JUNE 17, 2020

Offline Reinforcement Learning

Deep Learning Sessions Lisbon

Offline Reinforcement Learning
Main Objectives

Why Reinforcement Learning ?

Why Offline Reinforcement Learning?

What are the main challenges to Offline RL?

What are the most promising approaches in Offline RL?

What are the unresolved issues in Offline RL?

How can I use Offline RL?

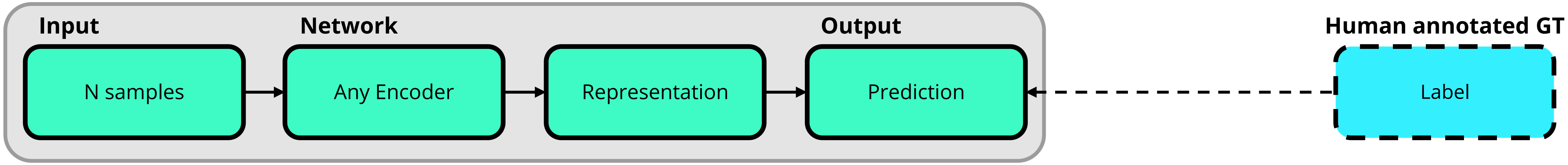
Reinforcement Learning vs **Supervised Learning**

Formalisms

Supervised Learning

Feed Forward, Recurrent, Convolutional Neural Network (CNN)

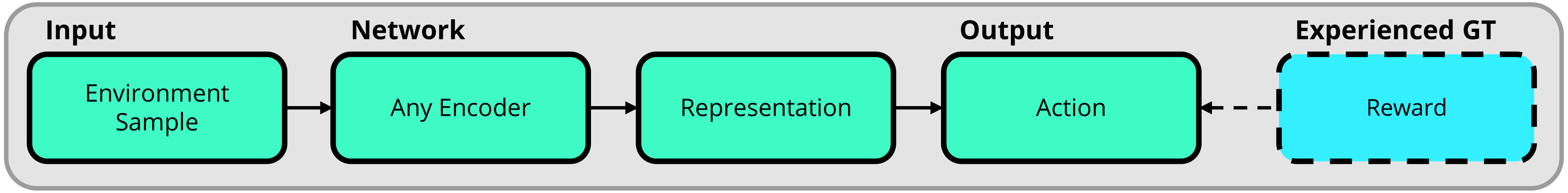
"Teach by example"



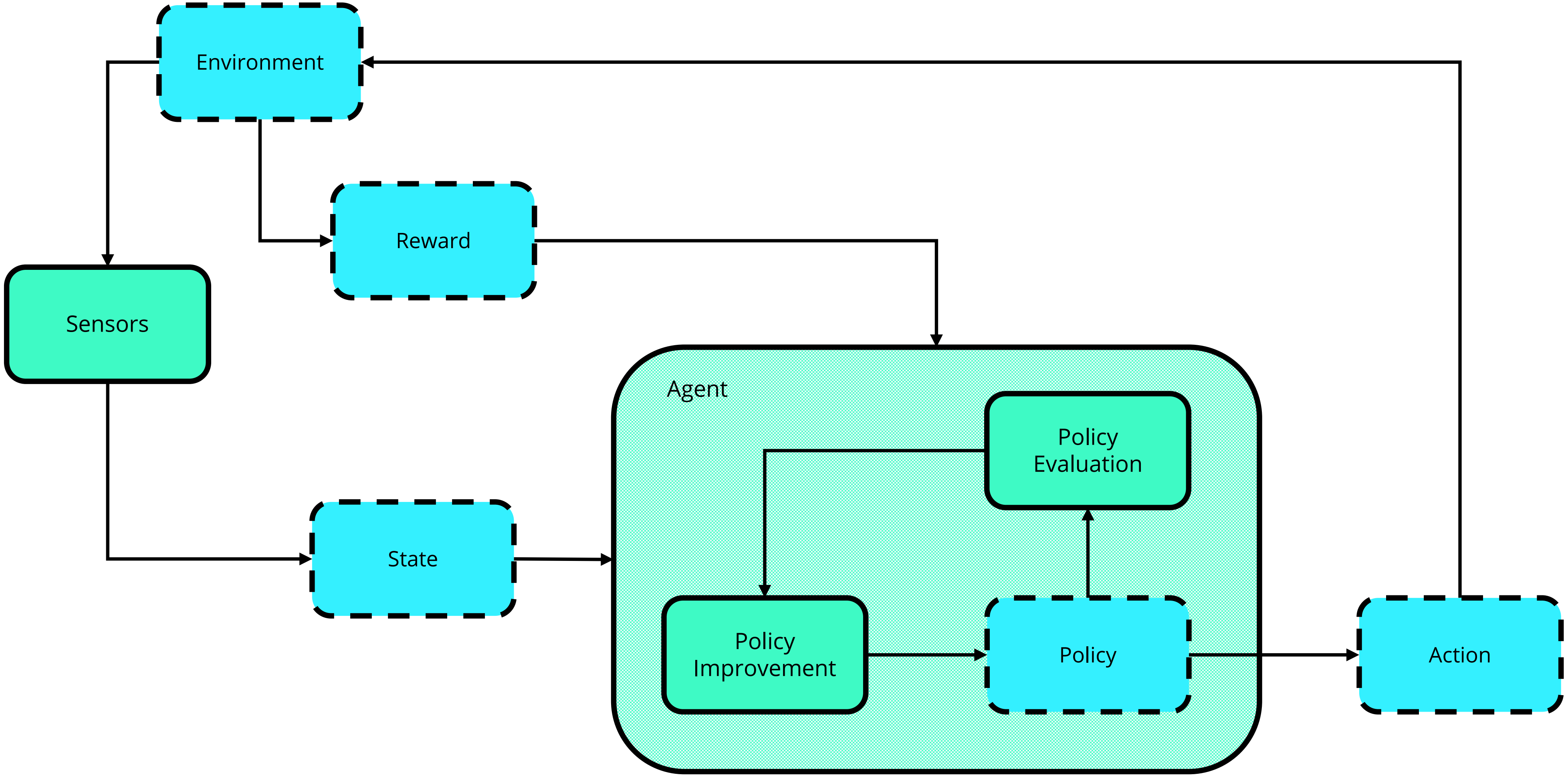
Reinforcement Learning

Networks for learning *actions, values, policies,* and/or *models*

"Teach by experience"



Reinforcement Learning
Formalisms



Reinforcement Learning vs **Supervised Learning**

What makes RL interesting in the real world ?

Sequential Nature

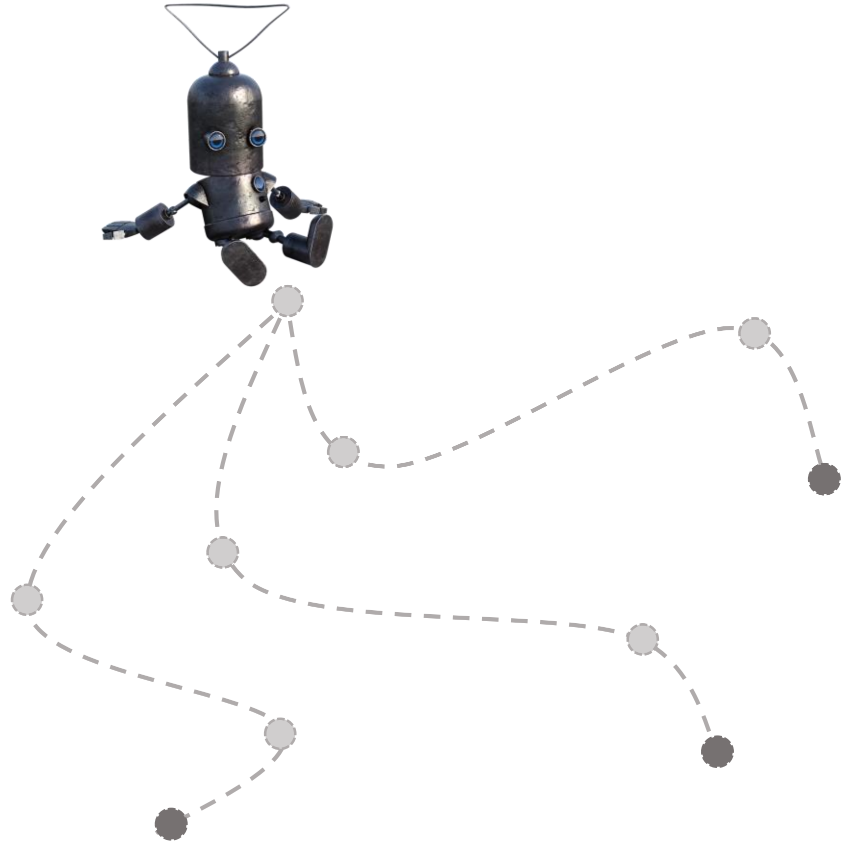
MDPs embed the notions of sequences of steps very naturally

$\boldsymbol{S}$: *state space*

$\boldsymbol{A}$: *action space*

$\boldsymbol{P}(\boldsymbol{s}' | \boldsymbol{s}', \boldsymbol{a})$: *transition probability*

$\boldsymbol{R}(\boldsymbol{s}, \boldsymbol{s}')$: *reward function*



Rewards vs **Labels**

Rewards measure how good a particular situation is; they don't prescribe correct behavior which is harder to get, and more restrictive.

Offline Reinforcement Learning
Definition and Nomenclature

RL algorithms that require no interaction with an environment during training

No Interaction

No interaction means that we'll have to learn from a **fixed batch of data**

Behavioural Cloning

No attempt to achieve better performance that the agent used to generate the batch of data

Pre-training

This is fundamentally different from a paradigm where we pre-train an RL model to accelerate convergence

"Data-driven Reinforcement Learning"

"Batch Reinforcement Learning"

"Truly off-policy Reinforcement Learning"

"Offline Reinforcement Learning"

RL Recent Milestones

What do they have in common?

Simulation Environments

A cheap way to interact or simulate interaction with the world

Why is it the case?

Does it have to be like that?

Is this an acceptable constraint?

Simulation Environments
Practicality and Consequences

Building a simulation environment

Costly to build

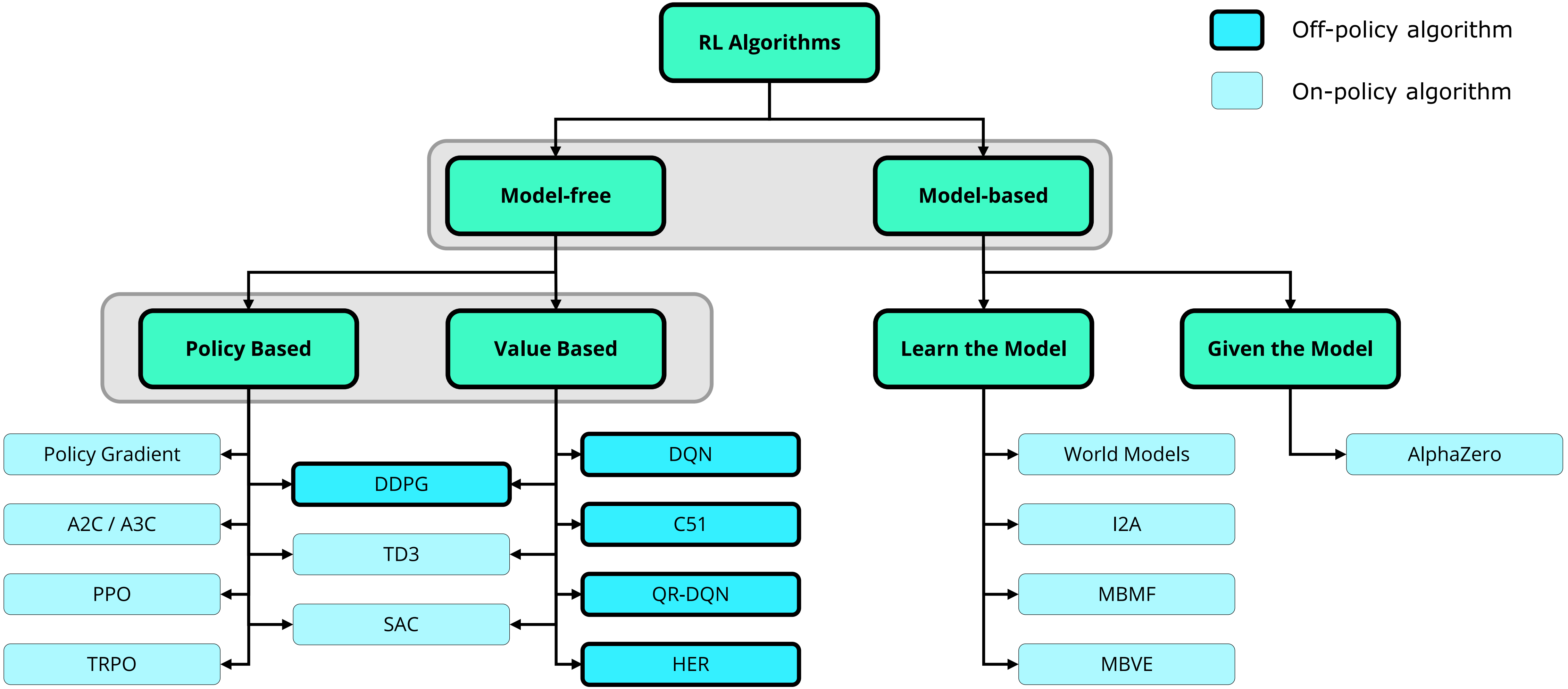
May require expert knowledge

Not knowing system dynamics

Impossible to build a simulation

Interacting with real world might be unacceptable

Reinforcement Learning
Taxonomy



Reinforcement Learning

Taxonomy

Model Free

Don't care how the world works as long as we know how to act in it

Model Based

Understand the world **in order to** learn how to act in it

Policy Based

Explicitly improve the policy we have (tend to be on-policy)

Value Based

Evaluate the intrinsic value of each state; derive a policy from that

On-Policy

We need to act and see the effect on the world in order to improve how we act

Off-Policy

Behavioral policy is potentially **unrelated** to the policy being optimized

Off-Policy RL
How off policy can we go?

Algorithm 1: deep Q-learning with experience replay.
Initialize replay memory D to capacity N
Initialize action-value function Q with random weights θ
Initialize target action-value function $\hat{Q}$ with weights $\theta^{-} = \theta$
For episode = 1, M **do**
  Initialize sequence $s_1 = \{x_1\}$ and preprocessed sequence $\phi_1 = \phi(s_1)$
  For t = 1,T **do**
    With probability ε select a random action a_t
    otherwise select $a_t = \mathrm{argmax}_a Q(\phi(s_t), a; \theta)$
    Execute action a_t in emulator and observe reward r_t and image x_{t+1}
    Set $s_{t+1} = s_t, a_t, x_{t+1}$ and preprocess $\phi_{t+1} = \phi(s_{t+1})$
    Store transition $(\phi_t, a_t, r_t, \phi_{t+1})$ in D
    Sample random minibatch of transitions $(\phi_j, a_j, r_j, \phi_{j+1})$ from D
    Set $y_j = \begin{cases} r_j & \text{if episode terminates at step } j+1 \\ r_j + \gamma \max_{a'} \hat{Q}(\phi_{j+1}, a'; \theta^{-}) & \text{otherwise} \end{cases}$
    Perform a gradient descent step on $\left(y_j - Q(\phi_j, a_j; \theta)\right)^2$ with respect to the network parameters θ
    Every C steps reset $\hat{Q} = Q$
  End For
End For

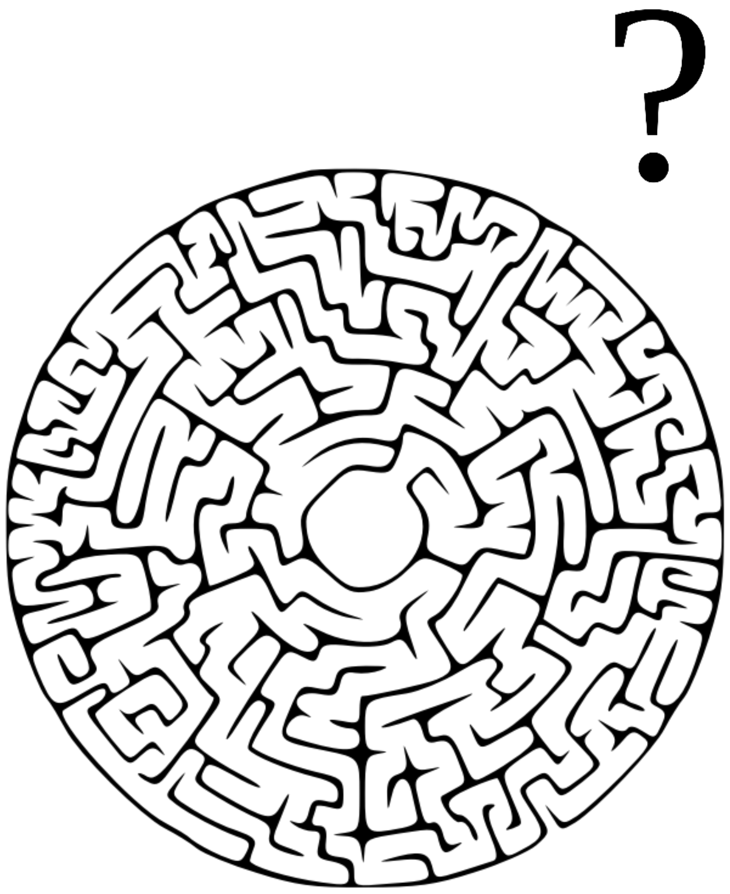
From: Original DQN paper

- a_j and $argmax_a$ are potentially unrelated
- we could replace these steps by a batch of data

What could go wrong?

Offline RL

The consequences



Forfeit the right to explore

Training from a fixed batch of data means we cannot improve exploration

What if …

Offline RL is actually about guessing the consequences of actions not taken

Offline Reinforcement Learning

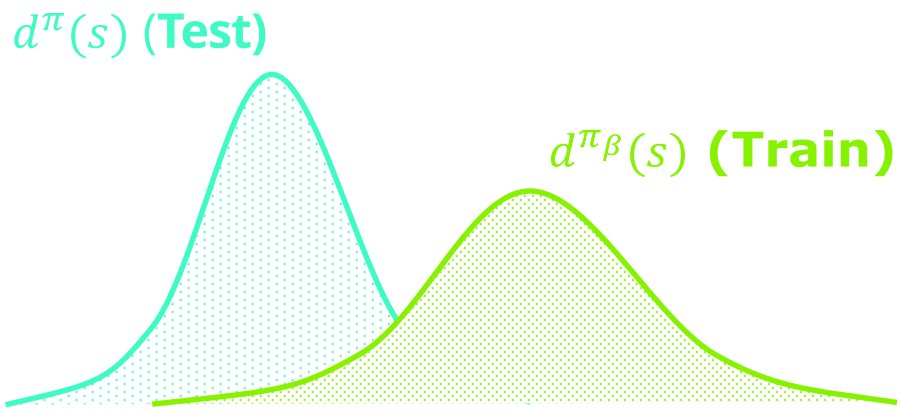
The problems

What about iid?

Finding $\pi \neq \pi_\beta$, may lead to $d^{\pi}(s)$ being very different than $d^{\pi_\beta}(s)$

Distributional Shift

Most models used are based on the assumption that training data is identical to the data seen by the model once deployed



Sequential nature

With no empirical error minimization for ood states and actions, the sequential nature of the RL framework makes it easy for errors to accumulate and be propagated to other states

Offline RL

What is to be done?

Constraining Policies

The basic approach to Offline RL is to try and make sure $d^{\pi_\beta}(s)$ is not that different from $d^{\pi}(s)$.

Many ways to achieve

The constraining of policies might be achieved in many different ways

Offline RL
Policy Constraints

Constraining π

If we explicitly constraint π to be "close" to π_β, we'll probably reduce the impact of ood states.

Assumptions

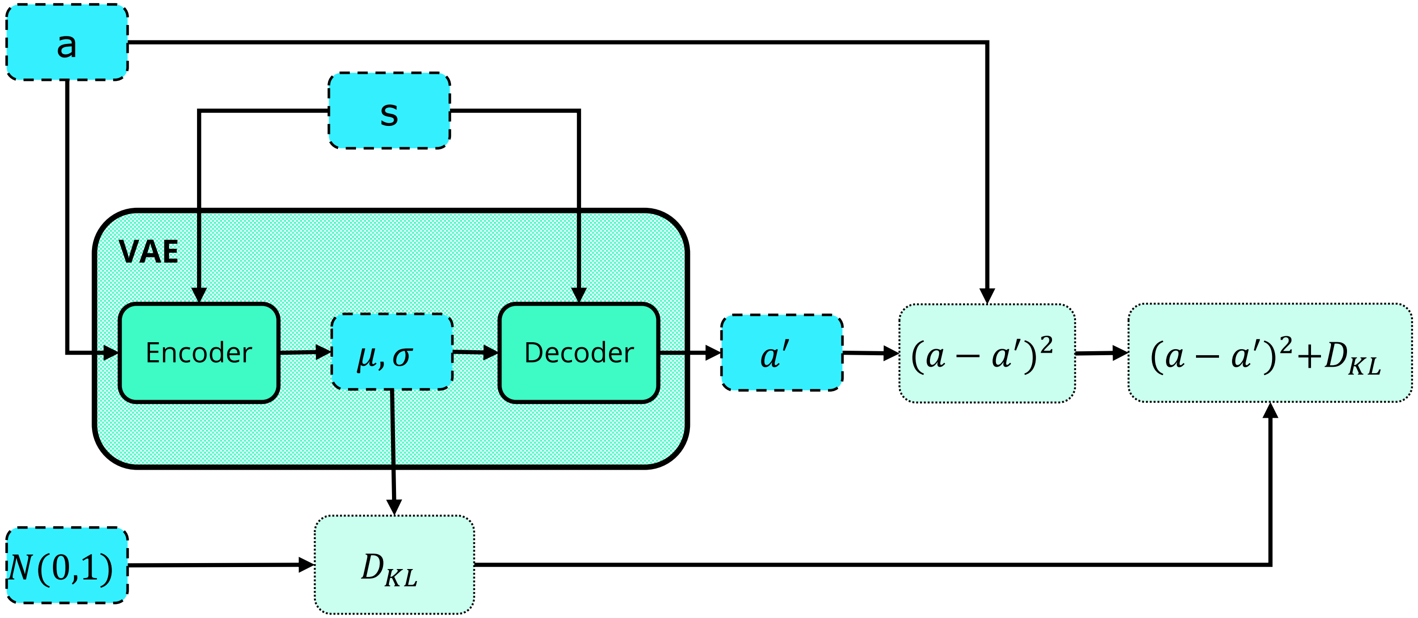
- Similar actions generate similar results
- Making π_β be similar to π is actually a good thing

Algorithm 1 BCQ

Input: Batch $\mathcal{B}$, horizon T, target network update rate τ, mini-batch size N, max perturbation Φ, number of sampled actions n, minimum weighting λ.
Initialize Q-networks $Q_{\theta_1}, Q_{\theta_2}$, perturbation network ξ_ϕ, and VAE $G_\omega = \{E_{\omega_1}, D_{\omega_2}\}$, with random parameters θ_1, θ_2, ϕ, ω, and target networks $Q_{\theta'_1}, Q_{\theta'_2}$, $\xi_{\phi'}$ with $\theta'_1 \leftarrow \theta_1, \theta'_2 \leftarrow \theta_2$, $\phi' \leftarrow \phi$.
for $t = 1$ **to** T **do**
  Sample mini-batch of N transitions (s, a, r, s') from $\mathcal{B}$
  $\mu, \sigma = E_{\omega_1}(s, a), \quad \tilde{a} = D_{\omega_2}(s, z), \quad z \sim \mathcal{N}(\mu, \sigma)$
  $\omega \leftarrow \operatorname{argmin}_\omega \sum (a - \tilde{a})^2 + D_{\mathrm{KL}}(\mathcal{N}(\mu, \sigma) || \mathcal{N}(0, 1))$
  Sample n actions: $\{a_i \sim G_\omega(s')\}_{i=1}^n$
  Perturb each action: $\{a_i = a_i + \xi_\phi(s', a_i, \Phi)\}_{i=1}^n$
  Set value target y (Eqn. 13)
  $\theta \leftarrow \operatorname{argmin}_\theta \sum (y - Q_\theta(s, a))^2$
  $\phi \leftarrow \operatorname{argmax}_\phi \sum Q_{\theta_1}(s, a + \xi_\phi(s, a, \Phi)), a \sim G_\omega(s)$
  Update target networks: $\theta'_i \leftarrow \tau\theta + (1 - \tau)\theta'_i$
  $\phi' \leftarrow \tau\phi + (1 - \tau)\phi'$
end for

From: BCQ paper (https://arxiv.org/pdf/1812.02900.pdf*)*

BCQ
Training VAE

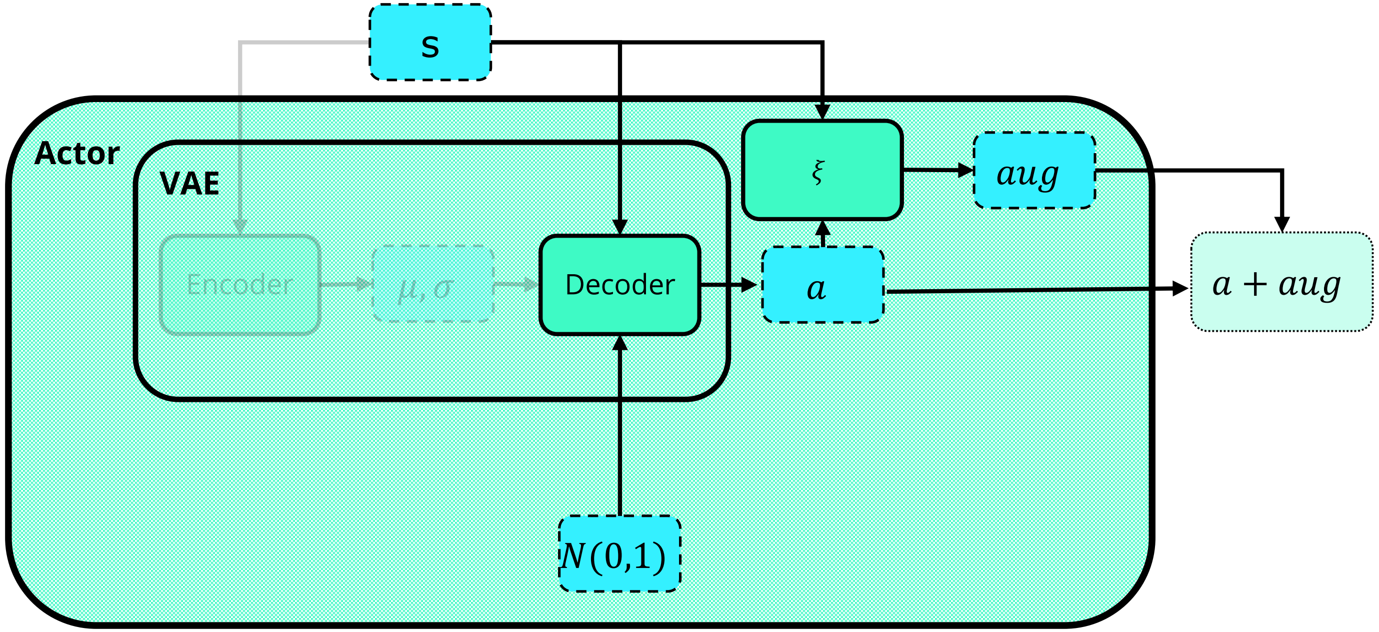


Algorithm 1 BCQ

Input: Batch $\mathcal{B}$, horizon T, target network update rate τ, mini-batch size N, max perturbation Φ, number of sampled actions n, minimum weighting λ.
Initialize Q-networks $Q_{\theta_1}, Q_{\theta_2}$, perturbation network ξ_ϕ, and VAE $G_\omega = \{E_{\omega_1}, D_{\omega_2}\}$, with random parameters θ_1, θ_2, ϕ, ω, and target networks $Q_{\theta'_1}, Q_{\theta'_2}, \xi_{\phi'}$ with $\theta'_1 \leftarrow \theta_1, \theta'_2 \leftarrow \theta_2$, $\phi' \leftarrow \phi$.
for $t = 1$ **to** T **do**
  Sample mini-batch of N transitions (s, a, r, s') from $\mathcal{B}$
  $\mu, \sigma = E_{\omega_1}(s, a), \quad \tilde{a} = D_{\omega_2}(s, z), \quad z \sim \mathcal{N}(\mu, \sigma)$
  $\omega \leftarrow \mathrm{argmin}_\omega \sum (a - \tilde{a})^2 + D_{\mathrm{KL}}(\mathcal{N}(\mu, \sigma) || \mathcal{N}(0, 1))$
  Sample n actions: $\{a_i \sim G_\omega(s')\}_{i=1}^n$
  Perturb each action: $\{a_i = a_i + \xi_\phi(s', a_i, \Phi)\}_{i=1}^n$
  Set value target y (Eqn. 13)
  $\theta \leftarrow \mathrm{argmin}_\theta \sum (y - Q_\theta(s, a))^2$
  $\phi \leftarrow \mathrm{argmax}_\phi \sum Q_{\theta_1}(s, a + \xi_\phi(s, a, \Phi)), a \sim G_\omega(s)$
  Update target networks: $\theta'_i \leftarrow \tau\theta + (1 - \tau)\theta'_i$
  $\phi' \leftarrow \tau\phi + (1 - \tau)\phi'$
end for

From: BCQ paper (https://arxiv.org/pdf/1812.02900.pdf)

BCQ
Acting VAE



Algorithm 1 BCQ

Input: Batch $\mathcal{B}$, horizon T, target network update rate τ, mini-batch size N, max perturbation Φ, number of sampled actions n, minimum weighting λ.
Initialize Q-networks $Q_{\theta_1}, Q_{\theta_2}$, perturbation network ξ_ϕ, and VAE $G_\omega = \{E_{\omega_1}, D_{\omega_2}\}$, with random parameters θ_1, θ_2, ϕ, ω, and target networks $Q_{\theta'_1}, Q_{\theta'_2}$, $\xi_{\phi'}$ with $\theta'_1 \leftarrow \theta_1, \theta'_2 \leftarrow \theta_2$, $\phi' \leftarrow \phi$.
for $t = 1$ **to** T **do**
  Sample mini-batch of N transitions (s, a, r, s') from $\mathcal{B}$
  $\mu, \sigma = E_{\omega_1}(s, a), \quad \tilde{a} = D_{\omega_2}(s, z), \quad z \sim \mathcal{N}(\mu, \sigma)$
  $\omega \leftarrow \text{argmin}_\omega \sum (a - \tilde{a})^2 + D_{\text{KL}}(\mathcal{N}(\mu, \sigma) || \mathcal{N}(0, 1))$
  Sample n actions: $\{a_i \sim G_\omega(s')\}_{i=1}^n$
  Perturb each action: $\{a_i = a_i + \xi_\phi(s', a_i, \Phi)\}_{i=1}^n$
  Set value target y (Eqn. 13)
  $\theta \leftarrow \text{argmin}_\theta \sum (y - Q_\theta(s, a))^2$
  $\phi \leftarrow \text{argmax}_\phi \sum Q_{\theta_1}(s, a + \xi_\phi(s, a, \Phi)), a \sim G_\omega(s)$
  Update target networks: $\theta'_i \leftarrow \tau\theta + (1 - \tau)\theta'_i$
  $\phi' \leftarrow \tau\phi + (1 - \tau)\phi'$
end for

From: BCQ paper (https://arxiv.org/pdf/1812.02900.pdf)

Offline RL
Uncertainty Estimation

$$\pi_\phi := \max_{\pi \in \Delta_{|S|}} \mathbb{E}_{s \sim \mathcal{D}} \mathbb{E}_{a \sim \pi(\cdot|s)} \left[\min_{j=1,..,K} \hat{Q}_j(s,a) \right] \quad \text{s.t.} \quad \mathbb{E}_{s \sim \mathcal{D}}[\text{MMD}(\mathcal{D}(s), \pi(\cdot|s))] \leq \varepsilon \qquad (1)$$

Managing uncertainty

Use a measure of uncertainty to restrict the actions taken

Algorithm 1 BEAR Q-Learning (BEAR-QL)

input : Dataset $\mathcal{D}$, target network update rate τ, mini-batch size N, sampled actions for MMD n, minimum λ

Initialize Q-ensemble $\{Q_{\theta_i}\}_{i=1}^K$, actor π_ϕ, Lagrange multiplier α, target networks $\{Q_{\theta'_i}\}_{i=1}^K$, and a target actor $\pi_{\phi'}$, with $\phi' \leftarrow \phi, \theta'_i \leftarrow \theta_i$

for t in $\{1, \ldots, \text{N}\}$ **do**

Sample mini-batch of transitions $(s, a, r, s') \sim \mathcal{D}$

Q-update:

Sample p action samples, $\{a_i \sim \pi_{\phi'}(\cdot|s')\}_{i=1}^p$

Define $y(s,a) := \max_{a_i} [\lambda \min_{j=1,..,K} Q_{\theta'_j}(s', a_i) + (1-\lambda) \max_{j=1,..,K} Q_{\theta'_j}(s', a_i)]$

$\forall i, \theta_i \leftarrow \arg\min_{\theta_i} (Q_{\theta_i}(s,a) - (r + \gamma y(s,a)))^2$

Policy-update:

Sample actions $\{\hat{a}_i \sim \pi_\phi(\cdot|s)\}_{i=1}^m$ and $\{a_j \sim \mathcal{D}(s)\}_{j=1}^n$, n preferably an intermediate integer(1-10)

Update ϕ, α by minimizing Equation 1 by using dual gradient descent with Lagrange multiplier α

Update Target Networks: $\theta'_i \leftarrow \tau\theta_i + (1-\tau)\theta'_i$; $\phi' \leftarrow \tau\phi + (1-\tau)\phi'$

end for

Assumptions

(epistemic) Uncertainty will be larger for ood states

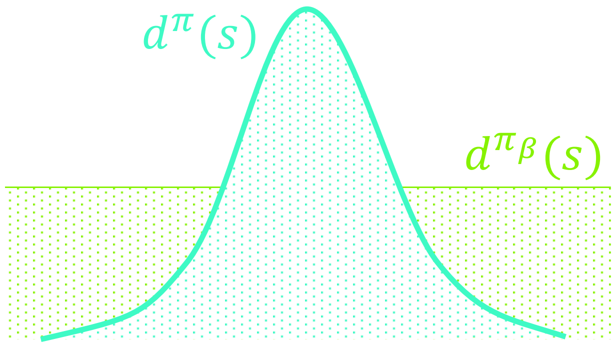
From: BEAR paper (https://arxiv.org/pdf/1906.00949.pdf)

Offline RL

Uncertainty Estimation vs Policy Constraints

Preventing ood states and actions

Constraining π to be close to π_β might be too restrictive. (e.g. uniform π_β)



In Practice

Pure Uncertainty estimation methods don’t seem to work that well in practice

Offline RL
Other Approaches

Policy Gradient Methods

Some variation of importance sampling might be used used but it suffers from some problems, namely high variance

$$\nabla_\theta J(\pi_\theta) = \mathbb{E}_{\tau\sim\pi_\beta(\tau)}\left[\frac{\pi_\theta(\tau)}{\pi_\beta(\tau)}\sum_{t=0}^{H}\gamma^t\nabla_\theta\log\pi_\theta(\mathbf{a}_t|\mathbf{s}_t)\hat{A}(\mathbf{s}_t,\mathbf{a}_t)\right]$$
$$= \mathbb{E}_{\tau\sim\pi_\beta(\tau)}\left[\left(\prod_{t=0}^{H}\frac{\pi_\theta(\mathbf{a}_t|\mathbf{s}_t)}{\beta(\mathbf{a}_t|\mathbf{s}_t)}\right)\sum_{t=0}^{H}\gamma^t\nabla_\theta\log\pi_\theta(\mathbf{a}_t|\mathbf{s}_t)\hat{A}(\mathbf{s}_t,\mathbf{a}_t)\right]$$

From https://arxiv.org/pdf/2005.01643.pdf

Standard RL approaches

Distributional DQN seems to cope very well with the challenges of Offline RL

Offline RL

Important Issues & Open Problems

Sample breathiness

Very “focused” sample batches might not be enough to improve behavior

Model Evaluation

Hyper-parameter tuning

Interaction for evaluation

off-policy Evaluation

Sample acquisition

Non RL agents (markov property)

Multiple agents

Suboptimal agents

Benchmarking

Some datasets and research

Initial stages

Offline RL
Model Evaluation

Off Policy Evaluation

Particular settings of binary RL allow for an off policy evaluation (e.g. Off-Policy Evaluation via Off-Policy Classification Paper)

Using other performance measures

Recommendation Systems, for example, may use different measures of performance that are not average reward (e.g. precision @ k, recall @ k)

Offline Reinforcement Learning

Key Takeaway

Why Reinforcement Learning ?	Handling sequences, rewards vs labels.
Why Offline Reinforcement Learning?	No need for simulation environments. Use of big datasets.
What are the main challenges to Offline RL?	Distributional shift.
What are the most promising approaches in Offline RL?	Policy Constraints and Uncertainty Estimation methods.
What are the unresolved issues in Offline RL?	Model Evaluation, Sample acquisition, Benchmarks.
How can I use Offline RL?	

Offline Reinforcement Learning
References and useful links

http://papers.nips.cc/paper/8783-off-policy-evaluation-via-off-policy-classification.pdf - Off-Policy Evaluation

https://arxiv.org/pdf/1812.02900.pdf - BCQ Paper

https://arxiv.org/pdf/1907.04543.pdf - REM Model Paper

https://arxiv.org/pdf/1906.00949.pdf - BEAR Paper

https://ai.googleblog.com/2020/04/an-optimistic-perspective-on-offline.html - Blog post on BEAR Paper

https://ai.googleblog.com/2019/06/off-policy-classification-new.html - Blog post on Off Policy Model evaluation

https://arxiv.org/pdf/2005.01643.pdf - Very complete survey of offline Reinforcement Learning

https://arxiv.org/pdf/2004.07219.pdf - Paper about benchmarking Offline Reinforcement Learning agents

Thank you!
Any questions?

José Trocado Moreira
josmoreira@deloitte.pt