535514: Reinforcement Learning

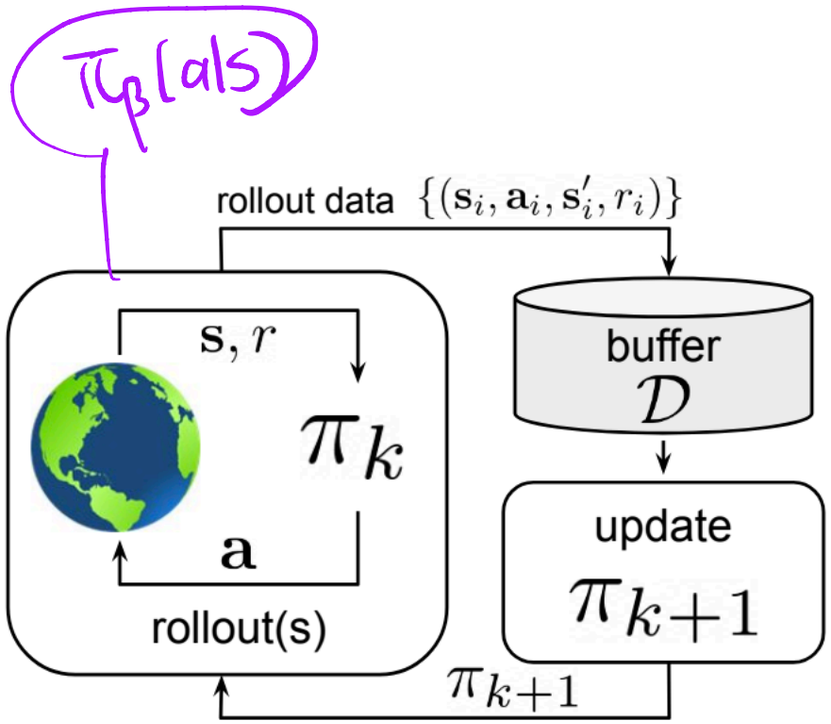
Lecture 23 — Distributional RL

Ping-Chun Hsieh

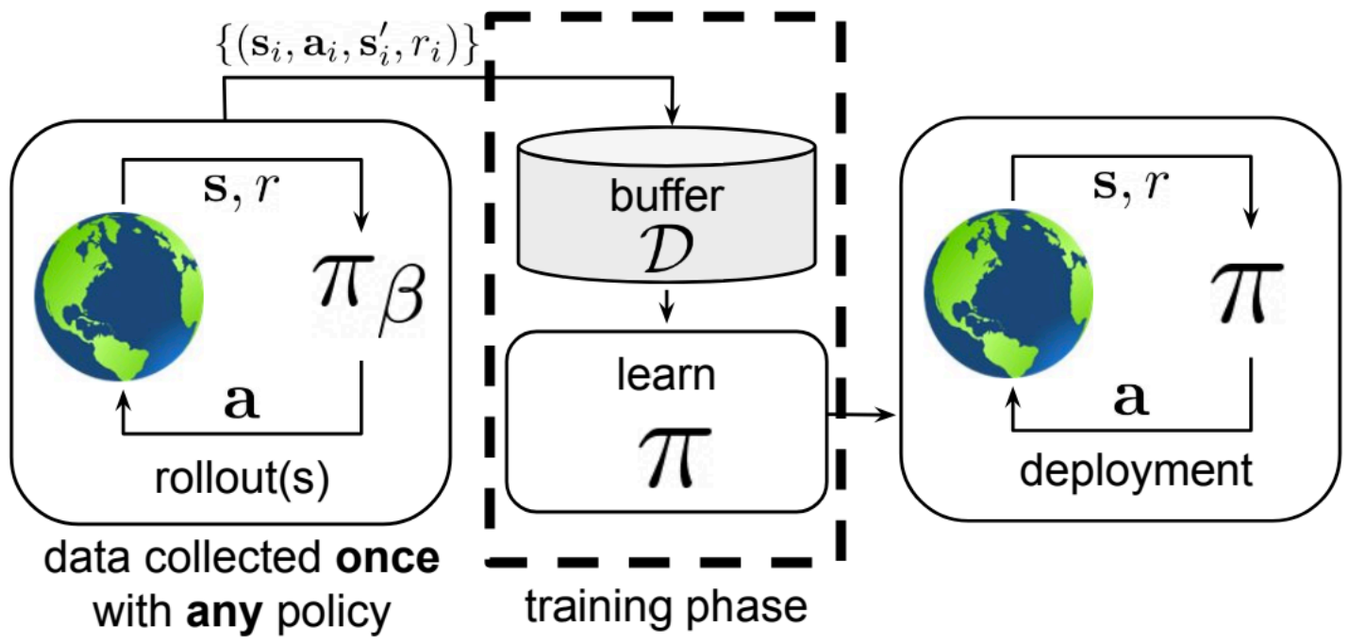
May 13, 2024

Off-Policy RL vs Offline RL

Off-Policy RL



Offline-RL



What are the differences?

1. Online interactions possible
2. Knowledge about behavior policy

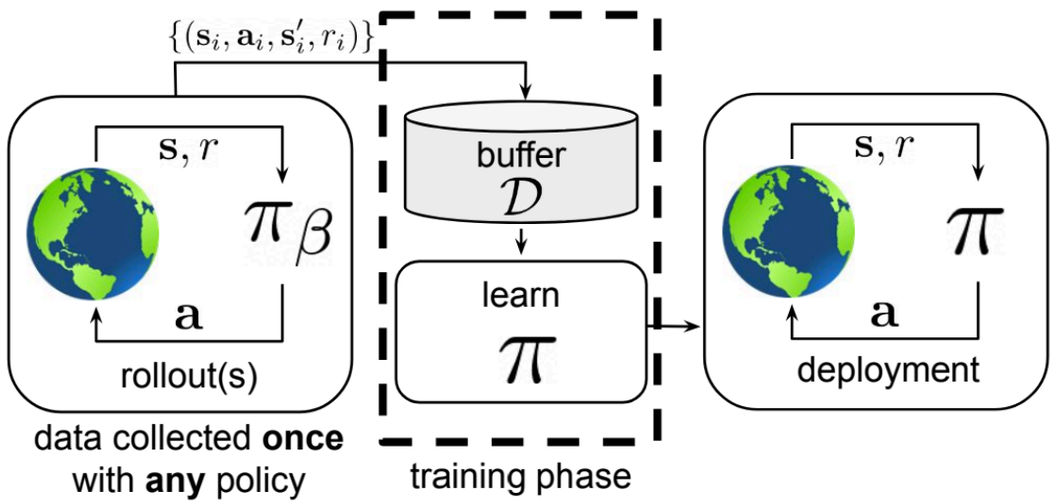
A Classic Offline RL Method: Fitted Q Iteration

- **Fitted Q Iteration (FQI):** Given an offline dataset $D = \{(s_i, a_i, r_i, s_i')\}$

In each iteration k:

Step 1. Let $y_i = r_i + \gamma \max_{a \in A} Q(s_i', a; \mathbf{w}_k)$, for each i

Step 2. Set $\mathbf{w}_{k+1} \leftarrow \arg\min_{\mathbf{w}} \frac{1}{2} \sum_{(s_i, a_i, r_i, s_i') \in D} \|Q_{\mathbf{w}}(s_i, a_i) - y_i\|_2^2$



If s_i and a_i are draw from exploratory distributions $\mu(s)$ and $\beta(a)$, then FQI has nice convergence

https://nanjiang.cs.illinois.edu/files/cs598/note5.pdf

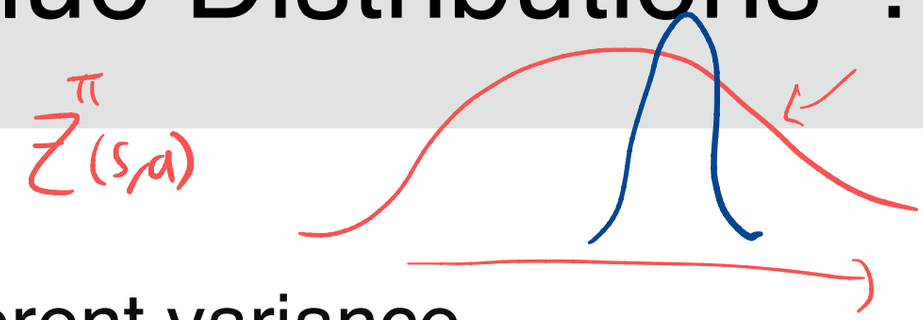
On-Policy vs Off-Policy Methods

	Policy Optimization	Value-Based	Model-Based	Imitation-Based
On-Policy	**Exact PG** **REINFORCE (w/i baseline)** **A2C** **On-policy DAC** **TRPO** **Natural PG (NPG)** **PPO-KL & PPO-Clip** **RLHF by PPO-KL**	**Epsilon-Greedy MC** **Sarsa** **Expected Sarsa**	**Model-Predictive Control (MPC)** **PETS**	**IRL** **GAIL** **IQ-Learn**
Off-Policy	**Off-policy DPG & DDPG** **Twin Delayed DDPG (TD3)**	**Q-learning** **Double Q-learning** **DQN & DDQN** **Rainbow** **C51 / QR-DQN / IQN** **Soft Actor-Critic (SAC)**		

Distributional Q-Learning

(Learn value distribution $Z(s, a)$ & use $E[Z(s, a)]$ as $Q(s, a)$ in Q-Learning)

Why Shall We Consider “Value Distributions”?



- Risky vs safe choices
  - E.g., Same expected return but different variance
- Good empirical performance (despite that the underlying root cause is not fully known)
  - C51 [Belleware et al., ICML 2017]
  - QR-DQN [Dabney et al., AAAI 2018]
  - IQN [Dabney et al., ICML 2018]
- New approaches for exploration
  - Information-directed exploration [Nikolov et al., ICLR 2019]
  - Distributional RL for efficient exploration [Mavrin et al., ICML 2019]
- Learn better critics
  - Truncated Quantile Critics (TQC) [Kuznetsov et al., ICML 2020]

Question: How to learn the complete value distribution (instead of merely the expectation)?

Sample Action-Value $Z^\pi(s,a)$

- Sample action-value $Z^\pi(s,a)$: sample return if we start from state s and take action a, and then follow policy π

$$Q^\pi(s,a) = \mathbb{E}[Z^\pi(s,a)] = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t,a_t)\right]$$

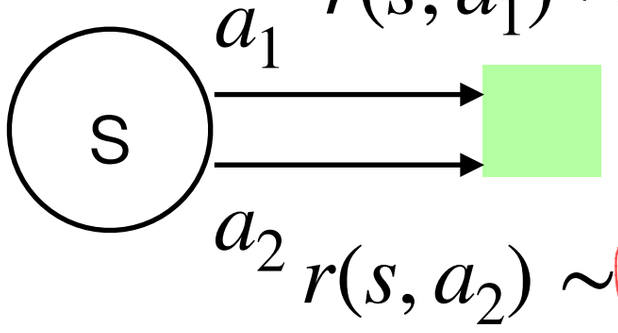
$s_0 = s, a_0 = a$

random variable

random variables

- $Z^\pi(s,a)$ is essentially a random variable

- Example: 1-state MDP with 2 actions and $\pi(s) = a_1$



- $Q^\pi(s,a_1)$ = ? Distribution of $Z^\pi(s,a_1)$?

0

$\mathcal{N}(0,1)$

- $Q^\pi(s,a_2)$ = ? Distribution of $Z^\pi(s,a_2)$?

1

$\mathcal{N}(1,10)$

Finding Z^π via Distributional Bellman Equation

- Mild assumption: $Z^\pi(s,a)$ has bounded moments
- Distributional Bellman equation for $Z^\pi(s,a)$: Given s, a, we have

$$Z^\pi(s,a) \overset{D}{=} r(s,a) + \gamma Z^\pi(s',a')$$

a random variable

a random variable

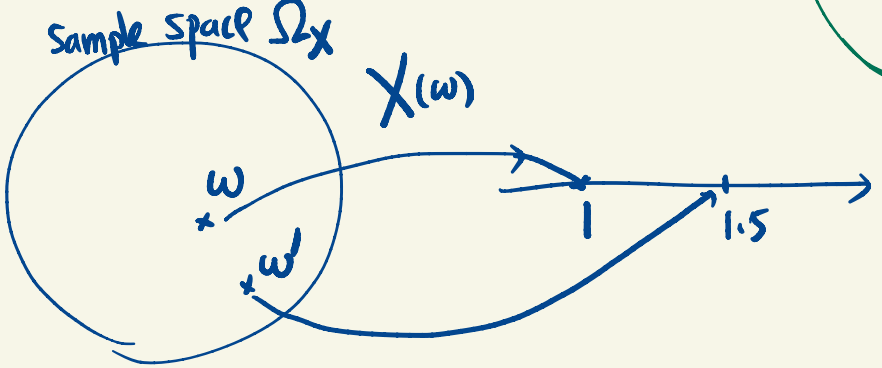
($\overset{D}{=}$: equal in distribution)

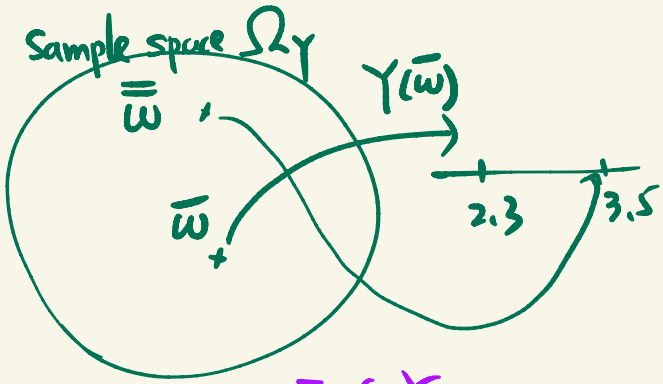
- Question: How to interpret this equation?
- Question: Are $r(s,a)$ and $Z^\pi(s',a')$ independent? (conditionally) Yes! (they involve different sources of randomness)
- Question: Is this consistent with Bellman expectation equation?

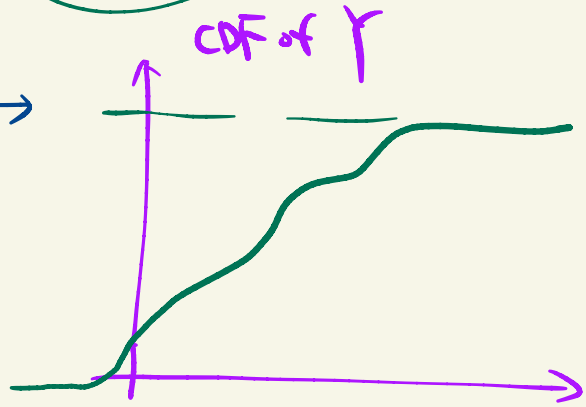
$$Q^\pi(s,a) = R(s,a) + \gamma \cdot E_{s',a'}[Q^\pi(s',a')]$$

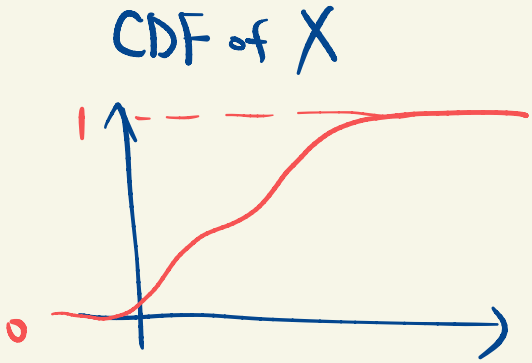
Two random variables:

X Y

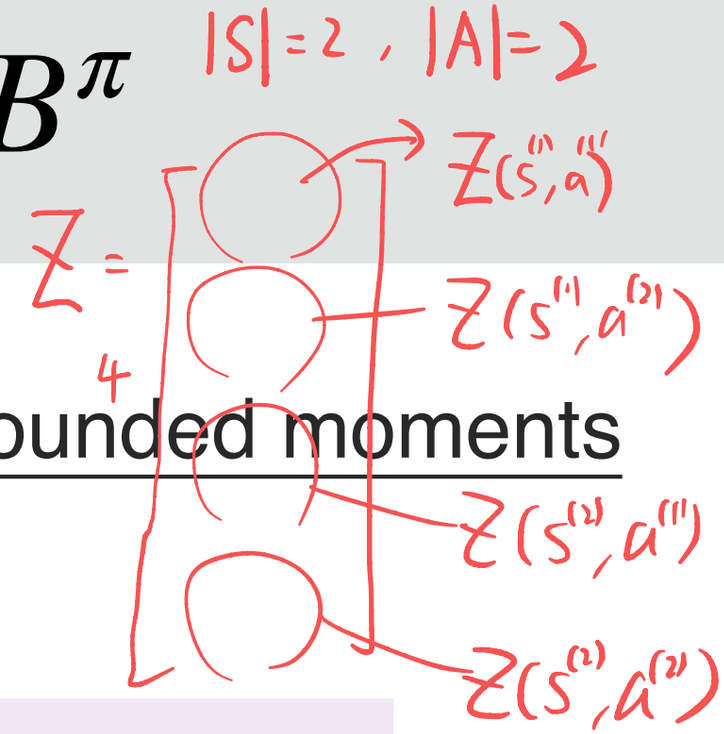








Distributional Bellman Operator B^π



- $\mathcal{Z}$: the space of all value distributions with bounded moments
- Transition operator $P^\pi : \mathcal{Z} \to \mathcal{Z}$

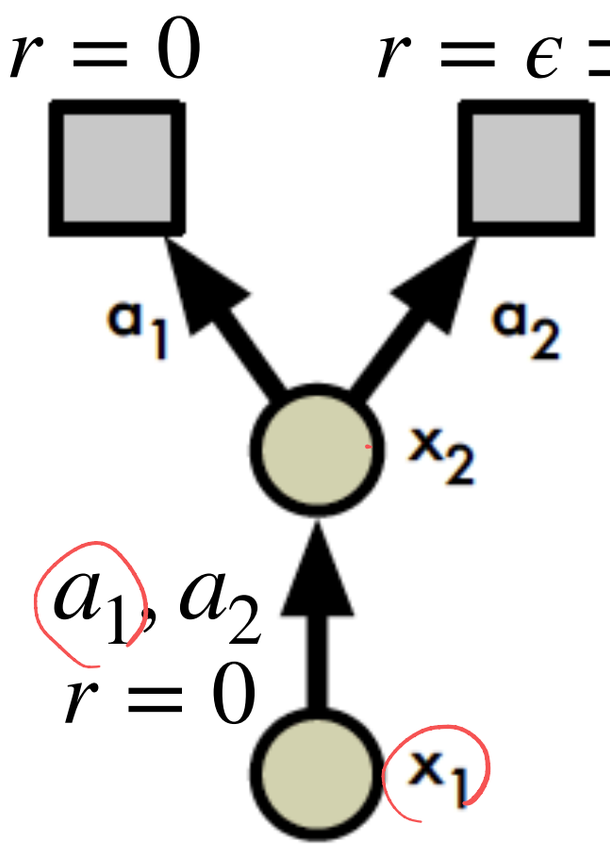
$$P^\pi Z(s,a) :\overset{D}{=} Z(s',a')$$
$$s' \sim P(\cdot \mid s,a),\ a' \sim \pi(\cdot \mid s')$$

- Distributional Bellman operator $B^\pi : \mathcal{Z} \to \mathcal{Z}$

$$B^\pi Z(s,a) :\overset{D}{=} r(s,a) + \gamma P^\pi Z(s,a)$$

An Example of Applying B^π

- Example: 2 states x_1, x_2 and 2 actions a_1, a_2
- $\pi(a_1 | x_2) = 0.3$, $\pi(a_2 | x_2) = 0.7$, and $\gamma = 0.9$

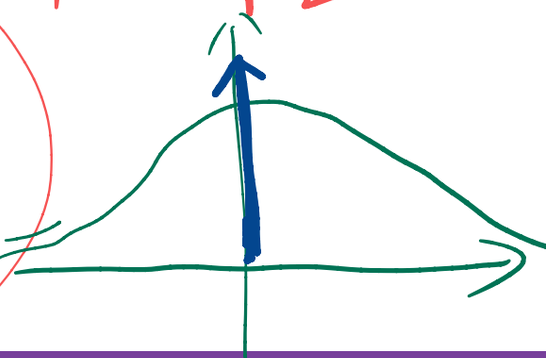


$$B^\pi Z(s,a) :\overset{D}{=} r(s,a) + \gamma P^\pi Z(s,a)$$

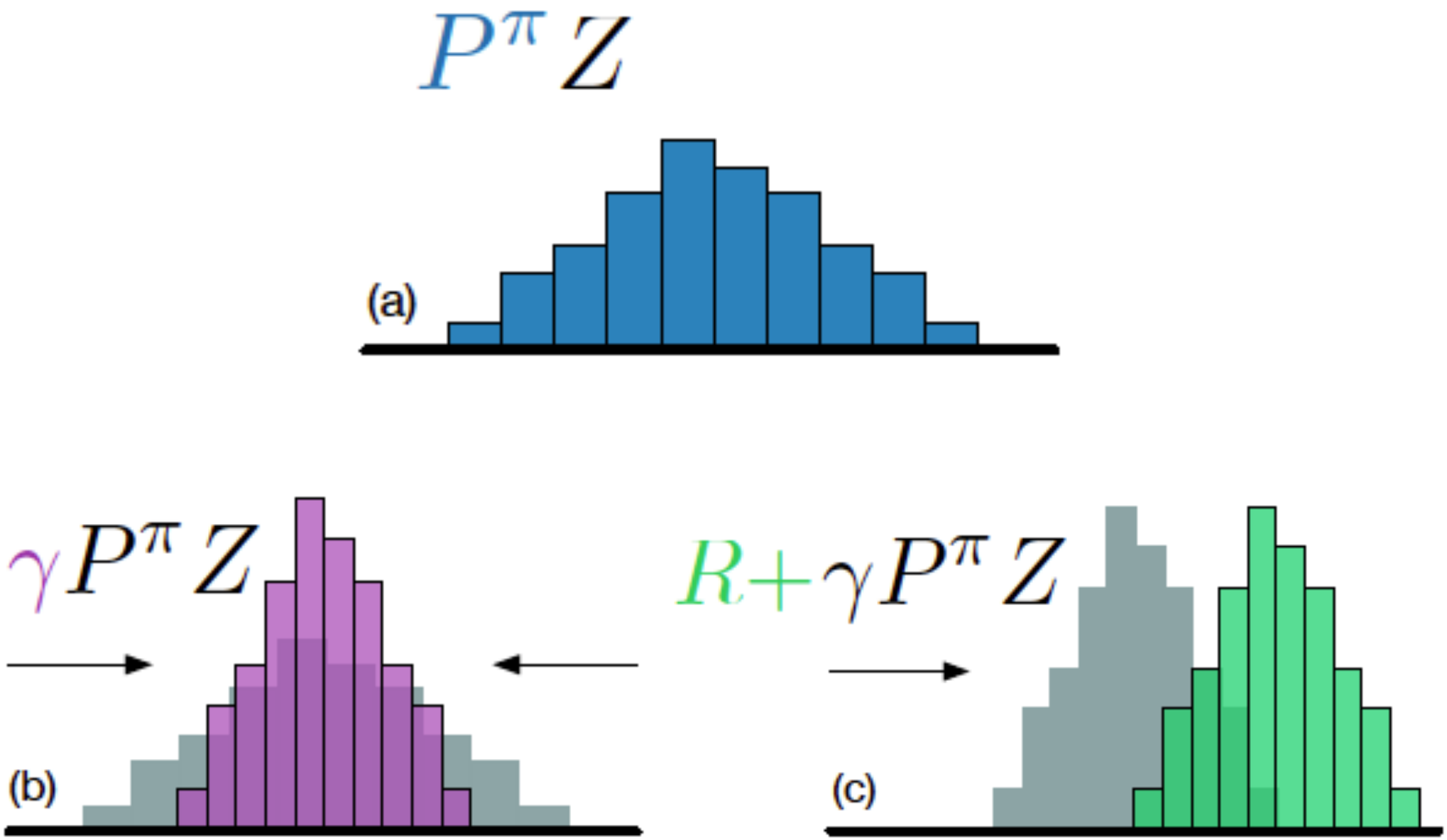
- Suppose $Z(x_1, a_1) = 0, Z(x_2, a_1) = 0$ with probability 1 and $Z(x_2, a_2) \sim \mathcal{N}(0,1)$
- Question: $B^\pi Z(x_2, a_2) = ?$ $B^\pi Z(x_1, a_1) = ?$

$B^\pi Z(x_2, a_2) = r(x_2, a_2) = \begin{cases} \epsilon+1, & \text{w.p. } \frac{1}{2} \\ \epsilon-1, & \text{w.p. } \frac{1}{2} \end{cases}$

$B^\pi Z(x_1, a_1) = \begin{cases} \gamma Z(x_2, a_1), & \text{w.p. } 0.3 \\ \quad \| \\ \quad 0 \\ \gamma Z(x_2, a_2), & \text{w.p. } 0.7 \end{cases}$ $\mathcal{N}(0,1)$



Visualization of Distributional Bellman Operator



Distributional "Optimality" Operator

Recall— Distributional Bellman operator $B^\pi : \mathcal{Z} \to \mathcal{Z}$

$$B^\pi Z(s,a) \overset{D}{:=} r(s,a) + \gamma P^\pi Z(s,a)$$

- Distributional **optimality** operator B^*: The B^π resulting from a greedy policy π, i.e,., $B^* \equiv B^{\pi_{greedy}}$

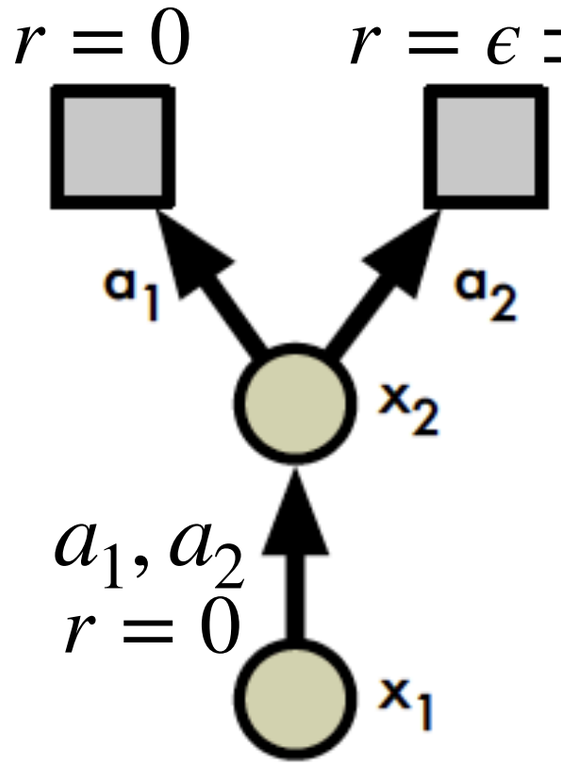
$P^\pi Z \equiv Z(s',a')$

$s' \sim P(\cdot|s,a)$, $a' \sim \pi(\cdot|s')$

What does "greedy" mean here?

Under π_{greedy}, $a' = \underset{\hat{a} \in A}{\text{argmax}}\, \mathbb{E}[Z(s',\hat{a})]$

An Example of B^*



Suppose we have the following:

- $\pi(a_1 \mid x_2) = 0.3, \pi(a_2 \mid x_2) = 0.7$, and $\gamma = 1$
- $Z(x_1, a_1) = 0, Z(x_2, a_1) = 0$ with probability 1
- $Z(x_2, a_2) \sim \mathcal{N}(1,2)$

Question: What's the PDF of $B^*Z(x_1, a_1)$ =?

$B^*Z(x_1,a_1) = Z(x_2,a_2) = \mathcal{N}(1,2)$

$B^*Z(s,a) :\overset{D}{=} r(s,a) + \gamma P^{\pi_{greedy}} Z(s,a)$

$Z(s',a')$

$E[Z(x_1,a_1)] = 0$

$E[Z(x_2,a_2)] = 1$

A Case Study on Distributional Q-Learning: C51

Bellemare et al., A Distributional Perspective on Reinforcement Learning, ICML 2017

Let's Design (Tabular) "Distributional" Q-Learning

- Recall: Standard Q-Learning

Step 1: Initialize $Q(s,a)$ ($Z(s,a)$) for all (s,a), and initial state s_0

Step 2: For each step $t = 0,1,2,\cdots$

Select a_t using ε-greedy w.r.t $Q(s_t,\cdot)$ $\mathbb{E}[Z(s,a)]$

Observe (r_{t+1}, s_{t+1})

$$Q(s_t,a_t) \leftarrow Q(s_t,a_t) + \alpha_t(s_t,a_t)\left(r_{t+1} + \gamma \max_{a'} Q(s_{t+1},a') - Q(s_t,a_t)\right)$$

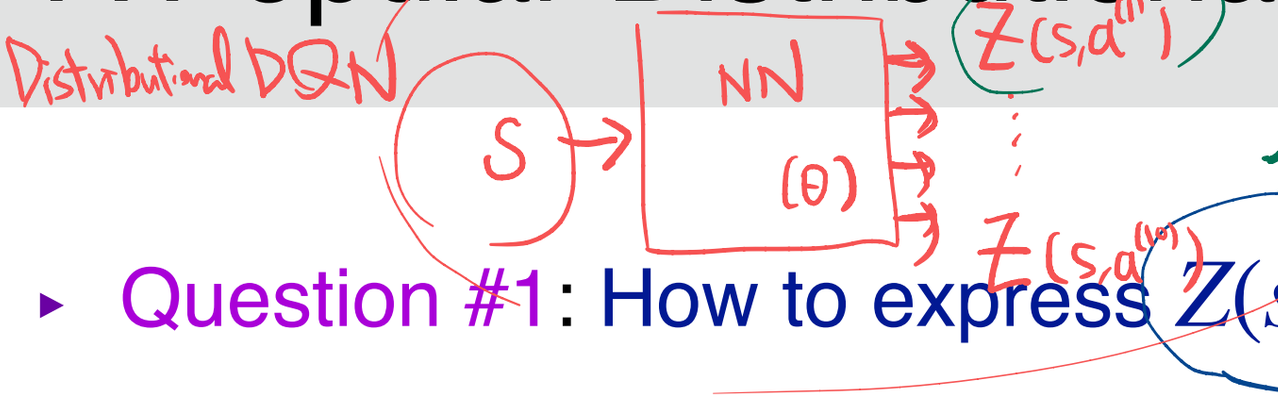
$$Z(s_t,a_t) \leftarrow (1-\alpha_t(s_t,a_t))Z(s_t,a_t) + \alpha_t(s_t,a_t)B^*Z(s_t,a_t)$$

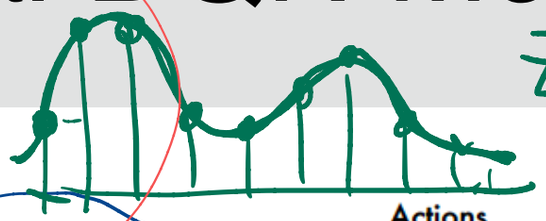
To get "distributional Q-learning", we need two modifications:

1. Action selection in ε-"greedy"
2. "Distributional" TD update

Next Question: “**Function approximation**” for distributional Q-learning?

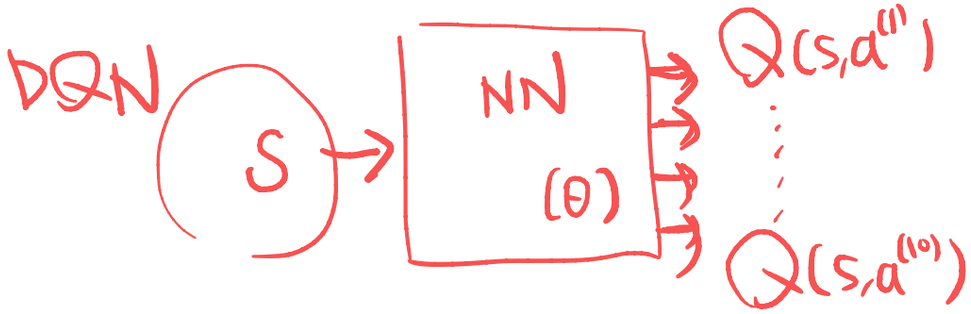
A Popular Distributional DQN Method: *C51*

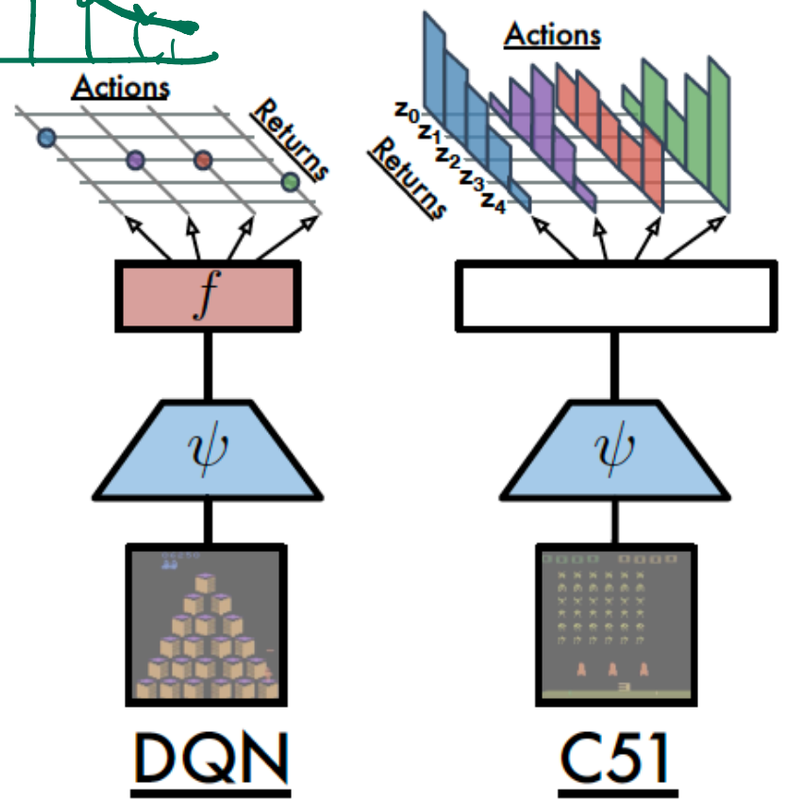




- Question #1: How to express $Z(s, a)$?

(C1) Categorical distributions for parametrizing $Z_\theta(s, a)$





- Question #2: How to update $Z(s, a)$ during training?

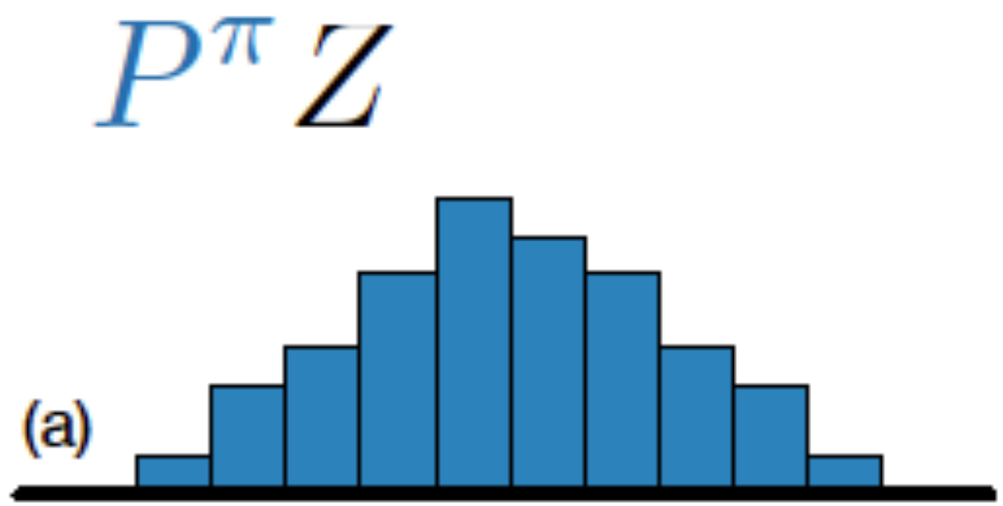
(C2) Mimicking B^* for learning with sample transitions (s, a, r, s')

(C3) Cramer Projection Φ for support mismatch caused by $B^* Z_\theta(s, a)$

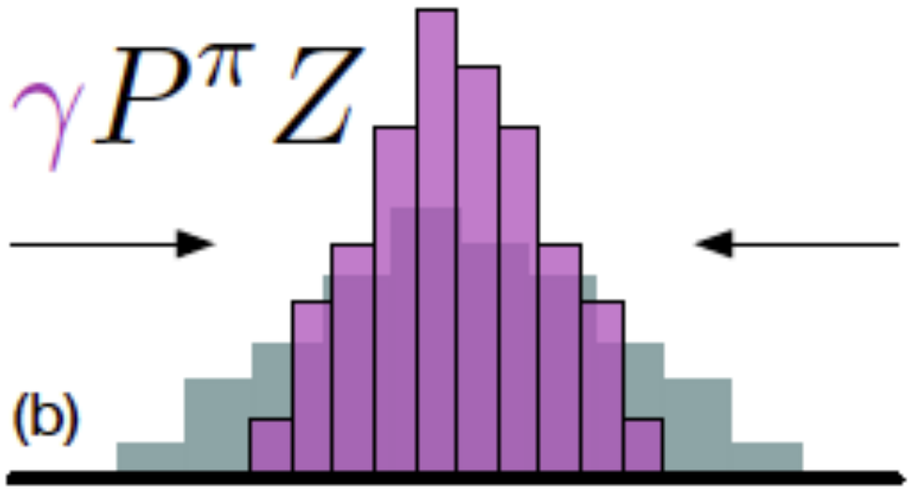
(C4) Minimize $L_{C51}(s, a, r, s'; \theta) := D_{KL}(\Phi B^* Z_{\bar{\theta}}(s, a) \| Z_\theta(s, a))$

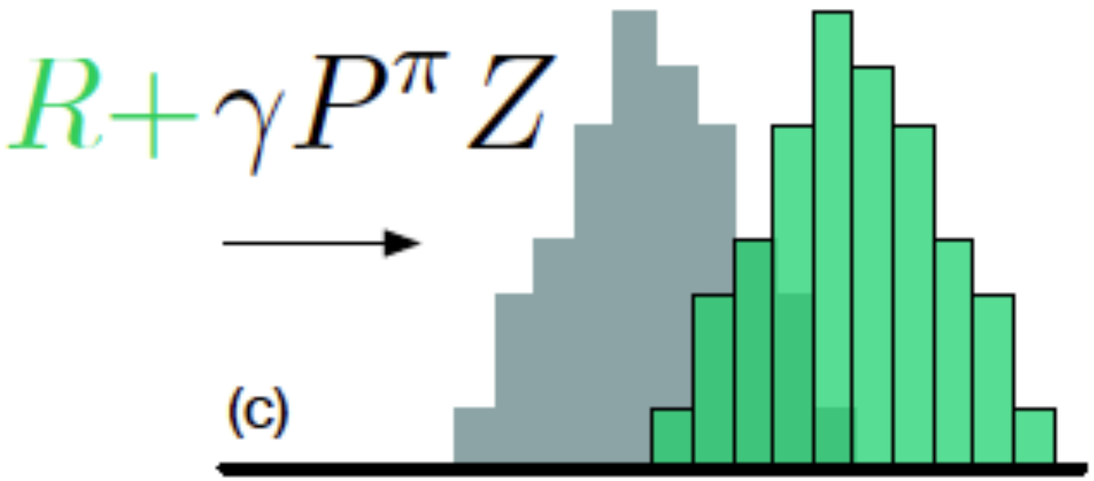
Visualization of C51: Distributional Optimality Operator + Projection

(Let π be a greedy policy)

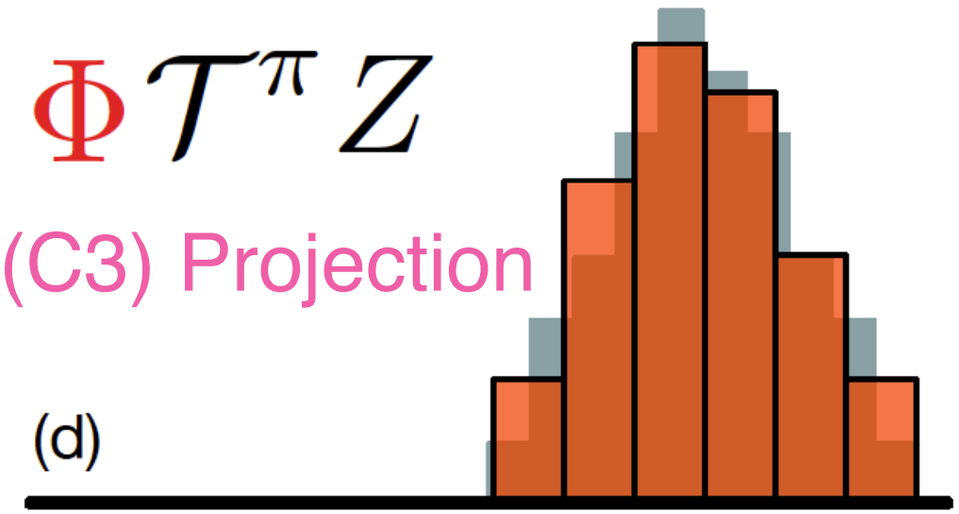


(C1) categorical distributions





(C2) Mimic B^*



(C3) Projection

(C1) Categorical Distributions for $Z_\theta(s, a)$

- Example: Categorical distributions with 3 "atoms"

	0	1	2
$Z_\theta(s_1, a_1)$	0.35	0.17	0.48
$Z_\theta(s_1, a_2)$	0.06	0.62	0.32

Possible values (each value is called an "atom")

Categorical distribution

- With more atoms, $Z_\theta(s, a)$ can be approximated more accurately

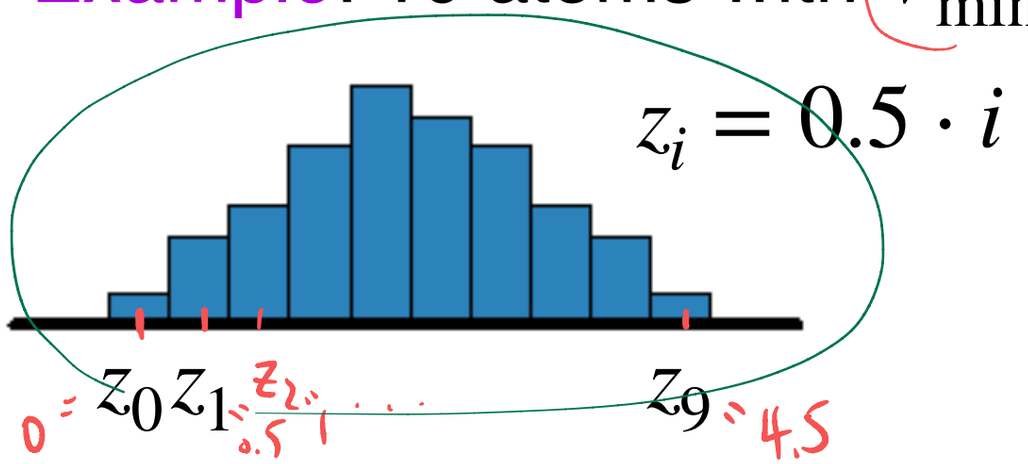
- Question: Any inherent assumption about using categorical distributions? Upper & lower limits of possible values are known

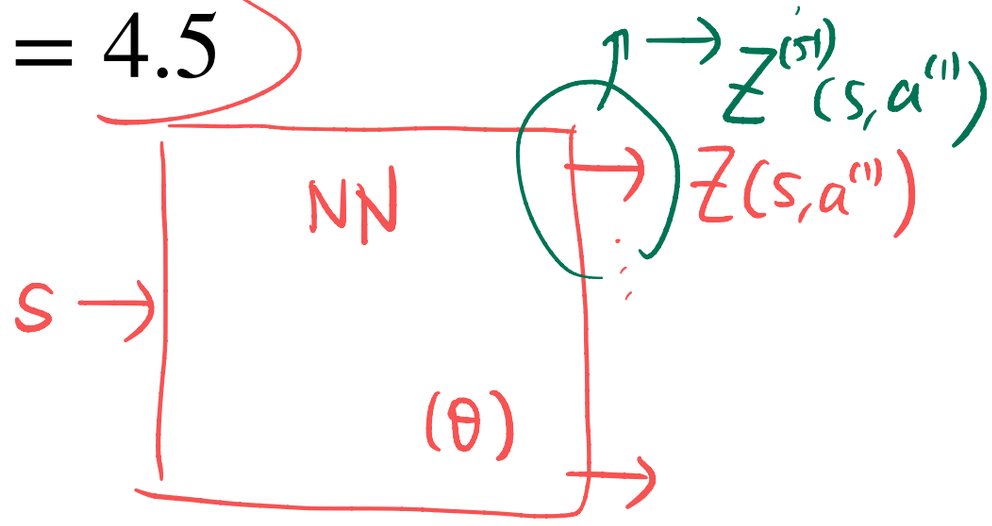
(C1) Categorical Distributions for $Z_\theta(s,a)$ (Cont.)

- Idea: Choose $V_{\max}, V_{\min}$ from preliminary experiments and select the number of atoms (denoted by N)

$$\Delta = \frac{V_{\max} - V_{\min}}{N-1}$$

- Example: 10 atoms with $V_{\min} = 0, V_{\max} = 4.5$





- Remark: C51 suggests using 51 atoms (why?)

- To achieve categorical distributions, one simple approach is to use

Softmax parameterization: $P(Z_\theta(s,a) = z_i) = \frac{e^{f_{\theta_i}(s,a)}}{\sum_j e^{f_{\theta_j}(s,a)}}$

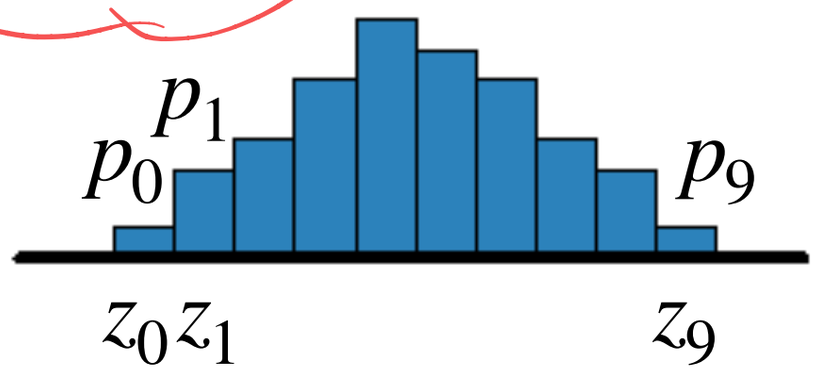
(C2) Mimicking B^* for Learning With Sample Transitions

$B^*Z = r + \gamma P^{\pi_{greedy}} Z$

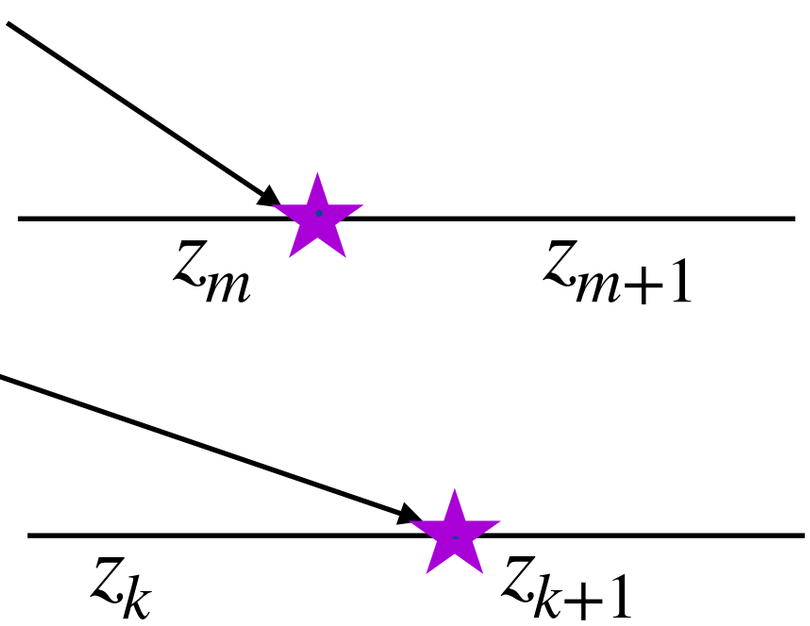
- Here we presume a greedy policy w.r.t Q function for B^*
- Question: Given only transitions (s, a, r, s'), how to enforce B^* to update $Z(s, a)$ on categorical distributions?

$$a^* = \arg\max_a Q(s', a) \qquad E[Z(s', a)]$$

$Z(s', a^*)$



$[r + \gamma z_0]_{V_{min}}^{V_{max}}$



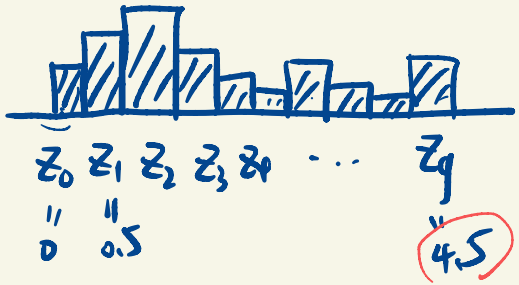
Q: What's the distribution of $r + \gamma Z(s', a^*)$?

- Question: Any issue?

Mismatch in atom positions

$V_{min} = 0$, $V_{max} = 4.5$

$Z(s', a^*)$

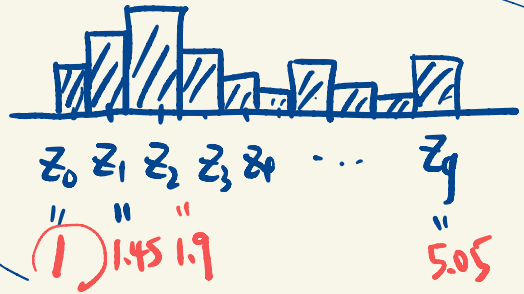


Suppose $\gamma = 1$.

$r + Z(s', a^*)$

$r + \gamma Z(s', a^*)$

$\gamma = 0.9$

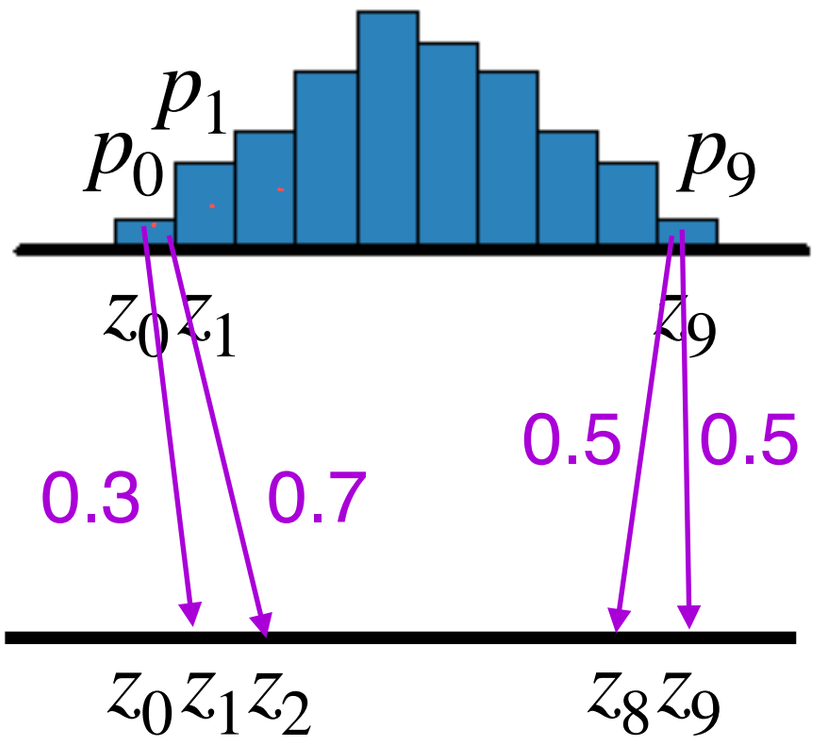


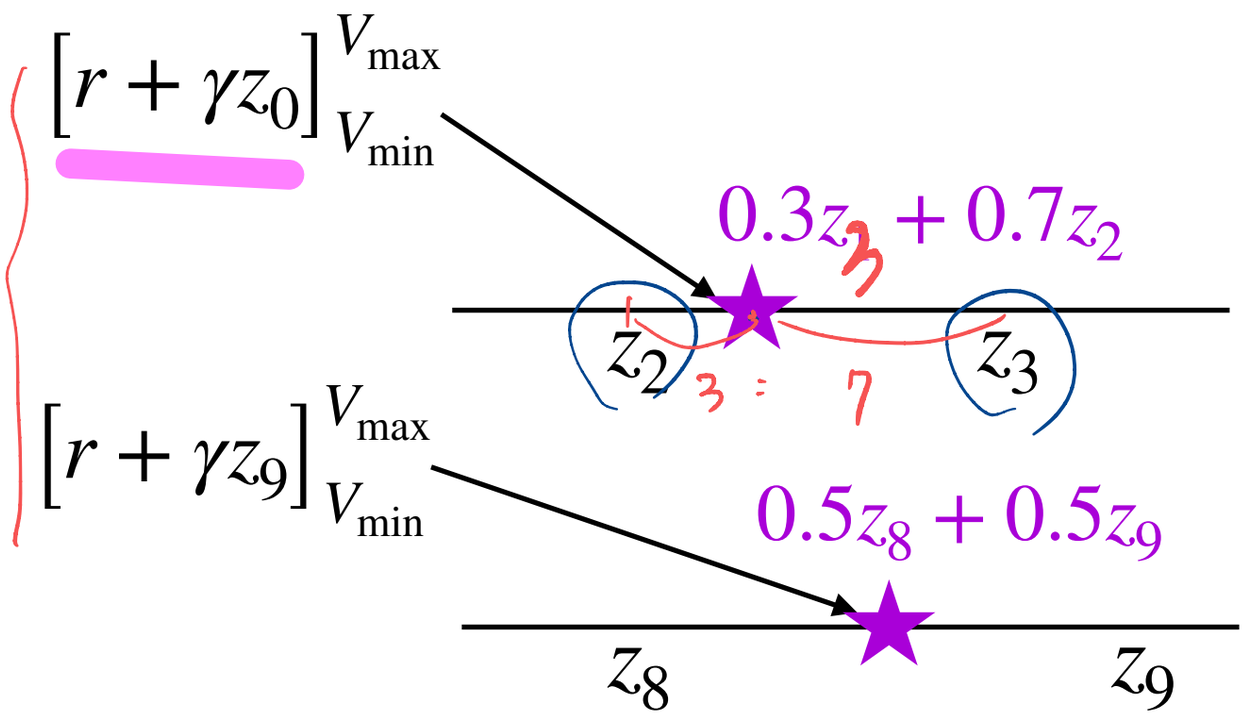
(C3) Cramer Projection Φ for Support Mismatch Caused by $B^*Z_\theta(s,a)$

- Issue: $\left[r+\gamma z_i\right]_{V_{\min}}^{V_{\max}}$ almost always leads to support mismatch
- Idea: Projection onto the set of atoms
- Example:

$$a^* = \arg\max_a Q(s',a)$$

$Z(s',a^*)$





(distribute probability mass to neighboring atoms ≡ projection Φ)

$$Z(s,a) = [\Phi B^* Z](s,a)$$

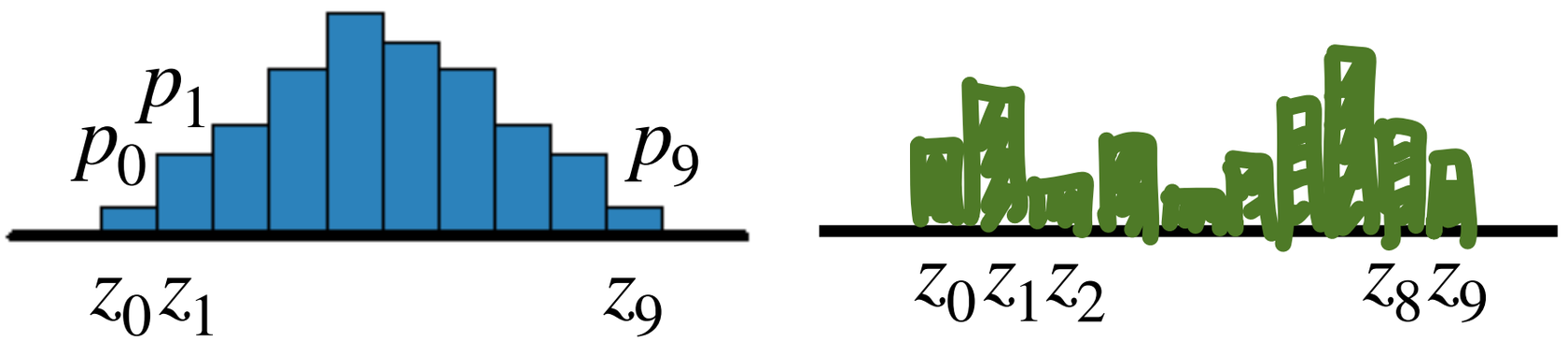
(C4) Update θ by Minimizing

$$L(s, a, r, s'; \theta) := D_{KL}(\Phi \hat{B}^* Z_{\bar{\theta}}(s, a) \| Z_\theta(s, a))$$

$$a^* = \arg\max_a Q(s', a)$$

$Z(s', a^*)$ $\quad$ $[\Phi B^* Z](s, a)$

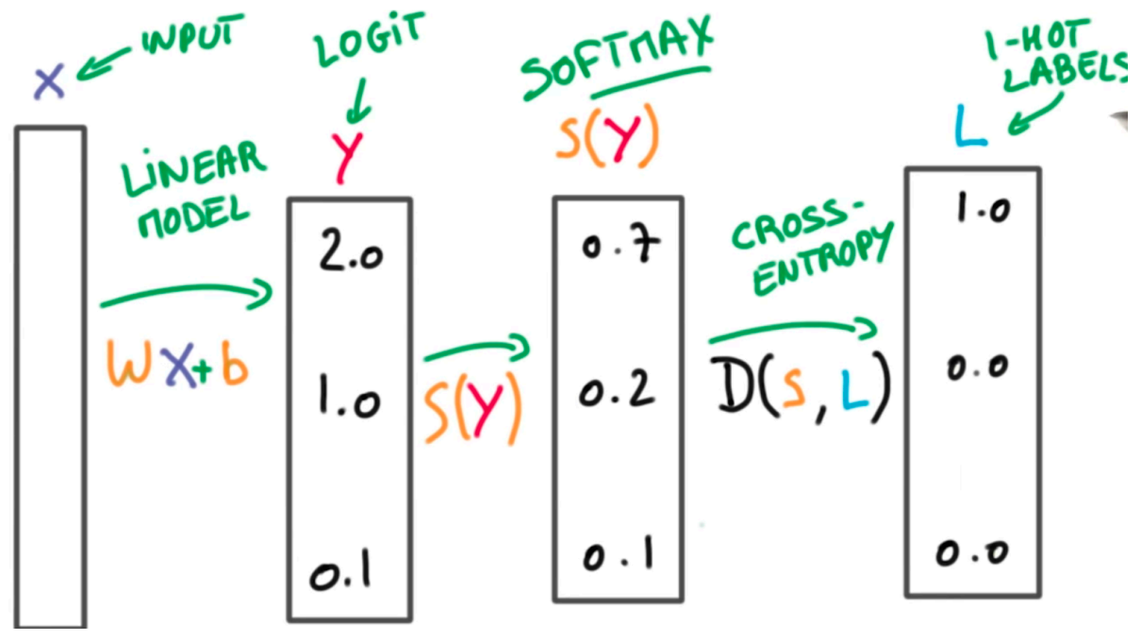
Goal: Update θ such that $Z_\theta(s, a)$ gets closer to $[\Phi B^* Z](s, a)$



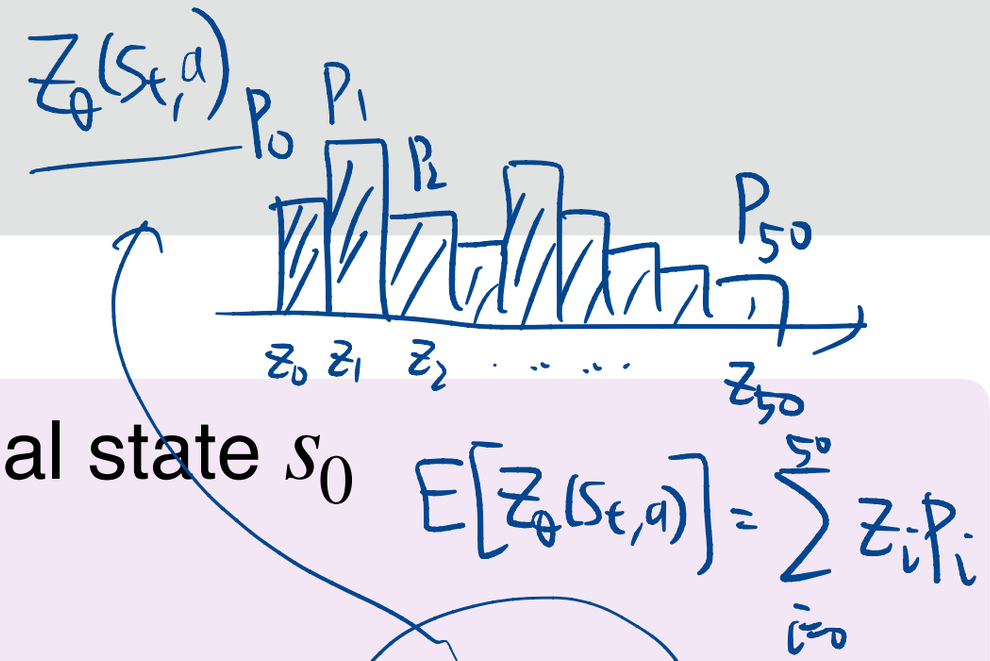
- Observation: Equivalent to multi-class classification

Multi-class classification (with NN)

$$L(s, a, r, s'; \theta) := D_{KL}(\Phi B^* Z_{\bar{\theta}}(s, a) \| Z_\theta(s, a))$$



C51 Algorithm (Formally)



Step 1: Initialize θ for $Z_\theta(s, a)$ and initial state s_0

Step 2: For each step $t = 0,1,2,\cdots$

Select a_t using ε-greedy w.r.t $Q(s_t, a) \equiv \mathbb{E}[Z_\theta(s_t, a)]$

Observe (r_{t+1}, s_{t+1}) and store $(s_t, a_t, r_{t+1}, s_{t+1})$ in the buffer

Draw a mini-batch of samples B from the replay buffer

Update θ by minimizing loss as follows:

$$\theta \leftarrow \theta - \alpha \nabla_\theta \sum_{(s,a,r,s') \in B} L_{C51}(s, a, r, s'; \theta)$$

Evaluation of Distributional DQN (C51) in Rainbow

C51 is a strong enhancement to vanilla DQN

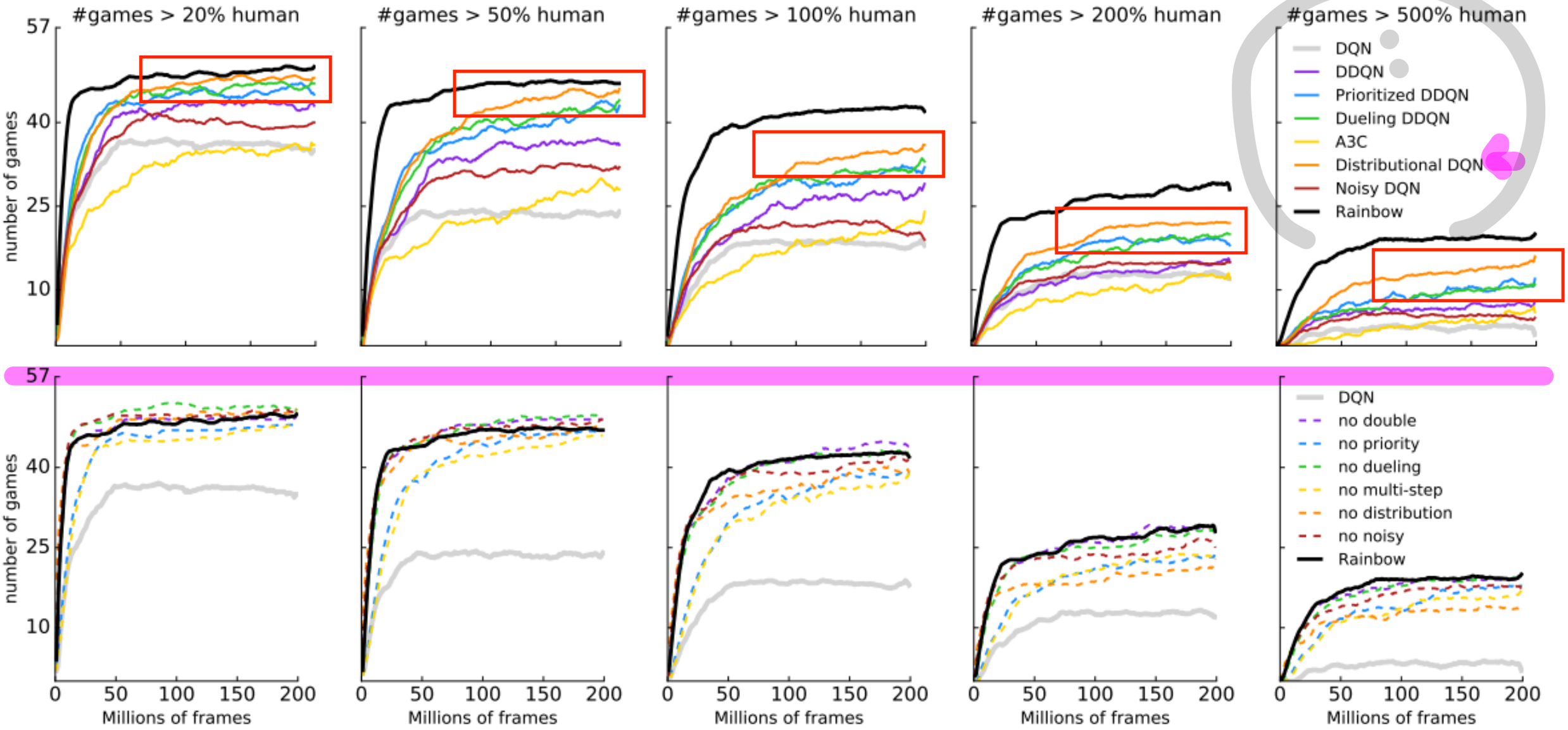


Figure 2: Each plot shows, for several agents, the number of games where they have achieved at least a given fraction of human performance, as a function of time. From left to right we consider the 20%, 50%, 100%, 200% and 500% thresholds. On the first row we compare Rainbow to the baselines. On the second row we compare Rainbow to its ablations.

Hessel et al., Rainbow: Combining Improvements in Deep Reinforcement Learning, AAAI 2018

Issues With C51

- In C51, $Z_\theta(s, a)$ is approximated by a categorical distribution
- Question: Any issues with C51?

Issue 1: Need to pre-specify bounds on the support (while the range of total return may vary greatly across states)

Issue 2: Require the projection Φ due to support mismatch

- Question: Any other way to express $Z(s, a)$?

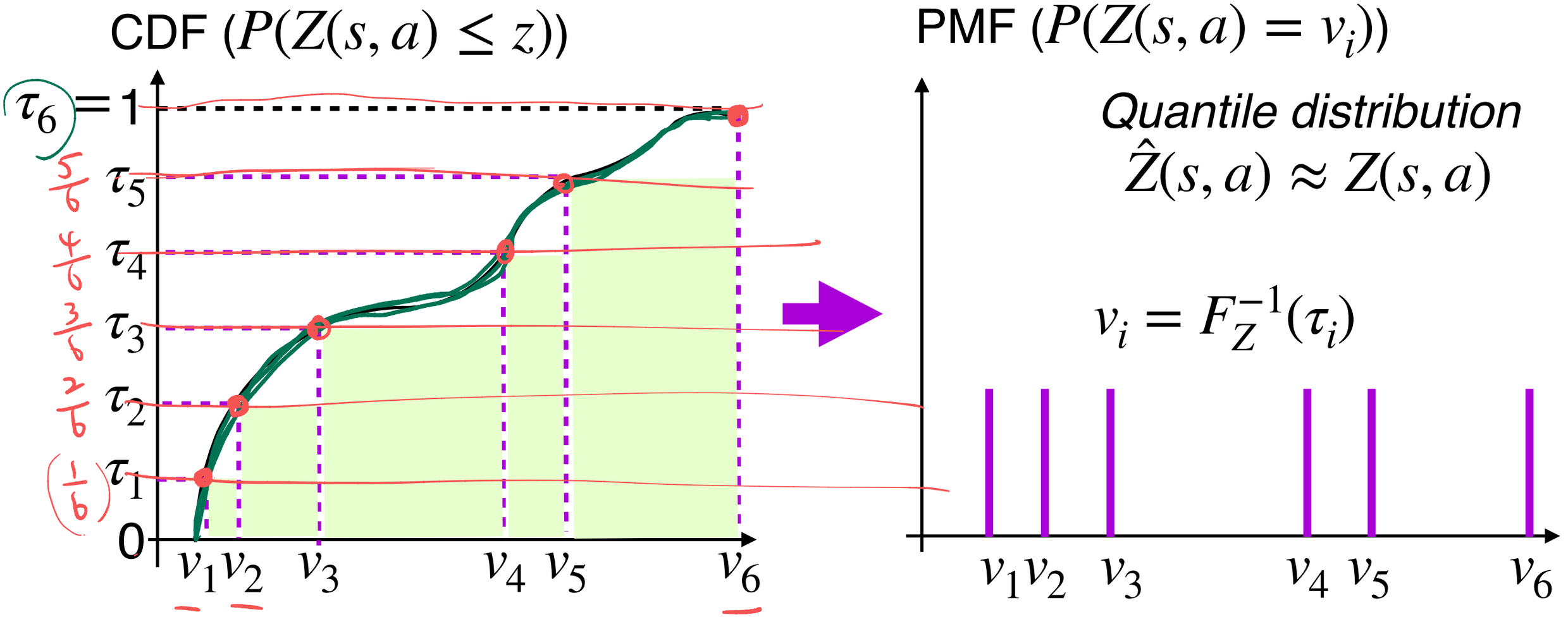
  Express $Z(s, a)$ using CDF (instead PMF or PDF)

QR-DQN

Dabney et al., Distributional Reinforcement Learning with Quantile Regression, AAAI 2018

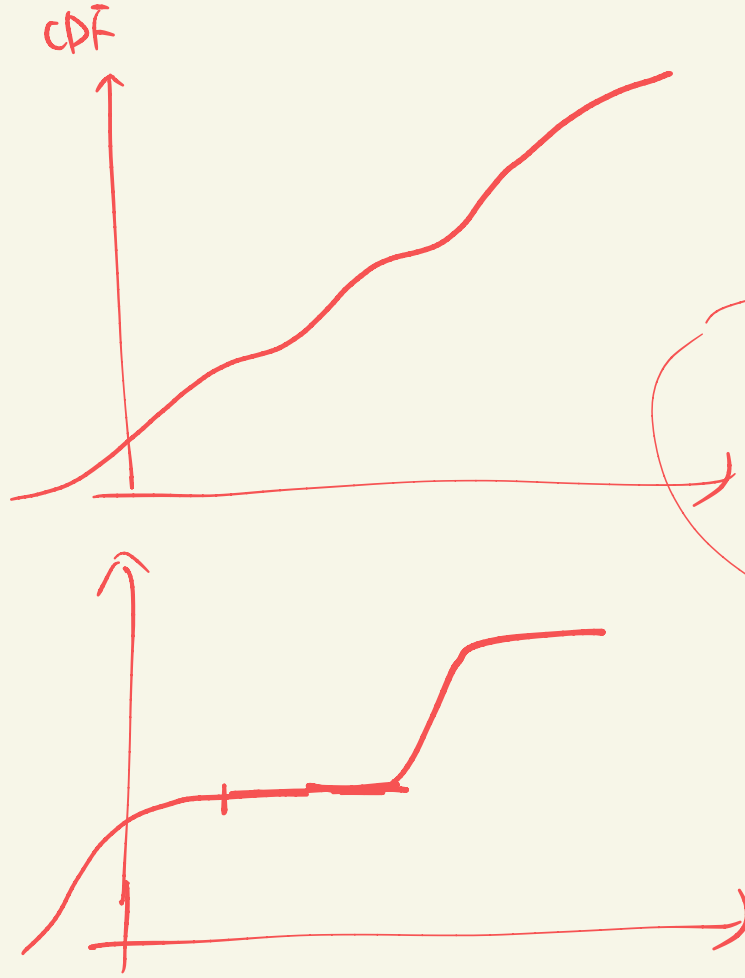
Quantile-Based Parametrization of $Z(s, a)$

- **Idea: Express $Z(s, a)$ using CDF (instead PDF)**



- Quantile function: $F_Z^{-1}(\tau) := \inf\{z : P(Z \leq z) \geq \tau\}$

$\tau \in [0, 1]$



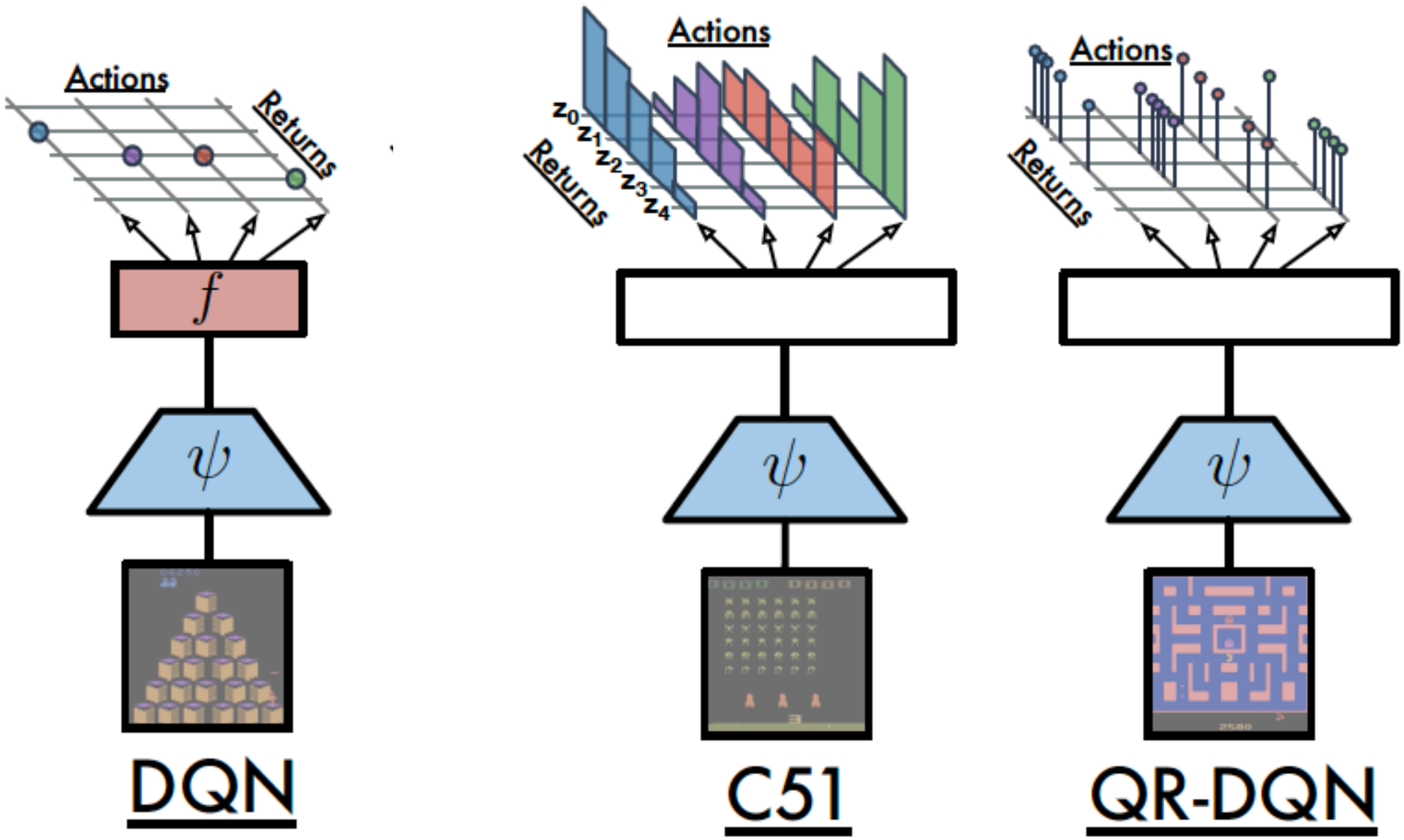
X, Y

$P(X=x, Y=y) = P(X=x) \cdot P(Y=y)$

$P(X=x, Y=y \mid E)$

$= P(X=x \mid E) \cdot P(Y=y \mid E)$

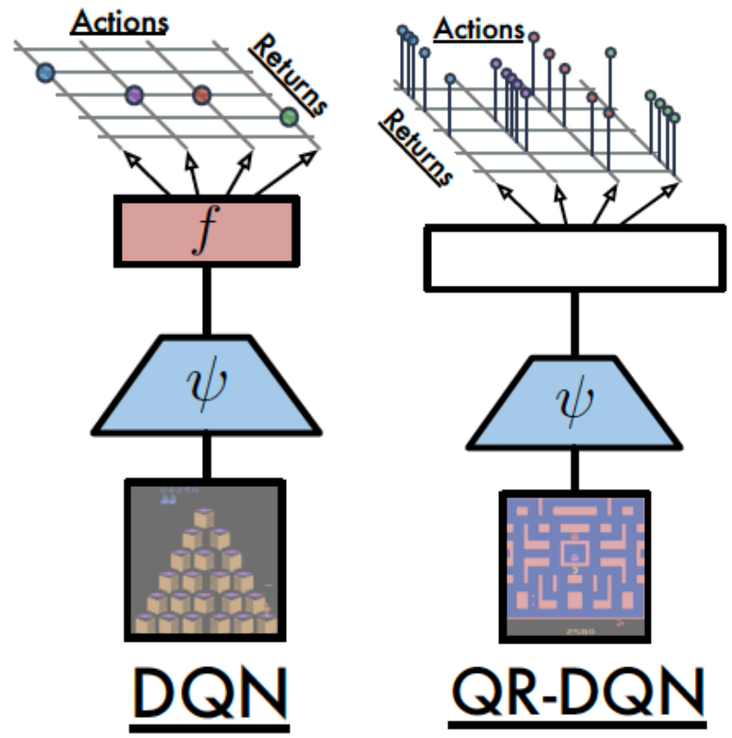
A Comparison of NN Architecture



QR-DQN: Another Popular Distributional DQN

- Q1: How to express $Z(s,a)$?

(D1) **Quantile** distributions for $Z_\theta(s,a)$



- Q2: How to update $Z(s,a)$ during training?

(D2) Mimicking B^* for learning with sample transitions (s,a,r,s')

(D3) Minimize $L_{QR}(s,a,r,s';\theta) := D(B^*Z_{\bar{\theta}}(s,a)\|Z_\theta(s,a))$

Quantile Regression DQN (Formally)

Step 1: Initialize $Z_\theta(s, a)$ and initial state s_0

Step 2: For each step $t = 0,1,2,\cdots$

Select a_t using ε-greedy w.r.t $Q(s_t, a) \equiv \mathbb{E}[Z_\theta(s_t, a)]$

Observe (r_{t+1}, s_{t+1}) and store $(s_t, a_t, r_{t+1}, s_{t+1})$ in the buffer

Draw a mini-batch of samples B from the replay buffer

Update θ by minimizing QR loss as follows:

$$\theta \leftarrow \theta - \alpha \nabla_\theta \sum_{(s,a,r,s') \in B} L_{QRDQN}(s, a, r, s'; \theta)$$

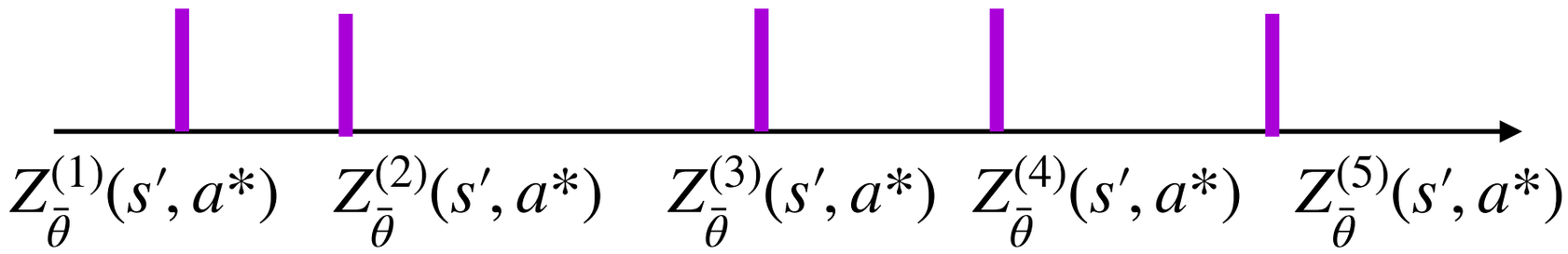
Under quantile distributions, $\mathbb{E}[Z_\theta(s, a)] = \sum_{i=1}^{N} \frac{1}{N} Z_\theta^{(i)}(s, a)$

(D2) Mimicking B^* for Learning With Sample Transitions

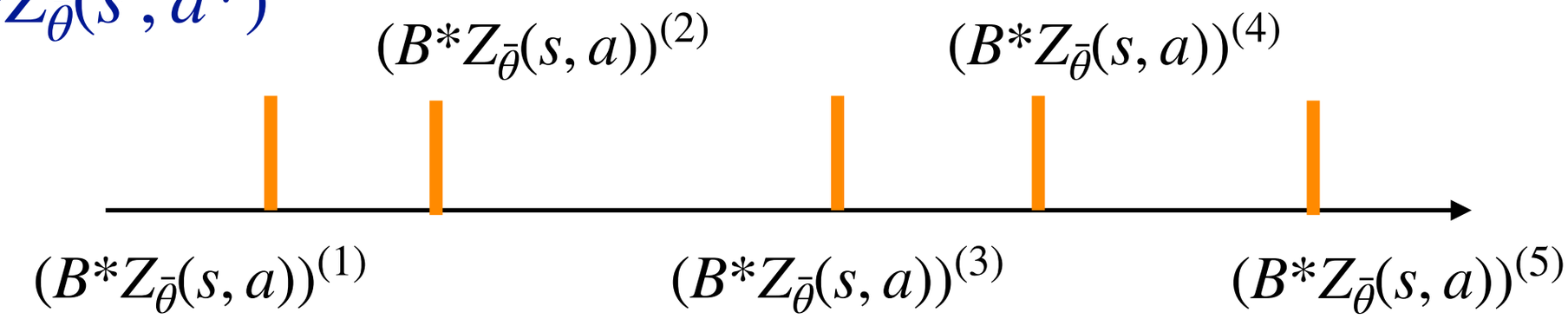
- Here we presume a greedy policy w.r.t Q function for B^*
- Question: Given only transitions (s, a, r, s'), how to enforce B^* to update $Z(s, a)$ on *quantile* distributions?

$$a^* = \arg\max_a Q(s', a) \equiv \arg\max_a \mathbb{E}[Z_{\bar{\theta}}(s', a)]$$

$Z_{\bar{\theta}}(s', a^*)$



$B^*Z_{\bar{\theta}}(s, a) = r + \gamma Z_{\bar{\theta}}(s', a^*)$



(D3) Loss Function

- We still need to choose a distance function $D(\,\cdot\|\cdot\,)$ in $L_{QRDQN}(s,a,r,s';\theta) := D(B^*Z_{\bar{\theta}}(s,a)\|Z_\theta(s,a))$

- There are many possibilities, e.g., total variation or KL divergence

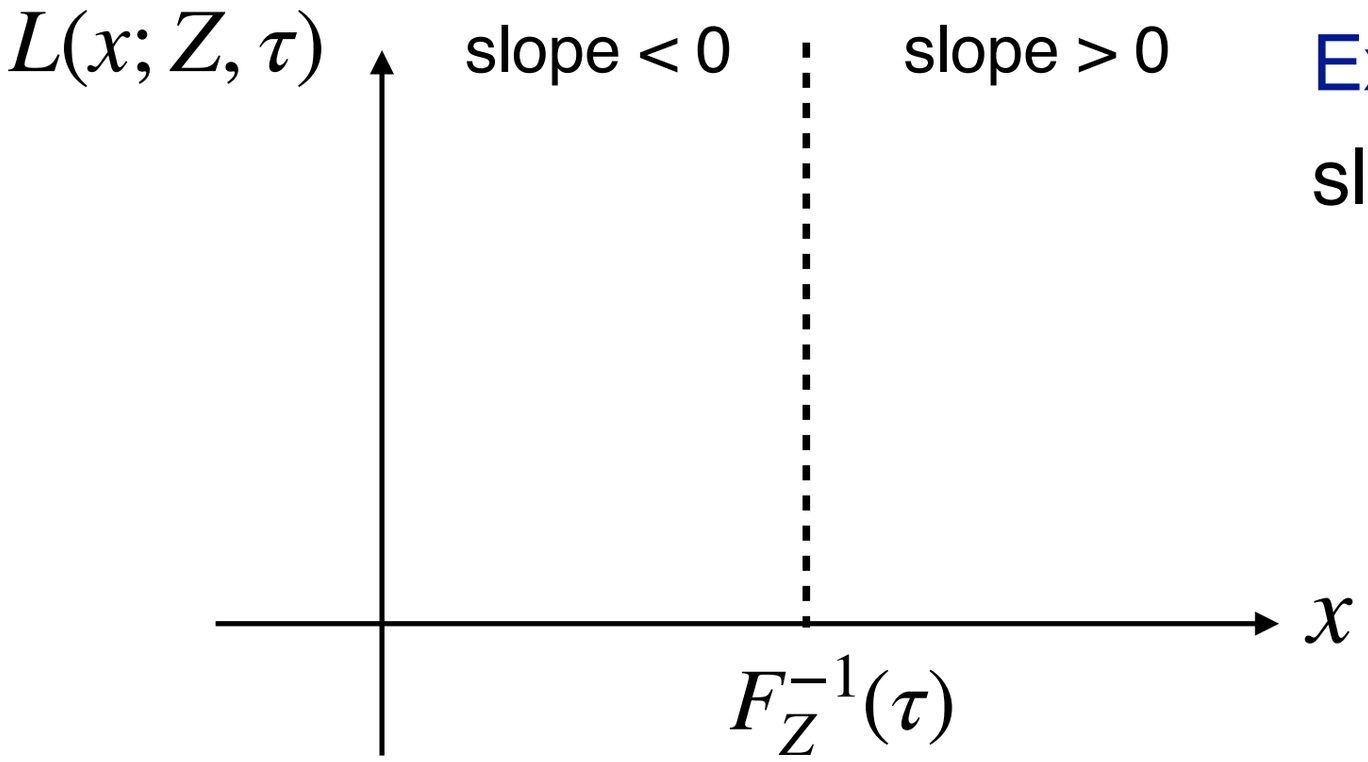
- QR-DQN uses the quantile regression loss
  - Motivation: Both $B^*Z_{\bar{\theta}}(s,a)$ and $Z_\theta(s,a)$ are quantile distributions

Quantile Regression Loss

- Idea: Finding a quantile $F_Z^{-1}(\tau)$ by minimizing loss $L(x; Z, \tau)$

$$F_Z^{-1}(\tau) = \arg\min_{x \in \mathbb{R}} L(x; Z, \tau)$$

- $L(x; Z, \tau)$ is *easy-to-optimize* when it is strictly convex



Example: Let $L(x; Z, \tau)$ have a slope of $F_Z(x) - \tau$, for all $x \in \mathbb{R}$

(Aka "quantile regression" loss)

Roger Koenker, Quantile Regression, 2005

The Quantile Regression Loss

- Given that the derivative of $L(x; Z, \tau)$ is $F_Z(x) - \tau$, we can recover the QR loss by integration

Quantile regression (QR) loss:

$$L_{QR}(x; Z, \tau) = (\tau - 1) \int_{-\infty}^{x} (z - x) dF_Z(z) + \tau \int_{x}^{\infty} (z - x) dF_Z(z)$$

(It is easy to verify that $\frac{d}{dx} L(x; Z, \tau) = F_Z(x) - \tau$ by the Leibniz integral rule)

An alternative expression of QR loss:

$$\rho_\tau(y) := y(\tau - \mathbb{I}\{y < 0\})$$

$$L_{QR}(x; Z, \tau) = E_Z[\rho_\tau(Z - x)]$$

Summary: Loss Function of QR-DQN

$$L_{QRDQN}(s, a, r, s'; \theta) := \sum_{i=1}^{N} L_{QR}(B^* Z_{\bar{\theta}}(s, a); Z_\theta(s, a), \tau_i)$$

$$= \sum_{i=1}^{N} \mathbb{E}_{z \sim B^* Z_{\bar{\theta}}(s,a)}[\rho_{\tau_i}(z - Z_\theta(s, a))]$$

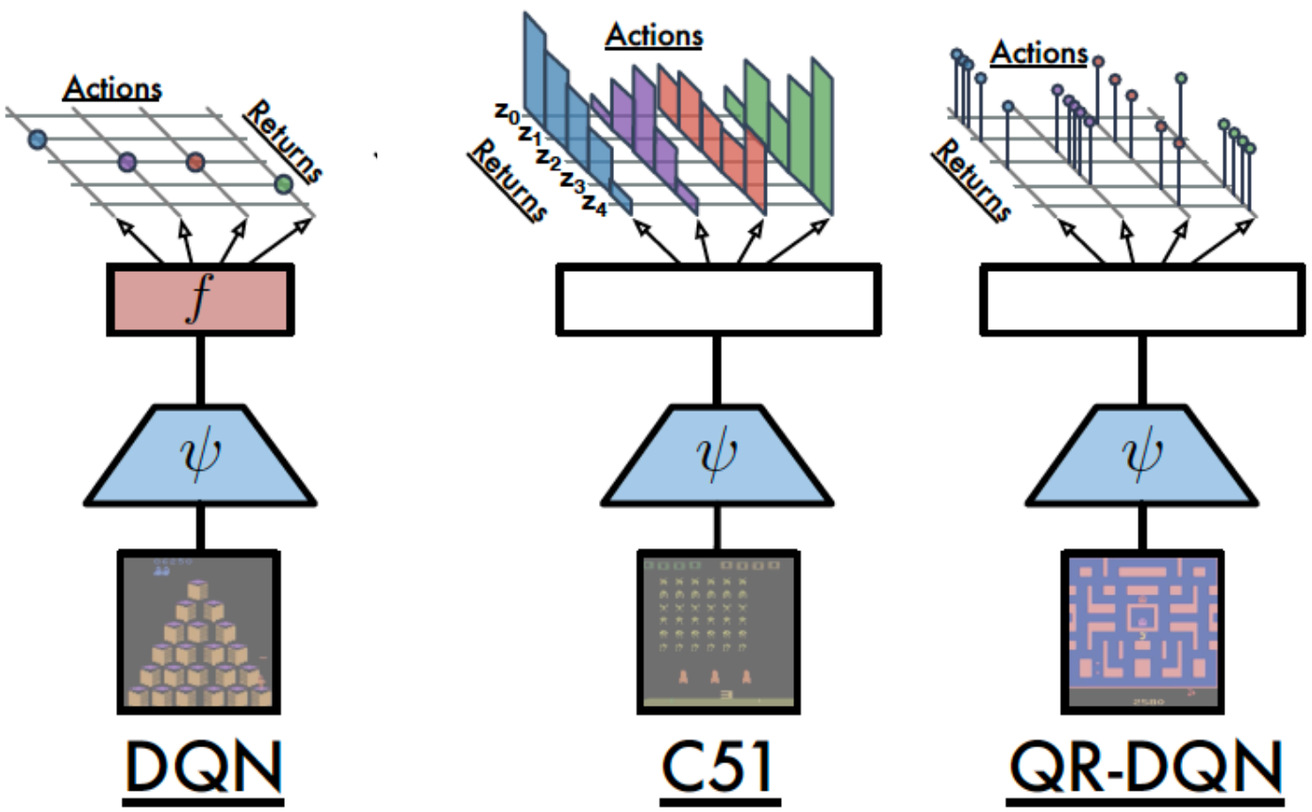
- Question: Is $L_{QRDQN}(s, a, r, s'; \theta)$ easy to compute during training?

Implicit Quantile Networks (IQN)

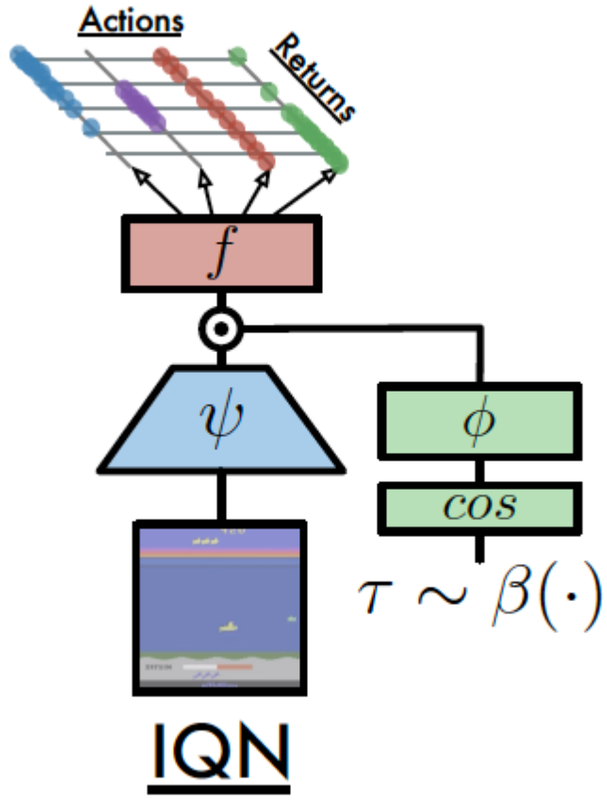
Dabney et al., Implicit Quantile Networks for Distributional Reinforcement Learning, ICML 2018

IQN: A Generative Approach to Distributional RL

- An illustrative comparison of distributional Q-learning methods



Distributional RL via explicitly expressing the distribution $Z(s, a)$



Distributional RL via a generative model for distribution $Z(s, a)$

➡ Need sufficiently many atoms or quantiles for an accurate representation of $Z(s, a)$

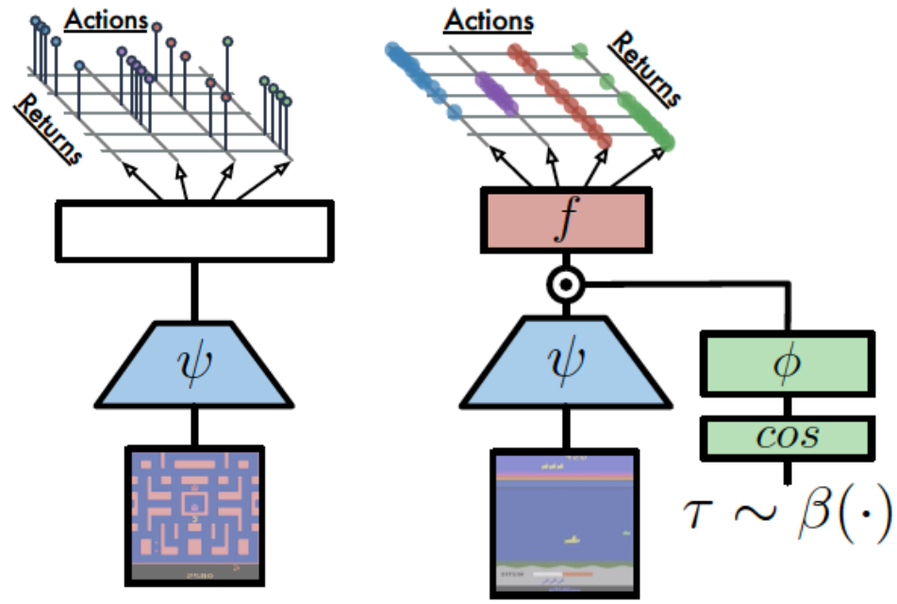
(Figure credit: Will Dabney)

Calculate QR Loss by *Sampling*

QR loss:

$$\rho_\tau(y) := y(\tau - \mathbb{I}\{y < 0\})$$

$$L(x; Z, \tau) = E_{z\sim Z}[\rho_\tau(z - x)]$$



- Recall QR-DQN:
  - The QR loss is calculated explicitly
  - $Z \Rightarrow$ target distribution induced by $\{\bar{\theta}_1, \cdots, \bar{\theta}_N\}$
- Question: Is there any other way to calculate the QR loss?

  Sampling!

  $$L(x; Z, \tau) \approx$$

- IQN **implicitly** parameterizes Z by constructing a generator for Z

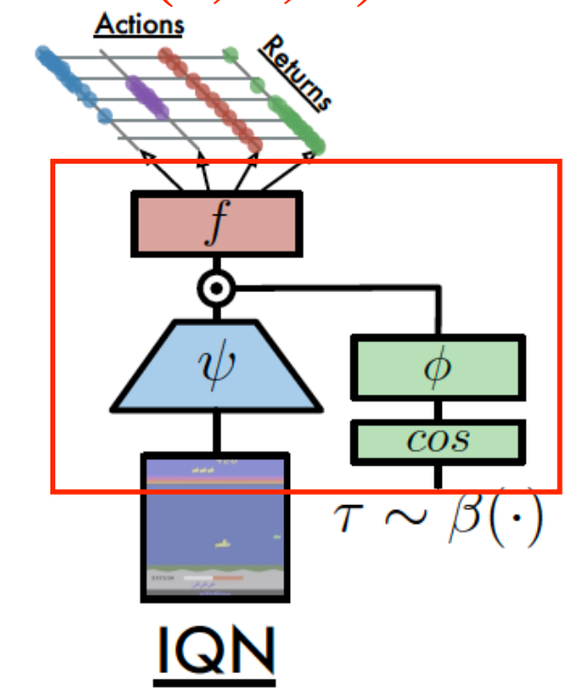
QR Loss and Inverse Transform Sampling

QR loss:

$$\rho_\tau(y) := y(\tau - \mathbb{I}\{y < 0\})$$

$$L(x; Z, \tau) = E_{z\sim Z}[\rho_\tau(z - x)] \approx \frac{1}{K}\sum_{k=1}^{K} \rho_\tau(z_k - x)$$
$$(z_1, \cdots, z_K \sim Z)$$

$\text{IQN}(s, \tau; a) \equiv F^{-1}$



Inverse Transform Sampling (ITS): Generate <u>any</u> random variable with CDF F from a uniform random variable

1. Generate a random variable $U \sim \text{Unif}(0,1)$
2. Let $X = F^{-1}(U)$, where $F^{-1}(u) := \inf\{z : F(z) \geq u\}$

- ITS is essentially a generative approach!

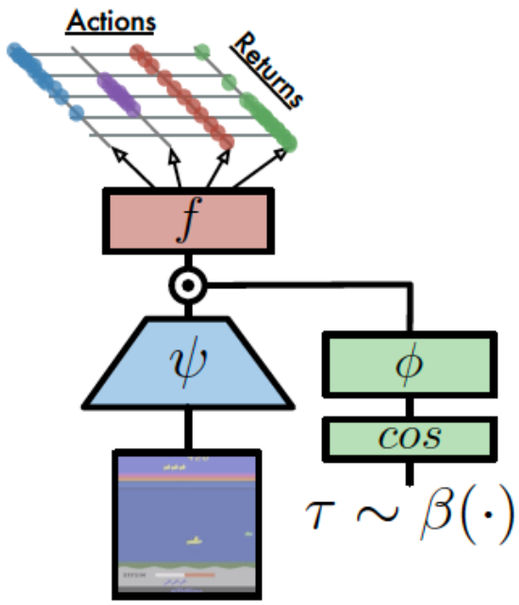
Calculating QR Loss in IQN

QR loss:

$$\rho_\tau(y) := y(\tau - \mathbb{I}\{y < 0\})$$

$$L(x; Z, \tau) = E_{z\sim Z}[\rho_\tau(z - x)] \approx \frac{1}{K}\sum_{k=1}^{K} \rho_\tau(z_k - x)$$

$$(z_1, \cdots, z_K \sim Z)$$



(Recall that Z corresponds to the target distribution in QR-DQN)

At each update, given (s, a, r, s'), for a given $\tau \in [0,1]$:

1. Draw $\tau'_1, \cdots, \tau'_K \sim \text{Unif}(0,1)$ ←a generative step!
2. Get $z_1, \cdots, z_K$ by $z_i = r + \gamma \cdot \overline{\text{IQN}}(s', a'; \tau'_i)$
3. QR loss in IQN = $\frac{1}{K}\sum_{i=1}^{K} \rho_\tau(z_i - \text{IQN}(s, a; \tau))$ (can be readily extended to multiple τ)