Incremental Least-Squares Temporal Difference Learning

Alborz Geramifard
December, 2006
alborz@cs.ualberta.ca

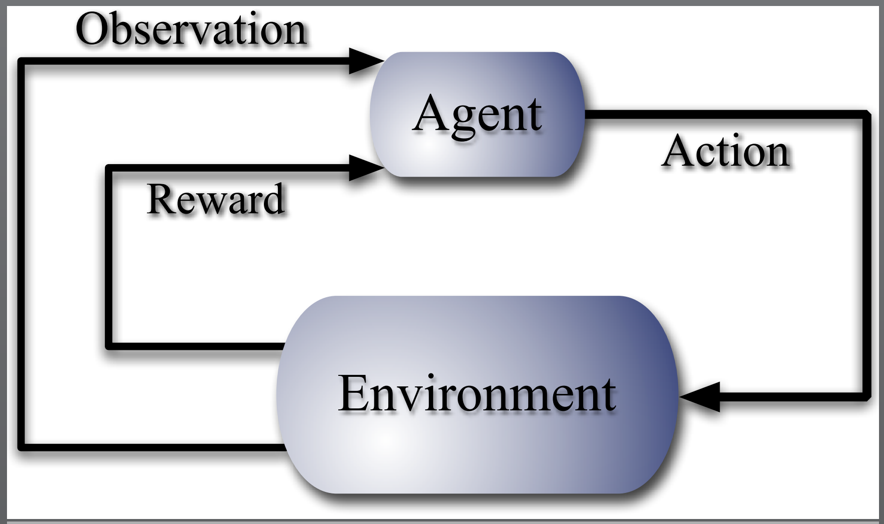
Incremental Least-Squares Temporal Difference Learning

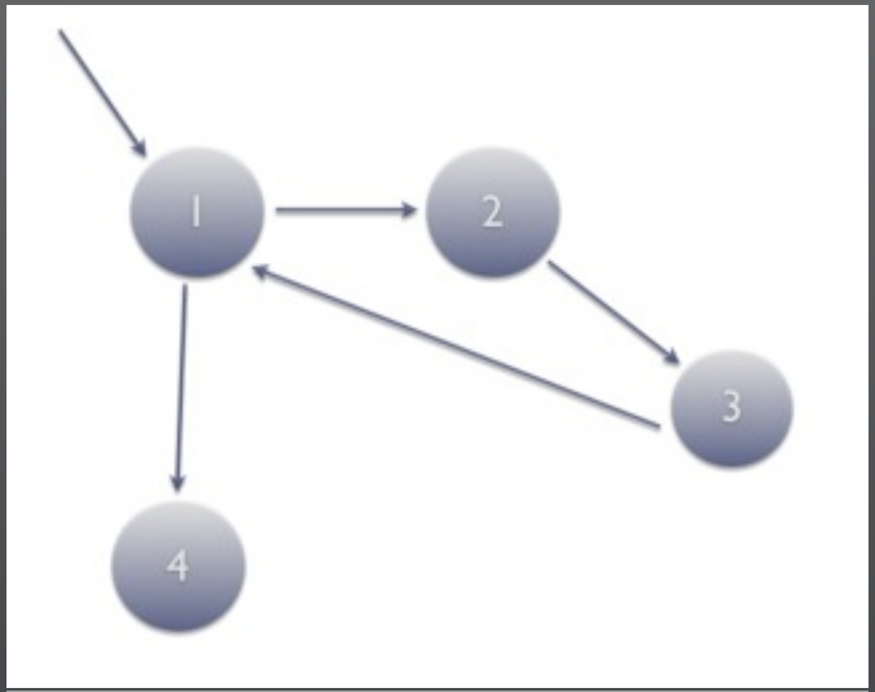
Alborz Geramifard
December, 2006
alborz@cs.ualberta.ca

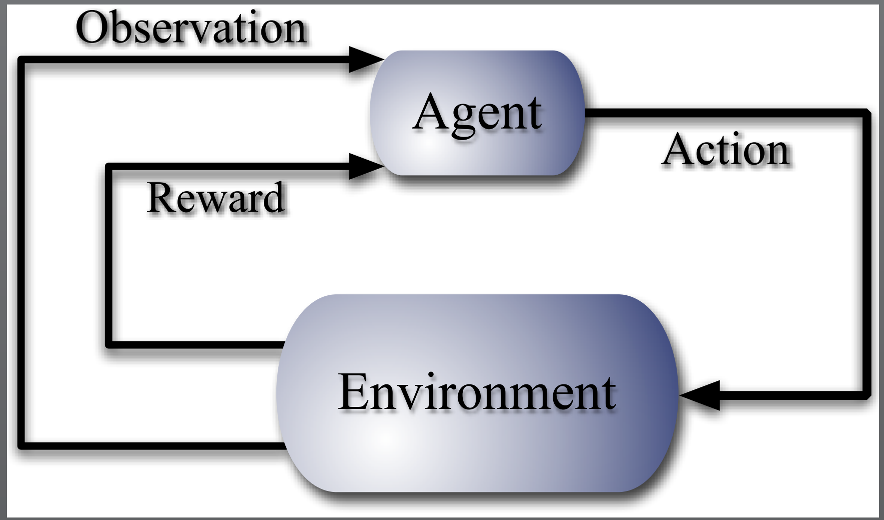
UNIVERSITY OF ALBERTA
QUAECUMQUE VERA

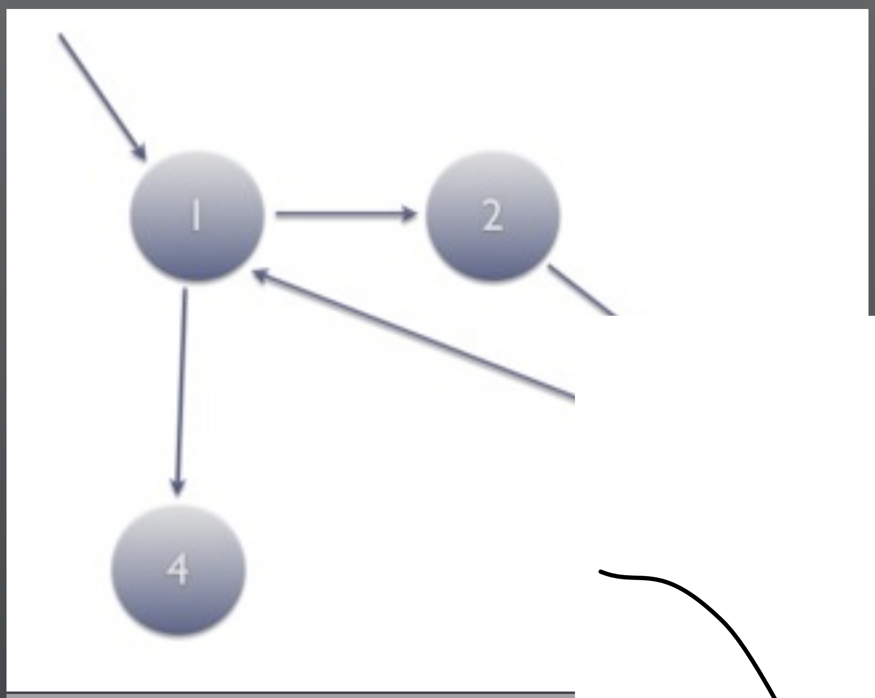
RL & AI

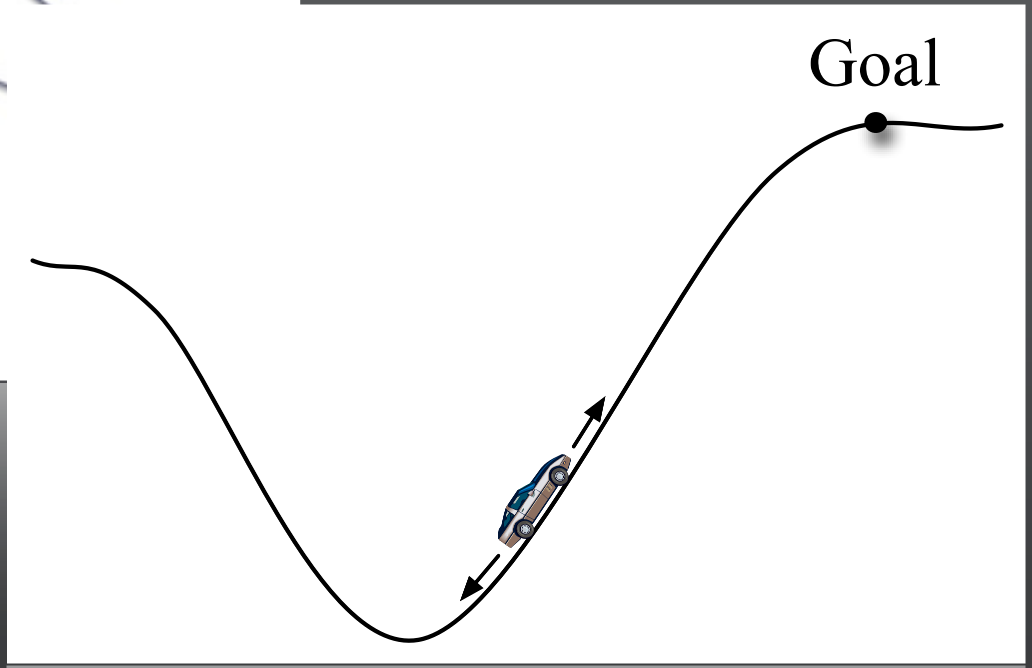

Observation
Agent
Action
Reward
Environment

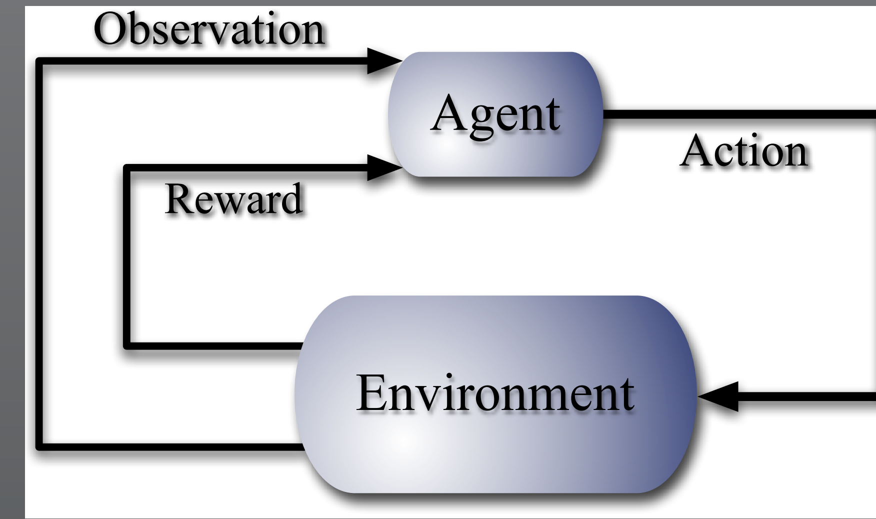

Observation
Agent
Action
Reward
Environment

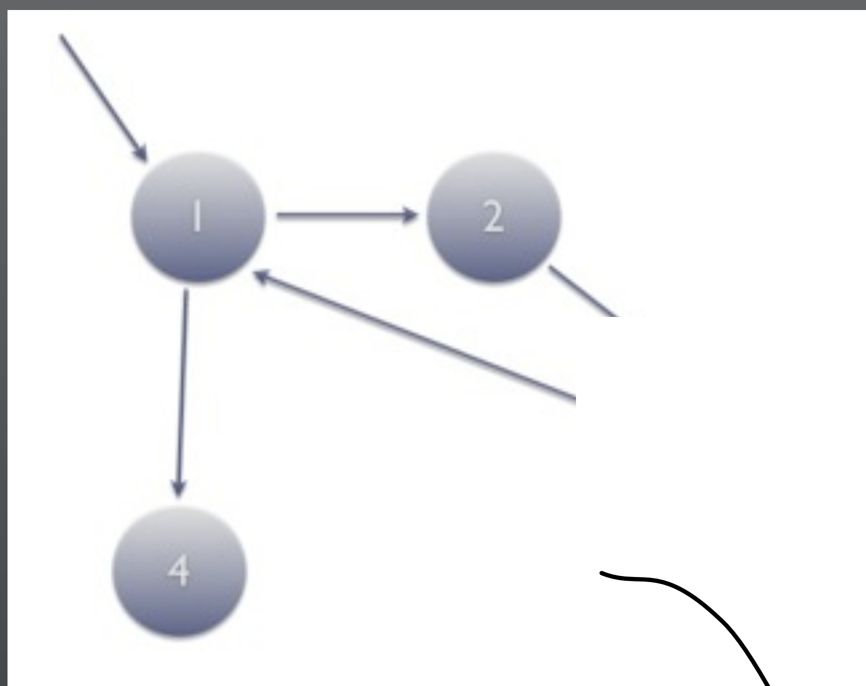

1
2
3
4

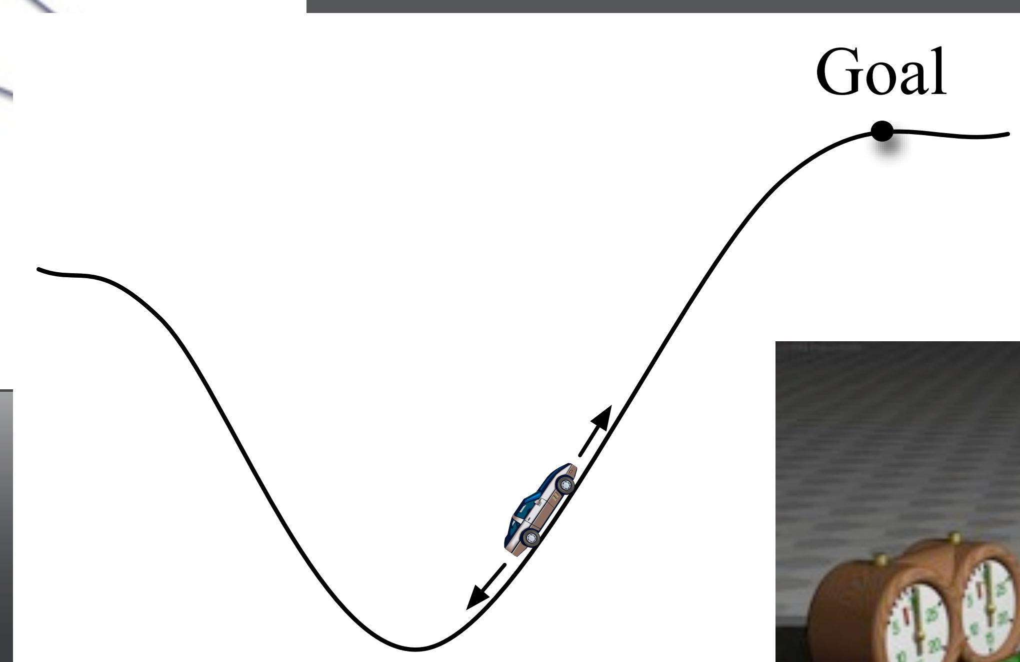

Observation
Agent
Action
Reward
Environment

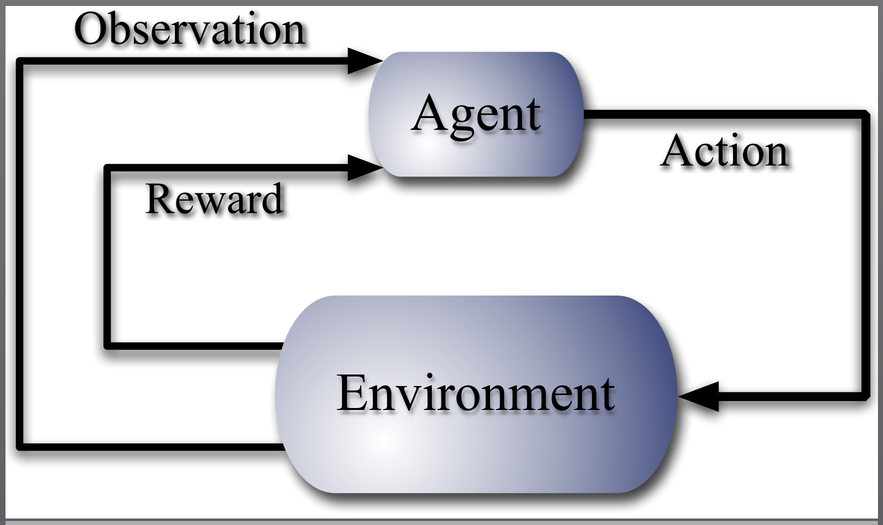

1
2
4


Goal

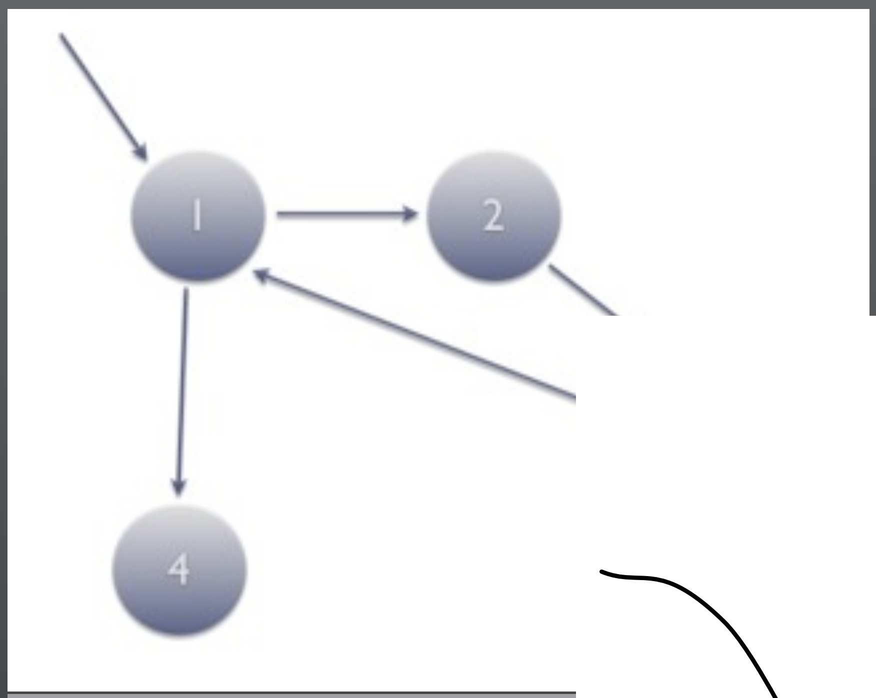

Observation
Agent
Action
Reward
Environment

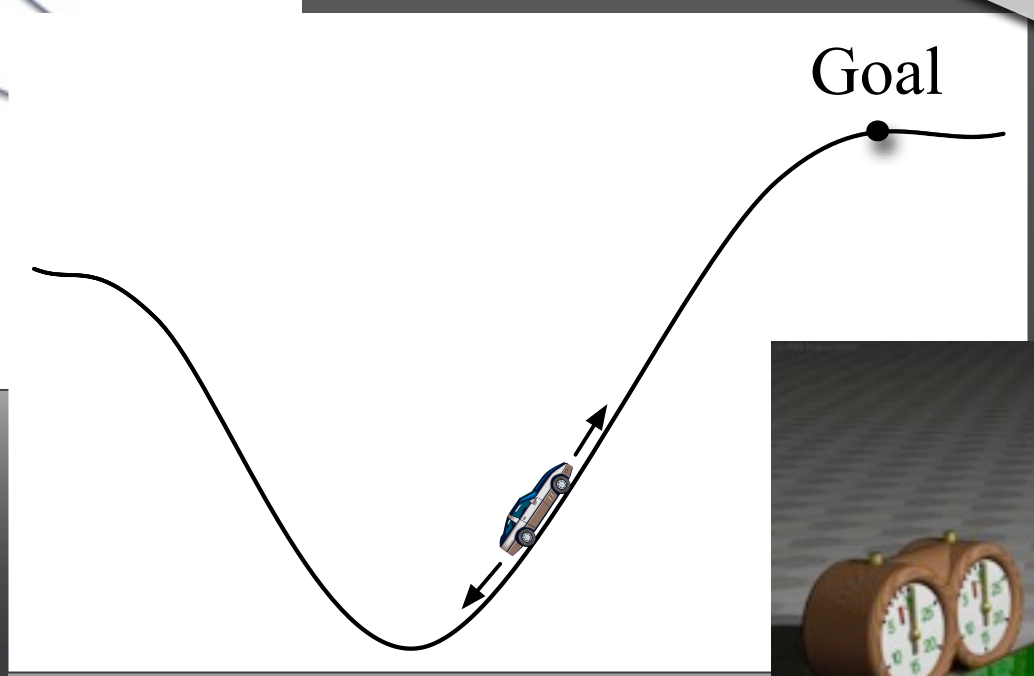

1
2
4


Goal



More Complicated State Space

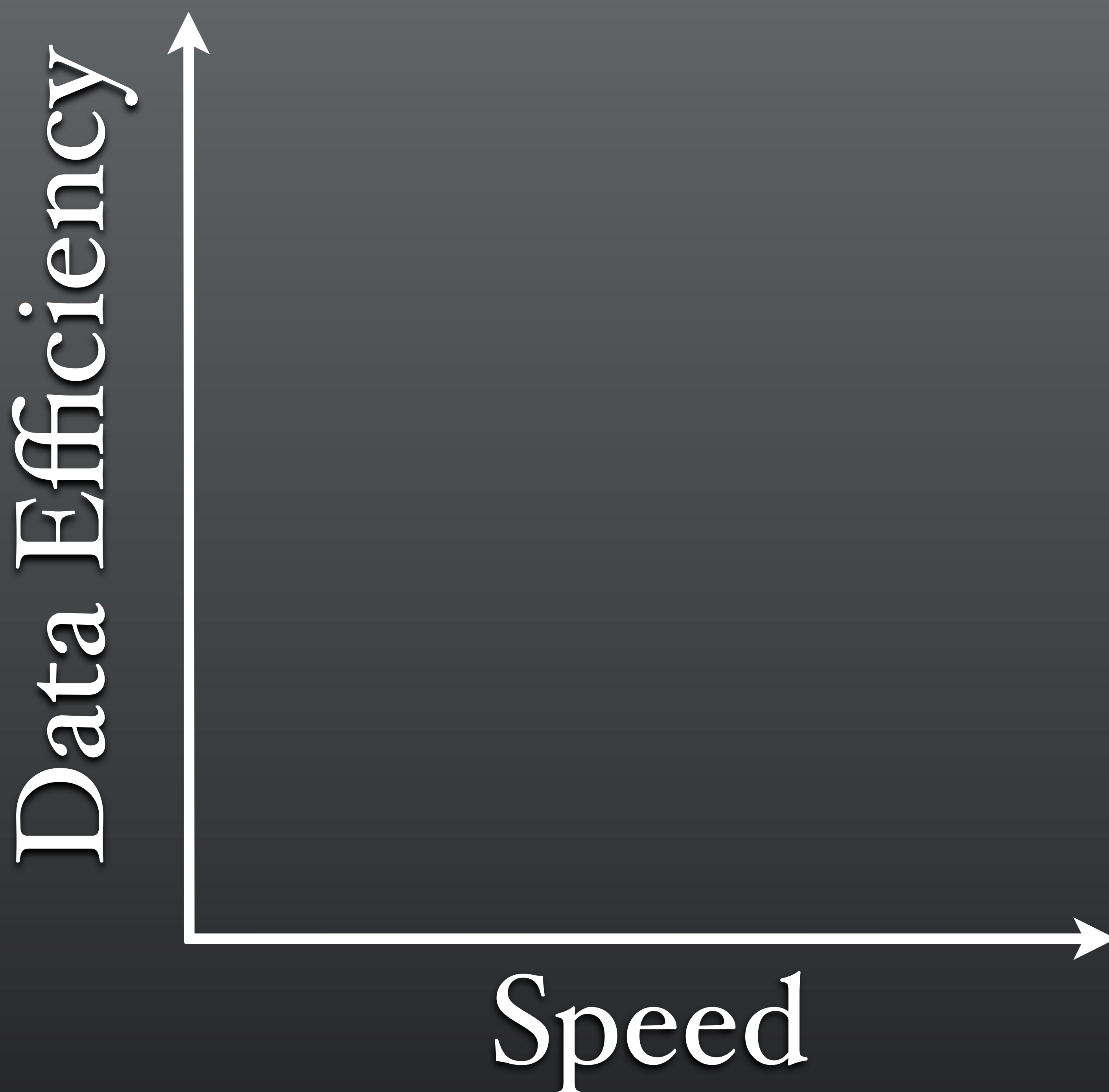




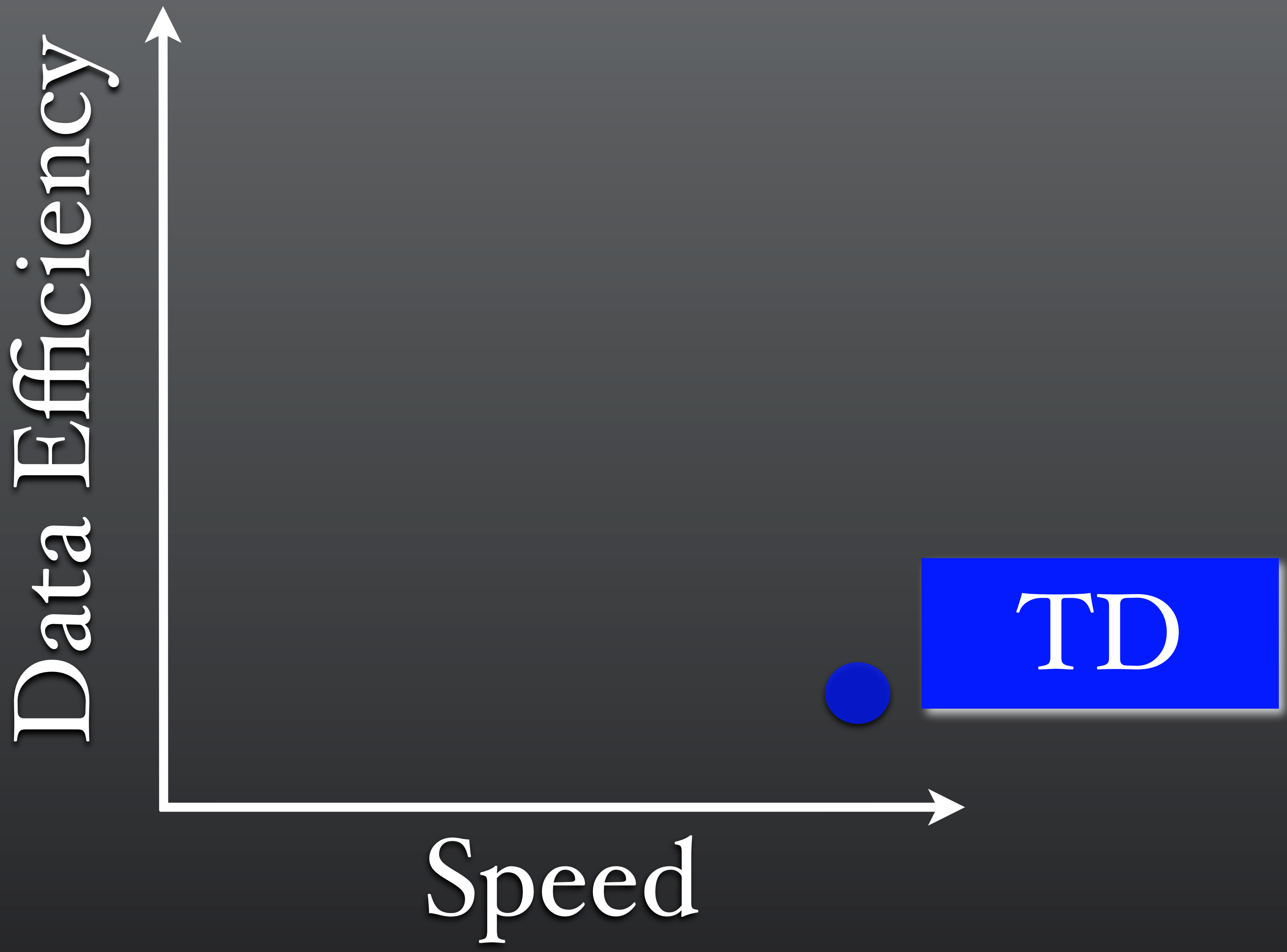
More Complicated Tasks

Summary

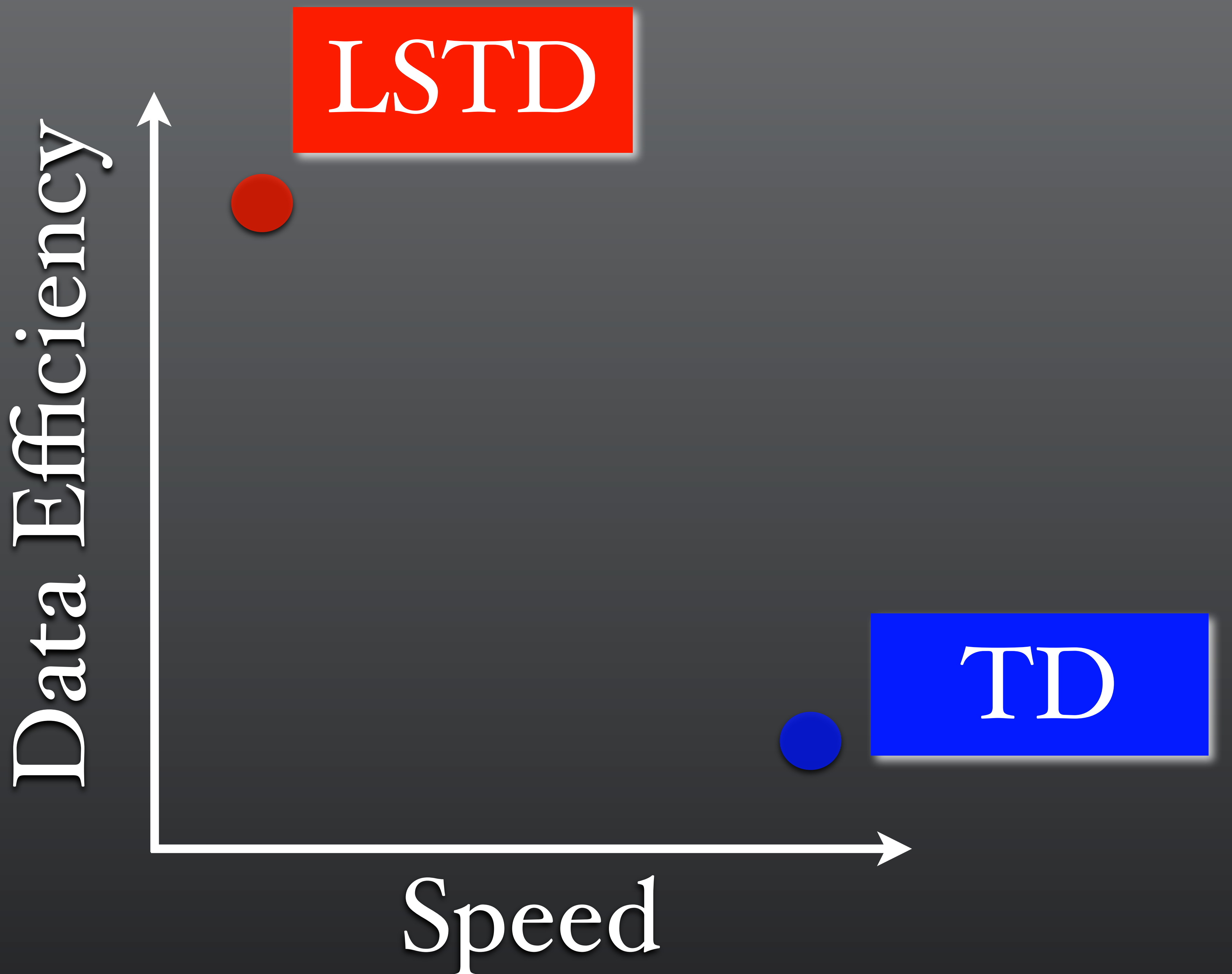
Summary



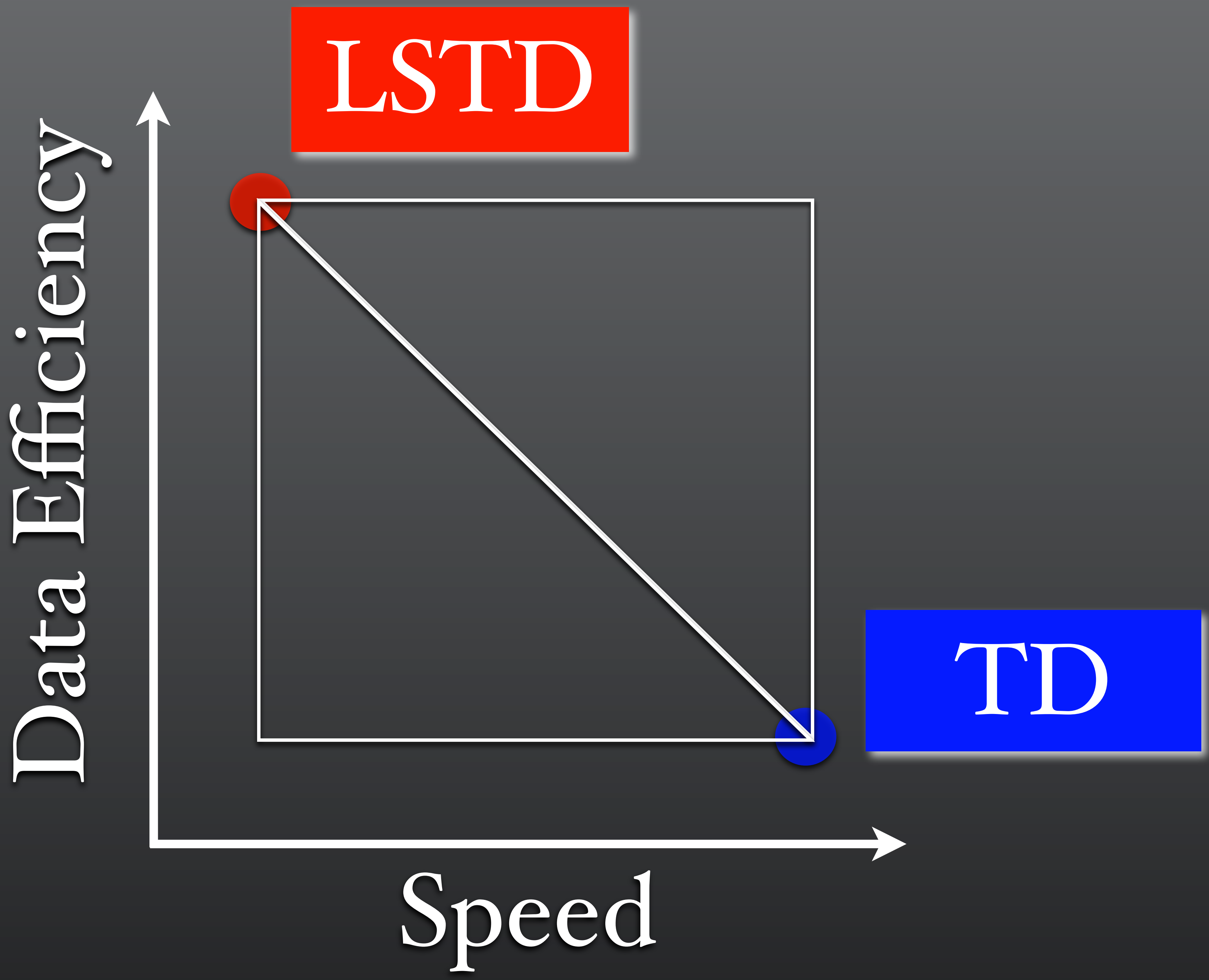
Summary



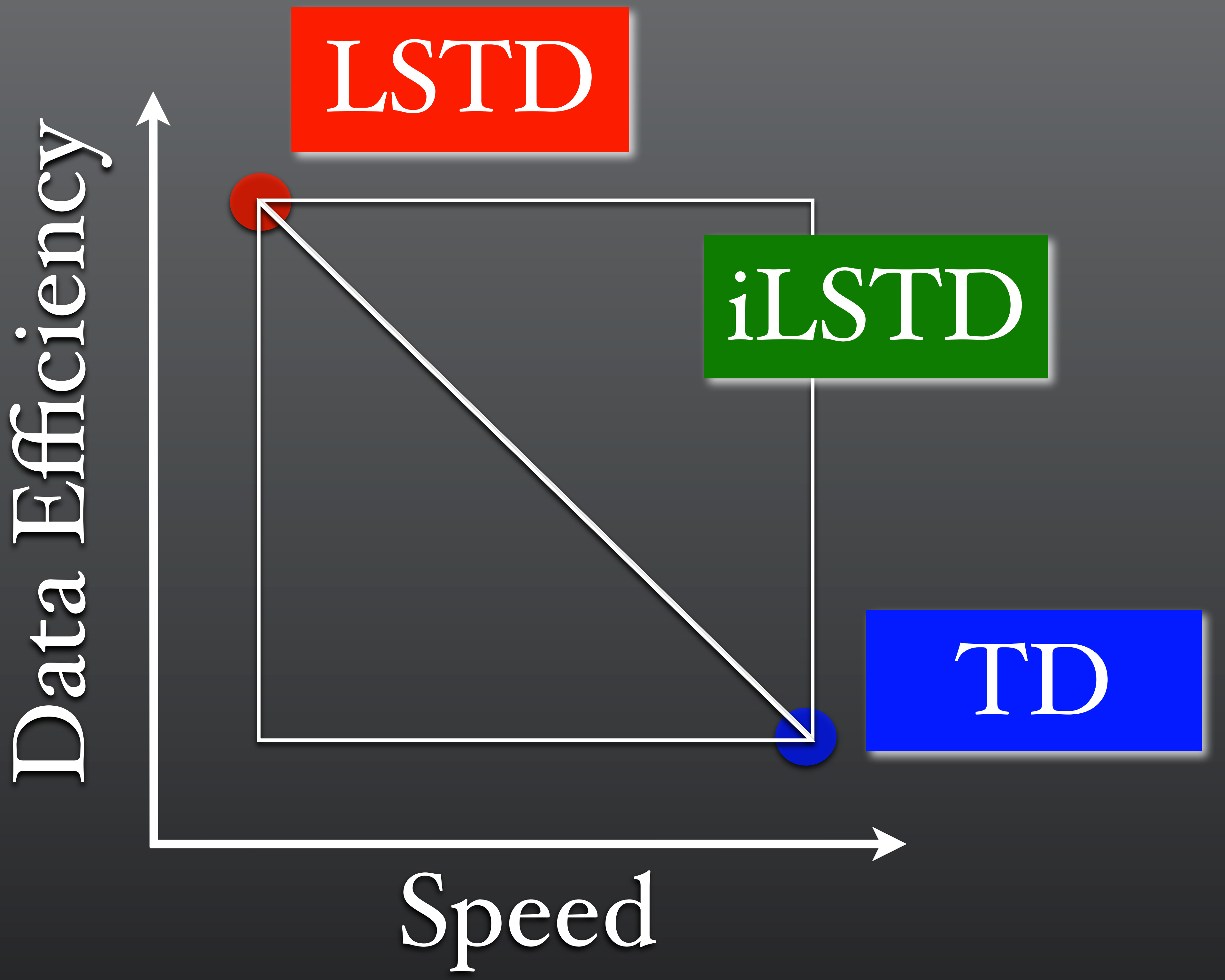
Summary



Summary



Summary



Contributions

Contributions

- iLSTD: A new policy evaluation algorithm
- Extension with eligibility traces
- Running time analysis
- Dimension selection methods
- Proof of convergence
- Empirical results

Contributions

- iLSTD: A new policy evaluation algorithm
- Extension with eligibility traces
- Running time analysis
- Dimension selection methods
- Proof of convergence
- Empirical results

Outline

Outline

- Motivation
- Introduction
- The New Approach
- Eligibility Traces
- Dimension Selection
- Conclusion

Outline

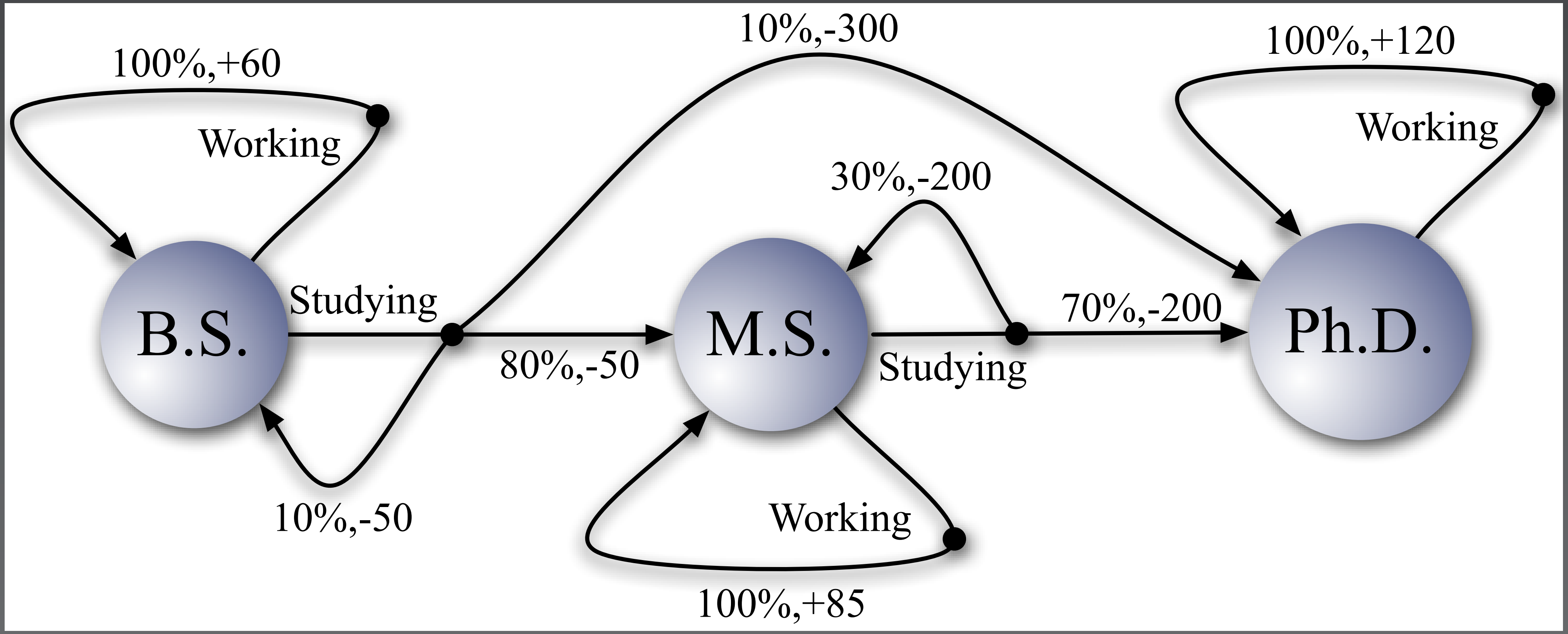
- Motivation
- Introduction ←
- The New Approach
- Eligibility Traces
- Dimension Selection
- Conclusion

Markov Decision Process

$$\langle \mathcal{S}, \mathcal{A}, \mathcal{P}^a_{ss'}, \mathcal{R}^a_{ss'}, \gamma \rangle$$

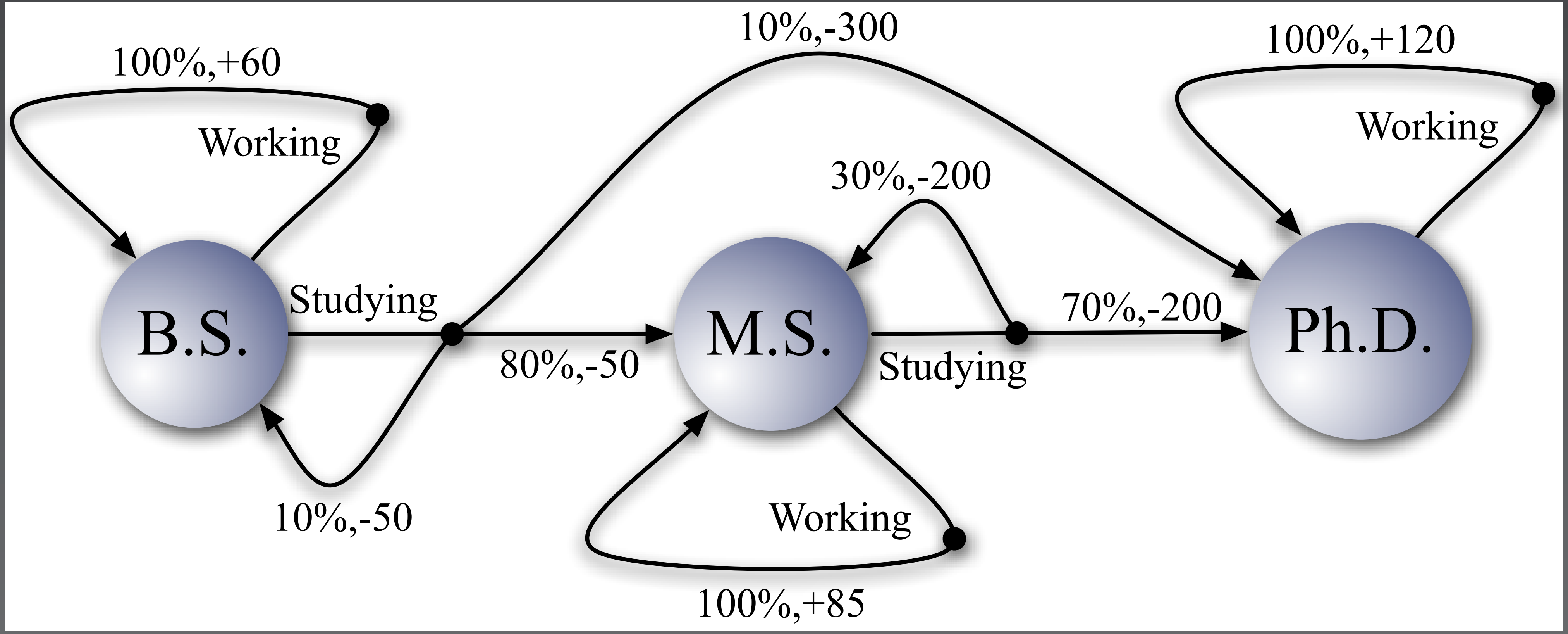
Markov Decision Process

$$\langle \mathcal{S}, \mathcal{A}, \mathcal{P}^a_{ss'}, \mathcal{R}^a_{ss'}, \gamma \rangle$$



Markov Decision Process

$$\langle \mathcal{S}, \mathcal{A}, \mathcal{P}^a_{ss'}, \mathcal{R}^a_{ss'}, \gamma \rangle$$



B.S., Working, +60, B.S. Studying, -50, M.S. , ...

Policy Evaluation

Policy Evaluation

↓

Policy Improvement

Policy Evaluation

Policy Improvement


Policy Evaluation
Policy Improvement

Notation

Scalar	Regular	$V^\pi(s)$	r_{t+1}
Vector	Bold Lower Case	$\boldsymbol{\phi}(s)$	$\boldsymbol{\mu}(\boldsymbol{\theta})$
Matrix	Bold Upper Case	$\mathbf{A}_t$	$\tilde{\mathbf{A}}$

Policy Evaluation

$$V^{\pi}(s) = E\left[\sum_{t=1}^{\infty} \gamma^{t-1} r_t \middle| s_0 = s, \pi\right]$$

Linear Function Approximation

$$V(s) = \boldsymbol{\theta} \cdot \boldsymbol{\phi}(s) = \sum_{i=1}^{n} \theta_i \phi_i(s)$$

Sparsity of features

- Sparsity: Only k features are active at any given moment.

$$k \ll n$$

Sparsity of features

- Sparsity: Only k features are active at any given moment.

$$k \ll n$$

  - Acrobot [Sutton *96*]: 48 << 18,648

Sparsity of features

- Sparsity: Only k features are active at any given moment.

$$k \ll n$$

  - Acrobot [Sutton *96*]: 48 << 18,648
  - Card game [Bowling *et al.* 02]: 3 << 10^6

Sparsity of features

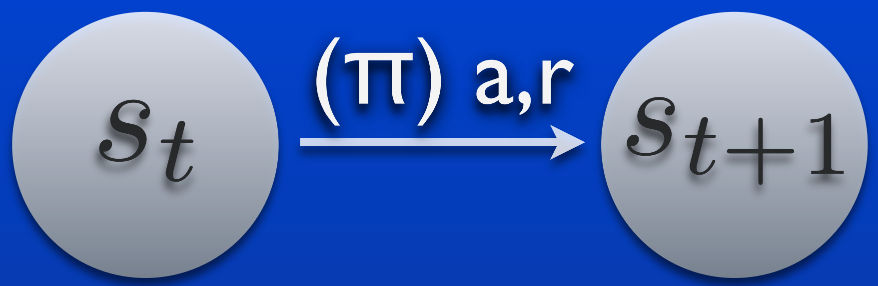
- Sparsity: Only k features are active at any given moment.

$$k \ll n$$

  - Acrobot [Sutton *96*]: 48 << 18,648
  - Card game [Bowling *et al.* 02]: 3 << 10^6
  - Keep away soccer [Stone *et al.* 05]: 416 << 10^4

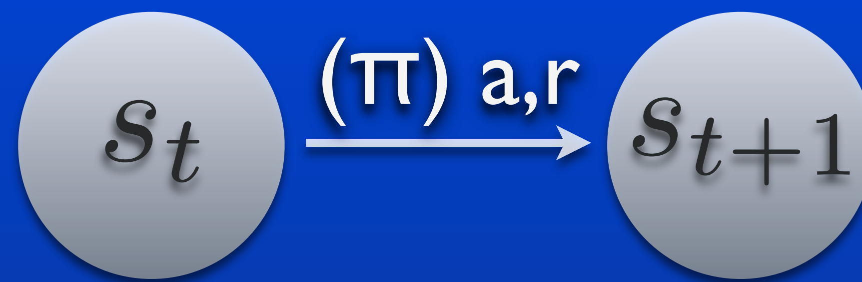
Temporal Difference Learning
TD(0)

Temporal Difference Learning TD(0)



[Sutton 88]

Temporal Difference Learning
TD(0)



- Tabular Representation

$$\delta_t = r_t + \gamma V(s_{t+1}) - V(s_t).$$

- Linear Function Approximation

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \alpha_t \boldsymbol{\phi}(s_t)\delta_t(V)$$

[Sutton 88]

TD(0) Properties

- Computational complexity

$$O(k) \text{ per time step}$$

- Data inefficient
  - Only last transition

TD(0) Properties

- Computational complexity

Constant

$$O(k) \text{ per time step}$$

- Data inefficient
  - Only last transition

Least-Squares TD (LSTD)

- Sum of TD updates

Least-Squares TD (LSTD)

[Bradtke, Barto 96]

- Sum of TD updates

Least-Squares TD (LSTD)

[Bradtke, Barto 96]

- Sum of TD updates

$$\boldsymbol{\mu}_t(\boldsymbol{\theta}) = \sum_{i=1}^{t} \phi_i \delta_i(V_{\boldsymbol{\theta}})$$

Least-Squares TD (LSTD)

[Bradtke, Barto 96]

- Sum of TD updates

$$\boldsymbol{\mu}_t(\boldsymbol{\theta}) = \sum_{i=1}^{t} \phi_i \delta_i(V_{\boldsymbol{\theta}})$$

$$= \underbrace{\sum_{i=1}^{t} \phi_i r_{i+1}}_{\mathbf{b}_t} - \underbrace{\sum_{i=1}^{t} \phi_i (\phi_i - \gamma \phi_{i+1})^T}_{\mathbf{A}_t} \boldsymbol{\theta}$$

Least-Squares TD (LSTD)

[Bradtke, Barto 96]

- Sum of TD updates

$$\mu_t(\boldsymbol{\theta}) = \sum_{i=1}^{t} \phi_i \delta_i(V_{\boldsymbol{\theta}})$$

$$= \underbrace{\sum_{i=1}^{t} \phi_i r_{i+1}}_{\mathbf{b}_t} - \underbrace{\sum_{i=1}^{t} \phi_i (\phi_i - \gamma \phi_{i+1})^T}_{\mathbf{A}_t} \boldsymbol{\theta}$$

$$= \mathbf{b}_t - \mathbf{A}_t \boldsymbol{\theta}$$

Least-Squares TD (LSTD)

[Bradtke, Barto 96]

- Sum of TD updates

$$\mu_t(\boldsymbol{\theta}) = \sum_{i=1}^{t} \phi_i \delta_i(V_{\boldsymbol{\theta}})$$

$$= \underbrace{\sum_{i=1}^{t} \phi_i r_{i+1}}_{\mathbf{b}_t} - \underbrace{\sum_{i=1}^{t} \phi_i(\phi_i - \gamma\phi_{i+1})^T}_{\mathbf{A}_t} \boldsymbol{\theta}$$

$$= \mathbf{b}_t - \mathbf{A}_t \boldsymbol{\theta}$$

$$\mu_t(\boldsymbol{\theta}) = \mathbf{0} \quad \longrightarrow \quad \boldsymbol{\theta} = \mathbf{A}^{-1}\mathbf{b}$$

LSTD Properties

- Computational complexity

$$O(n^2) \text{ per time step}$$

- Data efficient
  - Look through all data

LSTD Properties

- Computational complexity

Quadratic

$$O(n^2) \text{ per time step}$$

- Data efficient
  - Look through all data

Outline

Outline

- Motivation
- Introduction
- The New Approach ←
- Eligibility Traces
- Dimension Selection
- Conclusion

The New Approach

The New Approach

- **A** and **b** matrices change on each iteration.

The New Approach

- **A** and **b** matrices change on each iteration.

$$\boldsymbol{\mu}_t(\boldsymbol{\theta}) = \underbrace{\sum_{i=1}^{t} \phi_i r_{i+1}}_{\mathbf{b}_t} - \underbrace{\sum_{i=1}^{t} \phi_i(\phi_i - \gamma\phi_{i+1})^T}_{\mathbf{A}_t} \boldsymbol{\theta}$$

The New Approach

- **A** and **b** matrices change on each iteration.

$$\boldsymbol{\mu}_t(\boldsymbol{\theta}) = \underbrace{\sum_{i=1}^{t} \phi_i r_{i+1}}_{\mathbf{b}_t} - \underbrace{\sum_{i=1}^{t} \phi_i (\phi_i - \gamma \phi_{i+1})^T}_{\mathbf{A}_t} \boldsymbol{\theta}$$

$$\mathbf{b}_t = \mathbf{b}_{t-1} + \underbrace{r_t \phi_t}_{\Delta \mathbf{b}_t}$$

$$\mathbf{A}_t = \mathbf{A}_{t-1} + \underbrace{\phi_t (\phi_t - \gamma \phi_{t+1})^T}_{\Delta \mathbf{A}_t}$$

Incremental LSTD

$$\boldsymbol{\mu}_t(\boldsymbol{\theta}) = \mathbf{b}_t - \mathbf{A}_t\boldsymbol{\theta}$$

Incremental LSTD

$$\boldsymbol{\mu}_t(\boldsymbol{\theta}) = \mathbf{b}_t - \mathbf{A}_t\boldsymbol{\theta}$$

[Geramifard, Bowling, Sutton 06]

Incremental LSTD

$$\mu_t(\theta) = \mathbf{b}_t - \mathbf{A}_t\theta$$

- Fixed $\boldsymbol{\theta}$

- Fixed $\mathbf{A}$ and $\mathbf{b}$

[Geramifard, Bowling, Sutton 06]

Incremental LSTD

$$\boldsymbol{\mu}_t(\boldsymbol{\theta}) = \mathbf{b}_t - \mathbf{A}_t\boldsymbol{\theta}$$

- Fixed **θ**

$$\boldsymbol{\mu}_t(\boldsymbol{\theta}) = \boldsymbol{\mu}_{t-1}(\boldsymbol{\theta}) + \Delta\mathbf{b}_t - (\Delta\mathbf{A}_t)\boldsymbol{\theta}.$$

- Fixed **A** and **b**

[Geramifard, Bowling, Sutton 06]

Incremental LSTD

$$\boldsymbol{\mu}_t(\boldsymbol{\theta}) = \mathbf{b}_t - \mathbf{A}_t\boldsymbol{\theta}$$

- Fixed **θ**

$$\boldsymbol{\mu}_t(\boldsymbol{\theta}) = \boldsymbol{\mu}_{t-1}(\boldsymbol{\theta}) + \Delta\mathbf{b}_t - (\Delta\mathbf{A}_t)\boldsymbol{\theta}.$$

- Fixed **A** and **b**

$$\boldsymbol{\mu}_t(\boldsymbol{\theta}_{t+1}) = \boldsymbol{\mu}_t(\boldsymbol{\theta}_t) - \mathbf{A}_t(\Delta\boldsymbol{\theta}_t).$$

[Geramifard, Bowling, Sutton 06]

iLSTD

$$\boldsymbol{\mu}_t(\boldsymbol{\theta}_{t+1}) = \boldsymbol{\mu}_t(\boldsymbol{\theta}_t) - \mathbf{A}_t(\Delta\boldsymbol{\theta}_t).$$

iLSTD

$$\boldsymbol{\mu}_t(\boldsymbol{\theta}_{t+1}) = \boldsymbol{\mu}_t(\boldsymbol{\theta}_t) - \mathbf{A}_t(\Delta\boldsymbol{\theta}_t).$$

- How to change **θ** ?

iLSTD

$$\boldsymbol{\mu}_t(\boldsymbol{\theta}_{t+1}) = \boldsymbol{\mu}_t(\boldsymbol{\theta}_t) - \mathbf{A}_t(\Delta\boldsymbol{\theta}_t).$$

- How to change **θ ?**
  - Descent in the direction of $\boldsymbol{\mu}_t(\boldsymbol{\theta})$?

iLSTD

$$\boldsymbol{\mu}_t(\boldsymbol{\theta}_{t+1}) = \boldsymbol{\mu}_t(\boldsymbol{\theta}_t) - \mathbf{A}_t(\Delta\boldsymbol{\theta}_t).$$

- How to change **θ** ?
  - Descent in the direction of $\boldsymbol{\mu}_t(\boldsymbol{\theta})$?

iLSTD

$$\boldsymbol{\mu}_t(\boldsymbol{\theta}_{t+1}) = \boldsymbol{\mu}_t(\boldsymbol{\theta}_t) - \mathbf{A}_t(\Delta\boldsymbol{\theta}_t).$$

- How to change **θ** ?
  - Descent in the direction of $\boldsymbol{\mu}_t(\boldsymbol{\theta})$?

iLSTD

$$\boldsymbol{\mu}_t(\boldsymbol{\theta}_{t+1}) = \boldsymbol{\mu}_t(\boldsymbol{\theta}_t) - \mathbf{A}_t(\Delta\boldsymbol{\theta}_t).$$

- How to change **θ** ?
  - Descent in the direction of $\boldsymbol{\mu}_t(\boldsymbol{\theta})$?

iLSTD

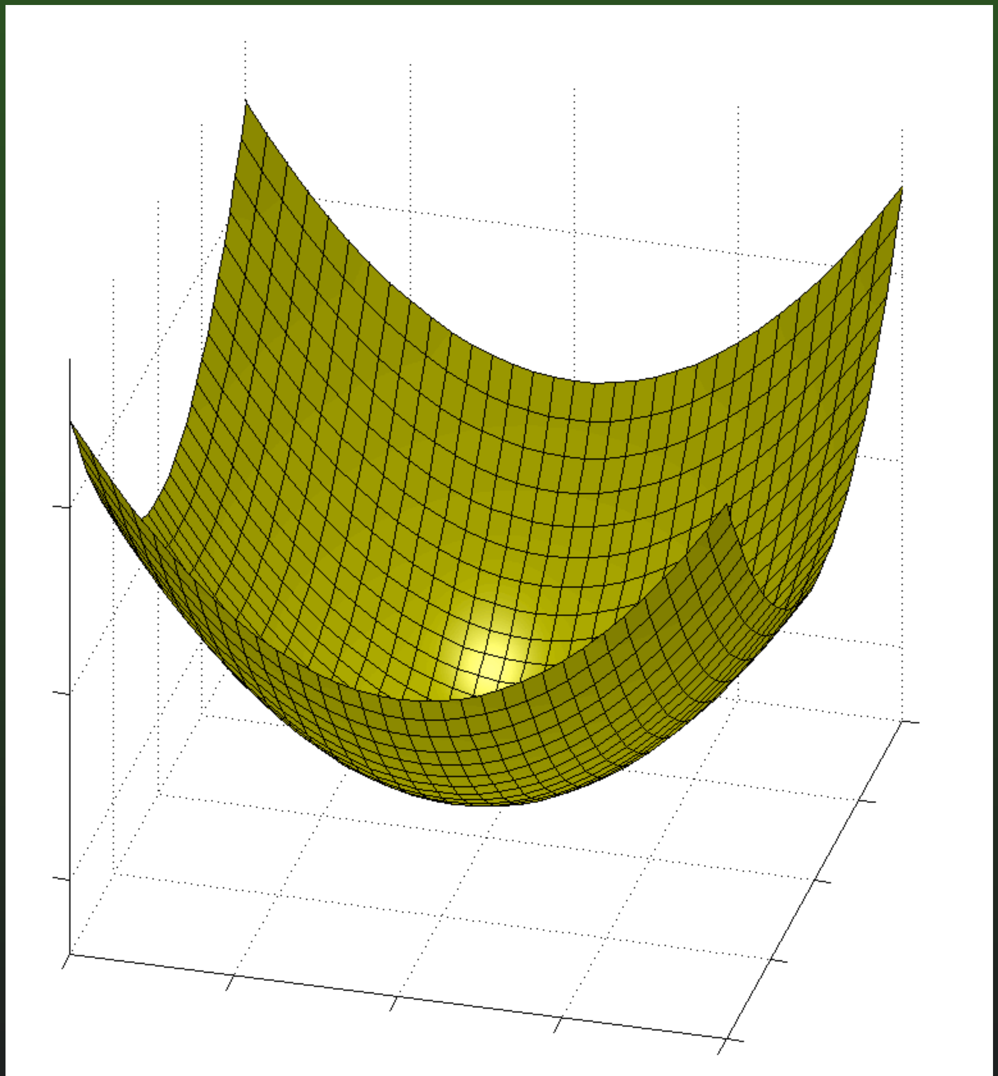
$$\boldsymbol{\mu}_t(\boldsymbol{\theta}_{t+1}) = \boldsymbol{\mu}_t(\boldsymbol{\theta}_t) - \mathbf{A}_t(\Delta\boldsymbol{\theta}_t).$$

- How to change **θ ?**
  - Descent in the direction of $\boldsymbol{\mu}_t(\boldsymbol{\theta})$? $O(n^2)$

iLSTD

$$\boldsymbol{\mu}_t(\boldsymbol{\theta}_{t+1}) = \boldsymbol{\mu}_t(\boldsymbol{\theta}_t) - \mathbf{A}_t(\Delta\boldsymbol{\theta}_t).$$

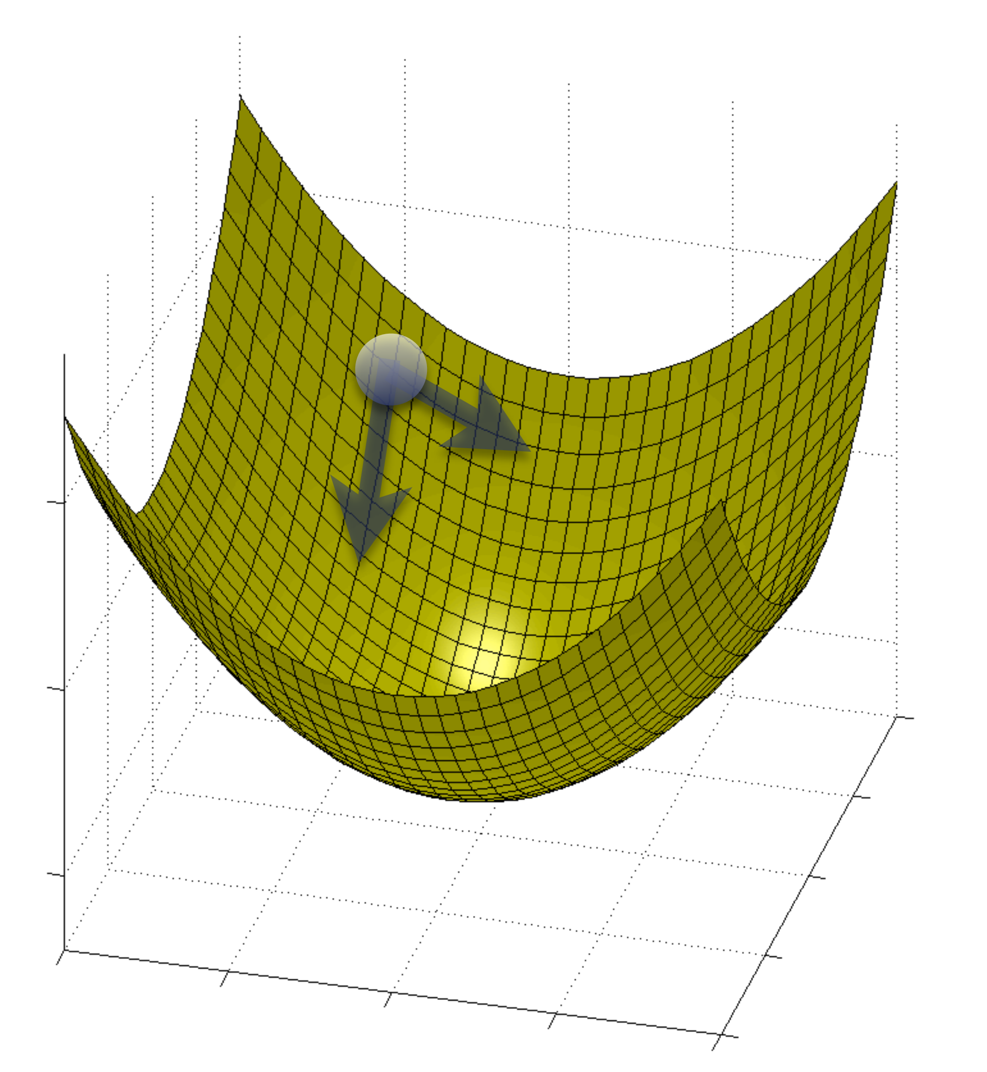
- How to change **θ** ?
  - Descent in the direction of $\boldsymbol{\mu}_t(\boldsymbol{\theta})$? $O(n^2)$



iLSTD

$$\boldsymbol{\mu}_t(\boldsymbol{\theta}_{t+1}) = \boldsymbol{\mu}_t(\boldsymbol{\theta}_t) - \mathbf{A}_t(\Delta\boldsymbol{\theta}_t).$$

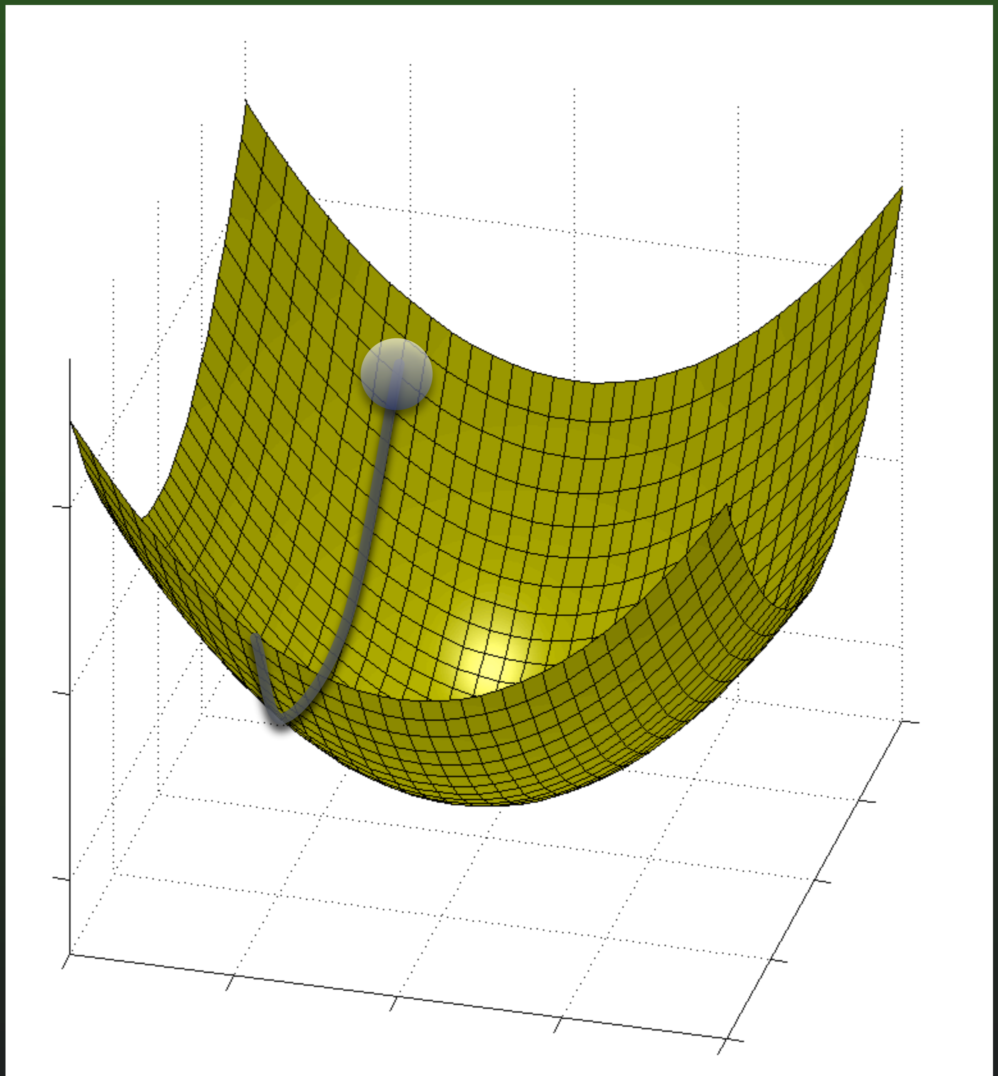
- How to change **θ** ?
  - Descent in the direction of $\boldsymbol{\mu}_t(\boldsymbol{\theta})$? $O(n^2)$



iLSTD

$$\boldsymbol{\mu}_t(\boldsymbol{\theta}_{t+1}) = \boldsymbol{\mu}_t(\boldsymbol{\theta}_t) - \mathbf{A}_t(\Delta\boldsymbol{\theta}_t).$$

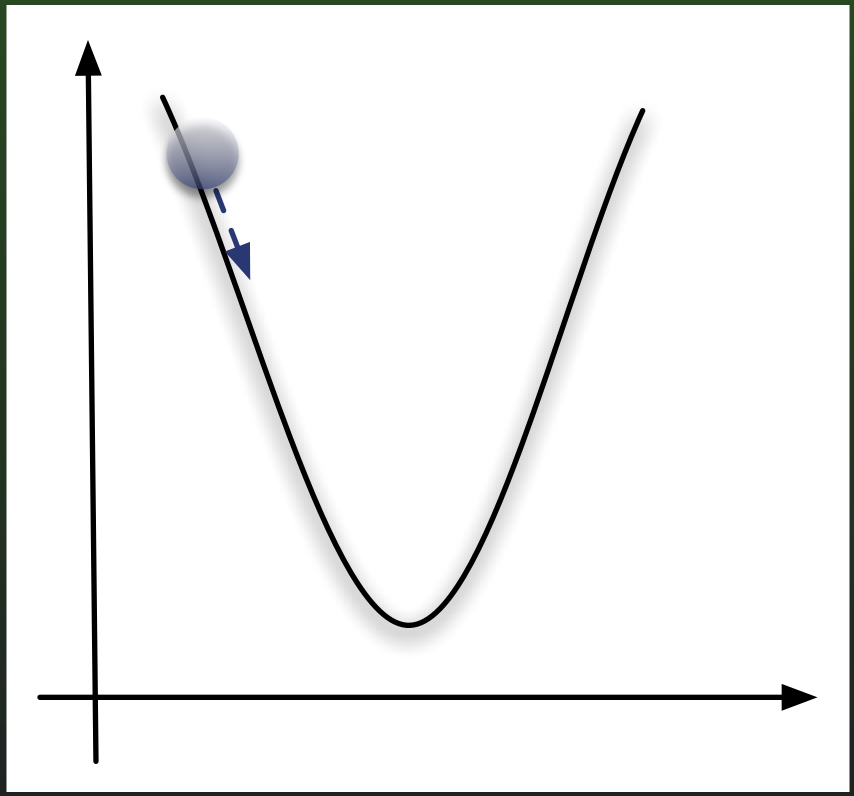
- How to change **θ** ?
  - Descent in the direction of $\boldsymbol{\mu}_t(\boldsymbol{\theta})$? $O(n^2)$



iLSTD

$$\boldsymbol{\mu}_t(\boldsymbol{\theta}_{t+1}) = \boldsymbol{\mu}_t(\boldsymbol{\theta}_t) - \mathbf{A}_t(\Delta\boldsymbol{\theta}_t).$$

- How to change **θ** ?
  - Descent in the direction of $\boldsymbol{\mu}_t(\boldsymbol{\theta})$? $O(n^2)$



iLSTD Algorithm

$s \leftarrow s_0$, $\mathbf{A} \leftarrow \mathbf{0}$, $\boldsymbol{\mu} \leftarrow \mathbf{0}$, $t \leftarrow 0$

Initialize $\boldsymbol{\theta}$ arbitrarily

iLSTD Algorithm

$s \leftarrow s_0$, $\mathbf{A} \leftarrow \mathbf{0}$, $\boldsymbol{\mu} \leftarrow \mathbf{0}$, $t \leftarrow 0$
Initialize $\boldsymbol{\theta}$ arbitrarily
repeat
  Take action according to π and observe r, s'
  $t \leftarrow t + 1$
  $\Delta\mathbf{b} \leftarrow \phi(s)r$
  $\Delta\mathbf{A} \leftarrow \phi(s)(\phi(s) - \gamma\phi(s'))^T$
  $\mathbf{A} \leftarrow \mathbf{A} + \Delta\mathbf{A}$
  $\boldsymbol{\mu} \leftarrow \boldsymbol{\mu} + \Delta\mathbf{b} - (\Delta\mathbf{A})\boldsymbol{\theta}$

iLSTD Algorithm

$s \leftarrow s_0$, $\mathbf{A} \leftarrow \mathbf{0}$, $\boldsymbol{\mu} \leftarrow \mathbf{0}$, $t \leftarrow 0$
Initialize $\boldsymbol{\theta}$ arbitrarily
repeat
  Take action according to π and observe r, s'
  $t \leftarrow t + 1$
  $\Delta\mathbf{b} \leftarrow \phi(s)r$
  $\Delta\mathbf{A} \leftarrow \phi(s)(\phi(s) - \gamma\phi(s'))^T$
  $\mathbf{A} \leftarrow \mathbf{A} + \Delta\mathbf{A}$
  $\boldsymbol{\mu} \leftarrow \boldsymbol{\mu} + \Delta\mathbf{b} - (\Delta\mathbf{A})\boldsymbol{\theta}$
  for i from 1 to m **do**
    $j \leftarrow$ choose an index of $\boldsymbol{\mu}$ using a *dimension selection mechanism*
    $\theta_j \leftarrow \theta_j + \alpha\mu_j$
    $\boldsymbol{\mu} \leftarrow \boldsymbol{\mu} - \alpha\mu_j\mathbf{A}\boldsymbol{e}_j$
  end for
  $s \leftarrow s'$
end repeat

iLSTD

- Per-time-step computational complexity

$$O(mn + k^2)$$

- More data efficient than TD

iLSTD

- Per-time-step computational complexity

$$O(mn + k^2)$$

Number of iterations per time step

- More data efficient than TD

iLSTD

- Per-time-step computational complexity

$$O(mn + k^2)$$

- m: Number of iterations per time step
- n: Number of features

- More data efficient than TD

iLSTD

- Per-time-step computational complexity

$$O(mn + k^2)$$

- k: Maximum number of active features
- n: Number of features
- m: Number of iterations per time step

- More data efficient than TD

iLSTD

- Per-time-step computational complexity

$$O(mn + k^2)$$

Linear

- m: Number of iterations per time step
- n: Number of features
- k: Maximum number of active features

- More data efficient than TD

iLSTD

- *Theorem* : iLSTD converges with probability one to the same solution as TD, under the usual step-size conditions, for any dimension selection method such that all dimensions for which μ_t is non-zero are selected in the limit an infinite number of times.

Empirical Results

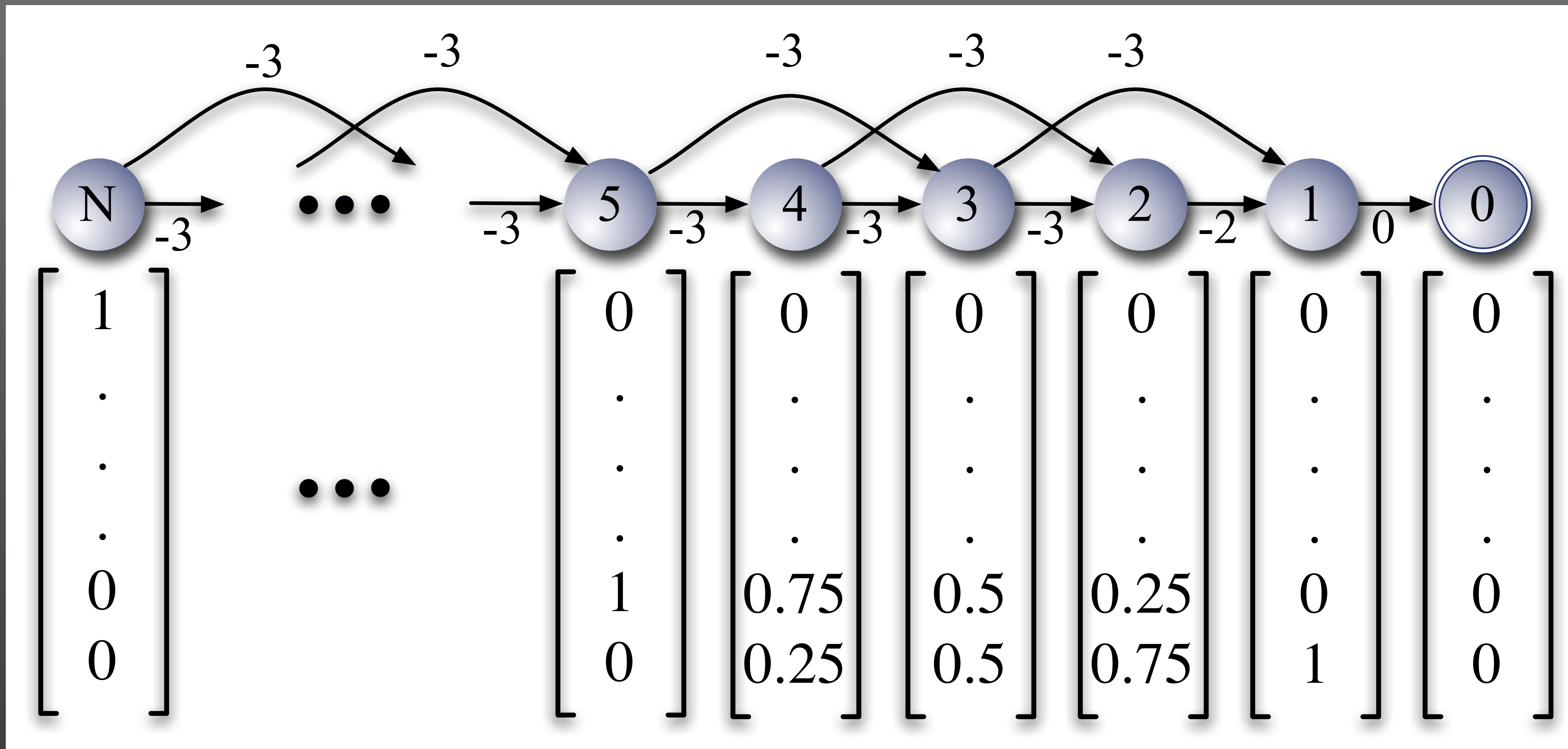
Settings

Settings

- Averaged over 30 runs
- Same random seed for all methods
- Sparse matrix representation
- iLSTD
  - Non-zero random selection
  - One descent per iteration

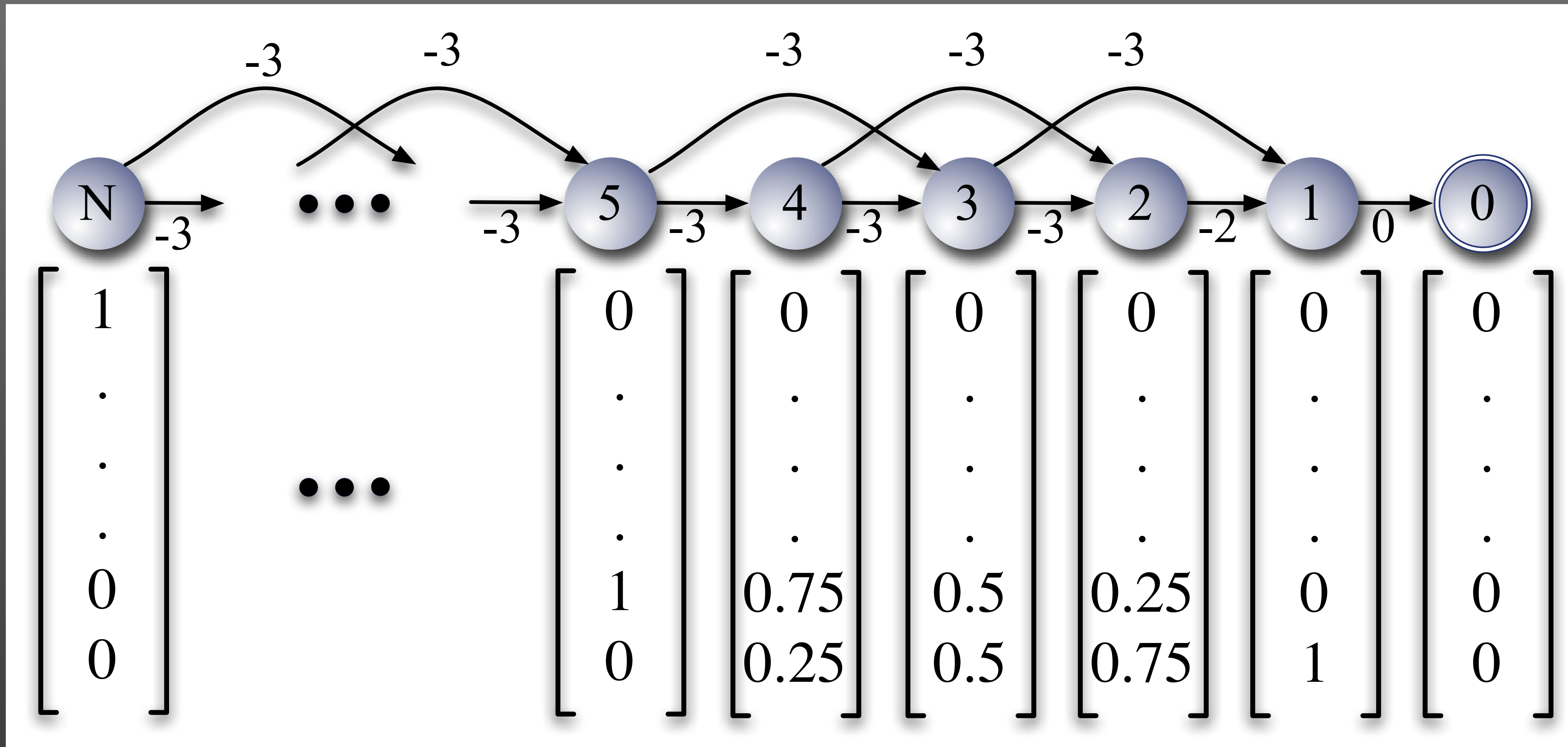
Boyan Chain

Boyan Chain



[Boyan 99]

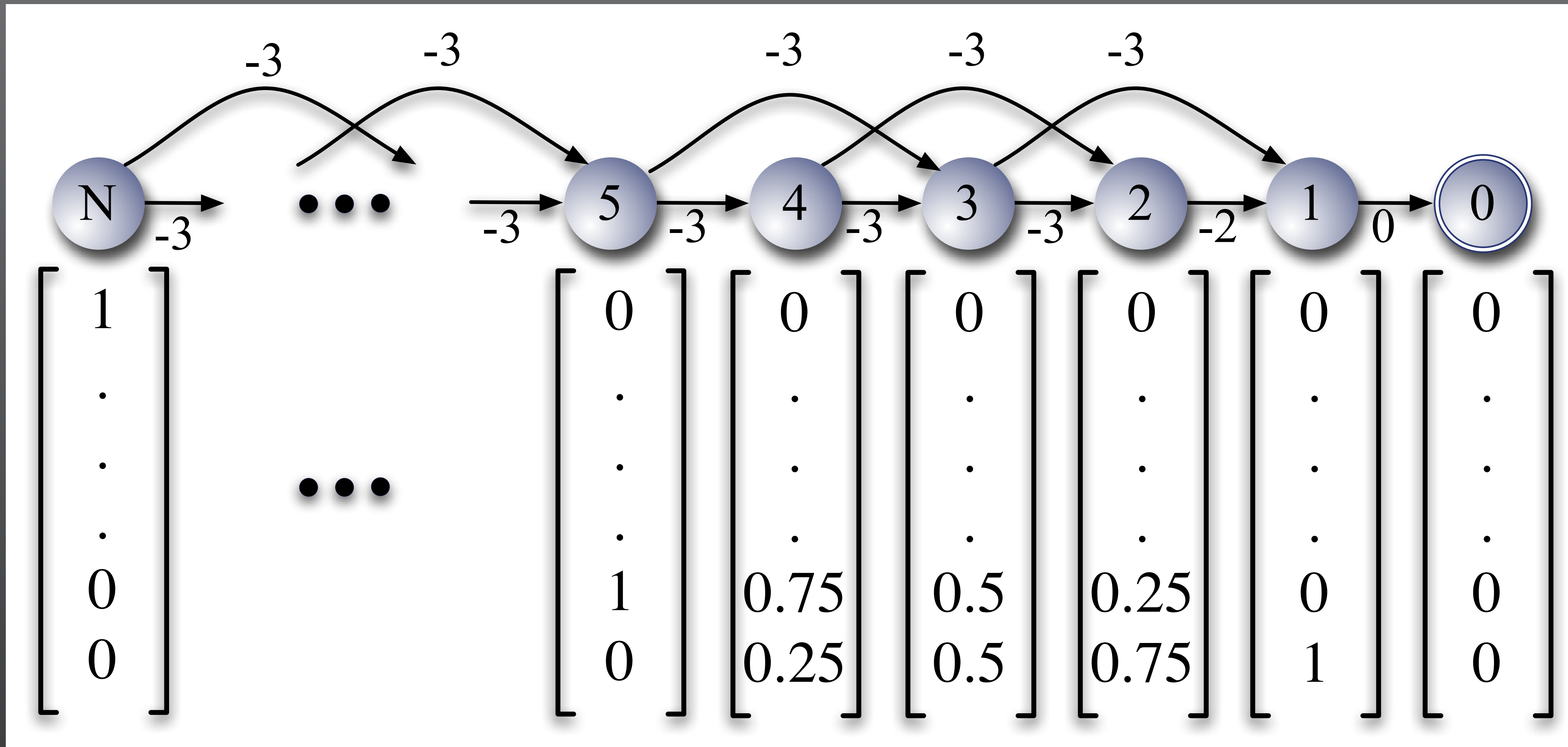
Boyan Chain



- n = 4 (Small)
  n = 25 (Medium)
  n = 100 (Large)

[Boyan 99]

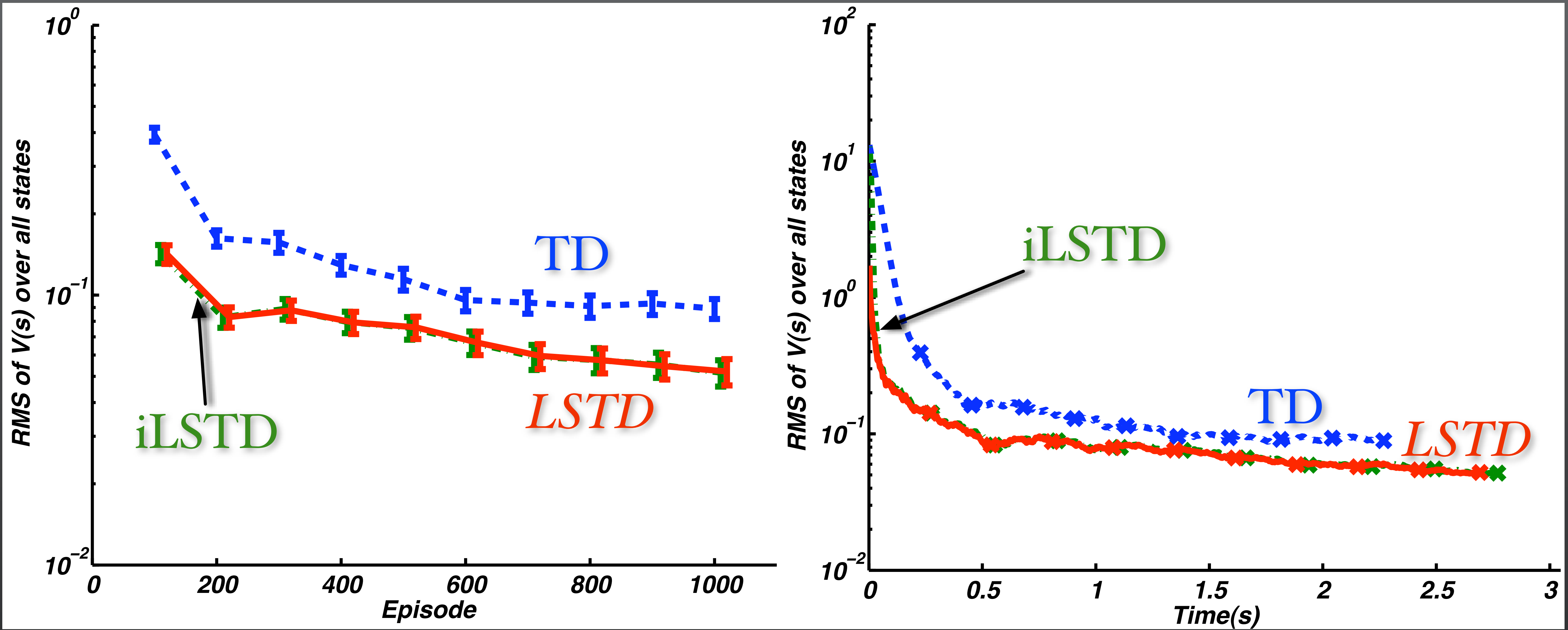
Boyan Chain



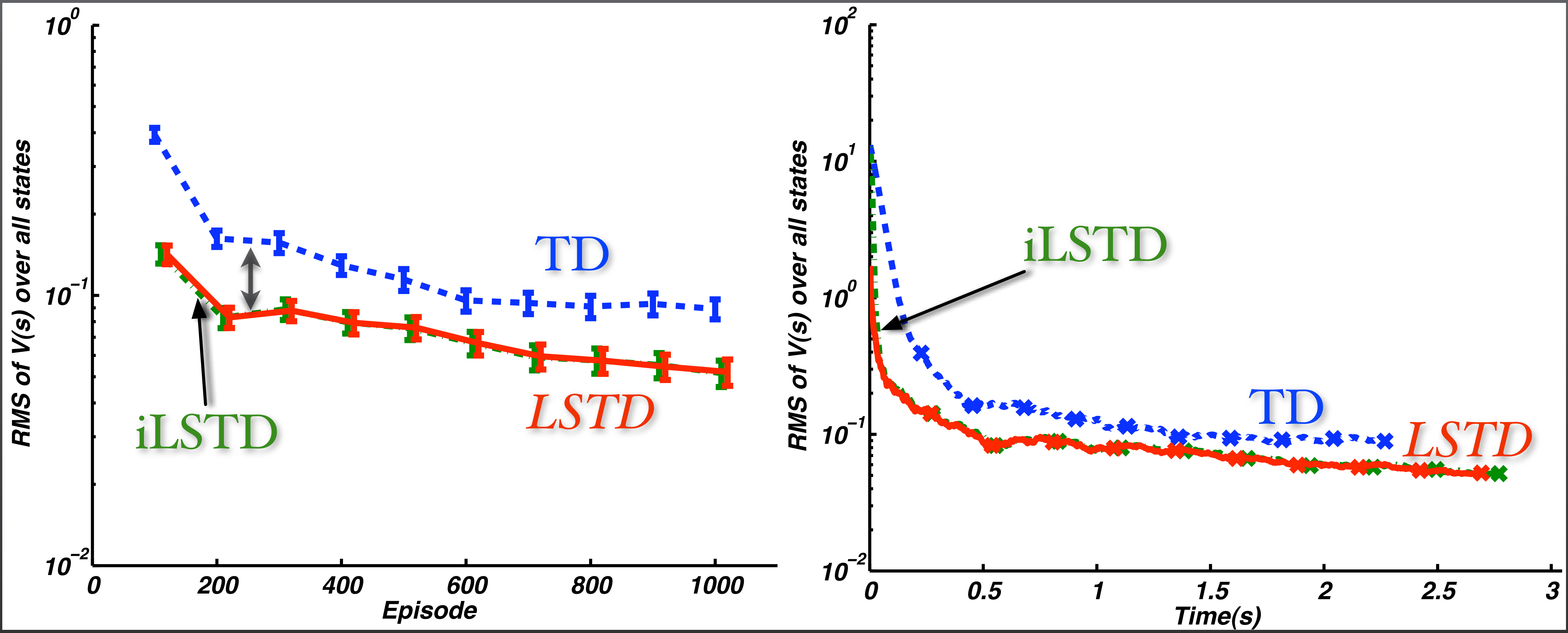
- n = 4 (Small)
  n = 25 (Medium)
  n = 100 (Large)
- k = 2

[Boyan 99]

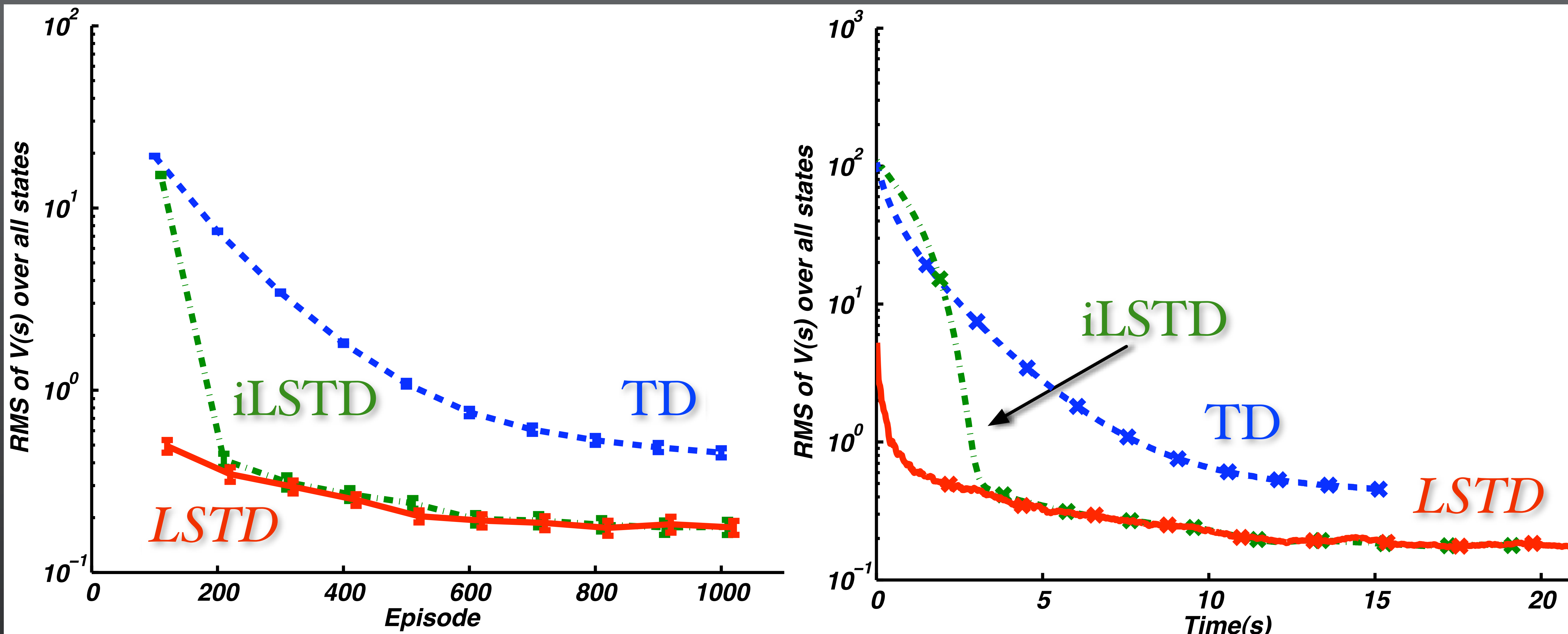
Small Boyan Chain



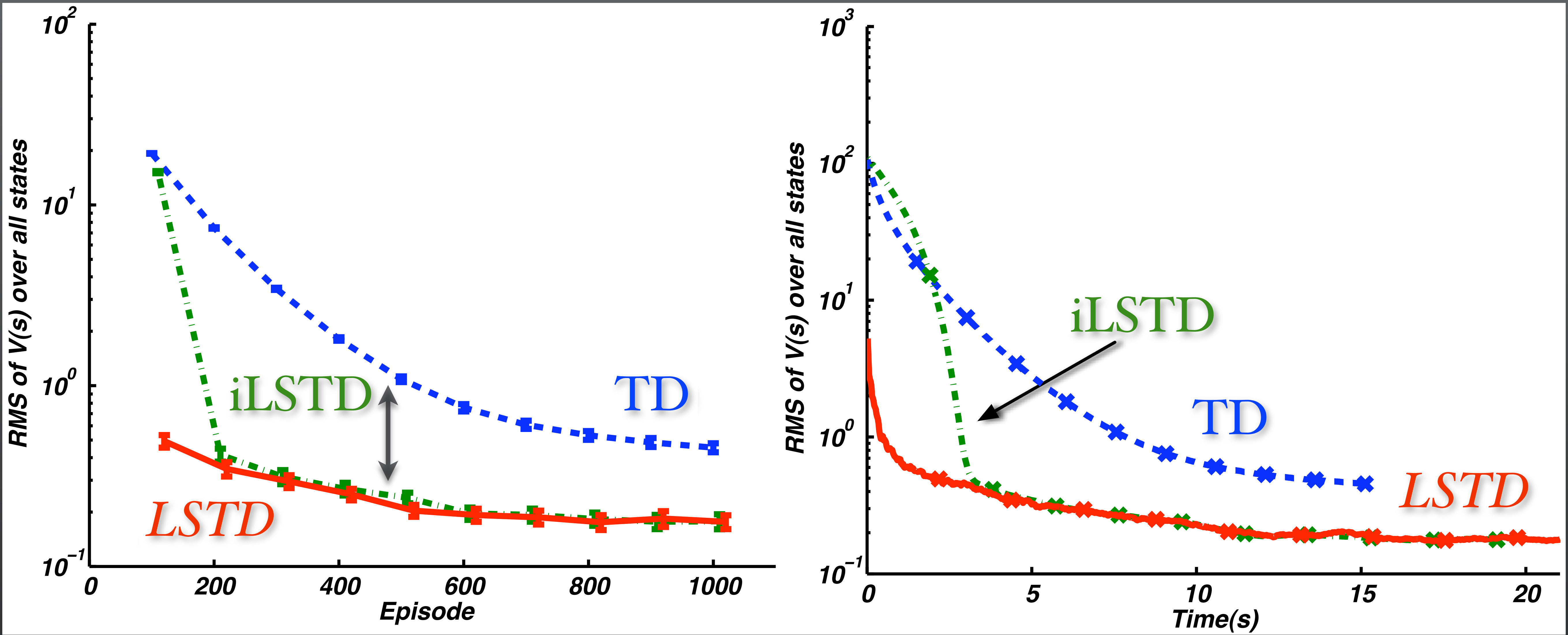
Small Boyan Chain



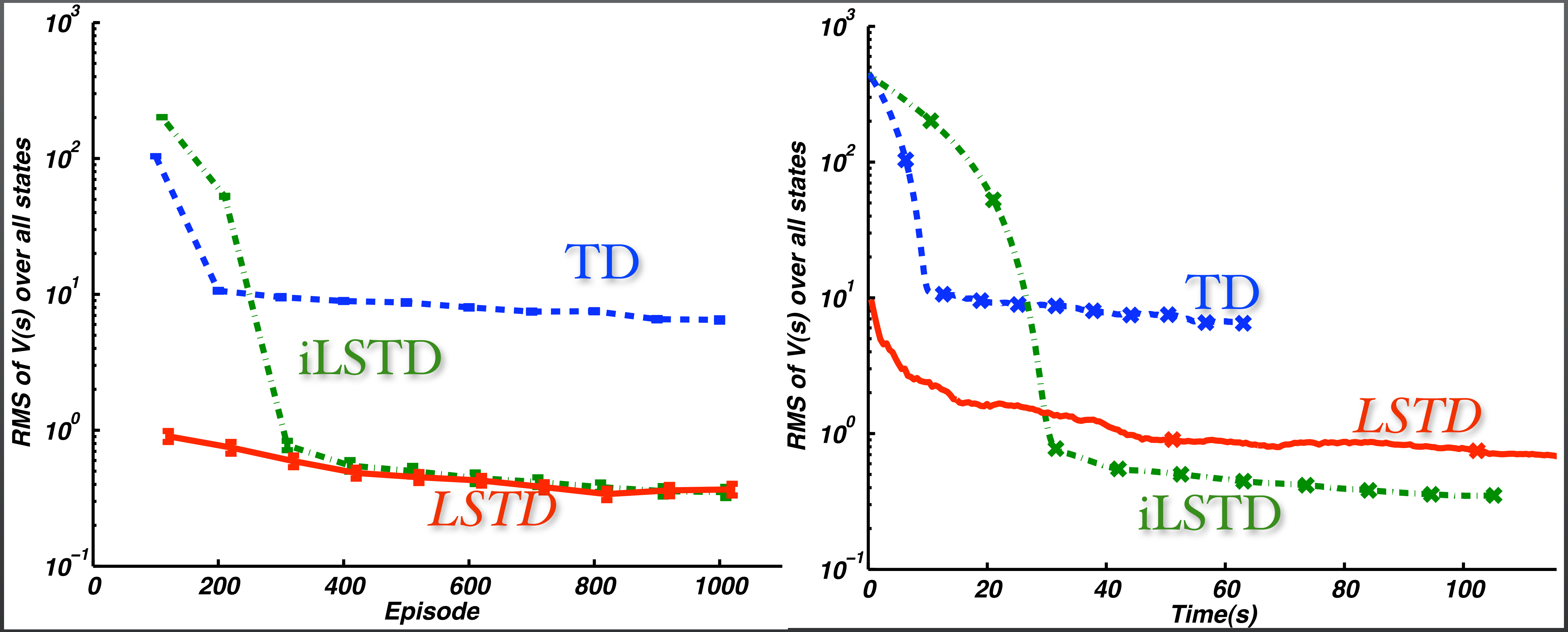
Medium Boyan Chain



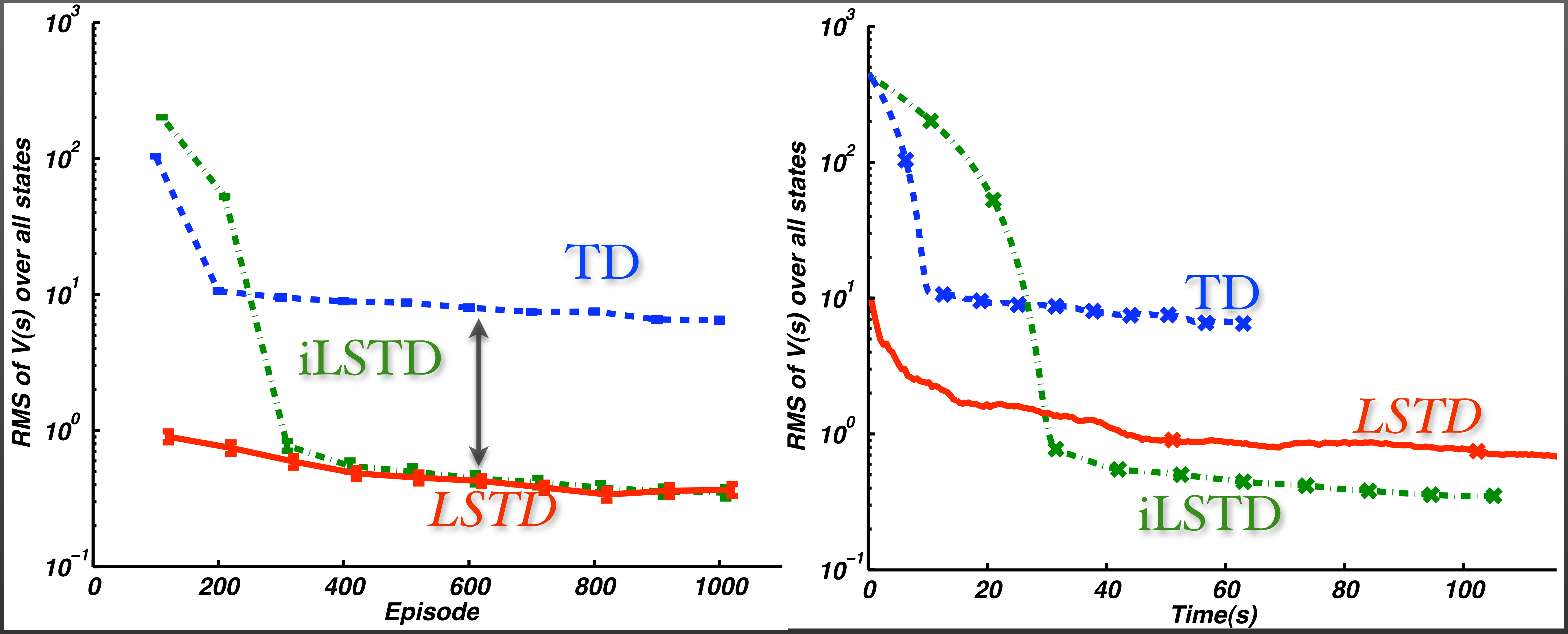
Medium Boyan Chain



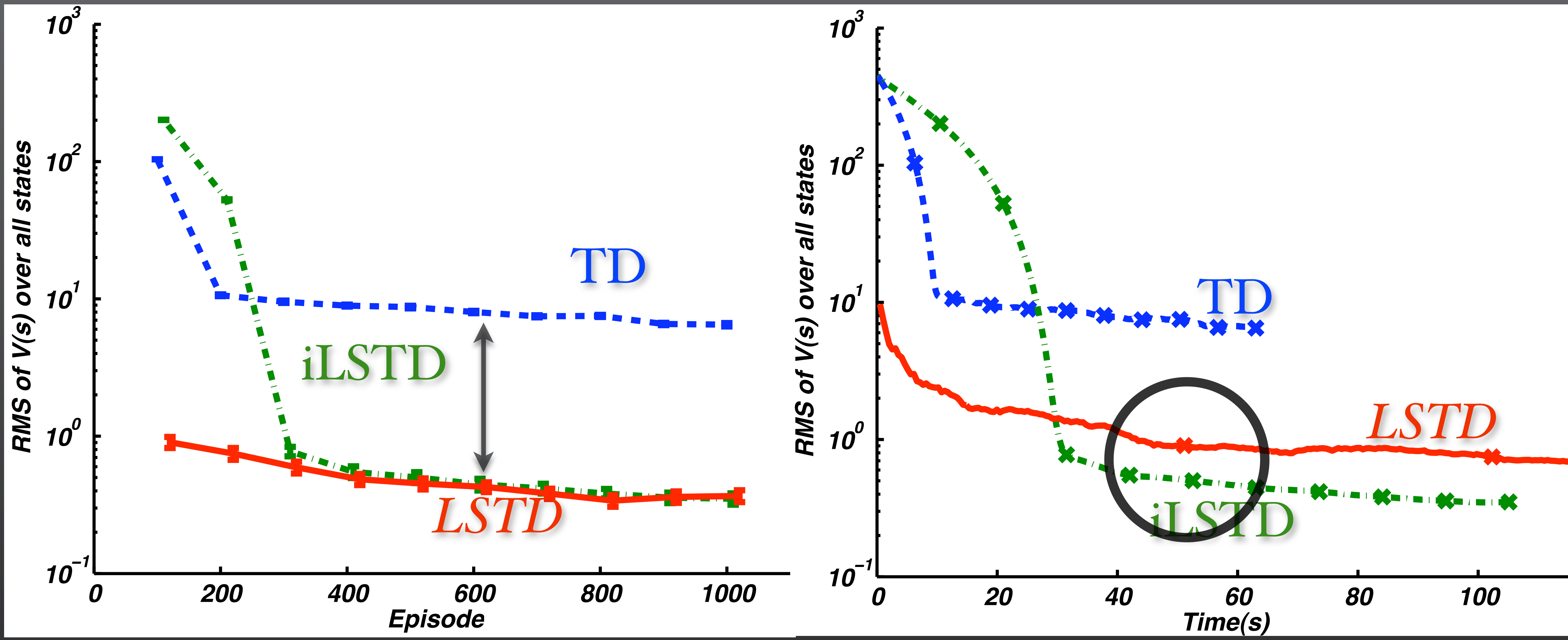
Large Boyan Chain



Large Boyan Chain

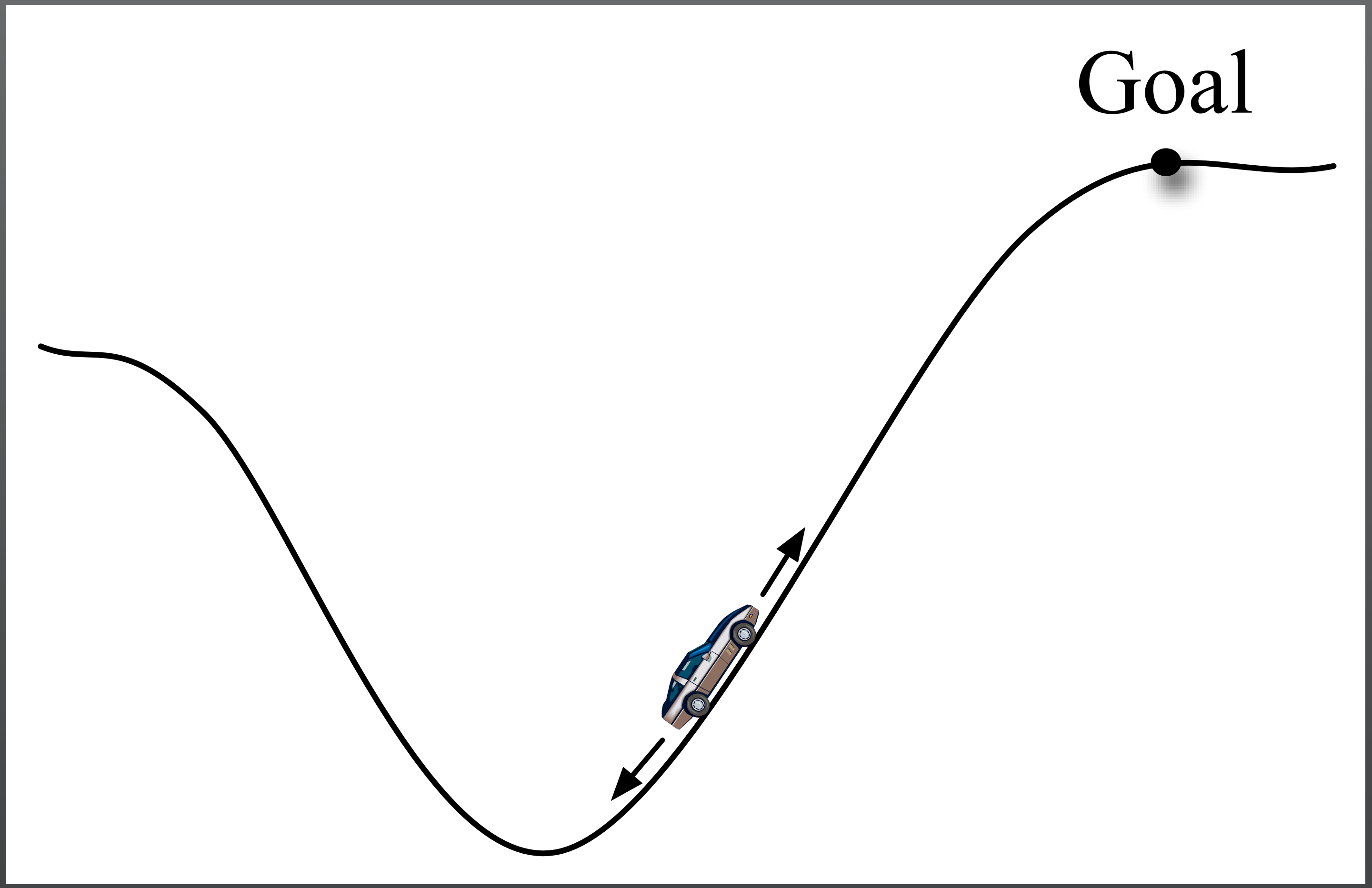


Large Boyan Chain



Mountain Car

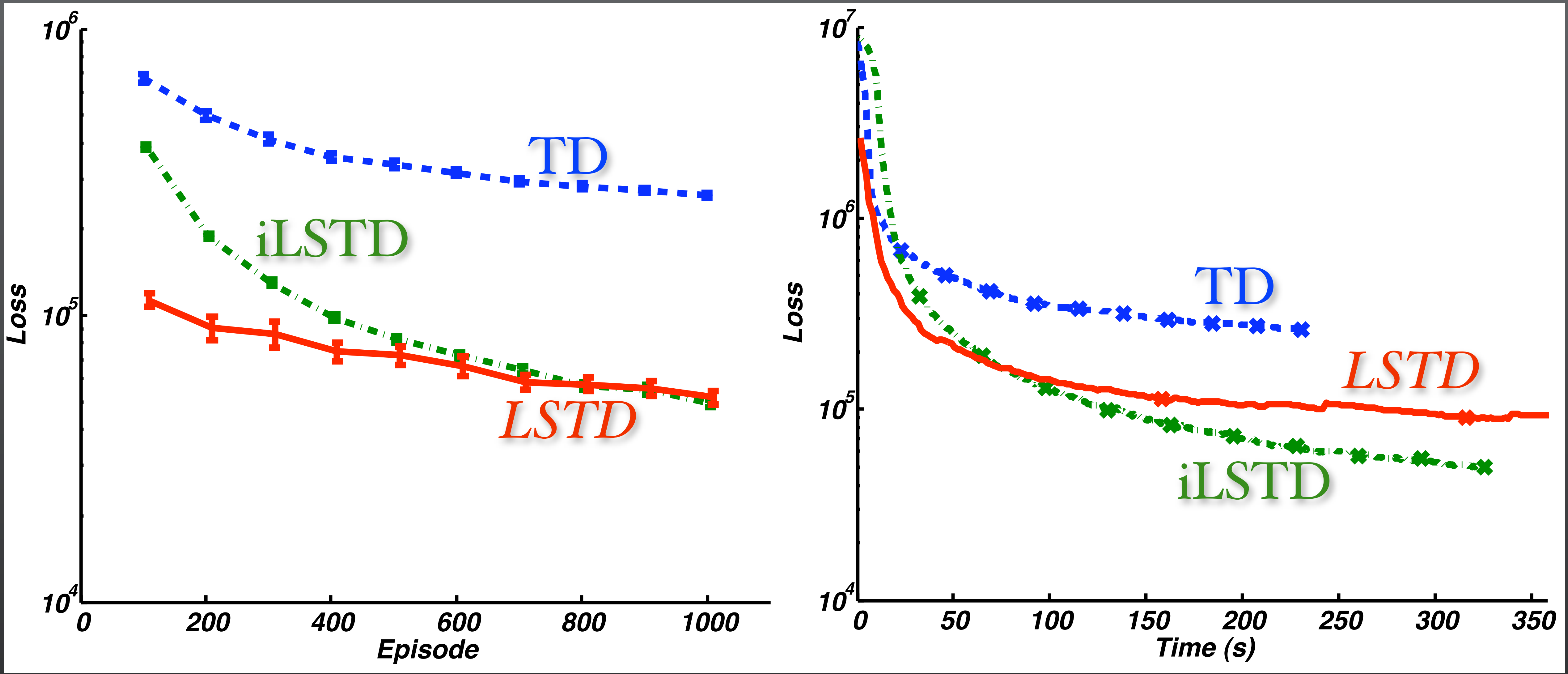
Mountain Car



- Mountain Car
  Position = -1 (Easy)
  Position = -.5 (Hard)
- Tile coding
  n = 10,000
  k = 10

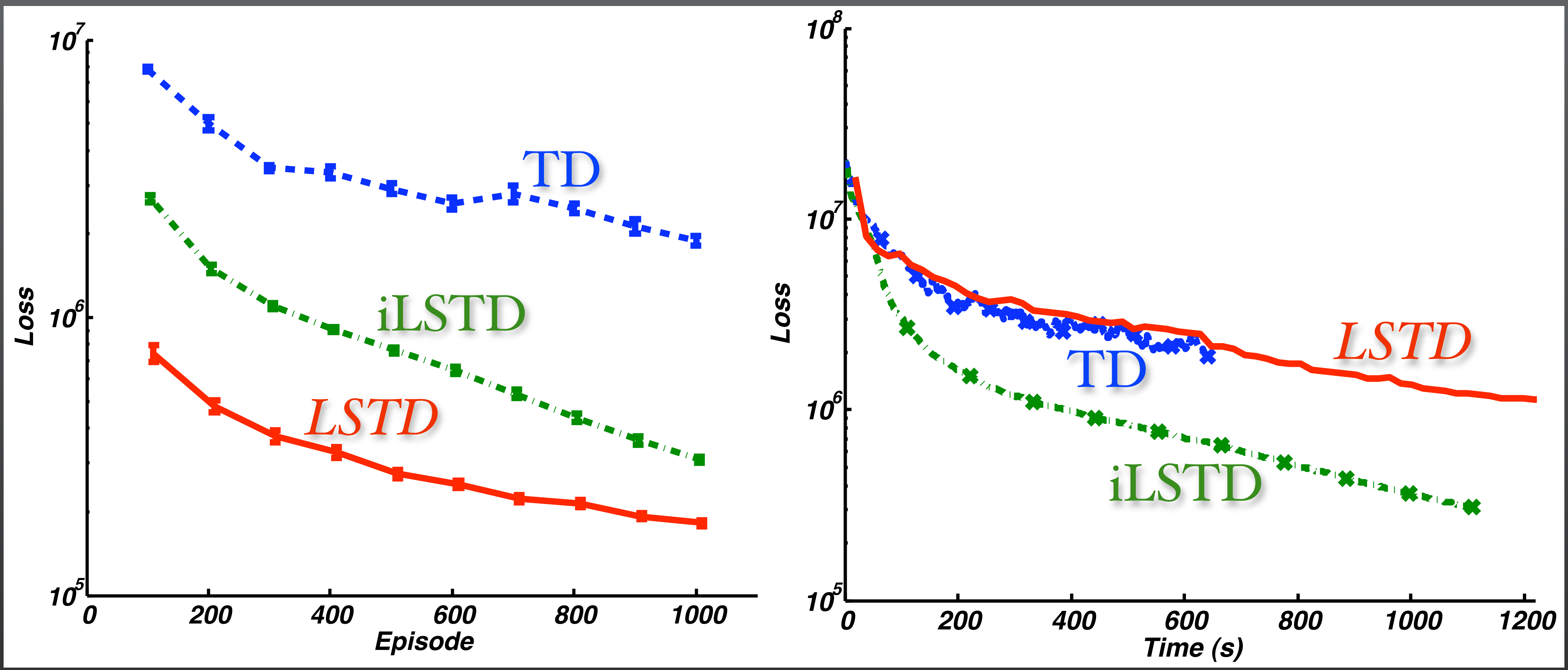
[For details see RL-Library]

Easy Mountain Car

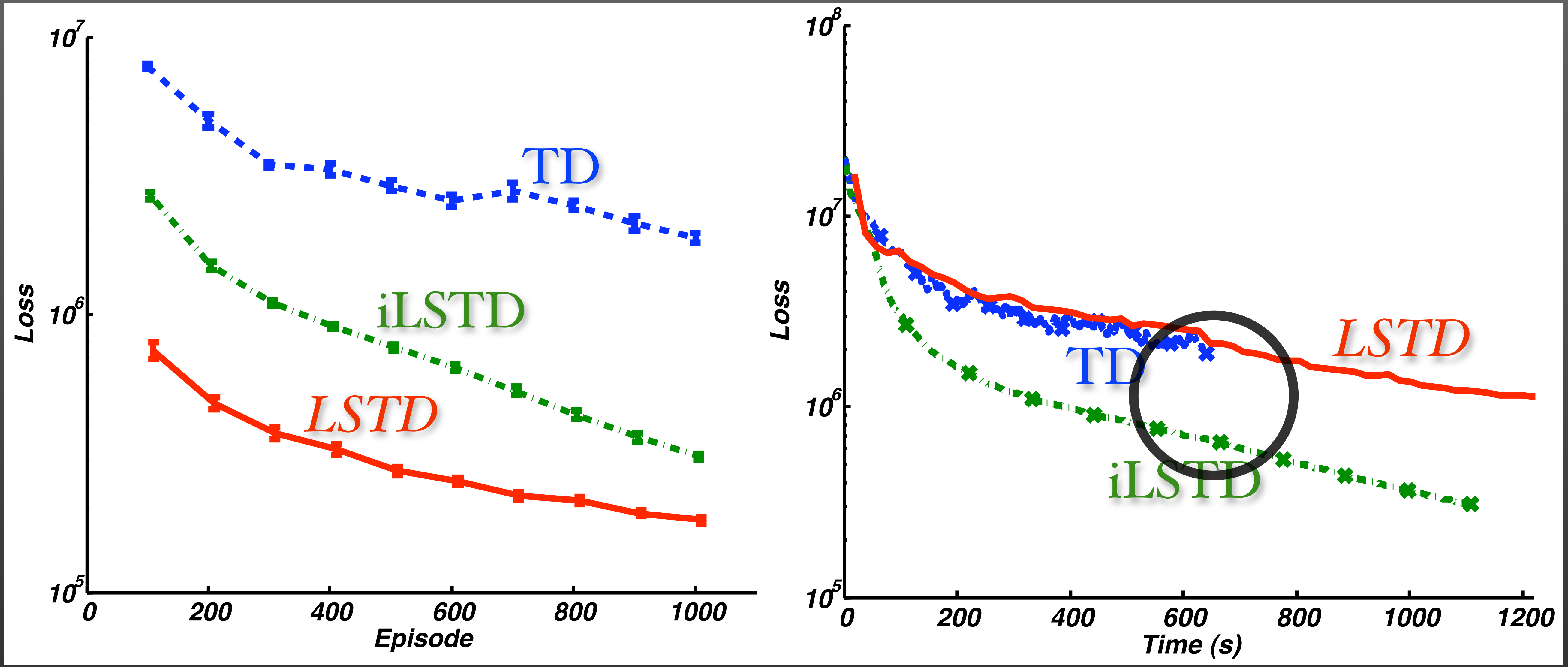


$$\text{Loss} = ||\mathbf{b}^* - \mathbf{A}^*\theta||_2$$

Hard Mountain Car

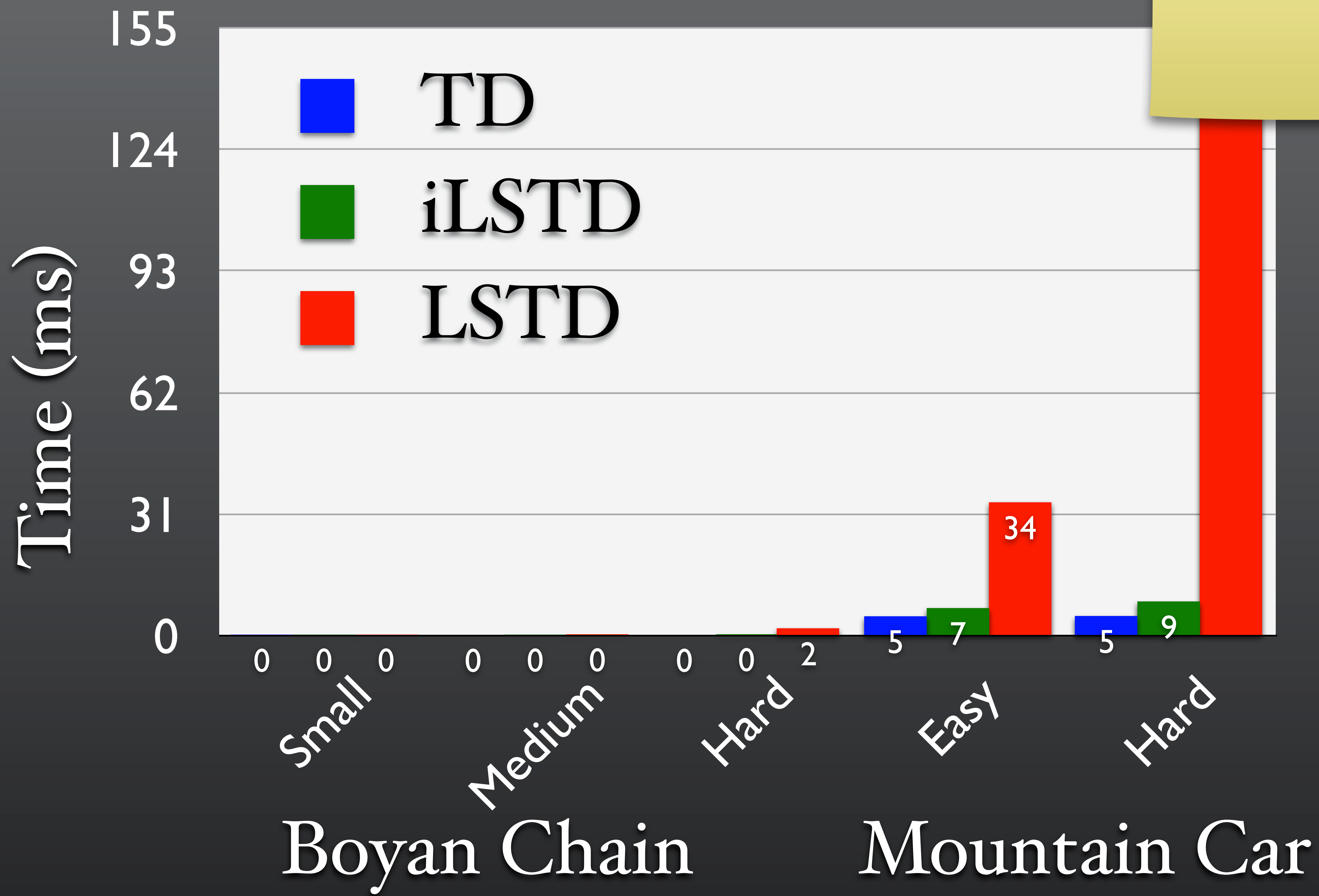


Hard Mountain Car

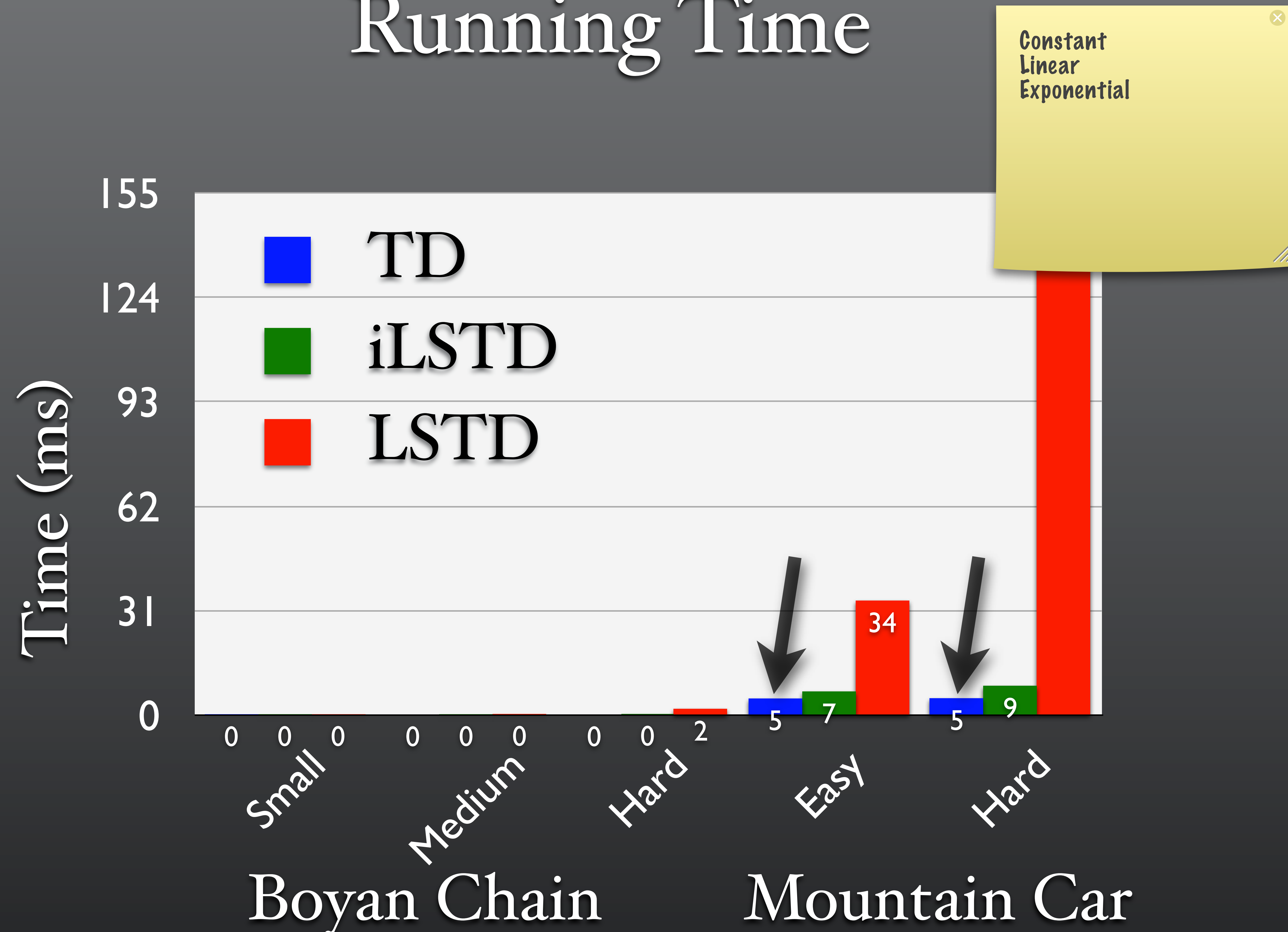


Running Time

Constant
Linear
Exponential



Running Time



Outline

Outline

- Motivation
- Introduction
- The New Approach
- Eligibility Traces ←
- Dimension Selection
- Conclusion

Eligibility Traces for Function Approximation

$$\mathbf{z}_t(i) = \begin{cases} \gamma\lambda\mathbf{z}_{t-1}(i) + 1 & \mathbf{z}(i) \in \text{active features of } \phi(s_t); \\ \gamma\lambda\mathbf{z}_{t-1}(i) & \text{otherwise;} \end{cases}$$

- A threshold for faster computation

$$\lambda^l < \xi$$

TD (λ)

$$\boldsymbol{\theta}_t = \boldsymbol{\theta}_{t-1} + \alpha \mathbf{z}_t \delta_t (V_{\boldsymbol{\theta}_t})$$

- Per-time-step computational complexity

$$O(lk)$$

- More data efficient than TD(0)

$$\mathbf{z}_t$$

TD (λ)

$$\boldsymbol{\theta}_t = \boldsymbol{\theta}_{t-1} + \alpha \mathbf{z}_t \delta_t (V_{\boldsymbol{\theta}_t})$$

- Per-time-step computational complexity

$$O(lk)$$

- More data efficient than TD(0)

$$\mathbf{z}_t$$

[Sutton 88]

TD (λ)

$$\boldsymbol{\theta}_t = \boldsymbol{\theta}_{t-1} + \alpha \mathbf{z}_t \delta_t (V_{\boldsymbol{\theta}_t})$$

- Per-time-step computational complexity

$$O(lk)$$ **Constant**

- More data efficient than TD(0)

$$\mathbf{z}_t$$

[Sutton 88]

TD (λ)

Proof of convergence

$$\boldsymbol{\theta}_t = \boldsymbol{\theta}_{t-1} + \alpha \mathbf{z}_t \delta_t(V_{\boldsymbol{\theta}_t})$$

- Per-time-step computational complexity

$O(lk)$ Constant

- More data efficient than TD(0)

$\mathbf{z}_t$

[Sutton 88]

LSTD(λ)

$$\boldsymbol{\mu}_t(\boldsymbol{\theta}) = \underbrace{\sum_{i=1}^{t} z_i r_{i+1}}_{\mathbf{b}_t} - \underbrace{\sum_{i=1}^{t} z_i(\boldsymbol{\phi}_i - \gamma \boldsymbol{\phi}_{i+1})^T}_{\mathbf{A}_t} \boldsymbol{\theta}$$

- Per-time-step computational complexity

$$O(n^2)$$

LSTD(λ)

$$\mu_t(\boldsymbol{\theta}) = \underbrace{\sum_{i=1}^{t} z_i r_{i+1}}_{\mathbf{b}_t} - \underbrace{\sum_{i=1}^{t} z_i (\boldsymbol{\phi}_i - \gamma \boldsymbol{\phi}_{i+1})^T}_{\mathbf{A}_t} \boldsymbol{\theta}$$

- Per-time-step computational complexity

$$O(n^2)$$

[Boyan 99]

LSTD(λ)

$$\boldsymbol{\mu}_t(\boldsymbol{\theta}) = \underbrace{\sum_{i=1}^{t} z_i r_{i+1}}_{\mathbf{b}_t} - \underbrace{\sum_{i=1}^{t} z_i (\boldsymbol{\phi}_i - \gamma \boldsymbol{\phi}_{i+1})^T}_{\mathbf{A}_t} \boldsymbol{\theta}$$

- Per-time-step computational complexity

$$O(n^2)$$ **Quadratic**

[Boyan 99]

LSTD(λ)

Proof of convergence

$$\mu_t(\theta) = \underbrace{\sum_{i=1}^{t} z_i r_{i+1}}_{\mathbf{b}_t} - \underbrace{\sum_{i=1}^{t} z_i(\phi_i - \gamma\phi_{i+1})^T}_{\mathbf{A}_t} \theta$$

- Per-time-step computational complexity

$$O(n^2)$$

Quadratic

[Boyan 99]

iLSTD(λ)

$$\boldsymbol{\mu}_t(\boldsymbol{\theta}) = \underbrace{\sum_{i=1}^{t} z_i r_{i+1}}_{\mathbf{b}_t} - \underbrace{\sum_{i=1}^{t} z_i (\boldsymbol{\phi}_i - \gamma \boldsymbol{\phi}_{i+1})^T}_{\mathbf{A}_t} \boldsymbol{\theta}$$

- Per-time-step computational complexity

$$O(mn + lk^2)$$

iLSTD(λ)

$$\boldsymbol{\mu}_t(\boldsymbol{\theta}) = \underbrace{\sum_{i=1}^{t} z_i r_{i+1}}_{\mathbf{b}_t} - \underbrace{\sum_{i=1}^{t} z_i (\boldsymbol{\phi}_i - \gamma \boldsymbol{\phi}_{i+1})^T}_{\mathbf{A}_t} \boldsymbol{\theta}$$

- Per-time-step computational complexity

$$O(mn + lk^2)$$

[Geramifard, Bowling, Zinkevich, Sutton 07]

iLSTD(λ)

$$\boldsymbol{\mu}_t(\boldsymbol{\theta}) = \underbrace{\sum_{i=1}^{t} z_i r_{i+1}}_{\mathbf{b}_t} - \underbrace{\sum_{i=1}^{t} z_i(\boldsymbol{\phi}_i - \gamma\boldsymbol{\phi}_{i+1})^T}_{\mathbf{A}_t} \boldsymbol{\theta}$$

- Per-time-step computational complexity

$$O(mn + lk^2)$$

Linear

[Geramifard, Bowling, Zinkevich, Sutton 07]

Proof of convergence

iLSTD(λ)

$$\mu_t(\theta) = \underbrace{\sum_{i=1}^{t} z_i r_{i+1}}_{\mathbf{b}_t} - \underbrace{\sum_{i=1}^{t} z_i(\phi_i - \gamma\phi_{i+1})^T}_{\mathbf{A}_t} \theta$$

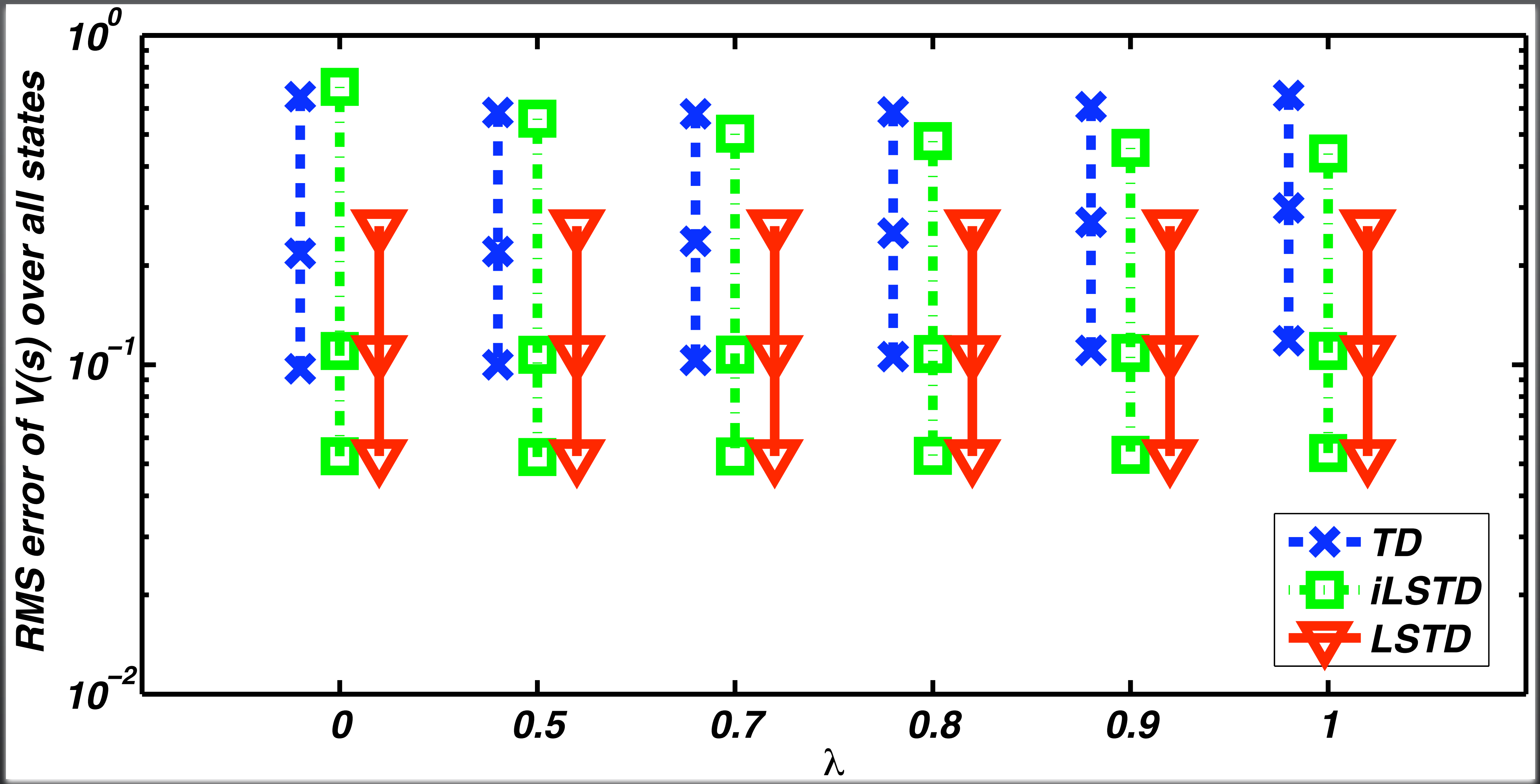
- Per-time-step computational complexity

$$O(mn + lk^2)$$

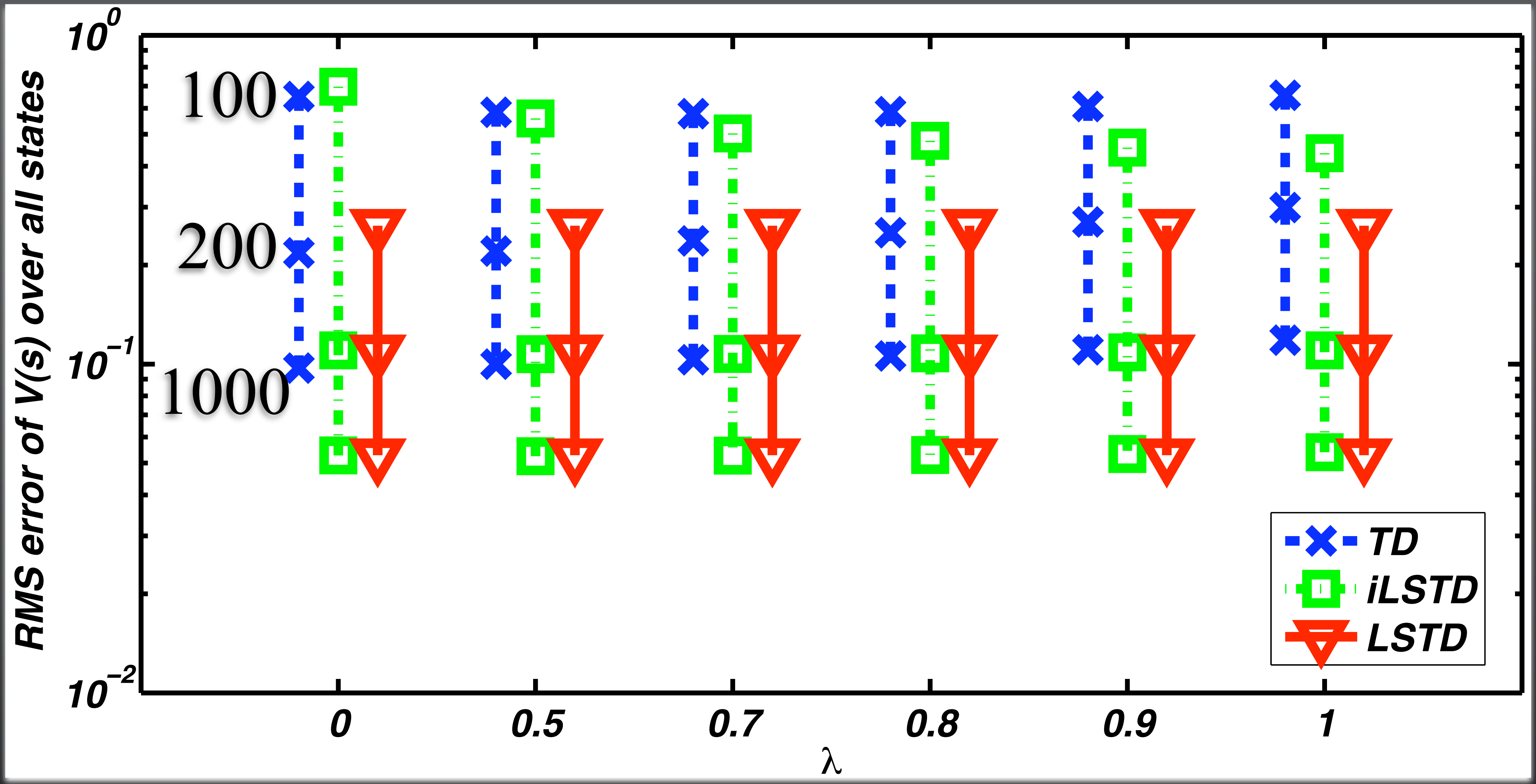
Linear

[Geramifard, Bowling, Zinkevich, Sutton 07]

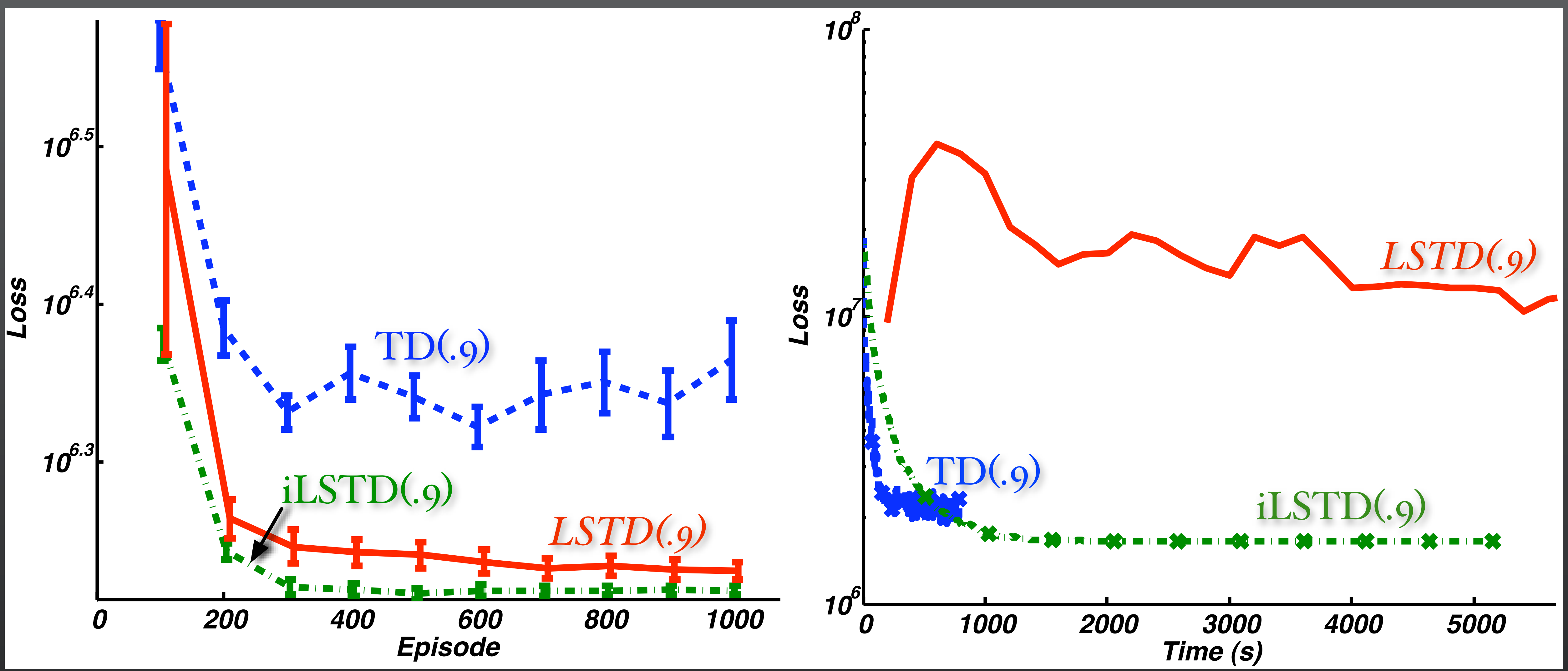
Results on Small Boyan Chain



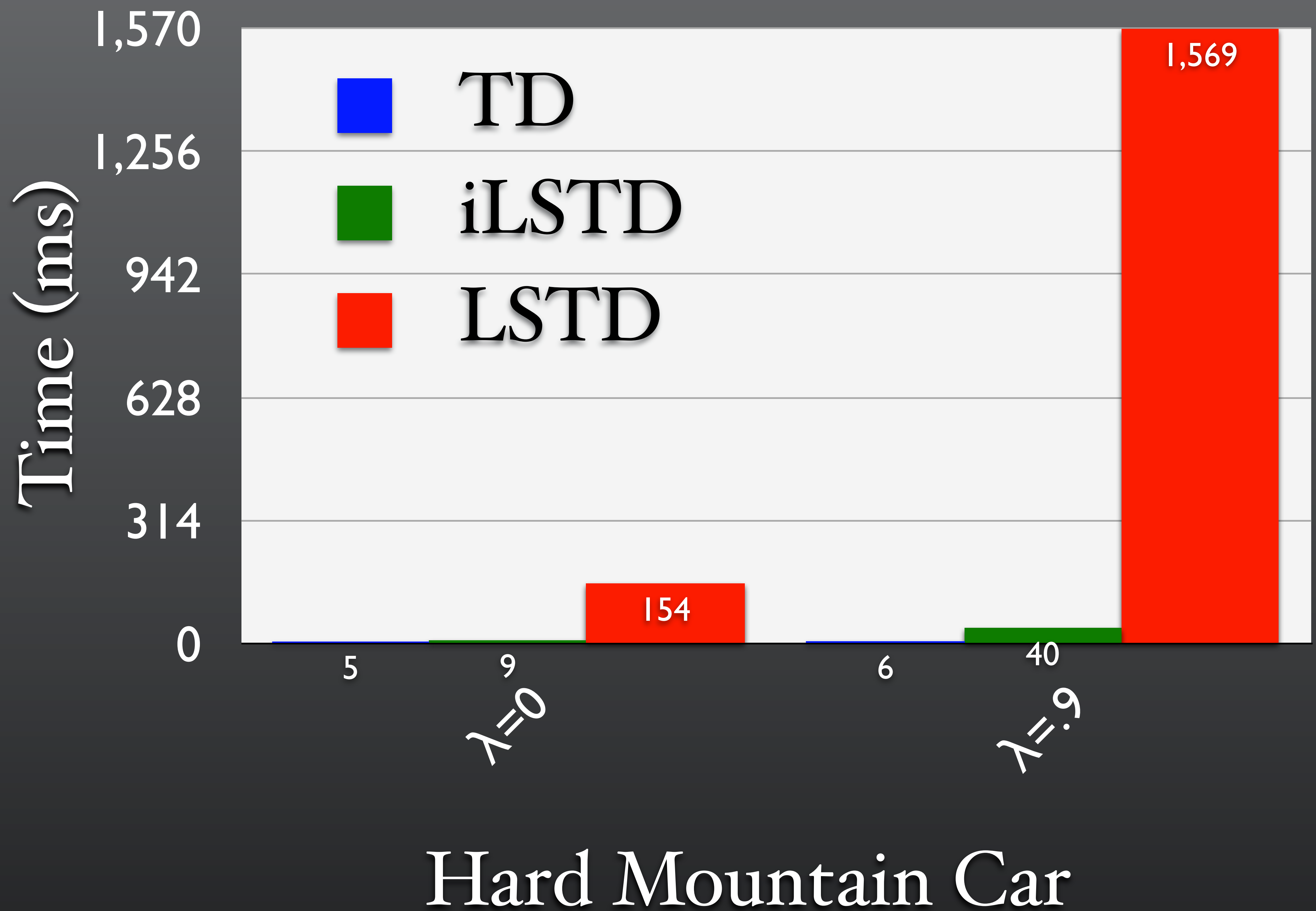
Results on Small Boyan Chain



Results on Hard mountain car



Running Time



Outline

Outline

- Motivation
- Introduction
- The New Approach
- Eligibility Traces
- Dimension Selection
- Conclusion

Dimension Selection

- Random ✓

Dimension Selection

- Random ✓
- Greedy

Dimension Selection

- Random ✓
- Greedy
- ε-Greedy

Dimension Selection

- Random ✓
- Greedy
- ε-Greedy
- Boltzmann

Greedy Dimension Selection

- Pick the one with highest value of

$$|\boldsymbol{\mu}_t(i)|$$

Greedy Dimension Selection

- Pick the one with highest value of

$$|\boldsymbol{\mu}_t(i)|$$

- Not proven to converge.

ε-Greedy Dimension Selection

- ε : Non-Zero Random
- (1-ε) : Greedy

ε-Greedy Dimension Selection

- ε : Non-Zero Random
- (1-ε) : Greedy
- Convergence proof applies.

Boltzmann Component Selection

- Boltzmann Distribution + Non-Zero Random

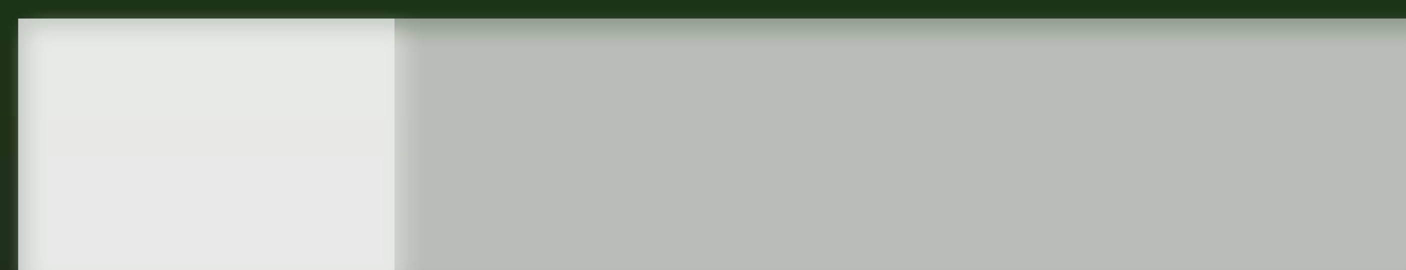
- Convergence proof applies.

Boltzmann Component Selection

- Boltzmann Distribution + Non-Zero Random
- Convergence proof applies.

Boltzmann Component Selection

- Boltzmann Distribution + Non-Zero Random



$\psi \times m$

- Convergence proof applies.

Boltzmann Component Selection

- Boltzmann Distribution + Non-Zero Random

$$\psi \times m$$

- Convergence proof applies.

Boltzmann Component Selection

- Boltzmann Distribution + Non-Zero Random

Boltzmann Distribution

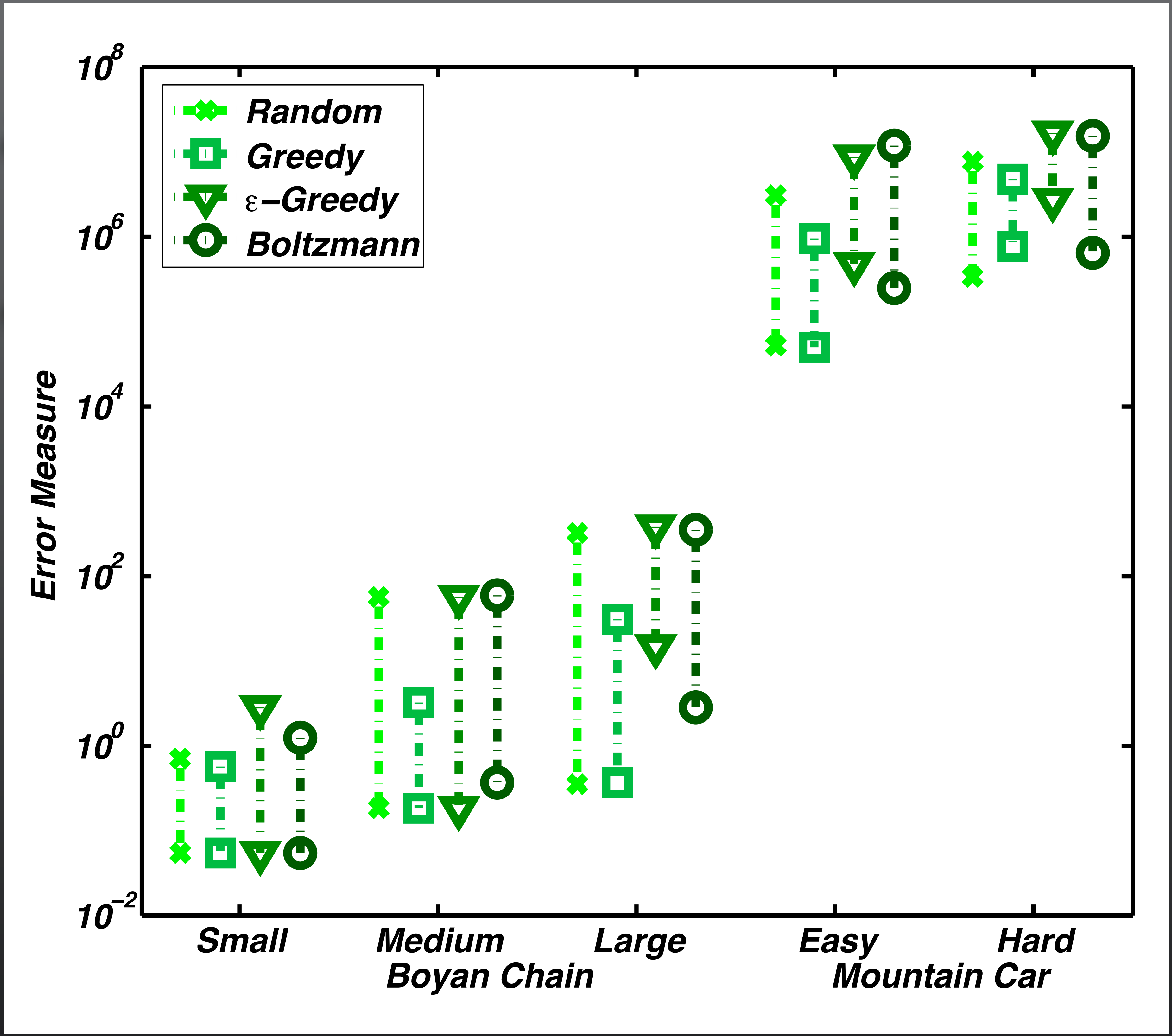
$$\psi \times m$$

- Convergence proof applies.

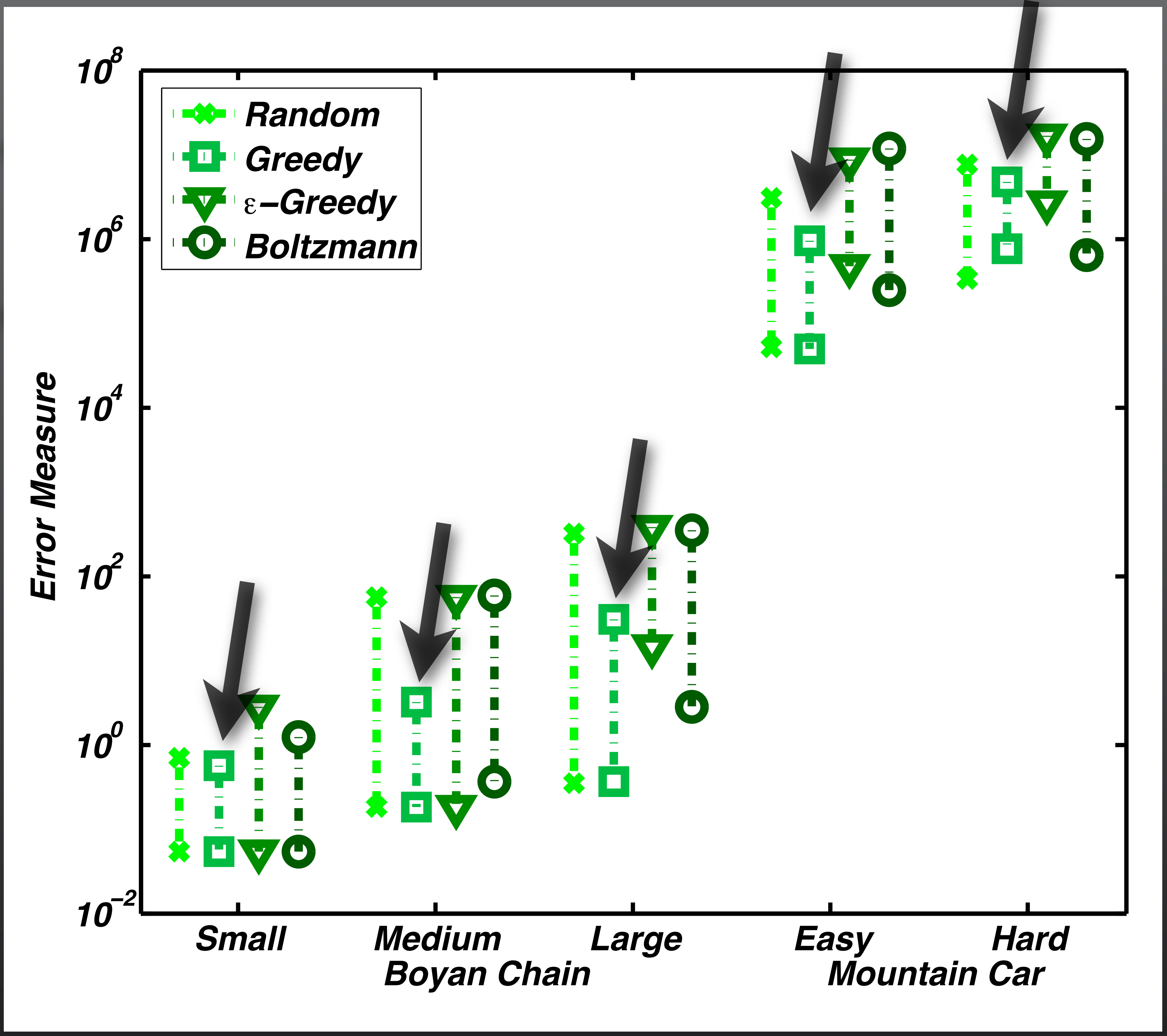
Empirical Results

- ε-Greedy: $\varepsilon = .1$
- Boltzmann: $\psi = 10^{-9}$, $\tau = 1$

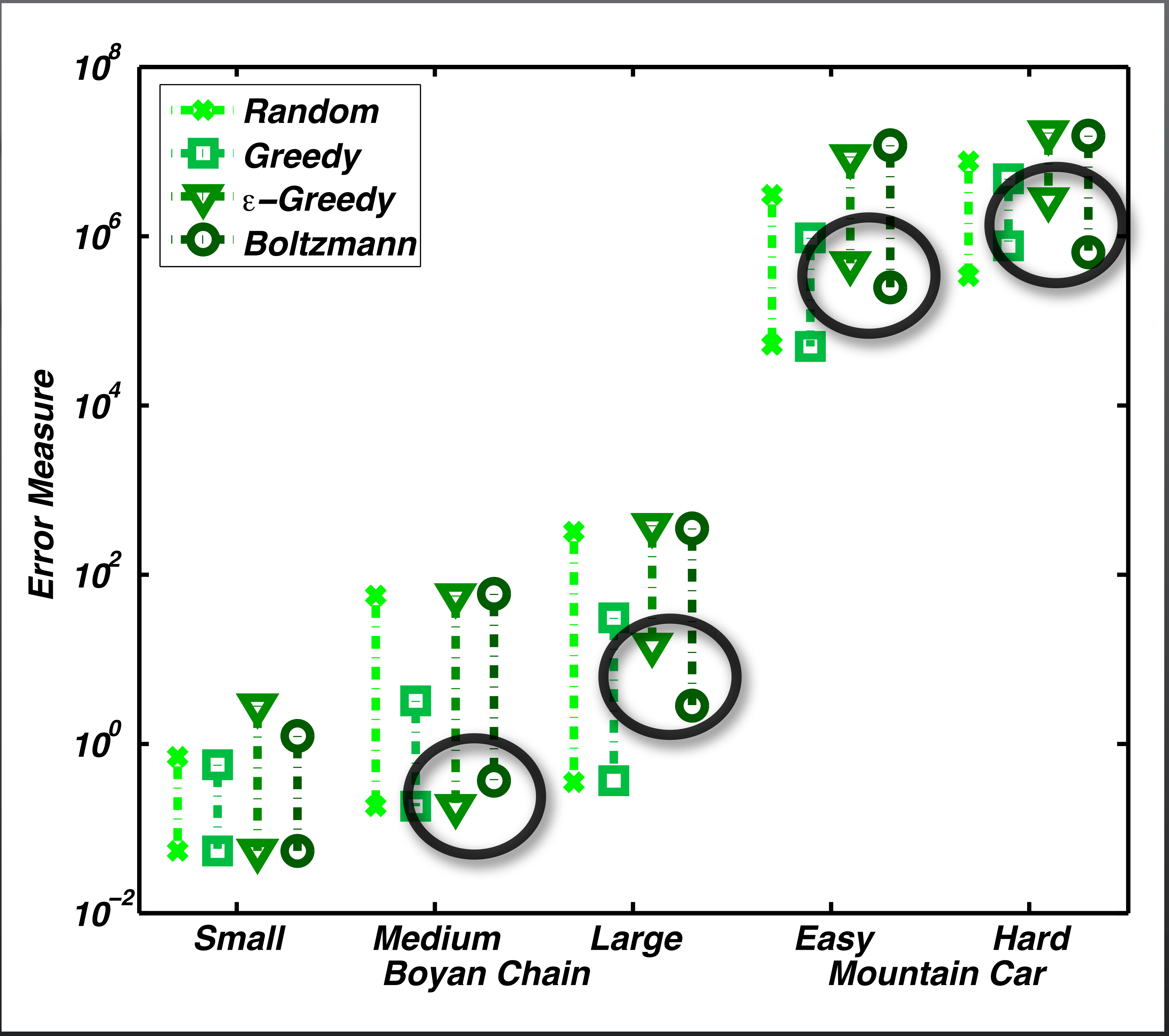
Empirical Results



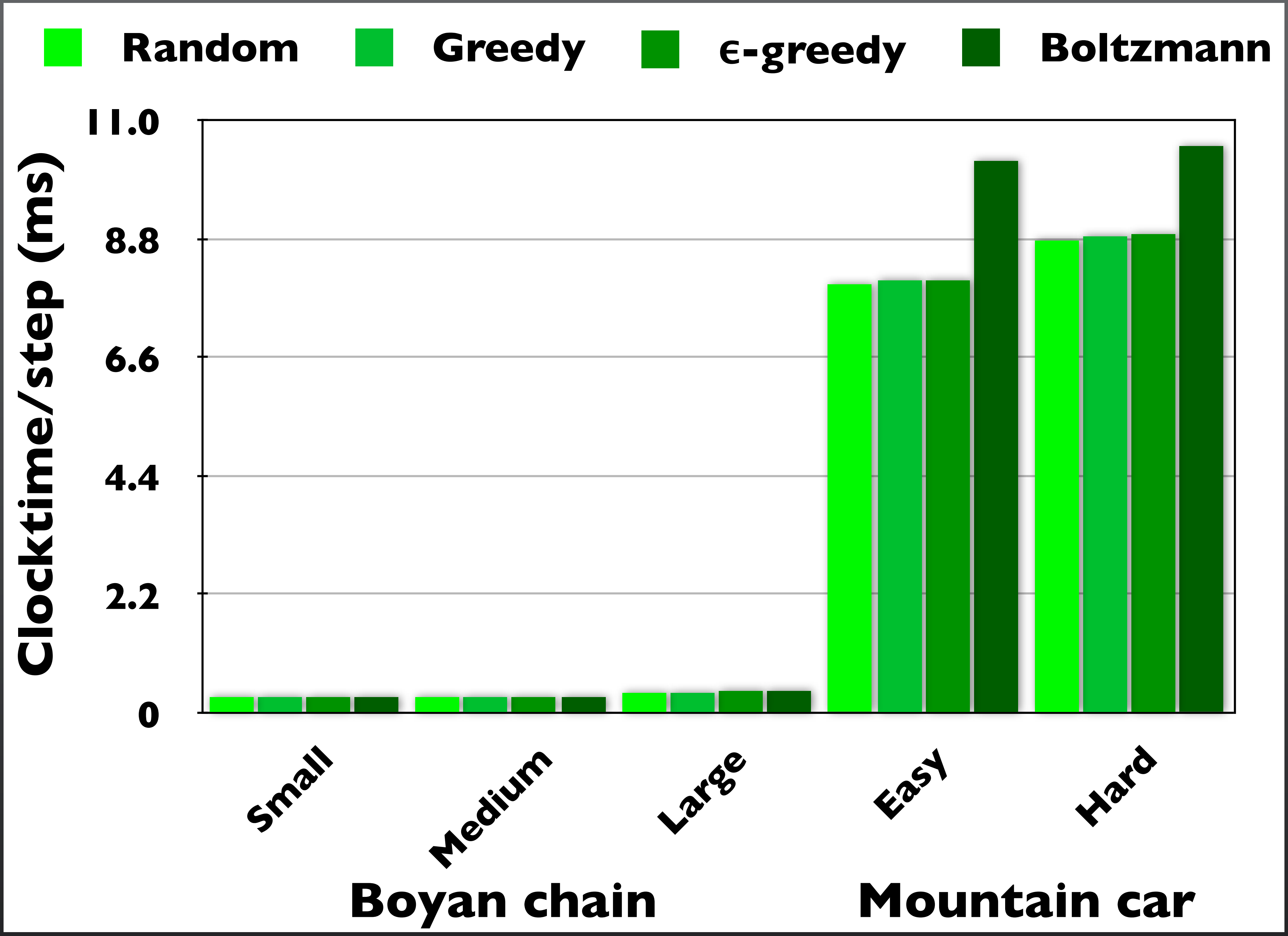
Empirical Results



Empirical Results



Running Time



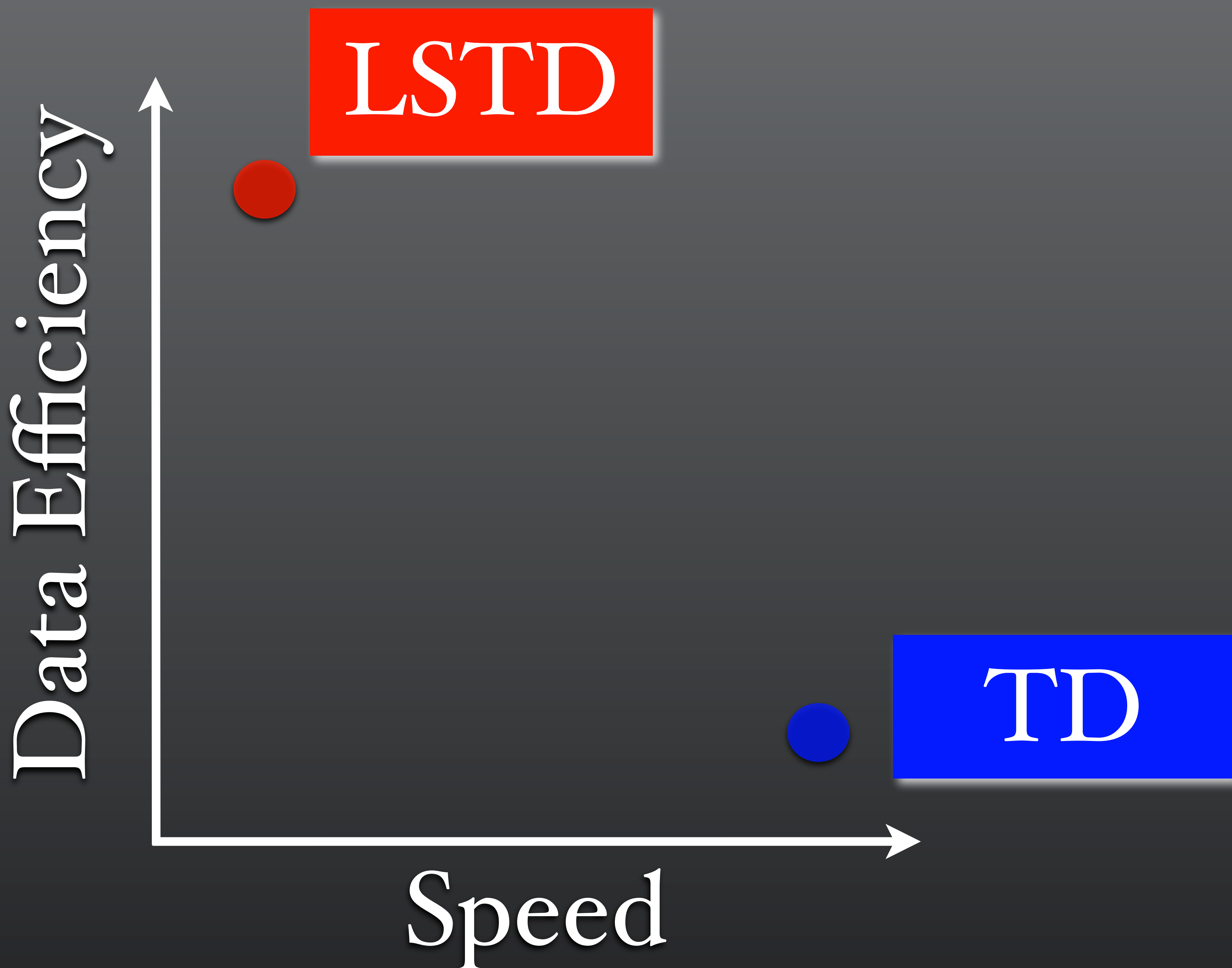
Outline

Outline

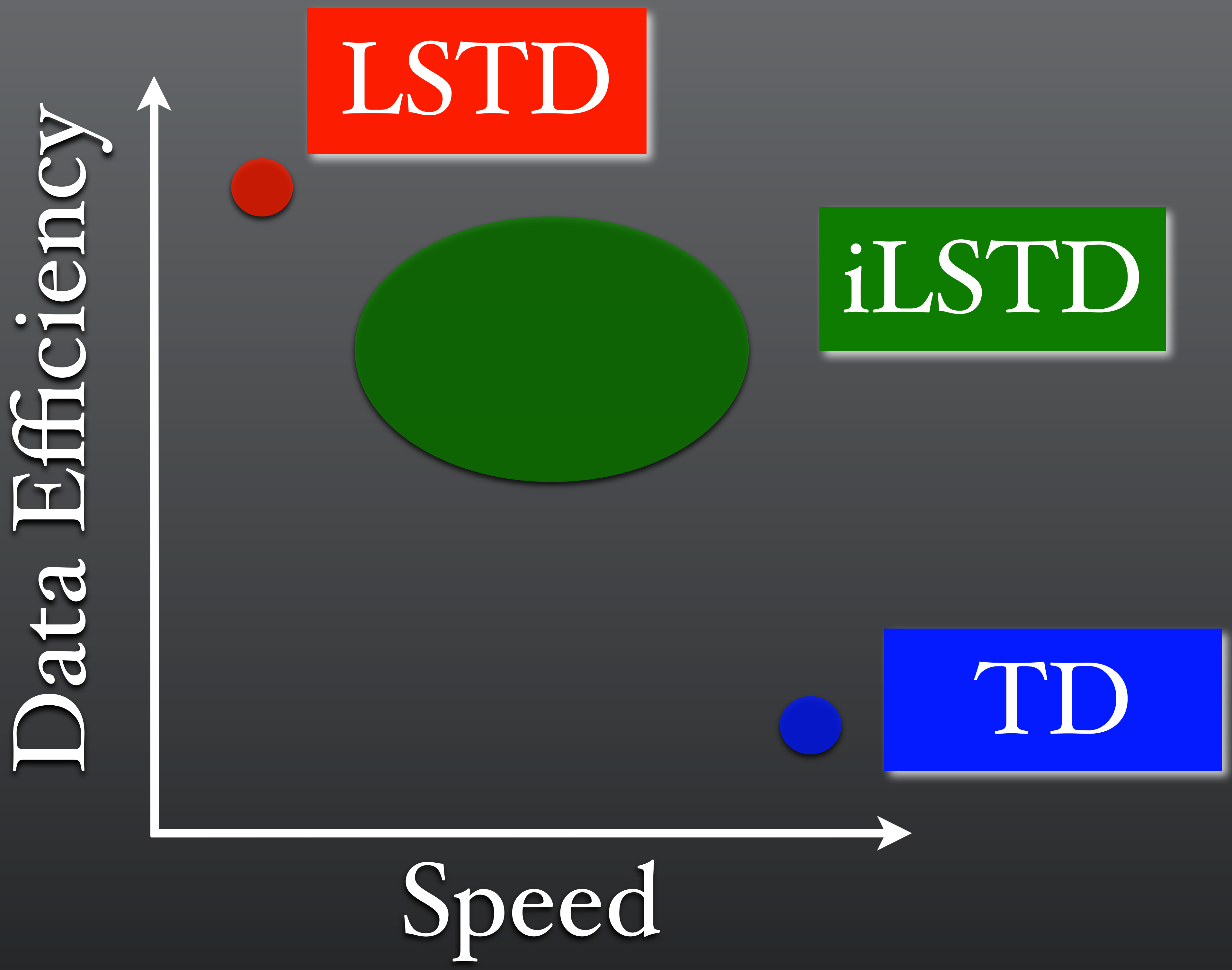
- Motivation
- Introduction
- The New Approach
- Eligibility Traces
- Dimension Selection
- Conclusion ←

Conclusion

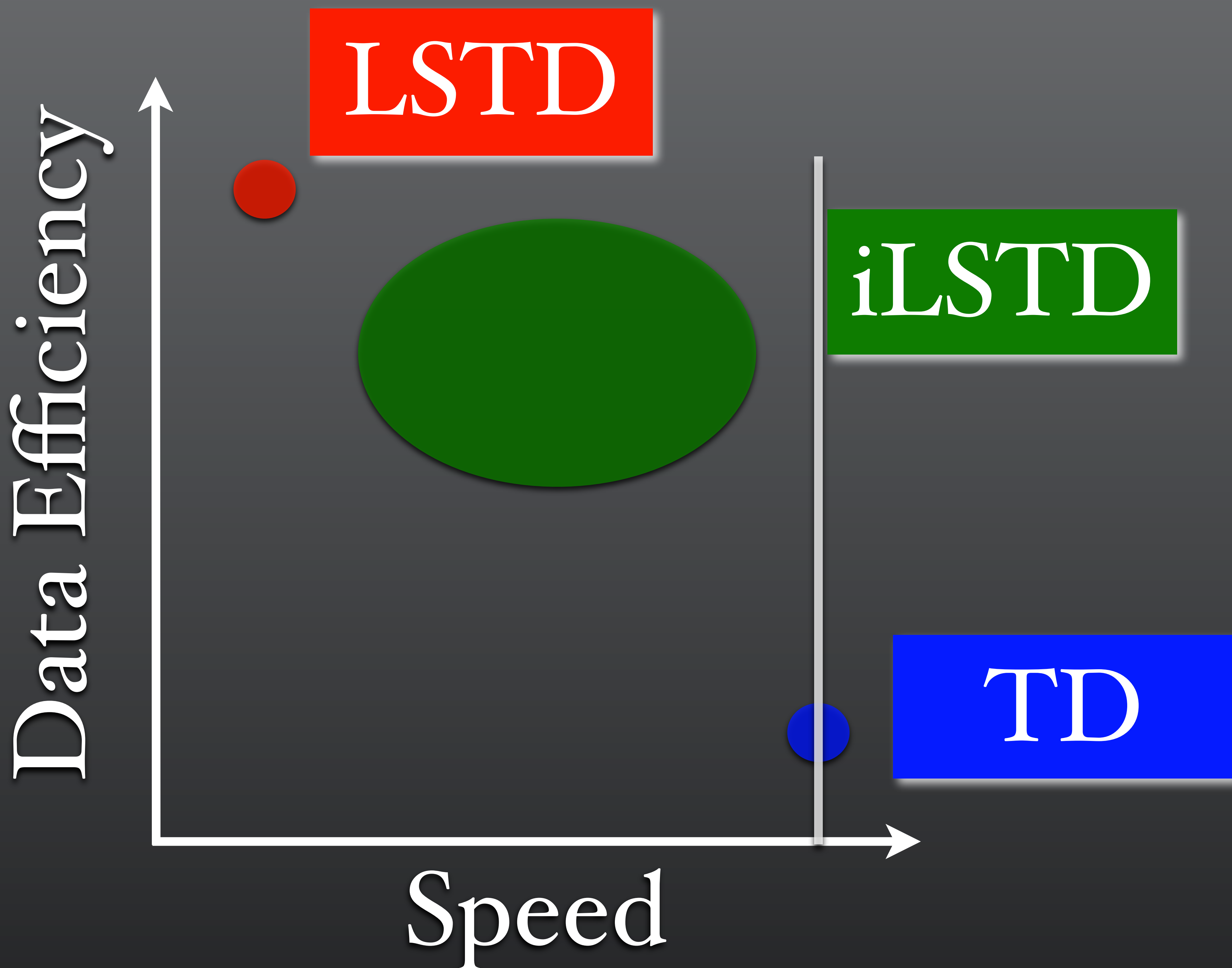
Conclusion



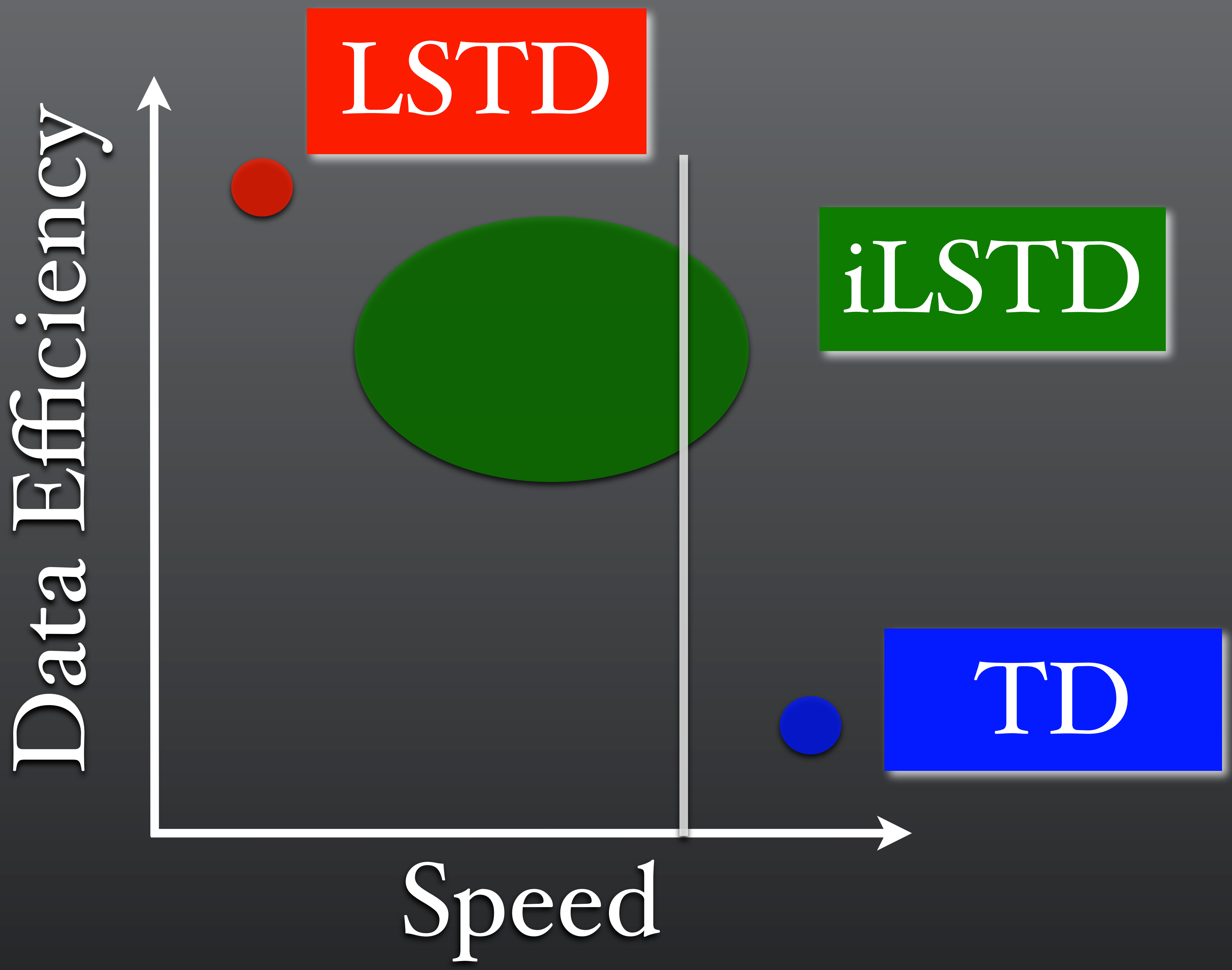
Conclusion



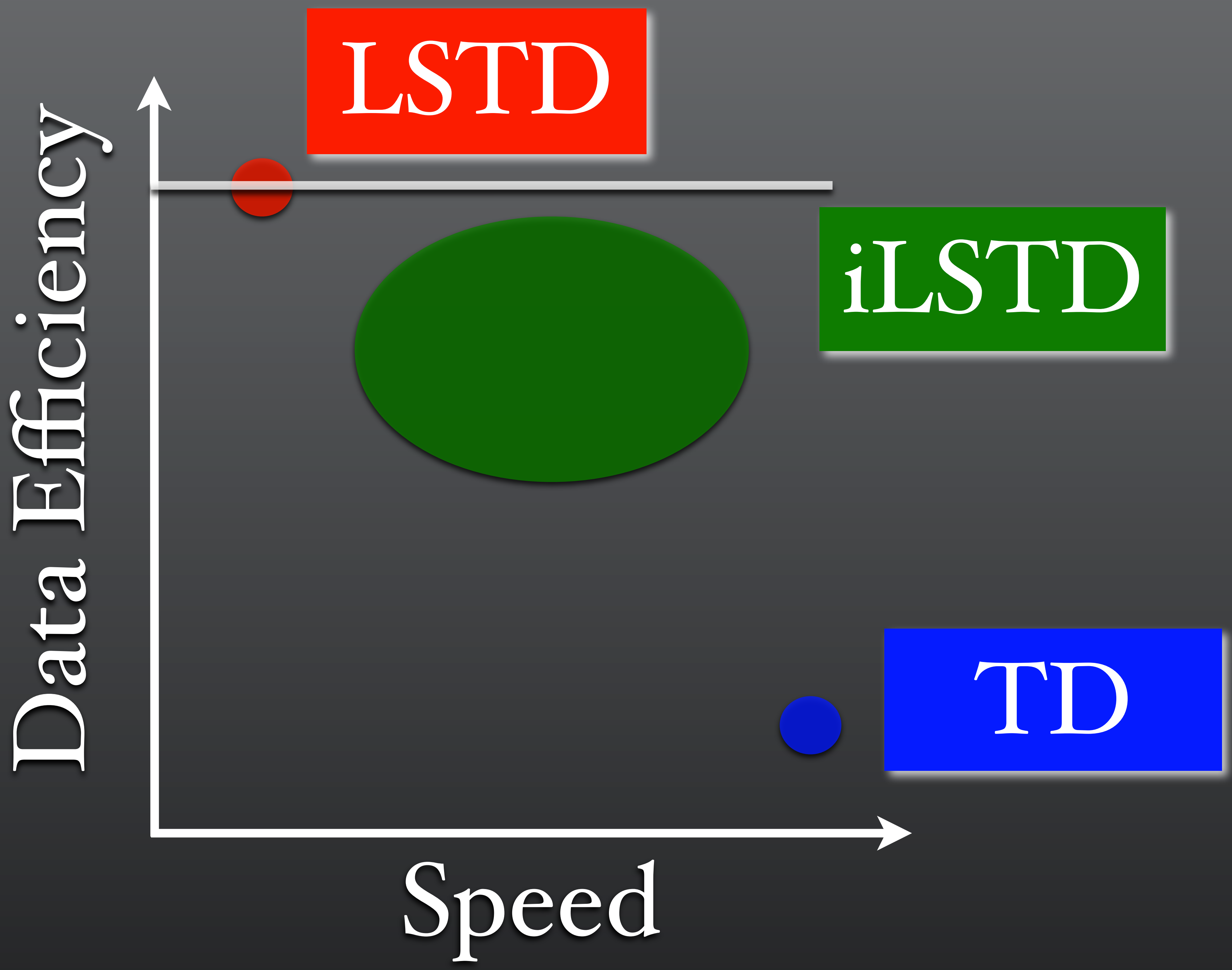
Conclusion



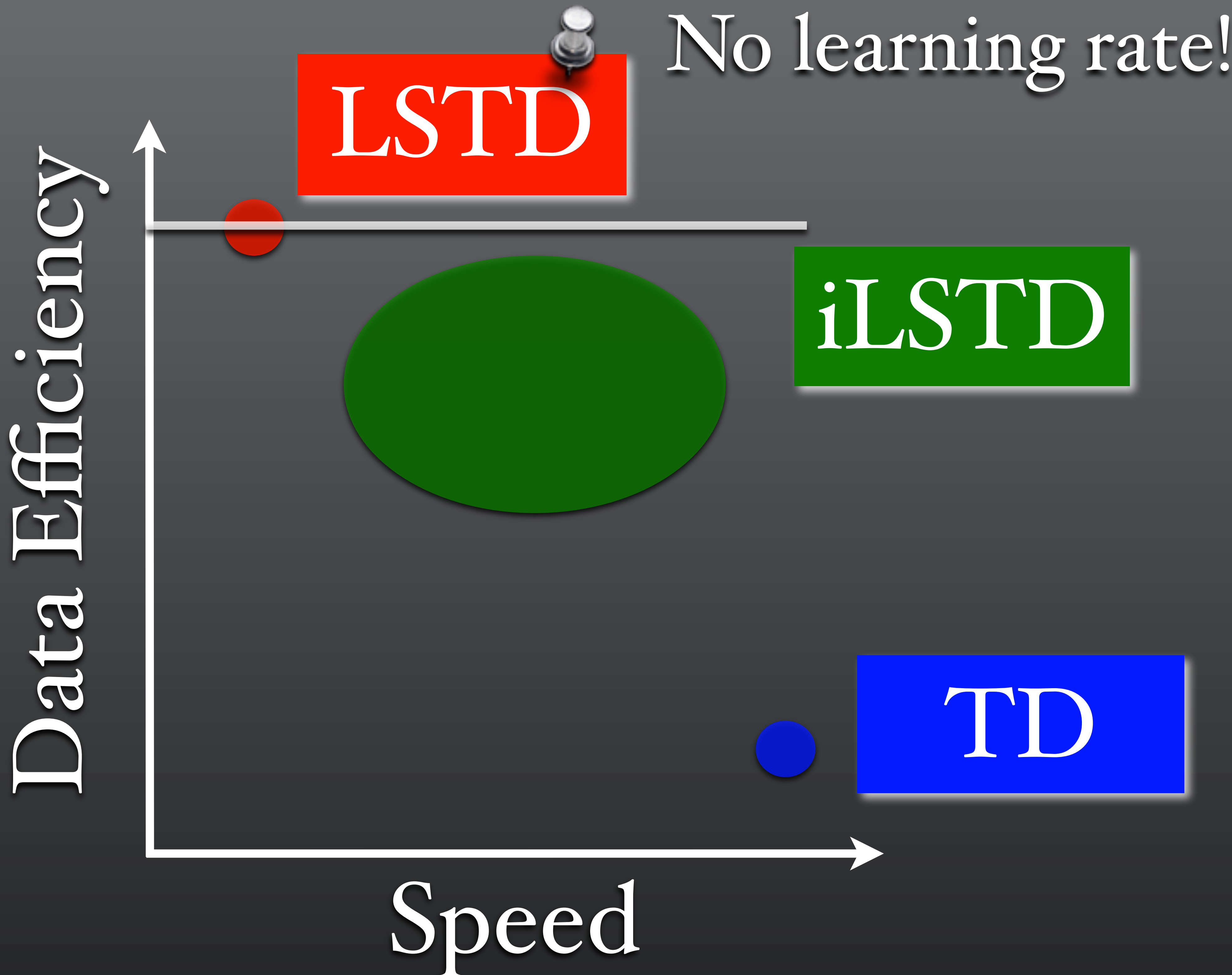
Conclusion



Conclusion



Conclusion



Questions ?

Questions ...

Questions ...

- What if someone uses batch-LSTD?

Questions ...

- What if someone uses batch-LSTD?
- Why iLSTD takes simple descent?

Questions ...

- What if someone uses batch-LSTD?
- Why iLSTD takes simple descent?
- Hmm ... What about control?

Thanks ...