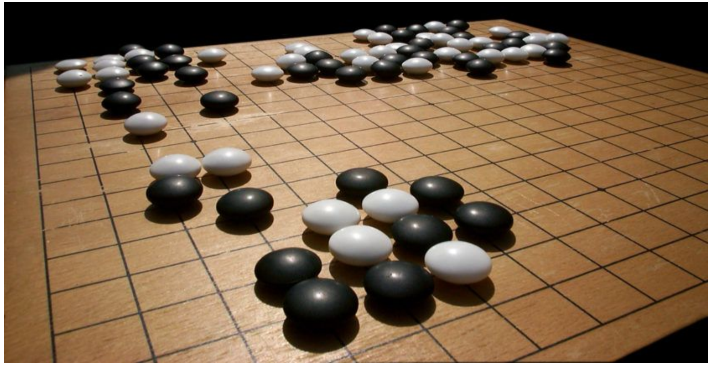
Distributional Reinforcement Learning

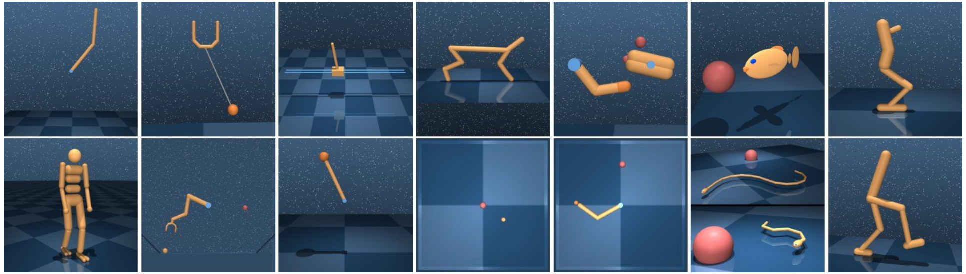



Marc Bellemare, Will Dabney, Georg Ostrovski, Mark Rowland, Rémi Munos

DeepMind

Deep RL is already a successful *empirical* research domain





Can we make it a *fundamental* research domain?

Related fundamental works:

- **RL side**: tabular, linear TD, ADP, sample complexity, ...
- **Deep learning side**: VC-dim, convergence, stability, robustness, ...

Nice theoretical results, but how much do they tell us about deepRL?

Can we make it a *fundamental* research domain?

Related fundamental works:

- **RL side**: tabular, linear TD, ADP, sample complexity, ...
- **Deep learning side**: VC-dim, convergence, stability, robustness, ...

Nice theoretical results, but how much do they tell us about deepRL?

What is specific about RL when combined with deep learning?

Distributional-RL

Shows interesting interactions between RL and deep-learning

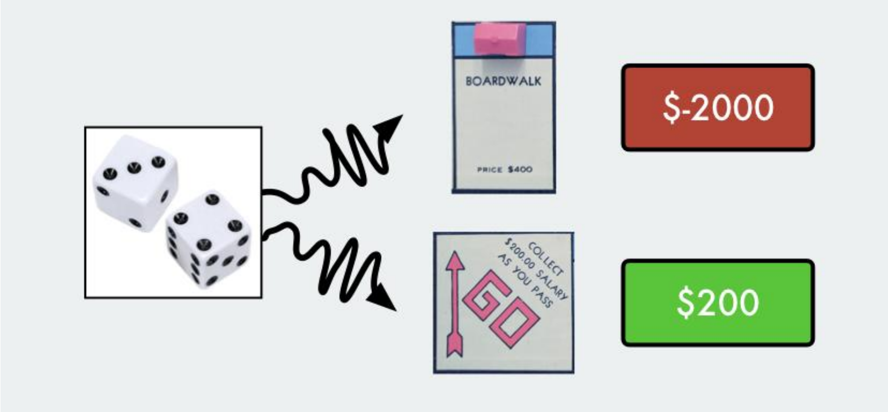
Outline:

- The idea of distributional-RL
- The theory
- How to represents distributions?
- Neural net implementation
- Results
- Why does this work?

Random immediate reward

Expected immediate reward

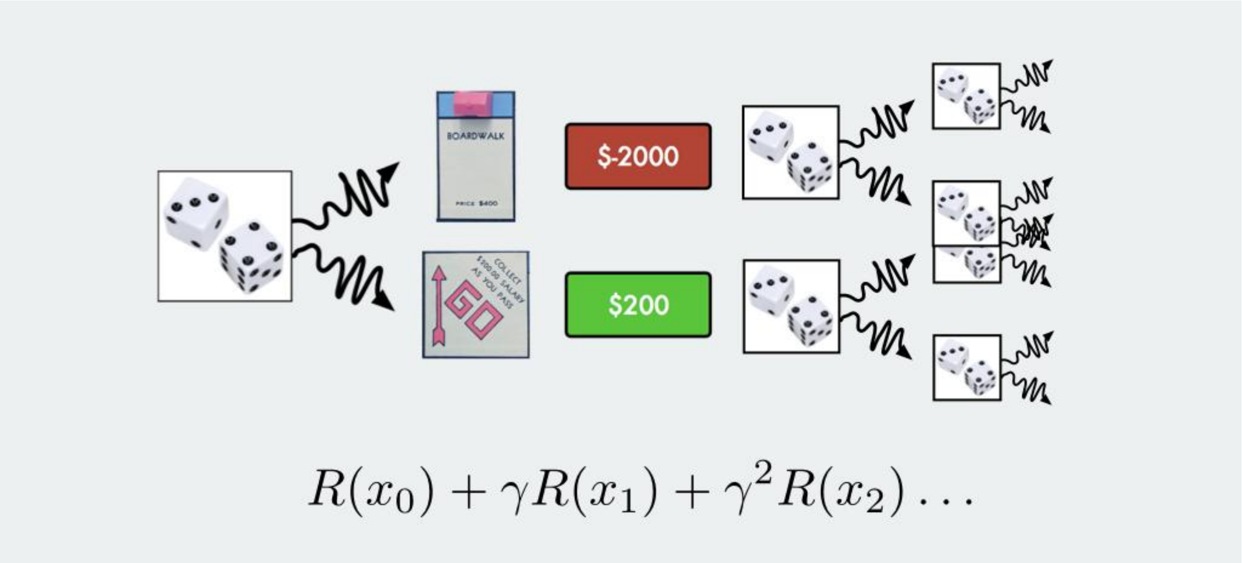
$$\mathbb{E}[R(x)] = \frac{1}{36} \times (-2000) + \frac{35}{36} \times (200) = 138.88$$



Random variable reward:

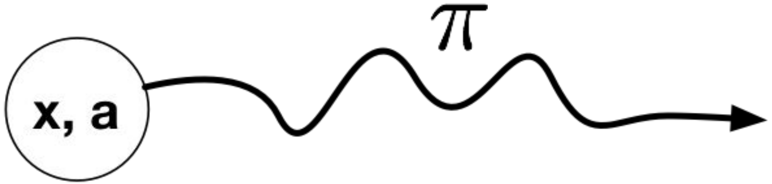
$$R(x) = \begin{cases} -2000 \text{ w.p. } 1/36 \\ \quad 200 \text{ w.p. } 35/36 \end{cases}$$

The return = sum of future discounted rewards



- Returns are often complex, multimodal
- Modelling the expected return hides this intrinsic randomness
- Model all possible returns!

The r.v. Return $Z^\pi(x, a)$

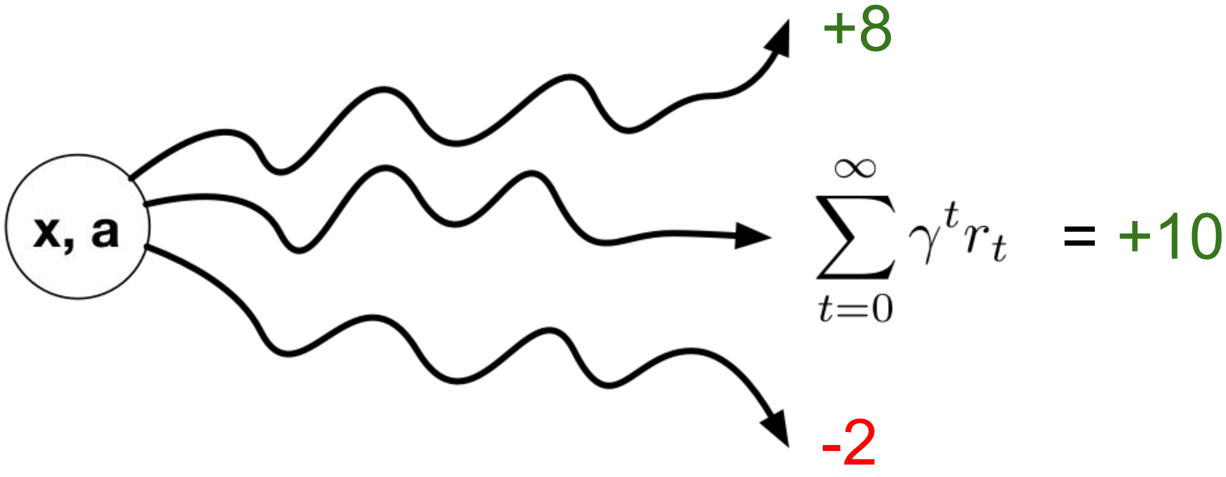


$$\sum_{t=0}^{\infty} \gamma^t r_t = +10$$

Captures intrinsic randomness from:

- Immediate rewards
- Stochastic dynamics
- Possibly stochastic policy

The r.v. Return $Z^{\pi}(x, a)$



$$Z^{\pi}(x, a) = \sum_{t \geq 0} \gamma^t r(x_t, a_t) \Big|_{x_0 = x, a_0 = a, \pi}$$

The expected Return

The value function $Q^\pi(x,a) = \mathbb{E}[Z^\pi(x,a)]$

Satisfies the Bellman equation

$$Q^\pi(x,a) = \mathbb{E}[r(x,a) + \gamma Q^\pi(x',a')]$$

$$\text{where } x' \sim p(\cdot|x,a) \text{ and } a' \sim \pi(\cdot|x')$$

Distributional Bellman equation?

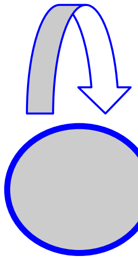
We would like to write a Bellman equation for the distributions:

$$Z^\pi(x,a) \stackrel{D}{=} R(x,a) + \gamma Z^\pi(x',a')$$

$$\text{where } x' \sim p(\cdot|x,a) \text{ and } a' \sim \pi(\cdot|x')$$

Does this equation make sense?

Example



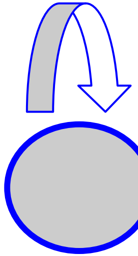
$$R = \begin{cases} 1 & \text{w.p. } 1/2 \\ 0 & \text{w.p. } 1/2 \end{cases}$$

Reward = Bernoulli (½), discount factor γ = ½

Bellman equation: $V = \frac{1}{2} + \frac{1}{2} V$, thus V = 1

Return $Z = \sum_{t \geq 0} 2^{-t} R_t$ Distribution?

Example



$$R = \begin{cases} 1 & \text{w.p. } 1/2 \\ 0 & \text{w.p. } 1/2 \end{cases}$$

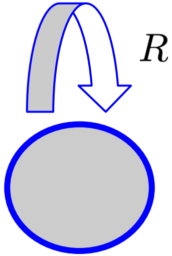
Reward = Bernoulli (½), discount factor γ = ½

Bellman equation: $V = \frac{1}{2} + \frac{1}{2}V$, thus V = 1

Return $Z = \sum_{t \geq 0} 2^{-t} R_t$ Distribution? $\mathcal{U}([0,2])$

(rewards = binary expansion of a real number)

Example



$$R = \begin{cases} 1 & \text{w.p. } 1/2 \\ 0 & \text{w.p. } 1/2 \end{cases}$$

Reward = Bernoulli (½), discount factor γ = ½

Bellman equation: $V = \frac{1}{2} + \frac{1}{2}V$, thus V = 1

Return $Z = \sum_{t\geq 0} 2^{-t} R_t$ Distribution? $\mathcal{U}([0,2])$

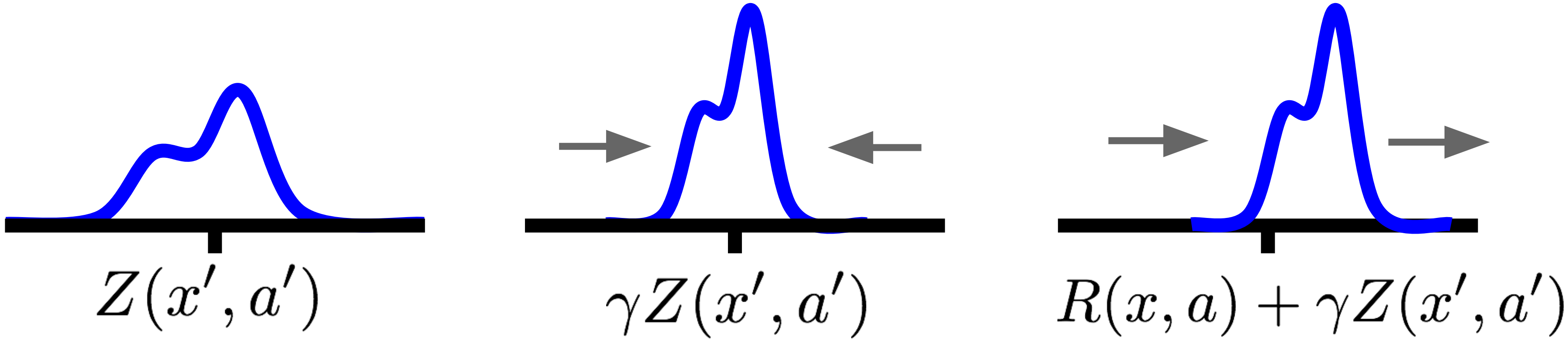
Distributional Bellman equation: $Z = \mathcal{B}(\frac{1}{2}) + \frac{1}{2}Z$

In terms of distribution:

$$\eta(z) = \frac{1}{2}\big(\delta(0) + \delta(1)\big) * 2\eta(2z)$$
$$= \eta(2z) + \eta(2(z-1))$$

Distributional Bellman operator

$$T^{\pi} Z(x,a) = R(x,a) + \gamma Z(x',a')$$



Does there exists a fixed point?

Properties

Theorem [Rowland et al., 2018]

$$\boxed{T^{\pi} \text{ is a contraction in Cramer metric}}$$

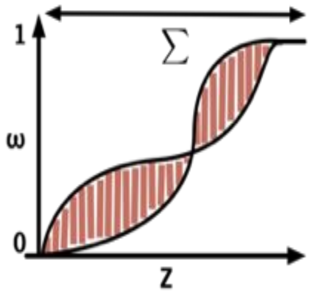
$$\ell_2(X,Y) = \left(\int_{\mathbb{R}} (F_X(t) - F_Y(t))^2 dt \right)^{1/2}$$

Theorem [Bellemare et al., 2017]

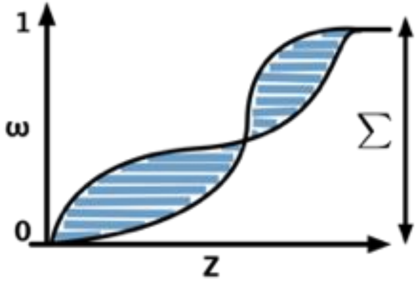
$$\boxed{T^{\pi} \text{ is a contraction in Wasserstein metric,}}$$

$$w_p(X,Y) = \left(\int_{\mathbb{R}} (F_X^{-1}(t) - F_Y^{-1}(t))^p dt \right)^{1/p}$$

(but not in KL neither in total variation)
Intuition: the size of the support shrinks.



Cramer



Wasserstein

Distributional dynamic programming

Thus T^{π} has a unique fixed point, and it is Z^{π}

Policy evaluation:
For a given policy π, iterate $Z \leftarrow T^{\pi} Z$ converges to Z^{π}

Distributional dynamic programming

Thus T^π has a unique fixed point, and it is Z^π

Policy evaluation:
For a given policy π, iterate $Z \leftarrow T^\pi Z$ converges to Z^π

Policy iteration:

- For current policy π_k, compute Z^{π_k}
- Improve policy

$$\pi_{k+1}(x) = \arg\max_a \mathbb{E}[Z^{\pi_k}(x, a)]$$

Does Z^{π_k} converge to the return distribution for the optimal policy?

Distributional Bellman optimality operator

$$TZ(x,a) \stackrel{D}{=} r(x,a) + \gamma Z(x', \pi_Z(x'))$$

$$\text{where } x' \sim p(\cdot|x,a) \text{ and } \pi_Z(x') = \arg\max_{a'} \mathbb{E}[Z(x',a')]$$

Is this operator a contraction mapping?

Distributional Bellman optimality operator

$$\mathcal{T}Z(x,a) \stackrel{D}{=} r(x,a) + \gamma Z(x', \pi_Z(x'))$$

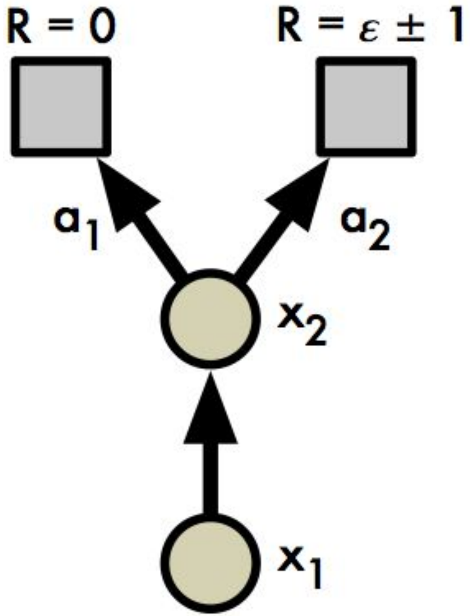
$$\text{where } x' \sim p(\cdot|x,a) \text{ and } \pi_Z(x') = \arg\max_{a'} \mathbb{E}[Z(x',a')]$$

Is this operator a contraction mapping?

No!

It’s not even continuous

The dist. opt. Bellman operator is not smooth



Consider distributions Z_ϵ

If $\varepsilon > 0$ we back up a bimodal distribution

If $\varepsilon < 0$ we back up a Dirac in 0

Thus the map $Z_\epsilon \mapsto TZ_\epsilon$ is not continuous

Distributional Bellman optimality operator

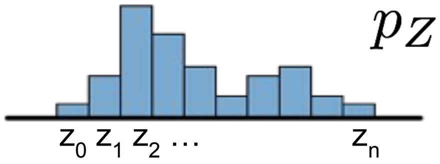
Theorem [Bellemare et al., 2017]

if the optimal policy is unique, then the iterates $Z_{k+1} \leftarrow TZ_k$ converge to Z^{π^*}

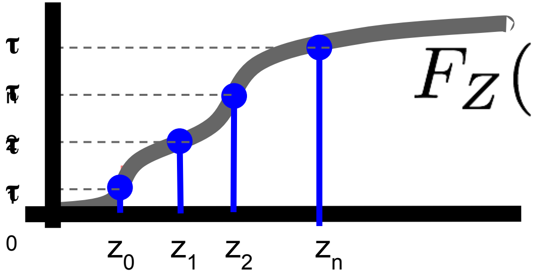
Intuition: The distributional Bellman operator preserves the mean, thus the mean will converge to the optimal policy π^* eventually. If the policy is unique, we revert to iterating T^{π^*}, which is a contraction.

How to represent distributions?

- Categorical



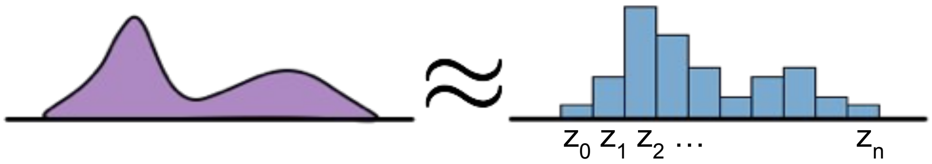
- Inverse CDF for specific quantile levels



- Parametric inverse CDF

$$\tau \mapsto F_Z^{-1}(\tau)$$

Categorical distributions

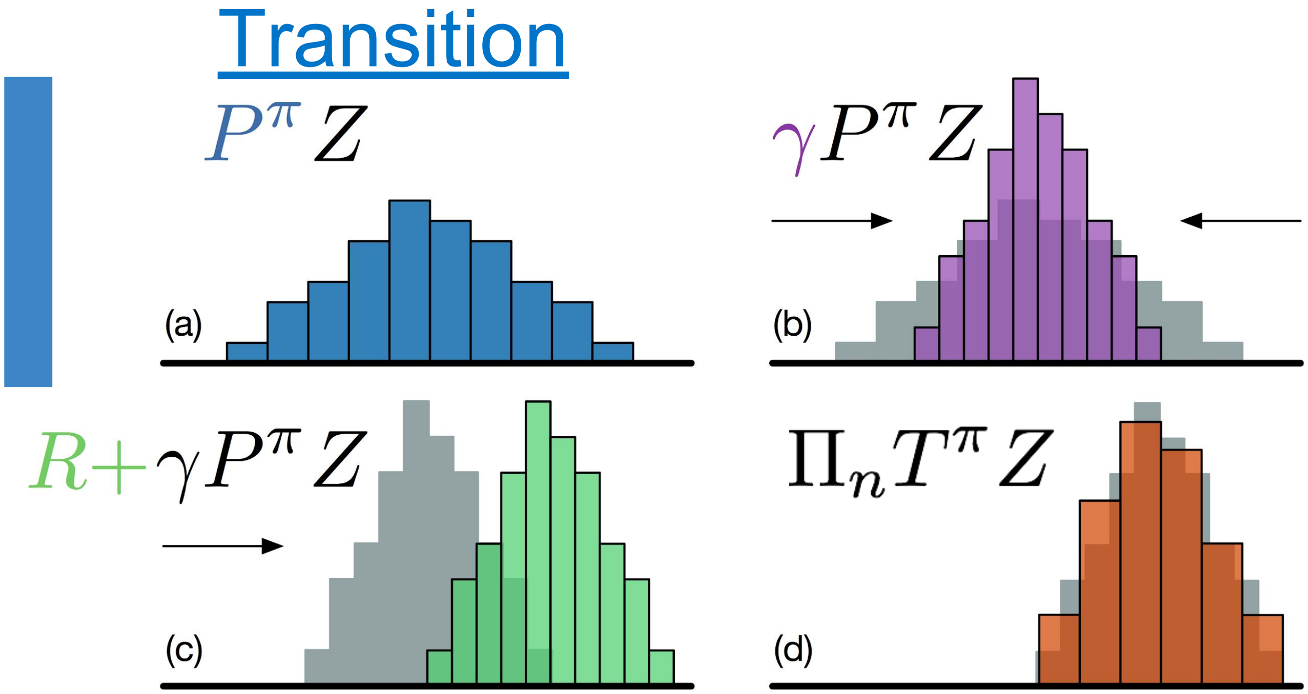


Distributions supported on a finite support $\{z_1, \ldots, z_n\}$

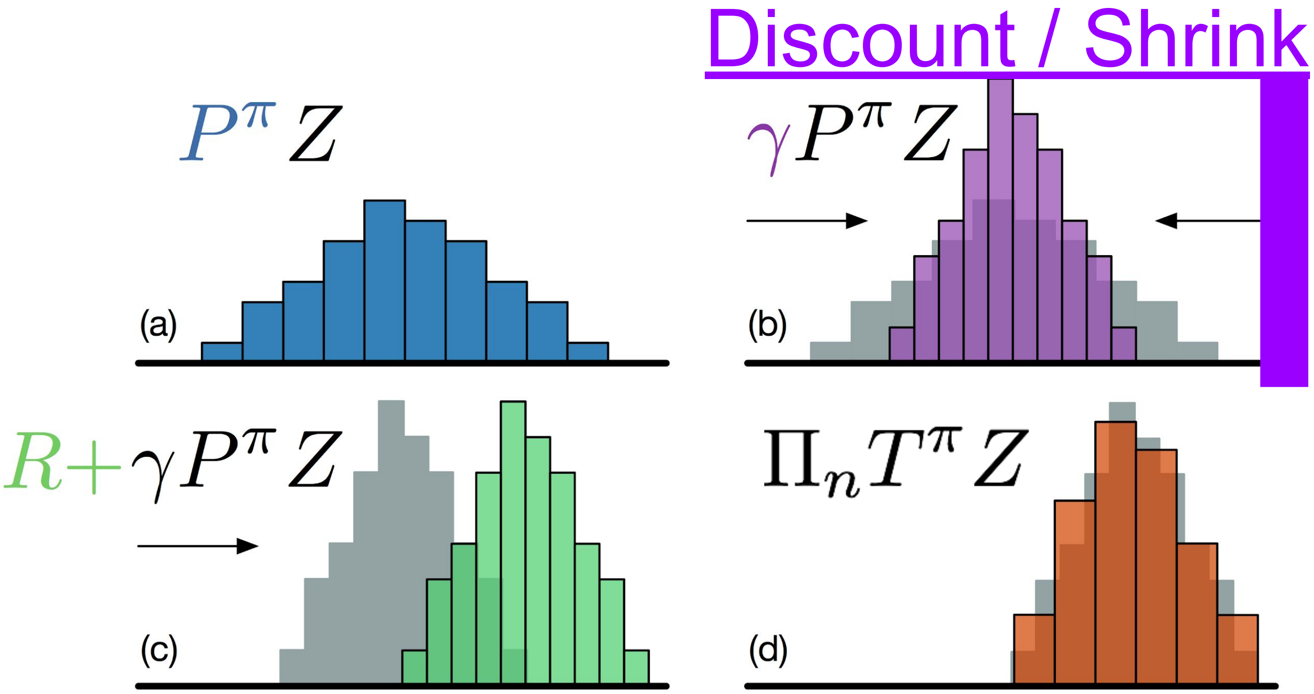
Discrete distribution $\{p_i(x, a)\}_{1 \leq i \leq n}$

$$Z(x, a) = \sum_i p_i(x, a)\delta_{z_i}$$

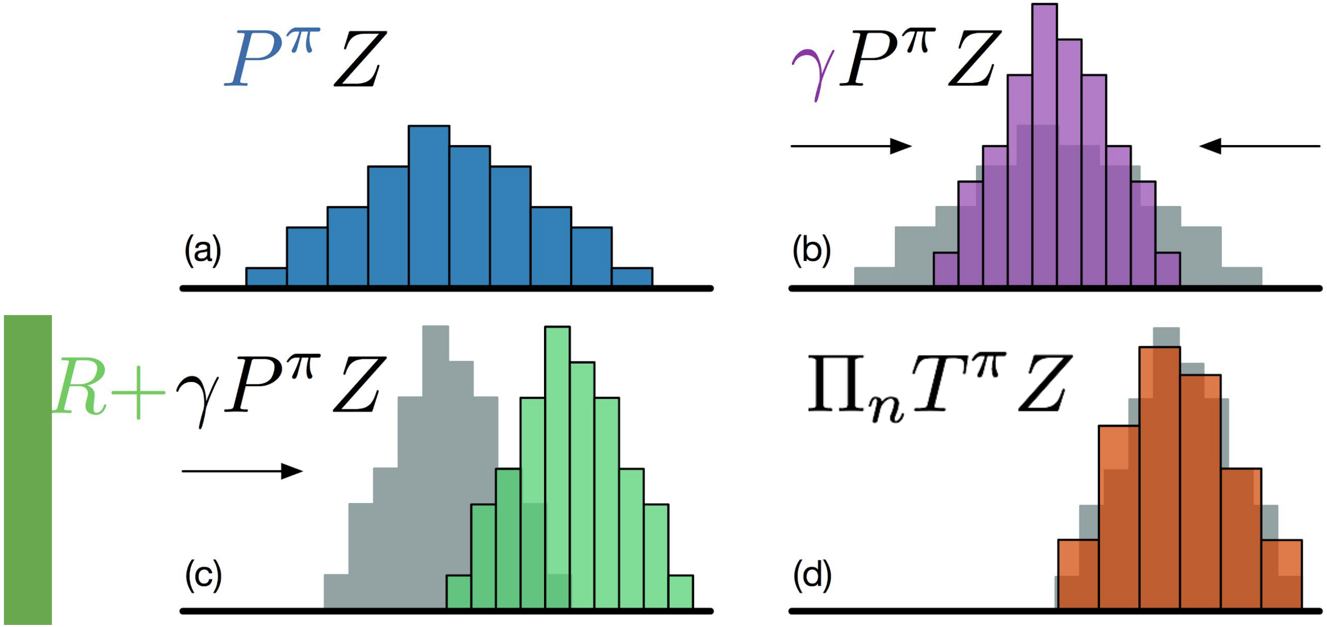
Projected Distributional Bellman Update



Projected Distributional Bellman Update

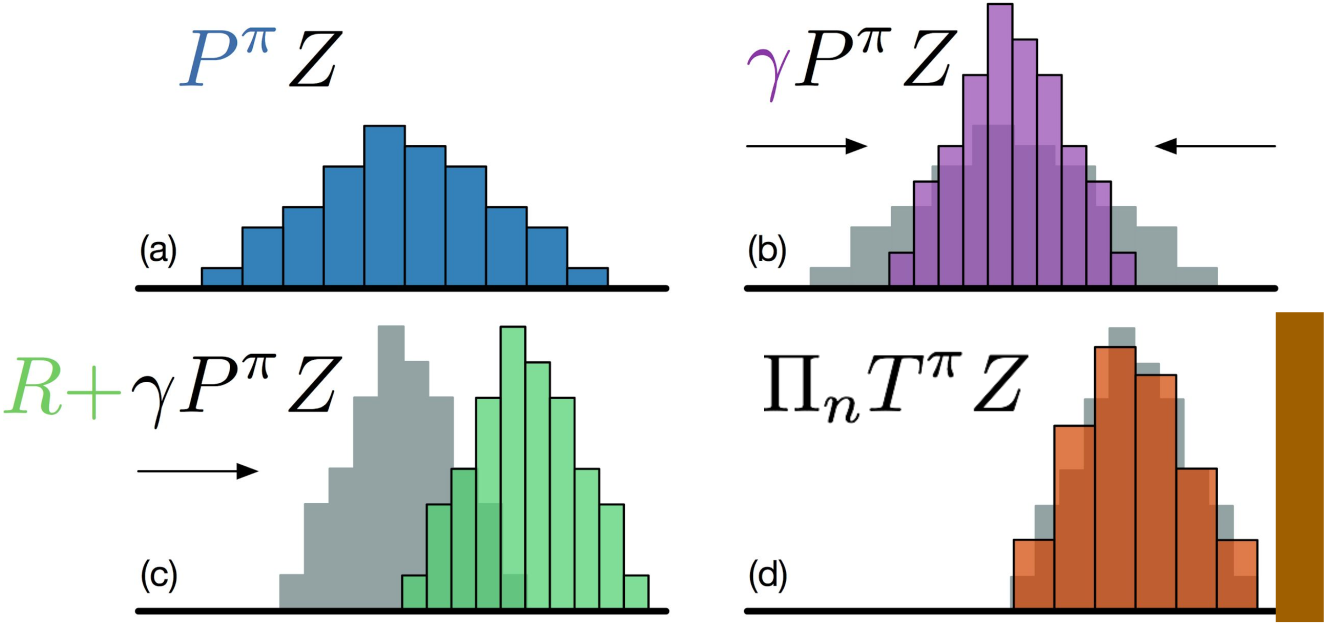


Projected Distributional Bellman Update



Reward / Shift

Projected Distributional Bellman Update



Fit / Project

Projected distributional Bellman operator

Let Π_n be the projection onto the support (piecewise linear interpolation)

Theorem: $\boxed{\Pi_n T^\pi \text{ is a contraction (in Cramer distance)}}$

Intuition: Π_n is a non-expansion (in Cramer distance).

Its fixed point Z_n can be computed by value iteration $Z \leftarrow \Pi_n T^\pi Z$

Theorem: $$\boxed{\ell_2^2(Z_n, Z^\pi) \leq \frac{1}{(1-\gamma)} \max_{1 \leq i < n} |z_{i+1} - z_i|}$$ [Rowland et al., 2018]

Projected distributional Bellman operator

Policy iteration: iterate

- Policy evaluation: $Z_k = \Pi_n T^{\pi_k} Z_k$
- Policy improvement: $\pi_{k+1}(x) = \arg\max_a \mathbb{E}[Z^{\pi_k}(x, a)]$

Theorem:

Assume there is a unique optimal policy.
Z_k converges to $Z_n^{\pi^*}$, whose greedy policy is optimal.

Categorical distributional Q-learning

Observe transition samples $x_t, a_t \xrightarrow{r_t} x_{t+1}$

Update:

$$Z(x_t, a_t) = (1 - \alpha_t) Z(x_t, a_t) + \alpha_t \Pi_C (r_t + \gamma Z(x_{t+1}, \pi_Z(x_{t+1}))$$

Theorem

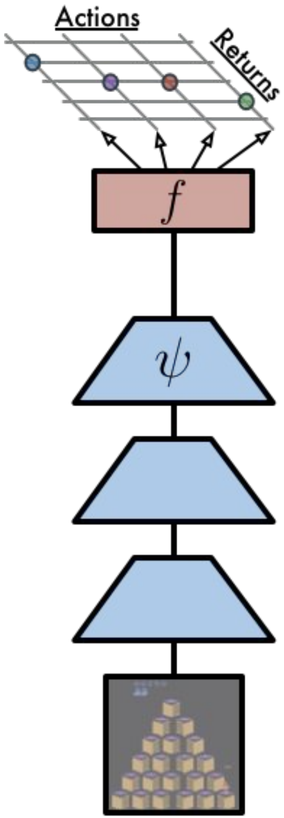
Under the same assumption as for Q-learning, assume there is a unique optimal policy π^*, then $Z \to Z_n^{\pi^*}$ and the resulting policy is optimal.

[Rowland et al., 2018]

DeepRL implementation

DQN

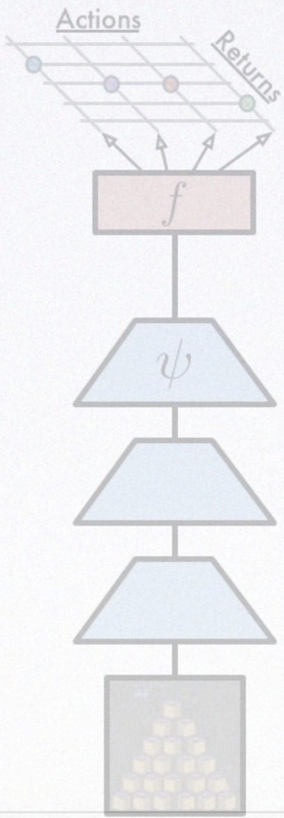
[Mnih et al., 2013]

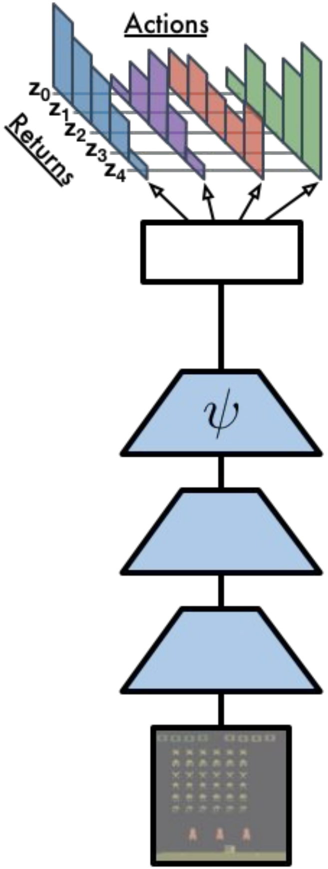


DQN

C51

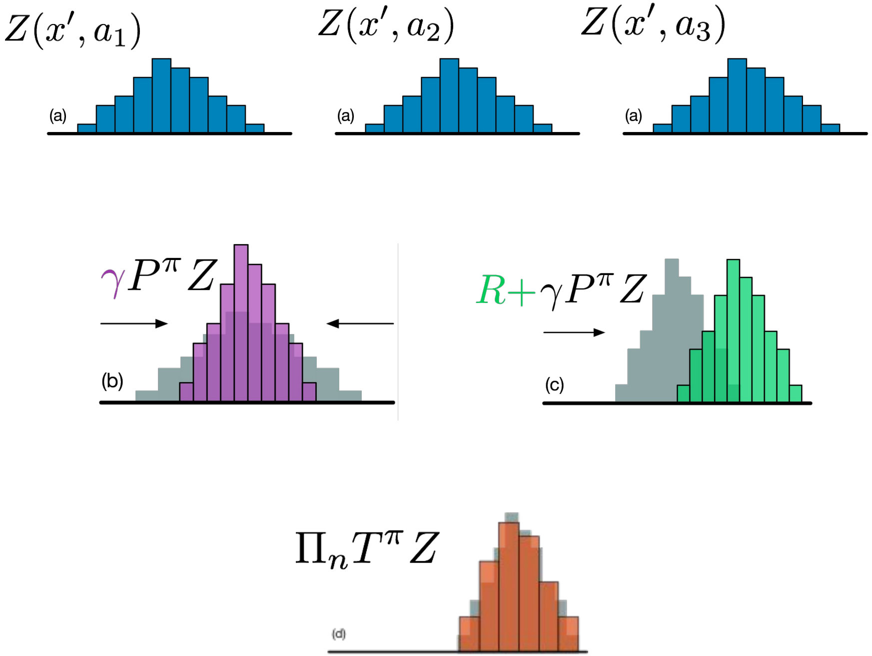
[Bellemare et al., 2017]



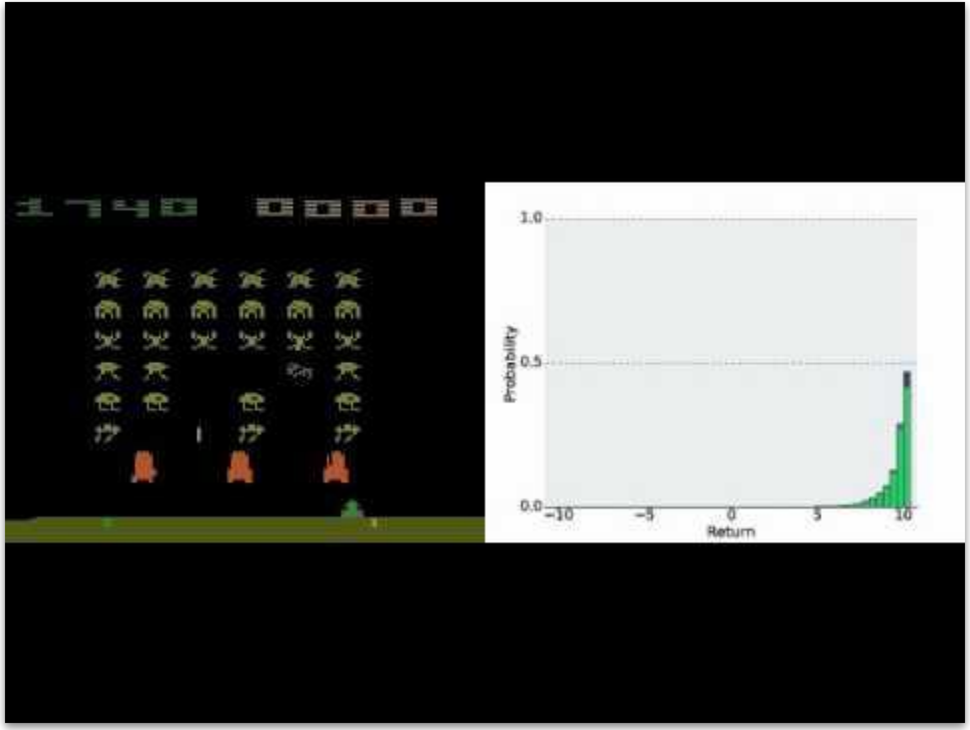


C51 (categorical distributional DQN)

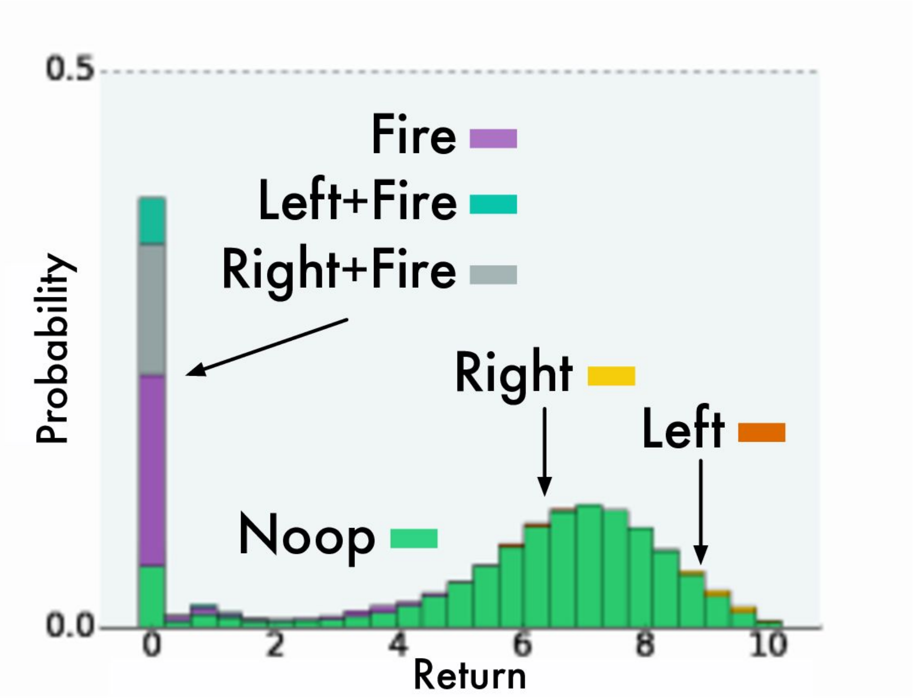
1. Transition x, a → x'
2. Select best action at x'
3. Compute Bellman backup
4. Project onto support
5. Update toward projection (e.g., by minimize a kl-loss)



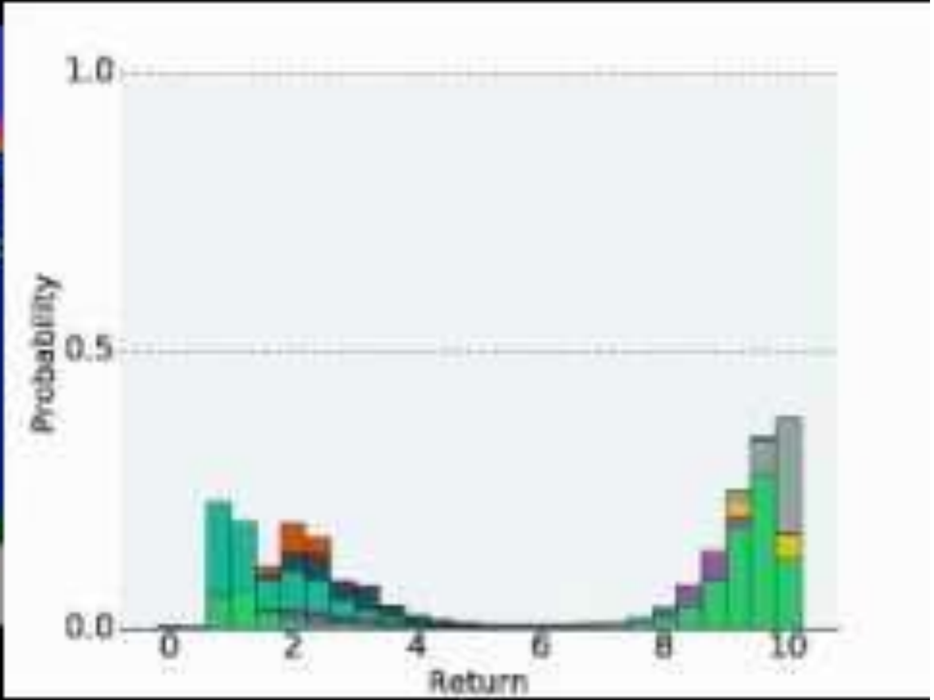
Categorical DQN

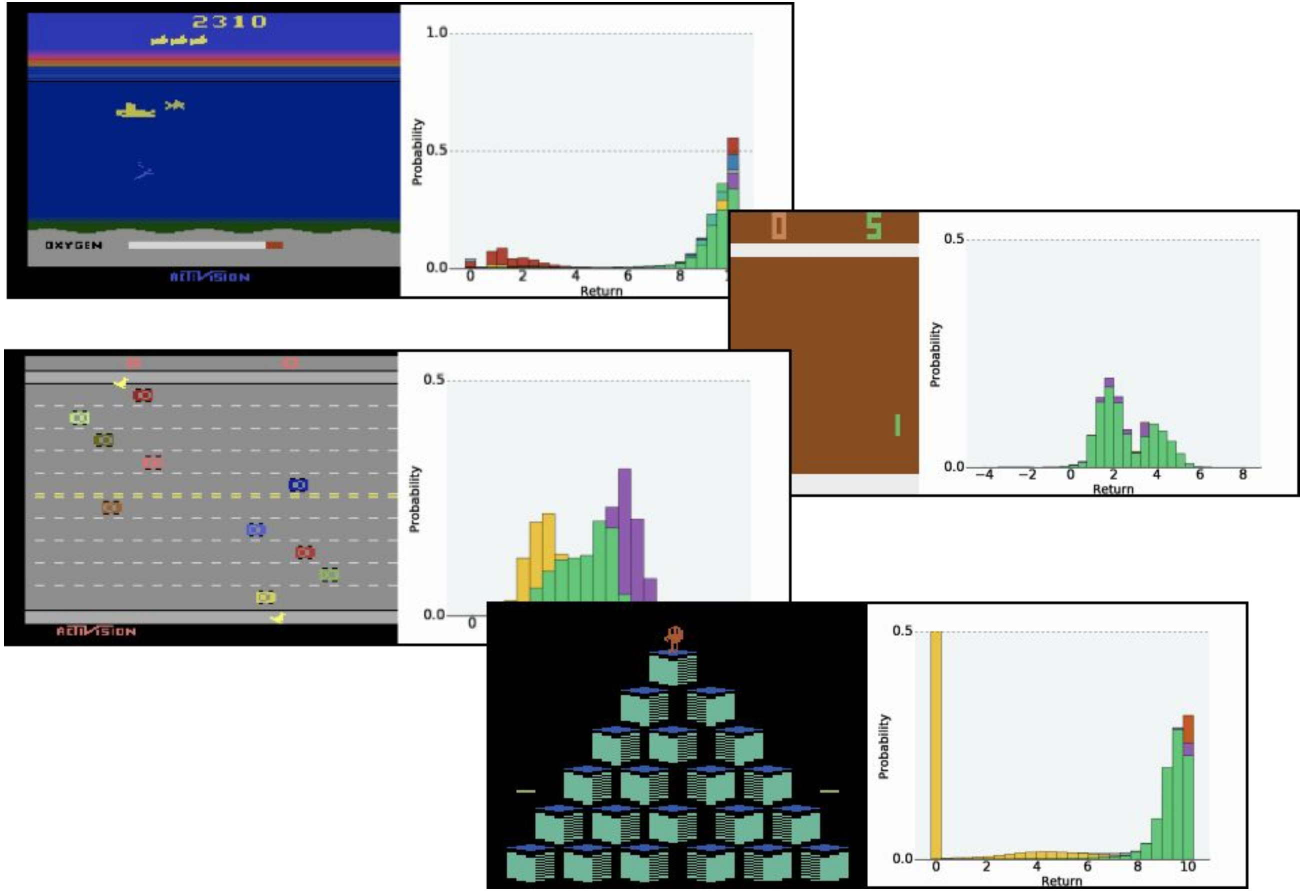


Randomness from future choices

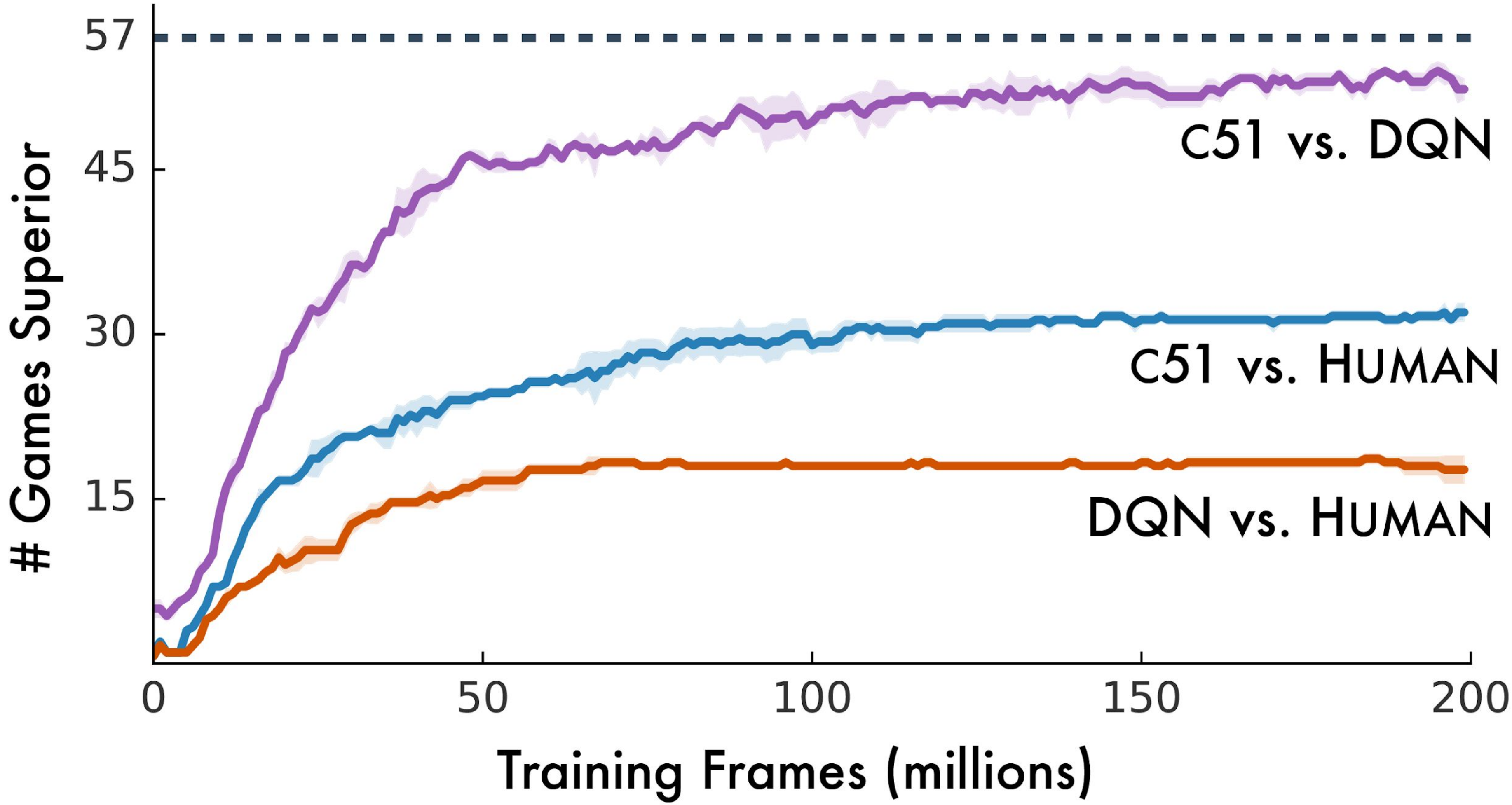



27140
OXYGEN


1.0
0.5
0.0
Probability
0
2
4
6
8
10
Return



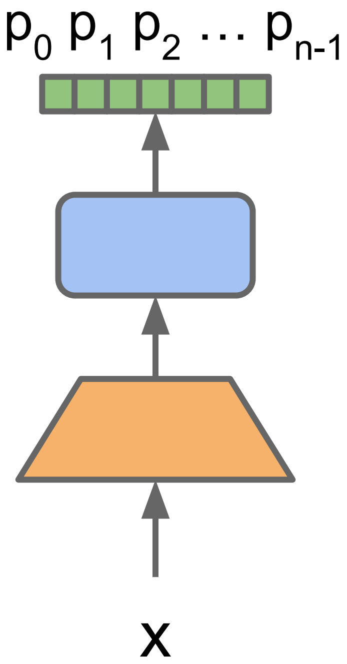
2310
OXYGEN
ACTIVISION
Probability
Return
Probability
Return
Probability
Probability
Return
ACTIVISION

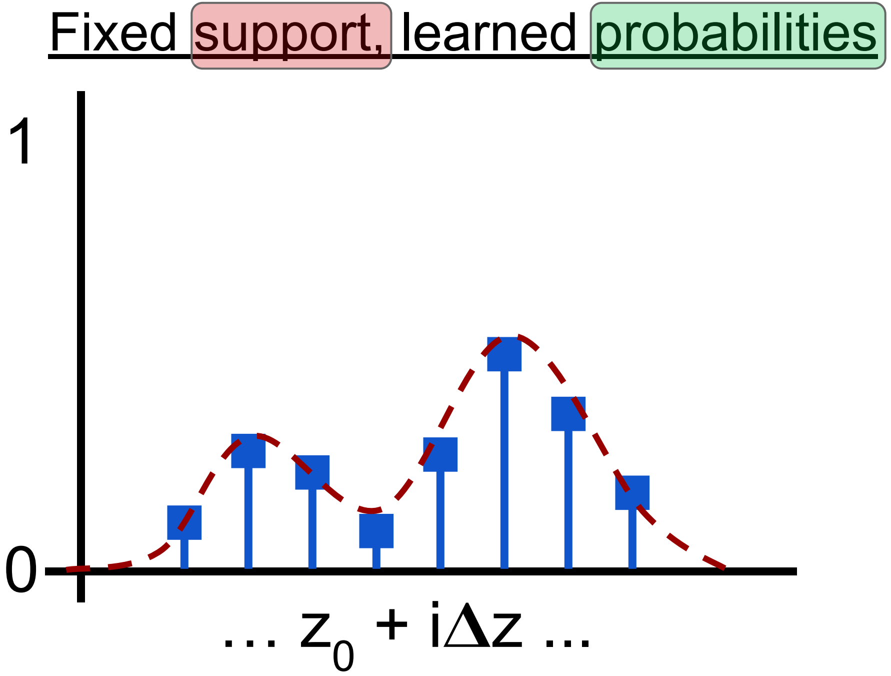

57
45
30
15
Games Superior
C51 vs. DQN
C51 vs. HUMAN
DQN vs. HUMAN
0
50
100
150
200
Training Frames (millions)

Results on 57 games Atari 2600

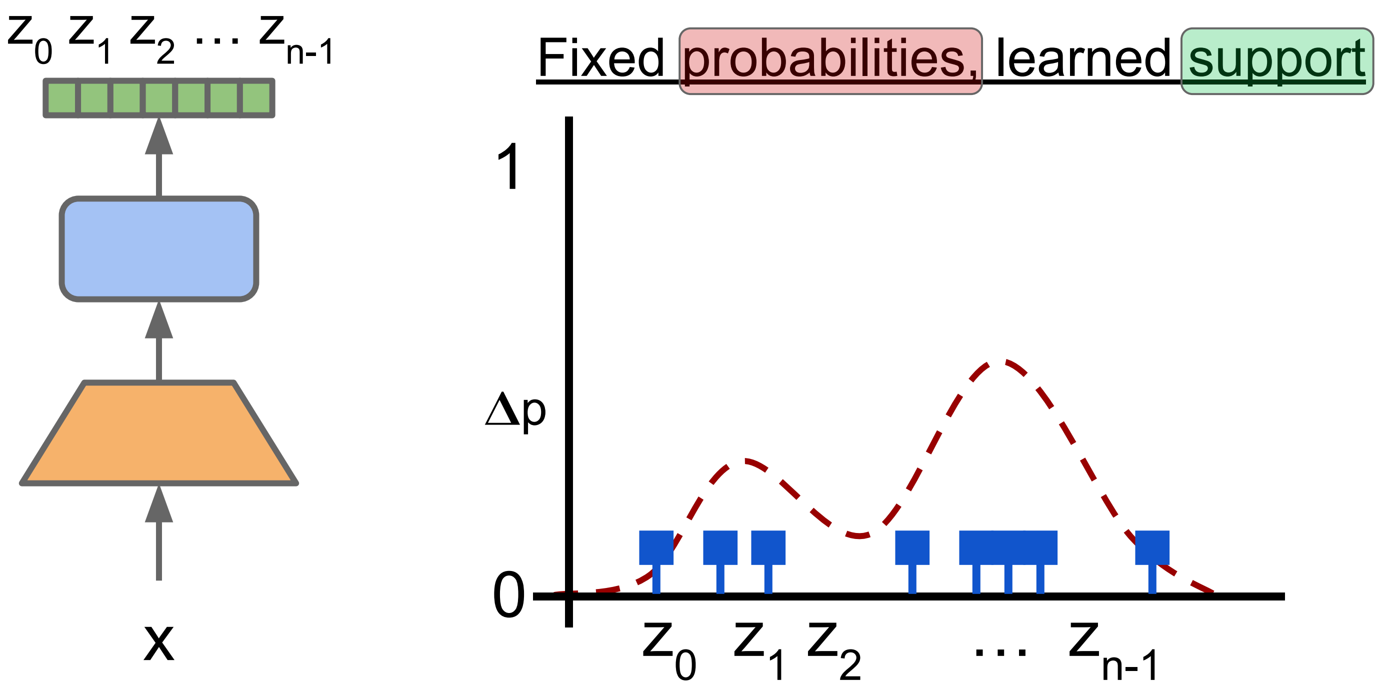
	Mean	**Median**	**>human**
DQN	228%	79%	24
Double DQN	307%	118%	33
Dueling	373%	151%	37
Prio. Duel.	592%	172%	39
C51	701%	178%	40

Categorical representation

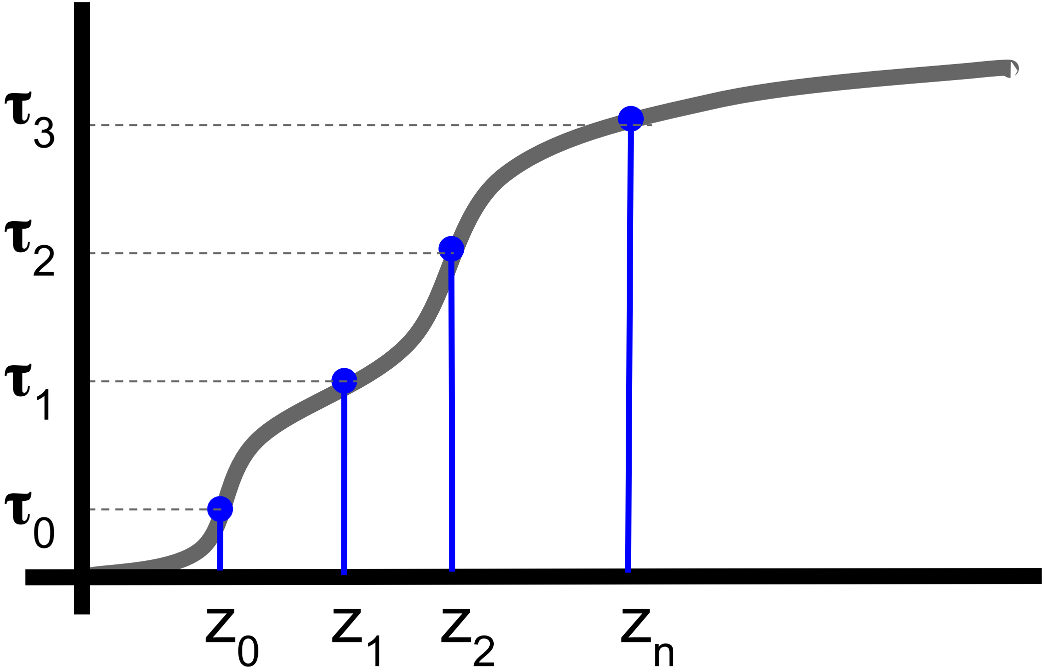




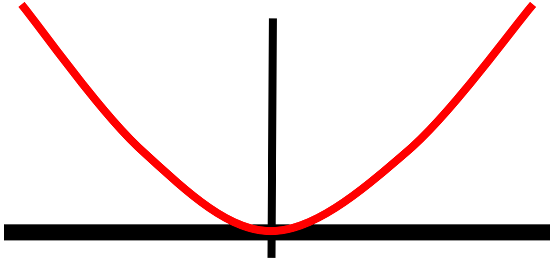
Quantile Regression Networks



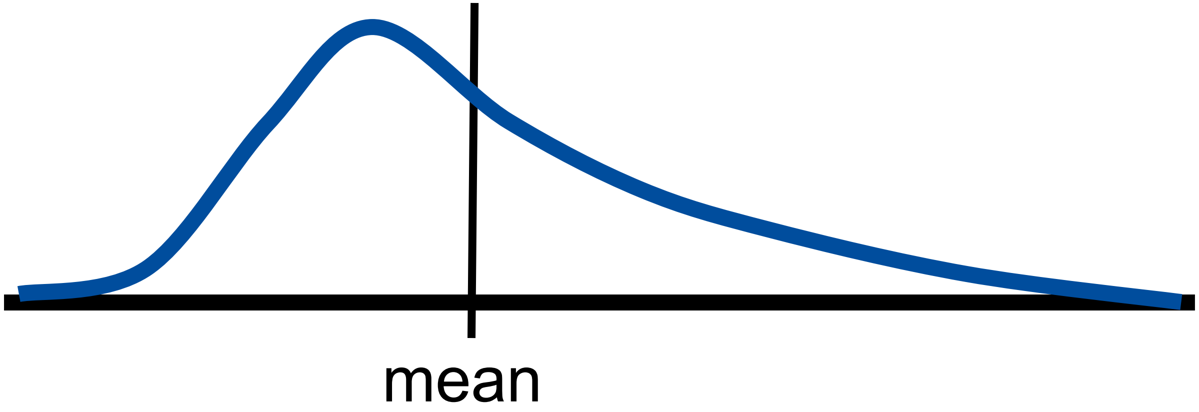
Inverse CDF learnt by Quantile Regression



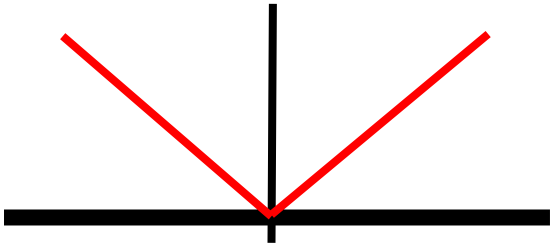
l2-regression



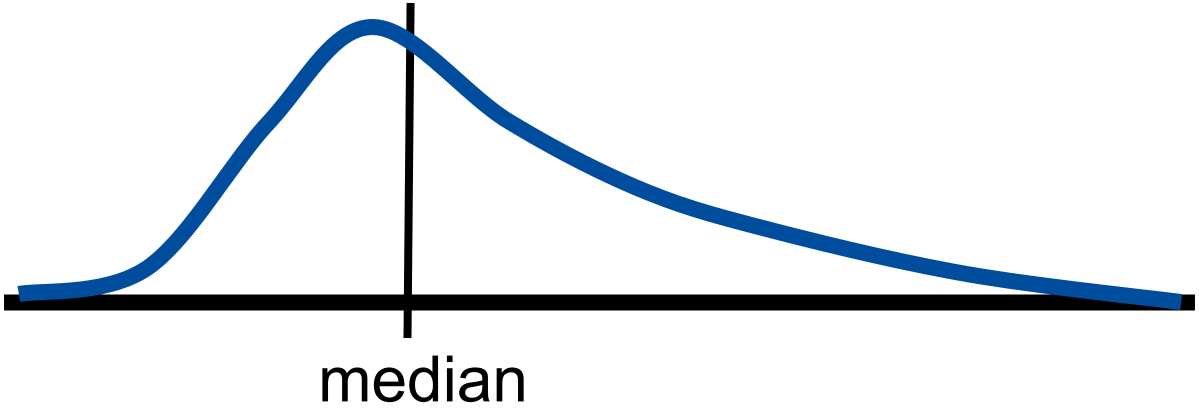
$$loss = x^2$$



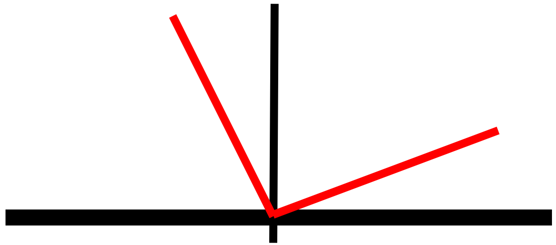
l1-regression



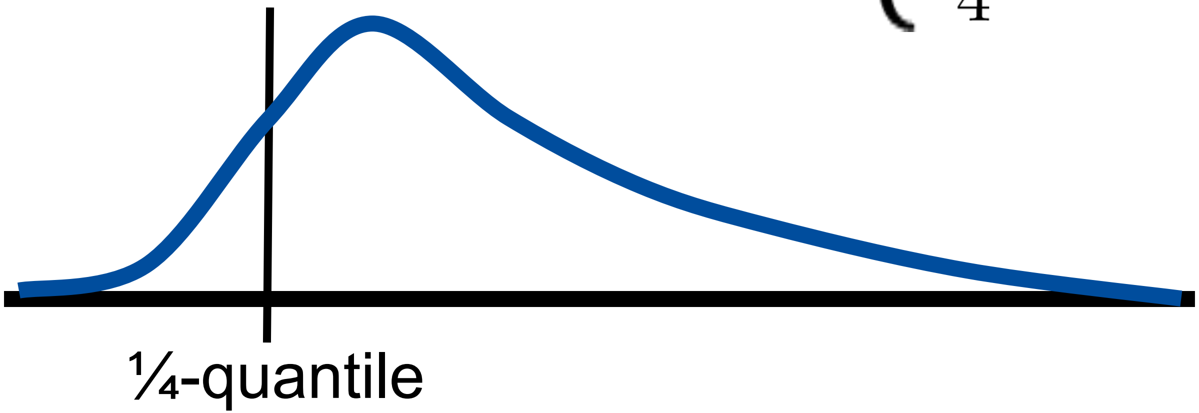
$$loss = |x|$$



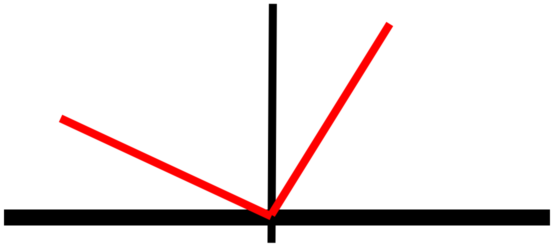
¼-quantile-regression



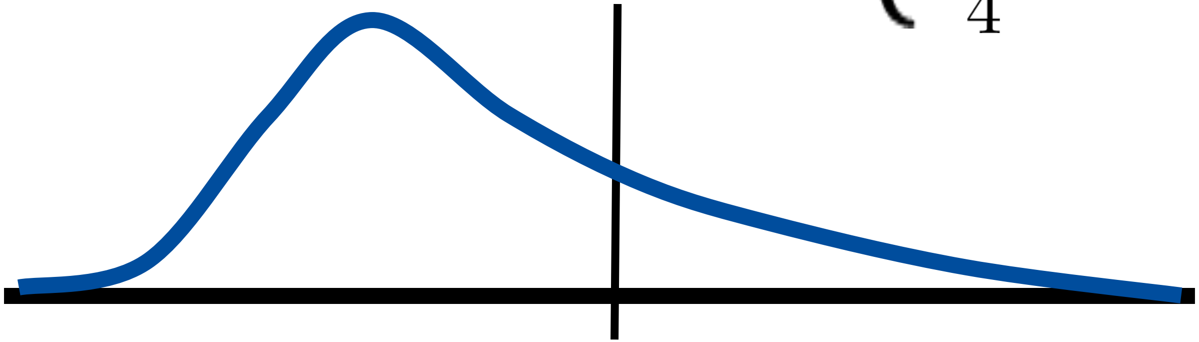
$$loss = \begin{cases} \frac{1}{4}x, & \text{for } x \geq 0 \\ -\frac{3}{4}x, & \text{for } x < 0 \end{cases}$$



¾-quantile-regression

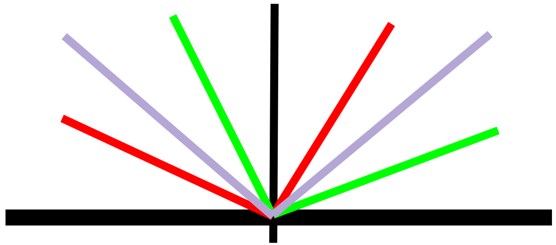


$$loss = \begin{cases} \frac{3}{4}x, & \text{for } x \geq 0 \\ -\frac{1}{4}x, & \text{for } x \geq 0 \end{cases}$$

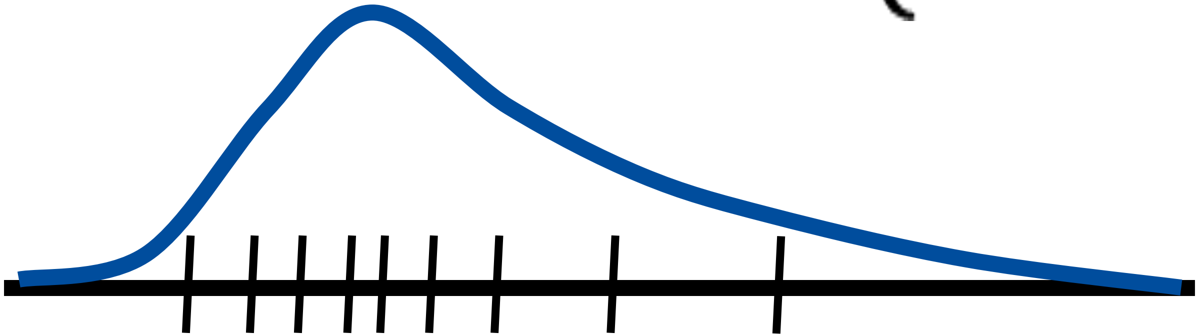


¾-quantile

many-quantiles-regression

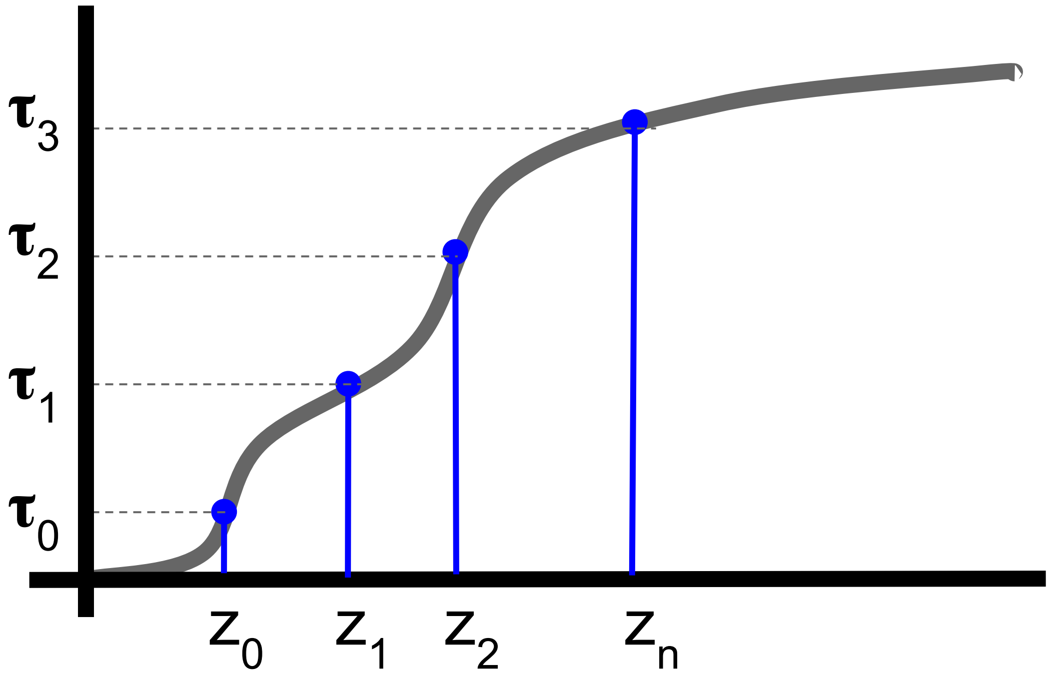


$$loss = \begin{cases} \tau x, \ \text{for } x \geq 0 \\ (\tau - 1)x, \ \text{for } x \geq 0 \end{cases}$$



many-quantiles

Inverse CDF learnt by Quantile Regression



Quantile Regression DQN

$$z \sim Z_\tau(x_t, a_t)$$

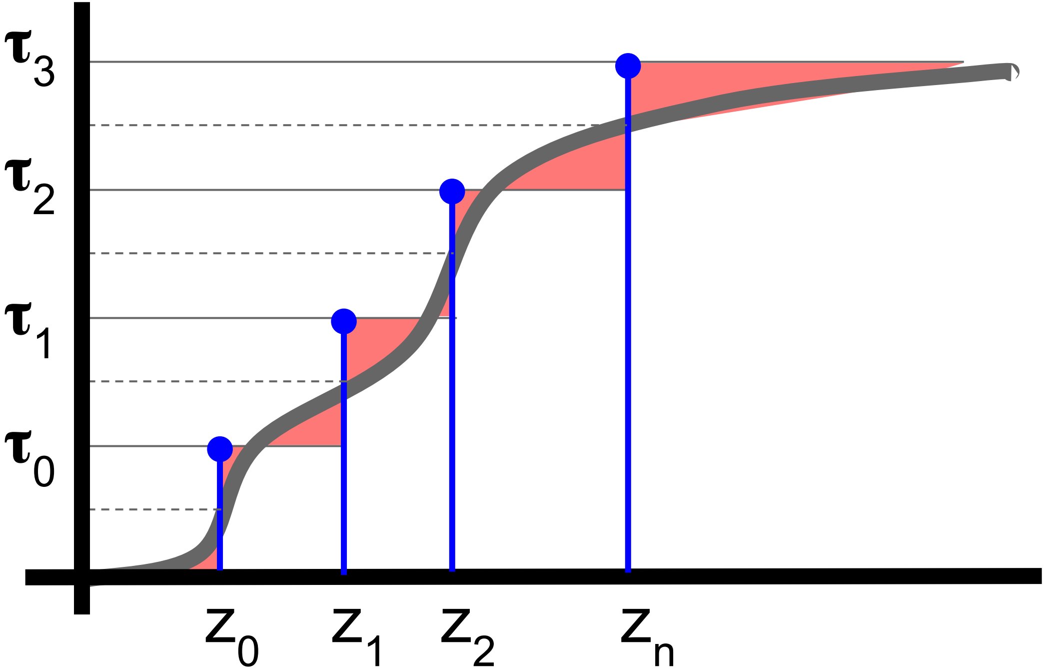
$$z' \sim Z_\tau(x_{t+1}, a^*)$$

$$\delta_t = r_t + \gamma z' - z$$

$$\text{QR loss: } \rho_\tau(\delta) = \delta(\tau - \mathbb{I}_{\delta<0})$$

Quantile Regression = projection in Wasserstein!

(on a uniform grid)



QR distributional Bellman operator

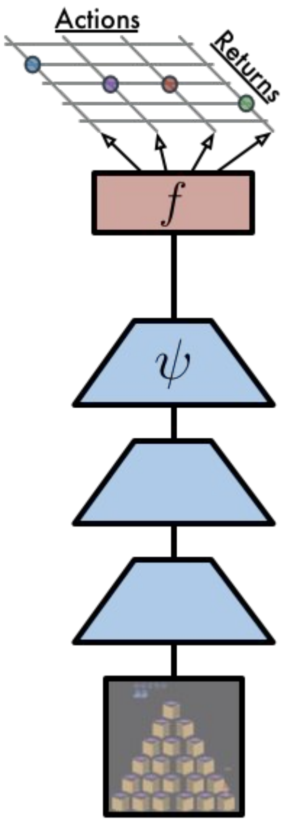
Theorem: $\boxed{\Pi_{QR} T^{\pi} \text{ is a contraction (in Wasserstein)}}$ [Dabney et al., 2018]

Intuition: quantile regression = projection in Wasserstein

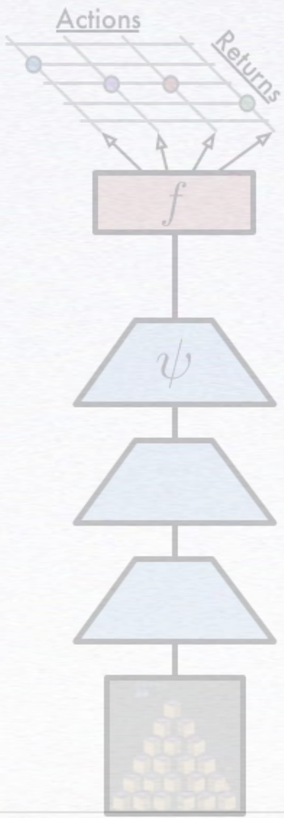
Reminder:

- T^{π} is a contraction (both in Cramer and Wasserstein)
- $\Pi_n T^{\pi}$ is a contraction (in Cramer)

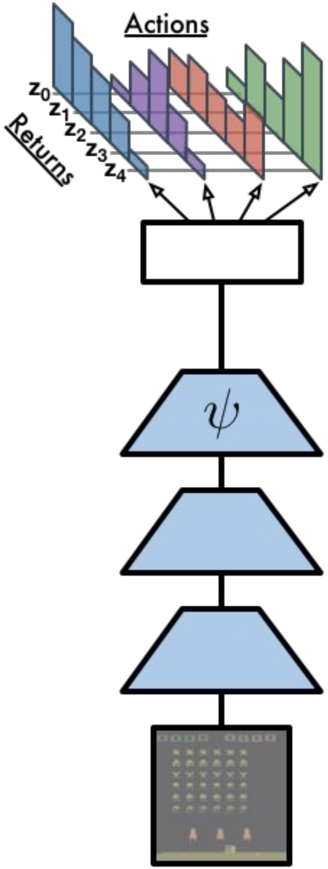
DQN



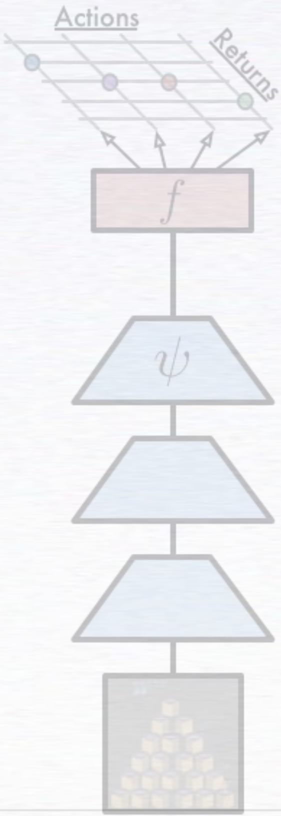
DQN



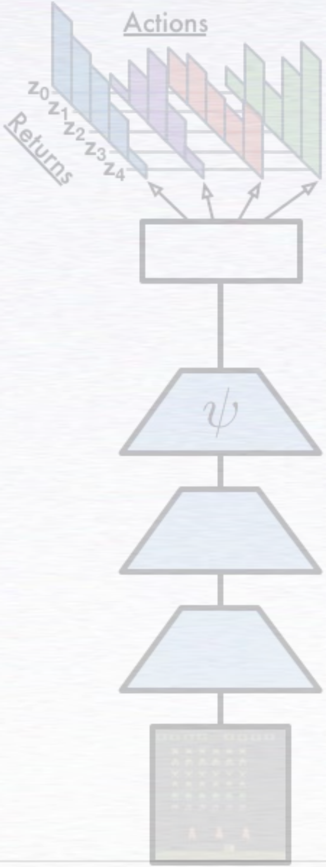
C51



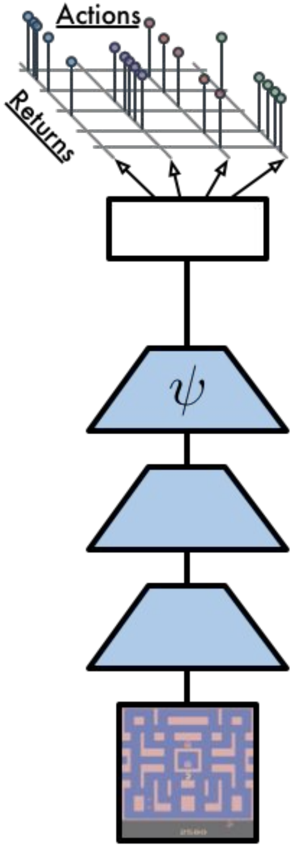
DQN



C51



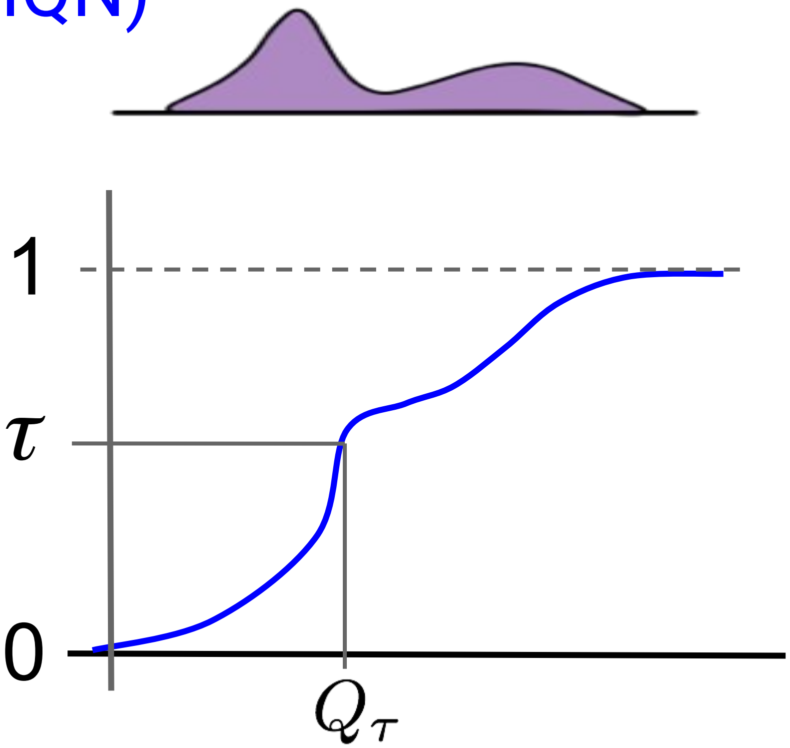
QR-DQN



Quantile-Regression DQN

	Mean	Median
DQN	228%	79%
Double DQN	307%	118%
Dueling	373%	151%
Prio. Duel.	592%	172%
C51	701%	178%
QR-DQN	864%	193%

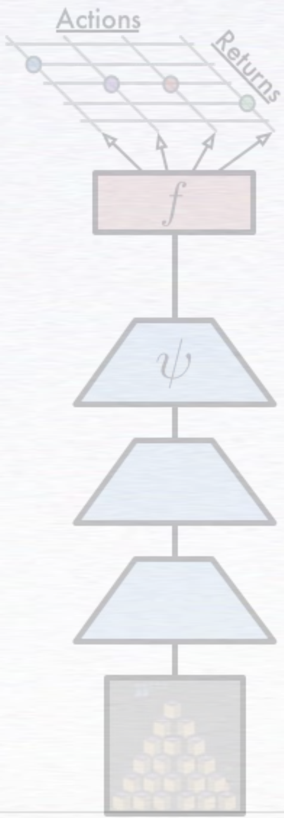
Implicit Quantile Networks (IQN)



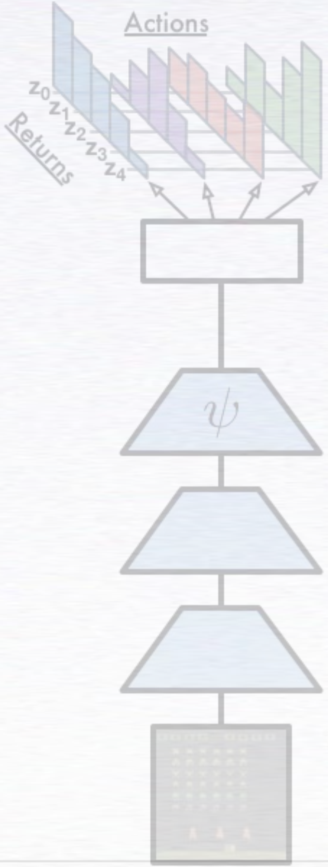
Learn a parametric inverse CDF

$$\tau \mapsto F_Z^{-1}(\tau)$$

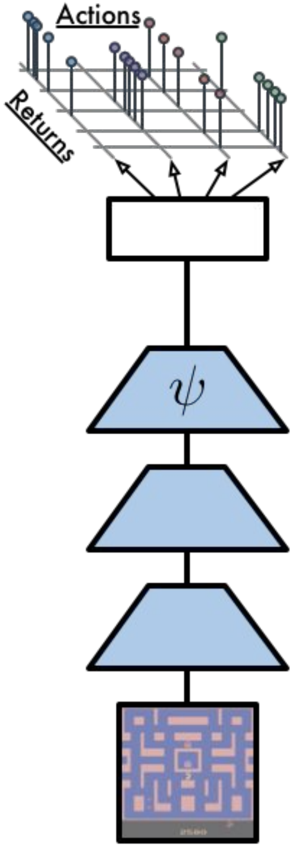
[DQN](#)



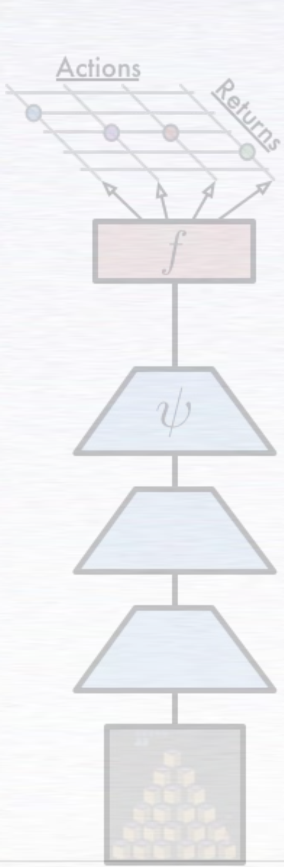
[C51](#)



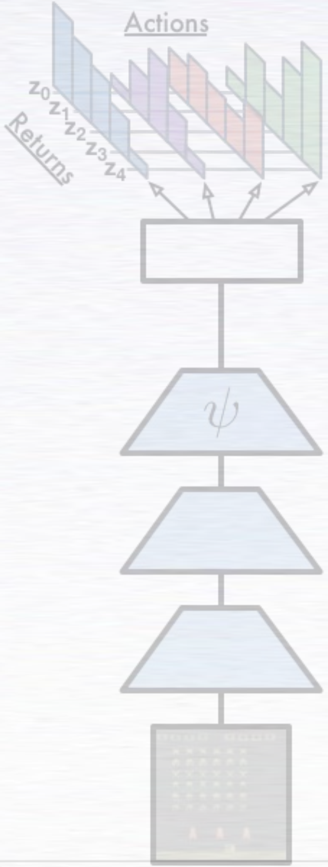
[QR-DQN](#)



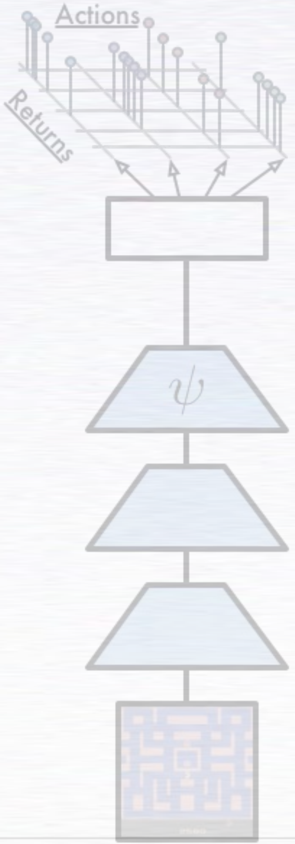
DQN



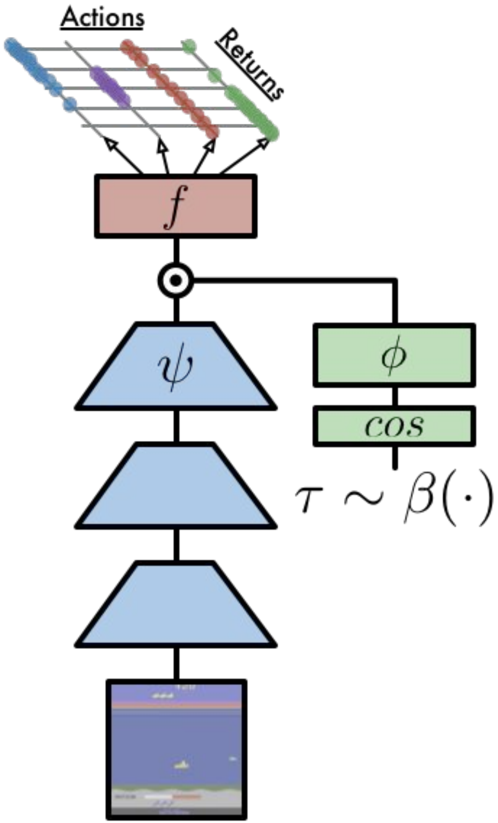
C51



QR-DQN



IQN

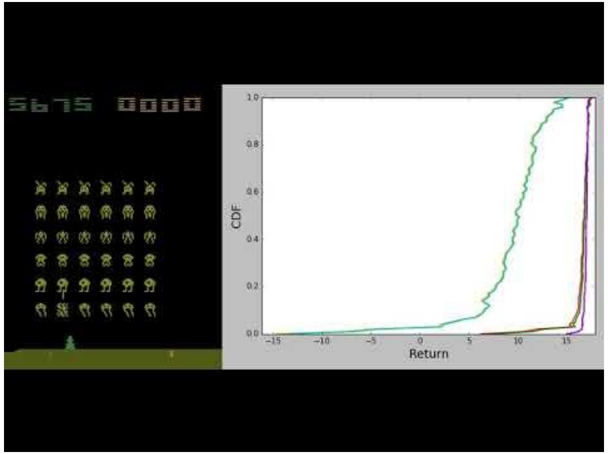
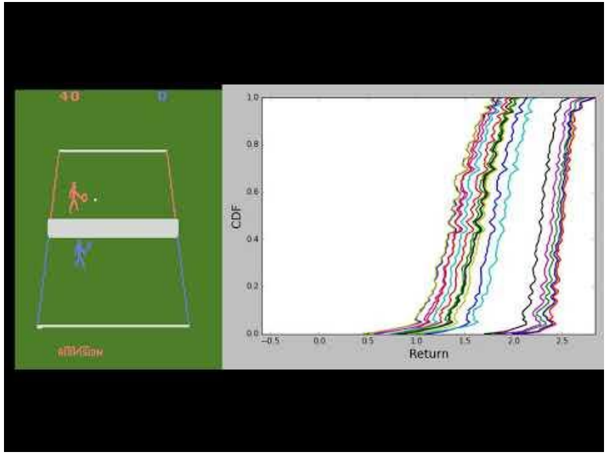
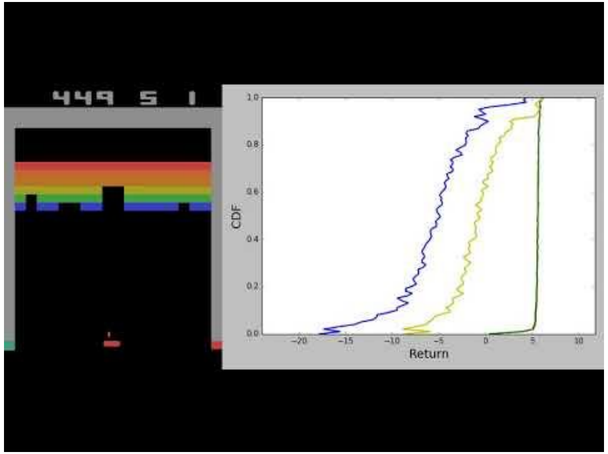
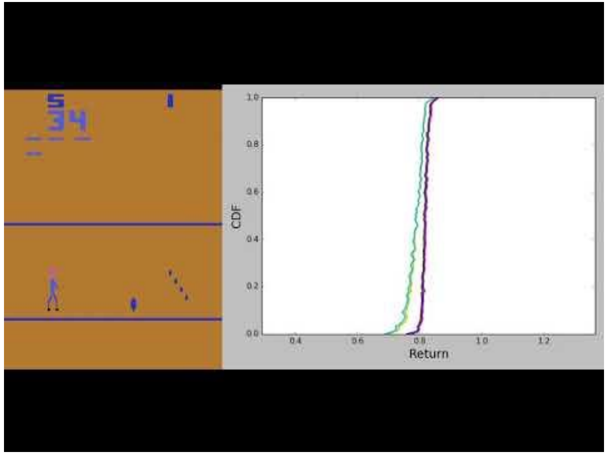
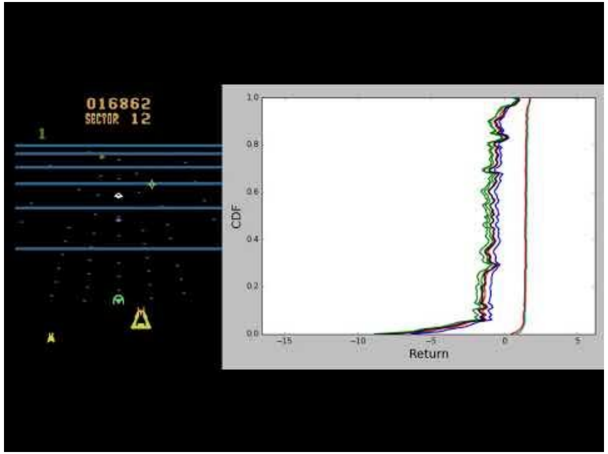
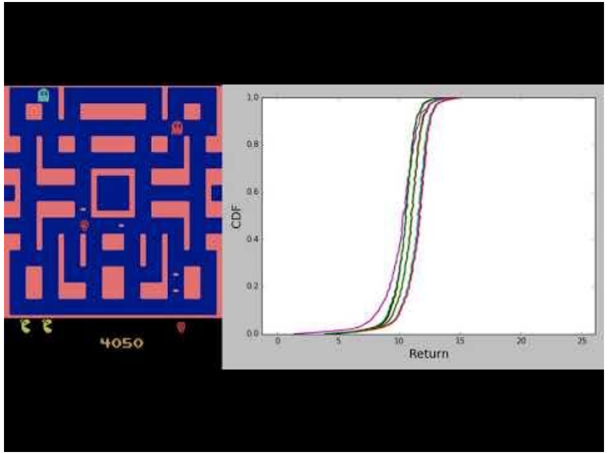


Implicit Quantile Networks for TD

$$\tau \sim \mathcal{U}[0,1], \qquad z = Z_\tau(x_t, a_t)$$
$$\tau' \sim \mathcal{U}[0,1], \qquad z' = Z_\tau(x_{t+1}, a^*)$$

$$\delta_t = r_t + \gamma z' - z$$

$$\text{QR loss: } \rho_\tau(\delta) = \delta(\tau - \mathbb{I}_{\delta<0})$$


CDF
Return

CDF
Return

CDF
Return

CDF
Return

016862
SECTOR 12
CDF
Return

4050
CDF
Return

Implicit Quantile Networks

	Mean	Median	Human starts
DQN	228%	79%	68%
Prio. Duel.	592%	172%	128%
C51	701%	178%	116%
QR-DQN	864%	193%	153%
IQN	**1019%**	**218%**	**162%**

Implicit Quantile Networks

	Mean	**Median**	**Human starts**
DQN	228%	79%	68%
Prio. Duel.	592%	172%	128%
C51	701%	178%	116%
QR-DQN	864%	193%	153%
IQN	1019%	218%	**162%**
Rainbow	**1189%**	**230%**	125%

Almost as good as SOTA (Rainbow/Reactor) which combine prio/dueling/categorical/...

Why does it work?

- In the end **we only use the mean** of these distributions

Why does it work?

- In the end **we only use the mean** of these distributions

When we use deep networks, maybe:

- Auxiliary task effect:
  - Same signal to learn from but more predictions
  - More predictions → richer signal → better representations
  - Reduce state aliasing (disambiguate different states based on return)
- Density estimation instead of l2-regressions
  - RL uses same tools as deep learning
  - Lower variance gradient
- Other reasons?

Algorithms

Evaluation

Policy:
- **Risk-neutral**
- Risk seeking/averse
- Exploration: (optimism, Thompson sampling)

Algorithms:
- **Value-based**
- Policy-based

Agents:
DQN, A3C, Impala, DDPG, TRPO, PPO, ...

Distribution over
- **Returns**
- Policies

Environments
Atari, DMLab30, Control suite, Go,...

Distributional RL

Other:
- State aliasing
- Reward clipping
- Undiscounted RL

Deep Learning impact:
- Lower variance gradients
- Richer representations

Convergence analysis
- **Contraction property**
- Control case
- SGD friendly

Representation of distributions
- **Categorical**
- **Quantile regression**
- Mixture of Gaussians
- Generative models

Distributional loss
- **Wasserstein**
- **Cramer**
- other?

Theory

Deep Learning

References

- *A distributional perspective on reinforcement learning*, Bellemare, Dabney, Munos, ICML2017
- *An Analysis of Categorical Distributional Reinforcement Learning*, Rowland, Bellemare, Dabney, Munos, Teh, AISTATS2018
- *Distributional reinforcement learning with quantile regression*, Dabney, Rowland, Bellemare, Munos, AAAI2018
- *Implicit Quantile Networks for Distributional Reinforcement Learning*, Dabney, Ostrovski, Silver, Munos, ICML2018