Trust Region
Policy Optimization & NPG

Recap on NPG:

At iteration t:

$$\max_{\pi_\theta} \mathbb{E}_{s\sim d_\mu^{\pi_{\theta_t}}}\left[\mathbb{E}_{a\sim\pi_\theta(s)} A^{\pi_{\theta_t}}(s,a)\right]$$

$$\text{s.t., } KL\left(\rho_{\pi_{\theta_t}} | \rho_{\pi_\theta}\right) \leq \delta$$

Intuition: maximize local adv subject to being incremental (in KL);

Recap on NPG:

At iteration t:

$$\max_{\pi_\theta} \mathbb{E}_{s\sim d_\mu^{\pi_{\theta_t}}}\left[\mathbb{E}_{a\sim\pi_\theta(s)} A^{\pi_{\theta_t}}(s,a)\right] \longrightarrow$$ First-order Taylor expansion at θ_t

$$\text{s.t., } KL\left(\rho_{\pi_{\theta_t}} | \rho_{\pi_\theta}\right) \leq \delta \longrightarrow$$ second-order Taylor expansion at θ_t

Intuition: maximize local adv subject to being incremental (in KL);

Recap on NPG:

At iteration t:

$$\max_{\pi_\theta} \mathbb{E}_{s\sim d_\mu^{\pi_{\theta_t}}}\left[\mathbb{E}_{a\sim\pi_\theta(s)} A^{\pi_{\theta_t}}(s,a)\right] \longrightarrow$$ First-order Taylor expansion at θ_t

s.t., $KL\left(\rho_{\pi_{\theta_t}} | \rho_{\pi_\theta}\right) \leq \delta \longrightarrow$ second-order Taylor expansion at θ_t

Intuition: maximize local adv subject to being incremental (in KL);

$$\max_{\theta} \nabla_\theta J(\pi_{\theta_t})^\top(\theta - \theta_t)$$

s.t. $(\theta - \theta_t)^\top F_{\theta_t}(\theta - \theta_t) \leq \delta$

Recap on NPG:

At iteration t:

$$\max_{\pi_\theta} \mathbb{E}_{s\sim d_\mu^{\pi_{\theta_t}}}\left[\mathbb{E}_{a\sim\pi_\theta(s)} A^{\pi_{\theta_t}}(s,a)\right]$$ → First-order Taylor expansion at θ_t

s.t., $KL\left(\rho_{\pi_{\theta_t}} | \rho_{\pi_\theta}\right) \leq \delta$ → second-order Taylor expansion at θ_t

Intuition: maximize local adv subject to being incremental (in KL);

$$\max_{\theta} \nabla_\theta J(\pi_{\theta_t})^\top (\theta - \theta_t)$$

s.t. $(\theta - \theta_t)^\top F_{\theta_t}(\theta - \theta_t) \leq \delta$

$$\theta_{t+1} = \theta_t + \eta F_{\theta_t}^{-1} \nabla_\theta J(\pi_{\theta_t})$$ ←

NPG

Recap on NPG:

At iteration t:

$$\max_{\pi_\theta} \mathbb{E}_{s\sim d_\mu^{\pi_{\theta_t}}}\left[\mathbb{E}_{a\sim\pi_\theta(s)} A^{\pi_{\theta_t}}(s,a)\right]$$

First-order Taylor expansion at θ_t

s.t., $KL\left(\rho_{\pi_{\theta_t}} | \rho_{\pi_\theta}\right) \leq \delta$

second-order Taylor expansion at θ_t

Intuition: maximize local adv subject to being incremental (in KL);

$$\max_{\theta} \nabla_\theta J(\pi_{\theta_t})^\top(\theta - \theta_t)$$

s.t. $(\theta - \theta_t)^\top F_{\theta_t}(\theta - \theta_t) \leq \delta$

$$\theta_{t+1} = \theta_t + \eta F_{\theta_t}^{-1} \nabla_\theta J(\pi_{\theta_t})$$

NPG

$$F_{\theta_t} := \mathbb{E}_{s,a\sim d_\mu^{\pi_{\theta_t}}}\left[\nabla_\theta \ln \pi_{\theta_t}(a|s)\left(\nabla_\theta \ln \pi_{\theta_t}(a|s)\right)^\top\right] \in \mathbb{R}^{dim_\theta \times dim_\theta}$$

Outline for Today:

1. Derivation of the closed-form NPG update

2. Intuitive Explanation of Natural (Policy) Gradient

3. Review of Policy Optimization (API, CPI, PG, and NPG) & a new algorithm

At iteration t, NPG solves a convex constrained optimization problem:

$$\max_{\theta} \nabla_\theta J(\pi_{\theta_t})^\top (\theta - \theta_t)$$

$$\text{s.t. } (\theta - \theta_t)^\top F_{\theta_t} (\theta - \theta_t) \leq \delta$$

At iteration t, NPG solves a convex constrained optimization problem:

$$\max_{\theta} \nabla_{\theta} J(\pi_{\theta_t})^{\top}(\theta - \theta_t)$$

$$\text{s.t. } (\theta - \theta_t)^{\top} F_{\theta_t} (\theta - \theta_t) \leq \delta$$

$\xrightarrow{\text{Notation simplification}}$

At iteration t, NPG solves a convex constrained optimization problem:

$$\max_{\theta} \nabla_\theta J(\pi_{\theta_t})^\top (\theta - \theta_t) \quad \text{s.t. } (\theta - \theta_t)^\top F_{\theta_t} (\theta - \theta_t) \le \delta$$

$$\xrightarrow{\text{Notation simplification}}$$

$$\max_{\Delta} \nabla^\top \Delta, \quad \text{s.t. } \Delta^\top F \Delta \le \delta$$

At iteration t, NPG solves a convex constrained optimization problem:

$$\max_{\theta} \nabla_\theta J(\pi_{\theta_t})^\top(\theta - \theta_t)$$
$$\text{s.t. } (\theta - \theta_t)^\top F_{\theta_t}(\theta - \theta_t) \le \delta$$

$$\xrightarrow{\text{Notation simplification}}$$

$$\max_{\Delta} \nabla^\top \Delta,$$
$$\text{s.t. } \Delta^\top F \Delta \le \delta$$

$$\Big\downarrow \; \widetilde{\Delta} := F^{1/2}\Delta$$

At iteration t, NPG solves a convex constrained optimization problem:

$$\max_{\theta} \nabla_\theta J(\pi_{\theta_t})^\top (\theta - \theta_t) \quad \text{s.t. } (\theta - \theta_t)^\top F_{\theta_t} (\theta - \theta_t) \le \delta$$

$$\xrightarrow{\text{Notation simplification}}$$

$$\max_{\Delta} \nabla^\top \Delta, \quad \text{s.t. } \Delta^\top F \Delta \le \delta$$

$$\Big\downarrow \; \widetilde{\Delta} := F^{1/2}\Delta$$

$$\max_{\widetilde{\Delta}} \left(F^{-1/2}\nabla\right)^\top \widetilde{\Delta},$$

$$\text{s.t. } \widetilde{\Delta}^\top \widetilde{\Delta} \le \delta$$

At iteration t, NPG solves a convex constrained optimization problem:

$$\max_{\theta} \nabla_{\theta} J(\pi_{\theta_t})^{\top}(\theta - \theta_t)$$
$$\text{s.t. } (\theta - \theta_t)^{\top} F_{\theta_t} (\theta - \theta_t) \le \delta$$

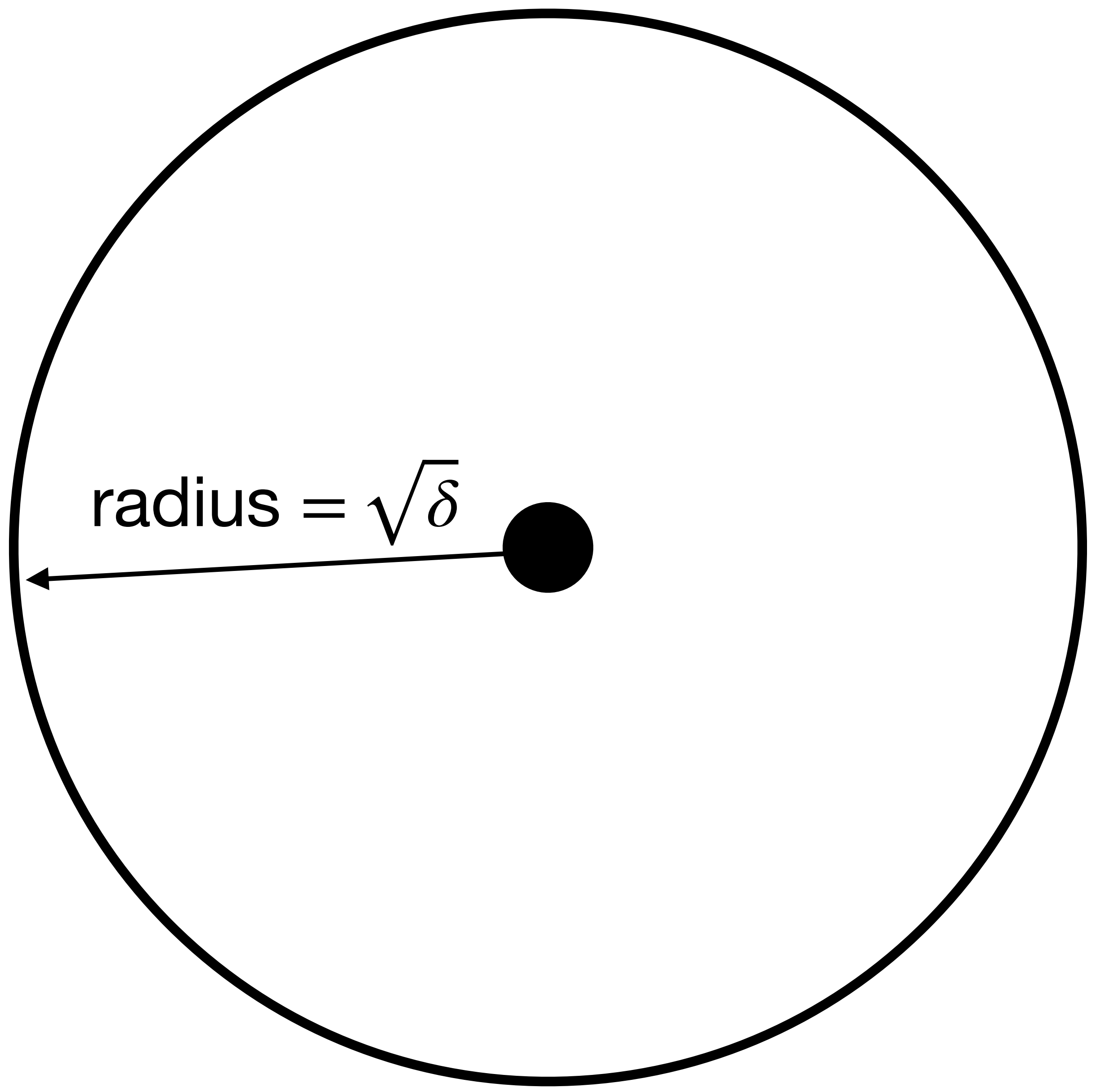
$\xrightarrow{\text{Notation simplification}}$

$$\max_{\Delta} \nabla^{\top} \Delta,$$
$$\text{s.t. } \Delta^{\top} F \Delta \le \delta$$

$\widetilde{\Delta} := F^{1/2}\Delta$

$$\max_{\widetilde{\Delta}} \left(F^{-1/2}\nabla\right)^{\top} \widetilde{\Delta},$$

$$\text{s.t. } \widetilde{\Delta}^{\top} \widetilde{\Delta} \le \delta$$



At iteration t, NPG solves a convex constrained optimization problem:

$$\max_{\theta} \nabla_{\theta} J(\pi_{\theta_t})^{\top}(\theta - \theta_t)$$
$$\text{s.t. } (\theta - \theta_t)^{\top} F_{\theta_t} (\theta - \theta_t) \le \delta$$

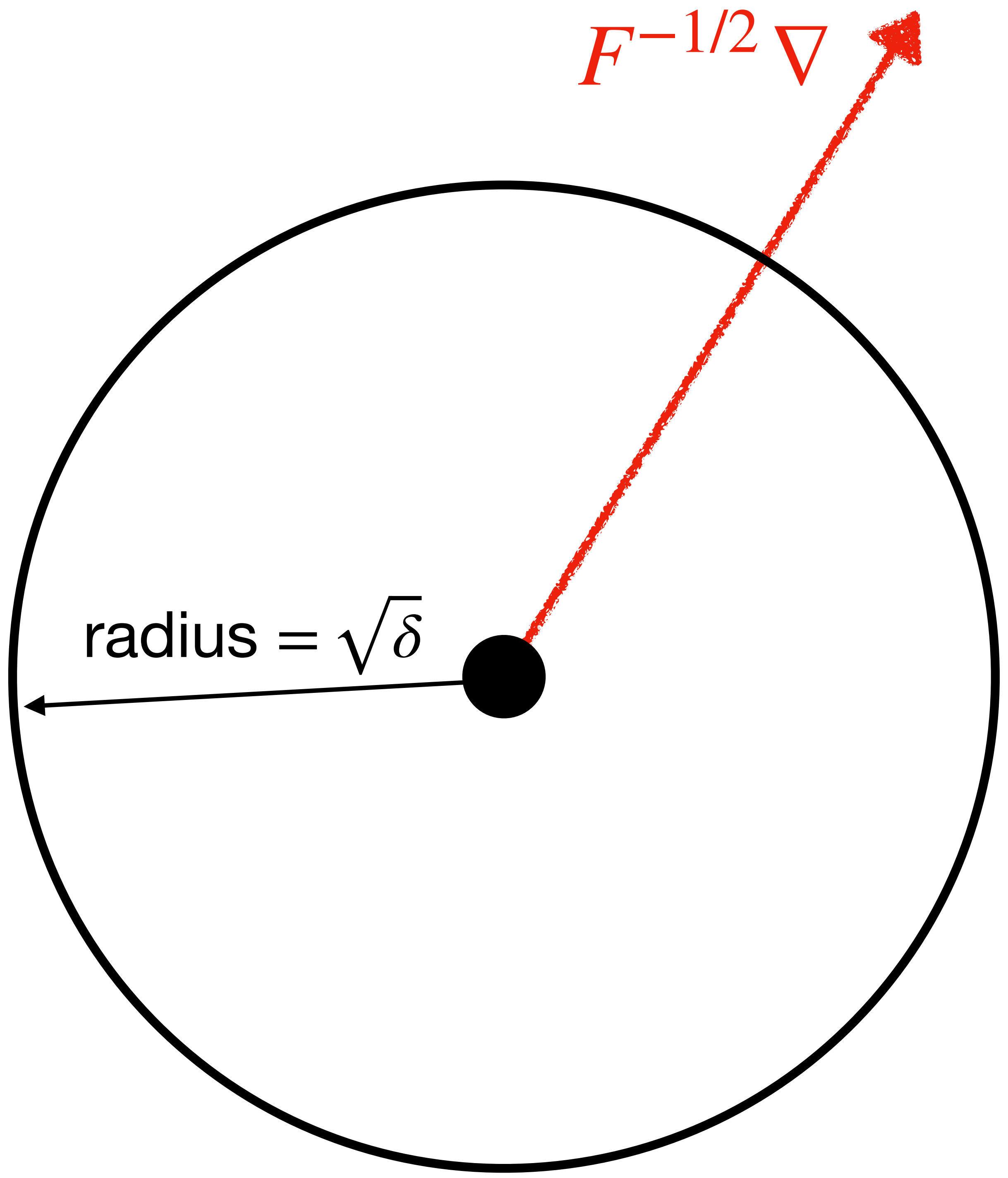
$$\xrightarrow{\text{Notation simplification}}$$

$$\max_{\Delta} \nabla^{\top} \Delta,$$
$$\text{s.t. } \Delta^{\top} F \Delta \le \delta$$

$$\Big\downarrow \ \widetilde{\Delta} := F^{1/2}\Delta$$

$$\max_{\widetilde{\Delta}} \left(F^{-1/2}\nabla\right)^{\top} \widetilde{\Delta},$$

$$\text{s.t. } \widetilde{\Delta}^{\top} \widetilde{\Delta} \le \delta$$



At iteration t, NPG solves a convex constrained optimization problem:

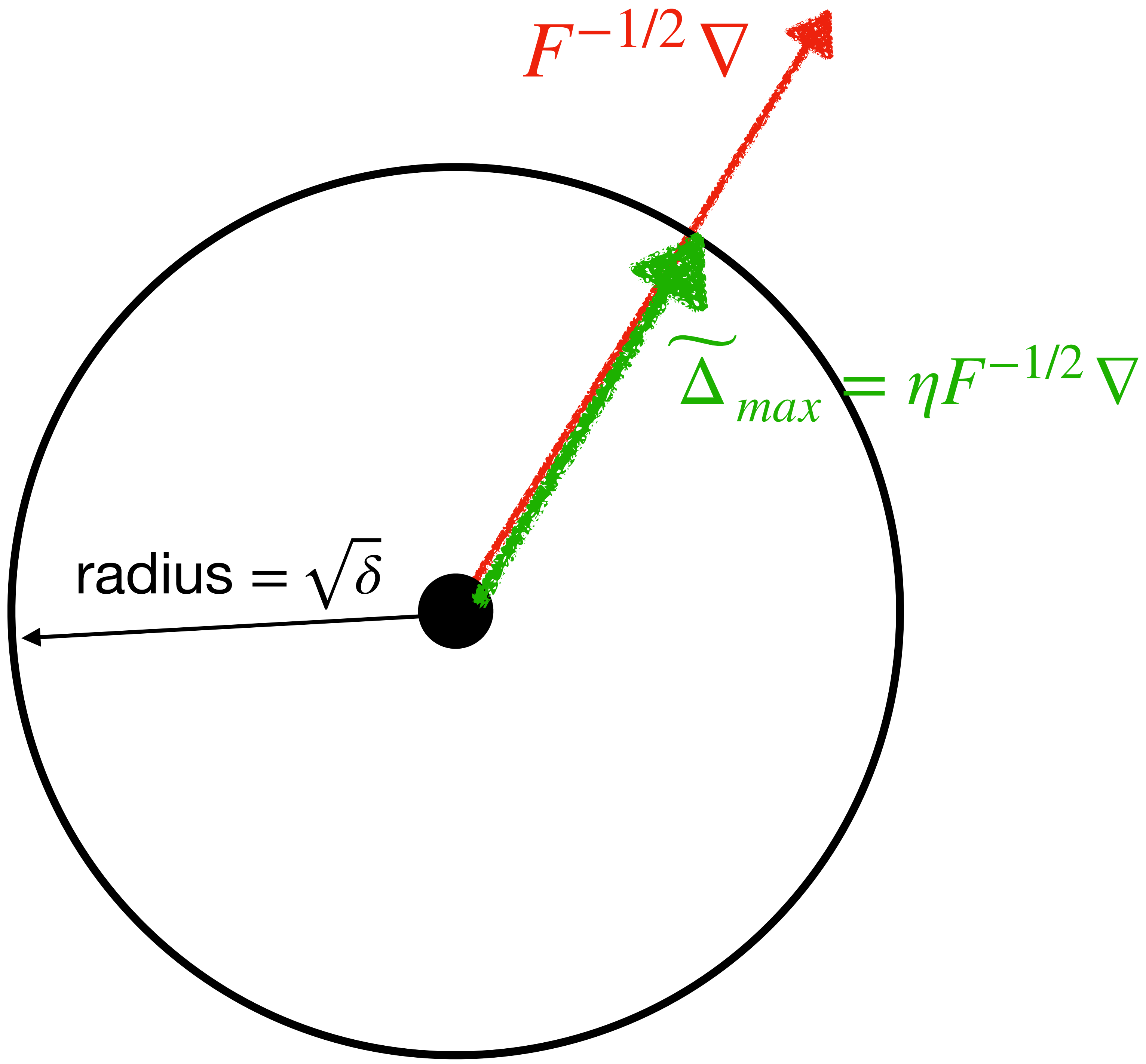
$$\max_{\theta} \nabla_{\theta} J(\pi_{\theta_t})^{\top}(\theta - \theta_t)$$
$$\text{s.t. } (\theta - \theta_t)^{\top} F_{\theta_t}(\theta - \theta_t) \leq \delta$$

$$\xrightarrow{\text{Notation simplification}}$$

$$\max_{\Delta} \nabla^{\top} \Delta,$$
$$\text{s.t. } \Delta^{\top} F \Delta \leq \delta$$

$$\Big\downarrow \; \widetilde{\Delta} := F^{1/2}\Delta$$

$$\max_{\widetilde{\Delta}} \left(F^{-1/2}\nabla\right)^{\top} \widetilde{\Delta},$$
$$\text{s.t. } \widetilde{\Delta}^{\top} \widetilde{\Delta} \leq \delta$$



At iteration t, NPG solves a convex constrained optimization problem:

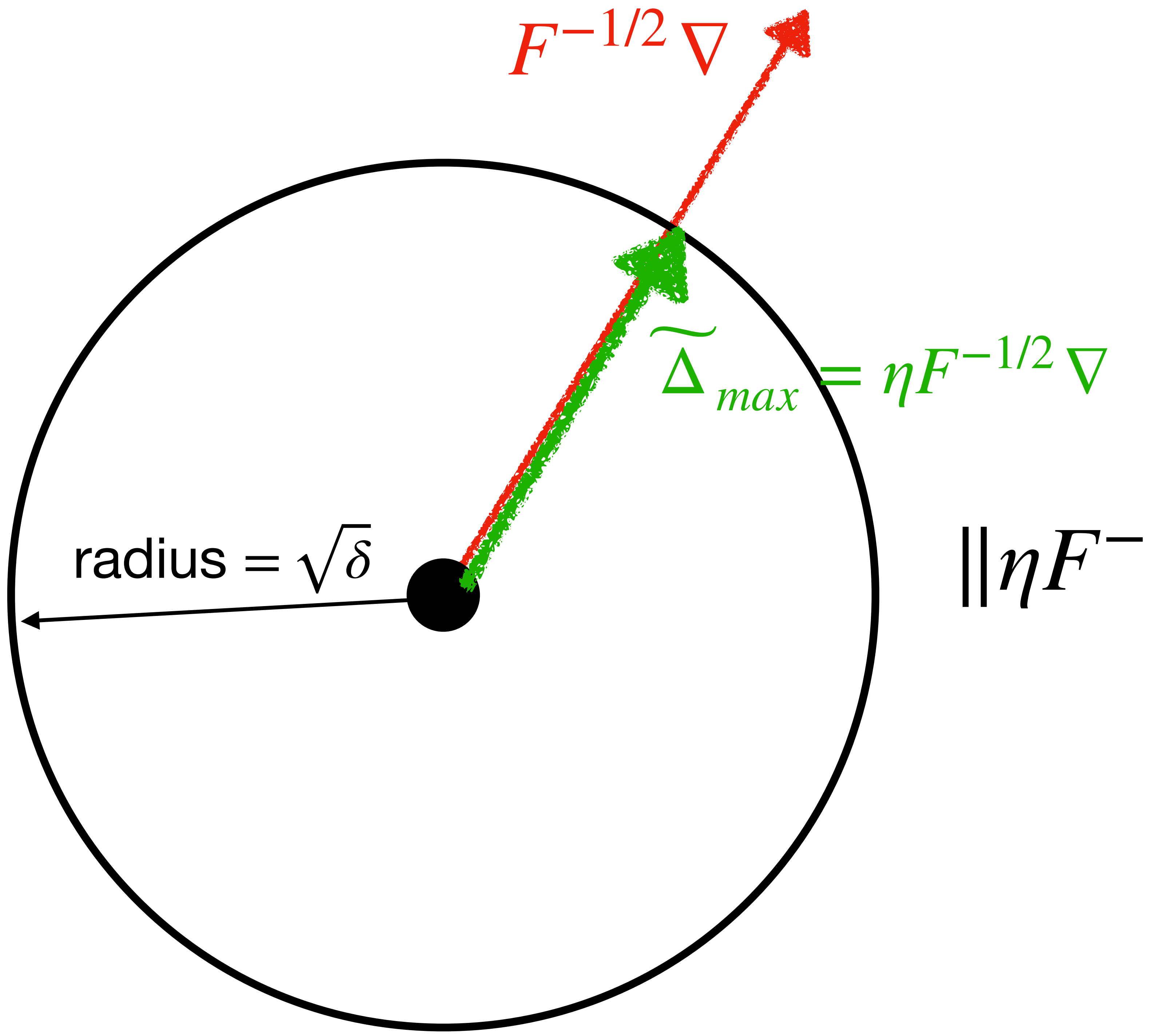
$$\max_{\theta} \nabla_\theta J(\pi_{\theta_t})^\top (\theta - \theta_t)$$
$$\text{s.t. } (\theta - \theta_t)^\top F_{\theta_t} (\theta - \theta_t) \le \delta$$

Notation simplification $\longrightarrow$

$$\max_{\Delta} \nabla^\top \Delta,$$
$$\text{s.t. } \Delta^\top F \Delta \le \delta$$

$\big\downarrow \; \widetilde{\Delta} := F^{1/2}\Delta$

$$\max_{\widetilde{\Delta}} \left(F^{-1/2}\nabla\right)^\top \widetilde{\Delta},$$
$$\text{s.t. } \widetilde{\Delta}^\top \widetilde{\Delta} \le \delta$$



$$\|\eta F^{-1/2}\nabla\|_2 = \sqrt{\delta}$$

At iteration t, NPG solves a convex constrained optimization problem:

$$\max_{\theta} \nabla_\theta J(\pi_{\theta_t})^\top (\theta - \theta_t)$$
$$\text{s.t. } (\theta - \theta_t)^\top F_{\theta_t} (\theta - \theta_t) \leq \delta$$

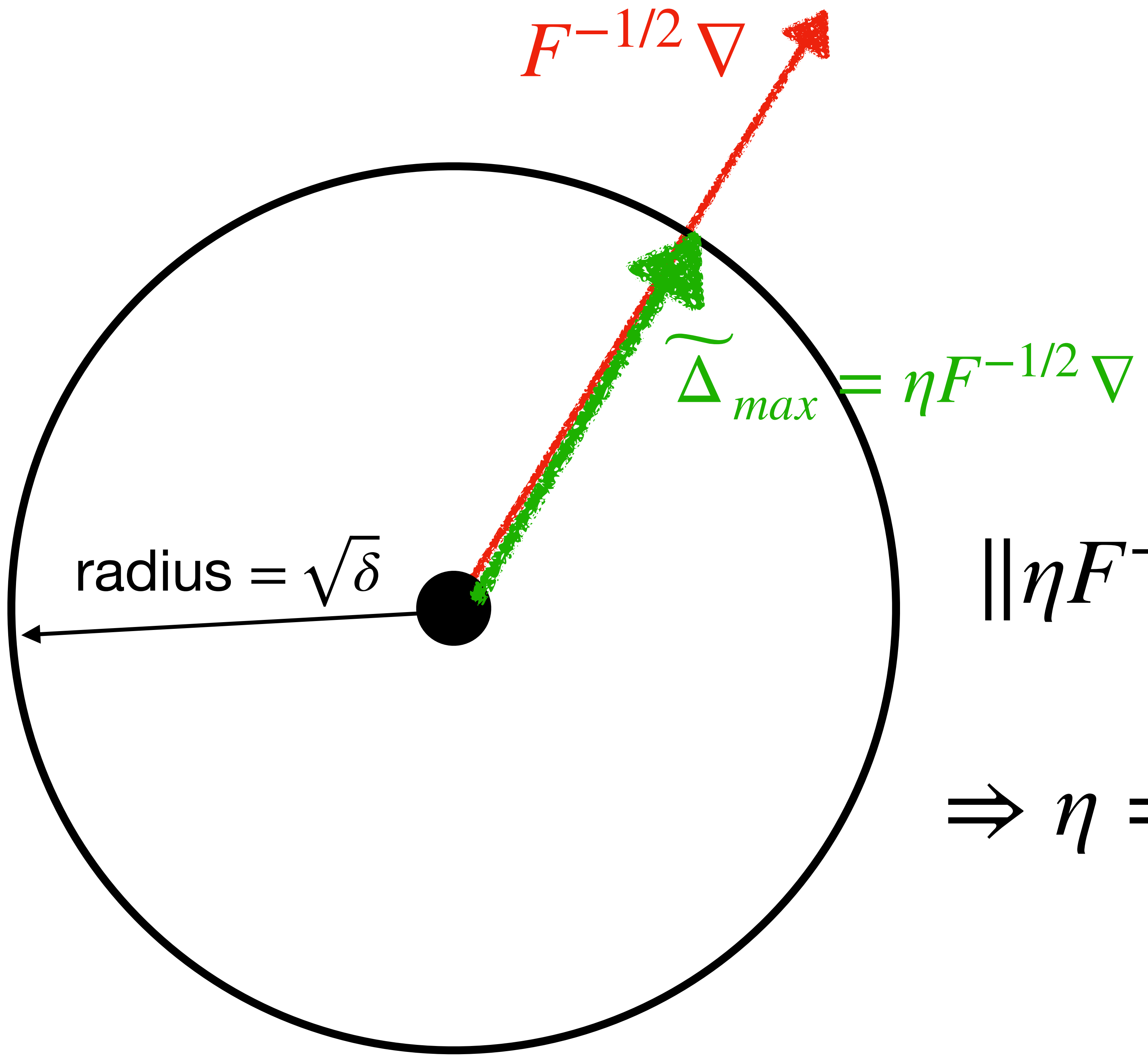
$\xrightarrow{\text{Notation simplification}}$

$$\max_{\Delta} \nabla^\top \Delta,$$
$$\text{s.t. } \Delta^\top F \Delta \leq \delta$$

$\Big\downarrow \; \widetilde{\Delta} := F^{1/2}\Delta$

$$\max_{\widetilde{\Delta}} \left(F^{-1/2}\nabla\right)^\top \widetilde{\Delta},$$

$$\text{s.t. } \widetilde{\Delta}^\top \widetilde{\Delta} \leq \delta$$



$$\|\eta F^{-1/2}\nabla\|_2 = \sqrt{\delta}$$
$$\Rightarrow \eta = \sqrt{\frac{\delta}{\nabla^\top F^{-1}\nabla}}$$

At iteration t, NPG solves a convex constrained optimization problem:

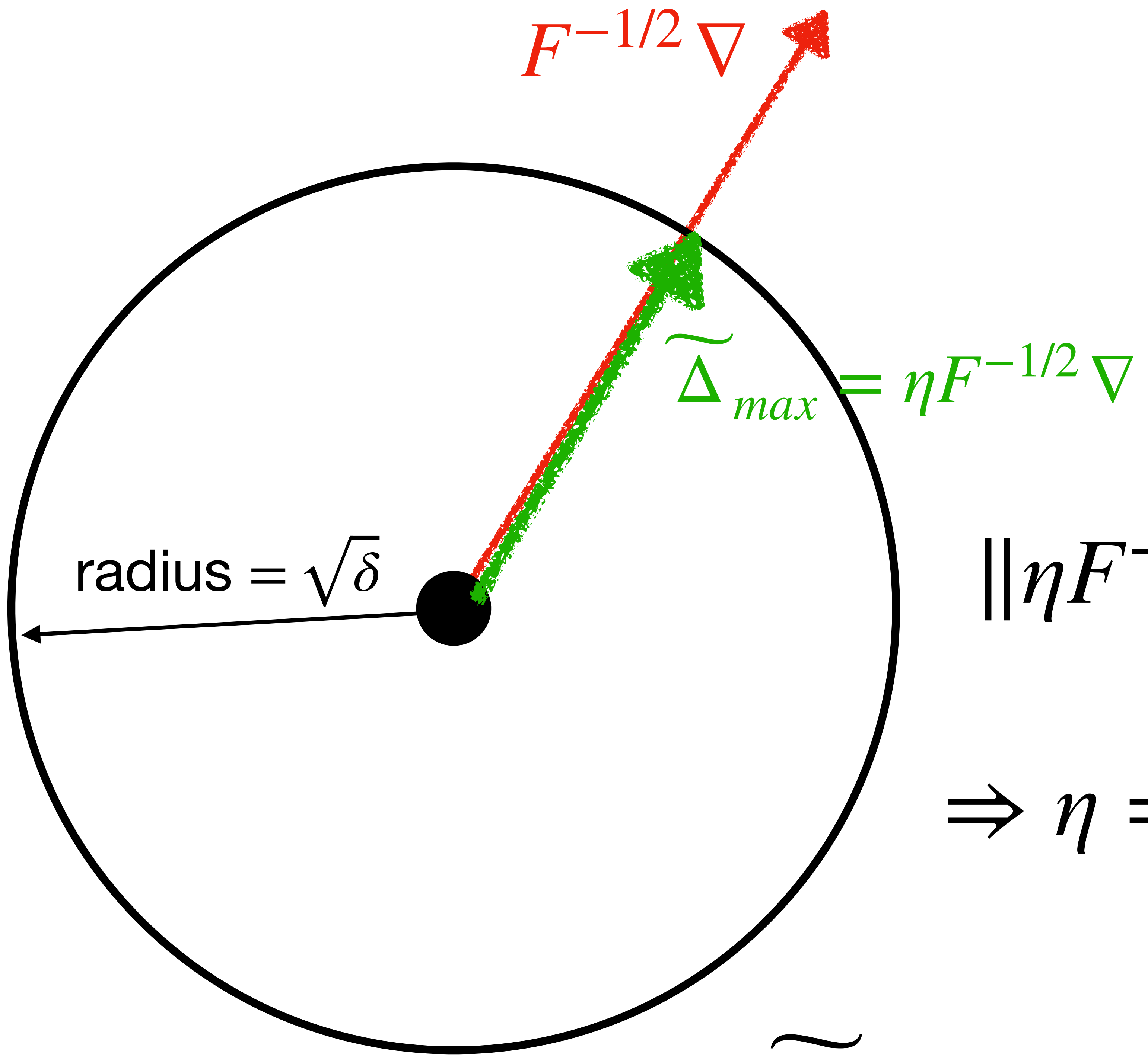
$$\max_{\theta} \nabla_\theta J(\pi_{\theta_t})^\top(\theta - \theta_t)$$
$$\text{s.t. } (\theta - \theta_t)^\top F_{\theta_t}(\theta - \theta_t) \le \delta$$

Notation simplification $\longrightarrow$

$$\max_{\Delta} \nabla^\top \Delta,$$
$$\text{s.t. } \Delta^\top F \Delta \le \delta$$

$\Big\downarrow \; \widetilde{\Delta} := F^{1/2}\Delta$

$$\max_{\widetilde{\Delta}} \left(F^{-1/2}\nabla\right)^\top \widetilde{\Delta},$$
$$\text{s.t. } \widetilde{\Delta}^\top \widetilde{\Delta} \le \delta$$



$$\|\eta F^{-1/2}\nabla\|_2 = \sqrt{\delta}$$
$$\Rightarrow \eta = \sqrt{\frac{\delta}{\nabla^\top F^{-1}\nabla}}$$
$$\widetilde{\Delta}_{max} := \sqrt{\frac{\delta}{\nabla^\top F^{-1}\nabla}} F^{-1/2}\nabla$$

At iteration t, NPG solves a convex constrained optimization problem:

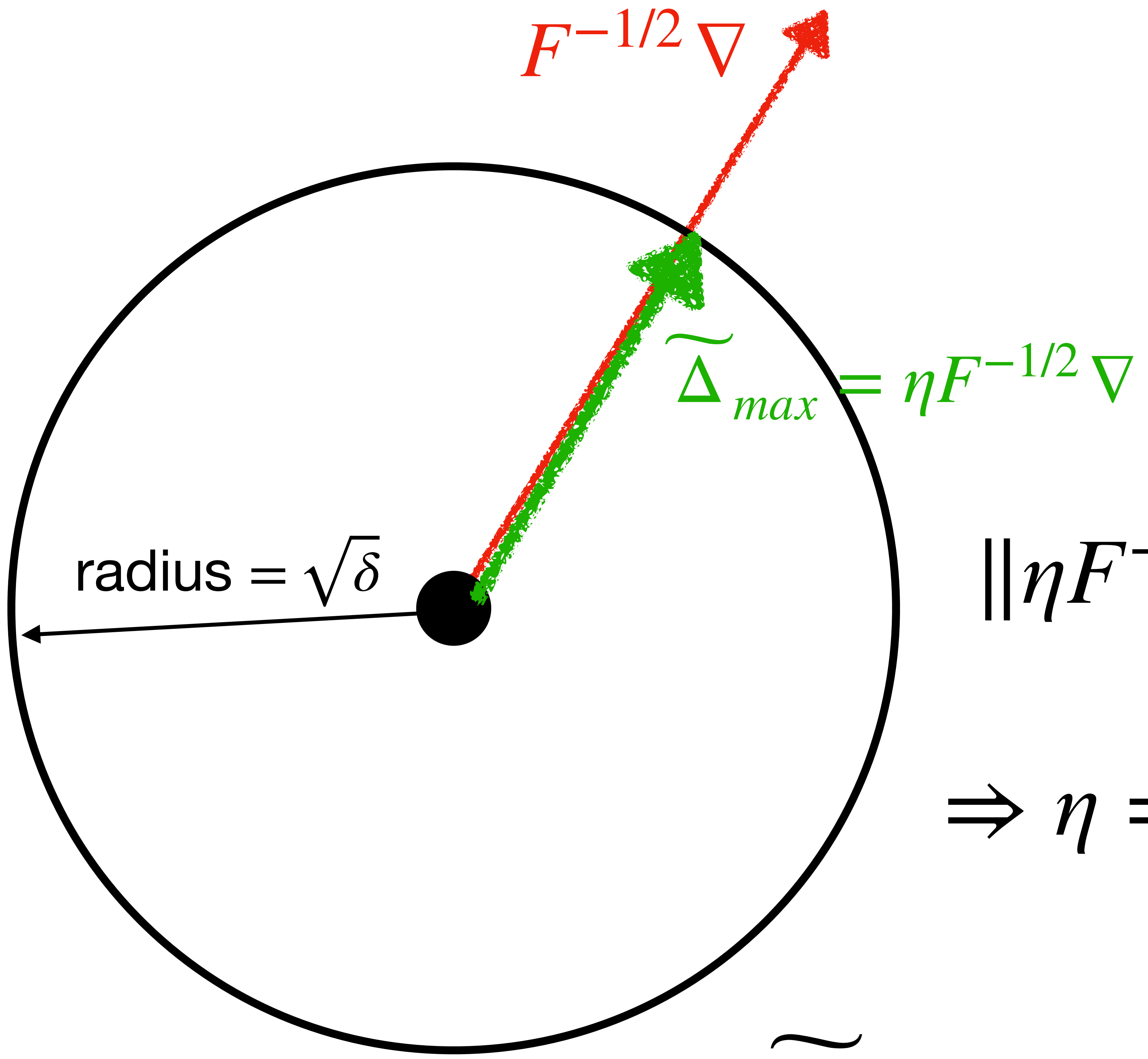
$$\max_{\theta} \nabla_\theta J(\pi_{\theta_t})^\top (\theta - \theta_t)$$
$$\text{s.t. } (\theta - \theta_t)^\top F_{\theta_t} (\theta - \theta_t) \le \delta$$

$\xrightarrow{\text{Notation simplification}}$

$$\max_{\Delta} \nabla^\top \Delta,$$
$$\text{s.t. } \Delta^\top F \Delta \le \delta$$

$\Big\downarrow\ \widetilde{\Delta} := F^{1/2}\Delta$

$$\max_{\widetilde{\Delta}} \left(F^{-1/2}\nabla\right)^\top \widetilde{\Delta},$$
$$\text{s.t. } \widetilde{\Delta}^\top \widetilde{\Delta} \le \delta$$



$$\|\eta F^{-1/2}\nabla\|_2 = \sqrt{\delta}$$
$$\Rightarrow \eta = \sqrt{\frac{\delta}{\nabla^\top F^{-1}\nabla}}$$
$$\widetilde{\Delta}_{max} := \sqrt{\frac{\delta}{\nabla^\top F^{-1}\nabla}} F^{-1/2}\nabla$$

$$\boxed{\Delta_{max} := \sqrt{\frac{\delta}{\nabla^\top F^{-1}\nabla}} F^{-1}\nabla}$$

At iteration t, NPG solves a convex constrained optimization problem:

$$\max_{\theta} \nabla_\theta J(\pi_{\theta_t})^\top (\theta - \theta_t)$$
$$\text{s.t. } (\theta - \theta_t)^\top F_{\theta_t} (\theta - \theta_t) \leq \delta$$

A more standard and straightway is to use Lagrange multiplier $\lambda \leq 0$:

At iteration t, NPG solves a convex constrained optimization problem:

$$\max_{\theta} \nabla_\theta J(\pi_{\theta_t})^\top (\theta - \theta_t)$$
$$\text{s.t. } (\theta - \theta_t)^\top F_{\theta_t} (\theta - \theta_t) \leq \delta$$

A more standard and straightway is to use Lagrange multiplier $\lambda \leq 0$:

$$\min_{\lambda \leq 0} \max_{\theta} \nabla_\theta J(\pi_{\theta_t})^\top (\theta - \theta_t) + \lambda \left((\theta - \theta_t)^\top F_{\theta_t} (\theta - \theta_t) - \delta \right)$$

At iteration t, NPG solves a convex constrained optimization problem:

$$\max_{\theta} \nabla_\theta J(\pi_{\theta_t})^\top(\theta - \theta_t)$$
$$\text{s.t. } (\theta - \theta_t)^\top F_{\theta_t}(\theta - \theta_t) \leq \delta$$

A more standard and straightway is to use Lagrange multiplier $\lambda \leq 0$:

$$\min_{\lambda \leq 0} \max_{\theta} \nabla_\theta J(\pi_{\theta_t})^\top(\theta - \theta_t) + \lambda \left((\theta - \theta_t)^\top F_{\theta_t}(\theta - \theta_t) - \delta \right)$$

(This is optional: Lagrange formulation is out of scope)

At iteration t, NPG solves a convex constrained optimization problem:

$$\max_{\theta} \nabla_\theta J(\pi_{\theta_t})^\top (\theta - \theta_t)$$
$$\text{s.t. } (\theta - \theta_t)^\top F_{\theta_t} (\theta - \theta_t) \le \delta$$

A more standard and straightway is to use Lagrange multiplier $\lambda \le 0$:

$$\min_{\lambda \le 0} \max_{\theta} \nabla_\theta J(\pi_{\theta_t})^\top (\theta - \theta_t) + \lambda \left((\theta - \theta_t)^\top F_{\theta_t} (\theta - \theta_t) - \delta \right)$$

(This is optional: Lagrange formulation is out of scope)

Summary: at this stage, we complete the NPG algorithm derivation

Outline for Today:

1. Derivation of the closed-form NPG update

2. Intuitive Explanation of Natural (Policy) Gradient

3. Review of Policy Optimization (API, CPI, PG, and NPG) & a new algorithm

NPG update: $\theta_1 = \theta_0 + \eta F_{\theta_0}^{-1} \nabla_{\theta_0}$

NPG update: $\theta_1 = \theta_0 + \eta F_{\theta_0}^{-1} \nabla_{\theta_0}$

$$KL\left(\rho_{\pi_{\theta_0}} | \rho_{\pi_\theta}\right) \leq \delta \Rightarrow (\theta - \theta_0)^\top F_{\theta_0}(\theta - \theta_0) \leq \delta$$

NPG update: $\theta_1 = \theta_0 + \eta F_{\theta_0}^{-1} \nabla_{\theta_0}$

$$KL\left(\rho_{\pi_{\theta_0}} | \rho_{\pi_\theta}\right) \leq \delta \Rightarrow (\theta - \theta_0)^\top F_{\theta_0}(\theta - \theta_0) \leq \delta$$

Our goal is to make sure two distributions do not change to much,
but parameters θ could potential change a lot!

NPG update: $\theta_1 = \theta_0 + \eta F_{\theta_0}^{-1} \nabla_{\theta_0}$

$$KL\left(\rho_{\pi_{\theta_0}} | \rho_{\pi_\theta}\right) \leq \delta \Rightarrow (\theta - \theta_0)^\top F_{\theta_0}(\theta - \theta_0) \leq \delta$$

Our goal is to make sure two distributions do not change to much,
but parameters θ could potential change a lot!

Consider special case where F_{θ_0} is a diagonal matrix: $F_{\theta_0} = \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{bmatrix}$

NPG update: $\theta_1 = \theta_0 + \eta F_{\theta_0}^{-1} \nabla_{\theta_0}$

$$KL\left(\rho_{\pi_{\theta_0}} | \rho_{\pi_\theta}\right) \leq \delta \Rightarrow (\theta - \theta_0)^\top F_{\theta_0}(\theta - \theta_0) \leq \delta$$

Our goal is to make sure two distributions do not change to much, but parameters θ could potential change a lot!

Consider special case where F_{θ_0} is a diagonal matrix: $F_{\theta_0} = \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{bmatrix}$

$$\forall i: \ \theta_1[i] = \theta_0[i] + \left(\eta \sigma_i^{-1}\right) \nabla_{\theta_0}[i]$$

NPG update: $\theta_1 = \theta_0 + \eta F_{\theta_0}^{-1} \nabla_{\theta_0}$

$$KL\left(\rho_{\pi_{\theta_0}} | \rho_{\pi_\theta}\right) \leq \delta \Rightarrow (\theta - \theta_0)^\top F_{\theta_0} (\theta - \theta_0) \leq \delta$$

Our goal is to make sure two distributions do not change to much, but parameters θ could potential change a lot!

Consider special case where F_{θ_0} is a diagonal matrix: $F_{\theta_0} = \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{bmatrix}$

$$\forall i: \ \theta_1[i] = \theta_0[i] + \left(\eta \sigma_i^{-1}\right) \nabla_{\theta_0}[i]$$

For tiny σ_i, we indeed have a **huge** learning rate, i.e., $\eta \sigma_i^{-1}$, at coordinate i !

NPG update: $\theta_1 = \theta_0 + \eta F_{\theta_0}^{-1} \nabla_{\theta_0}$

$$KL\left(\rho_{\pi_{\theta_0}} | \rho_{\pi_\theta}\right) \leq \delta \Rightarrow (\theta - \theta_0)^\top F_{\theta_0}(\theta - \theta_0) \leq \delta$$

Our goal is to make sure two distributions do not change to much,
but parameters θ could potential change a lot!

Consider special case where F_{θ_0} is a diagonal matrix: $F_{\theta_0} = \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{bmatrix}$

$$\forall i: \ \ \theta_1[i] = \theta_0[i] + \left(\eta\sigma_i^{-1}\right) \nabla_{\theta_0}[i]$$

For tiny σ_i, we indeed have a **huge** learning rate, i.e., $\eta\sigma_i^{-1}$, at coordinate i !

In other words, NPG **allows a big jump** on some coordinates which do not affect KL-div too much

Example of Natural Gradient on 1-d problem:

$$p_\theta = \left(\frac{\exp(\theta)}{1 + \exp(\theta)}, \frac{1}{1 + \exp(\theta)} \right)$$

$$g(\theta) = 100 \cdot p_\theta[1] + 1 \cdot p_\theta[2]$$

Example of Natural Gradient on 1-d problem:

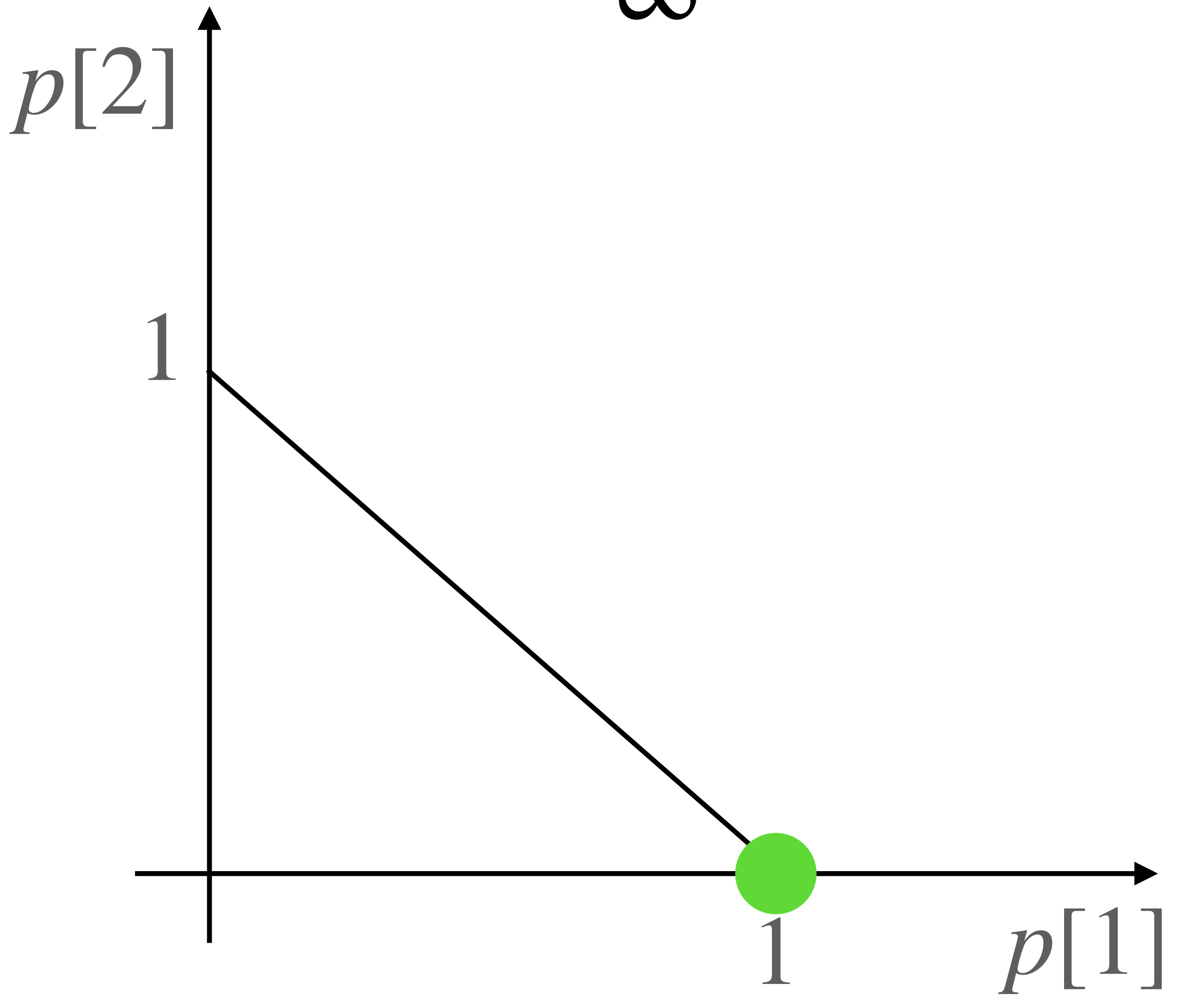
$$p_\theta = \left(\frac{\exp(\theta)}{1 + \exp(\theta)}, \frac{1}{1 + \exp(\theta)} \right)$$

$$g(\theta) = 100 \cdot p_\theta[1] + 1 \cdot p_\theta[2]$$

Example of Natural Gradient on 1-d problem:

$$p_\theta = \left(\frac{\exp(\theta)}{1+\exp(\theta)}, \frac{1}{1+\exp(\theta)} \right)$$

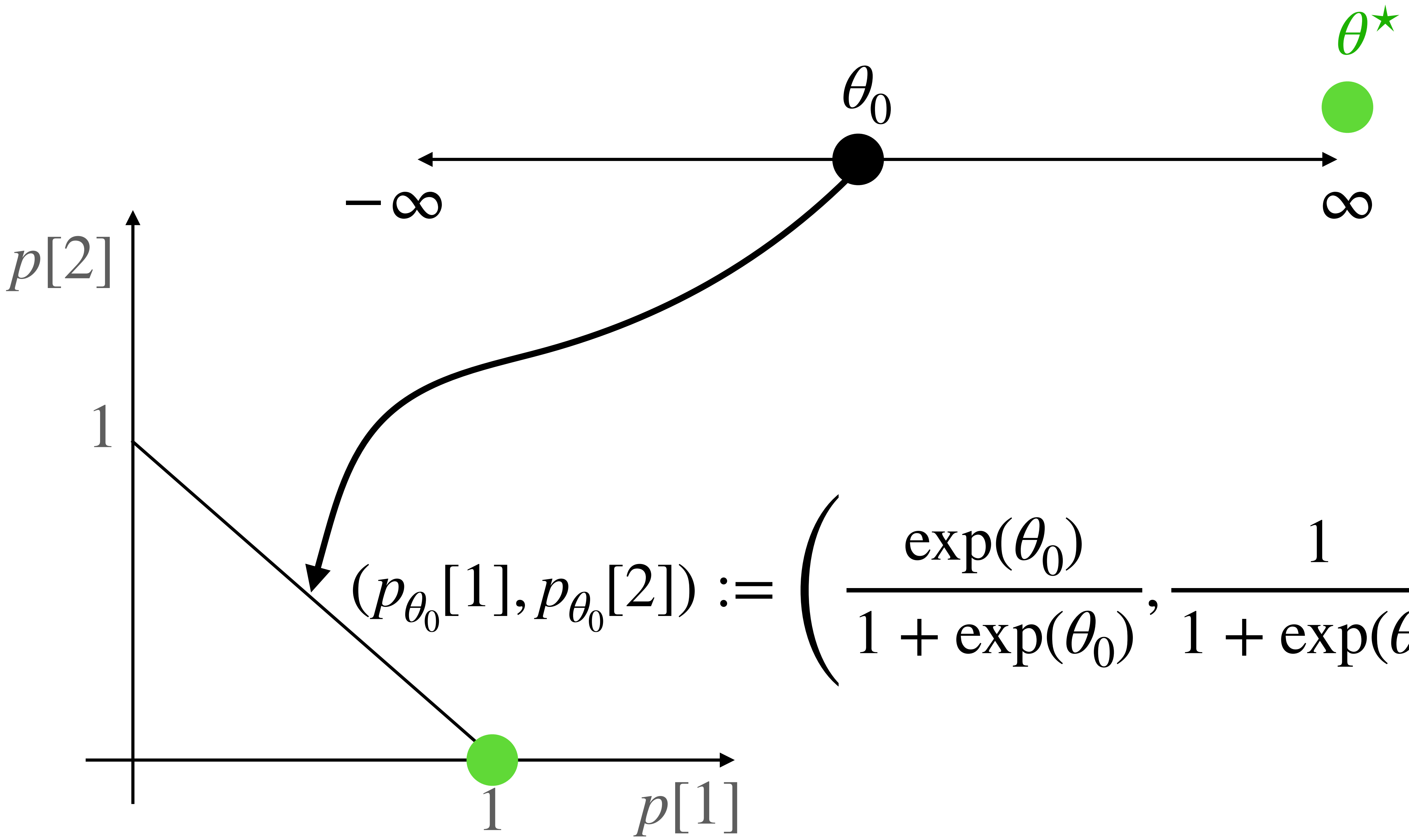
$$g(\theta) = 100 \cdot p_\theta[1] + 1 \cdot p_\theta[2]$$



Example of Natural Gradient on 1-d problem:

$$p_\theta = \left(\frac{\exp(\theta)}{1+\exp(\theta)}, \frac{1}{1+\exp(\theta)} \right)$$

$$g(\theta) = 100 \cdot p_\theta[1] + 1 \cdot p_\theta[2]$$



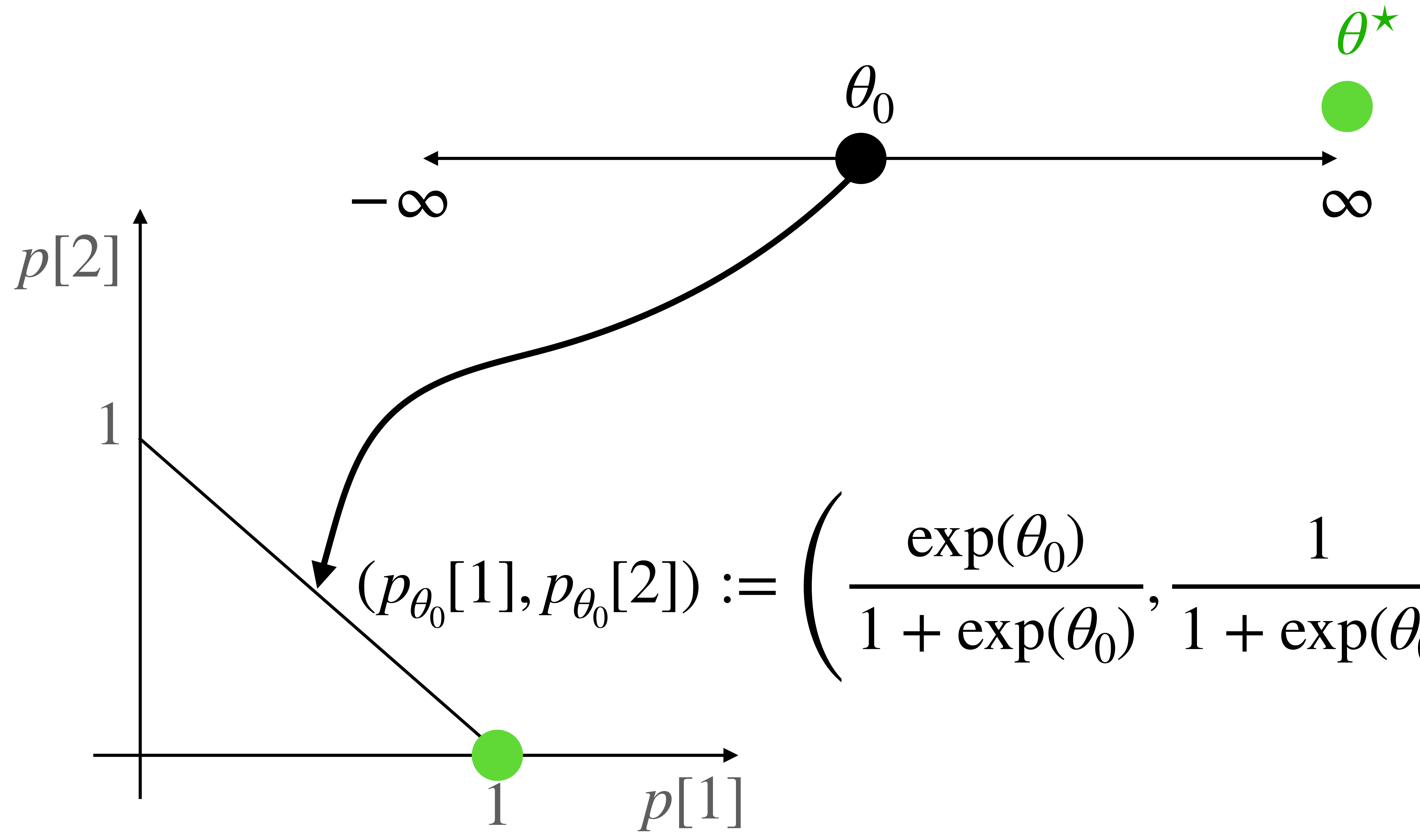
$$(p_{\theta_0}[1], p_{\theta_0}[2]) := \left(\frac{\exp(\theta_0)}{1+\exp(\theta_0)}, \frac{1}{1+\exp(\theta_0)} \right)$$

Example of Natural Gradient on 1-d problem:

$$p_\theta = \left(\frac{\exp(\theta)}{1+\exp(\theta)}, \frac{1}{1+\exp(\theta)} \right)$$

Fisher information scalar: $f_{\theta_0} = \dfrac{\exp(\theta_0)}{(1+\exp(\theta_0))^2}$

$$g(\theta) = 100 \cdot p_\theta[1] + 1 \cdot p_\theta[2]$$



$$(p_{\theta_0}[1], p_{\theta_0}[2]) := \left(\frac{\exp(\theta_0)}{1+\exp(\theta_0)}, \frac{1}{1+\exp(\theta_0)} \right)$$

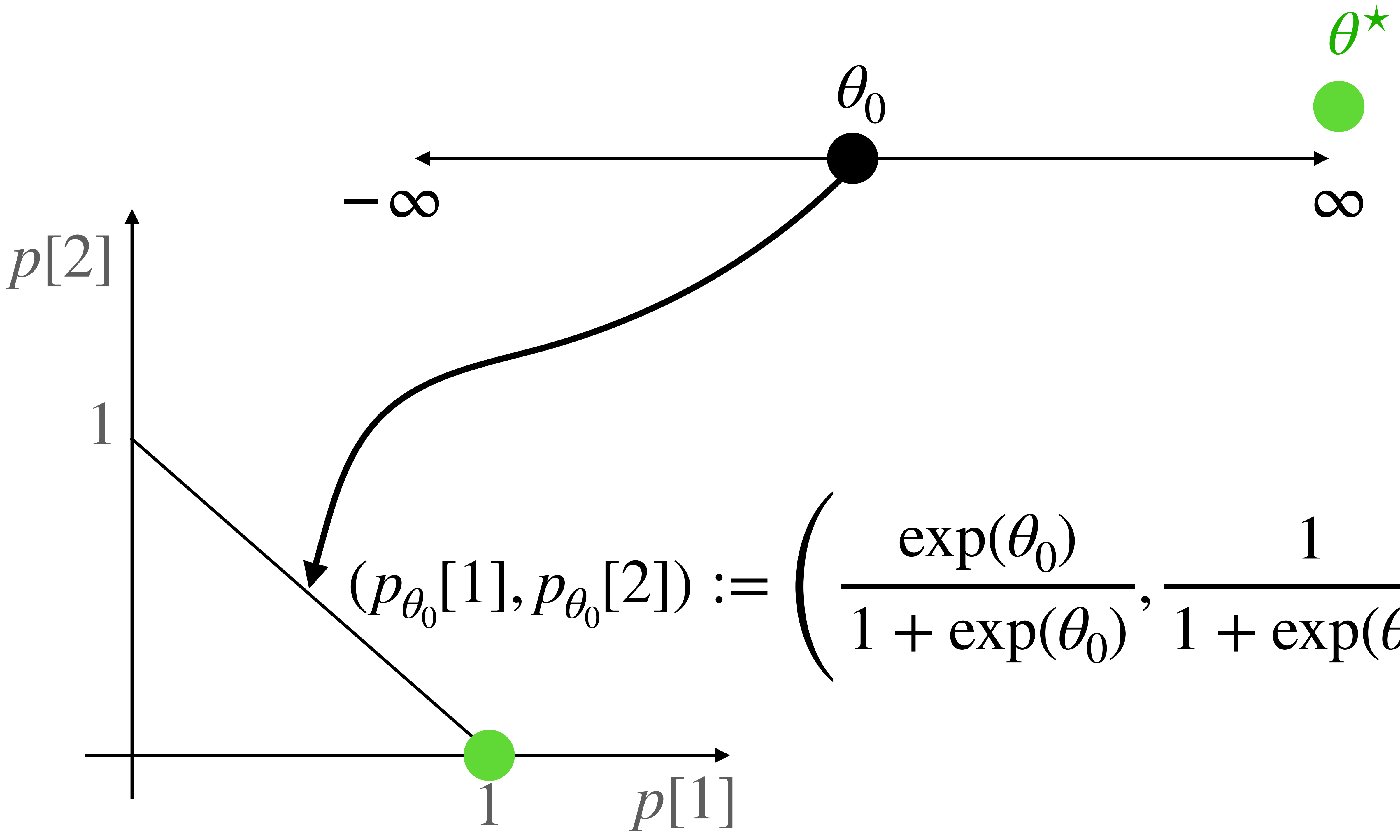
Example of Natural Gradient on 1-d problem:

$$p_\theta = \left(\frac{\exp(\theta)}{1+\exp(\theta)}, \frac{1}{1+\exp(\theta)} \right)$$

$$g(\theta) = 100 \cdot p_\theta[1] + 1 \cdot p_\theta[2]$$

Fisher information scalar: $f_{\theta_0} = \dfrac{\exp(\theta_0)}{(1+\exp(\theta_0))^2}$

Hence: $f_{\theta_0} \to 0^+$, as $\theta_0 \to \infty$



$$(p_{\theta_0}[1], p_{\theta_0}[2]) := \left(\frac{\exp(\theta_0)}{1+\exp(\theta_0)}, \frac{1}{1+\exp(\theta_0)} \right)$$

Example of Natural Gradient on 1-d problem:

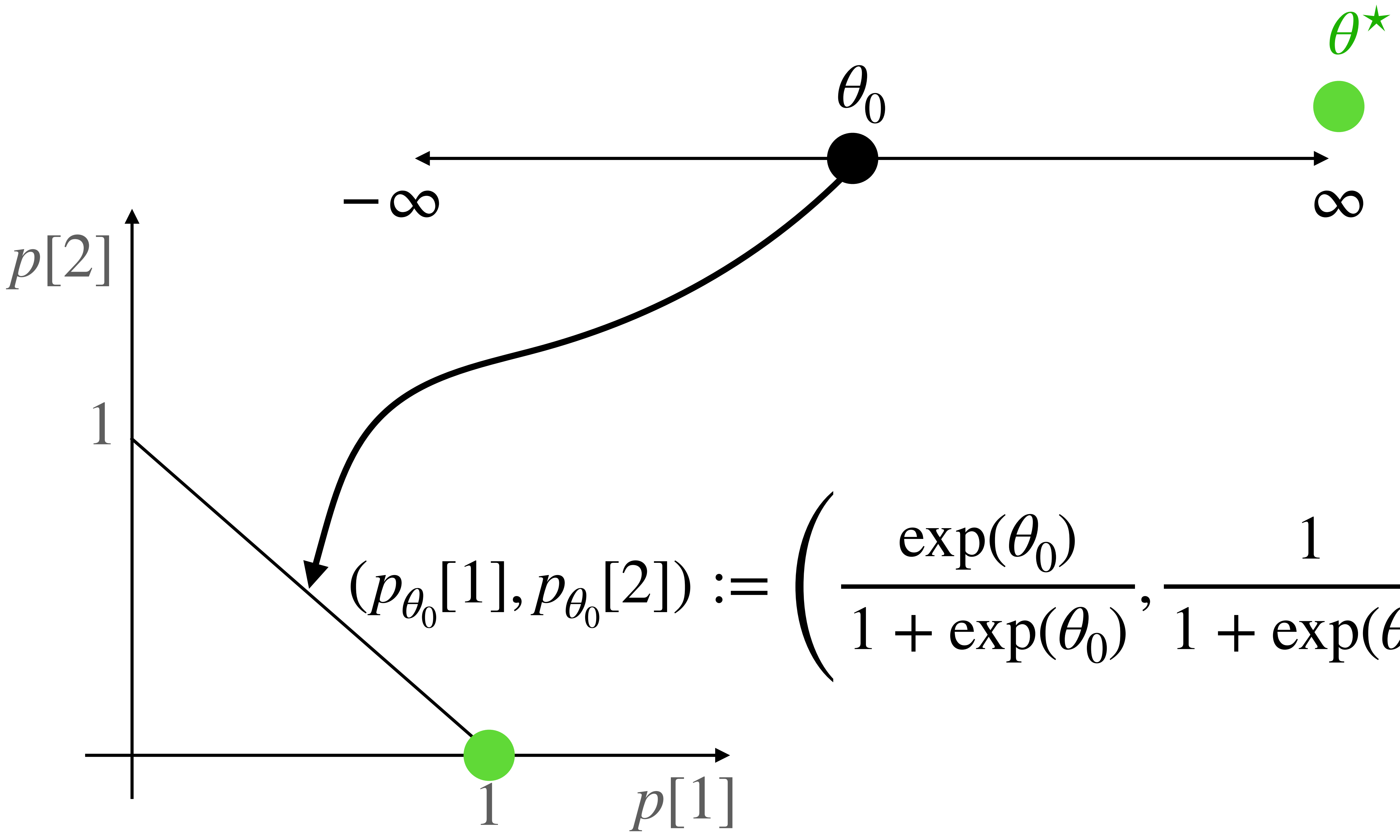
$$p_\theta = \left(\frac{\exp(\theta)}{1+\exp(\theta)}, \frac{1}{1+\exp(\theta)}\right)$$

$$g(\theta) = 100 \cdot p_\theta[1] + 1 \cdot p_\theta[2]$$

Fisher information scalar: $f_{\theta_0} = \dfrac{\exp(\theta_0)}{(1+\exp(\theta_0))^2}$

Hence: $f_{\theta_0} \to 0^+$, as $\theta_0 \to \infty$

NPG: $\theta_1 = \theta_0 + \eta \dfrac{g'(\theta_0)}{f_{\theta_0}}$

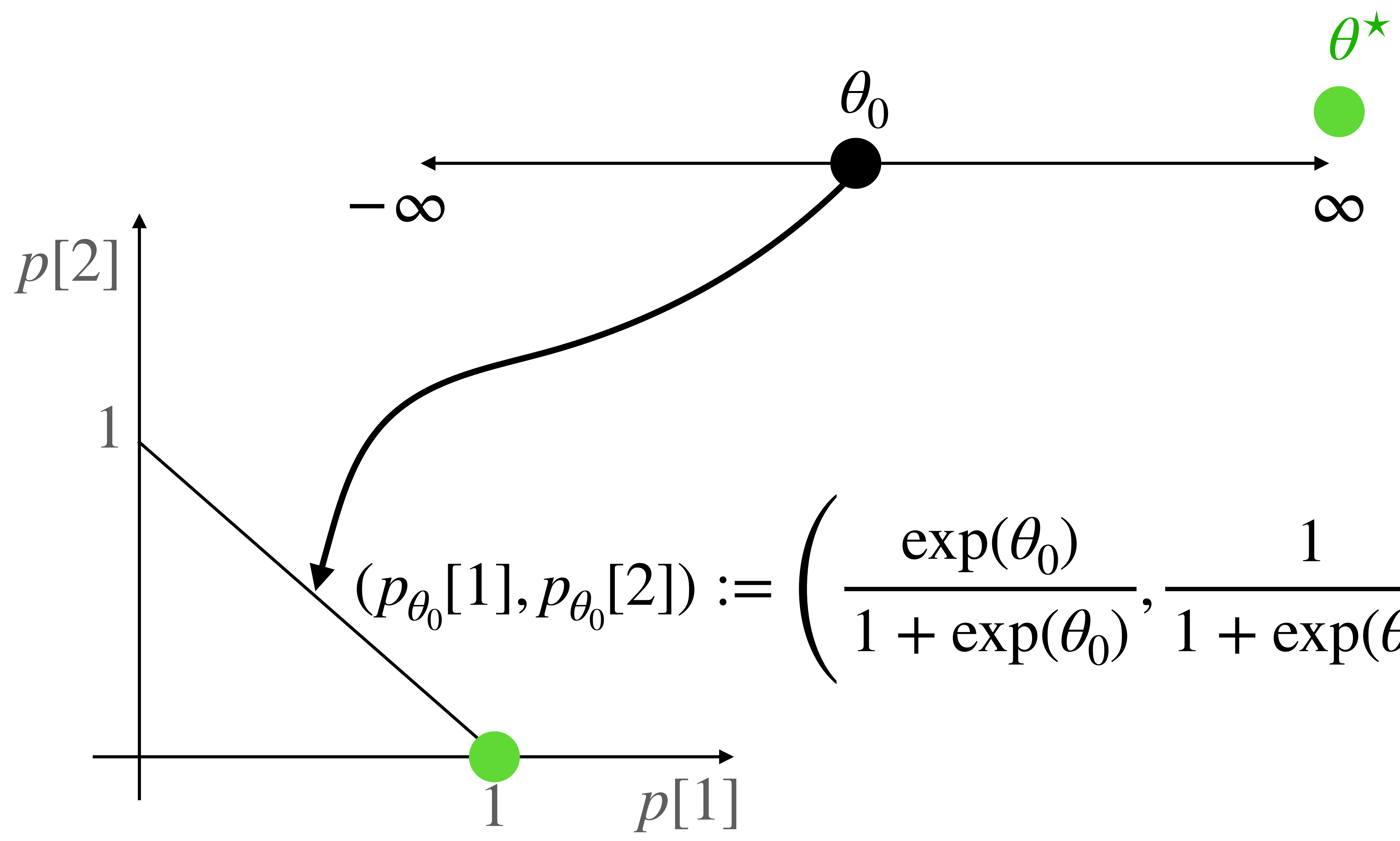


$$(p_{\theta_0}[1], p_{\theta_0}[2]) := \left(\frac{\exp(\theta_0)}{1+\exp(\theta_0)}, \frac{1}{1+\exp(\theta_0)}\right)$$

Example of Natural Gradient on 1-d problem:

$$p_\theta = \left(\frac{\exp(\theta)}{1+\exp(\theta)}, \frac{1}{1+\exp(\theta)} \right)$$

$$g(\theta) = 100 \cdot p_\theta[1] + 1 \cdot p_\theta[2]$$



$$(p_{\theta_0}[1], p_{\theta_0}[2]) := \left(\frac{\exp(\theta_0)}{1+\exp(\theta_0)}, \frac{1}{1+\exp(\theta_0)} \right)$$

Fisher information scalar: $f_{\theta_0} = \dfrac{\exp(\theta_0)}{(1+\exp(\theta_0))^2}$

Hence: $f_{\theta_0} \to 0^+$, as $\theta_0 \to \infty$

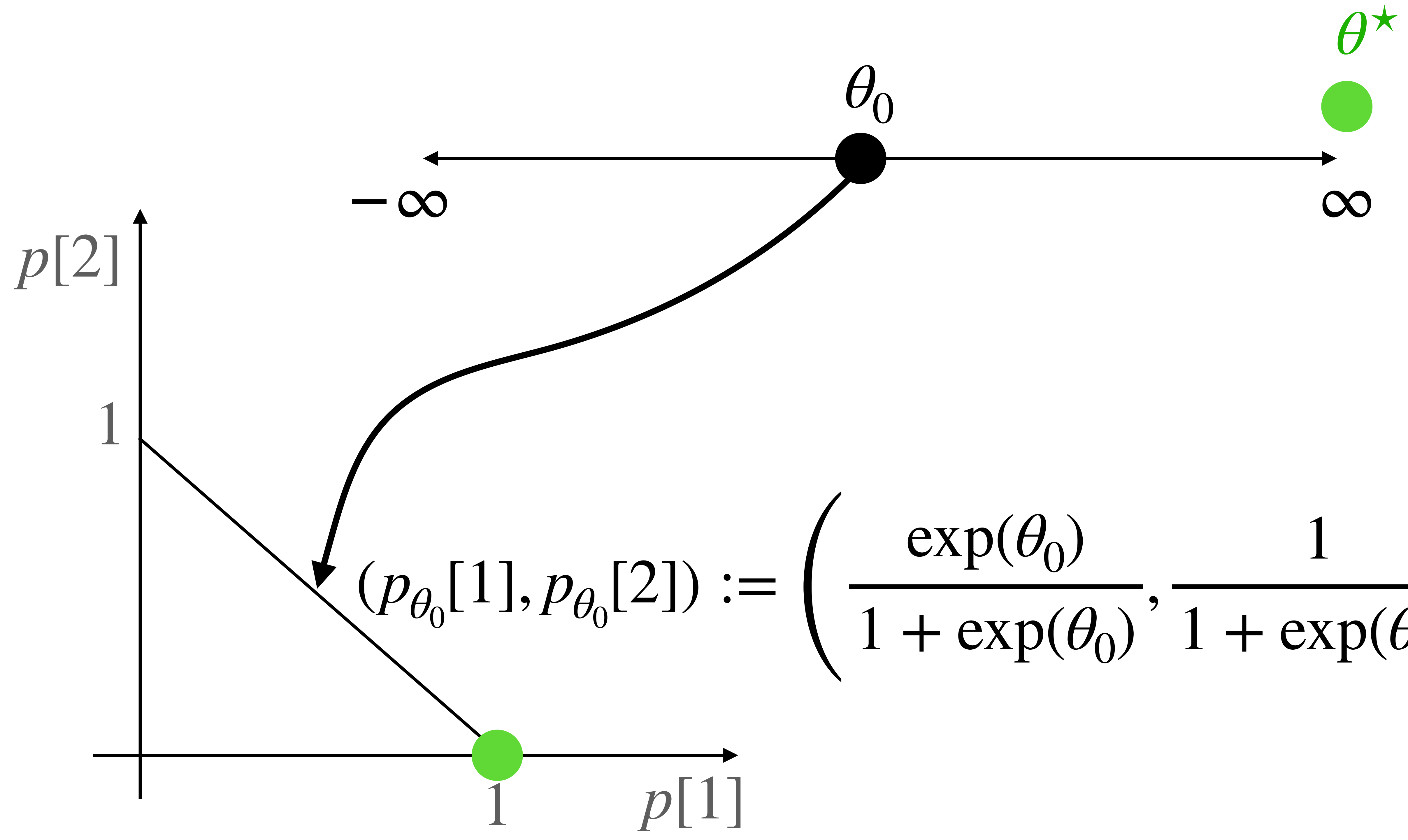
NPG: $\theta_1 = \theta_0 + \eta \dfrac{g'(\theta_0)}{f_{\theta_0}}$

GA: $\theta_1 = \theta_0 + \eta g'(\theta_0)$

Example of Natural Gradient on 1-d problem:

$$p_\theta = \left(\frac{\exp(\theta)}{1+\exp(\theta)}, \frac{1}{1+\exp(\theta)} \right)$$

$$g(\theta) = 100 \cdot p_\theta[1] + 1 \cdot p_\theta[2]$$



$$(p_{\theta_0}[1], p_{\theta_0}[2]) := \left(\frac{\exp(\theta_0)}{1+\exp(\theta_0)}, \frac{1}{1+\exp(\theta_0)} \right)$$

Fisher information scalar: $f_{\theta_0} = \dfrac{\exp(\theta_0)}{(1+\exp(\theta_0))^2}$

Hence: $f_{\theta_0} \to 0^+$, as $\theta_0 \to \infty$

NPG: $\theta_1 = \theta_0 + \eta \dfrac{g'(\theta_0)}{f_{\theta_0}}$

GA: $\theta_1 = \theta_0 + \eta g'(\theta_0)$

i.e., Plain GA in θ will move to $\theta = \infty$ at a constant speed,
while Natural GA can traverse faster and faster when θ gets bigger
(subject to the same learning rate)

Outline for Today:

1. Derivation of the closed-form NPG update

2. Intuitive Explanation of Natural (Policy) Gradient

(In HW2, try to compare PG and NPG, see how they perform differently in practice!)

3. Review of Policy Optimization (API, CPI, PG, and NPG) & a new algorithm

Review on Policy Optimization:

We have huge space space, i.e., $|S|$ might be $255^{3\times512\times512}$

We can only reset from initial state distribution $s_0 \sim \mu$

Review on Policy Optimization:

We have huge space space, i.e., $|S|$ might be $255^{3\times512\times512}$

We can only reset from initial state distribution $s_0 \sim \mu$

Numeration over state (e.g., a for loop) is not possible!

Review on Policy Optimization:

We have huge space space, i.e., $|S|$ might be $255^{3\times512\times512}$

We can only reset from initial state distribution $s_0 \sim \mu$

Numeration over state (e.g., a for loop) is not possible!

Goal: learn w/ function approximation

Review on Policy Optimization:

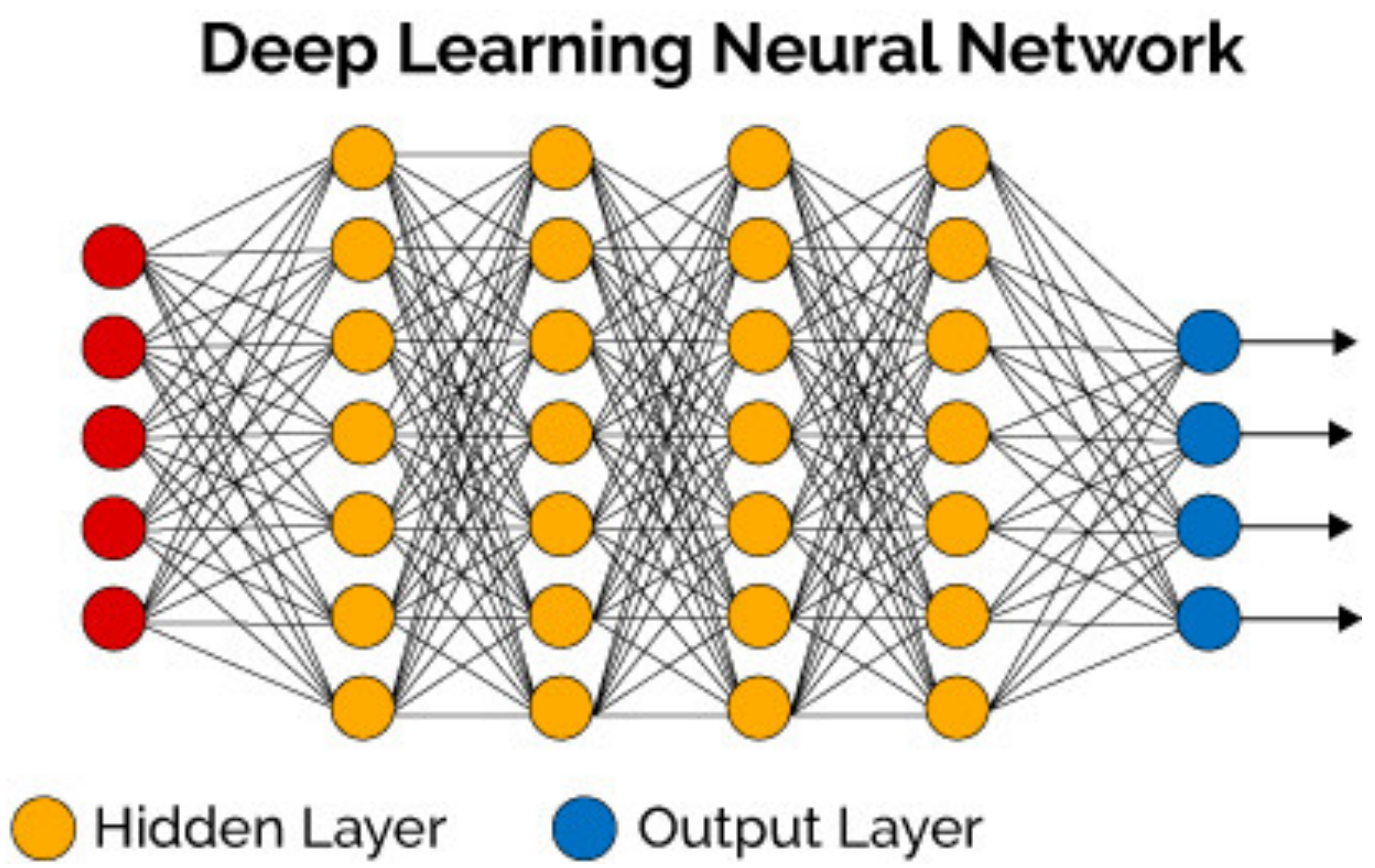
We have huge space space, i.e., $|S|$ might be $255^{3\times512\times512}$

We can only reset from initial state distribution $s_0 \sim \mu$

Numeration over state (e.g., a for loop) is not possible!

Goal: learn w/ function approximation

A Policy is a classifier w/ A many classes



Review on Policy Optimization:

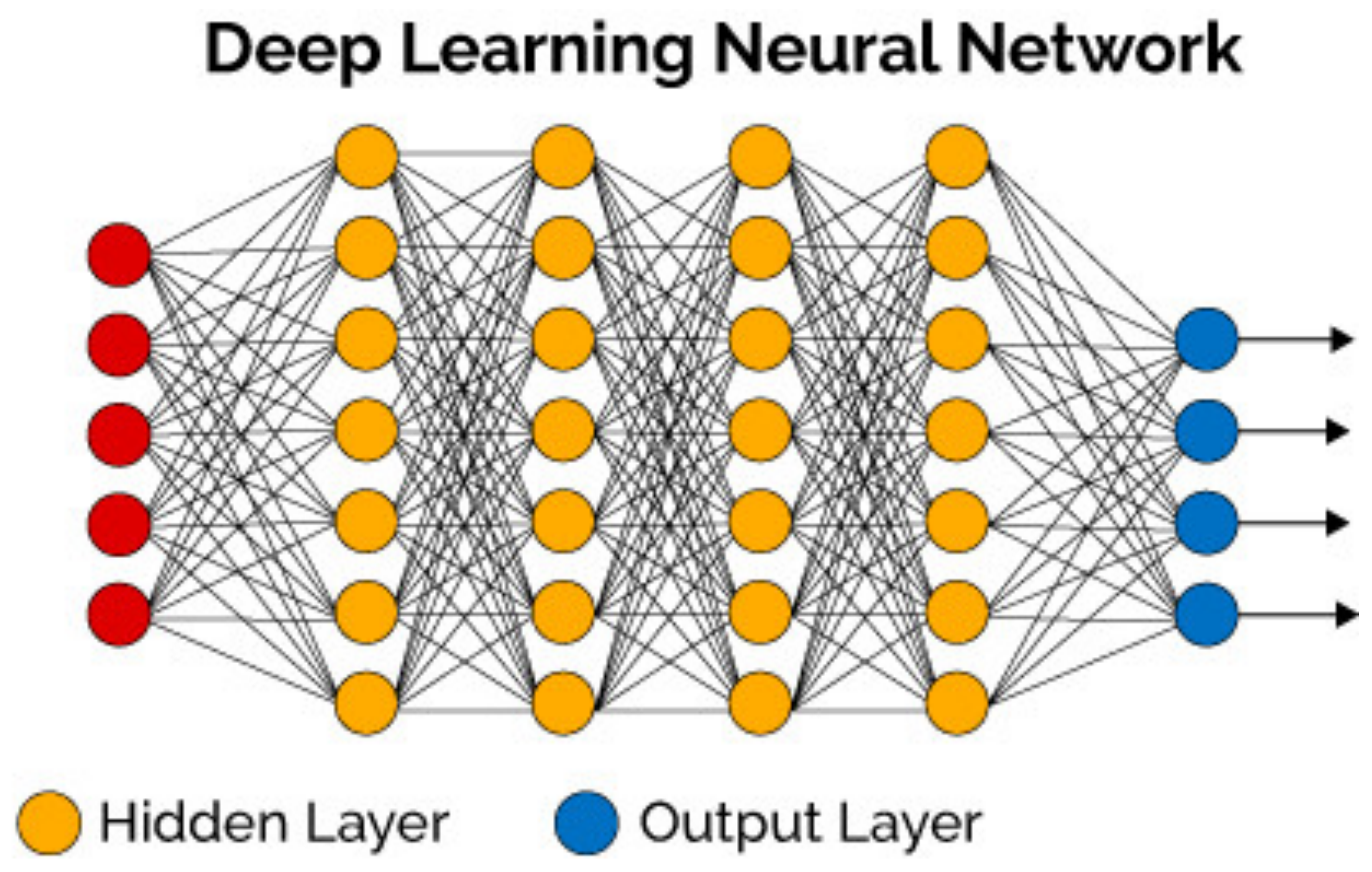
We have huge space space, i.e., $|S|$ might be $255^{3\times512\times512}$

We can only reset from initial state distribution $s_0 \sim \mu$

Numeration over state (e.g., a for loop) is not possible!

Goal: learn w/ function approximation

A Policy is a classifier w/ A many classes



What about continuous actions $a \in \mathbb{R}^d$?

Review on Policy Optimization:

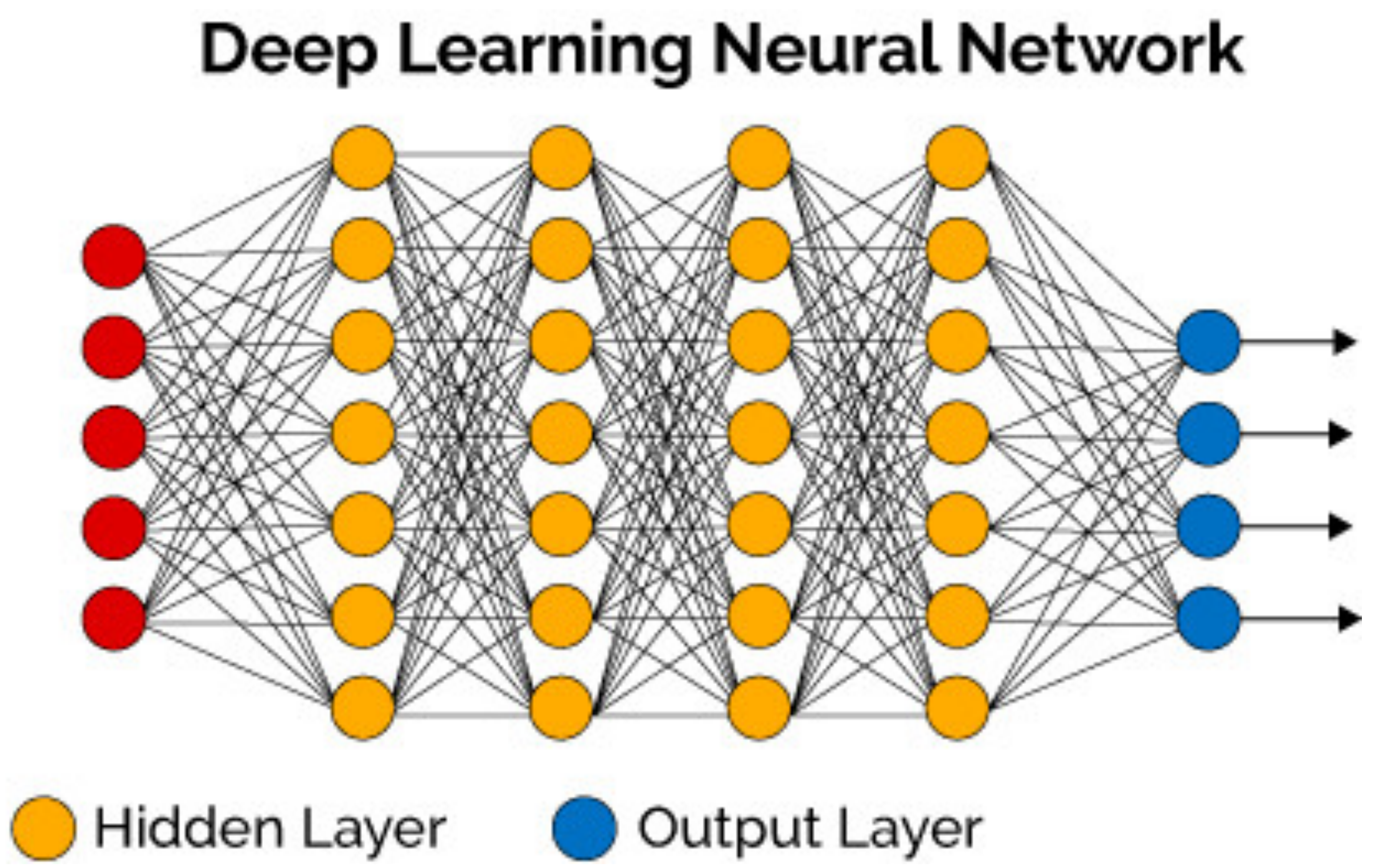
We have huge space space, i.e., $|S|$ might be $255^{3\times512\times512}$

We can only reset from initial state distribution $s_0 \sim \mu$

Numeration over state (e.g., a for loop) is not possible!

Goal: learn w/ function approximation

A Policy is a classifier w/ A many classes



What about continuous actions $a \in \mathbb{R}^d$?

$$\pi_{\beta,\alpha}(\cdot\,|\,s) = \mathcal{N}\left(\mu_\beta(s), \exp(\alpha) I_{d\times d}\right)$$

$$\theta := [\beta, \alpha]$$

Review on Policy Optimization:

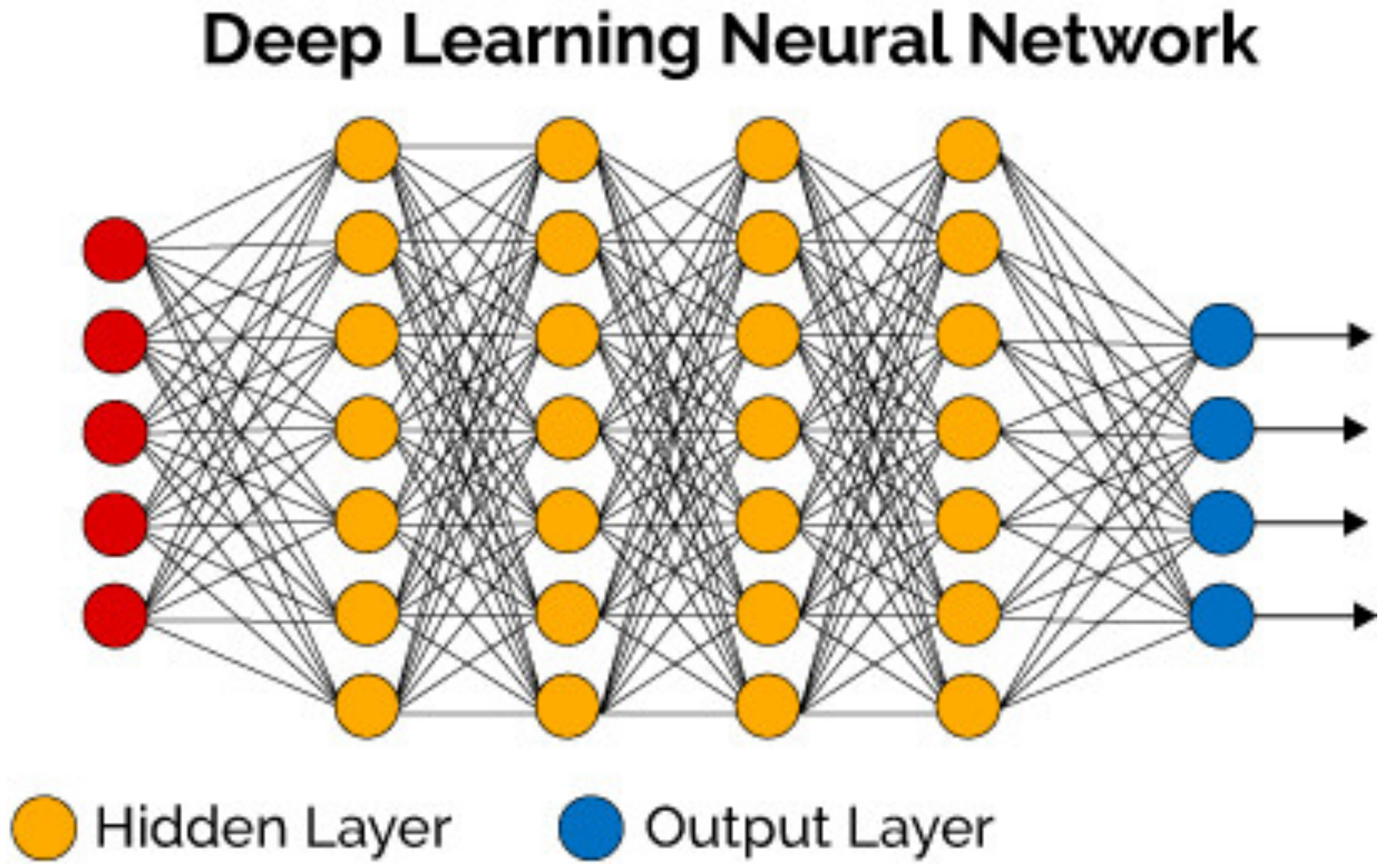
We have huge space space, i.e., $|S|$ might be $255^{3\times 512\times 512}$

We can only reset from initial state distribution $s_0 \sim \mu$

Numeration over state (e.g., a for loop) is not possible!

Goal: learn w/ function approximation

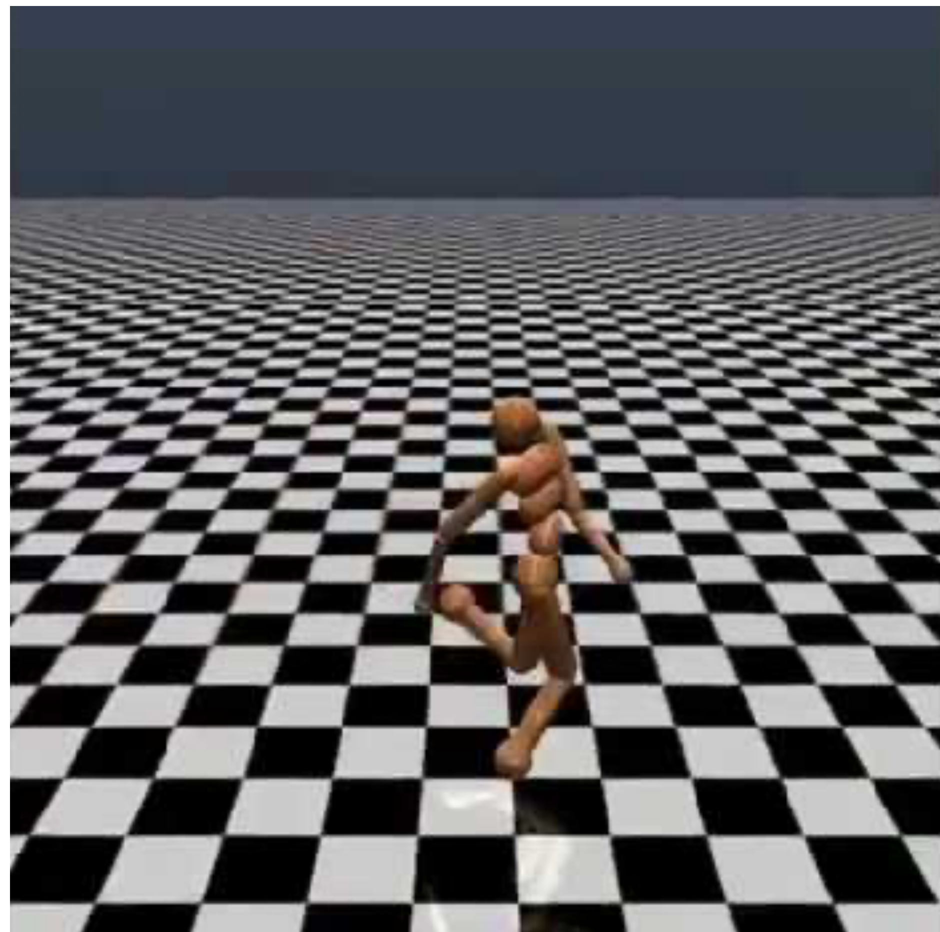
A Policy is a classifier w/ A many classes



What about continuous actions $a \in \mathbb{R}^d$?

$$\pi_{\beta,\alpha}(\cdot \mid s) = \mathcal{N}\left(\mu_\beta(s), \exp(\alpha) I_{d\times d}\right)$$

$$\theta := [\beta, \alpha]$$



Review on Policy Optimization: API

Given an current policy π^t, we perform policy update to π^{t+1}

First attempt: **Approximate Policy Iteration**

$$\pi^{t+1} = \arg\max_{\pi \in \Pi} \mathbb{E}_{s \sim d_\mu^{\pi_t}} \left[A^{\pi^t}(s, \pi(s)) \right]$$

Review on Policy Optimization: API

Given an current policy π^t, we perform policy update to π^{t+1}

First attempt: **Approximate Policy Iteration**

$$\pi^{t+1} = \arg\max_{\pi \in \Pi} \mathbb{E}_{s \sim d_\mu^{\pi_t}} \left[A^{\pi^t}(s, \pi(s)) \right]$$

i.e., find the greedy policy that maximizes the local advantage (e.g., via regression)

Review on Policy Optimization: API

Given an current policy π^t, we perform policy update to π^{t+1}

First attempt: **Approximate Policy Iteration**

$$\pi^{t+1} = \arg\max_{\pi\in\Pi} \mathbb{E}_{s\sim d_\mu^{\pi_t}} \left[A^{\pi^t}(s, \pi(s)) \right]$$

i.e., find the greedy policy that maximizes the local advantage (e.g., via regression)

Unfortunately, π^{t+1} might be very different from π^t,
and API could fail to make any progress

Review on Policy Optimization: CPI

Given an current policy π^t, we perform policy update to π^{t+1}

second attempt: **Conservative Policy Iteration**

$$\pi_{grd} = \arg\max_{\pi \in \Pi} \mathbb{E}_{s \sim d_\mu^{\pi_t}} \left[A^{\pi^t}(s, \pi(s)) \right]$$

Review on Policy Optimization: CPI

Given an current policy π^t, we perform policy update to π^{t+1}

second attempt: **Conservative Policy Iteration**

$$\pi_{grd} = \arg\max_{\pi \in \Pi} \mathbb{E}_{s \sim d_\mu^{\pi_t}} \left[A^{\pi^t}(s, \pi(s)) \right]$$

$$\forall s : \pi^{t+1}(\cdot \mid s) = (1 - \alpha)\pi^t(\cdot \mid s) + \alpha \pi_{grd}(\cdot \mid s)$$

Review on Policy Optimization: CPI

Given an current policy π^t, we perform policy update to π^{t+1}

second attempt: **Conservative Policy Iteration**

$$\pi_{grd} = \arg\max_{\pi \in \Pi} \mathbb{E}_{s \sim d_\mu^{\pi_t}} \left[A^{\pi^t}(s, \pi(s)) \right]$$

$$\forall s : \pi^{t+1}(\cdot \mid s) = (1 - \alpha)\pi^t(\cdot \mid s) + \alpha \pi_{grd}(\cdot \mid s)$$

i.e., CPI find the greedy policy, and move towards it a little bit!

Review on Policy Optimization: CPI

Given an current policy π^t, we perform policy update to π^{t+1}

second attempt: **Conservative Policy Iteration**

$$\pi_{grd} = \arg\max_{\pi\in\Pi} \mathbb{E}_{s\sim d_\mu^{\pi_t}} \left[A^{\pi^t}(s, \pi(s))\right]$$

$$\forall s : \pi^{t+1}(\cdot\,|\,s) = (1-\alpha)\pi^t(\cdot\,|\,s) + \alpha\pi_{grd}(\cdot\,|\,s)$$

i.e., CPI find the greedy policy, and move towards it a little bit!

Two nice properties:

Review on Policy Optimization: CPI

Given an current policy π^t, we perform policy update to π^{t+1}

second attempt: **Conservative Policy Iteration**

$$\pi_{grd} = \arg\max_{\pi \in \Pi} \mathbb{E}_{s \sim d_\mu^{\pi_t}} \left[A^{\pi^t}(s, \pi(s)) \right]$$

$$\forall s : \pi^{t+1}(\cdot \mid s) = (1 - \alpha)\pi^t(\cdot \mid s) + \alpha \pi_{grd}(\cdot \mid s)$$

i.e., CPI find the greedy policy, and move towards it a little bit!

Two nice properties:

$\left\| d^{\pi^t}(\cdot) - d^{\pi^{t+1}}(\cdot) \right\|_1 \leq O\left(\dfrac{\alpha}{1-\gamma}\right)$, $\quad V^{\pi^{t+1}} > V^{\pi^t}$ (if not terminate yet)

Review on Policy Optimization: PG

Given an current policy π^t, we perform policy update to π^{t+1}

Third attempt: **PG on parameterized policy**

$$\max_{\theta} \mathbb{E}_{s \sim d_{\mu}^{\pi_{\theta_t}}} \left[\mathbb{E}_{a \sim \pi_{\theta}(\cdot|s)} A^{\pi_{\theta_t}}(s, a) \right]$$

Review on Policy Optimization: PG

Given an current policy π^t, we perform policy update to π^{t+1}

Third attempt: **PG on parameterized policy**

$$\max_{\theta} \mathbb{E}_{s\sim d_{\mu}^{\pi_{\theta_t}}}\left[\mathbb{E}_{a\sim\pi_{\theta}(\cdot|s)} A^{\pi_{\theta_t}}(s,a)\right]$$

Locally Improve the local-adv a little bit via one-step gradient ascent:

Review on Policy Optimization: PG

Given an current policy π^t, we perform policy update to π^{t+1}

Third attempt: **PG on parameterized policy**

$$\max_{\theta} \mathbb{E}_{s \sim d_\mu^{\pi_{\theta_t}}} \left[\mathbb{E}_{a \sim \pi_\theta(\cdot | s)} A^{\pi_{\theta_t}}(s, a) \right]$$

Locally Improve the local-adv a little bit via one-step gradient ascent:

$$\theta_{t+1} = \theta_t + \eta \cdot \mathbb{E}_{s \sim d_\mu^{\pi_{\theta_t}}} \left[\mathbb{E}_{a \sim \pi_{\theta_t}(s)} \nabla \ln \pi_{\theta_t}(a \,|\, s) \cdot A^{\pi_{\theta_t}}(s, a) \right]$$

Review on Policy Optimization: PG

Given an current policy π^t, we perform policy update to π^{t+1}

Third attempt: **PG on parameterized policy**

$$\max_\theta \mathbb{E}_{s \sim d_\mu^{\pi_{\theta_t}}} \left[\mathbb{E}_{a \sim \pi_\theta(\cdot|s)} A^{\pi_{\theta_t}}(s, a) \right]$$

Locally Improve the local-adv a little bit via one-step gradient ascent:

$$\theta_{t+1} = \theta_t + \eta \cdot \mathbb{E}_{s \sim d_\mu^{\pi_{\theta_t}}} \left[\mathbb{E}_{a \sim \pi_{\theta_t}(s)} \nabla \ln \pi_{\theta_t}(a \mid s) \cdot A^{\pi_{\theta_t}}(s, a) \right]$$

When $\eta \to 0^+$, gradient ascent ensures
we improve the objective function

Review on Policy Optimization: NPG

Given an current policy π^t, we perform policy update to π^{t+1}

Fourth attempt: **Natural Policy Gradient**

$$\max_{\theta} \mathbb{E}_{s \sim d_{\mu}^{\pi_{\theta_t}}} \left[\mathbb{E}_{a \sim \pi_{\theta}(\cdot | s)} A^{\pi_{\theta_t}}(s, a) \right]$$

Review on Policy Optimization: NPG

Given an current policy π^t, we perform policy update to π^{t+1}

Fourth attempt: **Natural Policy Gradient**

$$\max_{\theta} \mathbb{E}_{s\sim d_\mu^{\pi_{\theta_t}}}\left[\mathbb{E}_{a\sim\pi_\theta(\cdot|s)} A^{\pi_{\theta_t}}(s,a)\right]$$

$$s.t., \mathrm{KL}(\rho_{\theta_t} | \rho_\theta) \leq \delta$$

Review on Policy Optimization: NPG

Given an current policy π^t, we perform policy update to π^{t+1}

Fourth attempt: **Natural Policy Gradient**

$$\max_{\theta} \mathbb{E}_{s\sim d_{\mu}^{\pi_{\theta_t}}}\left[\mathbb{E}_{a\sim\pi_\theta(\cdot|s)} A^{\pi_{\theta_t}}(s,a)\right]$$

$$s.t., \mathrm{KL}(\rho_{\theta_t} | \rho_\theta) \leq \delta$$

Define fisher info-matrix $F_{\theta_t} = \nabla_\theta^2 \mathrm{KL}(\rho_{\theta_t} | \rho_\theta)|_{\theta=\theta_t}$,

a convex approximation, e.g., linearize obj and quadratize constraint,

gives us the following NPG update:

Review on Policy Optimization: NPG

Given an current policy π^t, we perform policy update to π^{t+1}

Fourth attempt: **Natural Policy Gradient**

$$\max_\theta \mathbb{E}_{s\sim d_\mu^{\pi_{\theta_t}}} \left[\mathbb{E}_{a\sim\pi_\theta(\cdot|s)} A^{\pi_{\theta_t}}(s,a) \right]$$

$$s.t., \text{KL}(\rho_{\theta_t} | \rho_\theta) \leq \delta$$

Define fisher info-matrix $F_{\theta_t} = \nabla_\theta^2 \text{KL}(\rho_{\theta_t} | \rho_\theta)|_{\theta=\theta_t}$,
a convex approximation, e.g., linearize obj and quadratize constraint,
gives us the following NPG update:

$$\max_\theta \nabla_\theta J(\pi_{\theta_t})^\top (\theta - \theta_t), \text{ s.t., } (\theta - \theta_t)^\top F_{\theta_t}(\theta - \theta_t) \leq \delta$$

An extension of NPG (even faster in practice):

Given an current policy π^t, we perform policy update to π^{t+1}

fifth attempt (new): **Proximal Policy Optimization (PPO)**

$$\max_{\theta} \mathbb{E}_{s\sim d_\mu^{\pi_{\theta_t}}}\left[\mathbb{E}_{a\sim\pi_\theta(\cdot|s)} A^{\pi_{\theta_t}}(s,a)\right]$$

An extension of NPG (even faster in practice):

Given an current policy π^t, we perform policy update to π^{t+1}

fifth attempt (new): **Proximal Policy Optimization (PPO)**

$$\max_{\theta} \mathbb{E}_{s\sim d_\mu^{\pi_{\theta_t}}}\left[\mathbb{E}_{a\sim\pi_\theta(\cdot|s)} A^{\pi_{\theta_t}}(s,a)\right] \underbrace{-\lambda \mathbb{E}_{s\sim d_\mu^{\pi^t}}\left[\mathrm{KL}\left(\pi_{\theta_t}(a|s) | \pi_\theta(a|s)\right)\right]}_{\text{regularization}}$$

An extension of NPG (even faster in practice):

Given an current policy π^t, we perform policy update to π^{t+1}

fifth attempt (new): **Proximal Policy Optimization (PPO)**

$$\max_{\theta} \mathbb{E}_{s \sim d_\mu^{\pi_{\theta_t}}} \left[\mathbb{E}_{a \sim \pi_\theta(\cdot|s)} A^{\pi_{\theta_t}}(s, a) \right] \underbrace{- \lambda \mathbb{E}_{s \sim d_\mu^{\pi^t}} \left[\mathrm{KL} \left(\pi_{\theta_t}(a \mid s) \mid \pi_\theta(a \mid s) \right) \right]}_{\text{regularization}}$$

Use importance weighting & expand KL divergence:

An extension of NPG (even faster in practice):

Given an current policy π^t, we perform policy update to π^{t+1}

fifth attempt (new): **Proximal Policy Optimization (PPO)**

$$\max_{\theta} \mathbb{E}_{s\sim d_\mu^{\pi_{\theta_t}}}\left[\mathbb{E}_{a\sim\pi_\theta(\cdot|s)} A^{\pi_{\theta_t}}(s,a)\right] \underbrace{-\lambda \mathbb{E}_{s\sim d_\mu^{\pi^t}}\left[\mathrm{KL}\left(\pi_{\theta_t}(a\,|\,s)\,|\,\pi_\theta(a\,|\,s)\right)\right]}_{\text{regularization}}$$

Use importance weighting & expand KL divergence:

$$\ell(\theta) := \mathbb{E}_{s\sim d_\mu^{\pi_{\theta_t}}}\left[\mathbb{E}_{a\sim\pi_{\theta_t}(\cdot|s)} \frac{\pi_\theta(a\,|\,s)}{\pi_{\theta_t}(a\,|\,s)} A^{\pi_{\theta_t}}(s,a)\right] - \lambda \mathbb{E}_{s\sim d_\mu^{\pi_{\theta_t}}} \mathbb{E}_{a\sim\pi_{\theta_t}(\cdot|s)}\left[-\ln \pi_\theta(a\,|\,s)\right]$$

An extension of NPG (even faster in practice):

Given an current policy π^t, we perform policy update to π^{t+1}

fifth attempt (new): **Proximal Policy Optimization (PPO)**

$$\max_\theta \mathbb{E}_{s\sim d_\mu^{\pi_{\theta_t}}}\left[\mathbb{E}_{a\sim\pi_\theta(\cdot|s)} A^{\pi_{\theta_t}}(s,a)\right] \underbrace{-\lambda \mathbb{E}_{s\sim d_\mu^{\pi^t}}\left[\mathrm{KL}\left(\pi_{\theta_t}(a|s) | \pi_\theta(a|s)\right)\right]}_{\text{regularization}}$$

Use importance weighting & expand KL divergence:

$$\ell(\theta) := \mathbb{E}_{s\sim d_\mu^{\pi_{\theta_t}}}\left[\mathbb{E}_{a\sim\pi_{\theta_t}(\cdot|s)} \frac{\pi_\theta(a|s)}{\pi_{\theta_t}(a|s)} A^{\pi_{\theta_t}}(s,a)\right] - \lambda \mathbb{E}_{s\sim d_\mu^{\pi_{\theta_t}}} \mathbb{E}_{a\sim\pi_{\theta_t}(\cdot|s)}\left[-\ln \pi_\theta(a|s)\right]$$

PPO: Perform a few steps of mini-batch SGA on $\ell(\theta)$ to approximate $\arg\max_\theta \ell(\theta)$

Next a few lectures:

Imitation Learning
(Learning from Demonstrations)

Can we learn a good policy purely from expert demonstrations?