Stochastic Second Order Methods and Finite Time Analysis of Policy Gradient Methods

Rui Yuan

PhD Thesis Defense - 17 March 2023

Meta AI

TELECOM
Paris

INSTITUT POLYTECHNIQUE DE PARIS IP PARIS

Thank you to

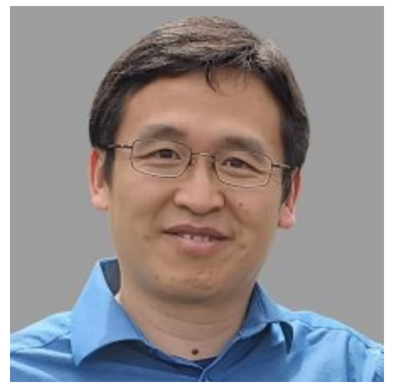
- My advisors:

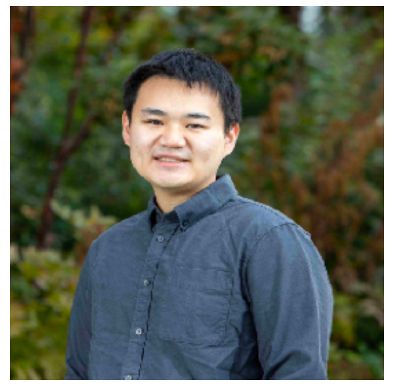
Robert M. Gower

Alessandro Lazaric

François Roueff

- My collaborators:





Outline

Outline

1. Stochastic Second Order Methods

Optimization

Outline

1. Stochastic Second Order Methods
    - A principled approach to design stochastic Newton methods
    - Convergence guarantees

Optimization

Outline

1. Stochastic Second Order Methods
    - A principled approach to design stochastic Newton methods
    - Convergence guarantees

Optimization

2. Finite Time Analysis of Policy Gradient Methods

Reinforcement Learning

Outline

1. Stochastic Second Order Methods
   - A principled approach to design stochastic Newton methods
   - Convergence guarantees

Optimization

2. Finite Time Analysis of Policy Gradient Methods
   - Vanilla policy gradient
   - Natural policy gradient

Reinforcement Learning

Outline

1. Stochastic Second Order Methods
   - A principled approach to design stochastic Newton methods
   - Convergence guarantees

Optimization

2. Finite Time Analysis of Policy Gradient Methods
   - Vanilla policy gradient
   - Natural policy gradient

Reinforcement Learning

3. Discussion & Connections to each other

— Part I —

Stochastic Second Order Methods in Optimization

Introduction (Part I)

Artificial Intelligence

Artificial Intelligence

Artificial Intelligence

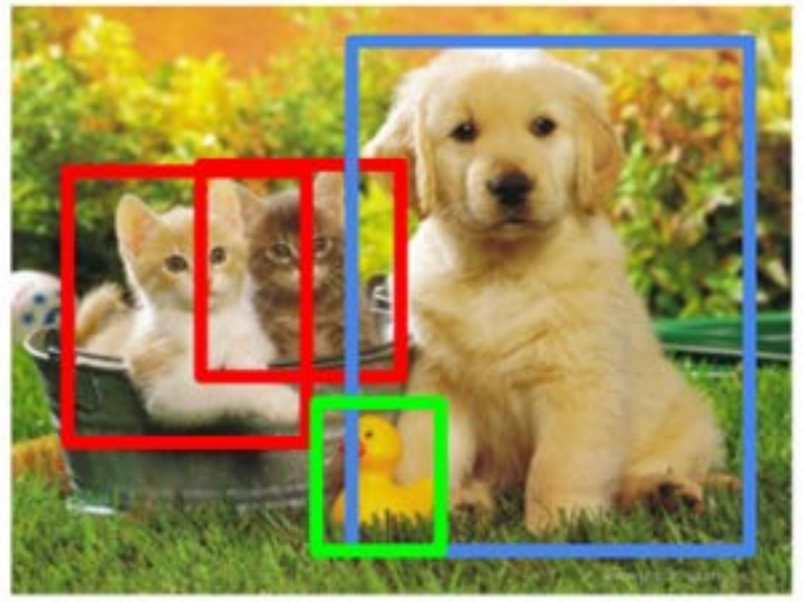
Artificial Intelligence

ChatGPT

Artificial Intelligence

ChatGPT

CAT



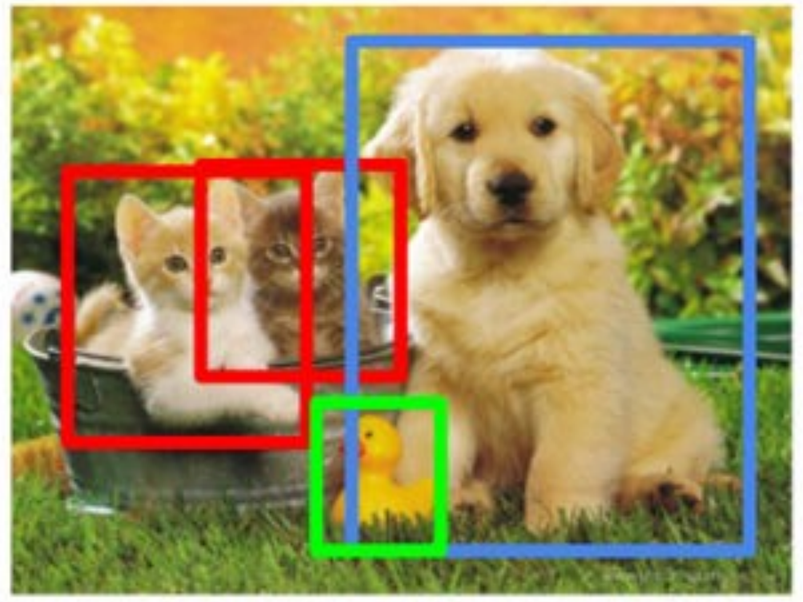
CAT, DOG, DUCK

Artificial Intelligence

ChatGPT

$\min_{w \in \mathbb{R}^d} f(w)$

CAT



CAT, DOG, DUCK

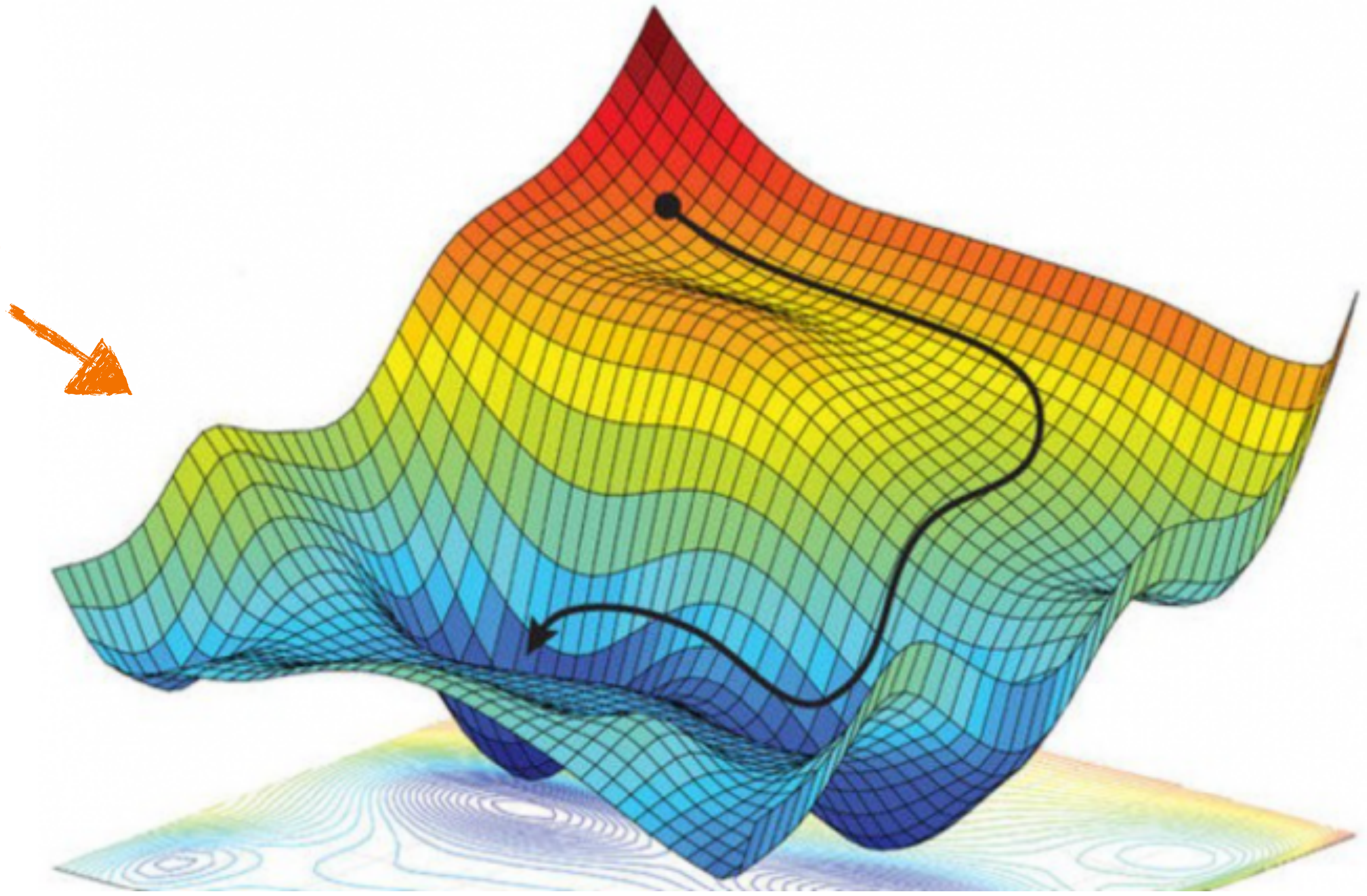
Artificial Intelligence

Optimization

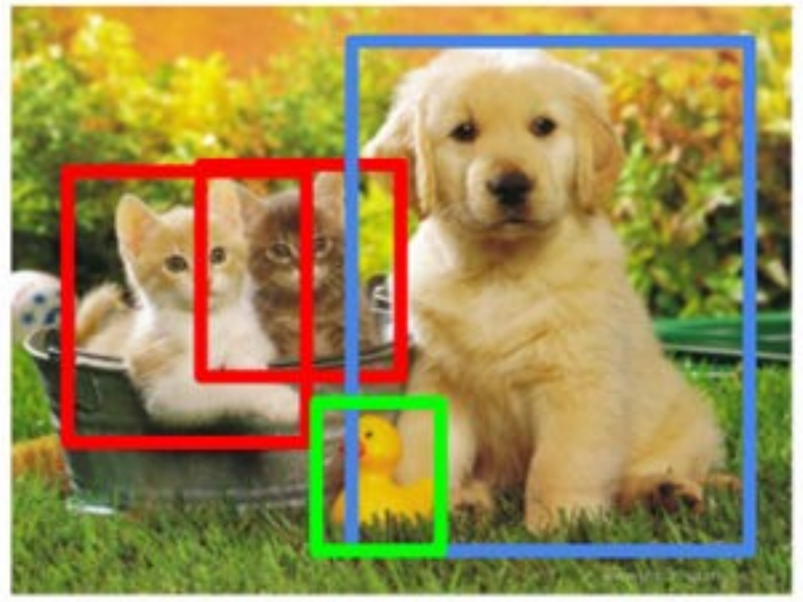
ChatGPT

$\min_{w \in \mathbb{R}^d} f(w)$ →

$f(w)$



CAT



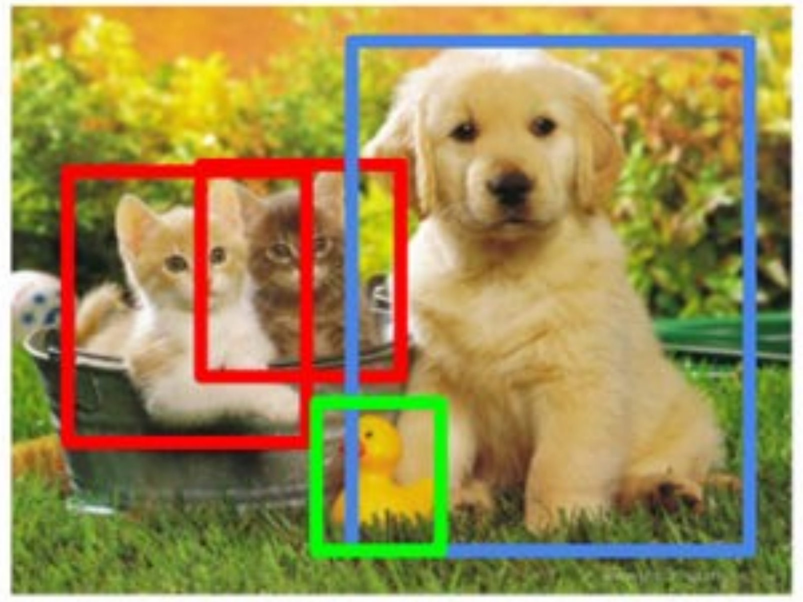
CAT, DOG, DUCK

Artificial Intelligence

Optimization

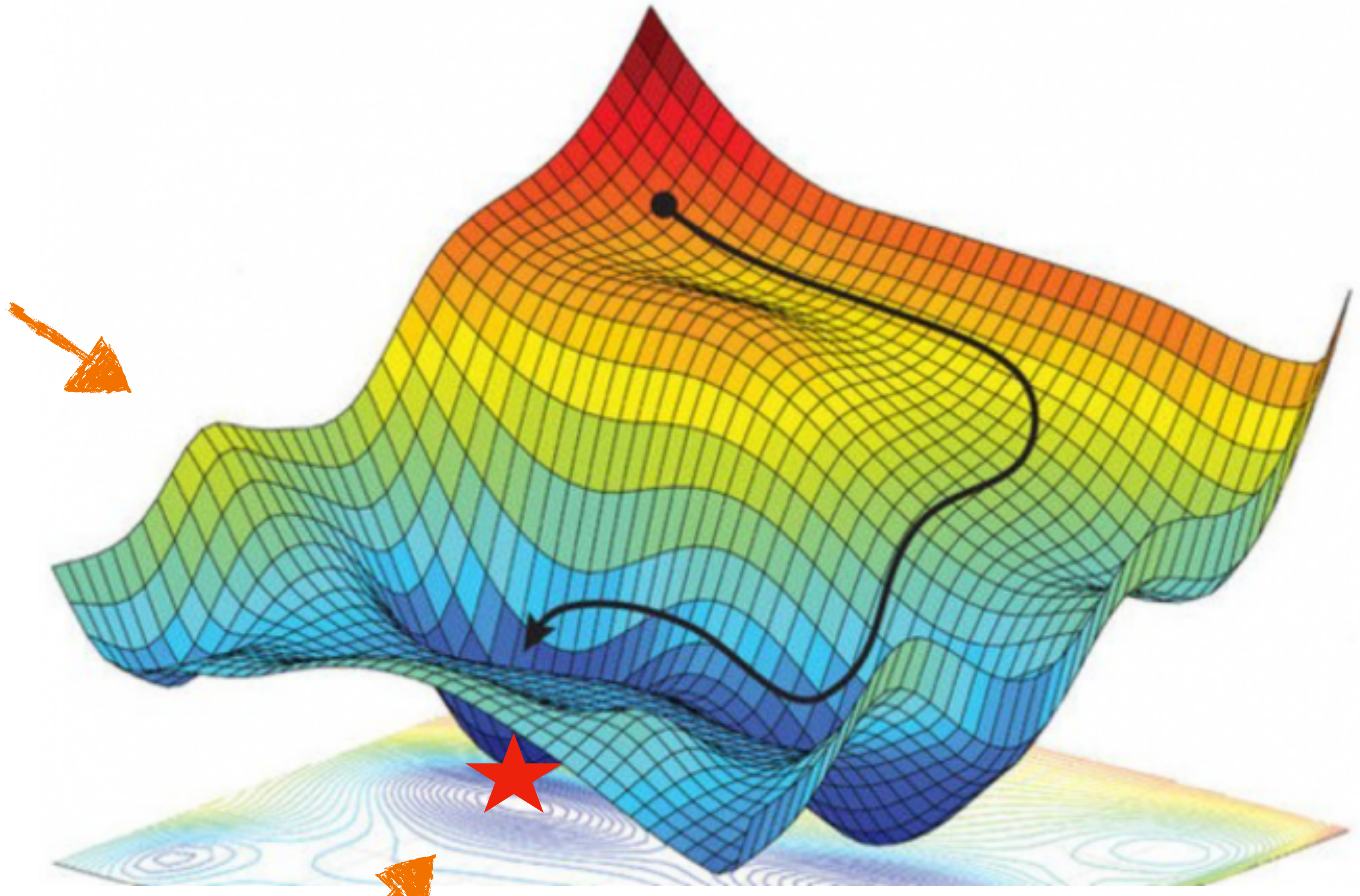
ChatGPT

CAT



CAT, DOG, DUCK

$\min_{w \in \mathbb{R}^d} f(w)$



$f(w)$

Optimal solution w^*

Gradient descent to solve $\min_{w \in \mathbb{R}^d} f(w)$

Gradient descent to solve $\min_{w \in \mathbb{R}^d} f(w)$

$$w^{k+1} = w^k - \eta^k \nabla f(w^k)$$

Gradient descent to solve $\min_{w \in \mathbb{R}^d} f(w)$

a.k.a First-order methods

$$w^{k+1} = w^k - \eta^k \nabla f(w^k)$$

Gradient descent to solve $\min_{w\in\mathbb{R}^d} f(w)$

a.k.a First-order methods

Step size /
Learning rate

$$w^{k+1} = w^k - \boxed{\eta^k}\nabla f(w^k)$$

Gradient descent to solve $\min_{w \in \mathbb{R}^d} f(w)$

a.k.a First-order methods

Step size / Learning rate

$$w^{k+1} = w^k - \boxed{\eta^k} \nabla f(w^k)$$

⚠ Step size depends on the scale of the function

Gradient descent to solve $\min_{w \in \mathbb{R}^d} f(w)$

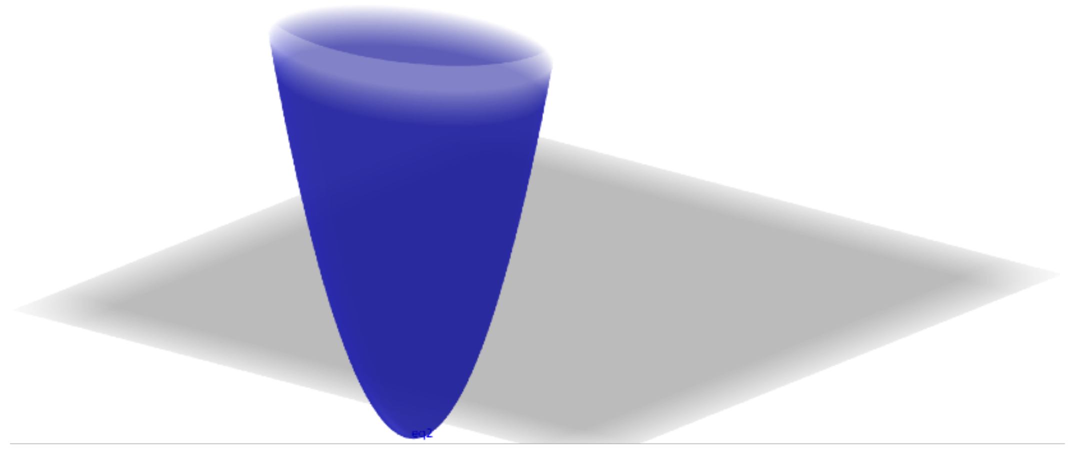
a.k.a First-order methods

Step size / Learning rate

$$w^{k+1} = w^k - \boxed{\eta^k} \nabla f(w^k)$$

⚠ Step size depends on the scale of the function

$$\arg\min_{w \in \mathbb{R}^d} f(w)$$



Gradient descent to solve $\min_{w\in\mathbb{R}^d} f(w)$

a.k.a First-order methods

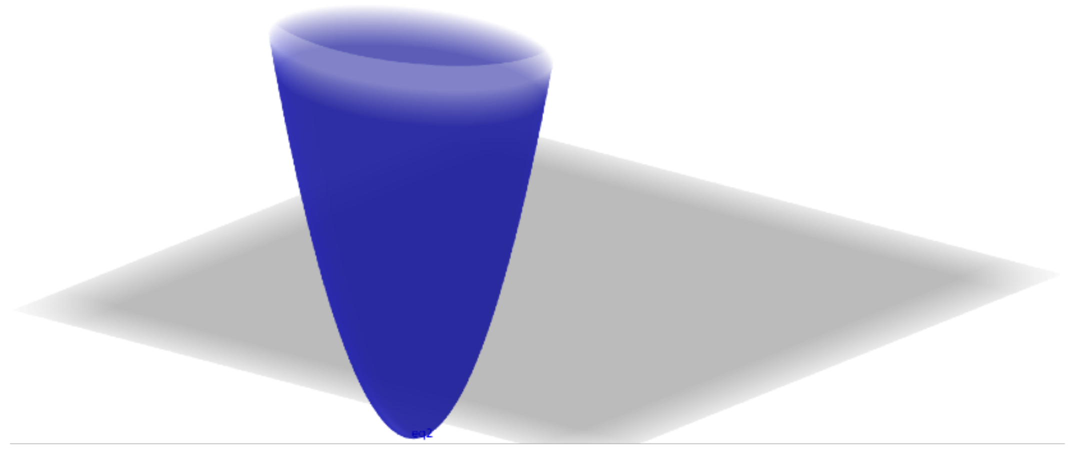
Step size / Learning rate

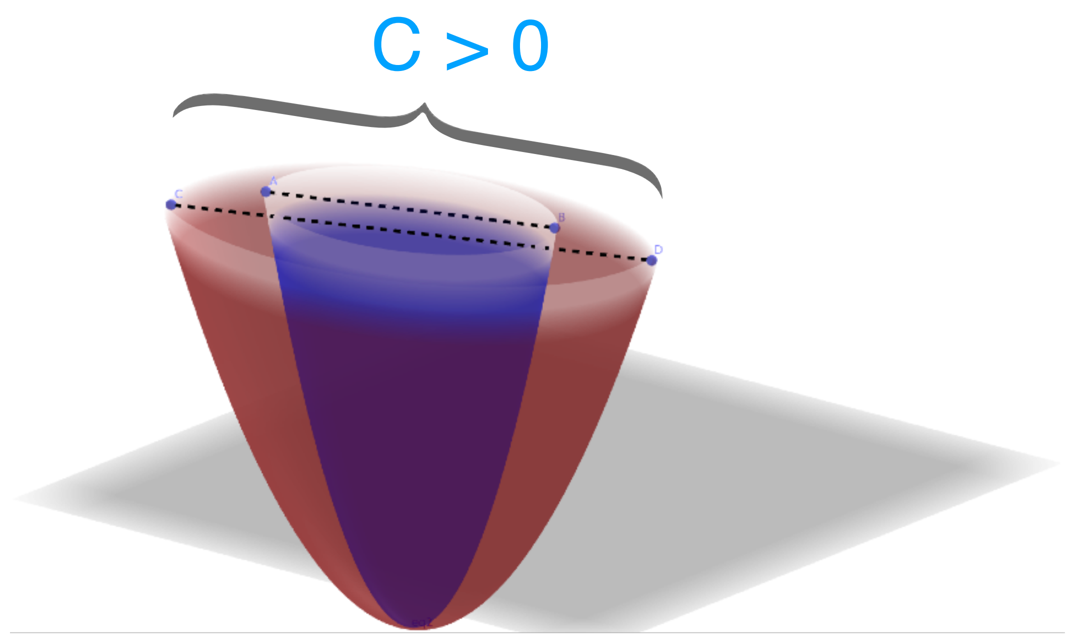
$$w^{k+1} = w^k - \boxed{\eta^k} \nabla f(w^k)$$

⚠ Step size depends on the scale of the function

C > 0

$$\arg\min_{w\in\mathbb{R}^d} f(w) \quad \Leftrightarrow \quad \arg\min_{w\in\mathbb{R}^d} C \times f(w)$$





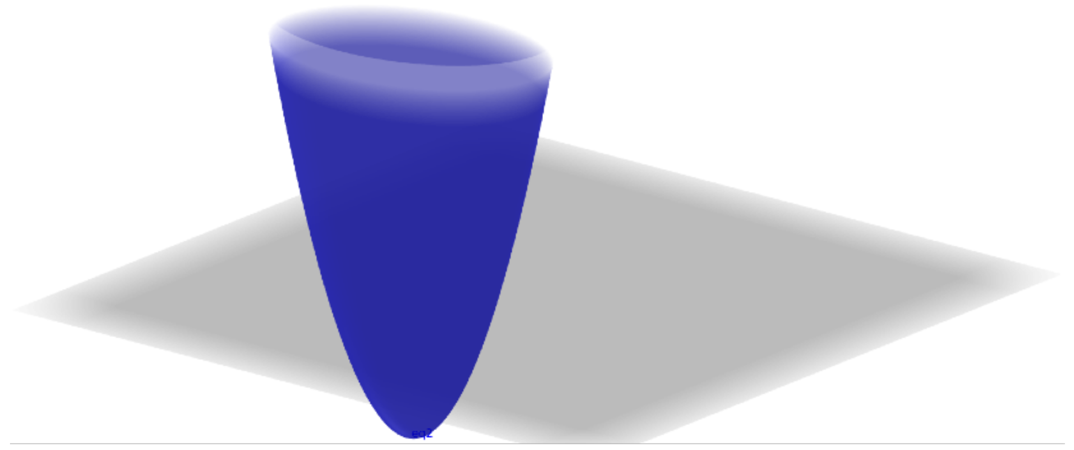
Gradient descent to solve $\min_{w\in\mathbb{R}^d} f(w)$

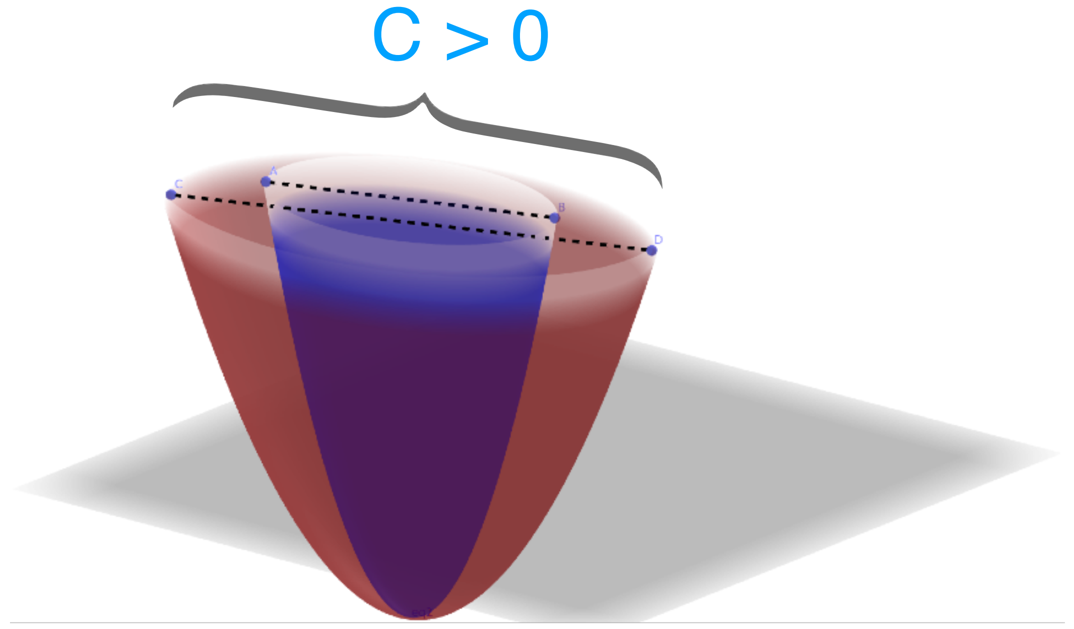
a.k.a First-order methods

$$w^{k+1} = w^k - \overset{\text{Step size / Learning rate}}{\boxed{\eta^k}} \nabla f(w^k)$$

⚠ Step size depends on the scale of the function

$$\arg\min_{w\in\mathbb{R}^d} f(w) \quad \Leftrightarrow \quad \arg\min_{w\in\mathbb{R}^d} \overset{C>0}{C} \times f(w)$$





$$w^{k+1} = w^k - \eta^k \nabla f(w^k) \quad \not\Leftrightarrow \quad w^{k+1} = w^k - \eta^k C \nabla f(w^k)$$

Gradient descent to solve $\min_{w\in\mathbb{R}^d} f(w)$

a.k.a First-order methods

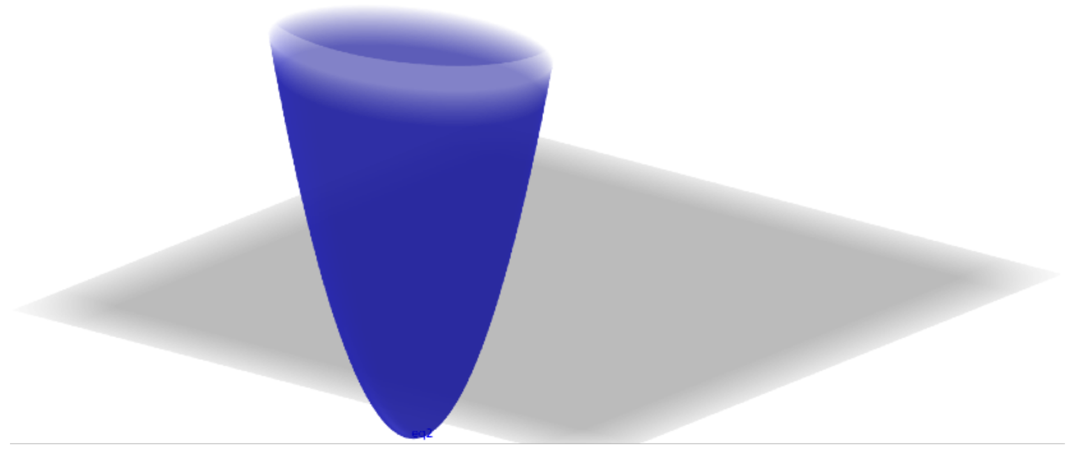
Step size / Learning rate

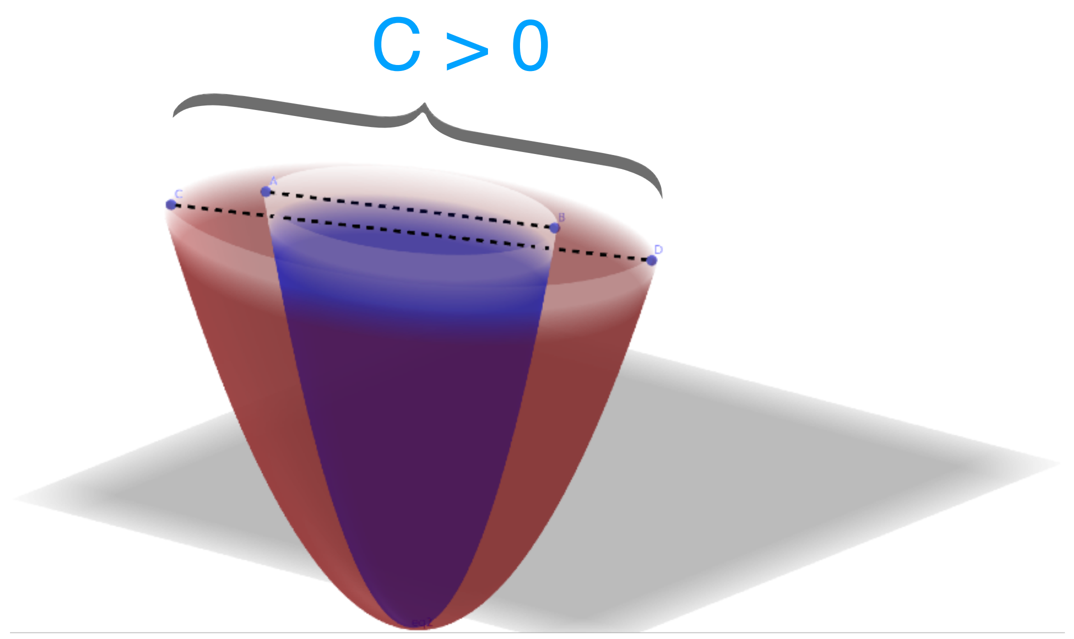
$$w^{k+1} = w^k - \boxed{\eta^k} \nabla f(w^k)$$

⚠ Step size depends on the scale of the function

C > 0

$$\underset{w\in\mathbb{R}^d}{\arg\min}\ f(w) \quad \Leftrightarrow \quad \underset{w\in\mathbb{R}^d}{\arg\min}\ C \times f(w)$$





$$w^{k+1} = w^k - \eta^k \nabla f(w^k) \quad \not\Leftrightarrow \quad w^{k+1} = w^k - \eta^k C \nabla f(w^k)$$

⚠ Hard to tune

Invariance of Newton method

Invariance of Newton method

$$w^{k+1} \;=\; w^k \;-\; \eta\,\nabla^2 f(w^k)^{-1}\,\nabla f(w^k)$$

Invariance of Newton method

a.k.a Second-order methods

$$w^{k+1} \ = \ w^k \ - \ \eta \nabla^2 f(w^k)^{-1} \nabla f(w^k)$$

Invariance of Newton method

a.k.a Second-order methods

$$w^{k+1} = w^k - \eta \nabla^2 f(w^k)^{-1} \nabla f(w^k) \quad \Leftrightarrow \quad w^{k+1} = w^k - \eta \nabla^2 (C f(w^k))^{-1} \nabla (C f(w^k))$$

💡 Scale invariant, i.e. easy to tune the step size

Invariance of Newton method

a.k.a Second-order methods

$$w^{k+1} = w^k - \eta \nabla^2 f(w^k)^{-1} \nabla f(w^k) \quad \Leftrightarrow \quad w^{k+1} = w^k - \eta \boxed{\nabla^2 (C f(w^k))^{-1}} \nabla (C f(w^k))$$

- 💡 Scale invariant, i.e. easy to tune the step size

- ⚠ Cost per iteration is $O\left(d^3\right)$ which is prohibitive when d is large

Invariance of Newton method

a.k.a Second-order methods

$$w^{k+1} = w^k - \eta \nabla^2 f(w^k)^{-1} \nabla f(w^k) \quad \Leftrightarrow \quad w^{k+1} = w^k - \eta \boxed{\nabla^2 (Cf(w^k))^{-1}} \nabla (Cf(w^k))$$

💡 Scale invariant, i.e. easy to tune the step size

⚠ Cost per iteration is $O\left(d^3\right)$ which is prohibitive when d is large

Motivations

- Less parameters tuning, e.g. step size
- Computational efficiency, as cheap as first order methods

Invariance of Newton method

a.k.a Second-order methods

$$w^{k+1} = w^k - \eta \nabla^2 f(w^k)^{-1} \nabla f(w^k) \quad \Leftrightarrow \quad w^{k+1} = w^k - \eta \boxed{\nabla^2 (Cf(w^k))^{-1}} \nabla (Cf(w^k))$$

💡 Scale invariant, i.e. easy to tune the step size

⚠ Cost per iteration is $O\left(d^3\right)$ which is prohibitive when d is large

Motivations

- Less parameters tuning, e.g. step size
- Computational efficiency, as cheap as (stochastic) first order methods

Sketched Newton-Raphson

Rui Yuan, Alessandro Lazaric, Robert M. Gower
Sketched Newton-Raphson, Society for Industrial and Applied Mathematics (SIAM) Journal on Optimization (SIOPT), 2022.

Context

Context

- Solving non linear equations $F(x) = 0$ with $F : \mathbb{R}^p \to \mathbb{R}^m$

Context

- Solving non linear equations $F(x) = 0$ with $F : \mathbb{R}^p \to \mathbb{R}^m$
- Main interest: Solving machine learning problems (e.g. generalized linear models)

Context

- Solving non linear equations $F(x) = 0$ with $F : \mathbb{R}^p \to \mathbb{R}^m$
- Main interest: Solving machine learning problems (e.g. generalized linear models)
- Newton-Raphson (NR) method

$$x^{k+1} = x^k - \eta \left(DF(x^k)^\top\right)^\dagger F(x^k)$$

Context

- Solving non linear equations $F(x) = 0$ with $F : \mathbb{R}^p \to \mathbb{R}^m$
- Main interest: Solving machine learning problems (e.g. generalized linear models)
- Newton-Raphson (NR) method

$$x^{k+1} = x^k - \eta \left(DF(x^k)^\top\right)^\dagger F(x^k)$$

$DF(x) = \left[\nabla F_1(x) \cdots \nabla F_m(x)\right] \in \mathbb{R}^{p\times m}$: transpose of the Jacobian matrix of F at x

Context

- Solving non linear equations $F(x) = 0$ with $F : \mathbb{R}^p \to \mathbb{R}^m$
- Main interest: Solving machine learning problems (e.g. generalized linear models)
- Newton-Raphson (NR) method

$$x^{k+1} = x^k - \eta \left(DF(x^k)^\top\right)^\dagger F(x^k)$$

$DF(x) = \left[\nabla F_1(x) \cdots \nabla F_m(x)\right] \in \mathbb{R}^{p \times m}$: transpose of the Jacobian matrix of F at x

$\left(DF(x^k)^\top\right)^\dagger$: Moore-Penrose pseudoinverse of $DF(x^k)^\top$

Context

- Solving non linear equations $F(x) = 0$ with $F : \mathbb{R}^p \to \mathbb{R}^m$
- Main interest: Solving machine learning problems (e.g. generalized linear models)
- Newton-Raphson (NR) method

$$x^{k+1} = x^k - \eta \boxed{\left(DF(x^k)^\top\right)^\dagger} F(x^k)$$

$DF(x) = \left[\nabla F_1(x) \cdots \nabla F_m(x)\right] \in \mathbb{R}^{p\times m}$: transpose of the Jacobian matrix of F at x

$\left(DF(x^k)^\top\right)^\dagger$: Moore-Penrose pseudoinverse of $DF(x^k)^\top$

⚠ Cost per iteration is $O\left(\min\{pm^2, mp^2\}\right)$ which is prohibitive when both p and m are large

Sketch – and – project

📗 [Gower and Richtárik, 2015]

Sketch – and – project

📗 [Gower and Richtárik, 2015]

- Newton-Raphson (NR) method

$$x^{k+1} = x^k - \eta \left(DF(x^k)^\top\right)^\dagger F(x^k)$$

Sketch – and – project

[Gower and Richtárik, 2015]

- Newton-Raphson (NR) method

$$\begin{aligned} x^{k+1} &= x^k - \eta \left(DF(x^k)^\top\right)^\dagger F(x^k) \\ &= \arg\min_{x \in \mathbb{R}^p} \|x - x^k\|_2^2 \\ &\quad \text{subject to} \quad DF(x^k)^\top (x - x^k) = -\eta F(x^k)\,. \end{aligned}$$

Sketch – and – project

[Gower and Richtárik, 2015]

- Newton-Raphson (NR) method

$$
\begin{aligned}
x^{k+1} &= x^k - \eta \left(DF(x^k)^\top\right)^\dagger F(x^k) \\
&= \arg\min_{x \in \mathbb{R}^p} \|x - x^k\|_2^2 \\
&\quad \boxed{\text{subject to} \quad DF(x^k)^\top (x - x^k) = -\,\eta F(x^k)\,.} \longrightarrow \text{Newton System}
\end{aligned}
$$

Sketch – and – project

[Gower and Richtárik, 2015]

- Newton-Raphson (NR) method

$$
\begin{aligned}
x^{k+1} &= x^k - \eta \left(DF(x^k)^\top\right)^{\dagger} F(x^k) \\
&= \arg\min_{x\in\mathbb{R}^p} \|x - x^k\|_2^2 \\
&\quad \boxed{\text{subject to} \quad DF(x^k)^\top (x - x^k) = -\,\eta F(x^k)\,.} \longrightarrow \text{Newton System}
\end{aligned}
$$

- Sketched Newton-Raphson (SNR) method

$Sketch - and - project$

[Gower and Richtárik, 2015]

- Newton-Raphson (NR) method

$$\begin{aligned} x^{k+1} &= x^k - \eta \left(DF(x^k)^\top \right)^\dagger F(x^k) \\ &= \arg\min_{x \in \mathbb{R}^p} \|x - x^k\|_2^2 \\ &\quad \boxed{\text{subject to} \quad DF(x^k)^\top (x - x^k) = -\eta F(x^k)\,.} \longrightarrow \text{Newton System} \end{aligned}$$

- Sketched Newton-Raphson (SNR) method

$$\begin{aligned} x^{k+1} &= \arg\min_{x \in \mathbb{R}^p} \|x - x^k\|_2^2 \\ &\quad \text{subject to} \quad \mathbf{S}_k^\top DF(x^k)^\top (x - x^k) = -\eta \mathbf{S}_k^\top F(x^k)\,. \end{aligned}$$

Sketch – and – project

[Gower and Richtárik, 2015]

- Newton-Raphson (NR) method

$$
\begin{aligned}
x^{k+1} &= x^k - \eta \left(DF(x^k)^\top\right)^\dagger F(x^k) \\
&= \arg\min_{x\in\mathbb{R}^p} \|x - x^k\|_2^2
\end{aligned}
$$

$$\boxed{\text{subject to} \quad DF(x^k)^\top(x - x^k) = -\eta F(x^k)\,.} \longrightarrow \text{Newton System}$$

- Sketched Newton-Raphson (SNR) method

$$x^{k+1} = \arg\min_{x\in\mathbb{R}^p} \|x - x^k\|_2^2$$

$$\boxed{\text{subject to} \quad \mathbf{S}_k^\top DF(x^k)^\top(x - x^k) = -\eta \mathbf{S}_k^\top F(x^k)\,.} \longrightarrow \text{Sketched Newton System}$$

Sketch – and – project

[Gower and Richtárik, 2015]

- Newton-Raphson (NR) method

$$
\begin{aligned}
x^{k+1} &= x^k - \eta \left(DF(x^k)^\top \right)^\dagger F(x^k) \\
&= \arg\min_{x \in \mathbb{R}^p} \|x - x^k\|_2^2 \\
&\quad \boxed{\text{subject to} \quad DF(x^k)^\top (x - x^k) = -\eta F(x^k)\,.} \longrightarrow \text{Newton System}
\end{aligned}
$$

- Sketched Newton-Raphson (SNR) method

$$
\begin{aligned}
x^{k+1} &= \arg\min_{x \in \mathbb{R}^p} \|x - x^k\|_2^2 \\
&\quad \boxed{\text{subject to} \quad \mathbf{S}_k^\top DF(x^k)^\top (x - x^k) = -\eta \mathbf{S}_k^\top F(x^k)\,.} \longrightarrow \text{Sketched Newton System}
\end{aligned}
$$

$\mathbf{S}_k \sim \mathscr{D}$: sketching matrix of size $m \times \tau$ with $\tau \ll m$, low rank

Sketch – and – project

[Gower and Richtárik, 2015]

- Newton-Raphson (NR) method

$$
\begin{aligned}
x^{k+1} &= x^k - \eta \left(DF(x^k)^\top\right)^\dagger F(x^k) \\
&= \arg\min_{x\in\mathbb{R}^p} \|x - x^k\|_2^2 \\
&\quad \boxed{\text{subject to} \quad DF(x^k)^\top(x - x^k) = -\eta F(x^k)\,.} \longrightarrow \text{Newton System}
\end{aligned}
$$

- Sketched Newton-Raphson (SNR) method

$$
\begin{aligned}
x^{k+1} &= \arg\min_{x\in\mathbb{R}^p} \|x - x^k\|_2^2 \\
&\quad \boxed{\text{subject to} \quad \mathbf{S}_k^\top DF(x^k)^\top(x - x^k) = -\eta \mathbf{S}_k^\top F(x^k)\,.} \longrightarrow \text{Sketched Newton System}
\end{aligned}
$$

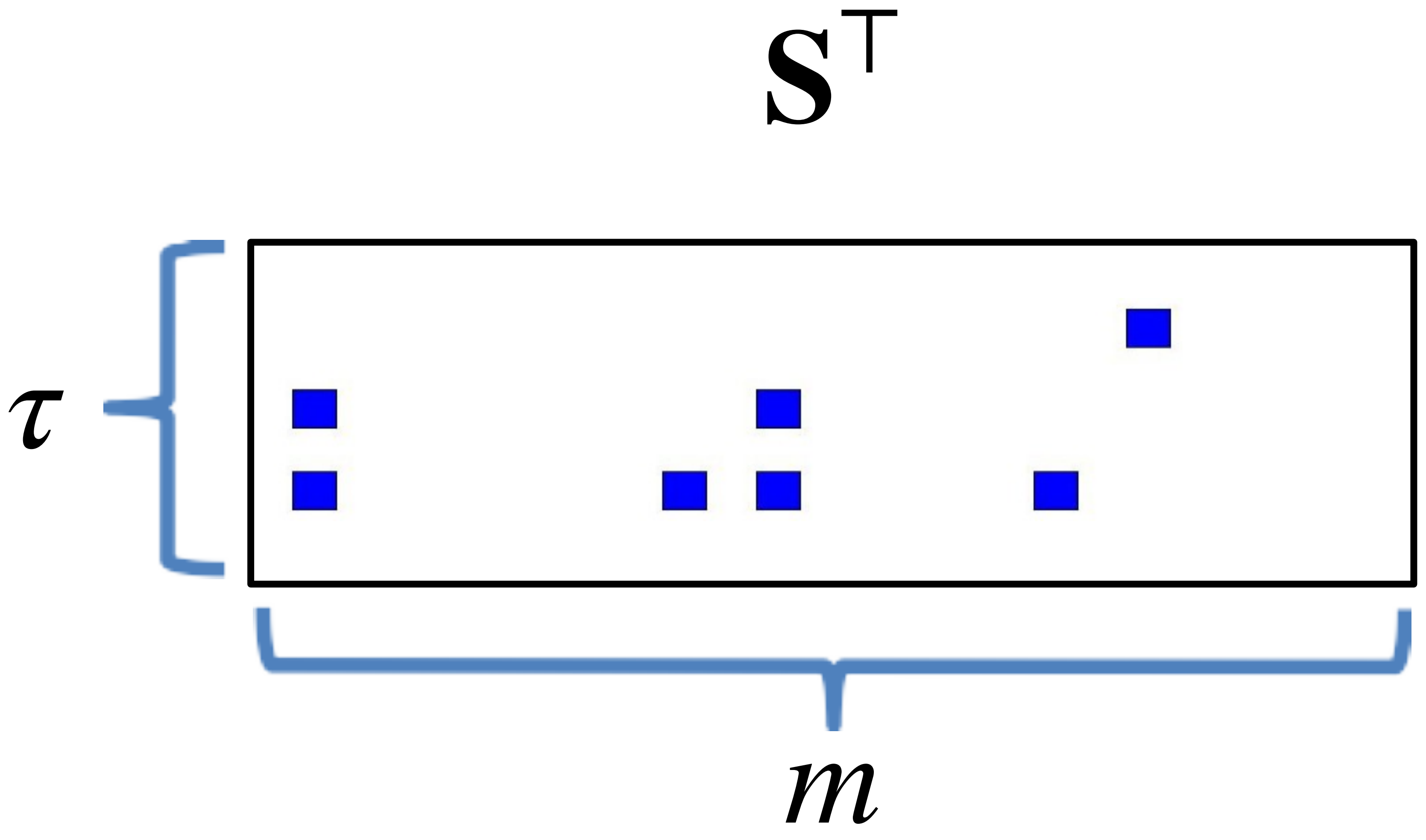
$\mathbf{S}_k \sim \mathscr{D}$: sketching matrix of size $m \times \tau$ with $\tau \ll m$, low rank

Cost per iteration $O(p)$

Decrease dimension using sketching

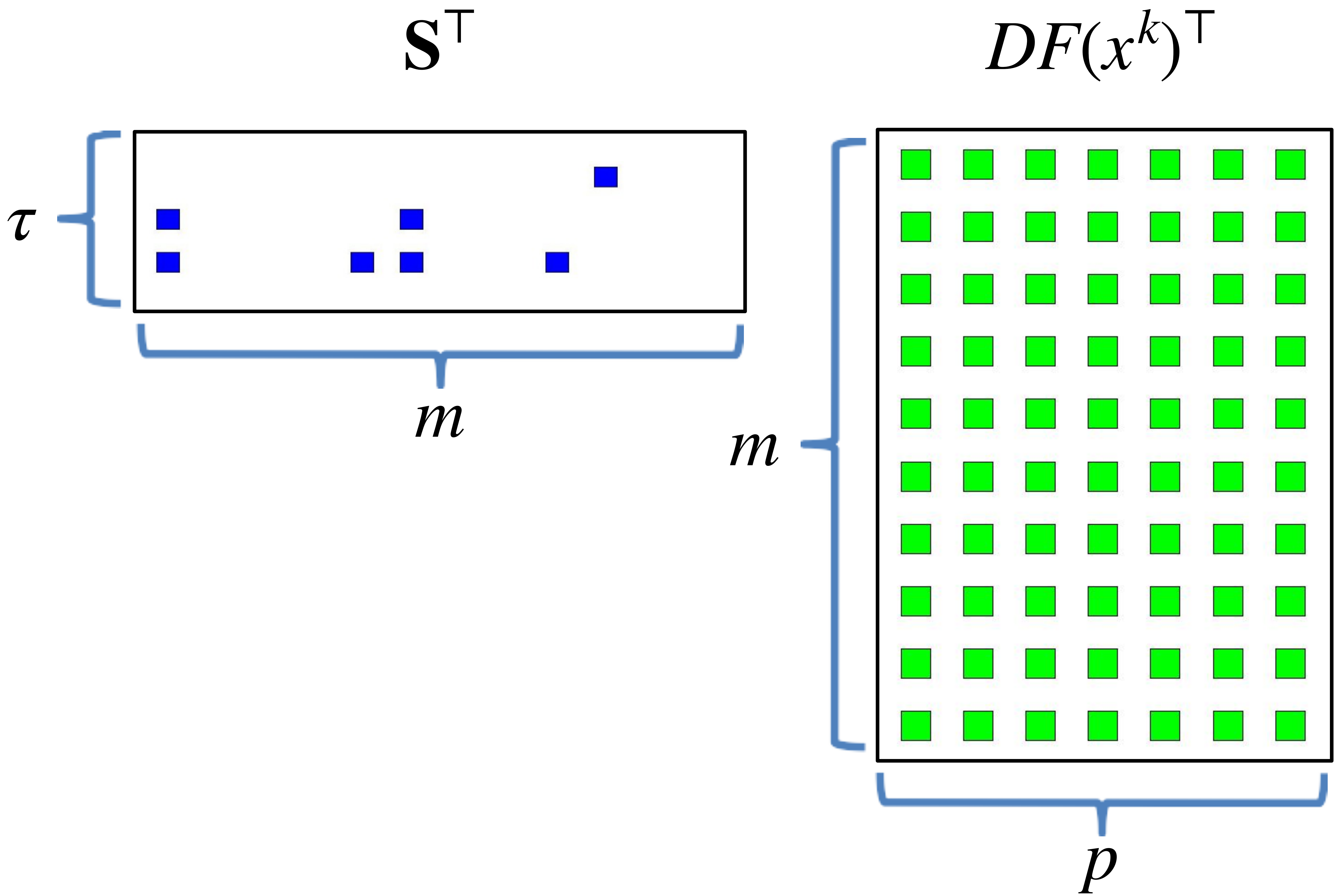
Decrease dimension using sketching

The sketching matrix $\mathbf{S} \sim \mathcal{D}$ a distribution over $\mathbf{S} \in \mathbb{R}^{m \times \tau}$ and $\tau \ll m$



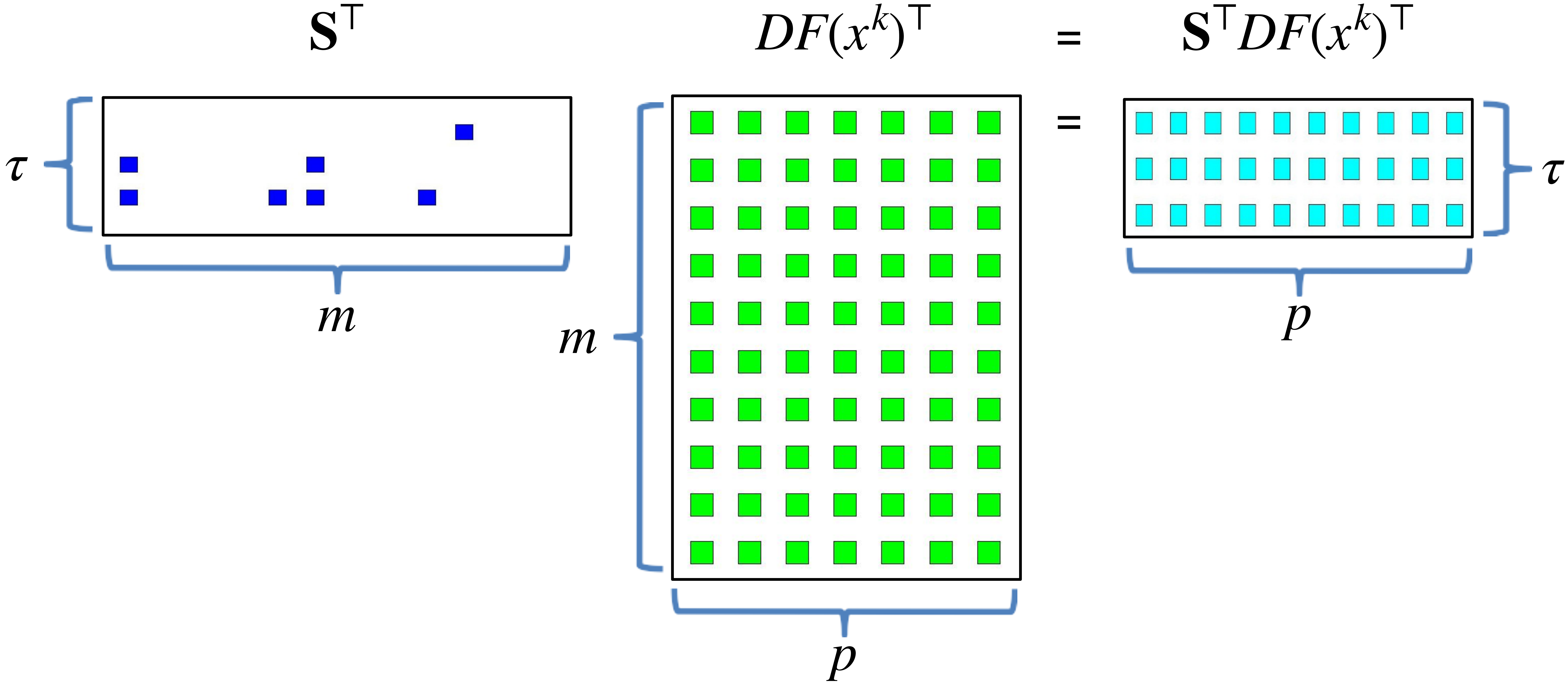
Decrease dimension using sketching

The sketching matrix $\mathbf{S} \sim \mathscr{D}$ a distribution over $\mathbf{S} \in \mathbb{R}^{m \times \tau}$ and $\tau \ll m$



Decrease dimension using sketching

The sketching matrix $\mathbf{S} \sim \mathscr{D}$ a distribution over $\mathbf{S} \in \mathbb{R}^{m \times \tau}$ and $\tau \ll m$



Simple examples of sketches

Simple examples of sketches

- Sample

$$\mathbf{S} \;=\; \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} = e_j \qquad \Longrightarrow \qquad \mathbf{S}^\top DF(x)^\top \;=\; \nabla F_j(x)^\top$$

Simple examples of sketches

- Sample $\mathbf{S} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} = e_j \quad \Longrightarrow \quad \mathbf{S}^\top DF(x)^\top = \nabla F_j(x)^\top$

- Average sample $\mathbf{S} = \begin{bmatrix} a_1 \\ 0 \\ a_3 \\ a_4 \end{bmatrix} = \sum_{i \in I} a_i e_i \quad \Longrightarrow \quad \mathbf{S}^\top DF(x)^\top = \sum_{i \in I} a_i \nabla F_i(x)^\top$

Simple examples of sketches

- Sample $$\mathbf{S} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} = e_j \quad \Longrightarrow \quad \mathbf{S}^\top DF(x)^\top = \nabla F_j(x)^\top$$

- Average sample $$\mathbf{S} = \begin{bmatrix} a_1 \\ 0 \\ a_3 \\ a_4 \end{bmatrix} = \sum_{i \in I} a_i e_i \quad \Longrightarrow \quad \mathbf{S}^\top DF(x)^\top = \sum_{i \in I} a_i \nabla F_i(x)^\top$$

- Batch sample $$\mathbf{S} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} e_i & e_j & e_k \end{bmatrix} \quad \Longrightarrow \quad \mathbf{S}^\top DF(x)^\top = \begin{bmatrix} \nabla F_i(x)^\top \\ \nabla F_j(x)^\top \\ \nabla F_k(x)^\top \end{bmatrix} \in \mathbb{R}^{\tau \times p}$$

Sketched Newton-Raphson (SNR)

$$x^{k+1} = \arg\min_{x \in \mathbb{R}^p} \|x - x^k\|_2^2$$

$$\text{subject to} \quad \mathbf{S}_k^\top DF(x^k)^\top (x - x^k) = -\eta \mathbf{S}_k^\top F(x^k)\,.$$

Sketched Newton-Raphson (SNR)

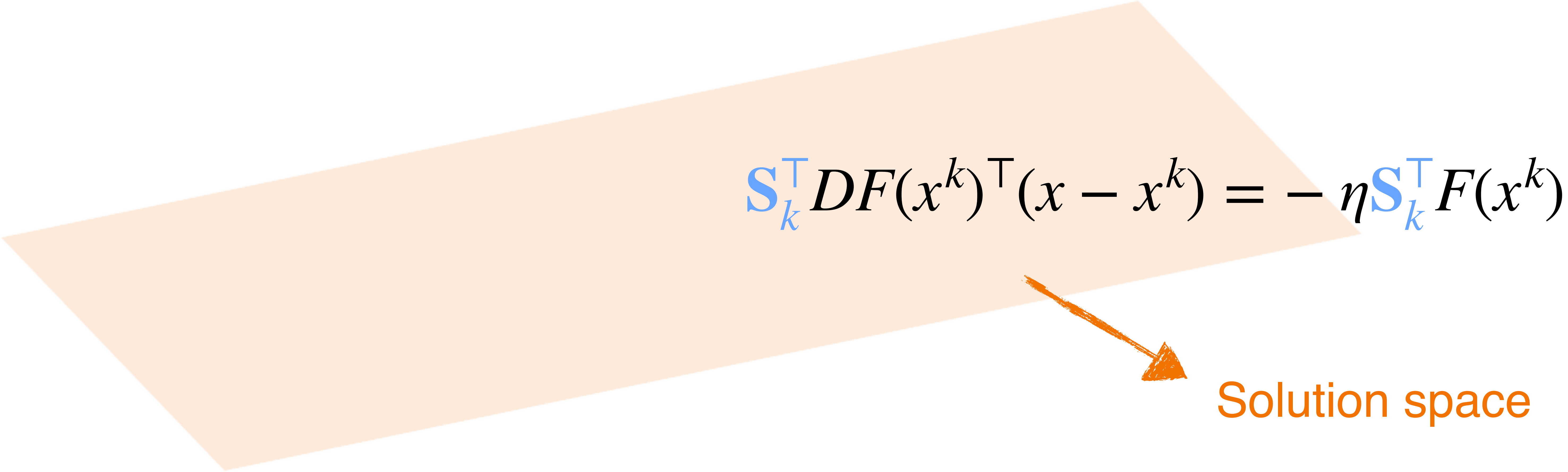
$$x^{k+1} = \arg\min_{x \in \mathbb{R}^p} \|x - x^k\|_2^2$$

$$\boxed{\text{subject to} \quad \mathbf{S}_k^\top DF(x^k)^\top (x - x^k) = -\eta \mathbf{S}_k^\top F(x^k)\,.}$$

Sketched Newton-Raphson (SNR)

$$x^{k+1} = \arg\min_{x \in \mathbb{R}^p} \|x - x^k\|_2^2$$

$$\boxed{\text{subject to} \quad \mathbf{S}_k^\top DF(x^k)^\top (x - x^k) = -\eta \mathbf{S}_k^\top F(x^k)\,.}$$



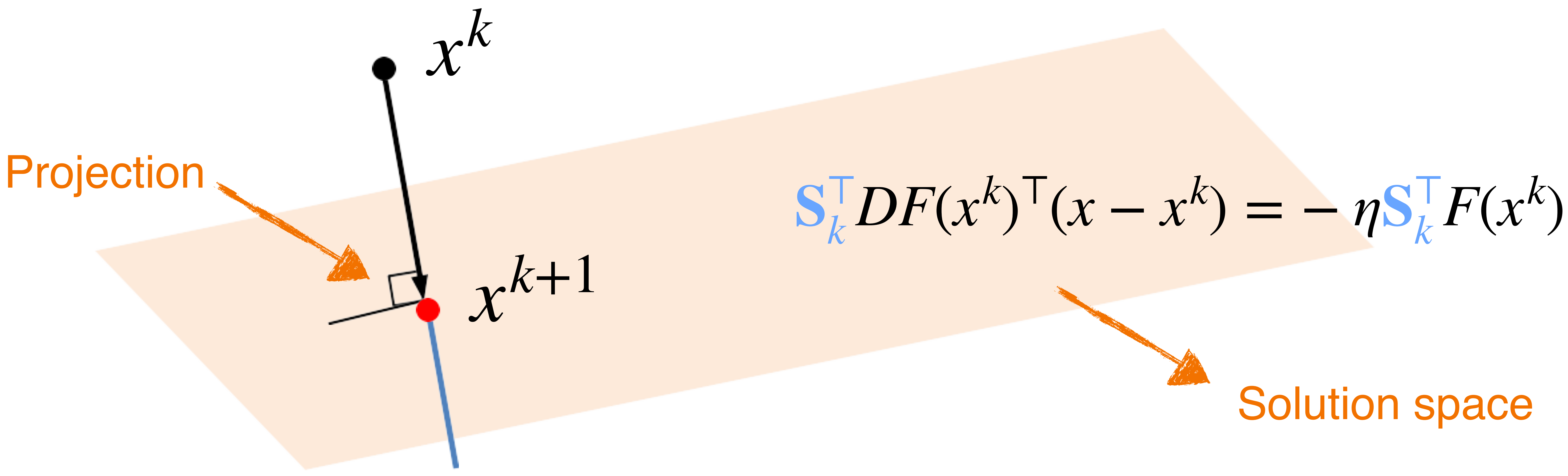
$\mathbf{S}_k^\top DF(x^k)^\top (x - x^k) = -\eta \mathbf{S}_k^\top F(x^k)$

Solution space

Sketched Newton-Raphson (SNR)

$$x^{k+1} = \boxed{\arg\min_{x\in\mathbb{R}^p} \|x - x^k\|_2^2}$$

subject to $\mathbf{S}_k^\top DF(x^k)^\top (x - x^k) = -\eta \mathbf{S}_k^\top F(x^k)$.


x^k
Projection
x^{k+1}
$\mathbf{S}_k^\top DF(x^k)^\top (x - x^k) = -\eta \mathbf{S}_k^\top F(x^k)$
Solution space

Convergence theories of SNR

(see paper for technique details and additional properties)

Convergence theories of SNR

(see paper for technique details and additional properties)

- Reformulation as online stochastic gradient descent (SGD)

Convergence theories of SNR

(see paper for technique details and additional properties)

- Reformulation as online stochastic gradient descent (SGD)
- The reformulation has a gratuitous smoothness property

Convergence theories of SNR

(see paper for technique details and additional properties)

- Reformulation as online stochastic gradient descent (SGD)
- The reformulation has a gratuitous smoothness property
- The reformulation has a gratuitous interpolation condition, i.e. zero noise for stochastic gradient at the optimum

Convergence theories of SNR

(see paper for technique details and additional properties)

- Reformulation as online stochastic gradient descent (SGD)
- The reformulation has a gratuitous smoothness property
- The reformulation has a gratuitous interpolation condition, i.e. zero noise for stochastic gradient at the optimum
- Global convergence theory and rates of convergence guaranteed under convex type assumptions

Applications

(see paper for additional applications)

Applications

(see paper for additional applications)

- When $\mathbf{S}_k = \mathbf{I}_m$, i.e. no sketch, new global convergence theory for the original Newton-Raphson method under strictly weaker assumptions

Applications

(see paper for additional applications)

- When $\mathbf{S}_k = \mathbf{I}_m$, i.e. no sketch, new global convergence theory for the original Newton-Raphson method under strictly weaker assumptions
- When $\mathbf{S}_k = e_i$, i.e., single row sampling, new nonlinear Kaczmarz method

Applications

(see paper for additional applications)

- When $\mathbf{S}_k = \mathbf{I}_m$, i.e. no sketch, new global convergence theory for the original Newton-Raphson method under strictly weaker assumptions
- When $\mathbf{S}_k = e_i$, i.e., single row sampling, new nonlinear Kaczmarz method
- Recover the stochastic Newton method [Rodomanov and Kropotov, 2016; Kovalev et al., 2019] (First global convergence theory)

Applications

(see paper for additional applications)

- When $\mathbf{S}_k = \mathbf{I}_m$, i.e. no sketch, new global convergence theory for the original Newton-Raphson method under strictly weaker assumptions

- When $\mathbf{S}_k = e_i$, i.e., single row sampling, new nonlinear Kaczmarz method

- Recover the stochastic Newton method [Rodomanov and Kropotov, 2016; Kovalev et al., 2019] (First global convergence theory)

- New method for solving generalized linear models (GLM)

Generalized linear models (GLMs)

Generalized linear models (GLMs)

- Generalized linear models

$$\min_{w\in\mathbb{R}^d}\left[f(w) := \frac{1}{n}\sum\nolimits_{i=1}^{n}\phi_i(a_i^\top w) + \frac{\lambda}{2}\|w\|^2\right]$$

Generalized linear models (GLMs)

- Generalized linear models

Training problem ←

$$\min_{w \in \mathbb{R}^d} \left[f(w) := \frac{1}{n} \sum_{i=1}^{n} \phi_i(a_i^\top w) + \frac{\lambda}{2} \|w\|^2 \right]$$

Generalized linear models (GLMs)

- Generalized linear models

n := Number of samples

Training problem

$$\min_{w \in \mathbb{R}^d} \left[f(w) := \frac{1}{\boxed{n}} \sum_{i=1}^{n} \phi_i(a_i^\top w) + \frac{\lambda}{2} \|w\|^2 \right]$$

Generalized linear models (GLMs)

n := Number of samples

a_i := The ith sample of the dataset

- Generalized linear models

Training problem

$$\min_{w\in\mathbb{R}^d}\left[f(w) := \frac{1}{\boxed{n}}\sum_{i=1}^{n}\phi_i(\boxed{a_i}^\top w) + \frac{\lambda}{2}\|w\|^2\right]$$

Generalized linear models (GLMs)

n := Number of samples

a_i := The ith sample of the dataset

- Generalized linear models

Training problem ←

$$\min_{w \in \mathbb{R}^d} \left[f(w) := \frac{1}{\boxed{n}} \sum_{i=1}^{n} \boxed{\phi_i(\boxed{a_i}^\top w)} + \frac{\lambda}{2} \|w\|^2 \right]$$

ϕ_i := The loss over the ith batch of data

Generalized linear models (GLMs)

- Generalized linear models

n := Number of samples

a_i := The ith sample of the dataset

Training problem

$$\min_{w\in\mathbb{R}^d}\left[f(w) := \frac{1}{\boxed{n}}\sum_{i=1}^{n} \boxed{\phi_i(\boxed{a_i}^\top w)} + \boxed{\frac{\lambda}{2}\|w\|^2}\right]$$

Regularization on w

ϕ_i := The loss over the ith batch of data

Generalized linear models (GLMs)

- Generalized linear models

n := Number of samples

a_i := The ith sample of the dataset

Training problem ←

$$\min_{w\in\mathbb{R}^d}\left[f(w) := \frac{1}{\boxed{n}}\sum_{i=1}^{n} \boxed{\phi_i(\boxed{a_i}^\top w)} + \boxed{\frac{\lambda}{2}\|w\|^2}\right]$$

→ Regularization on w

ϕ_i := The loss over the ith batch of data

- We want to solve $\nabla f(w) = 0$

$$\nabla f(w) = \frac{1}{n}\sum_{i=1}^{n} \phi_i'(a_i^\top w)a_i + \lambda w = 0$$

Tossing-coin-sketch (TCS) for solving GLMs

Objective: $\nabla f(w) = \frac{1}{n} \sum_{i=1}^{n} \phi_i'(a_i^\top w) a_i + \lambda w = 0$

Tossing-coin-sketch (TCS) for solving GLMs

Objective: $\nabla f(w) = \frac{1}{n} \sum_{i=1}^{n} \underbrace{\phi_i'(a_i^\top w)}_{-\alpha_i} a_i + \lambda w = 0$

Tossing-coin-sketch (TCS) for solving GLMs

Objective: $\nabla f(w) = \frac{1}{n} \sum_{i=1}^{n} \underbrace{\phi_i'(a_i^\top w)}_{-\alpha_i} a_i + \lambda w = 0$

- Fixed point equations

$$\alpha_i = -\phi_i'(a_i^\top w), \quad \text{for } i = 1, \ldots, n,$$
$$w = \frac{1}{\lambda n} A \alpha \in \mathbb{R}^d.$$

Tossing-coin-sketch (TCS) for solving GLMs

Objective: $\nabla f(w) = \frac{1}{n}\sum_{i=1}^{n} \underbrace{\phi_i'(a_i^\top w)}_{-\alpha_i} a_i + \lambda w = 0$

- Fixed point equations

$$\alpha_i = -\phi_i'(a_i^\top w), \quad \text{for } i = 1,\ldots,n,$$

$$w = \frac{1}{\lambda n} A\alpha \in \mathbb{R}^d.$$

$$\begin{cases} \mathbf{A} := [a_1 \ \cdots \ a_n] \in \mathbb{R}^{d\times n} \\ \alpha := [\alpha_1 \ \cdots \ \alpha_n]^\top \in \mathbb{R}^n \end{cases}$$

Tossing-coin-sketch (TCS) for solving GLMs

Objective: $\nabla f(w) = \frac{1}{n} \sum_{i=1}^{n} \underbrace{\phi_i'(a_i^\top w)}_{-\alpha_i} a_i + \lambda w = 0$

- Fixed point equations

$$\alpha_i = -\phi_i'(a_i^\top w), \quad \text{for } i = 1, \ldots, n,$$
$$w = \frac{1}{\lambda n} A \alpha \in \mathbb{R}^d .$$

$$\begin{cases} \mathbf{A} & := \begin{bmatrix} a_1 & \cdots & a_n \end{bmatrix} \in \mathbb{R}^{d \times n} \\ \alpha & := \begin{bmatrix} \alpha_1 & \cdots & \alpha_n \end{bmatrix}^\top \in \mathbb{R}^n \end{cases}$$

- $F(x) = 0$ where $F : \mathbb{R}^{n+d} \to \mathbb{R}^{n+d}$, i.e. $p = m = n + d$ and $x = [\alpha; w] \in \mathbb{R}^{n+d}$

Tossing-coin-sketch (TCS) for solving GLMs

Objective: $\nabla f(w) = \frac{1}{n}\sum_{i=1}^{n} \underbrace{\phi_i'(a_i^\top w)}_{-\alpha_i} a_i + \lambda w = 0$

- Fixed point equations

$$\alpha_i = -\phi_i'(a_i^\top w), \quad \text{for } i = 1,\dots,n,$$
$$w = \frac{1}{\lambda n} A\alpha \in \mathbb{R}^d .$$

$$\begin{cases} \mathbf{A} := [a_1 \ \cdots \ a_n] \in \mathbb{R}^{d\times n} \\ \alpha := [\alpha_1 \ \cdots \ \alpha_n]^\top \in \mathbb{R}^n \end{cases}$$

- $F(x) = 0$ where $F : \mathbb{R}^{n+d} \to \mathbb{R}^{n+d}$, i.e. $p = m = n + d$ and $x = [\alpha; w] \in \mathbb{R}^{n+d}$

- Toss a coin to decide which block to sketch 🎲

Tossing-coin-sketch (TCS) for solving GLMs

Objective: $\nabla f(w) = \frac{1}{n}\sum_{i=1}^{n} \underbrace{\phi_i'(a_i^\top w)}_{-\alpha_i} a_i + \lambda w = 0$

- Fixed point equations

With probability $b \in (0,1)$

$$\boxed{\alpha_i = -\,\phi_i'(a_i^\top w), \quad \text{for } i = 1,\ldots,n,}$$

$$w = \frac{1}{\lambda n} A\alpha \in \mathbb{R}^d\,.$$

$$\begin{cases} \mathbf{A} := \left[a_1 \ \cdots \ a_n\right] \in \mathbb{R}^{d\times n} \\ \alpha := \left[\alpha_1 \ \cdots \ \alpha_n\right]^\top \in \mathbb{R}^n \end{cases}$$

- $F(x) = 0$ where $F : \mathbb{R}^{n+d} \to \mathbb{R}^{n+d}$, i.e. $p = m = n + d$ and $x = [\alpha; w] \in \mathbb{R}^{n+d}$
- Toss a coin to decide which block to sketch 🎲

Tossing-coin-sketch (TCS) for solving GLMs

Objective: $\nabla f(w) = \frac{1}{n}\sum_{i=1}^{n} \underbrace{\phi_i'(a_i^\top w)}_{-\alpha_i} a_i + \lambda w = 0$

- Fixed point equations

$$\alpha_i = -\,\phi_i'(a_i^\top w), \quad \text{for } i = 1,\ldots,n,$$

With probability $1-b$

$$\boxed{w = \frac{1}{\lambda n} A\alpha \in \mathbb{R}^d.}$$

$$\begin{cases} \mathbf{A} & := \begin{bmatrix} a_1 & \cdots & a_n \end{bmatrix} \in \mathbb{R}^{d\times n} \\ \alpha & := \begin{bmatrix} \alpha_1 & \cdots & \alpha_n \end{bmatrix}^\top \in \mathbb{R}^n \end{cases}$$

- $F(x) = 0$ where $F : \mathbb{R}^{n+d} \to \mathbb{R}^{n+d}$, i.e. $p = m = n + d$ and $x = [\alpha; w] \in \mathbb{R}^{n+d}$

- Toss a coin to decide which block to sketch 🎲

Tossing-coin-sketch (TCS) for solving GLMs

Objective: $\nabla f(w) = \frac{1}{n} \sum_{i=1}^{n} \underbrace{\phi_i'(a_i^\top w)}_{-\alpha_i} a_i + \lambda w = 0$

- Fixed point equations

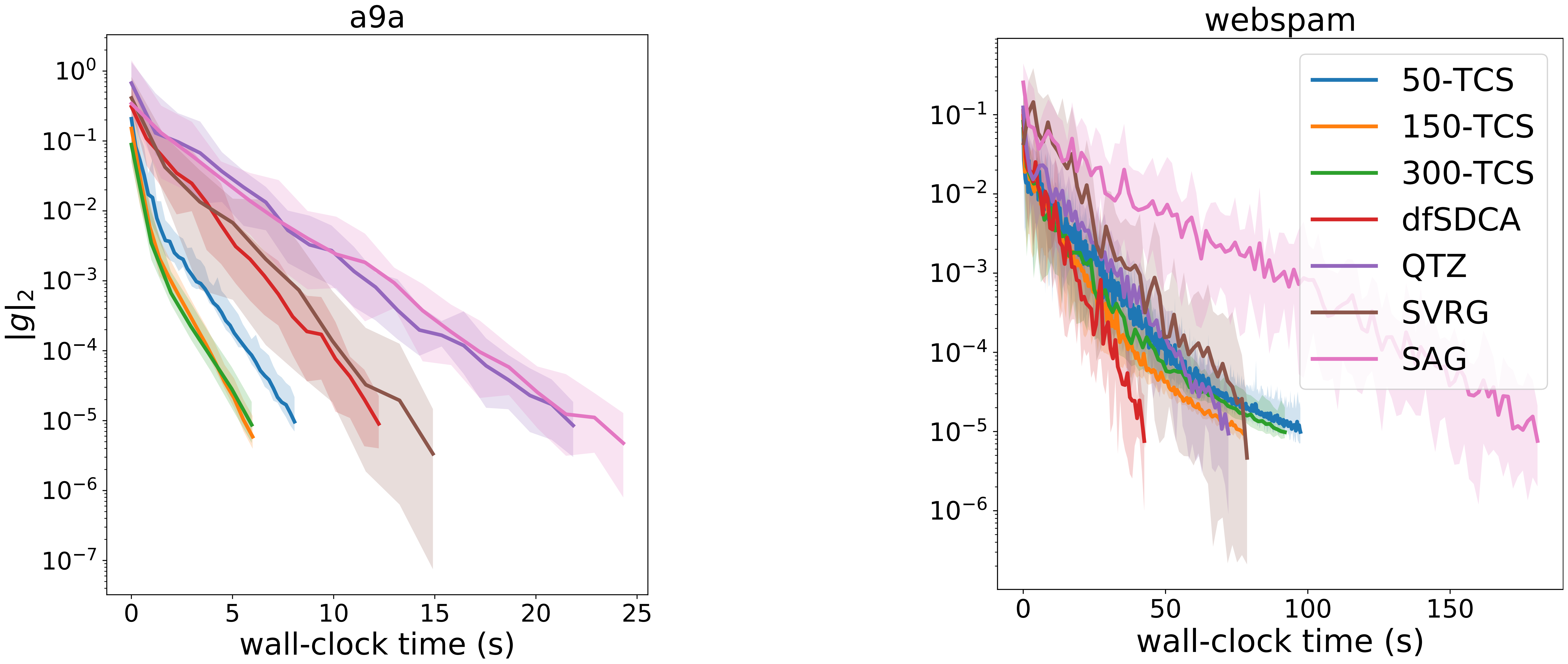
$$\begin{aligned} \alpha_i &= -\phi_i'(a_i^\top w), \quad \text{for } i = 1,\ldots,n, \\ w &= \frac{1}{\lambda n} A\alpha \in \mathbb{R}^d. \end{aligned}$$

$$\begin{cases} \mathbf{A} & := [a_1 \ \cdots \ a_n] \in \mathbb{R}^{d\times n} \\ \alpha & := [\alpha_1 \ \cdots \ \alpha_n]^\top \in \mathbb{R}^n \end{cases}$$

- $F(x) = 0$ where $F : \mathbb{R}^{n+d} \to \mathbb{R}^{n+d}$, i.e. $p = m = n + d$ and $x = [\alpha; w] \in \mathbb{R}^{n+d}$
- Toss a coin to decide which block to sketch 🎲
- Cost per iteration $O(d)$ when the sketch size is $O(1)$ 💡

Logistic regression for binary classification

(see paper for additional experiments)



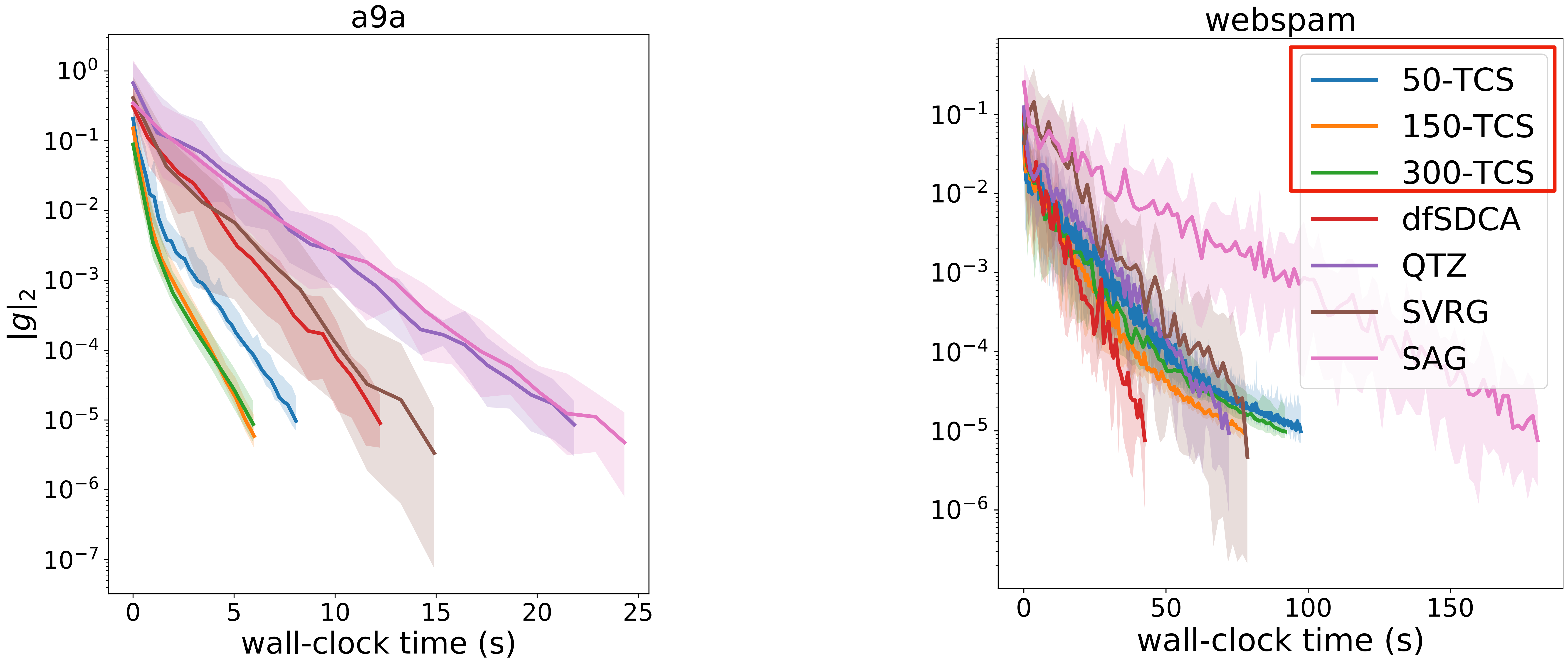
(a) a9a ($d : 123, n : 32561$)

(b) webspam ($d : 254, n : 350000$)

Figure: Experiments for TCS method applied for generalized linear model.

Logistic regression for binary classification

(see paper for additional experiments)



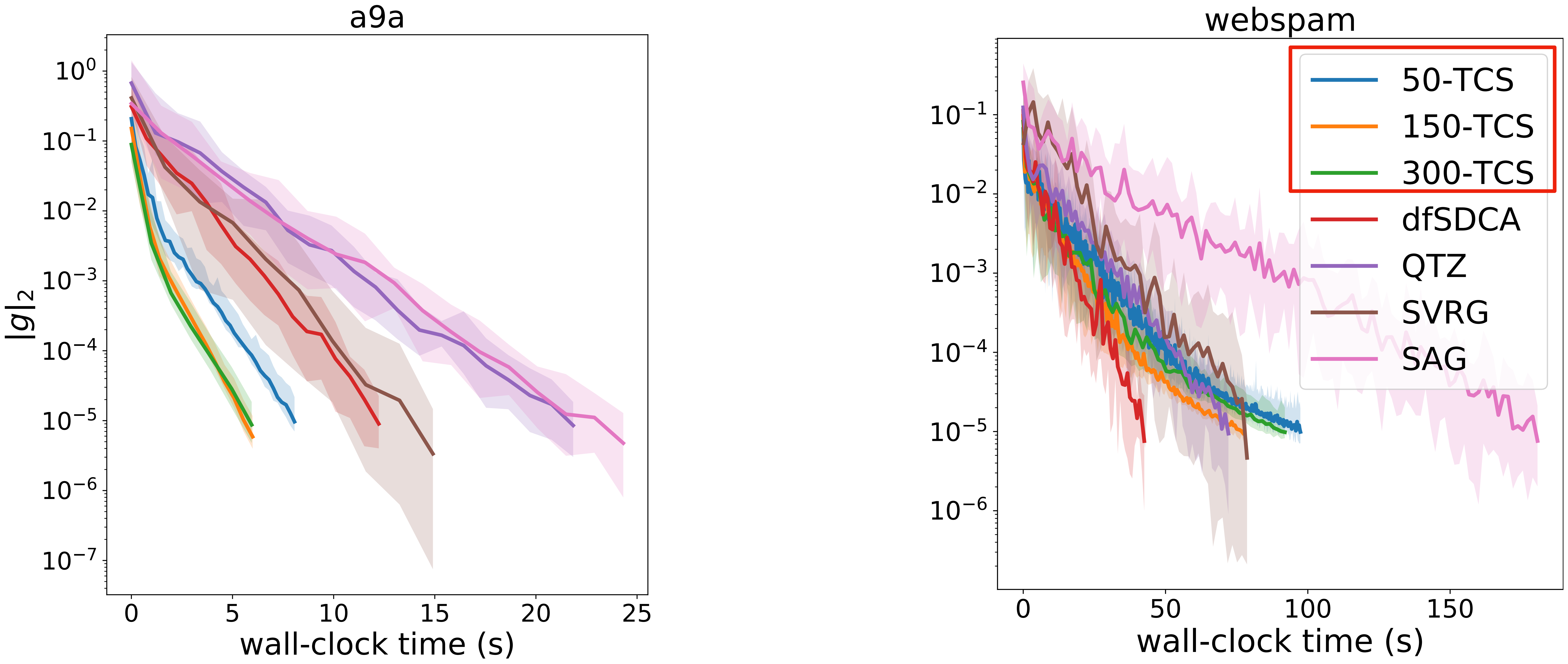
(a) a9a (d : 123, n : 32561)

(b) webspam (d : 254, n : 350000)

Figure: Experiments for TCS method applied for generalized linear model.

Logistic regression for binary classification

(see paper for additional experiments)



⚠ Need to tune the sketch size

(a) a9a (d : 123, n : 32561)

(b) webspam (d : 254, n : 350000)

Figure: Experiments for TCS method applied for generalized linear model.

Design new stochastic second order methods

Motivations

- Develop a second order method for machine learning problems that is incremental, efficient, scales well with the dimension d, and that requires less parameter tuning.

SAN: Stochastic Average Newton

Jiabin Chen*, Rui Yuan*, Guillaume Garrigos, Robert M. Gower
SAN: Stochastic Average Newton Algorithm for Minimizing Finite Sums, AISTATS, 2022.

Finite-sum minimization problem

Finite-sum minimization problem

- Solving a finite-sum minimization problem

$$\min_{w \in \mathbb{R}^d} \left[f(w) := \frac{1}{n} \sum\nolimits_{i=1}^{n} f_i(w) \right]$$

Finite-sum minimization problem

- Solving a finite-sum minimization problem

$$\min_{w\in\mathbb{R}^d}\left[f(w) := \frac{1}{\boxed{n}}\sum\nolimits_{i=1}^{n} f_i(w)\right]$$

n := Number of samples

Finite-sum minimization problem

- Solving a finite-sum minimization problem

$f_i(w)$:= The loss over the ith batch of data

$$\min_{w\in\mathbb{R}^d}\left[f(w) := \frac{1}{\boxed{n}}\sum_{i=1}^{n}\boxed{f_i(w)}\right]$$

n := Number of samples

Finite-sum minimization problem

- Solving a finite-sum minimization problem

$f_i(w)$:= The loss over the ith batch of data

$$\min_{w \in \mathbb{R}^d} \left[f(w) := \frac{1}{\boxed{n}} \sum_{i=1}^{n} \boxed{f_i(w)} \right]$$

n := Number of samples

- Finding a stationary point of the gradient of f : $\nabla f(w) = \frac{1}{n} \sum_{i=1}^{n} \nabla f_i(w) = 0$

SAN: Stochastic Average Newton (1/2)

SAN: Stochastic Average Newton (1/2)

- 1) Rewrite the optimality conditions $\nabla f(w) = \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(w) = 0$ as

$$(1) \qquad \frac{1}{n}\sum_{i=1}^{n} \alpha_i = 0,$$

$$(2) \qquad \alpha_i = \nabla f_i(w) \in \mathbb{R}^d, \quad \forall i \in \{1,\ldots,n\}\,.$$

SAN: Stochastic Average Newton (1/2)

- 1) Rewrite the optimality conditions $\nabla f(w) = \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(w) = 0$ as

$$(1) \qquad \frac{1}{n}\sum_{i=1}^{n} \alpha_i = 0,$$

$$(2) \qquad \alpha_i = \nabla f_i(w) \in \mathbb{R}^d, \quad \forall i \in \{1,\ldots,n\}.$$

- (n+1) equations ((n+1)d rows)

SAN: Stochastic Average Newton (1/2)

- 1) Rewrite the optimality conditions $\nabla f(w) = \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(w) = 0$ as

$$(1) \qquad \frac{1}{n}\sum_{i=1}^{n} \alpha_i = 0,$$

$$(2) \qquad \alpha_i = \nabla f_i(w) \in \mathbb{R}^d, \quad \forall i \in \{1,\ldots,n\}\,.$$

- (n+1) equations ((n+1)d rows)
- (n+1) variables $\left[w; \alpha_1; \cdots; \alpha_n\right] \in \mathbb{R}^{(n+1)d}$

SAN: Stochastic Average Newton (2/2)

(n+1) equations: (1) : $\frac{1}{n}\sum_{i=1}^{n} \alpha_i = 0,$ (2) : $\alpha_i = \nabla f_i(w), \quad \forall i \in \{1,\ldots,n\}$

SAN: Stochastic Average Newton (2/2)

(n+1) equations: (1) : $\frac{1}{n}\sum_{i=1}^{n}\alpha_i = 0,$ (2) : $\alpha_i = \nabla f_i(w), \quad \forall i \in \{1,\ldots,n\}$

- 2) ☞ Sketched Newton Raphson 📗 [Yuan et al., 2022]

SAN: Stochastic Average Newton (2/2)

(n+1) equations: (1) : $\frac{1}{n}\sum_{i=1}^{n} \alpha_i = 0$, (2) : $\alpha_i = \nabla f_i(w)$, $\forall i \in \{1,\ldots,n\}$

- 2) 👉 Sketched Newton Raphson 📗 [Yuan et al., 2022]
  - With probability 1/(n+1), *sample* eq. (1) and *project* onto its set of solutions:

$$\alpha_1^{k+1},\ldots,\alpha_n^{k+1} = \underset{\alpha_1,\ldots,\alpha_n \in \mathbb{R}^d}{\arg\min} \sum_{i=1}^{n} \|\alpha_i - \alpha_i^k\|^2$$

$$\text{s.t. } \frac{1}{n}\sum_{i=1}^{n} \alpha_i = 0$$

SAN: Stochastic Average Newton (2/2)

(n+1) equations: $\boxed{(1) : \frac{1}{n}\sum_{i=1}^{n} \alpha_i = 0,}$ $(2) : \alpha_i = \nabla f_i(w), \quad \forall i \in \{1,\ldots,n\}$

- 2) 👉 Sketched Newton Raphson 📗 [Yuan et al., 2022]
  - With probability 1/(n+1), *sample* eq. (1) and *project* onto its set of solutions:

$$\alpha_1^{k+1},\ldots,\alpha_n^{k+1} = \underset{\alpha_1,\ldots,\alpha_n \in \mathbb{R}^d}{\arg\min} \sum_{i=1}^{n} \|\alpha_i - \alpha_i^k\|^2$$

$$\text{s.t. } \frac{1}{n}\sum_{i=1}^{n} \alpha_i = 0$$

SAN: Stochastic Average Newton (2/2)

(n+1) equations: $\boxed{(1) : \frac{1}{n}\sum_{i=1}^{n} \alpha_i = 0,}$ $(2) : \alpha_i = \nabla f_i(w), \quad \forall i \in \{1,\ldots,n\}$

- 2) ☞ Sketched Newton Raphson 📗 [Yuan et al., 2022]
  - With probability 1/(n+1), *sample* eq. (1) and *project* onto its set of solutions:

$$\alpha_1^{k+1}, \ldots, \alpha_n^{k+1} = \underset{\alpha_1,\ldots,\alpha_n \in \mathbb{R}^d}{\arg\min} \sum_{i=1}^{n} \|\alpha_i - \alpha_i^k\|^2$$

$$\boxed{\text{s.t. } \frac{1}{n}\sum_{i=1}^{n} \alpha_i = 0}$$

SAN: Stochastic Average Newton (2/2)

(n+1) equations: (1) : $\frac{1}{n}\sum_{i=1}^{n} \alpha_i = 0,$ (2) : $\alpha_i = \nabla f_i(w), \quad \forall i \in \{1,\ldots,n\}$

- 2) 👉 Sketched Newton Raphson 📗 [Yuan et al., 2022]
  - With probability 1/(n+1), *sample* eq. (1) and *project* onto its set of solutions:

$$\alpha_1^{k+1}, \ldots, \alpha_n^{k+1} \boxed{= \underset{\alpha_1,\ldots,\alpha_n \in \mathbb{R}^d}{\arg\min} \sum_{i=1}^{n} \|\alpha_i - \alpha_i^k\|^2}$$

$$\text{s.t. } \frac{1}{n}\sum_{i=1}^{n} \alpha_i = 0$$

SAN: Stochastic Average Newton (2/2)

(n+1) equations: (1) : $\frac{1}{n}\sum_{i=1}^{n} \alpha_i = 0$, (2) : $\alpha_i = \nabla f_i(w), \quad \forall i \in \{1,\ldots,n\}$

- 2) ☞ Sketched Newton Raphson 📗 [Yuan et al., 2022]
  - With probability 1/(n+1), *sample* eq. (1) and *project* onto its set of solutions:

$$\alpha_1^{k+1}, \ldots, \alpha_n^{k+1} = \underset{\alpha_1,\ldots,\alpha_n \in \mathbb{R}^d}{\arg\min} \sum_{i=1}^{n} \|\alpha_i - \alpha_i^k\|^2$$

$$\text{s.t. } \frac{1}{n}\sum_{i=1}^{n} \alpha_i = 0$$

  - With probability 1/(n+1), *sample* the j-th eq. of (2), and *project* onto the set of solutions of its *linearization* at w^k:

$$\alpha_j^{k+1}, w^{k+1} = \underset{\alpha_j, w \in \mathbb{R}^d}{\arg\min} \|\alpha_j - \alpha_j^k\|^2 + \|w - w^k\|^2_{\nabla^2 f_j(w^k)}$$

$$\text{s.t. } \nabla f_j(w^k) + \nabla^2 f_j(w^k)(w - w^k) = \alpha_j$$

SAN: Stochastic Average Newton (2/2)

(n+1) equations: (1) : $\frac{1}{n}\sum_{i=1}^{n} \alpha_i = 0,$ $\boxed{\text{(2)} : \alpha_i = \nabla f_i(w), \quad \forall i \in \{1,\ldots,n\}}$

- 2) 👉 Sketched Newton Raphson 📗 [Yuan et al., 2022]
  - With probability 1/(n+1), *sample* eq. (1) and *project* onto its set of solutions:

$$\alpha_1^{k+1},\ldots,\alpha_n^{k+1} = \underset{\alpha_1,\ldots,\alpha_n \in \mathbb{R}^d}{\arg\min} \; \sum_{i=1}^{n} \|\alpha_i - \alpha_i^k\|^2$$

$$\text{s.t. } \frac{1}{n}\sum_{i=1}^{n} \alpha_i = 0$$

  - With probability 1/(n+1), *sample* the j-th eq. of (2), and *project* onto the set of solutions of its *linearization* at w^k:

$$\alpha_j^{k+1}, w^{k+1} = \underset{\alpha_j, w \in \mathbb{R}^d}{\arg\min} \; \|\alpha_j - \alpha_j^k\|^2 + \|w - w^k\|^2_{\nabla^2 f_j(w^k)}$$

$$\text{s.t. } \nabla f_j(w^k) + \nabla^2 f_j(w^k)(w - w^k) = \alpha_j$$

SAN: Stochastic Average Newton (2/2)

(n+1) equations: (1) : $\frac{1}{n}\sum_{i=1}^{n}\alpha_i = 0,$ $\boxed{(2) : \alpha_i = \nabla f_i(w), \quad \forall i \in \{1,\ldots,n\}}$

- 2) ☞ Sketched Newton Raphson [Yuan et al., 2022]
  - With probability 1/(n+1), *sample* eq. (1) and *project* onto its set of solutions:

$$\alpha_1^{k+1},\ldots,\alpha_n^{k+1} = \underset{\alpha_1,\ldots,\alpha_n\in\mathbb{R}^d}{\arg\min}\ \sum_{i=1}^{n}\|\alpha_i-\alpha_i^k\|^2$$

$$\text{s.t. } \frac{1}{n}\sum_{i=1}^{n}\alpha_i = 0$$

  - With probability 1/(n+1), *sample* the j-th eq. of (2), and *project* onto the set of solutions of its *linearization* at w^k:

$$\alpha_j^{k+1}, w^{k+1} = \underset{\alpha_j,w\in\mathbb{R}^d}{\arg\min}\ \|\alpha_j-\alpha_j^k\|^2 + \|w-w^k\|^2_{\nabla^2 f_j(w^k)}$$

$$\boxed{\text{s.t. } \nabla f_j(w^k) + \nabla^2 f_j(w^k)(w-w^k) = \alpha_j}$$

SAN: Stochastic Average Newton (2/2)

(n+1) equations: (1) : $\frac{1}{n}\sum_{i=1}^{n}\alpha_i = 0,$ (2) : $\alpha_i = \nabla f_i(w), \quad \forall i \in \{1,\ldots,n\}$

- 2) 👉 Sketched Newton Raphson 📗 [Yuan et al., 2022]
  - With probability 1/(n+1), *sample* eq. (1) and *project* onto its set of solutions:

$$\alpha_1^{k+1},\ldots,\alpha_n^{k+1} = \underset{\alpha_1,\ldots,\alpha_n\in\mathbb{R}^d}{\arg\min}\ \sum_{i=1}^{n}\|\alpha_i-\alpha_i^k\|^2$$

$$\text{s.t. } \frac{1}{n}\sum_{i=1}^{n}\alpha_i = 0$$

  - With probability 1/(n+1), *sample* the j-th eq. of (2), and *project* onto the set of solutions of its *linearization* at w^k:

$$\alpha_j^{k+1}, w^{k+1} \boxed{= \underset{\alpha_j, w\in\mathbb{R}^d}{\arg\min}\ \|\alpha_j-\alpha_j^k\|^2 + \|w-w^k\|^2_{\nabla^2 f_j(w^k)}}$$

$$\text{s.t. } \nabla f_j(w^k) + \nabla^2 f_j(w^k)(w-w^k) = \alpha_j$$

SAN: Stochastic Average Newton (2/2)

(n+1) equations: (1) : $\frac{1}{n}\sum_{i=1}^{n}\alpha_i = 0$, (2) : $\alpha_i = \nabla f_i(w)$, $\forall i \in \{1,\ldots,n\}$

- 2) ☞ Sketched Newton Raphson 📗 [Yuan et al., 2022]
  - With probability 1/(n+1), *sample* eq. (1) and *project* onto its set of solutions:

$$\alpha_1^{k+1},\ldots,\alpha_n^{k+1} = \underset{\alpha_1,\ldots,\alpha_n\in\mathbb{R}^d}{\arg\min}\ \sum_{i=1}^{n}\|\alpha_i-\alpha_i^k\|^2$$

$$\text{s.t. } \frac{1}{n}\sum_{i=1}^{n}\alpha_i = 0$$

  - With probability 1/(n+1), *sample* the j-th eq. of (2), and *project* onto the set of solutions of its *linearization* at w^k:

$$\alpha_j^{k+1}, w^{k+1} = \underset{\alpha_j, w\in\mathbb{R}^d}{\arg\min}\ \|\alpha_j-\alpha_j^k\|^2 + \|w-w^k\|^2_{\boxed{\nabla^2 f_j(w^k)}}$$

$$\text{s.t. } \nabla f_j(w^k) + \nabla^2 f_j(w^k)(w-w^k) = \alpha_j$$

What’s the point by doing this ?

(see paper for technique details and additional properties)

What’s the point by doing this ?

(see paper for technique details and additional properties)

- It turns out that SAN

What’s the point by doing this ?

(see paper for technique details and additional properties)

- It turns out that SAN

❶ is *incremental*, i.e. samples only one single data point per iteration;

What's the point by doing this ?

(see paper for technique details and additional properties)

- It turns out that SAN

❶ is *incremental*, i.e. samples only one single data point per iteration;

❷ is *efficient* and scales well with the dimension d, i.e. costs $O(d)$ per iteration for generalized linear models;

What’s the point by doing this ?

(see paper for technique details and additional properties)

- It turns out that SAN

❶ is *incremental*, i.e. samples only one single data point per iteration;

❷ is *efficient* and scales well with the dimension d, i.e. costs $O(d)$ per iteration for generalized linear models;

❸ requires less parameter tuning (*e.g. learning rate, sketch size*).

What’s the point by doing this ?

(see paper for technique details and additional properties)

- It turns out that SAN

❶ is *incremental*, i.e. samples only one single data point per iteration;

❷ is *efficient* and scales well with the dimension d, i.e. costs $O(d)$ per iteration for generalized linear models;

❸ requires less parameter tuning (*e.g. learning rate, sketch size*).

👉 We provide a *global linear convergence theory* of SAN

What's the point by doing this ?

(see paper for technique details and additional properties)

- It turns out that SAN

❶ is *incremental*, i.e. samples only one single data point per iteration;

❷ is *efficient* and scales well with the dimension d, i.e. costs $O(d)$ per iteration for generalized linear models;

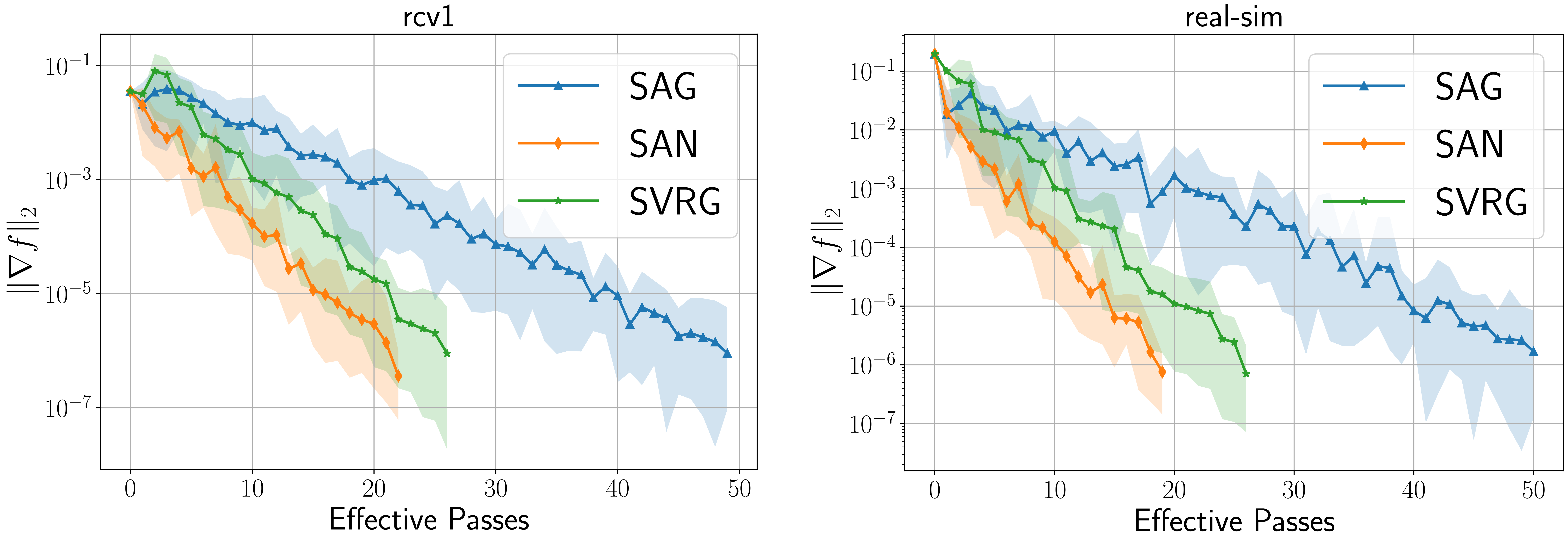
❸ requires less parameter tuning (*e.g. learning rate, sketch size*).

☞ We provide a *global linear convergence theory* of SAN

☞ Using our approach, we develop other new stochastic Newton methods, e.g., SANA and SNRVM

Logistic regression for binary classification

(see paper for additional experiments)



(a) rcv1 (d : 47236, n : 20242)

(b) real-sim (d : 20958, n : 72309)

Figure: Experiments for SAN applied for generalized linear model.

— Part II —

Finite Time Analysis of Policy Gradient Methods in Reinforcement Learning

Introduction (Part II)

Impressive Reinforcement Learning (RL) Results

Impressive Reinforcement Learning (RL) Results

Board Game

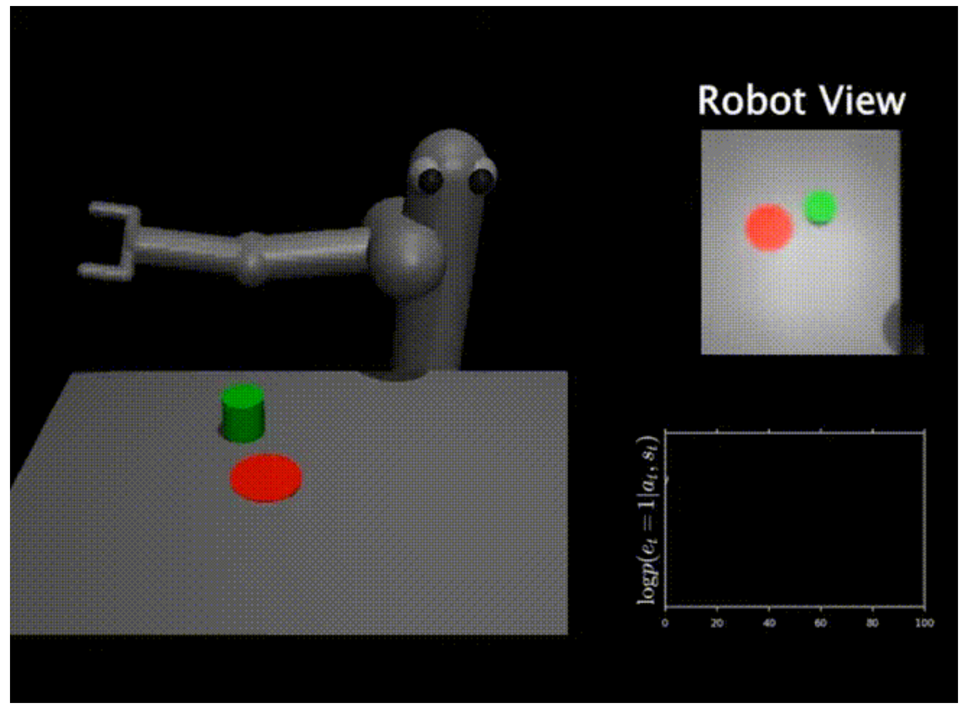
nature
THE INTERNATIONAL WEEKLY JOURNAL OF SCIENCE
At last — a computer program that can beat a champion Go player PAGE 484
ALL SYSTEMS GO

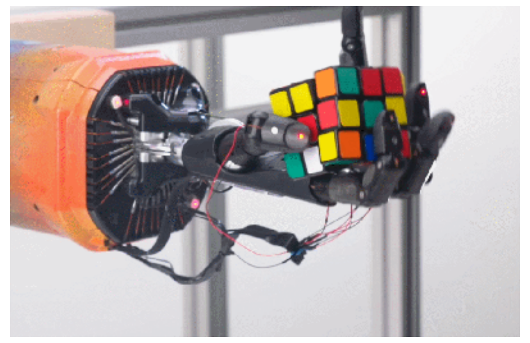
Impressive Reinforcement Learning (RL) Results

Board Game

nature
THE INTERNATIONAL WEEKLY JOURNAL OF SCIENCE
At last — a computer program that can beat a champion Go player PAGE 484
ALL SYSTEMS GO

Robotic Manipulation





Impressive Reinforcement Learning (RL) Results

Board Game

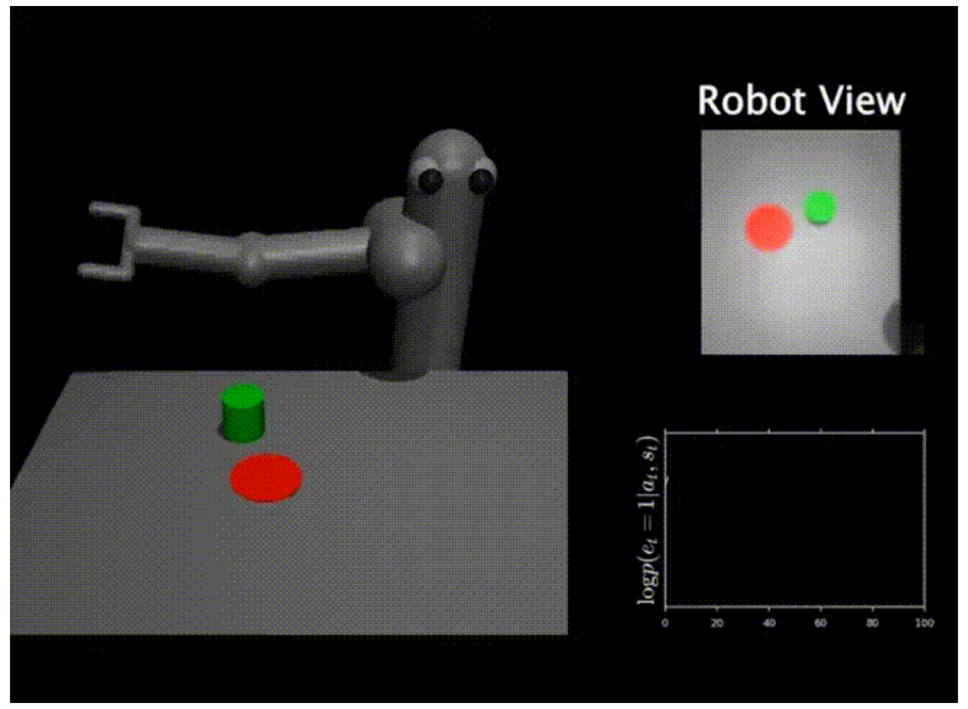
nature

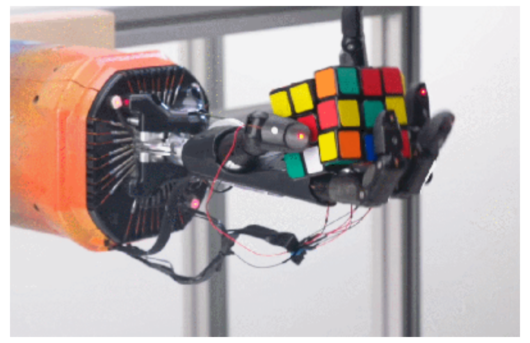
THE INTERNATIONAL WEEKLY JOURNAL OF SCIENCE

At last — a computer program that can beat a champion Go player PAGE 484

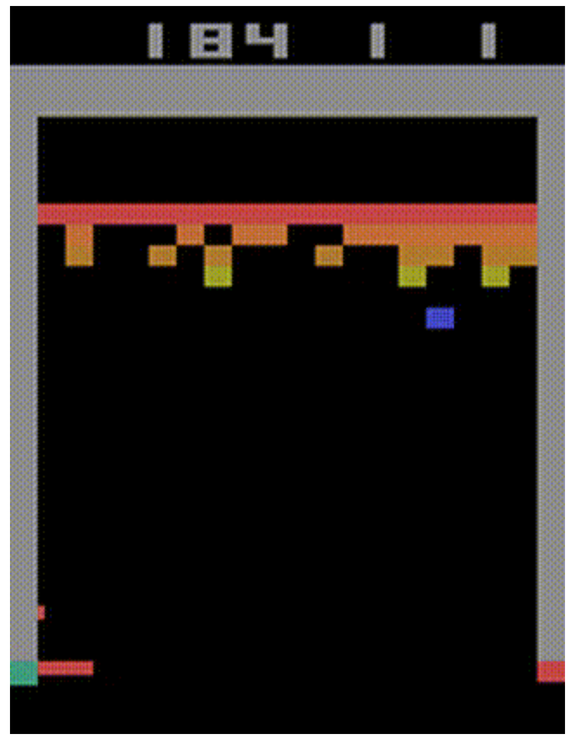
ALL SYSTEMS GO

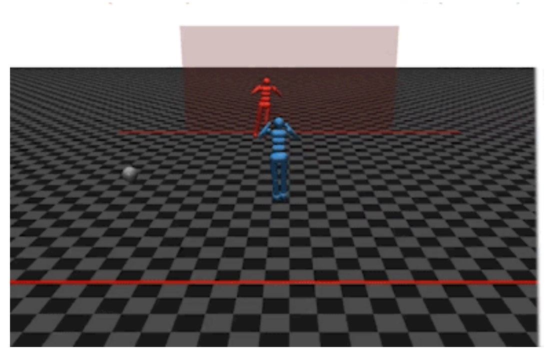
Robotic Manipulation

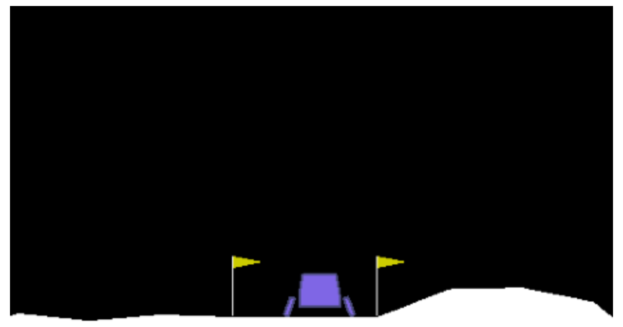




Game Playing

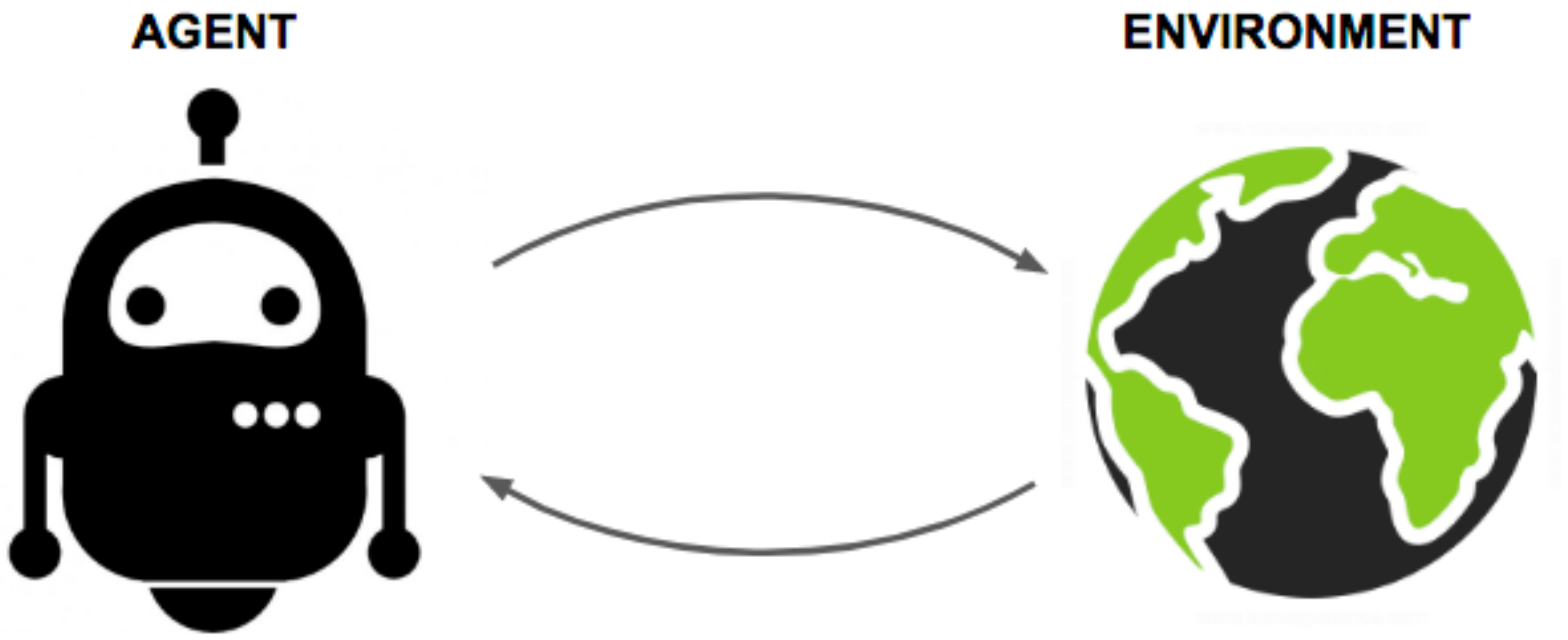






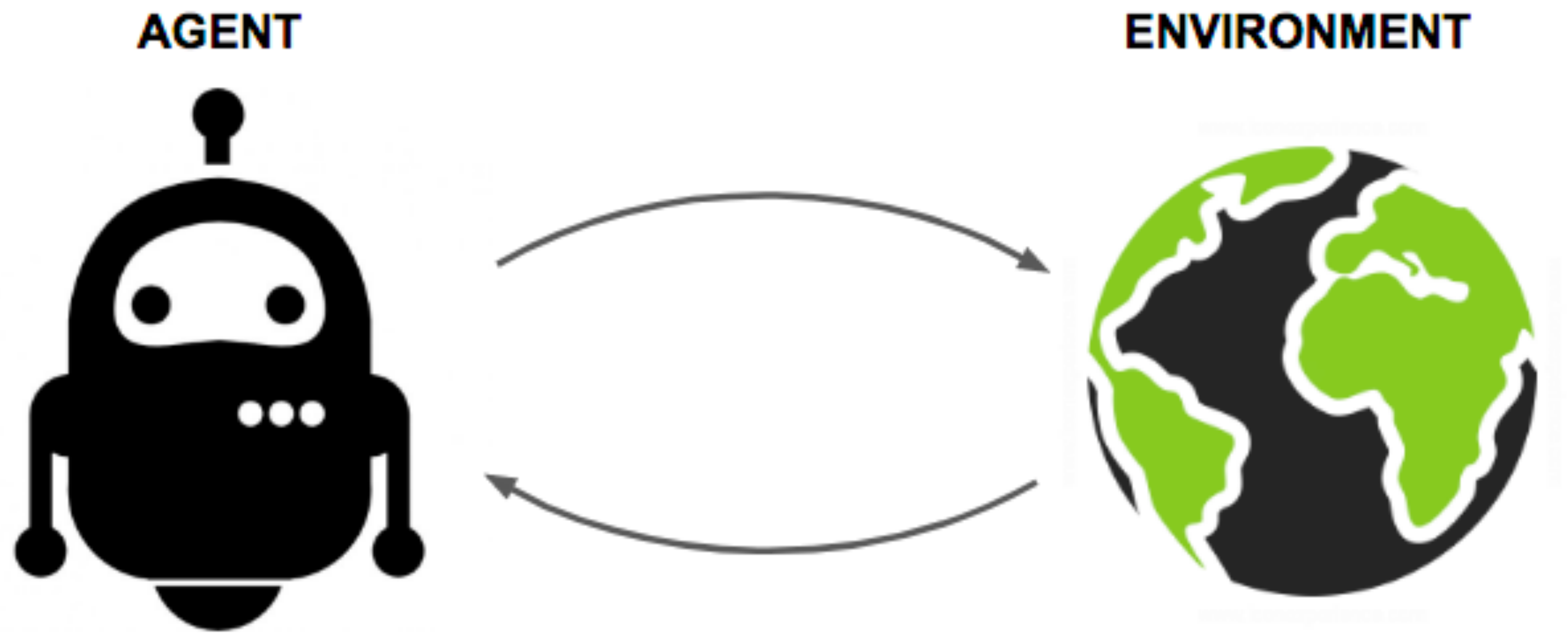
Reinforcement Learning

Sequential decision making problems



Reinforcement Learning

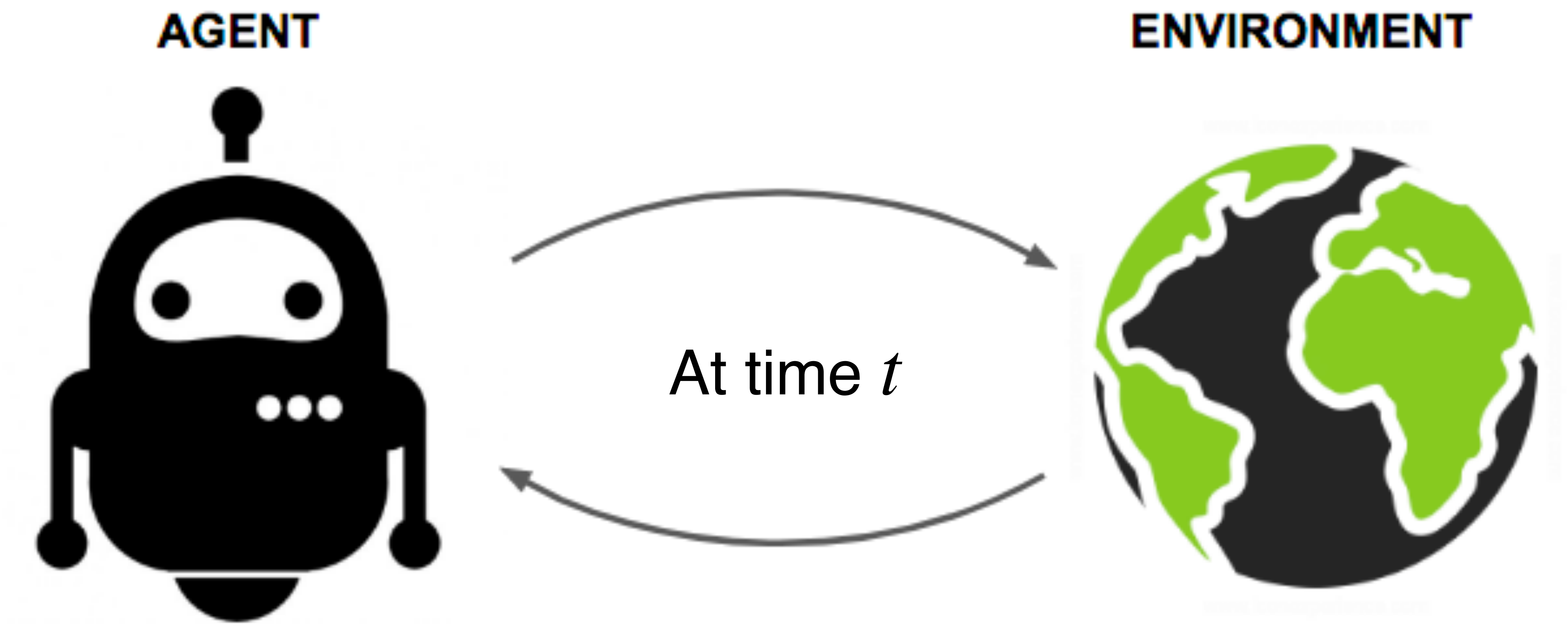
Sequential decision making problems



Markov decision Process (MDP)

Reinforcement Learning

Sequential decision making problems

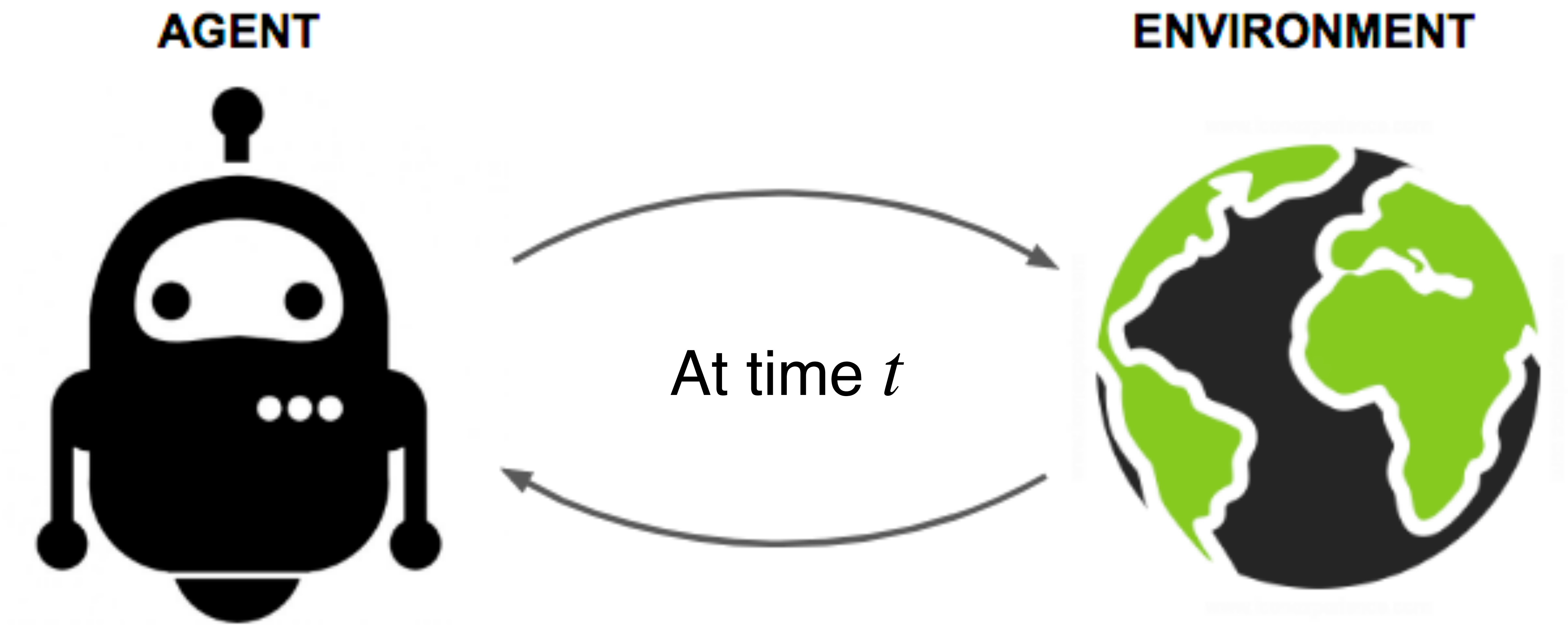


Markov decision Process (MDP)

Reinforcement Learning

Sequential decision making problems

- State $s_t \in \mathcal{S}$

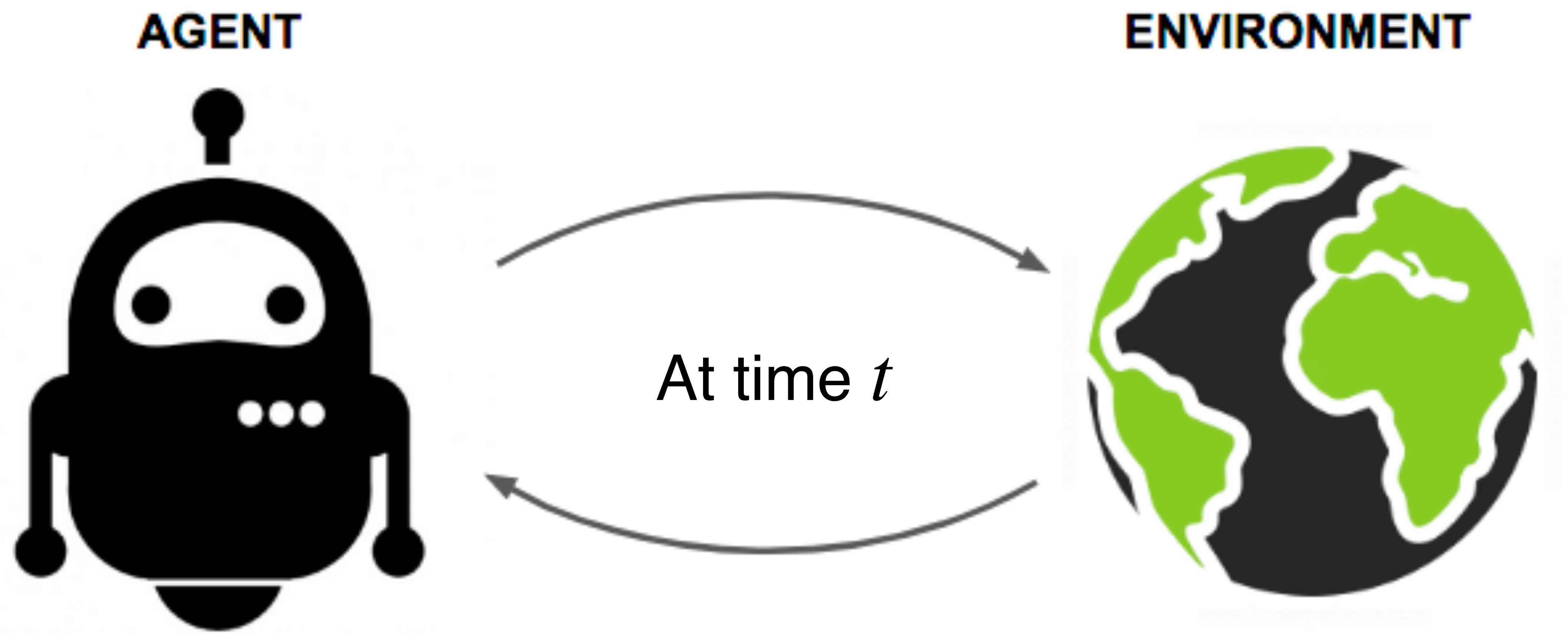


Markov decision Process (MDP)

Reinforcement Learning

Sequential decision making problems

- State $s_t \in \mathcal{S}$



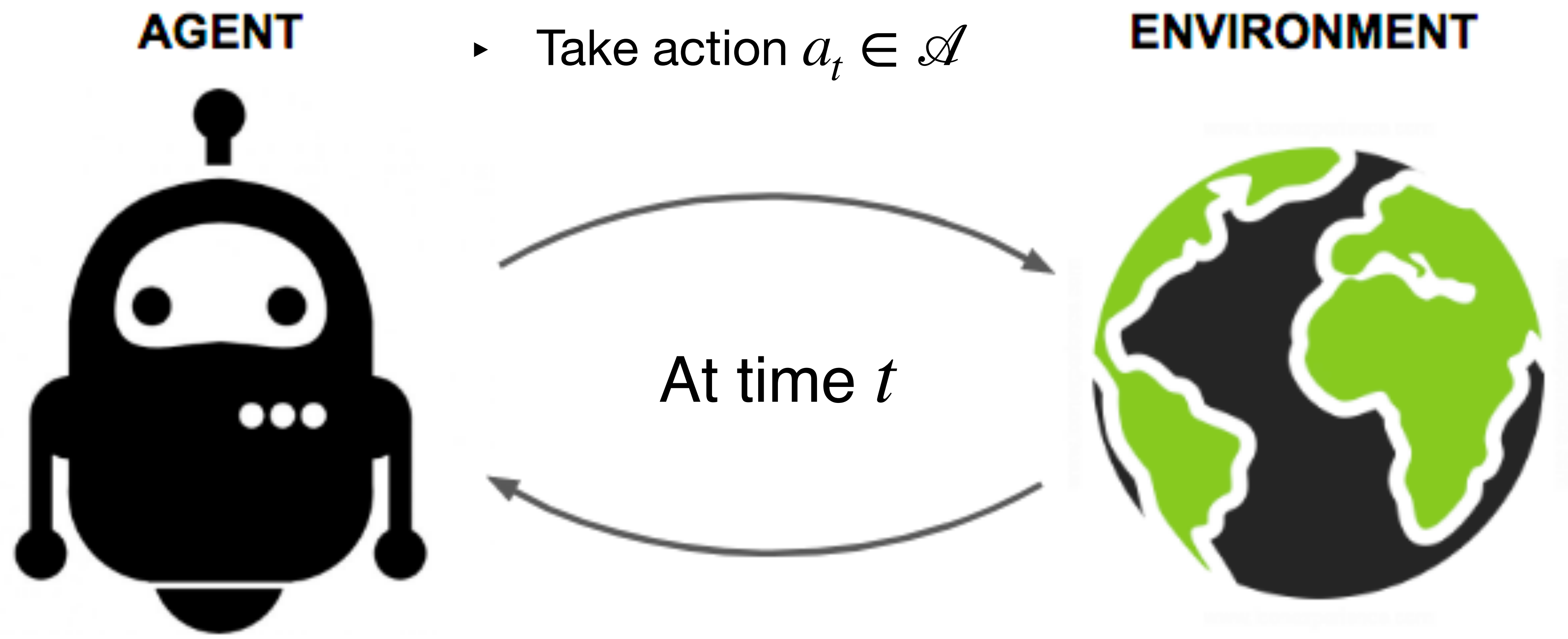
Markov decision Process (MDP)

- State space $\mathcal{S}$

Reinforcement Learning

Sequential decision making problems

- State $s_t \in \mathcal{S}$
- Take action $a_t \in \mathcal{A}$

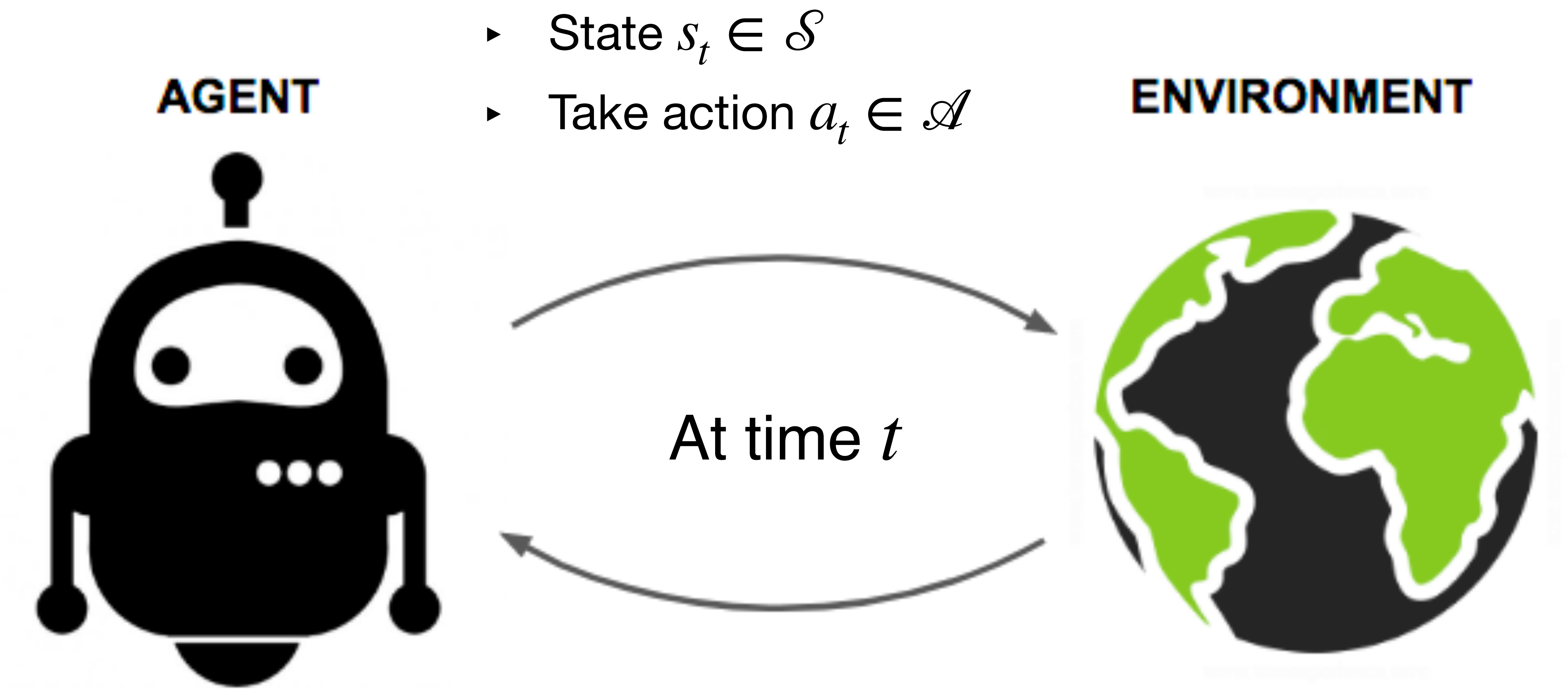


Markov decision Process (MDP)

- State space $\mathcal{S}$

Reinforcement Learning

Sequential decision making problems



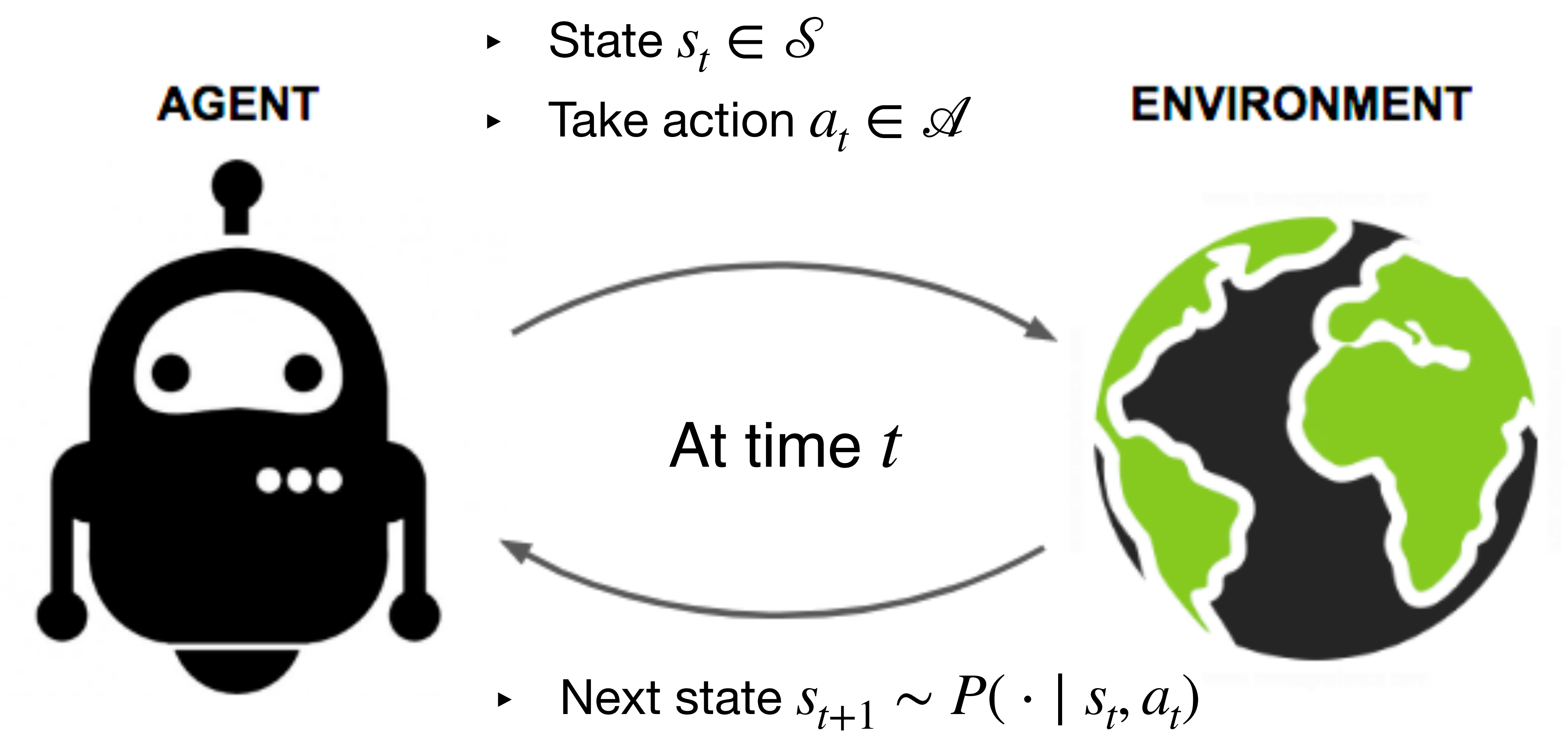
- State $s_t \in \mathcal{S}$
- Take action $a_t \in \mathcal{A}$

Markov decision Process (MDP)

- State space $\mathcal{S}$
- Action space $\mathcal{A}$

Reinforcement Learning

Sequential decision making problems



- State $s_t \in \mathcal{S}$
- Take action $a_t \in \mathcal{A}$
- Next state $s_{t+1} \sim P(\cdot \mid s_t, a_t)$

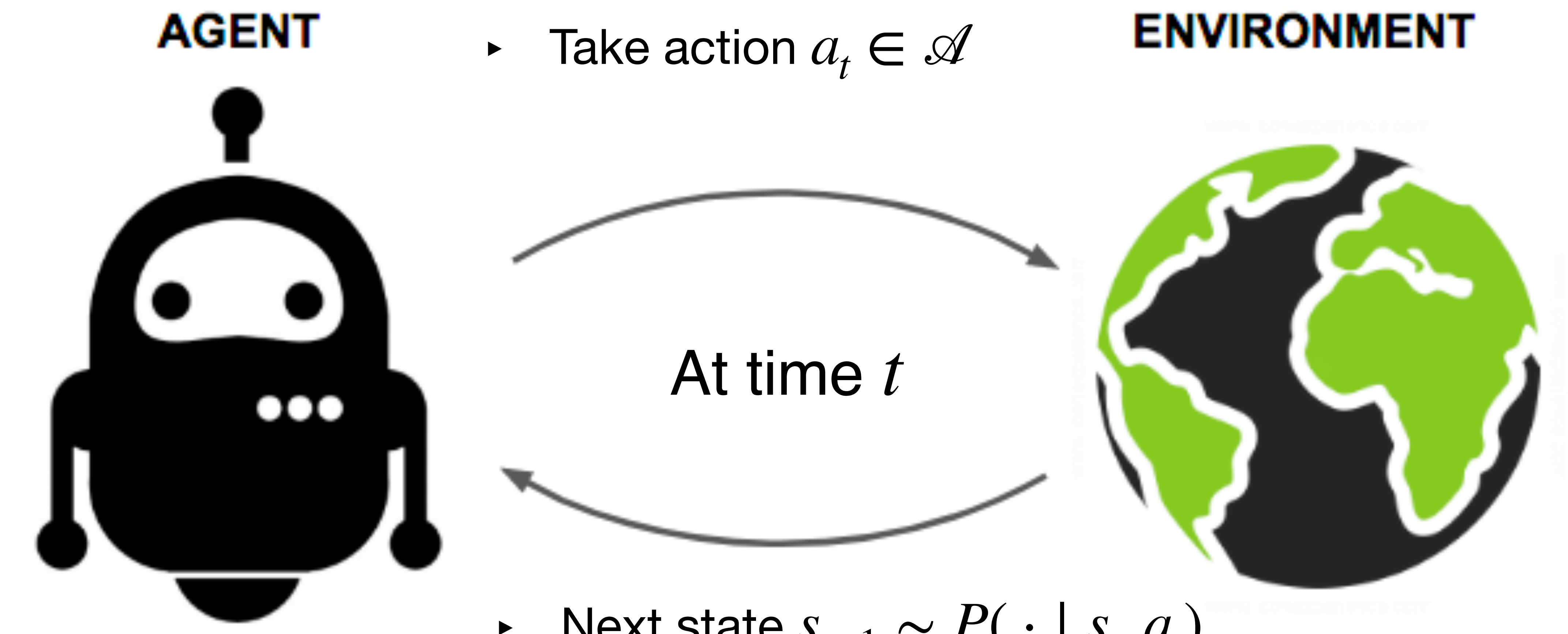
Markov decision Process (MDP)

- State space $\mathcal{S}$
- Action space $\mathcal{A}$

Reinforcement Learning

Sequential decision making problems

- State $s_t \in \mathscr{S}$
- Take action $a_t \in \mathscr{A}$



At time t

- Next state $s_{t+1} \sim P(\cdot \mid s_t, a_t)$

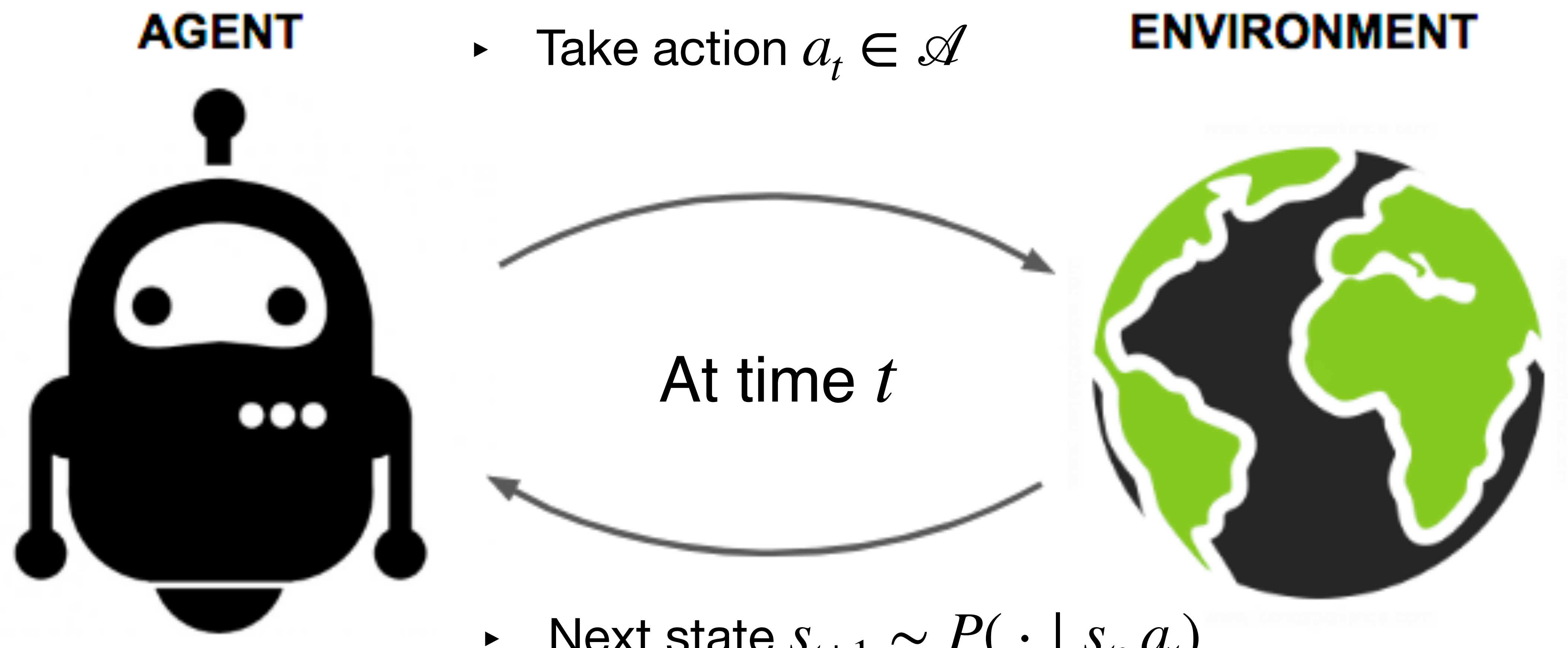
Markov decision Process (MDP)

- State space $\mathscr{S}$
- Action space $\mathscr{A}$
- Transition probabilities P

Reinforcement Learning

Sequential decision making problems

- State $s_t \in \mathcal{S}$
- Take action $a_t \in \mathcal{A}$



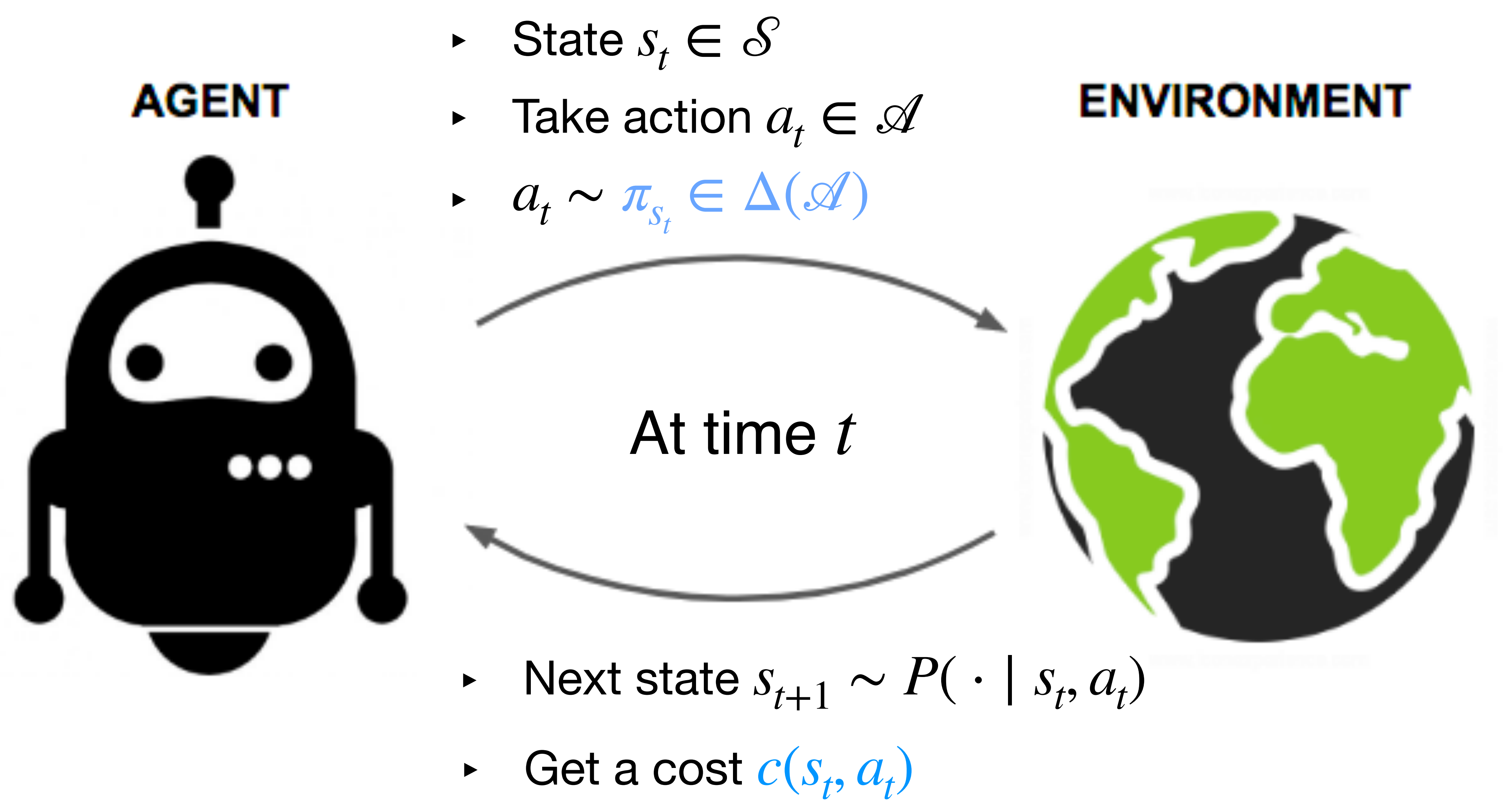
- Next state $s_{t+1} \sim P(\cdot \mid s_t, a_t)$
- Get a cost $c(s_t, a_t)$

Markov decision Process (MDP)

- State space $\mathcal{S}$
- Action space $\mathcal{A}$
- Transition probabilities P

Reinforcement Learning

Sequential decision making problems



AGENT

ENVIRONMENT

- State $s_t \in \mathcal{S}$
- Take action $a_t \in \mathcal{A}$
- $a_t \sim \pi_{s_t} \in \Delta(\mathcal{A})$

At time t

- Next state $s_{t+1} \sim P(\cdot \mid s_t, a_t)$
- Get a cost $c(s_t, a_t)$

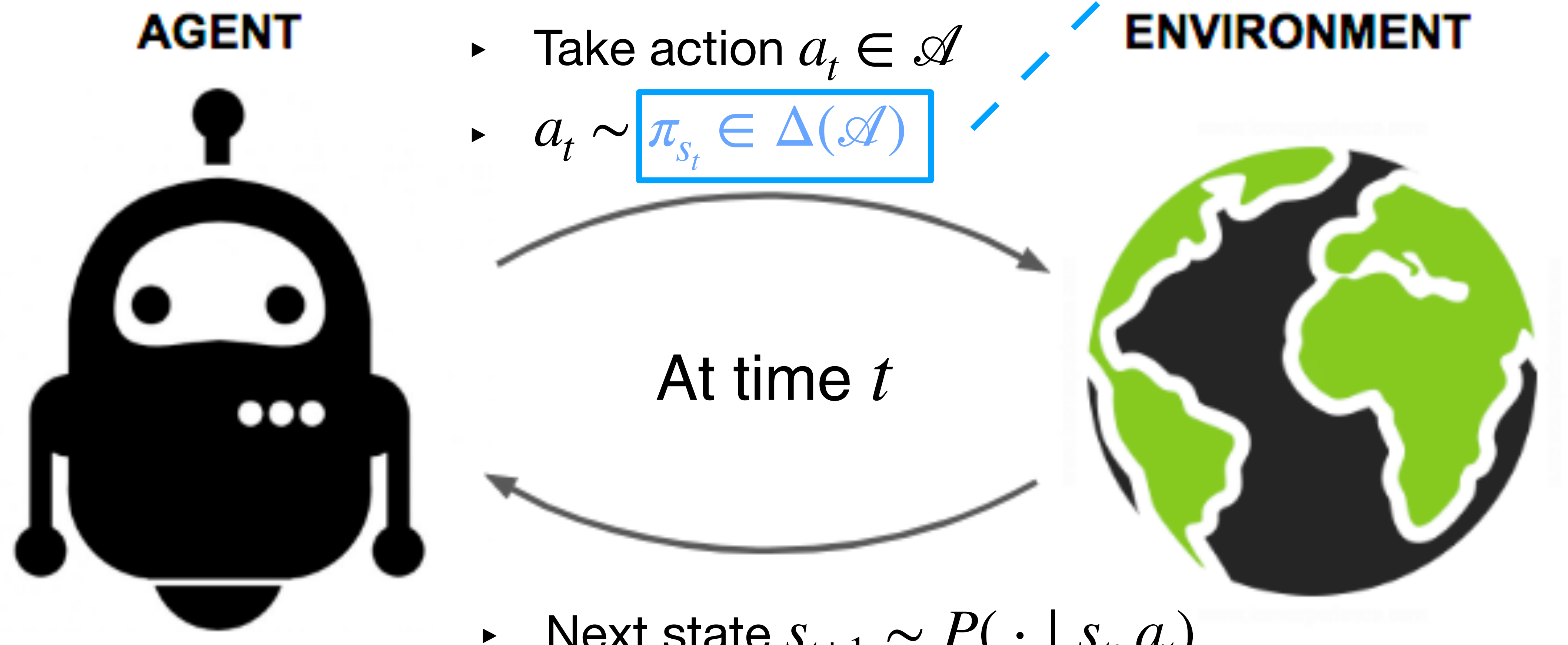
Markov decision Process (MDP)

- State space $\mathcal{S}$
- Action space $\mathcal{A}$
- Transition probabilities P

Reinforcement Learning

Sequential decision making problems

Policy $\pi : \mathscr{S} \to \Delta(\mathscr{A})$,
$\pi_{s_t,a_t} \in \mathbb{R}$ is the density of the distribution over actions at $s_t \in \mathscr{S}$



- State $s_t \in \mathscr{S}$
- Take action $a_t \in \mathscr{A}$
- $a_t \sim \boxed{\pi_{s_t} \in \Delta(\mathscr{A})}$
- Next state $s_{t+1} \sim P(\cdot \mid s_t, a_t)$
- Get a cost $c(s_t, a_t)$

Markov decision Process (MDP)

- State space $\mathscr{S}$
- Action space $\mathscr{A}$
- Transition probabilities P

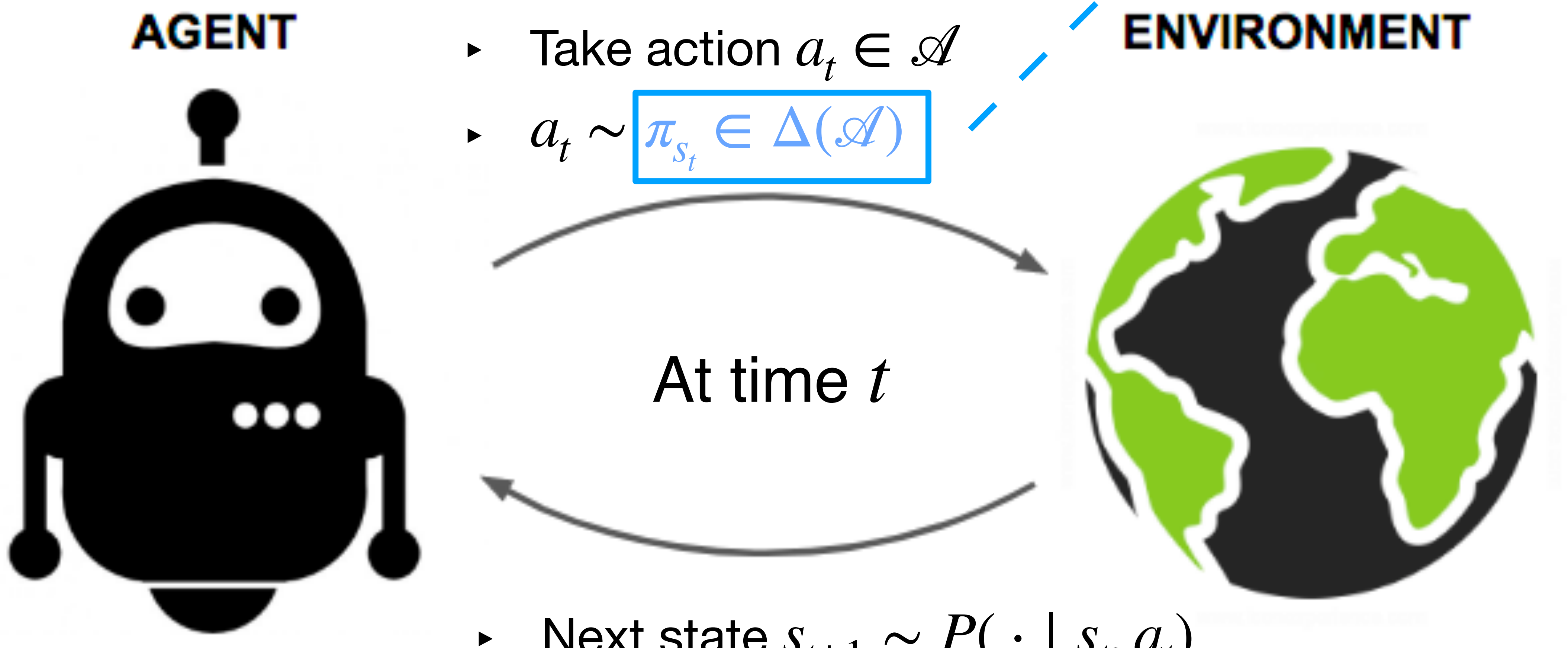
Reinforcement Learning

Sequential decision making problems

Policy $\pi : \mathscr{S} \to \Delta(\mathscr{A})$,

$\pi_{s_t,a_t} \in \mathbb{R}$ is the density of the distribution over actions at $s_t \in \mathscr{S}$

- State $s_t \in \mathscr{S}$
- Take action $a_t \in \mathscr{A}$
- $a_t \sim \boxed{\pi_{s_t} \in \Delta(\mathscr{A})}$



- Next state $s_{t+1} \sim P(\cdot \mid s_t, a_t)$
- Get a cost $c(s_t, a_t)$

Markov decision Process (MDP)

- State space $\mathscr{S}$
- Action space $\mathscr{A}$
- Transition probabilities P

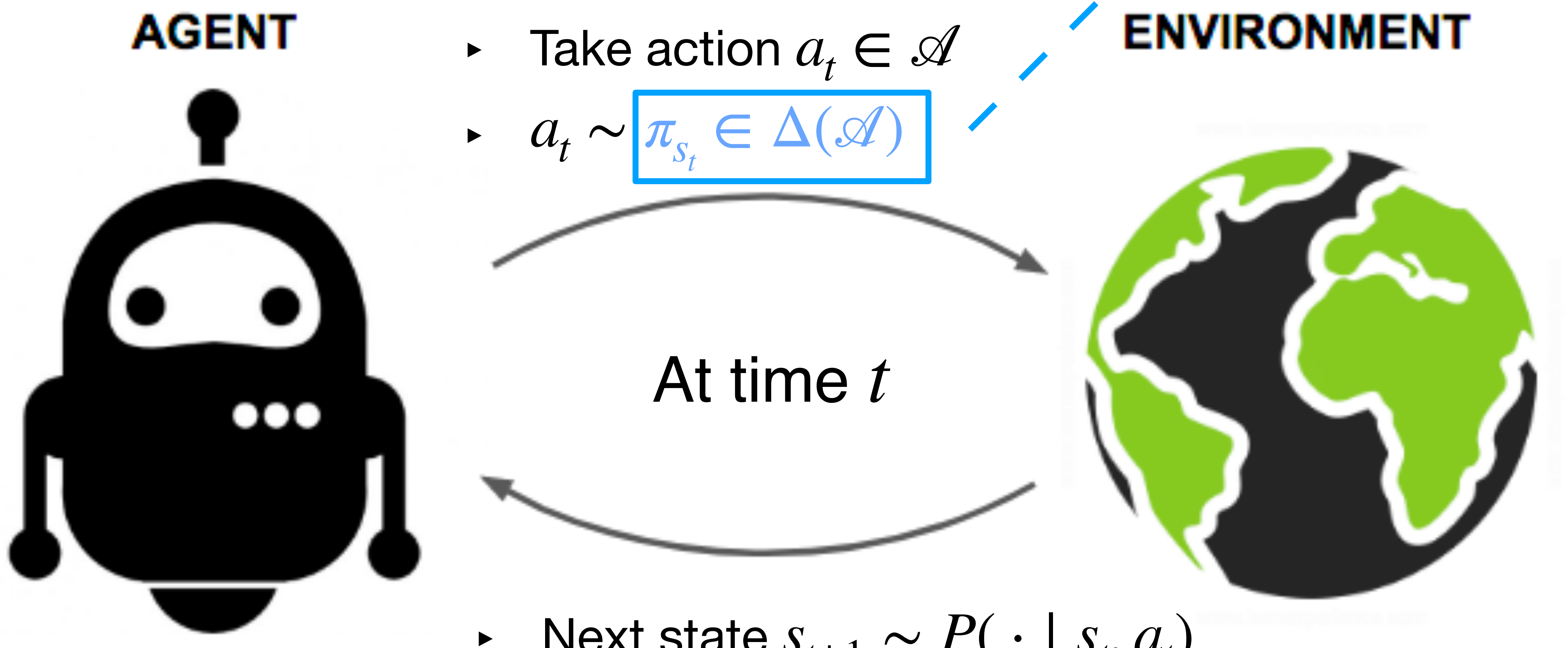
Solve an MDP to minimize total expected cost (a.k.a. policy optimization)

$$\arg\min_{\pi} V_\rho(\pi) := \mathbb{E}_{s_0 \sim \rho,\ a_t \sim \pi_{s_t},\ s_{t+1} \sim P(\cdot|s_t,a_t)} \left[\sum_{t=0}^{\infty} \gamma^t c(s_t, a_t) \right]$$

Reinforcement Learning

Sequential decision making problems

Policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$,
$\pi_{s_t,a_t} \in \mathbb{R}$ is the density of the distribution over actions at $s_t \in \mathcal{S}$



- State $s_t \in \mathcal{S}$
- Take action $a_t \in \mathcal{A}$
- $a_t \sim \boxed{\pi_{s_t} \in \Delta(\mathcal{A})}$
- Next state $s_{t+1} \sim P(\cdot \mid s_t, a_t)$
- Get a cost $c(s_t, a_t)$

Markov decision Process (MDP)

- State space $\mathcal{S}$
- Action space $\mathcal{A}$
- Transition probabilities P

Solve an MDP to minimize total expected cost (a.k.a. policy optimization)

$$\arg\min_{\pi} \boxed{V_\rho(\pi) := \mathbb{E}_{s_0\sim\rho,\ a_t\sim\pi_{s_t},\ s_{t+1}\sim P(\cdot|s_t,a_t)}\left[\sum_{t=0}^{\infty}\gamma^t c(s_t, a_t)\right]}$$

→ Cost function

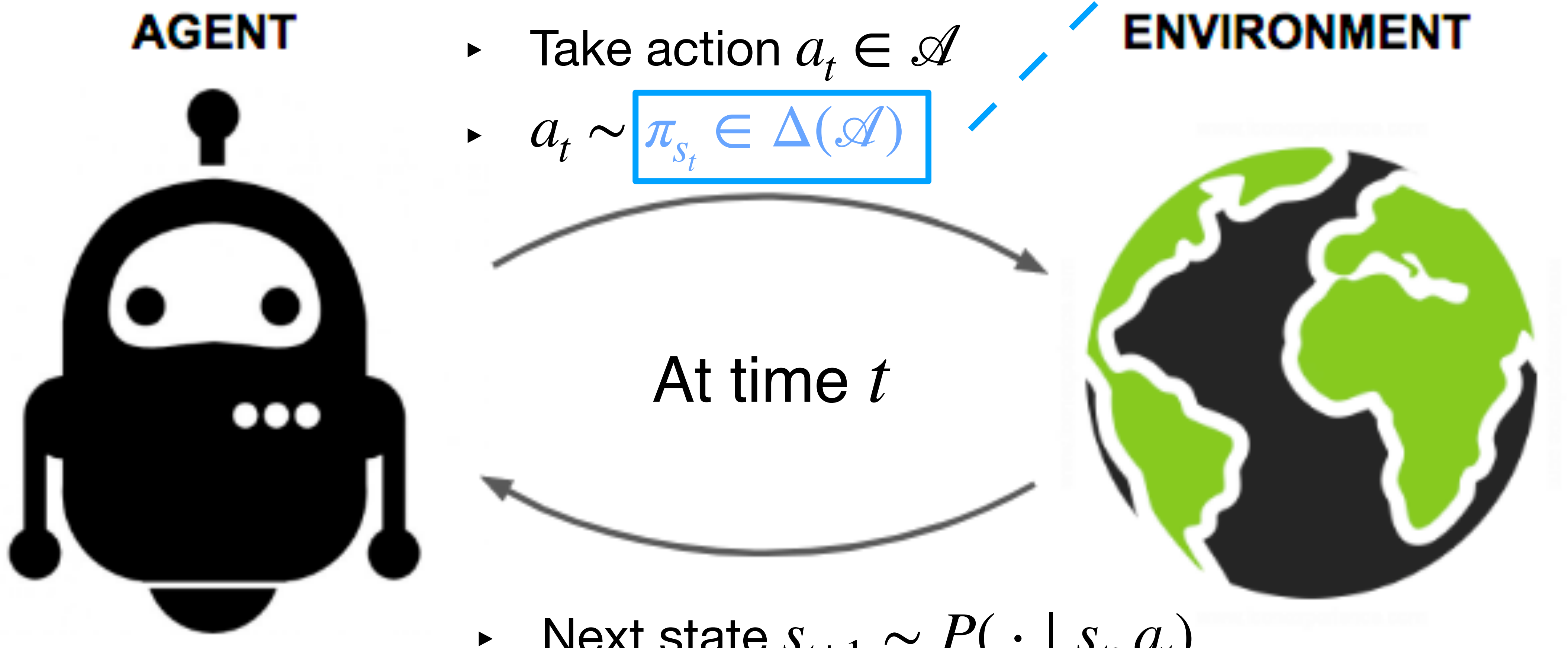
Reinforcement Learning

Sequential decision making problems

Policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$,

$\pi_{s_t,a_t} \in \mathbb{R}$ is the density of the distribution over actions at $s_t \in \mathcal{S}$

- State $s_t \in \mathcal{S}$
- Take action $a_t \in \mathcal{A}$
- $a_t \sim \boxed{\pi_{s_t} \in \Delta(\mathcal{A})}$



- Next state $s_{t+1} \sim P(\cdot \mid s_t, a_t)$
- Get a cost $c(s_t, a_t)$

Markov decision Process (MDP)

- State space $\mathcal{S}$
- Action space $\mathcal{A}$
- Transition probabilities P

Solve an MDP to minimize total expected cost (a.k.a. policy optimization)

$$\arg\min_{\pi} \boxed{V_\rho(\pi) := \mathbb{E}_{\boxed{s_0\sim\rho,}\, a_t\sim\pi_{s_t},\ s_{t+1}\sim P(\cdot|s_t,a_t)}\left[\sum_{t=0}^{\infty}\gamma^t c(s_t,a_t)\right]} \longrightarrow \text{Cost function}$$

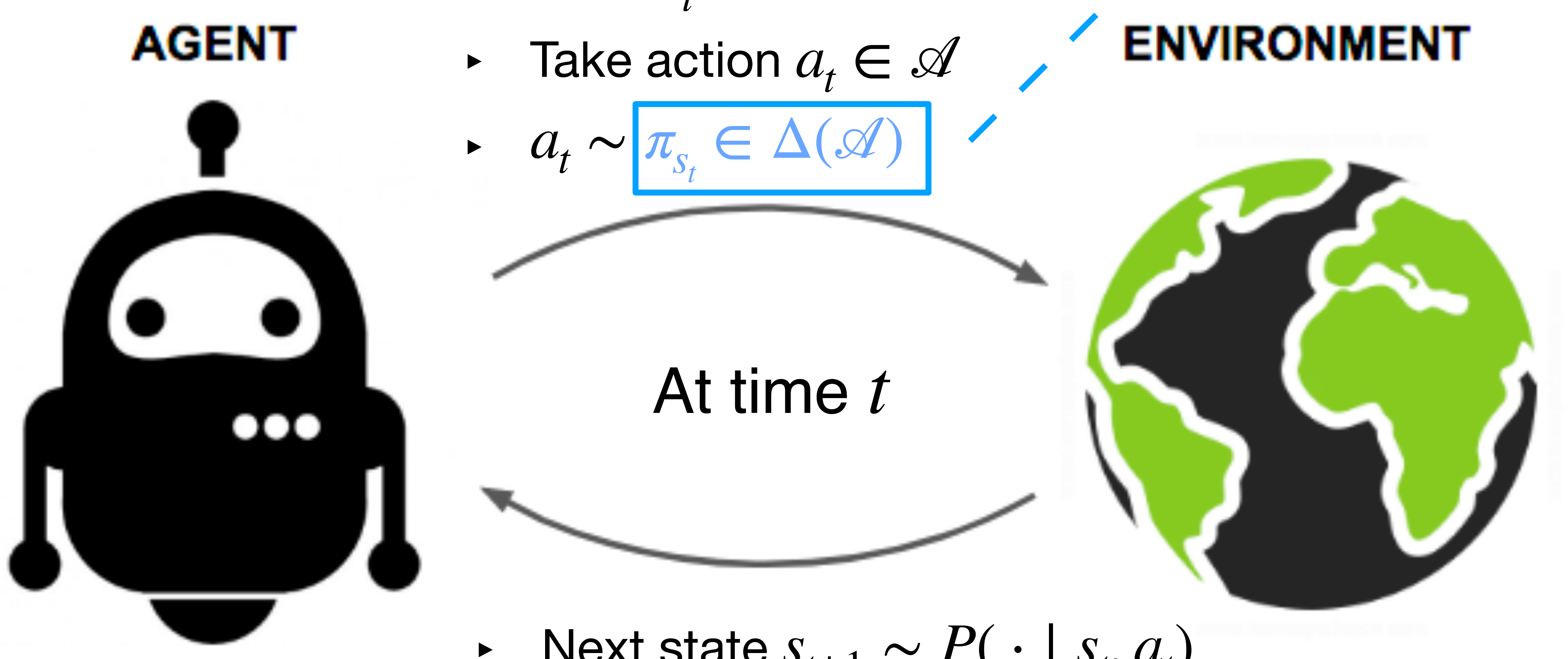
Reinforcement Learning

Sequential decision making problems

Policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$,

$\pi_{s_t,a_t} \in \mathbb{R}$ is the density of the distribution over actions at $s_t \in \mathcal{S}$

- State $s_t \in \mathcal{S}$
- Take action $a_t \in \mathcal{A}$
- $a_t \sim \boxed{\pi_{s_t} \in \Delta(\mathcal{A})}$



- Next state $s_{t+1} \sim P(\cdot \mid s_t, a_t)$
- Get a cost $c(s_t, a_t)$

Markov decision Process (MDP)

- State space $\mathcal{S}$
- Action space $\mathcal{A}$
- Transition probabilities P
- Initial state distribution ρ

Solve an MDP to minimize total expected cost (a.k.a. policy optimization)

$$\arg\min_{\pi} \boxed{V_\rho(\pi) := \mathbb{E}_{\boxed{s_0\sim\rho,}\, a_t\sim\pi_{s_t},\, s_{t+1}\sim P(\cdot|s_t,a_t)} \left[\sum_{t=0}^{\infty} \gamma^t c(s_t, a_t)\right]} \longrightarrow \text{Cost function}$$

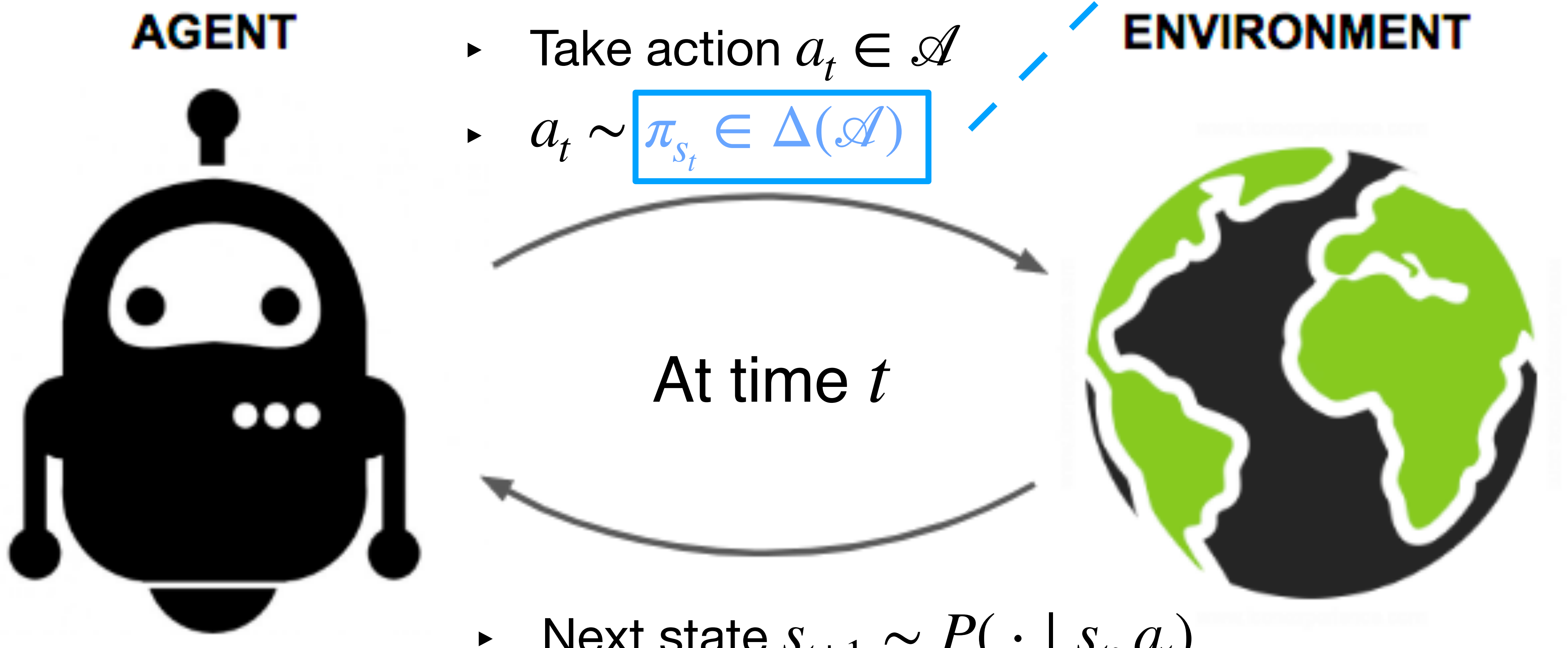
Reinforcement Learning

Sequential decision making problems

Policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$,

$\pi_{s_t,a_t} \in \mathbb{R}$ is the density of the distribution over actions at $s_t \in \mathcal{S}$

- State $s_t \in \mathcal{S}$
- Take action $a_t \in \mathcal{A}$
- $a_t \sim \boxed{\pi_{s_t} \in \Delta(\mathcal{A})}$



- Next state $s_{t+1} \sim P(\cdot \mid s_t, a_t)$
- Get a cost $c(s_t, a_t)$

Markov decision Process (MDP)

- State space $\mathcal{S}$
- Action space $\mathcal{A}$
- Transition probabilities P
- Initial state distribution ρ

Solve an MDP to minimize total expected cost (a.k.a. policy optimization)

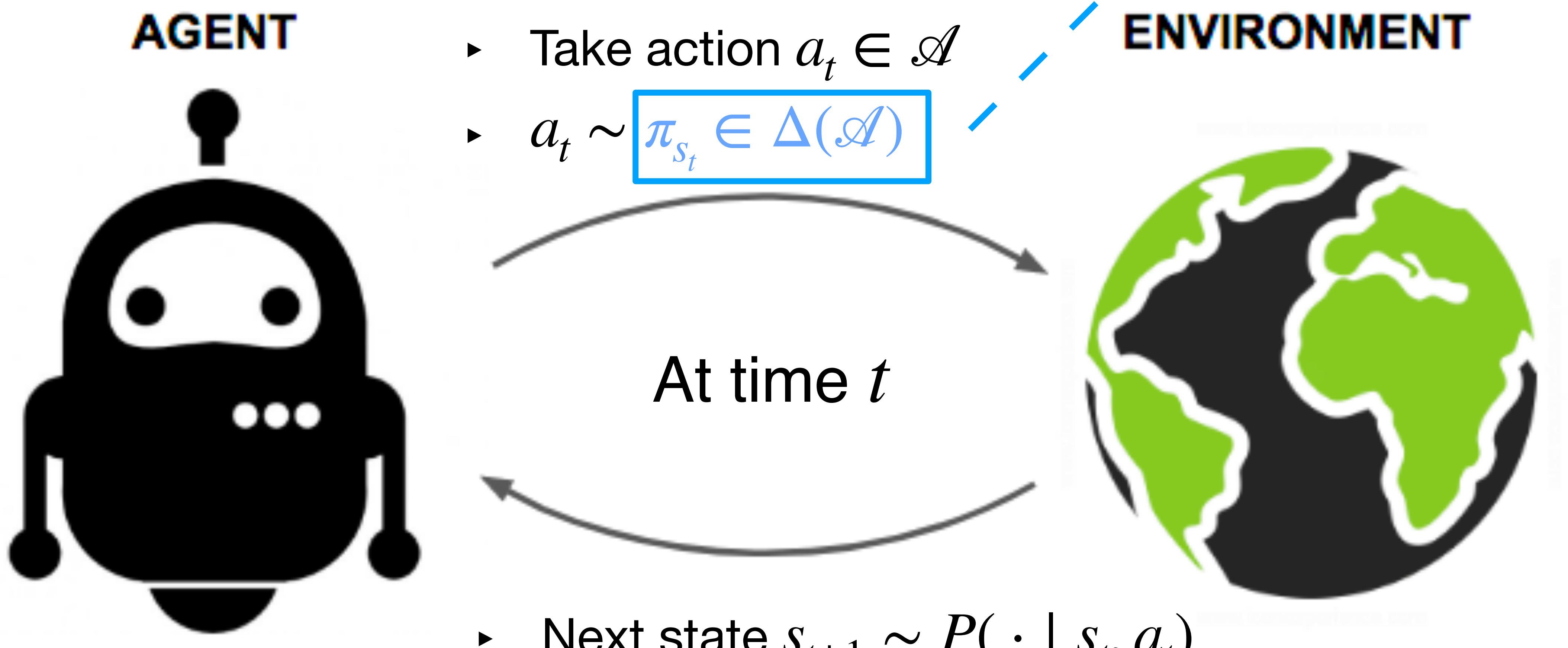
$$\arg\min_{\pi} \boxed{V_\rho(\pi) := \mathbb{E}_{\boxed{s_0\sim\rho,}\, a_t\sim\pi_{s_t},\, s_{t+1}\sim P(\cdot|s_t,a_t)} \left[\sum_{t=0}^{\infty} \boxed{\gamma^t} c(s_t, a_t) \right]} \longrightarrow \text{Cost function}$$

Reinforcement Learning

Sequential decision making problems

Policy $\pi : \mathcal{S} \rightarrow \Delta(\mathcal{A})$,

$\pi_{s_t,a_t} \in \mathbb{R}$ is the density of the distribution over actions at $s_t \in \mathcal{S}$



- State $s_t \in \mathcal{S}$
- Take action $a_t \in \mathcal{A}$
- $a_t \sim \boxed{\pi_{s_t} \in \Delta(\mathcal{A})}$
- Next state $s_{t+1} \sim P(\cdot \mid s_t, a_t)$
- Get a cost $c(s_t, a_t)$

Markov decision Process (MDP)

- State space $\mathcal{S}$
- Action space $\mathcal{A}$
- Transition probabilities P
- Initial state distribution ρ
- Discounted factor $\gamma \in (0,1)$

Solve an MDP to minimize total expected cost (a.k.a. policy optimization)

$$\arg\min_{\pi} \boxed{V_\rho(\pi) := \mathbb{E}_{\boxed{s_0 \sim \rho,}\, a_t \sim \pi_{s_t},\ s_{t+1} \sim P(\cdot|s_t,a_t)} \left[\sum_{t=0}^{\infty} \boxed{\gamma^t} c(s_t, a_t) \right]} \longrightarrow \text{Cost function}$$

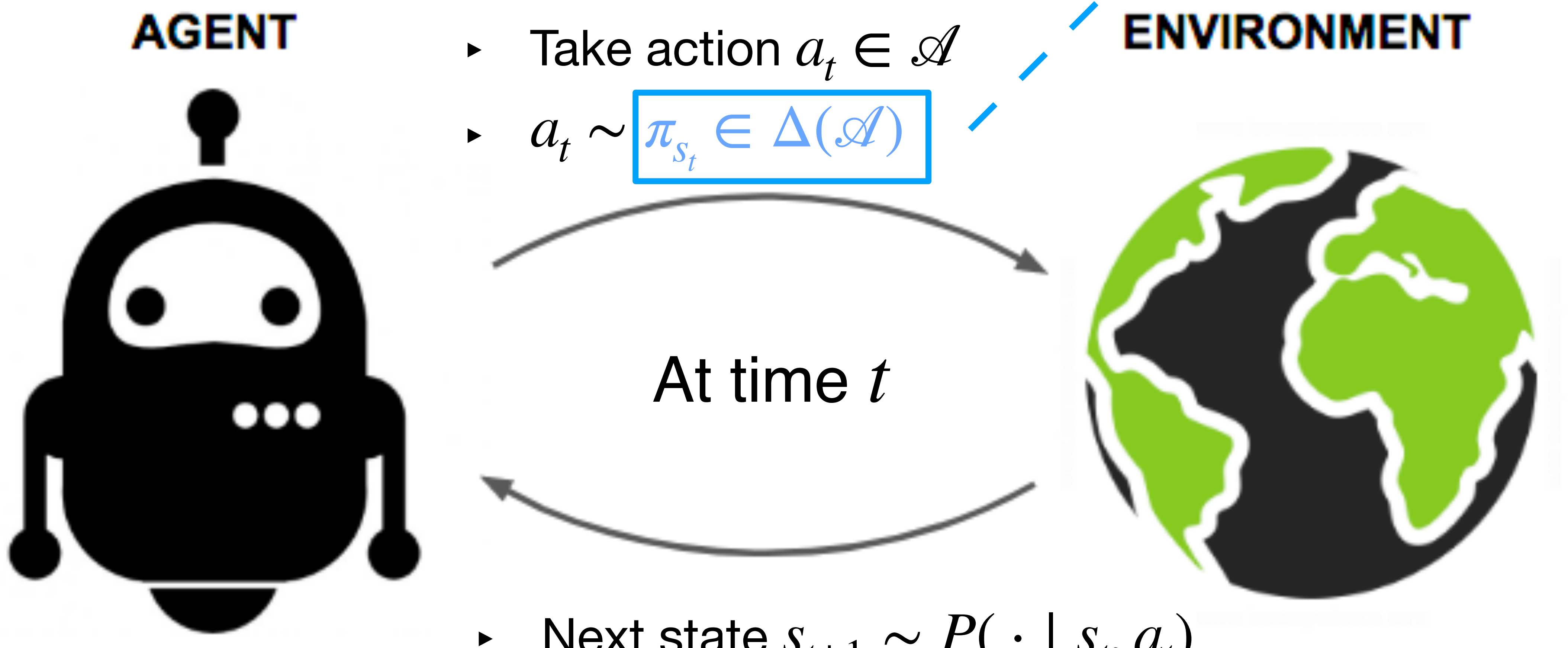
Reinforcement Learning

Sequential decision making problems

Policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$,

$\pi_{s_t,a_t} \in \mathbb{R}$ is the density of the distribution over actions at $s_t \in \mathcal{S}$

- State $s_t \in \mathcal{S}$
- Take action $a_t \in \mathcal{A}$
- $a_t \sim \boxed{\pi_{s_t} \in \Delta(\mathcal{A})}$



- Next state $s_{t+1} \sim P(\cdot \mid s_t, a_t)$
- Get a cost $c(s_t, a_t)$

Markov decision Process (MDP)

- State space $\mathcal{S}$
- Action space $\mathcal{A}$
- Transition probabilities P
- Initial state distribution ρ
- Discounted factor $\gamma \in (0,1)$

Solve an MDP to minimize total expected cost (a.k.a. policy optimization)

$$\arg\min_{\theta\in\mathbb{R}^d} V_\rho(\theta) := \mathbb{E}_{s_0\sim\rho,\ a_t\sim\pi_{s_t}(\theta),\ s_{t+1}\sim P(\cdot|s_t,a_t)}\left[\sum_{t=0}^{\infty}\gamma^t c(s_t,a_t)\right]$$

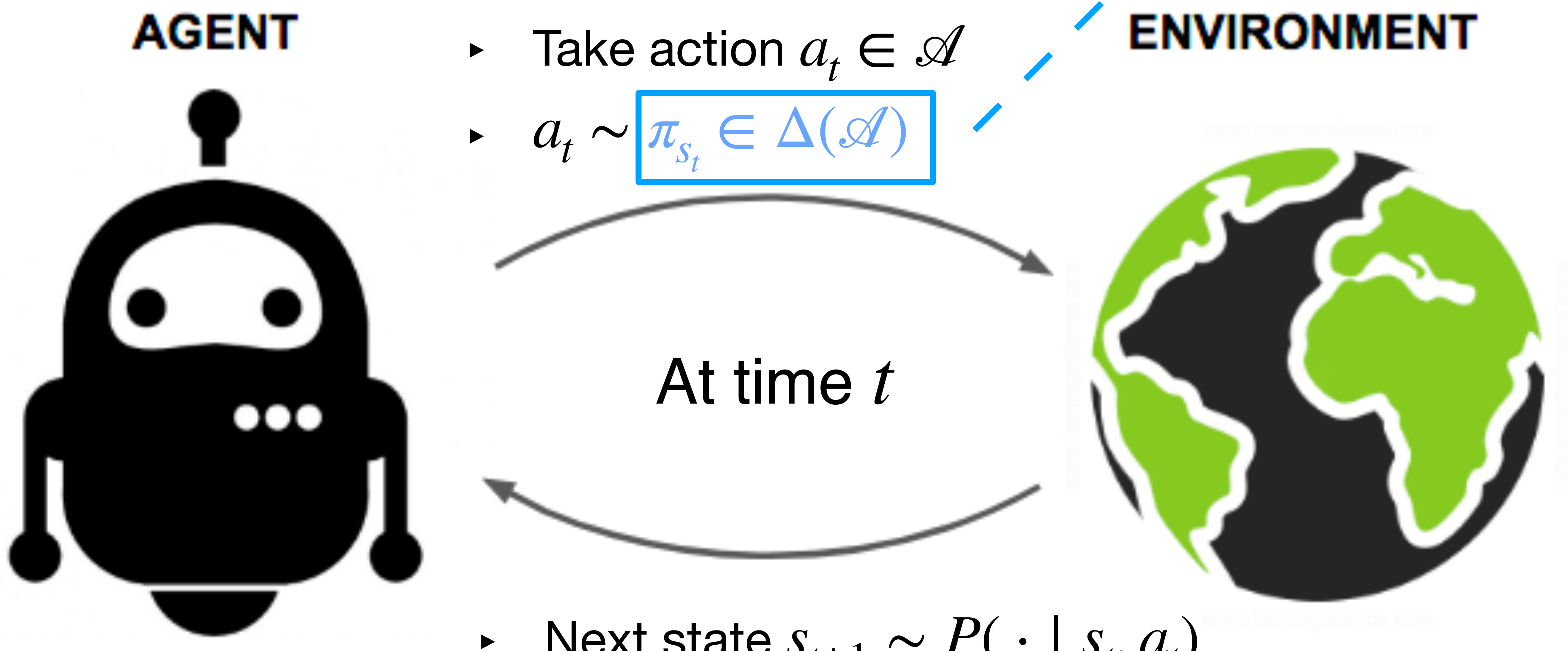
Reinforcement Learning

Sequential decision making problems

Policy $\pi : \mathscr{S} \to \Delta(\mathscr{A})$,

$\pi_{s_t,a_t} \in \mathbb{R}$ is the density of the distribution over actions at $s_t \in \mathscr{S}$

- State $s_t \in \mathscr{S}$
- Take action $a_t \in \mathscr{A}$
- $a_t \sim \boxed{\pi_{s_t} \in \Delta(\mathscr{A})}$



- Next state $s_{t+1} \sim P(\cdot \mid s_t, a_t)$
- Get a cost $c(s_t, a_t)$

Markov decision Process (MDP)

- State space $\mathscr{S}$
- Action space $\mathscr{A}$
- Transition probabilities P
- Initial state distribution ρ
- Discounted factor $\gamma \in (0,1)$

Solve an MDP to minimize total expected cost (a.k.a. policy optimization)

$$\arg\min_{\theta \in \mathbb{R}^d} V_\rho(\theta) := \mathbb{E}_{s_0 \sim \rho,\ \boxed{a_t \sim \pi_{s_t}(\theta),}\ s_{t+1} \sim P(\cdot|s_t,a_t)} \left[\sum_{t=0}^{\infty} \gamma^t c(s_t, a_t) \right]$$

Objective: $\arg\min_{\theta \in \mathbb{R}^d} V_\rho(\theta)$

Policy gradient (PG) methods

Objective: $\arg\min_{\theta \in \mathbb{R}^d} V_\rho(\theta)$

Policy gradient (PG) methods

Objective: $\arg\min_{\theta \in \mathbb{R}^d} V_\rho(\theta)$

- Simplicity

Policy gradient (PG) methods

Objective: $\arg\min_{\theta \in \mathbb{R}^d} V_\rho(\theta)$

- Simplicity
  - Easy to implement and use in practice

Policy gradient (PG) methods

Objective: $\arg\min_{\theta \in \mathbb{R}^d} V_\rho(\theta)$

- Simplicity
  - Easy to implement and use in practice
  - Can solve a wide range of problems (e.g. partially-observable environments)

Policy gradient (PG) methods

Objective: $\arg\min_{\theta \in \mathbb{R}^d} V_\rho(\theta)$

- Simplicity
  - Easy to implement and use in practice
  - Can solve a wide range of problems (e.g. partially-observable environments)
- Versatility

Policy gradient (PG) methods

Objective: $\arg\min_{\theta\in\mathbb{R}^d} V_\rho(\theta)$

- Simplicity
  - Easy to implement and use in practice
  - Can solve a wide range of problems (e.g. partially-observable environments)
- Versatility
  - Actor-critic [Konda and Tsitsiklis, 2000], natural PG[Kakade, 2001], policy mirror descent, etc.

Policy gradient (PG) methods

Objective: $\arg\min_{\theta\in\mathbb{R}^d} V_\rho(\theta)$

- Simplicity
  - Easy to implement and use in practice
  - Can solve a wide range of problems (e.g. partially-observable environments)
- Versatility
  - Actor-critic [Konda and Tsitsiklis, 2000], natural PG[Kakade, 2001], policy mirror descent, etc.
  - Trust-region (e.g. TRPO, PPO [Schulman et al., 2015; 2017]), variance reduction techniques [Papini et al., 2018; Shen et al., 2019; Xu et al., 2020; Huang et al., 2020]

Main challenge about PG methods

Main challenge about PG methods

A solid theoretical understanding of even the “vanilla” PG has long been elusive until recent, and it is messy.

Main challenge about PG methods

A solid theoretical understanding of even the "vanilla" PG has long been elusive until recent, and it is messy.

Unlike value-based methods, sample efficiency in theory lacks for existing gradient-based RL methods.

Vanilla Policy Gradient

Rui Yuan, Robert M. Gower, Alessandro Lazaric
A general sample complexity analysis of vanilla policy gradient, AISTATS, 2022.

Policy gradient methods as gradient descent

Objective: $\arg\min_{\theta \in \mathbb{R}^d} V_\rho(\theta)$

Policy gradient methods as gradient descent

Objective: $\arg\min_{\theta \in \mathbb{R}^d} V_\rho(\theta)$

- PG methods

Policy gradient methods as gradient descent

Objective: $\arg\min_{\theta \in \mathbb{R}^d} V_\rho(\theta)$

- PG methods

$$\theta^{(k+1)} = \theta^{(k)} - \eta_k \nabla_\theta V_\rho(\theta^{(k)})$$

Policy gradient methods as gradient descent

Objective: $\arg\min_{\theta \in \mathbb{R}^d} V_\rho(\theta)$

- PG methods

Step size

$$\theta^{(k+1)} = \theta^{(k)} - \boxed{\eta_k} \nabla_\theta V_\rho(\theta^{(k)})$$

Policy gradient methods as gradient descent

Objective: $\arg\min_{\theta \in \mathbb{R}^d} V_\rho(\theta)$

- PG methods

Step size

$$\theta^{(k+1)} = \theta^{(k)} - \boxed{\eta_k}\boxed{\nabla_\theta V_\rho(\theta^{(k)})}$$

Gradient of $V_\rho(\theta)$

Policy gradient methods as gradient descent

Objective: $\arg\min_{\theta \in \mathbb{R}^d} V_\rho(\theta)$

- PG methods

Step size

$$\theta^{(k+1)} = \theta^{(k)} - \boxed{\eta_k}\boxed{\nabla_\theta V_\rho(\theta^{(k)})}$$

Gradient of $V_\rho(\theta)$

- Compute $\nabla_\theta V_\rho(\theta)$:

Policy gradient methods as gradient descent

Objective: $\arg\min_{\theta\in\mathbb{R}^d} V_\rho(\theta)$

- PG methods

Step size

$$\theta^{(k+1)} = \theta^{(k)} - \boxed{\eta_k}\boxed{\nabla_\theta V_\rho(\theta^{(k)})}$$

Gradient of $V_\rho(\theta)$

- Compute $\nabla_\theta V_\rho(\theta)$: $\nabla_\theta V_\rho(\theta) = \nabla_\theta \mathbb{E}_{s_0\sim\rho,\ a_t\sim\pi_{s_t}(\theta),\ s_{t+1}\sim P(\cdot|s_t,a_t)}\left[\sum_{t=0}^{\infty}\gamma^t c(s_t, a_t)\right]$

Policy gradient methods as gradient descent

Objective: $\arg\min_{\theta\in\mathbb{R}^d} V_\rho(\theta)$

- PG methods

Step size

$$\theta^{(k+1)} = \theta^{(k)} - \boxed{\eta_k}\boxed{\nabla_\theta V_\rho(\theta^{(k)})}$$

Gradient of $V_\rho(\theta)$

- Compute $\nabla_\theta V_\rho(\theta)$: $\nabla_\theta V_\rho(\theta) = \nabla_\theta \mathbb{E}_{\boxed{s_0\sim\rho,\ a_t\sim\pi_{s_t}(\theta),\ s_{t+1}\sim P(\cdot|s_t,a_t)}}\left[\sum_{t=0}^{\infty}\gamma^t c(s_t, a_t)\right]$

Policy gradient methods as gradient descent

Objective: $\arg\min_{\theta\in\mathbb{R}^d} V_\rho(\theta)$

- PG methods

Step size

$$\theta^{(k+1)} = \theta^{(k)} - \boxed{\eta_k}\boxed{\nabla_\theta V_\rho(\theta^{(k)})}$$

Gradient of $V_\rho(\theta)$

- Compute $\nabla_\theta V_\rho(\theta)$:

$$\nabla_\theta V_\rho(\theta) = \nabla_\theta \mathbb{E}_{\boxed{s_0\sim\rho,\ a_t\sim\pi_{s_t}(\theta),\ s_{t+1}\sim P(\cdot|s_t,a_t)}}\left[\sum_{t=0}^{\infty}\gamma^t c(s_t, a_t)\right]$$

Trajectory $\tau = (s_0, a_1, s_1, a_1, \cdots)$

Policy gradient methods as gradient descent

Objective: $\arg\min_{\theta\in\mathbb{R}^d} V_\rho(\theta)$

- PG methods

Step size

$$\theta^{(k+1)} = \theta^{(k)} - \boxed{\eta_k}\boxed{\nabla_\theta V_\rho(\theta^{(k)})}$$

Gradient of $V_\rho(\theta)$

- Compute $\nabla_\theta V_\rho(\theta)$:

$$\nabla_\theta V_\rho(\theta) = \nabla_\theta \mathbb{E}_{\boxed{s_0\sim\rho,\ a_t\sim\pi_{s_t}(\theta),\ s_{t+1}\sim P(\cdot|s_t,a_t)}}\left[\sum_{t=0}^{\infty}\gamma^t c(s_t,a_t)\right]$$

Trajectory $\tau = (s_0, a_1, s_1, a_1, \cdots)$

Probability of sampling a trajectory τ:

$$p(\tau \mid \theta) = \rho(s_0)\Pi_{t'=0}^{\infty}\pi_{s_{t'},a_{t'}}(\theta)P(s_{t'+1}\,|\,s_{t'},a_{t'})$$

Policy gradient methods as gradient descent

Objective: $\arg\min_{\theta\in\mathbb{R}^d} V_\rho(\theta)$

- PG methods

Step size

$$\theta^{(k+1)} = \theta^{(k)} - \boxed{\eta_k}\boxed{\nabla_\theta V_\rho(\theta^{(k)})}$$

Gradient of $V_\rho(\theta)$

- Compute $\nabla_\theta V_\rho(\theta)$:

$$\nabla_\theta V_\rho(\theta) = \nabla_\theta \mathbb{E}_{\boxed{s_0\sim\rho,\ a_t\sim\pi_{s_t}(\theta),\ s_{t+1}\sim P(\cdot|s_t,a_t)}}\left[\sum_{t=0}^{\infty}\gamma^t c(s_t,a_t)\right]$$

$$= \int \left(\sum_{t=0}^{\infty}\gamma^t c(s_t,a_t)\right)\nabla_\theta p(\tau\mid\theta)d\tau$$

Trajectory $\tau = (s_0, a_1, s_1, a_1, \cdots)$

Probability of sampling a trajectory τ:

$p(\tau\mid\theta) = \rho(s_0)\Pi_{t'=0}^{\infty}\pi_{s_{t'},a_{t'}}(\theta)P(s_{t'+1}\,|\,s_{t'},a_{t'})$

Policy gradient methods as gradient descent

Objective: $\arg\min_{\theta\in\mathbb{R}^d} V_\rho(\theta)$

- PG methods

Step size

$$\theta^{(k+1)} = \theta^{(k)} - \boxed{\eta_k}\boxed{\nabla_\theta V_\rho(\theta^{(k)})}$$

Gradient of $V_\rho(\theta)$

- Compute $\nabla_\theta V_\rho(\theta)$:

$$\nabla_\theta V_\rho(\theta) = \nabla_\theta \mathbb{E}_{\boxed{s_0\sim\rho,\ a_t\sim\pi_{s_t}(\theta),\ s_{t+1}\sim P(\cdot|s_t,a_t)}}\left[\sum_{t=0}^{\infty}\gamma^t c(s_t,a_t)\right]$$

$$= \int\left(\sum_{t=0}^{\infty}\gamma^t c(s_t,a_t)\right)\nabla_\theta p(\tau \mid \theta)d\tau$$

$$= \int\left(\sum_{t=0}^{\infty}\gamma^t c(s_t,a_t)\right)(\nabla_\theta p(\tau \mid \theta)/p(\tau \mid \theta))p(\tau \mid \theta)d\tau$$

Trajectory $\tau = (s_0, a_1, s_1, a_1, \cdots)$

Probability of sampling a trajectory τ:

$p(\tau \mid \theta) = \rho(s_0)\Pi_{t'=0}^{\infty}\pi_{s_{t'},a_{t'}}(\theta)P(s_{t'+1}\,|\,s_{t'},a_{t'})$

Policy gradient methods as gradient descent

Objective: $\arg\min_{\theta \in \mathbb{R}^d} V_\rho(\theta)$

- PG methods

Step size

$$\theta^{(k+1)} = \theta^{(k)} - \boxed{\eta_k} \boxed{\nabla_\theta V_\rho(\theta^{(k)})}$$

Gradient of $V_\rho(\theta)$

- Compute $\nabla_\theta V_\rho(\theta)$:

$$\nabla_\theta V_\rho(\theta) = \nabla_\theta \mathbb{E}_{\boxed{s_0 \sim \rho,\ a_t \sim \pi_{s_t}(\theta),\ s_{t+1} \sim P(\cdot|s_t, a_t)}} \left[\sum_{t=0}^{\infty} \gamma^t c(s_t, a_t) \right]$$

$$= \int \left(\sum_{t=0}^{\infty} \gamma^t c(s_t, a_t) \right) \nabla_\theta p(\tau \mid \theta) d\tau$$

$$= \int \left(\sum_{t=0}^{\infty} \gamma^t c(s_t, a_t) \right) (\nabla_\theta p(\tau \mid \theta) / p(\tau \mid \theta)) p(\tau \mid \theta) d\tau$$

$$= \mathbb{E}_{p(\tau|\theta)} \left[\left(\sum_{t=0}^{\infty} \gamma^t c(s_t, a_t) \right) \nabla_\theta \log p(\tau \mid \theta) \right]$$

Trajectory $\tau = (s_0, a_1, s_1, a_1, \cdots)$

Probability of sampling a trajectory τ:

$p(\tau \mid \theta) = \rho(s_0) \Pi_{t'=0}^{\infty} \pi_{s_{t'}, a_{t'}}(\theta) P(s_{t'+1} \,|\, s_{t'}, a_{t'})$

Policy gradient methods as gradient descent

Objective: $\arg\min_{\theta\in\mathbb{R}^d} V_\rho(\theta)$

- PG methods

Step size

$$\theta^{(k+1)} = \theta^{(k)} - \boxed{\eta_k}\boxed{\nabla_\theta V_\rho(\theta^{(k)})}$$

Gradient of $V_\rho(\theta)$

- Compute $\nabla_\theta V_\rho(\theta)$:

$$\nabla_\theta V_\rho(\theta) = \nabla_\theta \mathbb{E}_{\boxed{s_0\sim\rho,\ a_t\sim\pi_{s_t}(\theta),\ s_{t+1}\sim P(\cdot|s_t,a_t)}}\left[\sum_{t=0}^{\infty} \gamma^t c(s_t, a_t)\right]$$

$$= \int \left(\sum_{t=0}^{\infty} \gamma^t c(s_t, a_t)\right) \nabla_\theta p(\tau \mid \theta) d\tau$$

$$= \int \left(\sum_{t=0}^{\infty} \gamma^t c(s_t, a_t)\right) \underbrace{(\nabla_\theta p(\tau \mid \theta)/p(\tau \mid \theta))} p(\tau \mid \theta) d\tau$$

$$= \mathbb{E}_{p(\tau|\theta)}\left[\left(\sum_{t=0}^{\infty} \gamma^t c(s_t, a_t)\right) \nabla_\theta \log p(\tau \mid \theta)\right]$$

Trajectory $\tau = (s_0, a_1, s_1, a_1, \cdots)$

Probability of sampling a trajectory τ:

$$p(\tau \mid \theta) = \rho(s_0)\Pi_{t'=0}^{\infty}\pi_{s_{t'},a_{t'}}(\theta)P(s_{t'+1} \,|\, s_{t'}, a_{t'})$$

Policy gradient methods as gradient descent

Objective: $\arg\min_{\theta \in \mathbb{R}^d} V_\rho(\theta)$

- PG methods

Step size

$$\theta^{(k+1)} = \theta^{(k)} - \boxed{\eta_k} \boxed{\nabla_\theta V_\rho(\theta^{(k)})}$$

Gradient of $V_\rho(\theta)$

- Compute $\nabla_\theta V_\rho(\theta)$:

$$\nabla_\theta V_\rho(\theta) = \nabla_\theta \mathbb{E}_{\boxed{s_0 \sim \rho,\ a_t \sim \pi_{s_t}(\theta),\ s_{t+1} \sim P(\cdot|s_t, a_t)}} \left[\sum_{t=0}^{\infty} \gamma^t c(s_t, a_t) \right]$$

$$= \int \left(\sum_{t=0}^{\infty} \gamma^t c(s_t, a_t) \right) \nabla_\theta p(\tau \mid \theta) d\tau$$

$$= \int \left(\sum_{t=0}^{\infty} \gamma^t c(s_t, a_t) \right) \underbrace{(\nabla_\theta p(\tau \mid \theta) / p(\tau \mid \theta))}_{} p(\tau \mid \theta) d\tau$$

$$= \mathbb{E}_{p(\tau|\theta)} \left[\left(\sum_{t=0}^{\infty} \gamma^t c(s_t, a_t) \right) \nabla_\theta \log p(\tau \mid \theta) \right]$$

Trajectory $\tau = (s_0, a_1, s_1, a_1, \cdots)$

Probability of sampling a trajectory τ:

$p(\tau \mid \theta) = \rho(s_0) \Pi_{t'=0}^{\infty} \pi_{s_{t'}, a_{t'}}(\theta) P(s_{t'+1} \,|\, s_{t'}, a_{t'})$

Policy gradient methods as gradient descent

Objective: $\arg\min_{\theta\in\mathbb{R}^d} V_\rho(\theta)$

- PG methods

Step size

$$\theta^{(k+1)} = \theta^{(k)} - \boxed{\eta_k}\boxed{\nabla_\theta V_\rho(\theta^{(k)})}$$

Gradient of $V_\rho(\theta)$

- Compute $\nabla_\theta V_\rho(\theta)$:

$$\nabla_\theta V_\rho(\theta) = \nabla_\theta \mathbb{E}_{\boxed{s_0\sim\rho,\ a_t\sim\pi_{s_t}(\theta),\ s_{t+1}\sim P(\cdot|s_t,a_t)}}\left[\sum_{t=0}^{\infty}\gamma^t c(s_t,a_t)\right]$$

$$= \int\left(\sum_{t=0}^{\infty}\gamma^t c(s_t,a_t)\right)\nabla_\theta p(\tau\mid\theta)d\tau$$

$$= \int\left(\sum_{t=0}^{\infty}\gamma^t c(s_t,a_t)\right)\underbrace{(\nabla_\theta p(\tau\mid\theta)/p(\tau\mid\theta))}p(\tau\mid\theta)d\tau$$

$$= \mathbb{E}_{p(\tau|\theta)}\left[\left(\sum_{t=0}^{\infty}\gamma^t c(s_t,a_t)\right)\nabla_\theta \log p(\tau\mid\theta)\right]$$

$$= \mathbb{E}_{p(\tau|\theta)}\left[\sum_{t=0}^{\infty}\gamma^t c(s_t,a_t)\sum_{t'=0}^{\infty}\nabla_\theta \log \pi_{s_{t'},a_{t'}}(\theta)\right]$$

Trajectory $\tau = (s_0, a_1, s_1, a_1, \cdots)$

Probability of sampling a trajectory τ:

$p(\tau\mid\theta) = \rho(s_0)\Pi_{t'=0}^{\infty}\pi_{s_{t'},a_{t'}}(\theta)P(s_{t'+1}\,|\,s_{t'},a_{t'})$

Vanilla policy gradient

Vanilla policy gradient

- Recall $\nabla_\theta V_\rho(\theta) = \mathbb{E}_{p(\tau|\theta)}\left[\sum_{t=0}^{\infty} \gamma^t c(s_t, a_t) \sum_{t'=0}^{\infty} \nabla_\theta \log \pi_{s_{t'}, a_{t'}}(\theta)\right]$

Vanilla policy gradient

- Recall $\nabla_\theta V_\rho(\theta) = \mathbb{E}_{p(\tau|\theta)}\left[\sum_{t=0}^{\infty}\gamma^t c(s_t, a_t)\sum_{t'=0}^{\infty}\nabla_\theta \log \pi_{s_{t'},a_{t'}}(\theta)\right]$

- Compute an empirical estimator of the gradient by sampling m truncated trajectories $\tau = \left(s_0, a_0, s_1, a_1, \cdots, s_{H-1}, a_{H-1}\right)$

Vanilla policy gradient

- Recall $\nabla_\theta V_\rho(\theta) = \mathbb{E}_{p(\tau|\theta)}\left[\sum_{t=0}^{\infty} \gamma^t c(s_t, a_t) \sum_{t'=0}^{\infty} \nabla_\theta \log \pi_{s_{t'}, a_{t'}}(\theta)\right]$

- Compute an empirical estimator of the gradient by sampling m truncated trajectories $\tau = \left(s_0, a_0, s_1, a_1, \cdots, s_{H-1}, a_{H-1}\right)$

$$\hat{\nabla}_m V_\rho(\theta) := \frac{1}{m} \sum_{i=1}^{m} \sum_{t=0}^{H-1} \gamma^t c(s_t^i, a_t^i) \cdot \sum_{t'=0}^{H-1} \nabla_\theta \log \pi_{s_{t'}^i, a_{t'}^i}(\theta)$$

Vanilla policy gradient

- Recall $\nabla_\theta V_\rho(\theta) = \mathbb{E}_{p(\tau|\theta)}\left[\sum_{t=0}^{\infty}\gamma^t c(s_t, a_t)\sum_{t'=0}^{\infty}\nabla_\theta \log \pi_{s_{t'},a_{t'}}(\theta)\right]$

- Compute an empirical estimator of the gradient by sampling m truncated trajectories $\tau = \left(s_0, a_0, s_1, a_1, \cdots, s_{H-1}, a_{H-1}\right)$

$$\hat{\nabla}_m V_\rho(\theta) := \frac{1}{m}\sum_{i=1}^{m}\sum_{t=0}^{H-1}\gamma^t c(s_t^i, a_t^i)\cdot\sum_{t'=0}^{H-1}\nabla_\theta \log \pi_{s_{t'}^i,a_{t'}^i}(\theta)$$

- Vanilla PG (REINFORCE [Williams, 1992], GPOMDP [Baxter and Bartlett, 2001])

$$\theta^{(k+1)} = \theta^{(k)} - \eta\,\hat{\nabla}_m V_\rho(\theta^{(k)})$$

Current literatures of vanilla PG: *fragmentary* !

Current literatures of vanilla PG: *fragmentary* !

- Exact PG [Agarwal et al., 2019, Zhang et al., 2020a, Mei et al., 2020] vs stochastic PG [Papini et al., 2019, Liu et al., 2020, Zhang et al., 2020c, Xiong et al., 2021]

Current literatures of vanilla PG: *fragmentary* !

- Exact PG [Agarwal et al., 2019, Zhang et al., 2020a, Mei et al., 2020] vs stochastic PG [Papini et al., 2019, Liu et al., 2020, Zhang et al., 2020c, Xiong et al., 2021]

- Different criteria of the convergence results: first-order stationary point [Papini et al., 2019, Zhang et al., 2020c], global optimum [Agarwal et al., 2019, Zhang et al., 2020a, Mei et al., 2020], average regret to the global optimum [Zhang et al., 2020b, Liu et al., 2020]

Current literatures of vanilla PG: *fragmentary* !

- Exact PG [Agarwal et al., 2019, Zhang et al., 2020a, Mei et al., 2020] vs stochastic PG [Papini et al., 2019, Liu et al., 2020, Zhang et al., 2020c, Xiong et al., 2021]

- Different criteria of the convergence results: first-order stationary point [Papini et al., 2019, Zhang et al., 2020c], global optimum [Agarwal et al., 2019, Zhang et al., 2020a, Mei et al., 2020], average regret to the global optimum [Zhang et al., 2020b, Liu et al., 2020]

- Different RL settings: softmax tabular policy w/o different regularizations [Agarwal et al., 2019, Zhang et al., 2020a,b, Mei et al., 2020], Fisher-non-degenerate policy [Liu et al., 2020, Ding et al., 2021]

Current literatures of vanilla PG: *fragmentary* !

- Exact PG [Agarwal et al., 2019, Zhang et al., 2020a, Mei et al., 2020] vs stochastic PG [Papini et al., 2019, Liu et al., 2020, Zhang et al., 2020c, Xiong et al., 2021]

- Different criteria of the convergence results: first-order stationary point [Papini et al., 2019, Zhang et al., 2020c], global optimum [Agarwal et al., 2019, Zhang et al., 2020a, Mei et al., 2020], average regret to the global optimum [Zhang et al., 2020b, Liu et al., 2020]

- Different RL settings: softmax tabular policy w/o different regularizations [Agarwal et al., 2019, Zhang et al., 2020a,b, Mei et al., 2020], Fisher-non-degenerate policy [Liu et al., 2020, Ding et al., 2021]

- Different assumptions: Lipschitz and smooth policy [Liu et al., 2020, Zhang et al., 2020c, Xiong et al., 2021], bijection between the primal and the dual space [Zhang et al., 2020a]

Current literatures of vanilla PG: *fragmentary* !

- Exact PG [Agarwal et al., 2019, Zhang et al., 2020a, Mei et al., 2020] vs stochastic PG [Papini et al., 2019, Liu et al., 2020, Zhang et al., 2020c, Xiong et al., 2021]

- Different criteria of the convergence results: first-order stationary point [Papini et al., 2019, Zhang et al., 2020c], global optimum [Agarwal et al., 2019, Zhang et al., 2020a, Mei et al., 2020], average regret to the global optimum [Zhang et al., 2020b, Liu et al., 2020]

- Different RL settings: softmax tabular policy w/o different regularizations [Agarwal et al., 2019, Zhang et al., 2020a,b, Mei et al., 2020], Fisher-non-degenerate policy [Liu et al., 2020, Ding et al., 2021]

- Different assumptions: Lipschitz and smooth policy [Liu et al., 2020, Zhang et al., 2020c, Xiong et al., 2021], bijection between the primal and the dual space [Zhang et al., 2020a]

- Large mini-batch (e.g. $O(\epsilon^{-1})$, $O(\epsilon^{-2})$) per iteration for stochastic updates [Papini et al., 2019, Liu et al., 2020, Zhang et al., 2020c, Xiong et al., 2021]

Contribution

Contribution

- A general PG analysis with weaker assumptions

Contribution

- A general PG analysis with weaker assumptions
  - Unify much of the fragmented results in the literature under one guise without lost of the performance.

Contribution

- A general PG analysis with weaker assumptions
  - Unify much of the fragmented results in the literature under one guise without lost of the performance.
  - Recover existing $O(\epsilon^{-4})$ sample complexity guarantees with weaker assumptions for *wider ranges* of parameters (e.g. mini-batch m from 1 to $O(\epsilon^{-2})$)

Contribution

- A general PG analysis with weaker assumptions
  - Unify much of the fragmented results in the literature under one guise without lost of the performance.
  - Recover existing $O(\epsilon^{-4})$ sample complexity guarantees with weaker assumptions for *wider ranges* of parameters (e.g. mini-batch m from 1 to $O(\epsilon^{-2})$)
- New $O(\epsilon^{-3})$ sample complexity for global optimum guarantees with additional relaxed weak gradient domination assumption, including Fisher-non-degenerate parametrized policies as special case

Contribution

- A general PG analysis with weaker assumptions
  - Unify much of the fragmented results in the literature under one guise without lost of the performance.
  - Recover existing $O(\epsilon^{-4})$ sample complexity guarantees with weaker assumptions for *wider ranges* of parameters (e.g. mini-batch m from 1 to $O(\epsilon^{-2})$)
- New $O(\epsilon^{-3})$ sample complexity for global optimum guarantees with additional relaxed weak gradient domination assumption, including Fisher-non-degenerate parametrized policies as special case

Main assumption: ABC Assumption

[Khaled and Richtárik, 2020]

Main assumption: ABC Assumption

[Khaled and Richtárik, 2020]

- We assume that, for some $A, B, C \geq 0$ and all $\theta \in \mathbb{R}^d$, the stochastic gradient satisfies

Main assumption: ABC Assumption

📗 [Khaled and Richtárik, 2020]

- We assume that, for some $A, B, C \geq 0$ and all $\theta \in \mathbb{R}^d$, the stochastic gradient satisfies

$$\mathbb{E}\left[\|\hat{\nabla}_m V_\rho(\theta)\|^2\right] \leq 2A(V_\rho(\theta) - V^*) + B\|\nabla V_\rho^H(\theta)\|^2 + C$$

Main assumption: ABC Assumption

📗 [Khaled and Richtárik, 2020]

- We assume that, for some $A, B, C \geq 0$ and all $\theta \in \mathbb{R}^d$, the stochastic gradient satisfies

$$\mathbb{E}\left[\|\hat{\nabla}_m V_\rho(\theta)\|^2\right] \leq 2A(V_\rho(\theta) - V^*) + B\|\nabla V_\rho^H(\theta)\|^2 + C$$

Here V^* is the optimum cost function.

Main assumption: ABC Assumption

📗 [Khaled and Richtárik, 2020]

- We assume that, for some $A, B, C \geq 0$ and all $\theta \in \mathbb{R}^d$, the stochastic gradient satisfies

$$\mathbb{E}\left[\|\hat{\nabla}_m V_\rho(\theta)\|^2\right] \leq 2A(V_\rho(\theta) - V^*) + B\|\nabla V_\rho^H(\theta)\|^2 + C$$

Here V^* is the optimum cost function.

$V_\rho^H(\theta) = \mathbb{E}\left[\sum_{t=0}^{H-1} \gamma^t c(s_t, a_t)\right]$ is the expected truncated total cost function.

Main assumption: ABC Assumption

[Khaled and Richtárik, 2020]

- We assume that, for some $A, B, C \geq 0$ and all $\theta \in \mathbb{R}^d$, the stochastic gradient satisfies

$$\mathbb{E}\left[\|\hat{\nabla}_m V_\rho(\theta)\|^2\right] \leq 2A\underbrace{(V_\rho(\theta) - V^*)}_{\text{Suboptimality gap}}+B\|\nabla V_\rho^H(\theta)\|^2 + C$$

Here V^* is the optimum cost function.

$V_\rho^H(\theta) = \mathbb{E}\left[\sum_{t=0}^{H-1} \gamma^t c(s_t, a_t)\right]$is the expected truncated total cost function.

Main assumption: ABC Assumption

[Khaled and Richtárik, 2020]

- We assume that, for some $A, B, C \geq 0$ and all $\theta \in \mathbb{R}^d$, the stochastic gradient satisfies

$$\mathbb{E}\left[\|\hat{\nabla}_m V_\rho(\theta)\|^2\right] \leq 2A(\underbrace{V_\rho(\theta) - V^*}_{\text{Suboptimality gap}}) + B\underbrace{\|\nabla V_\rho^H(\theta)\|^2}_{\text{Exact gradient}} + C$$

Here V^* is the optimum cost function.

$V_\rho^H(\theta) = \mathbb{E}\left[\sum_{t=0}^{H-1} \gamma^t c(s_t, a_t)\right]$ is the expected truncated total cost function.

Simple examples of ABC Assumption

ABC Assumption : $\mathbb{E}\left[\|\hat{\nabla}_m V_\rho(\theta)\|^2\right] \leq 2A(V_\rho(\theta) - V^*) + B\|\nabla V_\rho^H(\theta)\|^2 + C$

Simple examples of ABC Assumption

ABC Assumption : $\mathbb{E}\left[\|\hat{\nabla}_m V_\rho(\theta)\|^2\right] \leq 2A(V_\rho(\theta) - V*)+B\|\nabla V_\rho^H(\theta)\|^2 + C$

- If $H = m = \infty$, then ABC Assumption holds with the exact gradient. That is,

Simple examples of ABC Assumption

ABC Assumption : $\mathbb{E}\left[\|\hat{\nabla}_m V_\rho(\theta)\|^2\right] \leq 2A(V_\rho(\theta) - V*)+B\|\nabla V_\rho^H(\theta)\|^2 + C$

- If $H = m = \infty$, then ABC Assumption holds with the exact gradient. That is,

$$\hat{\nabla}_m V_\rho(\theta) = \nabla V_\rho(\theta), \quad \text{and} \quad A = C = 0, \quad B = 1;$$

Simple examples of ABC Assumption

ABC Assumption : $\mathbb{E}\left[\|\hat{\nabla}_m V_\rho(\theta)\|^2\right] \leq 2A(V_\rho(\theta) - V^*) + B\|\nabla V_\rho^H(\theta)\|^2 + C$

- If $H = m = \infty$, then ABC Assumption holds with the exact gradient. That is,

$$\hat{\nabla}_m V_\rho(\theta) = \nabla V_\rho(\theta), \quad \text{and} \quad A = C = 0, \quad B = 1;$$

- If $A = 0$ and $B = 1$, then ABC Assumption recovers the bounded variance of the stochastic gradient assumption [Ghadimi and Lan, 2013].

Simple examples of ABC Assumption

ABC Assumption : $\mathbb{E}\left[\|\hat{\nabla}_m V_\rho(\theta)\|^2\right] \leq 2A(V_\rho(\theta) - V^*) + B\|\nabla V_\rho^H(\theta)\|^2 + C$

- If $H = m = \infty$, then ABC Assumption holds with the exact gradient. That is,

$$\hat{\nabla}_m V_\rho(\theta) = \nabla V_\rho(\theta), \quad \text{and} \quad A = C = 0, \quad B = 1;$$

- If $A = 0$ and $B = 1$, then ABC Assumption recovers the bounded variance of the stochastic gradient assumption [Ghadimi and Lan, 2013].

$$\mathbb{E}\left[\|\nabla V_\rho^H(\theta) - \hat{\nabla}_m V_\rho(\theta)\|^2\right] \leq C$$

Simple examples of ABC Assumption

ABC Assumption : $\mathbb{E}\left[\|\hat{\nabla}_m V_\rho(\theta)\|^2\right] \leq 2A(V_\rho(\theta) - V^*) + B\|\nabla V_\rho^H(\theta)\|^2 + C$

- If $H = m = \infty$, then ABC Assumption holds with the exact gradient. That is,

$$\hat{\nabla}_m V_\rho(\theta) = \nabla V_\rho(\theta), \quad \text{and} \quad A = C = 0, \quad B = 1;$$

- If $A = 0$ and $B = 1$, then ABC Assumption recovers the bounded variance of the stochastic gradient assumption [Ghadimi and Lan, 2013].

$$\mathbb{E}\left[\|\nabla V_\rho^H(\theta) - \hat{\nabla}_m V_\rho(\theta)\|^2\right] \leq C$$

$$\Longrightarrow \quad \mathbb{E}\left[\|\hat{\nabla}_m V_\rho(\theta)\|^2\right] \leq \|\nabla V_\rho^H(\theta)\|^2 + C$$

Sample complexity under ABC Assumption

Sample complexity under ABC Assumption

- With a set of parameters (η, K, H), first-order stationary point convergence:

Sample complexity under ABC Assumption

- With a set of parameters (η, K, H), first-order stationary point convergence:

$$\min_{0 \le k \le K-1} \mathbb{E}\left[\|\nabla V_\rho(\theta^{(k)})\|^2\right] = O(\epsilon^2)$$

Sample complexity under ABC Assumption

- With a set of parameters (η, K, H), first-order stationary point convergence:

$$\min_{0 \le k \le \boxed{K-1}} \mathbb{E}\left[\|\nabla V_\rho(\theta^{(k)})\|^2\right] = O(\epsilon^2)$$

Total number of iterations

Sample complexity under ABC Assumption

- With a set of parameters (η, K, H), first-order stationary point convergence:

$$\min_{0\le k\le \boxed{K-1}} \mathbb{E}\left[\|\nabla V_\rho(\theta^{(k)})\|^2\right] = O(\epsilon^2)$$

Total number of iterations

- Sample complexity (i.e., single step interaction (s_t, a_t) with the environment among single sampled trajectory per iteration): $KH = \tilde{O}(\epsilon^{-4})$

Sample complexity under ABC Assumption

- With a set of parameters (η, K, H), first-order stationary point convergence:

$$\min_{0\leq k\leq \boxed{K-1}} \mathbb{E}\left[\|\nabla V_\rho(\theta^{(k)})\|^2\right] = O(\epsilon^2)$$

Total number of iterations

- Sample complexity (i.e., single step interaction (s_t, a_t) with the environment among single sampled trajectory per iteration): $KH = \tilde{O}(\epsilon^{-4})$

- For the exact PG ($A = C = 0,\ B = 1$ and $H = \infty$): $K = O(\epsilon^{-2})$

Applications

Applications

- Different settings that satisfy ABC Assumption

Applications

- Different settings that satisfy ABC Assumption
    - Softmax with log barrier regularization

Applications

- Different settings that satisfy ABC Assumption
  - Softmax with log barrier regularization
  - Softmax with entropy regularization

Applications

- Different settings that satisfy ABC Assumption
  - Softmax with log barrier regularization
  - Softmax with entropy regularization
  - Expected Lipschitz and smooth policy (Gaussian and softmax policies)

Expected Lipschitz and smooth (E-LS) policy

📗 [Papini et al., 2019] (Gaussian and softmax policies satisfy E-LS)

Expected Lipschitz and smooth (E-LS) policy

📗 [Papini et al., 2019] (Gaussian and softmax policies satisfy E-LS)

- There exists constants G, F > 0 such that for each state $s \in \mathcal{S}$, we have

Expected Lipschitz and smooth (E-LS) policy

📗 [Papini et al., 2019] (Gaussian and softmax policies satisfy E-LS)

- There exists constants G, F > 0 such that for each state $s \in \mathcal{S}$, we have

$$\mathbb{E}_{a \sim \pi_s(\theta)}\left[\|\nabla_\theta \log \pi_{s,a}(\theta)\|^2\right] \leq G^2,$$

$$\mathbb{E}_{a \sim \pi_s(\theta)}\left[\|\nabla_\theta^2 \log \pi_{s,a}(\theta)\|\right] \leq F\,.$$

Expected Lipschitz and smooth (E-LS) policy

[Papini et al., 2019] (Gaussian and softmax policies satisfy E-LS)

- There exists constants G, F > 0 such that for each state $s \in \mathcal{S}$, we have

$$\mathbb{E}_{a\sim\pi_s(\theta)}\left[\|\nabla_\theta \log \pi_{s,a}(\theta)\|^2\right] \leq G^2,$$

$$\mathbb{E}_{a\sim\pi_s(\theta)}\left[\|\nabla^2_\theta \log \pi_{s,a}(\theta)\|\right] \leq F\,.$$

- ABC Assumption holds with $A = 0,\ B = 1 - 1/m$ and $C = \nu/m$. That is,

Expected Lipschitz and smooth (E-LS) policy

[Papini et al., 2019] (Gaussian and softmax policies satisfy E-LS)

- There exists constants G, F > 0 such that for each state $s \in \mathcal{S}$, we have

$$\mathbb{E}_{a\sim\pi_s(\theta)}\left[\|\nabla_\theta \log \pi_{s,a}(\theta)\|^2\right] \leq G^2,$$

$$\mathbb{E}_{a\sim\pi_s(\theta)}\left[\|\nabla^2_\theta \log \pi_{s,a}(\theta)\|\right] \leq F\,.$$

- ABC Assumption holds with $A = 0$, $B = 1 - 1/m$ and $C = \nu/m$. That is,

$$\mathbb{E}\left[\|\hat{\nabla}_m V_\rho(\theta)\|^2\right] \leq \left(1 - \frac{1}{m}\right)\|\nabla V^H_\rho(\theta)\|^2 + \frac{\nu}{m}$$

Expected Lipschitz and smooth (E-LS) policy

[Papini et al., 2019] (Gaussian and softmax policies satisfy E-LS)

- There exists constants G, F > 0 such that for each state $s \in \mathcal{S}$, we have

$$\mathbb{E}_{a \sim \pi_s(\theta)}\left[\|\nabla_\theta \log \pi_{s,a}(\theta)\|^2\right] \leq G^2,$$

$$\mathbb{E}_{a \sim \pi_s(\theta)}\left[\|\nabla_\theta^2 \log \pi_{s,a}(\theta)\|\right] \leq F\,.$$

- ABC Assumption holds with $A = 0$, $B = 1 - 1/m$ and $C = \nu/m$. That is,

$$\mathbb{E}\left[\|\hat{\nabla}_m V_\rho(\theta)\|^2\right] \leq \left(1 - \frac{1}{m}\right)\|\nabla V_\rho^H(\theta)\|^2 + \frac{\boxed{\nu}}{m}$$

Expected Lipschitz and smooth (E-LS) policy

[Papini et al., 2019] (Gaussian and softmax policies satisfy E-LS)

- There exists constants G, F > 0 such that for each state $s \in \mathcal{S}$, we have

$$\mathbb{E}_{a\sim\pi_s(\theta)}\left[\|\nabla_\theta \log \pi_{s,a}(\theta)\|^2\right] \leq G^2,$$

$$\mathbb{E}_{a\sim\pi_s(\theta)}\left[\|\nabla^2_\theta \log \pi_{s,a}(\theta)\|\right] \leq F\,.$$

- ABC Assumption holds with $A = 0,\ B = 1 - 1/m$ and $C = \nu/m$. That is,

$$\mathbb{E}\left[\|\hat{\nabla}_m V_\rho(\theta)\|^2\right] \leq \left(1 - \frac{1}{m}\right)\|\nabla V_\rho^H(\theta)\|^2 + \frac{\boxed{\nu}}{m}$$

- Sample complexity: $KmH = \tilde{O}(\epsilon^{-4})$

Expected Lipschitz and smooth (E-LS) policy

[Papini et al., 2019] (Gaussian and softmax policies satisfy E-LS)

- There exists constants G, F > 0 such that for each state $s \in \mathcal{S}$, we have

$$\mathbb{E}_{a\sim\pi_s(\theta)}\left[\|\nabla_\theta \log \pi_{s,a}(\theta)\|^2\right] \le G^2,$$

$$\mathbb{E}_{a\sim\pi_s(\theta)}\left[\|\nabla_\theta^2 \log \pi_{s,a}(\theta)\|\right] \le F\,.$$

- ABC Assumption holds with $A = 0,\ B = 1 - 1/m$ and $C = \nu/m$. That is,

$$\mathbb{E}\left[\|\hat{\nabla}_m V_\rho(\theta)\|^2\right] \le \left(1 - \frac{1}{m}\right)\|\nabla V_\rho^H(\theta)\|^2 + \frac{\boxed{\nu}}{m}$$

- Sample complexity: $KmH = \tilde{O}(\epsilon^{-4})$

💡 Wider range of parameters

$$m \in \left[1,\ \frac{2\nu}{\epsilon^2}\right]$$

Expected Lipschitz and smooth (E-LS) policy

[Papini et al., 2019] (Gaussian and softmax policies satisfy E-LS)

- There exists constants G, F > 0 such that for each state $s \in \mathcal{S}$, we have

$$\mathbb{E}_{a\sim\pi_s(\theta)}\left[\|\nabla_\theta \log \pi_{s,a}(\theta)\|^2\right] \le G^2,$$

$$\mathbb{E}_{a\sim\pi_s(\theta)}\left[\|\nabla^2_\theta \log \pi_{s,a}(\theta)\|\right] \le F\,.$$

- ABC Assumption holds with $A = 0,\ B = 1 - 1/m$ and $C = \nu/m$. That is,

$$\mathbb{E}\left[\|\hat{\nabla}_m V_\rho(\theta)\|^2\right] \le \left(1-\frac{1}{m}\right)\|\nabla V_\rho^H(\theta)\|^2 + \frac{\boxed{\nu}}{m}$$

- Sample complexity: $\boxed{KmH = \tilde{O}(\epsilon^{-4})}$

Wider range of parameters

$$m \in \left[1, \frac{2\nu}{\epsilon^2}\right]$$

Expected Lipschitz and smooth (E-LS) policy

📗 [Papini et al., 2019] (Gaussian and softmax policies satisfy E-LS)

- There exists constants G, F > 0 such that for each state $s \in \mathcal{S}$, we have

$$\mathbb{E}_{a\sim\pi_s(\theta)}\left[\|\nabla_\theta \log \pi_{s,a}(\theta)\|^2\right] \leq G^2,$$

$$\mathbb{E}_{a\sim\pi_s(\theta)}\left[\|\nabla_\theta^2 \log \pi_{s,a}(\theta)\|\right] \leq F\,.$$

- ABC Assumption holds with $A = 0,\ B = 1 - 1/m$ and $C = \nu/m$. That is,

$$\mathbb{E}\left[\|\hat{\nabla}_m V_\rho(\theta)\|^2\right] \leq \left(1 - \frac{1}{m}\right)\|\nabla V_\rho^H(\theta)\|^2 + \frac{\boxed{\nu}}{m}$$

- Sample complexity: $\boxed{KmH = \tilde{O}(\epsilon^{-4})}$

⚠ Not sample efficiency

💡 Wider range of parameters

$$m \in \left[1,\ \frac{2\nu}{\epsilon^2}\right]$$

Natural Policy Gradient

Rui Yuan, Simon S. Du, Robert M. Gower, Alessandro Lazaric, Lin Xiao
Linear Convergence of Natural Policy Gradient Methods with Log-Linear Policies, ICLR, 2023.

Context

Objective: $\arg\min_{\theta \in \mathbb{R}^d} V_\rho(\theta)$

Context

Objective: $\arg\min_{\theta \in \mathbb{R}^d} V_\rho(\theta)$

- Vanilla PG is not sample efficient

Context

Objective: $\arg\min_{\theta \in \mathbb{R}^d} V_\rho(\theta)$

- Vanilla PG is not sample efficient
- Natural PG (NPG)[Kakade, 2001] uses a preconditioner to improve PG direction

Context

Objective: $\arg\min_{\theta \in \mathbb{R}^d} V_\rho(\theta)$

- Vanilla PG is not sample efficient
- Natural PG (NPG)[Kakade, 2001] uses a preconditioner to improve PG direction
- NPG is the building block of several state-of-the-art algorithms (TRPO, PPO)

Context

Objective: $\arg\min_{\theta \in \mathbb{R}^d} V_\rho(\theta)$

- Vanilla PG is not sample efficient
- Natural PG (NPG)[Kakade, 2001] uses a preconditioner to improve PG direction
- NPG is the building block of several state-of-the-art algorithms (TRPO, PPO)
- Linear convergence of NPG is established for tabular case [Xiao, 2022]

Context

Objective: $\arg\min_{\theta \in \mathbb{R}^d} V_\rho(\theta)$

- Vanilla PG is not sample efficient
- Natural PG (NPG)[Kakade, 2001] uses a preconditioner to improve PG direction
- NPG is the building block of several state-of-the-art algorithms (TRPO, PPO)
- Linear convergence of NPG is established for tabular case [Xiao, 2022]

Context

Objective: $\arg\min_{\theta \in \mathbb{R}^d} V_\rho(\theta)$

- Vanilla PG is not sample efficient
- Natural PG (NPG)[Kakade, 2001] uses a preconditioner to improve PG direction
- NPG is the building block of several state-of-the-art algorithms (TRPO, PPO)
- Linear convergence of NPG is established for tabular case [Xiao, 2022]

Motivations

- Extend linear convergence of NPG from tabular to function approximation regime.

Natural policy gradient

Natural policy gradient

- State-action cost function (a.k.a Q-function) & advantage function

Natural policy gradient

- State-action cost function (a.k.a Q-function) & advantage function

$$Q_{s,a}(\theta) := \mathbb{E}_{a_t \sim \pi_{s_t}(\theta),\ s_{t+1} \sim P(\cdot | s_t, a_t)} \left[\sum_{t=0}^{\infty} \gamma^t c(s_t, a_t) \mid s_0 = s, a_0 = a \right]$$

Natural policy gradient

- State-action cost function (a.k.a Q-function) & advantage function

$$Q_{s,a}(\theta) := \mathbb{E}_{\,a_t \sim \pi_{s_t}(\theta),\ s_{t+1} \sim P(\cdot|s_t,a_t)} \left[\sum_{t=0}^{\infty} \gamma^t c(s_t, a_t) \mid s_0 = s, a_0 = a \right]$$

$$A_{s,a}(\theta) := Q_{s,a}(\theta) - \mathbb{E}_{a' \sim \pi_s(\theta)}[Q_{s,a'}(\theta)]$$

Natural policy gradient

- State-action cost function (a.k.a Q-function) & advantage function

$$Q_{s,a}(\theta) := \mathbb{E}_{\,a_t \sim \pi_{s_t}(\theta),\ s_{t+1} \sim P(\cdot|s_t,a_t)} \left[\sum_{t=0}^{\infty} \gamma^t c(s_t, a_t) \mid s_0 = s, a_0 = a \right]$$

$$A_{s,a}(\theta) := Q_{s,a}(\theta) - \mathbb{E}_{a' \sim \pi_s(\theta)}[Q_{s,a'}(\theta)]$$

- Policy gradient theorem [Sutton et al., 2000]

Natural policy gradient

- State-action cost function (a.k.a Q-function) & advantage function

$$Q_{s,a}(\theta) := \mathbb{E}_{\,a_t \sim \pi_{s_t}(\theta),\; s_{t+1} \sim P(\cdot | s_t, a_t)} \left[\sum_{t=0}^{\infty} \gamma^t c(s_t, a_t) \mid s_0 = s, a_0 = a \right]$$

$$A_{s,a}(\theta) := Q_{s,a}(\theta) - \mathbb{E}_{a' \sim \pi_s(\theta)}[Q_{s,a'}(\theta)]$$

- Policy gradient theorem [Sutton et al., 2000]

$$\nabla_\theta V_\rho(\theta) = \frac{1}{1-\gamma} \mathbb{E}_{(s,a) \sim \mathcal{D}(\theta)} \left[A_{s,a}(\theta)\, \nabla_\theta \log \pi_{s,a}(\theta) \right]$$

Natural policy gradient

- State-action cost function (a.k.a Q-function) & advantage function

$$Q_{s,a}(\theta) := \mathbb{E}_{a_t\sim\pi_{s_t}(\theta),\ s_{t+1}\sim P(\cdot|s_t,a_t)}\left[\sum_{t=0}^{\infty}\gamma^t c(s_t, a_t) \mid s_0 = s, a_0 = a\right]$$

$$A_{s,a}(\theta) := Q_{s,a}(\theta) - \mathbb{E}_{a'\sim\pi_s(\theta)}[Q_{s,a'}(\theta)]$$

- Policy gradient theorem [Sutton et al., 2000]

$$\nabla_\theta V_\rho(\theta) = \frac{1}{1-\gamma}\mathbb{E}_{\boxed{(s,a)\sim\mathscr{D}(\theta)}}\left[A_{s,a}(\theta)\,\nabla_\theta \log \pi_{s,a}(\theta)\right]$$

Stationary distribution of the MDP

Natural policy gradient

- State-action cost function (a.k.a Q-function) & advantage function

$$Q_{s,a}(\theta) := \mathbb{E}_{\,a_t\sim\pi_{s_t}(\theta),\ s_{t+1}\sim P(\cdot|s_t,a_t)}\left[\sum\nolimits_{t=0}^{\infty}\gamma^t c(s_t, a_t) \mid s_0 = s, a_0 = a\right]$$

$$A_{s,a}(\theta) := Q_{s,a}(\theta) - \mathbb{E}_{a'\sim\pi_s(\theta)}[Q_{s,a'}(\theta)]$$

- Policy gradient theorem [Sutton et al., 2000]

Stationary distribution of the MDP

$$\nabla_\theta V_\rho(\theta) = \frac{1}{1-\gamma}\mathbb{E}_{\boxed{(s,a)\sim\mathcal{D}(\theta)}}\left[A_{s,a}(\theta)\,\nabla_\theta \log \pi_{s,a}(\theta)\right]$$

- Natural policy gradient

Natural policy gradient

- State-action cost function (a.k.a Q-function) & advantage function

$$Q_{s,a}(\theta) := \mathbb{E}_{\,a_t\sim\pi_{s_t}(\theta),\ s_{t+1}\sim P(\cdot|s_t,a_t)}\left[\sum_{t=0}^{\infty}\gamma^t c(s_t, a_t) \mid s_0 = s, a_0 = a\right]$$

$$A_{s,a}(\theta) := Q_{s,a}(\theta) - \mathbb{E}_{a'\sim\pi_s(\theta)}[Q_{s,a'}(\theta)]$$

- Policy gradient theorem [Sutton et al., 2000]

Stationary distribution of the MDP

$$\nabla_\theta V_\rho(\theta) = \frac{1}{1-\gamma}\mathbb{E}_{\boxed{(s,a)\sim\mathcal{D}(\theta)}}\left[A_{s,a}(\theta)\,\nabla_\theta \log \pi_{s,a}(\theta)\right]$$

- Natural policy gradient

$$\theta^{(k+1)} = \theta^{(k)} - \eta_k F_\rho(\theta^{(k)})^\dagger\, \nabla_\theta V_\rho(\theta^{(k)})$$

Natural policy gradient

- State-action cost function (a.k.a Q-function) & advantage function

$$Q_{s,a}(\theta) := \mathbb{E}_{\,a_t\sim\pi_{s_t}(\theta),\ s_{t+1}\sim P(\cdot|s_t,a_t)}\left[\sum_{t=0}^{\infty}\gamma^t c(s_t,a_t) \mid s_0 = s, a_0 = a\right]$$

$$A_{s,a}(\theta) := Q_{s,a}(\theta) - \mathbb{E}_{a'\sim\pi_s(\theta)}[Q_{s,a'}(\theta)]$$

- Policy gradient theorem [Sutton et al., 2000]

Stationary distribution of the MDP

$$\nabla_\theta V_\rho(\theta) = \frac{1}{1-\gamma}\mathbb{E}_{\boxed{(s,a)\sim\mathscr{D}(\theta)}}\left[A_{s,a}(\theta)\,\nabla_\theta \log \pi_{s,a}(\theta)\right]$$

- Natural policy gradient

$$\theta^{(k+1)} = \theta^{(k)} - \eta_k F_\rho(\theta^{(k)})^\dagger\, \nabla_\theta V_\rho(\theta^{(k)})$$

$F_\rho(\theta) = \mathbb{E}_{(s,a)\sim\mathscr{D}(\theta)}\left[\nabla_\theta \log \pi_{s,a}(\theta)(\nabla_\theta \log \pi_{s,a}(\theta))^\top\right]$: Fisher information matrix

Natural policy gradient

With log-linear policies

- State-action cost function (a.k.a Q-function) & advantage function

$$Q_{s,a}(\theta) := \mathbb{E}_{\,a_t\sim\pi_{s_t}(\theta),\ s_{t+1}\sim P(\cdot|s_t,a_t)}\left[\sum_{t=0}^{\infty}\gamma^t c(s_t,a_t) \mid s_0 = s, a_0 = a\right]$$

$$A_{s,a}(\theta) := Q_{s,a}(\theta) - \mathbb{E}_{a'\sim\pi_s(\theta)}[Q_{s,a'}(\theta)]$$

- Policy gradient theorem [Sutton et al., 2000]

Stationary distribution of the MDP

$$\nabla_\theta V_\rho(\theta) = \frac{1}{1-\gamma}\mathbb{E}_{\boxed{(s,a)\sim\mathscr{D}(\theta)}}\left[A_{s,a}(\theta)\,\nabla_\theta \log \pi_{s,a}(\theta)\right]$$

- Natural policy gradient

$$\theta^{(k+1)} = \theta^{(k)} - \eta_k F_\rho(\theta^{(k)})^\dagger\,\nabla_\theta V_\rho(\theta^{(k)})$$

$F_\rho(\theta) = \mathbb{E}_{(s,a)\sim\mathscr{D}(\theta)}\left[\nabla_\theta \log \pi_{s,a}(\theta)(\nabla_\theta \log \pi_{s,a}(\theta))^\top\right]$: Fisher information matrix

Natural policy gradient

With log-linear policies

Log-linear policy:

$$\pi_{s,a}(\theta) = \frac{\exp \phi_{s,a}^\top \theta}{\sum_{a' \in \mathscr{A}} \exp \phi_{s,a'}^\top \theta}$$

- State-action cost function (a.k.a Q-function) & advantage function

$$Q_{s,a}(\theta) := \mathbb{E}_{\, a_t \sim \pi_{s_t}(\theta),\ s_{t+1} \sim P(\cdot | s_t, a_t)} \left[\sum_{t=0}^{\infty} \gamma^t c(s_t, a_t) \mid s_0 = s, a_0 = a \right]$$

$$A_{s,a}(\theta) := Q_{s,a}(\theta) - \mathbb{E}_{a' \sim \pi_s(\theta)}[Q_{s,a'}(\theta)]$$

- Policy gradient theorem [Sutton et al., 2000]

Stationary distribution of the MDP

$$\nabla_\theta V_\rho(\theta) = \frac{1}{1-\gamma} \mathbb{E}_{\boxed{(s,a) \sim \mathscr{D}(\theta)}} \left[A_{s,a}(\theta) \nabla_\theta \log \pi_{s,a}(\theta) \right]$$

- Natural policy gradient

$$\theta^{(k+1)} = \theta^{(k)} - \eta_k F_\rho(\theta^{(k)})^\dagger \nabla_\theta V_\rho(\theta^{(k)})$$

$F_\rho(\theta) = \mathbb{E}_{(s,a) \sim \mathscr{D}(\theta)} \left[\nabla_\theta \log \pi_{s,a}(\theta) (\nabla_\theta \log \pi_{s,a}(\theta))^\top \right]$: Fisher information matrix

Natural policy gradient

With log-linear policies

Log-linear policy:

$$\pi_{s,a}(\theta) = \frac{\exp \phi_{s,a}^{\top}\theta}{\sum_{a' \in \mathcal{A}} \exp \boxed{\phi_{s,a'}^{\top}}\theta}$$

Feature map $\phi_{s,a'} \in \mathbb{R}^d$ over $\mathcal{S} \times \mathcal{A}$

- State-action cost function (a.k.a Q-function) & advantage function

$$Q_{s,a}(\theta) := \mathbb{E}_{\, a_t \sim \pi_{s_t}(\theta),\ s_{t+1} \sim P(\cdot | s_t, a_t)} \left[\sum_{t=0}^{\infty} \gamma^t c(s_t, a_t) \mid s_0 = s, a_0 = a \right]$$

$$A_{s,a}(\theta) := Q_{s,a}(\theta) - \mathbb{E}_{a' \sim \pi_s(\theta)}[Q_{s,a'}(\theta)]$$

- Policy gradient theorem [Sutton et al., 2000]

Stationary distribution of the MDP

$$\nabla_\theta V_\rho(\theta) = \frac{1}{1-\gamma} \mathbb{E}_{\boxed{(s,a) \sim \mathcal{D}(\theta)}} \left[A_{s,a}(\theta) \nabla_\theta \log \pi_{s,a}(\theta) \right]$$

- Natural policy gradient

$$\theta^{(k+1)} = \theta^{(k)} - \eta_k F_\rho(\theta^{(k)})^{\dagger} \nabla_\theta V_\rho(\theta^{(k)})$$

$F_\rho(\theta) = \mathbb{E}_{(s,a) \sim \mathcal{D}(\theta)} \left[\nabla_\theta \log \pi_{s,a}(\theta) (\nabla_\theta \log \pi_{s,a}(\theta))^{\top} \right]$: Fisher information matrix

NPG with compatible function approximation

NPG with compatible function approximation

- Compatible function approximation

NPG with compatible function approximation

- Compatible function approximation

$$L(w,\theta,\zeta) = \mathbb{E}_{(s,a)\sim\zeta}\left[(w^\top \nabla_\theta \log \pi_{s,a}(\theta) - A_{s,a}(\theta))^2\right]$$

NPG with compatible function approximation

- Compatible function approximation

$$L(w, \theta, \zeta) = \mathbb{E}_{(s,a)\sim\zeta}\left[(w^\top \nabla_\theta \log \pi_{s,a}(\theta) - A_{s,a}(\theta))^2\right]$$

- NPG can be rewritten as

NPG with compatible function approximation

- Compatible function approximation

$$L(w, \theta, \zeta) = \mathbb{E}_{(s,a)\sim\zeta}\left[(w^\top \nabla_\theta \log \pi_{s,a}(\theta) - A_{s,a}(\theta))^2\right]$$

- NPG can be rewritten as

$$\theta^{(k+1)} = \theta^{(k)} - \eta_k w_\star^{(k)}, \qquad w_\star^{(k)} \in \arg\min_{w\in\mathbb{R}^d} L(w, \theta^{(k)}, \mathscr{D}(\theta^{(k)}))$$

NPG with compatible function approximation

- Compatible function approximation

$$L(w, \theta, \zeta) = \mathbb{E}_{(s,a)\sim\zeta}\left[(w^\top \nabla_\theta \log \pi_{s,a}(\theta) - A_{s,a}(\theta))^2\right]$$

- NPG can be rewritten as

$$\theta^{(k+1)} = \theta^{(k)} - \eta_k w_\star^{(k)}, \qquad \boxed{w_\star^{(k)} \in \arg\min_{w\in\mathbb{R}^d} L(w, \theta^{(k)}, \mathscr{D}(\theta^{(k)}))}$$

NPG with compatible function approximation

- Compatible function approximation

$$L(w, \theta, \zeta) = \mathbb{E}_{(s,a)\sim\zeta}\left[\underbrace{(w^\top \nabla_\theta \log \pi_{s,a}(\theta) - A_{s,a}(\theta))^2}_{\text{Linear approximation of the advantage function}}\right]$$

- NPG can be rewritten as

$$\theta^{(k+1)} = \theta^{(k)} - \eta_k w_\star^{(k)}, \qquad \boxed{w_\star^{(k)} \in \arg\min_{w\in\mathbb{R}^d} L(w, \theta^{(k)}, \mathcal{D}(\theta^{(k)}))}$$

NPG with log-linear as policy mirror descent

NPG with log-linear as policy mirror descent

Log-linear policy:

$$\pi_{s,a}(\theta) = \frac{\exp \phi_{s,a}^\top \theta}{\sum_{a' \in \mathcal{A}} \exp \phi_{s,a'}^\top \theta}$$

NPG with log-linear as policy mirror descent

Log-linear policy:

$$\pi_{s,a}(\theta) = \frac{\exp \phi_{s,a}^\top \theta}{\sum_{a' \in \mathscr{A}} \exp \phi_{s,a'}^\top \theta}$$

- NPG with log-linear can also be written as

NPG with log-linear as policy mirror descent

Log-linear policy:

$$\pi_{s,a}(\theta) = \frac{\exp \phi_{s,a}^\top \theta}{\sum_{a' \in \mathcal{A}} \exp \phi_{s,a'}^\top \theta}$$

- NPG with log-linear can also be written as

$$\pi_s(\theta^{(k+1)}) = \arg \min_{p \in \Delta(\mathcal{A})} \left\{ \eta_k \langle \bar{\Phi}_s^{(k)} w_\star^{(k)}, p \rangle + \mathrm{KL}(p, \pi_s(\theta^{(k)})) \right\}$$

NPG with log-linear as policy mirror descent

Log-linear policy:

$$\pi_{s,a}(\theta) = \frac{\exp \phi_{s,a}^\top \theta}{\sum_{a' \in \mathscr{A}} \exp \phi_{s,a'}^\top \theta}$$

- NPG with log-linear can also be written as

$$\pi_s(\theta^{(k+1)}) = \boxed{\arg \min_{p \in \Delta(\mathscr{A})} \left\{ \eta_k \langle \bar{\Phi}_s^{(k)} w_\star^{(k)}, p \rangle + \mathrm{KL}(p, \pi_s(\theta^{(k)})) \right\}} \to \text{Policy mirror descent}$$

NPG with log-linear as policy mirror descent

Log-linear policy:

$$\pi_{s,a}(\theta) = \frac{\exp \phi_{s,a}^\top \theta}{\sum_{a' \in \mathscr{A}} \exp \phi_{s,a'}^\top \theta}$$

- NPG with log-linear can also be written as

$$\pi_s(\theta^{(k+1)}) = \boxed{\arg\min_{p \in \Delta(\mathscr{A})} \left\{ \eta_k \langle \bar{\Phi}_s^{(k)} w_\star^{(k)}, p \rangle + \mathrm{KL}(p, \pi_s(\theta^{(k)})) \right\}}$$

→ Policy mirror descent

$\bar{\Phi}_s^{(k)} \in \mathbb{R}^{|\mathscr{A}| \times d}$ is a matrix whose rows consist of the *centered feature maps*

NPG with log-linear as policy mirror descent

Log-linear policy:

$$\pi_{s,a}(\theta) = \frac{\exp \phi_{s,a}^\top \theta}{\sum_{a' \in \mathscr{A}} \exp \phi_{s,a'}^\top \theta}$$

- NPG with log-linear can also be written as

$$\pi_s(\theta^{(k+1)}) = \boxed{\arg\min_{p \in \Delta(\mathscr{A})} \left\{ \eta_k \langle \bar{\Phi}_s^{(k)} w_\star^{(k)}, p \rangle + \mathrm{KL}(p, \pi_s(\theta^{(k)})) \right\}}$$

→ Policy mirror descent

$\bar{\Phi}_s^{(k)} \in \mathbb{R}^{|\mathscr{A}| \times d}$ is a matrix whose rows consist of the *centered feature maps*

$$\bar{\phi}_{s,a}(\theta^{(k)}) := \nabla_\theta \log \pi_{s,a}(\theta^{(k)}) = \phi_{s,a} - \mathbb{E}_{a' \sim \pi_s(\theta^{(k)})}[\phi_{s,a'}]$$

NPG with log-linear as policy mirror descent

Log-linear policy:

$$\pi_{s,a}(\theta) = \frac{\exp \phi_{s,a}^\top \theta}{\sum_{a' \in \mathcal{A}} \exp \phi_{s,a'}^\top \theta}$$

- NPG with log-linear can also be written as

$$\pi_s(\theta^{(k+1)}) = \boxed{\arg\min_{p \in \Delta(\mathcal{A})} \left\{ \eta_k \langle \bar{\Phi}_s^{(k)} w_\star^{(k)}, p \rangle + \mathrm{KL}(p, \pi_s(\theta^{(k)})) \right\}}$$

→ Policy mirror descent

$\bar{\Phi}_s^{(k)} \in \mathbb{R}^{|\mathcal{A}| \times d}$ is a matrix whose rows consist of the *centered feature maps*

$$\bar{\phi}_{s,a}(\theta^{(k)}) := \nabla_\theta \log \pi_{s,a}(\theta^{(k)}) = \phi_{s,a} - \mathbb{E}_{a' \sim \pi_s(\theta^{(k)})}[\phi_{s,a'}]$$

$\mathrm{KL}(p, q) = \sum_{a \in \mathcal{A}} p_a \log(p_a / q_a)$ is the Kullback-Leibler (KL) divergence for $p, q \in \Delta(\mathcal{A})$

NPG with log-linear as policy mirror descent

Log-linear policy:

$$\pi_{s,a}(\theta) = \frac{\exp \phi_{s,a}^\top \theta}{\sum_{a' \in \mathscr{A}} \exp \phi_{s,a'}^\top \theta}$$

- NPG with log-linear can also be written as

$$\pi_s(\theta^{(k+1)}) = \boxed{\arg \min_{p \in \Delta(\mathscr{A})} \left\{ \eta_k \langle \bar{\Phi}_s^{(k)} w_\star^{(k)}, p \rangle + \mathrm{KL}(p, \pi_s(\theta^{(k)})) \right\}}$$

→ Policy mirror descent

$\bar{\Phi}_s^{(k)} \in \mathbb{R}^{|\mathscr{A}| \times d}$ is a matrix whose rows consist of the *centered feature maps*

$$\bar{\phi}_{s,a}(\theta^{(k)}) := \nabla_\theta \log \pi_{s,a}(\theta^{(k)}) = \phi_{s,a} - \mathbb{E}_{a' \sim \pi_s(\theta^{(k)})}[\phi_{s,a'}]$$

$\mathrm{KL}(p, q) = \sum_{a \in \mathscr{A}} p_a \log(p_a / q_a)$ is the Kullback-Leibler (KL) divergence for $p, q \in \Delta(\mathscr{A})$

- Connection with Policy Iteration

NPG with log-linear as policy mirror descent

Log-linear policy:

$$\pi_{s,a}(\theta) = \frac{\exp \phi_{s,a}^\top \theta}{\sum_{a' \in \mathscr{A}} \exp \phi_{s,a'}^\top \theta}$$

- NPG with log-linear can also be written as

$$\pi_s(\theta^{(k+1)}) = \boxed{\arg \min_{p \in \Delta(\mathscr{A})} \left\{ \eta_k \langle \bar{\Phi}_s^{(k)} w_\star^{(k)}, p \rangle + \mathrm{KL}(p, \pi_s(\theta^{(k)})) \right\}}$$

→ Policy mirror descent

$\bar{\Phi}_s^{(k)} \in \mathbb{R}^{|\mathscr{A}| \times d}$ is a matrix whose rows consist of the *centered feature maps*

$$\bar{\phi}_{s,a}(\theta^{(k)}) := \nabla_\theta \log \pi_{s,a}(\theta^{(k)}) = \phi_{s,a} - \mathbb{E}_{a' \sim \pi_s(\theta^{(k)})}[\phi_{s,a'}]$$

$\mathrm{KL}(p, q) = \sum_{a \in \mathscr{A}} p_a \log(p_a / q_a)$ is the Kullback-Leibler (KL) divergence for $p, q \in \Delta(\mathscr{A})$

- Connection with Policy Iteration

$$\pi_s(\theta^{(k+1)}) = \arg \min_{p \in \Delta(\mathscr{A})} \left\{ \eta_k \langle A_s(\theta^{(k)}), p \rangle \right\} \quad \text{with} \quad A_s(\theta^{(k)}) := [A_{s,a}(\theta^{(k)})]_a \in \mathbb{R}^{|\mathscr{A}|}$$

NPG with log-linear as policy mirror descent

Log-linear policy:

$$\pi_{s,a}(\theta) = \frac{\exp \phi_{s,a}^\top \theta}{\sum_{a'\in\mathscr{A}} \exp \phi_{s,a'}^\top \theta}$$

- NPG with log-linear can also be written as

$$\pi_s(\theta^{(k+1)}) = \boxed{\arg\min_{p\in\Delta(\mathscr{A})} \left\{\eta_k\langle \bar{\Phi}_s^{(k)} w_\star^{(k)}, p\rangle + \boxed{\mathrm{KL}(p, \pi_s(\theta^{(k)}))}\right\}} \rightarrow \text{Policy mirror descent}$$

Regularization

$\bar{\Phi}_s^{(k)} \in \mathbb{R}^{|\mathscr{A}|\times d}$ is a matrix whose rows consist of the *centered feature maps*

$$\bar{\phi}_{s,a}(\theta^{(k)}) := \nabla_\theta \log \pi_{s,a}(\theta^{(k)}) = \phi_{s,a} - \mathbb{E}_{a'\sim\pi_s(\theta^{(k)})}[\phi_{s,a'}]$$

$\mathrm{KL}(p,q) = \sum_{a\in\mathscr{A}} p_a \log(p_a/q_a)$ is the Kullback-Leibler (KL) divergence for $p, q \in \Delta(\mathscr{A})$

- Connection with Policy Iteration

$$\pi_s(\theta^{(k+1)}) = \arg\min_{p\in\Delta(\mathscr{A})} \left\{\eta_k\langle A_s(\theta^{(k)}), p\rangle\right\} \quad \text{with } A_s(\theta^{(k)}) := [A_{s,a}(\theta^{(k)})]_a \in \mathbb{R}^{|\mathscr{A}|}$$

NPG with log-linear as policy mirror descent

Log-linear policy:

$$\pi_{s,a}(\theta) = \frac{\exp \phi_{s,a}^\top \theta}{\sum_{a' \in \mathcal{A}} \exp \phi_{s,a'}^\top \theta}$$

- NPG with log-linear can also be written as

$$\pi_s(\theta^{(k+1)}) = \boxed{\arg\min_{p \in \Delta(\mathcal{A})} \left\{ \eta_k \langle \boxed{\bar{\Phi}_s^{(k)} w_\star^{(k)}}, p \rangle + \boxed{\mathrm{KL}(p, \pi_s(\theta^{(k)}))} \right\}}$$

→ Policy mirror descent

Regularization

$\bar{\Phi}_s^{(k)} \in \mathbb{R}^{|\mathcal{A}| \times d}$ is a matrix whose rows consist of the *centered feature maps*

$$\bar{\phi}_{s,a}(\theta^{(k)}) := \nabla_\theta \log \pi_{s,a}(\theta^{(k)}) = \phi_{s,a} - \mathbb{E}_{a' \sim \pi_s(\theta^{(k)})}[\phi_{s,a'}]$$

$\mathrm{KL}(p, q) = \sum_{a \in \mathcal{A}} p_a \log(p_a / q_a)$ is the Kullback-Leibler (KL) divergence for $p, q \in \Delta(\mathcal{A})$

- Connection with Policy Iteration

Linear approximation

$$\pi_s(\theta^{(k+1)}) = \arg\min_{p \in \Delta(\mathcal{A})} \left\{ \eta_k \langle \boxed{A_s(\theta^{(k)})}, p \rangle \right\} \quad \text{with} \quad A_s(\theta^{(k)}) := [A_{s,a}(\theta^{(k)})]_a \in \mathbb{R}^{|\mathcal{A}|}$$

Convergence theory

Convergence theory

- Three-point descent lemma [Chen and Teboulle, 1993]:

Convergence theory

- Three-point descent lemma [Chen and Teboulle, 1993]:

  For any $p \in \Delta(\mathscr{A})$,

Convergence theory

- Three-point descent lemma [Chen and Teboulle, 1993]:

  For any $p \in \Delta(\mathscr{A})$,

$$\begin{aligned}&\eta_k \langle \bar{\Phi}_s^{(k)} w_\star^{(k)}, \pi_s(\theta^{(k+1)}) \rangle + \mathrm{KL}(\pi_s(\theta^{(k+1)}), \pi_s(\theta^{(k)})) \\ \leq\ &\eta_k \langle \bar{\Phi}_s^{(k)} w_\star^{(k)}, p \rangle + \mathrm{KL}(p, \pi_s(\theta^{(k)})) - \mathrm{KL}(p, \pi_s(\theta^{(k+1)}))\end{aligned}$$

Convergence theory

- Three-point descent lemma [Chen and Teboulle, 1993]:

  For any $p \in \Delta(\mathscr{A})$,

$$\eta_k \langle \bar{\Phi}_s^{(k)} w_\star^{(k)}, \pi_s(\theta^{(k+1)}) \rangle + \mathrm{KL}(\pi_s(\theta^{(k+1)}), \pi_s(\theta^{(k)}))$$
$$\leq \eta_k \langle \bar{\Phi}_s^{(k)} w_\star^{(k)}, p \rangle + \mathrm{KL}(p, \pi_s(\theta^{(k)})) - \mathrm{KL}(p, \pi_s(\theta^{(k+1)}))$$

  One can let $p = \pi_s(\theta^{(k)})$ or be the optimal policy to derive a telescoping sum

Convergence theory

- Three-point descent lemma [Chen and Teboulle, 1993]:

  For any $p \in \Delta(\mathscr{A})$,

$$\eta_k \langle \bar{\Phi}_s^{(k)} w_\star^{(k)}, \pi_s(\theta^{(k+1)}) \rangle + \mathrm{KL}(\pi_s(\theta^{(k+1)}), \pi_s(\theta^{(k)}))$$
$$\leq \eta_k \langle \bar{\Phi}_s^{(k)} w_\star^{(k)}, p \rangle + \mathrm{KL}(p, \pi_s(\theta^{(k)})) - \mathrm{KL}(p, \pi_s(\theta^{(k+1)}))$$

  One can let $p = \pi_s(\theta^{(k)})$ or be the optimal policy to derive a telescoping sum

- Linear convergence to the global optimum by increasing step size by $1/\gamma$

Convergence theory

- Three-point descent lemma [Chen and Teboulle, 1993]:

  For any $p \in \Delta(\mathscr{A})$,

$$\eta_k \langle \bar{\Phi}_s^{(k)} w_\star^{(k)}, \pi_s(\theta^{(k+1)}) \rangle + \mathrm{KL}(\pi_s(\theta^{(k+1)}), \pi_s(\theta^{(k)}))$$

$$\leq \eta_k \langle \bar{\Phi}_s^{(k)} w_\star^{(k)}, p \rangle + \mathrm{KL}(p, \pi_s(\theta^{(k)})) - \mathrm{KL}(p, \pi_s(\theta^{(k+1)}))$$

  One can let $p = \pi_s(\theta^{(k)})$ or be the optimal policy to derive a telescoping sum

- Linear convergence to the global optimum by increasing step size by $1/\gamma$

$$\pi_s(\theta^{(k+1)}) = \arg \min_{p \in \Delta(\mathscr{A})} \left\{ \eta_k \langle \bar{\Phi}_s^{(k)} w_\star^{(k)}, p \rangle + \mathrm{KL}(p, \pi_s(\theta^{(k)})) \right\}$$

Convergence theory

- Three-point descent lemma [Chen and Teboulle, 1993]:

  For any $p \in \Delta(\mathscr{A})$,

  $$\eta_k \langle \bar{\Phi}_s^{(k)} w_\star^{(k)}, \pi_s(\theta^{(k+1)}) \rangle + \mathrm{KL}(\pi_s(\theta^{(k+1)}), \pi_s(\theta^{(k)}))$$
  $$\le \eta_k \langle \bar{\Phi}_s^{(k)} w_\star^{(k)}, p \rangle + \mathrm{KL}(p, \pi_s(\theta^{(k)})) - \mathrm{KL}(p, \pi_s(\theta^{(k+1)}))$$

  One can let $p = \pi_s(\theta^{(k)})$ or be the optimal policy to derive a telescoping sum

- Linear convergence to the global optimum by increasing step size by $1/\gamma$

$$\pi_s(\theta^{(k+1)}) = \arg\min_{p \in \Delta(\mathscr{A})} \left\{ \eta_k \langle \bar{\Phi}_s^{(k)} w_\star^{(k)}, p \rangle + \mathrm{KL}(p, \pi_s(\theta^{(k)})) \right\} \qquad \eta_k \longrightarrow \infty$$

Convergence theory

- Three-point descent lemma [Chen and Teboulle, 1993]:

  For any $p \in \Delta(\mathscr{A})$,

$$\eta_k \langle \bar{\Phi}_s^{(k)} w_\star^{(k)}, \pi_s(\theta^{(k+1)}) \rangle + \mathrm{KL}(\pi_s(\theta^{(k+1)}), \pi_s(\theta^{(k)}))$$
$$\leq \eta_k \langle \bar{\Phi}_s^{(k)} w_\star^{(k)}, p \rangle + \mathrm{KL}(p, \pi_s(\theta^{(k)})) - \mathrm{KL}(p, \pi_s(\theta^{(k+1)}))$$

  One can let $p = \pi_s(\theta^{(k)})$ or be the optimal policy to derive a telescoping sum

- Linear convergence to the global optimum by increasing step size by $1/\gamma$

$$\pi_s(\theta^{(k+1)}) = \arg \min_{p \in \Delta(\mathscr{A})} \left\{ \eta_k \langle \bar{\Phi}_s^{(k)} w_\star^{(k)}, p \rangle + \xcancel{\mathrm{KL}(p, \pi_s(\theta^{(k)}))} \right\} \qquad \eta_k \longrightarrow \infty$$

Convergence theory

- Three-point descent lemma [Chen and Teboulle, 1993]:

  For any $p \in \Delta(\mathscr{A})$,

$$\eta_k \langle \bar{\Phi}_s^{(k)} w_\star^{(k)}, \pi_s(\theta^{(k+1)}) \rangle + \mathrm{KL}(\pi_s(\theta^{(k+1)}), \pi_s(\theta^{(k)}))$$
$$\leq \eta_k \langle \bar{\Phi}_s^{(k)} w_\star^{(k)}, p \rangle + \mathrm{KL}(p, \pi_s(\theta^{(k)})) - \mathrm{KL}(p, \pi_s(\theta^{(k+1)}))$$

  One can let $p = \pi_s(\theta^{(k)})$ or be the optimal policy to derive a telescoping sum

- Linear convergence to the global optimum by increasing step size by $1/\gamma$

$$\pi_s(\theta^{(k+1)}) = \arg \min_{p \in \Delta(\mathscr{A})} \left\{ \eta_k \langle \boxed{\bar{\Phi}_s^{(k)} w_\star^{(k)}}, p \rangle + \mathrm{KL}(\xcancel{p, \pi_s}(\theta^{(k)})) \right\} \qquad \eta_k \longrightarrow \infty$$

Convergence theory

- Three-point descent lemma [Chen and Teboulle, 1993]:

  For any $p \in \Delta(\mathscr{A})$,

$$\eta_k \langle \bar{\Phi}_s^{(k)} w_\star^{(k)}, \pi_s(\theta^{(k+1)}) \rangle + \mathrm{KL}(\pi_s(\theta^{(k+1)}), \pi_s(\theta^{(k)}))$$
$$\leq \eta_k \langle \bar{\Phi}_s^{(k)} w_\star^{(k)}, p \rangle + \mathrm{KL}(p, \pi_s(\theta^{(k)})) - \mathrm{KL}(p, \pi_s(\theta^{(k+1)}))$$

  One can let $p = \pi_s(\theta^{(k)})$ or be the optimal policy to derive a telescoping sum

- Linear convergence to the global optimum by increasing step size by $1/\gamma$

$$\pi_s(\theta^{(k+1)}) = \arg \min_{p \in \Delta(\mathscr{A})} \left\{ \eta_k \langle \boxed{\bar{\Phi}_s^{(k)} w_\star^{(k)}}, p \rangle + \cancel{\mathrm{KL}(p, \pi_s(\theta^{(k)}))} \right\} \qquad \eta_k \longrightarrow \infty$$

Behave more and more like policy iteration

Convergence theory 2

Convergence theory 2

- Consequently, we obtain an $\tilde{O}(\epsilon^{-2})$ sample complexity for NPG

Convergence theory 2

- Consequently, we obtain an $\tilde{O}(\epsilon^{-2})$ sample complexity for NPG
- Similar linear convergence and $\tilde{O}(\epsilon^{-2})$ sample complexity results are also established for Q-NPG

Convergence theory 2

- Consequently, we obtain an $\tilde{O}(\epsilon^{-2})$ sample complexity for NPG
- Similar linear convergence and $\tilde{O}(\epsilon^{-2})$ sample complexity results are also established for Q-NPG
- Sublinear convergence for both NPG and Q-NPG with arbitrary large constant step size

Discussion

& Connections to each other

- SNR and SNRVM open the way to designing and analyzing a host of new stochastic second order methods (e.g. stochastic Polyak method [Gower et al., 2021])

- SNR and SNRVM open the way to designing and analyzing a host of new stochastic second order methods (e.g. stochastic Polyak method [Gower et al., 2021])

- The use of the gradient domination type assumption in the vanilla PG analysis influence the analysis of variance reduced PG methods [Fatkhullin et al., 2022]

- SNR and SNRVM open the way to designing and analyzing a host of new stochastic second order methods (e.g. stochastic Polyak method [Gower et al., 2021])
- The use of the gradient domination type assumption in the vanilla PG analysis influence the analysis of variance reduced PG methods [Fatkhullin et al., 2022]
- The linear convergence analysis of NPG with log-linear policy is extended to general parametrization [Alfano et al., 2023]

- SNR and SNRVM open the way to designing and analyzing a host of new stochastic second order methods (e.g. stochastic Polyak method [Gower et al., 2021])

- The use of the gradient domination type assumption in the vanilla PG analysis influence the analysis of variance reduced PG methods [Fatkhullin et al., 2022]

- The linear convergence analysis of NPG with log-linear policy is extended to general parametrization [Alfano et al., 2023]

- Stochastic second order methods for optimizing the expected cost in RL (e.g. sketched NPG ?)

Conclusion

A principled approach to

design stochastic Newton methods (Part I)

A better understanding and sample efficiency

in gradient-based RL (Part II)

List of Papers

- A Novel Framework for Policy Mirror Descent with General Parametrization and Linear Convergence, preprint, 2023.
  Carlo Alfano, Rui Yuan, Patrick Rebeschini

- Linear Convergence of Natural Policy Gradient Methods with Log-Linear Policies, ICLR 2023
  Rui Yuan, Simon S. Du, Robert M. Gower, Alessandro Lazaric, Lin Xiao

- A general sample complexity analysis of vanilla policy gradient, AISTATS 2022
  Rui Yuan, Robert M. Gower, Alessandro Lazaric

- SAN: Stochastic Average Newton Algorithm for Minimizing Finite Sums, AISTATS 2022
  Jiabin Chen*, Rui Yuan*, Guillaume Garrigos, Robert M. Gower

- Sketched Newton-Raphson, SIAM 2022
  Rui Yuan, Alessandro Lazaric, Robert M. Gower

Thank you !

References

- Robert M. Gower and Peter Richtárik. Randomized iterative methods for linear systems. SIAM Journal on Matrix Analysis and Applications, 36(4):1660–1690, 2015.
- A. Rodomanov and D. Kropotov. A superlinearly-convergent proximal newton-type method for the optimization of finite sums, in Proceedings of The 33rd International Conference on Machine Learning, vol. 48 of Proceedings of Machine Learning Research, PMLR, 20–22 Jun 2016, pp. 2597–2605.
- Dmitry Kovalev, Konstantin Mishchenko, and Peter Richtarik. Stochastic newton and cubic newton methods with simple local linear-quadratic rates. 2019.
- Vijay Konda and John Tsitsiklis. Actor-critic algorithms. In Advances in Neural Information Processing Systems, volume 12. MIT Press, 2000.
- Sham M Kakade. A natural policy gradient. In Advances in Neural Information Processing Systems, volume 14. MIT Press, 2001.
- John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In Francis Bach and David Blei, editors, Proceedings of the 32nd International Conference on Machine Learning, volume 37 of Proceedings of Machine Learning Research, pages 1889–1897, Lille, France, 07–09 Jul 2015. PMLR.
- John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017.
- Matteo Papini, Damiano Binaghi, Giuseppe Canonaco, Matteo Pirotta, and Marcello Restelli. Stochastic variance-reduced policy gradient. In Proceedings of the 35th International Conference on Machine Learning, volume 80, pages 4026–4035. PMLR, 2018.
- Zebang Shen, Alejandro Ribeiro, Hamed Hassani, Hui Qian, and Chao Mi. Hessian aided policy gradient. In Proceedings of the 36th International Conference on Machine Learning, volume 97 of Proceedings of Machine Learning Research, pages 5729– 5738. PMLR, 09–15 Jun 2019

References

- Pan Xu, Felicia Gao, and Quanquan Gu. Sample efficient policy gradient methods with recursive variance reduction. In International Conference on Learning Representations, 2020.
- Feihu Huang, Shangqian Gao, Jian Pei, and Heng Huang. Momentum-based policy gradient methods, 2020.
- R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. Machine Learning, 8:229–256, 1992.
- J. Baxter and P. L. Bartlett. Infinite-horizon policy-gradient estimation. Journal of Artificial Intelligence Research, 15:319–350, Nov 2001.
- Alekh Agarwal, Sham M. Kakade, Jason D. Lee, and Gaurav Mahajan. On the theory of policy gradient methods: Optimality, approximation, and distribution shift. 2019.
- Jincheng Mei, Chenjun Xiao, Csaba Szepesvari, and Dale Schuurmans. On the global convergence rates of softmax policy gradient methods. In Proceedings of the 37th International Conference on Machine Learning, volume 119 of Proceedings of Machine Learning Research, pages 6820–6829. PMLR, 13–18 Jul 2020.
- Matteo Papini, Matteo Pirotta, and Marcello Restelli. Smoothing policies and safe policy gradients, 2019.
- Yanli Liu, Kaiqing Zhang, Tamer Basar, and Wotao Yin. An improved analysis of (variance-reduced) policy gradient and natural policy gradient methods. In Advances in Neural Information Processing Systems, volume 33, pages 7624–7636, 2020
- Huaqing Xiong, Tengyu Xu, Yingbin Liang, and Wei Zhang. Non-asymptotic convergence of adam-type reinforcement learning algorithms under markovian sampling. Proceedings of the AAAI Conference on Artificial Intelligence, 35(12):10460–10468, May 2021.
- Junyu Zhang, Alec Koppel, Amrit Singh Bedi, Csaba Szepesvari, and Mengdi Wang. Variational policy gradient method for reinforcement learning with general utilities. In Advances in Neural Information Processing Systems, volume 33, pages 4572–4583. Curran Associates, Inc., 2020a.

References

- Junzi Zhang, Jongho Kim, Brendan O'Donoghue, and Stephen Boyd. Sample efficient reinforcement learning with reinforce, 2020b.
- Kaiqing Zhang, Alec Koppel, Hao Zhu, and Tamer Başar. Global convergence of policy gradient methods to (almost) locally optimal policies. SIAM Journal on Control and Optimization, 58(6):3586–3612, 2020c.
- Yuhao Ding, Junzi Zhang, and Javad Lavaei. On the global convergence of momentum-based policy gradient, 2021.
- Ahmed Khaled and Peter Richtárik. Better theory for sgd in the nonconvex world, 2020.
- Saeed Ghadimi and Guanghui Lan. Stochastic firstand zeroth-order methods for nonconvex stochastic programming. SIAM journal on optimization, 23 (4):2341–2368, 2013.
- Lin Xiao. On the convergence rates of policy gradient methods. Journal of Machine Learning Research, 23(282):1–36, 2022.
- Richard S Sutton, David A. McAllester, Satinder P. Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In Advances in Neural Information Processing Systems 12, pages 1057–1063. MIT Press, 2000.
- Gong Chen and Marc Teboulle. Convergence analysis of a proximal-like minimization algorithm using bregman functions. SIAM Journal on Optimization, 3(3):538–543, 1993.
- Gower, Robert M., Aaron Defazio, and Mike Rabbat. Stochastic Polyak Stepsize with a Moving Target. In Advances in neural information processing systems, 13th Annual Workshop on Optimization for Machine Learning (OPT2021), 2021
- Fatkhullin, Ilyas, Jalal Etesami, Niao He, and Negar Kiyavash (2022). Sharp Analysis of Stochastic Optimization under Global Kurdyka-Łojasiewicz Inequality. In Advances in Neural Information Processing Systems

Back-up Slides

Stochastic Newton method (SNM)

[Kovalev et al., 2019]

- Solving a finite-sum minimization problem

- Finding a stationary point of the gradient of f : $\nabla f(x) = \frac{1}{n} \sum_{i=1}^{n} \nabla f_i(x) = 0$

Stochastic Newton method (SNM)

[Kovalev et al., 2019]

- Solving a finite-sum minimization problem

$$\min_{x\in\mathbb{R}^d}\left[f(x) := \frac{1}{n}\sum_{i=1}^{n} f_i(x)\right]$$

- Finding a stationary point of the gradient of f : $\nabla f(x) = \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(x) = 0$

Stochastic Newton method (SNM)

[Kovalev et al., 2019]

- Solving a finite-sum minimization problem

$f_i(x)$:= The loss over the ith batch of data

$$\min_{x \in \mathbb{R}^d} \left[f(x) := \frac{1}{n} \sum_{i=1}^{n} \boxed{f_i(x)} \right]$$

- Finding a stationary point of the gradient of f : $\nabla f(x) = \dfrac{1}{n} \sum_{i=1}^{n} \nabla f_i(x) = 0$

Stochastic Newton method (SNM)

[Kovalev et al., 2019]

- Solving a finite-sum minimization problem

$f_i(x)$:= The loss over the ith batch of data

$$\min_{x\in\mathbb{R}^d}\left[f(x) := \frac{1}{\boxed{n}}\sum_{i=1}^{n}\boxed{f_i(x)}\right]$$

n := Number of samples

- Finding a stationary point of the gradient of f : $\nabla f(x) = \frac{1}{n}\sum_{i=1}^{n}\nabla f_i(x) = 0$

Stochastic Newton method (SNM)

[Kovalev et al., 2019]

- Solving a finite-sum minimization problem

$f_i(x)$:= The loss over the ith batch of data

Training problem

$$\boxed{\min_{x\in\mathbb{R}^d}\left[f(x) := \frac{1}{\boxed{n}}\sum_{i=1}^{n}\boxed{f_i(x)}\right]}$$

n := Number of samples

- Finding a stationary point of the gradient of f : $\nabla f(x) = \dfrac{1}{n}\sum_{i=1}^{n}\nabla f_i(x) = 0$

Objective: $\nabla f(x) = \frac{1}{n} \sum_{i=1}^{n} \nabla f_i(x) = 0$

Objective: $\nabla f(x) = \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(x) = 0$

- Rewrite the problem as

$$\frac{1}{n}\sum_{i=1}^{n} \nabla f_i(w^i) \ = \ 0, \qquad \text{and} \qquad x \ = \ w^i, \quad \text{for } i = 1,\ldots,n$$

- $F(x; w_i) = 0$ where $F : \mathbb{R}^{(n+1)d} \to \mathbb{R}^{(n+1)d}$, i.e. $p = m = (n+1)d$
- Sketching matrix : based on subsampling $(n+1)$ blocks and the Hessian matrices of the f_i functions

SNM is a special case of SNR!

Objective: $\nabla f(x) = \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(x) = 0$

- Rewrite the problem as

$$\frac{1}{n}\sum_{i=1}^{n} \nabla f_i(w^i) \;=\; 0, \qquad \text{and} \qquad x \;=\; w^i, \quad \text{for } i = 1,\ldots,n$$

- $F(x; w_i) = 0$ where $F : \mathbb{R}^{(n+1)d} \to \mathbb{R}^{(n+1)d}$, i.e. $p = m = (n+1)d$

- Sketching matrix : based on subsampling $(n+1)$ blocks and the Hessian matrices of the f_i functions

SNM is a special case of SNR!

Objective: $\nabla f(x) = \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(x) = 0$

- Rewrite the problem as

$$\frac{1}{n}\sum_{i=1}^{n} \nabla f_i(w^i) \ = \ 0, \qquad \text{and} \qquad x \ = \ w^i, \quad \text{for } i = 1,\ldots,n$$

- $F(x; w_i) = 0$ where $F : \mathbb{R}^{(n+1)d} \to \mathbb{R}^{(n+1)d}$, i.e. $p = m = (n+1)d$
- Sketching matrix : based on subsampling $(n+1)$ blocks and the Hessian matrices of the f_i functions

💡 Consequently, establish the first global convergence theory of SNM

Overview of convergence results for vanilla PG

Figure from [Yuan et al., 2022]

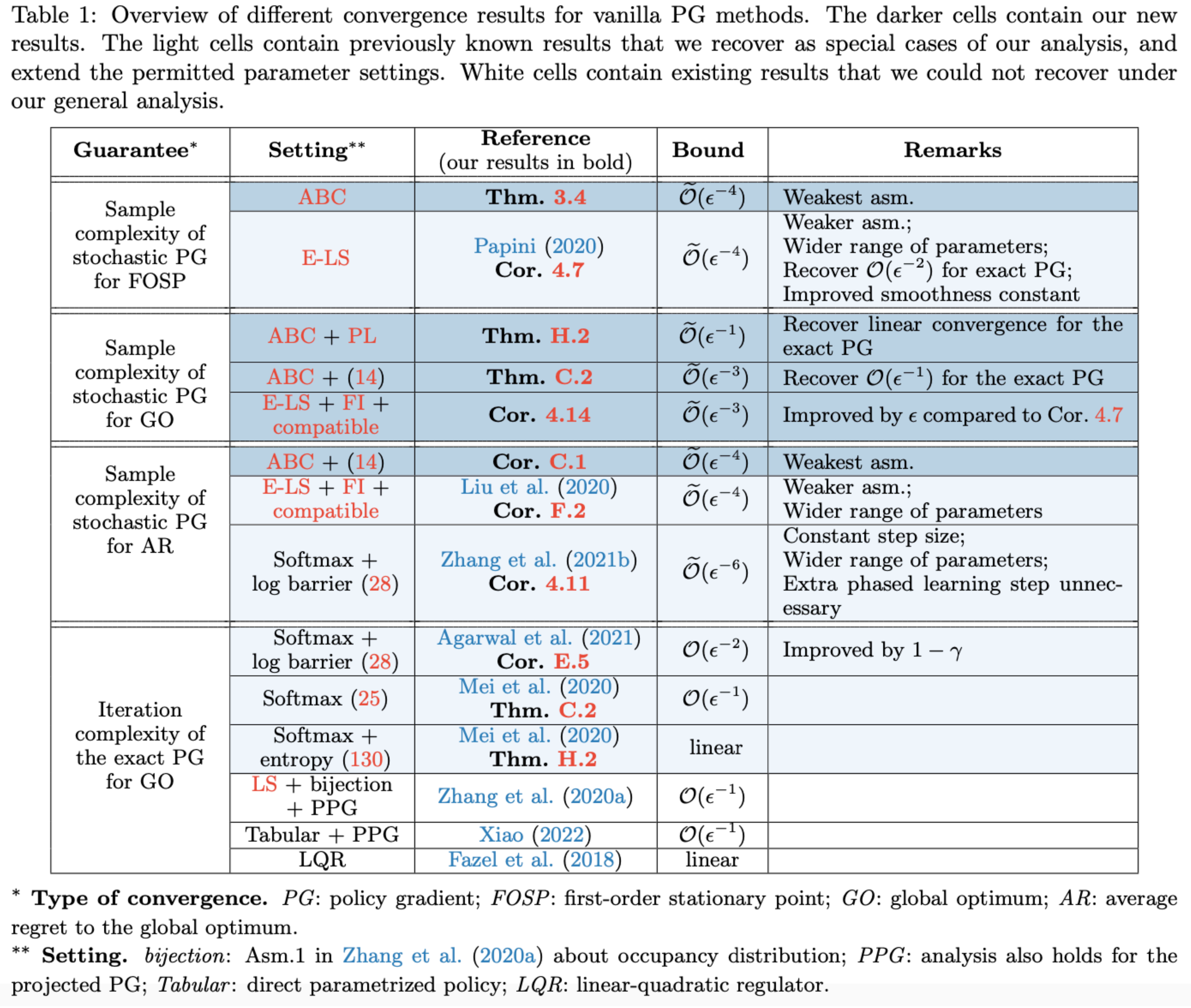


Table 1: Overview of different convergence results for vanilla PG methods. The darker cells contain our new results. The light cells contain previously known results that we recover as special cases of our analysis, and extend the permitted parameter settings. White cells contain existing results that we could not recover under our general analysis.

Guarantee*	**Setting****	**Reference** (our results in bold)	**Bound**	**Remarks**
Sample complexity of stochastic PG for FOSP	ABC	**Thm. 3.4**	$\widetilde{\mathcal{O}}(\epsilon^{-4})$	Weakest asm.
	E-LS	Papini (2020) **Cor. 4.7**	$\widetilde{\mathcal{O}}(\epsilon^{-4})$	Weaker asm.; Wider range of parameters; Recover $\mathcal{O}(\epsilon^{-2})$ for exact PG; Improved smoothness constant
Sample complexity of stochastic PG for GO	ABC + PL	**Thm. H.2**	$\widetilde{\mathcal{O}}(\epsilon^{-1})$	Recover linear convergence for the exact PG
	ABC + (14)	**Thm. C.2**	$\widetilde{\mathcal{O}}(\epsilon^{-3})$	Recover $\mathcal{O}(\epsilon^{-1})$ for the exact PG
	E-LS + FI + compatible	**Cor. 4.14**	$\widetilde{\mathcal{O}}(\epsilon^{-3})$	Improved by ϵ compared to Cor. 4.7
Sample complexity of stochastic PG for AR	ABC + (14)	**Cor. C.1**	$\widetilde{\mathcal{O}}(\epsilon^{-4})$	Weakest asm.
	E-LS + FI + compatible	Liu et al. (2020) **Cor. F.2**	$\widetilde{\mathcal{O}}(\epsilon^{-4})$	Weaker asm.; Wider range of parameters
	Softmax + log barrier (28)	Zhang et al. (2021b) **Cor. 4.11**	$\widetilde{\mathcal{O}}(\epsilon^{-6})$	Constant step size; Wider range of parameters; Extra phased learning step unnecessary
Iteration complexity of the exact PG for GO	Softmax + log barrier (28)	Agarwal et al. (2021) **Cor. E.5**	$\mathcal{O}(\epsilon^{-2})$	Improved by $1-\gamma$
	Softmax (25)	Mei et al. (2020) **Thm. C.2**	$\mathcal{O}(\epsilon^{-1})$	
	Softmax + entropy (130)	Mei et al. (2020) **Thm. H.2**	linear	
	LS + bijection + PPG	Zhang et al. (2020a)	$\mathcal{O}(\epsilon^{-1})$	
	Tabular + PPG	Xiao (2022)	$\mathcal{O}(\epsilon^{-1})$	
	LQR	Fazel et al. (2018)	linear	

* **Type of convergence.** *PG*: policy gradient; *FOSP*: first-order stationary point; *GO*: global optimum; *AR*: average regret to the global optimum.

** **Setting.** *bijection*: Asm.1 in Zhang et al. (2020a) about occupancy distribution; *PPG*: analysis also holds for the projected PG; *Tabular*: direct parametrized policy; *LQR*: linear-quadratic regulator.

A hierarchy between the assumptions

Figure from [Yuan et al., 2022]

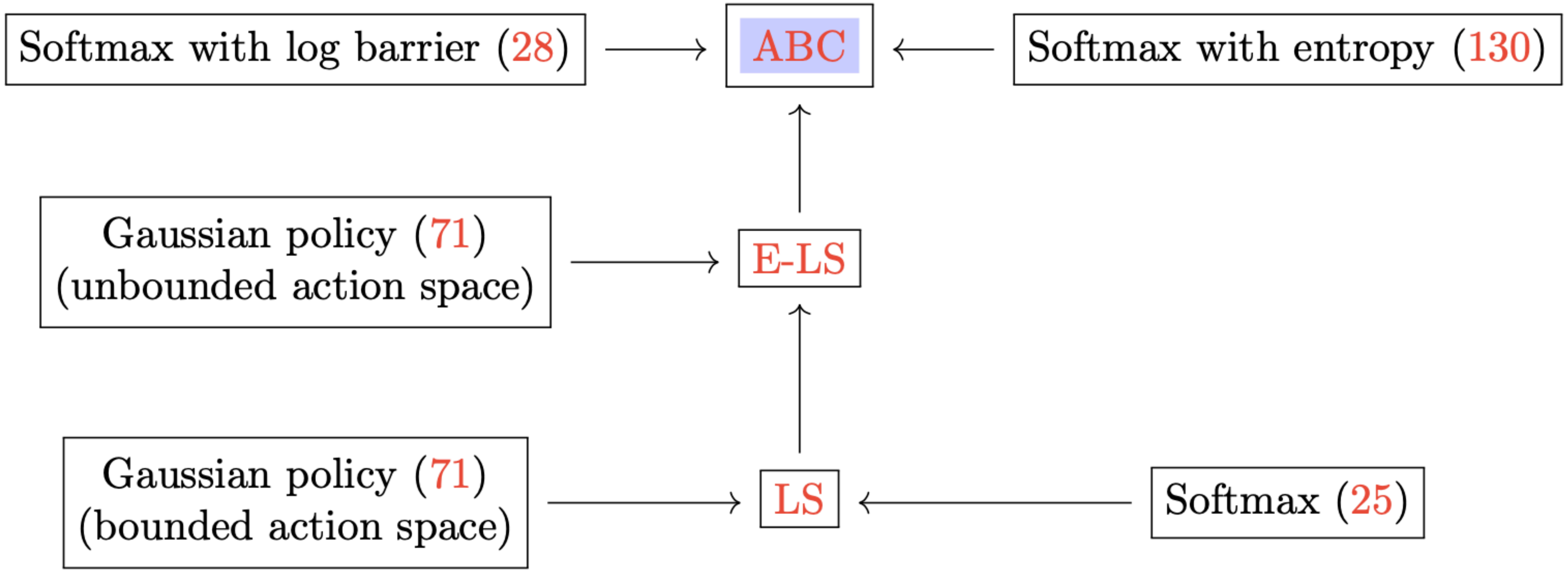


Figure 1: A hierarchy between the assumptions we present throughout the chapter. An arrow indicates an implication.

Overview of convergence results for NPG

Figure from [Yuan et al., 2023]

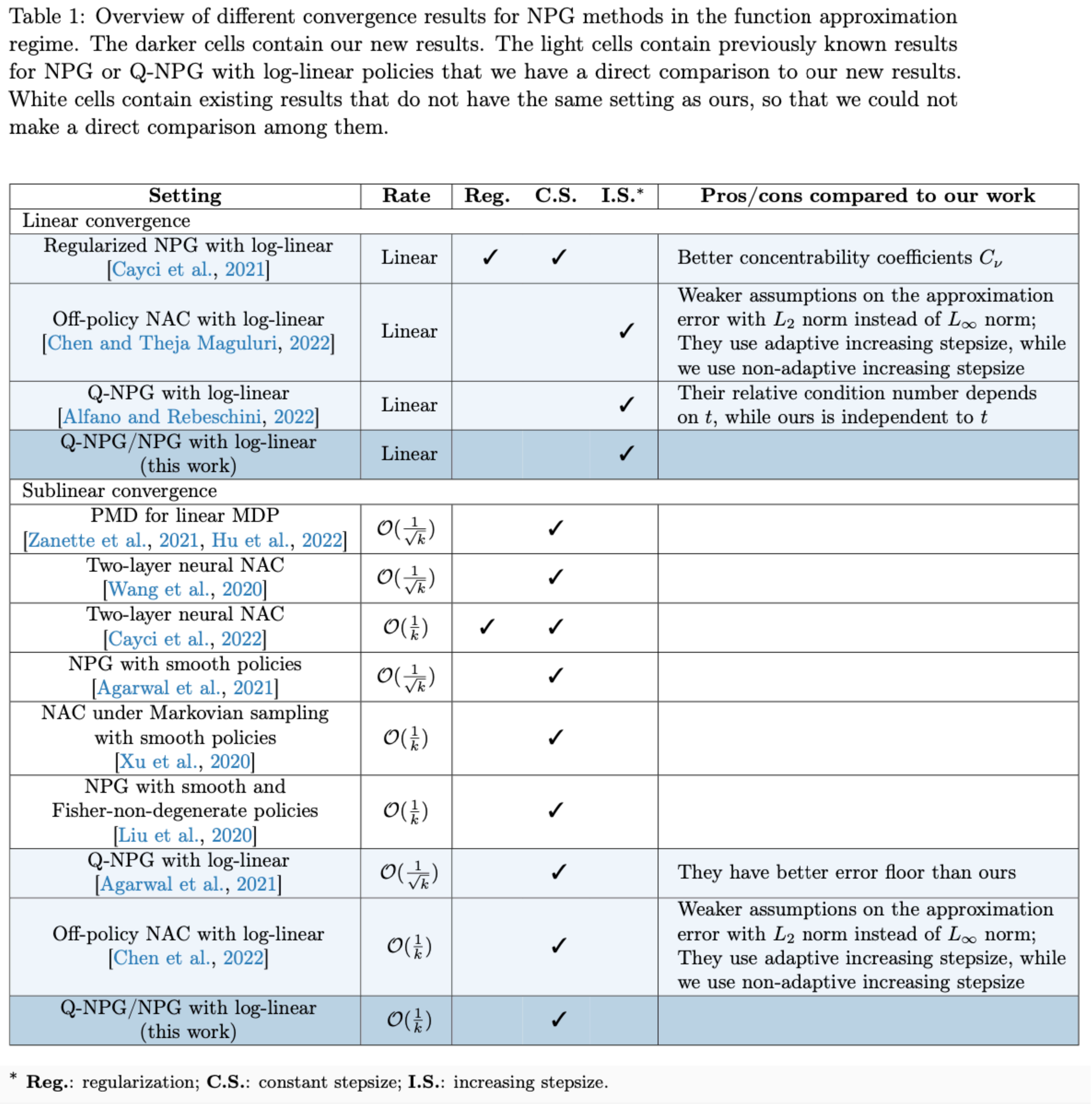


Table 1: Overview of different convergence results for NPG methods in the function approximation regime. The darker cells contain our new results. The light cells contain previously known results for NPG or Q-NPG with log-linear policies that we have a direct comparison to our new results. White cells contain existing results that do not have the same setting as ours, so that we could not make a direct comparison among them.

Setting	Rate	Reg.	C.S.	I.S.*	Pros/cons compared to our work
Linear convergence					
Regularized NPG with log-linear [Cayci et al., 2021]	Linear	✓	✓		Better concentrability coefficients C_ν
Off-policy NAC with log-linear [Chen and Theja Maguluri, 2022]	Linear			✓	Weaker assumptions on the approximation error with L_2 norm instead of L_∞ norm; They use adaptive increasing stepsize, while we use non-adaptive increasing stepsize
Q-NPG with log-linear [Alfano and Rebeschini, 2022]	Linear			✓	Their relative condition number depends on t, while ours is independent to t
Q-NPG/NPG with log-linear (this work)	Linear			✓	
Sublinear convergence					
PMD for linear MDP [Zanette et al., 2021, Hu et al., 2022]	$\mathcal{O}(\frac{1}{\sqrt{k}})$		✓		
Two-layer neural NAC [Wang et al., 2020]	$\mathcal{O}(\frac{1}{\sqrt{k}})$		✓		
Two-layer neural NAC [Cayci et al., 2022]	$\mathcal{O}(\frac{1}{k})$	✓	✓		
NPG with smooth policies [Agarwal et al., 2021]	$\mathcal{O}(\frac{1}{\sqrt{k}})$		✓		
NAC under Markovian sampling with smooth policies [Xu et al., 2020]	$\mathcal{O}(\frac{1}{k})$		✓		
NPG with smooth and Fisher-non-degenerate policies [Liu et al., 2020]	$\mathcal{O}(\frac{1}{k})$		✓		
Q-NPG with log-linear [Agarwal et al., 2021]	$\mathcal{O}(\frac{1}{\sqrt{k}})$		✓		They have better error floor than ours
Off-policy NAC with log-linear [Chen et al., 2022]	$\mathcal{O}(\frac{1}{k})$		✓		Weaker assumptions on the approximation error with L_2 norm instead of L_∞ norm; They use adaptive increasing stepsize, while we use non-adaptive increasing stepsize
Q-NPG/NPG with log-linear (this work)	$\mathcal{O}(\frac{1}{k})$		✓		

* **Reg.**: regularization; **C.S.**: constant stepsize; **I.S.**: increasing stepsize.