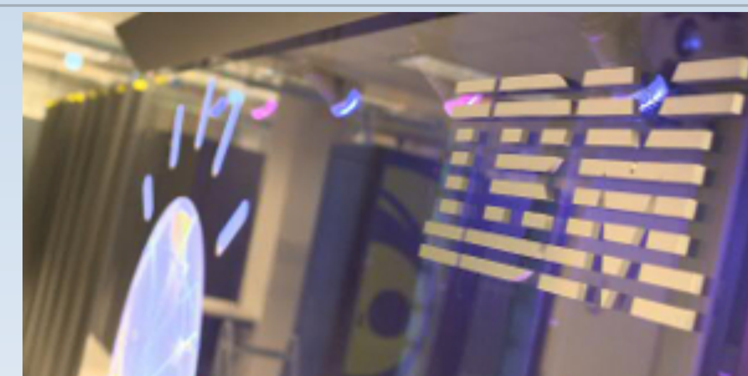
Proximal Reinforcement Learning: Learning to Act in Primal-Dual Spaces

Sridhar Mahadevan
Autonomous Learning Lab

UMASS CS 50 YEARS
College of Information and Computer Sciences



IBM Watson Research

National Science Foundation
WHERE DISCOVERIES BEGIN

RLDM 2015

Thanks to my collaborators

Bo Liu

Ian Gemp

Philip Thomas

Stephen Giguere

Nicholas Jacek

Will Dabney

UMass

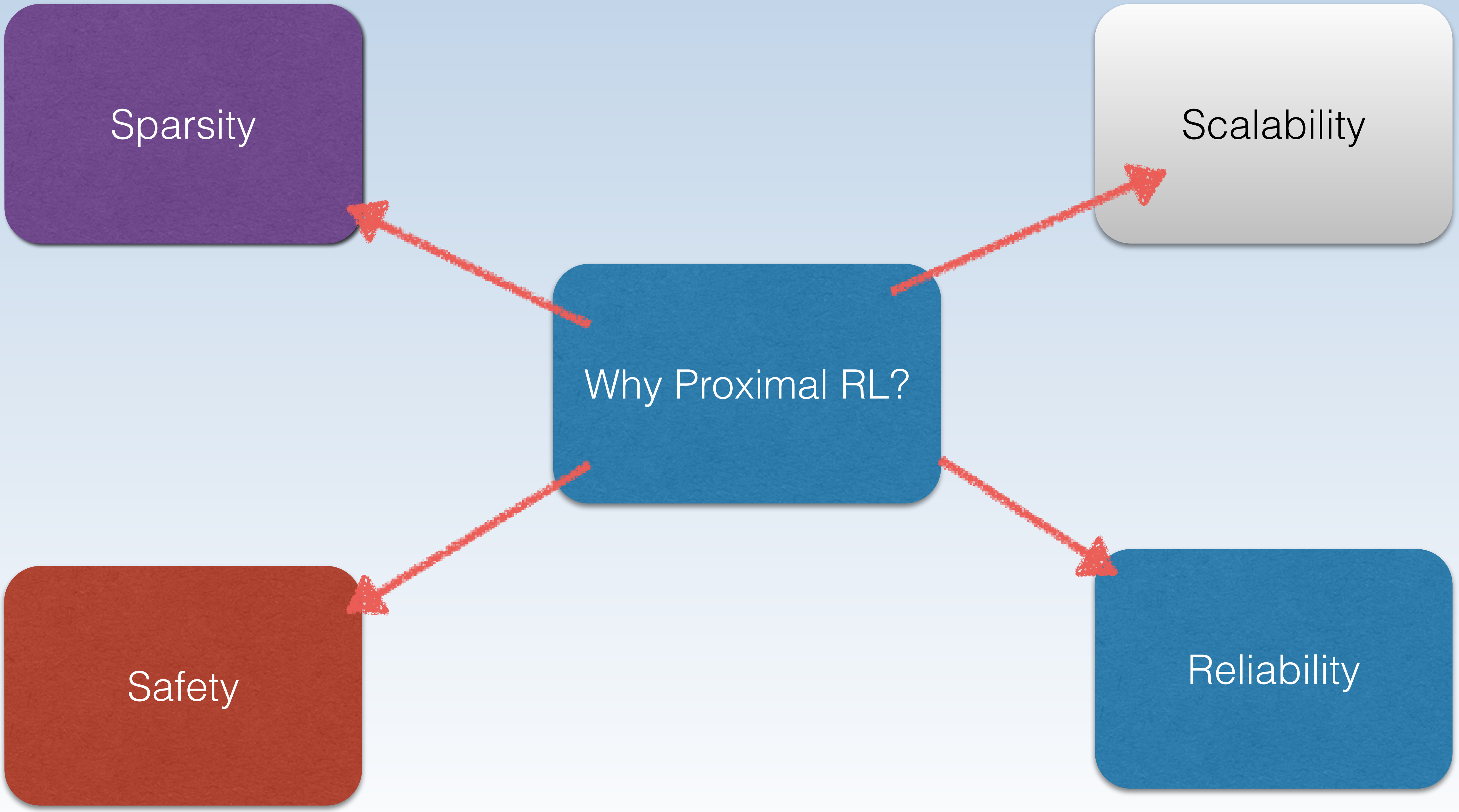
Mohammad Ghavamzadeh — Adobe

Ji Liu — Univ. of Rochester

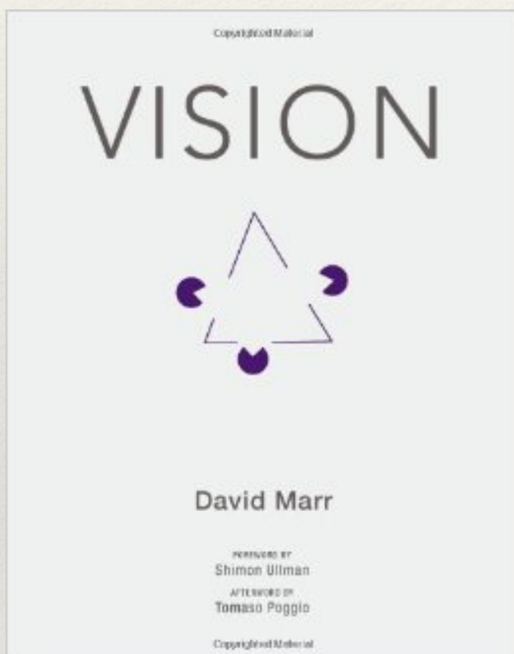
Marek Petrik — IBM

Proximal RL Framework

(Mahadevan et al., Arxiv 2014)



Three Level Analysis



Computational Theory

$$\min_\theta \max_y (L(\theta, y) = \langle b - A\theta, y \rangle - \frac{1}{2}\|y\|_M^2$$

Algorithmic Level

Algorithm 2 GTD2-MP

1: **for** $t = 1, \ldots, n$ **do**

2: Update parameters

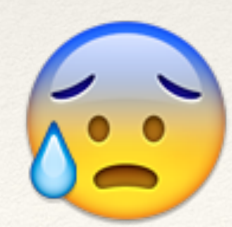
$$\delta_t = r_t - \theta_t^\top \Delta\phi_t$$
$$y_t^m = y_t + \alpha_t(\rho_t \delta_t - \phi_t^\top y_t)\phi_t$$
$$\theta_t^m = \theta_t + \alpha_t \rho_t \Delta\phi_t(\phi_t^\top y_t)$$
$$\delta_t^m = r_t - (\theta_t^m)^\top \Delta\phi_t$$
$$y_{t+1} = y_t + \alpha_t(\rho_t \delta_t^m - \phi_t^\top y_t^m)\phi_t$$
$$\theta_{t+1} = \theta_t + \alpha_t \rho_t \Delta\phi_t(\phi_t^\top y_t^m)$$

3: **end for**

4: OUTPUT

$$\bar{\theta}_n := \frac{\sum_{t=1}^n \alpha_t \theta_t}{\sum_{t=1}^n \alpha_t} \quad , \quad \bar{y}_n := \frac{\sum_{t=1}^n \alpha_t y_t}{\sum_{t=1}^n \alpha_t} \tag{34}$$

Neural Implementation



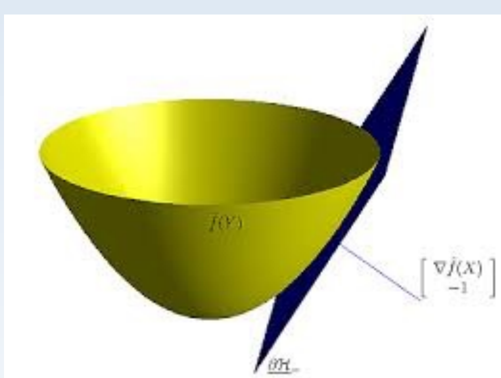
The Key Idea of Proximal RL

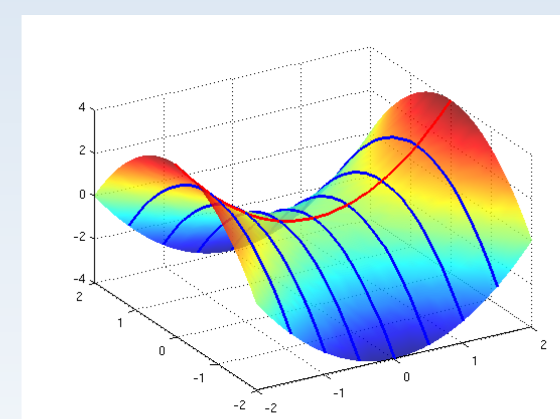
Primal Space
state,action,reward

$\theta \to \nabla \psi(\theta)$

$\lambda \to \nabla \psi^*(\lambda)$

Dual Space
operator splitting
gradient updates
Saddle point form





Recent publications

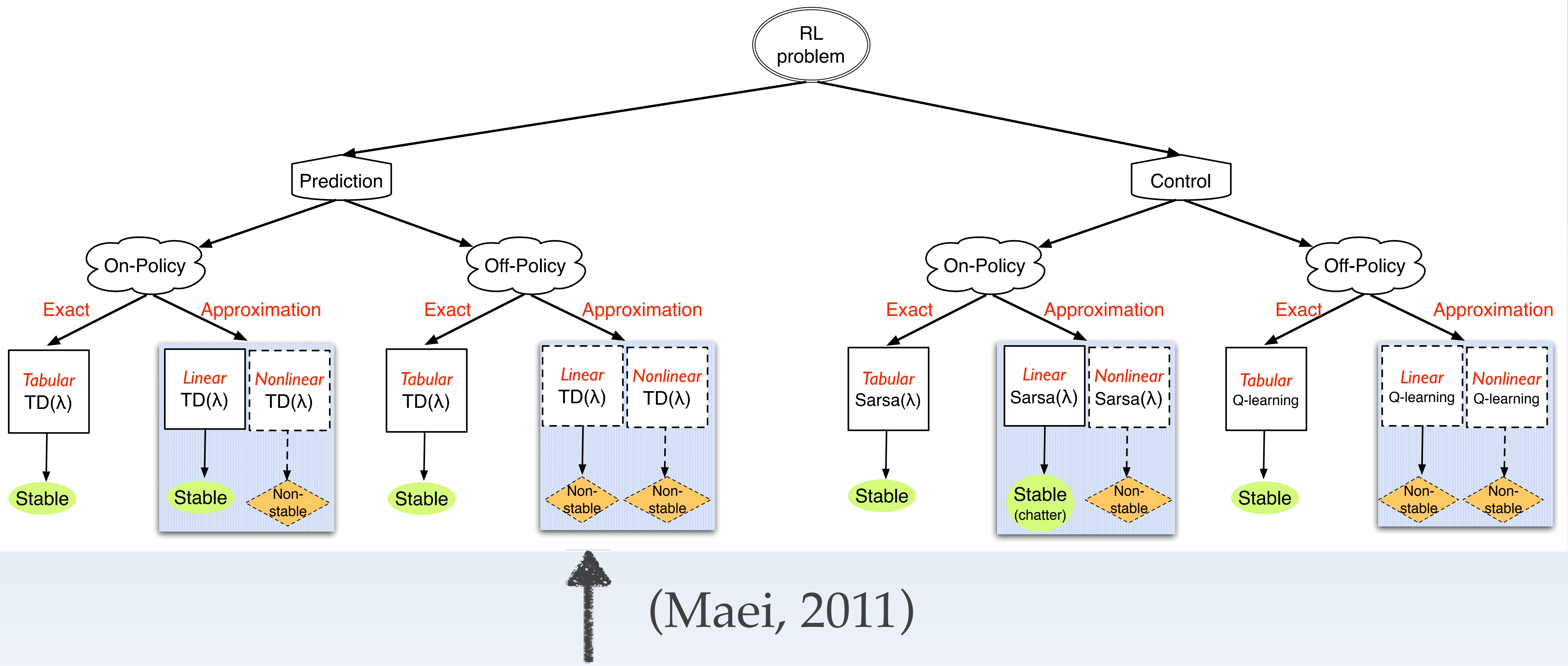
- ✓ UAI 2015: Finite sample analysis of proximal gradient TD algorithms
- ICML 2014: Generalization of natural gradient ascent
- NIPS 2013: Safe RL with Projected Natural Actor Critic
- AAAI 2013: Basis adaptation for sparse nonlinear RL
- NIPS 2012: Regularized off-policy TD-learning
- UAI 2012: Sparse Q-learning with mirror descent

Developing a True Stochastic Gradient TD Algorithm: The End of a 30 year Quest?

Bo Liu, Mohammad Ghavamzadeh,
Ji Liu, Marek Petrik, Sridhar Mahadevan

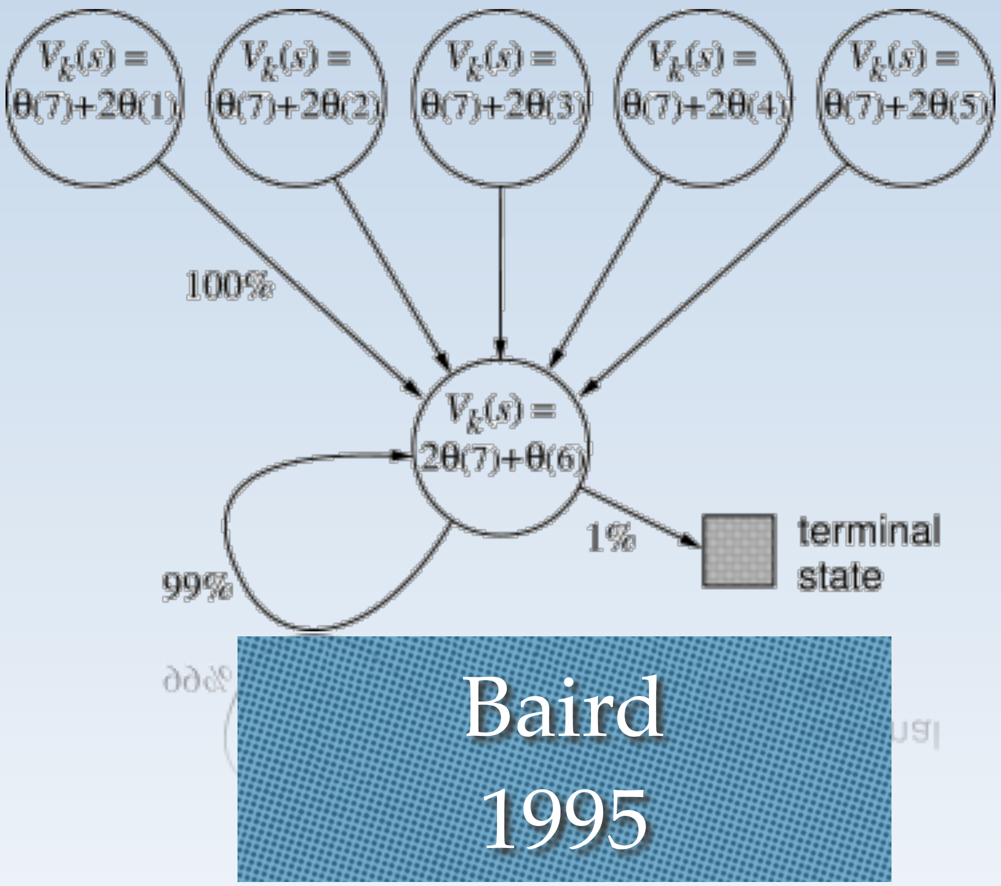
UAI 2015

Stability of RL Algorithms

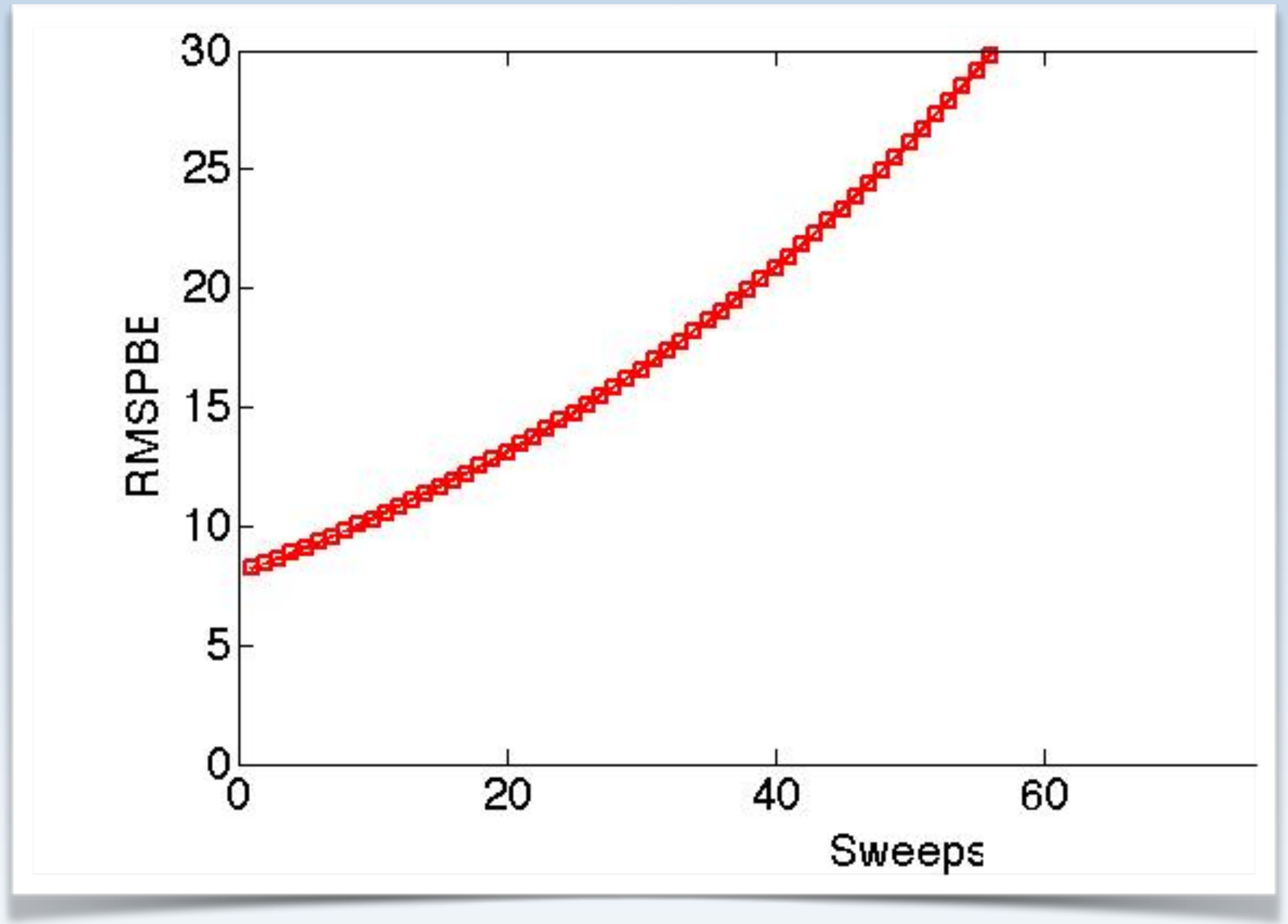


(Maei, 2011)

Instability of TD-Learning



TD diverges



Take the Blue Pill or the Red?

NEU
MSPBE
MSBE

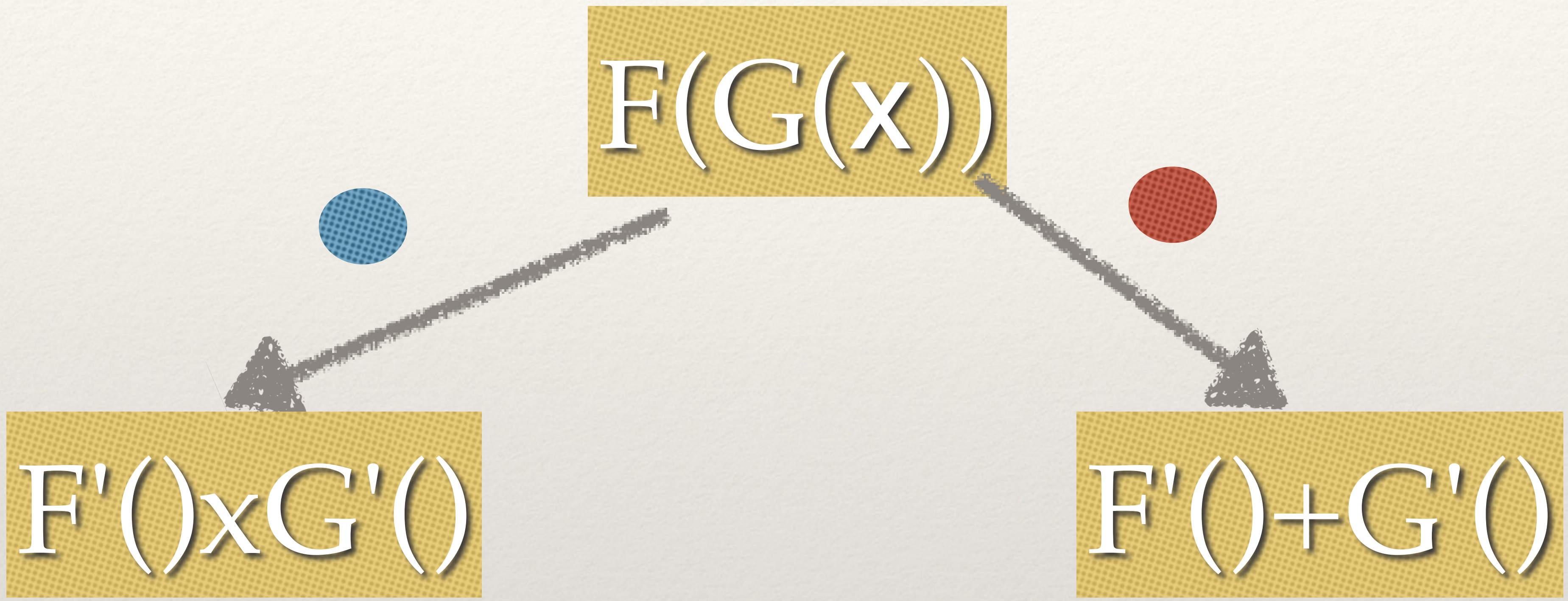
$$J(\theta) = \|b - A\theta\|^2_{M^{-1}}$$

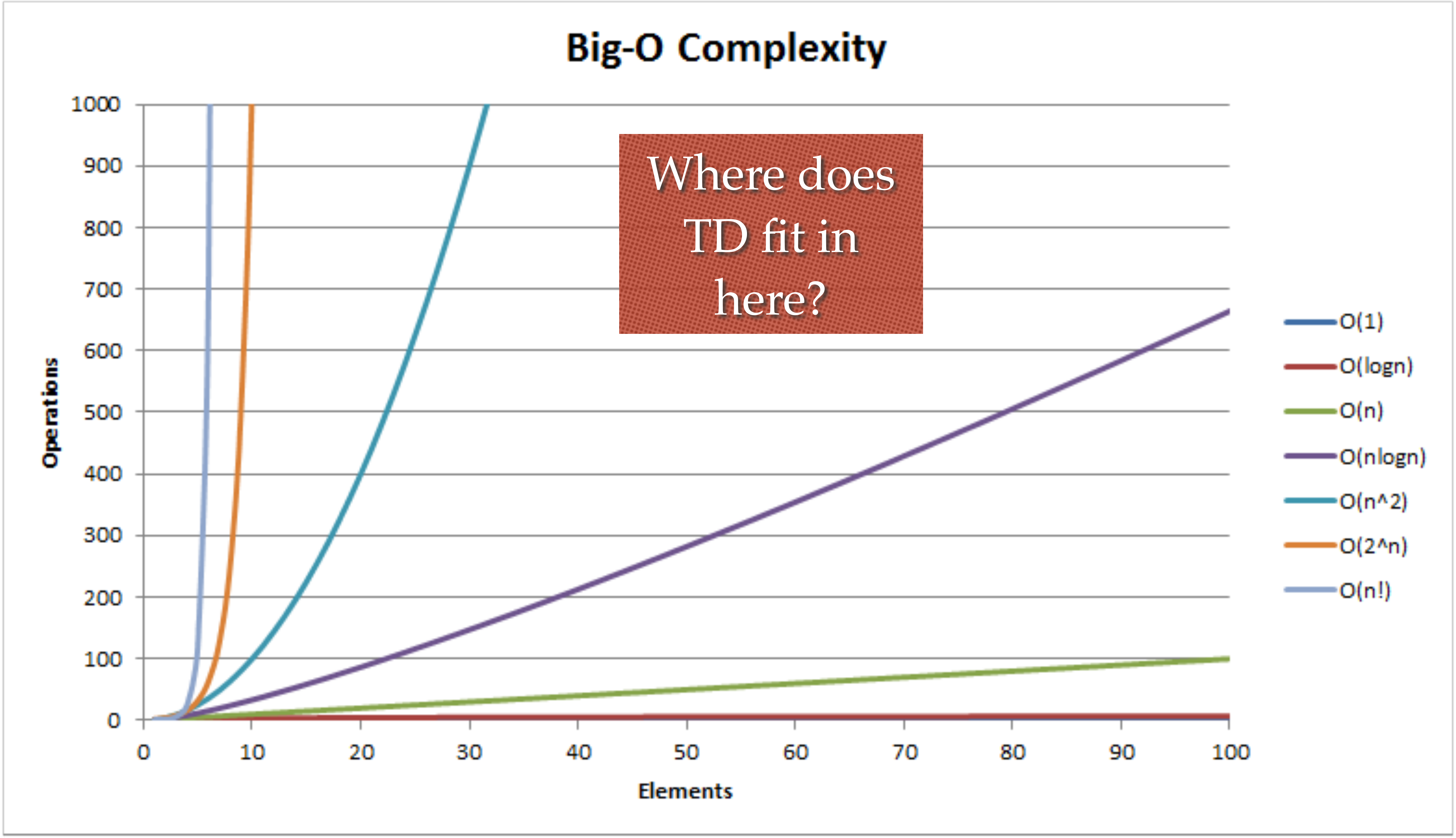
Baird,
Sutton et al.

Our
approach

$$\min_{\theta} \max_{y} (L(\theta, y) = \langle b - A\theta, y \rangle - \frac{1}{2}\|y\|^2_M$$

Operator Splitting





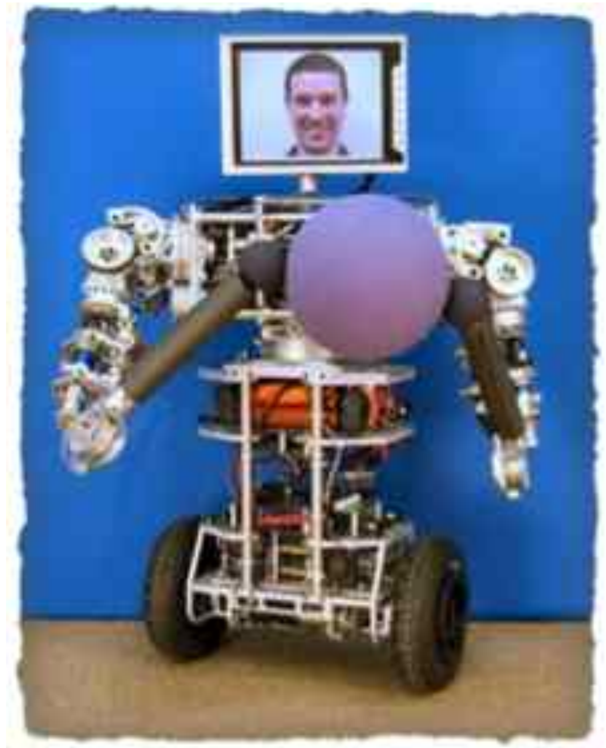
Big-O Complexity
1000
900
800
700
600
500
400
300
200
100
0
Operations
0
10
20
30
40
50
60
70
80
90
100
Elements
O(1)
O(logn)
O(n)
O(nlogn)
O(n^2)
O(2^n)
O(n!)
Where does TD fit in here?

Safe Reinforcement Learning with Projected Natural Actor Critic

Philip Thomas, Will Dabney, Sridhar Mahadevan, Steve Giguere

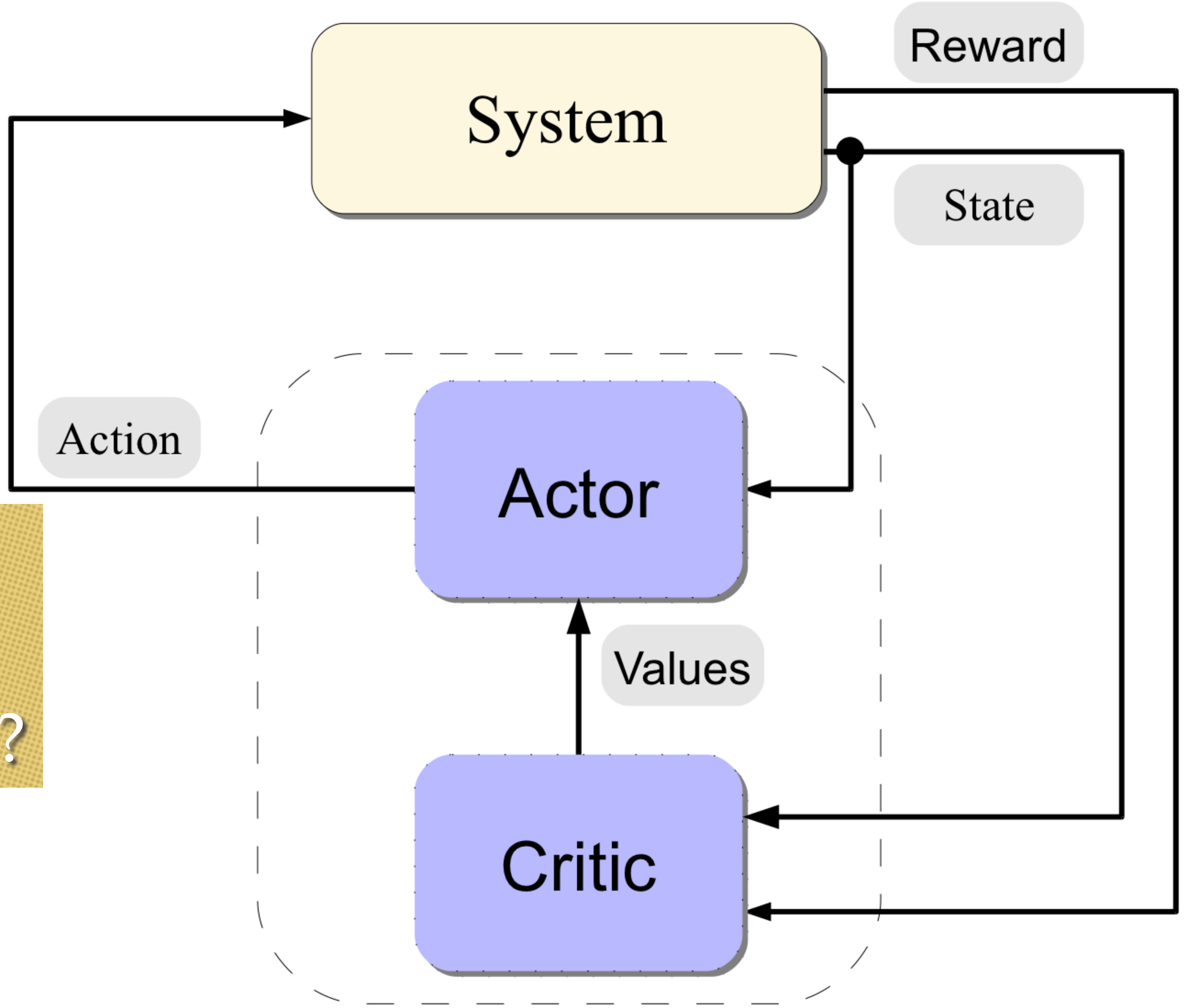
NIPS 2013

Natural Actor Critic



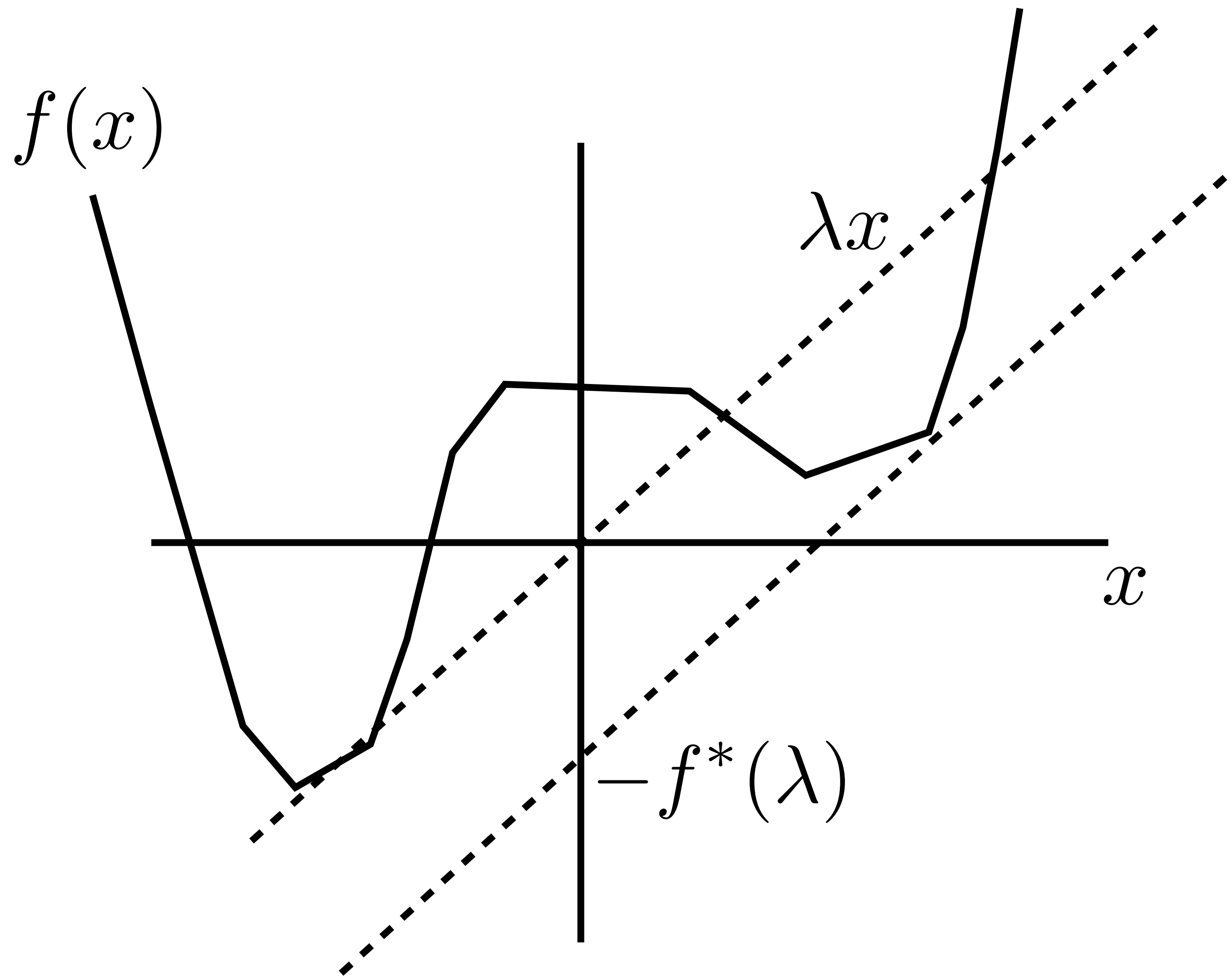
Actor update: $\omega_{t+1} = \omega_t + \alpha_t \delta_t \phi_t$

Critic update: $\theta_{t+1} = \theta_t + \beta_t \delta_t G_{t+1}^{-1} \psi(s_t, a_t)$



How to ensure safety in actor critic learning?

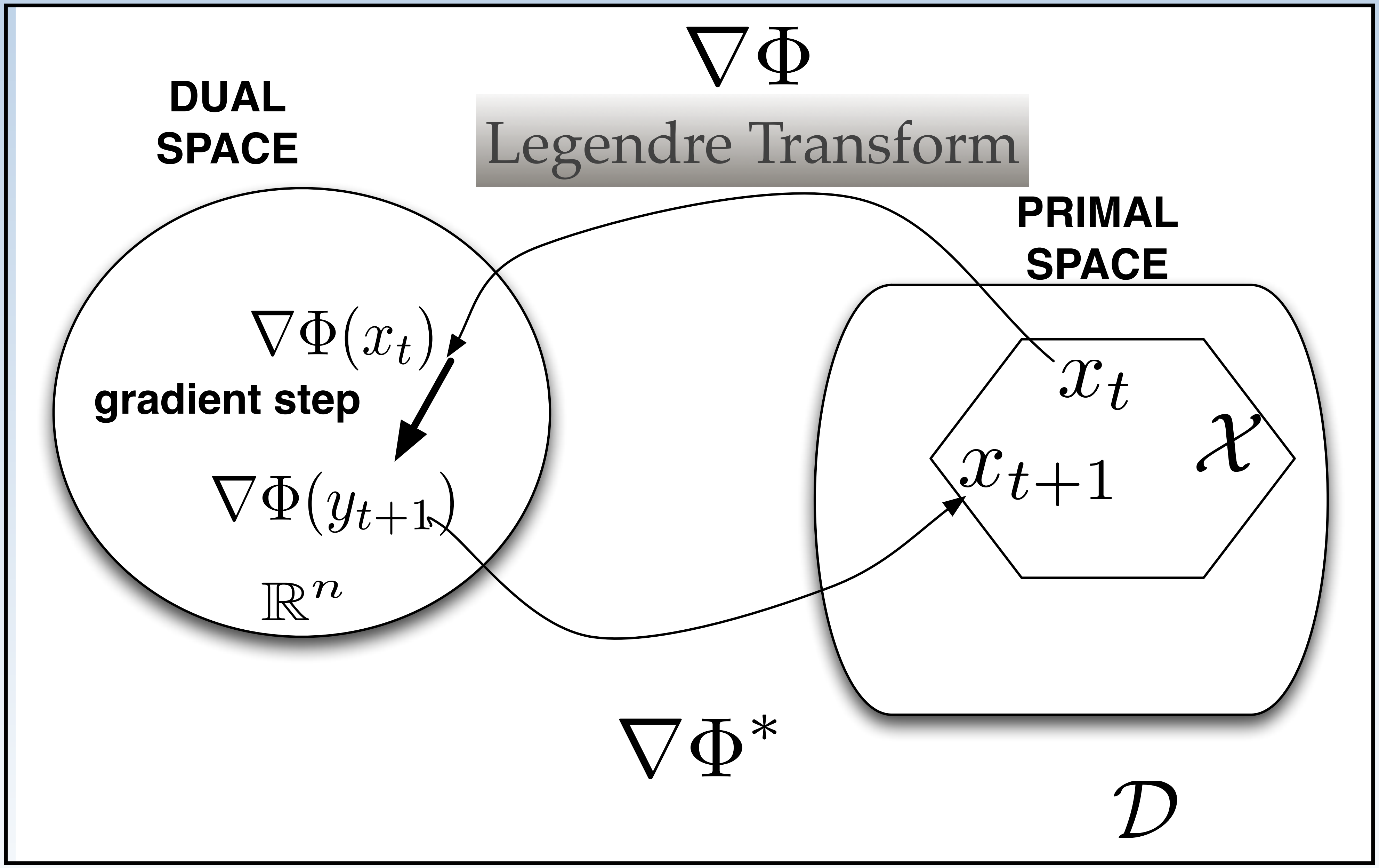
Conjugate Functions



$$f^*(\lambda) = \sup_x(\langle x, \lambda \rangle - f(x))$$

Mirror Maps

(Nemirovski and Yudin, 1980s)



Mirror Descent = Natural Gradient!

Thomas, Dabney, Mahadevan, Giguere, NIPS 2013

Natural gradient (Amari)

$$x_{k+1} = x_k - \alpha_k G_k^{-1} \nabla f(x_k)$$

Mirror Descent (Nemirovski and Yudin)

$$x_{k+1} = \nabla \psi_k^* \big(\nabla \psi_k(x_k) - \alpha_k \nabla f(x_k)\big)$$

We show these 30-year old techniques are closely related!

Proximal Mapping generalizes projections

- The proximal mapping of a convex function h is defined as

$$\mathrm{prox}_h(x) = \mathrm{argmin}_u \left(h(u) + \frac{1}{2}\|u - x\|_2^2 \right)$$

- Examples:

$$h(x) = 0, \mathrm{prox}_h(x) = x$$

$$h(x) = I_C(x), \mathrm{prox}_h(x) = P_C(x) = \mathrm{argmin}_{u \in C} \|u - x\|_2^2$$

Gradient Descent as proximal mapping

Answer:

$$w_{t+1} \leftarrow w_t - \alpha_t \nabla f(w_t)$$

Question?

$$w_{k+1} = \min_u (\langle \nabla f(w_k), u \rangle + \frac{1}{2\alpha} \| u - w_k \|^2)$$

Gradient Descent as proximal operator

Answer:

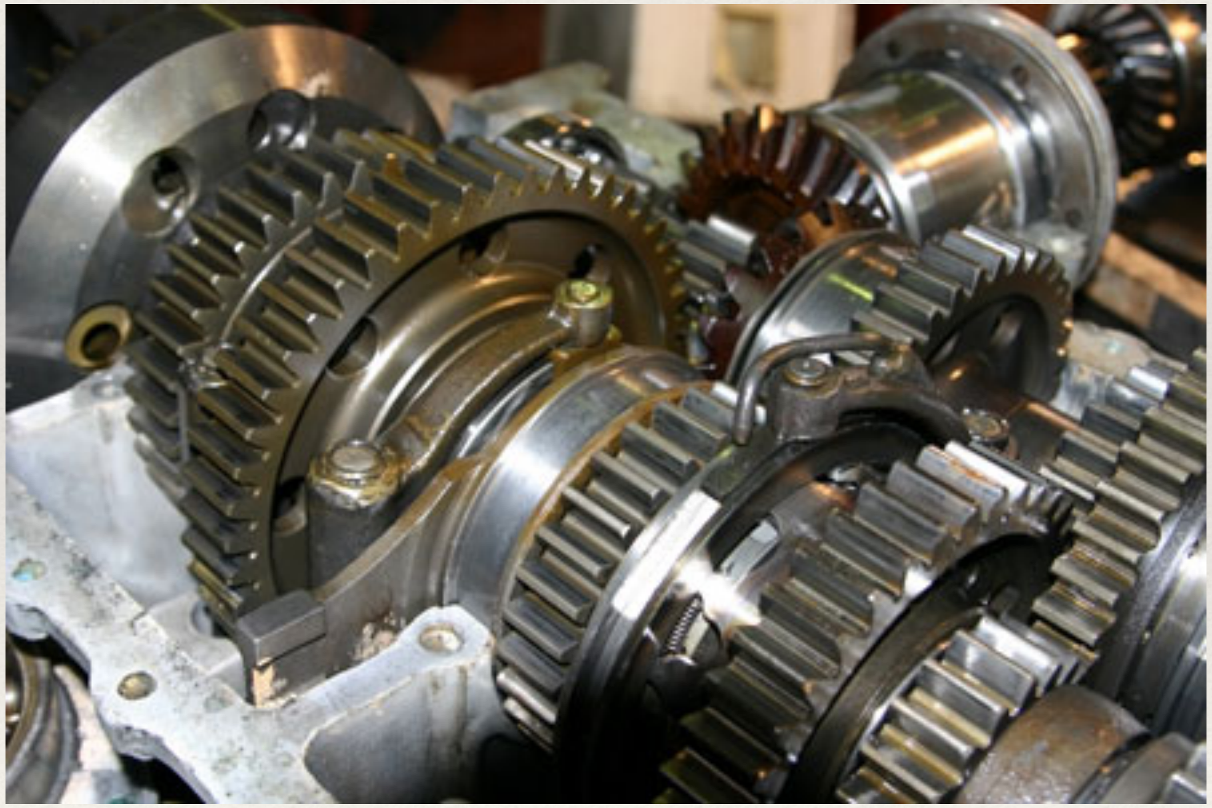
$$w_{k+1}^{j} = \frac{w_k^j \exp^{-\alpha_k \partial f_j(w_k)}}{\sum_{i=1}^{n} w_k^i \exp^{-\alpha_k \partial f_i(w_i)}}$$

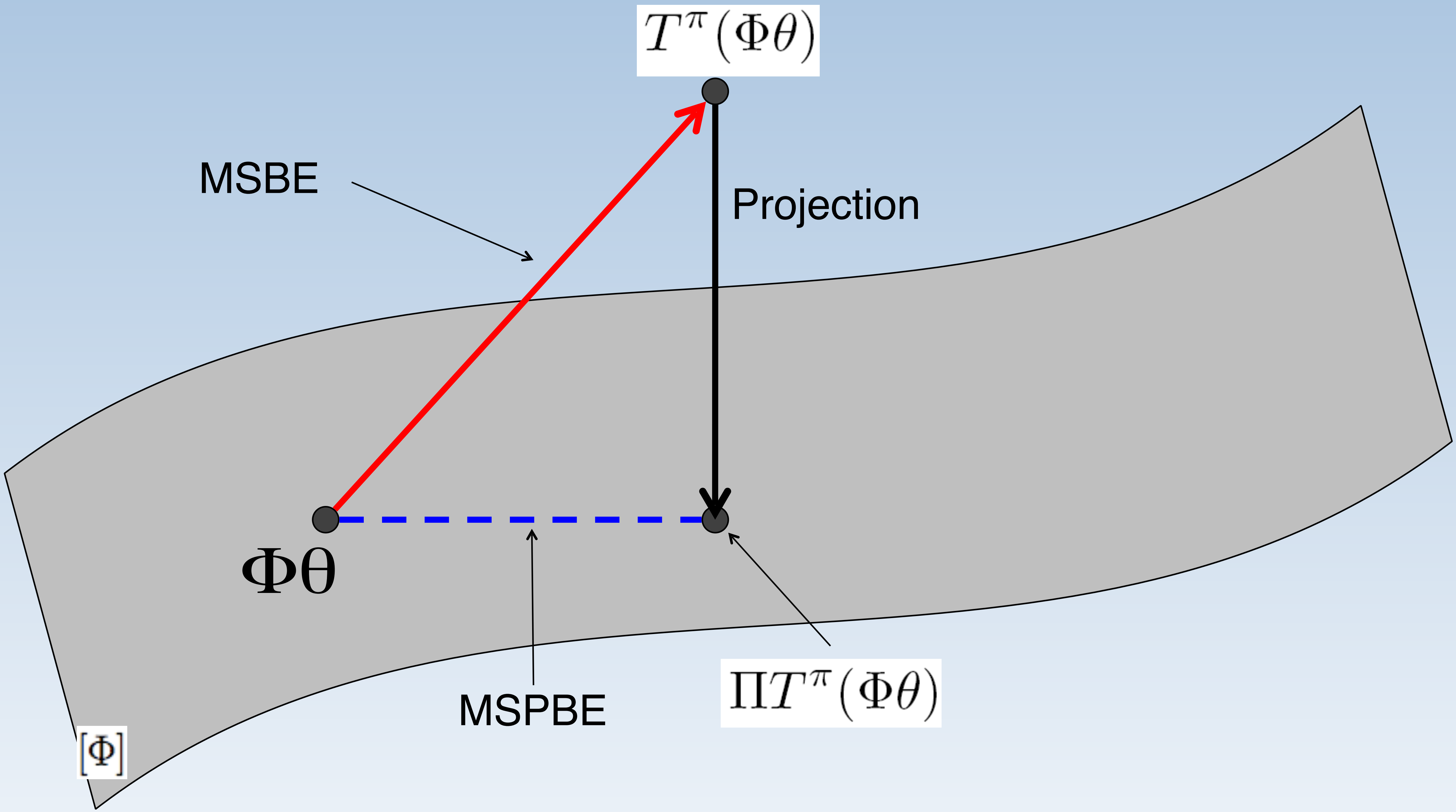
"Almost" dimension-free

Question?

$$w_{k+1} = \min_u (\langle \nabla f(w_k), u \rangle + \frac{1}{2\alpha} \mathrm{KL}(u, w_k))$$

Details of the framework





$$V^\pi = R^\pi + \gamma P^\pi V^\pi$$

Primal Approach to Gradient TD

Sutton, et al. 2009

$$
\begin{aligned}
&\mathrm{MSPBE}(\theta) \\
&= \| V_\theta - \Pi T V_\theta \|_D^2 \\
&= \| \Pi (V_\theta - T V_\theta) \|_D^2 \\
&= (\Pi (V_\theta - T V_\theta))^\top D (\Pi (V_\theta - T V_\theta)) \\
&= (V_\theta - T V_\theta)^\top \Pi^\top D \Pi (V_\theta - T V_\theta) \\
&= (V_\theta - T V_\theta)^\top D^\top \Phi (\Phi^\top D \Phi)^{-1} \Phi^\top D (V_\theta - T V_\theta) \\
&= (\Phi^\top D (T V_\theta - V_\theta))^\top (\Phi^\top D \Phi)^{-1} \Phi^\top D (T V_\theta - V_\theta) \\
&= \mathbb{E}[\delta \phi]^\top \, \mathbb{E}\left[\phi \phi^\top\right]^{-1} \mathbb{E}[\delta \phi] \, .
\end{aligned}
$$

Involves products of expectations

This cannot be easily sampled!

The key to our approach is to look at the dual problem, and use operator splitting (Mahadevan et al., Arxiv, 2014)

Linear System Reformulation of Gradient TD (Maei, 2011)

Lemma 1. *Let* $\mathcal{D} = \left\{\left(s_i, a_i, r_i, s_i'\right)\right\}_{i=1}^{n}$, $s_i \sim \xi$, $a_i \sim \pi_b(\cdot|s_i)$, $s_i' \sim P(\cdot|s_i, a_i)$ *be a training set generated by the behavior policy* π_b *and* T *be the Bellman operator of the target policy* π*. Then, we have*

$$\Phi^\top \Xi(T\hat{v} - \hat{v}) = \mathbb{E}\left[\rho_i \delta_i(\theta)\phi_i\right] = b - A\theta.$$

$$A := \mathbb{E}\left[\rho_i \phi_i (\Delta\phi_i)^\top\right], \quad b := \mathbb{E}\left[\rho_i \phi_i r_i\right], \quad C := \mathbb{E}[\phi_i \phi_i^\top]$$

weighting factor $\rho_i = \pi(a_i|s_i)/\pi_b(a_i|s_i)$

Unified Objective for Gradient TD

$$J(\theta) = ||\Phi^\top \Xi(T\hat{v} - \hat{v})||^2_{M^{-1}} = ||\mathbb{E}[\rho_i \delta_i(\theta)\phi_i]||^2_{M^{-1}}$$

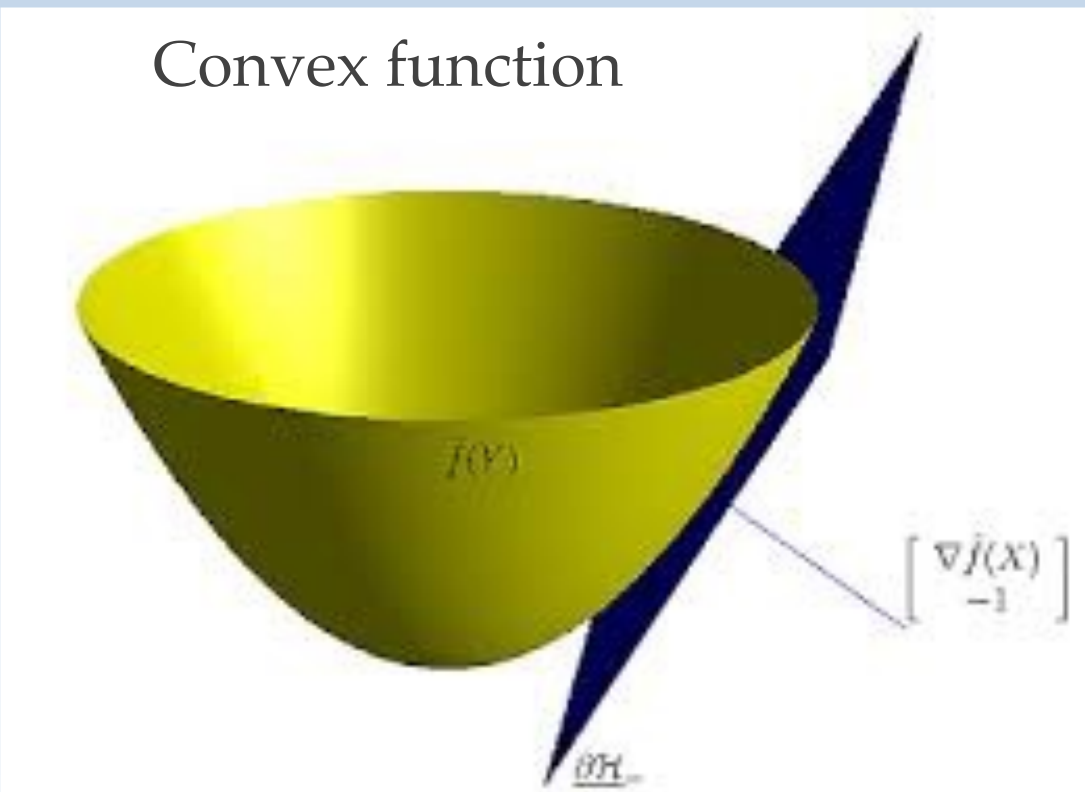
M = I for NEU

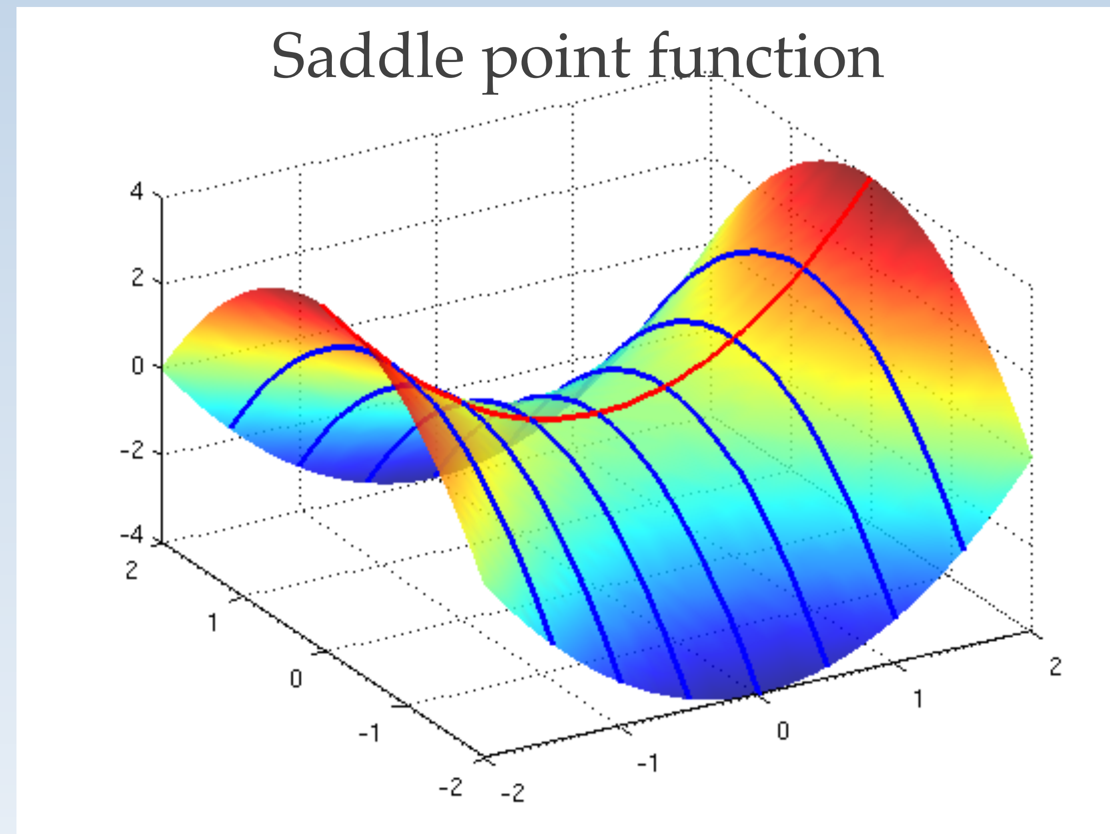
M = C for MSPBE

Primal objective function: $J(\theta) = \|b - A\theta\|^2_{M^{-1}}$

Saddle point formulation

$$J(\theta) = \|b - A\theta\|^2_{M^{-1}}$$





$$\min_{\theta} \max_{y} (L(\theta, y) = \langle b - A\theta, y \rangle - \frac{1}{2} \|y\|^2_M$$

Lemma

(Liu et al., UAI 2015)

Proposition 2. *GTD and GTD2 are true stochastic gradient algorithms w.r.t. the objective function $L(\theta, y)$ of the saddle-point problem* (14) *with $M = I$ and $M = C = \Phi^\top \Xi \Phi$ (the covariance matrix), respectively.*

Proof. It is easy to see that the gradient updates of the saddle-point problem (14) (ascending in y and descending in θ) may be written as

$$\begin{aligned} y_{t+1} &= y_t + \alpha_t \left(b - A\theta_t - M y_t \right), \\ \theta_{t+1} &= \theta_t + \alpha_t A^\top y_t. \end{aligned} \tag{16}$$

We may obtain the update rules of GTD and GTD2 by replacing A, b, and C in (16) with their unbiased estimates $\hat{A}$, $\hat{b}$, and $\hat{C}$ from Eq. 4, which completes the proof. □

“Gradient” TD Methods

(Sutton et al., 2009)

GTD:

$$y_{t+1} = y_t + \alpha_t\big(\rho_t \delta_t(\theta_t)\phi_t - y_t\big),$$
$$\theta_{t+1} = \theta_t + \alpha_t \rho_t \Delta\phi_t (y_t^\top \phi_t),$$

GTD2:

$$y_{t+1} = y_t + \alpha_t\big(\rho_t \delta_t(\theta_t) - \phi_t^\top y_t\big)\phi_t,$$
$$\theta_{t+1} = \theta_t + \alpha_t \rho_t \Delta\phi_t (y_t^\top \phi_t).$$

weighting factor $\rho_i = \pi(a_i|s_i)/\pi_b(a_i|s_i)$

Analysis of gradient TD

Proposition 3. *Let* $(\bar{\theta}_n, \bar{y}_n)$ *be the output of the GTD algorithm after* n *iterations (see Eq. 18). Then, with probability at least* $1-\delta$*, we have*

$$\mathrm{Err}(\bar{\theta}_n, \bar{y}_n) \leq \sqrt{\frac{5}{n}}(8 + 2\log\frac{2}{\delta})R^2 \times \left(\rho_{\max} L\left(2(1+\gamma)Ld + \frac{R_{\max}}{R}\right) + \tau + \frac{\sigma}{R}\right), \tag{24}$$

where $\mathrm{Err}(\bar{\theta}_n, \bar{y}_n)$ *is the error function of the saddle-point problem* (14) *defined by Eq. 13,* R *defined in Assumption 2,* σ *is from Eq. 21, and* $\tau = \sigma_{\max}(M)$ *is the largest singular value of* M*, which means* $\tau = 1$ *for GTD and* $\tau = \sigma_{\max}(C)$ *for GTD2.*

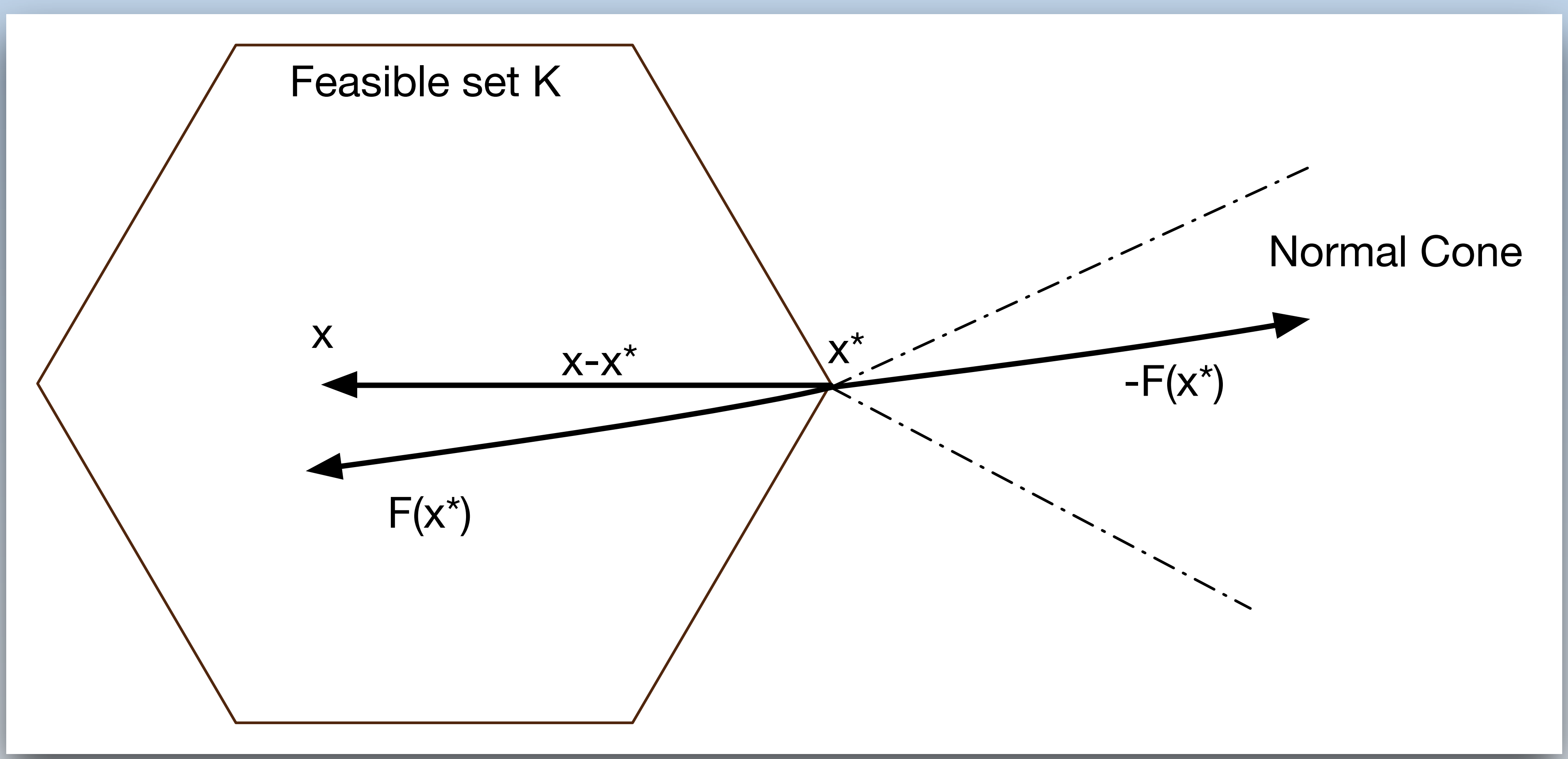
What is the “optimal” gradient TD method?

$$(\mathbf{GTD/GTD2}): \quad O\left(\frac{\tau + ||A||_2 + \sigma}{\sqrt{n}}\right)$$

$$\updownarrow$$

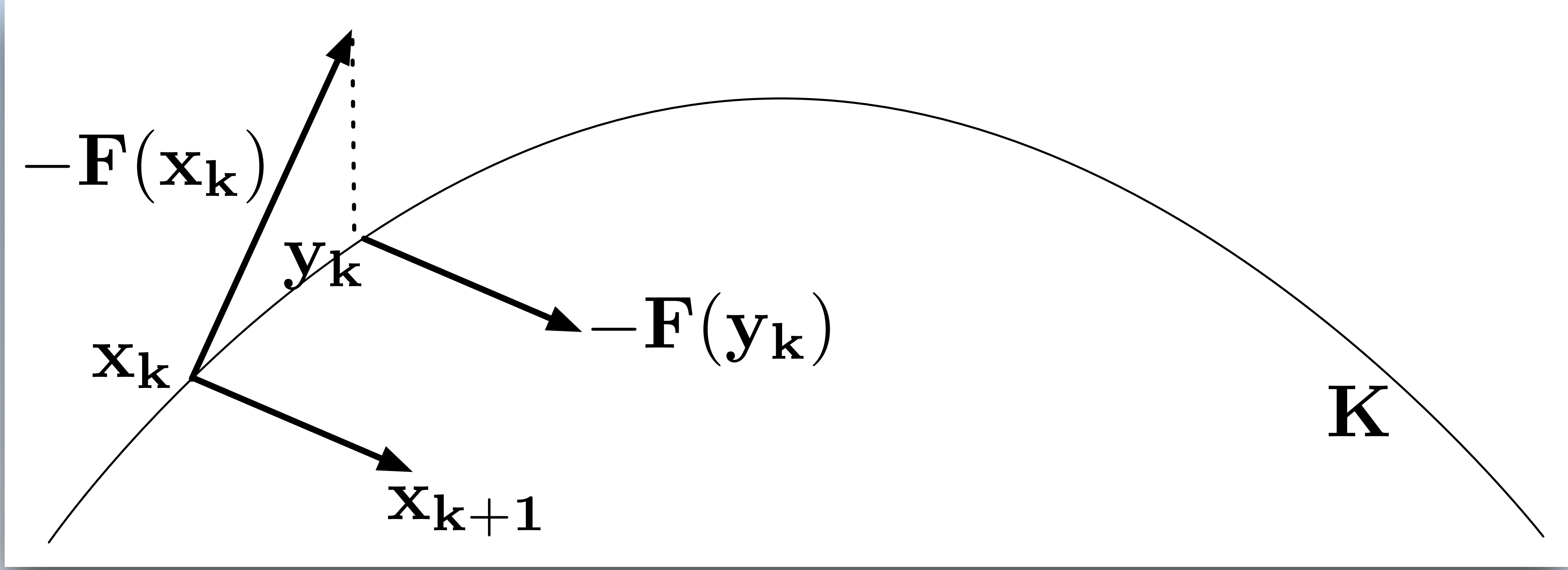
$$(\mathbf{Optimal}): \quad O\left(\frac{\tau}{n^2} + \frac{||A||_2}{n} + \frac{\sigma}{\sqrt{n}}\right)$$

Variational Inequality



$$\langle F(x^*), x - x^* \rangle \geq 0, \ \forall x \in K$$

Extragradient Method



Korpolevich (1970s) developed the extragradient method for solving saddle point problems and variational inequalities

Extragradient TD-Learning

Algorithm 2 GTD2-MP

for $t = 1, \ldots, n$ **do**

Update parameters

$$\begin{aligned}
\delta_t &= r_t - \theta_t^\top \Delta\phi_t \\
y_t^m &= y_t + \alpha_t(\rho_t\delta_t - \phi_t^\top y_t)\phi_t \\
\theta_t^m &= \theta_t + \alpha_t\rho_t\Delta\phi_t(\phi_t^\top y_t) \\
\delta_t^m &= r_t - (\theta_t^m)^\top \Delta\phi_t \\
y_{t+1} &= y_t + \alpha_t(\rho_t\delta_t^m - \phi_t^\top y_t^m)\phi_t \\
\theta_{t+1} &= \theta_t + \alpha_t\rho_t\Delta\phi_t(\phi_t^\top y_t^m)
\end{aligned}$$

end for

OUTPUT

$$\bar{\theta}_n := \frac{\sum_{t=1}^n \alpha_t\theta_t}{\sum_{t=1}^n \alpha_t} \quad , \quad \bar{y}_n := \frac{\sum_{t=1}^n \alpha_t y_t}{\sum_{t=1}^n \alpha_t} \tag{34}$$

What is the "optimal" gradient TD method?

$$(\textbf{GTD/GTD2}): \quad O\left(\frac{\tau + ||A||_2 + \sigma}{\sqrt{n}}\right)$$

$$(\textbf{SMP}): \quad O\left(\frac{\tau + ||A||_2}{n} + \frac{\sigma}{\sqrt{n}}\right)$$

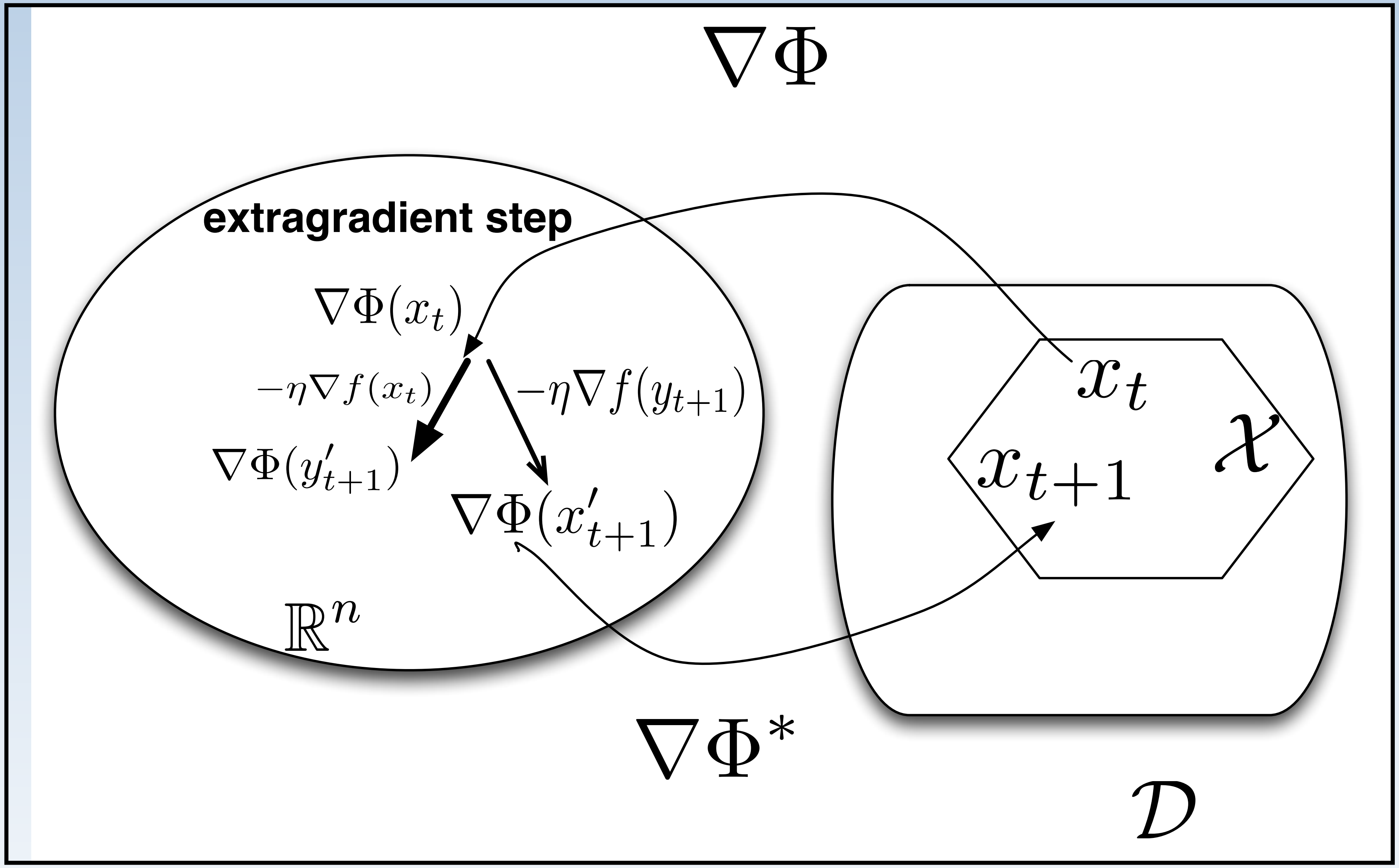
GTD-MP

"almost" dimension-free scalability

$$(\textbf{Optimal}): \quad O\left(\frac{\tau}{n^2} + \frac{||A||_2}{n} + \frac{\sigma}{\sqrt{n}}\right)$$

Mirror-Prox

(Nemirovski, 2005)



Proximal Gradient TD Algorithms

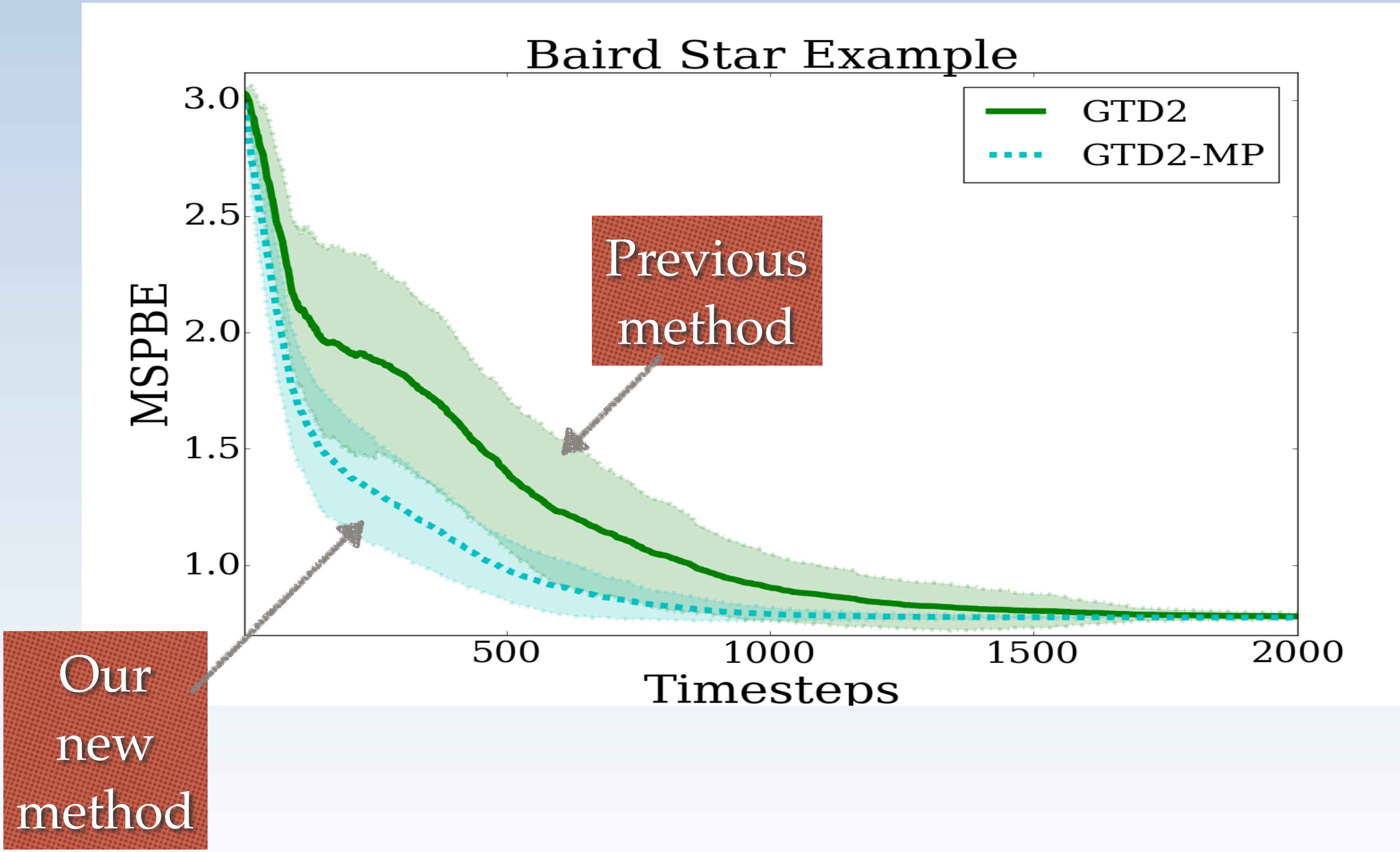
Algorithm 2 GTD2-MP

1. $w_{t+\frac{1}{2}} = w_t + \beta_t(\delta_t - \phi_t^T w_t)\phi_t,\ \theta_{t+\frac{1}{2}} = \text{prox}_{\alpha_t h}\left(\theta_t + \alpha_t(\phi_t - \gamma\phi_t')(\phi_t^T w_t)\right)$
2. $\delta_{t+\frac{1}{2}} = r_t + \gamma{\phi_t'}^T\theta_{t+\frac{1}{2}} - \phi_t^T\theta_{t+\frac{1}{2}}$
3. $w_{t+1} = w_t + \beta_t(\delta_{t+\frac{1}{2}} - \phi_t^T w_{t+\frac{1}{2}})\phi_t$, $\theta_{t+1} = \text{prox}_{\alpha_t h}\left(\theta_t + \alpha_t(\phi_t - \gamma\phi_t')(\phi_t^T w_{t+\frac{1}{2}})\right)$

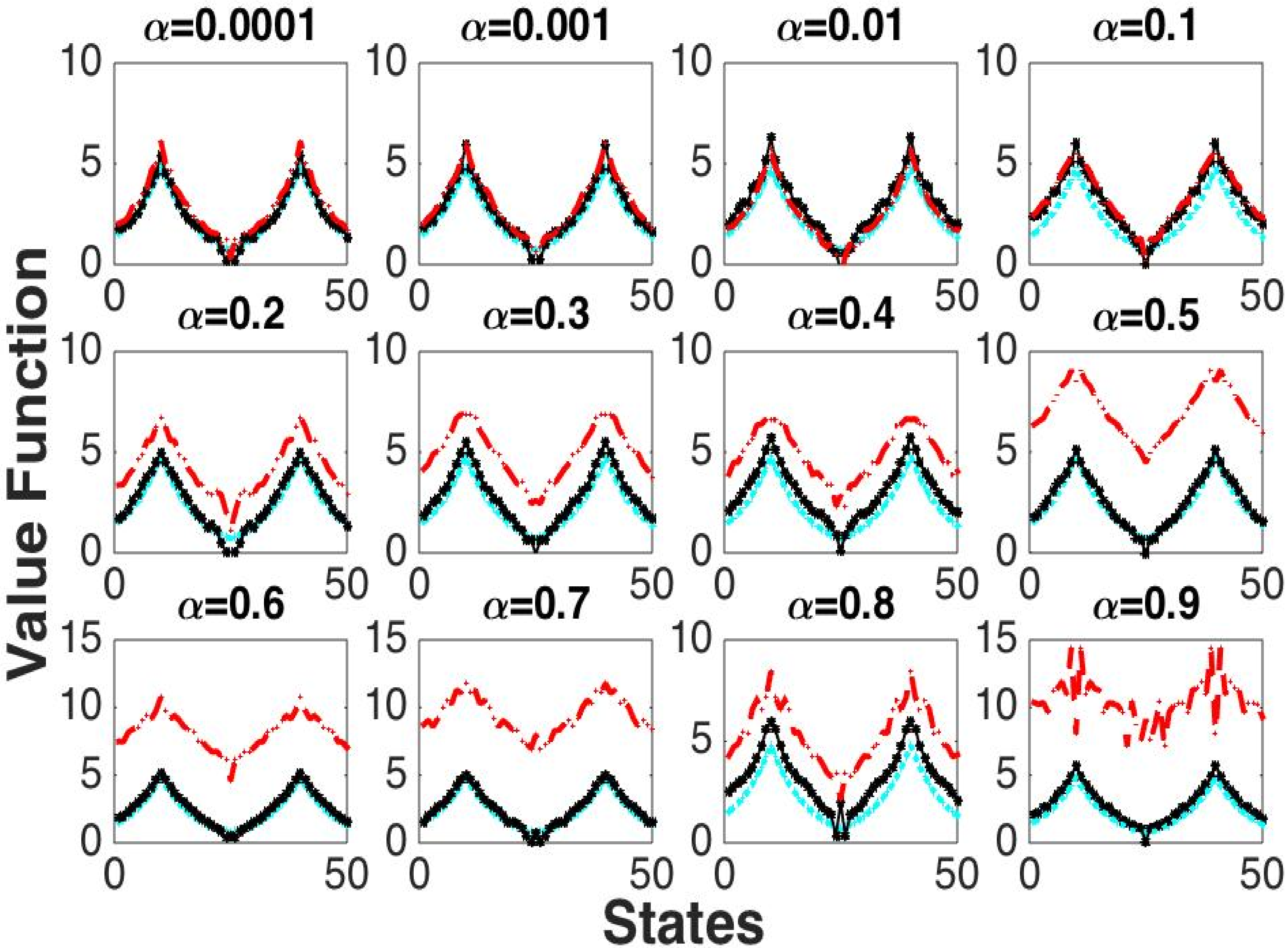
Algorithm 3 TDC-MP

1. $w_{t+\frac{1}{2}} = w_t + \beta_t(\delta_t - \phi_t^T w_t)\phi_t,\ \theta_{t+\frac{1}{2}} = \text{prox}_{\alpha_t h}\left(\theta_t + \alpha_t\delta_t\phi_t - \alpha_t\gamma{\phi_t}'(\phi_t^T w_t)\right)$
2. $\delta_{t+\frac{1}{2}} = r_t + \gamma{\phi_t'}^T\theta_{t+\frac{1}{2}} - \phi_t^T\theta_{t+\frac{1}{2}}$
3. $w_{t+1} = w_t + \beta_t(\delta_{t+\frac{1}{2}} - \phi_t^T w_{t+\frac{1}{2}})\phi_t$, $\theta_{t+1} = \text{prox}_{\alpha_t h}\left(\theta_t + \alpha_t\delta_{t+\frac{1}{2}}\phi_t - \alpha_t\gamma\phi_t'(\phi_t^T w_{t+\frac{1}{2}})\right)$

Baird MDP

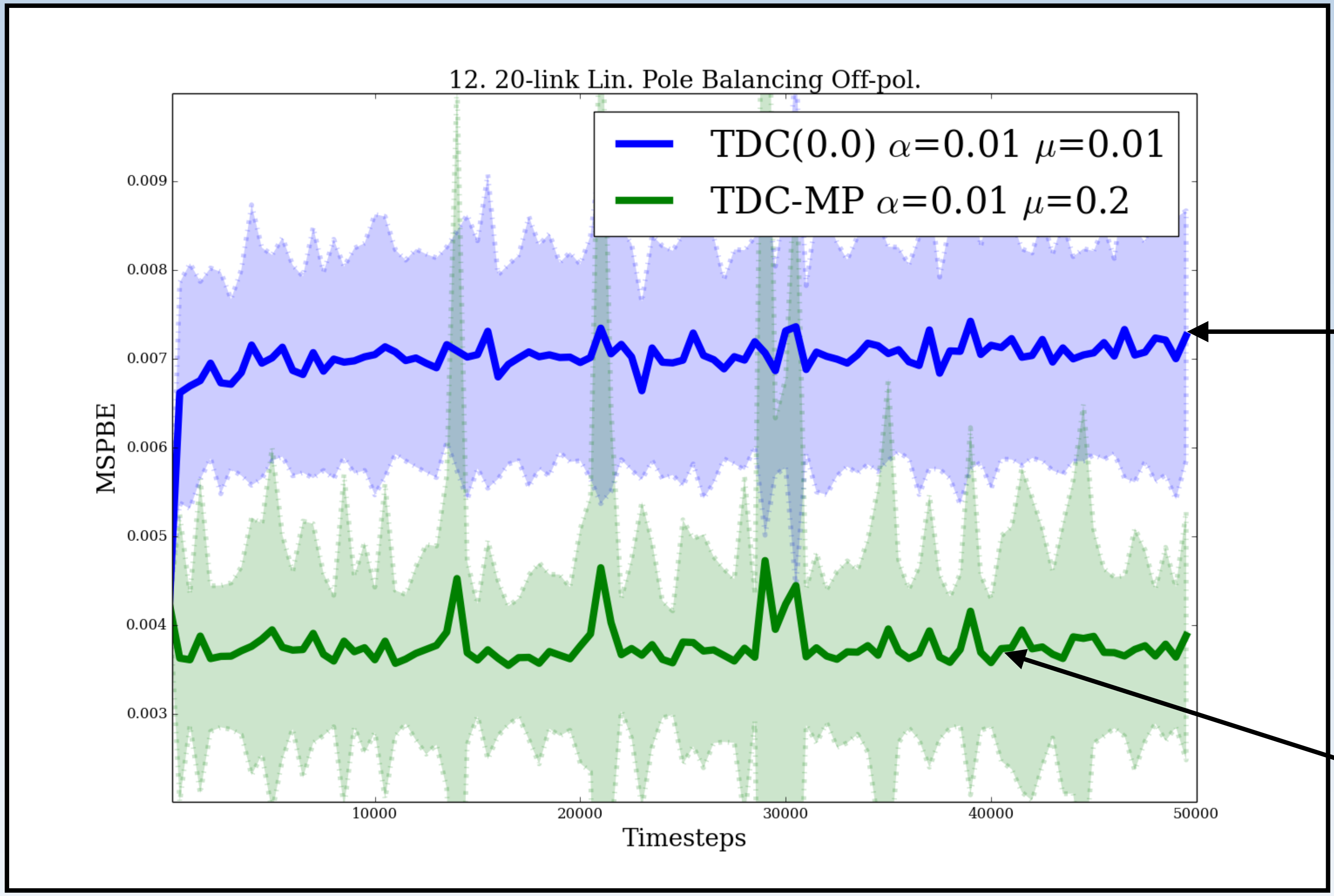

Baird Star Example
MSPBE
3.0
2.5
2.0
1.5
1.0
500
1000
1500
2000
Timesteps
GTD2
GTD2-MP
Previous method
Our new method

50 state chain domain



Red: GTD
Black: GTD-MP
Cyan: True VF

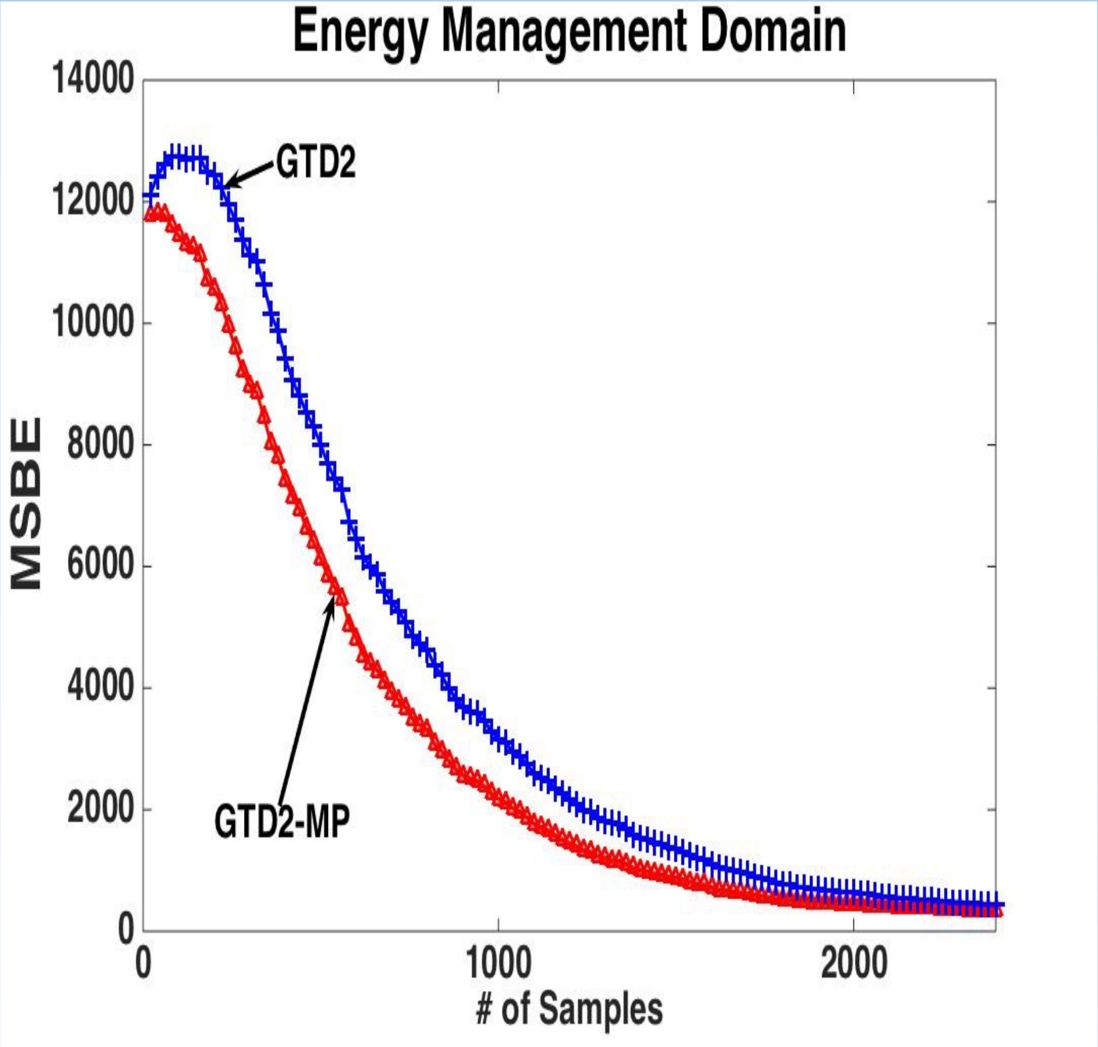
20-Dimensional Robot Arm

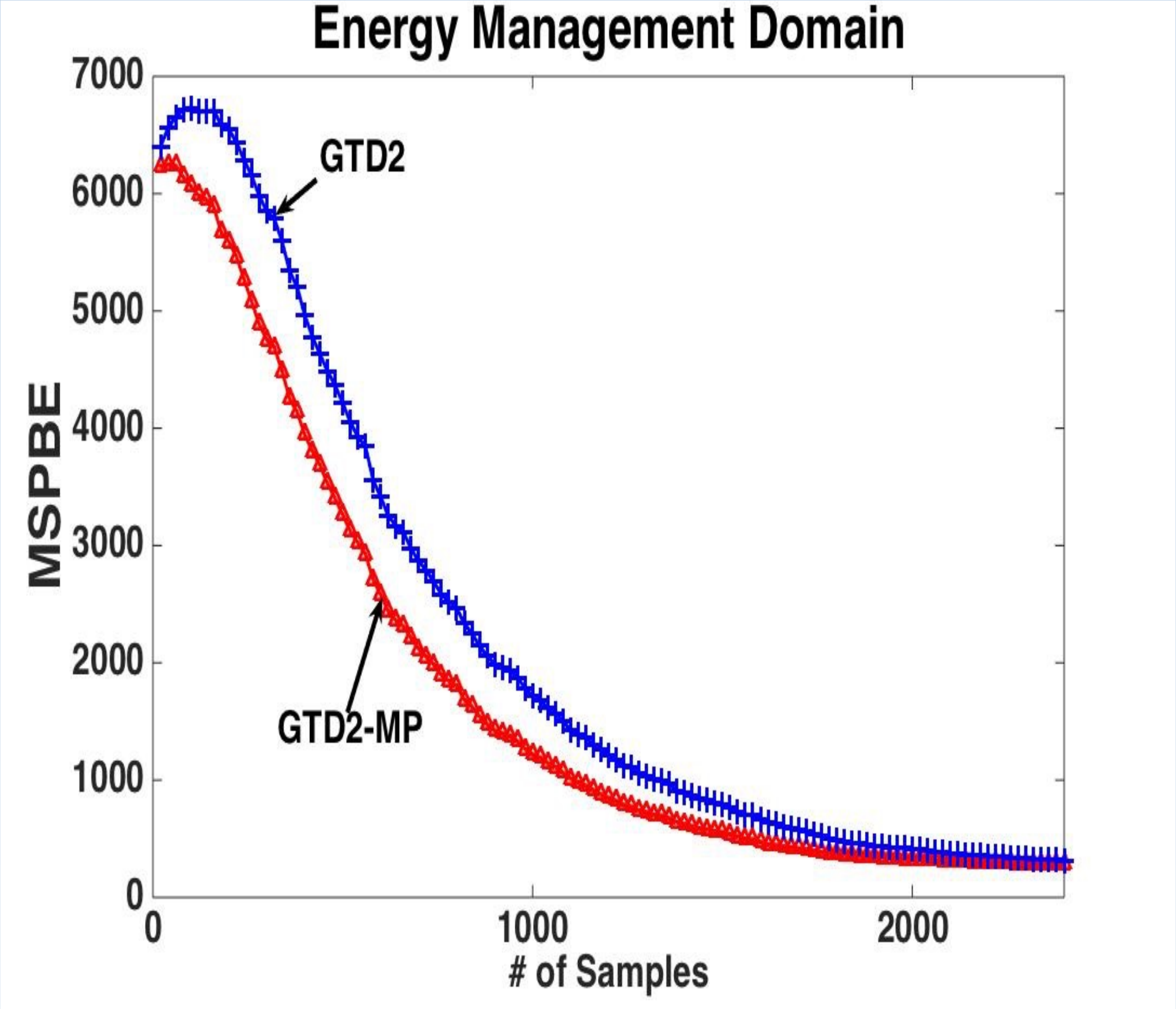


Sutton et al., 2009

Our new method

Energy Management





Algorithm	MSPBE	MSBE
GTD2	176.4	228.7
GTD2-MP	138.6	191.4

"Safe" Reinforcement Learning

Philip Thomas, Will Dabney,
Stephen Giguere, Sridhar Mahadevan

NIPS 2013, ICML 2014

5 Equivalence of Natural Gradient Descent and Mirror Descent

Theorem 5.1. *The natural gradient descent update at step k with metric tensor $G_k \triangleq G(x_k)$:*

$$x_{k+1} = x_k - \alpha_k G_k^{-1} \nabla f(x_k), \tag{2}$$

is equivalent to (1)*, the mirror descent update at step k, with $\psi_k(x) = (1/2) x^\intercal G_k x$.*

Proof. First, notice that $\nabla \psi_k(x) = G_k x$. Next, we derive a closed-form for ψ_k^*:

$$\psi_k^*(y) = \max_{x \in \mathbb{R}^n} \left\{ x^\intercal y - \frac{1}{2} x^\intercal G_k x \right\}. \tag{3}$$

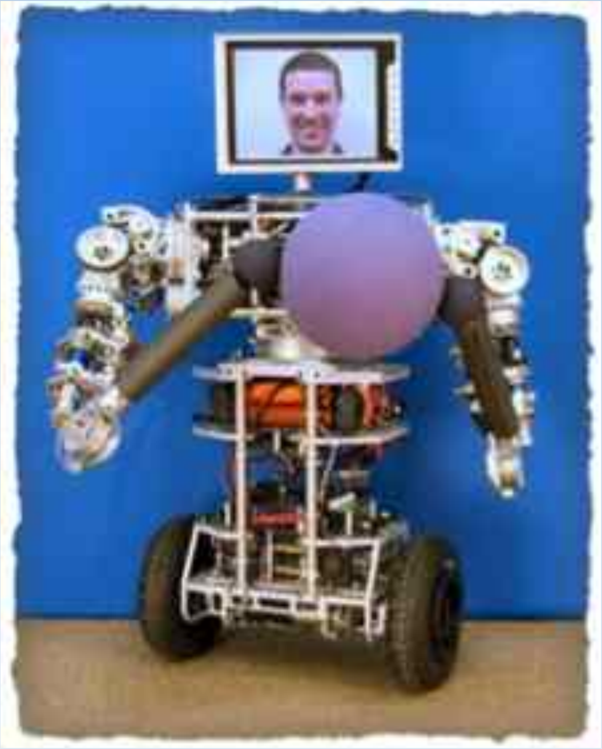
Since the function being maximized on the right hand side is strictly concave, the x that maximizes it is its critical point. Solving for this critical point, we get $x = G_k^{-1} y$. Substituting this into (3), we find that $\psi_k^*(y) = (1/2) y^\intercal G_k^{-1} y$. Hence, $\nabla \psi_k^*(y) = G_k^{-1} y$. Inserting the definitions of $\nabla \psi_k(x)$ and $\nabla \psi_k^*(y)$ into (1), we find that the mirror descent update is

$$x_{k+1} = G_k^{-1} \left(G_k x_k - \alpha_k \nabla f(x_k) \right) = x_k - \alpha_k G_k^{-1} \nabla f(x_k),$$

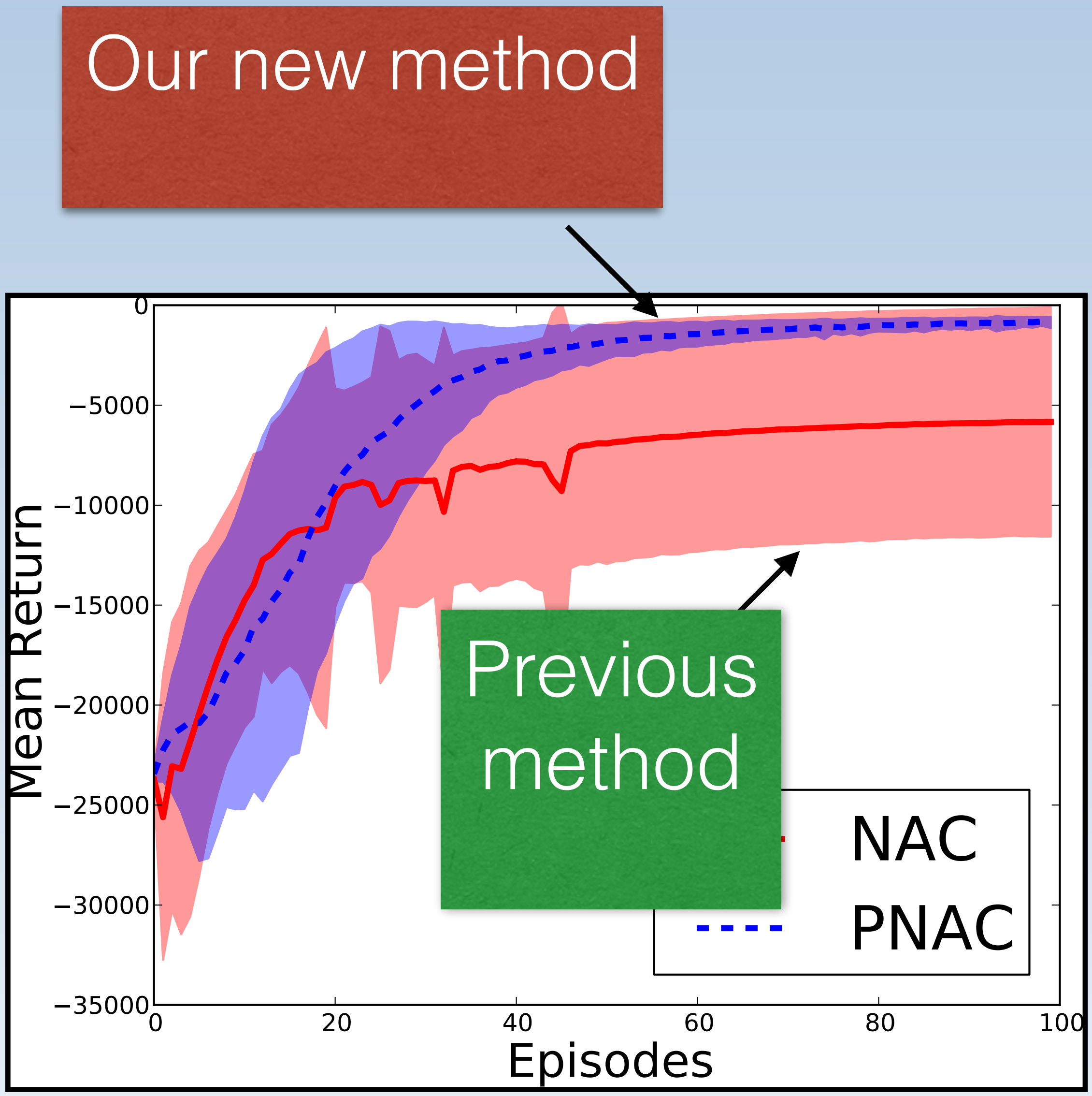
which is identical to (2). ■

Thomas, Dabney, Mahadevan, Giguere, NIPS 2013

Safe Robot Learning with PNAC



UBot, Laboratory of Perceptual Robotics



Thomas, Dabney, Mahadevan, Giguere, NIPS 2013

Proximal Reinforcement Learning: A New Theory of Sequential Decision Making in **Primal-Dual** Spaces,

Arxiv, May 26, 2014 (126 pages)

Sridhar Mahadevan, Bo Liu, Philip Thomas, Will Dabney, Stephen Giguere, Nicholas Jacek, Ian Gemp, Ji Liu

Ongoing Work

- New saddle-point formulation of gradient TD networks
  - Convergence rate and finite sample analysis
  - New scalable algorithms
- Applications
  - Integrating deep learning and gradient TD networks
  - Language models and transfer learning