$$w \leftarrow w - \frac{1}{2}\alpha \nabla \left[V_\pi(S_t) - \hat{V}(S_t, w) \right]^2$$

$$w \leftarrow w + \alpha \left[U_t - \hat{V}(S_t, w) \right] \cdot \nabla \hat{V}(S_t, w)$$

MC Estimate:

$$U_t = \sum_{i=t}^{T} \gamma^{i-t-1} R_i$$

- Unbiased
- fixed when w changes

Converges near global opt

Bootstrap Estimate:

$$U_t = R_t + \gamma \hat{V}(S_{t+1}, w)$$

- Biased
- Function of w → changes with w!
- Gradient calc treated U_t as a constant!

Converges to TD fixed point (local opt)

For TD fixed point w_{TD}:

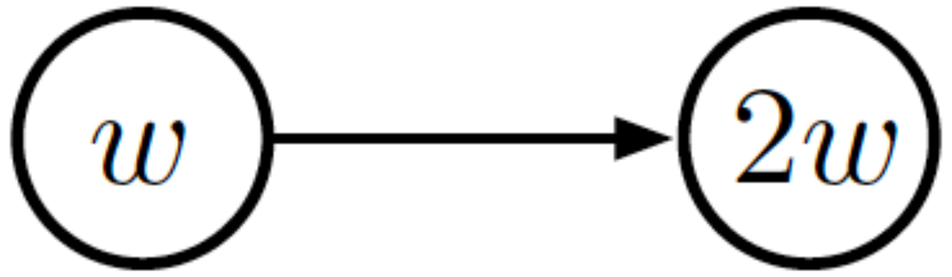
$$\overline{VE}(w_{TD}) \leq \frac{1}{1-\gamma} \min_w \left[\overline{VE}(w) \right]$$

The deadly triad

Divergence is possible when all 3 parts of the deadly triad are present:

- Function approximation
- Bootstrapping
- Off-Policy training

Off-policy semigradient methods



Stability of semigradient methods depends on on-policy distribution of updates. Why?

Imagine only updating one state S over and over again (i.e. off-policy):

- In tabular case, updating one state's value leaves all others changed
- With function approx + MC, multiple state values are updated, but V(S) is estimated independently of them via rewards only
- With function approx + TD (semigradient), multiple values are updated, which are then used to help estimate V(S) via bootstrapping, which are then updated again, which are then used to help estimates V(S)…

On-policy distribution forces state values to be "grounded" to something real

Proof of Convergence of Linear TD(0)

What properties assure convergence of the linear TD(0) algorithm (9.9)? Some insight can be gained by rewriting (9.10) as

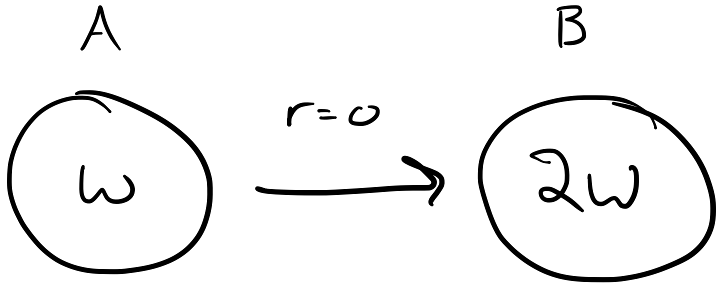
$$\mathbb{E}[\mathbf{w}_{t+1}|\mathbf{w}_t] = (\mathbf{I} - \alpha\mathbf{A})\mathbf{w}_t + \alpha\mathbf{b}. \tag{9.13}$$

Note that the matrix $\mathbf{A}$ multiplies the weight vector $\mathbf{w}_t$ and not $\mathbf{b}$; only $\mathbf{A}$ is important to convergence. To develop intuition, consider the special case in which $\mathbf{A}$ is a diagonal matrix. If any of the diagonal elements are negative, then the corresponding diagonal element of $\mathbf{I} - \alpha\mathbf{A}$ will be greater than one, and the corresponding component of $\mathbf{w}_t$ will be amplified, which will lead to divergence if continued. On the other hand, if the diagonal elements of $\mathbf{A}$ are all positive, then α can be chosen smaller than one over the largest of them, such that $\mathbf{I} - \alpha\mathbf{A}$ is diagonal with all diagonal elements between 0 and 1. In this case the first term of the update tends to shrink $\mathbf{w}_t$, and stability is assured. In general, $\mathbf{w}_t$ will be reduced toward zero whenever $\mathbf{A}$ is *positive definite*, meaning $y^\top \mathbf{A} y > 0$ for any real vector $y \neq 0$. Positive definiteness also ensures that the inverse $\mathbf{A}^{-1}$ exists.

For linear TD(0), in the continuing case with $\gamma < 1$, the $\mathbf{A}$ matrix (9.11) can be written

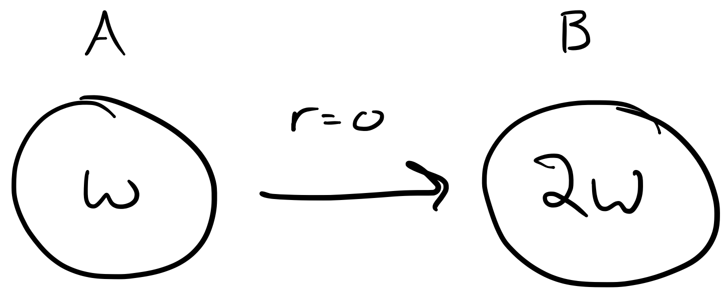
$$\begin{aligned}
\mathbf{A} &= \sum_s \mu(s) \sum_a \pi(a|s) \sum_{r,s'} p(r, s'|s, a)\mathbf{x}(s)\big(\mathbf{x}(s) - \gamma\mathbf{x}(s')\big)^\top \\
&= \sum_s \mu(s) \sum_{s'} p(s'|s)\mathbf{x}(s)\big(\mathbf{x}(s) - \gamma\mathbf{x}(s')\big)^\top \\
&= \sum_s \mu(s)\mathbf{x}(s)\Big(\mathbf{x}(s) - \gamma \sum_{s'} p(s'|s)\mathbf{x}(s')\Big)^\top \\
&= \mathbf{X}^\top \mathbf{D}(\mathbf{I} - \gamma\mathbf{P})\mathbf{X},
\end{aligned}$$

where $\mu(s)$ is the stationary distribution under π, $p(s'|s)$ is the probability of



Init: $w_0 = 10$ Thus: $\hat{V}(A) = 10$, $\hat{V}(B) = 20$

Assume $\alpha = 0.5$, $\gamma = 0.9$



Init: $w_0 = 10$ Thus: $\hat{v}(A) = 10$, $\hat{v}(B) = 20$

Assume $\alpha = 0.5$, $\gamma = 0.9$

observe Transition from A to B

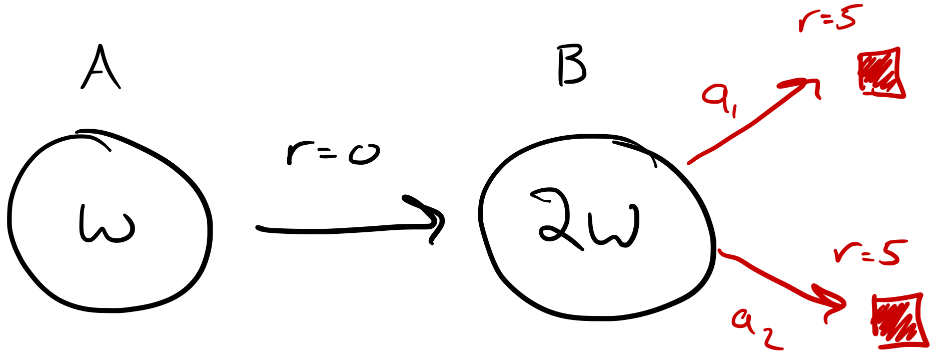
$$w_{t+1} = w_t + \alpha \rho \left[R_t + \gamma \hat{v}(B) - \hat{v}(A) \right] \nabla \hat{v}(A)$$

$$= 10 + (0.5)(1)\left[0 + .9(20) - 10\right] \cdot 1$$

$$= 10 + 4$$

$$= 14$$

Thus: $\hat{v}(A) = 14$, $\hat{v}(B) = 28$



$\pi_b(a_1, B) = 1$

$\pi(a_1, B) = 0$

- off policy ignores transition from B and diverges!
- on-policy uses transitions from B, which lowers $\hat{v}(B)$, and $\hat{v}(A)$ and converges

Init: $w_0 = 10$ Thus: $\hat{v}(A) = 10$, $\hat{v}(B) = 20$

Assume $\alpha = 0.5$, $\gamma = 0.9$

observe Transition from A to B

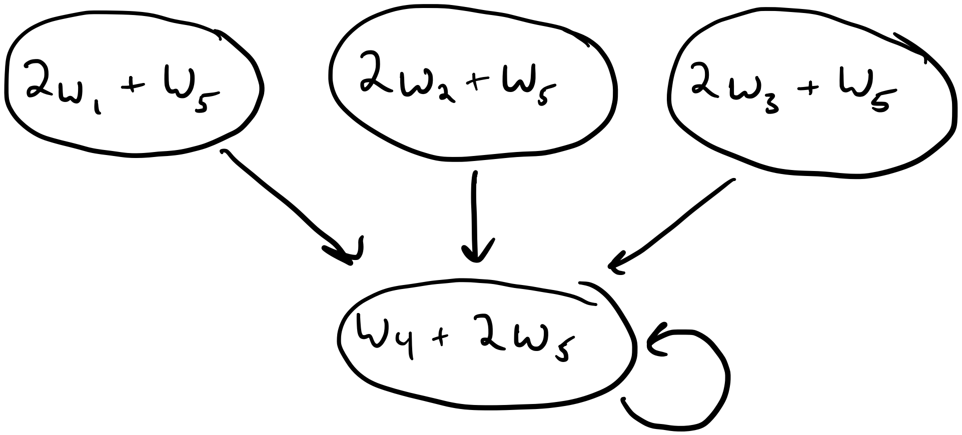
$$w_{t+1} = w_t + \alpha\rho\left[R_t + \gamma\hat{v}(B) - \hat{v}(A)\right]\nabla\hat{v}(A)$$

$$= 10 + (0.5)(1)\left[0 + .9(20) - 10\right]\cdot 1$$

$$= 10 + 4$$

$$= 14$$

Thus: $\hat{v}(A) = 14$, $\hat{v}(B) = 28$

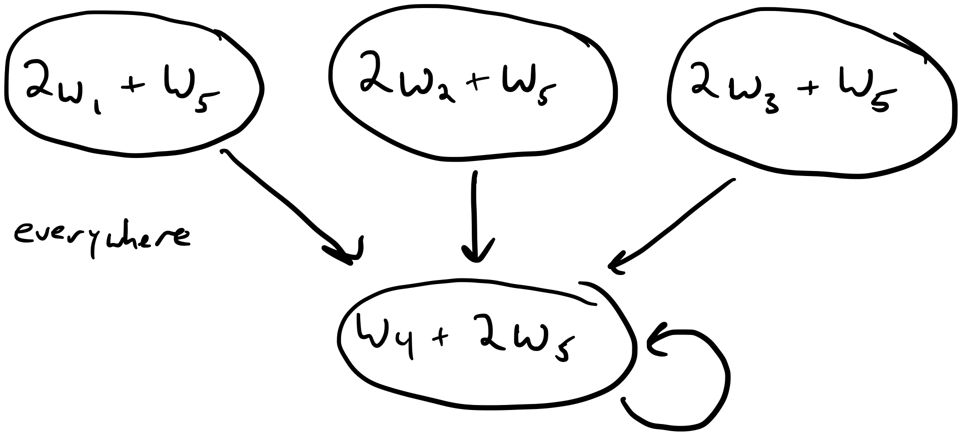


$\Rightarrow$

$$X: \begin{bmatrix} 2 & 0 & 0 & 0 & 1 \\ 0 & 2 & 0 & 0 & 1 \\ 0 & 0 & 2 & 0 & 1 \\ 0 & 0 & 0 & 1 & 2 \end{bmatrix}$$

if $w_0 = \begin{bmatrix} 1 & 1 & 1 & 10 & 1 \end{bmatrix}^T$ then,

$$\hat{V} = \begin{bmatrix} 3 & 3 & 3 & 12 \end{bmatrix}^T$$



$r = 0$ everywhere

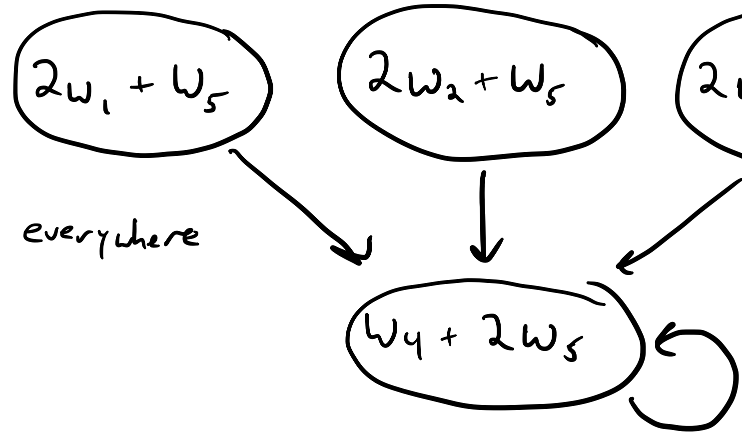
$$x: \begin{bmatrix} 2 & 0 & 0 & 0 & 1 \\ 0 & 2 & 0 & 0 & 1 \\ 0 & 0 & 2 & 0 & 1 \\ 0 & 0 & 0 & 1 & 2 \end{bmatrix}$$

if $w_0 = \begin{bmatrix} 1 & 1 & 1 & 10 & 1 \end{bmatrix}^T$ then,

$$\hat{V} = \begin{bmatrix} 3 & 3 & 3 & 12 \end{bmatrix}^T$$

DP update:

$$w_{t+1} = w_t + \frac{\alpha}{|S|} \sum_s \left(E_\pi[R_{t+1} + \gamma \hat{v}(s_{t+1})] - \hat{v}(s) \right) \nabla v(s)$$



$r = 0$ everywhere

$$\Rightarrow \quad x: \begin{bmatrix} 2 & 0 & 0 & 0 & 1 \\ 0 & 2 & 0 & 0 & 1 \\ 0 & 0 & 2 & 0 & 1 \\ 0 & 0 & 0 & 1 & 2 \end{bmatrix}$$

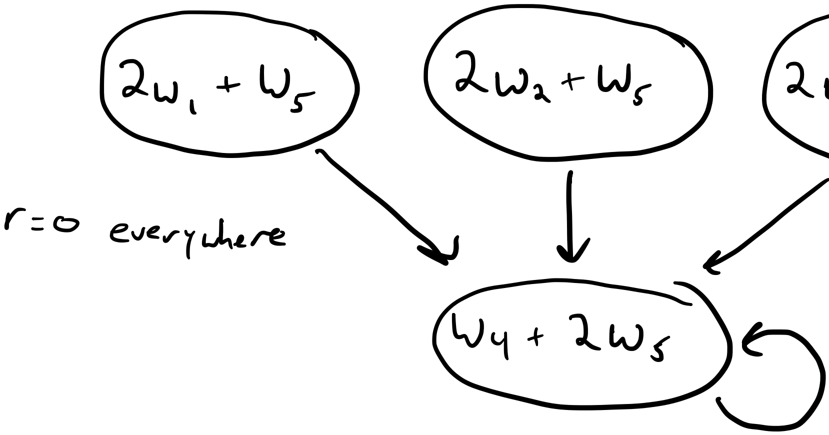
if $w_0 = \begin{bmatrix} 1 & 1 & 1 & 10 & 1 \end{bmatrix}^T$ then,

$$\hat{V} = \begin{bmatrix} 3 & 3 & 3 & 12 \end{bmatrix}^T$$

DP update:

$$w_{t+1} = w_t + \frac{\alpha}{|S|} \sum_s \left(E_\pi\left[R_{t+1} + \gamma \hat{v}(S_{t+1})\right] - \hat{v}(s) \right) \nabla v(s)$$

$$\begin{aligned} = w_t + 1/4 \Big[& (12-3)\begin{bmatrix} 2 & 0 & 0 & 0 & 1 \end{bmatrix}^T + \\ & (12-3)\begin{bmatrix} 0 & 2 & 0 & 0 & 1 \end{bmatrix}^T + \\ & (12-3)\begin{bmatrix} 0 & 0 & 2 & 0 & 1 \end{bmatrix}^T + \\ & (12-12)\begin{bmatrix} 0 & 0 & 0 & 1 & 2 \end{bmatrix}^T \Big] \end{aligned}$$



$$\Rightarrow \quad X: \begin{bmatrix} 2 & 0 & 0 & 0 & 1 \\ 0 & 2 & 0 & 0 & 1 \\ 0 & 0 & 2 & 0 & 1 \\ 0 & 0 & 0 & 1 & 2 \end{bmatrix}$$

if $w_0 = [1 \;\; 1 \;\; 1 \;\; 10 \;\; 1]^T$ then,

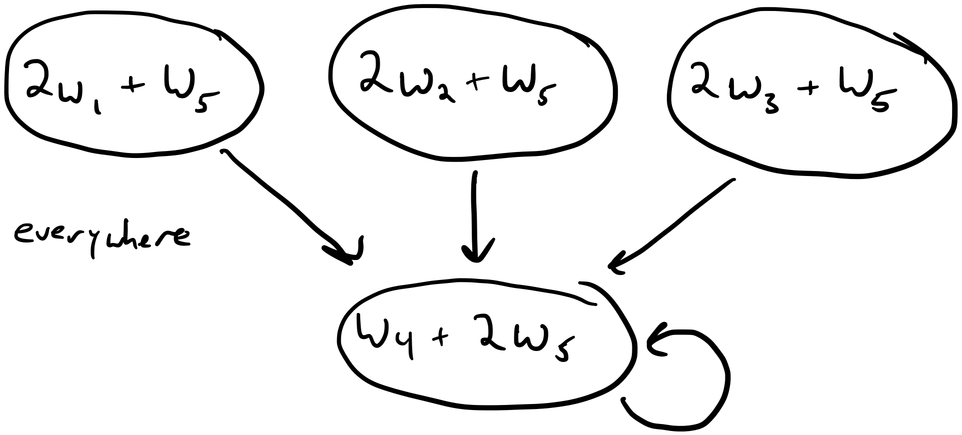
$$\hat{V} = [3 \;\; 3 \;\; 3 \;\; 12]^T$$

DP update:

$$w_{t+1} = w_t + \frac{\alpha}{|S|} \sum_s \left(E_\pi[R_{t+1} + \gamma \hat{v}(s_{t+1})] - \hat{v}(s) \right) \nabla v(s)$$

$$\begin{aligned} = w_t + 1/4 \Big[& (12-3)[2 \;\; 0 \;\; 0 \;\; 0 \;\; 1]^T + \\ & (12-3)[0 \;\; 2 \;\; 0 \;\; 0 \;\; 1]^T + \\ & (12-3)[0 \;\; 0 \;\; 2 \;\; 0 \;\; 1]^T + \\ & (12-12)[0 \;\; 0 \;\; 0 \;\; 1 \;\; 2]^T \Big] \end{aligned}$$

$$= [1 \;\; 1 \;\; 1 \;\; 10 \;\; 1]^T + 1/4 \cdot [18 \;\; 18 \;\; 18 \;\; 0 \;\; 54]^T$$



$r = 0$ everywhere

$\Rightarrow$ $X:$ $$\begin{bmatrix} 2 & 0 & 0 & 0 & 1 \\ 0 & 2 & 0 & 0 & 1 \\ 0 & 0 & 2 & 0 & 1 \\ 0 & 0 & 0 & 1 & 2 \end{bmatrix}$$

if $w_0 = [1 \;\; 1 \;\; 1 \;\; 10 \;\; 1]^T$ then,

$$\hat{V} = [3 \;\; 3 \;\; 3 \;\; 12]^T$$

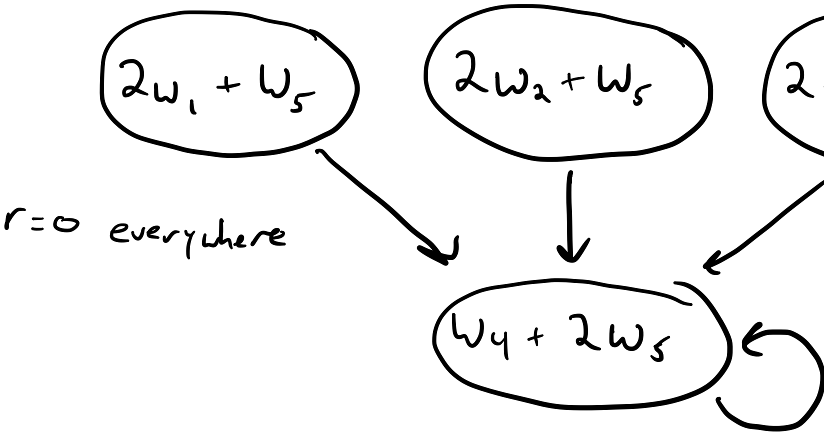
DP update:

$$w_{t+1} = w_t + \frac{\alpha}{|S|} \sum_s \Big(E_\pi[R_{t+1} + \gamma \hat{v}(s_{t+1})] - \hat{v}(s) \Big) \nabla v(s)$$

$$= w_t + 1/4 \Big[(12-3)[2 \;\; 0 \;\; 0 \;\; 0 \;\; 1]^T + (12-3)[0 \;\; 2 \;\; 0 \;\; 0 \;\; 1]^T + (12-3)[0 \;\; 0 \;\; 2 \;\; 0 \;\; 1]^T + (12-12)[0 \;\; 0 \;\; 0 \;\; 1 \;\; 2]^T \Big]$$

$$= [1 \;\; 1 \;\; 1 \;\; 10 \;\; 1]^T + 1/4 \cdot [18 \;\; 18 \;\; 18 \;\; 0 \;\; 54]^T$$

$$= [5.5 \;\; 5.5 \;\; 5.5 \;\; 10 \;\; 15]^T$$



$$\Rightarrow \quad X: \begin{bmatrix} 2 & 0 & 0 & 0 & 1 \\ 0 & 2 & 0 & 0 & 1 \\ 0 & 0 & 2 & 0 & 1 \\ 0 & 0 & 0 & 1 & 2 \end{bmatrix}$$

if $w_0 = [1 \;\; 1 \;\; 1 \;\; 10 \;\; 1]^T$ then,

$$\hat{V} = [3 \;\; 3 \;\; 3 \;\; 12]^T$$

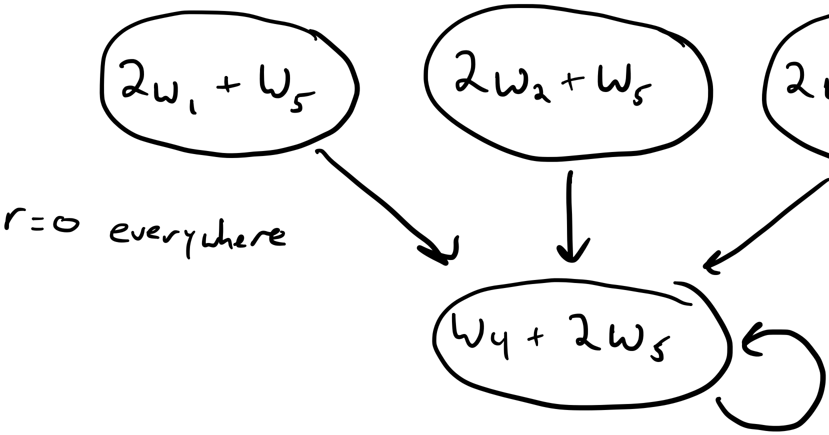
DP update:

$$w_{t+1} = w_t + \frac{\alpha}{|S|} \sum_s \left(E_\pi\left[R_{t+1} + \gamma \hat{v}(S_{t+1})\right] - \hat{v}(s) \right) \nabla v(s)$$

$$= w_t + 1/4 \Big[(12-3)[2 \;\; 0 \;\; 0 \;\; 0 \;\; 1]^T + (12-3)[0 \;\; 2 \;\; 0 \;\; 0 \;\; 1]^T + (12-3)[0 \;\; 0 \;\; 2 \;\; 0 \;\; 1]^T + (12-12)[0 \;\; 0 \;\; 0 \;\; 1 \;\; 2]^T \Big]$$

$$= [1 \;\; 1 \;\; 1 \;\; 10 \;\; 1]^T + 1/4 \cdot [18 \;\; 18 \;\; 18 \;\; 0 \;\; 54]^T$$

$$= [5.5 \;\; 5.5 \;\; 5.5 \;\; 10 \;\; 15]^T \quad \Longrightarrow \quad \hat{v} = [26 \;\; 26 \;\; 26 \;\; 40]^T$$



$$\Rightarrow \quad X: \begin{bmatrix} 2 & 0 & 0 & 0 & 1 \\ 0 & 2 & 0 & 0 & 1 \\ 0 & 0 & 2 & 0 & 1 \\ 0 & 0 & 0 & 1 & 2 \end{bmatrix}$$

if $w_0 = [1 \ 1 \ 1 \ 10 \ 1]^T$ then,

$$\hat{V} = [3 \ 3 \ 3 \ 12]^T$$

DP update:

$$w_{t+1} = w_t + \frac{\alpha}{|S|} \sum_s \Big(E_\pi\big[R_{t+1} + \gamma \hat{v}(S_{t+1})\big] - \hat{v}(s) \Big) \nabla v(s)$$

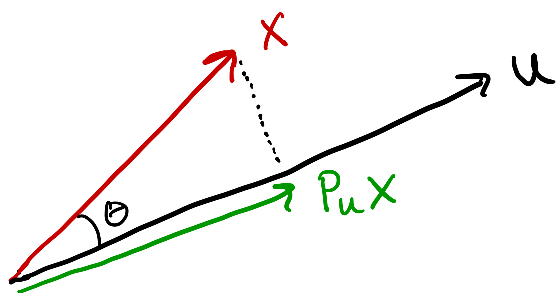
$$= w_t + 1/4 \Big[(12-3)[2 \ 0 \ 0 \ 0 \ 1]^T + (12-3)[0 \ 2 \ 0 \ 0 \ 1]^T + (12-3)[0 \ 0 \ 2 \ 0 \ 1]^T + (12-12)[0 \ 0 \ 0 \ 1 \ 2]^T \Big]$$

$$= [1 \ 1 \ 1 \ 10 \ 1]^T + 1/4 \cdot [18 \ 18 \ 18 \ 0 \ 54]^T$$

$$= [5.5 \ 5.5 \ 5.5 \ 10 \ 15]^T \quad \Rightarrow \quad \hat{v} = [26 \ 26 \ 26 \ 40]^T$$

- DP updates with off-policy state dist (uniform), but policy π always follows solid lines
- Updates top states more often than it should
- Bottom state looks better and raises value of top state
- ... which raises value of bottom state, which ...

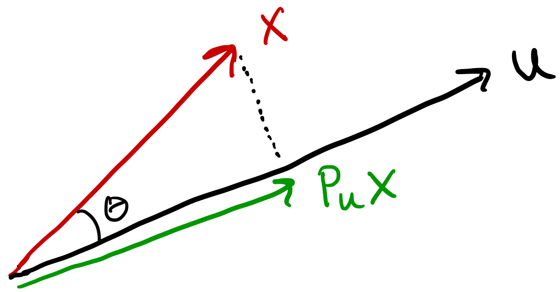
Orthogonal Projection



Projection to a unit vector : $P_u = uu^T$

So: $P_u x = uu^T x$

Orthogonal Projection



Projection to a unit vector: $P_u = uu^T$

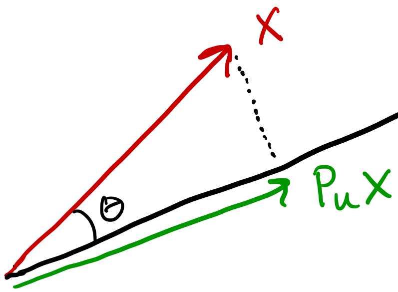
So: $P_u x = uu^T x$

Why?

$u^T x = \|x\| \cos\theta$

$uu^T x$ is a vector of magnitude $\|x\| \cos\theta$ in the direction of u

Orthogonal Projection



Projection to a unit vector : $P_u = uu^T$

So: $P_u x = uu^T x$

Why?

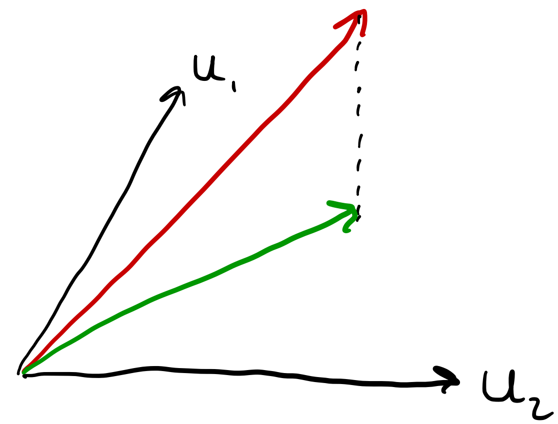
$u^T x = \|x\| \cos \theta$

$uu^T x$ is a vector of magnitude $\|x\| \cos \theta$
in the direction of u

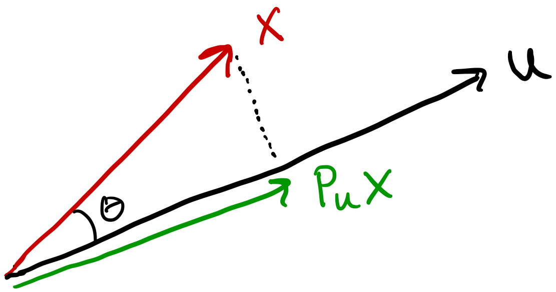
More Generally:

if $A = [u_1 \dots u_k]$ is an orthonormal basis of the subspace U, then:

$P_A = AA^T$



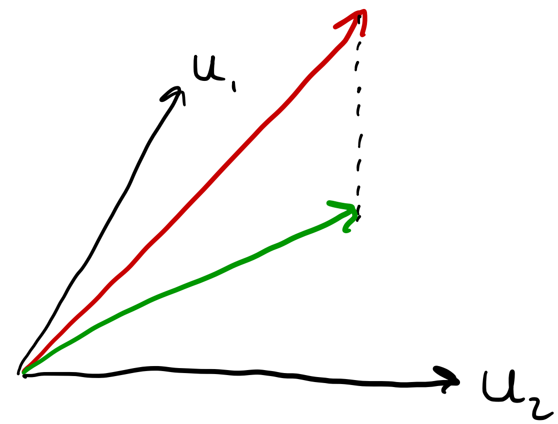
Orthogonal Projection



More Generally:

if $A = [u_1 \dots u_k]$ is an orthonormal basis of the subspace U, then:

$$P_A = AA^T$$



Projection to a unit vector: $P_u = uu^T$

So: $P_u x = uu^T x$

Why?

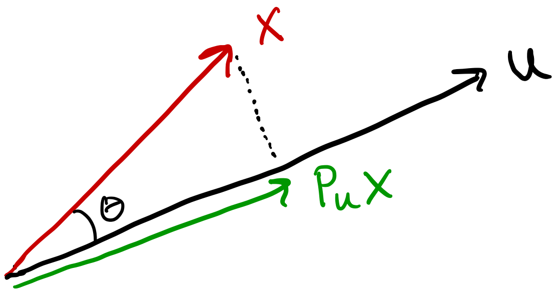
$u^T x = \|x\| \cos\theta$

$uu^T x$ is a vector of magnitude $\|x\| \cos\theta$ in the direction of u

what if $u_1 \dots u_k$ not orthonormal?

$$P_A = A \underbrace{(A^T A)^{-1}}_{\text{normalizing factor}} A^T$$

Orthogonal Projection



Projection to a unit vector : $P_u = uu^T$

So : $P_u x = uu^T x$

Why?

$u^T x = \|x\| \cos\theta$

$uu^T x$ is a vector of magnitude $\|x\| \cos\theta$ in the direction of u

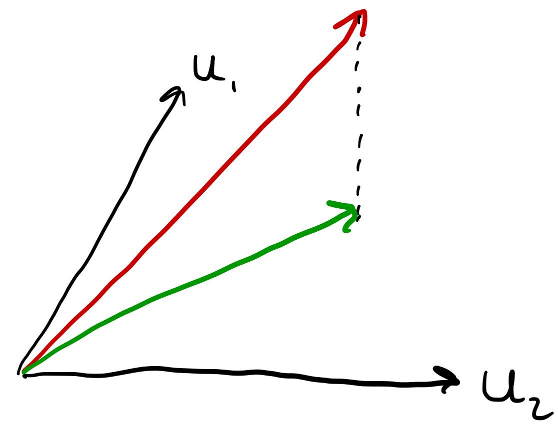
What if $u_1 \dots u_k$ not orthonormal?

$$P_A = A \underbrace{(A^T A)^{-1}}_{\text{normalizing factor}} A^T$$

More Generally:

if $A = [u_1 \dots u_k]$ is an orthonormal basis of the subspace U, then:

$$P_A = AA^T$$

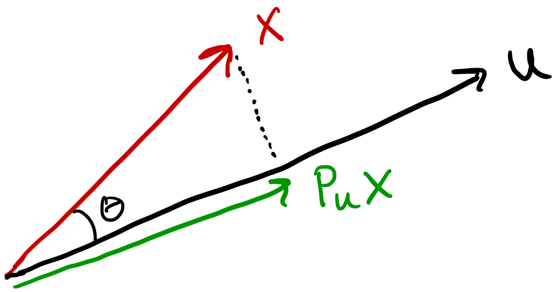


What about other inner products?

$$P_A = A(A^T D A)^{-1} A^T D$$

$\langle x, y \rangle = y^T D x$

Orthogonal Projection



Projection to a unit vector: $P_u = uu^T$

So: $P_u x = uu^T x$

Why?

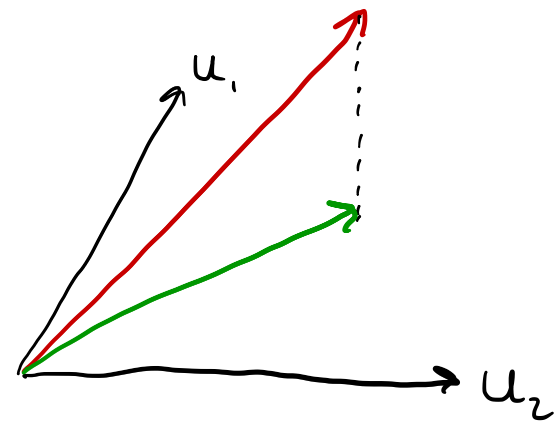
$u^T x = \|x\| \cos \theta$

$uu^T x$ is a vector of magnitude $\|x\| \cos \theta$ in the direction of u

More Generally:

if $A = [u_1 \dots u_k]$ is an orthonormal basis of the subspace U, then:

$P_A = AA^T$



What if $u_1 \dots u_k$ not orthonormal?

$$P_A = A \underbrace{(A^T A)^{-1}}_{\text{normalizing factor}} A^T$$

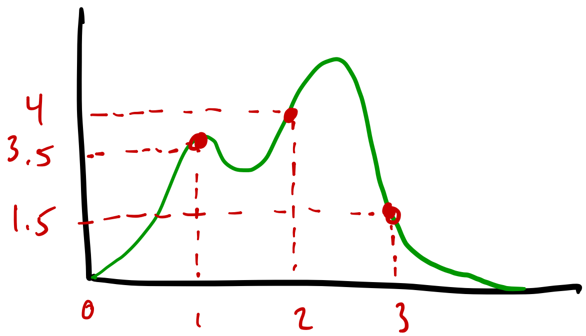
$\boxed{\text{Linear regression: } \hat{y} = X(X^T X)^{-1} X^T y}$

What about other inner products?

$P_A = A(A^T D A)^{-1} A^T D$

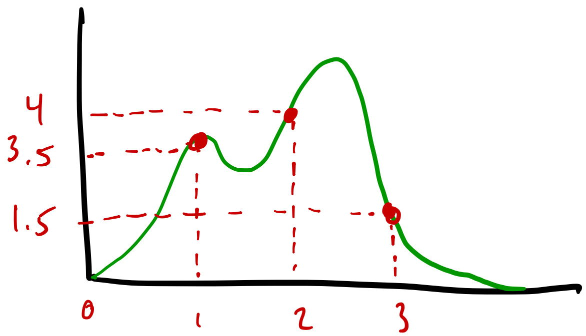
$\langle x, y \rangle = y^T D x$

Basis functions vs. vectors



$\Rightarrow \begin{bmatrix} 1.5 & 3.5 & 4 \end{bmatrix}$

Basis functions vs. vectors



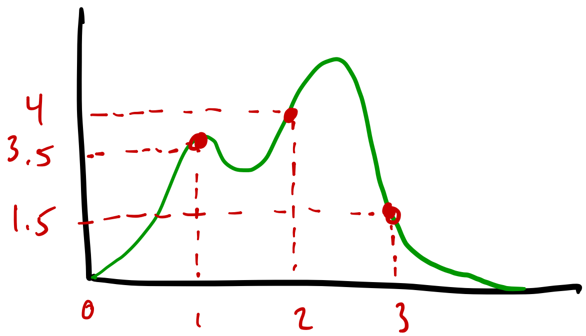
$\Rightarrow \begin{bmatrix} 1.5 & 3.5 & 4 \end{bmatrix}$

If increased to an infinite number of points, an infinite-dimensional vector can represent a function exactly!

Thus, basis vectors can represent basis functions

... and be projected onto!

Basis functions vs. vectors

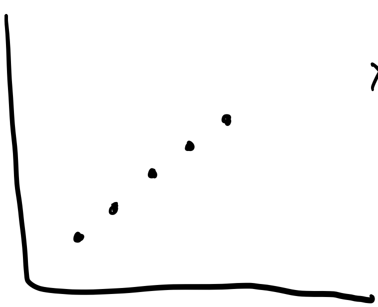


$\Rightarrow \begin{bmatrix} 1.5 & 3.5 & 4 \end{bmatrix}$

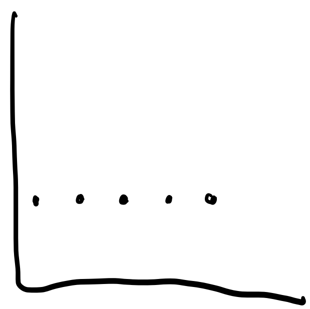
If increased to an infinite number of points, an infinite-dimensional vector can represent a function exactly!

Thus, basis vectors can represent basis functions ... and be projected onto!

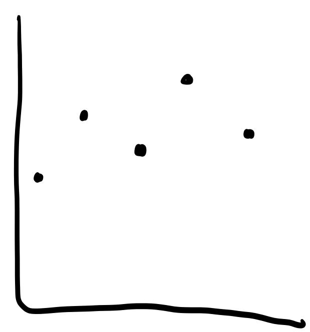
x_1:



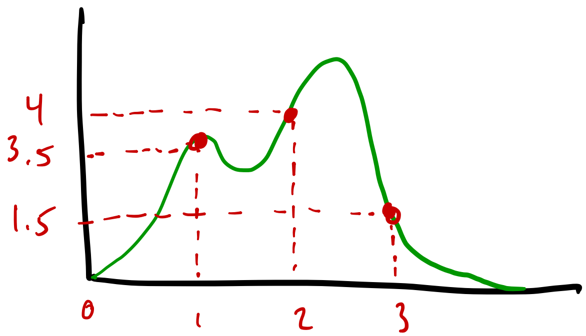
x_2:



y:



Basis functions vs. vectors



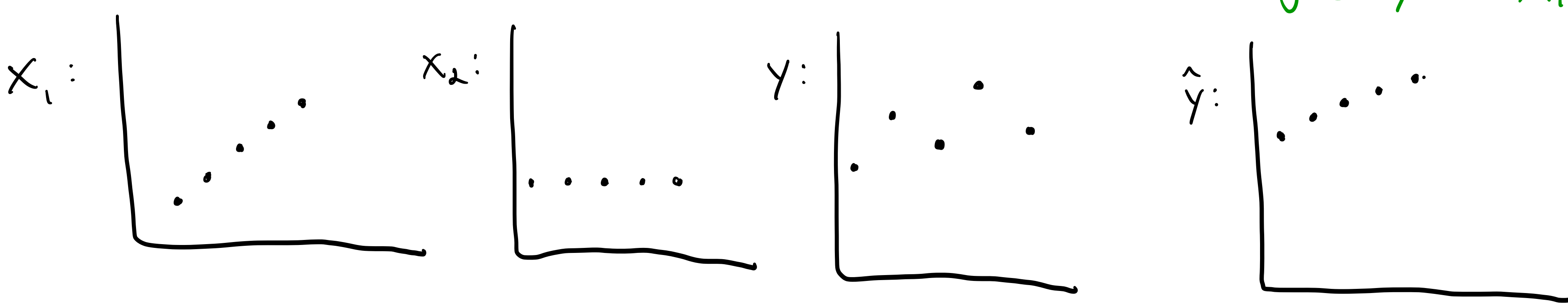
$\Rightarrow [1.5 \quad 3.5 \quad 4]$

If increased to an infinite number of points, an infinite-dimensional vector can represent a function exactly!

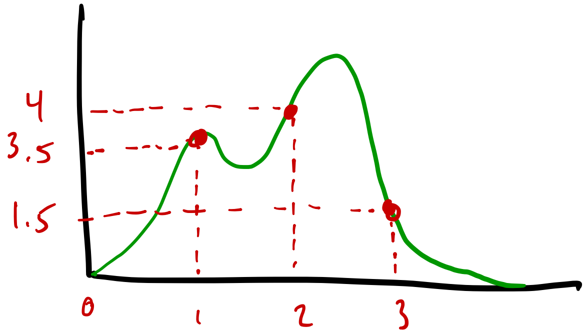
Thus, basis vectors can represent basis functions

... and be projected onto!

Project y onto X_1, X_2



Basis functions vs. vectors



$\Rightarrow \begin{bmatrix} 1.5 & 3.5 & 4 \end{bmatrix}$

$$P_X = X(X^T D X)^{-1} X^T D$$

where X is $|S| \times |\text{features}|$

and D is : $\begin{bmatrix} \mu(s_1) & & \\ & \ddots & \\ & & \mu(s_n) \end{bmatrix}$

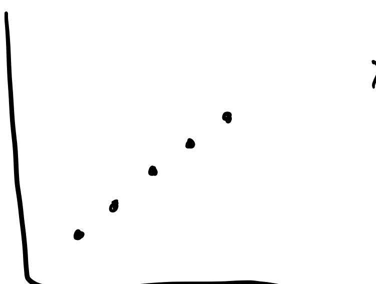
If increased to an infinite number of points, an infinite-dimensional vector can represent a function exactly!

Thus, basis vectors can represent basis functions

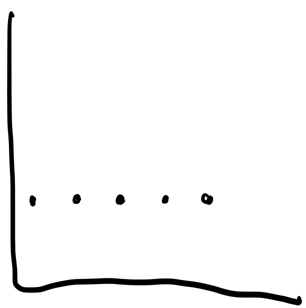
... and be projected onto!

Project y onto X_1, X_2

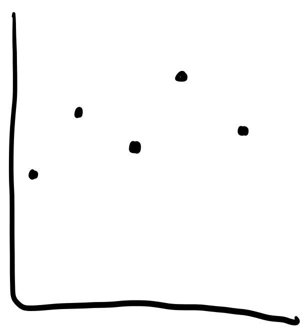
X_1:



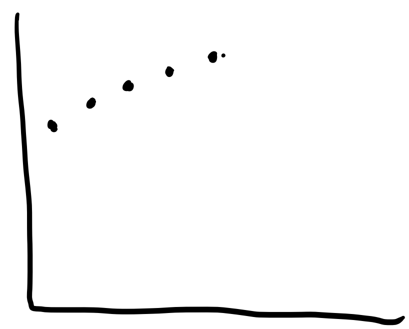
X_2:

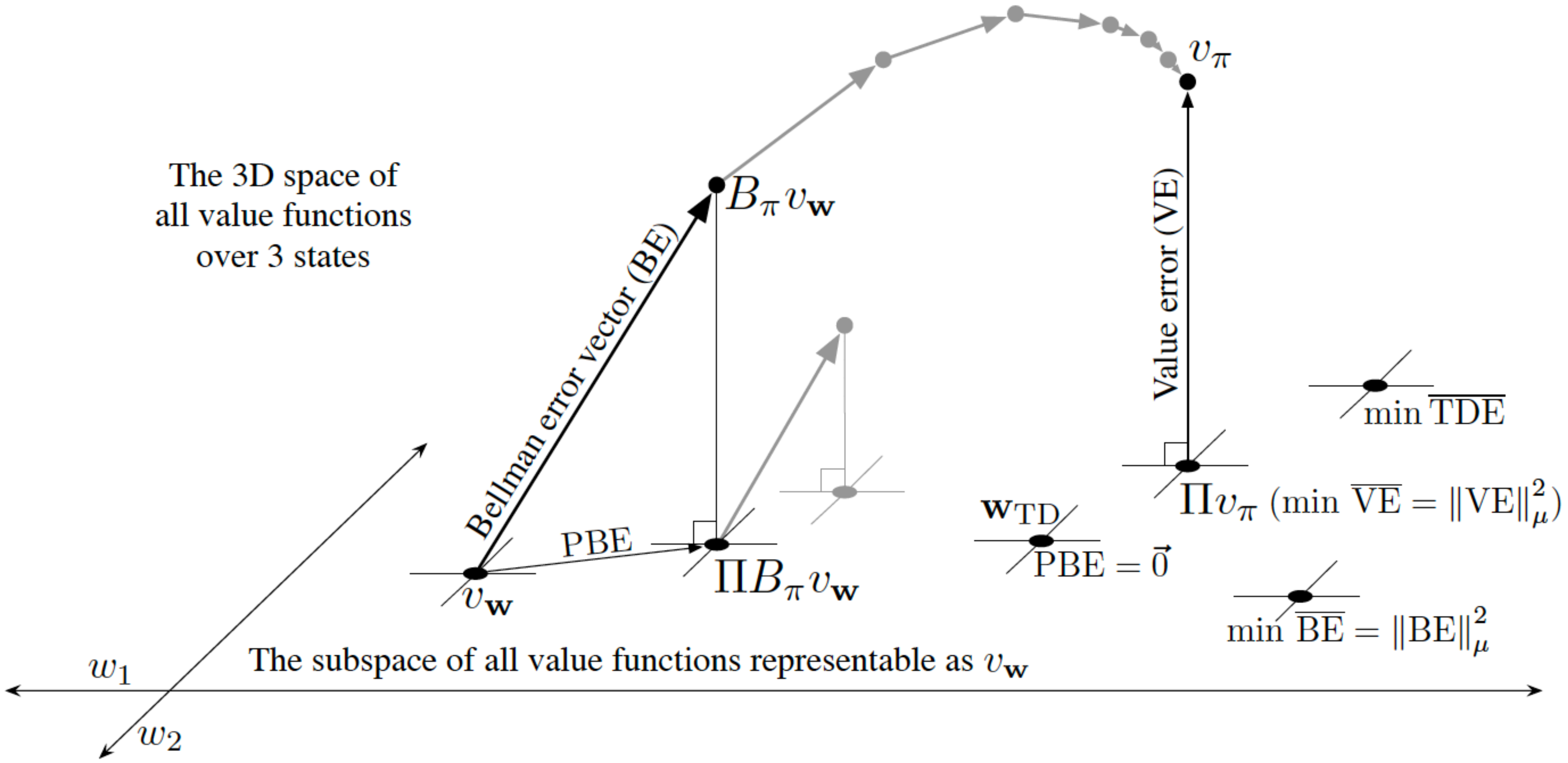


Y:



$\hat{Y}$:





The 3D space of all value functions over 3 states
v_π
$B_\pi v_{\mathbf{w}}$
Bellman error vector (BE)
Value error (VE)
PBE
$\Pi B_\pi v_{\mathbf{w}}$
$v_{\mathbf{w}}$
$\min \overline{\mathrm{TDE}}$
Πv_π $(\min \overline{\mathrm{VE}} = \|\mathrm{VE}\|_\mu^2)$
$\mathbf{w}_{\mathrm{TD}}$
$\mathrm{PBE} = \vec{0}$
$\min \overline{\mathrm{BE}} = \|\mathrm{BE}\|_\mu^2$
The subspace of all value functions representable as $v_{\mathbf{w}}$
w_1
w_2

The value function polytope (Dadashi et. al 2019)

Most expressible value functions don't belong to any policy in an MDP!

Question: Do all possible value functions (from real policies) have any structure?

Question: Is there a good VF representation that takes advantage of this structure?

The value function polytope (Dadashi et. al 2019)

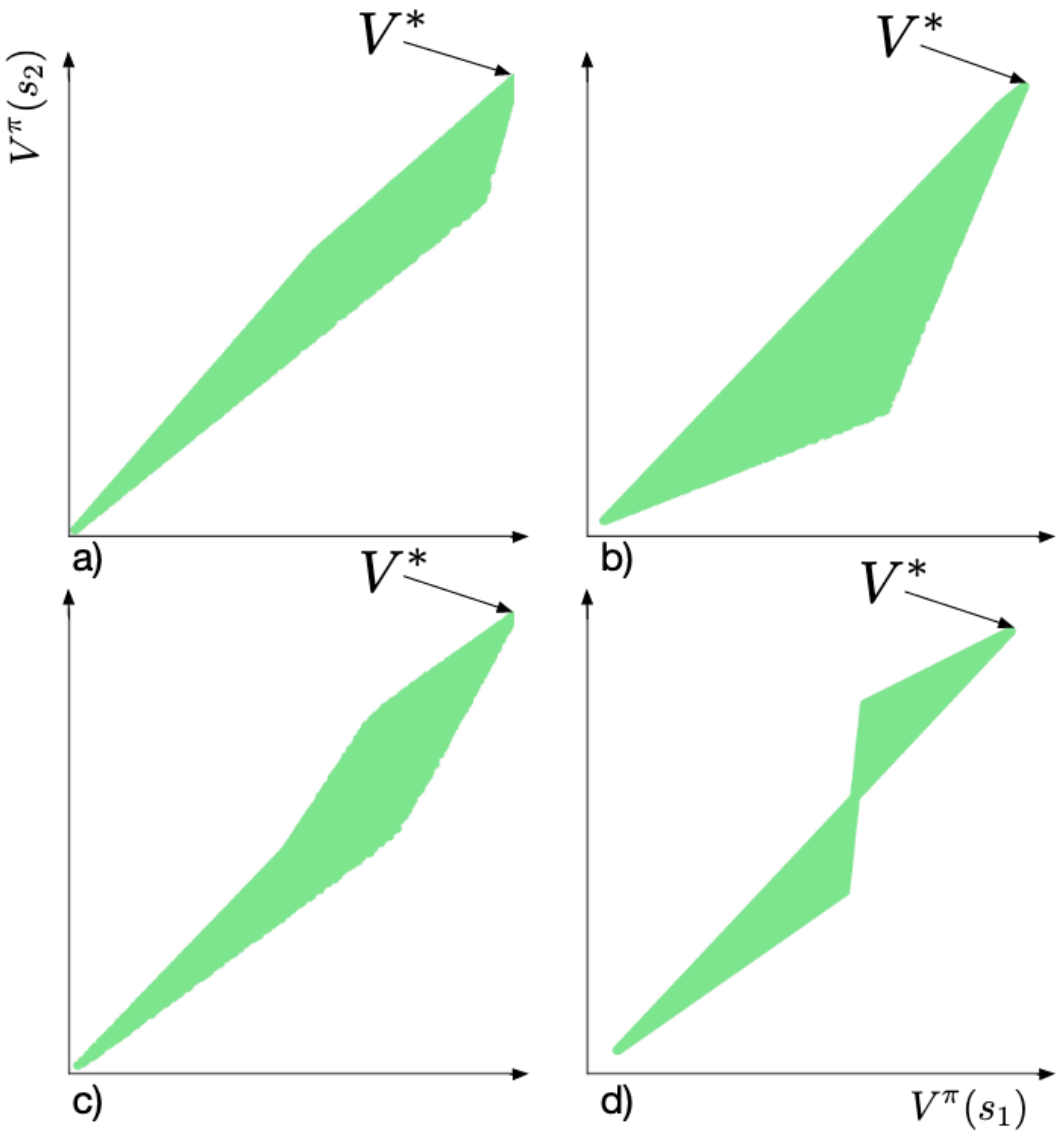


Figure 2. Space of value functions for various two-state MDPs.

The value function polytope (Dadashi et. al 2019)

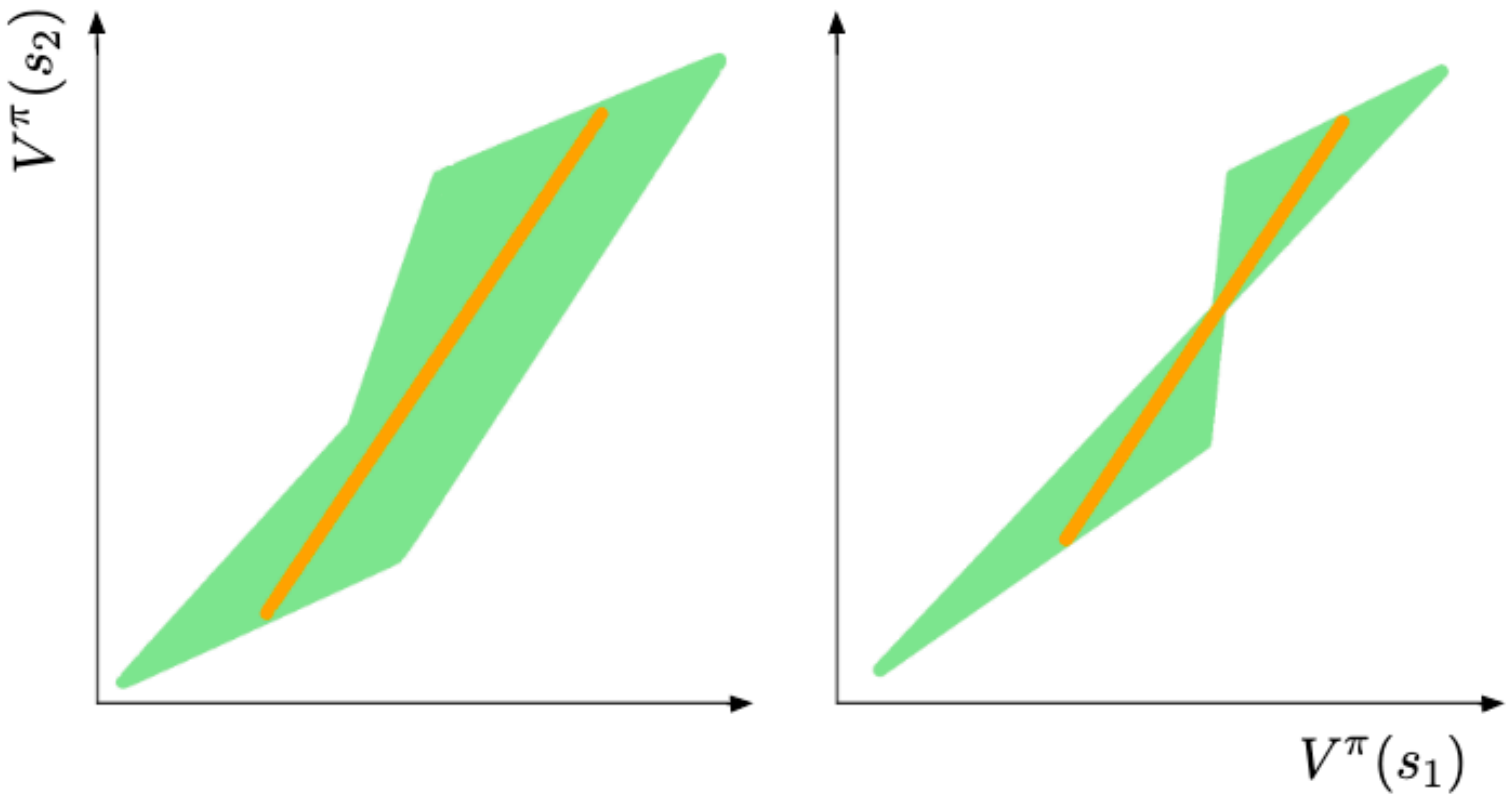


Figure 3. Illustration of Theorem 1. The orange points are the value functions of mixtures of policies that agree everywhere but one state.

The value function polytope (Dadashi et. al 2019)

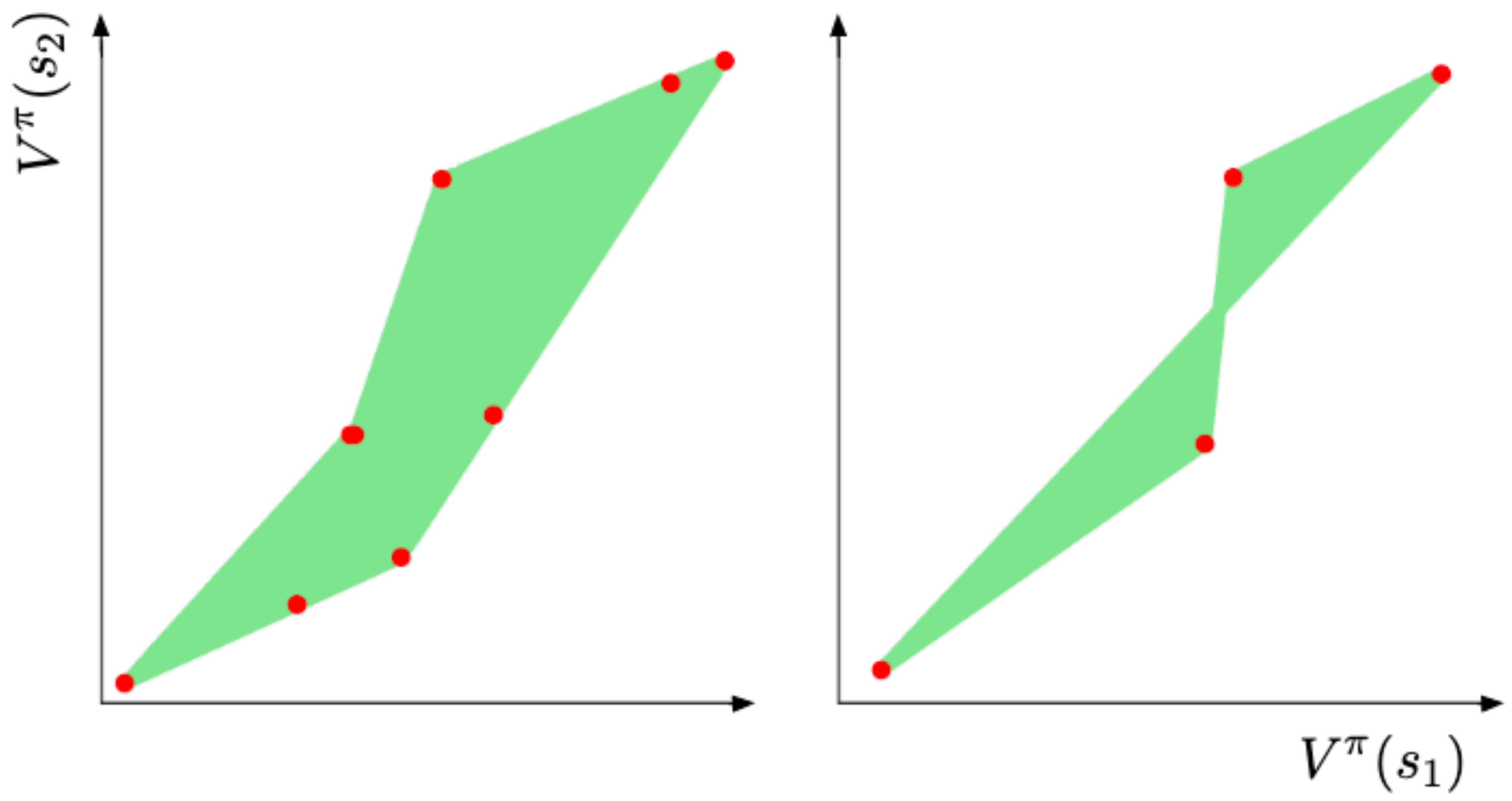


Figure 5. Visual representation of Corollary 1. The space of value functions is included in the convex hull of value functions of deterministic policies (red dots).

The value function polytope (Dadashi et. al 2019)

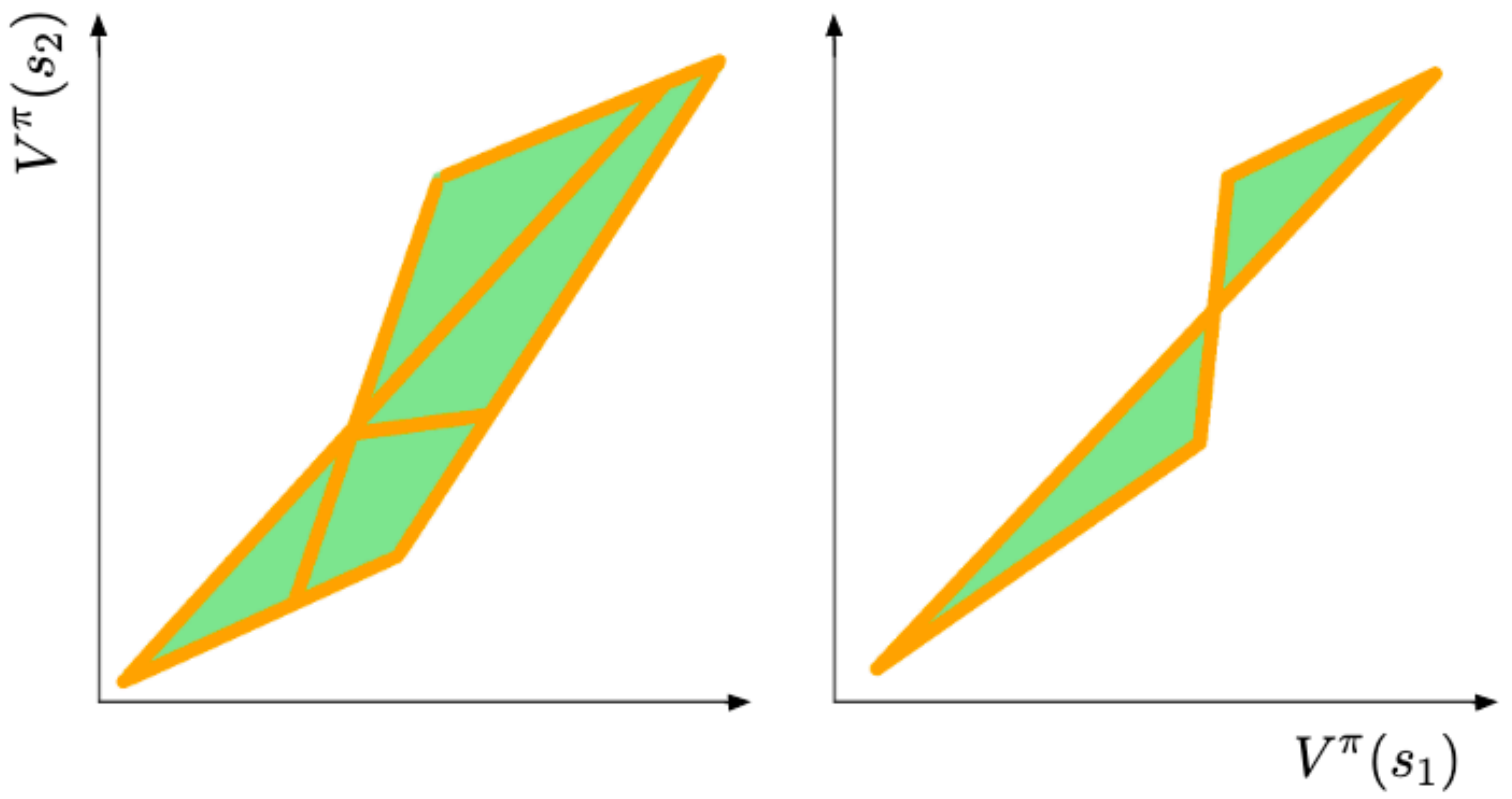


Figure 6. Visual representation of Corollary 3. The orange points are the value functions of semi-deterministic policies.

The value function polytope (Dadashi et. al 2019)

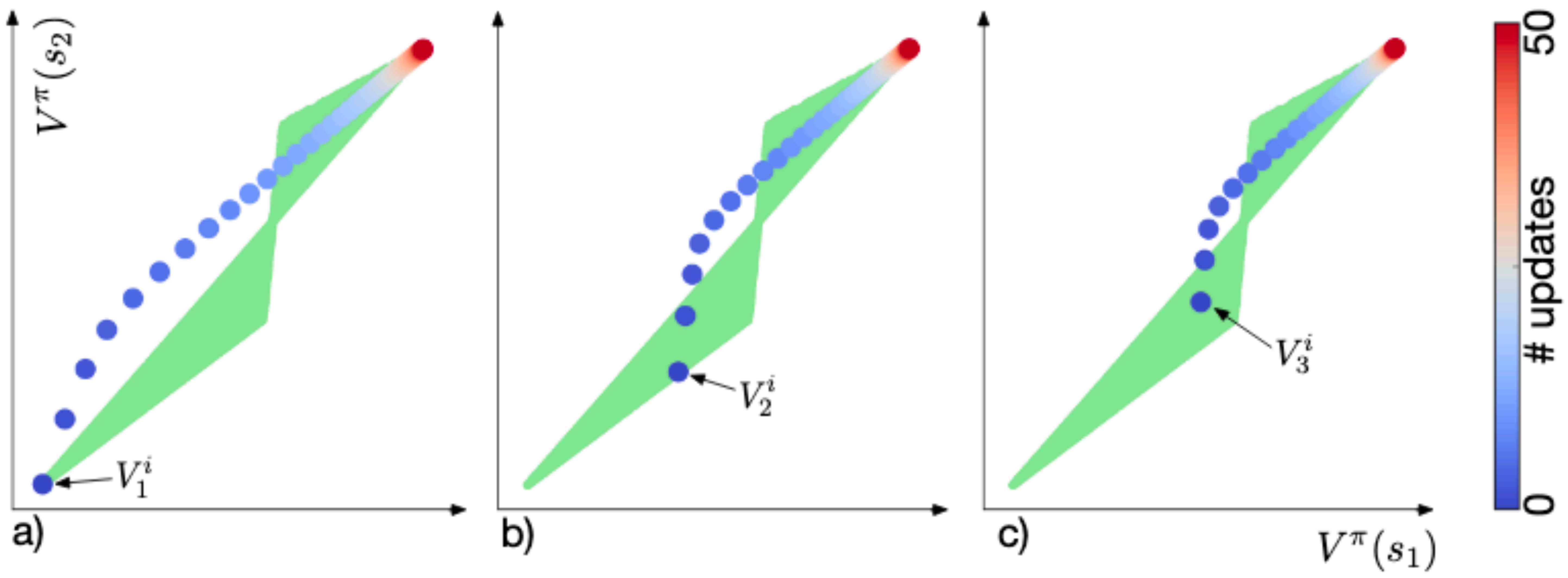


Figure 7. Value iteration dynamics for three initialization points.

The value function polytope (Dadashi et. al 2019)

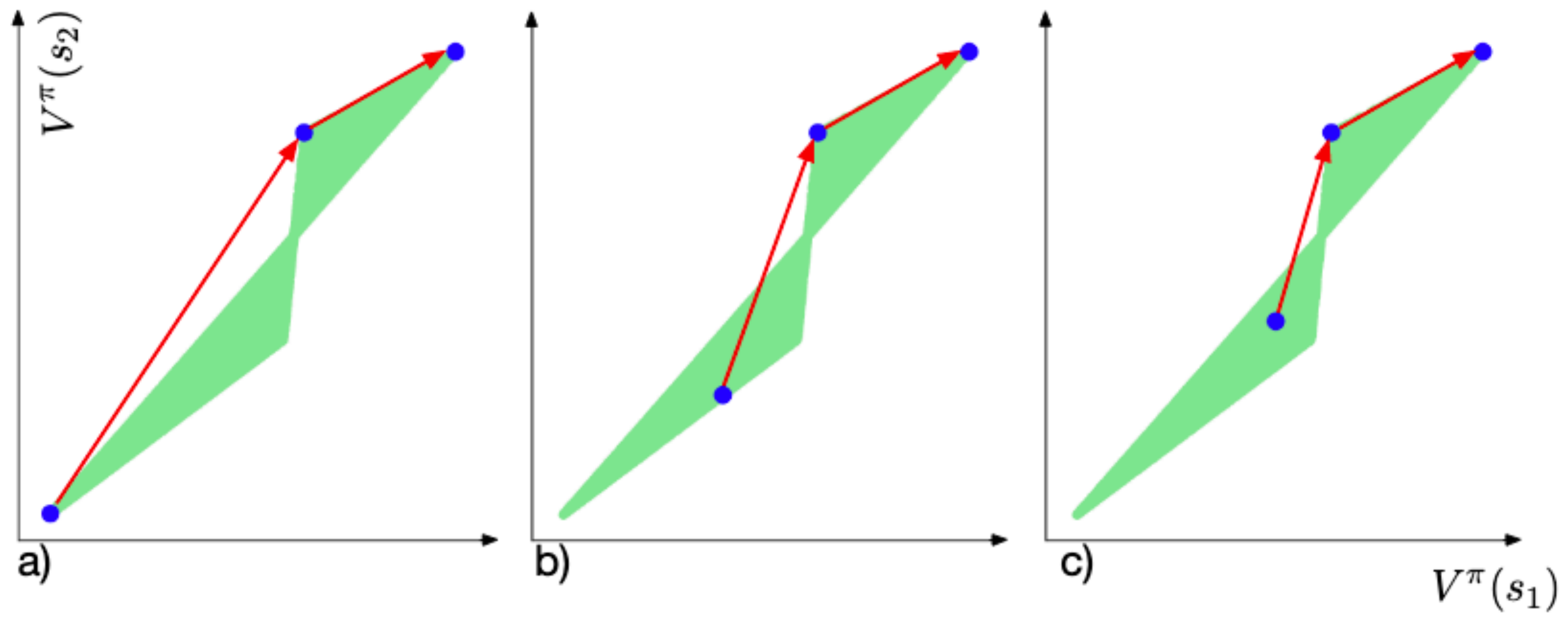


Figure 8. Policy iteration. The red arrows show the sequence of value functions (blue) generated by the algorithm.

The value function polytope (Dadashi et. al 2019)

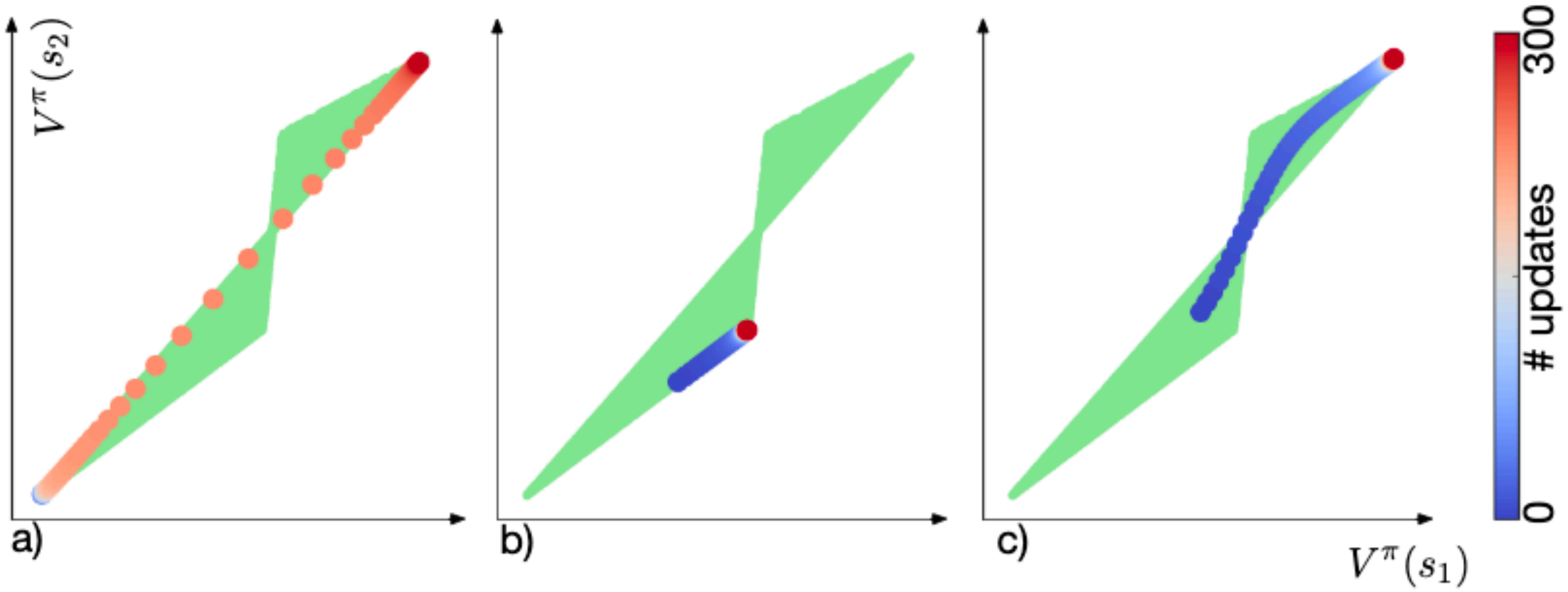


Figure 9. Value functions generated by policy gradient.

Adversarial value functions (Bellemare et. al 2019)