A PDE-based model-free algorithm for Continuous-time Reinforcement Learning

Yuhua Zhu - University of California, Los Angeles

Aug 27, Benasque
X Partial differential equations, optimal design and numerics

The PDE problem we are interested in

$$ds_t = b(s_t)\,dt + \sigma(s_t)\,dB_t$$

The PDE problem we are interested in

$$ds_t = b(s_t)\,dt + \sigma(s_t)\,dB_t$$

The drift and diffusion terms are unknown

The PDE problem we are interested in

$$ds_t = b(s_t)\,dt + \sigma(s_t)\,dB_t$$

The drift and diffusion terms are unknown

Trajectory data $\{s_0^j, s_{\Delta t}^j, \cdots, s_{m\Delta t}^j\}_{j=1}^J$ is available

The PDE problem we are interested in

$$ds_t = b(s_t)\,dt + \sigma(s_t)\,dB_t$$

The drift and diffusion terms are unknown

Trajectory data $\{s_0^j, s_{\Delta t}^j, \cdots, s_{m\Delta t}^j\}_{j=1}^J$ is available

The goal is NOT to estimate the dynamics or predict the trajectory

The PDE problem we are interested in

$$ds_t = b(s_t)\,dt + \sigma(s_t)\,dB_t$$

The drift and diffusion terms are unknown

Trajectory data $\{s_0^j, s_{\Delta t}^j, \cdots, s_{m\Delta t}^j\}_{j=1}^J$ is available

The goal is NOT to estimate the dynamics or predict the trajectory

Goal:

Estimate a function $V(s)$ related to the underlying dynamics

$$\mathscr{L}_{b,\sigma} V(s) = 0$$

The PDE problem we are interested in

$$ds_t = b(s_t)\,dt + \sigma(s_t)\,dB_t$$

← Continuous-time

The drift and diffusion terms are unknown

Trajectory data $\{s_0^j, s_{\Delta t}^j, \cdots, s_{m\Delta t}^j\}_{j=1}^J$ is available

The goal is NOT to estimate the dynamics or predict the trajectory

Goal:

Estimate a function $V(s)$ related to the underlying dynamics

$$\mathscr{L}_{b,\sigma} V(s) = 0$$

The PDE problem we are interested in

$$ds_t = b(s_t)\,dt + \sigma(s_t)\,dB_t$$ ⟵ Continuous-time

The drift and diffusion terms are unknown

Trajectory data $\{s_0^j, s_{\Delta t}^j, \cdots, s_{m\Delta t}^j\}_{j=1}^J$ is available ⟵ Discrete-time

The goal is NOT to estimate the dynamics or predict the trajectory

Goal:

Estimate a function $V(s)$ related to the underlying dynamics

$$\mathscr{L}_{b,\sigma} V(s) = 0$$

Question: Why not solving an inverse problem?

Step 1. Solve an inverse problem

$$\{s_0^j, s_{\Delta t}^j, \cdots, s_{m\Delta t}^j\}_{j=1}^J \longrightarrow \hat{b}(s_t), \hat{\sigma}(s_t)$$

Step 2. Solve the PDE with estimated dynamics

$$\hat{b}(s_t), \hat{\sigma}(s_t) \longrightarrow \mathscr{L}_{\hat{b},\hat{\sigma}} \hat{V}(s) = 0$$

Question: Why not solving an inverse problem?

Step 1. Solve an inverse problem

$$\{s_0^j, s_{\Delta t}^j, \cdots, s_{m\Delta t}^j\}_{j=1}^J \longrightarrow \hat{b}(s_t), \hat{\sigma}(s_t)$$

Step 2. Solve the PDE with estimated dynamics

$$\hat{b}(s_t), \hat{\sigma}(s_t) \longrightarrow \mathscr{L}_{\hat{b},\hat{\sigma}} \hat{V}(s) = 0$$

- **Cumulative** error (Step 1 + Step 2)
- Computationally **expensive** to solve the inverse problem

$$- \sum_{i,j} L\left(p_{\hat{b},\hat{\sigma}}(s_{i\Delta t}^j, \Delta t), s_{(i+1)\Delta t}^j\right)$$

  - Especially when Δt is large or number of data are large

Mismatch between continuous-time dynamics and discrete-time data

Step 1. Solve an inverse problem

$$\{s_0^j, s_{\Delta t}^j, \cdots, s_{m\Delta t}^j\}_{j=1}^J \longrightarrow \|(\hat{b}, \hat{\sigma}) - (b, \sigma)\| \geq \min_{(\hat{b},\hat{\sigma})\in\mathcal{F}} \|(\hat{b}, \hat{\sigma}) - (b, \sigma)\|$$

Mismatch between continuous-time dynamics and discrete-time data

Step 1. Solve an inverse problem

$$\{s_0^j, s_{\Delta t}^j, \cdots, s_{m\Delta t}^j\}_{j=1}^J \longrightarrow \|(\hat{b}, \hat{\sigma}) - (b, \sigma)\| \geq \min_{(\hat{b},\hat{\sigma})\in\mathcal{F}} \|(\hat{b}, \hat{\sigma}) - (b, \sigma)\|$$

Mismatch between continuous-time dynamics and discrete-time data

Step 1. Solve an inverse problem

$$\{s_0^j, s_{\Delta t}^j, \cdots, s_{m\Delta t}^j\}_{j=1}^J \longrightarrow \|(\hat{b}, \hat{\sigma}) - (b, \sigma)\| \geq \min_{(\hat{b},\hat{\sigma})\in\mathscr{F}} \|(\hat{b}, \hat{\sigma}) - (b, \sigma)\|$$

Mismatch between continuous-time dynamics and discrete-time data

Step 1. Solve an inverse problem

$$\{s_0^j, s_{\Delta t}^j, \cdots, s_{m\Delta t}^j\}_{j=1}^J \longrightarrow \|(\hat{b}, \hat{\sigma}) - (b, \sigma)\| \geq \min_{(\hat{b},\hat{\sigma})\in\mathscr{F}} \|(\hat{b}, \hat{\sigma}) - (b, \sigma)\|$$

- **Hard** to find a suitable functional space $\mathscr{F}$: $\mathscr{F}$ cannot be too small

Mismatch between continuous-time dynamics and discrete-time data

Step 1. Solve an inverse problem

$$\{s_0^j, s_{\Delta t}^j, \cdots, s_{m\Delta t}^j\}_{j=1}^J \longrightarrow \|(\hat{b}, \hat{\sigma}) - (b, \sigma)\| \geq \min_{(\hat{b},\hat{\sigma})\in\mathscr{F}} \|(\hat{b}, \hat{\sigma}) - (b, \sigma)\|$$

- **Hard** to find a suitable functional space $\mathscr{F}$: $\mathscr{F}$ cannot be too small or too large

Mismatch between continuous-time dynamics and discrete-time data

Step 1. Solve an inverse problem

$$\{s_0^j, s_{\Delta t}^j, \cdots, s_{m\Delta t}^j\}_{j=1}^J \longrightarrow \|(\hat{b}, \hat{\sigma}) - (b, \sigma)\| \geq \min_{(\hat{b},\hat{\sigma})\in\mathscr{F}} \|(\hat{b}, \hat{\sigma}) - (b, \sigma)\|$$

- **Hard** to find a suitable functional space $\mathscr{F}$: $\mathscr{F}$ cannot be too small or too large

$$\text{e.g. } \dot{s}_t = As_t, \quad s_t \in \mathbb{R}^2, \; A = \begin{bmatrix} 1, & 0 \\ 0, & 1 \end{bmatrix}$$

Mismatch between continuous-time dynamics and discrete-time data

Step 1. Solve an inverse problem

$$\{s_0^j, s_{\Delta t}^j, \cdots, s_{m\Delta t}^j\}_{j=1}^J \longrightarrow \|(\hat{b}, \hat{\sigma}) - (b, \sigma)\| \geq \min_{(\hat{b},\hat{\sigma})\in\mathscr{F}} \|(\hat{b}, \hat{\sigma}) - (b, \sigma)\|$$

- **Hard** to find a suitable functional space $\mathscr{F}$: $\mathscr{F}$ cannot be too small or too large

$$\text{e.g. } \dot{s}_t = As_t, \quad s_t \in \mathbb{R}^2, \; A = \begin{bmatrix} 1, & 0 \\ 0, & 1 \end{bmatrix}$$

The best one can do $s_{(i+1)\Delta t} = p_{\Delta t}(s_{i\Delta t})$

Mismatch between continuous-time dynamics and discrete-time data

Step 1. Solve an inverse problem

$$\{s_0^j, s_{\Delta t}^j, \cdots, s_{m\Delta t}^j\}_{j=1}^J \longrightarrow \|(\hat{b}, \hat{\sigma}) - (b, \sigma)\| \geq \min_{(\hat{b},\hat{\sigma})\in\mathscr{F}} \|(\hat{b}, \hat{\sigma}) - (b, \sigma)\|$$

- **Hard** to find a suitable functional space $\mathscr{F}$: $\mathscr{F}$ cannot be too small or too large

$$\text{e.g. } \dot{s}_t = As_t, \quad s_t \in \mathbb{R}^2, \; A = \begin{bmatrix} 1, & 0 \\ 0, & 1 \end{bmatrix}$$

The best one can do $s_{(i+1)\Delta t} = p_{\Delta t}(s_{i\Delta t}) \longrightarrow p_{\Delta t}(s) = e^{\Delta t} s$

Mismatch between continuous-time dynamics and discrete-time data

Step 1. Solve an inverse problem

$$\{s_0^j, s_{\Delta t}^j, \cdots, s_{m\Delta t}^j\}_{j=1}^J \longrightarrow \|(\hat{b}, \hat{\sigma}) - (b, \sigma)\| \geq \min_{(\hat{b},\hat{\sigma})\in\mathscr{F}} \|(\hat{b}, \hat{\sigma}) - (b, \sigma)\|$$

- **Hard** to find a suitable functional space $\mathscr{F}$: $\mathscr{F}$ cannot be too small or too large

$$\text{e.g. } \dot{s}_t = As_t, \quad s_t \in \mathbb{R}^2, \; A = \begin{bmatrix} 1, & 0 \\ 0, & 1 \end{bmatrix}$$

The best one can do $s_{(i+1)\Delta t} = p_{\Delta t}(s_{i\Delta t}) \longrightarrow p_{\Delta t}(s) = e^{\Delta t}s$ + $b(s)$ is linear

Mismatch between continuous-time dynamics and discrete-time data

Step 1. Solve an inverse problem

$$\{s_0^j, s_{\Delta t}^j, \cdots, s_{m\Delta t}^j\}_{j=1}^J \longrightarrow \|(\hat{b}, \hat{\sigma}) - (b, \sigma)\| \geq \min_{(\hat{b},\hat{\sigma})\in\mathscr{F}} \|(\hat{b}, \hat{\sigma}) - (b, \sigma)\|$$

- **Hard** to find a suitable functional space $\mathscr{F}$: $\mathscr{F}$ cannot be too small or too large

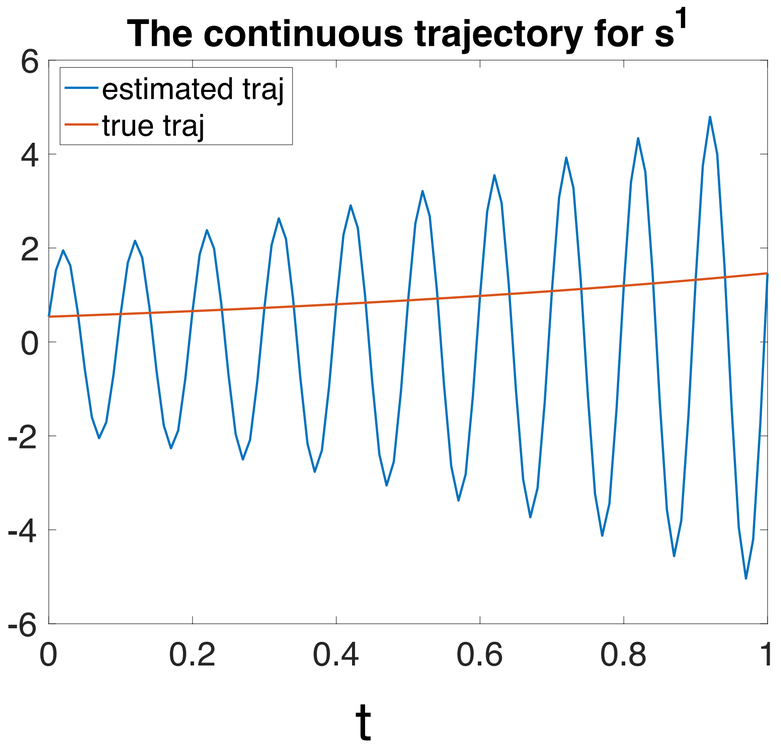
$$\text{e.g. } \dot{s}_t = As_t, \quad s_t \in \mathbb{R}^2, \; A = \begin{bmatrix} 1, & 0 \\ 0, & 1 \end{bmatrix}$$

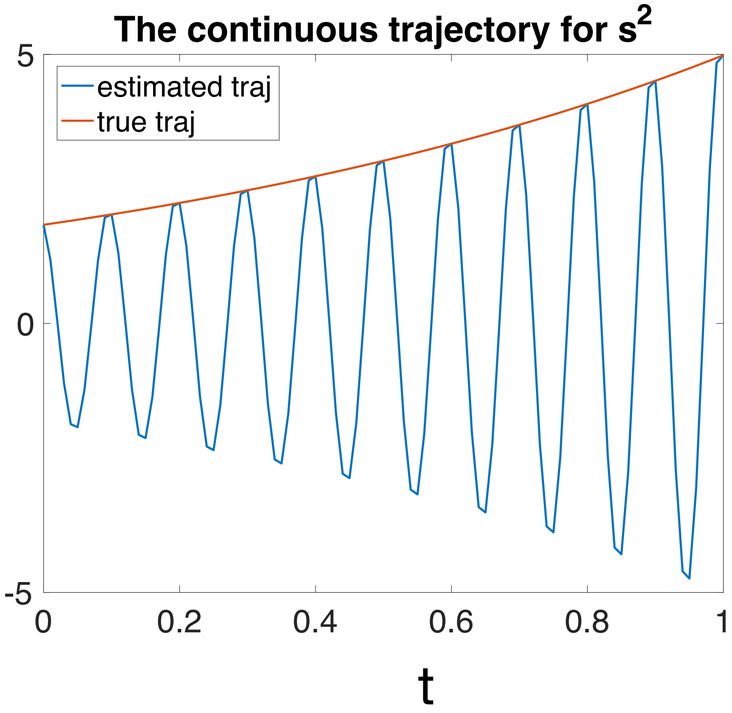
The best one can do $s_{(i+1)\Delta t} = p_{\Delta t}(s_{i\Delta t}) \longrightarrow p_{\Delta t}(s) = e^{\Delta t}s$ + $b(s)$ is linear

$$\hat{A} = \begin{bmatrix} 1, & \dfrac{2\pi}{\Delta t} \\ -\dfrac{2\pi}{\Delta t}, & 1 \end{bmatrix}$$

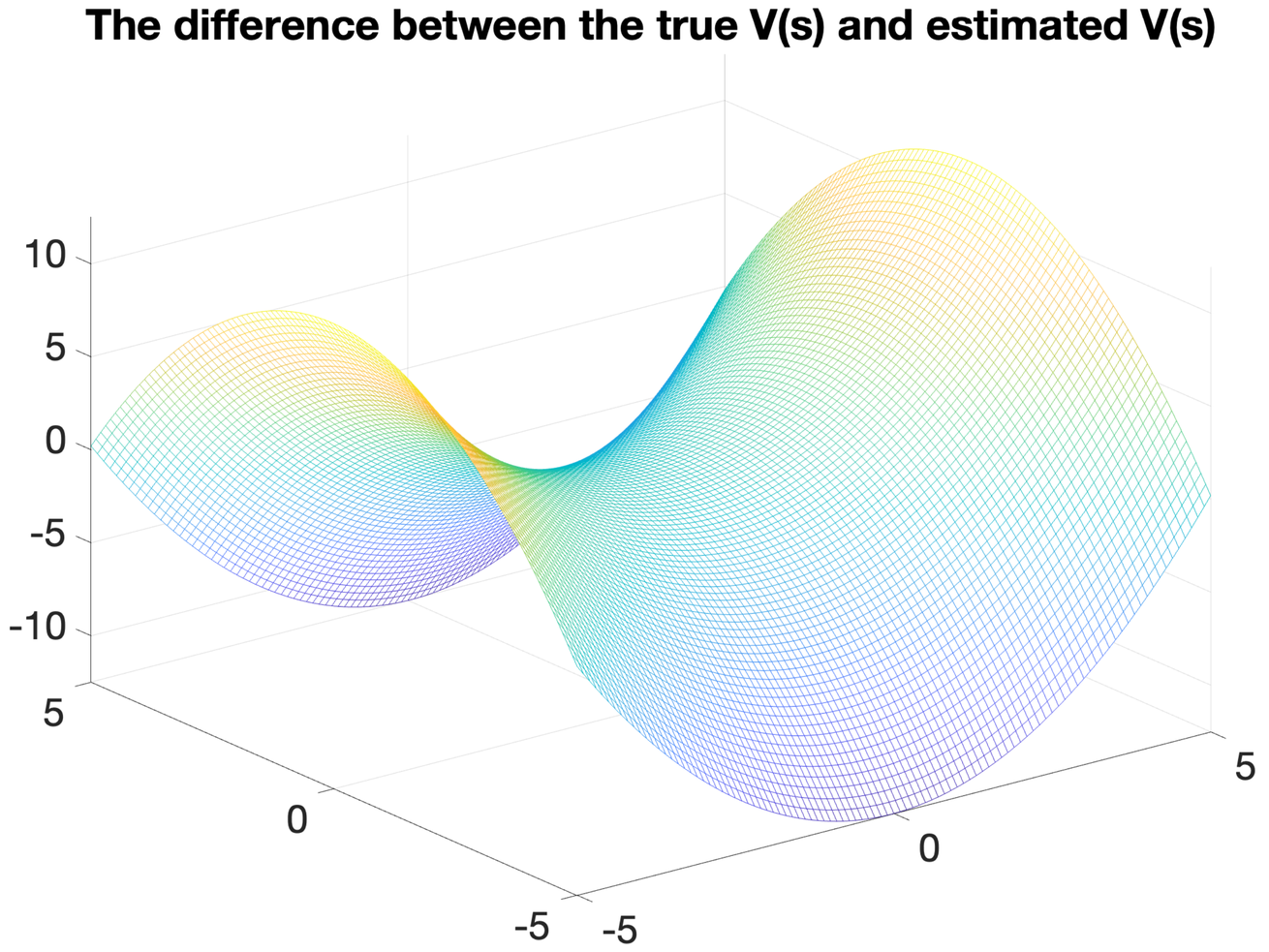
Mismatch between continuous-time dynamics and discrete-time data

Estimated trajectory is driven by $\hat{A}$; true trajectory is driven by A, $\Delta t = 0.1$



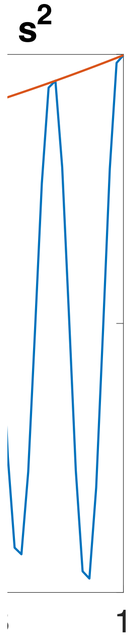


Mismatch between continuous-time dynamics and discrete-time data



I

$A, \Delta t = 0.1$



Model-based PDE formulation

Cons

- **Hard** to find a suitable functional space $\mathscr{F}$ for the dynamics
- **Cumulative** error (Step 1 + Step 1)
- Computationally **expensive** to solve the inverse problem
- An efficient method is **problem dependent**.

Model-based PDE formulation

Cons

- **Hard** to find a suitable functional space $\mathscr{F}$ for the dynamics
- **Cumulative** error (Step 1 + Step 1)
- Computationally **expensive** to solve the inverse problem
- An efficient method is **problem dependent**.

Pros

- Keep the **PDE form** to estimate $V(s)$
  - Model error small, then $\hat{V}(s)$ is close to $V(s)$
  - Well-developed PDE analysis tool
  - Keep the continuous-time structure

Model-based PDE formulation

Cons

- **Hard** to find a suitable functional space $\mathscr{F}$ for the dynamics
- **Cumulative** error (Step 1 + Step 1)
- Computationally **expensive** to solve the inverse problem
- An efficient method is **problem dependent**.

Pros

- Keep the **PDE form** to estimate $V(s)$
  - Model error small, then $\hat{V}(s)$ is close to $V(s)$
  - Well-developed PDE analysis tool
  - Keep the continuous-time structure

Can we skip the inverse problem and directly estimate $V(s)$?

Schedule

Problem Setting

Bellman equation ——Why it is not good

A PDE-based Bellman equation —— Why it is better

Algorithm

Continuous-time RL

Discrete-time RL (Markov Decision Process)

Continuous-time RL

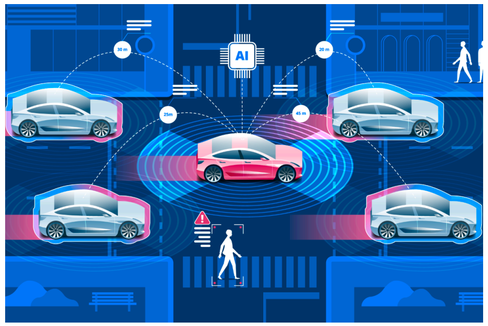
Discrete-time RL (Markov Decision Process)

Continuous-time RL (Stochastic optimal control with unknown dynamics)

Continuous-time RL

Discrete-time RL (Markov Decision Process)

Continuous-time RL (Stochastic optimal control with unknown dynamics)



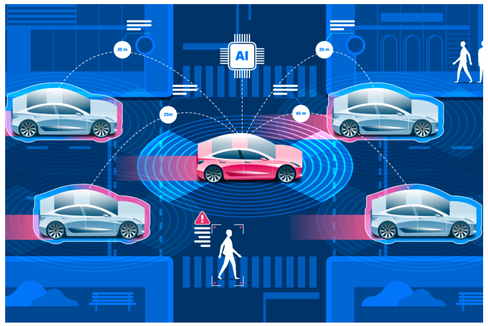
Autonomous vehicle

Robotics

Continuous-time RL

Discrete-time RL (Markov Decision Process)

Continuous-time RL (Stochastic optimal control with unknown dynamics)



Autonomous vehicle

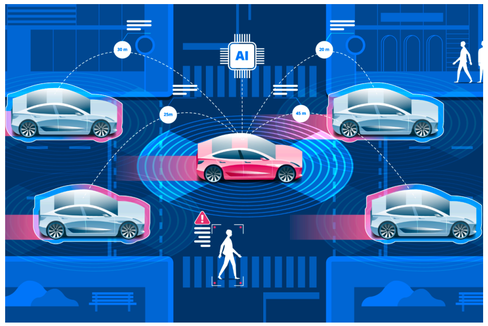
Robotics

Financial Market

Continuous-time RL

Discrete-time RL (Markov Decision Process)

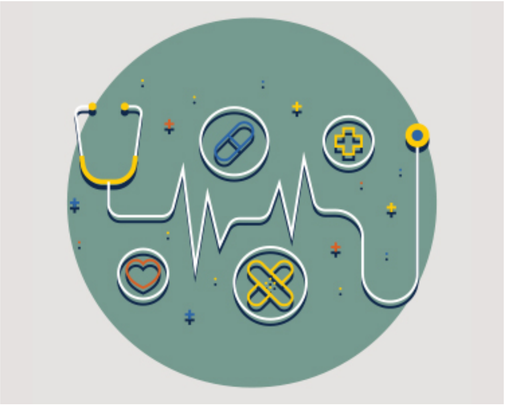
Continuous-time RL (Stochastic optimal control with unknown dynamics)



Autonomous vehicle

Robotics

Financial Market



Dynamic Treatment

Question: Why bother studying continuous-time RL?

Discrete-time data  why not treat it as a discrete-time RL?

Question: Why bother studying continuous-time RL?

Discrete-time data → why not treat it as a discrete-time RL?

Pros

- Model-free
  - One can skip the inverse problem and directly compute $V(s)$
- Many well-developed RL algorithms

Question: Why bother studying continuous-time RL?

Discrete-time data → why not treat it as a discrete-time RL?

Cons

Pros

- Model-free
  - One can skip the inverse problem and directly compute $V(s)$
- Many well-developed RL algorithms

Reinforcement Learning

Policy Evaluation

Policy Improvement

Reinforcement Learning

Policy Evaluation

Given a policy $a \sim \pi(a \mid s)$

Policy Improvement

Reinforcement Learning

Policy Evaluation

Given a policy $a \sim \pi(a \mid s)$  $V^{\pi}(s)$: Measures how good the policy π is

Policy Improvement

Reinforcement Learning

Policy Evaluation

Given a policy $a \sim \pi(a \mid s)$  $V^{\pi}(s)$: Measures how good the policy π is

Policy Improvement

Update the policy using gradient ascent

Reinforcement Learning

Policy Evaluation

Given a policy $a \sim \pi(a \mid s)$ → $V^{\pi}(s)$: Measures how good the policy π is

Policy Improvement

Update the policy using gradient ascent → $\pi_{k+1} = \pi_k + \eta \nabla_{\pi} V^{\pi_k}$

Reinforcement Learning

This talk

Policy Evaluation

Given a policy $a \sim \pi(a \mid s)$ → $V^{\pi}(s)$: Measures how good the policy π is

Policy Improvement

Update the policy using gradient ascent → $\pi_{k+1} = \pi_k + \eta \nabla_{\pi} V^{\pi_k}$

Policy Evaluation

Given a policy $a \sim \pi(a \mid s)$  $V^{\pi}(s)$: Measures how good the policy π is

Policy Evaluation

Given a policy $a \sim \pi(a \mid s)$ → $V^{\pi}(s)$: Measures how good the policy π is

Discrete-time RL (a.k.a. Markov Decision Process)

Policy Evaluation

Given a policy $a \sim \pi(a \mid s)$ → $V^{\pi}(s)$: Measures how good the policy π is

Discrete-time RL (a.k.a. Markov Decision Process)

s_0 ,

Policy Evaluation

Given a policy $a \sim \pi(a \mid s)$ → $V^{\pi}(s)$: Measures how good the policy π is

Discrete-time RL (a.k.a. Markov Decision Process)

s_0 , $a_0 \sim \pi(a \mid s_0)$

Policy Evaluation

Given a policy $a \sim \pi(a \mid s)$ → $V^{\pi}(s)$: Measures how good the policy π is

Discrete-time RL (a.k.a. Markov Decision Process)

s_0 , $a_0 \sim \pi(a \mid s_0) \longrightarrow \quad s_1$,

Policy Evaluation

Given a policy $a \sim \pi(a \mid s)$ → $V^{\pi}(s)$: Measures how good the policy π is

Discrete-time RL (a.k.a. Markov Decision Process)

$$s_0\ ,\ a_0 \sim \pi(a \mid s_0) \longrightarrow \quad s_1,\quad a_1 \sim \pi(a \mid s_1)$$

Policy Evaluation

Given a policy $a \sim \pi(a \mid s)$  $V^{\pi}(s)$: Measures how good the policy π is

Discrete-time RL (a.k.a. Markov Decision Process)

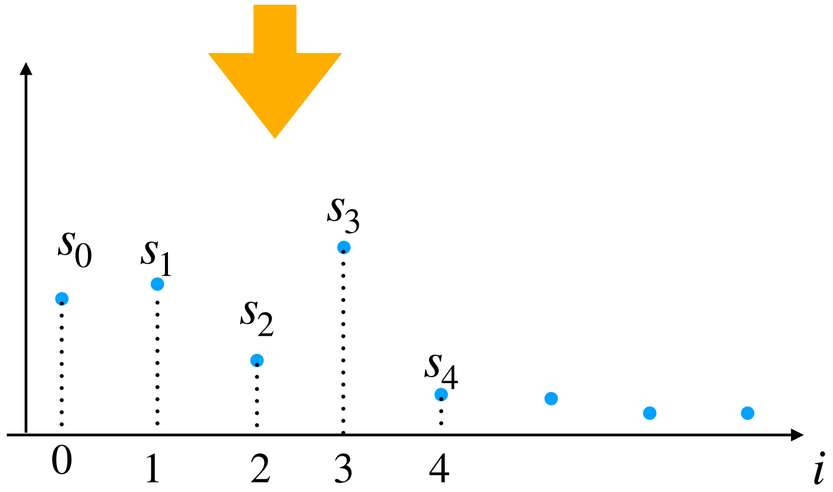
$$s_0\ ,\ a_0 \sim \pi(a \mid s_0) \longrightarrow \quad s_1,\quad a_1 \sim \pi(a \mid s_1) \longrightarrow \ \ldots\ldots$$

Policy Evaluation

Given a policy $a \sim \pi(a \mid s)$ → $V^{\pi}(s)$: Measures how good the policy π is

Discrete-time RL (a.k.a. Markov Decision Process)

$$s_0\ ,\ a_0 \sim \pi(a \mid s_0) \longrightarrow \quad s_1,\quad a_1 \sim \pi(a \mid s_1) \longrightarrow \ \ldots\ldots$$

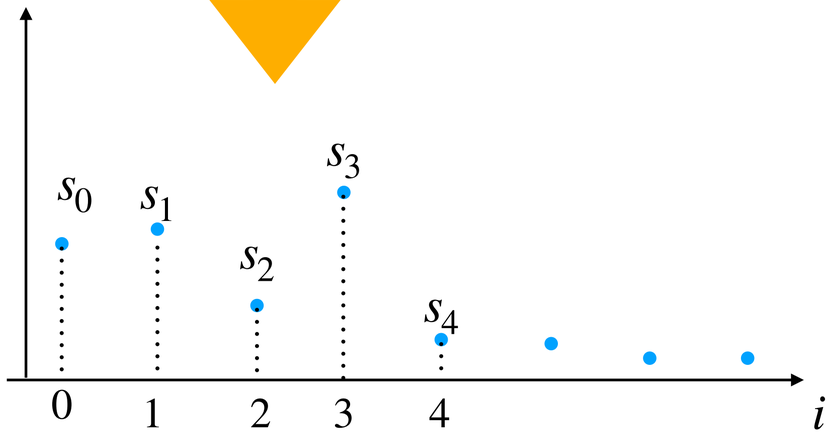


Policy Evaluation

Given a policy $a \sim \pi(a \mid s)$ → $V^{\pi}(s)$: Measures how good the policy π is

Discrete-time RL (a.k.a. Markov Decision Process)

s_0 , $a_0 \sim \pi(a \mid s_0) \longrightarrow$ s_1, $a_1 \sim \pi(a \mid s_1) \longrightarrow \ldots\ldots$

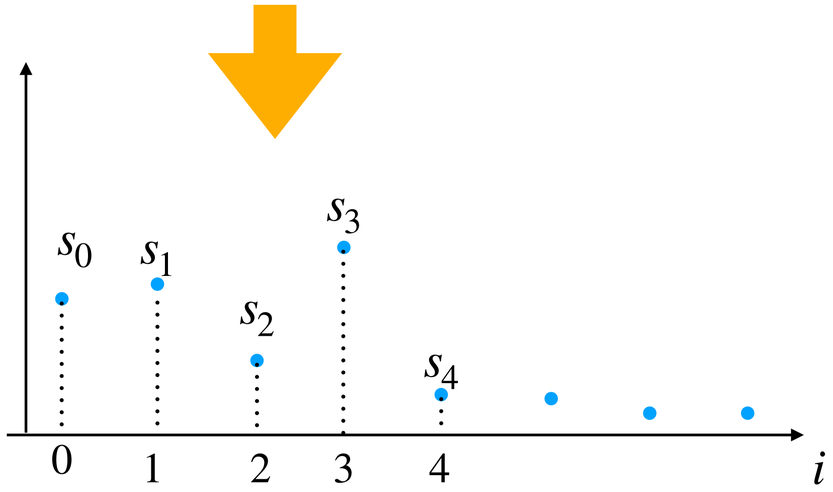


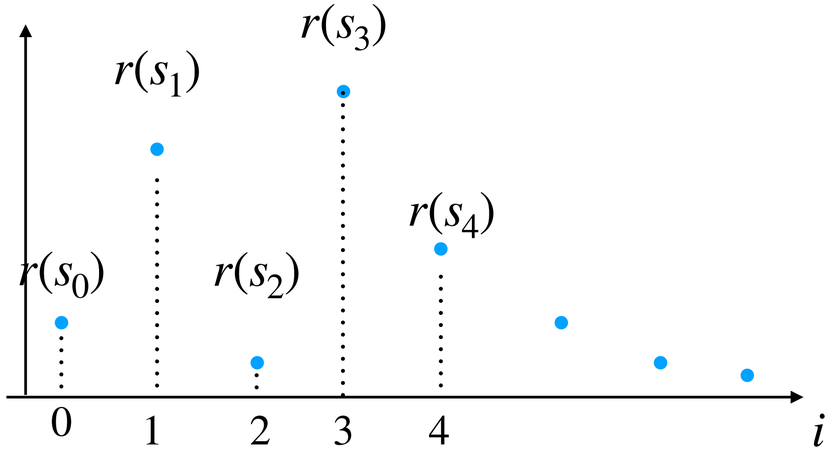
Policy Evaluation

Given a policy $a \sim \pi(a \mid s)$ → $V^{\pi}(s)$: Measures how good the policy π is

Discrete-time RL (a.k.a. Markov Decision Process)

$$s_0\,,\ a_0 \sim \pi(a \mid s_0) \longrightarrow \quad s_1,\quad a_1 \sim \pi(a \mid s_1) \longrightarrow \ \ldots\ldots$$



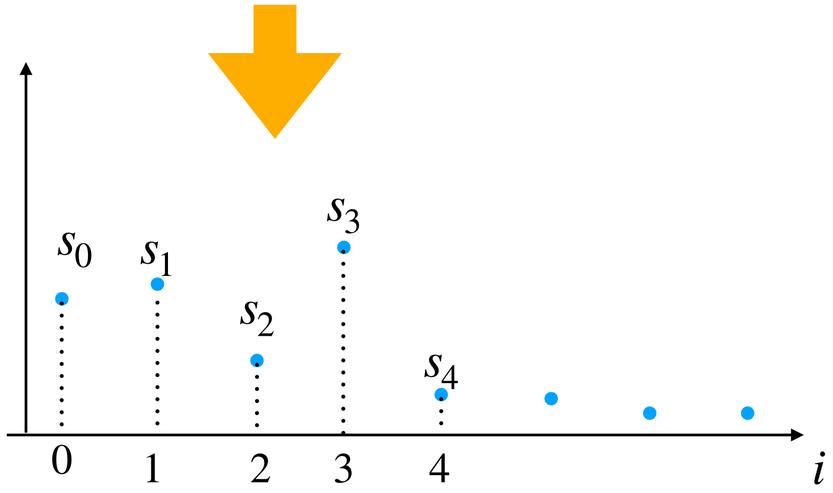


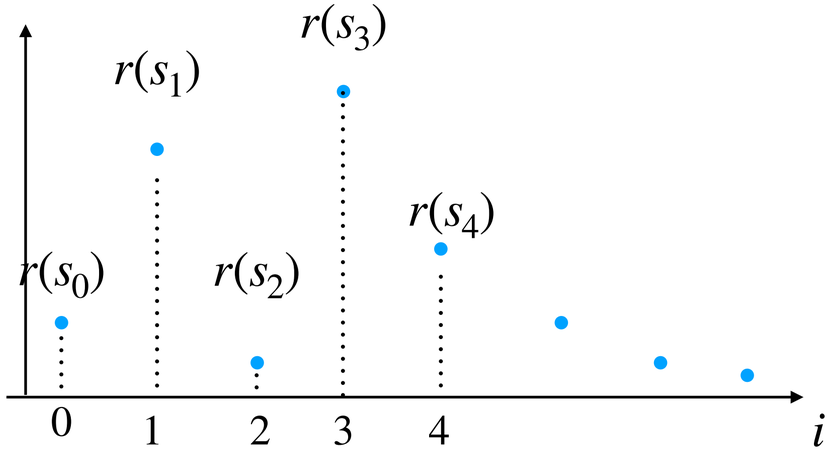
Policy Evaluation

Given a policy $a \sim \pi(a \mid s)$ → $V^{\pi}(s)$: Measures how good the policy π is

Discrete-time RL (a.k.a. Markov Decision Process)

$$s_0\,,\ a_0 \sim \pi(a \mid s_0) \longrightarrow \quad s_1,\quad a_1 \sim \pi(a \mid s_1) \longrightarrow \ \ldots\ldots \quad V^{\pi}(s) = \mathbb{E}\left[\sum_{i=0}^{\infty} \gamma^i r(s_i) \mid s_0 = s\right]$$



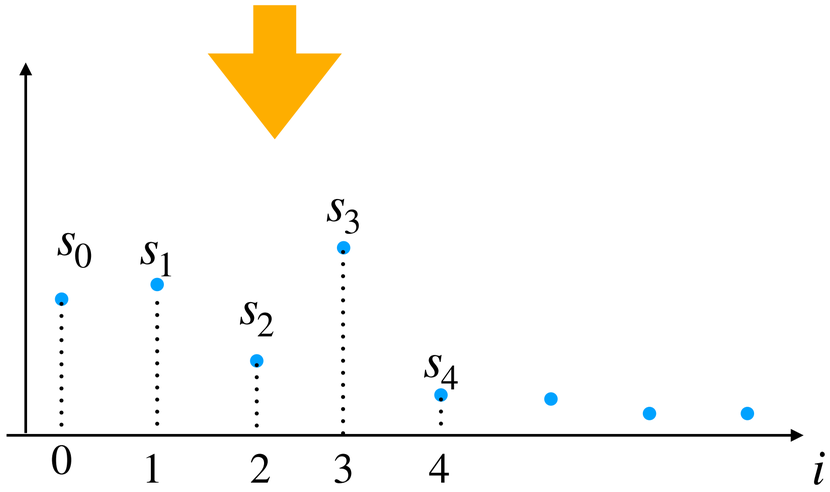


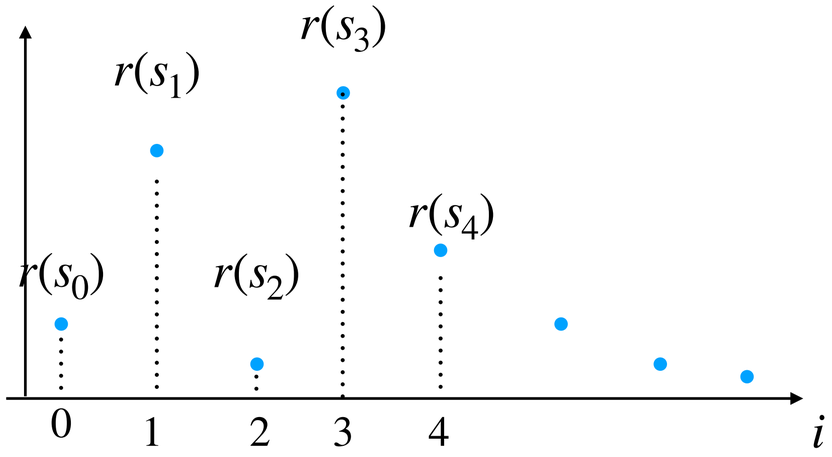
Policy Evaluation

Given a policy $a \sim \pi(a \mid s)$ → $V^{\pi}(s)$: Measures how good the policy π is

Continuous-time RL

$s_0\ ,\ a_0 \sim \pi(a \mid s_0) \longrightarrow \quad s_1,\quad a_1 \sim \pi(a \mid s_1) \longrightarrow \ \ldots\ldots$



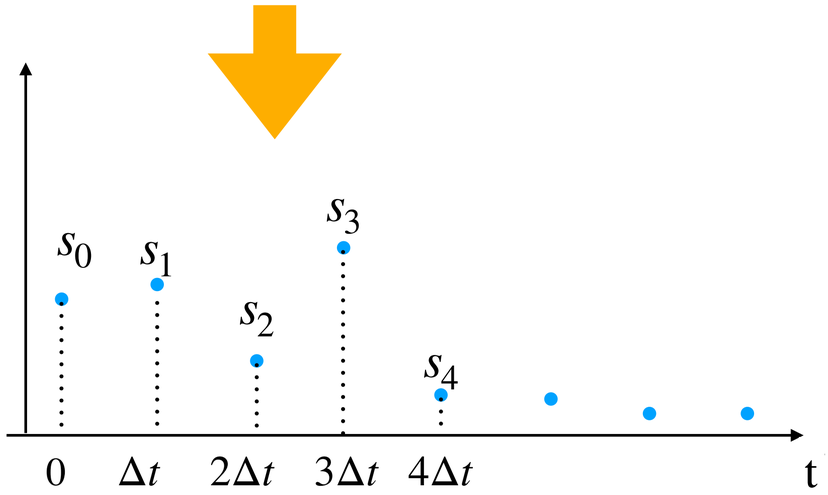


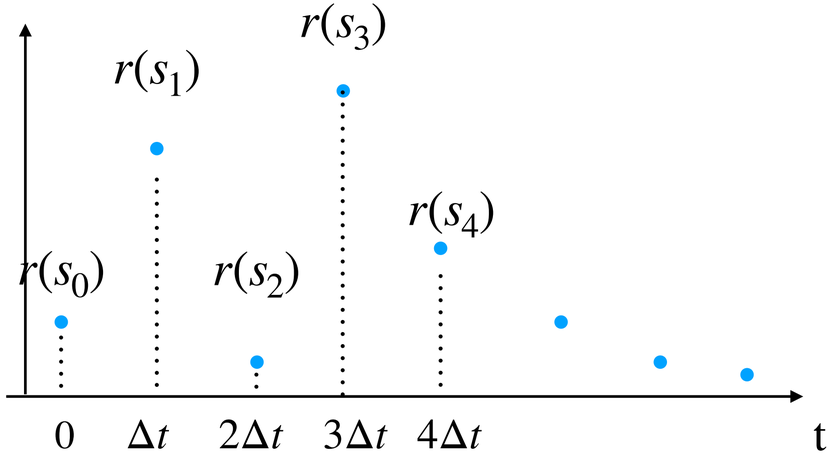
Policy Evaluation

Given a policy $a \sim \pi(a \mid s)$ → $V^{\pi}(s)$: Measures how good the policy π is

Continuous-time RL

$s_0\ ,\ a_0 \sim \pi(a \mid s_0) \longrightarrow \quad s_1,\quad a_1 \sim \pi(a \mid s_1) \longrightarrow \ \ldots\ldots$



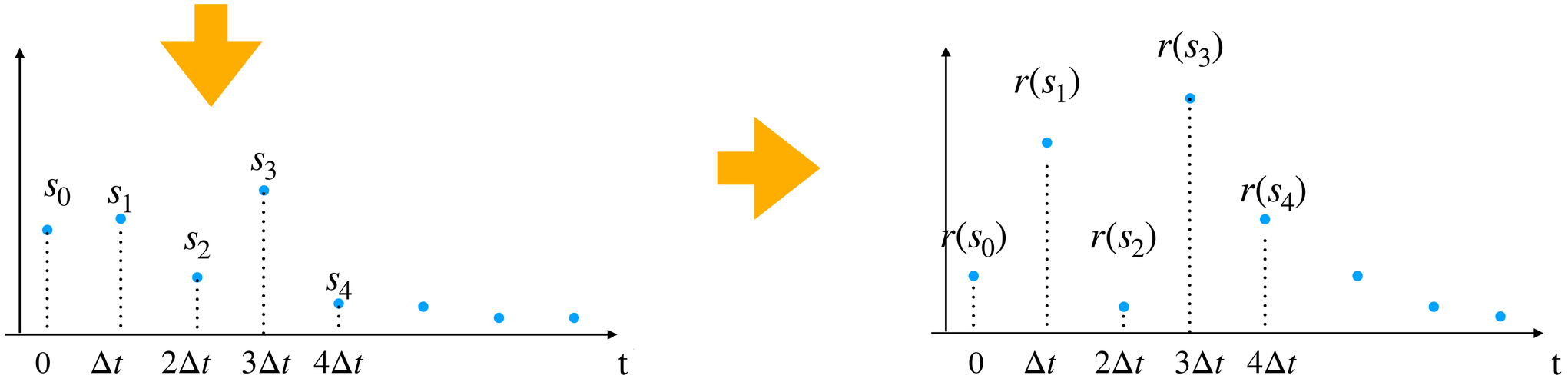


Policy Evaluation

Given a policy $a \sim \pi(a \mid s)$ → $V^{\pi}(s)$: Measures how good the policy π is

Continuous-time RL

$s_0\,,\ a_0 \sim \pi(a \mid s_0) \longrightarrow \quad s_1,\quad a_1 \sim \pi(a \mid s_1) \longrightarrow \ \ldots\ldots$



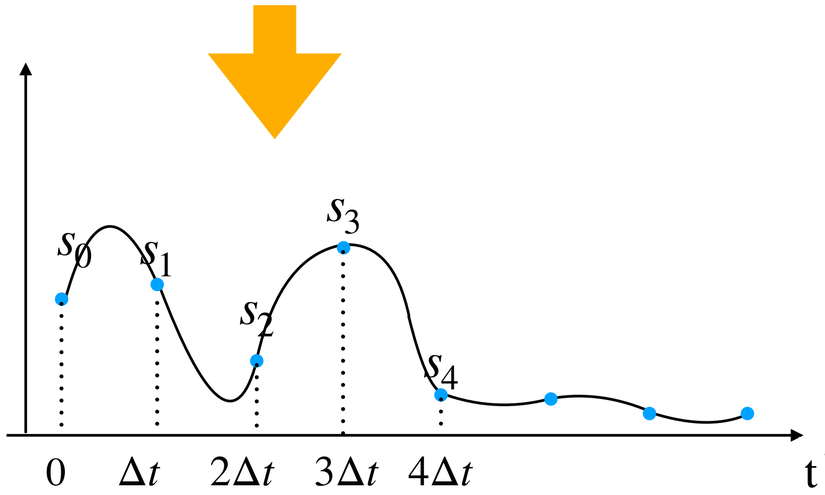
The underlying dynamics: $ds_t = b^{\pi}(s_t)\,dt + \sigma^{\pi}(s_t)\,dB_t$

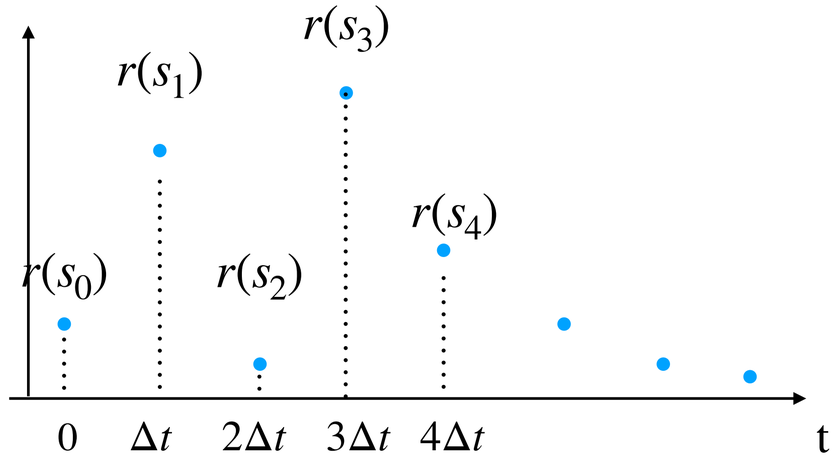
Policy Evaluation

Given a policy $a \sim \pi(a \mid s)$  $V^{\pi}(s)$: Measures how good the policy π is

Continuous-time RL

$s_0\ ,\ a_0 \sim \pi(a \mid s_0) \longrightarrow \quad s_1,\quad a_1 \sim \pi(a \mid s_1) \longrightarrow \ \ldots\ldots$





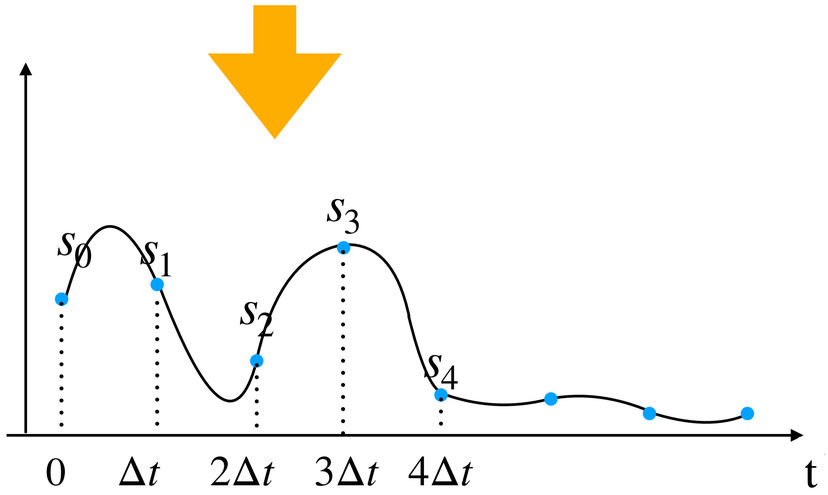
The underlying dynamics: $ds_t = b^{\pi}(s_t)\,dt + \sigma^{\pi}(s_t)\,dB_t$

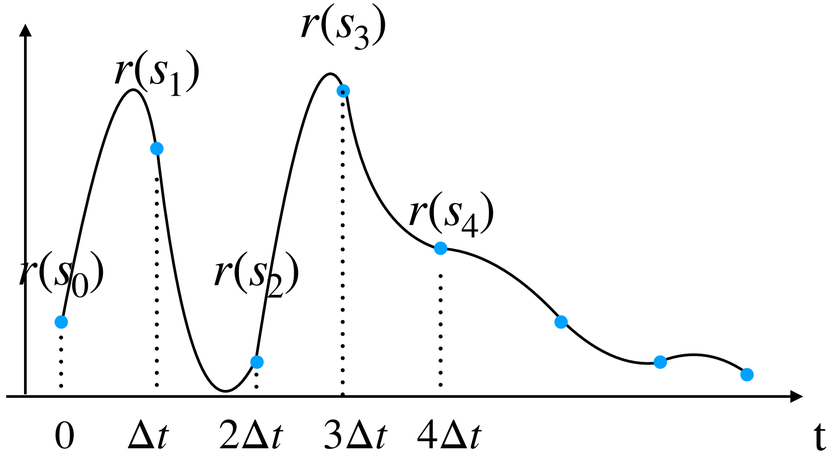
Policy Evaluation

Given a policy $a \sim \pi(a \mid s)$ → $V^{\pi}(s)$: Measures how good the policy π is

Continuous-time RL

$s_0\,,\ a_0 \sim \pi(a \mid s_0) \longrightarrow \quad s_1,\quad a_1 \sim \pi(a \mid s_1) \longrightarrow \ \ldots\ldots$





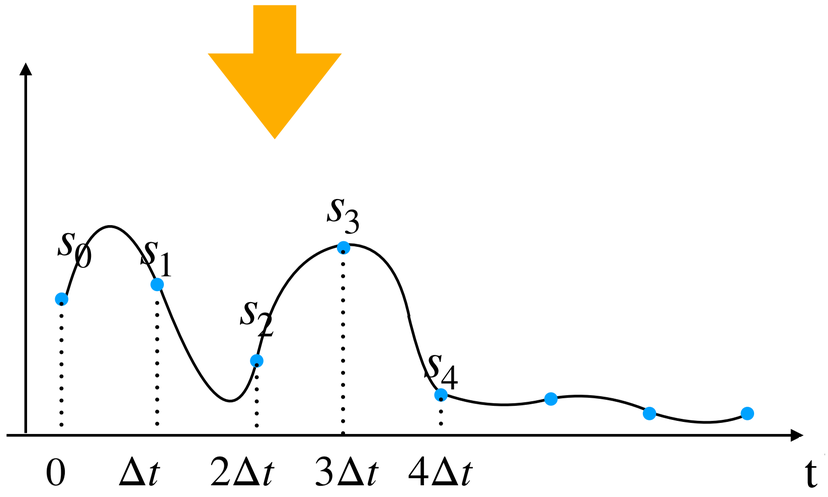
The underlying dynamics: $ds_t = b^{\pi}(s_t)\,dt + \sigma^{\pi}(s_t)\,dB_t$

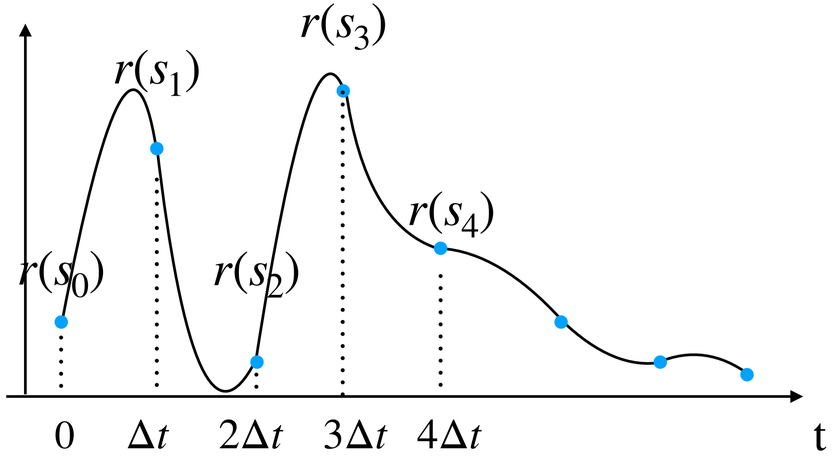
Policy Evaluation

Given a policy $a \sim \pi(a \mid s)$ → $V^{\pi}(s)$: Measures how good the policy π is

Continuous-time RL

$$s_0\,,\ a_0 \sim \pi(a \mid s_0) \longrightarrow \quad s_1,\quad a_1 \sim \pi(a \mid s_1) \longrightarrow \ldots\ldots V^{\pi}(s) = \mathbb{E}\left[\int_0^{\infty} e^{-\beta t} r(s_t)dt \mid s_0 = s\right]$$





The underlying dynamics: $ds_t = b^{\pi}(s_t)\,dt + \sigma^{\pi}(s_t)\,dB_t$

Policy Evaluation —— A PDE view

Goal:

Estimate a function $V(s)$ related to the underlying dynamics

$$\mathscr{L}_{b,\sigma} V(s) = 0$$

Policy Evaluation —— A PDE view

Goal:

Estimate a function $V(s)$ related to the underlying dynamics

$$\mathscr{L}_{b,\sigma} V(s) = 0$$

Policy Evaluation —— A PDE view

Goal:

Estimate a function $V(s)$ related to the underlying dynamics

$$\mathscr{L}_{b,\sigma} V(s) = 0$$

$$\beta V^{\pi}(s) = r^{\pi}(s) + b^{\pi}(s) \cdot \nabla V^{\pi}(s) + \frac{1}{2}\Sigma^{\pi}(s) : \nabla^2 V^{\pi}(s), \quad \Sigma = \sigma\sigma^{\top}$$

Policy Evaluation —— A PDE view

Goal:

Estimate a function $V(s)$ related to the underlying dynamics

$$\mathscr{L}_{b,\sigma} V(s) = 0$$

$$\beta V^{\pi}(s) = r^{\pi}(s) + b^{\pi}(s) \cdot \nabla V^{\pi}(s) + \frac{1}{2} \Sigma^{\pi}(s) : \nabla^2 V^{\pi}(s), \quad \Sigma = \sigma\sigma^{\top}$$

The underlying dynamics is unknown

Policy Evaluation —— A PDE view

Goal:

Estimate a function $V(s)$ related to the underlying dynamics

$$\mathscr{L}_{b,\sigma} V(s) = 0$$

$$\beta V^{\pi}(s) = r^{\pi}(s) + b^{\pi}(s) \cdot \nabla V^{\pi}(s) + \frac{1}{2} \Sigma^{\pi}(s) : \nabla^2 V^{\pi}(s), \quad \Sigma = \sigma\sigma^{\top}$$

The underlying dynamics is unknown

Only discrete-time information is given

Schedule

Continuos-time RL —— Setting

Model-free RL method —— Why not the optimal, either

A PDE-based Bellman equation —— Why it is better

Algorithm

Model-free RL method

$$V(s) = \mathbb{E}\left[\int_0^\infty e^{-\beta t} r(s_t) dt \,|\, s_0 \Big| = s\right] \longrightarrow \tilde{V}(s) = \mathbb{E}\left[\sum_{i=0}^{\infty} e^{-\beta \Delta t i} r(s_{\Delta t i}) \Delta t \,|\, s_0 \Big| = s\right]$$

Model-free RL method

$$V(s) = \mathbb{E}\left[\int_0^\infty e^{-\beta t} r(s_t) dt \,|\, s_0 \Big| = s\right] \longrightarrow \tilde{V}(s) = \mathbb{E}\left[\sum_{i=0}^{\infty} e^{-\beta \Delta t i} r(s_{\Delta t i}) \Delta t \,|\, s_0 \Big| = s\right]$$

$\longrightarrow$ $\tilde{V}(s)$ satisfies the discretized Bellman Equation (BE):

$$\tilde{V}(s) = \tilde{r}(s) + \gamma \mathbb{E}\left[\tilde{V}(s_{t+\Delta t}) \,|\, s_t = s\right] \quad \gamma = e^{-\beta \Delta t}, \;\; \tilde{r}(s) = r(s)\Delta t \,.$$

Model-free RL method

$$V(s) = \mathbb{E}\left[\int_0^\infty e^{-\beta t} r(s_t) dt \,|\, s_0 \Big| = s\right] \longrightarrow \tilde{V}(s) = \mathbb{E}\left[\sum_{i=0}^{\infty} e^{-\beta \Delta t i} r(s_{\Delta t i}) \Delta t \,|\, s_0 \Big| = s\right]$$

$\longrightarrow$ $\tilde{V}(s)$ satisfies the discretized Bellman Equation (BE):

$$\tilde{V}(s) = \tilde{r}(s) + \gamma \mathbb{E}\left[\tilde{V}(s_{t+\Delta t}) \,|\, s_t = s\right] \quad \gamma = e^{-\beta \Delta t}, \;\; \tilde{r}(s) = r(s)\Delta t\,.$$

Two important features about BE

Model-free RL method

$$V(s) = \mathbb{E}\left[\int_0^\infty e^{-\beta t} r(s_t) dt \,|\, s_0 \Big| = s\right] \longrightarrow \tilde{V}(s) = \mathbb{E}\left[\sum_{i=0}^{\infty} e^{-\beta \Delta t i} r(s_{\Delta t i}) \Delta t \,|\, s_0 \Big| = s\right]$$

$\longrightarrow$ $\tilde{V}(s)$ satisfies the discretized Bellman Equation (BE):

$$\tilde{V}(s) = \tilde{r}(s) + \gamma \mathbb{E}\left[\tilde{V}(s_{t+\Delta t}) \,|\, s_t = s\right] \quad \gamma = e^{-\beta \Delta t}, \;\; \tilde{r}(s) = r(s)\Delta t\,.$$

Two important features about BE

- The dynamics information is hidden in the expectation
  - The BE is the same for different dynamics $(b(s), \sigma(s))$

Model-free RL method

$$V(s) = \mathbb{E}\left[\int_0^\infty e^{-\beta t} r(s_t) dt \,|\, s_0 \Big| = s\right] \longrightarrow \tilde{V}(s) = \mathbb{E}\left[\sum_{i=0}^{\infty} e^{-\beta \Delta t i} r(s_{\Delta t i}) \Delta t \,|\, s_0 \Big| = s\right]$$

$\longrightarrow$ $\tilde{V}(s)$ satisfies the discretized Bellman Equation (BE):

$$\tilde{V}(s) = \tilde{r}(s) + \gamma \mathbb{E}\left[\tilde{V}(s_{t+\Delta t}) \,|\, s_t = s\right] \quad \gamma = e^{-\beta \Delta t}, \;\; \tilde{r}(s) = r(s)\Delta t\,.$$

$$s_{t+\Delta t} \sim \rho_{b,\sigma}(s', \Delta t \,|\, s_t)$$

Two important features about BE

- The dynamics information is hidden in the expectation
  - The BE is the same for different dynamics $(b(s), \sigma(s))$

Model-free RL method

$$V(s) = \mathbb{E}\left[\int_0^\infty e^{-\beta t} r(s_t) dt \,|\, s_0 \Big| = s\right] \longrightarrow \tilde{V}(s) = \mathbb{E}\left[\sum_{i=0}^{\infty} e^{-\beta \Delta t i} r(s_{\Delta t i}) \Delta t \,|\, s_0 \Big| = s\right]$$

$\longrightarrow$ $\tilde{V}(s)$ satisfies the discretized Bellman Equation (BE):

$$\tilde{V}(s) = \tilde{r}(s) + \gamma \mathbb{E}\left[\tilde{V}(s_{t+\Delta t}) \,|\, s_t = s\right] \quad \gamma = e^{-\beta \Delta t}, \;\; \tilde{r}(s) = r(s)\Delta t\,.$$

$$s_{t+\Delta t} \sim \rho_{b,\sigma}(s', \Delta t \,|\, s_t)$$

Two important features about BE

- The dynamics information is hidden in the expectation
  - The BE is the same for different dynamics $(b(s), \sigma(s))$
- Only depends on the current state and next state
  - Easy to plug in data directly

Least Square Temporal Difference

Least Square Temporal Difference

$$V(s) = \gamma \mathbb{E}[V(s_{\Delta t}) \,|\, s_0 = s] - \tilde{r}(s)$$

Least Square Temporal Difference

$$V(s) = \gamma \mathbb{E}[V(s_{\Delta t}) \,|\, s_0 = s] - \tilde{r}(s)$$

$$\tilde{V}_\theta(s) = \Phi(s)^\top \theta$$

Least Square Temporal Difference

$$V(s) = \gamma \mathbb{E}[V(s_{\Delta t}) \,|\, s_0 = s] - \tilde{r}(s)$$

$$\tilde{V}_\theta(s) = \Phi(s)^\top \theta$$

$$\langle \tilde{V}_\theta(s) = \gamma \mathbb{E}[\tilde{V}_\theta(s_{\Delta t}) \,|\, s_0 = s] - \tilde{r}(s), \Phi(s) \rangle$$

Least Square Temporal Difference

$$V(s) = \gamma \mathbb{E}[V(s_{\Delta t}) \,|\, s_0 = s] - \tilde{r}(s)$$

$$\tilde{V}_\theta(s) = \Phi(s)^\top \theta$$

$$\langle \tilde{V}_\theta(s) = \gamma \mathbb{E}[\tilde{V}_\theta(s_{\Delta t}) \,|\, s_0 = s] - \tilde{r}(s), \Phi(s) \rangle$$

LSTD in linear space

$$\tilde{A}\theta = \tilde{b}$$

$$\tilde{b} = \sum_{j=1}^{J} \sum_{i=0}^{m-1} \Phi(s^j_{i\Delta t}) r(s^j_{i\Delta t})$$

$$\tilde{A} = \frac{1}{\Delta t} \sum_{j=1}^{J} \sum_{i=0}^{m-1} \Phi(s^j_{i\Delta t}) \left(\Phi(s^j_{i\Delta t})^\top - \gamma \Phi(s^j_{(i+1)\Delta t})^\top \right)$$

Model-free RL method

Model-based

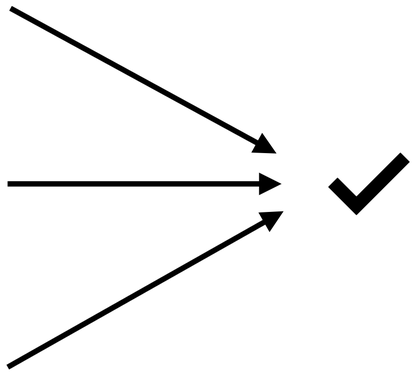
Model-free

- **Hard** to find an appropriate functional space $\mathcal{F}(\theta)$ for the dynamics
- **Cumulative** error (Step 1 + Step 1)
- Computationally **expensive** to solve the inverse problem
- The method is **problem dependent**.

Model-free RL method

Model-based

- **Hard** to find an appropriate functional space $\mathscr{F}(\theta)$ for the dynamics
- **Cumulative** error (Step 1 + Step 1)
- Computationally **expensive** to solve the inverse problem
- The method is **problem dependent**.



Model-free

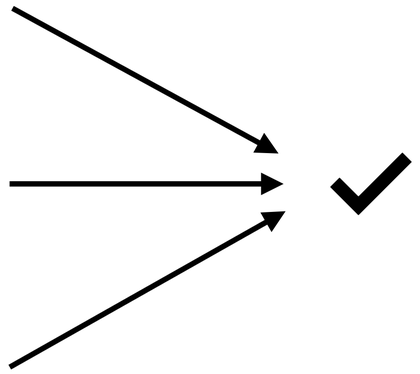
- No need to solve for the dynamics

Model-free RL method

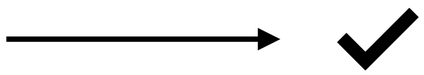
Model-based

- **Hard** to find an appropriate functional space $\mathscr{F}(\theta)$ for the dynamics
- **Cumulative** error (Step 1 + Step 1)
- Computationally **expensive** to solve the inverse problem
- The method is **problem dependent**.

Model-free



- No need to solve for the dynamics

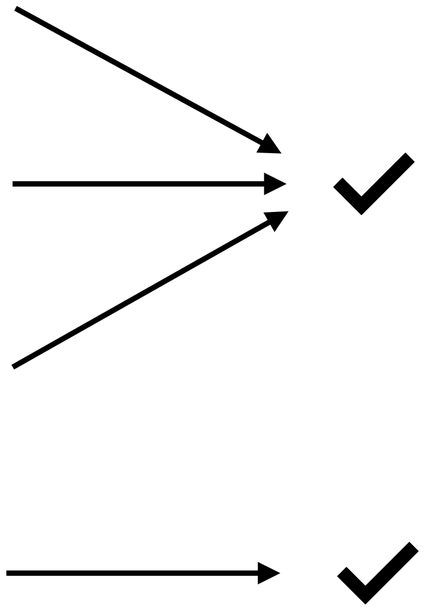


- The algorithm is the same for different dynamics

Model-free RL method

Model-based

- **Hard** to find an appropriate functional space $\mathscr{F}(\theta)$ for the dynamics
- **Cumulative** error (Step 1 + Step 1)
- Computationally **expensive** to solve the inverse problem
- The method is **problem dependent**.



Model-free

- No need to solve for the dynamics
- The algorithm is the same for different dynamics

Is it the optimal tool for continuous-time RL?

One example

One example

e.g.

Underlying dynamics: $\frac{d}{dt}s_t = \lambda s_t$, True value function: $V(s) = \cos^3(ks)$

One example

e.g. Underlying dynamics: $\frac{d}{dt}s_t = \lambda s_t$, True value function: $V(s) = \cos^3(ks)$

Approximate the value function linearly by finite bases:

$$\left\{ \frac{1}{\sqrt{2\pi}}, \frac{1}{\sqrt{\pi}}cos(ns_1), \frac{1}{\sqrt{\pi}}\sin(ns_1) \right\}_{n=1}^{N}$$

One example

e.g. Underlying dynamics: $\frac{d}{dt}s_t = \lambda s_t$, True value function: $V(s) = \cos^3(ks)$

Approximate the value function linearly by finite bases:

$$\left\{ \frac{1}{\sqrt{2\pi}}, \frac{1}{\sqrt{\pi}}cos(ns_1), \frac{1}{\sqrt{\pi}}\sin(ns_1) \right\}_{n=1}^{N}$$

Select N large enough s.t. $V(s)$ is in the finite space

One example

e.g. Underlying dynamics: $\frac{d}{dt}s_t = \lambda s_t$, True value function: $V(s) = \cos^3(ks)$

Approximate the value function linearly by finite bases:

$$\left\{ \frac{1}{\sqrt{2\pi}}, \frac{1}{\sqrt{\pi}}cos(ns_1), \frac{1}{\sqrt{\pi}}\sin(ns_1) \right\}_{n=1}^{N}$$

Select N large enough s.t. $V(s)$ is in the finite space

Discrete-time trajectory data: $\{s_0^j, s_{\Delta t}^j, \cdots, s_{I\Delta t}^j\}_{j=1}^{J}$

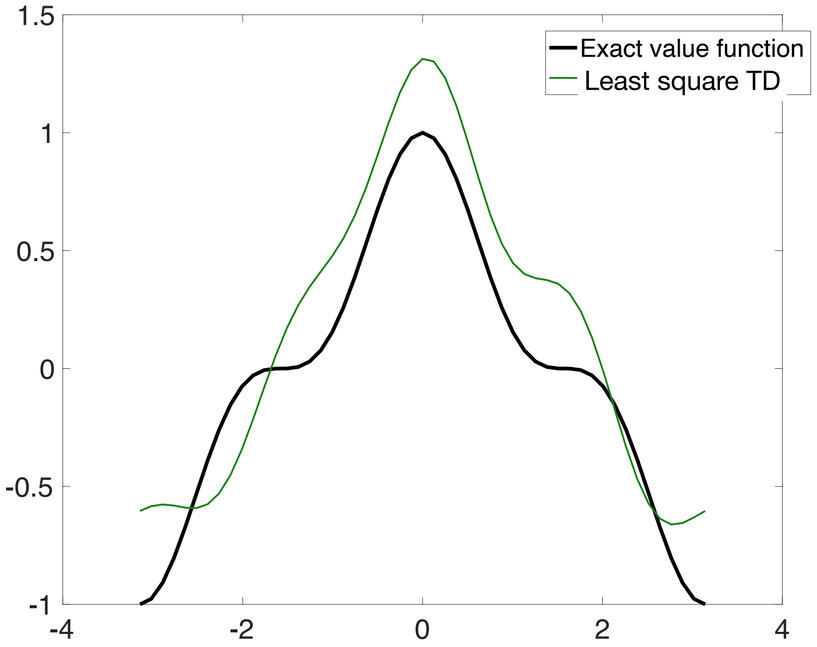
The performance of the RL algorithm

Underlying dynamics: $\frac{d}{dt}s_t = 0.05 s_t$

The performance of the RL algorithm

Underlying dynamics: $\frac{d}{dt}s_t = 0.05 s_t$

$\Delta t = 5, \beta = 0.1 (\gamma = 0.6), V(s) = \cos^3(s)$

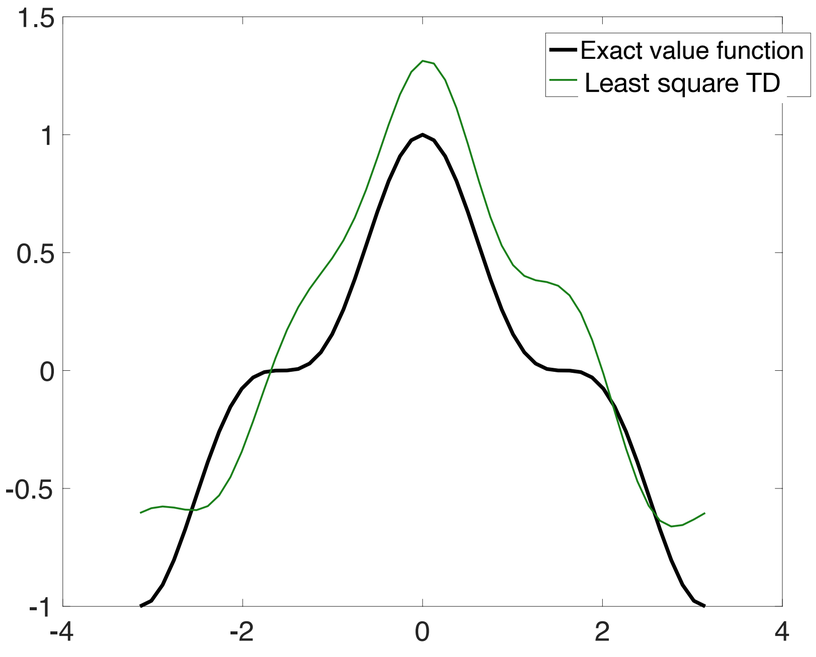


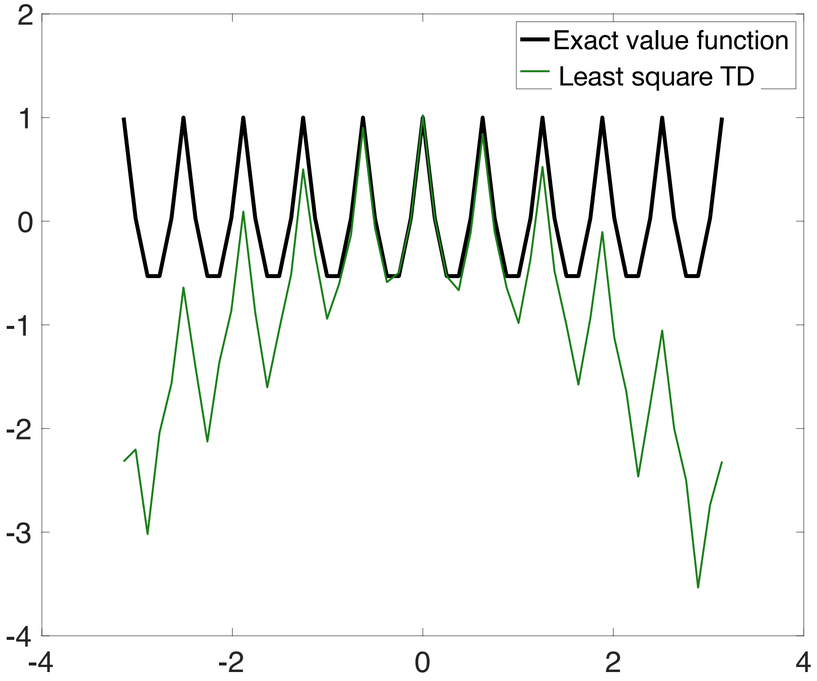
# of data: 40

The performance of the RL algorithm

Underlying dynamics: $\frac{d}{dt}s_t = 0.05 s_t$

$\Delta t = 5, \beta = 0.1 (\gamma = 0.6), V(s) = \cos^3(s)$ $\Delta t = 0.5, \beta = 0.1 (\gamma = 0.95), V(s) = \cos^3(10s)$





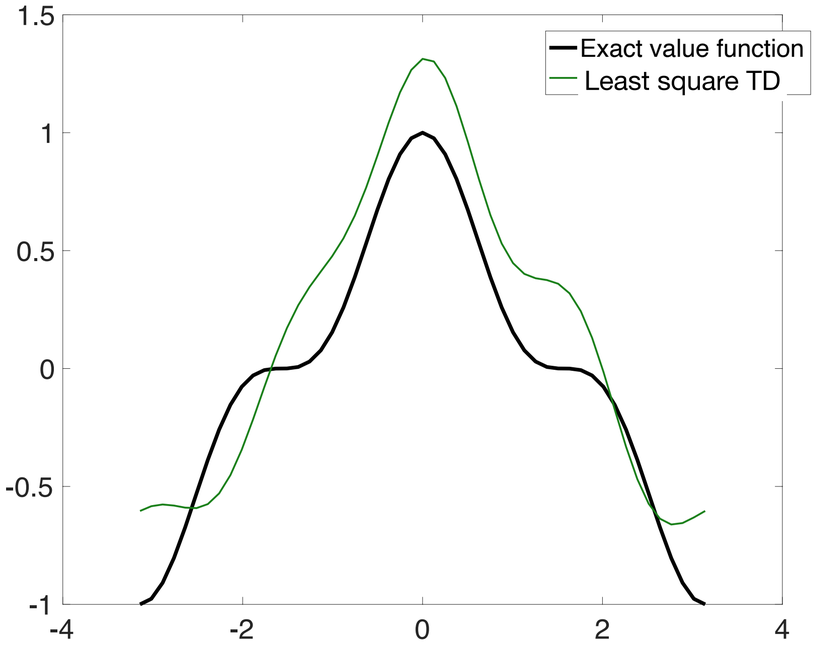
# of data: 40

# of data: 400

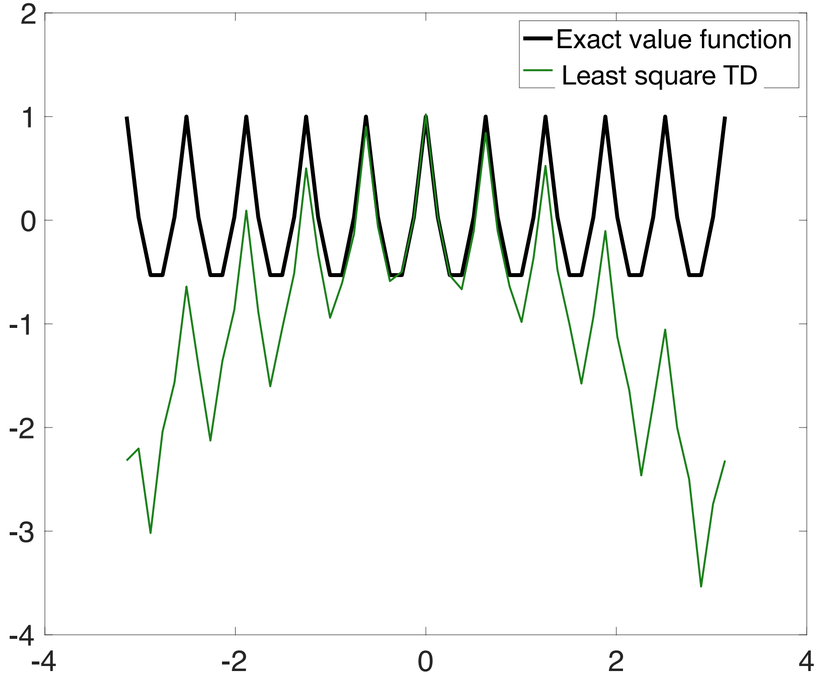
The performance of the RL algorithm

Underlying dynamics: $\frac{d}{dt}s_t = 0.05 s_t$

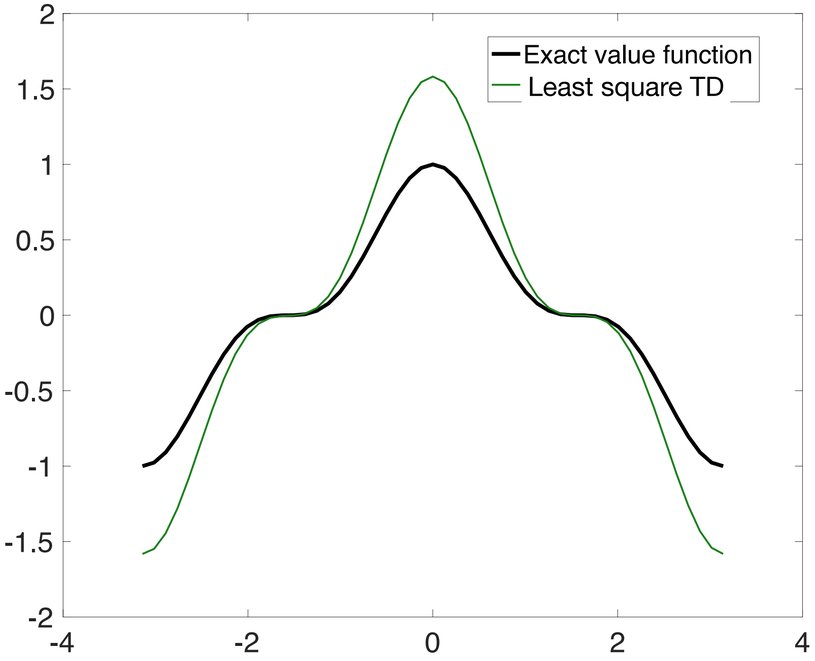
$\Delta t = 5, \beta = 0.1 (\gamma = 0.6), V(s) = \cos^3(s)$ $\Delta t = 0.5, \beta = 0.1 (\gamma = 0.95), V(s) = \cos^3(10s)$ $\Delta t = 0.1, \beta = 10 (\gamma = 0.37), V(s) = \cos^3(s)$



# of data: 40



# of data: 400



# of data: 40

Model-free RL algorithms

Cons

- It is not a good approximation for the continuous-time value function under certain circumstances

Pros

- Model-free
  - One can skip the inverse problem and directly compute $V(s)$
- Many well-developed RL algorithms

Model-free RL algorithms

Cons

- It is not a good approximation for the continuous-time value function under certain circumstances

Pros

- Model-free
  - One can skip the inverse problem and directly compute $V(s)$
- Many well-developed RL algorithms

Given the same information, can we do better than BE?

Schedule

Continuos-time RL —— Setting

Bellman equation ——Why it is not good

A PDE-based Model-free algorithm —— Why it is better

Algorithm

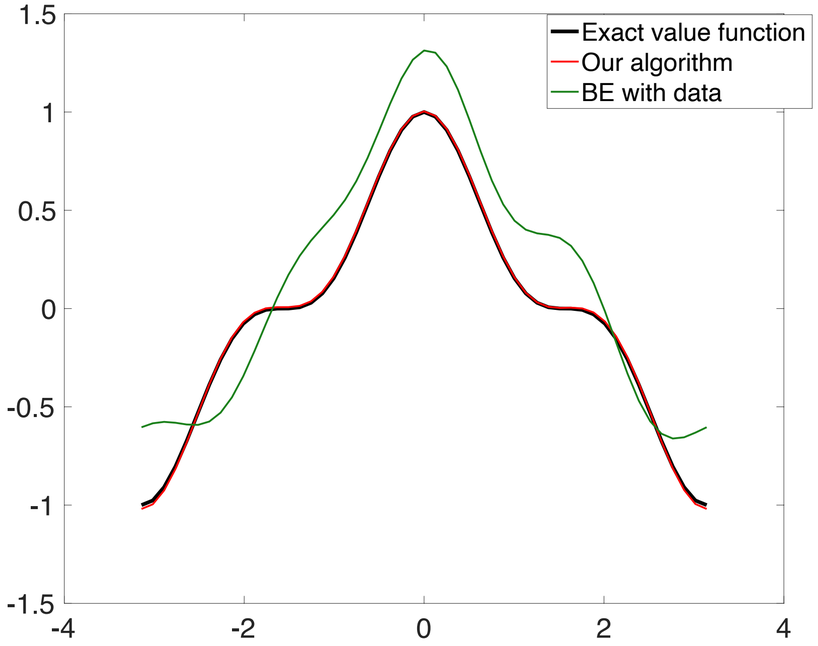
Given the same trajectory data: Our algorithm v.s. LSTD

Underlying dynamics: $\frac{d}{dt}s_t = 0.05 s_t$

Given the same trajectory data: Our algorithm v.s. LSTD

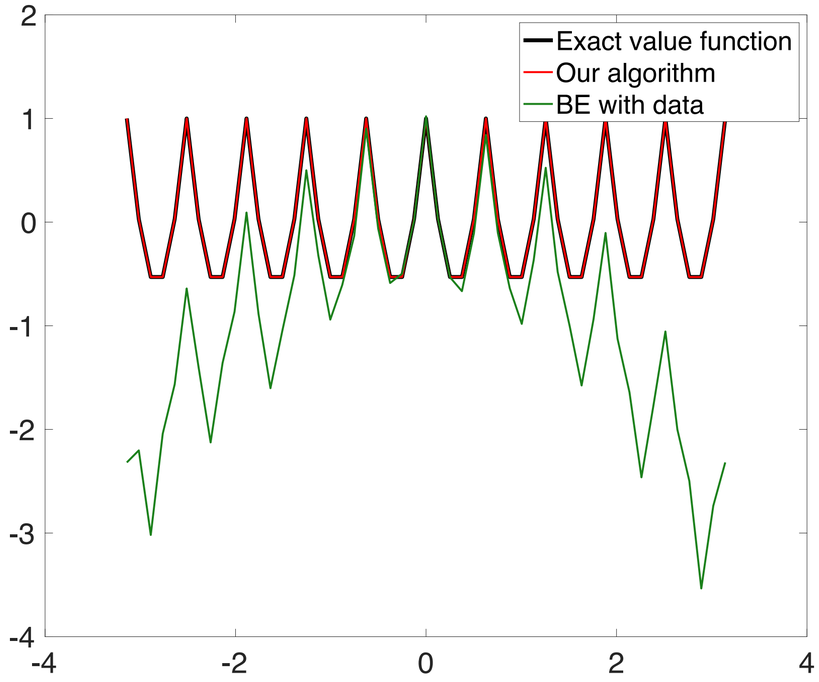
Underlying dynamics: $\frac{d}{dt}s_t = 0.05 s_t$

$\Delta t = 5, \beta = 0.1, V(s) = \cos^3(s)$



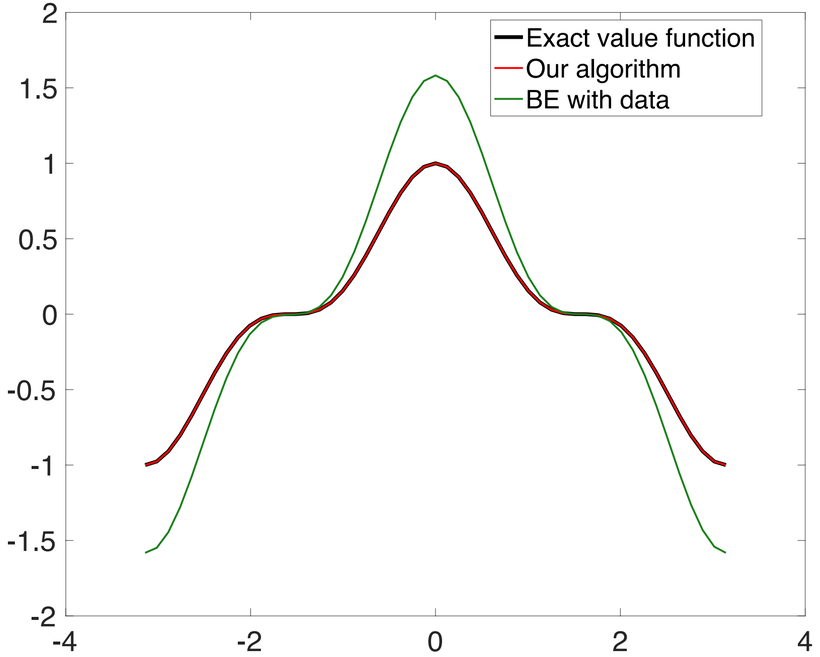
# of data: 40

$\Delta t = 0.5, \beta = 0.1, V(s) = \cos^3(10s)$



# of data: 400

$\Delta t = 0.1, \beta = 10, V(s) = \cos^3(s)$

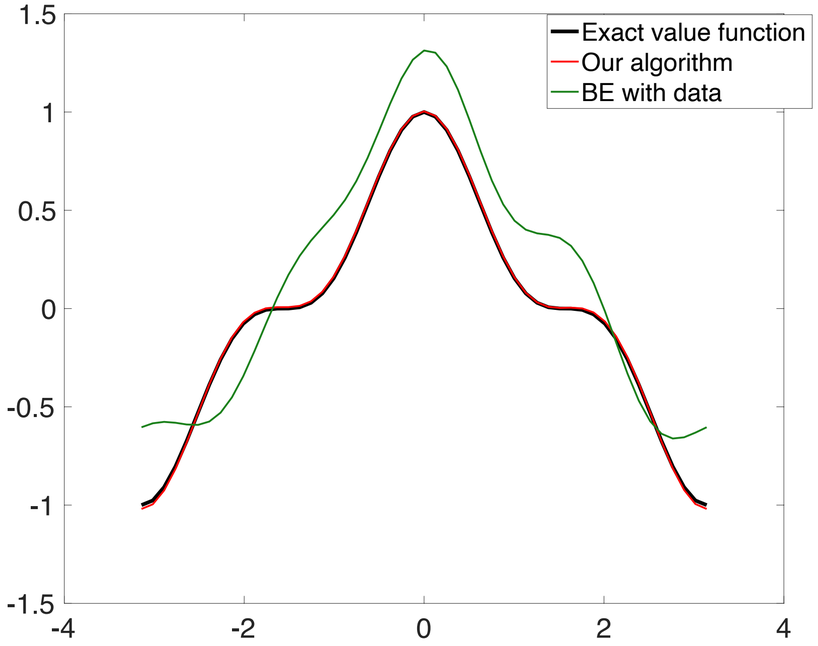


# of data: 40

Given the same trajectory data: Our algorithm v.s. LSTD

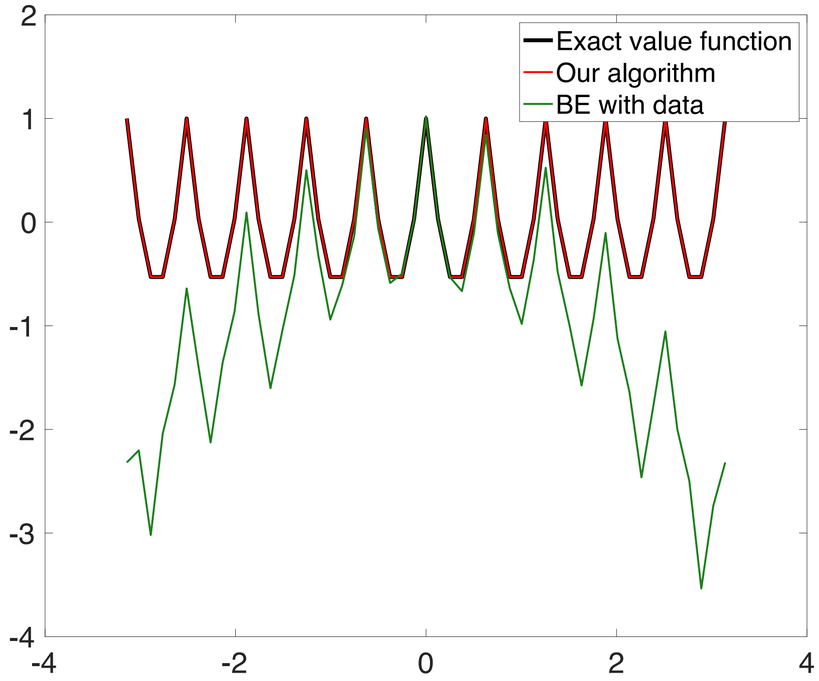
Underlying dynamics: $\frac{d}{dt}s_t = 0.05 s_t$

$\Delta t = 5, \beta = 0.1, V(s) = \cos^3(s)$



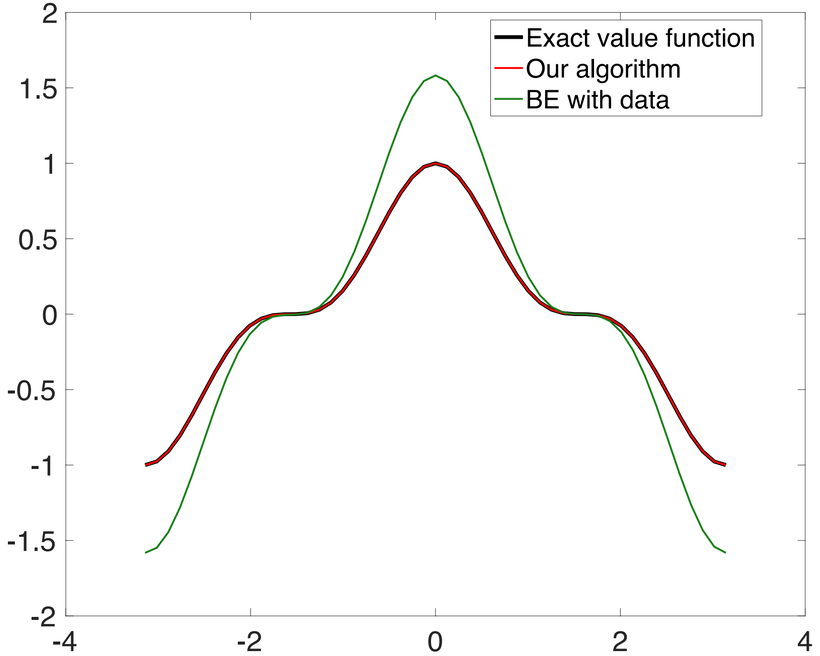
# of data: 40

$\Delta t = 0.5, \beta = 0.1, V(s) = \cos^3(10s)$



# of data: 400

$\Delta t = 0.1, \beta = 10, V(s) = \cos^3(s)$



# of data: 40

With the same computational cost!

Algorithm

LSTD

$$\tilde{V}_\theta(s) = \Phi(s)^\top \theta \quad <-- \quad \tilde{A}\theta = \tilde{b}$$

$$\tilde{b} = \sum_{j=1}^{J} \sum_{i=0}^{m-1} r(s^j_{i\Delta t}) \Phi(s^j_{i\Delta t})$$

$$\tilde{A} = \frac{1}{\Delta t} \sum_{j=1}^{J} \sum_{i=0}^{m-1} \left(\Phi(s^j_{i\Delta t}) - e^{\beta \Delta t} \Phi(s^j_{(i+1)\Delta t}) \right) \Phi(s^j_{i\Delta t})^\top$$

Our algorithm

$$\hat{V}_\theta(s) = \Phi(s)^\top \theta \quad <-- \quad \hat{A}\theta = \hat{b}$$

$$\hat{b} = \sum_{j=1}^{J} \sum_{i=0}^{m-1} r(s^j_{i\Delta t}) \Phi(s^j_{i\Delta t})$$

$$\hat{A} = \sum_{j=1}^{J} \sum_{i=0}^{m-1} \left(\beta \Phi(s^j_{i\Delta t}) - \frac{1}{\Delta t} \left(s^j_{(i+1)\Delta t} - s^j_{i\Delta t} \right) \cdot \nabla \Phi(s^j_{i\Delta t}) \right) \Phi(s^j_{i\Delta t})^\top$$

Why & When?

Why & When?

- Is it possible that it is the problem of LSTD? Is it possible that other RL algorithms work?
- When is BE not a good approximation to the continuous-time RL?
- What is the underlying equation behind our algorithm?
- When does our algorithm approximate work well?

Why LSTD is bad?

Most of the
RL algorithms

Why LSTD is bad?

Most of the RL algorithms $\xrightarrow[\text{to approximate}]{\text{use trajectory data}}$

Why LSTD is bad?

Most of the RL algorithms $\xrightarrow[\text{to approximate}]{\text{use trajectory data}}$ BE: $\tilde{V}(s) = \tilde{r}(s) + \gamma \mathbb{E}\left[\tilde{V}(s_{t+\Delta t}) \,|\, s_t = s\right]$

Why LSTD is bad?

Most of the RL algorithms $\xrightarrow[\text{to approximate}]{\text{use trajectory data}}$ BE: $\tilde{V}(s) = \tilde{r}(s) + \gamma \mathbb{E}\left[\tilde{V}(s_{t+\Delta t}) \,|\, s_t = s\right]$

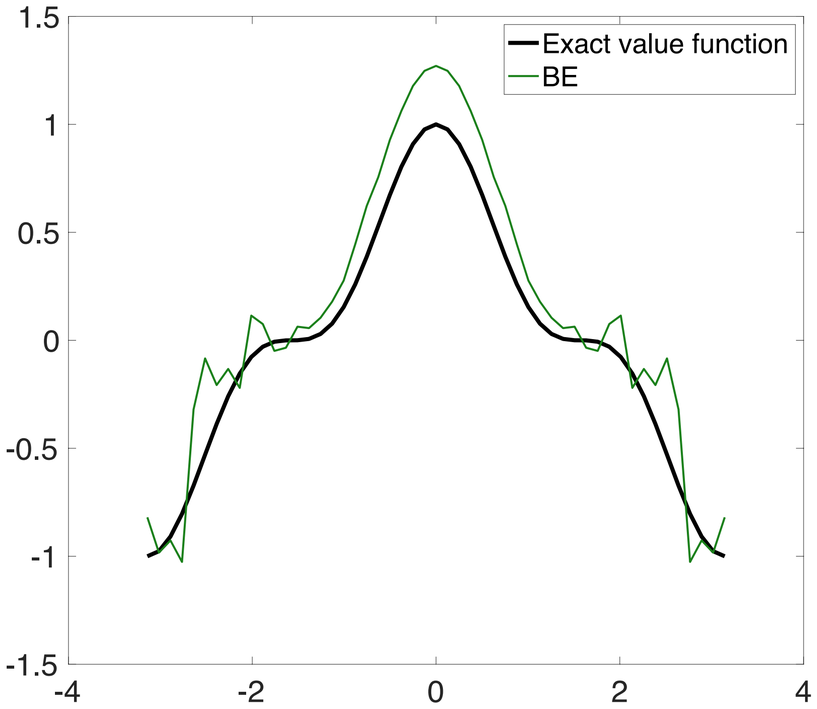
The solution to BE a NOT good approximation for continuous-time RL

Why LSTD is bad?

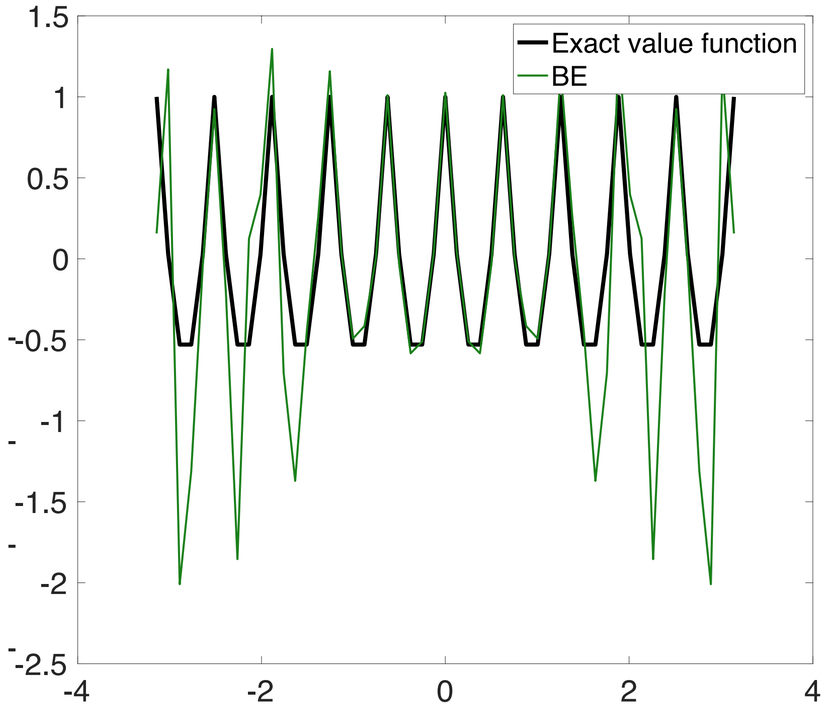
Most of the RL algorithms $\xrightarrow[\text{to approximate}]{\text{use trajectory data}}$ BE: $\tilde{V}(s) = \tilde{r}(s) + \gamma \mathbb{E}\left[\tilde{V}(s_{t+\Delta t}) \,|\, s_t = s\right]$

The solution to BE a NOT good approximation for continuous-time RL

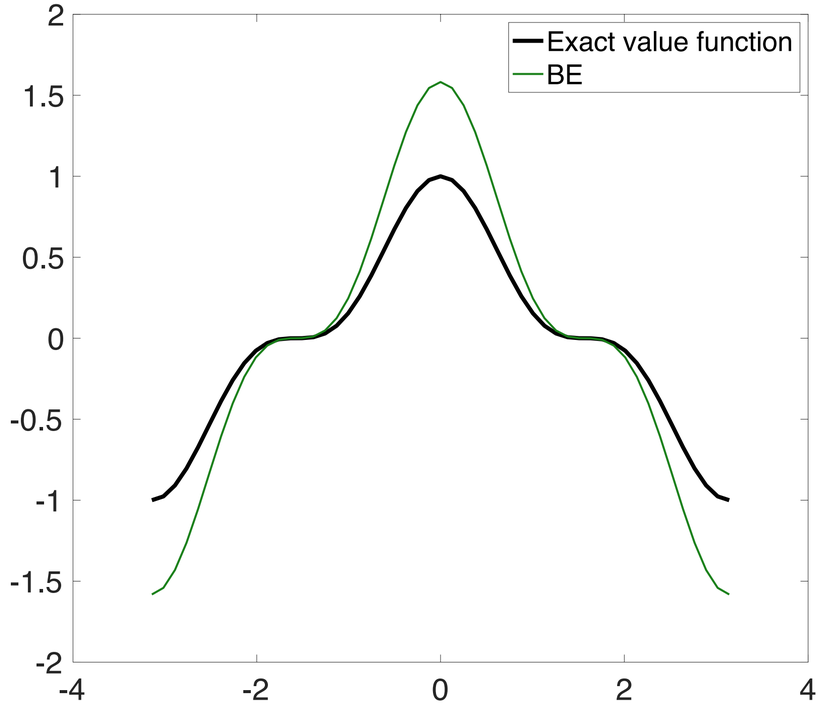
$\Delta t = 5, \beta = 0.1, V(s) = \cos^3(s)$



$\Delta t = 0.5, \beta = 0.1, V(s) = \cos^3(10s)$



$\Delta t = 0.1, \beta = 10, V(s) = \cos^3(s)$

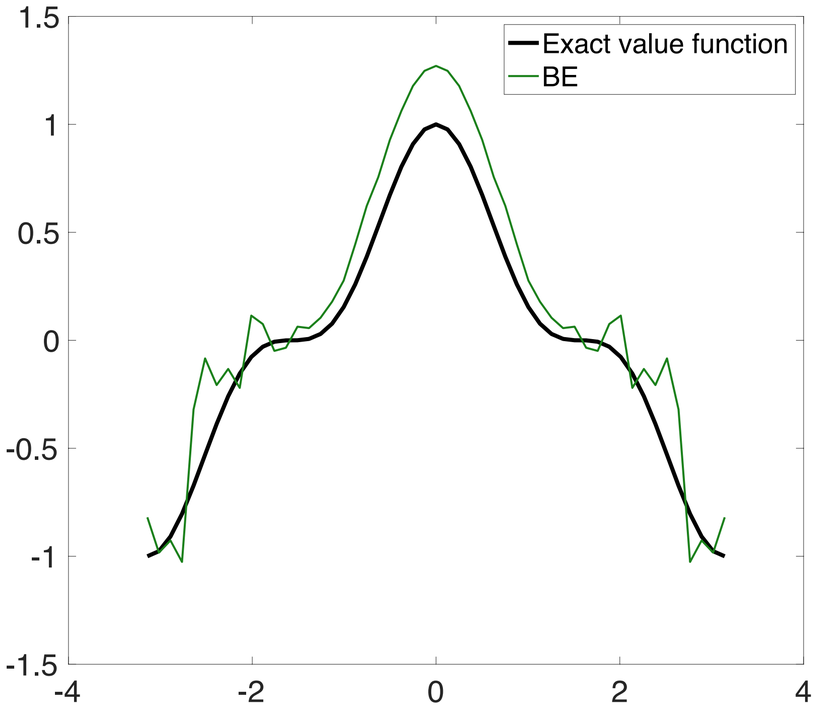


Why LSTD is bad?

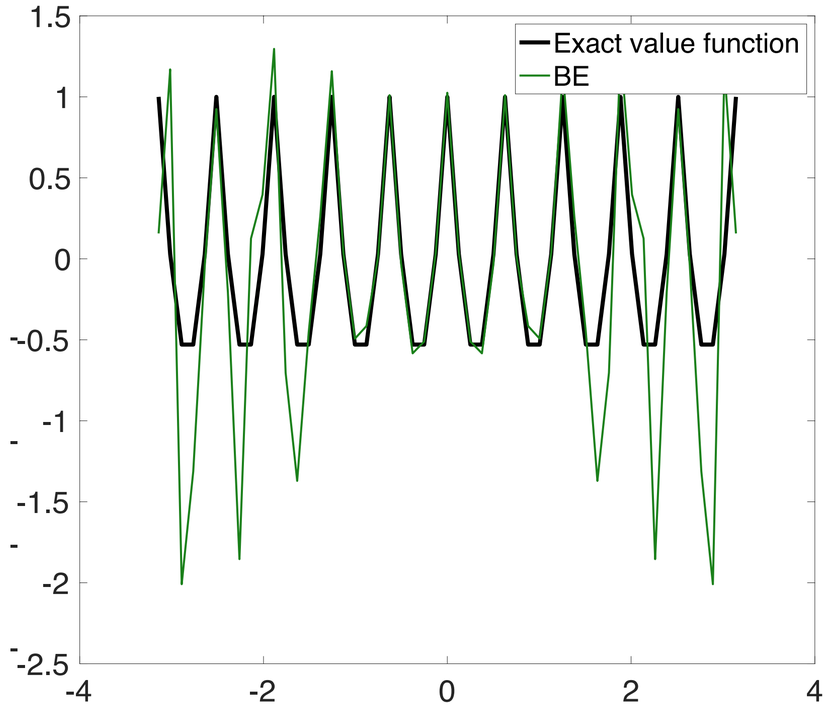
Most of the RL algorithms $\xrightarrow[\text{to approximate}]{\text{use trajectory data}}$ BE: $\tilde{V}(s) = \tilde{r}(s) + \gamma \mathbb{E}\left[\tilde{V}(s_{t+\Delta t}) \,|\, s_t = s\right]$

The solution to BE a NOT good approximation for continuous-time RL

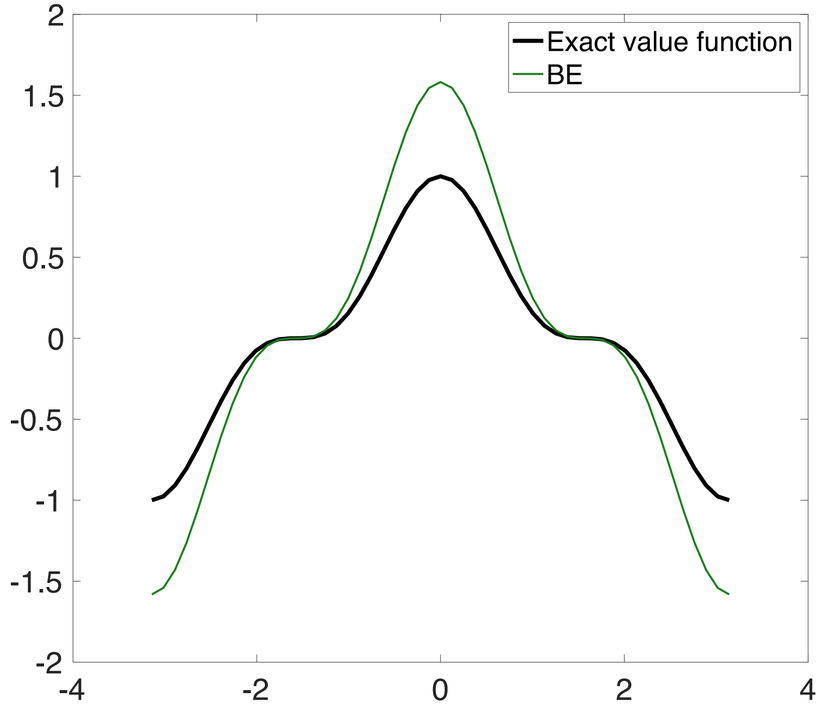
$\Delta t = 5, \beta = 0.1, V(s) = \cos^3(s)$



$\Delta t = 0.5, \beta = 0.1, V(s) = \cos^3(10s)$



$\Delta t = 0.1, \beta = 10, V(s) = \cos^3(s)$



- **The RL algorithm is converging to BE solution, not the true value function**

Why & When?

- Is it possible that it is the problem of LSTD? Is it possible that other RL algorithms work?
  - As long as the algorithm is derived from BE, it won't work under these circumstances.
- When is BE not a good approximation to the continuous-time RL?
- What is the underlying equation behind our algorithm?
- When does our algorithm approximate work well?

Why & When?

- Is it possible that it is the problem of LSTD? Is it possible that other RL algorithms work?
  - As long as the algorithm is derived from BE, it won't work under these circumstances.

- When is BE not a good approximation to the continuous-time RL?

- What is the underlying equation behind our algorithm?

- When does our algorithm approximate work well?

When is Bellman equation bad?

True value function: $\beta V(s) = r(s) + \mu(s) \cdot \nabla V(s) + \frac{1}{2}\Sigma(s) : \nabla^2 V(s)$

BE: $\tilde{V}(s) = r(s)\Delta t + e^{-\beta \Delta t}\mathbb{E}[V(s_{\Delta t}) \,|\, s_0 = s]$

Theorem for RL approximation [Z-24]

Assume that $\|r(s)\|_{L^\infty}$, $\|\mathscr{L}_{\mu,\Sigma} r(s)\|_{L^\infty}$ are bounded, then

$$\|V(s) - \tilde{V}(s)\|_{L^\infty} \lesssim \frac{\beta \|r\|_{L^\infty} + \|\mathscr{L}_{\mu,\Sigma} r\|_{L^\infty}}{\beta} \Delta t + o(\Delta t)$$

$$\mathscr{L}_{\mu,\Sigma} = \mu(s) \cdot \nabla + \frac{1}{2}\Sigma(s) : \nabla^2, \quad \text{where } \Sigma = \sigma\sigma^\top$$

Why & When?

- Is it possible that it is the problem of LSTD? Is it possible that other RL algorithms work?
  - As long as the algorithm is derived from BE, it won't work under these circumstances.

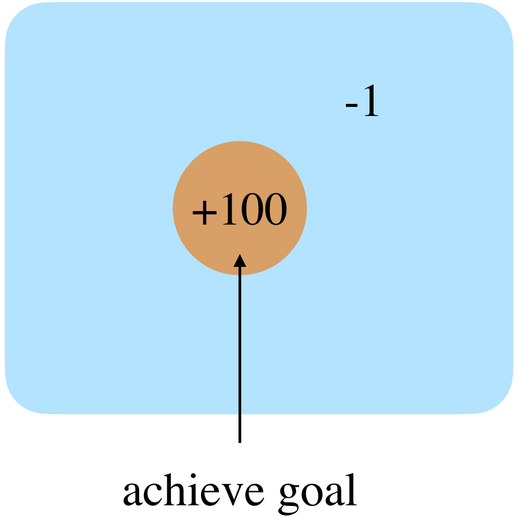
- When is BE not a good approximation to the continuous-time RL?
  - When the reward function has a large variation

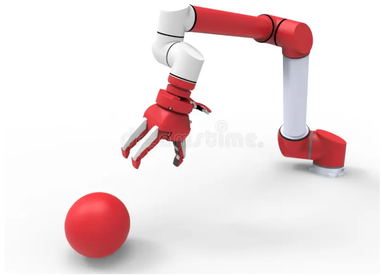
- What is the underlying equation behind our algorithm?

- When does our algorithm approximate work well?

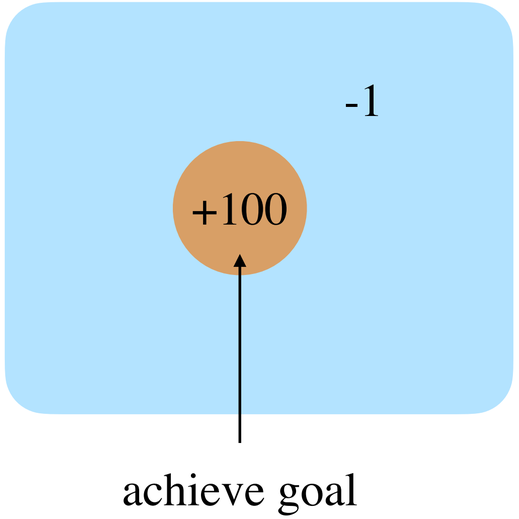
How does reward look like in RL applications?

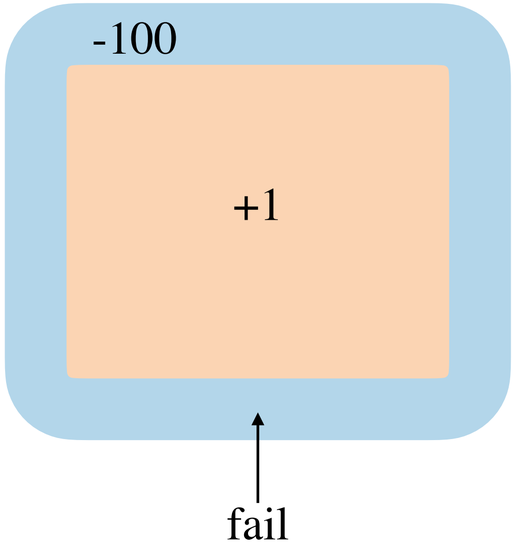
How does reward look like in RL applications?

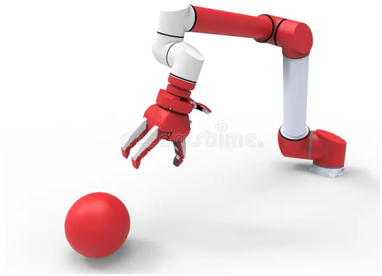


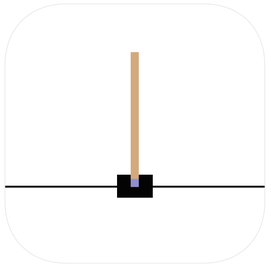


How does reward look like in RL applications?

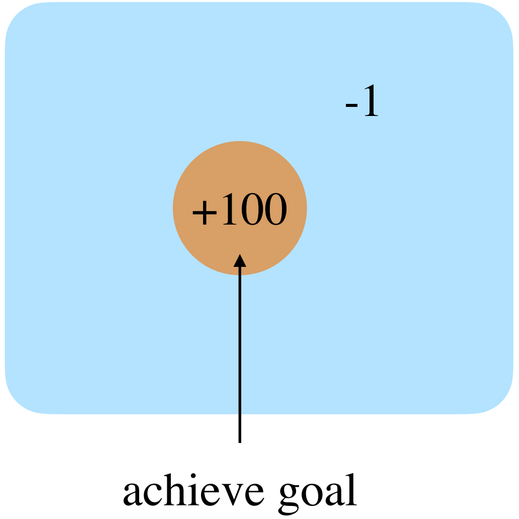


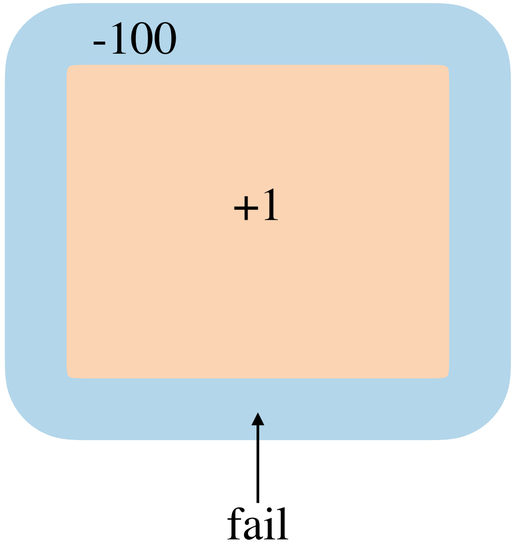


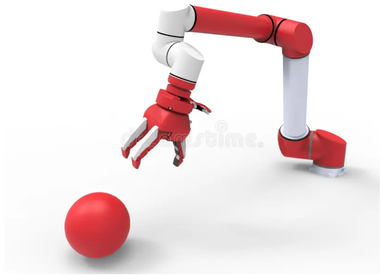


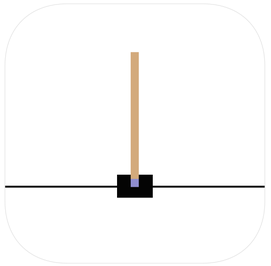


How does reward look like in RL applications?









The reward often varies a lot!

Why & When?

- Is it possible that it is the problem of LSTD? Is it possible that other RL algorithms work?
  - As long as the algorithm is derived from BE, it won't work under these circumstances.

- When is BE not a good approximation to the continuous-time RL?
  - When the reward function has a large variation

- What is the underlying equation behind our algorithm?

- When does our algorithm approximate work well?

What is the underlying equation behind our algorithm

Deterministic Dynamics

What is the underlying equation behind our algorithm

Deterministic Dynamics

- $p_{\Delta t}(s)$ represents the state at $t + \Delta t$ given state at t is s

$$\beta \hat{V}(s) = r(s) + \frac{1}{\Delta t}(p_{\Delta t}(s) - s) \cdot \nabla \hat{V}(s)$$

What is the underlying equation behind our algorithm

Deterministic Dynamics

- $p_{\Delta t}(s)$ represents the state at $t + \Delta t$ given state at t is s

$$\beta \hat{V}(s) = r(s) + \frac{1}{\Delta t}(p_{\Delta t}(s) - s) \cdot \nabla \hat{V}(s)$$

Bellman Equation

$$\tilde{V}(s) = r(s)\Delta t + e^{-\beta \Delta t} \tilde{V}(p_{\Delta t}(s))$$

What is the underlying equation behind our algorithm

Deterministic Dynamics

-$p_{\Delta t}(s)$ represents the state at $t + \Delta t$ given state at t is s

$$\beta \hat{V}(s) = r(s) + \frac{1}{\Delta t}(p_{\Delta t}(s) - s) \cdot \nabla \hat{V}(s)$$

Bellman Equation

$$\tilde{V}(s) = r(s)\Delta t + e^{-\beta \Delta t}\tilde{V}(p_{\Delta t}(s))$$

- Similarity: two important features of BE

  - Same formulation for different dynamics $(b(s), \sigma(s))$
  - Only depends on the current state and next state

- Difference: No smoothness infomation in continuous-time

What is the underlying equation behind our algorithm

Deterministic Dynamics

- $p_{\Delta t}(s)$ represents the state at $t + \Delta t$ given state at t is s

$$\beta \hat{V}(s) = r(s) + \frac{1}{\Delta t}(p_{\Delta t}(s) - s) \cdot \nabla \hat{V}(s)$$

Bellman Equation

$$\tilde{V}(s) = r(s)\Delta t + e^{-\beta \Delta t}\tilde{V}(p_{\Delta t}(s))$$

Hamilton-Jacobi Equation

$$\beta V(s) = r(s) + b(s) \cdot \nabla V(s)$$

- Similarity: two important features of BE

  - Same formulation for different dynamics $(b(s), \sigma(s))$
  - Only depends on the current state and next state

- Difference: No smoothness infomation in continuous-time

What is the underlying equation behind our algorithm

Deterministic Dynamics

- $p_{\Delta t}(s)$ represents the state at $t + \Delta t$ given state at t is s

$$\beta \hat{V}(s) = r(s) + \frac{1}{\Delta t}(p_{\Delta t}(s) - s) \cdot \nabla \hat{V}(s)$$

Bellman Equation

$$\tilde{V}(s) = r(s)\Delta t + e^{-\beta \Delta t}\tilde{V}(p_{\Delta t}(s))$$

Hamilton-Jacobi Equation

$$\beta V(s) = r(s) + b(s) \cdot \nabla V(s)$$

- Similarity: two important features of BE
  - Same formulation for different dynamics $(b(s), \sigma(s))$
  - Only depends on the current state and next state
- Difference: No smoothness infomation in continuous-time

- Similarity: PDE, containing continuous-time information
- Difference: Only continuous information, has to estimate the dynamics first

Model-based PDE formulation

Pros

- Keep the **PDE form** to estimate $V(s)$
  - Well-developed PDE analysis tool
  - Keep the continuous-time structure

Model-free RL algorithms

Pros

- **Model-free**
  - One can skip the inverse problem and directly compute $V(s)$
  - Many practical RL algorithms

Combine the advantages of PDE formulation with model-free algorithm

PhiBE in general dynamics

PhiBE: $$\beta\hat{V}_1(s) = r(s) + \frac{1}{\Delta t}\mathbb{E}[s_{\Delta t} - s \,|\, s_0 = s] \cdot \nabla \hat{V}_1(s) + \frac{1}{2\Delta t}\mathbb{E}[(s_{\Delta t} - s)(s_{\Delta t} - s)^\top \,|\, s_0 = s] : \nabla^2 \hat{V}_1(s)$$

PhiBE in general dynamics

PhiBE: $$\beta \hat{V}_1(s) = r(s) + \frac{1}{\Delta t}\mathbb{E}[s_{\Delta t} - s \,|\, s_0 = s] \cdot \nabla \hat{V}_1(s) + \frac{1}{2\Delta t}\mathbb{E}[(s_{\Delta t} - s)(s_{\Delta t} - s)^\top \,|\, s_0 = s] : \nabla^2 \hat{V}_1(s)$$

PhiBE, short for physics-informed Bellman equation

PhiBE in general dynamics

PhiBE: $$\beta \hat{V}_1(s) = r(s) + \frac{1}{\Delta t}\mathbb{E}[s_{\Delta t} - s \,|\, s_0 = s] \cdot \nabla \hat{V}_1(s) + \frac{1}{2\Delta t}\mathbb{E}[(s_{\Delta t} - s)(s_{\Delta t} - s)^\top \,|\, s_0 = s] : \nabla^2 \hat{V}_1(s)$$

PhiBE, short for physics-informed Bellman equation

The form of the PDE

PhiBE in general dynamics

PhiBE: $$\beta\hat{V}_1(s) = r(s) + \frac{1}{\Delta t}\mathbb{E}[s_{\Delta t} - s \,|\, s_0 = s] \cdot \nabla \hat{V}_1(s) + \frac{1}{2\Delta t}\mathbb{E}[(s_{\Delta t} - s)(s_{\Delta t} - s)^\top \,|\, s_0 = s] : \nabla^2 \hat{V}_1(s)$$

PhiBE, short for physics-informed Bellman equation

The form of the PDE

↑

derived from the true continuous-time physical environment

PhiBE in general dynamics

PhiBE: $$\beta\hat{V}_1(s) = r(s) + \frac{1}{\Delta t}\mathbb{E}[s_{\Delta t} - s \,|\, s_0 = s] \cdot \nabla \hat{V}_1(s) + \frac{1}{2\Delta t}\mathbb{E}[(s_{\Delta t} - s)(s_{\Delta t} - s)^\top \,|\, s_0 = s] : \nabla^2 \hat{V}_1(s)$$

PhiBE, short for physics-informed Bellman equation

The form of the PDE

↑

derived from the true continuous-time physical environment

+

Contains discrete-time information

PhiBE in general dynamics

PhiBE: $$\beta\hat{V}_1(s) = r(s) + \frac{1}{\Delta t}\mathbb{E}[s_{\Delta t} - s \,|\, s_0 = s] \cdot \nabla \hat{V}_1(s) + \frac{1}{2\Delta t}\mathbb{E}[(s_{\Delta t} - s)(s_{\Delta t} - s)^\top \,|\, s_0 = s] : \nabla^2 \hat{V}_1(s)$$

PhiBE, short for physics-informed Bellman equation

The form of the PDE ↑

derived from the true continuous-time physical environment

+

Contains discrete-time information ↑

similar to the Bellman equation

PhiBE in general dynamics

PhiBE: $$\beta\hat{V}_1(s) = r(s) + \frac{1}{\Delta t}\mathbb{E}[s_{\Delta t} - s \,|\, s_0 = s] \cdot \nabla \hat{V}_1(s) + \frac{1}{2\Delta t}\mathbb{E}[(s_{\Delta t} - s)(s_{\Delta t} - s)^\top \,|\, s_0 = s] : \nabla^2 \hat{V}_1(s)$$

PhiBE, short for physics-informed Bellman equation

The form of the PDE ↑ derived from the true continuous-time physical environment

+

Contains discrete-time information ↑ similar to the Bellman equation

PhiBE, incorporating discrete-time information into continuous-time PDE

Higher-order PhiBE in deterministic dynamics

i-th order PhiBE (deterministic dynamics)

i-th order PhiBE: $\beta \hat{V}_i(s) = r(s) + \hat{\mu}_i(s) \cdot \nabla \hat{V}_i(s)$

$$\text{with} \quad \hat{\mu}_i(s) = \frac{1}{\Delta t} \left[\sum_{j=1}^{i} a_j^{(i)} (s_{j\Delta t} - s_0) \,\middle|\, s_0 = s \right]$$

Here the coefficients $(a_0^{(i)}, \cdots, a_i^{(i)})^\top = (A^{(i)})^{-1} b^{(i)}$,
where $A_{kj}^{(i)} = j^k$, $b_k^{(i)} = (0,1,0,\cdots,0)^\top$

Higher-order PhiBE in deterministic dynamics

i-th order PhiBE (deterministic dynamics)

i-th order PhiBE: $\beta \hat{V}_i(s) = r(s) + \hat{\mu}_i(s) \cdot \nabla \hat{V}_i(s)$

$$\text{with} \quad \hat{\mu}_i(s) = \frac{1}{\Delta t} \left[\sum_{j=1}^{i} a_j^{(i)} (s_{j\Delta t} - s_0) \middle| s_0 = s \right]$$

Here the coefficients $(a_0^{(i)}, \cdots, a_i^{(i)})^\top = (A^{(i)})^{-1} b^{(i)}$,
where $A_{kj}^{(i)} = j^k, b_k^{(i)} = (0,1,0,\cdots,0)^\top$

Different from classical numerical schemes

Higher-order PhiBE in deterministic dynamics

i-th order PhiBE (deterministic dynamics)

i-th order PhiBE: $\beta \hat{V}_i(s) = r(s) + \hat{\mu}_i(s) \cdot \nabla \hat{V}_i(s)$

$$\text{with} \quad \hat{\mu}_i(s) = \frac{1}{\Delta t} \left[\sum_{j=1}^{i} a_j^{(i)} (s_{j\Delta t} - s_0) \middle| s_0 = s \right]$$

Here the coefficients $(a_0^{(i)}, \cdots, a_i^{(i)})^\top = (A^{(i)})^{-1} b^{(i)}$,
where $A_{kj}^{(i)} = j^k, b_k^{(i)} = (0,1,0,\cdots,0)^\top$

Different from classical numerical schemes

Numerical schemes known dynamics ——> discrete trajectory

PhiBE known discrete trajectory ——> dynamics

Why & When?

- Is it possible that it is the problem of LSTD? Is it possible that other RL algorithms work?
  - As long as the algorithm is derived from BE, it won't work under these circumstances.

- When is BE not a good approximation to the continuous-time RL?
  - When the reward function has a large variation

- What is the underlying equation behind our algorithm?
  - PhiBE & High-order PhiBE

- When does our algorithm approximate work well?

Why & When?

- Is it possible that it is the problem of LSTD? Is it possible that other RL algorithms work?
  - As long as the algorithm is derived from BE, it won't work under these circumstances.

- When is BE not a good approximation to the continuous-time RL?
  - When the reward function has a large variation

- What is the underlying equation behind our algorithm?
  - PhiBE & High-order PhiBE

- When does our algorithm approximate work well?

Theoretical Guarantees in general deterministic dynamics

$$\frac{d}{dt}s_t = \mu(s_t)$$

i-th order PhiBE [Z-24]

Assume that $\|\nabla r(s)\|_{L^\infty}$, $\|\mathscr{L}_\mu^i \mu(s)\|_{L^\infty}$ are bounded.
In addition, $\|\nabla \mu(s)\|_{L^\infty} < \beta$

$$\|V(s) - \hat{V}_i(s)\|_{L^\infty} \leq \frac{\|\nabla r\|_{L^\infty}\|\mathscr{L}_\mu^i \mu\|_{L^\infty}}{(\beta - \|\nabla \mu(s)\|_{L^\infty})^2}\Delta t^i$$

where $\mathscr{L}_\mu = \mu(s)\cdot\nabla$,

$$\|\mathscr{L}_\mu^i \mu\|_{L^\infty} = \left\| \frac{d^{i+1}}{dt^{i+1}} s_t \right\|_{L^\infty}$$

Theoretical Guarantees in general deterministic dynamics

$$\frac{d}{dt}s_t = \mu(s_t)$$

i-th order PhiBE [Z-24]

Assume that $\|\nabla r(s)\|_{L^\infty}$, $\|\mathscr{L}_\mu^i \mu(s)\|_{L^\infty}$ are bounded.
In addition, $\|\nabla \mu(s)\|_{L^\infty} < \beta$

$$\|V(s) - \hat{V}_i(s)\|_{L^\infty} \leq \frac{\|\nabla r\|_{L^\infty}\|\mathscr{L}_\mu^i \mu\|_{L^\infty}}{(\beta - \|\nabla \mu(s)\|_{L^\infty})^2}\Delta t^i$$

The dynamics changes slowly, the error is smaller

where $\mathscr{L}_\mu = \mu(s)\cdot\nabla$,

$$\|\mathscr{L}_\mu^i \mu\|_{L^\infty} = \left\| \frac{d^{i+1}}{dt^{i+1}} s_t \right\|_{L^\infty}$$

Theoretical Guarantees in general deterministic dynamics

$$\frac{d}{dt}s_t = \mu(s_t)$$

i-th order PhiBE [Z-24]

Assume that $\|\nabla r(s)\|_{L^\infty}$, $\|\mathscr{L}_\mu^i \mu(s)\|_{L^\infty}$ are bounded.
In addition, $\|\nabla \mu(s)\|_{L^\infty} < \beta$

$$\|V(s) - \hat{V}_i(s)\|_{L^\infty} \leq \frac{\|\nabla r\|_{L^\infty}\|\mathscr{L}_\mu^i \mu\|_{L^\infty}}{(\beta - \|\nabla \mu(s)\|_{L^\infty})^2}\Delta t^i$$

where $\mathscr{L}_\mu = \mu(s)\cdot\nabla$,

$$\|\mathscr{L}_\mu^i \mu\|_{L^\infty} = \left\| \frac{d^{i+1}}{dt^{i+1}} s_t \right\|_{L^\infty}$$

The dynamics changes slowly, the error is smaller

The advantage of error depending more on the dynamics instead of reward:

- More flexibility of designing the reward function
- Only need less data points to achieve the same error

Theoretical Guarantees in stochastic dynamics

Weighted L^2 norm

Define

$$\|f\|_\rho = \int f^2(s)\rho(s)ds,$$

where $\rho(s)$ is the stationary distribution of the SDE that satisfies

$$\nabla \cdot \left[\mu(s)\rho(s) + \frac{1}{2}\nabla \cdot (\Sigma(s)\rho(s))\right] = 0$$

Theoretical Guarantees in stochastic dynamics

Weighted L^2 norm

Define

$$\|f\|_\rho = \int f^2(s)\rho(s)ds,$$

where $\rho(s)$ is the stationary distribution of the SDE that satisfies

$$\nabla \cdot \left[\mu(s)\rho(s) + \frac{1}{2}\nabla \cdot (\Sigma(s)\rho(s))\right] = 0$$

Bellman Equation [Z-24]

Assume that $\|\mu(s)\|_\rho$, $\|\Sigma(s)\|_\rho$, $\|\nabla^k r(s)\|_\rho$ for $k = 0,1,2$ are bounded, then

$$\|V(s) - \tilde{V}(s)\|_\rho \leq \frac{C_1}{\beta}\Delta t + o(\Delta t)$$

Theoretical Guarantees in stochastic dynamics

Weighted L^2 norm

Define

$$\|f\|_\rho = \int f^2(s)\rho(s)ds,$$

where $\rho(s)$ is the stationary distribution of the SDE that satisfies

$$\nabla \cdot \left[\mu(s)\rho(s) + \frac{1}{2}\nabla \cdot (\Sigma(s)\rho(s))\right] = 0$$

Bellman Equation [Z-24]

Assume that $\|\mu(s)\|_\rho$, $\|\Sigma(s)\|_\rho$, $\|\nabla^k r(s)\|_\rho$ for $k = 0,1,2$ are bounded, then

$$\|V(s) - \tilde{V}(s)\|_\rho \leq \frac{C_1}{\beta}\Delta t + o(\Delta t)$$

PhiBE [Z-24]

Assume that $\lambda_{\min}(\Sigma(s)) \geq \lambda_{\min} > 0$,

$\|\nabla^k \mu(s)\|_{L^\infty}$, $\|\nabla^k \Sigma(s)\|_{L^\infty}$, for $k = 0,\cdots,2i$ are bounded,

$\max_{k,l} \sum_i \|\partial_{s_i} \Sigma_{kl}(s)\|_{L^\infty} \leq 2\lambda_{\min}$, then

$$\|V(s) - \hat{V}_i(s)\|_\rho \leq \frac{C_2}{\beta^2}\Delta t^i$$

Theoretical Guarantees in stochastic dynamics

Weighted L^2 norm

Define

$$\|f\|_\rho = \int f^2(s)\rho(s)ds,$$

where $\rho(s)$ is the stationary distribution of the SDE that satisfies

$$\nabla \cdot \left[\mu(s)\rho(s) + \frac{1}{2}\nabla \cdot (\Sigma(s)\rho(s))\right] = 0$$

Bellman Equation [Z-24]

Assume that $\|\mu(s)\|_\rho$, $\|\Sigma(s)\|_\rho$, $\|\nabla^k r(s)\|_\rho$ for $k = 0,1,2$ are bounded, then

$$\|V(s) - \tilde{V}(s)\|_\rho \leq \frac{C_1}{\beta}\Delta t + o(\Delta t)$$

When the dynamics changes slowly or the magnitude of the noise is large, 1st order PhiBE is better than BE.

PhiBE [Z-24]

Assume that $\lambda_{\min}(\Sigma(s)) \geq \lambda_{\min} > 0$,

$\|\nabla^k \mu(s)\|_{L^\infty}$, $\|\nabla^k \Sigma(s)\|_{L^\infty}$, for $k = 0,\cdots,2i$ are bounded,

$\max_{k,l} \sum_i \|\partial_{s_i} \Sigma_{kl}(s)\|_{L^\infty} \leq 2\lambda_{\min}$, then

$$\|V(s) - \hat{V}_i(s)\|_\rho \leq \frac{C_2}{\beta^2}\Delta t^i$$

Theoretical Guarantees for the linear approximation

$$\beta\hat{V}(s) = r(s) + \frac{1}{\Delta t}\mathbb{E}[s_{\Delta t} - s \,|\, s_0 = s] \cdot \nabla \hat{V}(s) + \frac{1}{2\Delta t}\mathbb{E}[(s_{\Delta t} - s)(s_{\Delta t} - s)^\top \,|\, s_0 = s] : \nabla^2 \hat{V}(s)$$

Theoretical Guarantees for the linear approximation

$$\beta\hat{V}(s) = r(s) + \frac{1}{\Delta t}\mathbb{E}[s_{\Delta t} - s \,|\, s_0 = s] \cdot \nabla \hat{V}(s) + \frac{1}{2\Delta t}\mathbb{E}[(s_{\Delta t} - s)(s_{\Delta t} - s)^\top \,|\, s_0 = s] : \nabla^2 \hat{V}(s)$$

$$\downarrow \quad \hat{V}^G(s) = \theta^\top \Phi(s)$$

Theoretical Guarantees for the linear approximation

$$\beta\hat{V}(s) = r(s) + \frac{1}{\Delta t}\mathbb{E}[s_{\Delta t} - s \,|\, s_0 = s] \cdot \nabla \hat{V}(s) + \frac{1}{2\Delta t}\mathbb{E}[(s_{\Delta t} - s)(s_{\Delta t} - s)^\top \,|\, s_0 = s] : \nabla^2 \hat{V}(s)$$

$$\hat{V}^G(s) = \theta^\top \Phi(s)$$

$$\langle \beta\hat{V}^G(s) = r(s) + \frac{1}{\Delta t}\mathbb{E}[s_{\Delta t} - s \,|\, s_0 = s] \cdot \nabla \hat{V}^G(s) + \frac{1}{2\Delta t}\mathbb{E}[(s_{\Delta t} - s)(s_{\Delta t} - s)^\top \,|\, s_0 = s] : \nabla^2 \hat{V}^G(s), \Phi \rangle$$

Theoretical Guarantees for the linear approximation

$$\beta\hat{V}(s) = r(s) + \frac{1}{\Delta t}\mathbb{E}[s_{\Delta t} - s \,|\, s_0 = s] \cdot \nabla \hat{V}(s) + \frac{1}{2\Delta t}\mathbb{E}[(s_{\Delta t} - s)(s_{\Delta t} - s)^\top \,|\, s_0 = s] : \nabla^2 \hat{V}(s)$$

$$\hat{V}^G(s) = \theta^\top \Phi(s)$$

$$\langle \beta\hat{V}^G(s) = r(s) + \frac{1}{\Delta t}\mathbb{E}[s_{\Delta t} - s \,|\, s_0 = s] \cdot \nabla \hat{V}^G(s) + \frac{1}{2\Delta t}\mathbb{E}[(s_{\Delta t} - s)(s_{\Delta t} - s)^\top \,|\, s_0 = s] : \nabla^2 \hat{V}^G(s), \Phi \rangle$$

Model-free algorithm is using data to approximate $\hat{V}^G$

PhiBE under linear approximation [Z-24]

Under mild assumption on $b(s), \Sigma(s), r(s)$, and boundedness of $\|\nabla \log \rho\|_{L^\infty}$ or $\|\Phi\|_{L^\infty}$, $\Delta t^i \le C_{\Delta t}$, then

$$\|\hat{V}_i^G(s) - V(s)\|_\rho \le \frac{C_2}{\beta^2}\Delta t^i + C^G \min_{\hat{V}_P = \theta^\top \Phi} \|V_i(s) - \hat{V}_P(s)\|_{H^1_\rho}$$

$C_2 \,.\, C^G$ decreases as the dynamics changes slowly

Theoretical Guarantees for the linear approximation

$$\beta\hat{V}(s) = r(s) + \frac{1}{\Delta t}\mathbb{E}[s_{\Delta t} - s \,|\, s_0 = s] \cdot \nabla \hat{V}(s) + \frac{1}{2\Delta t}\mathbb{E}[(s_{\Delta t} - s)(s_{\Delta t} - s)^\top \,|\, s_0 = s] : \nabla^2 \hat{V}(s)$$

$$\hat{V}^G(s) = \theta^\top \Phi(s)$$

$$\langle \beta\hat{V}^G(s) = r(s) + \frac{1}{\Delta t}\mathbb{E}[s_{\Delta t} - s \,|\, s_0 = s] \cdot \nabla \hat{V}^G(s) + \frac{1}{2\Delta t}\mathbb{E}[(s_{\Delta t} - s)(s_{\Delta t} - s)^\top \,|\, s_0 = s] : \nabla^2 \hat{V}^G(s), \Phi \rangle$$

Model-free algorithm is using data to approximate $\hat{V}^G$

PhiBE under linear approximation [Z-24]

Under mild assumption on $b(s), \Sigma(s), r(s)$, and boundedness of $\|\nabla \log \rho\|_{L^\infty}$ or $\|\Phi\|_{L^\infty}$, $\Delta t^i \le C_{\Delta t}$, then

$$\|\hat{V}_i^G(s) - V(s)\|_\rho \le \frac{C_2}{\beta^2}\Delta t^i + C^G \min_{\hat{V}_P = \theta^\top \Phi} \|V_i(s) - \hat{V}_P(s)\|_{H^1_\rho}$$

$C_2 \,.\, C^G$ decreases as the dynamics changes slowly

Open problem: sample complexity of the algorithm

Conclusion

- Is it possible that it is the problem of LSTD? Is it possible that other RL algorithms work?
  - As long as the algorithm is derived from BE, it won't work under these circumstances.

- When is BE not a good approximation to the continuous-time RL?
  - When the reward function has a large variation

- What is the underlying equation behind our algorithm?
  - PhiBE & High-order PhiBE

- When does our algorithm approximate work well?
  - When the dynamics change slowly

PhiBE in linear dynamics

Linear dynamics

Underlying dynamics: $\frac{d}{dt}s_t = \lambda s_t$

PhiBE in linear dynamics

Linear dynamics

Underlying dynamics: $\frac{d}{dt}s_t = \lambda s_t$

$$s_t = e^{\lambda t} s_0$$

When $\lambda > 0$, the dynamics change exponentially fast

PhiBE in linear dynamics

Linear dynamics

Underlying dynamics: $\frac{d}{dt}s_t = \lambda s_t$

$$s_t = e^{\lambda t} s_0$$

When $\lambda > 0$, the dynamics change exponentially fast

PhiBE for linear dynamics [Z-24]

$$\|V(s) - \tilde{V}(s)\|_{L^\infty} \lesssim \frac{1}{\beta}\left(\beta\|r(s)\|_{L^\infty} + |\lambda|\|s \cdot \nabla r(s)\|_{L^\infty}\right)\Delta t$$

$$\|V(s) - \hat{V}_i(s)\|_{L^\infty} \lesssim \frac{1}{\beta^2}|\lambda|^{i+1}\|s\nabla r\|\Delta t^i$$

PhiBE in linear dynamics

Linear dynamics

Underlying dynamics: $\frac{d}{dt}s_t = \lambda s_t$

$$s_t = e^{\lambda t} s_0$$

When $\lambda > 0$, the dynamics change exponentially fast

PhiBE for linear dynamics [Z-24]

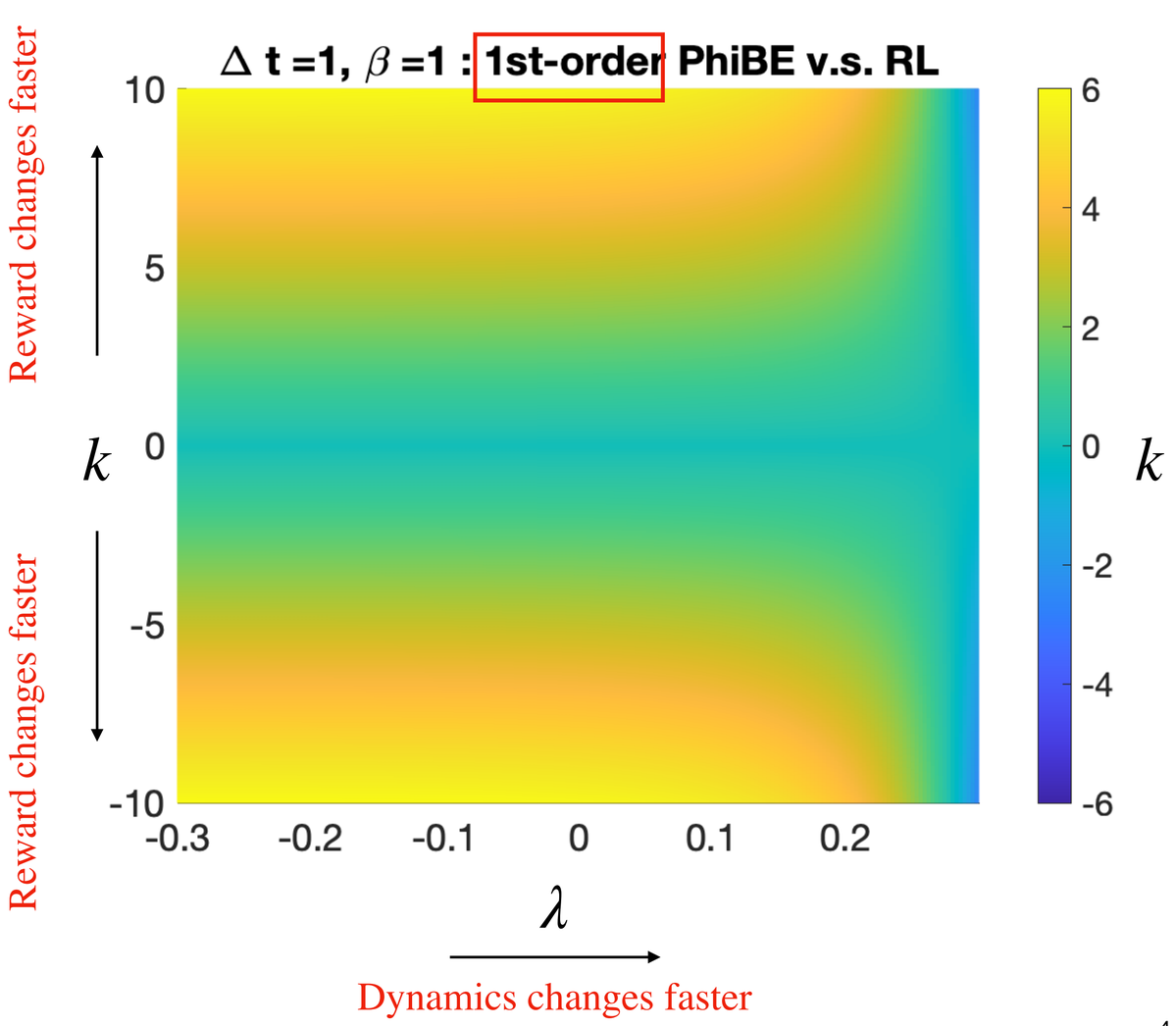
$$\|V(s) - \tilde{V}(s)\|_{L^\infty} \lesssim \frac{1}{\beta}\left(\beta\|r(s)\|_{L^\infty} + |\lambda|\|s \cdot \nabla r(s)\|_{L^\infty}\right)\Delta t$$

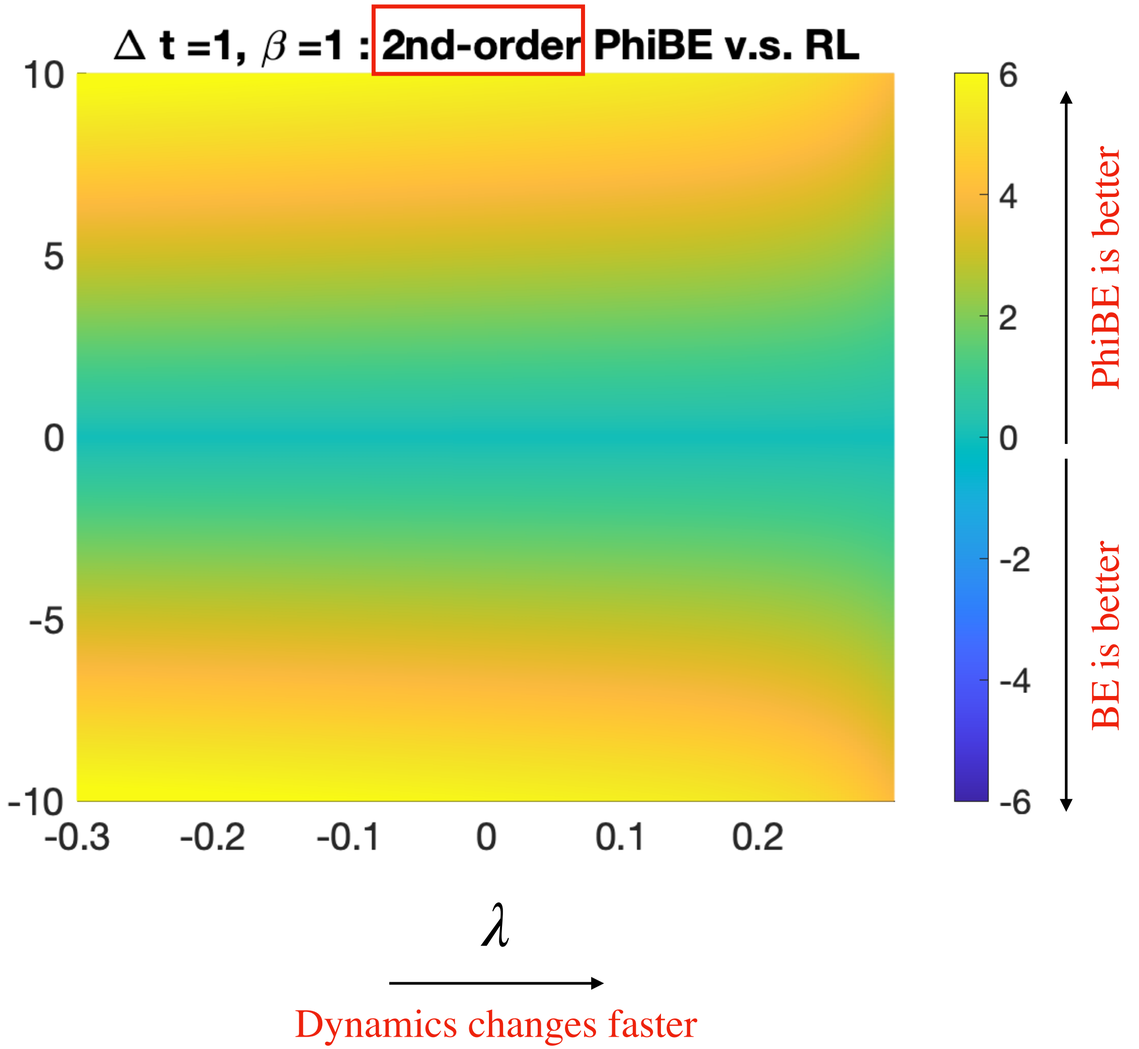
$$\|V(s) - \hat{V}_i(s)\|_{L^\infty} \lesssim \frac{1}{\beta^2}|\lambda|^{i+1}\|s\nabla r\|\Delta t^i$$

- When $|\lambda|$ is smaller, the error is smaller
- When $|\lambda|\Delta t < 1$, higher order PhiBE is better

LQR

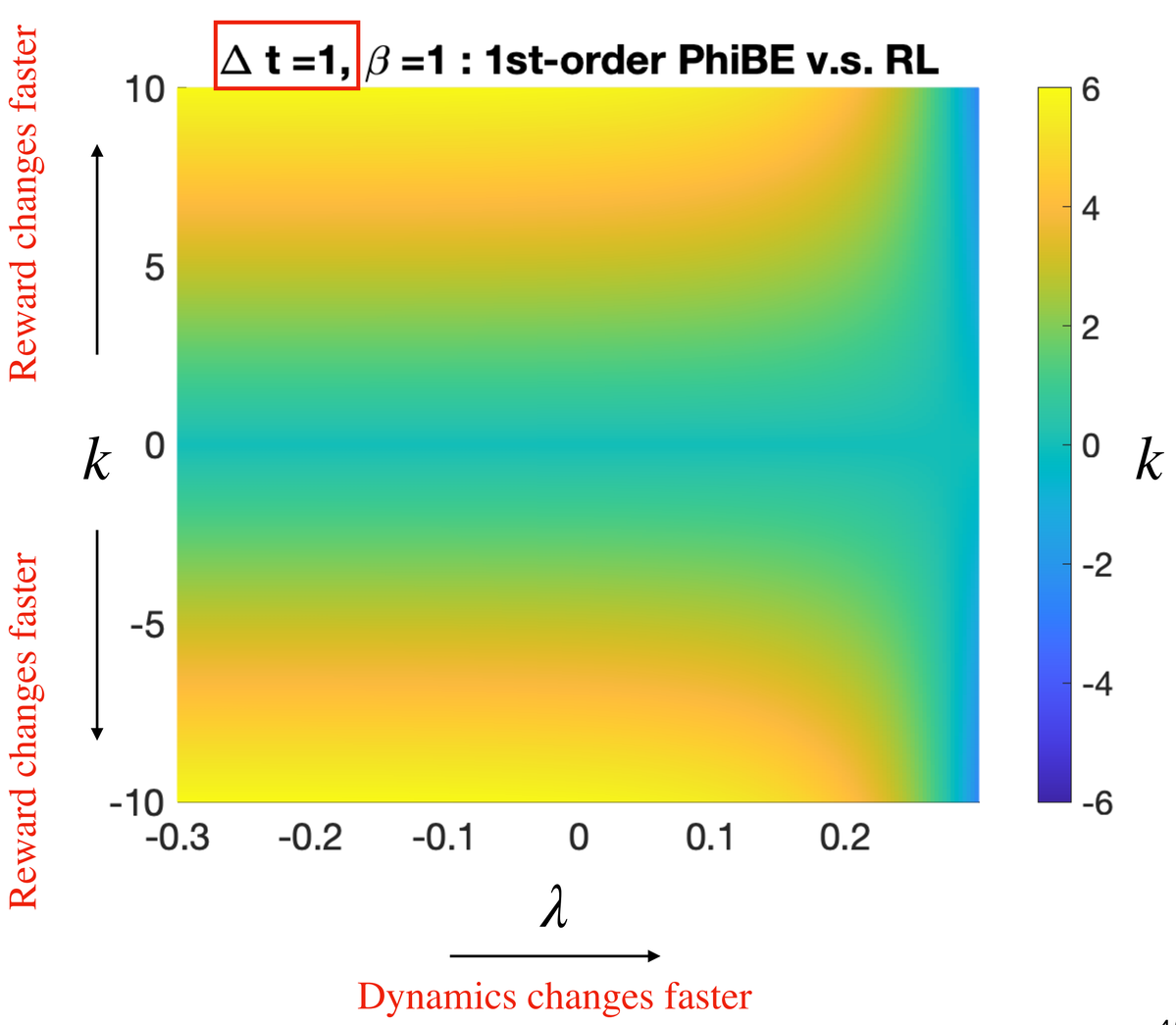
$$V(s) = \int_0^{\infty} e^{-\beta t} k s_t^2 dt, \quad \dot{s}_t = \lambda s_t$$

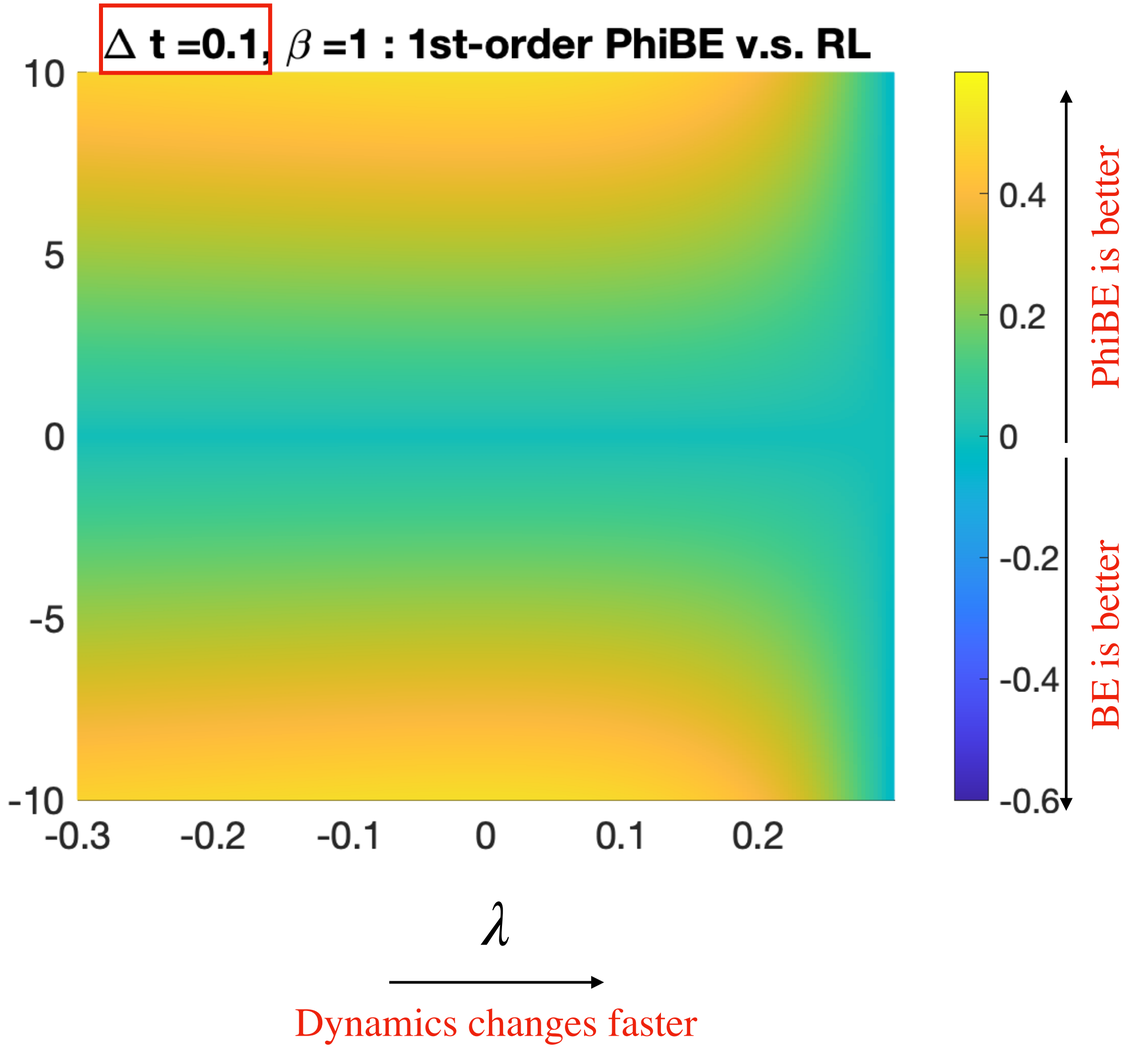




LQR

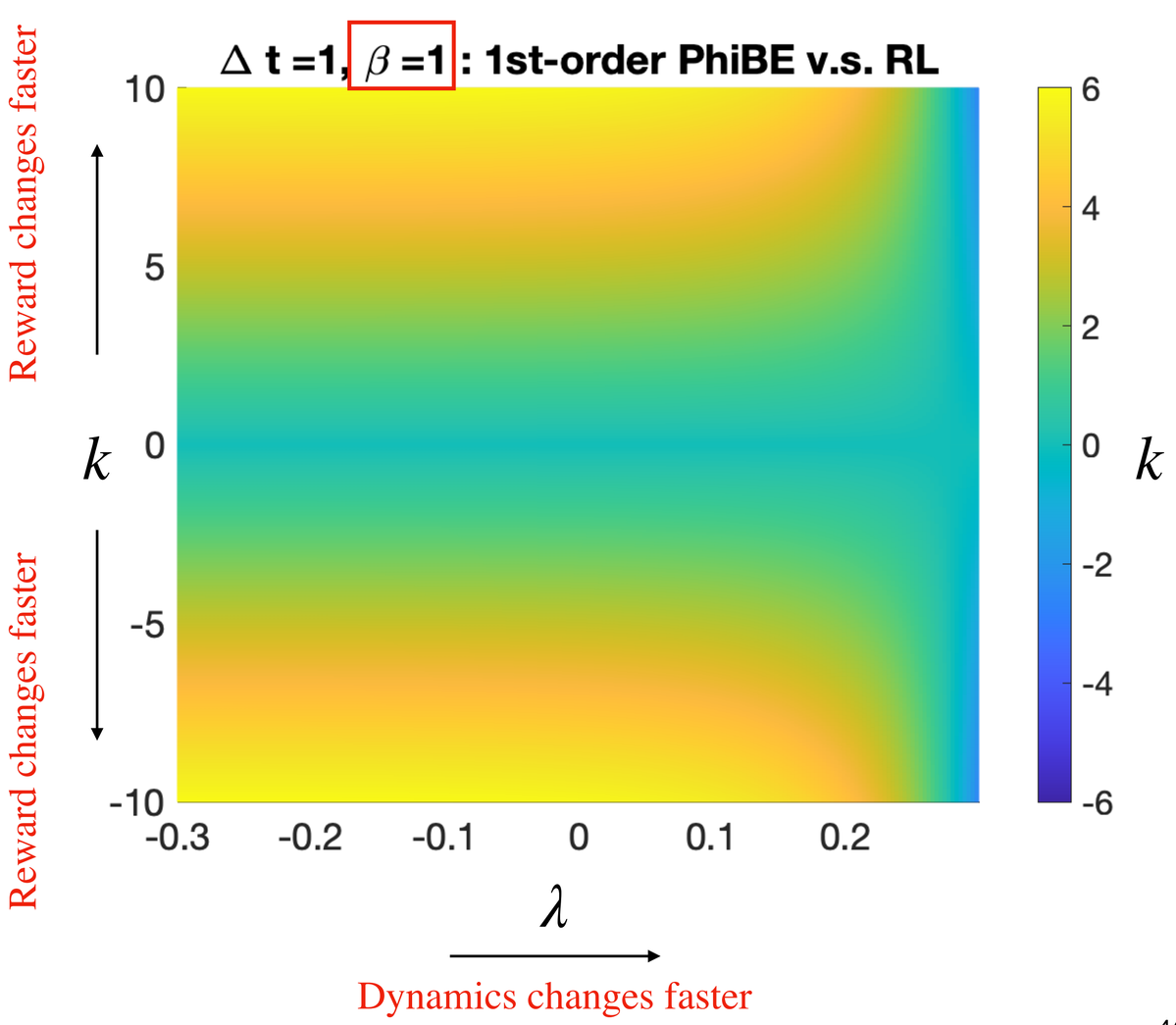
$$V(s) = \int_0^{\infty} e^{-\beta t} k s_t^2 dt, \quad \dot{s}_t = \lambda s_t$$

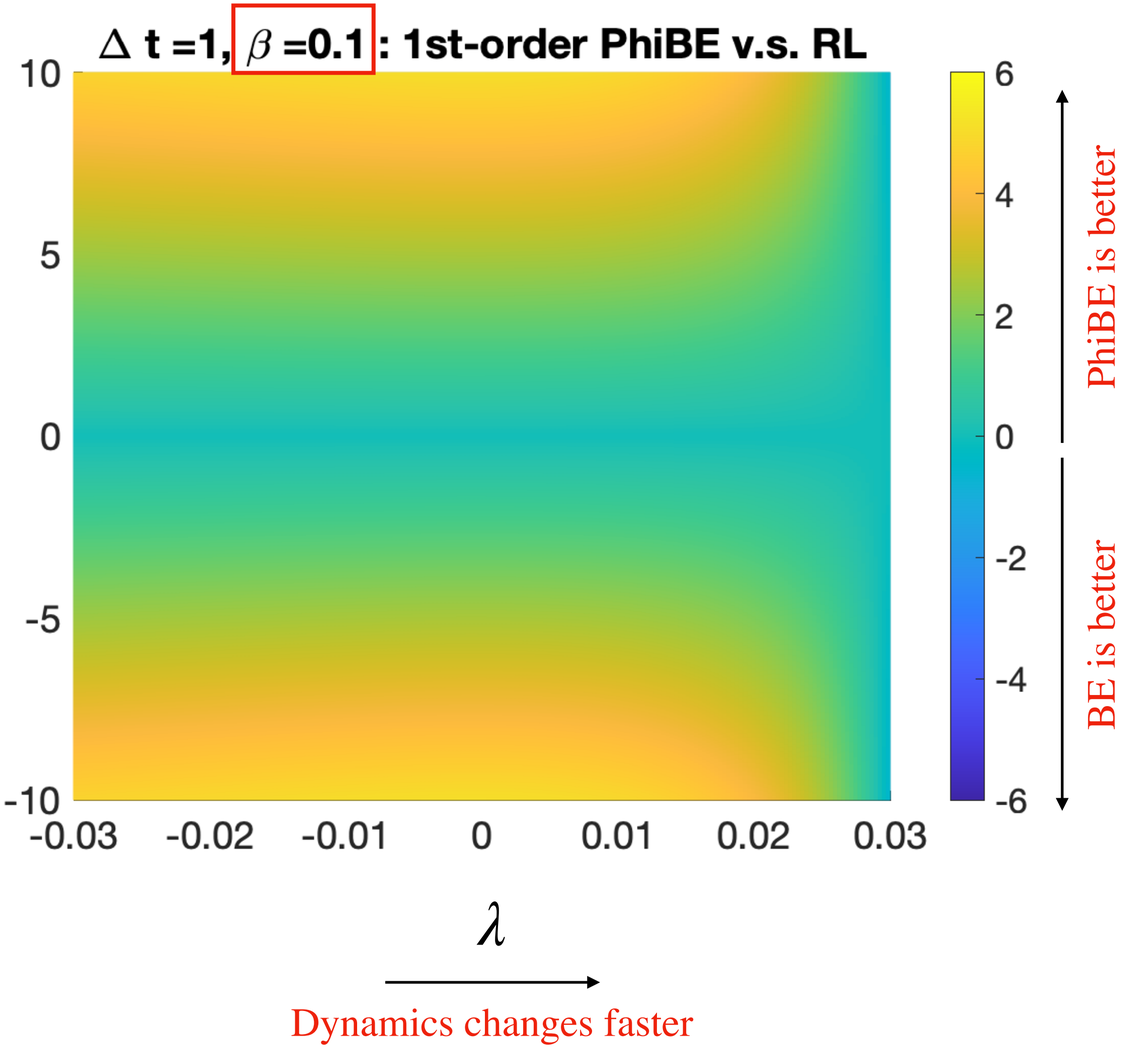




LQR

$$V(s) = \int_0^{\infty} e^{-\beta t} k s_t^2 dt, \quad \dot{s}_t = \lambda s_t$$





Schedule

Continuos-time RL —— Setting

Bellman equation ——Why it is not good

A PDE-based Bellman equation —— Why it is better

More numerical experiments

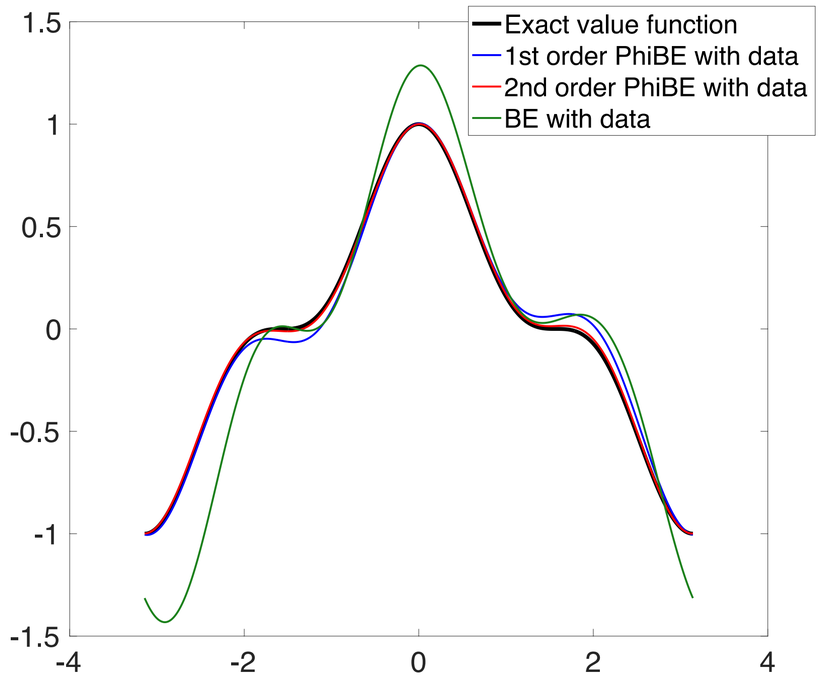
Given the same trajectory data: PhiBE v.s. BE

Underlying dynamics: $\frac{d}{dt}s_t = \lambda \sin(s_t)$

Given the same trajectory data: PhiBE v.s. BE

Underlying dynamics: $\frac{d}{dt}s_t = \lambda \sin(s_t)$

$\Delta t = 5, \beta = 0.1, \lambda = 0.05, V(s) = \cos^3(s)$

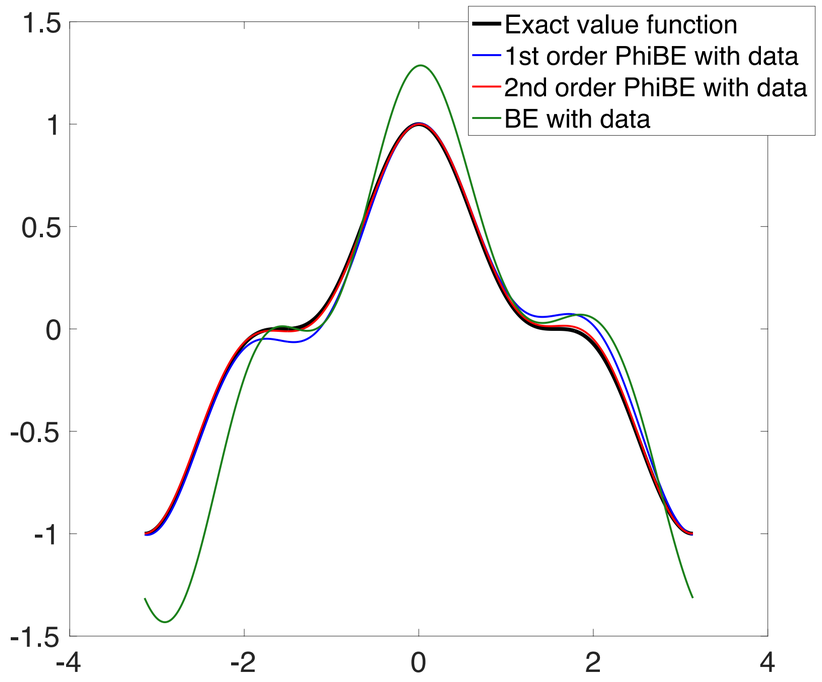


# of data: 40

Given the same trajectory data: PhiBE v.s. BE

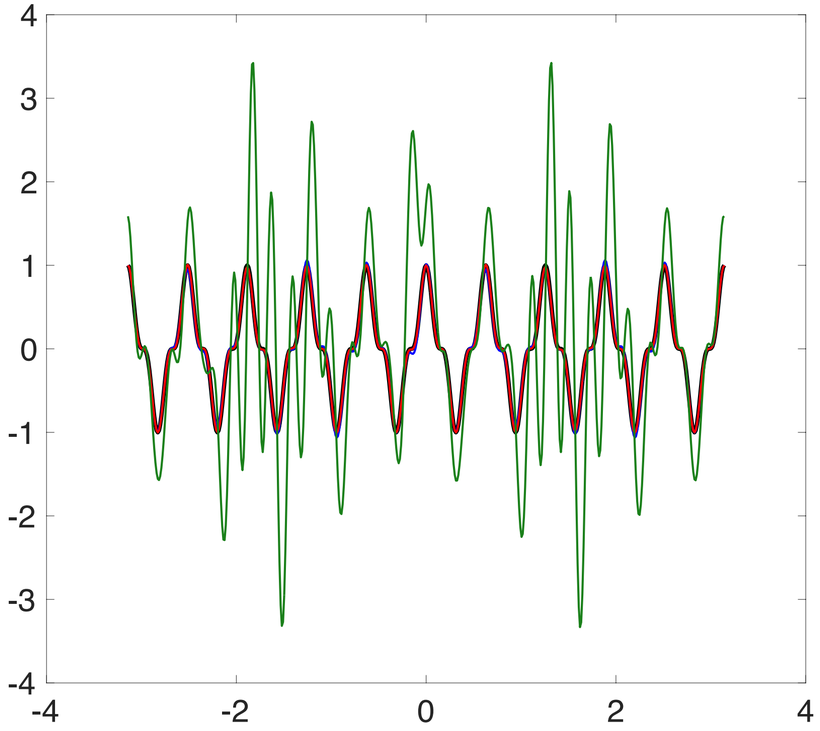
Underlying dynamics: $\frac{d}{dt}s_t = \lambda \sin(s_t)$

$\Delta t = 5, \beta = 0.1, \lambda = 0.05, V(s) = \cos^3(s)$



# of data: 40

$\Delta t = 0.1, \beta = 10, \lambda = 2, V(s) = \cos^3(10s)$

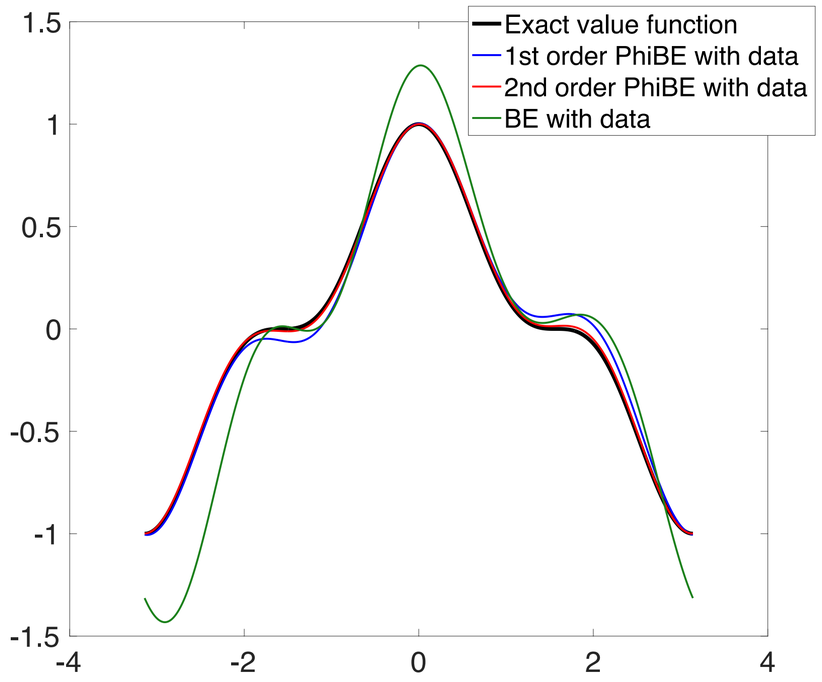


# of data: 400

Given the same trajectory data: PhiBE v.s. BE

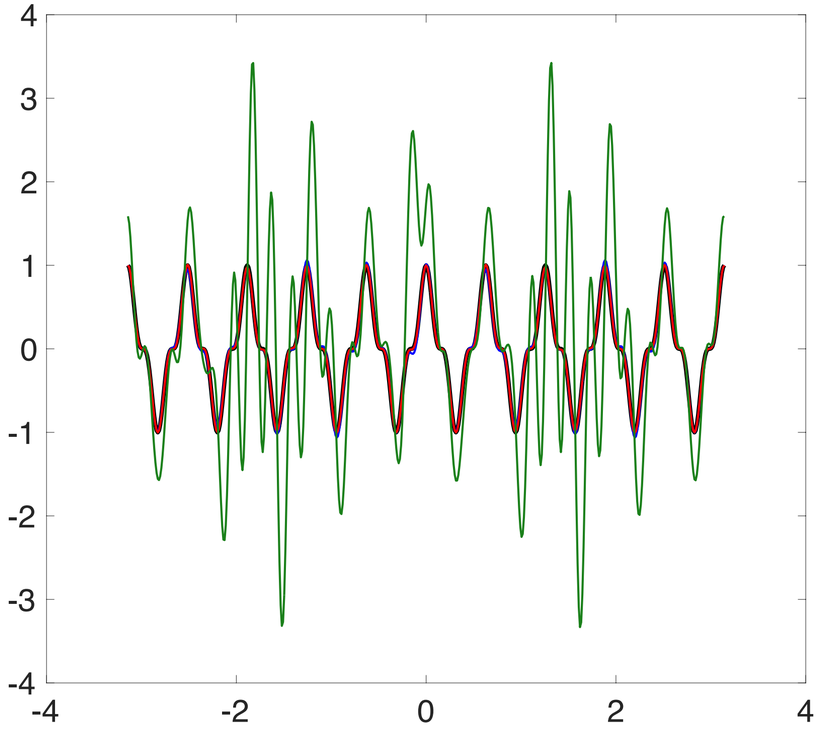
Underlying dynamics: $\frac{d}{dt}s_t = \lambda \sin(s_t)$

$\Delta t = 5, \beta = 0.1, \lambda = 0.05, V(s) = \cos^3(s)$



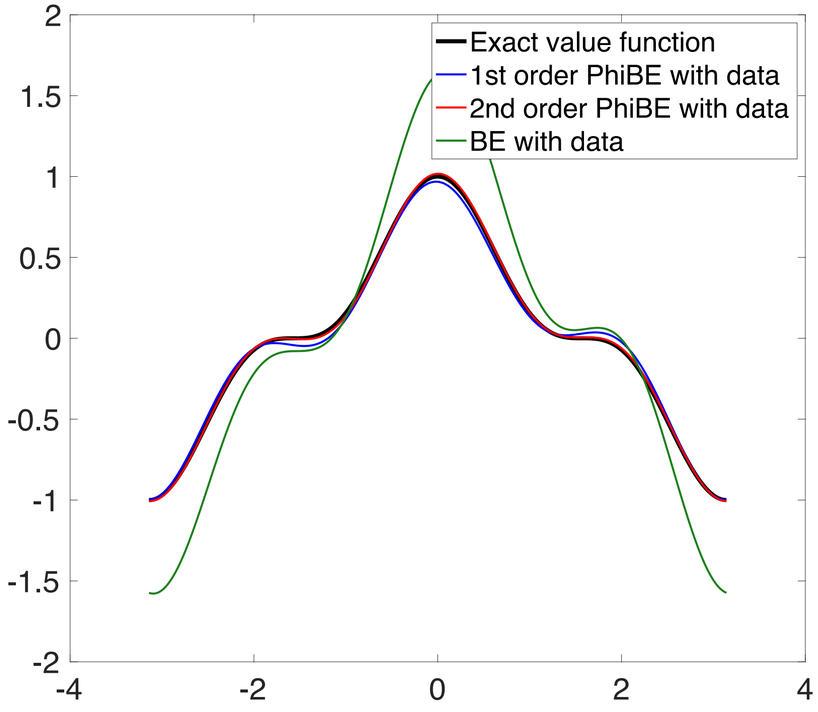
# of data: 40

$\Delta t = 0.1, \beta = 10, \lambda = 2, V(s) = \cos^3(10s)$



# of data: 400

$\Delta t = 0.1, \beta = 10, \lambda = 5, V(s) = \cos^3(s)$



# of data: 40

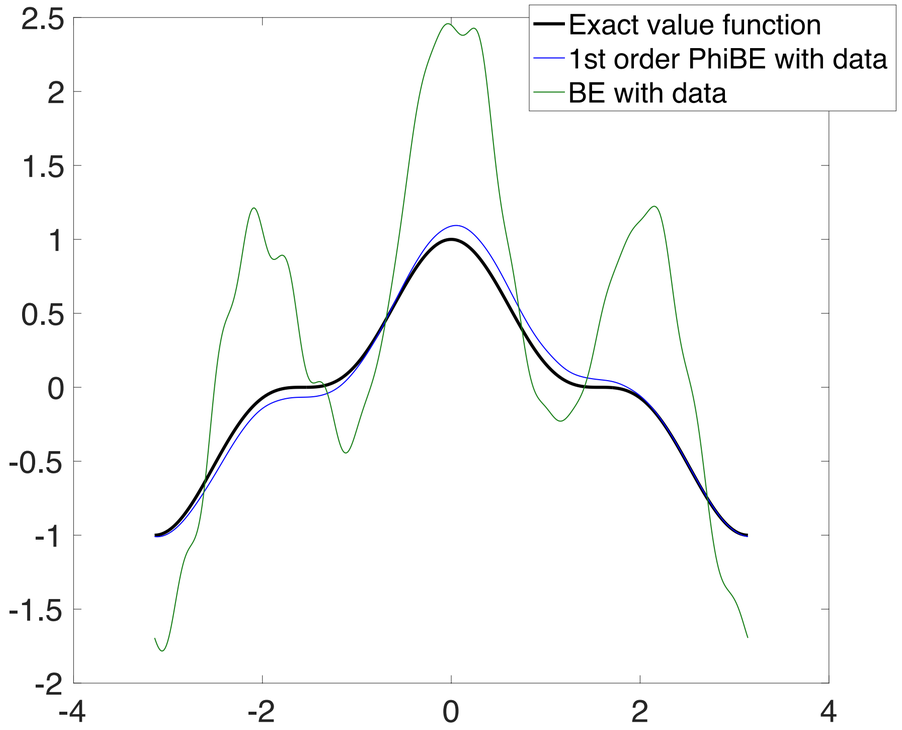
Given the same trajectory data: PhiBE v.s. BE

Underlying dynamics: $ds_t = 0.05 s_t + dB_t$

Given the same trajectory data: PhiBE v.s. BE

Underlying dynamics: $ds_t = 0.05 s_t + dB_t$

$\Delta t = 1, \ \beta = 0.1, V(s) = \cos^3(s)$

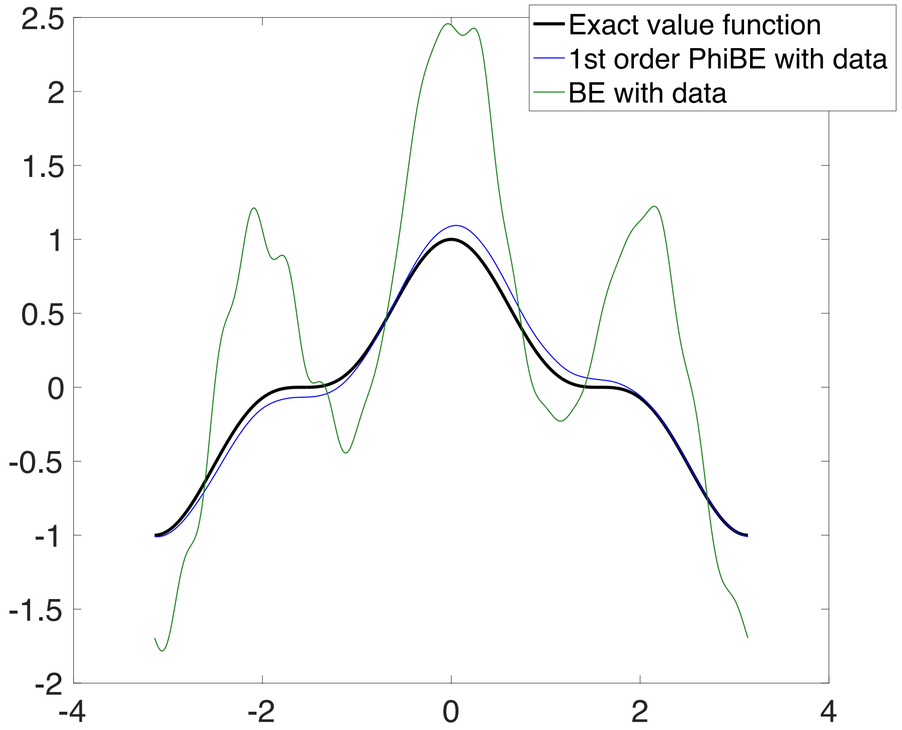


# of data: 4×10^3

Given the same trajectory data: PhiBE v.s. BE

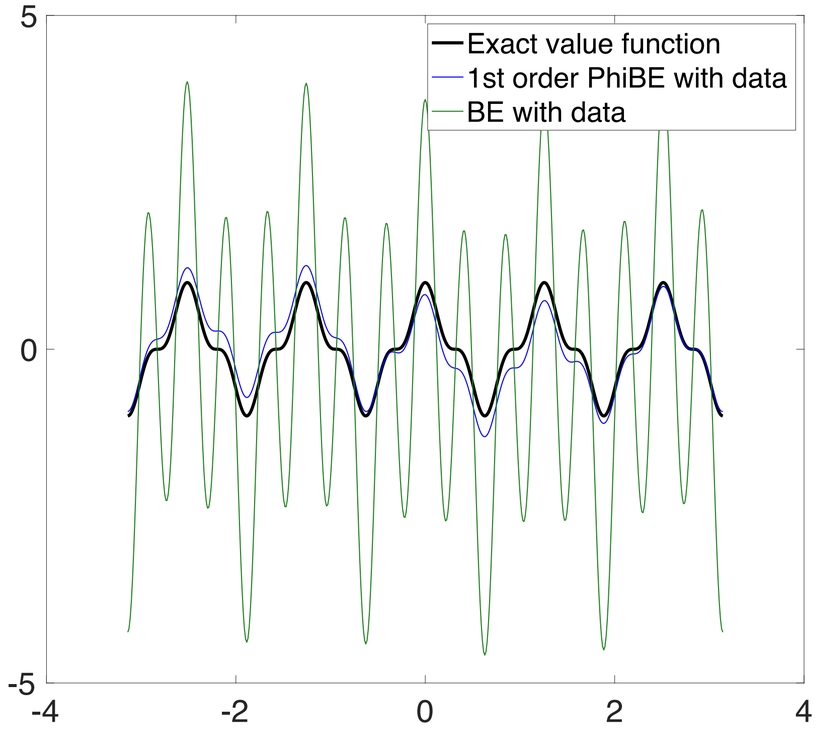
Underlying dynamics: $ds_t = 0.05 s_t + dB_t$

$\Delta t = 1,\ \beta = 0.1, V(s) = \cos^3(s)$



# of data: 4×10^3

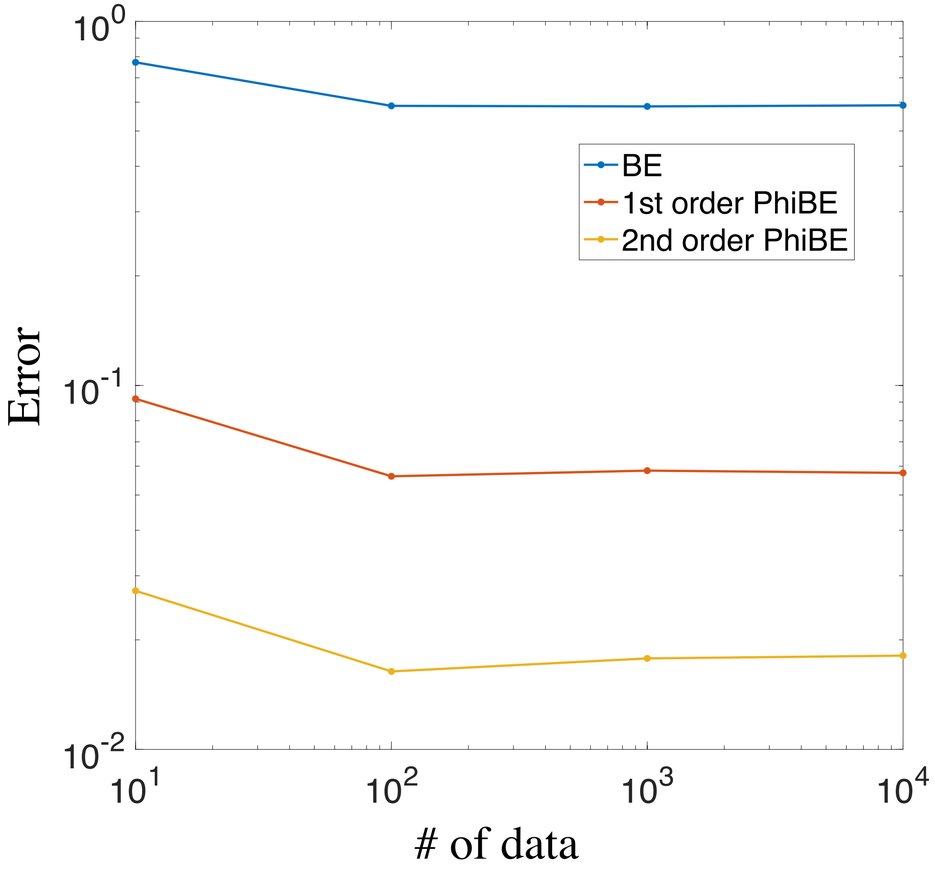
$\Delta t = 0.1,\ \beta = 0.1,\ V(s) = \cos^3(5s)$



# of data: 4×10^5

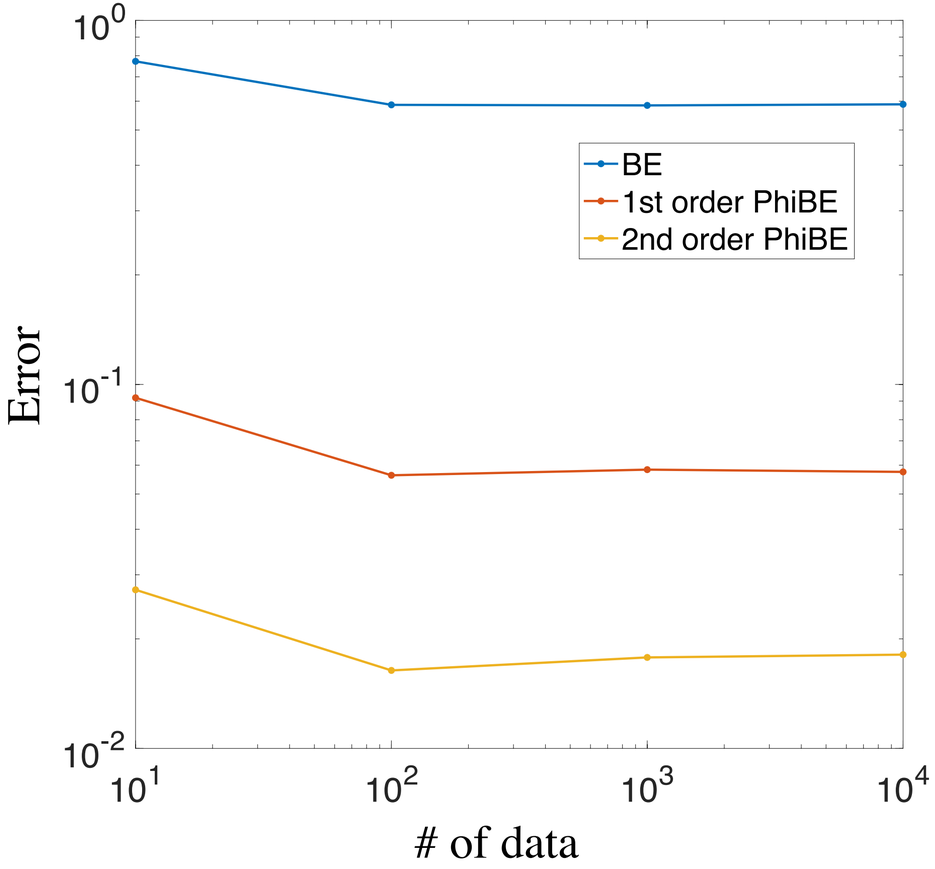
Decrease of error as # of data increases

Underlying dynamics: $\frac{d}{dt}s_t = 0.05 s_t$

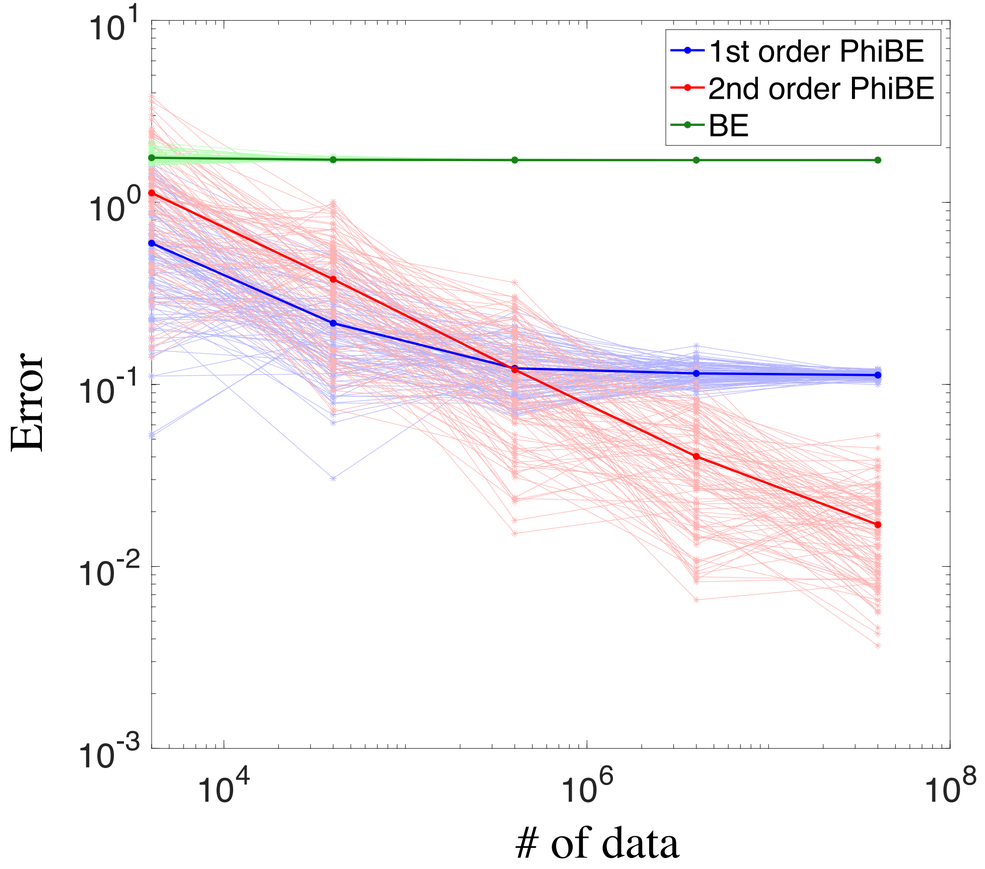


Decrease of error as # of data increases

Underlying dynamics: $\frac{d}{dt}s_t = 0.05 s_t$



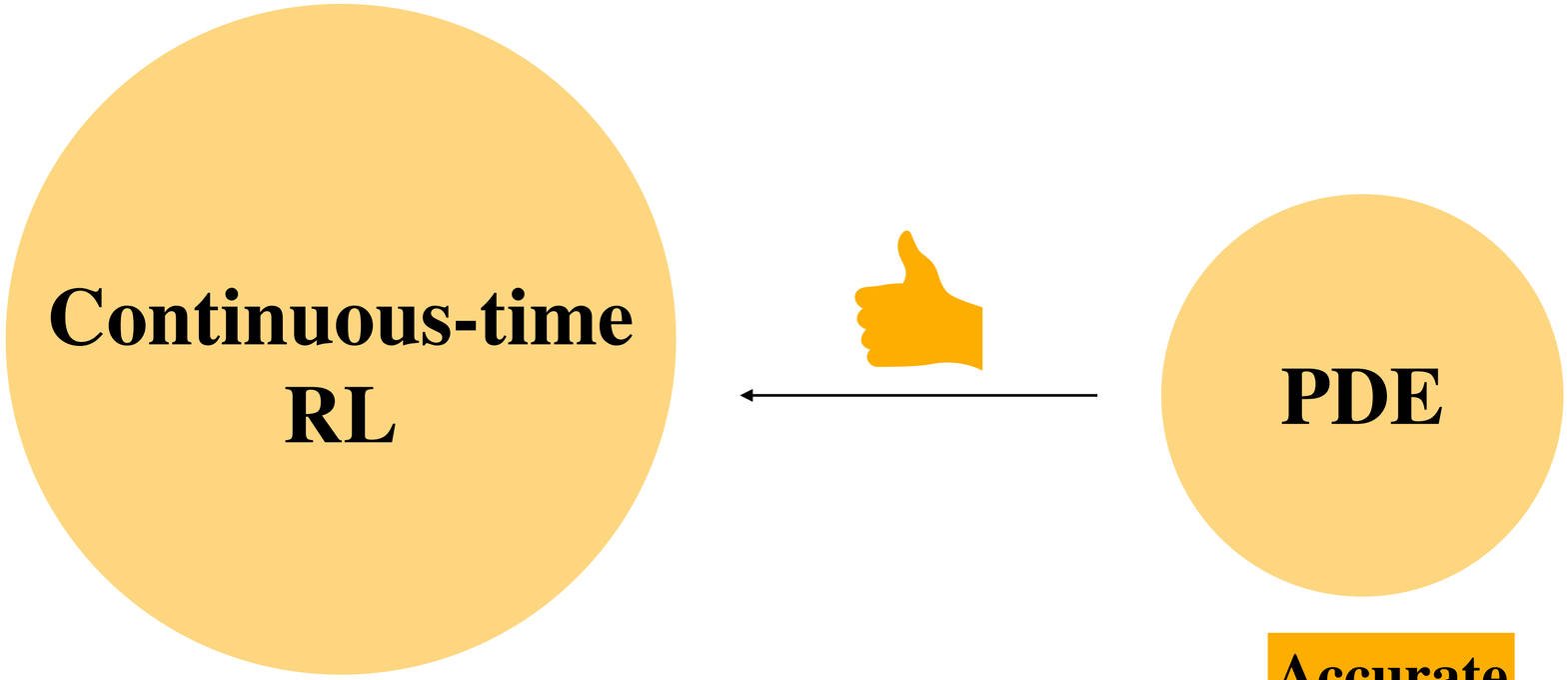
Underlying dynamics: $ds_t = 0.05 s_t + dB_t$



Conclusion

Continuous-time RL

Conclusion



Conclusion

Reference

Yuhua Zhu, 2024, PhiBE: A PDE-based Bellman Equation for Continuous-Time Policy Evaluation

Ongoing work

- Generalize to continuous-time RL
- Nonlinear approximation
- Time dependent dynamics
- Finite Horizon problems

Future work

- Multi-agent RL
- Mean-field game
- Application in robotics, autonomous driving and finance
- …