# On Finite-Sample Analysis of Offline RL with Deep ReLU Networks[1]

Thanh Nguyen-Tang
(Email: nguyent2792@gmail.com,
Website: thanhnguyentang.github.io)

Applied Artificial Intelligence Institute ($A^2I^2$), Deakin University, Australia

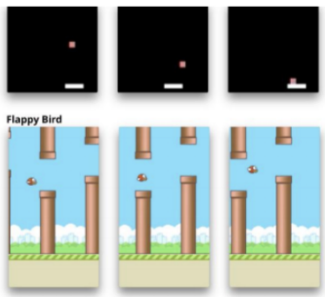
Viet.Operator.Theorists (Virtual) Seminar
April 20, 2021

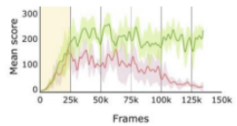
[1]Full paper at https://arxiv.org/abs/2103.06671; This is part of my thesis "*On Practical Considerations of Reinforcement Learning: Provable Robustness, Scalability, and Statistical Efficiency*".

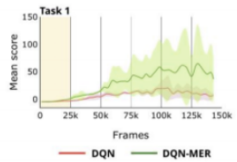
# Outline

# Motivation and Overview

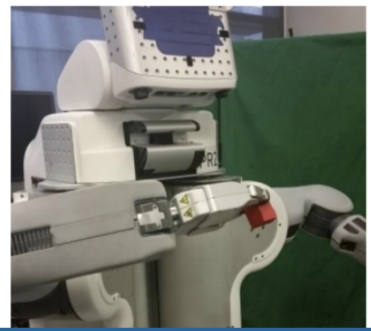
# Sequential Decision Making

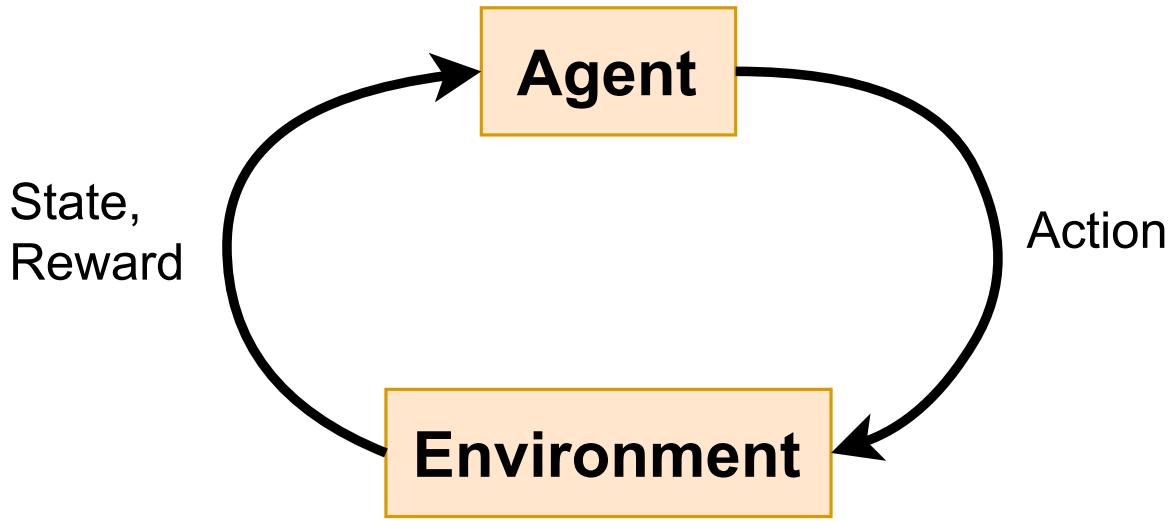







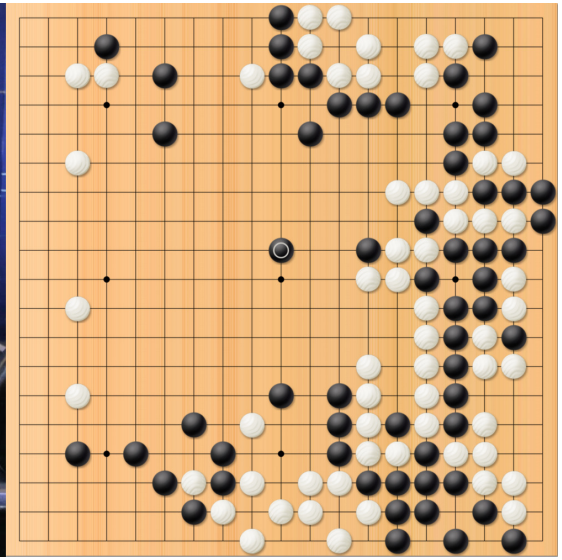

Main framework: **Reinforcement Learning (RL)**

# Reinforcement Learning




Markov decision process MDP$(\mathcal{S}, \mathcal{A}, P, R, \gamma, \rho)$.

# Efficiency

- **Sample efficiency**: Samples are the "*money*" of RL, collecting samples is expensive
- **Computational efficiency**: Training deep RL can be extremely time-consuming

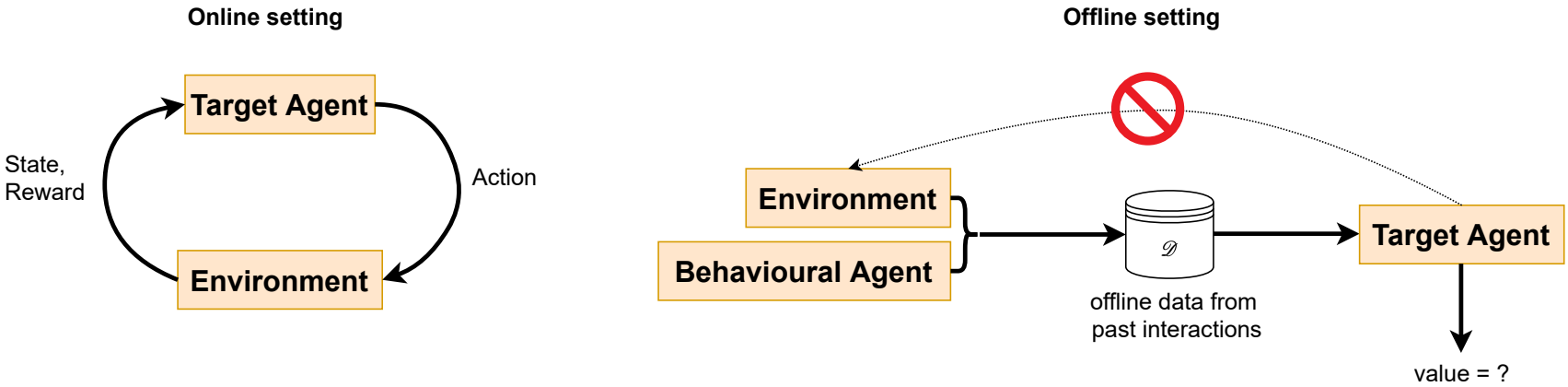
  AlphaGo Zero: trained on $\geq 10^7$ games, in $\geq 1$ month

# Data collection


In many practical settings, *online* interactions with the environment is impractical as collecting new online data is either

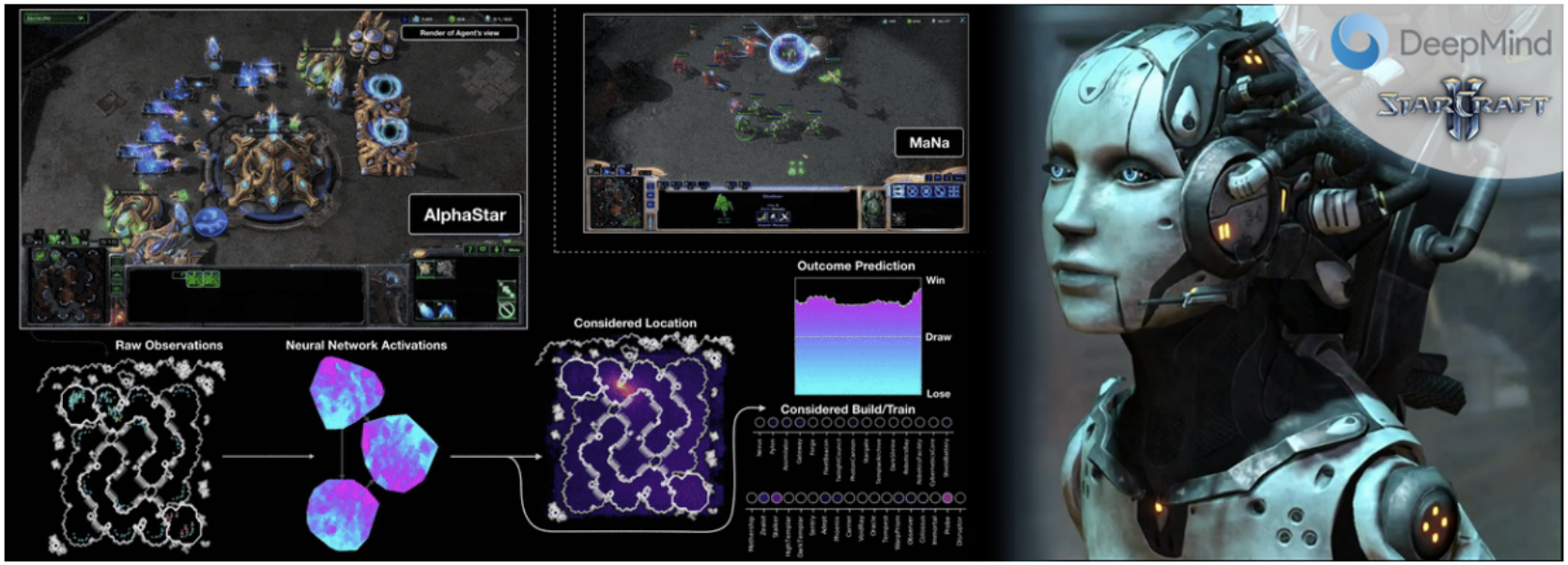
- Expensive (e.g., robotics, educational games, healthcare), or
- Dangerous and even unethical (e.g., autonomous driving, healthcare)

# Offline RL




**Offline RL**: Learn an optimal policy from offline data (collected a priori) without any further online interaction with the environment

# Function Approximation




- In practice, number of states $\geq 10^{100}$!
- It means most states are not visited.

Function approximation: Approximate *value* by functions in a parametric class $\mathcal{F}$ (e.g., deep neural networks)

# Challenges in Function Approximation

- *Generalization*: Generalize knowledge from the visited states to unobserved ones.
- *Expressiveness*: Approximate well a target function in a given function class $\mathcal{F}$ and handle functions outside the class.
- "*Distributional Shift*": Address the change in distributions in offline RL.

# Main Question

Is offline RL statistically efficient in the deep neural network function approximation?

**Contributions**:

1. Identify a general smoothness of RL problems;
2. Derive a comprehensive analysis that handles data-dependent structure;
3. Provide interpretation and insights of the interplay between offline RL and deep learning, an important research direction.

# Main Question

Is offline RL statistically efficient in the deep neural network function approximation?

**Contributions**:

1. Identify a general smoothness of RL problems;
2. Derive a comprehensive analysis that handles data-dependent structure;
3. Provide interpretation and insights of the interplay between offline RL and deep learning, an important research direction.

# Main Question

Is offline RL statistically efficient in the deep neural network function approximation?

**Contributions**:

1. Identify a general smoothness of RL problems;
2. Derive a comprehensive analysis that handles data-dependent structure;
3. Provide interpretation and insights of the interplay between offline RL and deep learning, an important research direction.

# Main Question

Is offline RL statistically efficient in the deep neural network function approximation?

**Contributions**:

1. Identify a general smoothness of RL problems;
2. Derive a comprehensive analysis that handles data-dependent structure;
3. Provide interpretation and insights of the interplay between offline RL and deep learning, an important research direction.

# Main Question

Is offline RL statistically efficient in the deep neural network function approximation?

**Contributions**:

1. Identify a general smoothness of RL problems;
2. Derive a comprehensive analysis that handles data-dependent structure;
3. Provide interpretation and insights of the interplay between offline RL and deep learning, an important research direction.

# Previous Attempts

Previous works in offline RL have limitations

- Limited function approximation: Tabular setting [Yin and Wang, AISTATS'20 ], linear function approximation [Duan and Wang, ICML'20]
- Limited smoothness: [Yang et al. (2019)] uses deep neural network function approximation but cannot handle *inhomogeneous* smoothness .
- Improper analysis: [Le et al. (ICML'19)] uses general function approximation but wrongly ignore the data-dependent structure in their analysis
- Unrealistic setting: [Yang et al. (2019)] does not reuse data for iterations, ignoring the technical bottleneck of the problem

# Previous Attempts

Previous works in offline RL have limitations

- Limited function approximation: Tabular setting [Yin and Wang, AISTATS'20 ], linear function approximation [Duan and Wang, ICML'20]
- Limited smoothness: [Yang et al. (2019)] uses deep neural network function approximation but cannot handle *inhomogeneous* smoothness .
- Improper analysis: [Le et al. (ICML'19)] uses general function approximation but wrongly ignore the data-dependent structure in their analysis
- Unrealistic setting: [Yang et al. (2019)] does not reuse data for iterations, ignoring the technical bottleneck of the problem

# Previous Attempts

Previous works in offline RL have limitations

- Limited function approximation: Tabular setting [Yin and Wang, AISTATS'20 ], linear function approximation [Duan and Wang, ICML'20]
- Limited smoothness: [Yang et al. (2019)] uses deep neural network function approximation but cannot handle *inhomogeneous* smoothness .
- Improper analysis: [Le et al. (ICML'19)] uses general function approximation but wrongly ignore the data-dependent structure in their analysis
- Unrealistic setting: [Yang et al. (2019)] does not reuse data for iterations, ignoring the technical bottleneck of the problem

# Previous Attempts

Previous works in offline RL have limitations

- Limited function approximation: Tabular setting [Yin and Wang, AISTATS'20 ], linear function approximation [Duan and Wang, ICML'20]
- Limited smoothness: [Yang et al. (2019)] uses deep neural network function approximation but cannot handle *inhomogeneous* smoothness .
- Improper analysis: [Le et al. (ICML'19)] uses general function approximation but wrongly ignore the data-dependent structure in their analysis
- Unrealistic setting: [Yang et al. (2019)] does not reuse data for iterations, ignoring the technical bottleneck of the problem

# A New Comprehensive Analysis

| Work | Function approximation | Regularity | Note | Sample complexity |
|---|---|---|---|---|
| Yin and Wang (2020) | Tabular | Tabular | Minimax-optimal | $\tilde{\mathcal{O}}\left(\frac{1}{\epsilon^2}\cdot\frac{SA}{(1-\gamma)^2}\right)$ |
| Duan and Wang (2020) | Linear | Linear | Minimax-optimal | $\tilde{\mathcal{O}}\left(\frac{1}{\epsilon^2}\cdot\frac{\kappa}{(1-\gamma)^4}\right)$ |
| Le et al. (2019) | General | General | No data-dependent structure | N/A |
| Yang et al. (2019) | ReLU networks | Hölder class | No data reuse | $\tilde{\mathcal{O}}\left(K\cdot\left(\frac{1}{\epsilon^2}\right)^{1+\frac{d}{2\alpha}}\cdot\left(\frac{\kappa\cdot A}{(1-\gamma)^2}\right)^{2+\frac{d}{\alpha}}\right)$ |
| Ours | ReLU networks | Besov class | Data reuse for all iterations | $\tilde{\mathcal{O}}\left(\left(\frac{1}{\epsilon^2}\right)^{1+\frac{d}{\alpha}}\cdot\left(\frac{\kappa}{(1-\gamma)^2}\right)^{1+\frac{d}{\alpha}}\right)$ |

Figure: Here $d$ is the dimension of the state-action space, $\alpha$ is the regularity of MDP, and $\kappa$ is a distributional shift measure.

This Talk: A new analysis of offline RL under deep neural networks.

It complements the previous attempts by

1. relying on a general smoothness: Besov smoothness,
2. handling the data-dependent structure,
3. quantifying an improved sample efficiency when data is reused.

# A New Comprehensive Analysis

| Work | Function approximation | Regularity | Note | Sample complexity |
|---|---|---|---|---|
| Yin and Wang (2020) | Tabular | Tabular | Minimax-optimal | $\tilde{\mathcal{O}}\left(\frac{1}{\epsilon^2}\cdot\frac{SA}{(1-\gamma)^2}\right)$ |
| Duan and Wang (2020) | Linear | Linear | Minimax-optimal | $\tilde{\mathcal{O}}\left(\frac{1}{\epsilon^2}\cdot\frac{\kappa}{(1-\gamma)^4}\right)$ |
| Le et al. (2019) | General | General | No data-dependent structure | N/A |
| Yang et al. (2019) | ReLU networks | Hölder class | No data reuse | $\tilde{\mathcal{O}}\left(K\cdot\left(\frac{1}{\epsilon^2}\right)^{1+\frac{d}{2\alpha}}\cdot\left(\frac{\kappa\cdot A}{(1-\gamma)^2}\right)^{2+\frac{d}{\alpha}}\right)$ |
| Ours | ReLU networks | Besov class | Data reuse for all iterations | $\tilde{\mathcal{O}}\left(\left(\frac{1}{\epsilon^2}\right)^{1+\frac{d}{\alpha}}\cdot\left(\frac{\kappa}{(1-\gamma)^2}\right)^{1+\frac{d}{\alpha}}\right)$ |

Figure: Here $d$ is the dimension of the state-action space, $\alpha$ is the regularity of MDP, and $\kappa$ is a distributional shift measure.

This Talk: A new analysis of offline RL under deep neural networks. It complements the previous attempts by

1. relying on a general smoothness: Besov smoothness,
2. handling the data-dependent structure,
3. quantifying an improved sample efficiency when data is reused.

# A New Comprehensive Analysis

| Work | Function approximation | Regularity | Note | Sample complexity |
|---|---|---|---|---|
| Yin and Wang (2020) | Tabular | Tabular | Minimax-optimal | $\tilde{\mathcal{O}}\left(\frac{1}{\epsilon^2}\cdot\frac{SA}{(1-\gamma)^2}\right)$ |
| Duan and Wang (2020) | Linear | Linear | Minimax-optimal | $\tilde{\mathcal{O}}\left(\frac{1}{\epsilon^2}\cdot\frac{\kappa}{(1-\gamma)^4}\right)$ |
| Le et al. (2019) | General | General | No data-dependent structure | N/A |
| Yang et al. (2019) | ReLU networks | Hölder class | No data reuse | $\tilde{\mathcal{O}}\left(K\cdot\left(\frac{1}{\epsilon^2}\right)^{1+\frac{d}{2\alpha}}\cdot\left(\frac{\kappa\cdot A}{(1-\gamma)^2}\right)^{2+\frac{d}{\alpha}}\right)$ |
| Ours | ReLU networks | Besov class | Data reuse for all iterations | $\tilde{\mathcal{O}}\left(\left(\frac{1}{\epsilon^2}\right)^{1+\frac{d}{\alpha}}\cdot\left(\frac{\kappa}{(1-\gamma)^2}\right)^{1+\frac{d}{\alpha}}\right)$ |

Figure: Here $d$ is the dimension of the state-action space, $\alpha$ is the regularity of MDP, and $\kappa$ is a distributional shift measure.

This Talk: A new analysis of offline RL under deep neural networks. It complements the previous attempts by

1. relying on a general smoothness: Besov smoothness,
2. handling the data-dependent structure,
3. quantifying an improved sample efficiency when data is reused.

# A New Comprehensive Analysis

| Work | Function approximation | Regularity | Note | Sample complexity |
|---|---|---|---|---|
| Yin and Wang (2020) | Tabular | Tabular | Minimax-optimal | $\tilde{\mathcal{O}}\left(\frac{1}{\epsilon^2}\cdot\frac{SA}{(1-\gamma)^2}\right)$ |
| Duan and Wang (2020) | Linear | Linear | Minimax-optimal | $\tilde{\mathcal{O}}\left(\frac{1}{\epsilon^2}\cdot\frac{\kappa}{(1-\gamma)^4}\right)$ |
| Le et al. (2019) | General | General | No data-dependent structure | N/A |
| Yang et al. (2019) | ReLU networks | Hölder class | No data reuse | $\tilde{\mathcal{O}}\left(K\cdot\left(\frac{1}{\epsilon^2}\right)^{1+\frac{d}{2\alpha}}\cdot\left(\frac{\kappa\cdot A}{(1-\gamma)^2}\right)^{2+\frac{d}{\alpha}}\right)$ |
| Ours | ReLU networks | Besov class | Data reuse for all iterations | $\tilde{\mathcal{O}}\left(\left(\frac{1}{\epsilon^2}\right)^{1+\frac{d}{\alpha}}\cdot\left(\frac{\kappa}{(1-\gamma)^2}\right)^{1+\frac{d}{\alpha}}\right)$ |

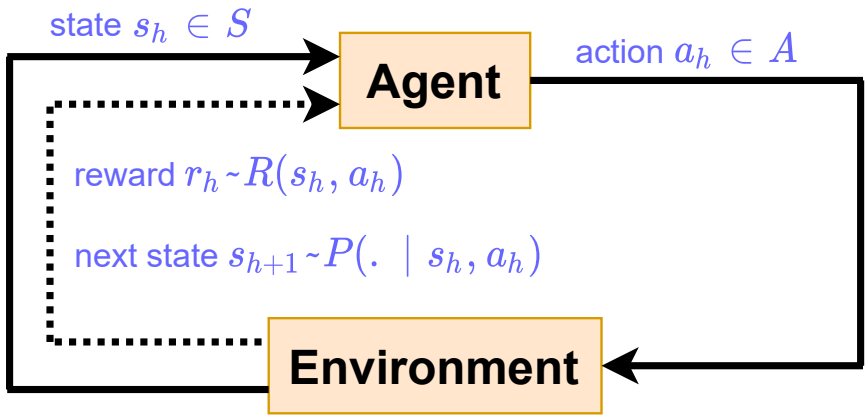
Figure: Here $d$ is the dimension of the state-action space, $\alpha$ is the regularity of MDP, and $\kappa$ is a distributional shift measure.

This Talk: A new analysis of offline RL under deep neural networks. It complements the previous attempts by

1. relying on a general smoothness: Besov smoothness,
2. handling the data-dependent structure,
3. quantifying an improved sample efficiency when data is reused.

# Formal Background and Setup

# Markov Decision Process (MDP)




- $\mathcal{S}$: infinite state space; $\mathcal{A}$: infinite action space; $\gamma \in [0, 1]$: discount factor; $\rho$: initial state distribution.
- Unknown reward distribution $R(s, a) \in \mathcal{P}([0, 1])$ with expected reward $r(s, a) = \mathbb{E}_{r \sim R(s,a)}[r]$
- Unknown transition kernel $P(\cdot|s, a) \in \mathcal{P}(\mathcal{S})$
- Policy $\pi : \mathcal{S} \to \mathcal{P}(\mathcal{A})$, $a \sim \pi(\cdot|s)$
- The expected discounted total reward:

$$v^{\pi} = \mathbb{E}_{\pi}\left[\sum_{h=0}^{\infty} \gamma^h r(s_h, a_h)\right]$$

# Dynamic Programming and Bellman Equation

- Value function: Expected cumulative reward from a state

$$V^{\pi}(s) = \mathbb{E}_{\pi}\left[\sum_{h=0}^{\infty} \gamma^h r(s_h, a_h) | s_0 = s\right].$$

- $Q$-function: Expected cumulative reward from a state-action pair

$$Q^{\pi}(s, a) = \mathbb{E}_{\pi}\left[\sum_{h=0}^{\infty} \gamma^h r(s_h, a_h) | s_0 = s, a_0 = a\right].$$

- Bellman operator: $\forall f \in (\mathcal{S} \times \mathcal{A}) \to \mathbb{R}$

$$(T^{\pi} f)(s, a) := r(s, a) + \gamma \mathbb{E}_{s' \sim P(\cdot|s,a), a' \sim \pi(\cdot|s')}\left[f(s', a')\right]$$

- Bellman equation: $T^{\pi} Q^{\pi} = Q^{\pi}$
- RL with function approximation: Function class $\mathcal{F} \subseteq \{\mathcal{S} \times \mathcal{A} \to \mathbb{R}\}$ approximates $Q^{\pi}$, e.g., linear function, RKHS, neural networks, ...

# Dynamic Programming and Bellman Equation

- Value function: Expected cumulative reward from a state

$$V^{\pi}(s) = \mathbb{E}_{\pi}\left[\sum_{h=0}^{\infty} \gamma^h r(s_h, a_h) | s_0 = s\right].$$

- $Q$-function: Expected cumulative reward from a state-action pair

$$Q^{\pi}(s, a) = \mathbb{E}_{\pi}\left[\sum_{h=0}^{\infty} \gamma^h r(s_h, a_h) | s_0 = s, a_0 = a\right].$$

- Bellman operator: $\forall f \in (\mathcal{S} \times \mathcal{A}) \to \mathbb{R}$

$$(\mathcal{T}^{\pi} f)(s, a) := r(s, a) + \gamma \mathbb{E}_{s' \sim P(\cdot|s,a), a' \sim \pi(\cdot|s')}\left[f(s', a')\right]$$

- Bellman equation: $\mathcal{T}^{\pi} Q^{\pi} = Q^{\pi}$
- RL with function approximation: Function class $\mathcal{F} \subseteq \{\mathcal{S} \times \mathcal{A} \to \mathbb{R}\}$ approximates $Q^{\pi}$, e.g., linear function, RKHS, neural networks, ...

# Dynamic Programming and Bellman Equation

- Value function: Expected cumulative reward from a state

$$V^{\pi}(s) = \mathbb{E}_{\pi}\left[\sum_{h=0}^{\infty} \gamma^h r(s_h, a_h) | s_0 = s\right].$$

- $Q$-function: Expected cumulative reward from a state-action pair

$$Q^{\pi}(s, a) = \mathbb{E}_{\pi}\left[\sum_{h=0}^{\infty} \gamma^h r(s_h, a_h) | s_0 = s, a_0 = a\right].$$

- Bellman operator: $\forall f \in (\mathcal{S} \times \mathcal{A}) \to \mathbb{R}$

$$(T^{\pi} f)(s, a) := r(s, a) + \gamma \mathbb{E}_{s' \sim P(\cdot|s,a), a' \sim \pi(\cdot|s')}\left[f(s', a')\right]$$

- Bellman equation: $T^{\pi} Q^{\pi} = Q^{\pi}$
- RL with function approximation: Function class $\mathcal{F} \subseteq \{\mathcal{S} \times \mathcal{A} \to \mathbb{R}\}$ approximates $Q^{\pi}$, e.g., linear function, RKHS, neural networks, ...

# Dynamic Programming and Bellman Equation

- Value function: Expected cumulative reward from a state

$$V^{\pi}(s) = \mathbb{E}_{\pi}\left[\sum_{h=0}^{\infty} \gamma^h r(s_h, a_h) | s_0 = s\right].$$

- $Q$-function: Expected cumulative reward from a state-action pair

$$Q^{\pi}(s, a) = \mathbb{E}_{\pi}\left[\sum_{h=0}^{\infty} \gamma^h r(s_h, a_h) | s_0 = s, a_0 = a\right].$$

- Bellman operator: $\forall f \in (\mathcal{S} \times \mathcal{A}) \to \mathbb{R}$

$$(T^{\pi} f)(s, a) := r(s, a) + \gamma \mathbb{E}_{s' \sim P(\cdot|s,a), a' \sim \pi(\cdot|s')}\left[f(s', a')\right]$$

- Bellman equation: $T^{\pi} Q^{\pi} = Q^{\pi}$
- RL with function approximation: Function class $\mathcal{F} \subseteq \{\mathcal{S} \times \mathcal{A} \to \mathbb{R}\}$ approximates $Q^{\pi}$, e.g., linear function, RKHS, neural networks, ...

# Dynamic Programming and Bellman Equation

- Value function: Expected cumulative reward from a state

$$V^{\pi}(s) = \mathbb{E}_{\pi}\left[\sum_{h=0}^{\infty} \gamma^h r(s_h, a_h) | s_0 = s\right].$$

- $Q$-function: Expected cumulative reward from a state-action pair

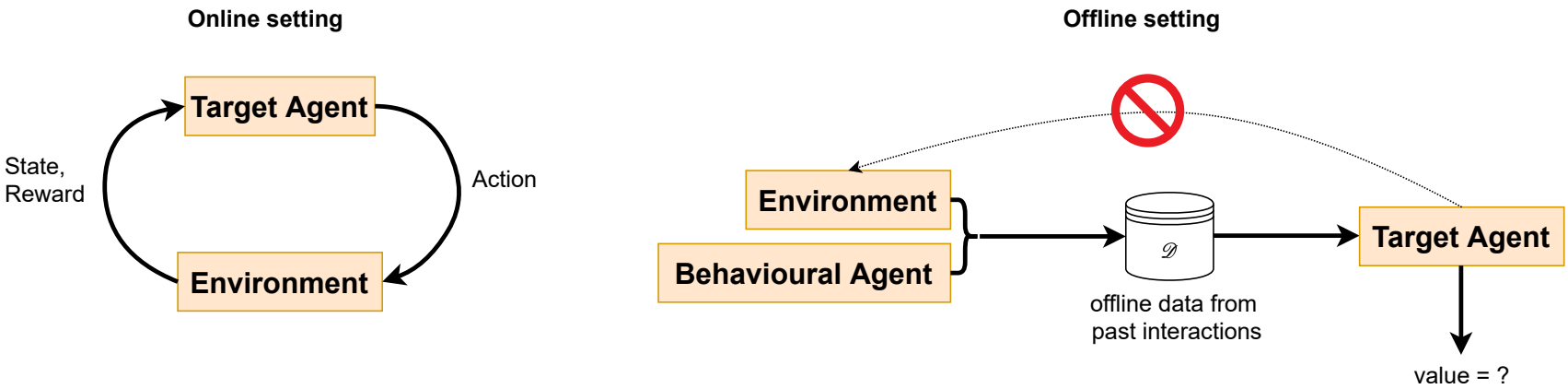
$$Q^{\pi}(s, a) = \mathbb{E}_{\pi}\left[\sum_{h=0}^{\infty} \gamma^h r(s_h, a_h) | s_0 = s, a_0 = a\right].$$

- Bellman operator: $\forall f \in (\mathcal{S} \times \mathcal{A}) \to \mathbb{R}$

$$(T^{\pi} f)(s, a) := r(s, a) + \gamma \mathbb{E}_{s' \sim P(\cdot|s,a), a' \sim \pi(\cdot|s')}\left[f(s', a')\right]$$

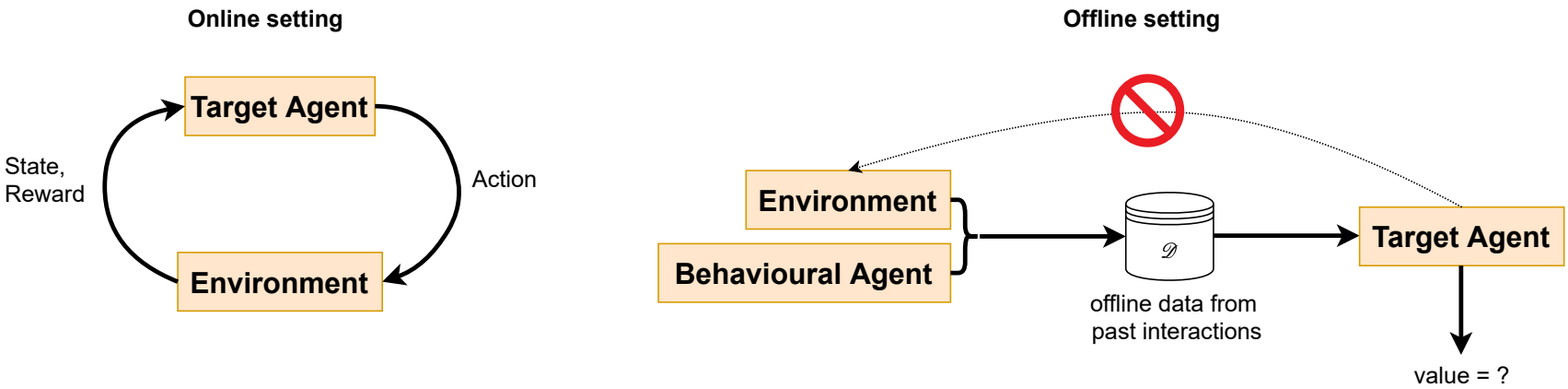
- Bellman equation: $T^{\pi} Q^{\pi} = Q^{\pi}$
- RL with function approximation: Function class $\mathcal{F} \subseteq \{\mathcal{S} \times \mathcal{A} \to \mathbb{R}\}$ approximates $Q^{\pi}$, e.g., linear function, RKHS, neural networks, ...

# Off-Policy Evaluation - A Central Subroutine of Offline RL



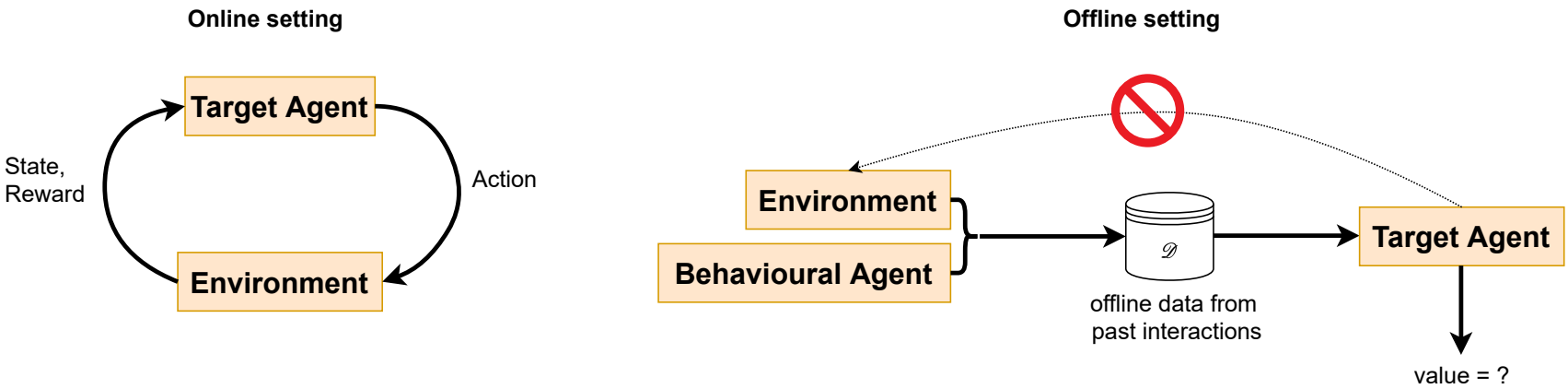

- **Offline data**: $\mathcal{D} = \{(s_i, a_i, r_i, s_i')\}_{i=1}^n$ collected a priori
- **Task**: For any policy $\pi$, estimate $v^\pi$ from $\mathcal{D}$ without any further interaction with MDP
- **Performance metric**: $|\hat{v} - v^\pi|$ where $\hat{v}$ is an (empirical) estimate from $\mathcal{D}$.

# Off-Policy Evaluation - A Central Subroutine of Offline RL




- **Offline data**: $\mathcal{D} = \{(s_i, a_i, r_i, s_i')\}_{i=1}^n$ collected a priori
- **Task**: For any policy $\pi$, estimate $v^\pi$ from $\mathcal{D}$ without any further interaction with MDP
- **Performance metric**: $|\hat{v} - v^\pi|$ where $\hat{v}$ is an (empirical) estimate from $\mathcal{D}$.

# Off-Policy Evaluation - A Central Subroutine of Offline RL




- **Offline data**: $\mathcal{D} = \{(s_i, a_i, r_i, s_i')\}_{i=1}^n$ collected a priori
- **Task**: For any policy $\pi$, estimate $v^\pi$ from $\mathcal{D}$ without any further interaction with MDP
- **Performance metric**: $|\hat{v} - v^\pi|$ where $\hat{v}$ is an (empirical) estimate from $\mathcal{D}$.

# Deep ReLU (Rectified Linear Unit) Networks

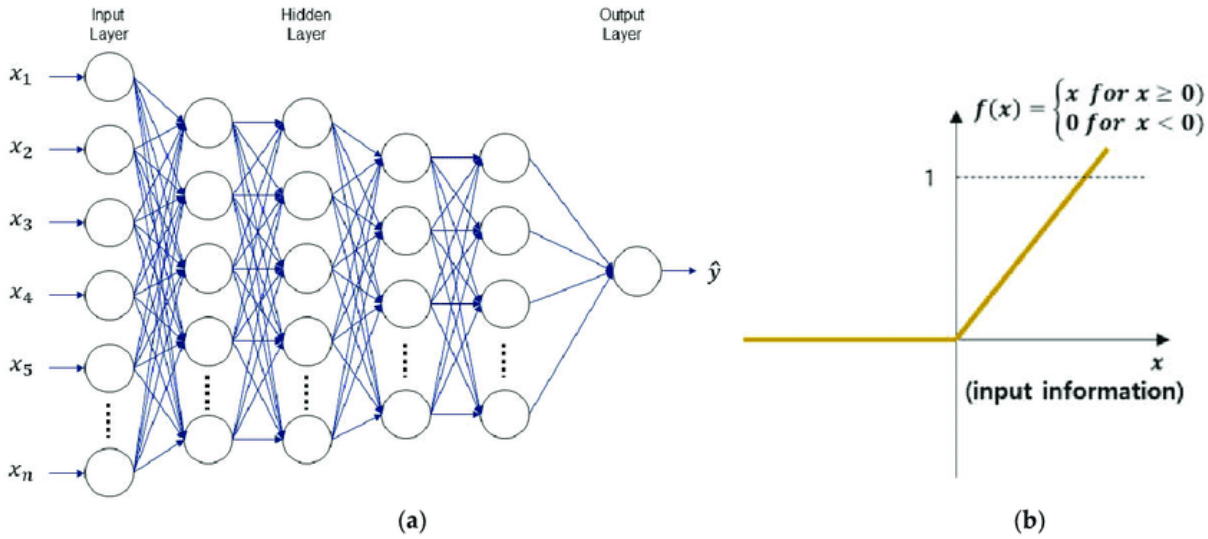



Figure: Deep ReLU networks (with some variants) account for most of the empirical success of deep learning in large-scale "pattern recognition".

- A hierarchy of layers to transform from the input space to the output space.
- Each hidden layer is an affine transformation followed by ReLU activation
- ReLU induces non-linearity and mitigates "gradient vanishing".

# Deep ReLU (Rectified Linear Unit) Networks

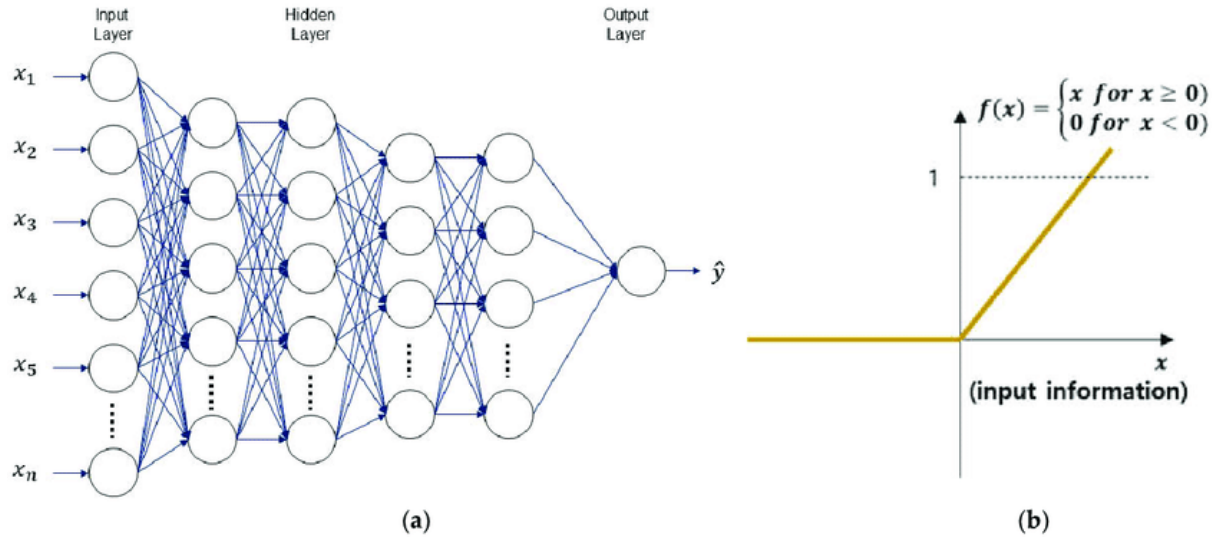



Figure: Deep ReLU networks (with some variants) account for most of the empirical success of deep learning in large-scale "pattern recognition".

- A hierarchy of layers to transform from the input space to the output space.
- Each hidden layer is an affine transformation followed by ReLU activation
- ReLU induces non-linearity and mitigates "gradient vanishing".

# Deep ReLU (Rectified Linear Unit) Networks

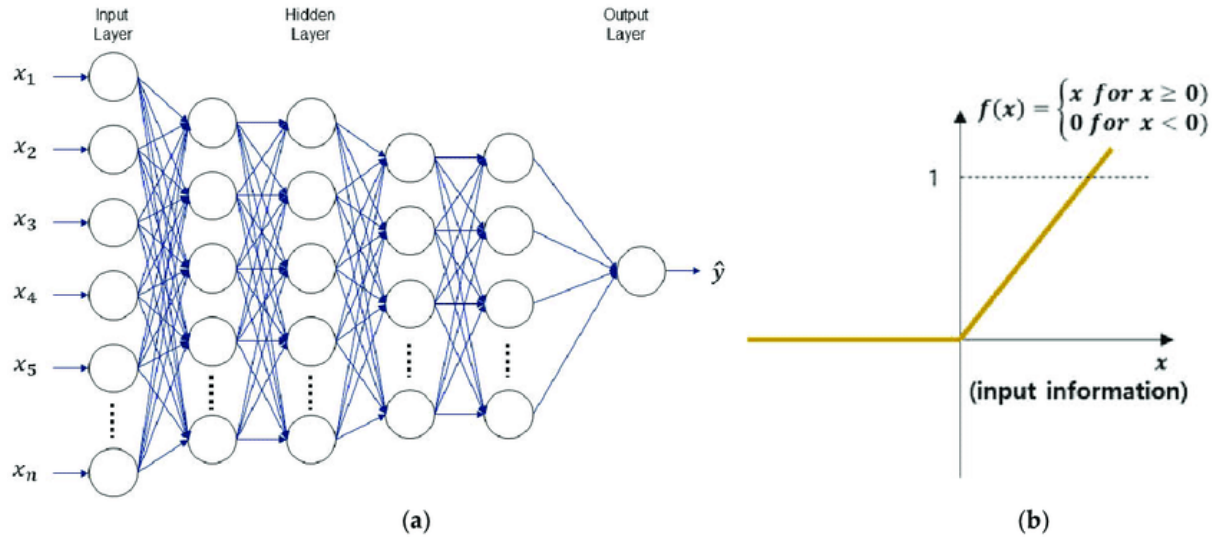



Figure: Deep ReLU networks (with some variants) account for most of the empirical success of deep learning in large-scale "pattern recognition".

- A hierarchy of layers to transform from the input space to the output space.
- Each hidden layer is an affine transformation followed by ReLU activation
- ReLU induces non-linearity and mitigates "gradient vanishing".

# Formal Description of Deep ReLU Networks

Let $\sigma(x) = \max\{x, 0\}$ be ReLU operator. We define a neural network with height $L$, width $W$, sparsity constraint $S$, and norm constraint $B$ as

$$\Phi(L, W, S, B) := \Big\{ (W^{(L)}\sigma(\cdot) + b^{(L)}) \circ \ldots \circ (W^{(2)}\sigma(\cdot) + b^{(2)}) \circ$$

$$(W^{(1)}Id(\cdot) + b^{(1)}) \Big| W^L \in \mathbb{R}^{1\times W}, b^L \in \mathbb{R}, W^{(1)} \in \mathbb{R}^{W\times d},$$

$$b^{(1)} \in \mathbb{R}^W, W^{(l)} \in \mathbb{R}^{W\times W}, b^{(l)} \in \mathbb{R}^W (1 < l < L),$$

$$\sum_{l=1}^{L} (\|W^{(l)}\|_0 + \|b^{(l)}\|_0) \leq S, \max_{1\leq l\leq L} \|W^{(l)}\|_\infty \vee \|b^{(l)}\|_\infty \leq B \Big\},$$

Function class:

$$\mathcal{F}_{NN} := \Big\{ f \in \Phi(L, W, S, B) : \|f\|_\infty \leq 1 \Big\}.$$

# Main Framework and Results

# Algorithm: Fitted Q-Evaluation (FQE)

**Algorithm 1** Fitted Q-Evaluation (FQE)

1: **Input:** MDP$(\mathcal{S}, \mathcal{A}, P, R, \gamma, \rho)$, function class $\mathcal{F}_{NN}$, number of iterations $K$, evaluation policy $\pi$, offline data $\mathcal{D}_n = \{(s_i, a_i, r_i, s_i')\}_{i=1}^n$ where $(s_i, a_i)_{i=1}^n \overset{i.i.d.}{\sim} \mu, s_i' \sim P(\cdot|s_i, a_i)$ and $r_i \sim R(s_i, a_i)$.
2: Initialize $\hat{Q}_0 \in \mathcal{F}_{NN}$.
3: **for** $k = 1$ **to** $K$ **do**
4: Compute $y_i = r_i + \gamma \int_{\mathcal{A}} \hat{Q}_{k-1}(s_i', a)\pi(da|s_i')$
5: $\hat{Q}_k \leftarrow \arg\min_{f \in \mathcal{F}_{NN}} \frac{1}{n} \sum_{i=1}^n (f(s_i, a_i) - y_i)^2$
6: **end for**
7: **Output:** $\hat{v}_K = \|\hat{Q}_K\|_{2,\rho} = \sqrt{\int \hat{Q}_K(s, a)^2 \rho(ds, da)}$

Iteratively compute an estimate using the least-squares estimation and the previous estimate.

Goal: Bounding $|\hat{v}_K - v^\pi|$.

# Regularity

- To obtain a nontrivial rate of convergence, it is necessary to make some assumption of the regularity of the target functions (no-free lunch theorem).
- This paper considers Besov smoothness which encompasses the classical Lipschitz, Hölder and Sobolev smoothness
- The smoothness is defined through the $L^p$-norm of its local oscillations

# Besov smoothness

## Definition (*Moduli of smoothness*)

For a function $f \in L^p(\mathcal{X})$ for some $p \in [1, \infty]$, we define its *r*th modulus of smoothness as

$$\omega_r^{t,p}(f) := \sup_{0 \le h \le t} \|\Delta_h^r(f)\|_p, t > 0, r \in \mathbb{N},$$

where the *r*th order translation-difference operator $\Delta_h^r = \Delta_h \circ \Delta_h^{r-1}$ is recursively defined as

$$\Delta_h^r(f)(\cdot) := (f(\cdot + h) - f(\cdot))^r = \sum_{k=0}^{r} \binom{r}{k} (-1)^{r-k} f(\cdot + k \cdot h).$$

# Some intuitions

- The quantity $\Delta_h^r(f)$ captures the local oscillation of function $f$ which is not necessarily differentiable
- In the case the $r$th order weak derivative $D^r f$ exists and is locally integrable, we approximately have

$$\frac{\Delta_h^r(f)(x)}{h^r} \approx D^r f(x) \text{ as } h \to 0.$$

# Some intuitions

- The quantity $\Delta_h^r(f)$ captures the local oscillation of function $f$ which is not necessarily differentiable
- In the case the $r$th order weak derivative $D^r f$ exists and is locally integrable, we approximately have

$$\frac{\Delta_h^r(f)(x)}{h^r} \approx D^r f(x) \text{ as } h \to 0.$$

# Besov spaces

## Definition (*Besov space* $B^{\alpha}_{p,q}(\mathcal{X})$)

For $1 \leq p, q \leq \infty$ and $\alpha > 0$, we define the norm $\|\cdot\|_{B^{\alpha}_{p,q}}$ of the Besov space $B^{\alpha}_{p,q}(\mathcal{X})$ as $\|f\|_{B^{\alpha}_{p,q}} := \|f\|_p + |f|_{B^{\alpha}_{p,q}}$ where

$$|f|_{B^{\alpha}_{p,q}} := \begin{cases} \left( \int_0^{\infty} \left( \frac{\omega^{t,p}_{\lfloor \alpha \rfloor + 1}(f)}{t^{\alpha}} \right)^q \frac{dt}{t} \right)^{1/q}, & 1 \leq q < \infty, \\ \sup_{t>0} \frac{\omega^{t,p}_{\lfloor \alpha \rfloor + 1}(f)}{t^{\alpha}}, & q = \infty, \end{cases}$$

is the Besov seminorm. Then, $B^{\alpha}_{p,q} := \{f \in L^p(\mathcal{X}) : \|f\|_{B^{\alpha}_{p,q}} < \infty\}$.

- A very rough intuition on the Besov seminorm: $|f|_{B^{\alpha}_{p,q}}$ is the $q$-norm of the $(\lfloor \alpha \rfloor + 1)$-th order (weak) derivative of $f$
- This intuition can be helpful to give an idea on Besov but do not take this intuition too seriously :)
- In this paper, we consider the Besov unit ball:

$$\bar{B}^{\alpha}_{p,q}(\mathcal{X}) := \{g \in B^{\alpha}_{p,q} : \|g\|_{B^{\alpha}_{p,q}} \leq 1 \text{ and } \|g\|_{\infty} \leq 1\}.$$

# Besov spaces

## Definition (*Besov space* $B^{\alpha}_{p,q}(\mathcal{X})$)

For $1 \leq p, q \leq \infty$ and $\alpha > 0$, we define the norm $\|\cdot\|_{B^{\alpha}_{p,q}}$ of the Besov space $B^{\alpha}_{p,q}(\mathcal{X})$ as $\|f\|_{B^{\alpha}_{p,q}} := \|f\|_p + |f|_{B^{\alpha}_{p,q}}$ where

$$|f|_{B^{\alpha}_{p,q}} := \begin{cases} \left( \int_0^{\infty} \left( \frac{\omega^{t,p}_{\lfloor \alpha \rfloor + 1}(f)}{t^{\alpha}} \right)^q \frac{dt}{t} \right)^{1/q}, & 1 \leq q < \infty, \\ \sup_{t>0} \frac{\omega^{t,p}_{\lfloor \alpha \rfloor + 1}(f)}{t^{\alpha}}, & q = \infty, \end{cases}$$

is the Besov seminorm. Then, $B^{\alpha}_{p,q} := \{f \in L^p(\mathcal{X}) : \|f\|_{B^{\alpha}_{p,q}} < \infty\}$.

- A very rough intuition on the Besov seminorm: $|f|_{B^{\alpha}_{p,q}}$ is the $q$-norm of the $(\lfloor \alpha \rfloor + 1)$-th order (weak) derivative of $f$
- This intuition can be helpful to give an idea on Besov but do not take this intuition too seriously :)
- In this paper, we consider the Besov unit ball:

$$\bar{B}^{\alpha}_{p,q}(\mathcal{X}) := \{g \in B^{\alpha}_{p,q} : \|g\|_{B^{\alpha}_{p,q}} \leq 1 \text{ and } \|g\|_{\infty} \leq 1\}.$$

# Besov spaces

## Definition (*Besov space* $B^{\alpha}_{p,q}(\mathcal{X})$)

For $1 \leq p, q \leq \infty$ and $\alpha > 0$, we define the norm $\|\cdot\|_{B^{\alpha}_{p,q}}$ of the Besov space $B^{\alpha}_{p,q}(\mathcal{X})$ as $\|f\|_{B^{\alpha}_{p,q}} := \|f\|_p + |f|_{B^{\alpha}_{p,q}}$ where

$$|f|_{B^{\alpha}_{p,q}} := \begin{cases} \left( \int_0^{\infty} \left( \frac{\omega^{t,p}_{\lfloor \alpha \rfloor + 1}(f)}{t^{\alpha}} \right)^q \frac{dt}{t} \right)^{1/q}, & 1 \leq q < \infty, \\ \sup_{t>0} \frac{\omega^{t,p}_{\lfloor \alpha \rfloor + 1}(f)}{t^{\alpha}}, & q = \infty, \end{cases}$$

is the Besov seminorm. Then, $B^{\alpha}_{p,q} := \{f \in L^p(\mathcal{X}) : \|f\|_{B^{\alpha}_{p,q}} < \infty\}$.

- A very rough intuition on the Besov seminorm: $|f|_{B^{\alpha}_{p,q}}$ is the $q$-norm of the $(\lfloor \alpha \rfloor + 1)$-th order (weak) derivative of $f$
- This intuition can be helpful to give an idea on Besov but do not take this intuition too seriously :)
- In this paper, we consider the Besov unit ball:

$$\bar{B}^{\alpha}_{p,q}(\mathcal{X}) := \{g \in B^{\alpha}_{p,q} : \|g\|_{B^{\alpha}_{p,q}} \leq 1 \text{ and } \|g\|_{\infty} \leq 1\}.$$

# Besov spaces

## Definition (*Besov space* $B^{\alpha}_{p,q}(\mathcal{X})$)

For $1 \leq p, q \leq \infty$ and $\alpha > 0$, we define the norm $\|\cdot\|_{B^{\alpha}_{p,q}}$ of the Besov space $B^{\alpha}_{p,q}(\mathcal{X})$ as $\|f\|_{B^{\alpha}_{p,q}} := \|f\|_p + |f|_{B^{\alpha}_{p,q}}$ where

$$|f|_{B^{\alpha}_{p,q}} := \begin{cases} \left( \int_0^{\infty} \left( \frac{\omega^{t,p}_{\lfloor \alpha \rfloor + 1}(f)}{t^{\alpha}} \right)^q \frac{dt}{t} \right)^{1/q}, & 1 \leq q < \infty, \\ \sup_{t>0} \frac{\omega^{t,p}_{\lfloor \alpha \rfloor + 1}(f)}{t^{\alpha}}, & q = \infty, \end{cases}$$

is the Besov seminorm. Then, $B^{\alpha}_{p,q} := \{f \in L^p(\mathcal{X}) : \|f\|_{B^{\alpha}_{p,q}} < \infty\}$.

- A very rough intuition on the Besov seminorm: $|f|_{B^{\alpha}_{p,q}}$ is the $q$-norm of the $(\lfloor \alpha \rfloor + 1)$-th order (weak) derivative of $f$
- This intuition can be helpful to give an idea on Besov but do not take this intuition too seriously :)
- In this paper, we consider the Besov unit ball:

$$\bar{B}^{\alpha}_{p,q}(\mathcal{X}) := \{g \in B^{\alpha}_{p,q} : \|g\|_{B^{\alpha}_{p,q}} \leq 1 \text{ and } \|g\|_{\infty} \leq 1\}.$$

# Assumption 1: Finite concentration coefficient

Assumption 1 (*Concentration coefficient*): There exists $\kappa_\mu < \infty$ such that $\|\frac{d\nu}{d\mu}\|_\infty \leq \kappa_\mu$ for any *realizable* distribution $\nu$. [a]

[a] $\nu$ is said to be realizable if there exists $t \geq 0$ and policy $\pi_1$ such that $\nu(s,a) = Pr(s_t = s, a_t = a | s_1 \sim \rho, \pi_1), \forall s, a$.

- Intuition: The sampling distribution $\mu$ is not too far from any realizable distribution, i.e., $\mu$ well covers the state-action spaces.
- This holds for a reasonably large class of MDPs (e.g., finite MDPs or MDPs with bounded transition kernels).

# Assumption 1: Finite concentration coefficient

Assumption 1 (*Concentration coefficient*): There exists $\kappa_\mu < \infty$ such that $\|\frac{d\nu}{d\mu}\|_\infty \leq \kappa_\mu$ for any *realizable* distribution $\nu$. [a]

[a] $\nu$ is said to be realizable if there exists $t \geq 0$ and policy $\pi_1$ such that $\nu(s,a) = Pr(s_t = s, a_t = a | s_1 \sim \rho, \pi_1), \forall s, a$.

- Intuition: The sampling distribution $\mu$ is not too far from any realizable distribution, i.e., $\mu$ well covers the state-action spaces.
- This holds for a reasonably large class of MDPs (e.g., finite MDPs or MDPs with bounded transition kernels).

# Assumption 2: Completeness

Assumption 2 (*Completeness*): $\forall f \in \mathcal{F}_{NN}(\mathcal{X}), T^\pi f \in \bar{B}^\alpha_{p,q}(\mathcal{X})$ for some $p, q \in [1, \infty]$ and $\alpha > \frac{d}{p \wedge 2}$.

- Intuition: Bellman operator $T^\pi$ applied on neural network functions is Besov (e.g., both the expected reward function $r(s, a)$ and kernel density function $P(s'|s, a)$ for any fixed $s'$ are Besov).
- Completeness is extremely common and is necessary in offline RL [Chen and Jiang, 2019].
- Our completeness is the most general of its kind that covers the previous cases considered in the offline RL literature.

# Assumption 2: Completeness

Assumption 2 (*Completeness*): $\forall f \in \mathcal{F}_{NN}(\mathcal{X}), T^\pi f \in \bar{B}^\alpha_{p,q}(\mathcal{X})$ for some $p, q \in [1, \infty]$ and $\alpha > \frac{d}{p \wedge 2}$.

- Intuition: Bellman operator $T^\pi$ applied on neural network functions is Besov (e.g., both the expected reward function $r(s, a)$ and kernel density function $P(s'|s, a)$ for any fixed $s'$ are Besov).
- Completeness is extremely common and is necessary in offline RL [Chen and Jiang, 2019].
- Our completeness is the most general of its kind that covers the previous cases considered in the offline RL literature.

# Assumption 2: Completeness

Assumption 2 (*Completeness*): $\forall f \in \mathcal{F}_{NN}(\mathcal{X}), T^\pi f \in \bar{B}^\alpha_{p,q}(\mathcal{X})$ for some $p, q \in [1, \infty]$ and $\alpha > \frac{d}{p \wedge 2}$.

- Intuition: Bellman operator $T^\pi$ applied on neural network functions is Besov (e.g., both the expected reward function $r(s, a)$ and kernel density function $P(s'|s, a)$ for any fixed $s'$ are Besov).
- Completeness is extremely common and is necessary in offline RL [Chen and Jiang, 2019].
- Our completeness is the most general of its kind that covers the previous cases considered in the offline RL literature.

# Assumption 2: Completeness

Assumption 2 (*Completeness*): $\forall f \in \mathcal{F}_{NN}(\mathcal{X}), T^\pi f \in \bar{B}^\alpha_{p,q}(\mathcal{X})$ for some $p, q \in [1, \infty]$ and $\alpha > \frac{d}{p \wedge 2}$.

- Intuition: Bellman operator $T^\pi$ applied on neural network functions is Besov (e.g., both the expected reward function $r(s, a)$ and kernel density function $P(s'|s, a)$ for any fixed $s'$ are Besov).
- Completeness is extremely common and is necessary in offline RL [Chen and Jiang, 2019].
- Our completeness is the most general of its kind that covers the previous cases considered in the offline RL literature.

# Main theorem

**Theorem**

*Under the completeness and finite concentration coefficient assumptions, for any $\epsilon > 0$ and $\delta \in (0,1)$, with probability at least $1-\delta$, we have $|v_K - v^\pi| \leq \frac{\sqrt{\kappa_\mu}}{1-\gamma}\epsilon + \frac{2\gamma^{K/2}}{(1-\gamma)^{1/2}}$, where*

$$n \gtrsim \left(\frac{1}{\epsilon^2}\right)^{1+\frac{d}{\alpha}} \log^6 n + \frac{1}{\epsilon^2}(\log(1/\delta) + \log\log n).$$

# Interpretation

- Significance: To our knowledge, this is the most general and "proper" analysis of offline RL with deep neural network
- Tightness: Our derived sample rate $\left(\frac{1}{\epsilon^2}\right)^{1+\frac{d}{\alpha}}$ (ignoring the log factor and the factor pertaining to $\kappa_\mu$ and $1/(1-\gamma)$) nearly matches ( but is a bit worse than) the minimax-optimal rate $\left(\frac{1}{\epsilon^2}\right)^{1+\frac{d}{2\alpha}}$ in nonparametric regression. This is understandable as we deal with an additional challenge: data-dependent structure.
- Improved bound: It is possible to retain the rate $\left(\frac{1}{\epsilon^2}\right)^{1+\frac{d}{2\alpha}}$ but at the cost that the sample complexity scales with $K$. $K$ could be very large in practice. Our bound is better than that in [Yang et al. (2019)]
- Curse of dimensionality: In high-dimensional OPE, FQE with deep ReLU networks can suffer from the curse of dimensionality.

# Interpretation

- Significance: To our knowledge, this is the most general and "proper" analysis of offline RL with deep neural network
- Tightness: Our derived sample rate $\left(\frac{1}{\epsilon^2}\right)^{1+\frac{d}{\alpha}}$ (ignoring the log factor and the factor pertaining to $\kappa_\mu$ and $1/(1-\gamma)$) nearly matches ( but is a bit worse than) the minimax-optimal rate $\left(\frac{1}{\epsilon^2}\right)^{1+\frac{d}{2\alpha}}$ in nonparametric regression. This is understandable as we deal with an additional challenge: data-dependent structure.
- Improved bound: It is possible to retain the rate $\left(\frac{1}{\epsilon^2}\right)^{1+\frac{d}{2\alpha}}$ but at the cost that the sample complexity scales with $K$. $K$ could be very large in practice. Our bound is better than that in [Yang et al. (2019)]
- Curse of dimensionality: In high-dimensional OPE, FQE with deep ReLU networks can suffer from the curse of dimensionality.

# Interpretation

- Significance: To our knowledge, this is the most general and "proper" analysis of offline RL with deep neural network
- Tightness: Our derived sample rate $\left(\frac{1}{\epsilon^2}\right)^{1+\frac{d}{\alpha}}$ (ignoring the log factor and the factor pertaining to $\kappa_\mu$ and $1/(1-\gamma)$) nearly matches ( but is a bit worse than) the minimax-optimal rate $\left(\frac{1}{\epsilon^2}\right)^{1+\frac{d}{2\alpha}}$ in nonparametric regression. This is understandable as we deal with an additional challenge: data-dependent structure.
- Improved bound: It is possible to retain the rate $\left(\frac{1}{\epsilon^2}\right)^{1+\frac{d}{2\alpha}}$ but at the cost that the sample complexity scales with $K$. $K$ could be very large in practice. Our bound is better than that in [Yang et al. (2019)]
- Curse of dimensionality: In high-dimensional OPE, FQE with deep ReLU networks can suffer from the curse of dimensionality.

# Interpretation

- Significance: To our knowledge, this is the most general and "proper" analysis of offline RL with deep neural network
- Tightness: Our derived sample rate $\left(\frac{1}{\epsilon^2}\right)^{1+\frac{d}{\alpha}}$ (ignoring the log factor and the factor pertaining to $\kappa_\mu$ and $1/(1-\gamma)$) nearly matches ( but is a bit worse than) the minimax-optimal rate $\left(\frac{1}{\epsilon^2}\right)^{1+\frac{d}{2\alpha}}$ in nonparametric regression. This is understandable as we deal with an additional challenge: data-dependent structure.
- Improved bound: It is possible to retain the rate $\left(\frac{1}{\epsilon^2}\right)^{1+\frac{d}{2\alpha}}$ but at the cost that the sample complexity scales with $K$. $K$ could be very large in practice. Our bound is better than that in [Yang et al. (2019)]
- Curse of dimensionality: In high-dimensional OPE, FQE with deep ReLU networks can suffer from the curse of dimensionality.

# Conclusion

# On the mitigation of the curse of dimensionality

- We precisely quantify the curse of dimensionality in high-dimensional offline RL
- Curse of dimensionality is unavoidable in Besov spaces, but it can be further mitigated for Besov spaces with *mixed smoothness*.

# On the mitigation of the curse of dimensionality

- We precisely quantify the curse of dimensionality in high-dimensional offline RL
- Curse of dimensionality is unavoidable in Besov spaces, but it can be further mitigated for Besov spaces with *mixed smoothness*.

# On the assumption of "uniform coverage" for the offline data $\mathcal{D}_n$

- Currently we assume the offline data is "benign": the offline data covers the state-action spaces uniformly.
- This may not hold in many cases in practice.
- A largely open question:

> What is the minimum structural assumption on the data coverage that guarantees sample efficiency in offline data?

# On the assumption of "uniform coverage" for the offline data $\mathcal{D}_n$

- Currently we assume the offline data is "benign": the offline data covers the state-action spaces uniformly.
- This may not hold in many cases in practice.
- A largely open question:

> What is the minimum structural assumption on the data coverage that guarantees sample efficiency in offline data?

# On the assumption of "uniform coverage" for the offline data $\mathcal{D}_n$

- Currently we assume the offline data is "benign": the offline data covers the state-action spaces uniformly.
- This may not hold in many cases in practice.
- A largely open question:

> What is the minimum structural assumption on the data coverage that guarantees sample efficiency in offline data?

# Beyond policy evaluation and toward offline learning

- Currently we focus on off-policy evaluation but offline learning is an ultimate goal for offline RL
- Pessimism is a promising direction to address offline learning: Extrapolate beyond the data support is erroneous, pessimism penalizes over-extrapolation.
- Current working project: "On Pessimistic Extrapolation in Offline RL with General Function Approximation" with ambitious goals of:
  - Providing a general statistical theory of offline RL;
  - Uncovering the minimal structural assumption on the (offline) data coverage thereafter;
  - (My bet) Largely closing the offline RL research and people will be happier with using offline RL in practical settings

# Beyond policy evaluation and toward offline learning

- Currently we focus on off-policy evaluation but offline learning is an ultimate goal for offline RL
- Pessimism is a promising direction to address offline learning: Extrapolate beyond the data support is erroneous, pessimism penalizes over-extrapolation.
- Current working project: "On Pessimistic Extrapolation in Offline RL with General Function Approximation" with ambitious goals of:
  - Providing a general statistical theory of offline RL;
  - Uncovering the minimal structural assumption on the (offline) data coverage thereafter;
  - (My bet) Largely closing the offline RL research and people will be happier with using offline RL in practical settings

# Beyond policy evaluation and toward offline learning

- Currently we focus on off-policy evaluation but offline learning is an ultimate goal for offline RL
- Pessimism is a promising direction to address offline learning: Extrapolate beyond the data support is erroneous, pessimism penalizes over-extrapolation.
- Current working project: "On Pessimistic Extrapolation in Offline RL with General Function Approximation" with ambitious goals of:
  - Providing a general statistical theory of offline RL;
  - Uncovering the minimal structural assumption on the (offline) data coverage thereafter;
  - (My bet) Largely closing the offline RL research and people will be happier with using offline RL in practical settings

# Beyond policy evaluation and toward offline learning

- Currently we focus on off-policy evaluation but offline learning is an ultimate goal for offline RL
- Pessimism is a promising direction to address offline learning: Extrapolate beyond the data support is erroneous, pessimism penalizes over-extrapolation.
- Current working project: "On Pessimistic Extrapolation in Offline RL with General Function Approximation" with ambitious goals of:
  - Providing a general statistical theory of offline RL;
  - Uncovering the minimal structural assumption on the (offline) data coverage thereafter;
  - (My bet) Largely closing the offline RL research and people will be happier with using offline RL in practical settings

# Beyond policy evaluation and toward offline learning

- Currently we focus on off-policy evaluation but offline learning is an ultimate goal for offline RL
- Pessimism is a promising direction to address offline learning: Extrapolate beyond the data support is erroneous, pessimism penalizes over-extrapolation.
- Current working project: "On Pessimistic Extrapolation in Offline RL with General Function Approximation" with ambitious goals of:
  - Providing a general statistical theory of offline RL;
  - Uncovering the minimal structural assumption on the (offline) data coverage thereafter;
  - (My bet) Largely closing the offline RL research and people will be happier with using offline RL in practical settings

# Beyond policy evaluation and toward offline learning

- Currently we focus on off-policy evaluation but offline learning is an ultimate goal for offline RL
- Pessimism is a promising direction to address offline learning: Extrapolate beyond the data support is erroneous, pessimism penalizes over-extrapolation.
- Current working project: "On Pessimistic Extrapolation in Offline RL with General Function Approximation" with ambitious goals of:
  - Providing a general statistical theory of offline RL;
  - Uncovering the minimal structural assumption on the (offline) data coverage thereafter;
  - (My bet) Largely closing the offline RL research and people will be happier with using offline RL in practical settings

# Proof Sketch

# Step 1: Error propagation through iterations

- Error propagation:

$$|\hat{v}_K - v^\pi| \leq \frac{\sqrt{\kappa_\mu}}{1-\gamma} \max_{0 \leq k \leq K-1} \|\hat{Q}_{k+1} - T^\pi \hat{Q}_k\|_{2,\mu} + \frac{2V_{max}\gamma^{K/2}}{(1-\gamma)^{1/2}},$$

- Now we need to bound $\max_{0 \leq k \leq K-1} \|\hat{Q}_{k+1} - T^\pi \hat{Q}_k\|_{2,\mu}$. Is this a standard nonparametric regression?

# Step 1: Error propagation through iterations

- Error propagation:

$$|\hat{v}_K - v^\pi| \leq \frac{\sqrt{\kappa_\mu}}{1-\gamma} \max_{0 \leq k \leq K-1} \|\hat{Q}_{k+1} - T^\pi \hat{Q}_k\|_{2,\mu} + \frac{2 V_{max} \gamma^{K/2}}{(1-\gamma)^{1/2}},$$

- Now we need to bound $\max_{0 \leq k \leq K-1} \|\hat{Q}_{k+1} - T^\pi \hat{Q}_k\|_{2,\mu}$. Is this a standard nonparametric regression?

# Step 2: Uniform convergence argument

- The offline data $\mathcal{D} = \{(s_i, a_i, r_i.s_i')\}_{i=1}^n$. For any function $f$, define

$$x_i := (s_i, a_i)$$
$$y_i^f := r_i + \gamma \langle f(s_i', \cdot), \pi(\cdot, s_i') \rangle_{\mathcal{A}}$$

- For any **fixed** $f$, we have

$$\mathbb{E}\left[ (T^\pi f)(x_i) - y_i^f \middle| x_i \right] = 0$$

- However, as $\hat{Q}_{k+1}$ depends on $x_i$,

$$\mathbb{E}\left[ (T^\pi \hat{Q}_k)(x_i) - y_i^{\hat{Q}_{k+1}} \middle| x_i \right] \neq 0$$

$\rightarrow$ Uniform convergence argument comes to rescue!

# Step 2: Uniform convergence argument

- The offline data $\mathcal{D} = \{(s_i, a_i, r_i.s_i')\}_{i=1}^n$. For any function $f$, define

$$x_i := (s_i, a_i)$$
$$y_i^f := r_i + \gamma \langle f(s_i', \cdot), \pi(\cdot, s_i') \rangle_{\mathcal{A}}$$

- For any **fixed** $f$, we have

$$\mathbb{E}\left[ (T^\pi f)(x_i) - y_i^f \middle| x_i \right] = 0$$

- However, as $\hat{Q}_{k+1}$ depends on $x_i$,

$$\mathbb{E}\left[ (T^\pi \hat{Q}_k)(x_i) - y_i^{\hat{Q}_{k+1}} \middle| x_i \right] \neq 0$$

$\rightarrow$ Uniform convergence argument comes to rescue!

# Step 2: Uniform convergence argument

- The offline data $\mathcal{D} = \{(s_i, a_i, r_i.s_i')\}_{i=1}^n$. For any function $f$, define

$$x_i := (s_i, a_i)$$
$$y_i^f := r_i + \gamma \langle f(s_i', \cdot), \pi(\cdot, s_i') \rangle_{\mathcal{A}}$$

- For any **fixed** $f$, we have

$$\mathbb{E}\left[ (T^\pi f)(x_i) - y_i^f \middle| x_i \right] = 0$$

- However, as $\hat{Q}_{k+1}$ depends on $x_i$,

$$\mathbb{E}\left[ (T^\pi \hat{Q}_k)(x_i) - y_i^{\hat{Q}_{k+1}} \middle| x_i \right] \neq 0$$

$\rightarrow$ Uniform convergence argument comes to rescue!

# Uniform convergence error decomposition

- For any $Q \in \mathcal{F}_{NN}$, define
  - Projection of $T^\pi Q$ onto $\mathcal{F}_{NN}$ :

$$f_\perp^Q := \arg\inf_{f \in \mathcal{F}_{NN}} \|f - T^\pi Q\|_{2,\mu}$$

  - Empirical risk minimizer:

$$\hat{f}^Q := \arg\inf_{f \in \mathcal{F}_{NN}} \sum_{i=1}^{n} (f(x_i) - y_i^Q)^2$$

- Define $l_f(x) := (x - f(x))^2$, we have

$$\max_{0 \le k \le K-1} \|\hat{Q}_{k+1} - T^\pi \hat{Q}_k\|_{2,\mu}^2 \le \underbrace{\sup_{Q \in \mathcal{F}_{NN}} (\mathbb{E} - \mathbb{E}_n)(l_{\hat{f}^Q} - l_{f_*^Q})}_{I_1, \text{empirical process term}} + \underbrace{\sup_{Q \in \mathcal{F}_{NN}} \mathbb{E}_n(l_{f_\perp^Q} - l_{f_*^Q})}_{I_2, \text{bias term}},$$

# Uniform convergence error decomposition

- For any $Q \in \mathcal{F}_{NN}$, define
  - Projection of $T^\pi Q$ onto $\mathcal{F}_{NN}$ :

$$f_\perp^Q := \arg\inf_{f \in \mathcal{F}_{NN}} \|f - T^\pi Q\|_{2,\mu}$$

  - Empirical risk minimizer:

$$\hat{f}^Q := \arg\inf_{f \in \mathcal{F}_{NN}} \sum_{i=1}^{n} (f(x_i) - y_i^Q)^2$$

- Define $l_f(x) := (x - f(x))^2$, we have

$$\max_{0 \le k \le K-1} \|\hat{Q}_{k+1} - T^\pi \hat{Q}_k\|_{2,\mu}^2 \le \underbrace{\sup_{Q \in \mathcal{F}_{NN}} (\mathbb{E} - \mathbb{E}_n)(l_{\hat{f}^Q} - l_{f_*^Q})}_{I_1,\text{empirical process term}} + \underbrace{\sup_{Q \in \mathcal{F}_{NN}} \mathbb{E}_n(l_{f_\perp^Q} - l_{f_*^Q})}_{I_2,\text{bias term}},$$

# Uniform convergence error decomposition

- For any $Q \in \mathcal{F}_{NN}$, define
  - Projection of $T^\pi Q$ onto $\mathcal{F}_{NN}$ :

$$f_\perp^Q := \arg\inf_{f \in \mathcal{F}_{NN}} \|f - T^\pi Q\|_{2,\mu}$$

  - Empirical risk minimizer:

$$\hat{f}^Q := \arg\inf_{f \in \mathcal{F}_{NN}} \sum_{i=1}^{n} (f(x_i) - y_i^Q)^2$$

- Define $l_f(x) := (x - f(x))^2$, we have

$$\max_{0 \le k \le K-1} \|\hat{Q}_{k+1} - T^\pi \hat{Q}_k\|_{2,\mu}^2 \le \underbrace{\sup_{Q \in \mathcal{F}_{NN}} (\mathbb{E} - \mathbb{E}_n)(l_{\hat{f}^Q} - l_{f_*^Q})}_{I_1, \text{empirical process term}} + \underbrace{\sup_{Q \in \mathcal{F}_{NN}} \mathbb{E}_n(l_{f_\perp^Q} - l_{f_*^Q})}_{I_2, \text{bias term}},$$

# Uniform convergence error decomposition

- For any $Q \in \mathcal{F}_{NN}$, define
  - Projection of $T^\pi Q$ onto $\mathcal{F}_{NN}$ :

$$f_\perp^Q := \arg\inf_{f \in \mathcal{F}_{NN}} \|f - T^\pi Q\|_{2,\mu}$$

  - Empirical risk minimizer:

$$\hat{f}^Q := \arg\inf_{f \in \mathcal{F}_{NN}} \sum_{i=1}^{n} (f(x_i) - y_i^Q)^2$$

- Define $l_f(x) := (x - f(x))^2$, we have

$$\max_{0 \le k \le K-1} \|\hat{Q}_{k+1} - T^\pi \hat{Q}_k\|_{2,\mu}^2 \le$$

$$\underbrace{\sup_{Q \in \mathcal{F}_{NN}} (\mathbb{E} - \mathbb{E}_n)(l_{\hat{f}^Q} - l_{f_*^Q})}_{I_1, \text{empirical process term}} + \underbrace{\sup_{Q \in \mathcal{F}_{NN}} \mathbb{E}_n(l_{f_\perp^Q} - l_{f_*^Q})}_{I_2, \text{bias term}},$$

# Next steps

- **Step 3**: Bounding the bias term using a concentration inequality similar to "uniform convergence" Bernstein's inequality.

**Lemma 5** (Theorem 11.6 in (Györfi et al., 2002)). *Let $B \geq 1$ and $\mathcal{F}$ be a set of functions $f : \mathbb{R}^d \to [0, B]$. Let $Z_1, ..., Z_n$ be i.i.d. $\mathbb{R}^d$-valued random variables. For any $\alpha > 0$, $0 < \epsilon < 1$, and $n \geq 1$, we have*

$$P\left\{\sup_{f\in\mathcal{F}} \frac{\frac{1}{n}\sum_{i=1}^{n} f(Z_i) - \mathbb{E}[f(Z)]}{\alpha + \frac{1}{n}\sum_{i=1}^{n} f(Z_i) + \mathbb{E}[f(Z)]} > \epsilon\right\} \leq 4\mathbb{E}N(\frac{\alpha\epsilon}{5}, \mathcal{F}|Z_1^n, n^{-1}\|\cdot\|_1)\exp\left(\frac{-3\epsilon^2\alpha n}{40B}\right).$$

- **Step 4**: Bounding the empirical process term using **local** Rademacher complexity

**Lemma 4** ((Bartlett et al., 2005)). *Let $r > 0$ and let*

$$\mathcal{F} \subseteq \{f : \mathcal{X} \to [a, b] : Var[f(X_1)] \leq r\}.$$

1. *For any $\lambda > 0$, we have with probability at least $1 - e^{-\lambda}$,*

$$\sup_{f\in\mathcal{F}} (\mathbb{E}f - \mathbb{E}_n f) \leq \inf_{\alpha>0}\left(2(1+\alpha)\mathbb{E}\left[R_n\mathcal{F}\right] + \sqrt{\frac{2r\lambda}{n}} + (b-a)\left(\frac{1}{3} + \frac{1}{\alpha}\right)\frac{\lambda}{n}\right).$$

# Next steps

- **Step 3**: Bounding the bias term using a concentration inequality similar to "uniform convergence" Bernstein's inequality.

**Lemma 5** (Theorem 11.6 in (Györfi et al., 2002)). *Let $B \geq 1$ and $\mathcal{F}$ be a set of functions $f : \mathbb{R}^d \to [0, B]$. Let $Z_1, ..., Z_n$ be i.i.d. $\mathbb{R}^d$-valued random variables. For any $\alpha > 0$, $0 < \epsilon < 1$, and $n \geq 1$, we have*

$$P\left\{\sup_{f\in\mathcal{F}} \frac{\frac{1}{n}\sum_{i=1}^{n} f(Z_i) - \mathbb{E}[f(Z)]}{\alpha + \frac{1}{n}\sum_{i=1}^{n} f(Z_i) + \mathbb{E}[f(Z)]} > \epsilon\right\} \leq 4\mathbb{E}N(\frac{\alpha\epsilon}{5}, \mathcal{F}|Z_1^n, n^{-1}\|\cdot\|_1)\exp\left(\frac{-3\epsilon^2\alpha n}{40B}\right).$$

- **Step 4**: Bounding the empirical process term using **local** Rademacher complexity

**Lemma 4** ((Bartlett et al., 2005)). *Let $r > 0$ and let*

$$\mathcal{F} \subseteq \{f : \mathcal{X} \to [a, b] : Var[f(X_1)] \leq r\}.$$

*1. For any $\lambda > 0$, we have with probability at least $1 - e^{-\lambda}$,*

$$\sup_{f\in\mathcal{F}} (\mathbb{E}f - \mathbb{E}_n f) \leq \inf_{\alpha>0}\left(2(1+\alpha)\mathbb{E}\left[R_n\mathcal{F}\right] + \sqrt{\frac{2r\lambda}{n}} + (b-a)\left(\frac{1}{3} + \frac{1}{\alpha}\right)\frac{\lambda}{n}\right).$$

# Next steps

- **Step 5**: Bounding the entire $I_1 + I_2$ using a localization argument and sub-root argument

Thus, with probability at least $1-\delta$, we have

$$\boxed{\|\hat{f}^{Q_{k-1}} - f_*^{Q_{k-1}}\|_{2,\mu} \lesssim d_{\mathcal{F}_{NN}} + \epsilon' + \sqrt{\frac{\log(2l/\delta)}{n}} + \sqrt{r_*}} \tag{8}$$

where

$$n \approx \frac{V_{\max}^2}{\epsilon'^2}\left(\log(8l/\delta) + \log \mathbb{E}N(\frac{\epsilon'^2}{40V_{\max}}, (\mathcal{F}_{NN} - T^\pi \mathcal{F}_{NN})|\{x_i\}_{i=1}^n, n^{-1}\|\cdot\|_1)\right).$$

- **Step 6**: Minimize the upper error bound from Step 5 with respect to the the network architecture. The optimized ReLU network architecture:

$$L \asymp \log N, W \asymp N, S \asymp N, \text{ and } B \asymp N^{d^{-1}+\nu^{-1}},$$

$$\text{where } \nu := \frac{\alpha - \delta}{2\delta}, \delta := d(p^{-1} - (1 + \lfloor\alpha\rfloor)^{-1})_+,$$

$$N \asymp n^{\frac{1}{2}(2\beta+1)\frac{d}{2\alpha+d}}, \beta = \left(2 + \frac{d^2}{\alpha(\alpha+d)}\right)^{-1}.$$

# Next steps

- **Step 5**: Bounding the entire $I_1 + I_2$ using a localization argument and sub-root argument

Thus, with probability at least $1-\delta$, we have

$$\boxed{\|\hat{f}^{Q_{k-1}} - f_*^{Q_{k-1}}\|_{2,\mu} \lesssim d_{\mathcal{F}_{NN}} + \epsilon' + \sqrt{\frac{\log(2l/\delta)}{n}} + \sqrt{r_*}} \tag{8}$$

where

$$n \approx \frac{V_{\max}^2}{\epsilon'^2}\left(\log(8l/\delta) + \log \mathbb{E}N(\frac{\epsilon'^2}{40V_{\max}}, (\mathcal{F}_{NN} - T^\pi \mathcal{F}_{NN})|\{x_i\}_{i=1}^n, n^{-1}\|\cdot\|_1))\right).$$

- **Step 6**: Minimize the upper error bound from Step 5 with respect to the the network architecture. The optimized ReLU network architecture:

$$L \asymp \log N, W \asymp N, S \asymp N, \text{ and } B \asymp N^{d^{-1}+\nu^{-1}},$$

$$\text{where } \nu := \frac{\alpha - \delta}{2\delta}, \delta := d(p^{-1} - (1 + \lfloor\alpha\rfloor)^{-1})_+,$$

$$N \asymp n^{\frac{1}{2}(2\beta+1)\frac{d}{2\alpha+d}}, \beta = \left(2 + \frac{d^2}{\alpha(\alpha+d)}\right)^{-1}.$$

Thank you for your attention.