POLICY SEARCH METHODS 3

Herke van Hoof

STAYING CLOSE TO PREVIOUS POLICIES

- Small policy update steps tends to be more ‘safe’
- Estimated value function can be imprecise (approximation or estimation errors)
- So we don’t want to fully trust the current best guess!

STAYING CLOSE TO PREVIOUS POLICIES

- Small policy update steps tends to be more ‘safe’
- How to measure the difference between policies?
- Policy gradient answer: l2 distance parameter vectors
- One way to derive this:

$$\boldsymbol{\theta}^* - \boldsymbol{\theta}_0 = \max_{d\boldsymbol{\theta}} J(\boldsymbol{\theta}_0 + d\boldsymbol{\theta}) \quad \text{s.t. } d\boldsymbol{\theta}^T d\boldsymbol{\theta} = c$$

$$\approx \max_{d\theta} J(\theta_0) + (\nabla_\theta J(\theta_0))^T d\theta \quad \text{s.t. } d\theta^T d\theta = c$$

$$\propto \nabla_\theta J(\theta_0)$$

STAYING CLOSE TO PREVIOUS POLICIES

- Small policy update steps tends to be more ‘safe’
- How to measure the difference between policies?
- Policy gradient answer: l2 distance parameter vectors
- One way to derive this:

$$\begin{aligned}
\boldsymbol{\theta}^* - \boldsymbol{\theta}_0 &= \max_{d\boldsymbol{\theta}} J(\boldsymbol{\theta}_0 + d\boldsymbol{\theta}) && \text{s.t. } d\boldsymbol{\theta}^T d\boldsymbol{\theta} = c \\
&\approx \max_{d\boldsymbol{\theta}} J(\boldsymbol{\theta}_0) + (\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}_0))^T d\boldsymbol{\theta} && \text{s.t. } d\boldsymbol{\theta}^T d\boldsymbol{\theta} = c
\end{aligned}$$

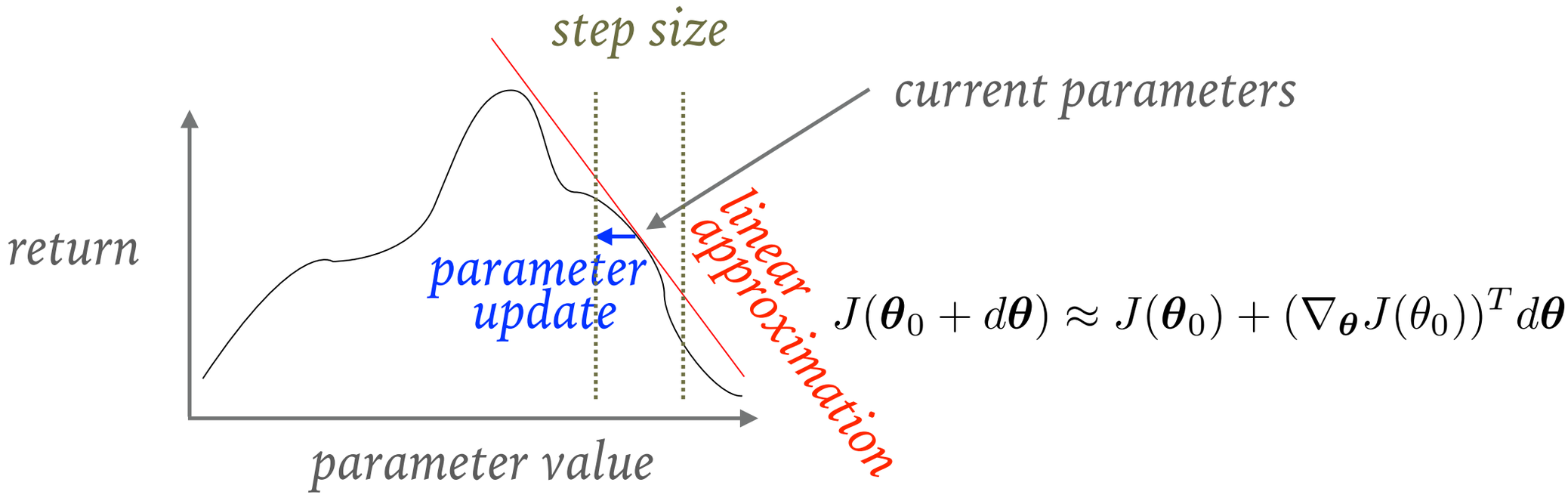
STAYING CLOSE TO PREVIOUS POLICIES

- Small policy update steps tends to be more ‘safe’
- How to measure the difference between policies?
- Policy gradient answer: l2 distance parameter vectors
- One way to derive this:

$$\begin{aligned}
\boldsymbol{\theta}^* - \boldsymbol{\theta}_0 &= \max_{d\boldsymbol{\theta}} J(\boldsymbol{\theta}_0 + d\boldsymbol{\theta}) && \text{s.t. } d\boldsymbol{\theta}^T d\boldsymbol{\theta} = c \\
&\approx \max_{d\boldsymbol{\theta}} J(\boldsymbol{\theta}_0) + (\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}_0))^T d\boldsymbol{\theta} && \text{s.t. } d\boldsymbol{\theta}^T d\boldsymbol{\theta} = c \\
&\propto \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}_0)
\end{aligned}$$

- Standard (‘vanilla’) policy gradients finds direction of most improvement per unit l2 distance in parameters

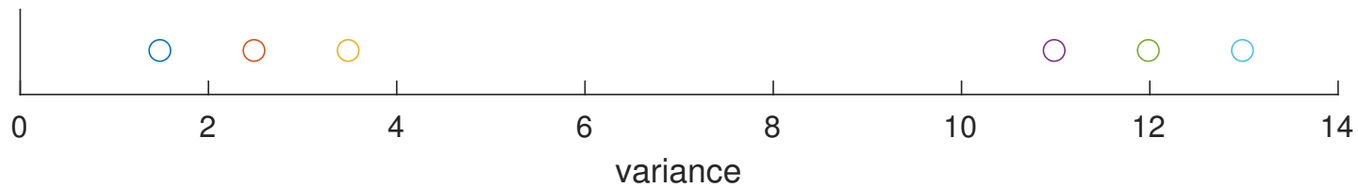
IN A PICTURE



$$J(\boldsymbol{\theta}_0 + d\boldsymbol{\theta}) \approx J(\boldsymbol{\theta}_0) + (\nabla_{\boldsymbol{\theta}} J(\theta_0))^T d\boldsymbol{\theta}$$

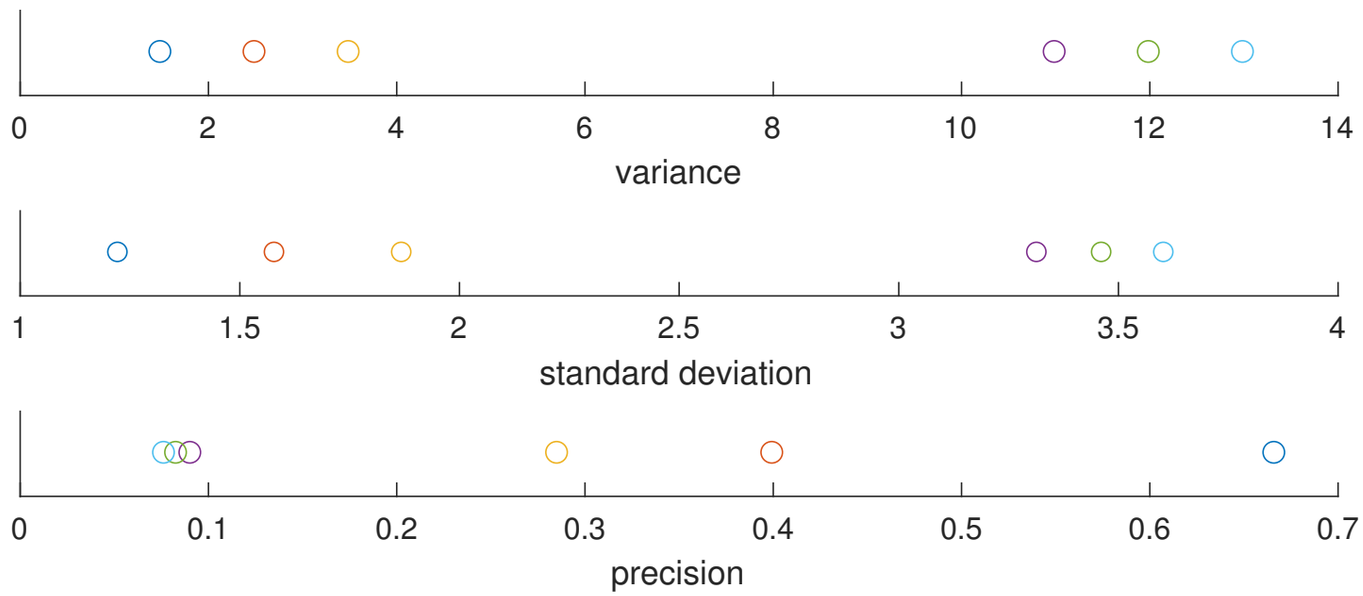
STAYING CLOSE TO PREVIOUS POLICIES

- Standard ('vanilla') policy gradients finds direction of most improvement per unit l2 distance in parameters
- This norm is sensitive to parametrisation:



STAYING CLOSE TO PREVIOUS POLICIES

- Standard ('vanilla') policy gradients finds direction of most improvement per unit l2 distance in parameters
- This norm is sensitive to parametrisation:



- How to express policy closeness covariantly?
- (covariant: independent of choice of parametrisation)

STAYING CLOSE TO PREVIOUS POLICIES

- Why do we want covariant norm (invariant to parametrisation)?
  - Don't waste time tuning parametrisation /hy perparameters
  - Parameters with different 'meaning': mean and precision
    - does a norm in this space make sense?
      - step size never right on all parameters if scale different
        (have to take step small enough for most sensitive direction)
- Correlations between parameters ignored
  (feature modulated by more parameters easier to change)

STAYING CLOSE TO PREVIOUS POLICIES

- Why do we want covariant norm (invariant to parametrisation)?
  - Don't waste time tuning parametrisation
  - Parameters with different 'meaning': mean and precision
    - does a norm in this space make sense?
      - step size never right on all parameters if scale different (have to take step small enough for most sensitive direction)
  - Correlations between parameters ignored (feature modulated by more parameters easier to change)
- Conceptually, it's not the change in parameters we care about!
  - Limit change in trajectories, states, and/or actions?

STAYING CLOSE TO PREVIOUS POLICIES

- How to express policy closeness covariantly?
- Kullback-Leibler (KL) divergence is information-theoretic quantification of difference between distributions

$$D_{\mathrm{KL}}(P||Q) = \int_{-\infty}^{\infty} p(x) \log \frac{p(x)}{q(x)} \mathrm{d}x$$

- Minimal value of 0 when p=q
- KL is invariant under parameter transformations
- Idea: find direction of maximal improvement per unit of KL between policies

STAYING CLOSE TO PREVIOUS POLICIES

- Idea: find direction of maximal improvement per unit of KL between policies
- Several algorithms can be understood using this idea
  - Natural policy gradient
  - Trust region policy optimization (TRPO)

NATURAL POLICY GRADIENT

- Idea: make policy gradients covariant [Kakade 2002]
- this yields an algorithm that exploits structure of parameters
- Here, will look how it relates to KL [Bagnell 2003]

[Kakade 2002, Bagnell 2003]

NATURAL POLICY GRADIENT

- Recall vanilla policy gradients

$$\begin{aligned}\boldsymbol{\theta}^* - \boldsymbol{\theta}_0 &= \max_{d\boldsymbol{\theta}} J(\boldsymbol{\theta}_0 + d\boldsymbol{\theta}) \\ &\approx \max_{d\boldsymbol{\theta}} J(\boldsymbol{\theta}_0) + (\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}_0))^T d\boldsymbol{\theta} \\ &\propto \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}_0)\end{aligned}$$

$$\text{s.t. } d\boldsymbol{\theta}^T d\boldsymbol{\theta} = c$$

$$\text{s.t. } d\boldsymbol{\theta}^T d\boldsymbol{\theta} = c$$

- replace constraint by quadratic expansion of KL divergence

$$c = \mathbb{E}_{\mathbf{s}} \left[D_{\mathrm{KL}}(\pi(\mathbf{a}|\mathbf{s}; \boldsymbol{\theta}_0) \| \pi(\mathbf{a}|\mathbf{s}; \boldsymbol{\theta}_0 + d\boldsymbol{\theta}) \right] = \mathrm{EKL}(d\boldsymbol{\theta})$$

$$\approx \mathrm{EKL}(\boldsymbol{\theta}_0) + d\boldsymbol{\theta}^T \nabla_{d\boldsymbol{\theta}} \mathrm{EKL}(d\boldsymbol{\theta}) + \frac{1}{2} d\boldsymbol{\theta}^T \left(\nabla^2_{d\theta} \mathrm{EKL}(d\boldsymbol{\theta}) \right) d\boldsymbol{\theta}$$

0

ge

(gradients and Hessians evaluated at $d\boldsymbol{\theta} = 0$)

NATURAL POLICY GRADIENT

- Recall vanilla policy gradients

$$\begin{aligned}
\boldsymbol{\theta}^* - \boldsymbol{\theta}_0 &= \max_{d\boldsymbol{\theta}} J(\boldsymbol{\theta}_0 + d\boldsymbol{\theta}) && \text{s.t. } d\boldsymbol{\theta}^T d\boldsymbol{\theta} = c \\
&\approx \max_{d\boldsymbol{\theta}} J(\boldsymbol{\theta}_0) + (\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}_0))^T d\boldsymbol{\theta} && \text{s.t. } d\boldsymbol{\theta}^T d\boldsymbol{\theta} = c \\
&\propto \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}_0)
\end{aligned}$$

- replace constraint by quadratic expansion of KL divergence

$$\begin{aligned}
c &= \mathbb{E}_{\mathbf{s}} \left[D_{\mathrm{KL}}(\pi(\mathbf{a}|\mathbf{s};\boldsymbol{\theta}_0) \| \pi(\mathbf{a}|\mathbf{s};\boldsymbol{\theta}_0 + d\boldsymbol{\theta}) \right] = \mathrm{EKL}(d\boldsymbol{\theta}) \\
&\approx \mathrm{EKL}\left(\boldsymbol{\theta}_0\right) + d\boldsymbol{\theta}^T \nabla_{d\boldsymbol{\theta}} \mathrm{EKL}\left(d\boldsymbol{\theta}\right) + \frac{1}{2} d\boldsymbol{\theta}^T \left(\nabla^2_{d\theta} \mathrm{EKL}(d\boldsymbol{\theta}) \right) d\boldsymbol{\theta} \\
c &= 0 + \frac{1}{2} d\boldsymbol{\theta}^T \left(\nabla^2_{d\theta} \mathrm{EKL}(d\boldsymbol{\theta}) \right) d\boldsymbol{\theta}
\end{aligned}$$

- since minimal value of 0 is reached if parameter doesn't change

NATURAL POLICY GRADIENT

$$c = \mathbb{E}_{\mathbf{s}}\left[D_{\mathrm{KL}}(\pi(\mathbf{a}|\mathbf{s};\boldsymbol{\theta}_0)\|\pi(\mathbf{a}|\mathbf{s};\boldsymbol{\theta}_0 + d\boldsymbol{\theta})\right] = \mathrm{EKL}(d\boldsymbol{\theta})$$

$$\approx \frac{1}{2} d\boldsymbol{\theta}^T \left(\nabla^2_{d\theta}\mathrm{EKL}(d\boldsymbol{\theta})\right) d\boldsymbol{\theta}$$

- This is the squared norm with respect to matrix

$$F = \nabla^2_{d\boldsymbol{\theta}}\mathrm{EKL} = \mathbb{E}_s\left[\nabla^2_{d\theta} D_{\mathrm{KL}}\left(\pi\left(\mathbf{a}|\mathbf{s};\boldsymbol{\theta}_0\right)\|\pi(\mathbf{a}|\mathbf{s};\boldsymbol{\theta}_0 + d\boldsymbol{\theta})\right)\right]$$

FIM

- F can be shown to be the expected Fisher information matrix of the policy
- F characterises information about parameters in action

[Kakade 2002, Bagnell 2003]

NATURAL POLICY GRADIENT

- Consider now the modified optimisation problem

$$\boldsymbol{\theta}^* - \boldsymbol{\theta}_0 = \max_{d\boldsymbol{\theta}} J(\boldsymbol{\theta}_0 + d\boldsymbol{\theta}) \qquad \text{s.t. } d\boldsymbol{\theta}^T F d\boldsymbol{\theta} = c$$

$$\approx \max_{d\boldsymbol{\theta}} J(\boldsymbol{\theta}_0) + (\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}_0))^T d\boldsymbol{\theta} \qquad \text{s.t. } d\boldsymbol{\theta}^T F d\boldsymbol{\theta} = c$$

- solve constraint optimisation problem: Lagrangian

$$L(d\theta, \lambda) = J(\theta_0) + (\nabla_\theta J(\theta_0))^T d\theta + \lambda (d\theta^T F d\theta - c)$$

- At optimality, partial derivatives of L are 0

$$\frac{\partial L(d\theta, \lambda)}{\partial d\theta} = 0 \Rightarrow (\nabla_\theta J(\theta_0)) + \lambda \frac{1}{2} F d\theta$$

$$\frac{\partial L(d\theta, \lambda)}{\partial \lambda} = 0 \Rightarrow d\theta^T F d\theta - c$$

[Kakade 2002, Bagnell 2003]

NATURAL POLICY GRADIENT

- Consider now the modified optimisation problem

$$\boldsymbol{\theta}^* - \boldsymbol{\theta}_0 = \max_{d\boldsymbol{\theta}} J(\boldsymbol{\theta}_0 + d\boldsymbol{\theta}) \qquad \text{s.t. } d\boldsymbol{\theta}^T F d\boldsymbol{\theta} = c$$

$$\approx \max_{d\boldsymbol{\theta}} J(\boldsymbol{\theta}_0) + (\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}_0))^T d\boldsymbol{\theta} \qquad \text{s.t. } d\boldsymbol{\theta}^T F d\boldsymbol{\theta} = c$$

- solve constraint optimisation problem: Lagrangian

$$L(d\boldsymbol{\theta}, \lambda) = J(\boldsymbol{\theta}_0) + \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}_0)^T d\boldsymbol{\theta} + \lambda(d\boldsymbol{\theta}^T \mathbf{F} d\boldsymbol{\theta} - c)$$

- At optimality, partial derivatives of L are 0

$$\frac{\partial L(d\boldsymbol{\theta}, \lambda)}{\partial d\boldsymbol{\theta}} = 0$$

$$\frac{\partial L(d\boldsymbol{\theta}, \lambda)}{\partial \lambda} = 0$$

$$(\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}_0)) + \lambda F d\boldsymbol{\theta} = 0$$

$$d\boldsymbol{\theta}^T F d\boldsymbol{\theta} = c$$

[Kakade 2002, Bagnell 2003]

NATURAL POLICY GRADIENT

- So optimality conditions are

$$\begin{aligned}(\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta})) + \lambda F d\boldsymbol{\theta} &= 0 \\ d\boldsymbol{\theta}^T F d\boldsymbol{\theta} &= c\end{aligned}$$

- From the first line, update direction

$$d\boldsymbol{\theta} \propto F^{-1} \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$$

- This is the **natural gradient**
  (natural gradients in ML used at least since [Amari, 1998], used in RL since [Kakade 2002])

NATURAL POLICY GRADIENT

- The policy is adapted using the **natural gradient** update rule:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \alpha F^{-1} \nabla_{\boldsymbol{\theta}_t} J(\boldsymbol{\theta}_t)$$

- We can use any known approach for the vanilla gradient
- Will this always improve J?
- For small enough step size, objective improves if

$$(\theta^x - \theta_0) \nabla J(\theta_0) > 0$$

$$\nabla_\theta J(\theta_0)^T F^{-1} \nabla J(\theta_0) \gtrless 0$$

NATURAL POLICY GRADIENT

- The policy is adapted using the **natural gradient** update rule:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \alpha F^{-1} \nabla_{\boldsymbol{\theta}_t} J(\boldsymbol{\theta}_t)$$

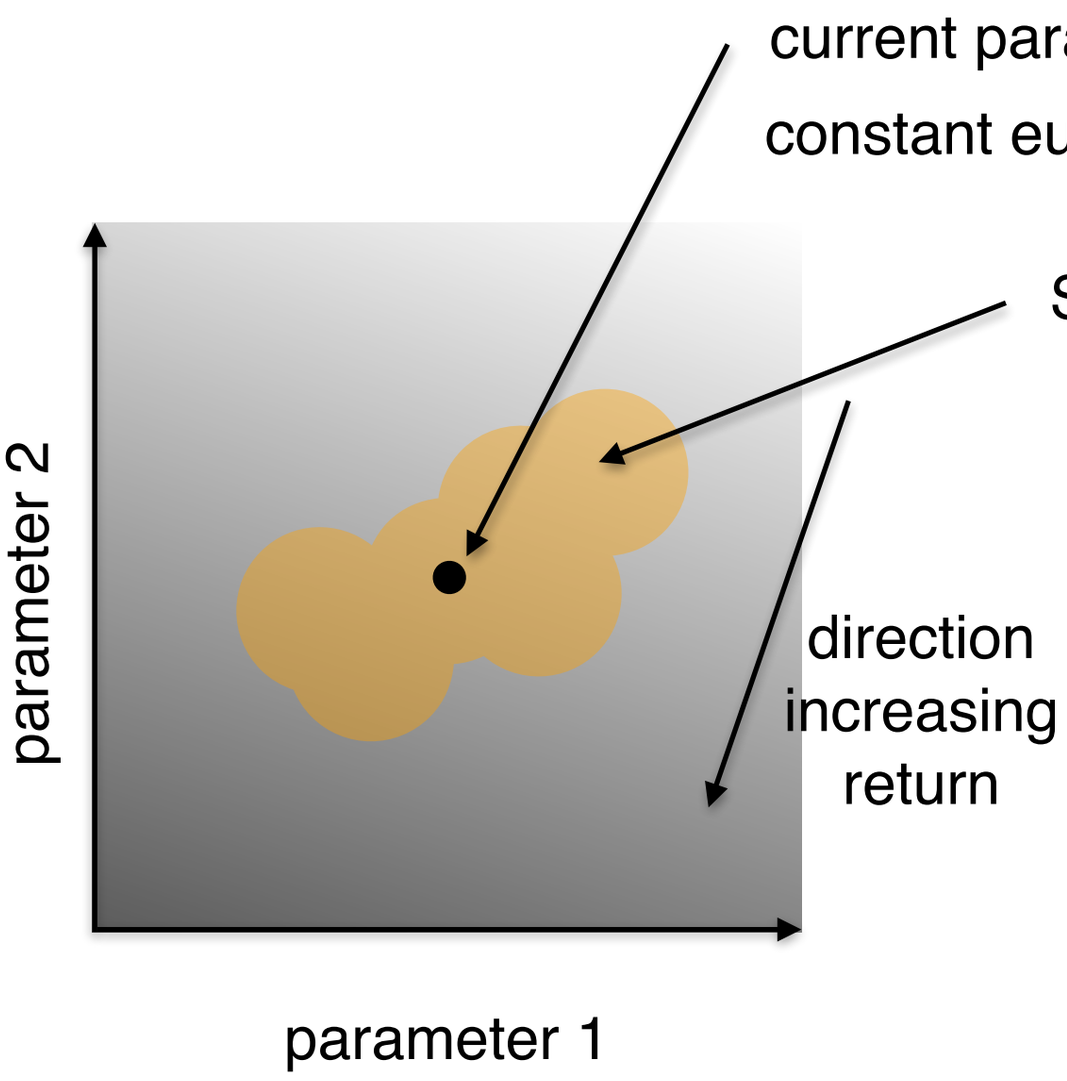
- We can use any known approach for the vanilla gradient
- Will this always improve J?
- For small enough step size, objective improves if

$$(\boldsymbol{\theta}^* - \boldsymbol{\theta}_0)^T \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}_0) > 0$$

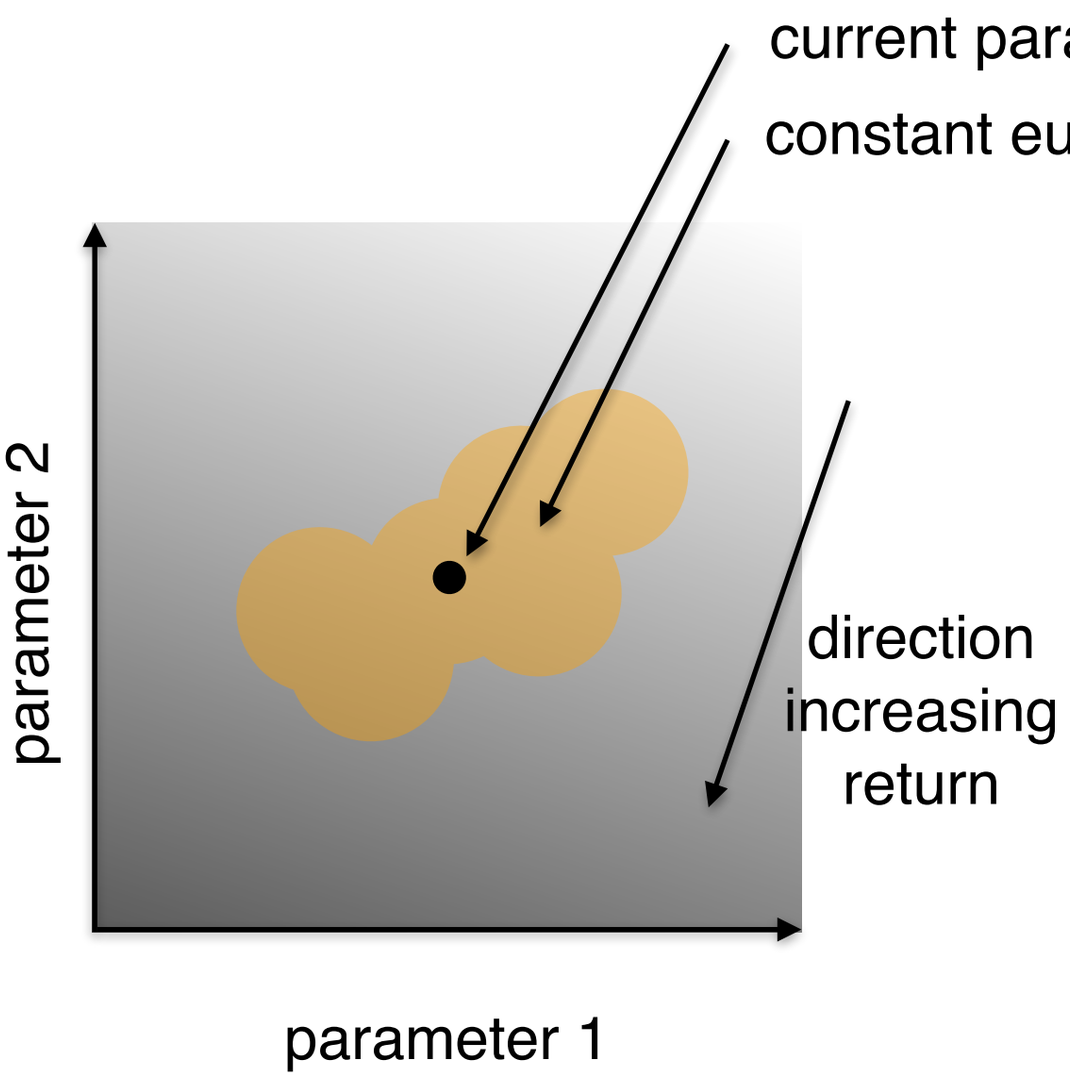
$$(\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}_0))^T F^{-1} \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}_0) \overset{?}{>} 0$$

- Since Fisher information is positive definite, answer is **yes**

NATURAL POLICY GRADIENT: SOME INTUITION

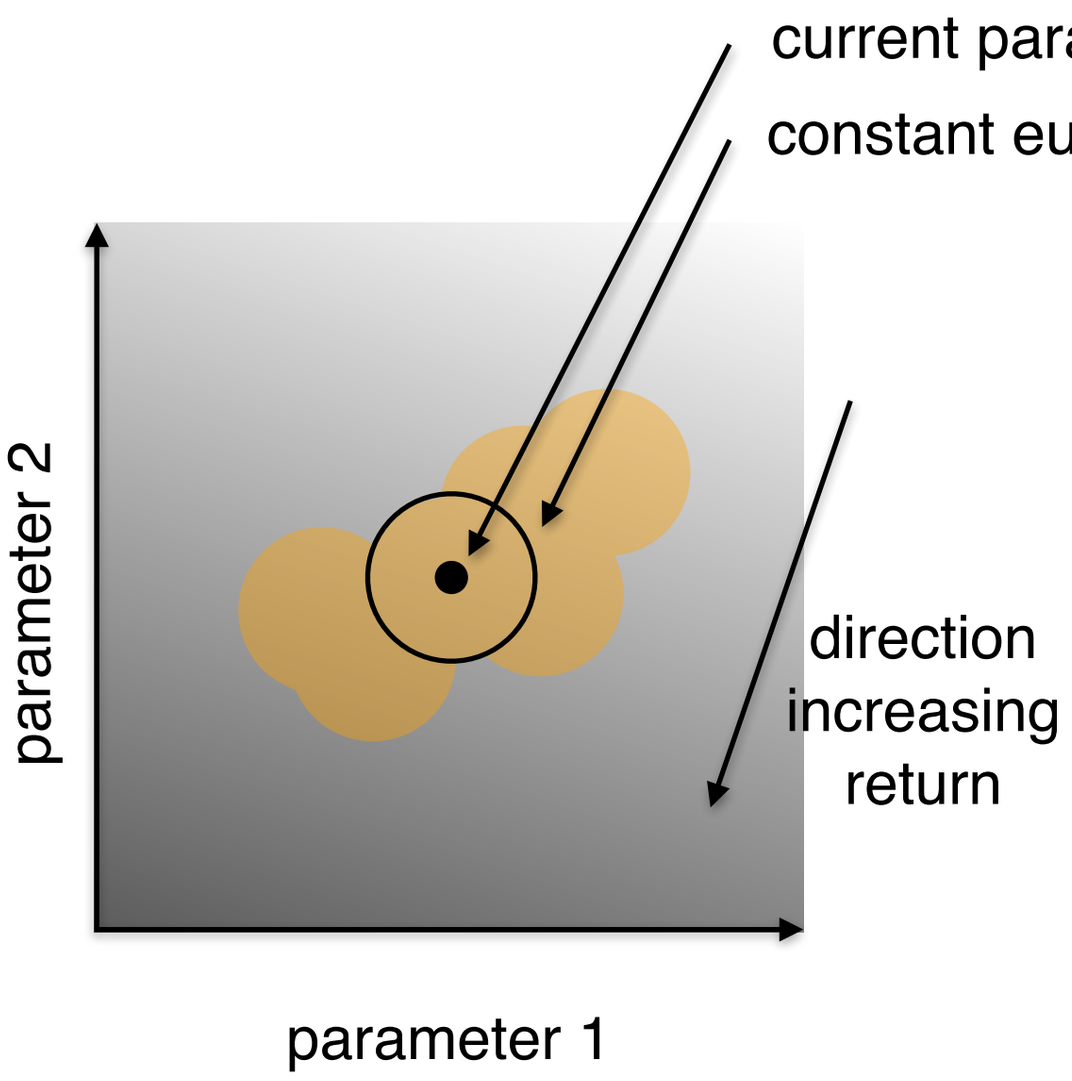


NATURAL POLICY GRADIENT: SOME INTUITION



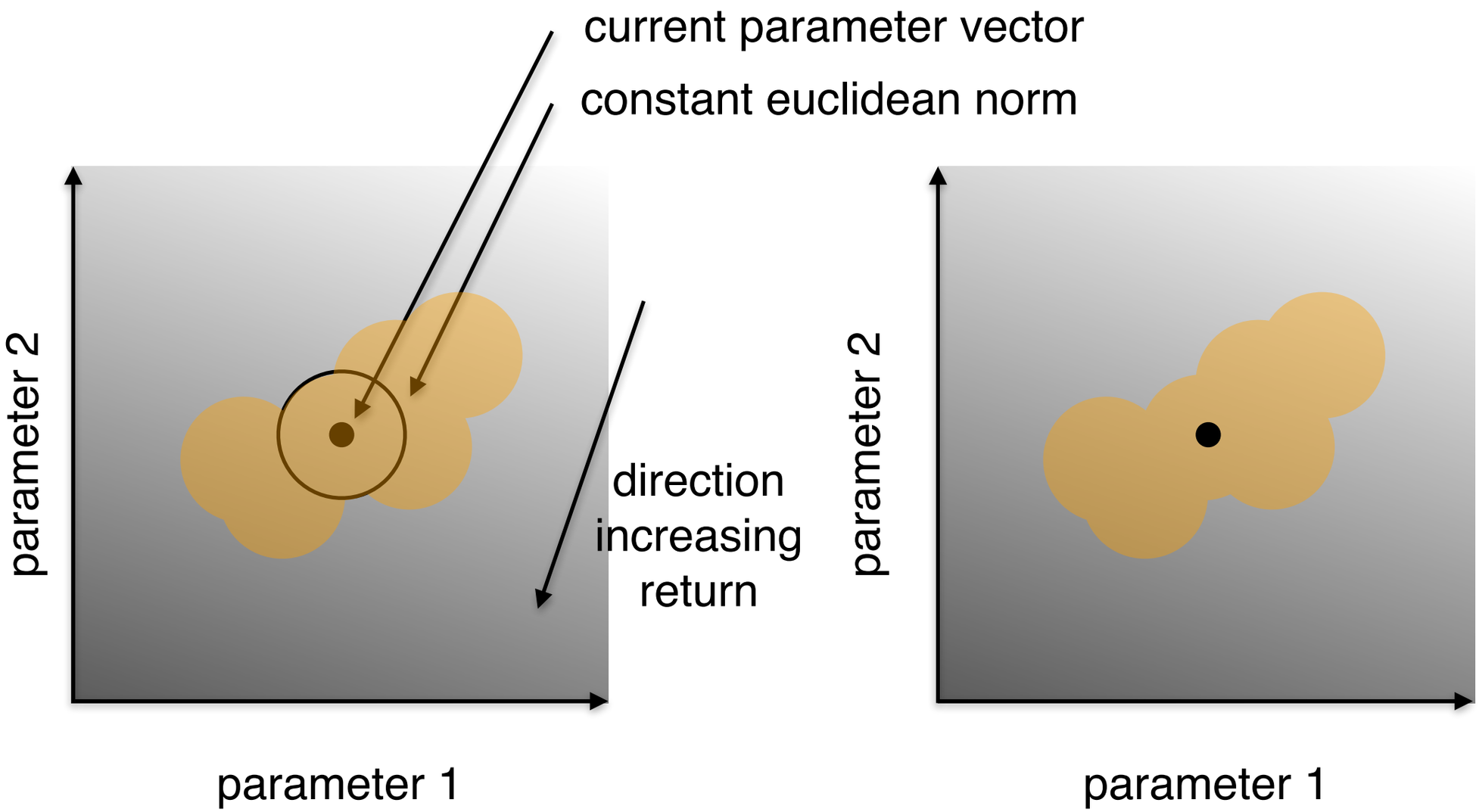
Regular gradient: increase step size till we notice we are outside of safe region

NATURAL POLICY GRADIENT: SOME INTUITION



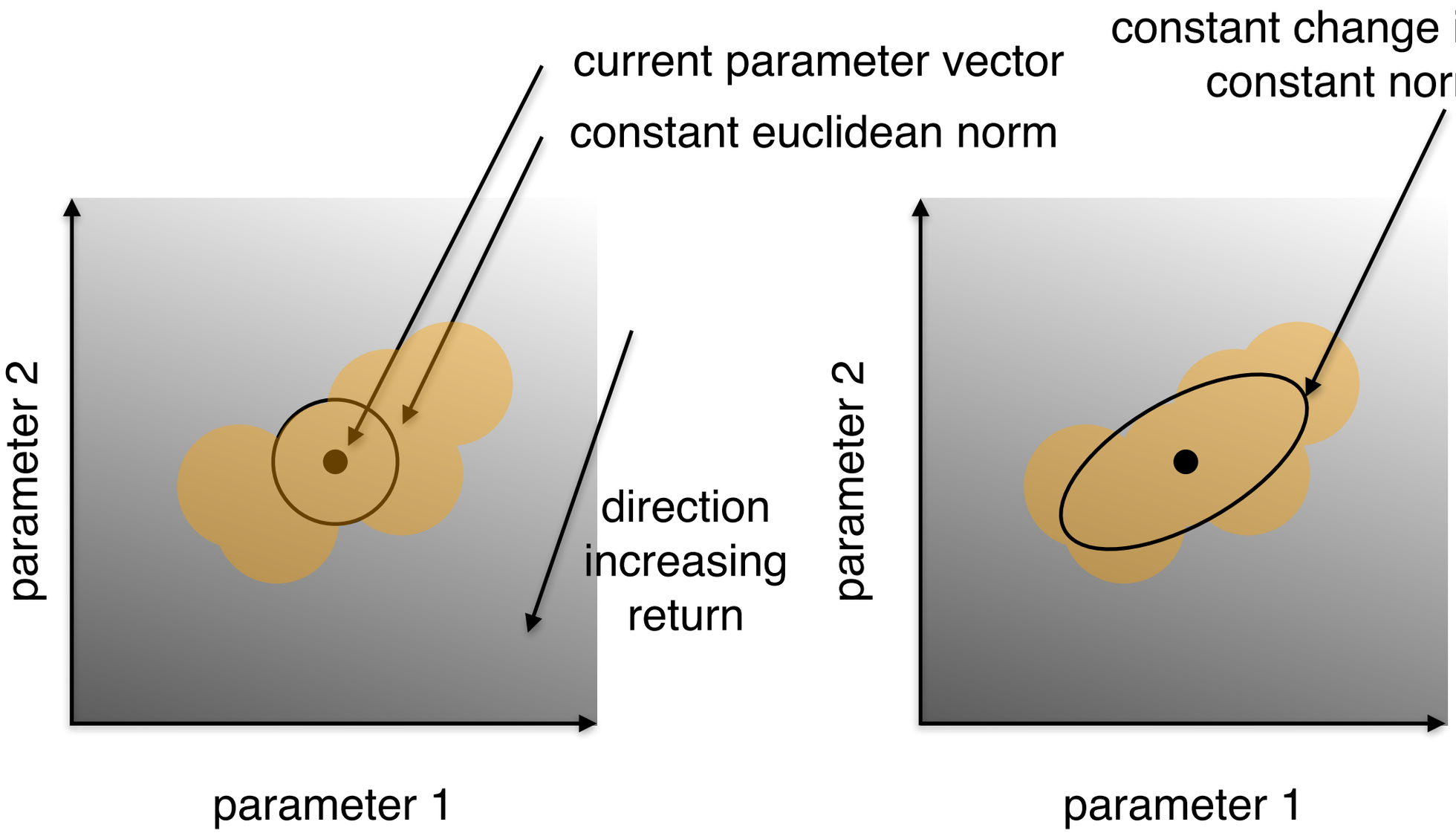
Regular gradient: increase step size till we notice we are outside of safe region

NATURAL POLICY GRADIENT: SOME INTUITION



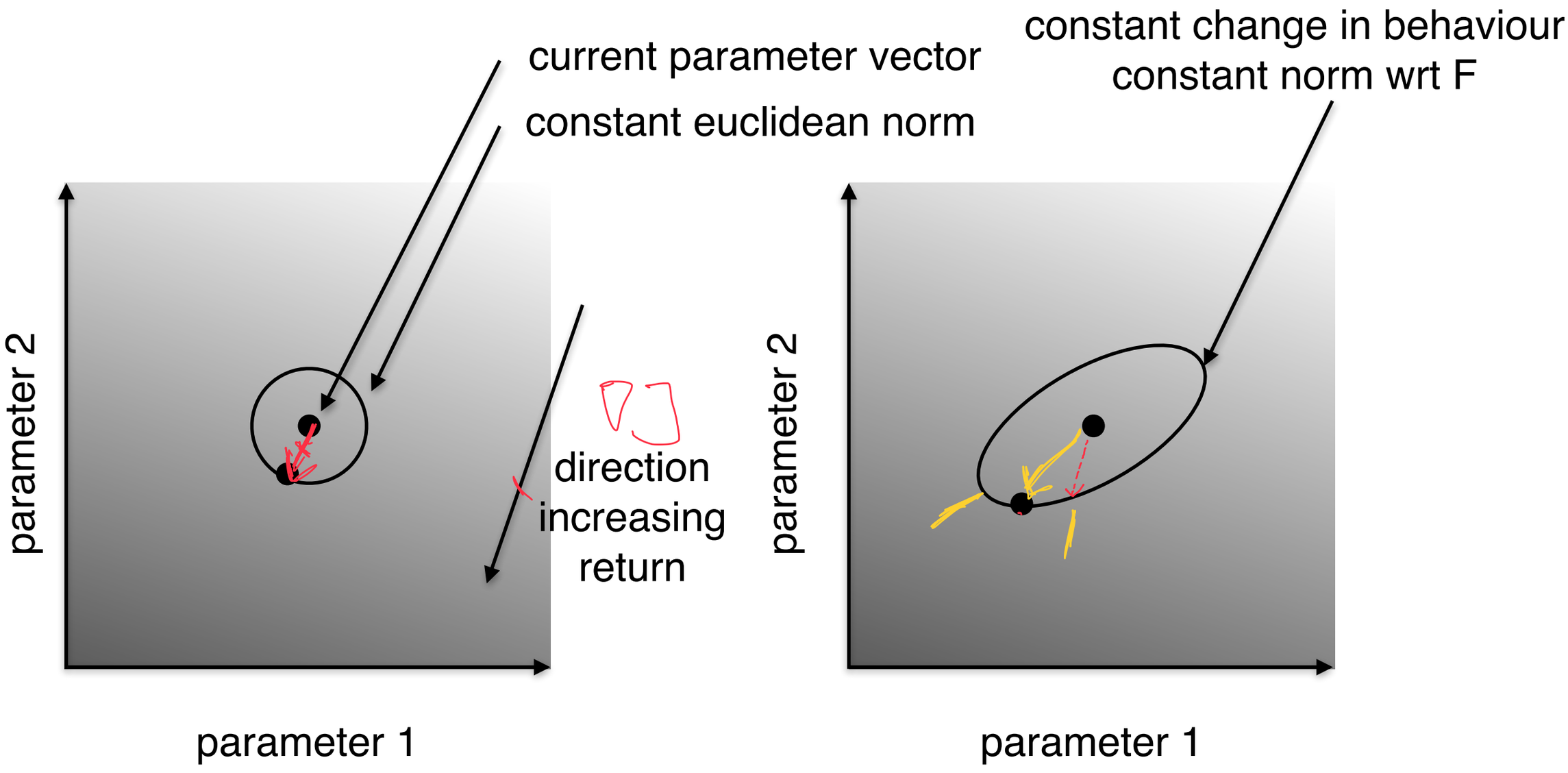
Do the same for natural gradient…

NATURAL POLICY GRADIENT: SOME INTUITION



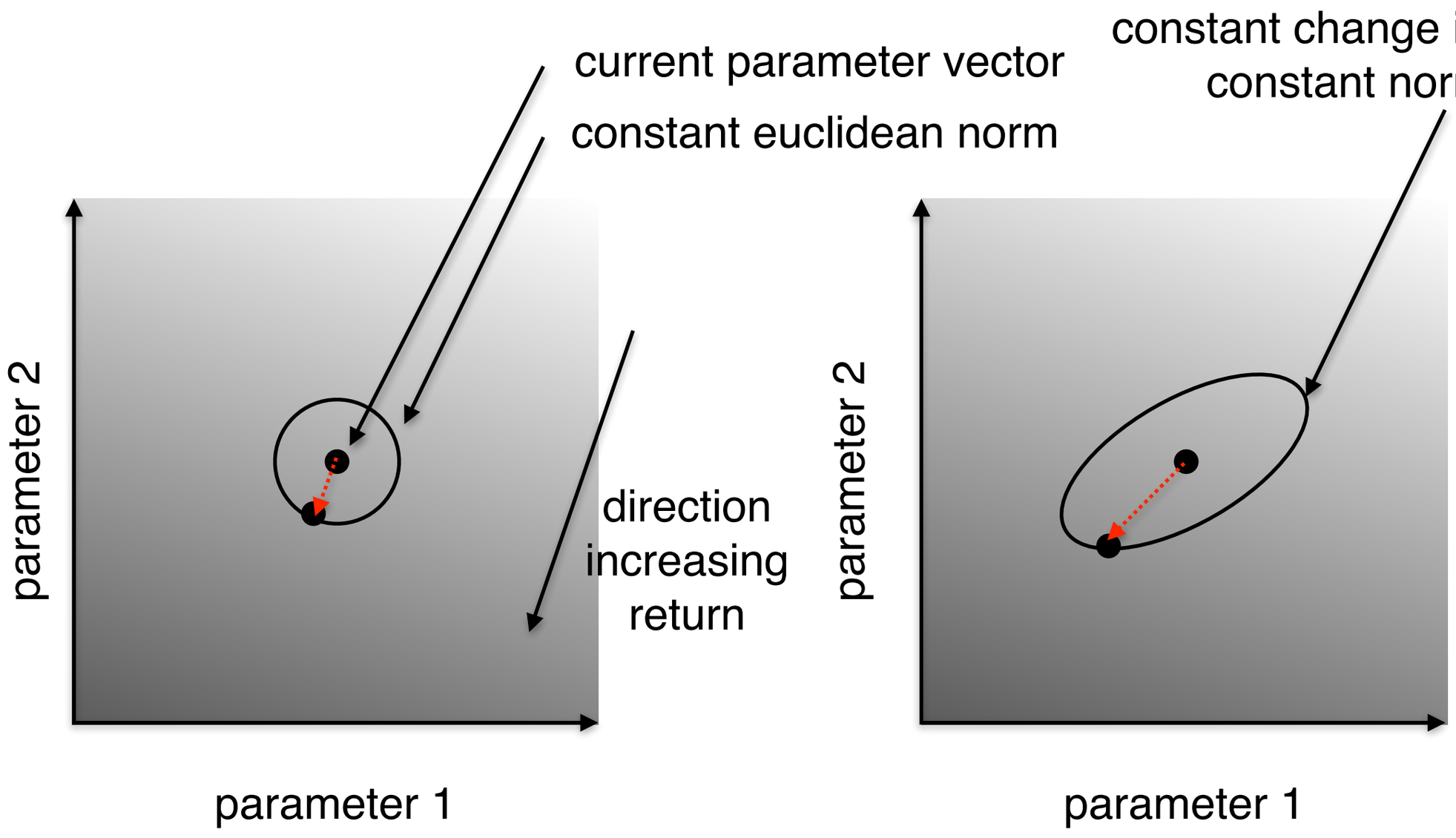
Do the same for natural gradient…

NATURAL POLICY GRADIENT: SOME INTUITION



As natural gradient can (hopefully!) better fit safe region, bigger steps possible

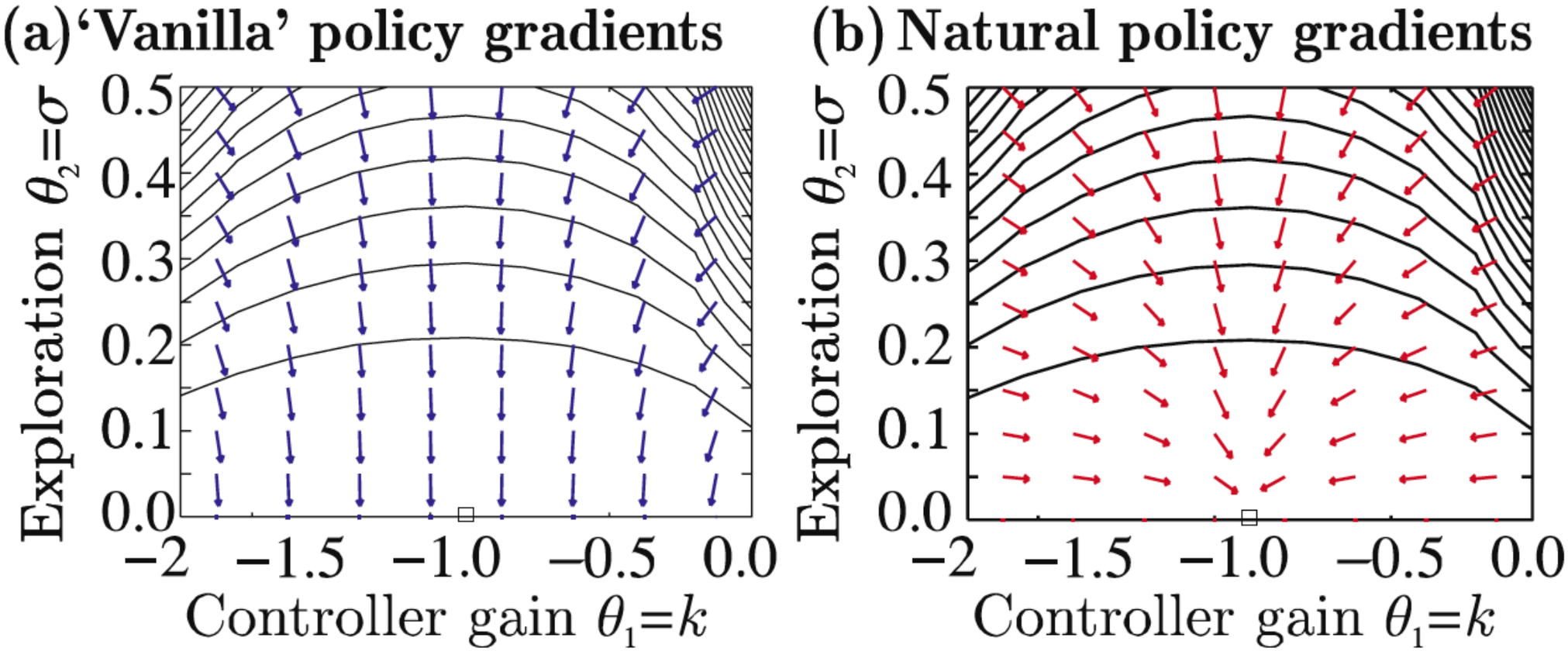
NATURAL POLICY GRADIENT: SOME INTUITION



Regular gradient: direction of steepest increase expected return
Natural gradient: within 90° of that direction: will improve objective
‘steepest return per unit policy change’

NATURAL ACTOR CRITIC

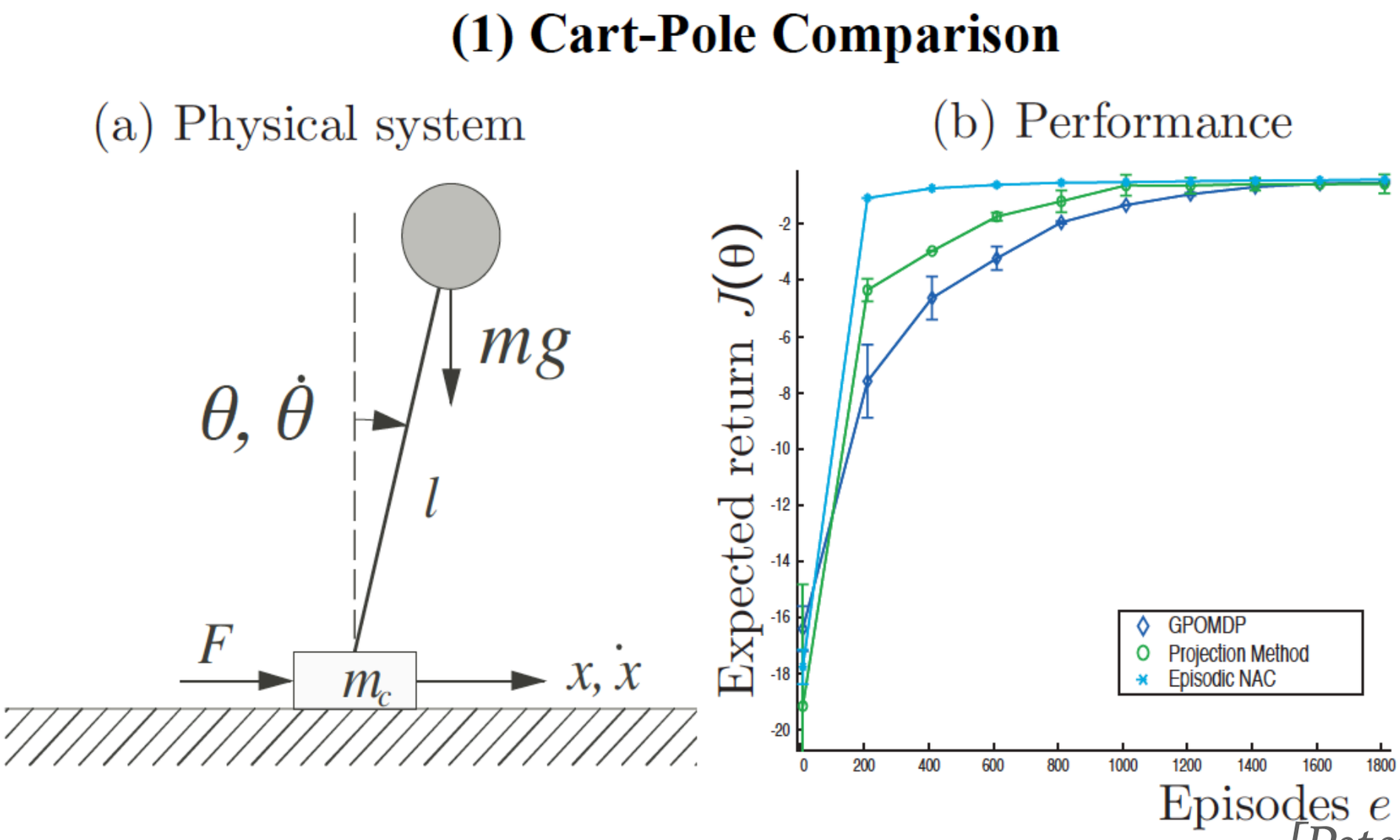
- Natural gradients can help where the likelihood is almost flat



[Peters 2008]

NATURAL POLICY GRADIENT

- Natural policy gradients can be used in actor-critic set-up



[Peters 2008]

NATURAL POLICY GRADIENT

- Advantages
  - Usually needs less training than regular policy gradients
  - Inherits advantageous properties from vanilla gradients
- Limitations
  - Need Fisher information matrix
    - Known for some standard distributions, e.g. Gaussian
  - Inherits disadvantages from PG (e.g., high variance)

STAYING CLOSE TO PREVIOUS POLICIES

- Idea: use KL to specify how the policy can change in one step
  - Natural policy gradient
  - Trust region policy optimization (TRPO)

TRUST REGION POLICY OPTIMISATION

- Trust region: region where approximation is valid
- Schulman's "Trust region policy optimisation" defines a trust region based RL algorithm inspired by a theoretical lower bound
- Theoretical starting point quite different than policy gradients, but practical implementation is quite similar

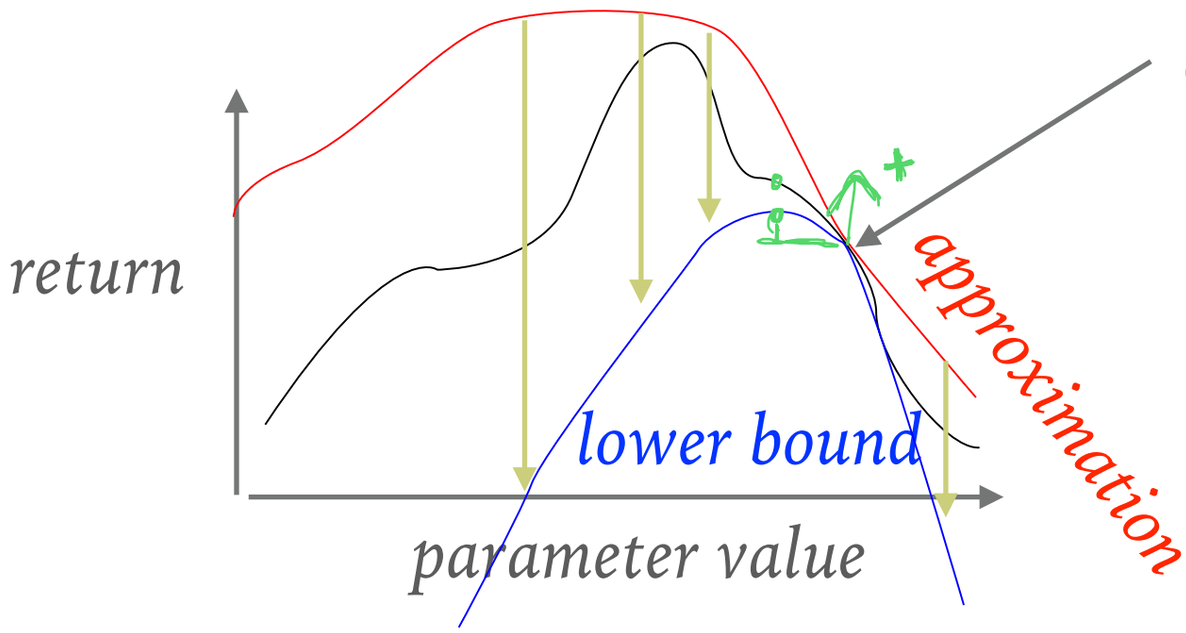
TRPO: THEORY

- Idea: take big steps while guaranteeing improvement

  1. approximate the return function
     (ignoring the difference between μ_b and μ_π)

  2. apply KL-based penalty term to yield lower bound

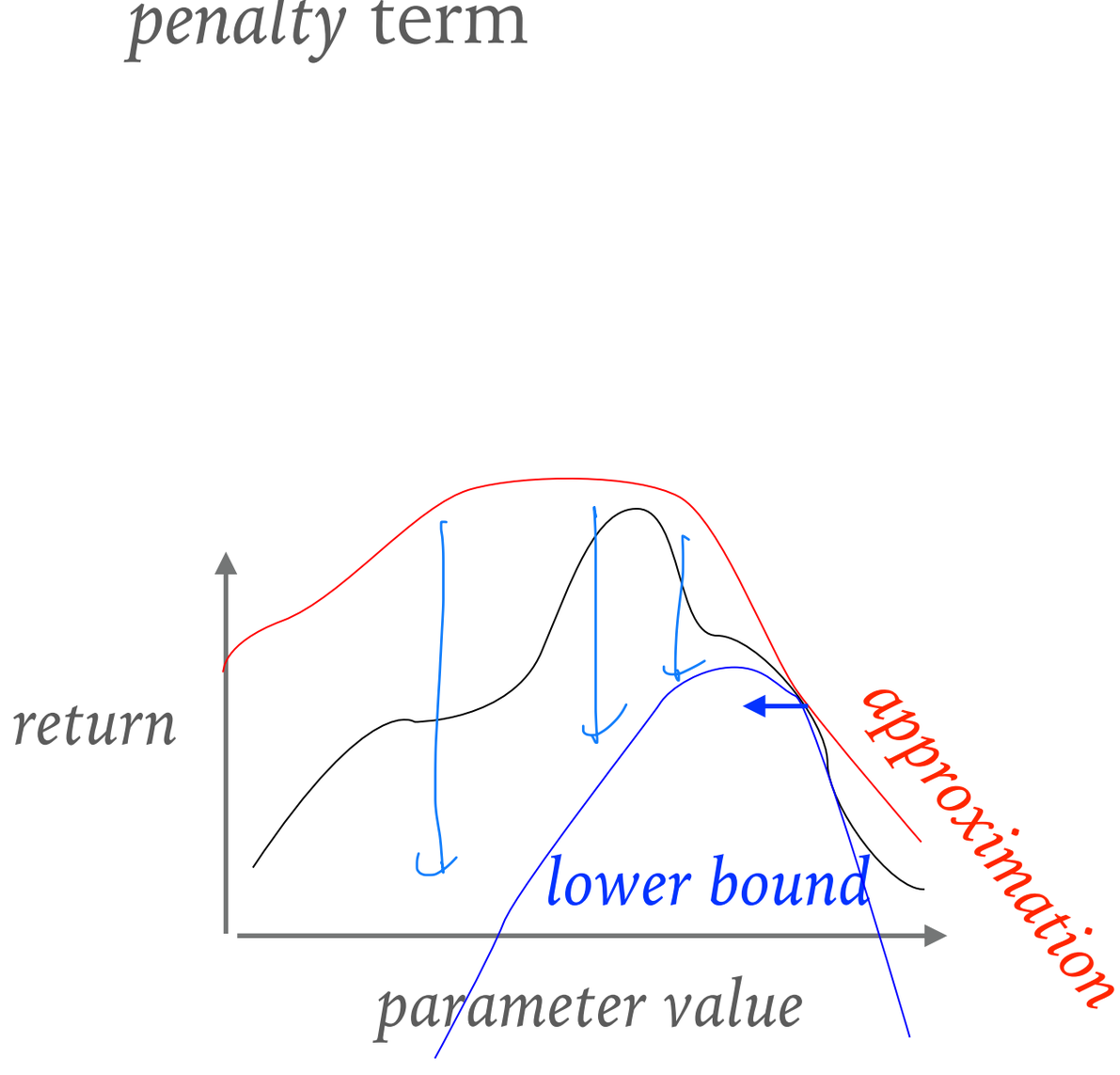
  3. maximize this lower bound (no need to specify step size!)

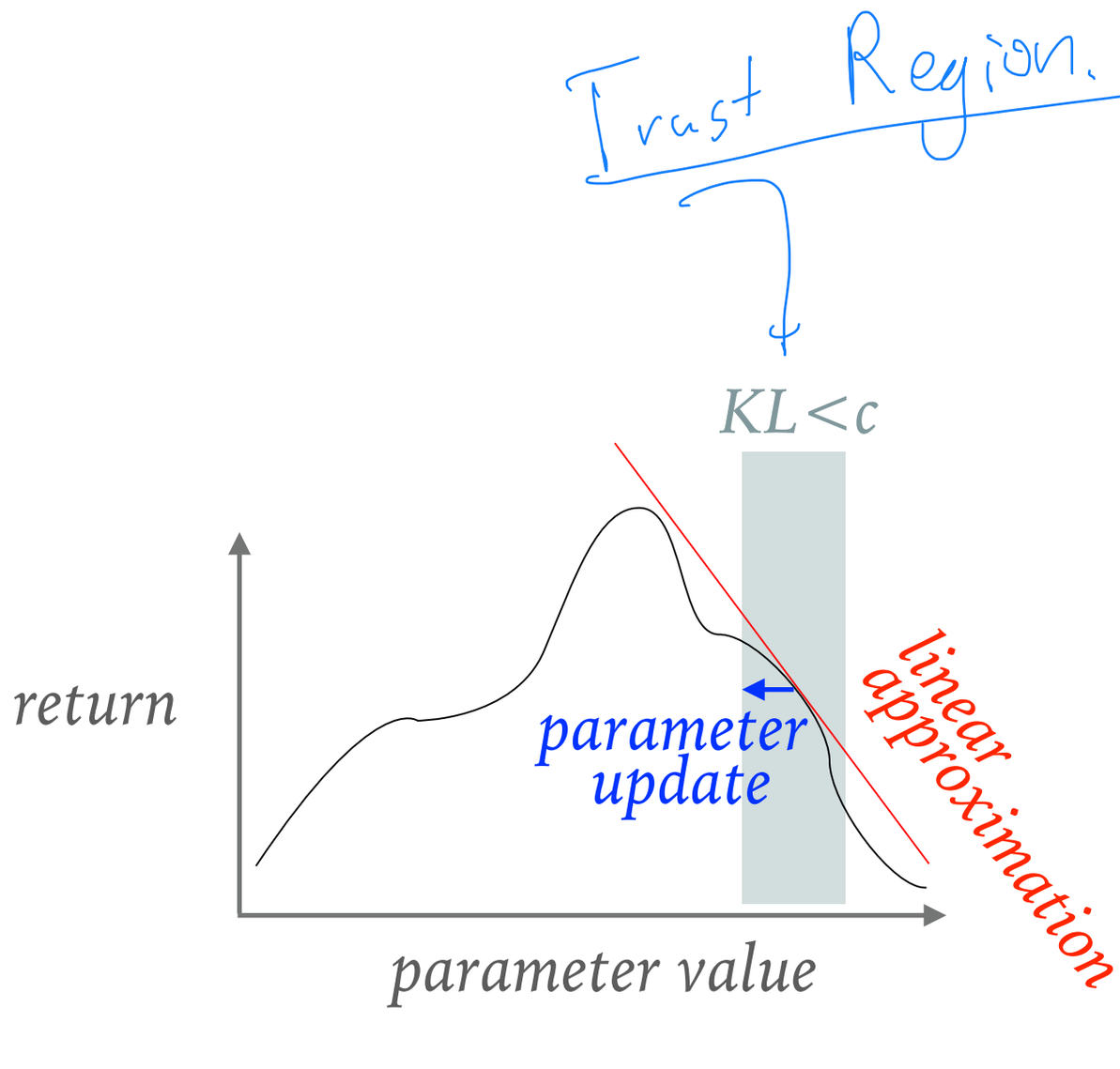


[Schulman 2016]

TRPO: PRACTICE

- Approximate theoretical update by maximising a linearised approximation under a KL *constraint* inspired by the KL-based *penalty* term





CONNECTION TO NATURAL GRADIENTS

- Maximising linearised performance under KL constraint is very reminiscent of our perspective on natural gradients!
- In natural gradients, we found the update direction with maximal improvement per unit KL

  $\mathbf{d}\boldsymbol{\theta} \propto \tilde{\nabla} J = F^{-1} \nabla_{\boldsymbol{d\theta}} J\left(\boldsymbol{\theta}_0 + \boldsymbol{d\theta}\right)$

  taking a fixed-size step β.
- The optimal step size might not be same everywhere in parameter space
- Instead: derive step-size from KL constraint.

CONNECTION TO NATURAL GRADIENTS

- Instead: derive step-size from KL constraint. Write $\mathbf{d}\boldsymbol{\theta} = \beta\tilde{\nabla} J$ and look again at the Taylor expansion of the expected KL

$$c = \mathbb{E}_{\mathbf{s}}\left[D_{\mathrm{KL}}(\pi(\mathbf{a}|\mathbf{s};\boldsymbol{\theta}_0)\|\pi(\mathbf{a}|\mathbf{s};\boldsymbol{\theta}_0 + \boldsymbol{d\theta})\right] = \mathrm{EKL}(\boldsymbol{d\theta})$$

$$\approx \mathrm{EKL}(\mathbf{0}) + d\boldsymbol{\theta}^T \nabla_{d\boldsymbol{\theta}} \mathrm{EKL}(\boldsymbol{d\theta}) + \frac{1}{2} d\boldsymbol{\theta}^T (\nabla^2_{d\theta} \mathrm{EKL}(\boldsymbol{d\theta}) d\boldsymbol{\theta}$$

$$c = \frac{1}{2} \cdot \beta (\tilde{\nabla} J)^T (\nabla_{d\theta} EKL(d\theta)) \beta (\tilde{\nabla} J)$$

$$\beta = \frac{1}{\sqrt{\frac{1}{c} \cdot \frac{1}{2} \cdot (\tilde{\nabla} J)^T (\nabla_{d\theta} EKL(d\theta))(\tilde{\nabla} J)}}$$

CONNECTION TO NATURAL GRADIENTS

- Instead: derive step-size from KL constraint. Write $\mathbf{d}\boldsymbol{\theta} = \beta\tilde{\nabla}J$ and look again at the Taylor expansion of the expected KL

$$c = \mathbb{E}_{\mathbf{s}}\left[D_{\mathrm{KL}}(\pi(\mathbf{a}|\mathbf{s};\boldsymbol{\theta}_0)\|\pi(\mathbf{a}|\mathbf{s};\boldsymbol{\theta}_0+\boldsymbol{d\theta})\right] = \mathrm{EKL}(\boldsymbol{d\theta})$$

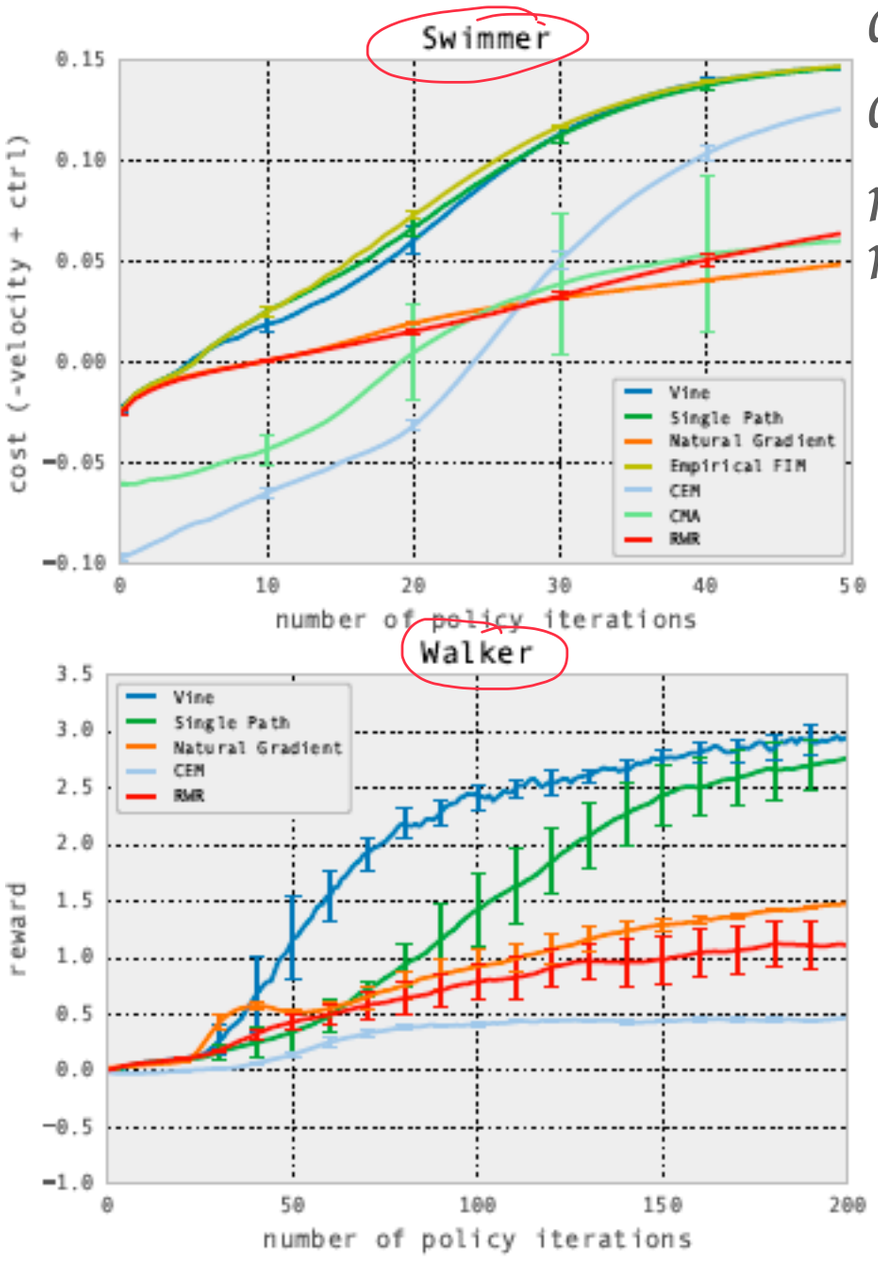
$$\approx \mathrm{EKL}\left(\mathbf{0}\right) + d\boldsymbol{\theta}^T\nabla_{d\boldsymbol{\theta}}\,\mathrm{EKL}(\boldsymbol{d\theta}) + \frac{1}{2}d\boldsymbol{\theta}^T(\nabla^2_{d\theta}\mathrm{EKL}(\boldsymbol{d\theta})d\boldsymbol{\theta}$$

$$= 0 \qquad + \frac{1}{2}\beta^2(\tilde{\nabla}J)^T\left(\nabla^2_{d\theta}\mathrm{EKL}\left(\boldsymbol{d\theta}\right)\right)(\tilde{\nabla}J)$$

F

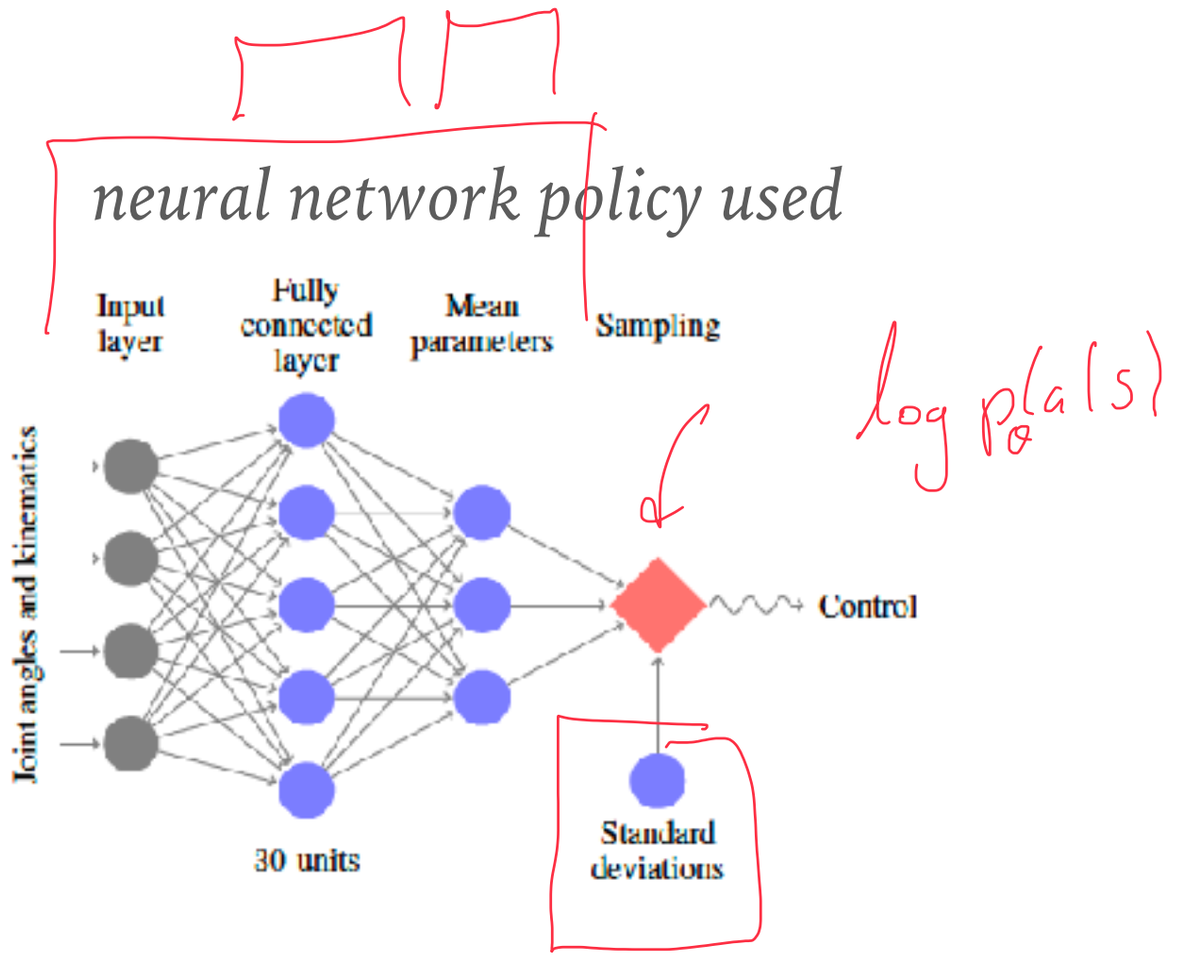
- $c \approx \beta^2(\tilde{\nabla}J)^T F(\tilde{\nabla}J)/2$ can now be solved to get β
- Based on approximation of objective and KL
- Make sure constraint is met using analytic objective & KL
- Additional tricks to work with large number of parameters (NN)

TRPO EVALUATION



different TRPO variants
direct policy search (meta heuristic)
natural gradients,
reward-weighted regression

neural network policy used



[Schulman, 2015]

TRPO EXAMPLE

TRPO with generalised advantage estimate, [Schulman 2016]

TRPO EXAMPLE

TRPO with generalised advantage estimate, [Schulman 2016]

Big picture: How to learn policies

$$D = \{(\mathbf{s}_i, \mathbf{a}_i, r_i, \mathbf{s}'_i)\}_{i=1 \ldots N}$$

Learn value function
$V(\mathbf{s}), Q(\mathbf{s}, \mathbf{a})$

Policy
$\pi(\mathbf{a}|\mathbf{s})$

Optimize policy
$\pi(\mathbf{a}|\mathbf{s})$

Learn dynamics model
$p(\mathbf{s}'|\mathbf{s}, \mathbf{a})$ $r(\mathbf{s}, \mathbf{a})$

Learn value or policy
$V(\mathbf{s}), Q(\mathbf{s}, \mathbf{a}), \pi(\mathbf{a}|\mathbf{s})$

Policy
$\pi(\mathbf{a}|\mathbf{s})$

Value-based

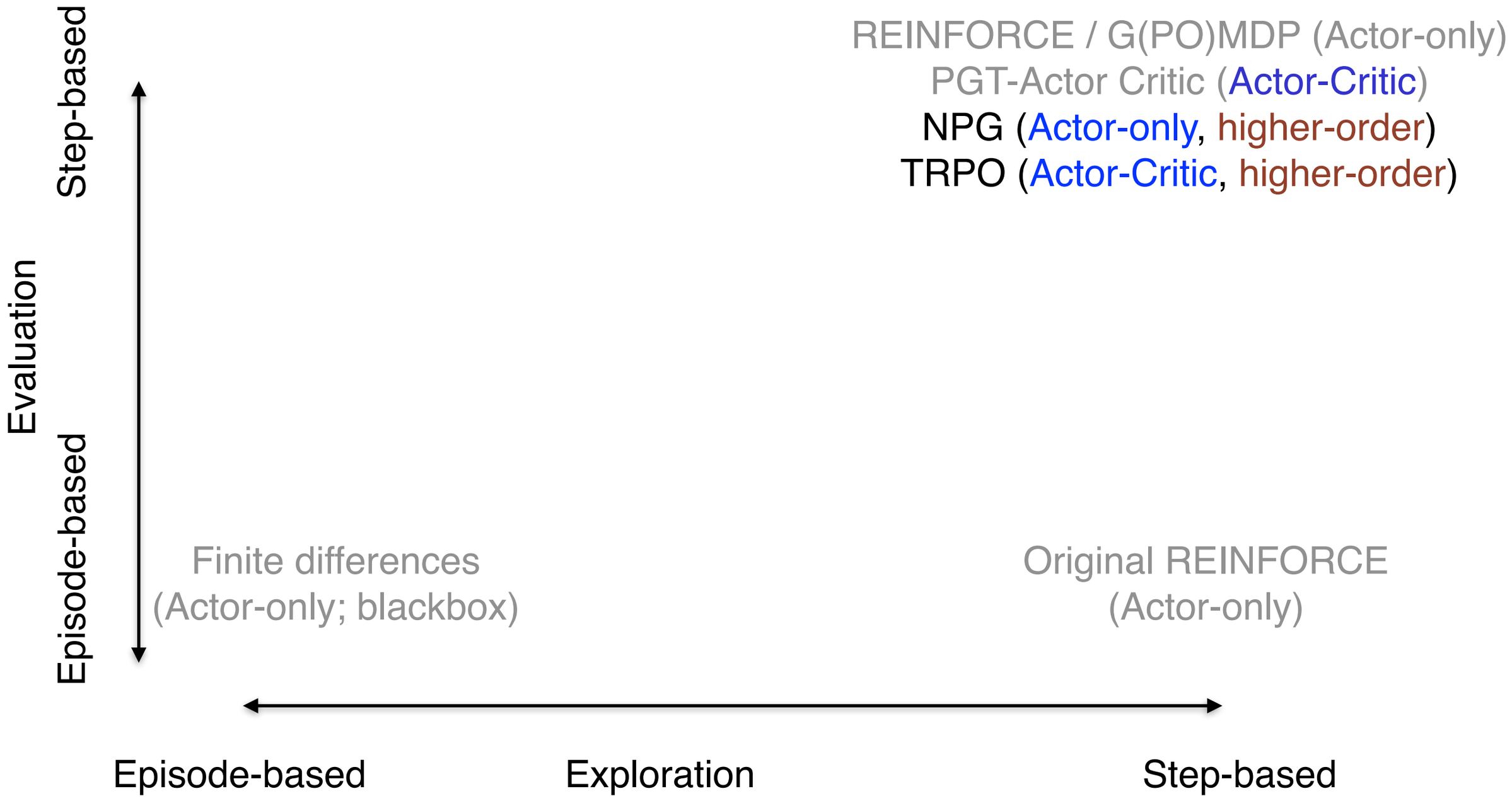
Policy-based

Model-free RL

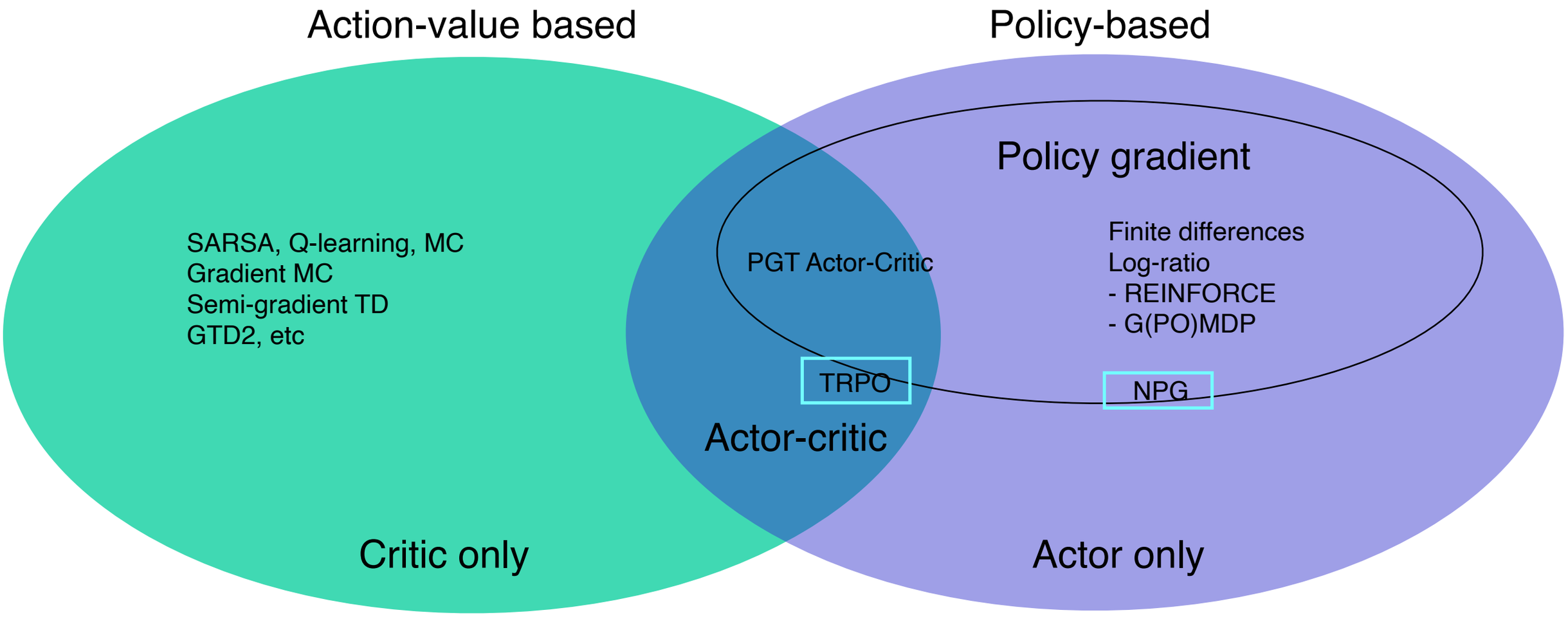
Model-based RL

Thanks to Jan Peters

Comparison


Step-based
Evaluation
Episode-based
REINFORCE / G(PO)MDP (Actor-only)
PGT-Actor Critic (Actor-Critic)
NPG (Actor-only, higher-order)
TRPO (Actor-Critic, higher-order)
Finite differences
(Actor-only; blackbox)
Original REINFORCE
(Actor-only)
Episode-based
Exploration
Step-based

Policies and action-values



CONCLUSIONS

- NPG, TRPO: all use improved metric for policy updates, exploit structure of policy
- Generally allows taking larger steps in policy space than 'vanilla' methods

→ Both have non-trivial computational cost compared to P-G.

- NPG: easier to implement, still manual step size
- TRPO: even larger steps (faster), step size from KL constraint

WHAT YOU SHOULD REMEMBER

- Advantage of covariant representation of distances?
- Advantage of specifying constraint instead of stepsize?
- Why do we need a constraint / penalty / stepsize?

Thanks for your attention!

Feedback?

h.c.vanhoof@uva.nl

REFERENCES

- [Bagnell 2003] Bagnell, J.A. and Schneider, J. Covariant policy search. IJCAI.
- [Kakade 2002] Kakade, S., 2002. A natural policy gradient. In: NeurIPS
- [Peters 2008] Peters, J. and Schaal, S., 2008. Natural actor-critic. Neurocomputing, 71(7), pp.1180-1190
- [Sutton 2000] Sutton, R.S., McAllester, D., Singh, S. & Mansour, Y. Policy Gradient Methods for Reinforcement Learning with Function Approximation. In: NeurIPS.
- [Schulman 2015] Schulman, J., Levine, S., Abbeel, P., Jordan, M.I. and Moritz, P. Trust Region Policy Optimization. In: ICML
- [Schulman 2016] Schulman, J., Moritz, P., Levine, S., Jordan, M.I. and Abbeel, P. High-dimensional continuous control using generalised advantage estimates. In ICLR.