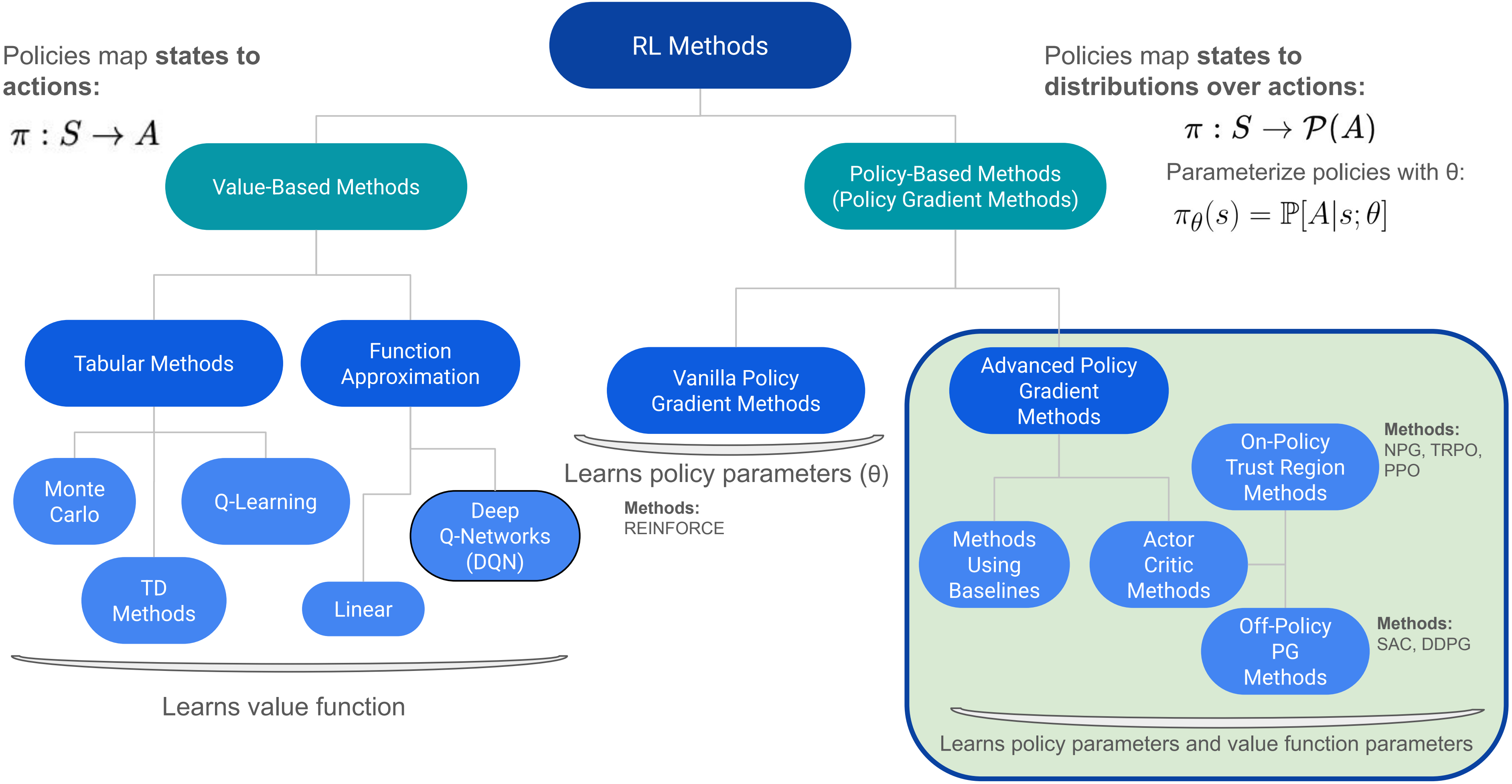
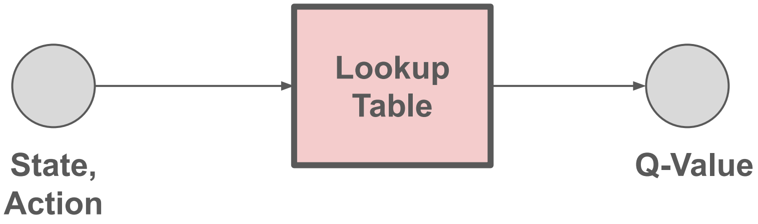
Deep Q-Networks

Nolan Bogumill


RL Methods
Policies map **states to actions:**
$\pi : S \to A$
Policies map **states to distributions over actions:**
$\pi : S \to \mathcal{P}(A)$
Parameterize policies with θ:
$\pi_\theta(s) = \mathbb{P}[A|s;\theta]$
Value-Based Methods
Policy-Based Methods (Policy Gradient Methods)
Tabular Methods
Function Approximation
Monte Carlo
Q-Learning
TD Methods
Linear
Deep Q-Networks (DQN)
Learns value function
Vanilla Policy Gradient Methods
Learns policy parameters (θ)
Methods: REINFORCE
Advanced Policy Gradient Methods
On-Policy Trust Region Methods
Methods: NPG, TRPO, PPO
Methods Using Baselines
Actor Critic Methods
Off-Policy PG Methods
Methods: SAC, DDPG
Learns policy parameters and value function parameters

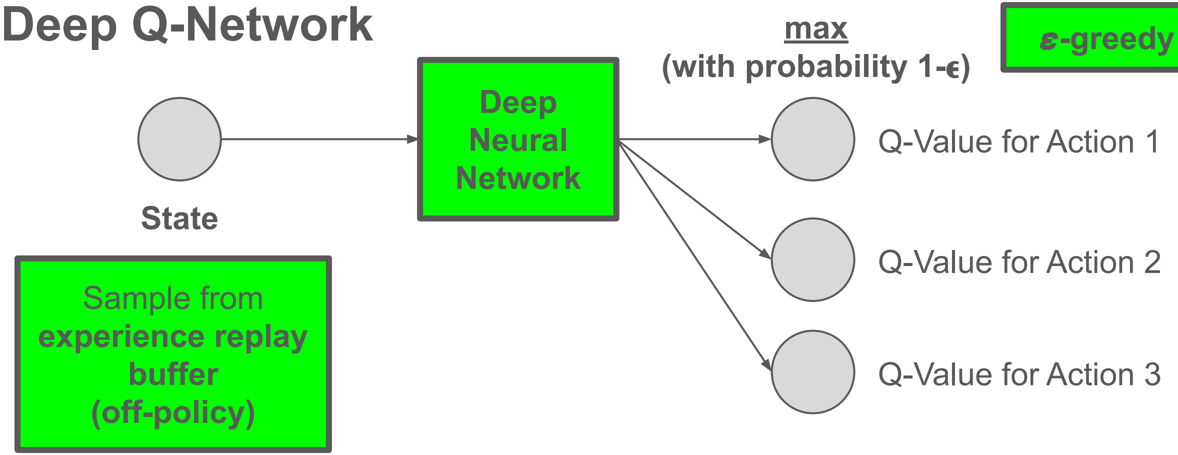
Q-Learning



Impractical to compute Q-values in high-dimensional action spaces

Solution: approximate Q-values instead!

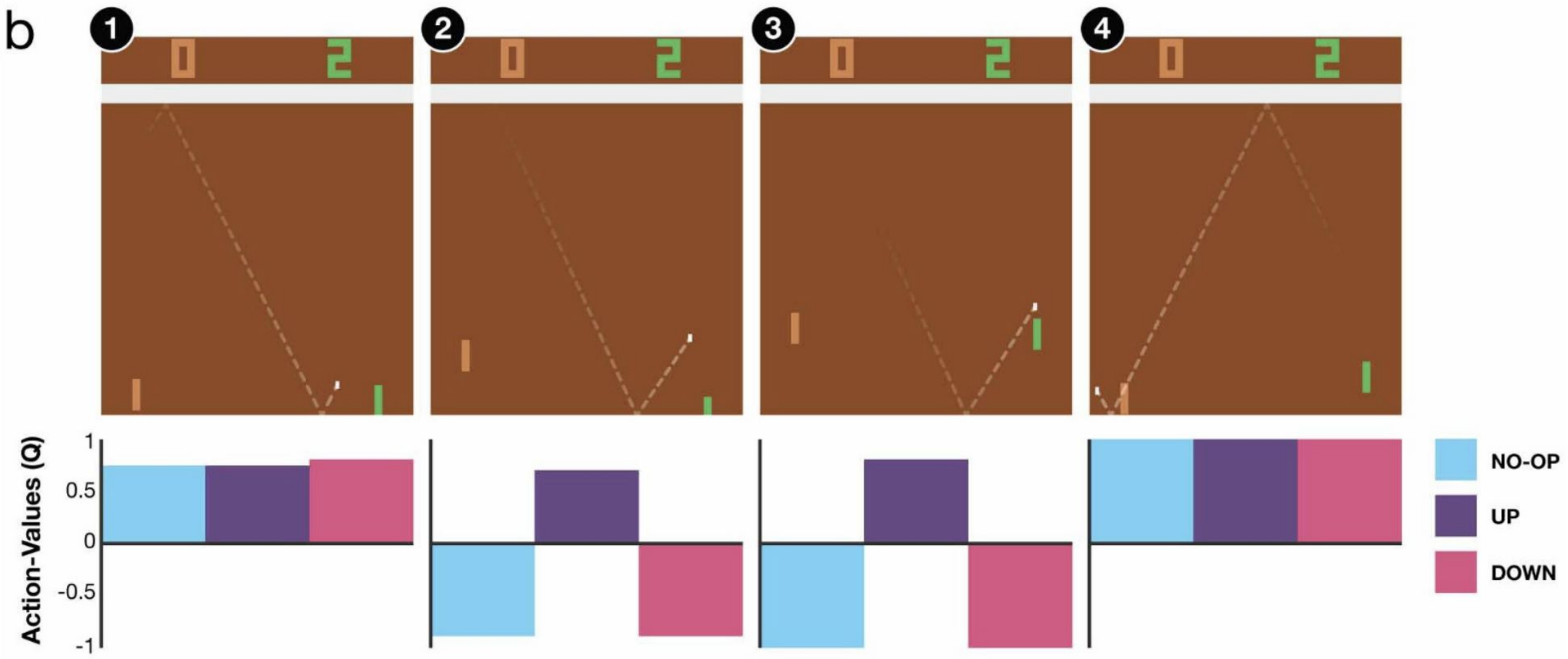
Deep Q-Network



***ε*-greedy** exploration

3 key concepts:
- DNNs
- Experience replay buffer
- Epsilon greedy exploration

Q-Values in Pong

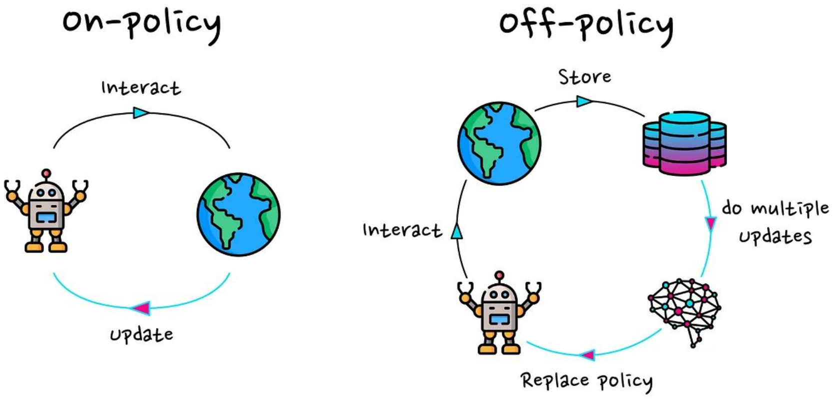


Off-Policy Methods

On-Policy vs. Off-Policy

Are we learning from data collected by the *current* policy, or data from any policy?

- Off-policy updates learn from any transition
    - More sample efficient
    - Implemented with **experience replay buffer**



Epsilon Greedy

- **Problem:** Exploration vs. exploitation trade-off

DQNs:

- With (1-ϵ) probability, select action with max Q-value:

$$a_t = \max_a Q^*(\phi(s_t), a; \theta)$$

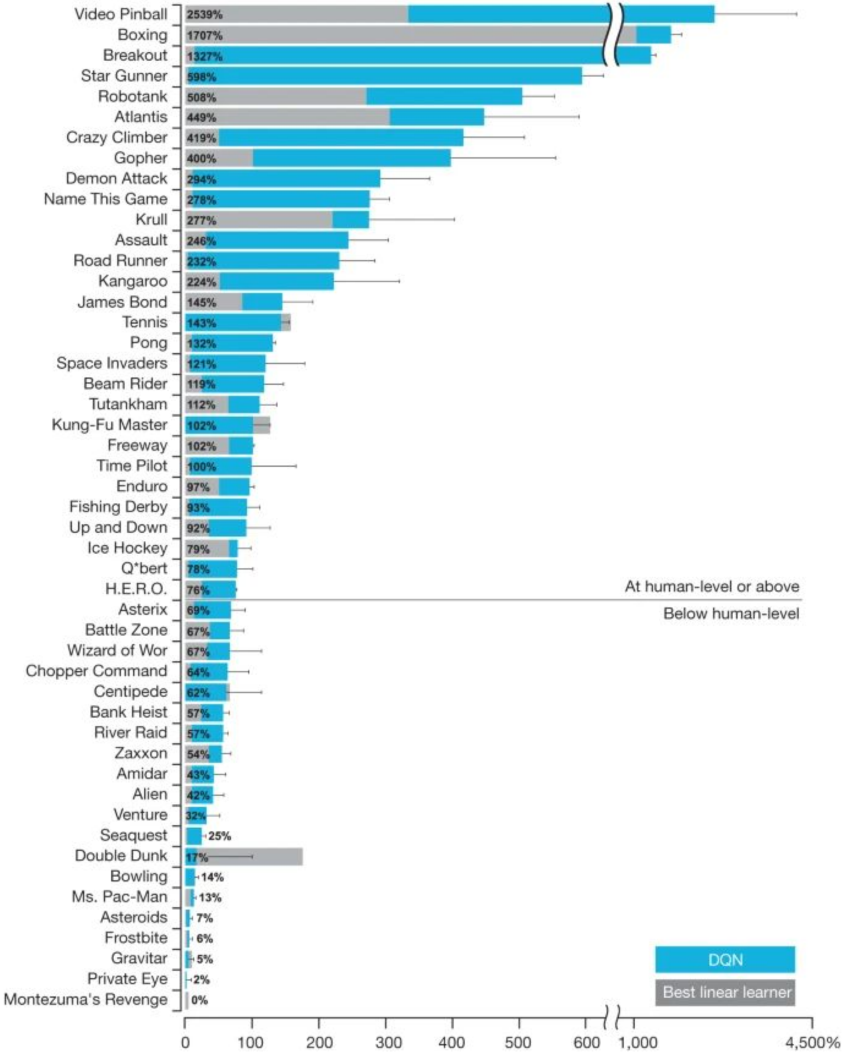
- Else sample random action

$$a_t$$

→ Sample random actions to explore action space for high-reward actions

Experiments

- DQN achieves more than 75% of the human score on 29/49 Atari games

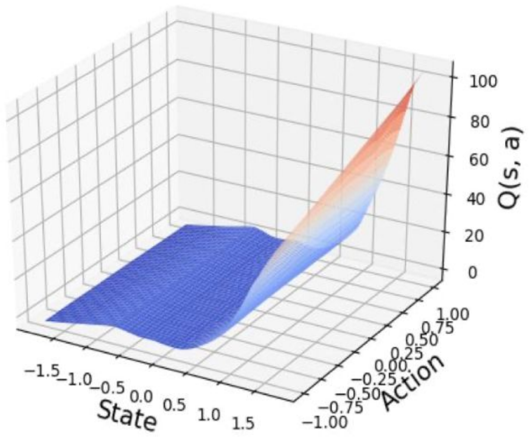


From DQN to Actor-Critic

- **2015 – A3C (Asynchronous Advantage Actor-Critic)** from the same team at DeepMind
  - DQN (Q-value approximation) + REINFORCE (direct policy optimization)

- **Key idea:** Combine **policy-based** and **value-based** methods
  - **Critic** takes states (or states+actions) and predicts values (or Q-values)
  - **Actor** takes states and predicts actions

Actor-Critic Methods



Two common reasons to use an actor-critic over DQNs:

1. **Learning high-dimensional continuous actions**
   - Actor directly represents actions
   - As opposed to learning values and searching for actions at test-time

2. **Reducing variance in a policy gradient**
   - Critic approximates a baseline function (usually advantage) that informs actor
   - Based to control variate methods from Monte Carlo literature

Hyperparameters

Hyperparameter	Value	Description
minibatch size	32	Number of training cases over which each stochastic gradient descent (SGD) update is computed.
replay memory size	1000000	SGD updates are sampled from this number of most recent frames.
agent history length	4	The number of most recent frames experienced by the agent that are given as input to the Q network.
target network update frequency	10000	The frequency (measured in the number of parameter updates) with which the target network is updated (this corresponds to the parameter *C* from Algorithm 1).
discount factor	0.99	Discount factor gamma used in the Q-learning update.
action repeat	4	Repeat each action selected by the agent this many times. Using a value of 4 results in the agent seeing only every 4th input frame.
update frequency	4	The number of actions selected by the agent between successive SGD updates. Using a value of 4 results in the agent selecting 4 actions between each pair of successive updates.
learning rate	0.00025	The learning rate used by RMSProp.
gradient momentum	0.95	Gradient momentum used by RMSProp.
squared gradient momentum	0.95	Squared gradient (denominator) momentum used by RMSProp.
min squared gradient	0.01	Constant added to the squared gradient in the denominator of the RMSProp update.
initial exploration	1	Initial value of ε in ε-greedy exploration.
final exploration	0.1	Final value of ε in ε-greedy exploration.
final exploration frame	1000000	The number of frames over which the initial value of ε is linearly annealed to its final value.
replay start size	50000	A uniform random policy is run for this number of frames before learning starts and the resulting experience is used to populate the replay memory.
no-op max	30	Maximum number of "do nothing" actions to be performed by the agent at the start of an episode.