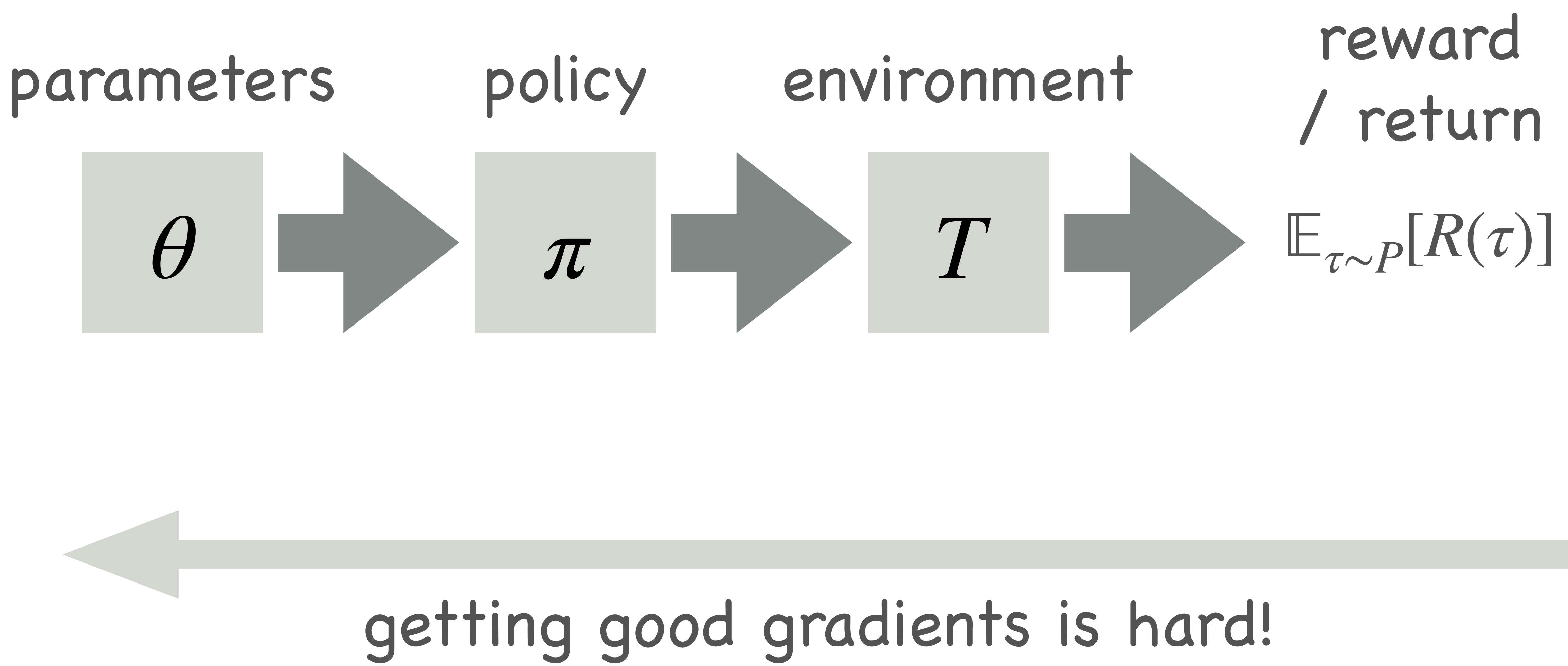
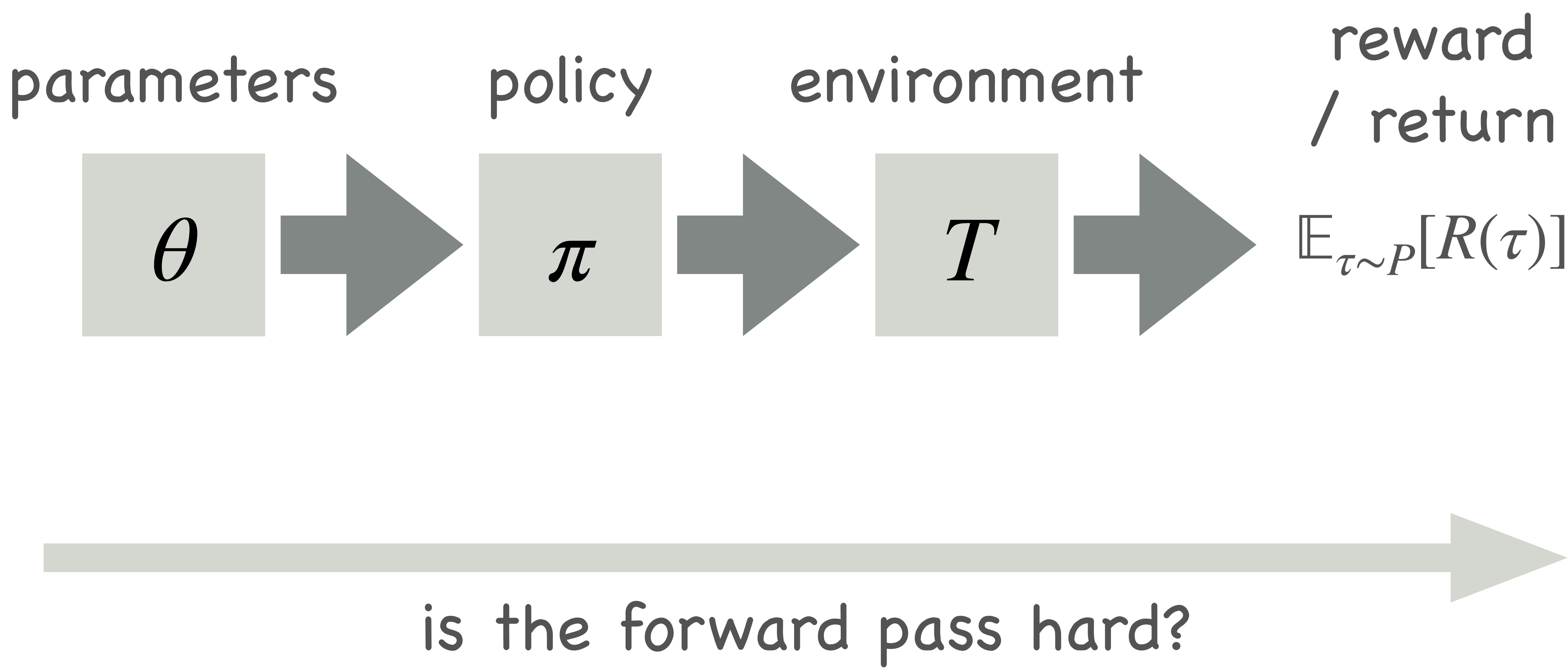
Gradient free optimization

Why do we need a gradient?



Why do we need a gradient?



What if we had an oracle?

- Given policies π_A and π_B
- which one is better?
  - rollout and compute return

Gradient Free Optimization

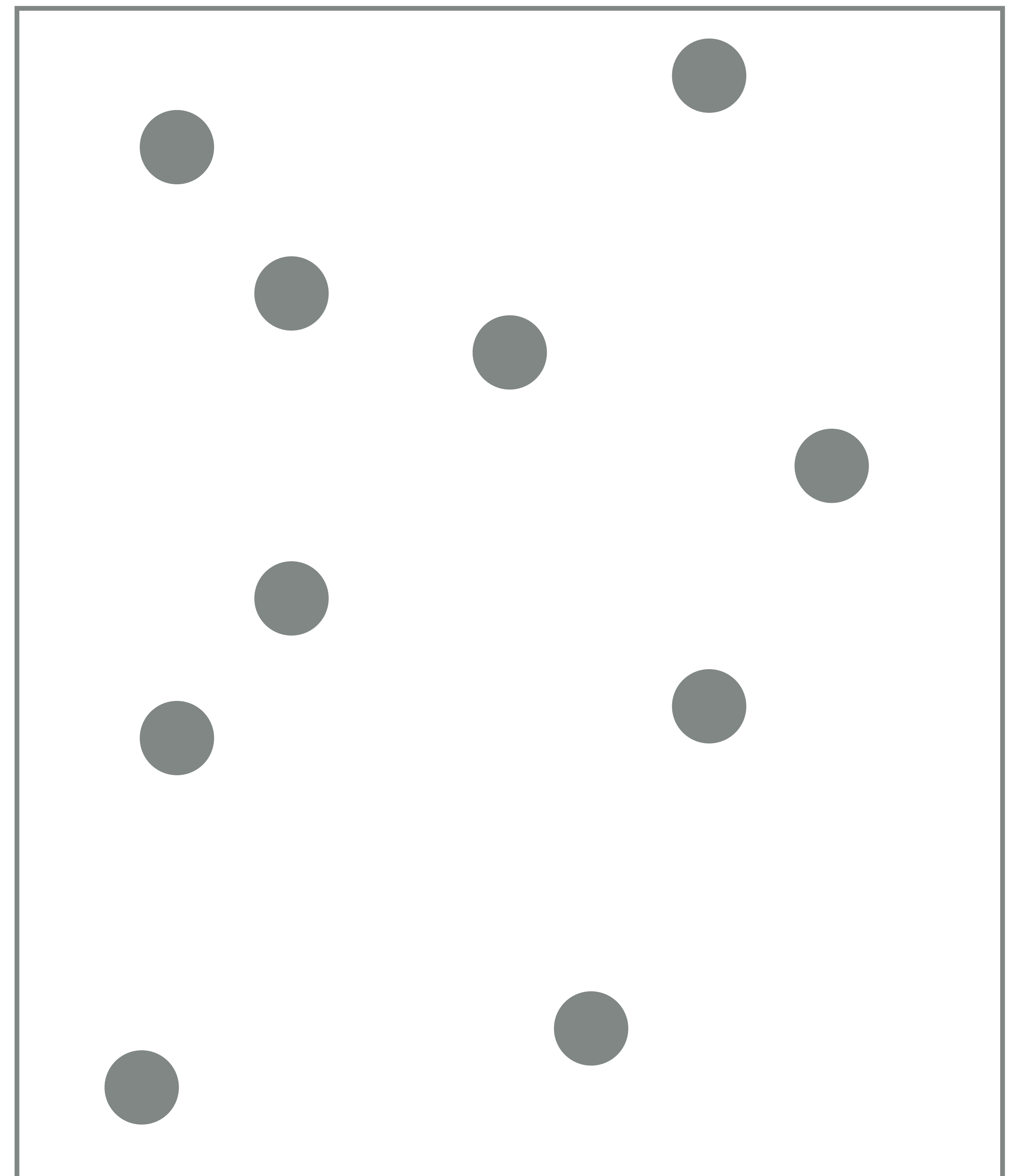
- maximize $f(\theta)$ w.r.t. θ
  - can only evaluate function value
  - no gradients
  - f is smooth
    - similar θ produce similar $f(\theta)$

black box
$f(\theta)$

Random Search

- Randomly sample θ
- pick highest $f(\theta)$

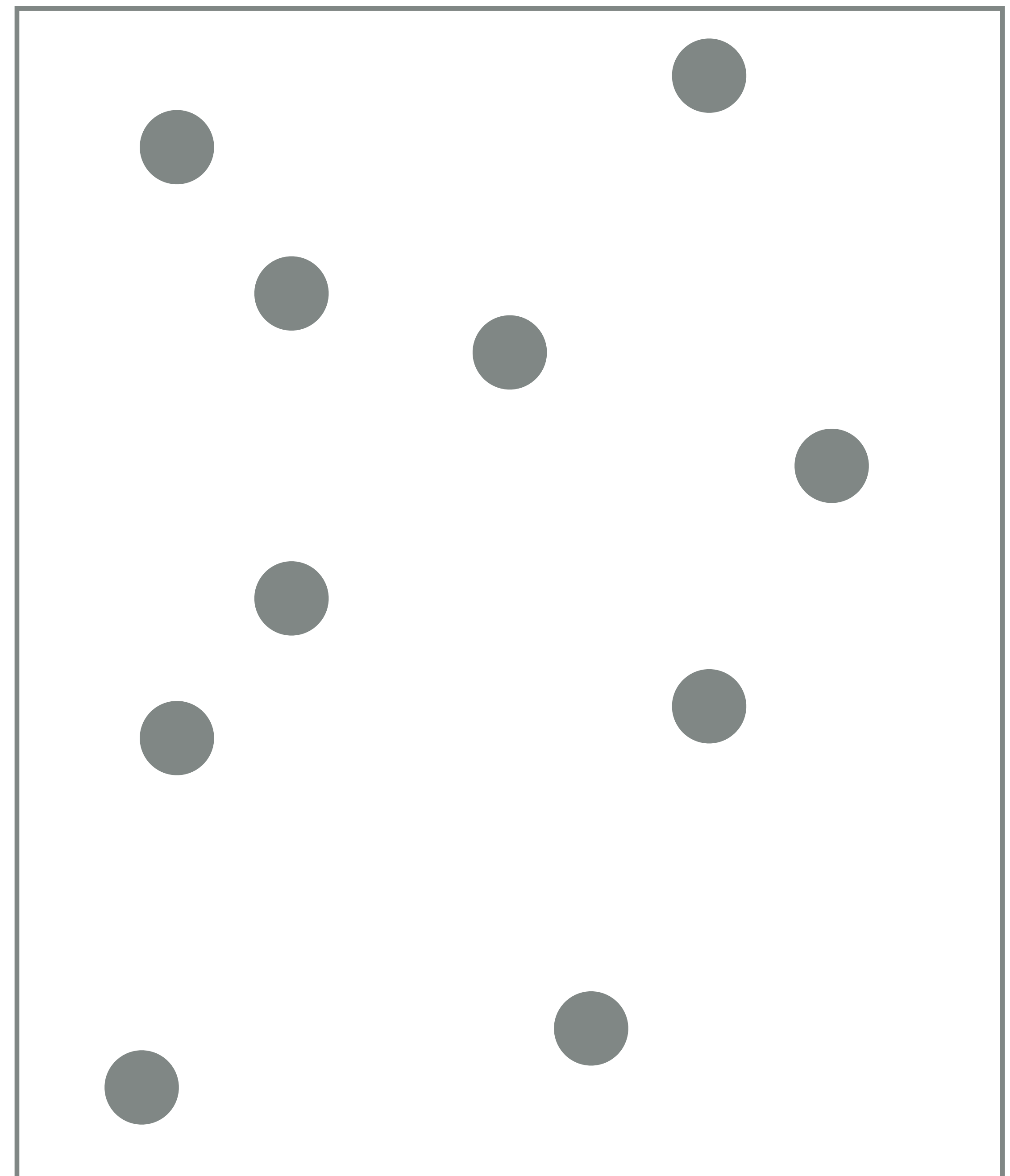
parameters space



Iterative Random Search

- Randomly sample θ
  - pick highest $f(\theta)$
  - Sample more points around maxima
  - repeat

parameters space



Cross entropy method

parameters space

- Initialize μ, σ
  - sample $\theta \sim \mathcal{N}(\mu, \sigma^2)$
  - compute reward $f(\theta)$
  - select top $p\,\%$ $(p = 20)$
  - fit Gaussian for new μ, σ
  - repeat

Evolution Strategies

parameters space

- Initialize θ
- Iterate
  - Sample $\epsilon_0, \epsilon_1, \ldots, \epsilon_n \sim \mathcal{N}(0,I)$
  - Compute returns $F_i = R(\theta + \sigma\epsilon_i)$
  - Normalize $\tilde{F}_i = \dfrac{F_i - \mu_F}{\sigma_F}$
  - Update $w := w + \dfrac{\alpha}{\sigma n} \sum_{i=1}^{n} \tilde{F}_i \epsilon_i$

Evolution strategies as a scalable alternative to reinforcement learning, Salimans et al., arXiv 2017

Augmented random search

parameters space

- Initialize θ
- Iterate
  - Sample $\epsilon_0, \epsilon_1, \ldots, \epsilon_n \sim \mathcal{N}(0,I)$
  - Compute returns $F_i^+ = R(\theta + \sigma\epsilon_i)$, $F_i^- = R(\theta - \sigma\epsilon_i)$
  - Update
    $$w := w + \frac{\alpha}{\sigma n}\sum_{i=1}^{n}(F_i^+ - F_i^-)\epsilon_i$$

- Simple random search provides a competitive approach to reinforcement learning, Mania et al., NIPS 2018
- Multivariate Stochastic Approximation Using a Simultaneous Perturbation Gradient Approximation, Spall, *Automatic Control, 1992*

Gradient free optimization

- Exponential in parameter space
- works better if
  - parameter space small
  - parameters correlate with expected return