Lecture 9: Distributional RL
CS885 Reinforcement Learning

2022-10-17

Complementary readings:
Bellemare, Dabney, Munos. A distributional perspective on reinforcement learning. ICML. 2017.
Bellemare, Dabney, Rolland. Distributional Reinforcement Learning, MIT Press, 2023.

Pascal Poupart
David R. Cheriton School of Computer Science

UNIVERSITY OF
WATERLOO

Outline

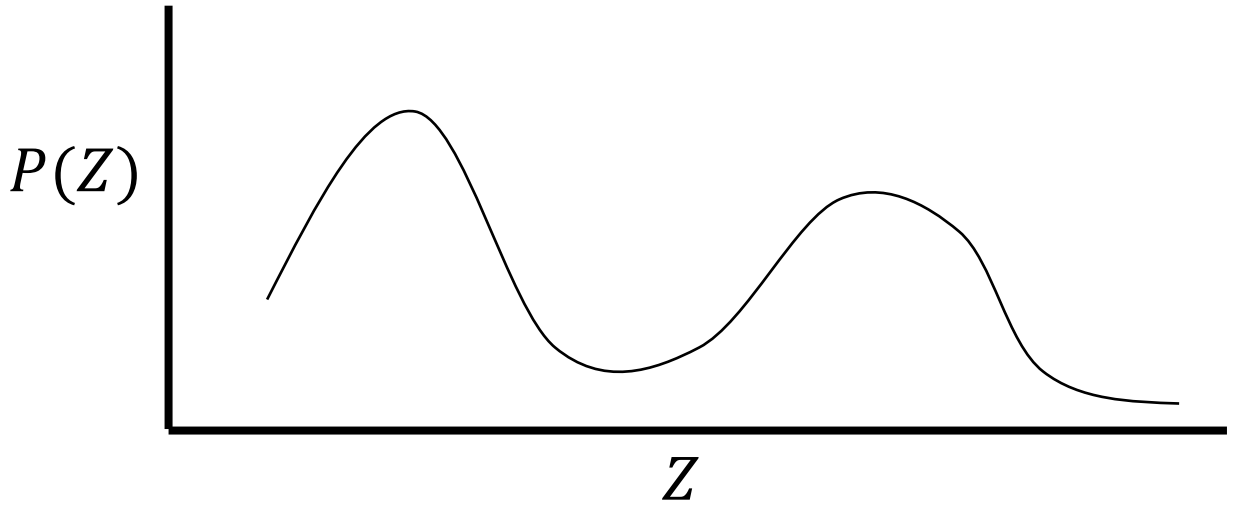
- Distributional Reinforcement Learning
  - Enables risk sensitive objectives
  - Distributional returns
  - C51 (Categorical DQN) Bellemare et al., 2017

Objective

- Let $Z = \sum_t \gamma^t R_t$ be the return random variable
- Traditional RL objective:
  - Mean: $E[Z]$
- Risk sensitive RL objectives:
  - Mean-variance: $E[Z] - \lambda V[Z]$
  - Cumulative distribution: $CDF_Z(z) = \Pr(Z \leq z)$
  - Value at risk: $\text{VaR}_\alpha(Z) = CDF_Z^{-1}(\alpha)$
  - Conditional value at risk: $CVaR_\alpha(Z) = E[Z \mid Z \geq VaR_\alpha(Z)]$

Distributional RL

- Idea: keep track of return distribution $P(Z)$



- Use $P(Z)$ to compute desired objective

Return Distribution

- Random variables:

$$R(s_t, a_t) \sim P(r_t|s_t, a_t),$$

$$s_{t+1} \sim P(s_{t+1}|s_t, a_t),$$

$$a_t \sim \pi(a_t|s_t),$$

- Return distribution: $Z^{\pi}(s_0) = \sum_{t=0} \gamma^t R(s_t, a_t)$
- Expected return: $V^{\pi}(s_0) = E_{P,\pi}[\sum_{t=0} \gamma^t R(s_t, a_t)]$

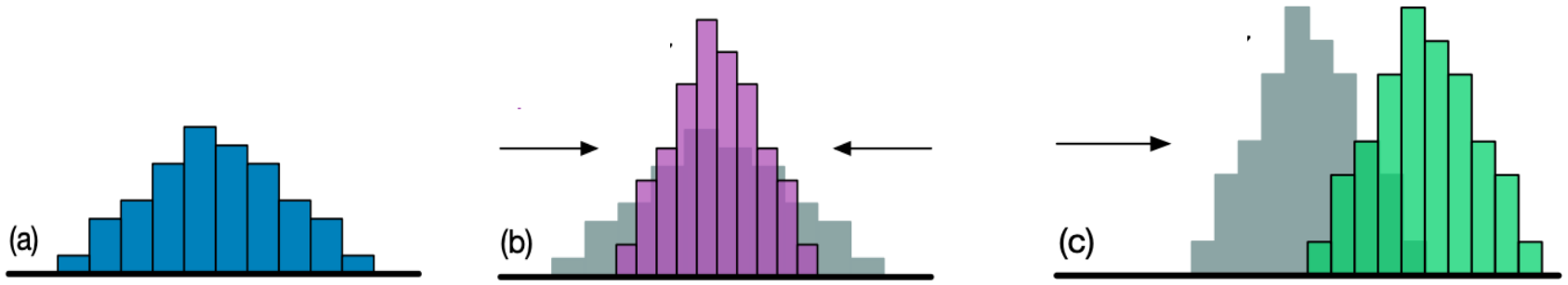
Policy Evaluation

- Policy evaluation

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

$$Z(s,a) = R(s,a) + \gamma Z(s',a')$$

$Z(s',a')$ $\gamma Z(s',a')$ $R(s,a) + \gamma Z(s',a')$



Graphs from Bellemare et al., 2017

Convergence

- Let $\mathcal{T}^\pi$ be the policy evaluation operator
- $\mathcal{T}^\pi Z(s,a) = R(s,a) + \gamma Z(s',a')$

Theorem: $\mathcal{T}^\pi$ converges to a unique return distribution

Proof sketch: $\mathcal{T}^\pi$ is a γ-contraction mapping according to the Wasserstein metric d_w

$$d_W\big(\mathcal{T}^\pi Z(s,a), \mathcal{T}^\pi Z'(s,a)\big) \leq \gamma d_W\big(Z(s,a), Z'(s,a)\big)$$

Bellman Equation

- Bellman Optimality Equation

$$Q(s,a) = E_P[R(s,a)] + \gamma E_{P,\pi}[Q(s', argmax_{a'} Q(s,a'))]$$

$$Z(s,a) = R(s,a) + \gamma\, Z(s', argmax_{a'} E[Z(s',a')])$$

- NB: If we replace $argmax_{a'} E[\cdot]$ by a greedy risk averse objective, $Z(s,a)$ may not be optimal since risk averse objectives cannot be computed exactly by dynamic programming.

Convergence

- Let $\mathcal{T}^*$ be the Bellman operator

$$\mathcal{T}^* Z(s,a) = R(s,a) + \gamma Z(s', argmax_{a'} E[Z(s',a')])$$

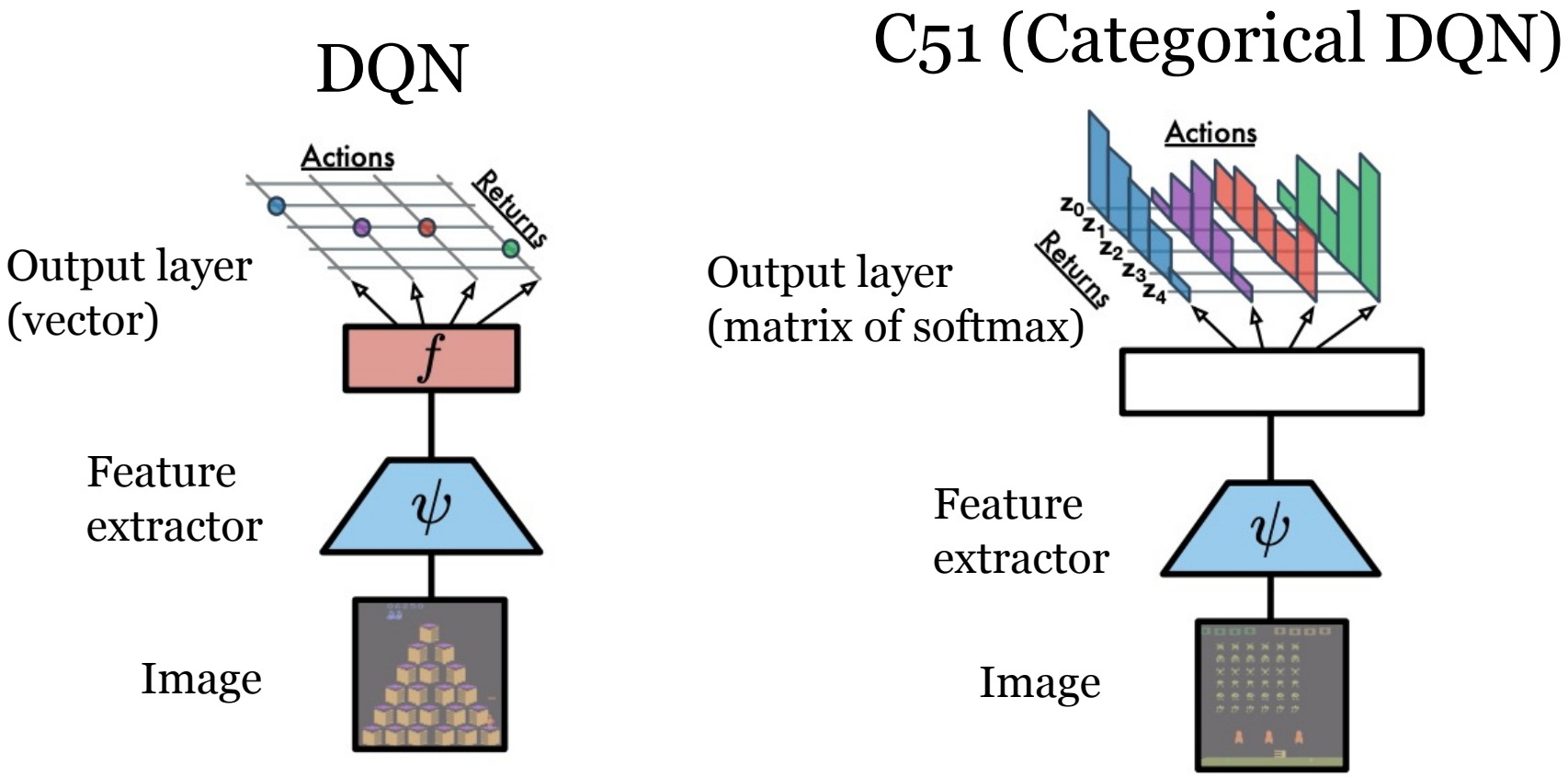
- NB: Optimal return distribution is not unique
  - Each optimal policy may have a different return distribution

Theorem: $\mathcal{T}^*$ converges to a set of optimal return distributions

Proof: complicated

- Cannot show that $\mathcal{T}^*$ is a contraction mapping

C51 (Categorical DQN)



Pictures from Dabney et al., 2018

C51 (Categorical DQN)

Initialize weights $\boldsymbol{w}$ and $\bar{\boldsymbol{w}}$ at random
Observe current state s
Loop
- Select action a and execute it
- Receive reward r and observe s'
- Add (s, a, s', r) to experience buffer
- Sample mini-batch of experiences from buffer
- For each experience (s, a, s', r) in mini-batch do
  - $p_i \leftarrow 0 \quad \forall i \in \{0,1, \dots, N\}$
  - Greedy action: $a' \leftarrow argmax_{a'} \sum_{i'} P_{\bar{\boldsymbol{w}}}(Z(s', a') = z_{i'}) z_{i'}$
  - For each $i' \in \{0,1, \dots, N\}$ do
    - Backup $z_{i'}$ and project it in $[z_{min}, z_{max}]$: $\hat{\mathcal{T}} z_{i'} \leftarrow \left[r + \gamma z_{i'}\right]_{z_{min}}^{z_{max}}$
    - Real index: $i \leftarrow (\hat{\mathcal{T}} z_{i'} - z_{min})/\Delta z$. (where $\Delta z = (z_{max} - z_{min})/N$)
    - Neighboring integer indices: $l \leftarrow \lfloor i \rfloor, \quad u \leftarrow \lceil i \rceil$
    - Distribute probability $P_{\bar{\boldsymbol{w}}}(Z(s', a') = z_{i'})$ of $\hat{\mathcal{T}} z_{i'}$:
      - $p_l \leftarrow p_l + P_{\bar{\boldsymbol{w}}}(Z(s', a') = z_{i'})(u - i)$
      - $p_u \leftarrow p_u + P_{\bar{\boldsymbol{w}}}(Z(s', a') = z_{i'})(i - l)$
  - Cross entropy loss: $L(\boldsymbol{w}) \leftarrow -\sum_i p_i \log P_{\boldsymbol{w}}(Z(s, a) = z_i)$
  - Update weights: $\boldsymbol{w} \leftarrow \boldsymbol{w} - \alpha \nabla_{\boldsymbol{w}} L(\boldsymbol{w})$
- Every c steps, update target: $\bar{\boldsymbol{w}} \leftarrow \boldsymbol{w}$

Advantage

- Why compute a value distribution when the objective is to maximize expected value?

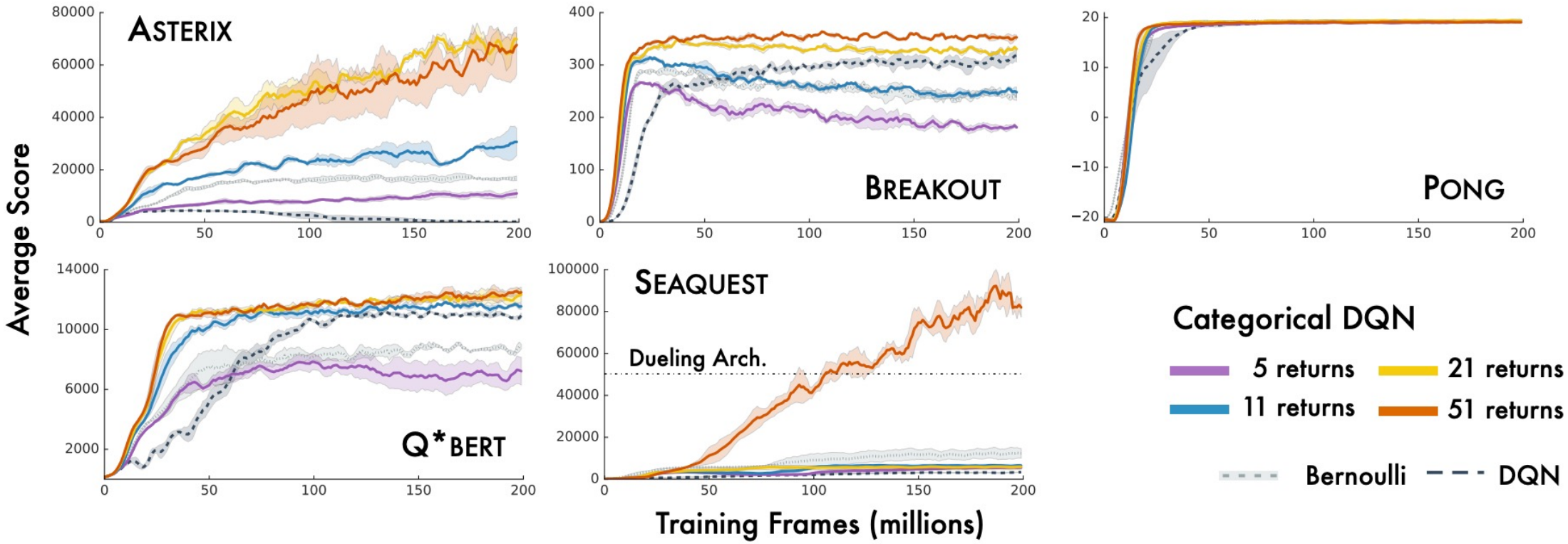
- Categorical DQN can be thought as computing a (weighted) ensemble of returns

$$Q(s,a) = \sum_i P_w(Z(s,a) = z_i) z_i$$

  - Errors in different returns/probabilities may cancel each other, yielding a more accurate estimate of the expectation

Atari Results

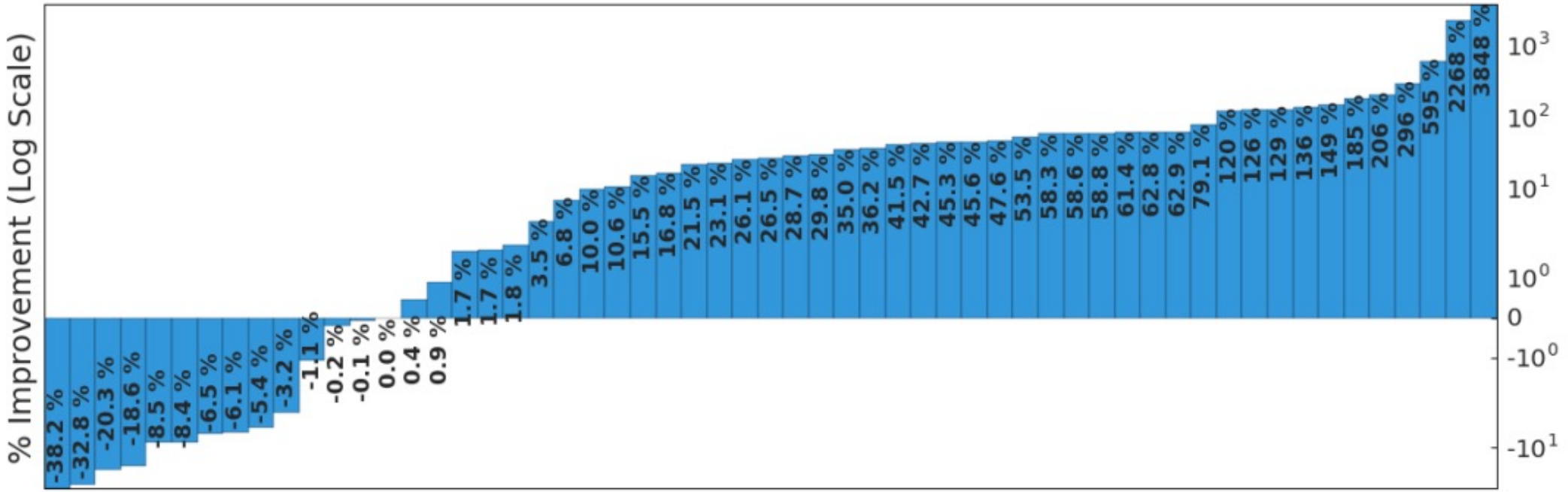
Graphs from Bellemare et al., 2017



UNIVERSITY OF WATERLOO

Atari Results

Improvement of Categorical DQN over Double DQN
new SOTA in 2017



Graph from Bellemare et al., 2017

Distributional Representations

- Return distribution:
  - Categorical: C51 (Bellemare et al., 2017), D4PG (Bath-Maron et al., 2018)
  - Samples: VDGL (Freirich et al., 2019) and SDPG (Singh et al., 2020)
- Quantile function (inverse of CDF): $CDF_Z^{-1}(\alpha)$
  - Step function: QR-DQN (Dabney et al., 2018b), IQN (Dabney et al., 2018), FQF (Yang et al., 2019), NC-QR-DQN (Zhou et al., 2020)
  - Piecewise linear: NDQFN (Zhou et al., 2021)
  - Spline: SPL-DQN (Luo et al., 2021)