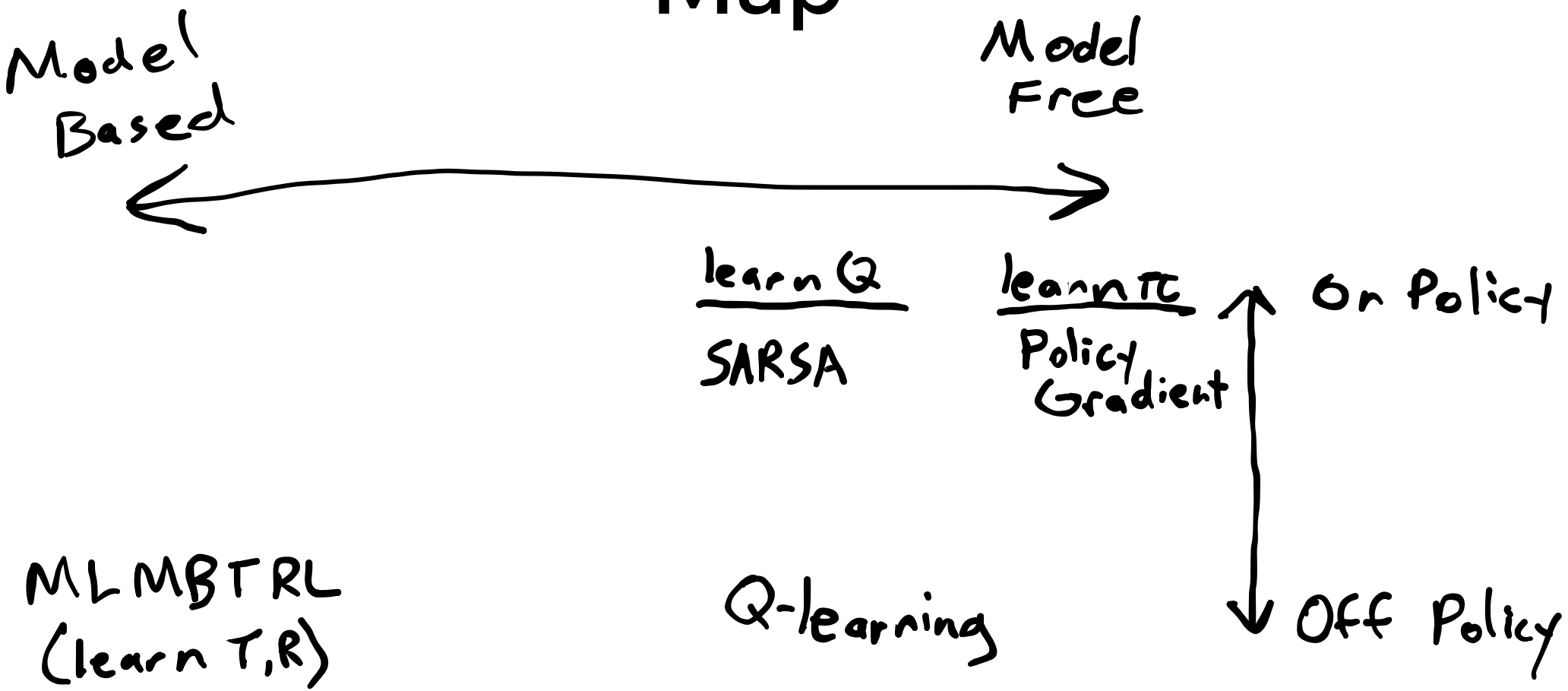
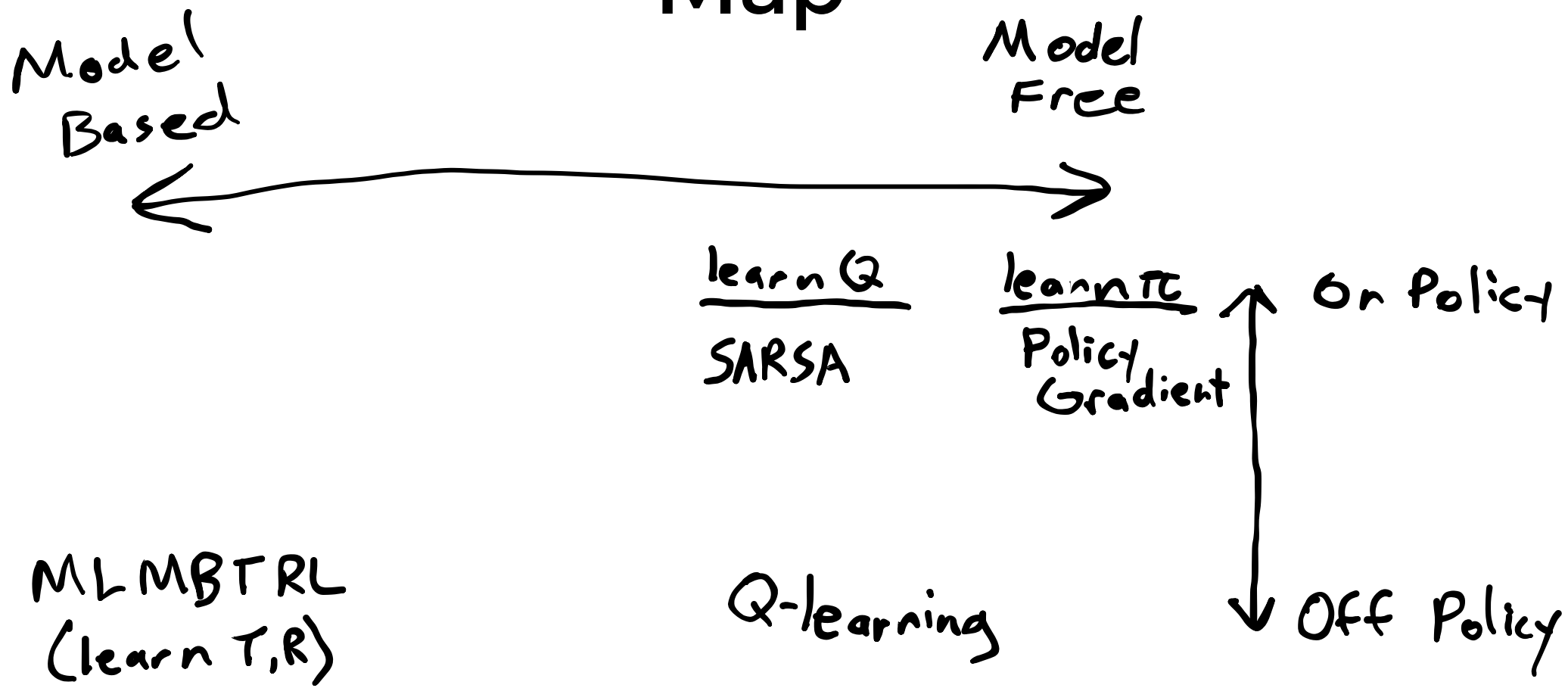
Map

Map


Model Based
Model Free
learn Q
SARSA
learn π
Policy Gradient
On Policy
Off Policy
ML MBTRL (learn T,R)
Q-learning

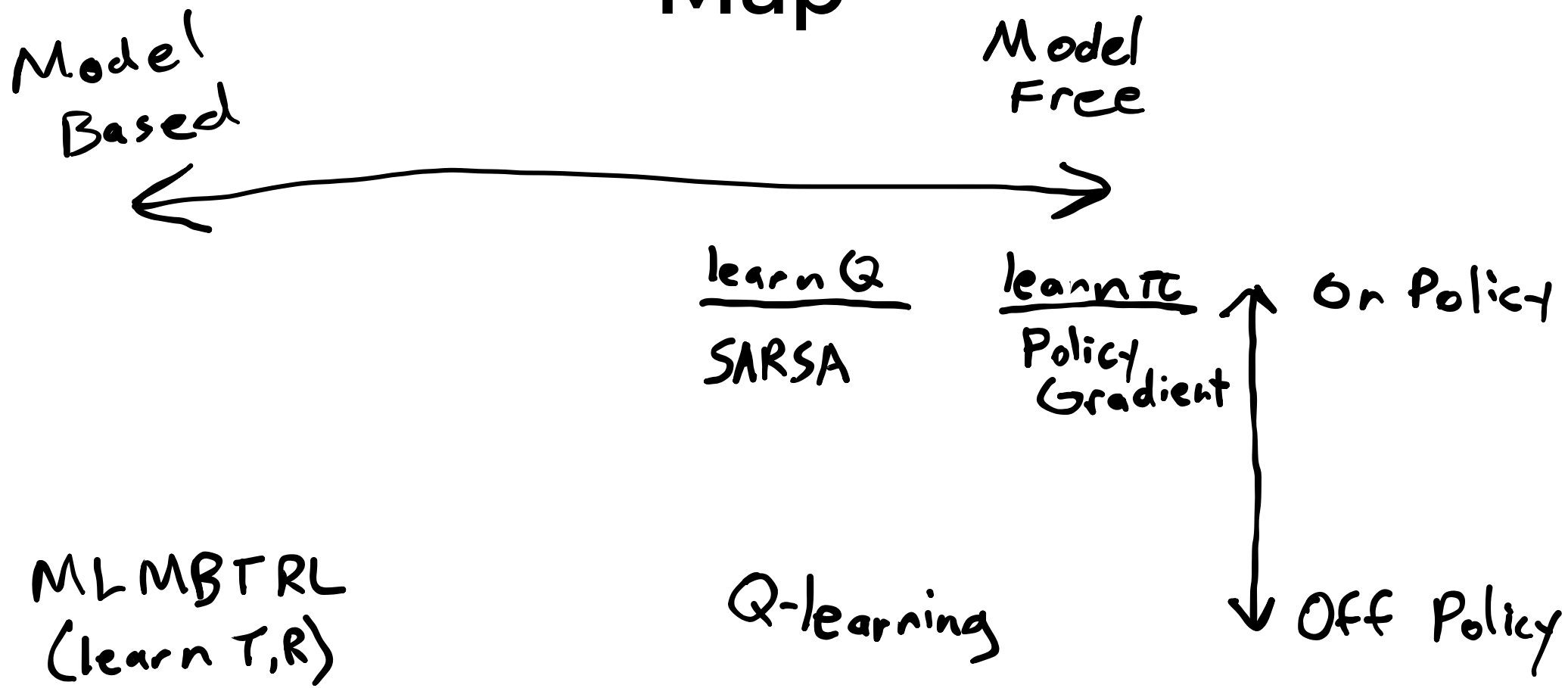
Map



Challenges:

1. Exploration vs Exploitation
2. Credit Assignment
3. Generalization

Map

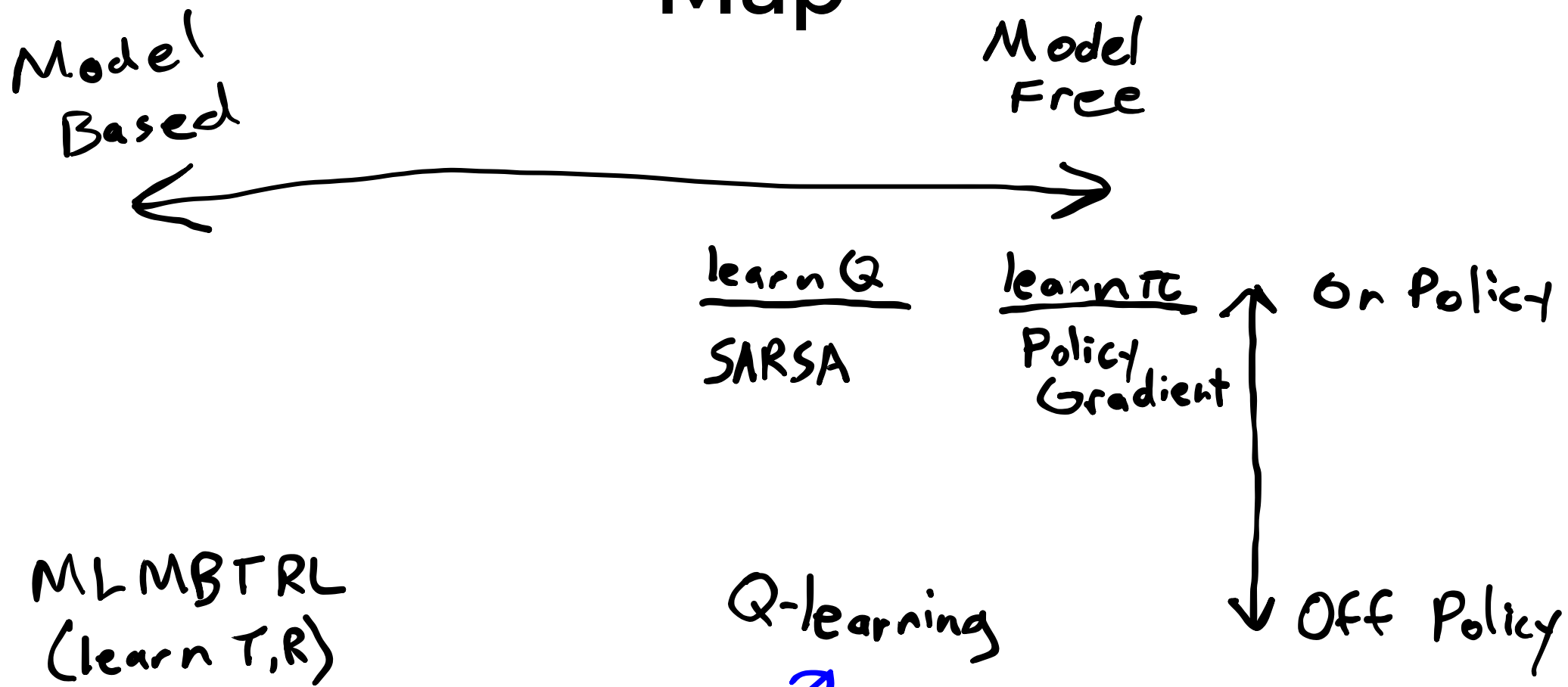


Challenges:

1. Exploration vs Exploitation
2. Credit Assignment
3. Generalization

Last Time: Neural Networks

Map

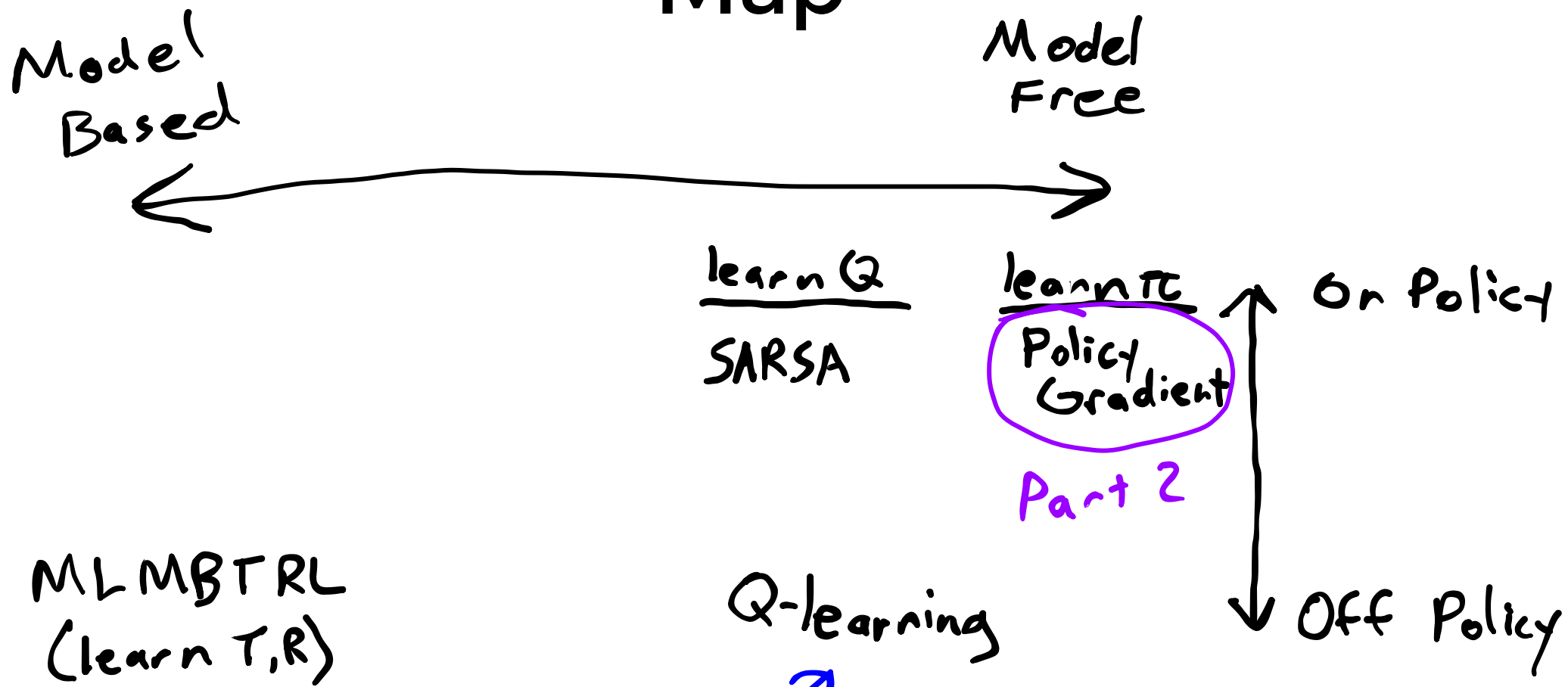


Challenges:

1. Exploration vs Exploitation
2. Credit Assignment
3. Generalization

Last Time: Neural Networks

Map



Challenges:

1. Exploration vs Exploitation
2. Credit Assignment
3. Generalization

Last Time: Neural Networks

Part 1
DQN

Discussion

Discussion

Q-Learning:

$$Q(s,a) \leftarrow Q(s,a) + \alpha \left(r + \gamma \max_{a'} Q(s',a') - Q(s,a)\right)$$

Discussion

Q-Learning:

$$Q(s,a) \leftarrow Q(s,a) + \alpha \left(r + \gamma \max_{a'} Q(s',a') - Q(s,a)\right)$$

Neural Networks

$$\theta^* = \arg\min_\theta \sum_{(x,y)\in\mathcal{D}} l(f_\theta(x), y)$$

$$\theta \leftarrow \theta - \alpha \nabla_\theta \, l(f_\theta(x), y)$$

Discussion

Q-Learning:

$$Q(s,a) \leftarrow Q(s,a) + \alpha \left(r + \gamma \max_{a'} Q(s',a') - Q(s,a)\right)$$

Neural Networks

$$\theta^* = \arg\min_\theta \sum_{(x,y)\in\mathcal{D}} l(f_\theta(x), y)$$

$$\theta \leftarrow \theta - \alpha \nabla_\theta \, l(f_\theta(x), y)$$

Deep Q learning:

- Approximate Q with Q_θ

Discussion

Q-Learning:

$$Q(s,a) \leftarrow Q(s,a) + \alpha \left(r + \gamma \max_{a'} Q(s',a') - Q(s,a)\right)$$

Neural Networks

$$\theta^* = \arg\min_\theta \sum_{(x,y)\in\mathcal{D}} l(f_\theta(x), y)$$

$$\theta \leftarrow \theta - \alpha \nabla_\theta \, l(f_\theta(x), y)$$

Deep Q learning:

- Approximate Q with Q_θ
- What should (x, y) be?

Discussion

Q-Learning:

$$Q(s,a) \leftarrow Q(s,a) + \alpha \left(r + \gamma \max_{a'} Q(s',a') - Q(s,a)\right)$$

Neural Networks

$$\theta^* = \arg\min_\theta \sum_{(x,y)\in\mathcal{D}} l(f_\theta(x), y)$$

$$\theta \leftarrow \theta - \alpha \nabla_\theta \, l(f_\theta(x), y)$$

Deep Q learning:

- Approximate Q with Q_θ
- What should (x, y) be?
- What should l be?

Discussion

Q-Learning:

$$Q(s,a) \leftarrow Q(s,a) + \alpha\ (r + \gamma \max_{a'} Q(s',a') - Q(s,a))$$

Neural Networks

$$\theta^* = \arg\min_\theta \sum_{(x,y)\in\mathcal{D}} l(f_\theta(x), y)$$

$$\theta \leftarrow \theta - \alpha \nabla_\theta\, l(f_\theta(x), y)$$

Deep Q learning:

- Approximate Q with Q_θ
- What should (x, y) be?
- What should l be?

Candidate Algorithm:

loop

$a \leftarrow \text{argmax}\, Q(s,a)$ w.p. $1-\epsilon$, $\quad \text{rand}(A)$ o.w.

$r \leftarrow \text{act!}(\text{env}, a)$

$s' \leftarrow \text{observe}(\text{env})$

$\theta \leftarrow \theta - \alpha \nabla_\theta \|r + \gamma \max_{a'} Q_\theta(s',a') - Q_\theta(s,a)\|_2$

$s \leftarrow s'$

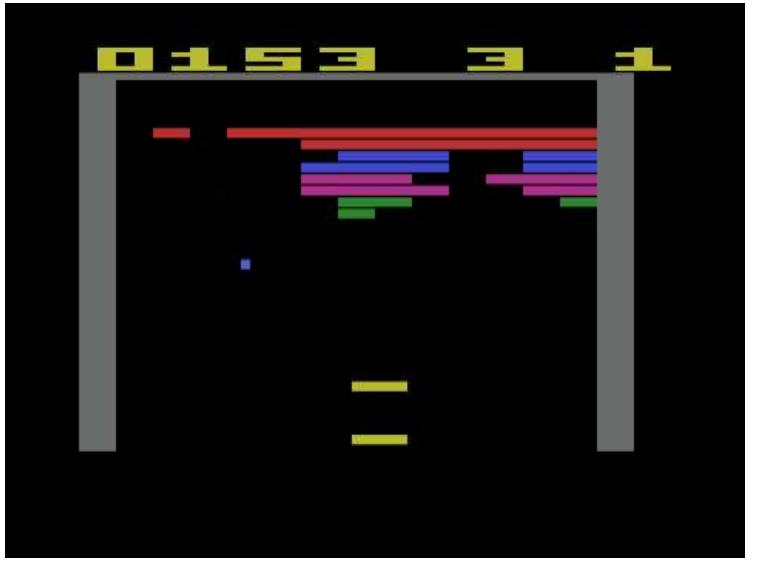
DQN: The Atari Benchmark

DQN: The Atari Benchmark

DQN: The Atari Benchmark

DQN: The Atari Benchmark

DQN: The Atari Benchmark



DQN: Problems with Naive Approach

Problems:

} data buffer / experience replay

} periodically freeze target

DQN: Problems with Naive Approach

Candidate Algorithm:

loop

$$a \leftarrow \text{argmax}\, Q(s,a) \text{ w.p. } 1-\epsilon, \quad \text{rand}(A) \text{ o.w.}$$

$$r \leftarrow \text{act!}(\text{env}, a)$$

$$s' \leftarrow \text{observe}(\text{env})$$

$$\theta \leftarrow \theta - \alpha \nabla_\theta \| r + \gamma \max_{a'} Q_\theta(s', a') - Q_\theta(s,a) \|_2$$

$$s \leftarrow s'$$

Problems:

} data buffer / experience replay

} periodically freeze target

DQN: Problems with Naive Approach

Candidate Algorithm:

loop

$$a \leftarrow \text{argmax}\, Q(s,a) \text{ w.p. } 1-\epsilon, \quad \text{rand}(A) \text{ o.w.}$$

$$r \leftarrow \text{act!}(\text{env}, a)$$

$$s' \leftarrow \text{observe}(\text{env})$$

$$\theta \leftarrow \theta - \alpha \nabla_\theta \| r + \gamma \max_{a'} Q_\theta(s', a') - Q_\theta(s,a) \|_2$$

$$s \leftarrow s'$$

Problems:

1. Samples Highly Correlated

} data buffer / experience replay

} periodically freeze target

DQN: Problems with Naive Approach

Candidate Algorithm:

loop

$$a \leftarrow \text{argmax}\, Q(s,a) \text{ w.p. } 1-\epsilon, \quad \text{rand}(A) \text{ o.w.}$$

$$r \leftarrow \text{act!}(\text{env}, a)$$

$$s' \leftarrow \text{observe}(\text{env})$$

$$\theta \leftarrow \theta - \alpha \nabla_\theta \| r + \gamma \max_{a'} Q_\theta(s', a') - Q_\theta(s,a) \|_2$$

$$s \leftarrow s'$$

Problems:

1. Samples Highly Correlated
2. Size-1 batches

} data buffer / experience replay

} periodically freeze target

DQN: Problems with Naive Approach

Candidate Algorithm:

loop

$$a \leftarrow \text{argmax}\, Q(s,a) \text{ w.p. } 1-\epsilon, \quad \text{rand}(A) \text{ o.w.}$$

$$r \leftarrow \text{act!}(\text{env}, a)$$

$$s' \leftarrow \text{observe}(\text{env})$$

$$\theta \leftarrow \theta - \alpha \nabla_\theta \| r + \gamma \max_{a'} Q_\theta(s', a') - Q_\theta(s,a) \|_2$$

$$s \leftarrow s'$$

Problems:

1. Samples Highly Correlated
2. Size-1 batches
3. Moving target

} data buffer / experience replay

} periodically freeze target

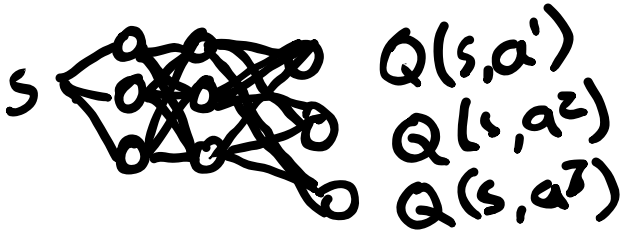
DQN

DQN

Q Network Structure:

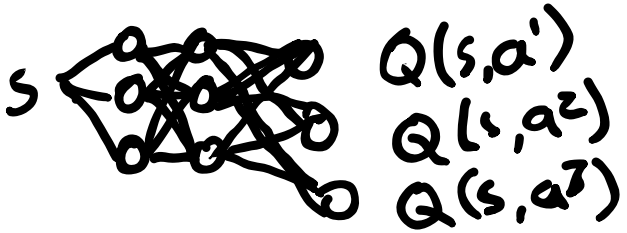
DQN

Q Network Structure:



DQN

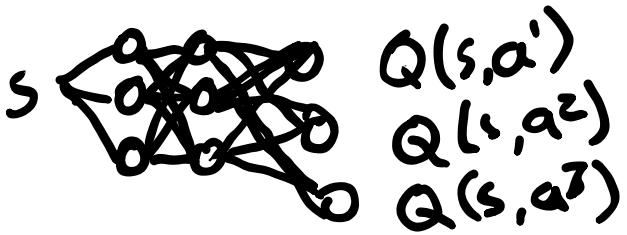
Q Network Structure:



Experience Tuple: (s, a, r, s')

DQN

Q Network Structure:



Experience Tuple: (s, a, r, s')

Loss:

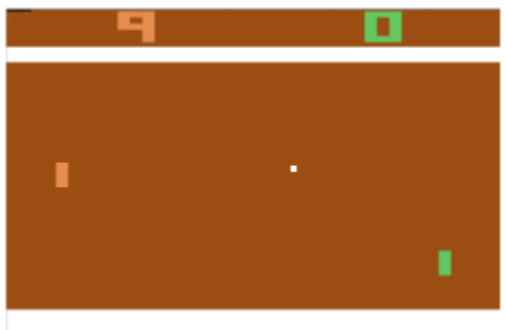
$$l(s, a, r, s') = \left(r + \gamma \max_{a'} Q_{\theta'}(s', a') - Q_{\theta}(s, a)\right)^2$$

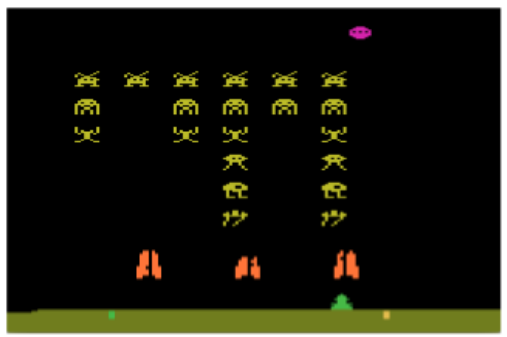
Playing Atari with Deep Reinforcement Learning

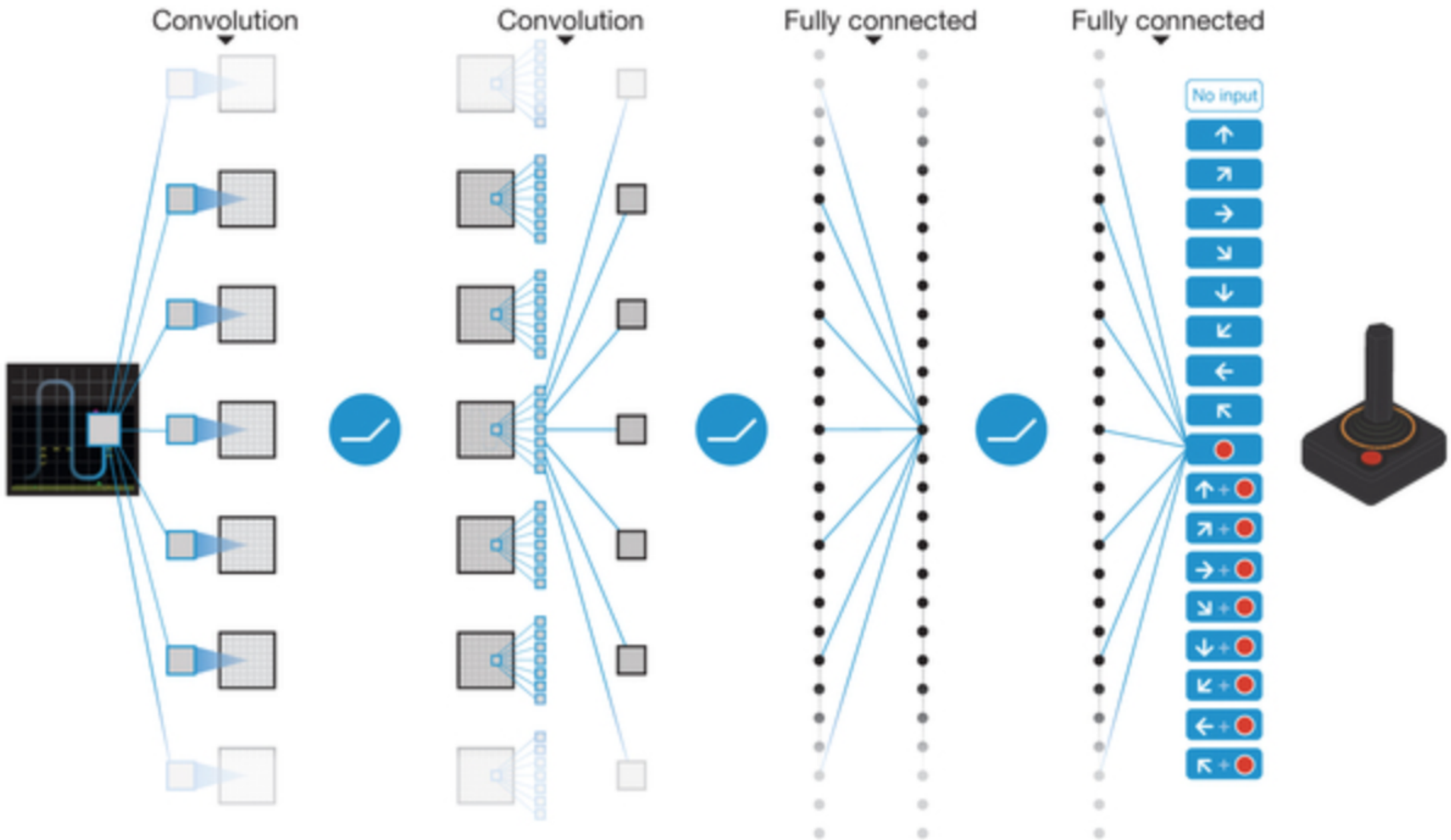
Volodymyr Mnih **Koray Kavukcuoglu** **David Silver** **Alex Graves** **Ioannis Antonoglou**

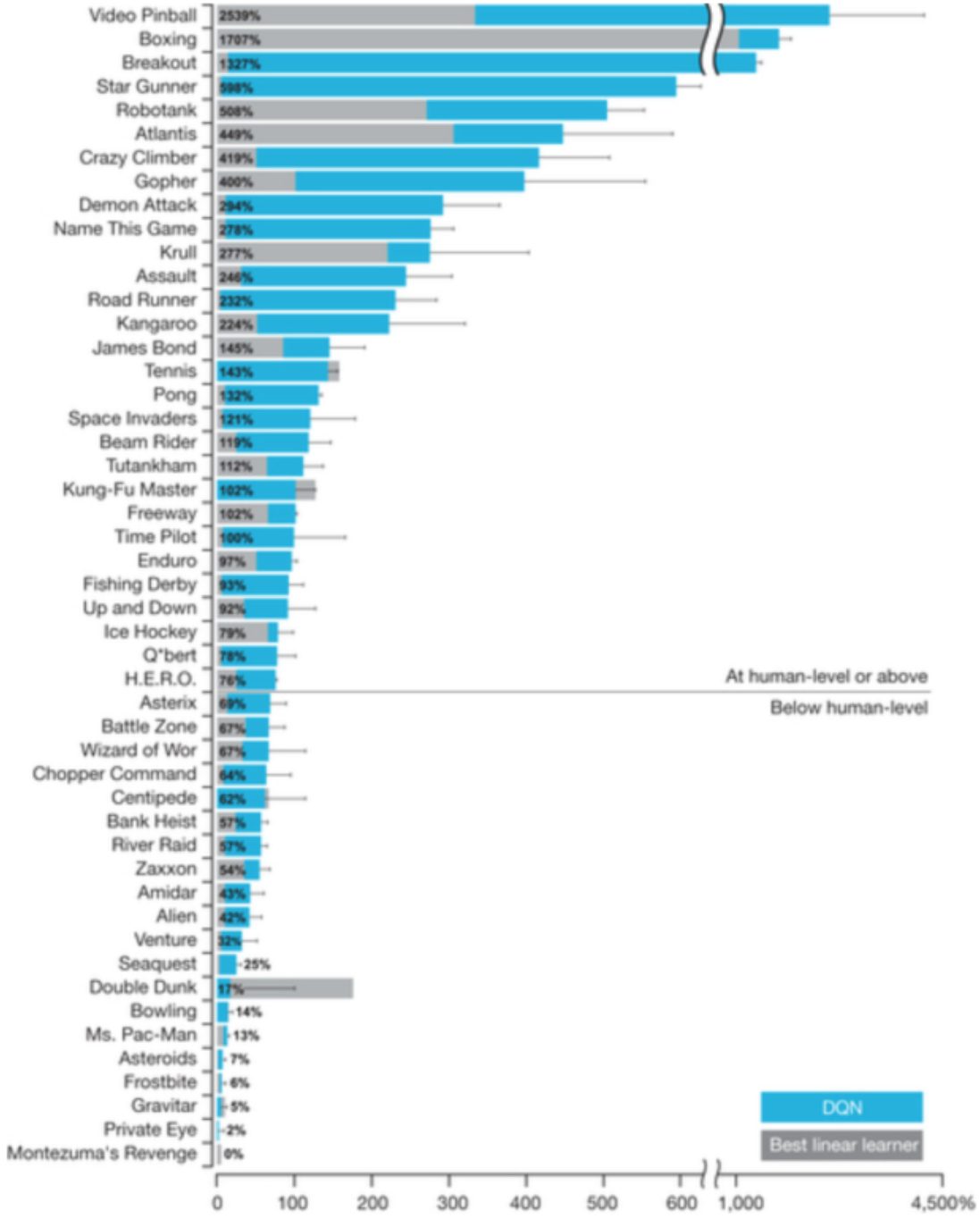
Daan Wierstra **Martin Riedmiller**

DeepMind Technologies

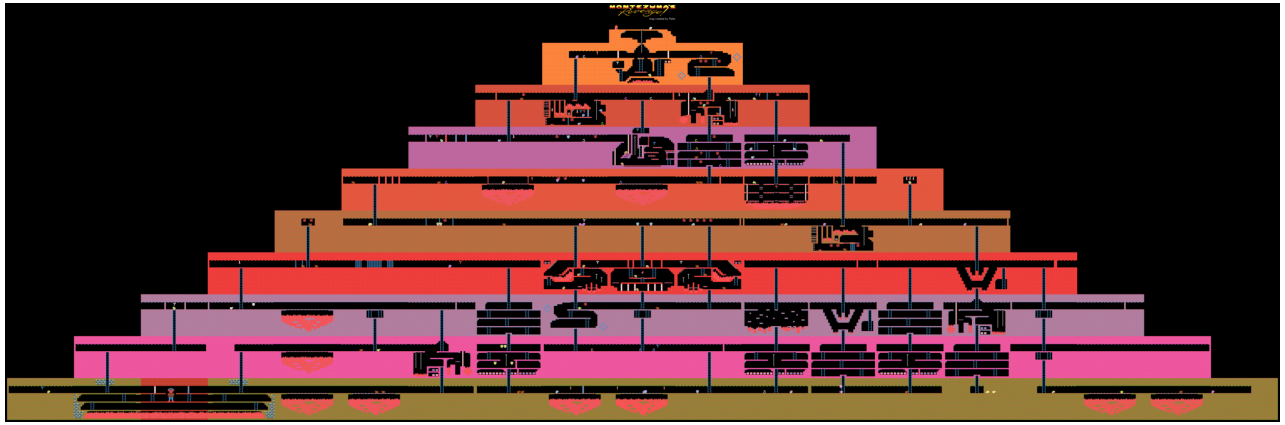






Convolution
Convolution
Fully connected
Fully connected
No input



https://www.youtube.com/watch?v=SuZVyOlgVek



Rainbow

Rainbow

- Double Q Learning

Rainbow

- Double Q Learning
- Prioritized Replay
  (priority proportional to last TD error)

Rainbow

- Double Q Learning
- Prioritized Replay
  (priority proportional to last TD error)
- Dueling networks
  Value network + advantage network
  $Q(s,a) = V(s) + A(s,a)$

Rainbow

- Double Q Learning
- Prioritized Replay
  (priority proportional to last TD error)
- Dueling networks
  Value network + advantage network
  $Q(s,a) = V(s) + A(s,a)$
- Multi-step learning
  $(r_t + \gamma r_{t+1} + \ldots + \gamma^{n-1} r_{t+n-1} + \gamma \max Q_\theta(s_{t+n}, a') - Q_\theta(s_t, a_t))^2$

Rainbow

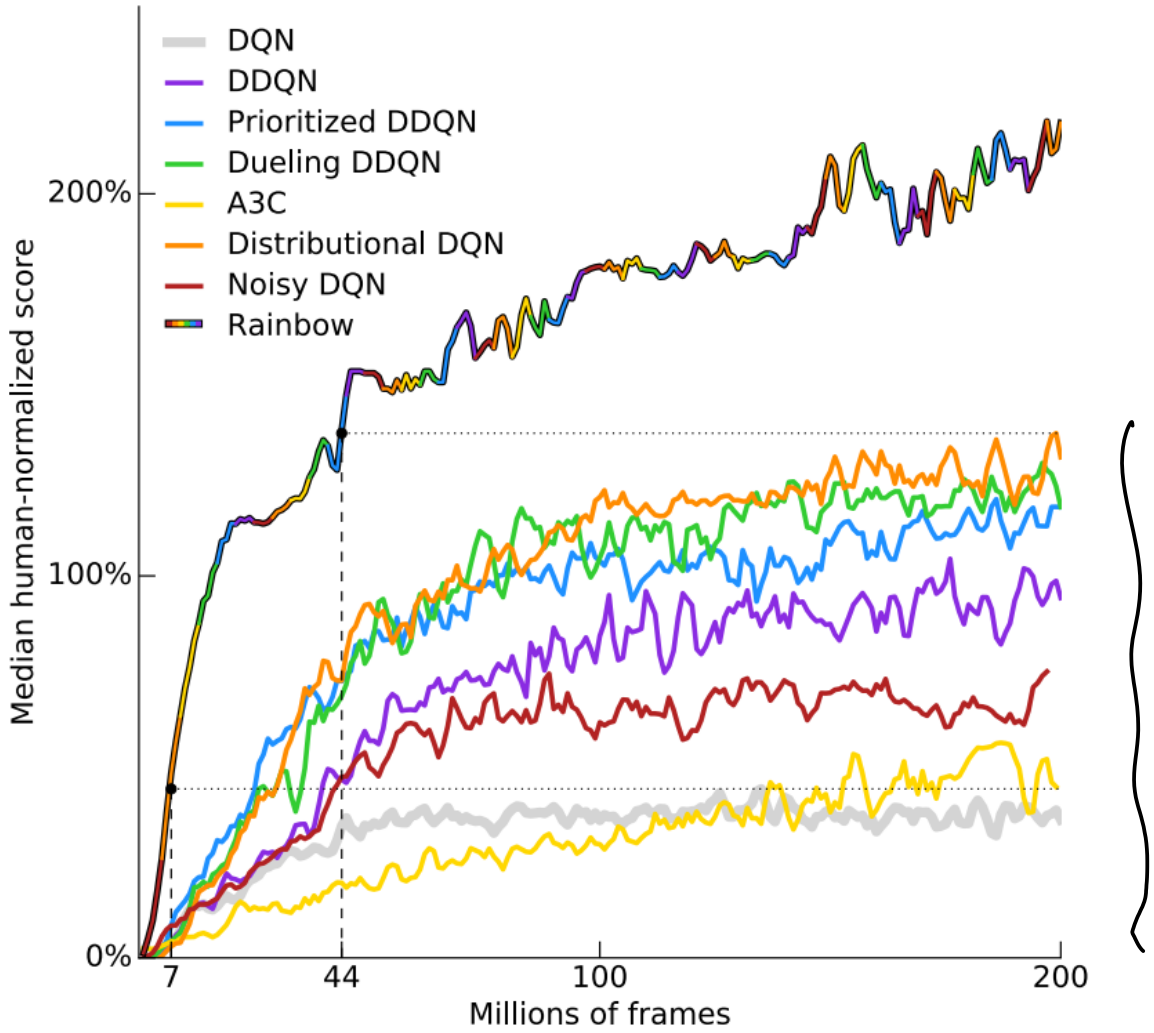
- Double Q Learning
- Prioritized Replay
  (priority proportional to last TD error)
- Dueling networks
  Value network + advantage network
  $Q(s,a) = V(s) + A(s,a)$
- Multi-step learning
  $(r_t + \gamma r_{t+1} + \ldots + \gamma^{n-1} r_{t+n-1} + \gamma \max Q_\theta(s_{t+n}, a') - Q_\theta(s_t, a_t))^2$
- Distributional RL
  predict an entire distribution of values instead of just Q

Rainbow

- Double Q Learning
- Prioritized Replay
  (priority proportional to last TD error)
- Dueling networks
  Value network + advantage network
  $Q(s,a) = V(s) + A(s,a)$
- Multi-step learning
  $(r_t + \gamma r_{t+1} + \ldots + \gamma^{n-1} r_{t+n-1} + \gamma \max Q_\theta(s_{t+n}, a') - Q_\theta(s_t, a_t))^2$
- Distributional RL
  predict an entire distribution of values instead of just Q
- Noisy Nets

Rainbow

- Double Q Learning
- Prioritized Replay
  (priority proportional to last TD error)
- Dueling networks
  Value network + advantage network
  $Q(s,a) = V(s) + A(s,a)$
- Multi-step learning
  $(r_t + \gamma r_{t+1} + \ldots + \gamma^{n-1} r_{t+n-1} + \gamma \max Q_\theta(s_{t+n}, a') - Q_\theta(s_t, a_t))^2$
- Distributional RL
  predict an entire distribution of values instead of just Q
- Noisy Nets



Part 2
Improved Policy Gradients

Restricted Gradient Update

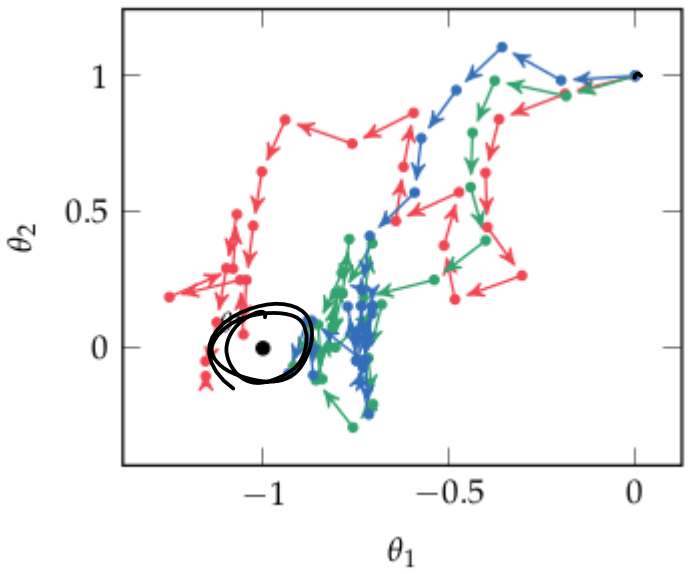
Restricted Gradient Update

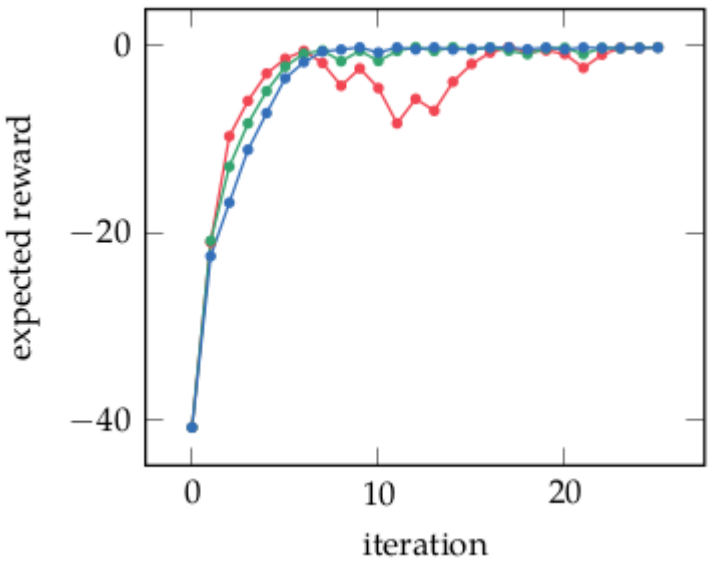
$$\nabla U(\theta) = E_\tau \left[\sum_{k=0}^{d} \nabla_\theta \log \pi_\theta(a_k \mid s_k) \gamma^k (r_{k,\text{to-go}} - r_{\text{base}}(s_k)) \right]$$

Restricted Gradient Update

$$\nabla U(\theta) = E_\tau \left[\sum_{k=0}^{d} \nabla_\theta \log \pi_\theta(a_k \mid s_k) \gamma^k (r_{k,\text{to-go}} - r_{\text{base}}(s_k)) \right]$$

$$\theta' = \theta + \alpha \nabla U(\theta)$$





likelihood ratio
reward-to-go
baseline subtraction

Restricted Gradient Update

$$\nabla U(\theta) = E_\tau \left[\sum_{k=0}^{d} \nabla_\theta \log \pi_\theta(a_k \mid s_k) \gamma^k (r_{k,\text{to-go}} - r_{\text{base}}(s_k)) \right]$$

$$\theta' = \theta + \alpha \nabla U(\theta)$$

$$U(\theta') \approx U(\theta) + \nabla U(\theta)^\top (\theta' - \theta)$$

Restricted Gradient Update

$$\nabla U(\theta) = E_\tau \left[\sum_{k=0}^{d} \nabla_\theta \log \pi_\theta(a_k \mid s_k) \gamma^k (r_{k,\text{to-go}} - r_{\text{base}}(s_k)) \right]$$

$$\theta' = \theta + \alpha \nabla U(\theta)$$

$$\begin{aligned} \underset{\theta'}{\text{maximize}} \quad & U(\theta') \approx U(\theta) + \nabla U(\theta)^\top (\theta' - \theta) \\ \text{subject to} \quad & \end{aligned}$$

Restricted Gradient Update

$$\nabla U(\theta) = E_\tau \left[\sum_{k=0}^{d} \nabla_\theta \log \pi_\theta(a_k \mid s_k) \gamma^k (r_{k,\text{to-go}} - r_{\text{base}}(s_k)) \right]$$

$$\theta' = \theta + \alpha \nabla U(\theta)$$

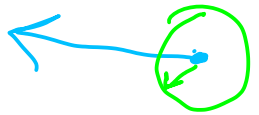
$$\begin{aligned} \underset{\theta'}{\text{maximize}} \quad & U(\theta') \approx U(\theta) + \nabla U(\theta)^\top (\theta' - \theta) \\ \text{subject to} \quad & g(\theta, \theta') \leq \epsilon \end{aligned}$$

Restricted Gradient Update

$$\nabla U(\theta) = E_\tau \left[\sum_{k=0}^{d} \nabla_\theta \log \pi_\theta(a_k \mid s_k) \gamma^k (r_{k,\text{to-go}} - r_{\text{base}}(s_k)) \right]$$

$$\theta' = \theta + \alpha \nabla U(\theta)$$

$$\begin{aligned} \underset{\theta'}{\text{maximize}} \quad & U(\theta') \approx U(\theta) + \nabla U(\theta)^\top (\theta' - \theta) \\ \text{subject to} \quad & g(\theta, \theta') \leq \epsilon \end{aligned}$$



$$g(\theta, \theta') = \|\theta - \theta'\|_2^2 = \frac{1}{2}(\theta' - \theta)^\top (\theta' - \theta)$$

Restricted Gradient Update

$$\nabla U(\theta) = E_\tau \left[\sum_{k=0}^{d} \nabla_\theta \log \pi_\theta(a_k \mid s_k) \gamma^k (r_{k,\text{to-go}} - r_{\text{base}}(s_k)) \right]$$

$$\theta' = \theta + \alpha \nabla U(\theta)$$

$$\begin{aligned} \underset{\theta'}{\text{maximize}} \quad & U(\theta') \approx U(\theta) + \nabla U(\theta)^\top (\theta' - \theta) \\ \text{subject to} \quad & g(\theta, \theta') \leq \epsilon \end{aligned}$$

$$g(\theta, \theta') = \|\theta - \theta'\|_2^2 = \frac{1}{2}(\theta' - \theta)^\top (\theta' - \theta)$$

$$\theta' = \theta + \mathbf{u}\sqrt{\frac{2\epsilon}{\mathbf{u}^\top \mathbf{u}}} = \theta + \sqrt{2\epsilon}\frac{\mathbf{u}}{\|\mathbf{u}\|}$$

Restricted Gradient Update

$$\nabla U(\theta) = E_\tau \left[\sum_{k=0}^{d} \nabla_\theta \log \pi_\theta(a_k \mid s_k) \gamma^k (r_{k,\text{to-go}} - r_{\text{base}}(s_k)) \right]$$

$$\theta' = \theta + \alpha \nabla U(\theta)$$

$$\begin{aligned} \underset{\theta'}{\text{maximize}} \quad & U(\theta') \approx U(\theta) + \nabla U(\theta)^\top (\theta' - \theta) \\ \text{subject to} \quad & g(\theta, \theta') \leq \epsilon \end{aligned}$$

$$g(\theta, \theta') = \|\theta - \theta'\|_2^2 = \frac{1}{2}(\theta' - \theta)^\top (\theta' - \theta)$$

$$\theta' = \theta + \mathbf{u}\sqrt{\frac{2\epsilon}{\mathbf{u}^\top \mathbf{u}}} = \theta + \sqrt{2\epsilon}\frac{\mathbf{u}}{\|\mathbf{u}\|}$$

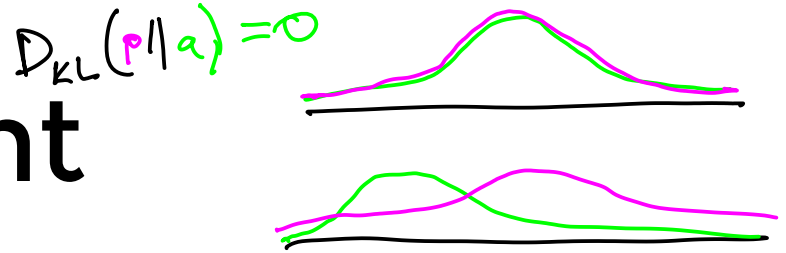
$$\mathbf{u} = \nabla U(\theta)$$

Natural Gradient

Natural Gradient

$$g(\theta, \theta') = D_{\mathrm{KL}}\left(p(\cdot \mid \theta) \,\middle|\middle|\, p(\cdot \mid \theta')\right) \leq \epsilon$$

Natural Gradient



$$g(\theta, \theta') = D_{\mathrm{KL}}\left(p(\cdot \mid \theta) \,\middle|\middle|\, p(\cdot \mid \theta')\right) \leq \epsilon$$

$$D_{\mathrm{KL}}(p \,||\, q) = \int p(x) \log \frac{p(x)}{q(x)} \, \mathrm{d}x = -\int p(x) \log \frac{q(x)}{p(x)} \, \mathrm{d}x$$

Natural Gradient

$$g(\theta, \theta') = D_{\mathrm{KL}}\left(p(\cdot \mid \theta) \,\middle|\middle|\, p(\cdot \mid \theta')\right) \leq \epsilon$$

$$D_{\mathrm{KL}}(p \,||\, q) = \int p(x) \log \frac{p(x)}{q(x)} \, \mathrm{d}x = -\int p(x) \log \frac{q(x)}{p(x)} \, \mathrm{d}x$$

$$g(\theta, \theta') = \frac{1}{2}(\theta' - \theta)^{\top} \mathbf{F}_{\theta} (\theta' - \theta) \leq \epsilon$$

Natural Gradient

$$g(\theta, \theta') = D_{\mathrm{KL}}\left(p(\cdot \mid \theta) \,\middle|\middle|\, p(\cdot \mid \theta')\right) \leq \epsilon$$

$$D_{\mathrm{KL}}(p \,||\, q) = \int p(x) \log \frac{p(x)}{q(x)} \, \mathrm{d}x = -\int p(x) \log \frac{q(x)}{p(x)} \, \mathrm{d}x$$

$$g(\theta, \theta') = \frac{1}{2}(\theta' - \theta)^\top \mathbf{F}_\theta (\theta' - \theta) \leq \epsilon$$

↳ Fischer Information Matrix

Natural Gradient

$$g(\theta, \theta') = D_{\mathrm{KL}}\left(p(\cdot \mid \theta) \,\middle|\middle|\, p(\cdot \mid \theta')\right) \leq \epsilon$$

$$D_{\mathrm{KL}}(p \,||\, q) = \int p(x) \log \frac{p(x)}{q(x)} \, \mathrm{d}x = -\int p(x) \log \frac{q(x)}{p(x)} \, \mathrm{d}x$$

$$g(\theta, \theta') = \frac{1}{2}(\theta' - \theta)^\top \mathbf{F}_\theta (\theta' - \theta) \leq \epsilon$$

↳ Fischer Information Matrix

$$\begin{aligned} \mathbf{F}_\theta &= \int p(\tau \mid \theta) \nabla \log p(\tau \mid \theta) \nabla \log p(\tau \mid \theta)^\top \, \mathrm{d}\tau \\ &= \mathbb{E}_\tau \left[\nabla \log p(\tau \mid \theta) \nabla \log p(\tau \mid \theta)^\top \right] \end{aligned}$$

Natural Gradient

$$g(\theta, \theta') = D_{\mathrm{KL}}\left(p(\cdot \mid \theta) \,||\, p(\cdot \mid \theta')\right) \leq \epsilon$$

$$D_{\mathrm{KL}}(p \,||\, q) = \int p(x) \log \frac{p(x)}{q(x)} \, \mathrm{d}x = -\int p(x) \log \frac{q(x)}{p(x)} \, \mathrm{d}x$$

$$g(\theta, \theta') = \frac{1}{2}(\theta' - \theta)^\top \mathbf{F}_\theta (\theta' - \theta) \leq \epsilon$$

↳ Fischer Information Matrix

$$\mathbf{F}_\theta = \int p(\tau \mid \theta) \nabla \log p(\tau \mid \theta) \nabla \log p(\tau \mid \theta)^\top \, \mathrm{d}\tau$$
$$= \mathbb{E}_\tau \left[\nabla \log p(\tau \mid \theta) \nabla \log p(\tau \mid \theta)^\top \right]$$

$$\theta' = \theta + \mathbf{u} \sqrt{\frac{2\epsilon}{\nabla U(\theta)^\top \mathbf{u}}}$$

Natural Gradient

$$g(\theta, \theta') = D_{\mathrm{KL}}\left(p(\cdot \mid \theta) \,||\, p(\cdot \mid \theta')\right) \leq \epsilon$$

$$D_{\mathrm{KL}}(p \,||\, q) = \int p(x) \log \frac{p(x)}{q(x)} \, \mathrm{d}x = -\int p(x) \log \frac{q(x)}{p(x)} \, \mathrm{d}x$$

$$g(\theta, \theta') = \frac{1}{2}(\theta' - \theta)^\top \mathbf{F}_\theta (\theta' - \theta) \leq \epsilon$$

↳ Fischer Information Matrix

$$\mathbf{F}_\theta = \int p(\tau \mid \theta) \nabla \log p(\tau \mid \theta) \nabla \log p(\tau \mid \theta)^\top \, \mathrm{d}\tau$$
$$= \mathbb{E}_\tau \left[\nabla \log p(\tau \mid \theta) \nabla \log p(\tau \mid \theta)^\top \right]$$

$$\theta' = \theta + \mathbf{u} \sqrt{\frac{2\epsilon}{\nabla U(\theta)^\top \mathbf{u}}} \qquad \mathbf{u} = \mathbf{F}_\theta^{-1} \nabla U(\theta)$$

Natural Gradient

$$g(\theta, \theta') = D_{\mathrm{KL}}\left(p(\cdot \mid \theta) \,||\, p(\cdot \mid \theta')\right) \leq \epsilon$$

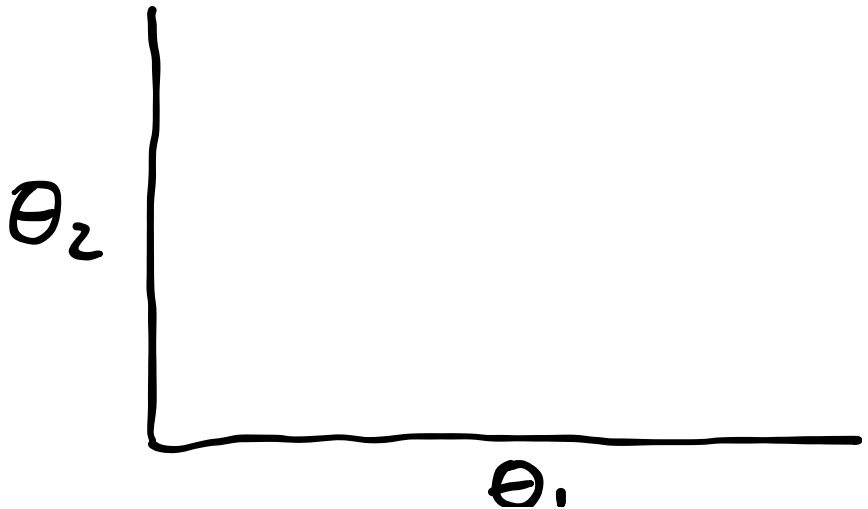
$$D_{\mathrm{KL}}(p \,||\, q) = \int p(x) \log \frac{p(x)}{q(x)} \, \mathrm{d}x = -\int p(x) \log \frac{q(x)}{p(x)} \, \mathrm{d}x$$

$$g(\theta, \theta') = \frac{1}{2}(\theta' - \theta)^\top \mathbf{F}_\theta (\theta' - \theta) \leq \epsilon$$

↳ Fischer Information Matrix

$$\mathbf{F}_\theta = \int p(\tau \mid \theta) \nabla \log p(\tau \mid \theta) \nabla \log p(\tau \mid \theta)^\top \, \mathrm{d}\tau$$
$$= \mathbb{E}_\tau \left[\nabla \log p(\tau \mid \theta) \nabla \log p(\tau \mid \theta)^\top \right]$$

$$\theta' = \theta + \mathbf{u} \sqrt{\frac{2\epsilon}{\nabla U(\theta)^\top \mathbf{u}}} \qquad \mathbf{u} = \mathbf{F}_\theta^{-1} \nabla U(\theta)$$



Natural Gradient

$$g(\theta, \theta') = D_{\mathrm{KL}}\left(p(\cdot \mid \theta) \,||\, p(\cdot \mid \theta')\right) \leq \epsilon$$

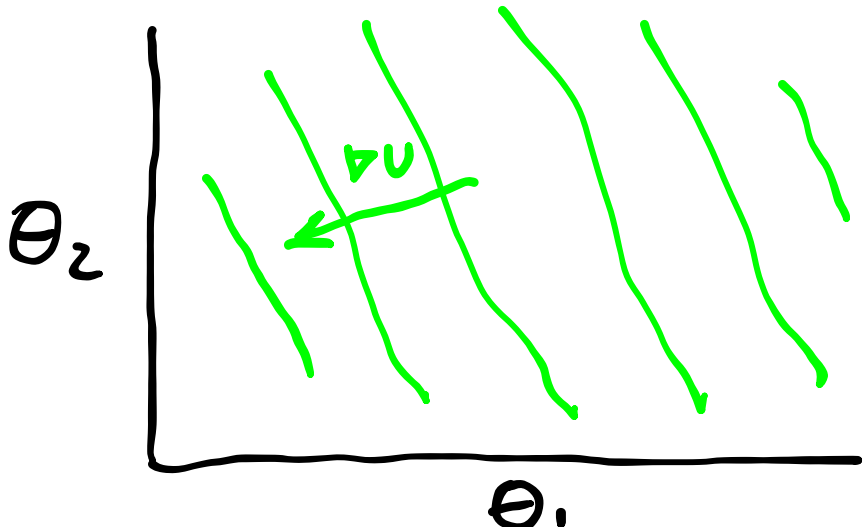
$$D_{\mathrm{KL}}(p \,||\, q) = \int p(x) \log \frac{p(x)}{q(x)} \, \mathrm{d}x = -\int p(x) \log \frac{q(x)}{p(x)} \, \mathrm{d}x$$

$$g(\theta, \theta') = \frac{1}{2}(\theta' - \theta)^\top \mathbf{F}_\theta (\theta' - \theta) \leq \epsilon$$

↳ Fischer Information Matrix

$$\mathbf{F}_\theta = \int p(\tau \mid \theta) \nabla \log p(\tau \mid \theta) \nabla \log p(\tau \mid \theta)^\top \mathrm{d}\tau$$
$$= \mathbb{E}_\tau \left[\nabla \log p(\tau \mid \theta) \nabla \log p(\tau \mid \theta)^\top \right]$$

$$\theta' = \theta + \mathbf{u} \sqrt{\frac{2\epsilon}{\nabla U(\theta)^\top \mathbf{u}}} \qquad \mathbf{u} = \mathbf{F}_\theta^{-1} \nabla U(\theta)$$



Natural Gradient

$$g(\theta, \theta') = D_{\mathrm{KL}}\left(p(\cdot \mid \theta) \,||\, p(\cdot \mid \theta')\right) \leq \epsilon$$

$$D_{\mathrm{KL}}(p \,||\, q) = \int p(x) \log \frac{p(x)}{q(x)} \, \mathrm{d}x = -\int p(x) \log \frac{q(x)}{p(x)} \, \mathrm{d}x$$

$$g(\theta, \theta') = \frac{1}{2}(\theta' - \theta)^\top \mathbf{F}_\theta (\theta' - \theta) \leq \epsilon$$

↳ Fischer Information Matrix

$$\mathbf{F}_\theta = \int p(\tau \mid \theta) \nabla \log p(\tau \mid \theta) \nabla \log p(\tau \mid \theta)^\top \mathrm{d}\tau$$
$$= \mathbb{E}_\tau \left[\nabla \log p(\tau \mid \theta) \nabla \log p(\tau \mid \theta)^\top \right]$$

$$\theta' = \theta + \mathbf{u} \sqrt{\frac{2\epsilon}{\nabla U(\theta)^\top \mathbf{u}}} \qquad \mathbf{u} = \mathbf{F}_\theta^{-1} \nabla U(\theta)$$

Natural Gradient

$$g(\theta, \theta') = D_{\mathrm{KL}}\left(p(\cdot \mid \theta) \,||\, p(\cdot \mid \theta')\right) \leq \epsilon$$

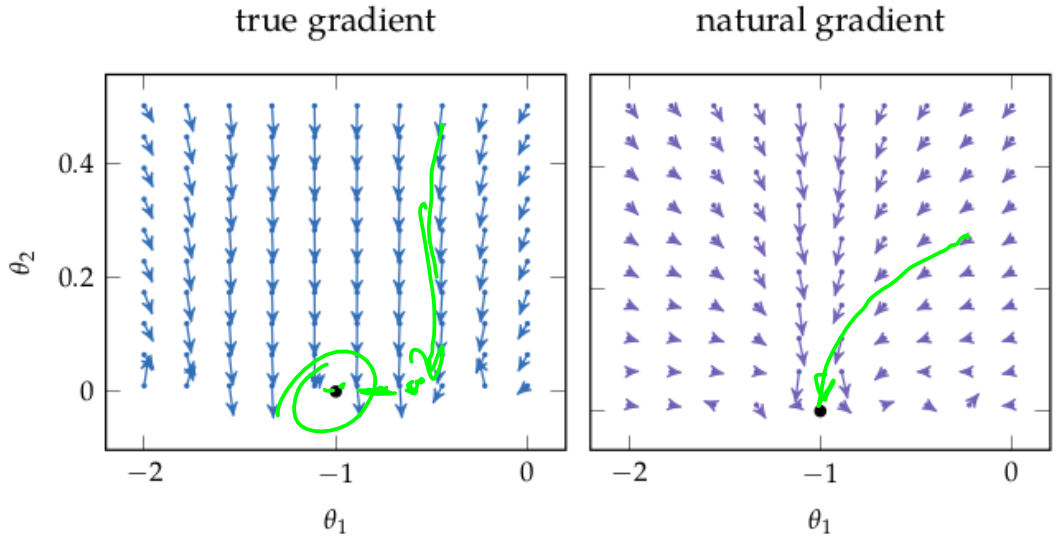
$$D_{\mathrm{KL}}(p \,||\, q) = \int p(x) \log \frac{p(x)}{q(x)} \, \mathrm{d}x = -\int p(x) \log \frac{q(x)}{p(x)} \, \mathrm{d}x$$

$$g(\theta, \theta') = \frac{1}{2}(\theta' - \theta)^\top \mathbf{F}_\theta (\theta' - \theta) \leq \epsilon$$

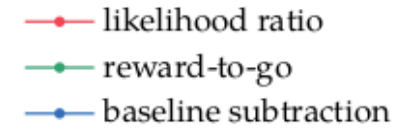
↳ Fischer Information Matrix

$$\mathbf{F}_\theta = \int p(\tau \mid \theta) \nabla \log p(\tau \mid \theta) \nabla \log p(\tau \mid \theta)^\top \, \mathrm{d}\tau$$
$$= \mathbb{E}_\tau \left[\nabla \log p(\tau \mid \theta) \nabla \log p(\tau \mid \theta)^\top \right]$$

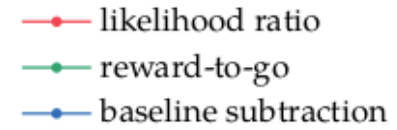
$$\theta' = \theta + \mathbf{u} \sqrt{\frac{2\epsilon}{\nabla U(\theta)^\top \mathbf{u}}} \qquad \mathbf{u} = \mathbf{F}_\theta^{-1} \nabla U(\theta)$$



TRPO and PPO



TRPO and PPO



TRPO = Trust Region Policy Optimization

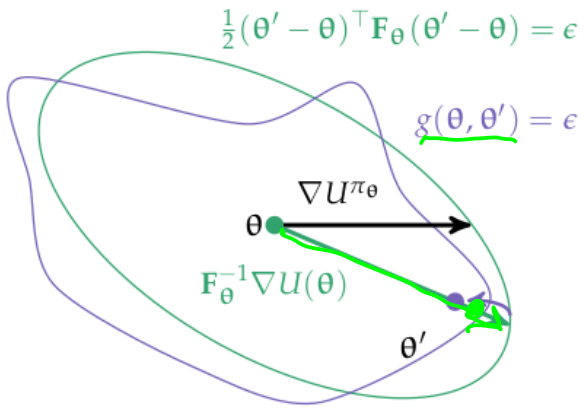
(Natural gradient + line search)

TRPO and PPO

likelihood ratio
reward-to-go
baseline subtraction

TRPO = Trust Region Policy Optimization

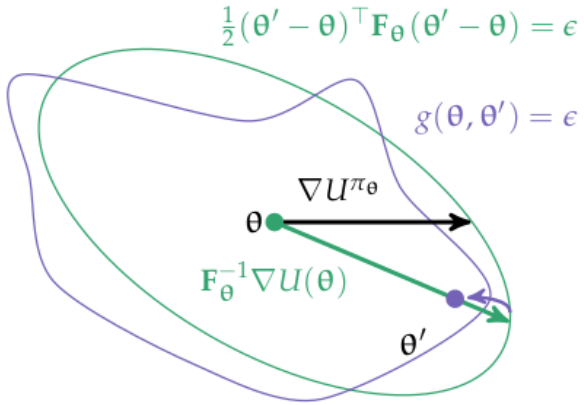
(Natural gradient + line search)

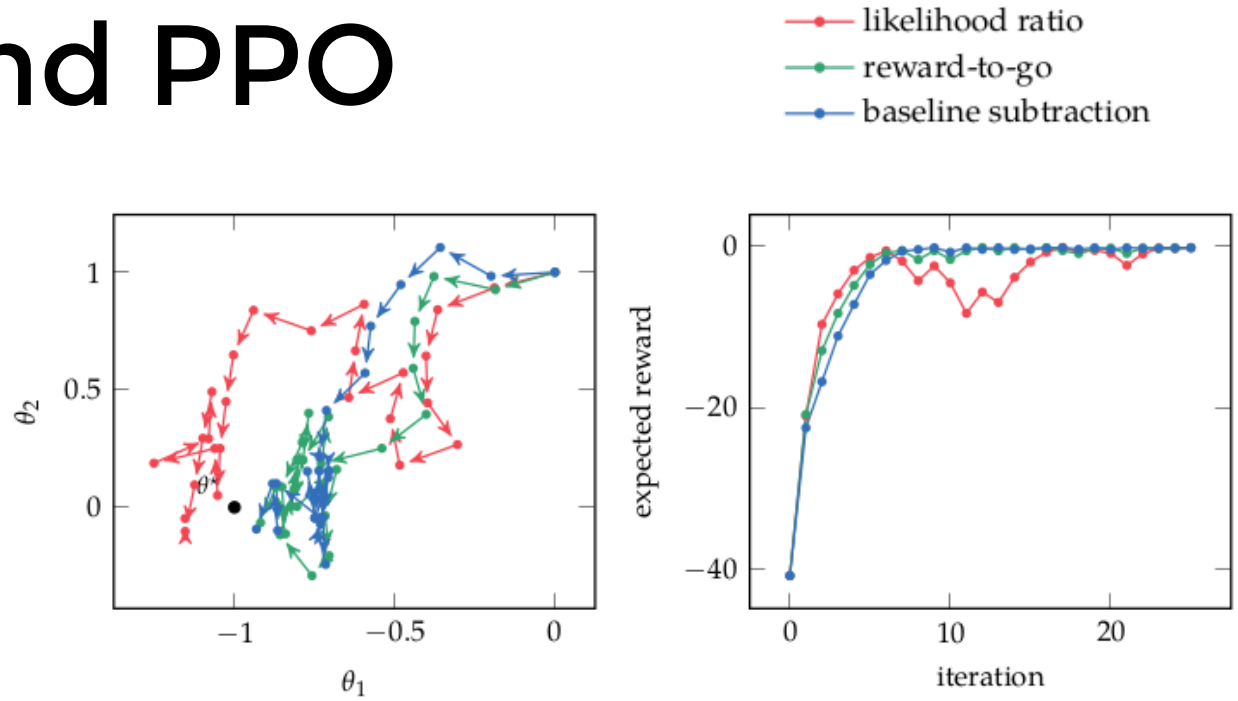


TRPO and PPO

TRPO = Trust Region Policy Optimization

(Natural gradient + line search)

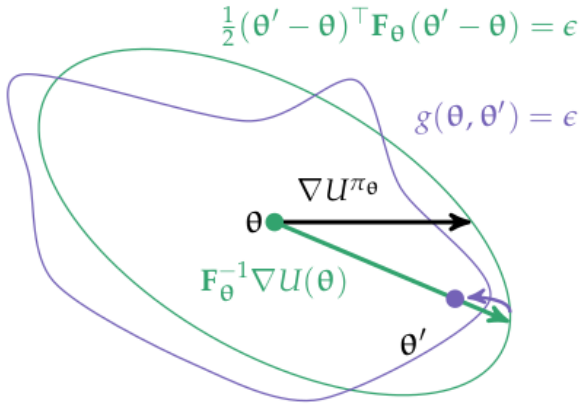


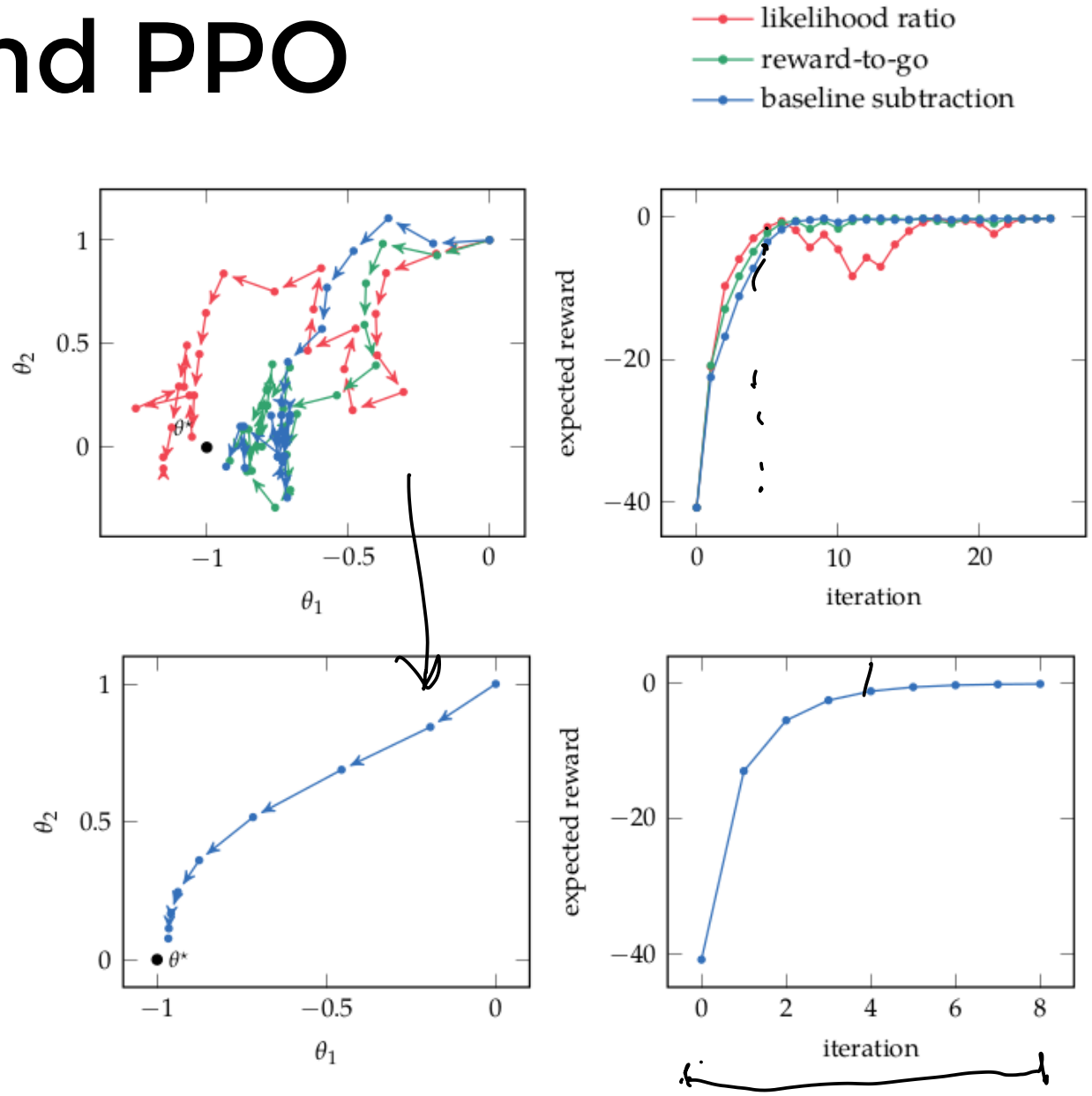


TRPO and PPO

TRPO = Trust Region Policy Optimization

(Natural gradient + line search)

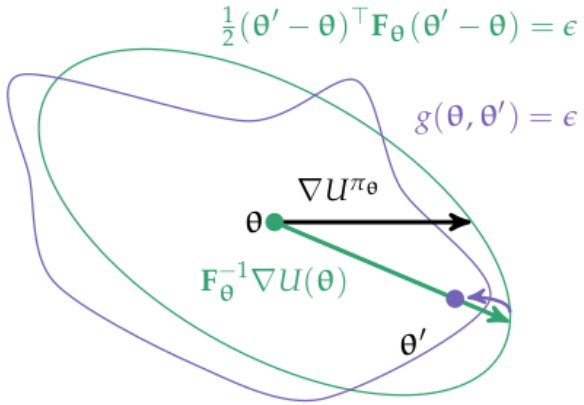




TRPO and PPO

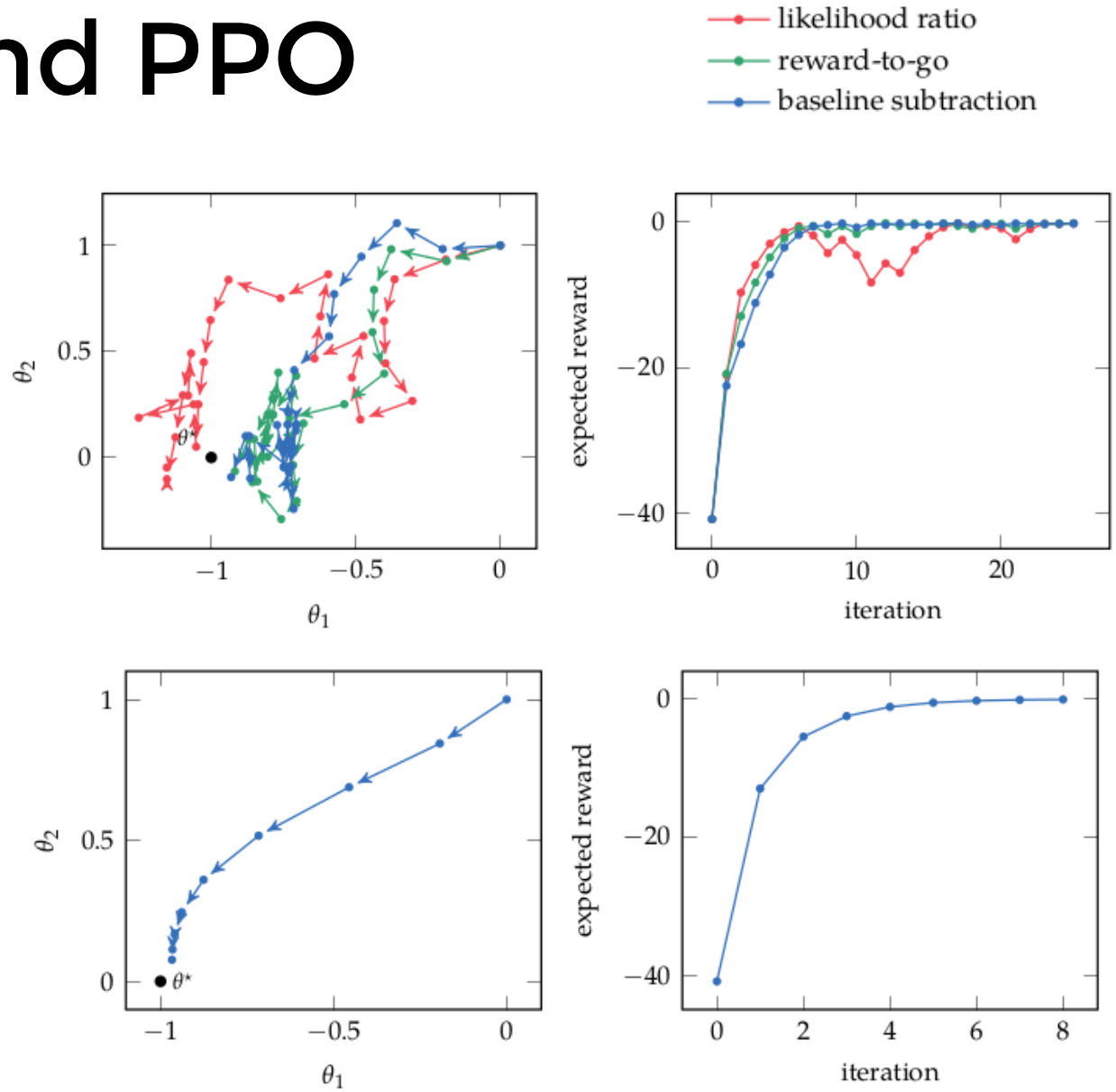
TRPO = Trust Region Policy Optimization

(Natural gradient + line search)



PPO = Proximal Policy Optimization

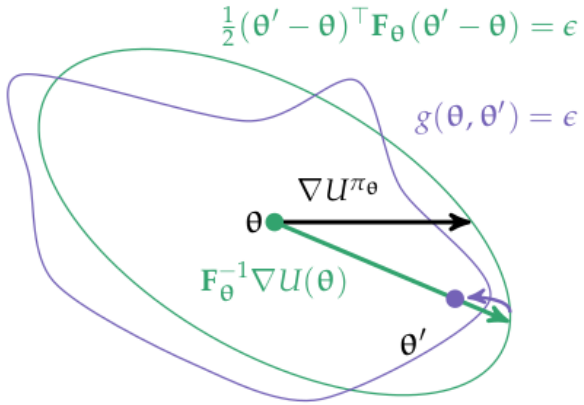
(Use clamped surrogate objective to remove the need for line search)

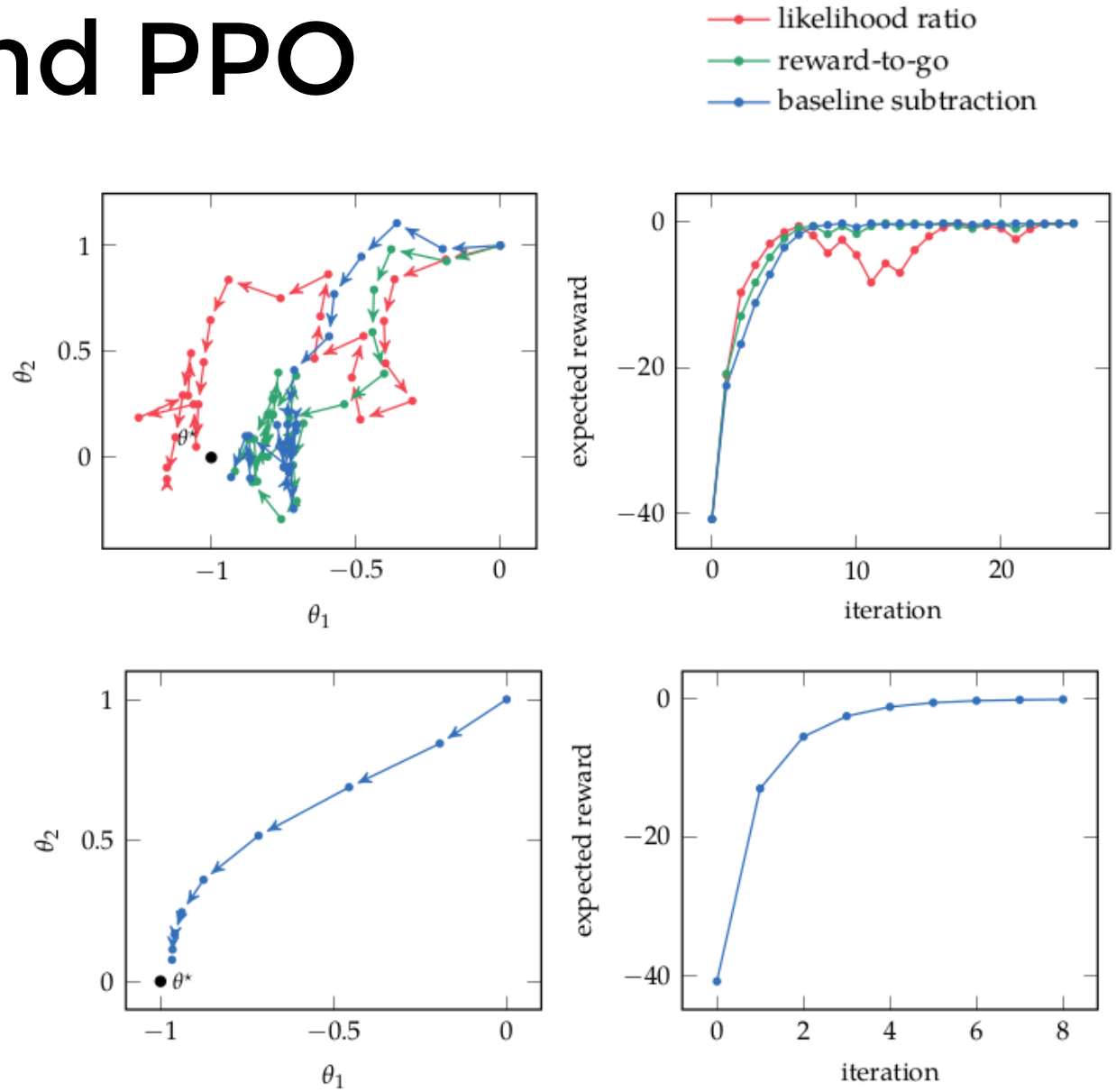


TRPO and PPO

TRPO = Trust Region Policy Optimization

(Natural gradient + line search)





PPO = Proximal Policy Optimization

(Use clamped surrogate objective to remove the need for line search)

https://openai.com/blog/openai-baselines-ppo/

Recap

DQN

Experience Buffer

Freezing Targets

Policy Grad

Natural Gradient

TRPO

PPO