Incremental Least-Square Temporal Difference Learning (iLSTD)

Alborz Geramifard, Michael Bowling, Richard Sutton

UNIVERSITY OF ALBERTA

{alborz, bowling, sutton}@cs.ualberta.ca

Outline

- Introduction
- Least-Square Methods
- iLSTD (Algorithm & Properties)
- Results
- Discussion

Outline

- Introduction
- Least-Square Methods
- iLSTD (Algorithm & Properties)
- Results
- Discussion

Notations

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

Introduction

- Markov Decision Process (MDP)

$$(\mathcal{S}, \mathcal{A}, \mathcal{P}^a_{ss'}, \mathcal{R}^a_{ss'}, \gamma)$$

- We focus on online policy evaluation

Introduction

- Markov Decision Process (MDP)

$$(\mathcal{S}, \mathcal{A}, \mathcal{P}^a_{ss'}, \mathcal{R}^a_{ss'}, \gamma)$$

- We focus on online policy evaluation

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

$$V^\pi(s) \quad = \quad \sum_a \pi(s,a) \sum_{s'} \mathcal{P}^a_{ss'} \left[\mathcal{R}^a_{ss'} + \gamma V^\pi(s')\right]$$

[Sutton, Barto 98]

Introduction

- Tabular Case

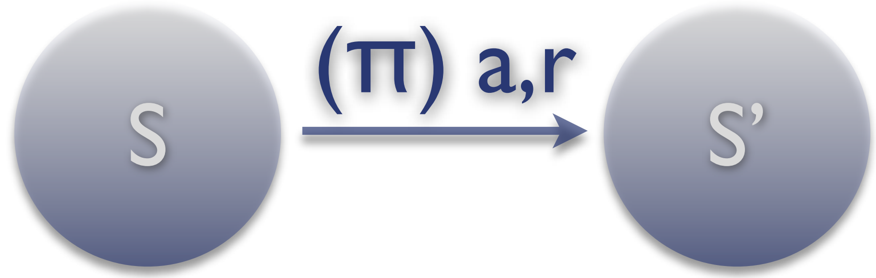
$$\delta_t(V) = r_{t+1} + \gamma V(s_{t+1}) - V(s_t).$$

- Using Linear Function Approximation

$$\begin{aligned}
\boldsymbol{\theta}_{t+1} &= \boldsymbol{\theta}_t + \alpha_t \mathbf{u}_t(\boldsymbol{\theta}_t), \\
\mathbf{u}_t(\boldsymbol{\theta}) &= \phi(s_t)\delta_t(V_{\boldsymbol{\theta}}).
\end{aligned}$$

Introduction

- Tabular Case



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

- Using Linear Function Approximation

$$\begin{aligned} \boldsymbol{\theta}_{t+1} &= \boldsymbol{\theta}_t + \alpha_t \mathbf{u}_t(\boldsymbol{\theta}_t), \\ \mathbf{u}_t(\boldsymbol{\theta}) &= \phi(s_t)\delta_t(V_{\boldsymbol{\theta}}). \end{aligned}$$

Outline

- Introduction
- Least-Square Methods
- iLSTD (Algorithm & Properties)
- Results
- Discussion

Outline

- Introduction
- Least-Square Methods
- iLSTD (Algorithm & Properties)
- Results
- Discussion

Least-Square Methods

- It minimizes the mean squared TD errors over all past experiences.

$$E_t(\theta) = \frac{1}{t}\sum_t \delta_t^2(\theta)$$

- It takes advantage of all experiment and does the update (Sum of the TD updates) [Bradtke, Barto 96]

$$\boldsymbol{\mu}_t(\boldsymbol{\theta}) \quad = \quad \sum_{i=1}^{t} \phi_t \delta_t(V_{\boldsymbol{\theta}})$$

Least-Square Methods

- It minimizes the mean squared TD errors over all past experiences.

$$E_t(\theta) = \frac{1}{t}\sum_t \delta_t^2(\theta)$$

- It takes advantage of all experiment and does the update (Sum of the TD updates) [Bradtke, Barto 96]

$$\boldsymbol{\mu}_t(\boldsymbol{\theta}) \quad = \quad \sum_{i=1}^{t} \phi_t \delta_t(V_{\boldsymbol{\theta}})$$

We call it "TD Gradient"

Least-Square Methods

- By plugging the definitions, we will have:

$$\begin{aligned}
\boldsymbol{\mu}_t(\boldsymbol{\theta}) &= \left(\underbrace{\sum_{i=1}^{t} \phi_t r_{t+1}}_{\mathbf{b}_t} - \underbrace{\sum_{i=1}^{t} \phi_t (\phi_t - \gamma \phi_{t+1})^T}_{\mathbf{A}_t} \boldsymbol{\theta} \right) \\
&= (\mathbf{b}_t - \mathbf{A}_t \boldsymbol{\theta}).
\end{aligned}$$

$$\boldsymbol{\theta}_{t+1} = \mathbf{A}_t^{-1} \mathbf{b}_t.$$

[Bradtke, Barto 96]

Least-Square Methods

- By plugging the definitions, we will have:

$$\begin{aligned}\boldsymbol{\mu}_t(\boldsymbol{\theta}) &= \left(\underbrace{\sum_{i=1}^{t}\phi_t r_{t+1}}_{\mathbf{b}_t} - \underbrace{\sum_{i=1}^{t}\phi_t(\phi_t - \gamma\phi_{t+1})^T}_{\mathbf{A}_t}\boldsymbol{\theta}\right)\\ &= (\mathbf{b}_t - \mathbf{A}_t\boldsymbol{\theta}).\end{aligned}$$

$$\xrightarrow{\text{LSTD}} \quad \boldsymbol{\theta}_{t+1} = \mathbf{A}_t^{-1}\mathbf{b}_t.$$

[Bradtke, Barto 96]

Least-Square Methods

$$\boldsymbol{\theta}_{t+1} = \mathbf{A}_t^{-1}\mathbf{b}_t.$$

Least-Square Methods

$$\boldsymbol{\theta}_{t+1} = \mathbf{A}_t^{-1}\mathbf{b}_t.$$

- Pros
  - Minimized the sum of TD errors with respect to all of the past experiences.

Least-Square Methods

$$\boldsymbol{\theta}_{t+1} = \mathbf{A}_t^{-1}\mathbf{b}_t.$$

- Pros
  - Minimized the sum of TD errors with respect to all of the past experiences.
- Cons
  - Needs at least
    needs $O(n^2)$ computation per time step (Using iterative matrix inversion)
  - *n* is the number of features which can be potentially large.

Outline

- Introduction
- Least-Square Methods
- iLSTD (Algorithm & Properties)
- Results
- Discussion

Outline

- Introduction
- Least-Square Methods
- iLSTD (Algorithm & Properties)
- Results
- Discussion

iLSTD

iLSTD

- Can we do something about the inverse ?

iLSTD

- Can we do something about the inverse ?
- We are interested in case of having k features "on" at any given moment (Tile Coding, RBFs, etc.) where k << n.

iLSTD

- Can we do something about the inverse ?
- We are interested in case of having k features “on” at any given moment (Tile Coding, RBFs, etc.) where k << n.
- We do not have to compute the exact solution since we change **A** matrix and **b** vector on each iteration.

iLSTD

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

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

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

iLSTD

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

- Incremental Computation

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

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

[Bradtke, Barto 96]

iLSTD

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

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

iLSTD

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

- Incremental Computation (when **A** and **b** are changed).

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

* Note that θ is fixed.

iLSTD

- Incremental Computation (When θ is changed).

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \Delta\boldsymbol{\theta}_t$$

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

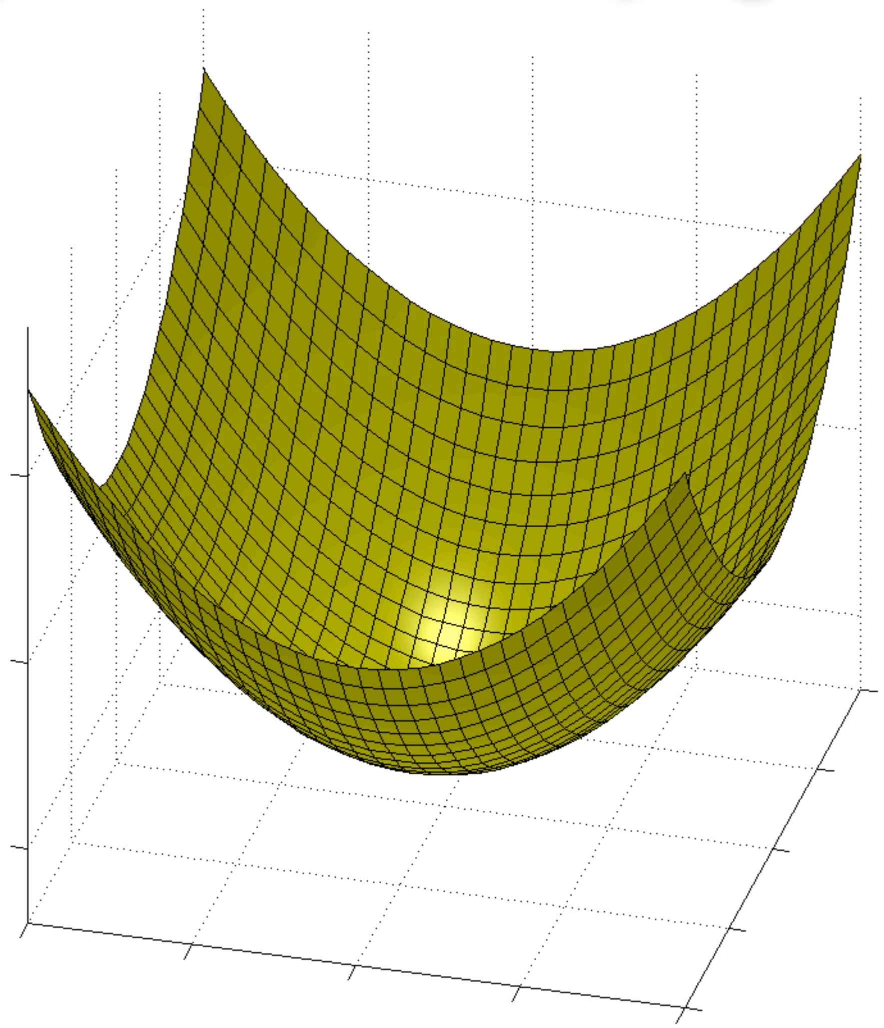
* Note that **A** is fixed.

iLSTD

- Use Gradient Descent in "best" dimension to update θ (w.r.t TD Gradient Vector)
- Similar to prioritized sweeping idea

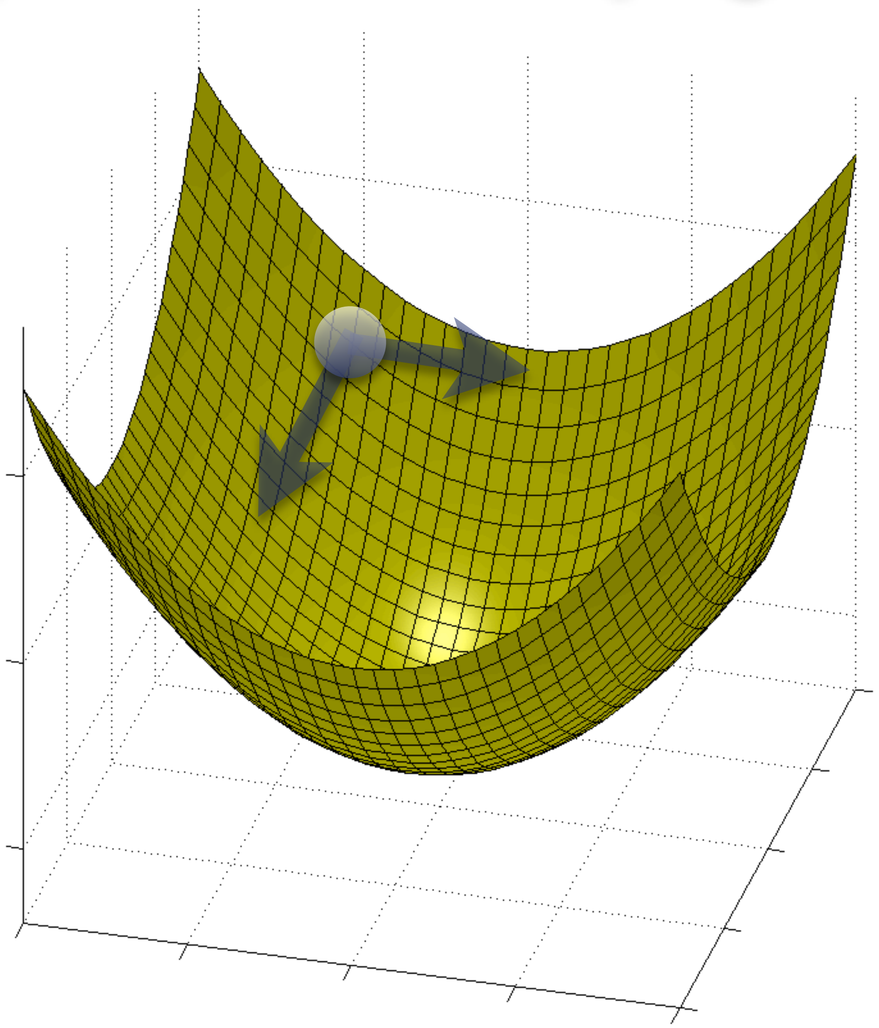
iLSTD

- Use Gradient Descent in “best” dimension to update θ (w.r.t TD Gradient Vector)
- Similar to prioritized sweeping idea



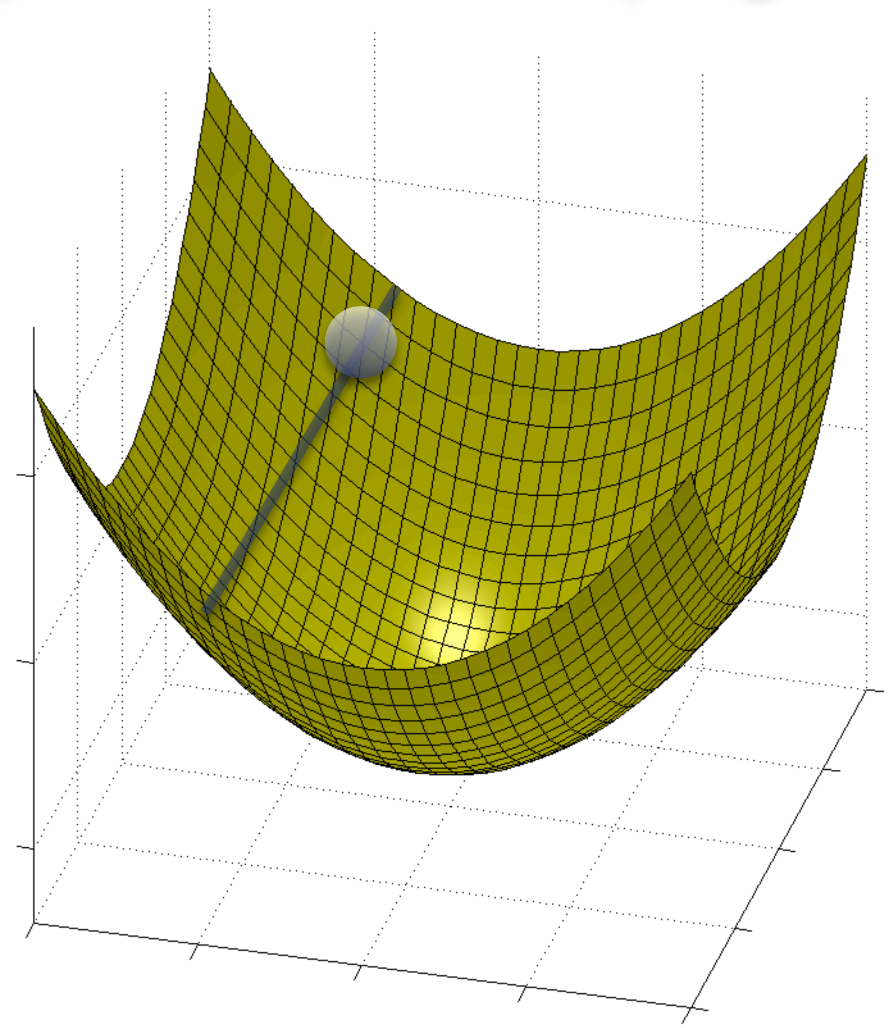
iLSTD

- Use Gradient Descent in "best" dimension to update θ (w.r.t TD Gradient Vector)
- Similar to prioritized sweeping idea



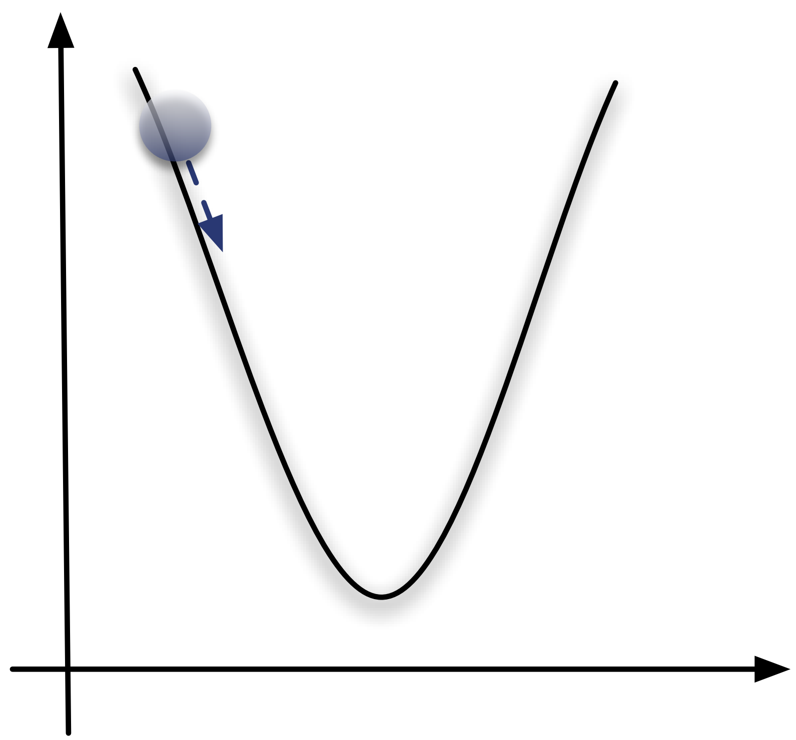
iLSTD

- Use Gradient Descent in "best" dimension to update θ (w.r.t TD Gradient Vector)
- Similar to prioritized sweeping idea



iLSTD

- Use Gradient Descent in "best" dimension to update θ (w.r.t TD Gradient Vector)
- Similar to prioritized sweeping idea



iLSTD Algorithm

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

[Geramifard, Bowling, Sutton 06]

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'
    $t \leftarrow t + 1$
    $\Delta\mathbf{b} \leftarrow \boldsymbol{\phi}(s) r$
    $\Delta\mathbf{A} \leftarrow \boldsymbol{\phi}(s)(\boldsymbol{\phi}(s) - \gamma\boldsymbol{\phi}(s'))^T$
    $\mathbf{A} \leftarrow \mathbf{A} + \Delta\mathbf{A}$
    $\boldsymbol{\mu} \leftarrow \boldsymbol{\mu} + \Delta\mathbf{b} - (\Delta\mathbf{A})\boldsymbol{\theta}$

Updating **A**,**b** and **μ** according to the interaction

[Geramifard, Bowling, Sutton 06]

iLSTD Algorithm

```
s ← s_0, A ← 0, μ ← 0, t ← 0
Initialize θ arbitrarily
repeat
    Take action according to π and observe r, s'
    t ← t + 1
    Δb ← φ(s)r
    ΔA ← φ(s)(φ(s) − γφ(s'))^T
    A ← A + ΔA
    μ ← μ + Δb − (ΔA)θ
    for i from 1 to m do
        j ← argmax(|μ_j|)
        θ_j ← θ_j + αμ_j
        μ ← μ − αμ_j A e_i
    end for
end repeat
```

Updating **A**,**b** and **μ** according to the interaction

Updating **μ** according to the change to **θ**

[Geramifard, Bowling, Sutton 06]

iLSTD

- Computational Complexity per time step

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

iLSTD

- Computational Complexity per time step

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

Number of gradient descent iterations

iLSTD

- Computational Complexity per time step

$$O(\underbrace{m}_{\text{Number of gradient descent iterations}}\underbrace{n}_{\text{Number of features}} + k^2)$$

Number of features

Number of gradient descent iterations

iLSTD

- Computational Complexity per time step

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

Maximum number of "On" features

Number of features

Number of gradient descent iterations

Outline

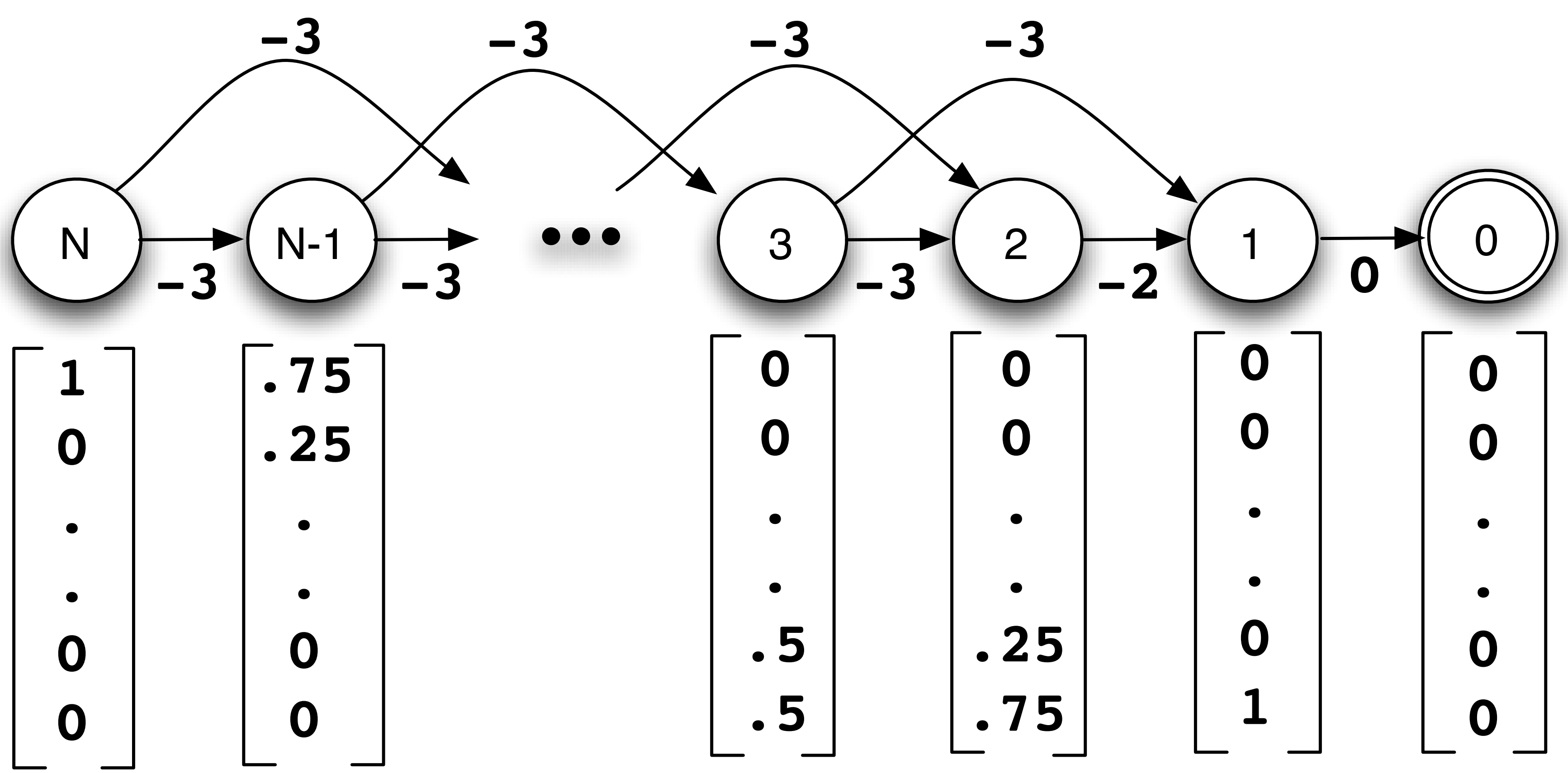
- Introduction
- Least-Square Methods
- iLSTD (Algorithm & Properties)
- Results
- Discussion

Outline

- Introduction
- Least-Square Methods
- iLSTD (Algorithm & Properties)
- Results
- Discussion

Results

- Chain example with correlated features



[Boyan 99]

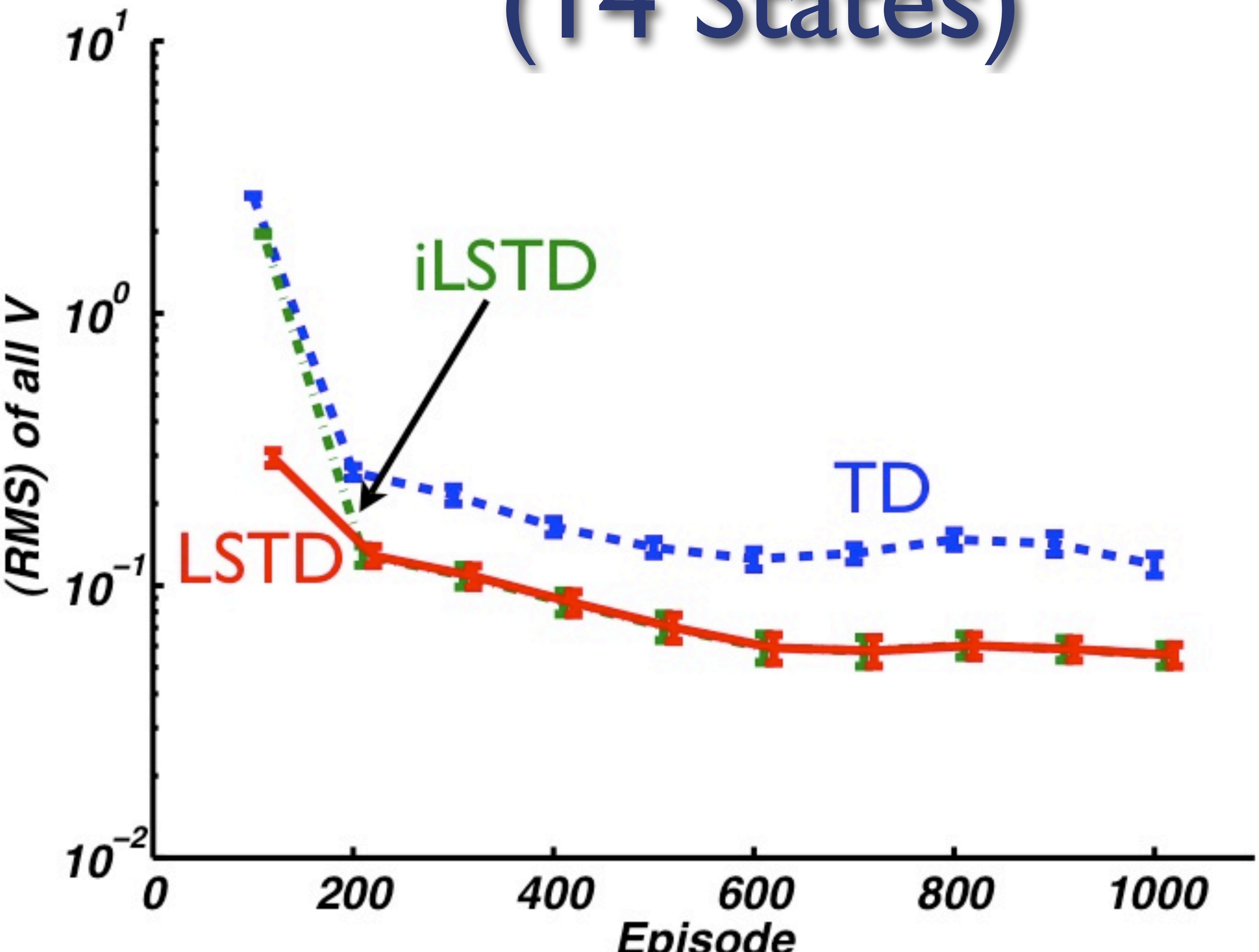
Results

- Parameters:
  - m = 1

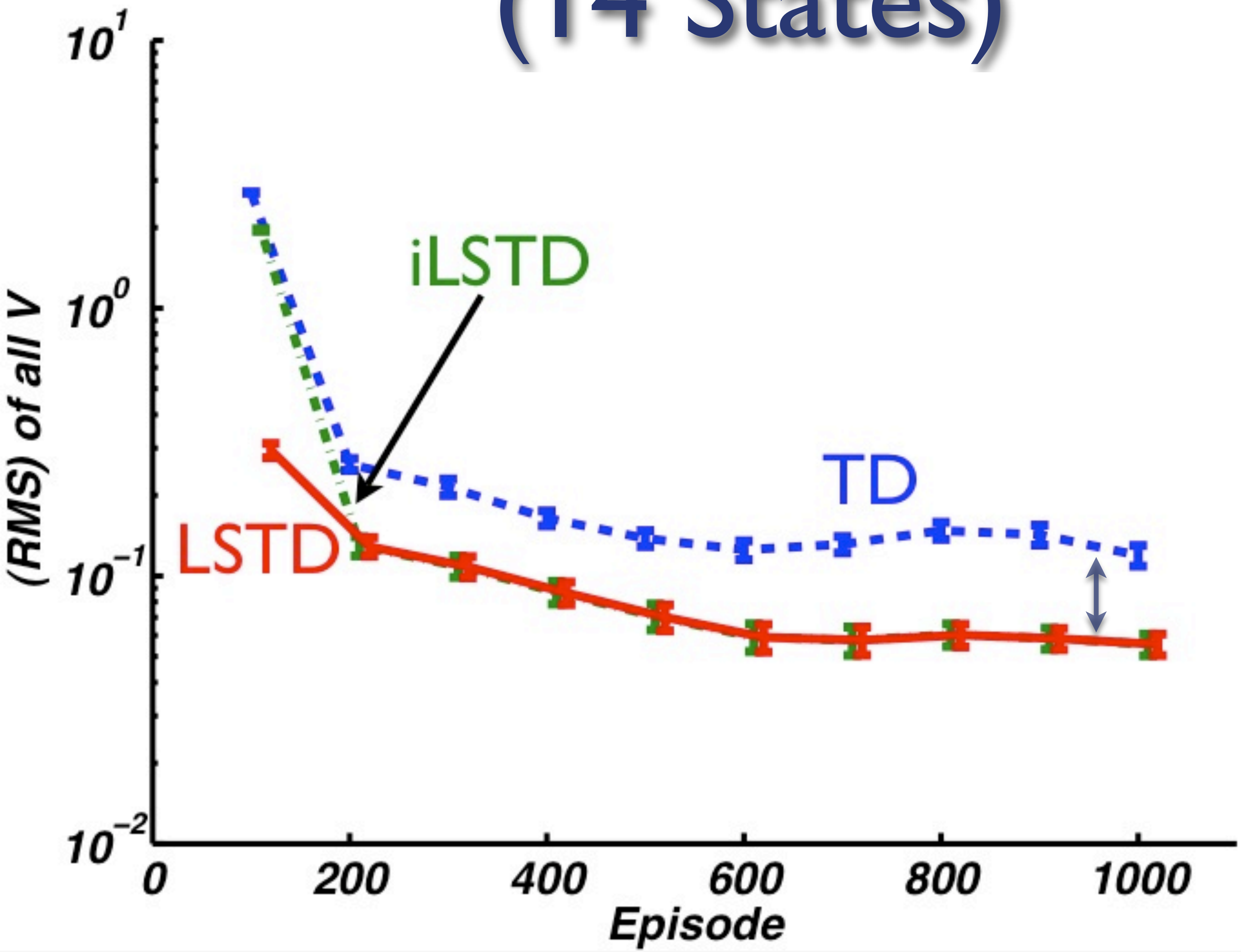
$$\alpha_t = \alpha_0 \frac{N_0 + 1}{N_0 + \text{Episode\#}}$$

  - $\alpha_0 \in \{0.01, 0.1, 1\}$
  - $N_0 \in \{100, 1000, 10^6\}$
  - Best selection for α_0 and N_0
  - Averaged over 30 runs
  - Same random seed for all methods.

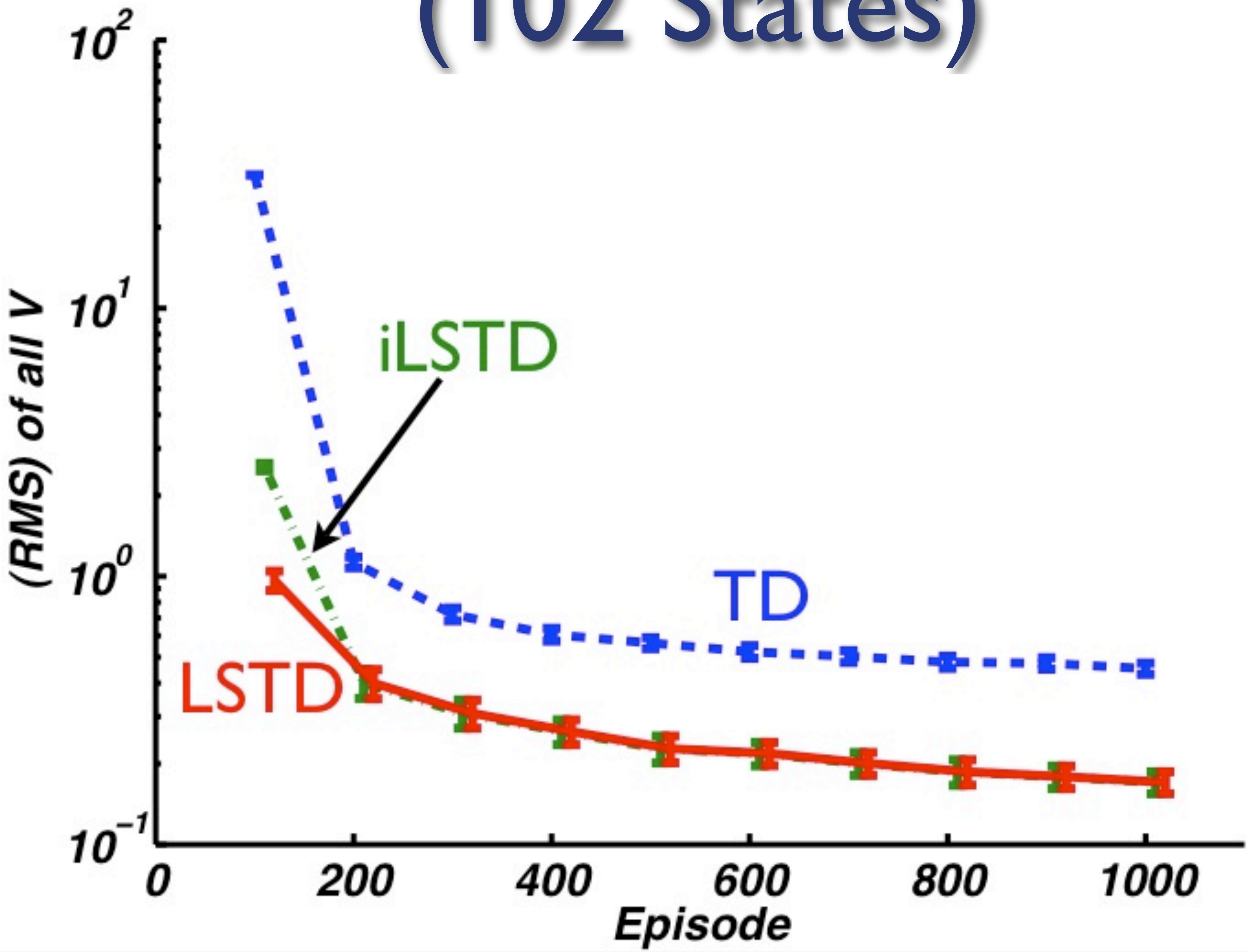
Results - small problem (14 States)



Results - small problem (14 States)

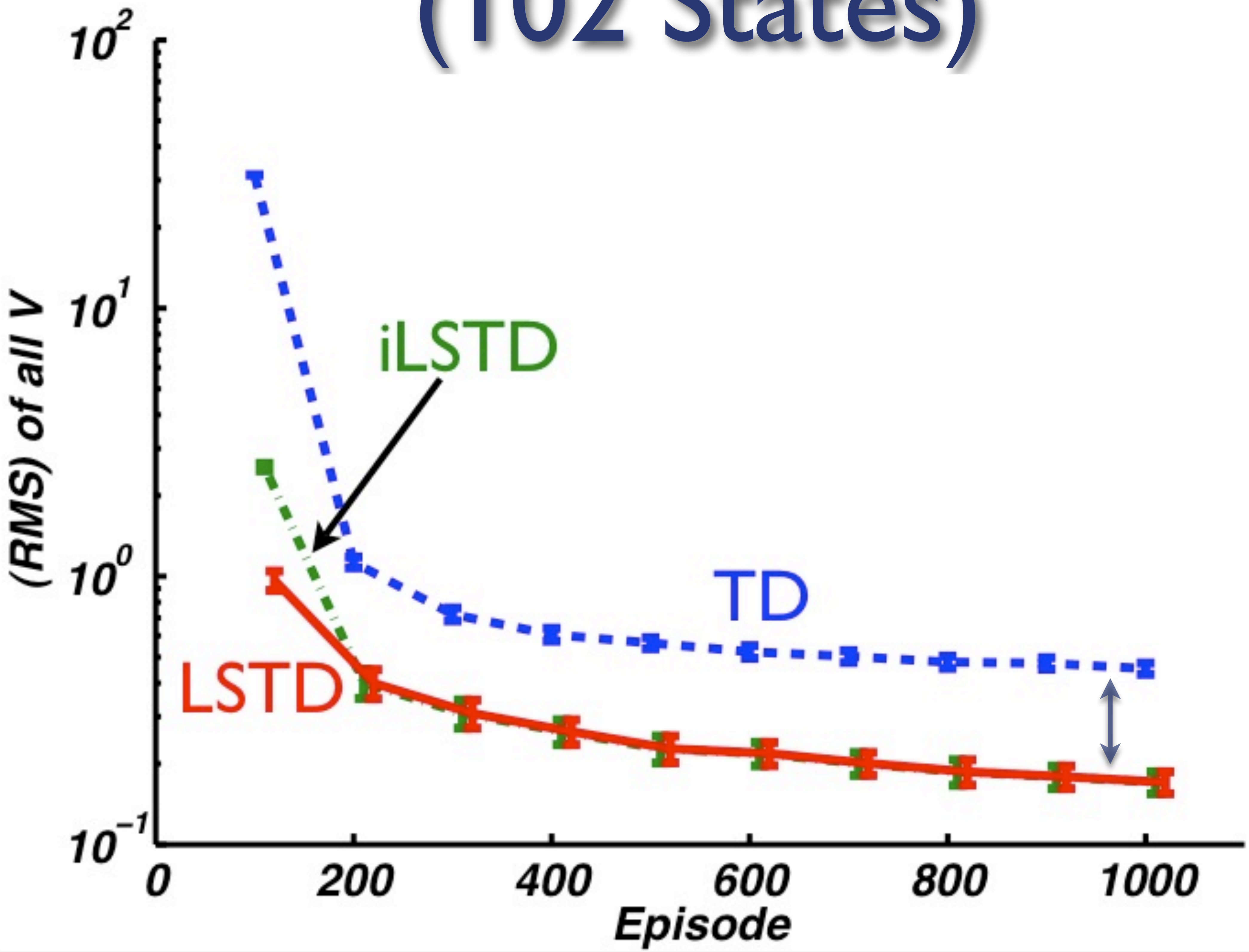


Results - medium problem (102 States)

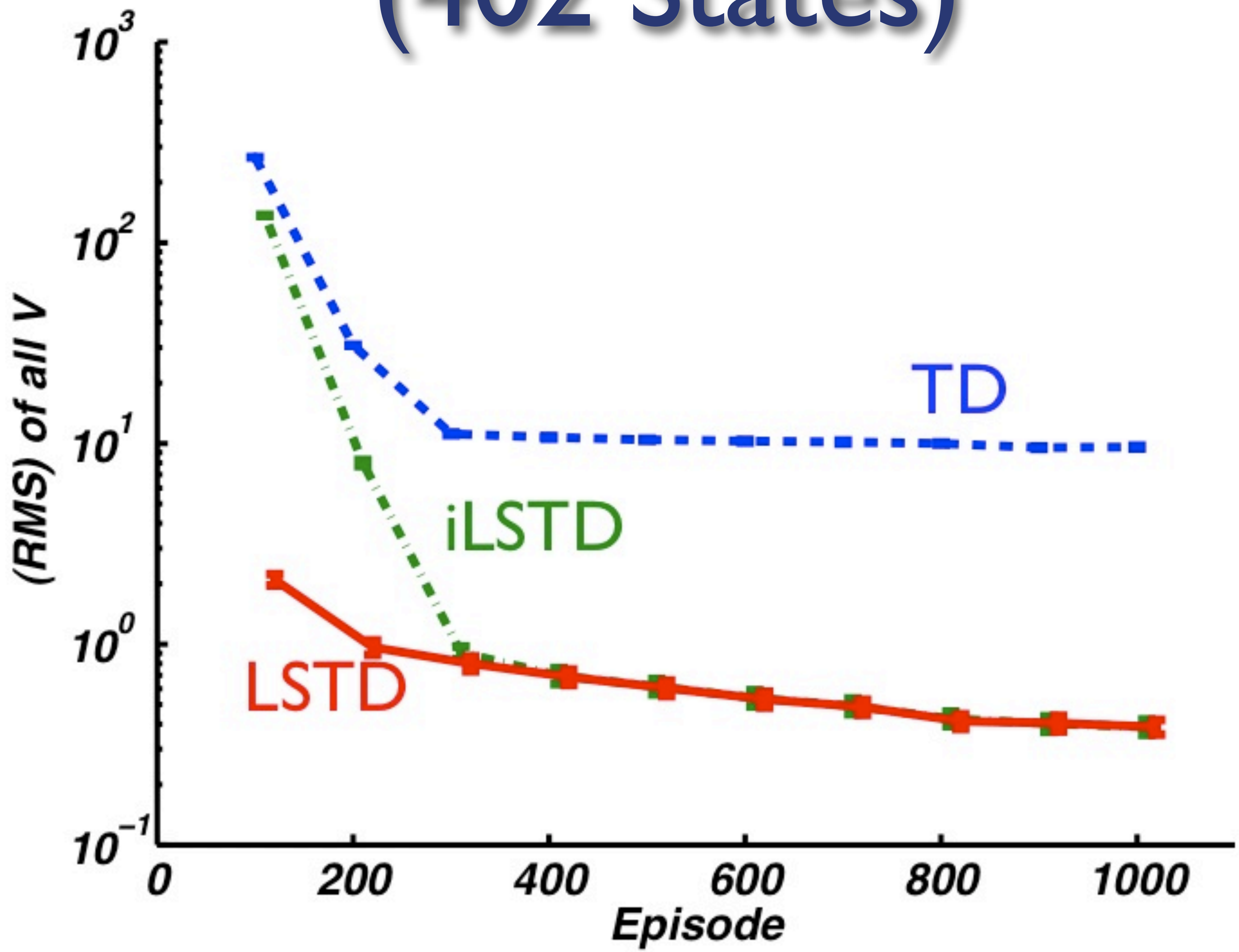


Tuesday, December 13, 2011

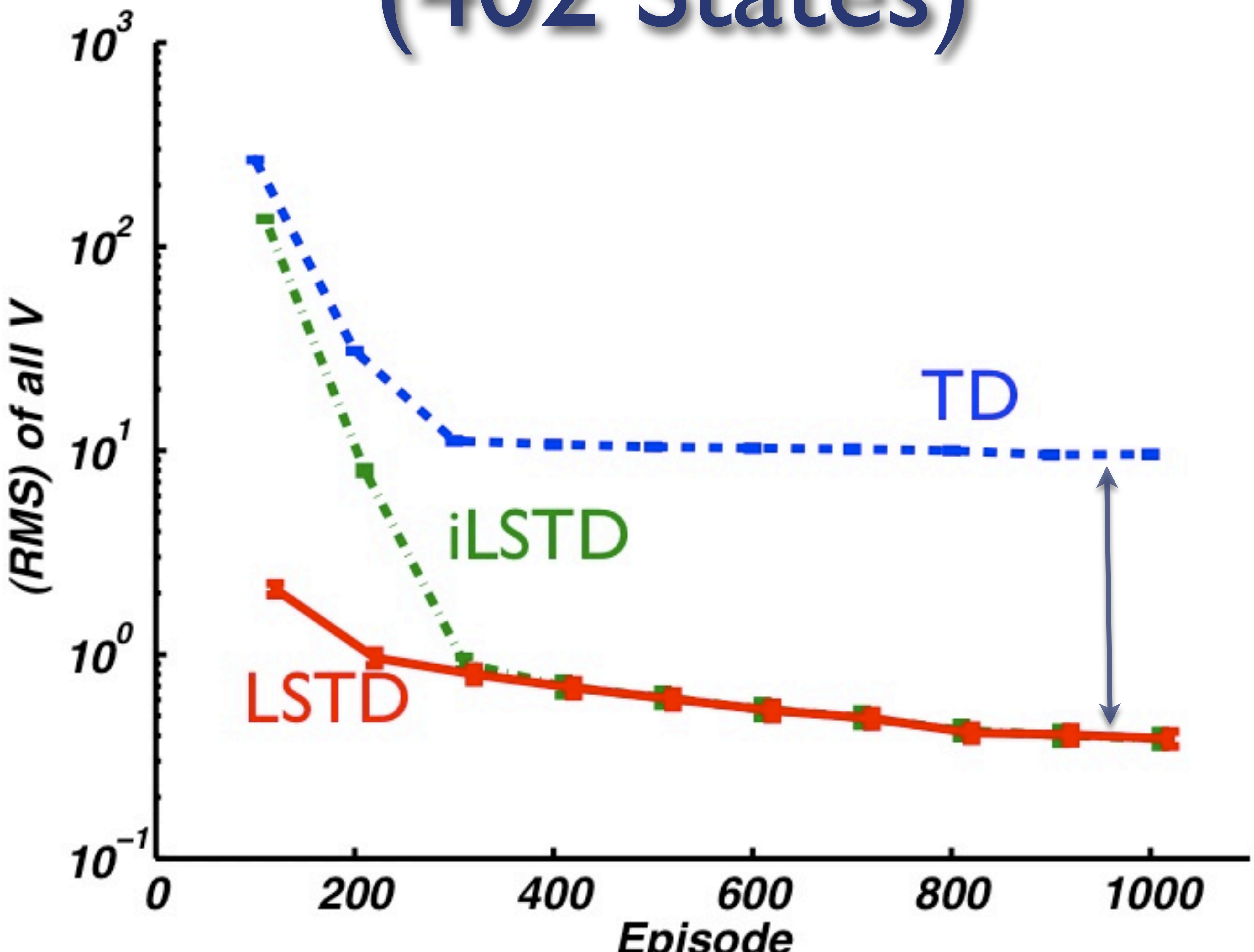
Results - medium problem (102 States)



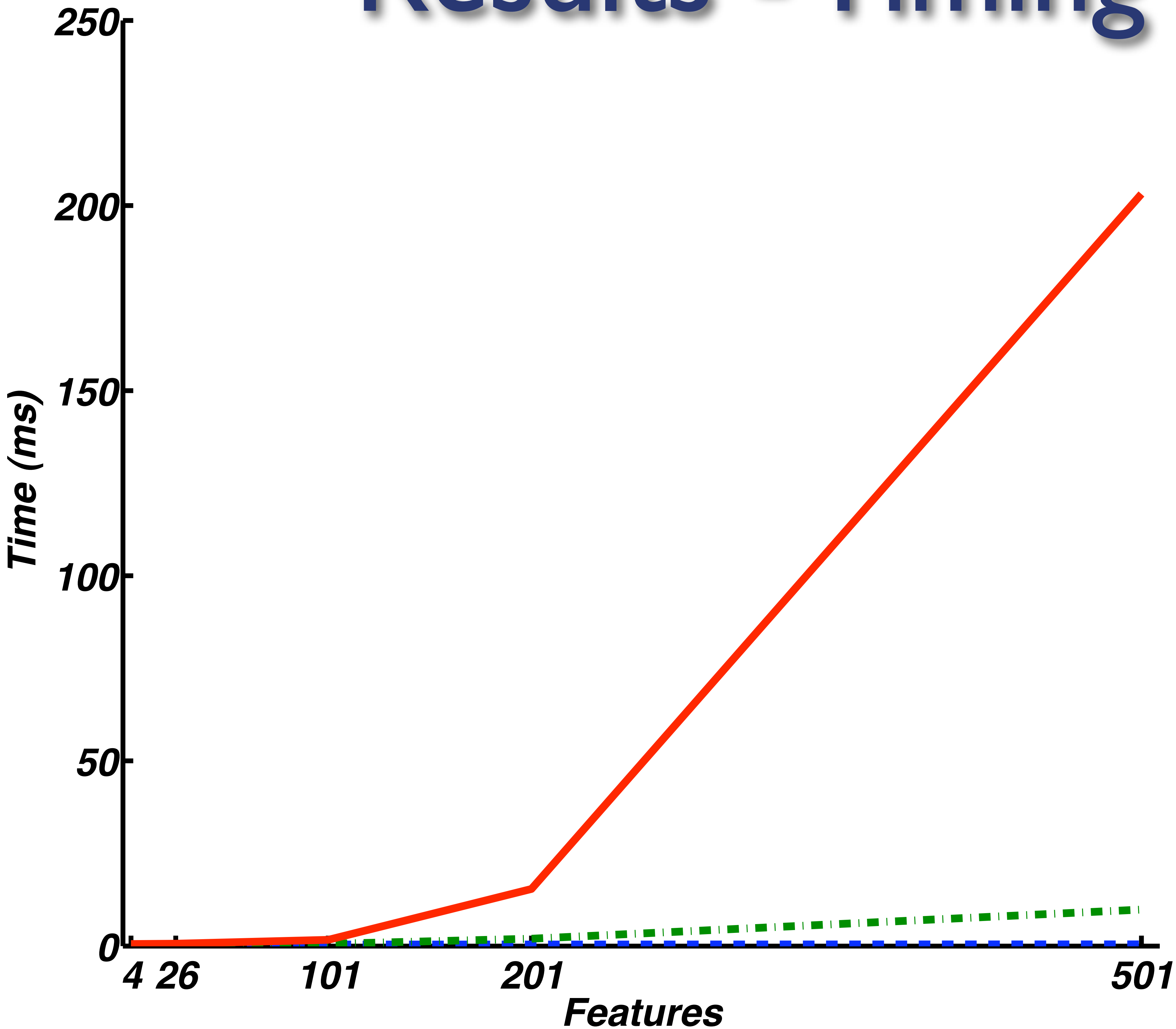
Results - large problem (402 States)



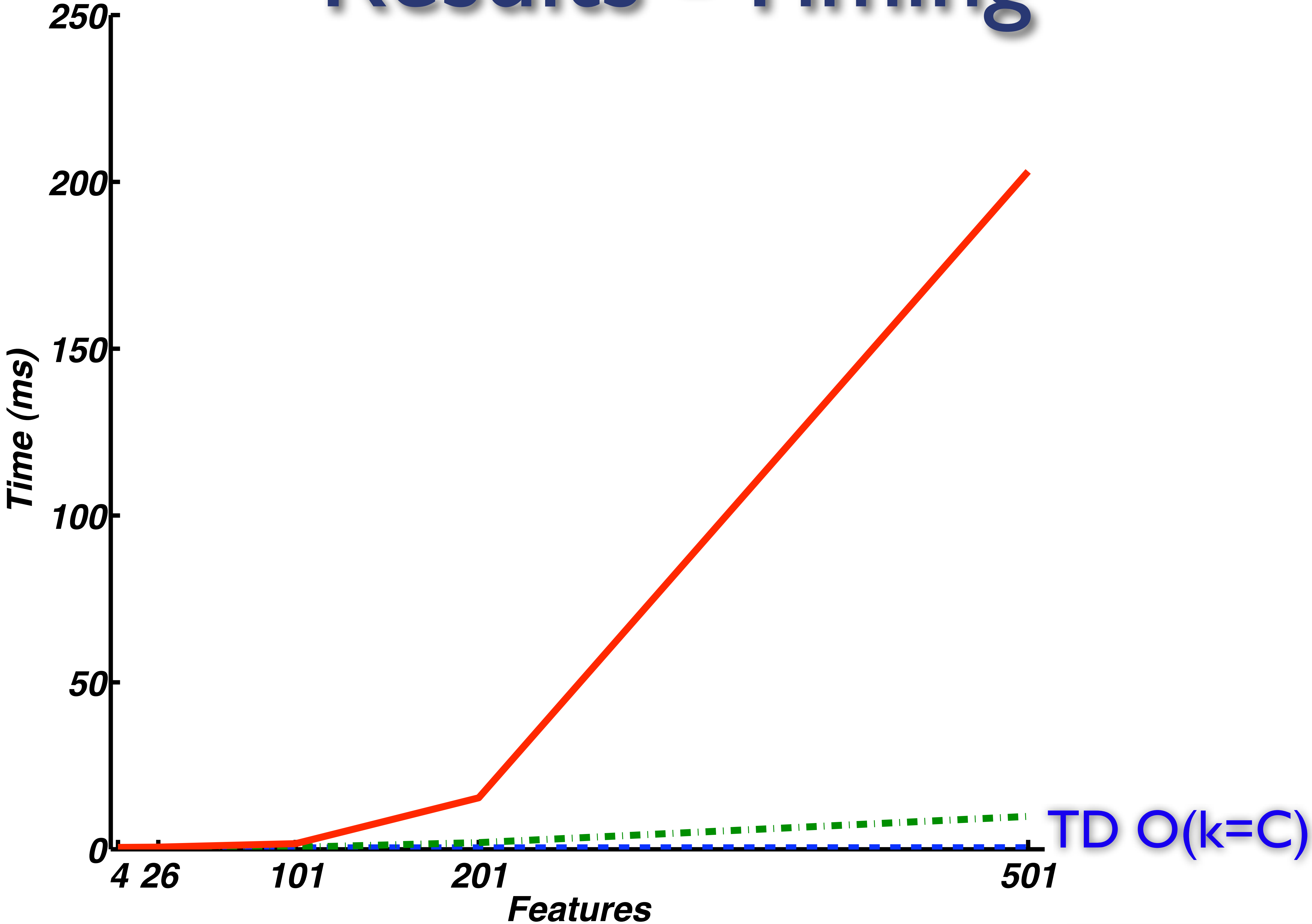
Results - large problem (402 States)



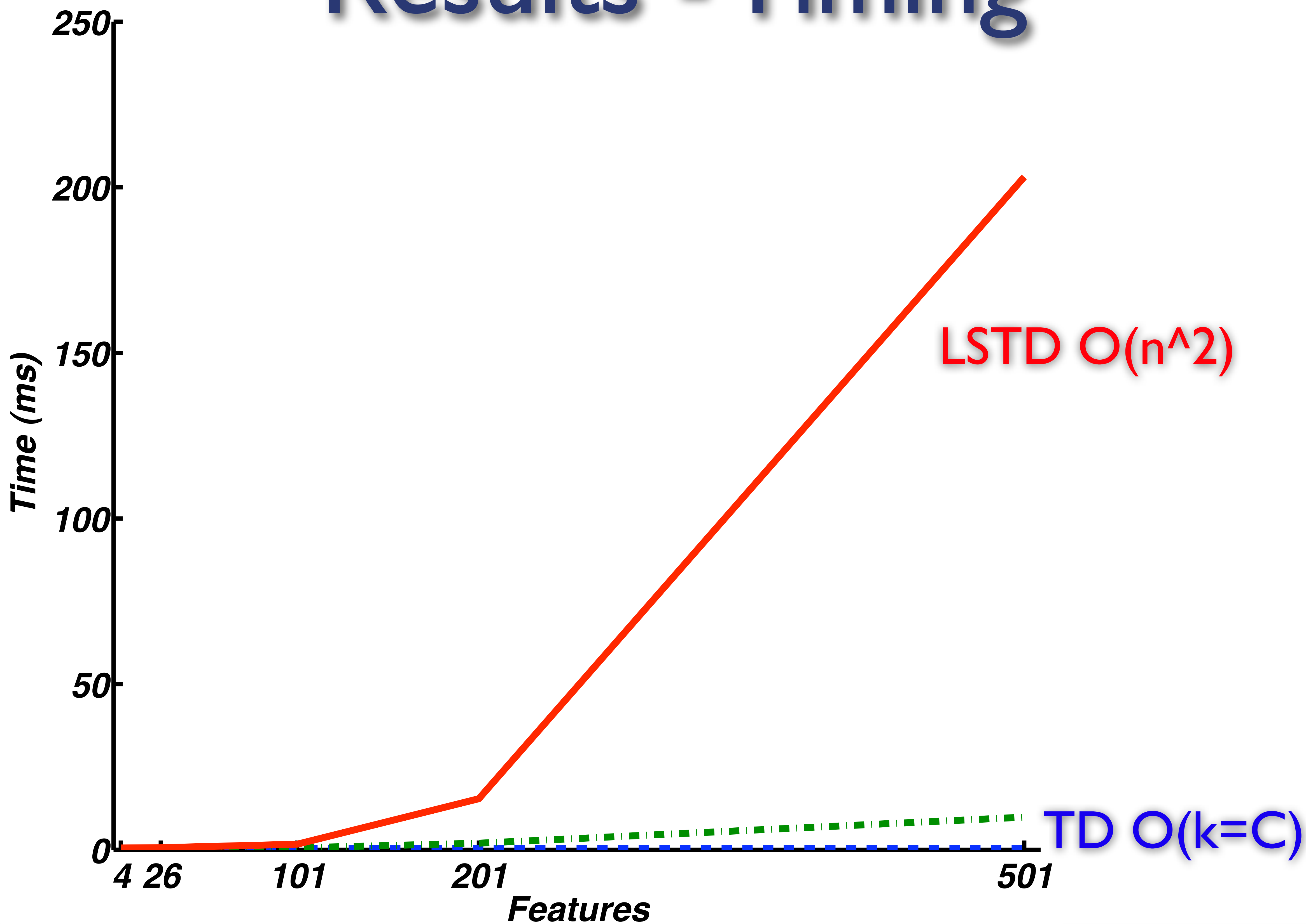
Results - Timing



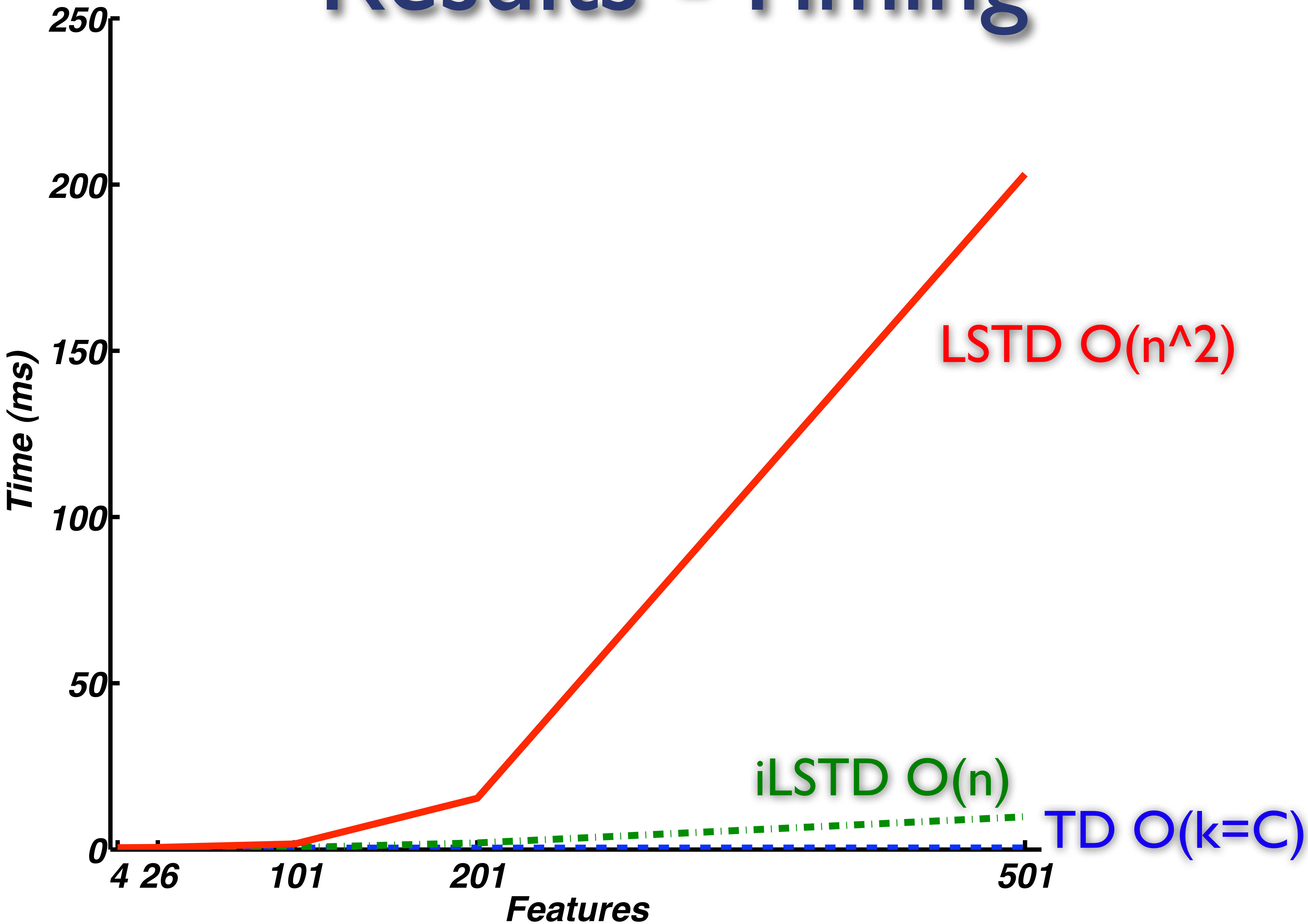
Results - Timing



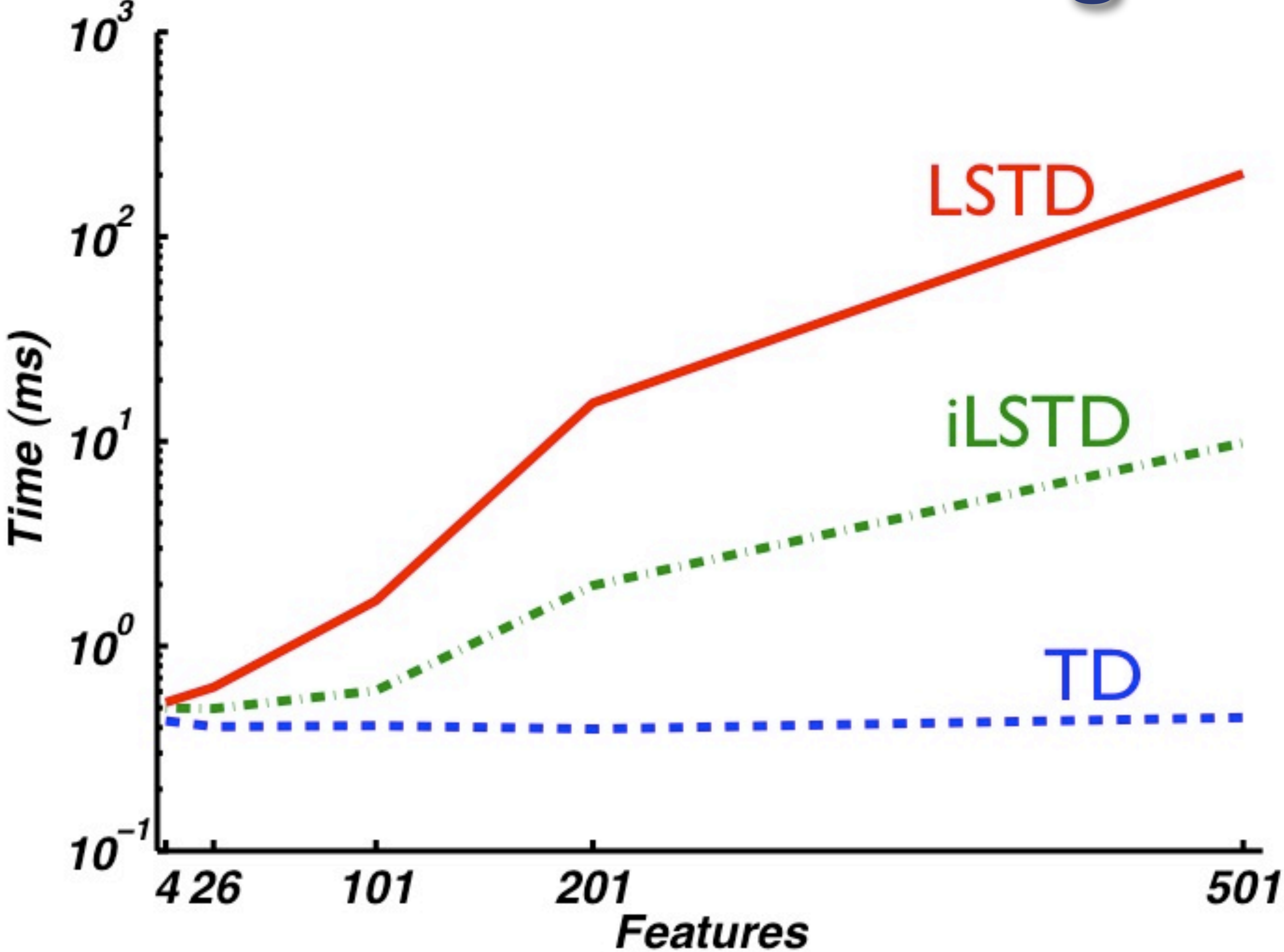
Results - Timing



Results - Timing



Results - Timing



Outline

- Introduction
- Least-Square Methods
- iLSTD (Algorithm & Properties)
- Results
- Discussion

Outline

- Introduction
- Least-Square Methods
- iLSTD (Algorithm & Properties)
- Results
- Discussion

Discussion

Discussion

- Which algorithm to pick?

Discussion

- Which algorithm to pick?
  - Data is extremely expensive, Task does not demand fast reaction $\Rightarrow$ LSTD

Discussion

- Which algorithm to pick?
  - Data is extremely expensive, Task does not demand fast reaction $\Rightarrow$ LSTD
  - Data is cheap but we need fast reaction with environment $\Rightarrow$ TD

Discussion

- Which algorithm to pick?
  - Data is extremely expensive, Task does not demand fast reaction $\Rightarrow$ LSTD
  - Data is cheap but we need fast reaction with environment $\Rightarrow$ TD
  - Between criteria $\Rightarrow$ iLSTD

Discussion

Discussion

- Important facts

Discussion

- Important facts
  - LSTD is still the optimum solution with respect to **all past experiences** and using TD methods.

Discussion

- Important facts
  - LSTD is still the optimum solution with respect to **all past experiences** and using TD methods.
  - TD is **faster** than iLSTD, and in case of having k features "on" in any moment, it is O(k) per time-step.

Discussion

- Important facts
  - LSTD is still the optimum solution with respect to **all past experiences** and using TD methods.
  - TD is **faster** than iLSTD, and in case of having k features "on" in any moment, it is O(k) per time-step.
  - iLSTD can be fit in many constraints **by adjusting *m* parameter**.

Discussion

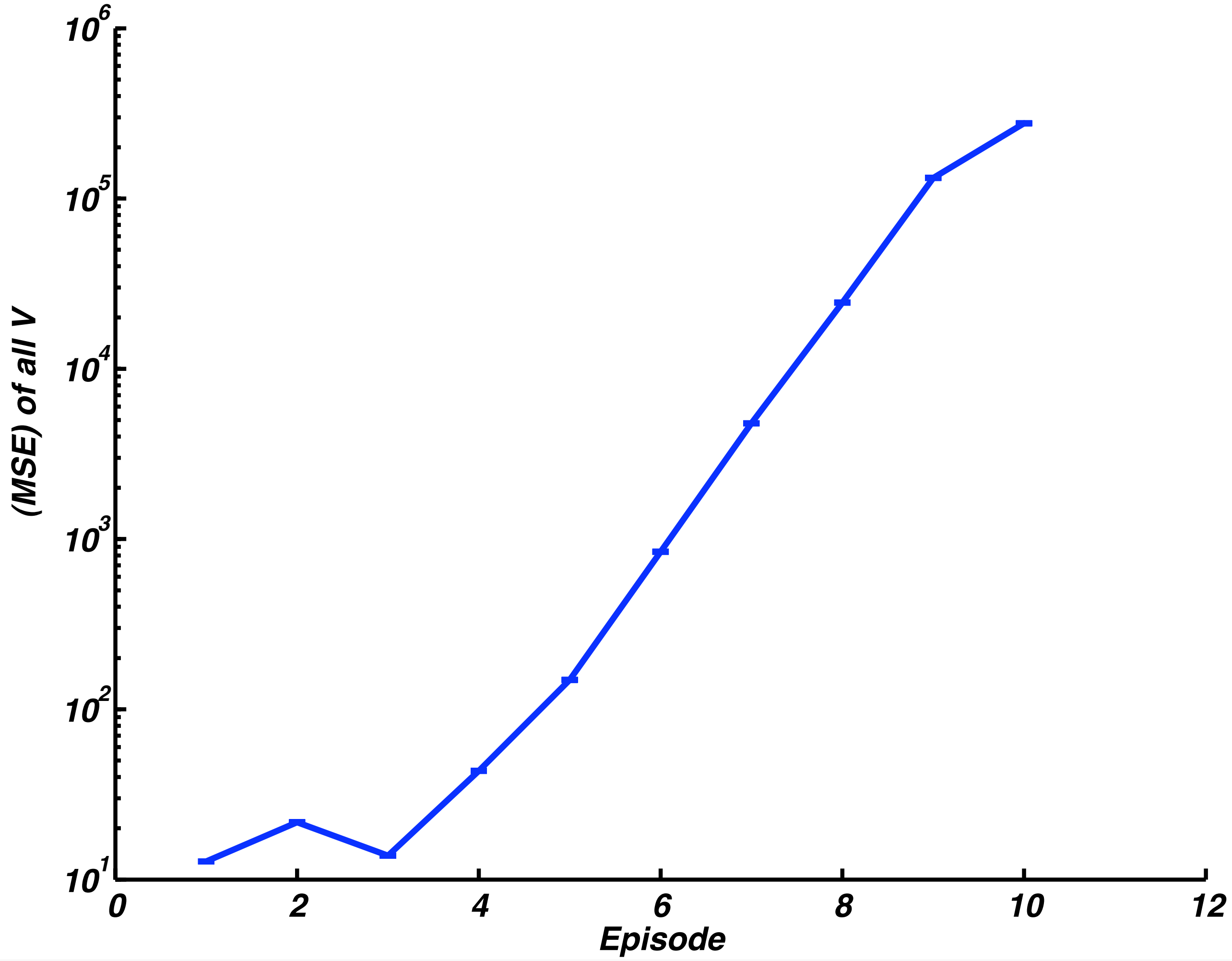
Discussion

- Can we use Coordinate Decent?
  - Equivalent to Gauss-Seidel method to solve a linear system of equations.
  - No step size parameter to tune!

Discussion

- Can we use Coordinate Decent?
  - Equivalent to Gauss-Seidel method to solve a linear system of equations.
  - No step size parameter to tune!

small problem



Discussion

- Can we use Coordinate Decent?
  - Equivalent to Gauss-Seidel method to solve a linear system of equations.
  - No step size parameter to tune!

"The Gauss-Seidel method is applicable to ***strictly diagonally dominant****, or* ***symmetric positive definite matrices****."*

Eric W. Weisstein et al. "Gauss-Seidel Method." From *MathWorld*--A Wolfram Web Resource.

Discussion

- Can we use Coordinate Decent?
  - Equivalent to Gauss-Seidel method to solve a linear system of equations.
  - No step size parameter to tune!

*"The Gauss-Seidel method is applicable to **strictly diagonally dominant**, or **symmetric positive definite matrices**."*

Eric W. Weisstein et al. "Gauss-Seidel Method." From *MathWorld*--A Wolfram Web Resource.

$$\sum_{i=1}^{t} \phi_t(\phi_t - \gamma\phi_{t+1})^T \theta$$

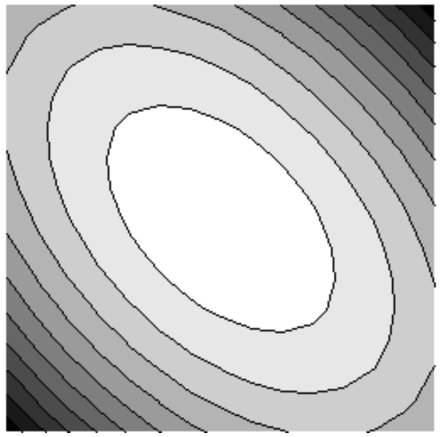
= μ is neither symmetric nor diagonally dominant.

Discussion

- We can make our matrix symmetric.

$$\mathbf{b}_t - \mathbf{A}_t \boldsymbol{\theta} = 0$$

$$\mathbf{A}_t^T \mathbf{b}_t - \mathbf{A}_t^T \mathbf{A}_t \boldsymbol{\theta} = 0$$



- **A** can be skewed, and this will make the convergence much slower.
- This is computationally more expensive. Choosing the best dimension would take $O(n^2)$

Discussion

$$\mathbf{A}_t^T \mathbf{b}_t - \mathbf{A}_t^T \mathbf{A}_t \boldsymbol{\theta} = 0$$

Discussion

$$\mathbf{A}_t^T \mathbf{b}_t - \mathbf{A}_t^T \mathbf{A}_t \boldsymbol{\theta} = 0$$

- Choosing the best dimension ✘

Discussion

$$\mathbf{A}_t^T \mathbf{b}_t - \mathbf{A}_t^T \mathbf{A}_t \boldsymbol{\theta} = 0$$

- Choosing the best dimension ✘
- Sweeping through dimensions ✓

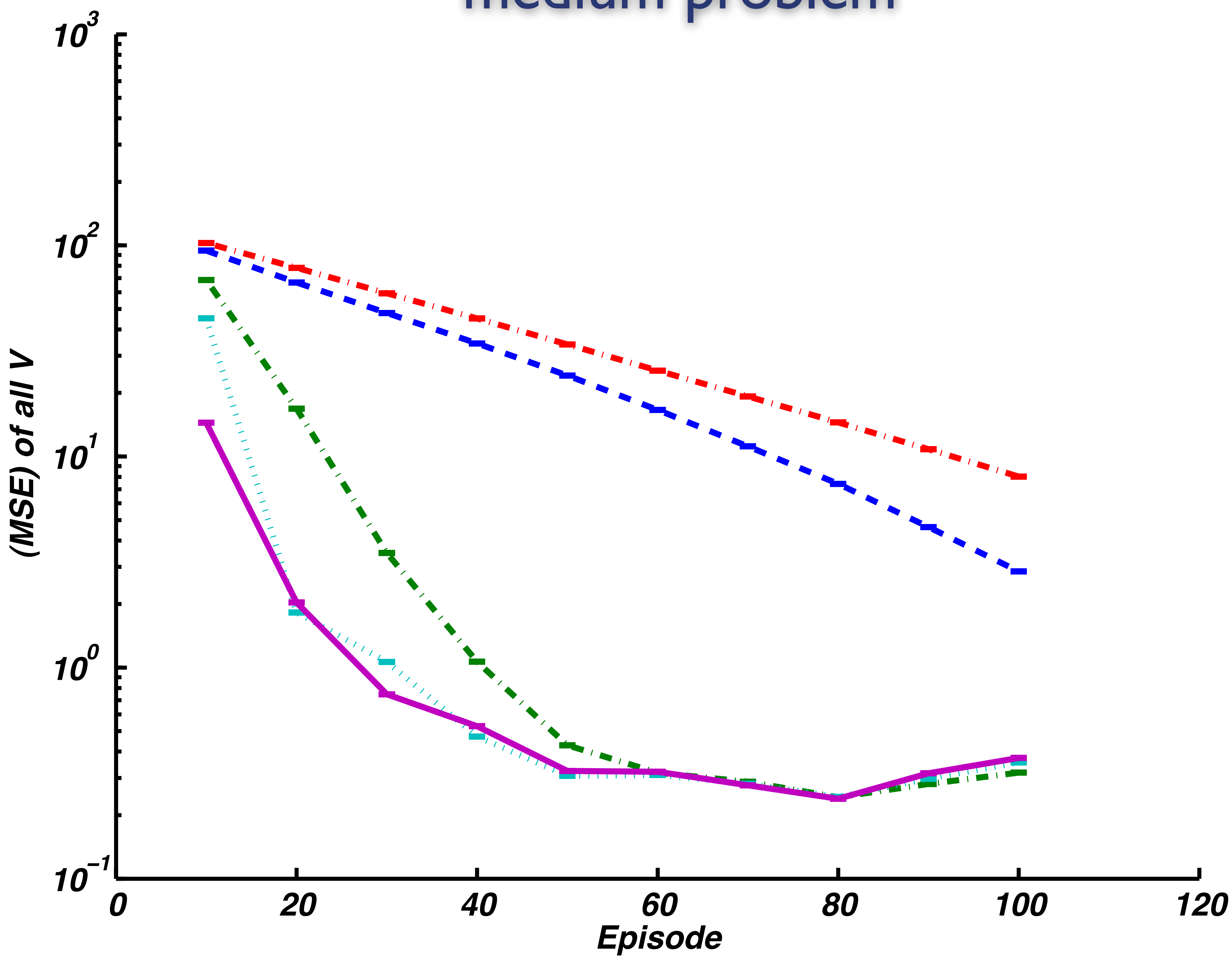
Discussion

$$\mathbf{A}_t^T \mathbf{b}_t - \mathbf{A}_t^T \mathbf{A}_t \boldsymbol{\theta} = 0$$

- Choosing the best dimension ✘
- Sweeping through dimensions ✓ ←
- Pick dimensions randomly ✓

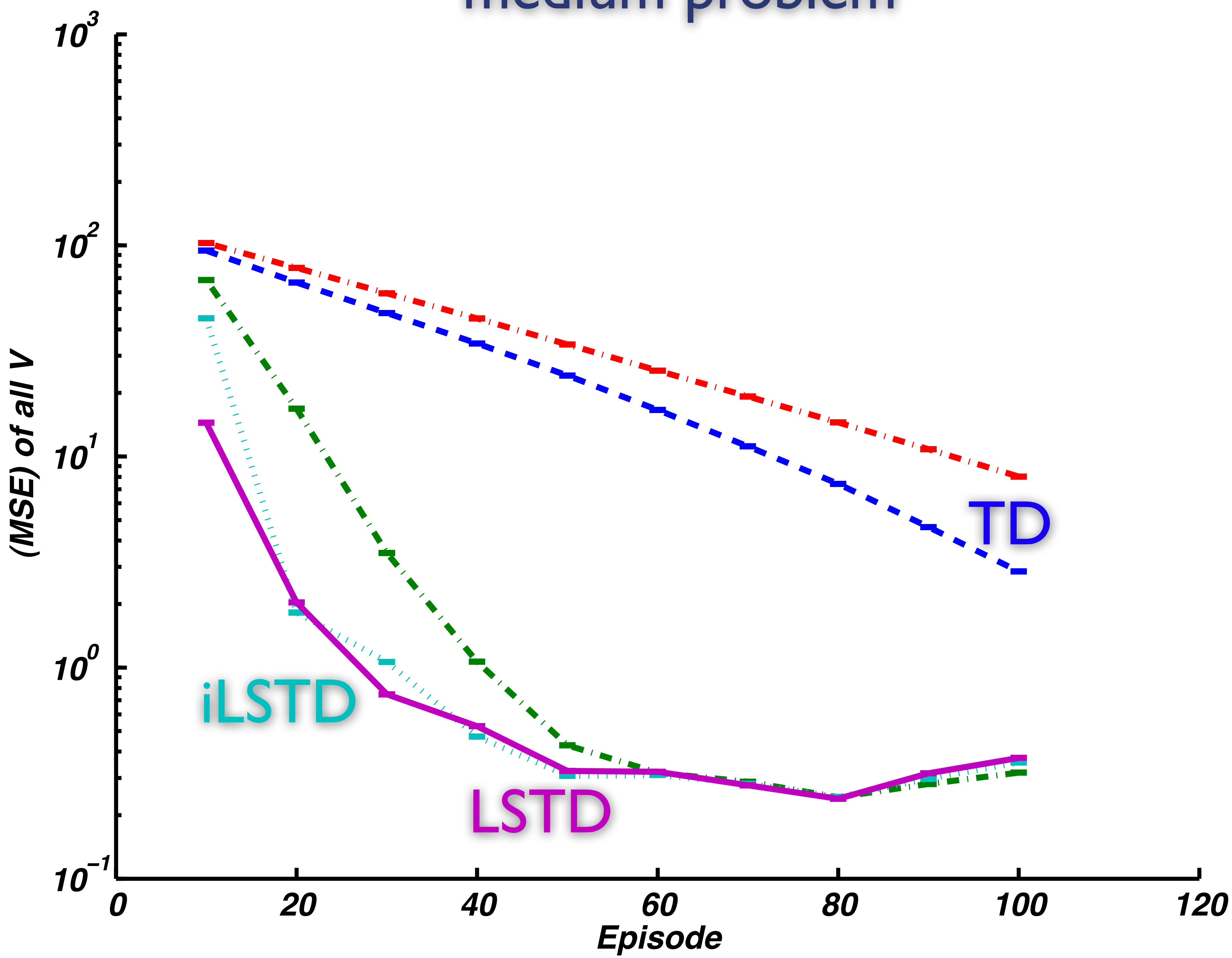
Discussion

medium problem



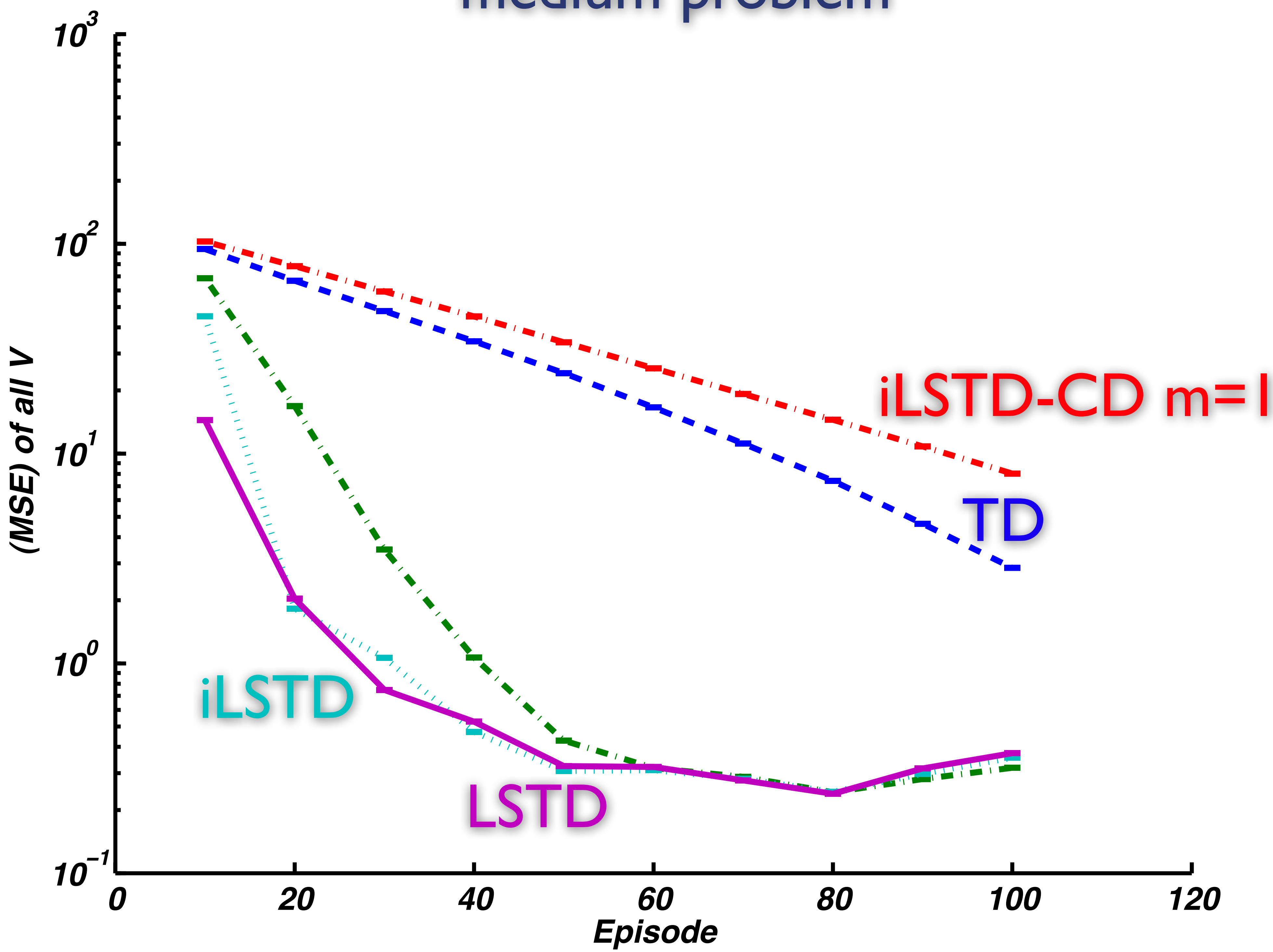
Discussion

medium problem



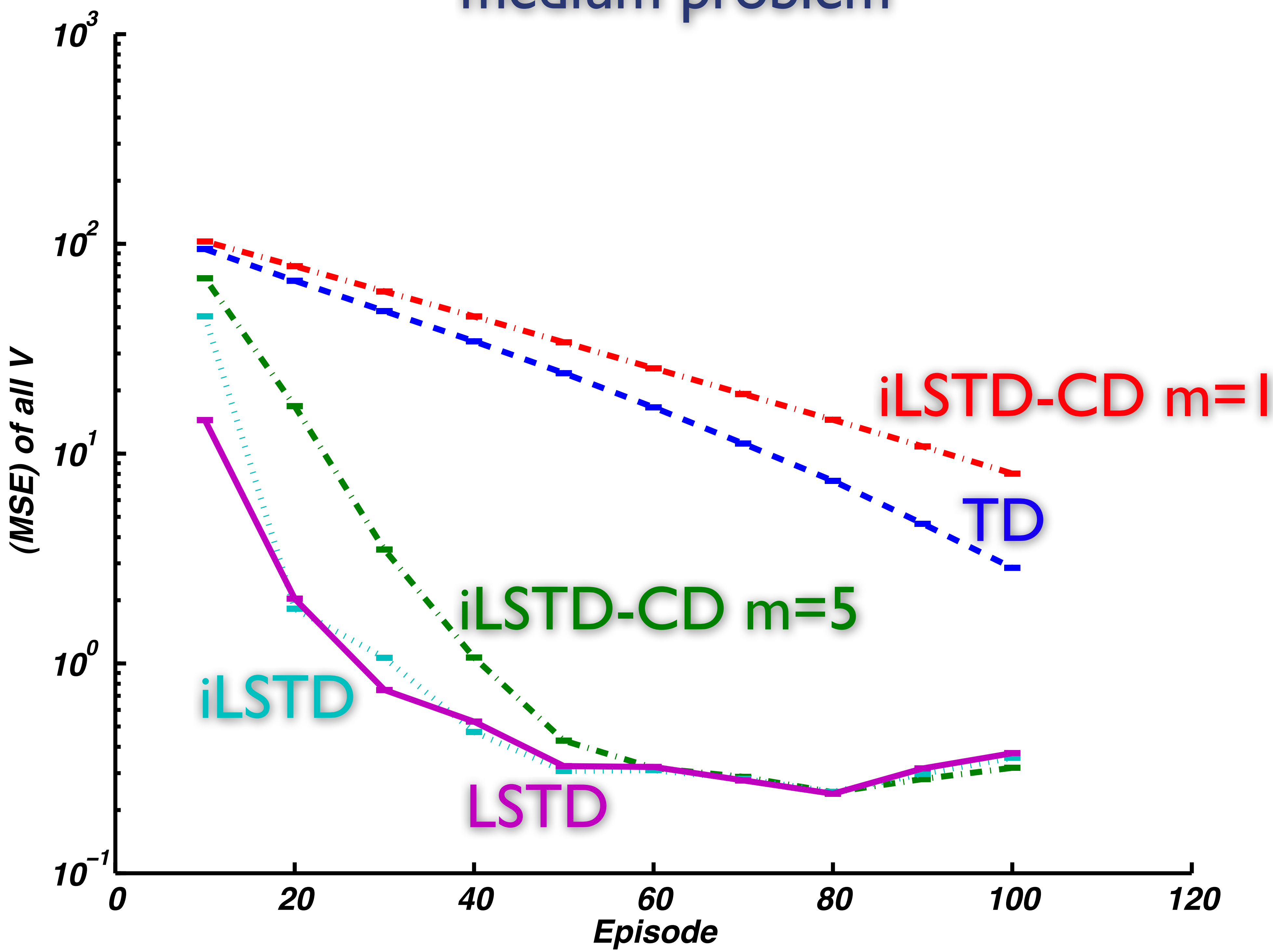
Discussion

medium problem

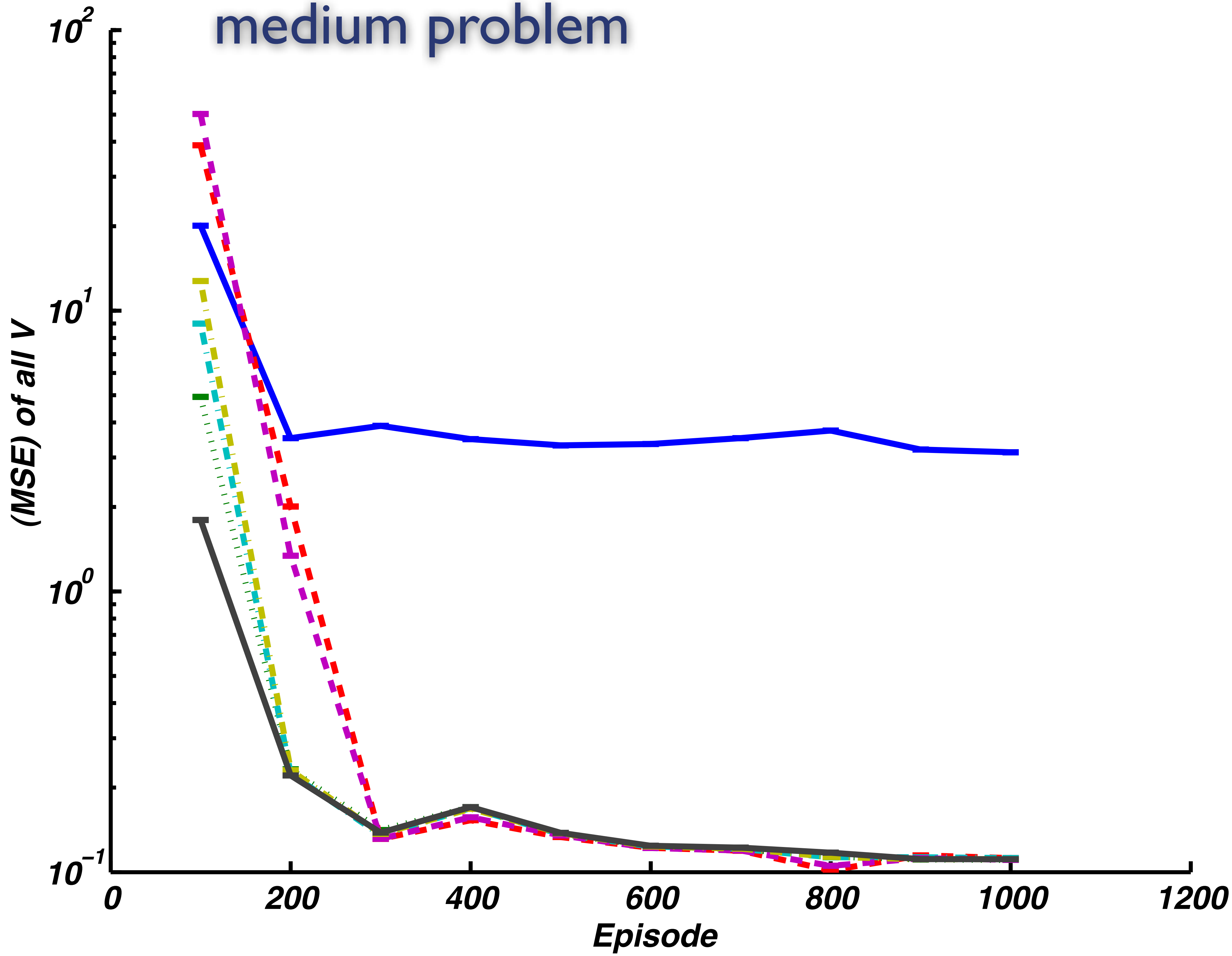


Discussion

medium problem

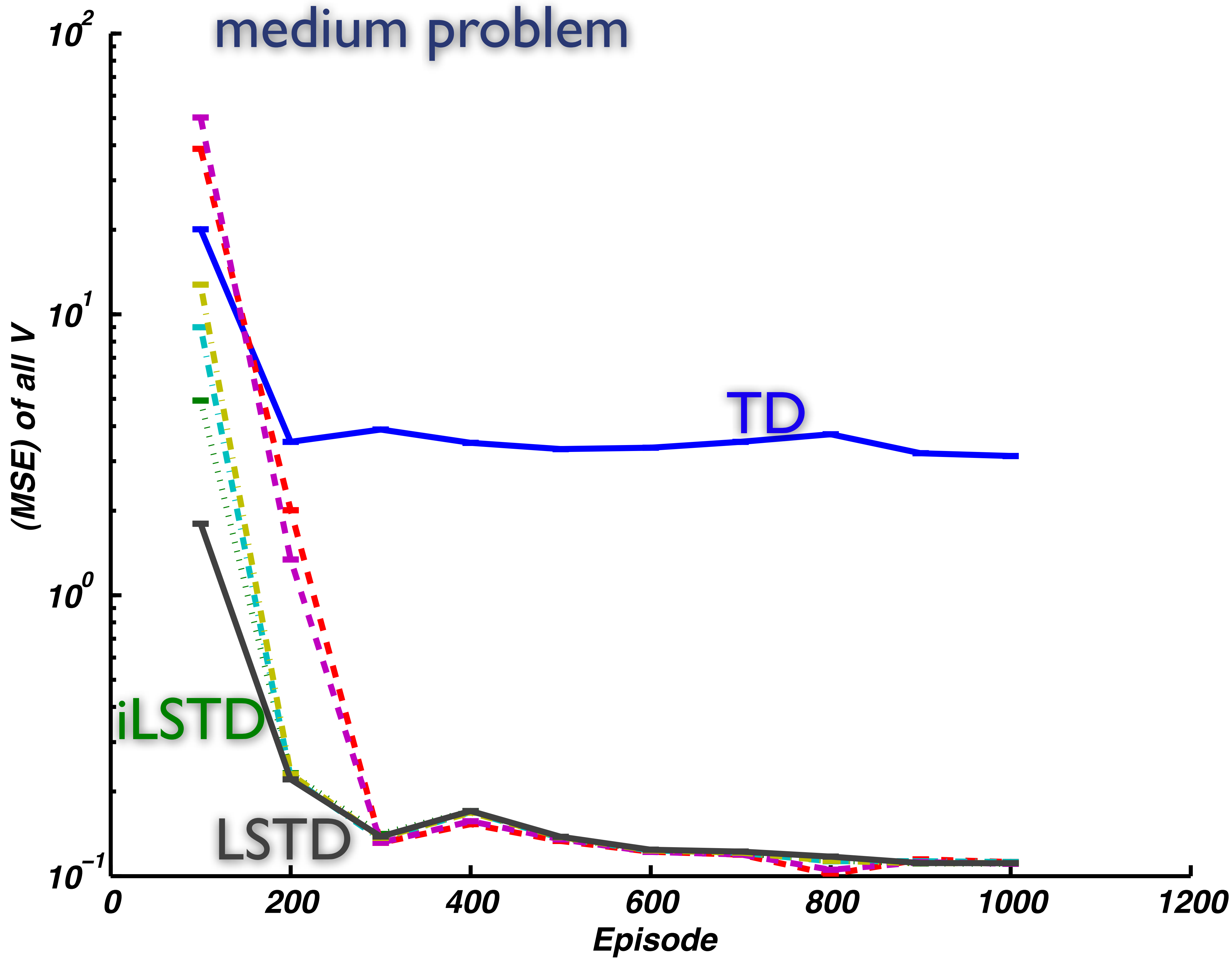


Discussion



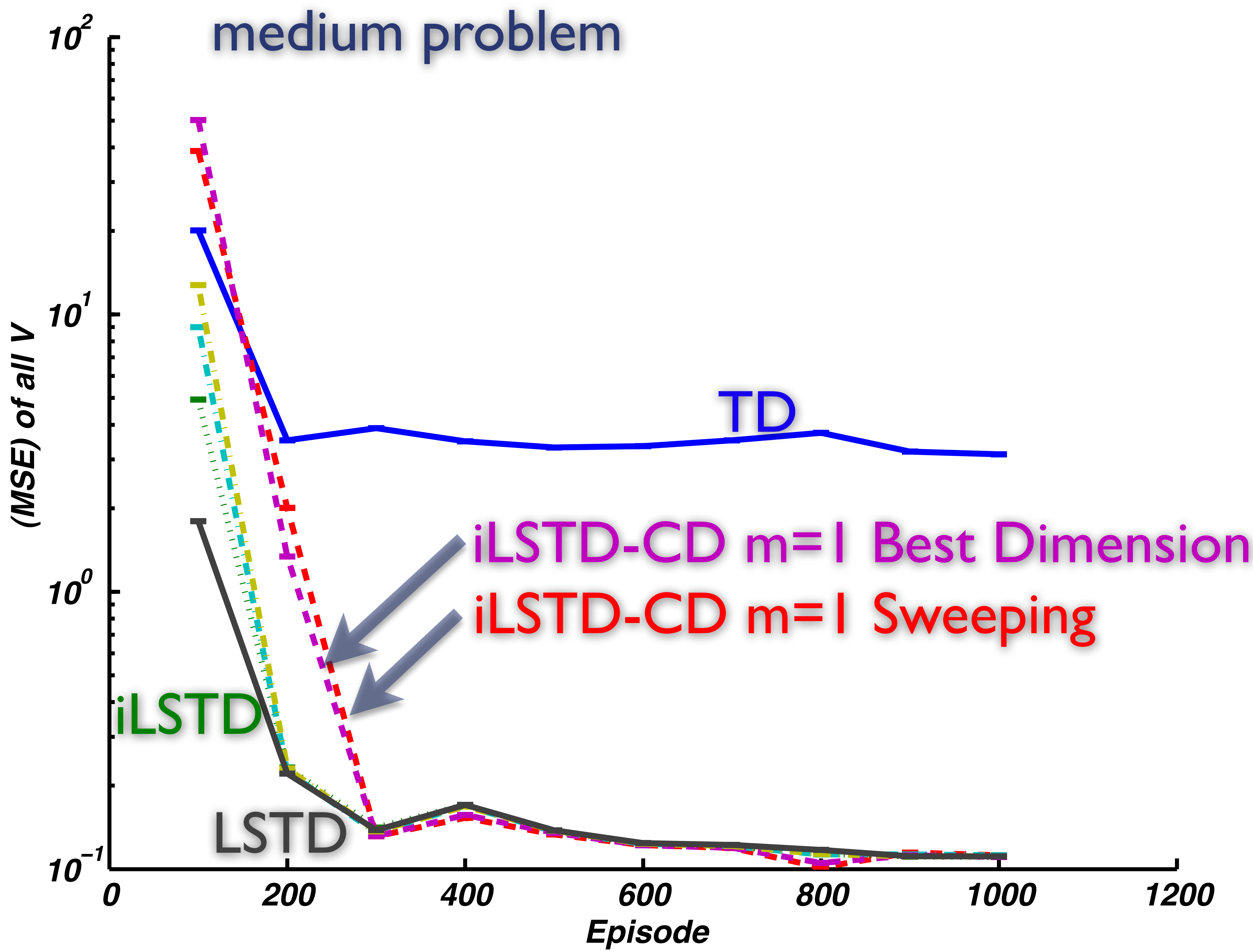
medium problem
10^2
10^1
10^0
10^{-1}
(MSE) of all V
0
200
400
600
800
1000
1200
Episode

Discussion



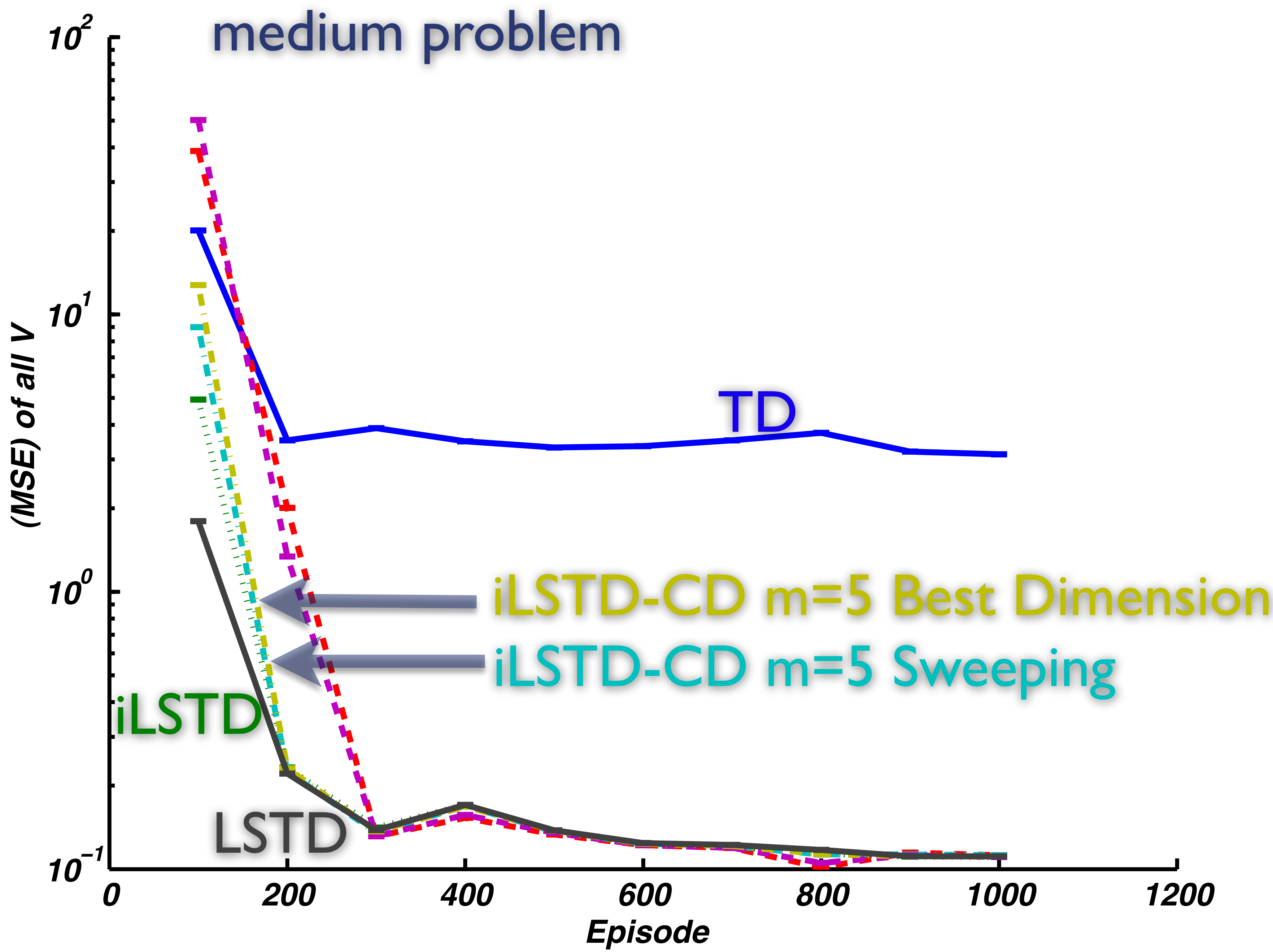
medium problem
10^2
10^1
10^0
10^{-1}
(MSE) of all V
TD
iLSTD
LSTD
0
200
400
600
800
1000
1200
Episode

Discussion



medium problem
10^2
10^1
10^0
10^{-1}
(MSE) of all V
0
200
400
600
800
1000
1200
Episode
TD
iLSTD-CD m=1 Best Dimension
iLSTD-CD m=1 Sweeping
iLSTD
LSTD

Discussion



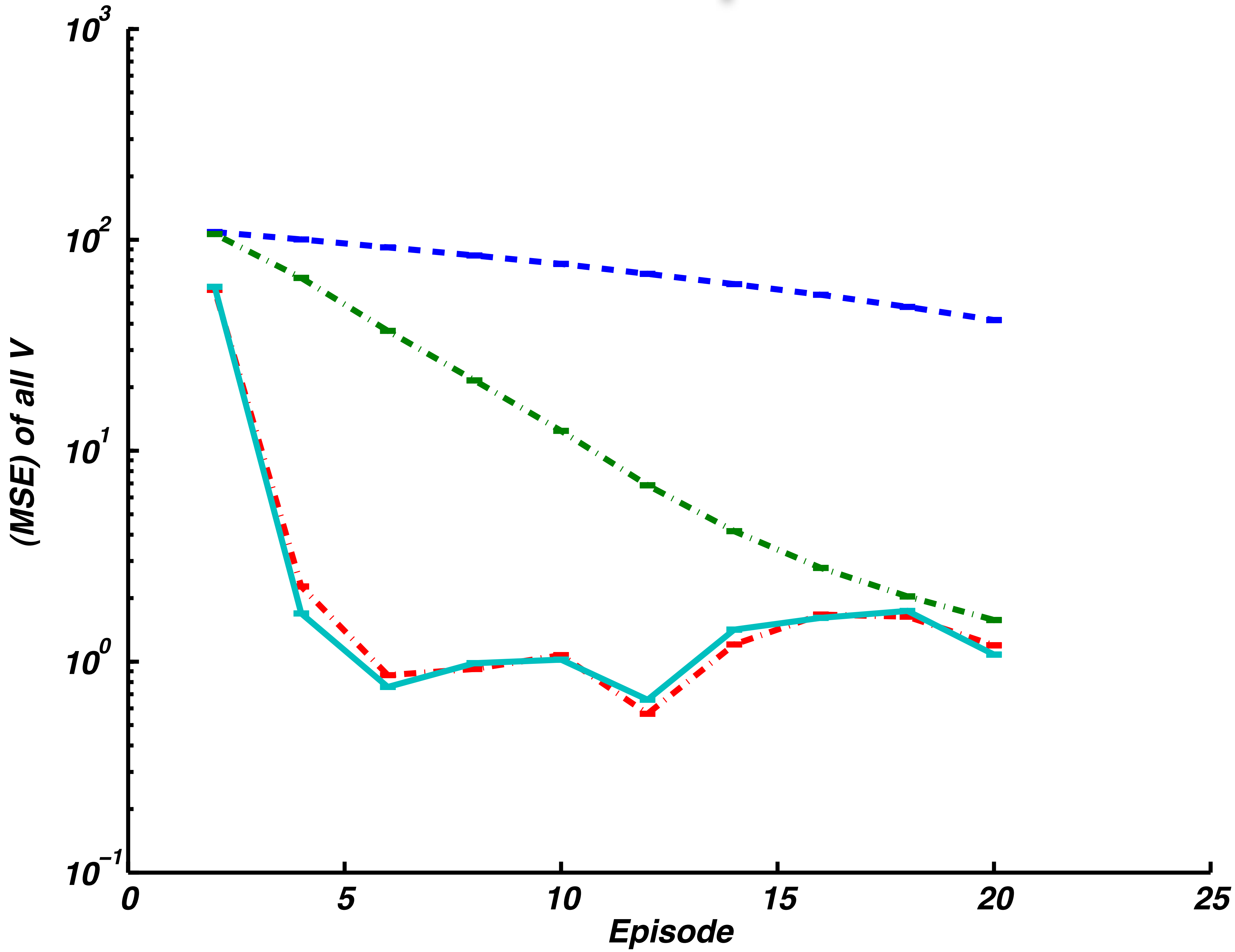
medium problem
10^2
10^1
10^0
10^{-1}
(MSE) of all V
0
200
400
600
800
1000
1200
Episode
TD
iLSTD-CD m=5 Best Dimension
iLSTD-CD m=5 Sweeping
iLSTD
LSTD

Discussion

- Generalized Minimal Residual method (GMRES) [Saad, Schultz 86]
  - Used for non symmetric matrices
  - Iterative
  - Results are interesting but the algorithm would be $O(n^2)$ per time step ...

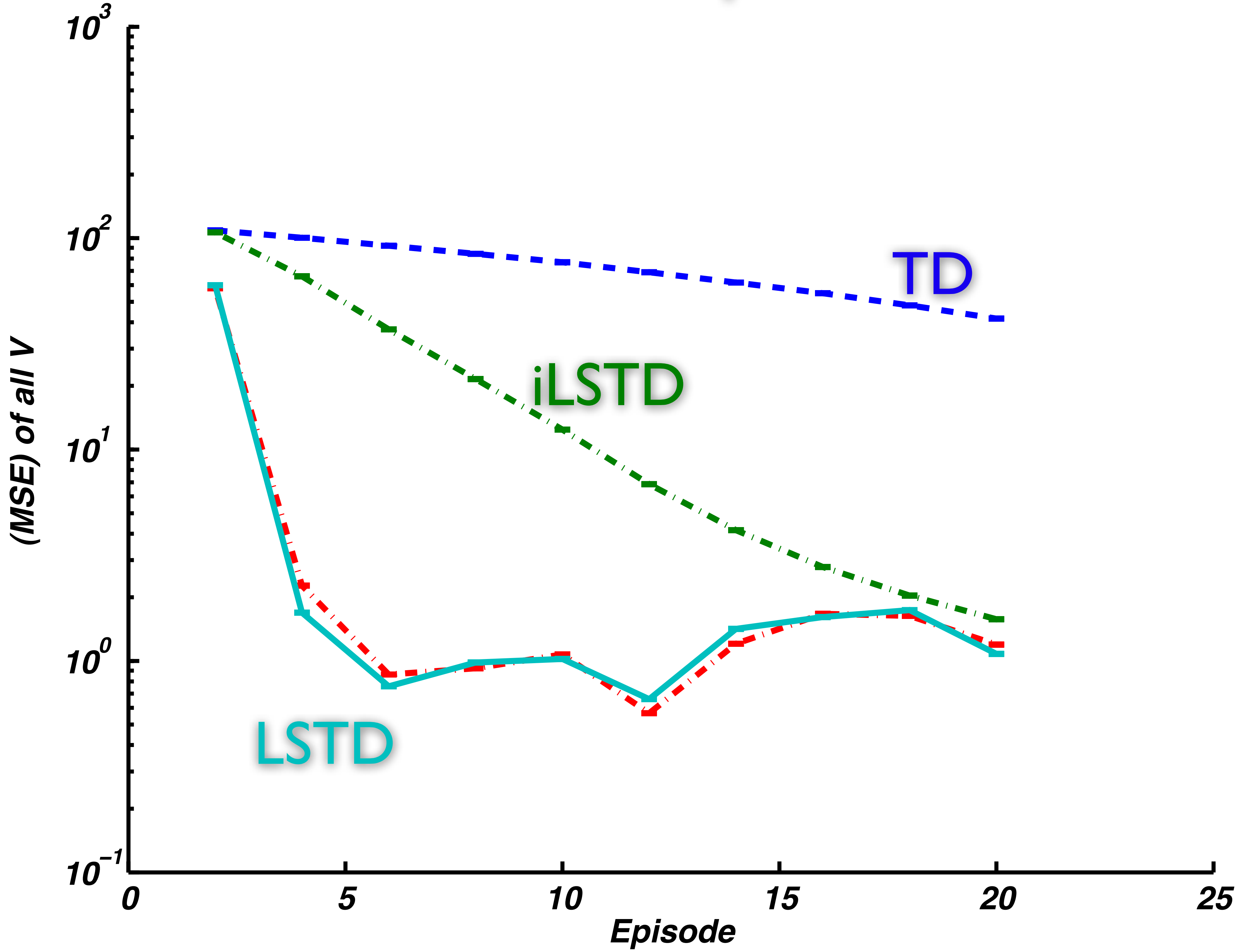
Discussion

medium problem



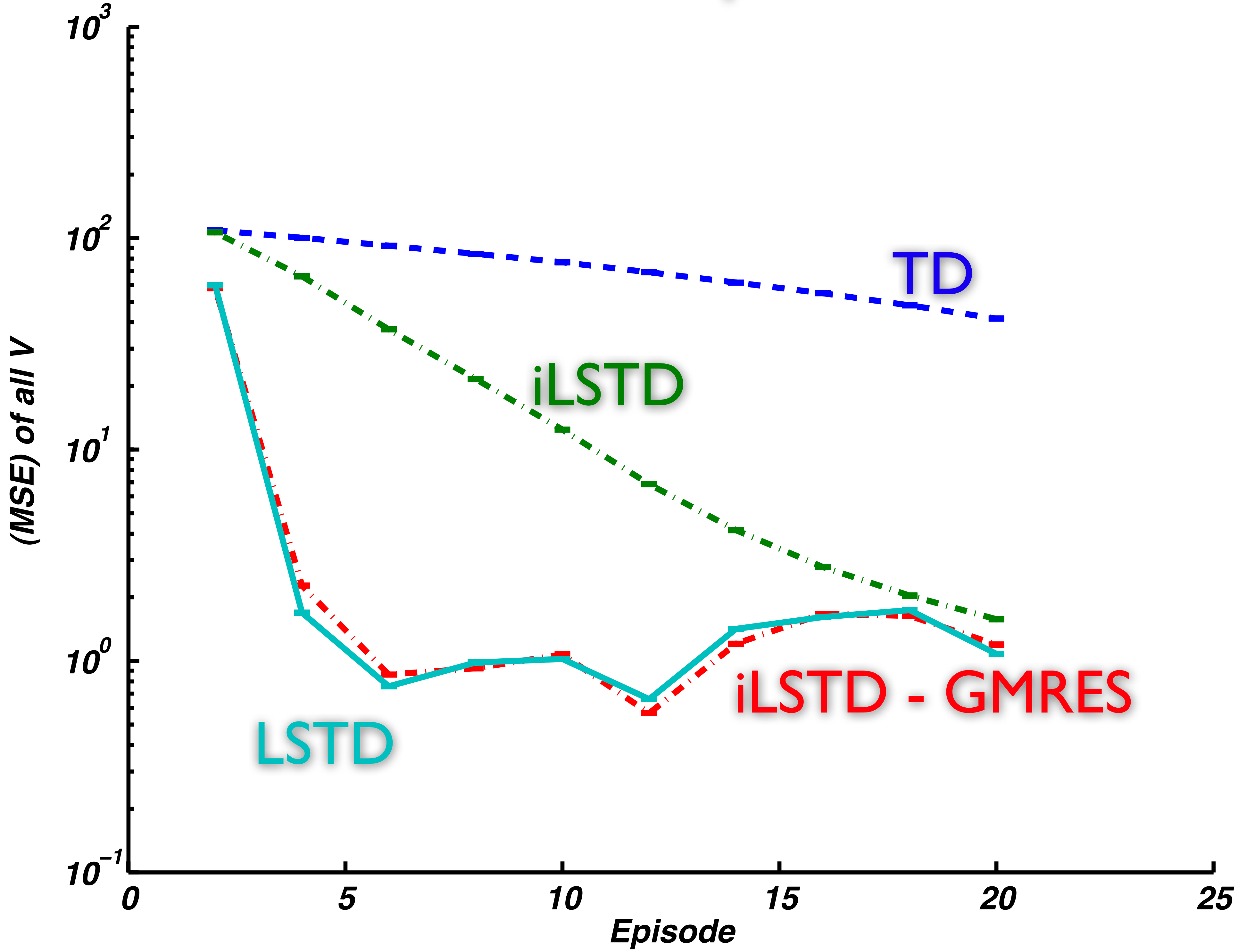
Discussion

medium problem



Discussion

medium problem



Discussion

Interesting Problems, Mountain Car, 10000 Memory, 10 Tilings

Discussion

Interesting Problems, Mountain Car, 10000 Memory, 10 Tilings

TD

iLSTD

LSTD

Discussion

Discussion

- Can iterative methods be superior than LSTD?
  - Many samples are needed to make **A** and **b** accurate when the environment is Stochastic
  - Small descents might be better than jumping to the solution of the estimated model ... (Future Work)

Discussion

- Can iterative methods be superior than LSTD?
  - Many samples are needed to make **A** and **b** accurate when the environment is Stochastic
  - Small descents might be better than jumping to the solution of the estimated model ... (Future Work)
- iLSTD(λ) → $O((m+k)n)$, Still $O(n)$

Discussion

- Can iterative methods be superior than LSTD?
  - Many samples are needed to make **A** and **b** accurate when the environment is Stochastic
  - Small descents might be better than jumping to the solution of the estimated model ... (Future Work)
- iLSTD(λ) → O((m+k)n), Still O(n)
- Larger Problems

Discussion

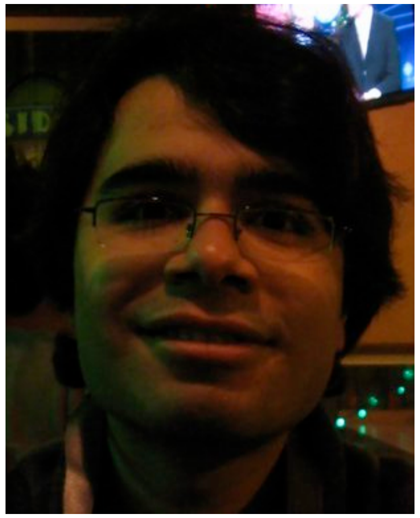
- Can iterative methods be superior than LSTD?
  - Many samples are needed to make **A** and **b** accurate when the environment is Stochastic
  - Small descents might be better than jumping to the solution of the estimated model ... (Future Work)
- iLSTD(λ) → $O((m+k)n)$, Still $O(n)$
- Larger Problems
- Proof of convergence

Acknowledgments

Dale Schuurmans

Dan Lizotte

Amirmasoud Farahmand



Mark Ring

References

- [Sutton, Barto 98]
  Sutton, R. S., and Barto, A. G. 1998. Reinforcement learning: An introduction.
- [Bradtke, Barto 96]
  Bradtke, S., and Barto, A. 1996. Linear least-squares algorithms for temporal difference learning. In *Machine Learning,*, volume 22, 33–57.
- [Boyan 99]
  Boyan, J. A. 1999. Least-squares temporal difference learning. In *Proc. 16th International Conf. on Machine Learning*, 49–56. Morgan Kaufmann, San Francisco, CA.

References

- [Saad, Schultz 86]
  Saad, Y. and Schultz, M. "GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems." *SIAM J. Sci. Statist. Comput.* **7**, 856-869, 1986.
- [Geramifard, Bowling, Sutton 06]
  A. Geramifard, M. Bowling, R. S. Sutton, "Iterative Lease Square Temporal Difference Learning" submitted to American Association for Artificial Intelligence (AAAI) 2006

Thanks...

- Questions

Thanks...

- Questions