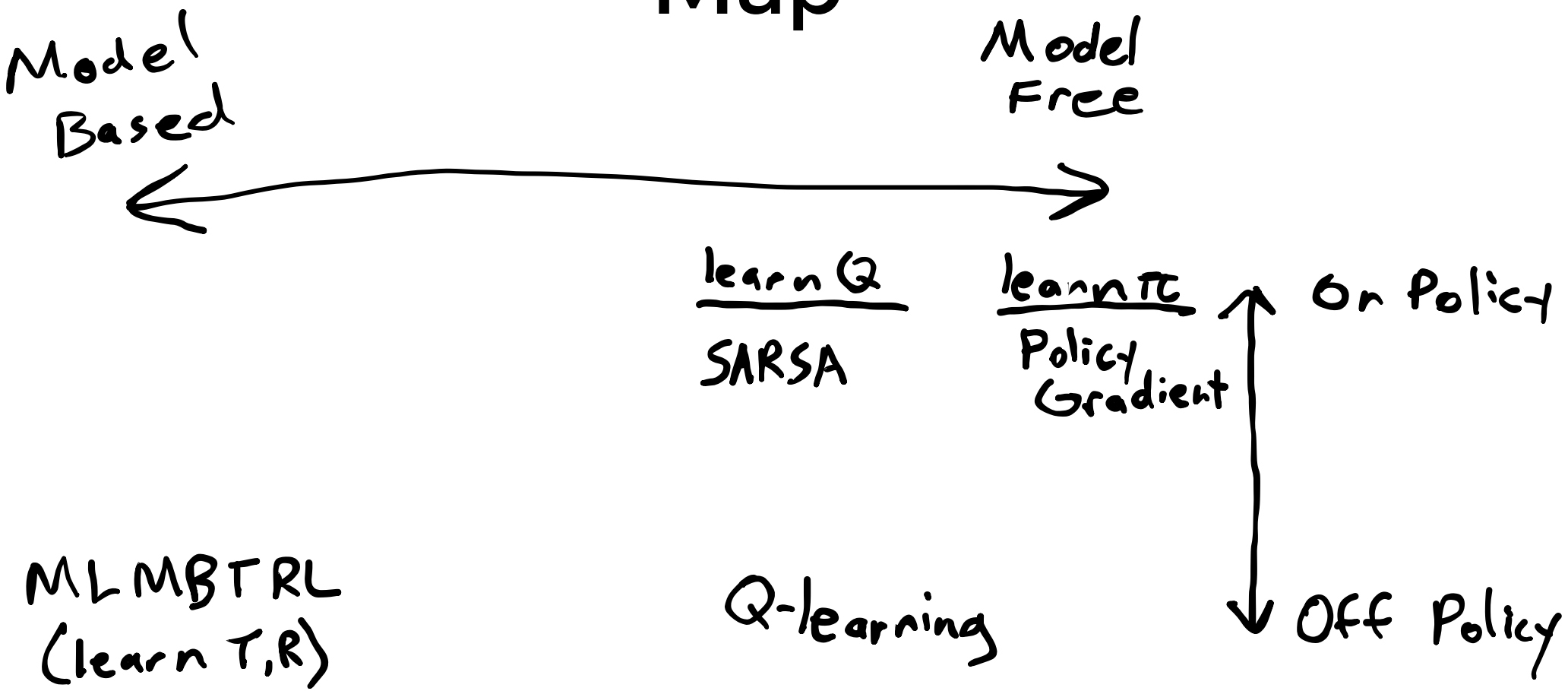
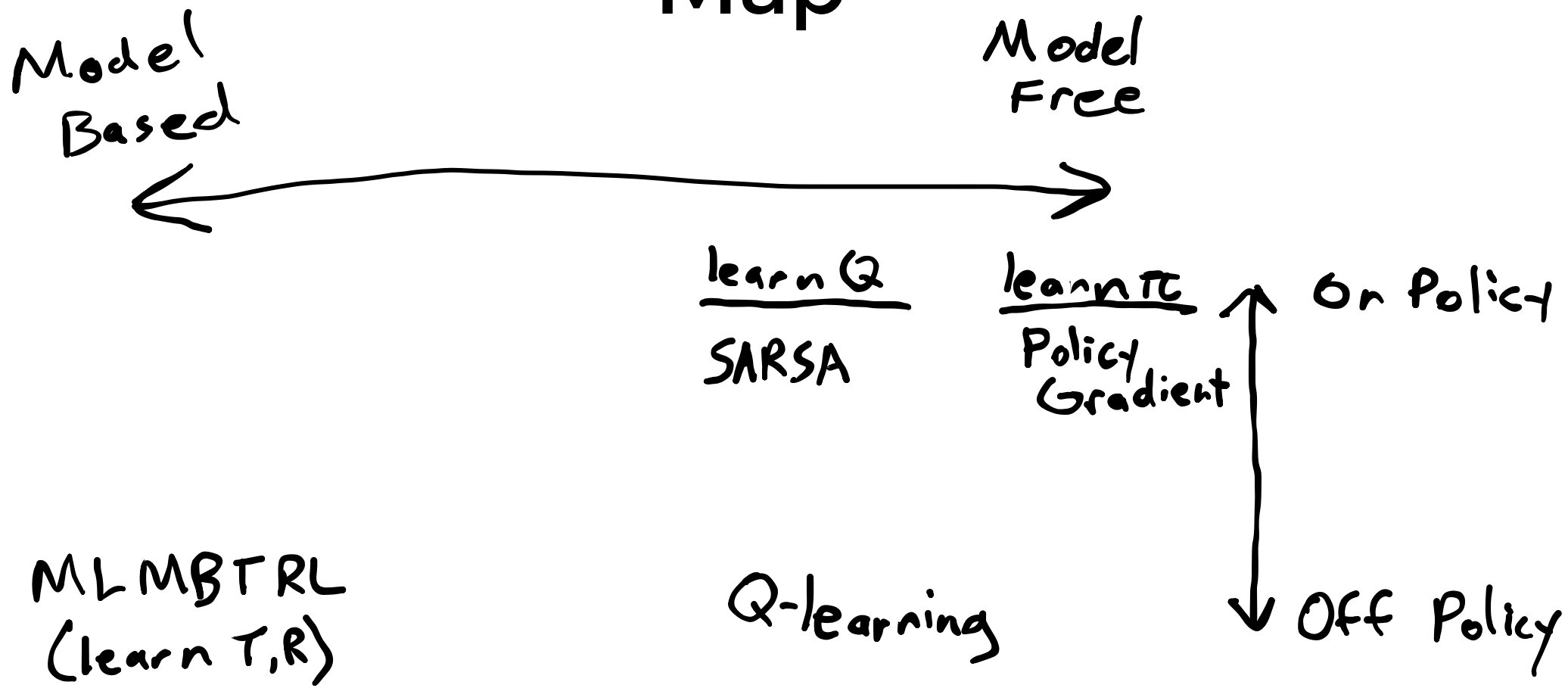
Map

Map


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

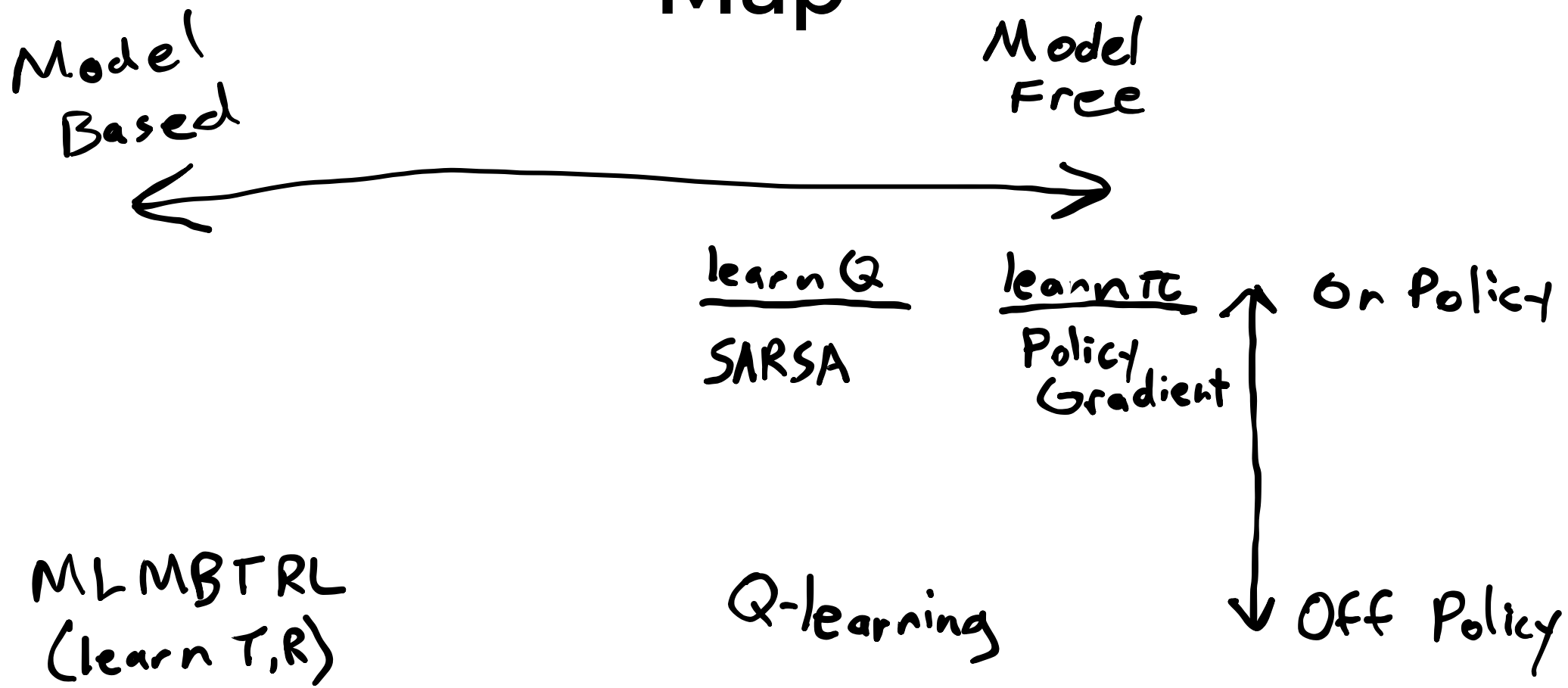
Map



Challenges:

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

Map

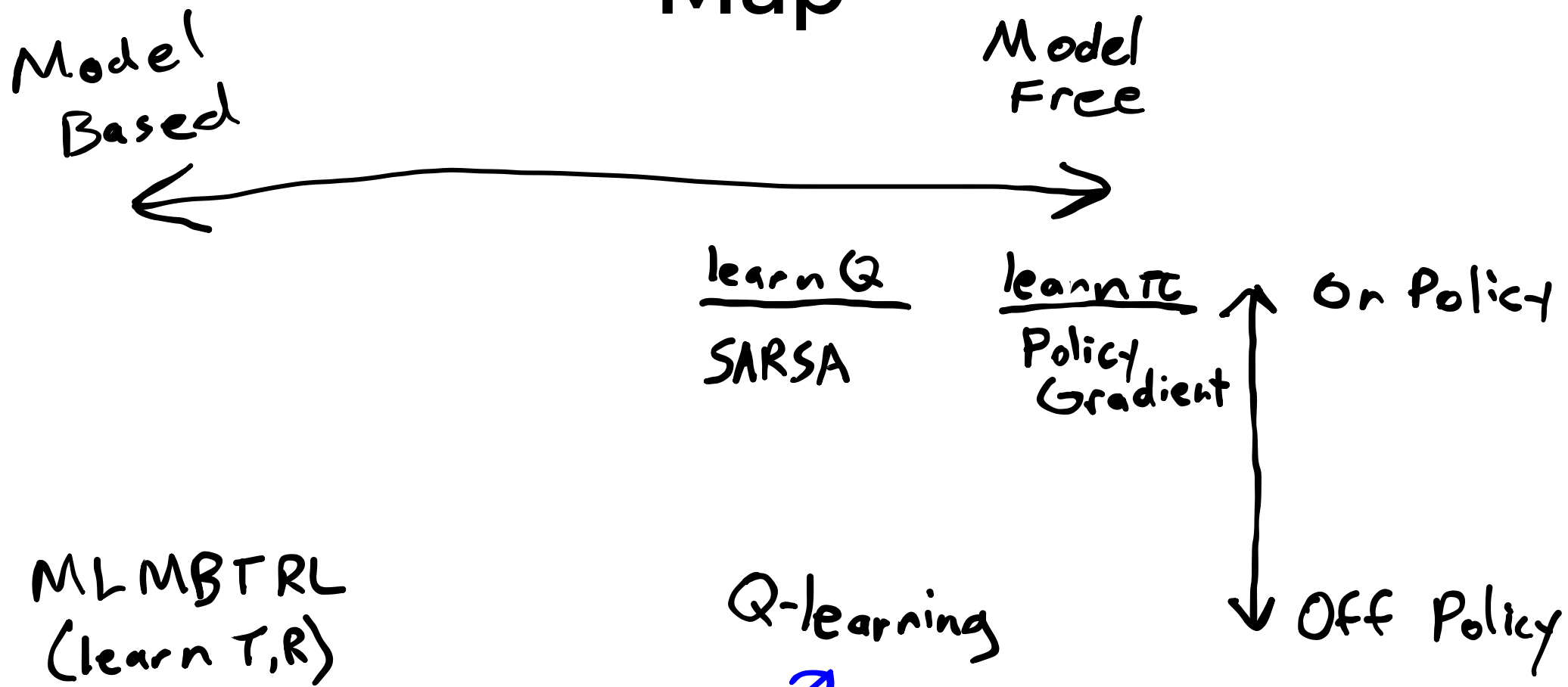


Challenges:

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

Last Time: Neural Networks

Map

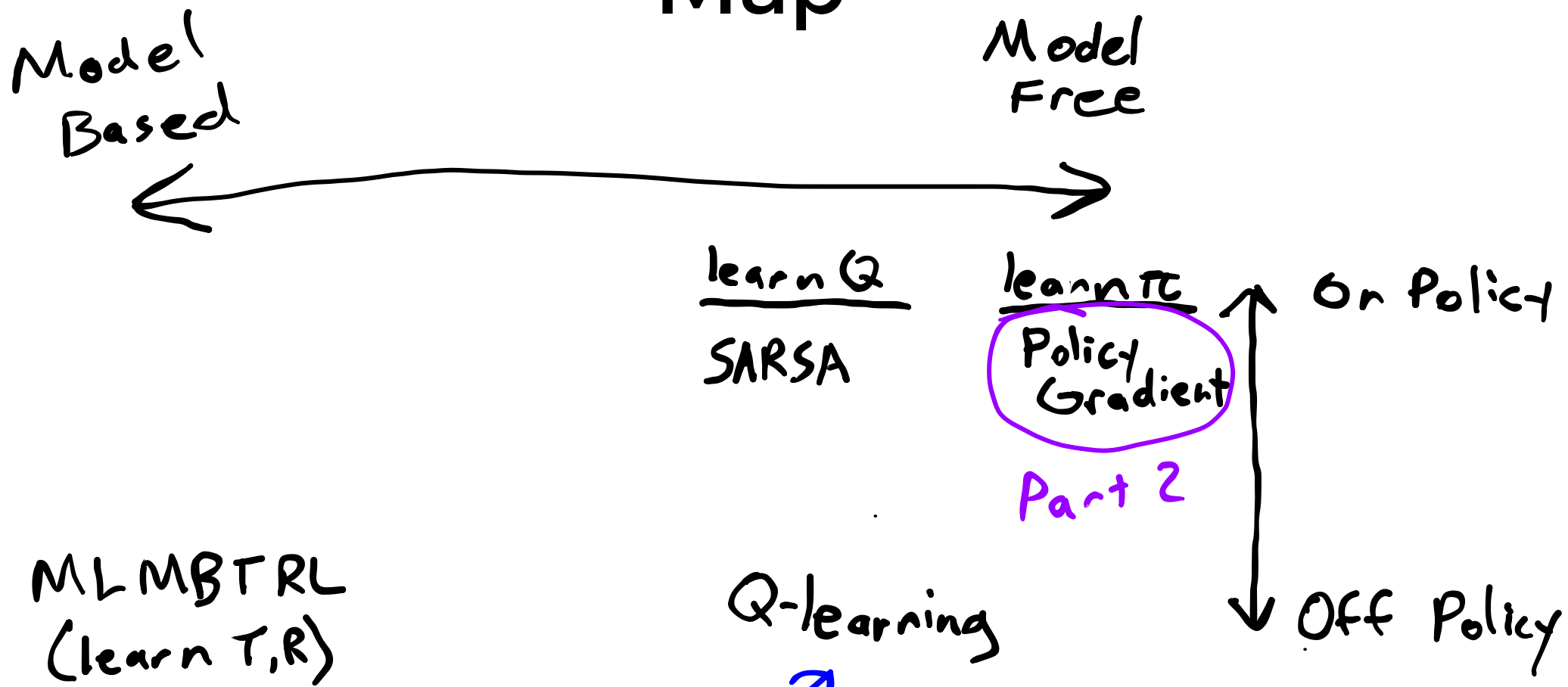


Challenges:

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?

(s, a, r, s') experience tuples

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

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

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

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

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

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

$$s \leftarrow s'$$

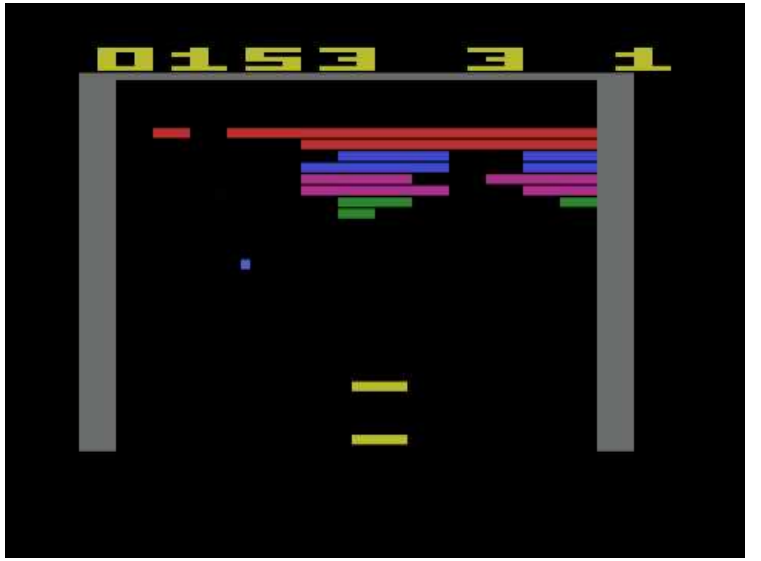
DQN: The Atari Benchmark

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

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

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

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

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

$$s \leftarrow s'$$

target network

Problems:

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

} data buffer / experience replay

} periodically freeze target

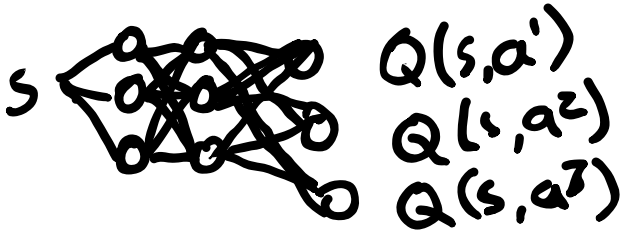
DQN

DQN

Q Network Structure:

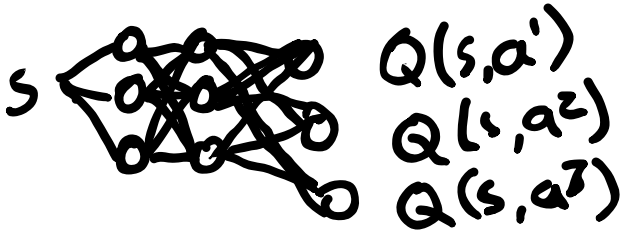
DQN

Q Network Structure:



DQN

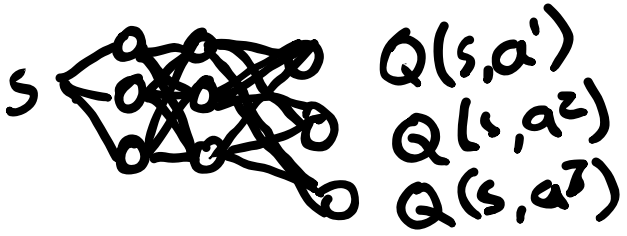
Q Network Structure:



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

DQN

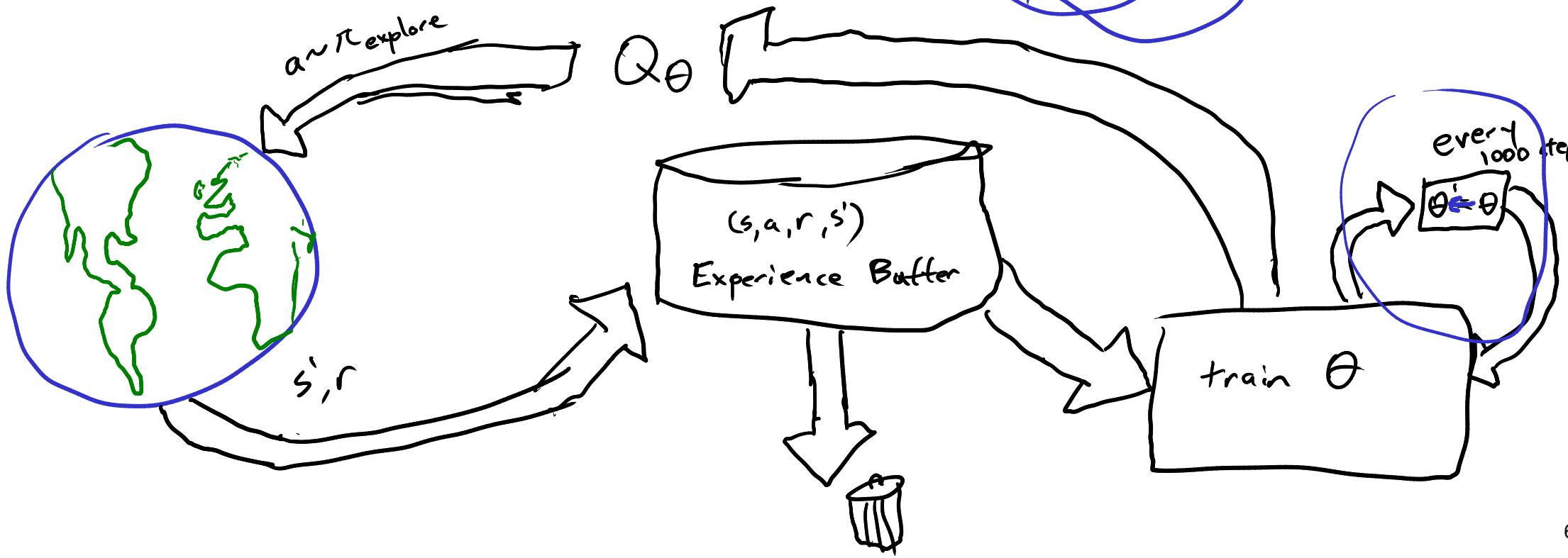
Q Network Structure:



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

Loss:

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

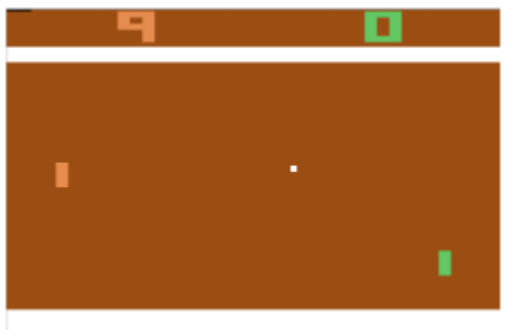


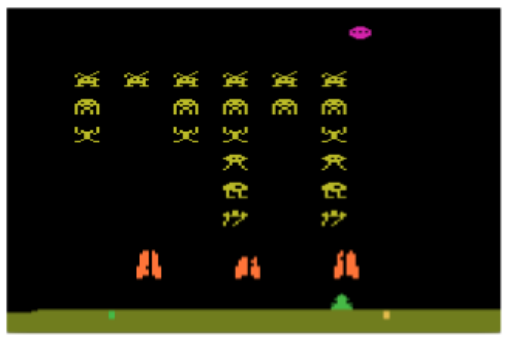
Playing Atari with Deep Reinforcement Learning

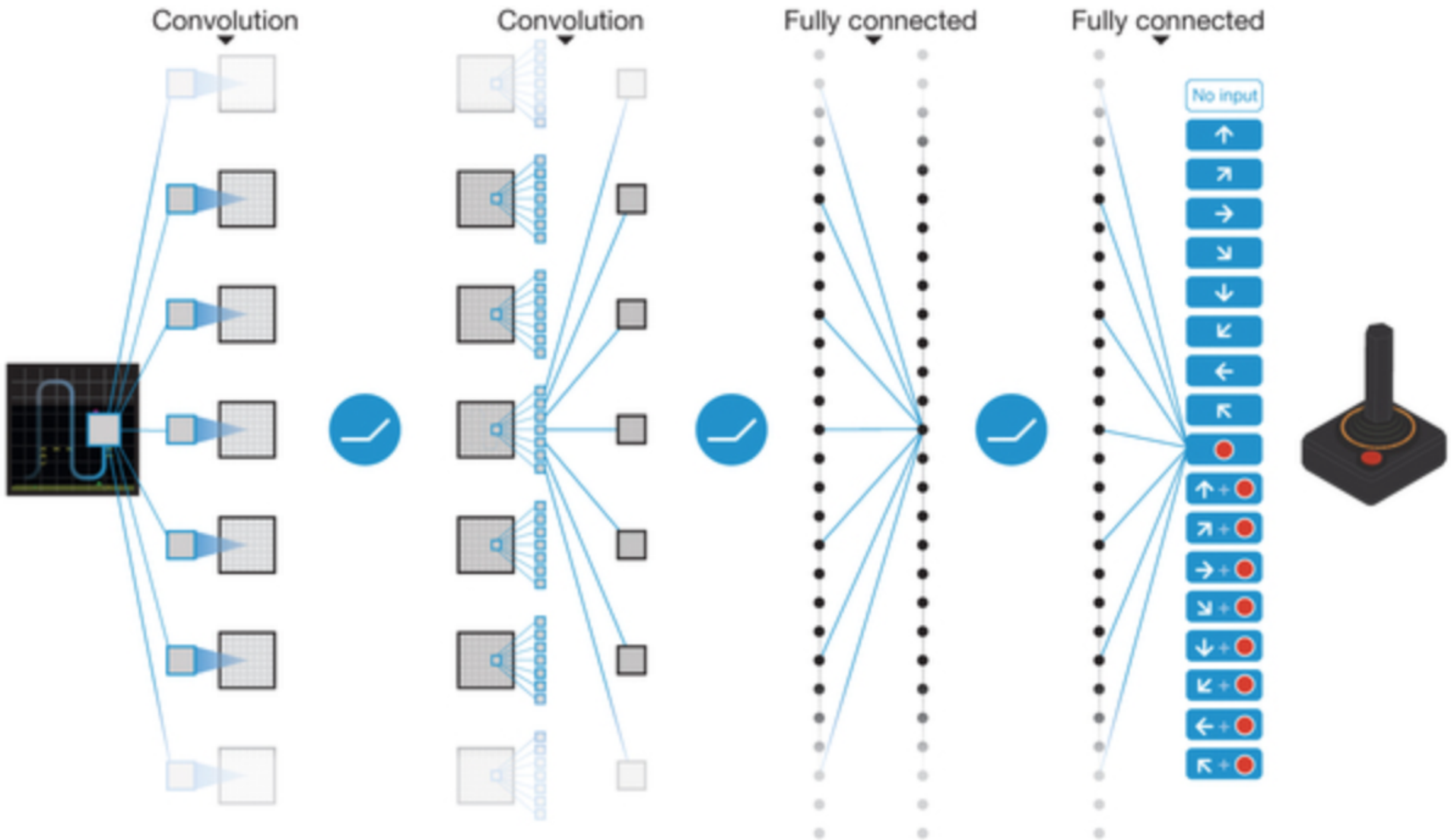
Volodymyr Mnih **Koray Kavukcuoglu** **David Silver** **Alex Graves** **Ioannis Antonoglou**

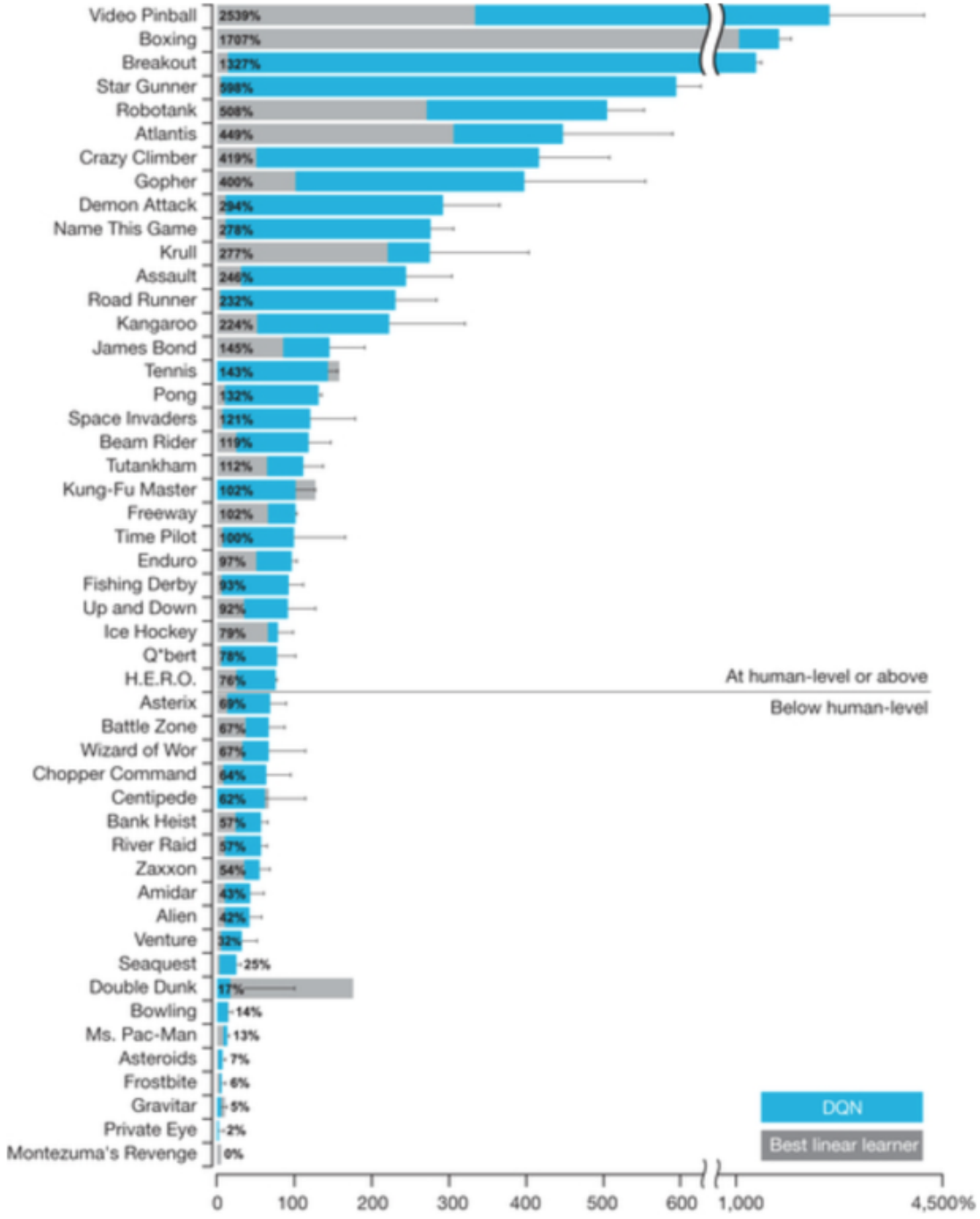
Daan Wierstra **Martin Riedmiller**

DeepMind Technologies

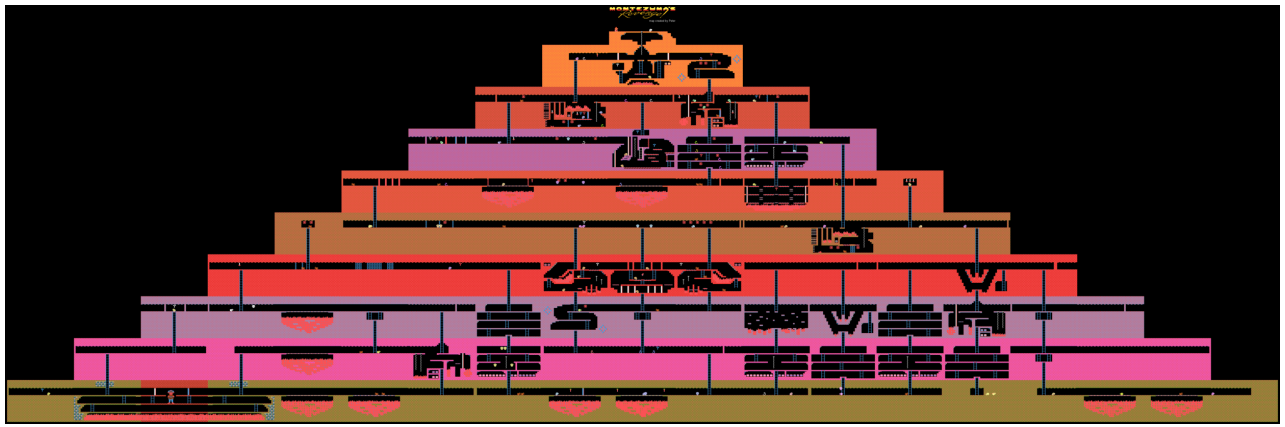






Convolution
Convolution
Fully connected
Fully connected
No input



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



Rainbow

Rainbow

- Double Q Learning

Rainbow

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