Deep Transfer Offline Reinforcement Learning

under nonstationary environments

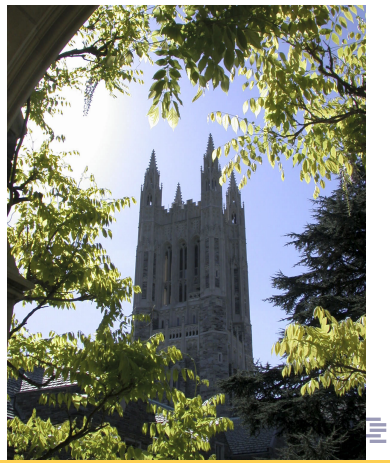
Jianqing Fan

Princeton University

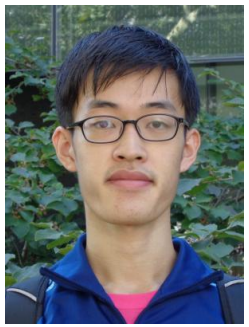
https://fan.princeton.edu/

Jinhang Chai, **Elynn Chen**

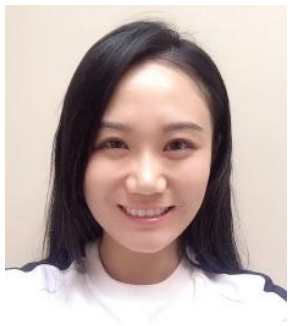




Outlines

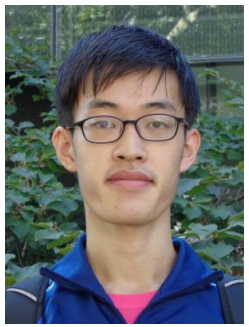


Jinhang Chai

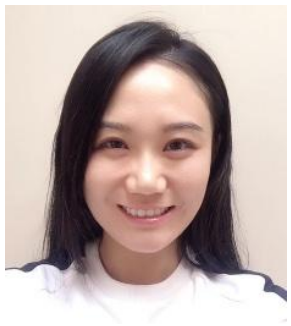


Elynn Chen

Outlines

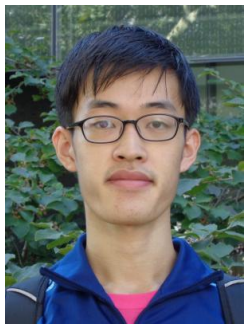


Jinhang Chai

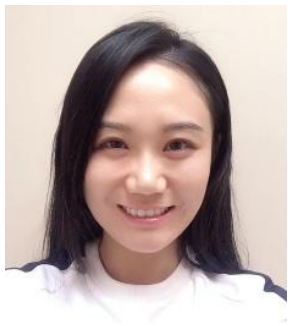


Elynn Chen

Outlines



Jinhang Chai



Elynn Chen

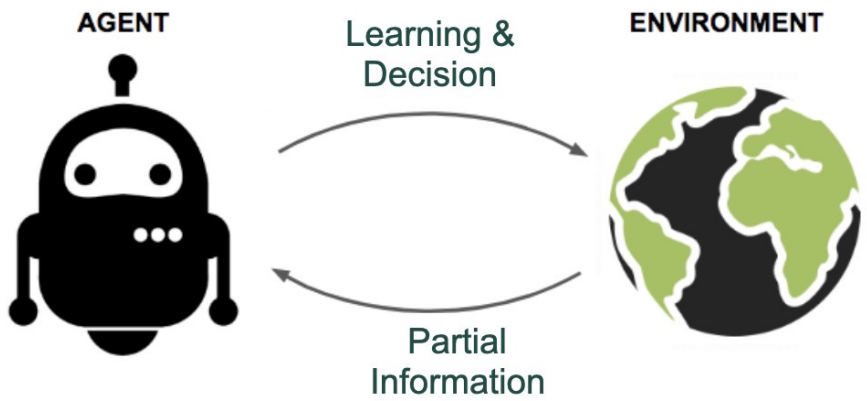
Introduction

Reinforcement Learning

■Dynamic decisions from **unknown** environments with **random** outcomes

Approach:

★Learning ★Optimization (planning)



★ **Business decision**: inventory control or dynamic pricing

★ **Clinical decision**: personalized health treatment.

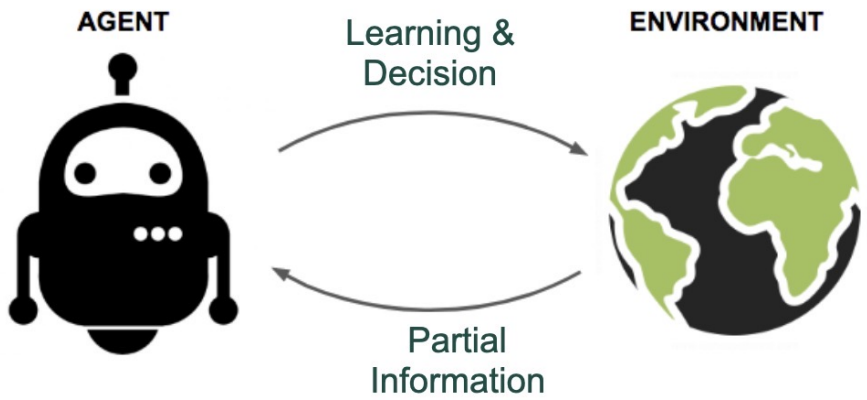
★ **Intelligent tutoring**: personalized math tutoring.

Reinforcement Learning

■Dynamic decisions from **unknown** environments with **random** outcomes

Approach:

★Learning ★Optimization (planning)



★ **Business decision**: inventory control or dynamic pricing

★ **Clinical decision**: personalized health treatment.

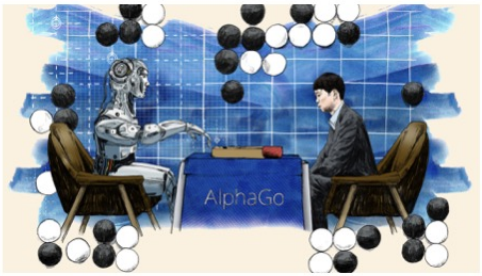
★ **Intelligent tutoring**: personalized math tutoring.

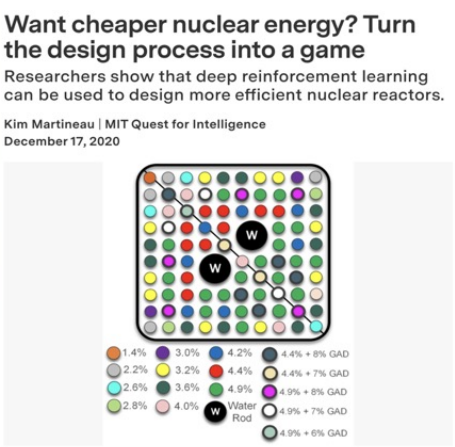
Synthetic and Scientific Systems

★Abundant data available from simulators or synthetic environments.

Open AI Dota Game-Playing

DeepMind AlphaGo





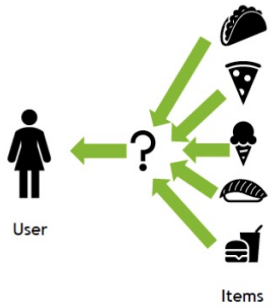
Reactor Design

Drug Design

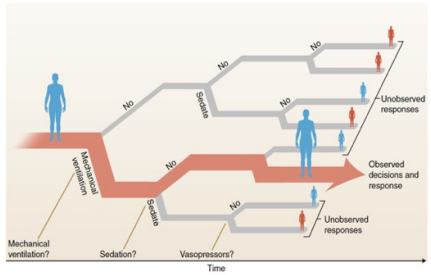
Scarcity of Interaction Data

★ Societal domains are different.

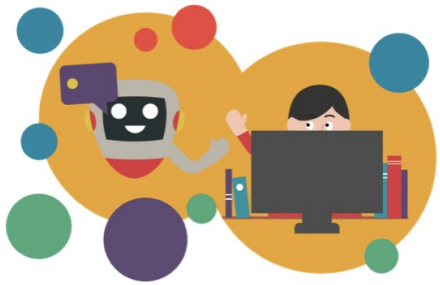
AI Marketing



AI Clinician



AI Tutor



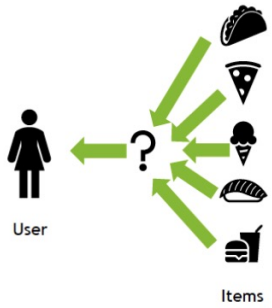
★ Data collection in societal problems:

- Expensive
- Time-consuming
- Risky or unethical

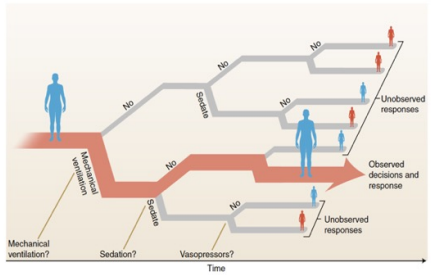
Scarcity of Interaction Data

★ Societal domains are different.

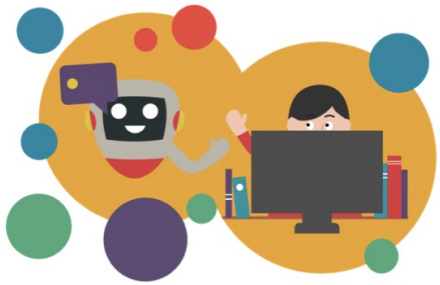
AI Marketing



AI Clinician



AI Tutor



★ Data collection in societal problems:

- Expensive
- Time-consuming
- Risky or unethical

Cost Reduction in Data Collection

★Leverage data from **similar products, users, or markets** **(transfer learning)**

Example: Use data from existing products to optimize pricing for a newly launched product, avoiding cold-start issues.

Cost Reduction in Data Collection

★Leverage data from **similar products, users, or markets** **(transfer learning)**

Example: Use data from existing products to optimize pricing for a newly launched product, avoiding cold-start issues.

Challenges: Heterogeneity across populations, locations, time, or contexts.

High-Stakes Scenarios

★Deploying exploratory or random actions is ethically unacceptable.

High-Stakes Scenarios

★Deploying exploratory or random actions is ethically unacceptable.

Transfer RL: Enables safe and rapid deployment of pre-trained policies, avoiding the need for extensive retraining or experiments in the field.

High-Stakes Scenarios

★Deploying exploratory or random actions is ethically unacceptable.

Transfer RL: Enables safe and rapid deployment of pre-trained policies, avoiding the need for extensive retraining or experiments in the field.

Example: For a new disease, it is unenthical to do random treatments, but use the treatment strategy from a similar disease.

Motivations for Transfer RL with heterogeneity

Data Scarcity: New products, markets, or users: Avoid cold-starts solutions.

Motivations for Transfer RL with heterogeneity

Data Scarcity: New products, markets, or users: Avoid cold-starts solutions.

Population Heterogeneity: Cross-Market (domain) Adaptation.

Motivations for Transfer RL with heterogeneity

Data Scarcity: New products, markets, or users: Avoid cold-starts solutions.

Population Heterogeneity: Cross-Market (domain) Adaptation.

High-Stakes: Random exploration is ethically unacceptable.

Contributions of this work

Transfer RL for Non-Stationary MDP with finite horizon using

- backward inductive Q-learning
- Neural networks

Rigorous Framework:
- Task similarity
- General function approx framework
- Instantiate on Deep NN (RKHS in Appendix)

Contributions of this work

Transfer RL for Non-Stationary MDP with finite horizon using

- backward inductive Q-learning
- Neural networks

Rigorous Framework: - Task similarity - General function approx framework

- Instantiate on Deep NN (RKHS in Appendix)

Contributions of this work

Transfer RL for Non-Stationary MDP with finite horizon using

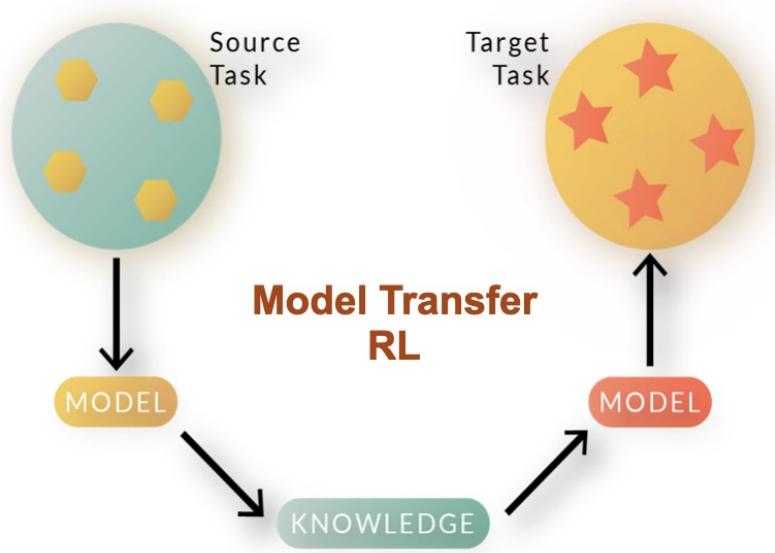
- backward inductive Q-learning
- Neural networks

Rigorous Framework: • Task similarity • General function approx framework

- Instantiate on Deep NN (RKHS in Appendix)

Challenges: Responses are **not** observable

Novelty: • Retargeting • Reweighting



Related Work: RL & NN

- **Offline RL**: Sutton and Barto (18) and Kosorok and Laber (19): **model-based** (Yang and Wang, 19; Li et al., 24a); **Q-learning** (Clifton and Laber, 20), policy eval. (Shi et al., 22a) and **policy opt** (Clifton and Laber, 20; Li et al., 24b)

- **Backward inductive Q-learning**: Murphy (2003, 2005) using **linear** (Chakraborty and Murphy, 2014; Laber et al., 2014; Song et al., 2015) and **nonlinear** models (Laber, Linn and Stefanski, 2014; Zhang et al., 2018).

- **Online RL**: Fan, et al (20); Li et al (21, 24a, b), Jin et al. (21, 23). Liao et al. (22) Shi et al., (2022a, b); Yan et al. (2023); Xia et al. (24); and Cai et al. (24)

- **Neural networks**: LeCun et al. (15), Telgarsky, (16); Yarotsky (17); Bauer and Kohler (19); Schmidt-Hieber (20); Farrell et al. (21); Kohler and Langer (21); Zhong, et al. (2022), Fan and Gu (24); Fan et al. (24); Bhattacharya et al. (24).

Related Work: RL & NN

- **Offline RL**: Sutton and Barto (18) and Kosorok and Laber (19): **model-based** (Yang and Wang, 19; Li et al., 24a); **Q-learning** (Clifton and Laber, 20), policy eval. (Shi et al., 22a) and **policy opt** (Clifton and Laber, 20; Li et al., 24b)

- **Backward inductive Q-learning**: Murphy (2003, 2005) using **linear** (Chakraborty and Murphy, 2014; Laber et al., 2014; Song et al., 2015) and **nonlinear** models (Laber, Linn and Stefanski, 2014; Zhang et al., 2018).

- **Online RL**: Fan, et al (20); Li et al (21, 24a, b), Jin et al. (21, 23). Liao et al. (22) Shi et al., (2022a, b); Yan et al. (2023); Xia et al. (24); and Cai et al. (24)

- **Neural networks**: LeCun et al. (15), Telgarsky, (16); Yarotsky (17); Bauer and Kohler (19); Schmidt-Hieber (20); Farrell et al. (21); Kohler and Langer (21); Zhong, et al. (2022), Fan and Gu (24); Fan et al. (24); Bhattacharya et al. (24).

Related Work: RL & NN

- **Offline RL**: Sutton and Barto (18) and Kosorok and Laber (19): **model-based** (Yang and Wang, 19; Li et al., 24a); **Q-learning** (Clifton and Laber, 20), policy eval. (Shi et al., 22a) and **policy opt** (Clifton and Laber, 20; Li et al., 24b)

- **Backward inductive Q-learning**: Murphy (2003, 2005) using **linear** (Chakraborty and Murphy, 2014; Laber et al., 2014; Song et al., 2015) and **nonlinear** models (Laber, Linn and Stefanski, 2014; Zhang et al., 2018).

- **Online RL**: Fan, et al (20); Li et al (21, 24a, b), Jin et al. (21, 23). Liao et al. (22) Shi et al., (2022a, b); Yan et al. (2023); Xia et al. (24); and Cai et al. (24)

- **Neural networks**: LeCun et al. (15), Telgarsky, (16); Yarotsky (17); Bauer and Kohler (19); Schmidt-Hieber (20); Farrell et al. (21); Kohler and Langer (21); Zhong, et al. (2022), Fan and Gu (24); Fan et al. (24); Bhattacharya et al. (24).

Related Work: RL & NN

- **Offline RL**: Sutton and Barto (18) and Kosorok and Laber (19): **model-based** (Yang and Wang, 19; Li et al., 24a); **Q-learning** (Clifton and Laber, 20), policy eval. (Shi et al., 22a) and **policy opt** (Clifton and Laber, 20; Li et al., 24b)

- **Backward inductive Q-learning**: Murphy (2003, 2005) using **linear** (Chakraborty and Murphy, 2014; Laber et al., 2014; Song et al., 2015) and **nonlinear** models (Laber, Linn and Stefanski, 2014; Zhang et al., 2018).

- **Online RL**: Fan, et al (20); Li et al (21, 24a, b), Jin et al. (21, 23). Liao et al. (22) Shi et al., (2022a, b); Yan et al. (2023); Xia et al. (24); and Cai et al. (24)

- **Neural networks**: LeCun et al. (15), Telgarsky, (16); Yarotsky (17); Bauer and Kohler (19); Schmidt-Hieber (20); Farrell et al. (21); Kohler and Langer (21); Zhong, et al. (2022), Fan and Gu (24); Fan et al. (24); Bhattacharya et al. (24).

Transfer Learning

- ★ **Domain & Model shift**: Ganin et al (2015), Zhuang et al (2020), Ma, Pathak and Wainwright (2023); Ge. et al (2024); Tang et al (2025); Wang (2023), Maity, Sun and Banerjee (2022).

- ★ **High-dimensional regression**: Pan and Yang (2009), Bengio et al (2010), Kweiss et al (2016), Kpotufe and Martinet (2021), Cai and Wei (2021); Li, Cai and Li (2022a; b),Tiang and Feng (2022), Gu, Han and Duan (2022), Li et al (2023), Cai and Pu (2022); Fan et al (2025).

- ★ **Transfer learning in RL**. Zhu et al. (2023), Agarwal et al. (2023); Ishfaq et al. (2024); Bose, Du and Fazel, (2024); Lu, Huang and Du (2021); Cheng et al. (2022), Chen, Chen and Jing, 2024), Chen, Li and Jordan (2022); Cai et al (2024)

Transfer Learning

- ★ **Domain & Model shift**: Ganin et al (2015), Zhuang et al (2020), Ma, Pathak and Wainwright (2023); Ge. et al (2024); Tang et al (2025); Wang (2023), Maity, Sun and Banerjee (2022).

- ★ **High-dimensional regression**: Pan and Yang (2009), Bengio et al (2010), Kweiss et al (2016), Kpotufe and Martinet (2021), Cai and Wei (2021); Li, Cai and Li (2022a; b),Tiang and Feng (2022), Gu, Han and Duan (2022), Li et al (2023), Cai and Pu (2022); Fan et al (2025).

- ★ **Transfer learning in RL**. Zhu et al. (2023), Agarwal et al. (2023); Ishfaq et al. (2024); Bose, Du and Fazel, (2024); Lu, Huang and Du (2021); Cheng et al. (2022), Chen, Chen and Jing, 2024), Chen, Li and Jordan (2022); Cai et al (2024)

Transfer Learning

- ★ **Domain & Model shift**: Ganin et al (2015), Zhuang et al (2020), Ma, Pathak and Wainwright (2023); Ge. et al (2024); Tang et al (2025); Wang (2023), Maity, Sun and Banerjee (2022).

- ★ **High-dimensional regression**: Pan and Yang (2009), Bengio et al (2010), Kweiss et al (2016), Kpotufe and Martinet (2021), Cai and Wei (2021); Li, Cai and Li (2022a; b),Tiang and Feng (2022), Gu, Han and Duan (2022), Li et al (2023), Cai and Pu (2022); Fan et al (2025).

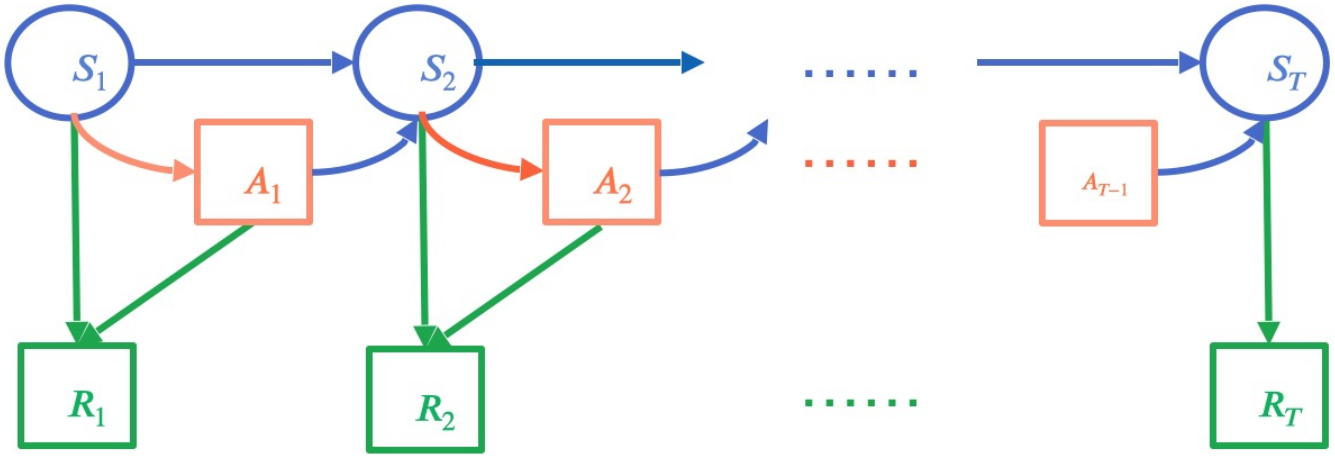
- ★ **Transfer learning in RL**. Zhu et al. (2023), Agarwal et al. (2023); Ishfaq et al. (2024); Bose, Du and Fazel, (2024); Lu, Huang and Du (2021); Cheng et al. (2022), Chen, Chen and Jing, 2024), Chen, Li and Jordan (2022); Cai et al (2024)

Problem Setup

Finite-horizon MDP

$\mathcal{M} = \{\mathcal{S}, \mathcal{A}, \mathbb{P}, r, \gamma, T\}$:

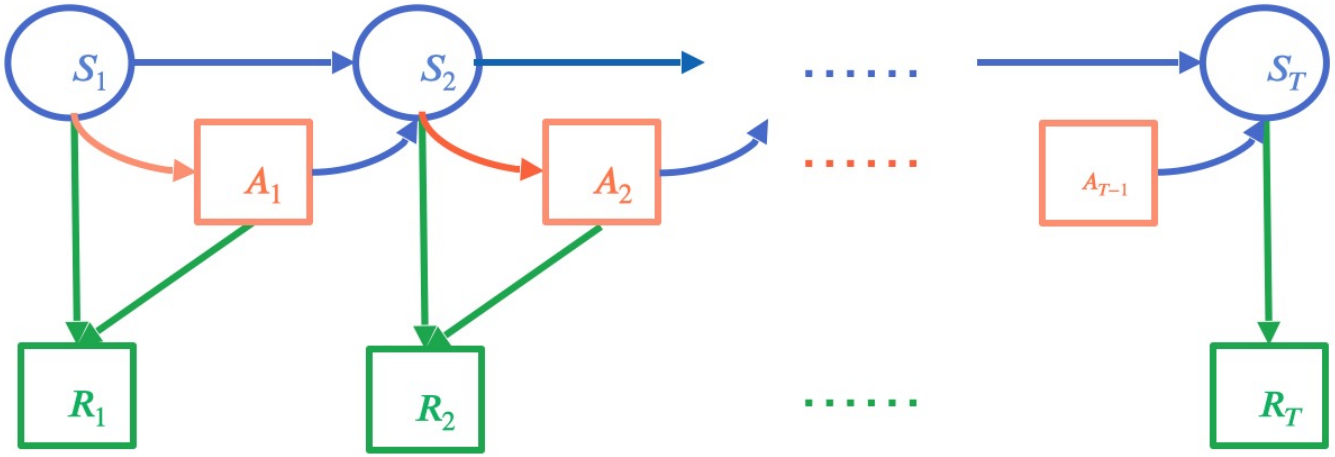
- State variable: $\mathbf{s}_t$.
- Finite action space: $a_t \in [M]$.
- $\mathbb{P}$ is the transition kernel **(time-varying)**; r is the reward.
- Goal: Estimate the optimal value function $Q_t^*(a, \mathbf{s})$ and policy $\pi_t^*(a_t|\mathbf{s}_t)$ for each stage $t \in [T]$.



Finite-horizon MDP

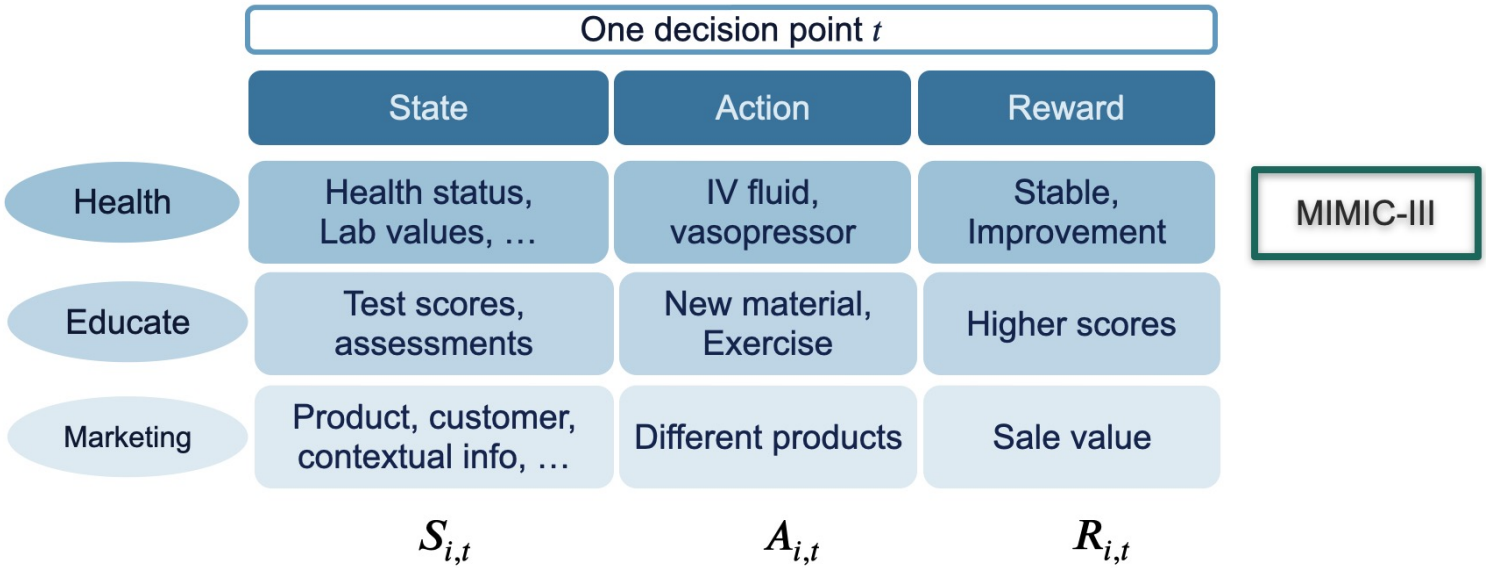
$\mathcal{M} = \{\mathcal{S}, \mathcal{A}, \mathbb{P}, r, \gamma, T\}$:

- State variable: $\mathbf{s}_t$.
- Finite action space: $a_t \in [M]$.
- $\mathbb{P}$ is the transition kernel **(time-varying)**; r is the reward.
- **Goal**: Estimate the optimal value function $Q_t^*(a, \mathbf{s})$ and policy $\pi_t^*(a_t|\mathbf{s}_t)$ for each stage $t \in [T]$.



Example

:

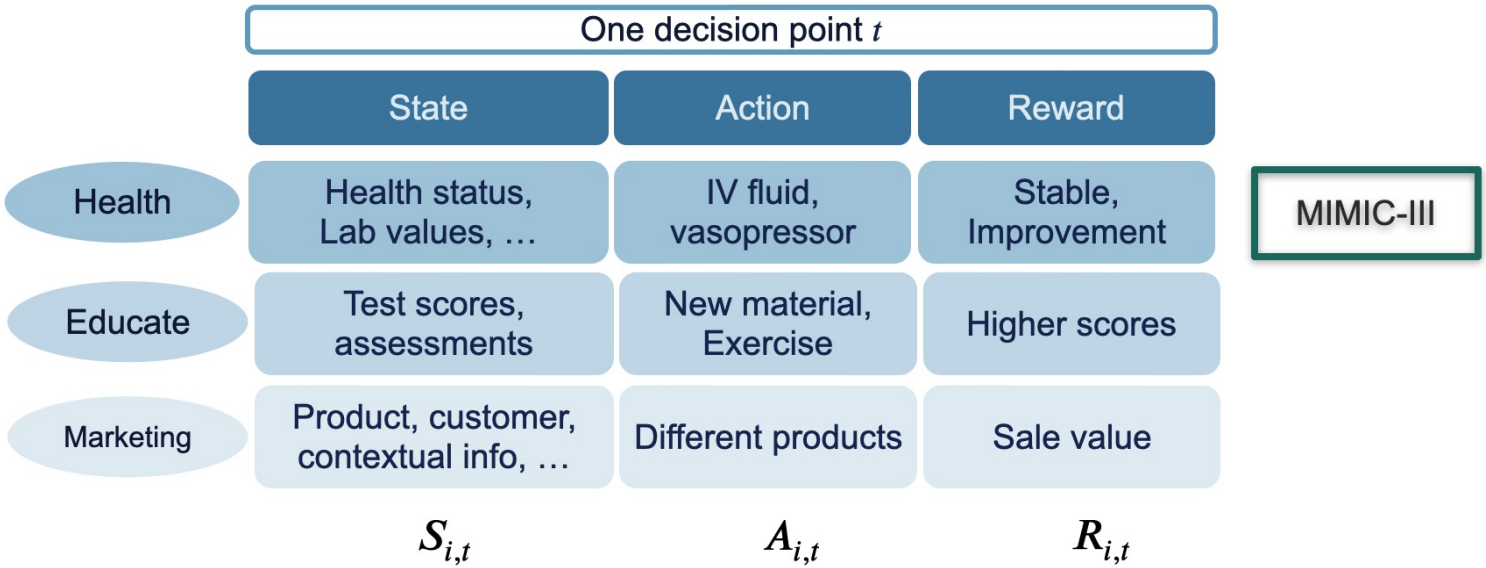


Source & Target: longitudinal data, n_k trajectories.

- Target RL task $\mathcal{M}^{(0)} = \{\mathcal{S}, \mathcal{A}, \mathbb{P}^{(0)}, r^{(0)}, \gamma, T\}$
- Source RL tasks $\mathcal{M}^{(k)} = \{\mathcal{S}, \mathcal{A}, \mathbb{P}^{(k)}, r^{(k)}, \gamma, T\}$, $k \in [M]$
- Data: $\{\mathbf{s}_{t,i}^{(k)}, a_{t,i}^{(k)}, r_{t,i}^{(k)}\}_{t \in [T], i \in [n_k]}$ for each k.

Example

:



Source & Target: longitudinal data, n_k trajectories.

- Target RL task $\mathcal{M}^{(0)} = \{\mathcal{S}, \mathcal{A}, \mathbb{P}^{(0)}, r^{(0)}, \gamma, T\}$
- Source RL tasks $\mathcal{M}^{(k)} = \{\mathcal{S}, \mathcal{A}, \mathbb{P}^{(k)}, r^{(k)}, \gamma, T\}$, $k \in [M]$
- **Data**: $\{\mathbf{s}_{t,i}^{(k)}, a_{t,i}^{(k)}, r_{t,i}^{(k)}\}_{t\in[T], i\in[n_k]}$ for each k.

Action-value function and Bellman Equation

Expected Reward: $r_t(\mathbf{s}_t, a_t) = \mathbb{E}[r_t|\mathbf{s}_t, a_t]$.

Action-value: $Q_t^{\pi}(\mathbf{s}, \mathbf{a}) = \mathbb{E}^{\pi}\left[\sum_{s=t}^{T} \gamma^{s-t} r(\mathbf{s}_s, a_s) \middle| \mathbf{s}_t = \mathbf{s}, \mathbf{a_t} = \mathbf{a}\right]$.

Optimal action-value: $Q_t^*(\mathbf{s}, a) = \sup_{\pi \in \Pi} Q_t^{\pi}(\mathbf{s}, a)$.

Bellman equation:

$$Q_t^*(\mathbf{s}_t, a_t) = \mathbb{E}\left\{ \underbrace{\mathbf{r_t} + \gamma \max_{\mathbf{a}' \in \mathcal{A}} \mathbf{Q}_{\mathbf{t+1}}^* \left(\mathbf{s_{t+1}}, \mathbf{a}'\right)}_{\mathbf{y_t}} \middle| \mathbf{s}_t, a_t \right\}.$$

Action-value function and Bellman Equation

Expected Reward: $r_t(\mathbf{s}_t, a_t) = \mathbb{E}[r_t|\mathbf{s}_t, a_t]$.

Action-value: $Q_t^{\pi}(\mathbf{s}, \mathbf{a}) = \mathbb{E}^{\pi}\left[\sum_{s=t}^{T} \gamma^{s-t} r(\mathbf{s}_s, a_s) \middle| \mathbf{s}_t = \mathbf{s}, \mathbf{a_t} = \mathbf{a}\right]$.

Optimal action-value: $Q_t^*(\mathbf{s}, a) = \sup_{\pi \in \Pi} Q_t^{\pi}(\mathbf{s}, a)$.

Bellman equation:

$$Q_t^*(\mathbf{s}_t, a_t) = \mathbb{E}\left\{ \underbrace{\mathbf{r_t} + \gamma \max_{\mathbf{a}' \in \mathcal{A}} \mathbf{Q}_{\mathbf{t+1}}^*\left(\mathbf{s_{t+1}}, \mathbf{a}'\right)}_{\mathbf{y_t}} \middle| \mathbf{s}_t, a_t \right\}.$$

Action-value function and Bellman Equation

Expected Reward: $r_t(\mathbf{s}_t, a_t) = \mathbb{E}[r_t|\mathbf{s}_t, a_t]$.

Action-value: $Q_t^{\pi}(\mathbf{s}, \mathbf{a}) = \mathbb{E}^{\pi}\left[\sum_{s=t}^{T}\gamma^{s-t} r(\mathbf{s}_s, a_s)\middle|\mathbf{s}_t = \mathbf{s}, \mathbf{a_t} = \mathbf{a}\right]$.

Optimal action-value: $Q_t^*(\mathbf{s}, a) = \sup_{\pi\in\Pi} Q_t^{\pi}(\mathbf{s}, a)$.

Bellman equation:

$$Q_t^*(\mathbf{s}_t, a_t) = \mathbb{E}\left\{\underbrace{\mathbf{r_t} + \gamma \max_{\mathbf{a}'\in\mathcal{A}} \mathbf{Q}_{\mathbf{t+1}}^*\left(\mathbf{s_{t+1}}, \mathbf{a}'\right)}_{\mathbf{y_t}}\middle|\mathbf{s}_t, a_t\right\}.$$

Action-value function and Bellman Equation

Expected Reward: $r_t(\mathbf{s}_t, a_t) = \mathbb{E}[r_t|\mathbf{s}_t, a_t]$.

Action-value: $Q_t^{\pi}(\mathbf{s}, \mathbf{a}) = \mathbb{E}^{\pi}\left[\sum_{s=t}^{T}\gamma^{s-t} r(\mathbf{s}_s, a_s)\middle|\mathbf{s}_t = \mathbf{s}, \mathbf{a_t} = \mathbf{a}\right]$.

Optimal action-value: $Q_t^*(\mathbf{s}, a) = \sup_{\pi\in\Pi} Q_t^{\pi}(\mathbf{s}, a)$.

Bellman equation:

$$Q_t^*(\mathbf{s}_t, a_t) = \mathbb{E}\left\{\underbrace{\mathbf{r_t} + \gamma \max_{\mathbf{a}'\in\mathcal{A}} \mathbf{Q}^*_{\mathbf{t+1}}\left(\mathbf{s_{t+1}}, \mathbf{a}'\right)}_{\mathbf{y_t}}\middle|\mathbf{s}_t, a_t\right\}.$$

Action-value function and Bellman Equation

Expected Reward: $r_t(\mathbf{s}_t, a_t) = \mathbb{E}[r_t|\mathbf{s}_t, a_t]$.

Action-value: $Q_t^{\pi}(\mathbf{s}, \mathbf{a}) = \mathbb{E}^{\pi}\left[\sum_{s=t}^{T}\gamma^{s-t} r(\mathbf{s}_s, a_s)\middle|\mathbf{s}_t = \mathbf{s}, \mathbf{a_t} = \mathbf{a}\right]$.

Optimal action-value: $Q_t^*(\mathbf{s}, a) = \sup_{\pi\in\Pi} Q_t^{\pi}(\mathbf{s}, a)$.

Bellman equation:

$$Q_t^*(\mathbf{s}_t, a_t) = \mathbb{E}\left\{\underbrace{\mathbf{r_t} + \gamma \max_{\mathbf{a}'\in\mathcal{A}} \mathbf{Q}^*_{\mathbf{t+1}}\left(\mathbf{s_{t+1}}, \mathbf{a}'\right)}_{\mathbf{y_t}}\middle|\mathbf{s}_t, a_t\right\}.$$

Action-value function and Bellman Equation

Expected Reward: $r_t(\mathbf{s}_t, a_t) = \mathbb{E}[r_t|\mathbf{s}_t, a_t]$.

Action-value: $Q_t^{\pi}(\mathbf{s}, \mathbf{a}) = \mathbb{E}^{\pi}\left[\sum_{s=t}^{T}\gamma^{s-t}r(\mathbf{s}_s, a_s)\middle|\mathbf{s}_t = \mathbf{s}, \mathbf{a_t} = \mathbf{a}\right]$.

Optimal action-value: $Q_t^*(\mathbf{s}, a) = \sup_{\pi\in\Pi} Q_t^{\pi}(\mathbf{s}, a)$.

Bellman equation: $\Longrightarrow$ **Backward induction (Murphy, 03, 05)**

$$Q_t^*(\mathbf{s}_t, a_t) = \mathbb{E}\left\{\underbrace{\mathbf{r_t} + \gamma \max_{\mathbf{a}'\in\mathcal{A}} \mathbf{Q}_{\mathbf{t+1}}^*\left(\mathbf{s_{t+1}}, \mathbf{a}'\right)}_{\mathbf{y_t}}\middle|\mathbf{s}_t, a_t\right\}.$$

★create psudo-sample at time $t+1$ and run regression to learn Q_t^* $\quad Q_{T+1}^* = 0$

Aggregation and Transferrability

★ Define difference of rewards for the target and source task k by

$$\delta_t^{(k)}(\mathbf{s}, a) = r_t^{(0)}(\mathbf{s}, a) - r_t^{(k)}(\mathbf{s}, a)$$

★ Define aggregated reward $r_t^{*\text{agg}} = \sum_{k=1}^{K} \bar{\upsilon}_t^{(k)} r_t^{(k)}$, where $\bar{\upsilon}_t^{(k)}(\mathbf{s}, a) = \mathbb{P}[k_i = k | \mathbf{s}_t = \mathbf{s}, a_t = a]$, and the agg. Q^* function as

$$Q_t^{*\text{agg}} = \sum_{k=1}^{K} \bar{\upsilon}_t^{(k)} \mathbb{E}^{(k)} \left[Y_{t,i}^{(rwt-k)} \mid S_{t,i}^{(k)} = \mathbf{s}, A_{t,i}^{(k)} = a \right].$$
$$= r_t^{*\text{agg}} + \gamma \mathbb{E}\left\{ \max_{a' \in \mathcal{A}} Q_{t+1}^{(0)}(\mathbf{s}_{t+1}, a') \,\middle|\, \mathbf{s}_t, a_t \right\}.$$

★ Agg. reward difference: $\delta_t^{*\text{agg}}(\mathbf{s}, a) = r_t^{*\text{agg}}(\mathbf{s}, a) - r_t^{(0)}(\mathbf{s}, a)$.

Aggregation and Transferrability

★ Define difference of rewards for the target and source task k by

$$\delta_t^{(k)}(\mathbf{s}, a) = r_t^{(0)}(\mathbf{s}, a) - r_t^{(k)}(\mathbf{s}, a)$$

★ Define **aggregated reward** $r_t^{*\text{agg}} = \sum_{k=1}^{K} \bar{\upsilon}_t^{(k)} r_t^{(k)}$, where $\bar{\upsilon}_t^{(k)}(\mathbf{s}, a) = \mathbb{P}[k_i = k | \mathbf{s}_t = \mathbf{s}, a_t = a]$, and the **agg. Q^* function** as

$$\begin{aligned} Q_t^{*\text{agg}} &= \sum_{k=1}^{K} \bar{\upsilon}_t^{(k)} \mathbb{E}^{(k)} \left[Y_{t,i}^{(rwt-k)} \mid S_{t,i}^{(k)} = \mathbf{s}, A_{t,i}^{(k)} = a \right]. \\ &= r_t^{*\text{agg}} + \gamma \mathbb{E}\left\{ \max_{a' \in \mathcal{A}} Q_{t+1}^{(0)}(\mathbf{s}_{t+1}, a') \,\middle|\, \mathbf{s}_t, a_t \right\}. \end{aligned}$$

★ **Agg. reward difference**: $\delta_t^{*\text{agg}}(\mathbf{s}, a) = r_t^{*\text{agg}}(\mathbf{s}, a) - r_t^{(0)}(\mathbf{s}, a)$.

Aggregation and Transferrability

★ Define difference of rewards for the target and source task k by

$$\delta_t^{(k)}(\mathbf{s}, a) = r_t^{(0)}(\mathbf{s}, a) - r_t^{(k)}(\mathbf{s}, a)$$

★ Define **aggregated reward** $r_t^{*\text{agg}} = \sum_{k=1}^{K} \bar{\upsilon}_t^{(k)} r_t^{(k)}$, where $\bar{\upsilon}_t^{(k)}(\mathbf{s}, a) = \mathbb{P}[k_i = k | \mathbf{s}_t = \mathbf{s}, a_t = a]$, and the **agg. Q^* function** as

$$\begin{aligned} Q_t^{*\text{agg}} &= \sum_{k=1}^{K} \bar{\upsilon}_t^{(k)} \mathbb{E}^{(k)} \left[Y_{t,i}^{(rwt-k)} \mid S_{t,i}^{(k)} = \mathbf{s}, A_{t,i}^{(k)} = a \right]. \\ &= r_t^{*\text{agg}} + \gamma \, \mathbb{E}\left\{ \max_{a' \in \mathcal{A}} Q_{t+1}^{(0)}(\mathbf{s}_{t+1}, a') \,\middle|\, \mathbf{s}_t, a_t \right\}. \end{aligned}$$

★ **Agg. reward difference**: $\delta_t^{*\text{agg}}(\mathbf{s}, a) = r_t^{*\text{agg}}(\mathbf{s}, a) - r_t^{(0)}(\mathbf{s}, a)$.

Similarity of MDPs

★ **Smoother** difference function: (Kohler and Krzyzak, 17)

$$Q_t^{*\,\mathrm{agg}} \in \mathrm{HCM}(\gamma_1), \qquad \delta_t^{*\,\mathrm{agg}} \in \mathrm{HCM}(\gamma_2) \qquad \text{and} \qquad \gamma_2 > \gamma_1.$$

★ **Smoother** Transition Ratio: $\omega_t^{(k)} = p_t^{(0)} / p_t^{(k)}$. (defined later)

Hierachical Composition Model: $\mathrm{HCM}(\mathcal{P})$ = Finite compositions

$Q^* = Q_1 \circ \cdots \circ Q_q$ of t-variate function with smoothness β, for $(t, \beta) \in \mathcal{P}$.

★ e.g. $Q^* = f_1(x_1) + \cdots + f_p(x_p)$ $\qquad$ $Q^* = f_1(x_1, f_2(x_2, x_3)) + f_3(x_2, x_5, x_9)$

★ Complexity determined by $\gamma^* = \min_{(t,\beta)\in\mathcal{P}} \frac{\beta}{t}$, $\qquad$ denoted by $\mathcal{H}(\gamma^*)$

★ **Adaptively** learned by neural networks

Similarity of MDPs

★ **Smoother** difference function: (Kohler and Krzyzak, 17)

$$Q_t^{*\,\mathrm{agg}} \in \mathrm{HCM}(\gamma_1), \qquad \delta_t^{*\,\mathrm{agg}} \in \mathrm{HCM}(\gamma_2) \qquad \text{and} \qquad \gamma_2 > \gamma_1.$$

★ **Smoother** Transition Ratio: $\omega_t^{(k)} = p_t^{(0)} / p_t^{(k)}$. (defined later)

Hierachical Composition Model: $\mathrm{HCM}(\mathcal{P}) =$ Finite compositions

$Q^* = Q_1 \circ \cdots \circ Q_q$ of t-variate function with smoothness β, for $(t, \beta) \in \mathcal{P}$.

★ e.g. $Q^* = f_1(x_1) + \cdots + f_p(x_p)$ $\qquad$ $Q^* = f_1(x_1, f_2(x_2, x_3)) + f_3(x_2, x_5, x_9)$

★ Complexity determined by $\gamma^* = \min_{(t,\beta)\in\mathcal{P}} \frac{\beta}{t}$, $\qquad$ denoted by $\mathcal{H}(\gamma^*)$

★ **Adaptively** learned by neural networks

Similarity of MDPs

★ **Smoother** difference function: (Kohler and Krzyzak, 17)

$$Q_t^{*\,\mathrm{agg}} \in \mathrm{HCM}(\gamma_1), \qquad \delta_t^{*\,\mathrm{agg}} \in \mathrm{HCM}(\gamma_2) \qquad \text{and} \qquad \gamma_2 > \gamma_1.$$

★ **Smoother** Transition Ratio: $\omega_t^{(k)} = p_t^{(0)} / p_t^{(k)}$. (defined later)

Hierachical Composition Model: $\mathrm{HCM}(\mathcal{P}) =$ Finite compositions

$Q^* = Q_1 \circ \cdots \circ Q_q$ of t-variate function with smoothness β, for $(t, \beta) \in \mathcal{P}$.

★ e.g. $Q^* = f_1(x_1) + \cdots + f_p(x_p)$ $\qquad$ $Q^* = f_1(x_1, f_2(x_2, x_3)) + f_3(x_2, x_5, x_9)$

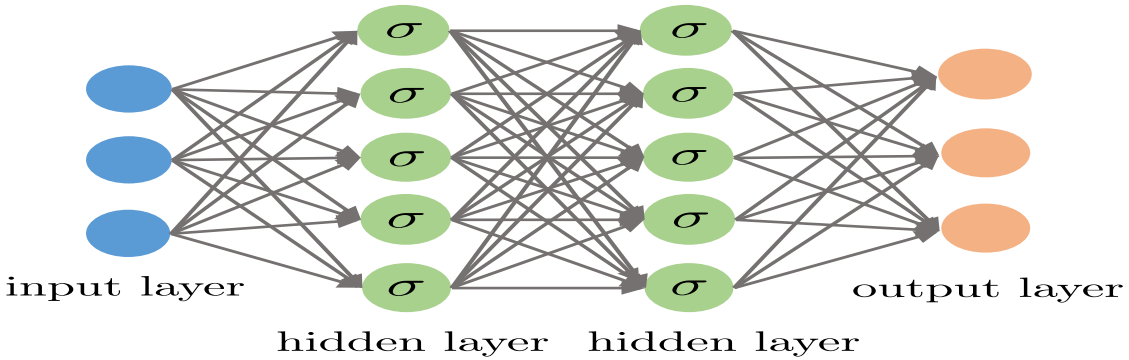
★ Complexity determined by $\gamma^* = \min_{(t,\beta)\in\mathcal{P}} \frac{\beta}{t}$, $\qquad$ denoted by $\mathcal{H}(\gamma^*)$

★ **Adaptively** learned by neural networks

Deep ReLU Neural Network

Deep ReLU neural network with width N and depth L

$$f : \mathbb{R}^d \to \mathbb{R}, \qquad f(\mathbf{x};\theta) = \mathcal{L}_{L+1} \circ \sigma \circ \mathcal{L}_L \circ \cdots \sigma \circ \mathcal{L}_2 \circ \sigma \circ \mathcal{L}_1(\mathbf{x})$$



- $\mathcal{L}_i(\mathbf{x}) = \mathbf{W}_i\mathbf{x} + \mathbf{b}_i$ with weights $\mathbf{W}_i, \mathbf{b}_i$, $\sigma(t) = t_+$ **ReLU** activation
- Dim: $(d_0, d_1, \cdots, d_L, d_{L+1}) = (d, N, \cdots, N, 1)$
- **Parameter** $\theta = \{\mathbf{W}_i, \mathbf{b}_i\}_{i=1}^L$ **# parameter** $\asymp N^2 L$

Bounded ReLU-DNN functions : $\mathcal{G}_n = \{\mathbf{T_M g(x)} : \sup_{\ell \in [L+1]} \|\mathbf{W}_\ell\|_{\max} \vee \|\mathbf{b}_\ell\|_\infty \le B\}$

Methodology

Pseudo Responses and Transferrable Data

★ **Pseudo Response**: $y_{t,i} := r_{t,i} + \gamma \max_{a' \in \mathcal{A}} Q^*_{t+1}(\mathbf{s}_{t+1,i}, a')$. Then

$$Q^*_t(\mathbf{s}, a) = \mathbb{E}[y_{t,i} \mid \mathbf{s}_{t,i} = \mathbf{s}, a_{t,i} = a], \qquad \textbf{(Bellman equation)}$$

★ **Target Data**: $\left\{(\mathbf{s}^{(0)}_{t,i}, y^{(0)}_{t,i})\right\}_{i=1}^{n_0}$. **Source**: $\left\{(\mathbf{s}^{(k)}_{t,j}, y^{(k)}_{t,j})\right\}_{i=1}^{n_k}$

★ Naively pooling RL samples does not work!

—Different Q^* and transition densities

Pseudo Responses and Transferrable Data

★ **Pseudo Response**: $y_{t,i} := r_{t,i} + \gamma \max_{a' \in \mathcal{A}} Q^*_{t+1}(\mathbf{s}_{t+1,i}, a')$. Then

$$Q^*_t(\mathbf{s}, a) = \mathbb{E}[y_{t,i} \mid \mathbf{s}_{t,i} = \mathbf{s}, a_{t,i} = a], \qquad \textbf{(Bellman equation)}$$

★ **Target Data**: $\left\{(\mathbf{s}^{(0)}_{t,i}, y^{(0)}_{t,i})\right\}_{i=1}^{n_0}$. **Source**: $\left\{(\mathbf{s}^{(k)}_{t,j}, y^{(k)}_{t,j})\right\}_{i=1}^{n_k}$

★ Naively pooling RL samples does not work!

—Different Q^* and transition densities

Pseudo Responses and Transferrable Data

★ **Pseudo Response**: $y_{t,i} := r_{t,i} + \gamma \max_{a' \in \mathcal{A}} Q^*_{t+1}(\mathbf{s}_{t+1,i}, a')$. Then

$$Q^*_t(\mathbf{s}, a) = \mathbb{E}[y_{t,i} \mid \mathbf{s}_{t,i} = \mathbf{s}, a_{t,i} = a], \qquad \textbf{(Bellman equation)}$$

★ **Target Data**: $\left\{(\mathbf{s}^{(0)}_{t,i}, y^{(0)}_{t,i})\right\}_{i=1}^{n_0}$. **Source**: $\left\{(\mathbf{s}^{(k)}_{t,j}, y^{(k)}_{t,j})\right\}_{i=1}^{n_k}$

★ Naively pooling RL samples does not work!

—Different Q^* and transition densities

Reweighting and Retargeting

Reweighted & retargeted response: W/ density ratio $\omega_t^{(k)} = p_t^{(0)} / p_t^{(k)}$,

$$y_{t,i}^{(rwt-k)} := r_{t,i}^{(k)} + \gamma \cdot \omega_{t,i}^{(k)} \cdot \max_{a \in \mathcal{A}} Q_{t+1}^{*(0)}(\mathbf{s}_{t+1,i}^{(k)}, a).$$

Transferable RL Samples:

$$\mathbb{E}_{\mathbb{P}^{(0)}}\left[y_{t,i}^{(0)} \mid \mathbf{s}_{t,i}^{(0)} = \mathbf{s}, a_{t,i}^{(0)} = \mathbf{a}\right] - \mathbb{E}_{\mathbb{P}^{(k)}}\left[y_{t,i}^{(rwt-k)} \mid \mathbf{s}_{t,i}^{(k)} = \mathbf{s}, a_{t,i}^{(k)} = \mathbf{a}\right]$$

Reweighting and Retargeting

Reweighted & retargeted response: W/ density ratio $\omega_t^{(k)} = p_t^{(0)} / p_t^{(k)}$,

$$y_{t,i}^{(rwt-k)} := r_{t,i}^{(k)} + \gamma \cdot \omega_{t,i}^{(k)} \cdot \max_{a \in \mathcal{A}} Q_{t+1}^{*(0)}(\mathbf{s}_{t+1,i}^{(k)}, a).$$

Transferable RL Samples:

$$\mathbb{E}_{\mathbb{P}^{(0)}}\left[y_{t,i}^{(0)} \mid \mathbf{s}_{t,i}^{(0)} = \mathbf{s}, a_{t,i}^{(0)} = \mathbf{a}\right] - \mathbb{E}_{\mathbb{P}^{(k)}}\left[y_{t,i}^{(rwt-k)} \mid \mathbf{s}_{t,i}^{(k)} = \mathbf{s}, a_{t,i}^{(k)} = \mathbf{a}\right]$$

Reweighting and Retargeting

Reweighted & retargeted response: W/ density ratio $\omega_t^{(k)} = p_t^{(0)} / p_t^{(k)}$,

$$y_{t,i}^{(rwt-k)} := r_{t,i}^{(k)} + \gamma \cdot \omega_{t,i}^{(k)} \cdot \max_{a \in \mathcal{A}} Q_{t+1}^{*(0)}(\mathbf{s}_{t+1,i}^{(k)}, a).$$

Transferable RL Samples: $r_t^{(0)}(\mathbf{s}, a) - r_t^{(k)}(\mathbf{s}, a) =$

$$\mathbb{E}_{\mathbb{P}^{(0)}}\left[y_{t,i}^{(0)} \mid \mathbf{s}_{t,i}^{(0)} = \mathbf{s}, a_{t,i}^{(0)} = \mathbf{a}\right] - \mathbb{E}_{\mathbb{P}^{(k)}}\left[y_{t,i}^{(rwt-k)} \mid \mathbf{s}_{t,i}^{(k)} = \mathbf{s}, a_{t,i}^{(k)} = \mathbf{a}\right]$$

Reweighting and Retargeting

Reweighted & retargeted response: W/ density ratio $\omega_t^{(k)} = p_t^{(0)} / p_t^{(k)}$,

$$y_{t,i}^{(rwt-k)} := r_{t,i}^{(k)} + \gamma \cdot \omega_{t,i}^{(k)} \cdot \max_{a \in \mathcal{A}} Q_{t+1}^{*(0)}(\mathbf{s}_{t+1,i}^{(k)}, a).$$

Transferable RL Samples: $\delta_t^{(k)}(\mathbf{s}, a) = \quad r_t^{(0)}(\mathbf{s}, a) - r_t^{(k)}(\mathbf{s}, a) =$

$$\mathbb{E}_{\mathbb{P}^{(0)}}\left[y_{t,i}^{(0)} \mid \mathbf{s}_{t,i}^{(0)} = \mathbf{s}, a_{t,i}^{(0)} = \mathbf{a}\right] - \mathbb{E}_{\mathbb{P}^{(k)}}\left[y_{t,i}^{(rwt-k)} \mid \mathbf{s}_{t,i}^{(k)} = \mathbf{s}, a_{t,i}^{(k)} = \mathbf{a}\right]$$

Reweighting and Retargeting

Reweighted & retargeted response: W/ density ratio $\omega_t^{(k)} = p_t^{(0)}/p_t^{(k)}$,

$$y_{t,i}^{(rwt-k)} := r_{t,i}^{(k)} + \gamma \cdot \omega_{t,i}^{(k)} \cdot \max_{a \in \mathcal{A}} Q_{t+1}^{*(0)}(\mathbf{s}_{t+1,i}^{(k)}, a).$$

Transferable RL Samples: $\delta_t^{(k)}(\mathbf{s}, a) = r_t^{(0)}(\mathbf{s}, a) - r_t^{(k)}(\mathbf{s}, a) =$

$$\mathbb{E}_{\mathbb{P}^{(0)}}\left[y_{t,i}^{(0)} \mid \mathbf{s}_{t,i}^{(0)} = \mathbf{s}, a_{t,i}^{(0)} = \mathbf{a}\right] - \mathbb{E}_{\mathbb{P}^{(k)}}\left[y_{t,i}^{(rwt-k)} \mid \mathbf{s}_{t,i}^{(k)} = \mathbf{s}, a_{t,i}^{(k)} = \mathbf{a}\right]$$

★Errors from task differences do not accumulate along steps.

★Pool data $\left\{\widehat{y}_{t,i}^{(rwt-k)}\right\}$ to get an estimator and correct biases.

Estimating Transition Densities

Population L_2 loss: For an estimator g of transition density, its MISE $=$

$$\frac{1}{2}\int\int\int \left(g(\mathbf{s},a,\mathbf{s}') - p_t^{(k)}(\mathbf{s}'|\mathbf{s},a)\right)^2 p_t^{(k)}(\mathbf{s},a)\,d\mathbf{s}\,da\,d\mathbf{s}'$$
$$= \frac{1}{2}\mathbb{E}_{\mathbb{P}^{(k)}}\int g(\mathbf{s},a,\mathbf{s}')^2\,d\mathbf{s}' - \mathbb{E}_{\mathbb{P}^{(k)}} g(\mathbf{s},a,\mathbf{s}') + \text{const}$$

M-estimator for transition density ($\mathbf{s}_i^\circ$ uniform in $\mathcal{S}$):

$$\widehat{\rho}_t^{(k)} := \underset{g\in\mathcal{G}(\bar{L},\bar{M},\bar{N},\bar{B})}{\arg\min} \frac{1}{2n_k}\sum_{i=1}^{n_k} g(\mathbf{s}_{t,i}^{(k)}, a_{t,i}^{(k)}, \mathbf{s}_i^\circ)^2 - \frac{1}{n_k}\sum_{i=1}^{n_k} g(\mathbf{s}_{t,i}^{(k)}, a_{t,i}^{(k)}, \mathbf{s}_{t,i}^{\prime(k)})$$

and the density ratio estimator is given by $\widehat{\omega}_t^{(k)} := \dfrac{\widehat{\rho}_t^{(0)}}{\max\{\widehat{\rho}_t^{(k)}, \Upsilon_1\}}$.

Estimating Transition Densities

Population L_2 loss: For an estimator g of transition density, its MISE $=$

$$\frac{1}{2}\int\int\int \left(g(\mathbf{s},a,\mathbf{s}') - p_t^{(k)}(\mathbf{s}'|\mathbf{s},a)\right)^2 p_t^{(k)}(\mathbf{s},a)\,d\mathbf{s}\,da\,d\mathbf{s}'$$
$$= \frac{1}{2}\mathbb{E}_{\mathbb{P}^{(k)}}\int g(\mathbf{s},a,\mathbf{s}')^2\,d\mathbf{s}' - \mathbb{E}_{\mathbb{P}^{(k)}} g(\mathbf{s},a,\mathbf{s}') + \text{const}$$

M-estimator for transition density ($\mathbf{s}_i^\circ$ uniform in $\mathcal{S}$):

$$\widehat{\rho}_t^{(k)} := \underset{g\in\mathcal{G}(\bar{L},\bar{M},\bar{N},\bar{B})}{\arg\min} \frac{1}{2n_k}\sum_{i=1}^{n_k} g(\mathbf{s}_{t,i}^{(k)}, a_{t,i}^{(k)}, \mathbf{s}_i^\circ)^2 - \frac{1}{n_k}\sum_{i=1}^{n_k} g(\mathbf{s}_{t,i}^{(k)}, a_{t,i}^{(k)}, \mathbf{s}_{t,i}^{\prime(k)})$$

and the density ratio estimator is given by $\widehat{\omega}_t^{(k)} := \dfrac{\widehat{\rho}_t^{(0)}}{\max\{\widehat{\rho}_t^{(k)}, \Upsilon_1\}}$.

Transition Density Estimation with Transfer

★When task and source tasks are close, density transfer is feasible:

$$\widehat{\omega}_t^{(k),\mathrm{tr}} = \arg \min_{g \in \mathcal{G}(\bar{L}_2, \bar{N}_2, \bar{M}_2, \bar{B}_2)} \frac{1}{2n_0} \sum_{i=1}^{n_0} (g \cdot \widehat{\rho}_t^{(k)})^2 (\mathbf{s}_{t,i}^{(0)}, a_{t,i}^{(0)}, \mathbf{s}_i^{\circ}) - 2(g \cdot \widehat{\rho}_t^{(k)})(\mathbf{s}_{t,i}^{(0)}, a_{t,i}^{(0)}, \mathbf{s}_{t,i}^{\prime(0)})$$

Transition density: Let $\rho_t^{(k)}(\mathbf{s}, a, \mathbf{s}') := p_t^{(k)}(\mathbf{s}'|\mathbf{s}, a)$. Assume that

(i) (Boundedness.) $\Upsilon_1 \leq |\rho_t^{(k)}| \leq \Upsilon_2$.

(ii) (Smoothness.) $\rho_t^{(k)} \in \mathcal{H}_3$ with $\gamma(\mathcal{H}_3) = \gamma_3$.

(iii) (Transferability.) $\rho_t^{(0)} / \rho_t^{(k)} \in \mathcal{H}_4$ with $\gamma(\mathcal{H}_4) = \gamma_4$, and $\boxed{\gamma_3 \leq \gamma_4}$.

Reweighted transfer Q-learnin

For $t = T, \cdots, 1$, compute $\widehat{y}_{t,i}^{(rwt-k)}$ and $\widehat{y}_{t,i}^{(0)}$, and run DNN:

$$\begin{aligned} \widehat{Q}_t^p &:= \arg\min_{g \in \mathcal{G}_1} \sum_{k \in [K]} \sum_{i \in [n_k]} \left(\widehat{y}_{t,i}^{(rwt-k)} - g(\mathbf{s}_{t,i}^{(k)}, a_{t,i}^{(k)}) \right)^2, \\ \widehat{\delta}_t &:= \arg\min_{g \in \mathcal{G}_2} \sum_{i \in [n_0]} \left(\widehat{y}_{t,i}^{(0)} - \widehat{Q}_t^p(\mathbf{s}_{t,i}^{(0)}, a_{t,i}^{(0)}) - g(\mathbf{s}_{t,i}^{(0)}, a_{t,i}^{(0)}) \right)^2, \end{aligned} \tag{1}$$

where $\mathcal{G}_1$ and $\mathcal{G}_2$ are function classes. Set $\widehat{Q}_t^{(0)} = \widehat{Q}_t^p + \widehat{\delta}_t$.

Theorectical Results

Assumptions

Positive Coverage

There exists a constant $\underline{c}$ such that for every t and almost surely for every $\mathbf{s}, a$,

$$\mathbb{P}_t^{\text{agg}}(A_t = a | S_t = \mathbf{s}) \geq \underline{c}.$$

Bounded Covariate Shift

The Radon-Nikodym derivative of $\mathbb{P}_t^{\text{agg}}$ and $\mathbb{P}_t^{(0)}$ satisfies

$$\eta \leq \frac{d\mathbb{P}_t^{\text{agg}}(\mathbf{s}, a)}{d\mathbb{P}_t^{(0)}(\mathbf{s}, a)} \leq \frac{1}{\eta}, \quad a.s.$$

Assumptions

Positive Coverage

There exists a constant $\underline{c}$ such that for every t and almost surely for every $\mathbf{s}, a$,

$$\mathbb{P}_t^{\text{agg}}(A_t = a | S_t = \mathbf{s}) \geq \underline{c}.$$

Bounded Covariate Shift

The Radon-Nikodym derivative of $\mathbb{P}_t^{\text{agg}}$ and $\mathbb{P}_t^{(0)}$ satisfies

$$\eta \leq \frac{d\mathbb{P}_t^{\text{agg}}(\mathbf{s}, a)}{d\mathbb{P}_t^{(0)}(\mathbf{s}, a)} \leq \frac{1}{\eta}, \quad a.s.$$

Main Result

Let $\widehat{Q}_t^{\mathrm{tr}}$ be the estimator with DNN approx. With probability $\geq 1 - 7Te^{-u}$,

$$\|\widehat{Q}_t^{\mathrm{tr}} - Q_t^*\|^2_{2,\mathbb{P}_t^{(0)}} \lesssim (T-t)\max\{\kappa,1\}^{T-t} \times \Bigg\{ \underbrace{\left(\frac{J\log n_0}{n_0}\right)^{\frac{2\gamma_2}{2\gamma_2+1}}}_{\text{est. err. of reward difference}} + \underbrace{\frac{1}{\eta}\left(\frac{J\log n_{\mathcal{M}}}{n_{\mathcal{M}}}\right)^{\frac{2\gamma_1}{2\gamma_1+1}}}_{\text{est. err. of reward aggregation}} + \underbrace{\frac{\gamma^2 T^2}{\eta}\max_{t\leq\tau\leq T}\widehat{\Omega}(\tau)}_{\text{est. err. of transition ratio}} + \frac{u}{\min\{n_0, n_{\mathcal{M}}\eta\}} \Bigg\},$$

where $J := |\mathcal{A}|$, $n_{\mathcal{M}} =$ No. of total tasks, $\widehat{\Omega}(t) = \frac{1}{n_{\mathcal{M}}}\sum_{i=1}^{n_{\mathcal{M}}} |\widehat{\omega}_{t,i}^{(k_i)} - \omega_{t,i}^{(k_i)}|^2$,

$\kappa := \left(\frac{\gamma^2}{c} + \frac{\gamma^2\Upsilon^2}{c\eta^2}\right)$, $\Upsilon = \Upsilon_2/\Upsilon_1$.

Remarks

1. **Policy learning**: Take the greedy policy of $\widehat{Q}_t^{\mathrm{tr}}(\mathbf{s}, a)$.
2. **Technical advantages**: ★allow temporal dependence instead of experience replay; ★remove the function class completeness ★no sample splitting; ★considering errors in transition density
3. **Benefit of transfer**: ★reward difference better estimated (lower complexity $\gamma_2 > \gamma_1$) ★errors in transition density negligible.
4. **Algorithmic learning with adaptive property**: DNN is fully algorithmic that is adaptive to the lower function structure:

Remarks

1. **Policy learning**: Take the greedy policy of $\widehat{Q}_t^{\mathrm{tr}}(\mathbf{s}, a)$.

2. **Technical advantages**: ★allow temporal dependence instead of experience replay; ★remove the function class completeness ★no sample splitting; ★considering errors in transition density

3. **Benefit of transfer**: ★reward difference better estimated (lower complexity $\gamma_2 > \gamma_1$) ★errors in transition density negligible.

4. **Algorithmic learning with adaptive property**: DNN is fully algorithmic that is adaptive to the lower function structure:

Remarks

1. **Policy learning**: Take the greedy policy of $\widehat{Q}_t^{\text{tr}}(\mathbf{s}, a)$.

2. **Technical advantages**: ★allow temporal dependence instead of experience replay; ★remove the function class completeness ★no sample splitting; ★considering errors in transition density

3. **Benefit of transfer**: ★reward difference better estimated (lower complexity $\gamma_2 > \gamma_1$) ★errors in transition density negligible.

4. **Algorithmic learning with adaptive property**: DNN is fully algorithmic that is adaptive to the lower function structure:

Remarks

1. **Policy learning**: Take the greedy policy of $\widehat{Q}_t^{\mathrm{tr}}(\mathbf{s}, a)$.

2. **Technical advantages**: ★allow temporal dependence instead of experience replay; ★remove the function class completeness ★no sample splitting; ★considering errors in transition density

3. **Benefit of transfer**: ★reward difference better estimated (lower complexity $\gamma_2 > \gamma_1$) ★errors in transition density negligible.

4. **Algorithmic learning with adaptive property**: DNN is fully algorithmic that is adaptive to the lower function structure:

Remarks

1. **Policy learning**: Take the greedy policy of $\widehat{Q}_t^{\mathrm{tr}}(\mathbf{s}, a)$.

2. **Technical advantages**: ★allow temporal dependence instead of experience replay; ★remove the function class completeness ★no sample splitting; ★considering errors in transition density

3. **Benefit of transfer**: ★reward difference better estimated (lower complexity $\gamma_2 > \gamma_1$) ★errors in transition density negligible.

4. **Algorithmic learning with adaptive property**: DNN is fully algorithmic that is adaptive to the lower function structure:
   ★For additive function $Q^* = f_1(x_1) + \cdots + f_p(x_p)$, we get one-d rate $n^{-2/5}$
   ★For comp. function $Q^* = f_1(x_1, f_2(x_2, x_3)) + f_4(x_2, x_9)$, we get two-d rate $n^{-1/3}$

Case I: Non-transferable transition densities

Let $\widehat{Q}_t^{(\mathrm{tr1})}$ = estimator w/ only reward transfers. Then, w/ prob. $> 1 - 2T(K+1)e^{-u}$,

$$\|\widehat{Q}_t^{(\mathrm{tr1})} - Q_t^*\|_{2,\mathbb{P}_t^{(0)}}^2 \lesssim \cdots \Bigg\{ \underbrace{\left(\frac{J\log n_0}{n_0}\right)^{\frac{2\gamma_2}{2\gamma_2+1}}}_{\text{est. err. of reward differences}} + \underbrace{\frac{1}{\eta}\left(\frac{J\log n_{\mathcal{M}}}{n_{\mathcal{M}}}\right)^{\frac{2\gamma_1}{2\gamma_1+1}}}_{\text{est. err. of reward aggregation}}$$

$$\underbrace{+\frac{\gamma^2 T^2}{\eta^2}\left(\frac{\log n_0}{n_0}\right)^{\frac{2\gamma_3}{2\gamma_3+1}} + \frac{\gamma^2 T^2}{\eta}\left(\frac{K\log(n_{\mathcal{M}})}{n_{\mathcal{M}}}\right)^{\frac{2\gamma_3}{2\gamma_3+1}}}_{\text{est. err. of transition ratio, no transfer}}$$

$$+ \max\{\frac{\gamma^2 T^2}{\eta}, 1\} \frac{u}{\min\{n_0, n_{\mathcal{M}}/K, n_{\mathcal{M}}\eta\}} \Bigg\}.$$

★Since transition density in $\mathcal{H}(\gamma_3)$, the rate is $n_k^{\frac{\gamma_3}{2\gamma_3+1}}$ and so are the density ratios.

Case II: Transferable densities

Let $\widehat{Q}_t^{(\mathrm{tr2})}$ be the estimator. Then, with probability $> 1 - 3T(K+1)e^{-u}$,

$$\|\widehat{Q}_t^{(\mathrm{tr2})} - Q_t^*\|_{2,\mathbb{P}_t^{(0)}}^2 \lesssim \cdots \Bigg\{ \underbrace{\left(\frac{J\log n_0}{n_0}\right)^{\frac{2\gamma_2}{2\gamma_2+1}}}_{\text{est. err. of reward differences}} + \underbrace{\frac{1}{\eta}\left(\frac{J\log n_{\mathcal{M}}}{n_{\mathcal{M}}}\right)^{\frac{2\gamma_1}{2\gamma_1+1}}}_{\text{est. err. of reward aggregation}}$$

$$+ \underbrace{\frac{\gamma^2 T^2}{\eta^2}\left(\frac{\log n_0}{n_0}\right)^{\frac{2\gamma_4}{2\gamma_4+1}}}_{\text{est. err. of transition differences}} + \underbrace{\frac{\gamma^2 T^2}{\eta^3}\left(\frac{K\log n_{\mathcal{M}}}{n_{\mathcal{M}}}\right)^{\frac{2\gamma_3}{2\gamma_3+1}}}_{\text{est. err. of transition aggregation}}$$

$$+ \max\left\{\frac{\gamma^2 T^2}{\eta}, 1\right\} \frac{u}{\min\{n_0\eta, n_{\mathcal{M}}\eta^2/K\}} + \frac{\gamma^2 T^2}{\eta}\frac{n_0^{\frac{1}{2\gamma_4+1}} K\log n_{\mathcal{M}}}{n_{\mathcal{M}}} \Bigg\}.$$

★Assume the transition ratio is smoother, in $\mathcal{H}(\gamma_4)$.

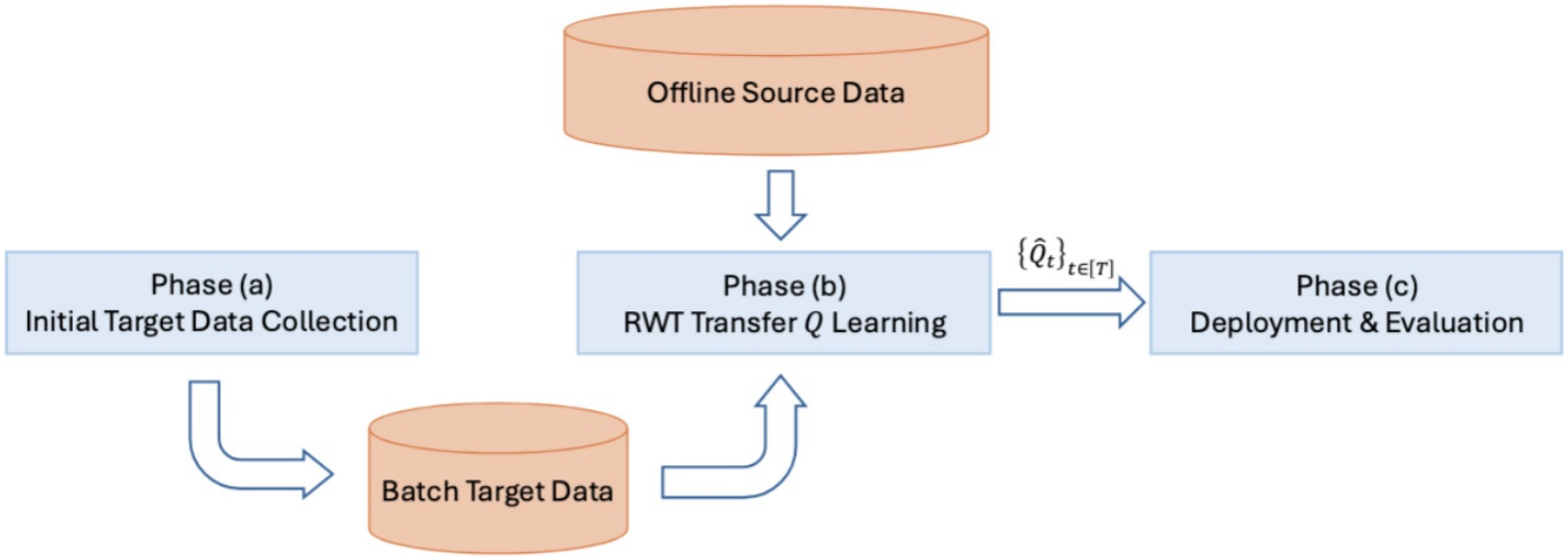
Numerical Studies

A Simulation Study

(a) Collect initial target data using uniform random policies.

(b) Apply RWT-Transfer Q-learning to both target and source data.

(c) Conduct online evaluation of the derived greedy policy from in the target environment.



Simulation Setting

★ **Two-stage MDP** *(Chakraborty et al, 2005, Song, et al, 2015)*

$$\Pr(S_1 = -1) = \Pr(S_1 = 1) = 0.5,$$
$$\Pr(A_t = -1) = \Pr(A_t = 1) = 0.5, \quad t = 1, 2,$$
$$\Pr(S_2 \mid S_1, A_1) = 1 - \Pr(S_2 \mid S_1, A_1) = \text{sig}(b_1 S_1 + b_2 A_1),$$

★ **Reward**: $R_1 = 0$, $\varepsilon_2 \sim N(0,1)$

$$R_2 = \kappa_1 + \kappa_2 S_1 + \kappa_3 A_1 + \kappa_4 S_1 A_1 + \kappa_5 A_2 + \kappa_6 S_2 A_2 + \kappa_7 A_1 A_2 + \varepsilon_2. \text{ (quadratic)}$$

★ True Q function has analytical form: $\theta_{2,j} = \kappa_j$, (Bellman eqn)

$$Q_2^*(S_2, A_2; \theta_2) = \theta_{2,1} + \theta_{2,2} S_1 + \theta_{2,3} A_1 + \theta_{2,4} S_1 A_1 + \theta_{2,5} A_2 + \theta_{2,6} S_2 A_2 + \theta_{2,7} A_1 A_2$$
$$Q_1^*(S_1, A_1; \theta_1) = \theta_{1,1} + \theta_{1,2} S_1 + \theta_{1,3} A_1 + \theta_{1,4} S_1 A_1$$

★ **Observed states**: $\mathbf{s}_t = (1, S_t, \tilde{\mathbf{s}}_t) \in \mathbb{R}^{31}$, $\tilde{\mathbf{s}}_t \sim N(0, \mathbf{I}_{29})$

Simulation Setting

★ **Two-stage MDP** *(Chakraborty et al, 2005, Song, et al, 2015)*

$$\Pr(S_1 = -1) = \Pr(S_1 = 1) = 0.5,$$
$$\Pr(A_t = -1) = \Pr(A_t = 1) = 0.5, \quad t = 1, 2,$$
$$\Pr(S_2 \mid S_1, A_1) = 1 - \Pr(S_2 \mid S_1, A_1) = \text{sig}(b_1 S_1 + b_2 A_1),$$

★ **Reward**: $R_1 = 0$, $\varepsilon_2 \sim N(0,1)$

$$R_2 = \kappa_1 + \kappa_2 S_1 + \kappa_3 A_1 + \kappa_4 S_1 A_1 + \kappa_5 A_2 + \kappa_6 S_2 A_2 + \kappa_7 A_1 A_2 + \varepsilon_2. \textbf{ (quadratic)}$$

★ True Q function has analytical form: $\theta_{2,j} = \kappa_j$, **(Bellman eqn)**

$$Q_2^*(S_2, A_2; \theta_2) = \theta_{2,1} + \theta_{2,2} S_1 + \theta_{2,3} A_1 + \theta_{2,4} S_1 A_1 + \theta_{2,5} A_2 + \theta_{2,6} S_2 A_2 + \theta_{2,7} A_1 A_2$$
$$Q_1^*(S_1, A_1; \theta_1) = \theta_{1,1} + \theta_{1,2} S_1 + \theta_{1,3} A_1 + \theta_{1,4} S_1 A_1$$

★ **Observed states**: $\mathbf{s}_t = (1, S_t, \tilde{\mathbf{s}}_t) \in \mathbb{R}^{31}$, $\tilde{\mathbf{s}}_t \sim N(0, \mathbf{I}_{29})$

Simulation Setting

★ **Two-stage MDP** *(Chakraborty et al, 2005, Song, et al, 2015)*

$$\Pr(S_1 = -1) = \Pr(S_1 = 1) = 0.5,$$
$$\Pr(A_t = -1) = \Pr(A_t = 1) = 0.5, \quad t = 1, 2,$$
$$\Pr(S_2 \mid S_1, A_1) = 1 - \Pr(S_2 \mid S_1, A_1) = \text{sig}(b_1 S_1 + b_2 A_1),$$

★ **Reward**: $R_1 = 0$, $\varepsilon_2 \sim N(0,1)$

$$R_2 = \kappa_1 + \kappa_2 S_1 + \kappa_3 A_1 + \kappa_4 S_1 A_1 + \kappa_5 A_2 + \kappa_6 S_2 A_2 + \kappa_7 A_1 A_2 + \varepsilon_2. \textbf{ (quadratic)}$$

★ True Q function has analytical form: $\theta_{2,j} = \kappa_j$, **(Bellman eqn)**

$$Q_2^*(S_2, A_2; \theta_2) = \theta_{2,1} + \theta_{2,2} S_1 + \theta_{2,3} A_1 + \theta_{2,4} S_1 A_1 + \theta_{2,5} A_2 + \theta_{2,6} S_2 A_2 + \theta_{2,7} A_1 A_2$$
$$Q_1^*(S_1, A_1; \theta_1) = \theta_{1,1} + \theta_{1,2} S_1 + \theta_{1,3} A_1 + \theta_{1,4} S_1 A_1$$

★ **Observed states**: $\mathbf{s}_t = (1, S_t, \tilde{\mathbf{s}}_t) \in \mathbb{R}^{31}$, $\tilde{\mathbf{s}}_t \sim N(0, \mathbf{I}_{29})$

Simulation Setting

★ **Two-stage MDP** *(Chakraborty et al, 2005, Song, et al, 2015)*

$$\Pr(S_1 = -1) = \Pr(S_1 = 1) = 0.5,$$
$$\Pr(A_t = -1) = \Pr(A_t = 1) = 0.5, \quad t = 1, 2,$$
$$\Pr(S_2 \mid S_1, A_1) = 1 - \Pr(S_2 \mid S_1, A_1) = \text{sig}(b_1 S_1 + b_2 A_1),$$

★ **Reward**: $R_1 = 0$, $\varepsilon_2 \sim N(0,1)$

$$R_2 = \kappa_1 + \kappa_2 S_1 + \kappa_3 A_1 + \kappa_4 S_1 A_1 + \kappa_5 A_2 + \kappa_6 S_2 A_2 + \kappa_7 A_1 A_2 + \varepsilon_2. \textbf{ (quadratic)}$$

★ True Q function has analytical form: $\theta_{2,j} = \kappa_j$, **(Bellman eqn)**

$$Q_2^*(S_2, A_2; \theta_2) = \theta_{2,1} + \theta_{2,2} S_1 + \theta_{2,3} A_1 + \theta_{2,4} S_1 A_1 + \theta_{2,5} A_2 + \theta_{2,6} S_2 A_2 + \theta_{2,7} A_1 A_2$$
$$Q_1^*(S_1, A_1; \theta_1) = \theta_{1,1} + \theta_{1,2} S_1 + \theta_{1,3} A_1 + \theta_{1,4} S_1 A_1$$

★ **Observed states**: $\mathbf{s}_t = (1, S_t, \tilde{\mathbf{s}}_t) \in \mathbb{R}^{31}$, $\tilde{\mathbf{s}}_t \sim N(0, \mathbf{I}_{29})$

Advantages of Transfer Learning

Target MDP: $b_1 = 1$, $b_2 = 1$, and $\kappa_j = 1$ for $j \leq 7$. **Source**: $\theta^{(1)}_{2,2} = 1.2$.

Target MDP: Stage 1 Reward coefficient: $\theta_{1,1}, \theta_{1,2}, \theta_{1,3}, \theta_{1,4} = 2.69, 1.19, 1.69, 1.19$

Source MDP: Stage 1 Reward coefficient: $\theta^{(1)}_{1,1}, \theta^{(1)}_{1,2}, \theta^{(1)}_{1,3}, \theta^{(1)}_{1,4} = 2.69, 1.39, 1.69, 1.19$.

Reward difference: $\theta_{1,2}$ for Q_1 and $\theta_{2,2}$ for Q_2 functions.

Advantages of Transfer Learning

Target MDP: $b_1 = 1$, $b_2 = 1$, and $\kappa_j = 1$ for $j \leq 7$. **Source**: $\theta^{(1)}_{2,2} = 1.2$.

Target MDP: Stage 1 Reward coefficient: $\theta_{1,1}, \theta_{1,2}, \theta_{1,3}, \theta_{1,4} = 2.69, 1.19, 1.69, 1.19$

Source MDP: Stage 1 Reward coefficient: $\theta^{(1)}_{1,1}, \theta^{(1)}_{1,2}, \theta^{(1)}_{1,3}, \theta^{(1)}_{1,4} = 2.69, 1.39, 1.69, 1.19$.

Reward difference: $\theta_{1,2}$ for Q_1 and $\theta_{2,2}$ for Q_2 functions.

Advantages of Transfer Learning

Target MDP: $b_1 = 1$, $b_2 = 1$, and $\kappa_j = 1$ for $j \leq 7$. **Source**: $\theta^{(1)}_{2,2} = 1.2$.

Target MDP: Stage 1 Reward coefficient: $\theta_{1,1}, \theta_{1,2}, \theta_{1,3}, \theta_{1,4} = 2.69, 1.19, 1.69, 1.19$

Source MDP: Stage 1 Reward coefficient: $\theta^{(1)}_{1,1}, \theta^{(1)}_{1,2}, \theta^{(1)}_{1,3}, \theta^{(1)}_{1,4} = 2.69, 1.39, 1.69, 1.19$.

Reward difference: $\theta_{1,2}$ for Q_1 and $\theta_{2,2}$ for Q_2 functions.

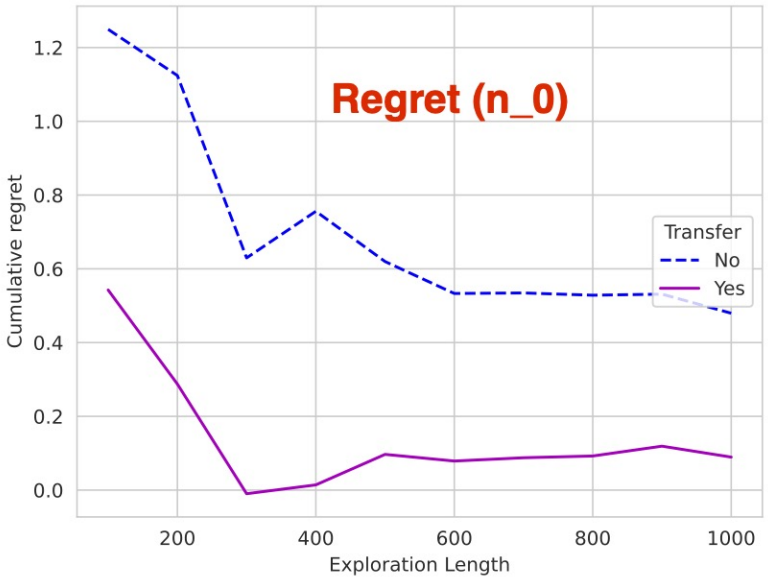
Advantages of Transfer Learning

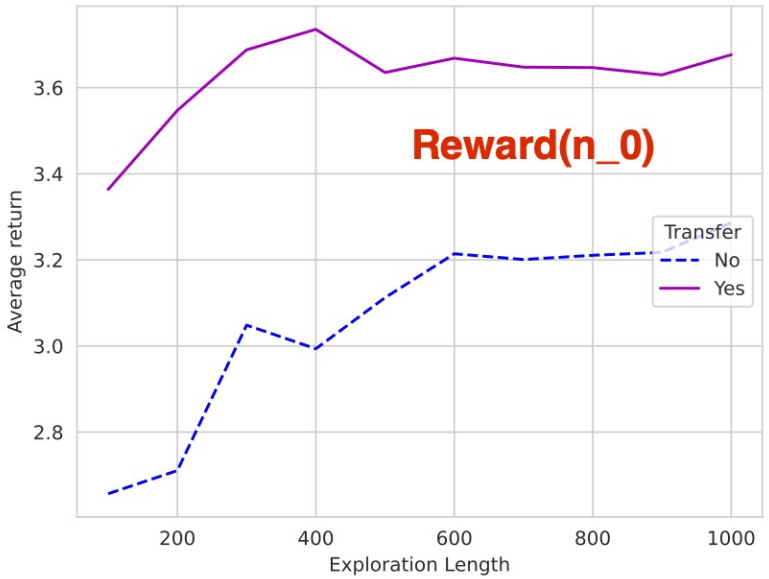
Target MDP: $b_1 = 1$, $b_2 = 1$, and $\kappa_j = 1$ for $j \leq 7$. **Source**: $\theta^{(1)}_{2,2} = 1.2$.

Target MDP: Stage 1 Reward coefficient: $\theta_{1,1}, \theta_{1,2}, \theta_{1,3}, \theta_{1,4} = 2.69, 1.19, 1.69, 1.19$

Source MDP: Stage 1 Reward coefficient: $\theta^{(1)}_{1,1}, \theta^{(1)}_{1,2}, \theta^{(1)}_{1,3}, \theta^{(1)}_{1,4} = 2.69, 1.39, 1.69, 1.19$.

Reward difference: $\theta_{1,2}$ for Q_1 and $\theta_{2,2}$ for Q_2 functions.





$n_0 = \text{x-axis}, \quad n_1 = 10K, \quad n_{eval} = 100$

MIMIC-III Sepsis Management

★ Medical Information Mart for Intensive Care III (*Johnson et al., 2016)*

- diverse and large no. patients of 6 ICUs in Boston (2001-12)
- high temporal resolution data ($T = 5$)
- lab results, electronic doc, and bedside monitor trends and waveforms.

★ Sepsis cohort (*Komorowski et al., 2018)*

- Entire sample 20,943 patients
- 11,704 (55.88 %) female (0), 9239 (44.11 %) male (1)
- Treatment decisions: volume of intravenous fluids and dose of vasopressors. Combinations of two treatments $= 3 \times 3 = 9$ actions.
- Rewards as in Prasad et al. (2017).

MIMIC-III Sepsis Management

★ Medical Information Mart for Intensive Care III (*Johnson et al., 2016)*

- diverse and large no. patients of 6 ICUs in Boston (2001-12)
- high temporal resolution data ($T = 5$)
- lab results, electronic doc, and bedside monitor trends and waveforms.

★ Sepsis cohort (*Komorowski et al., 2018)*

- Entire sample 20,943 patients
- 11,704 (55.88 %) female (0), 9239 (44.11 %) male (1)
- Treatment decisions: volume of intravenous fluids and dose of vasopressors. Combinations of two treatments $= 3 \times 3 = 9$ actions.
- Rewards as in Prasad et al. (2017).

MIMIC-III Sepsis Management

★ Medical Information Mart for Intensive Care III (*Johnson et al., 2016)*

- diverse and large no. patients of 6 ICUs in Boston (2001-12)
- high temporal resolution data ($T = 5$)
- lab results, electronic doc, and bedside monitor trends and waveforms.

★ Sepsis cohort (*Komorowski et al., 2018)*

- Entire sample 20,943 patients
- 11,704 (55.88 %) female (0), 9239 (44.11 %) male (1)
- Treatment decisions: volume of intravenous fluids and dose of vasopressors.
  Combinations of two treatments $= 3 \times 3 = 9$ actions.
- Rewards as in Prasad et al. (2017).

Empirical Study: MIMIC-III

Dataset Specifics

- ★ State features: $\mathbf{s}_{i,t} \in \mathbb{R}^3$. Top 3 PC = 99%.
  - •45 covariates, including demographics, vital signs, and laboratory values.
- ★ Action space [9]. •IV fluid $\in \{$low, med, high $\}$, •VASO $\in \{$low, med, high $\}$
- ★ n_0 varies, $n_1 = 10K$, $n_{eval} = 1000$. **Transfer** from female to male.

Empirical Study: MIMIC-III

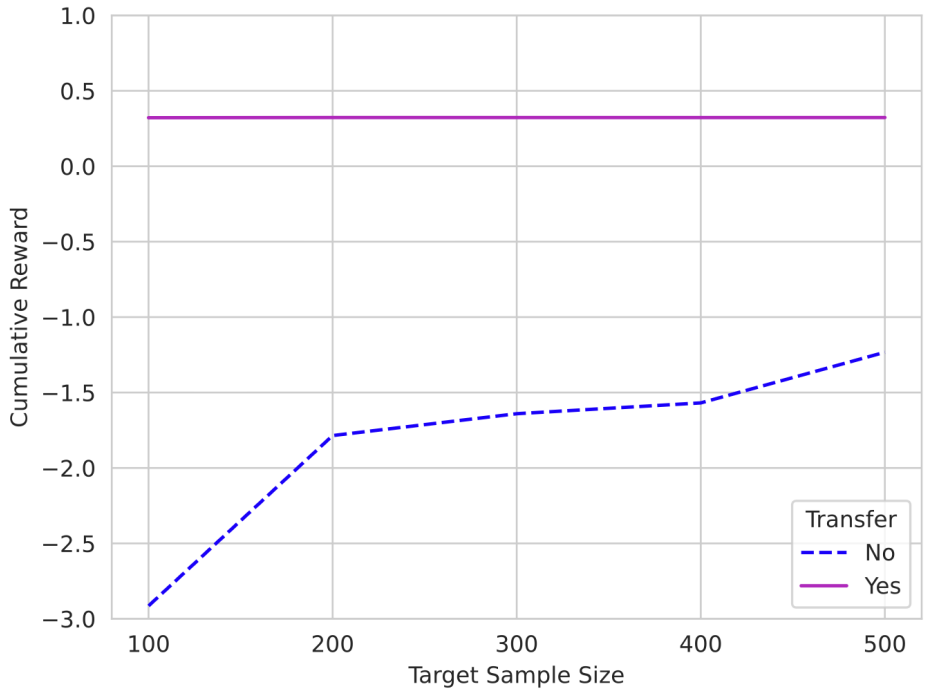
Dataset Specifics

★ State features: $\mathbf{s}_{i,t} \in \mathbb{R}^3$. Top 3 PC = 99%.

- 45 covariates, including demographics, vital signs, and laboratory values.

★ Action space [9]. - IV fluid $\in \{\text{low, med, high}\}$, - VASO $\in \{\text{low, med, high}\}$

★ n_0 varies, $n_1 = 10K$, $n_{eval} = 1000$. **Transfer** from female to male.



Summary

- ★ Propose a retargeting and reweighting method to construct transferable RL samples.
- ★ Propose general transfer RL framework for both value function and density ratio function estimation.
- ★ Establish $Q^\star$ estimation error using DNN and demonstrate the explicit **adaptive** rates of convergence for both **transferable** and **non-transferable** transition densities.
- ★ Implement our method both in simulated and real data.

Summary

- ★ Propose a retargeting and reweighting method to construct transferable RL samples.
- ★ Propose general transfer RL framework for both value function and density ratio function estimation.
- ★ Establish $Q^\star$ estimation error using DNN and demonstrate the explicit **adaptive** rates of convergence for both **transferable** and **non-transferable** transition densities.
- ★ Implement our method both in simulated and real data.

Summary

- ★ Propose a retargeting and reweighting method to construct transferable RL samples.
- ★ Propose general transfer RL framework for both value function and density ratio function estimation.
- ★ Establish $Q^\star$ estimation error using DNN and demonstrate the explicit **adaptive** rates of convergence for both **transferable** and **non-transferable** transition densities.
- ★ Implement our method both in simulated and real data.

Summary

- ★ Propose a retargeting and reweighting method to construct transferable RL samples.
- ★ Propose general transfer RL framework for both value function and density ratio function estimation.
- ★ Establish $Q^\star$ estimation error using DNN and demonstrate the explicit **adaptive** rates of convergence for both **transferable** and **non-transferable** transition densities.
- ★ Implement our method both in simulated and real data.

The End

Thank

You

—Chai, J., Chen, E., Fan, J.(2025). Deep Transfer Q-Learning for Offline Non-Stationary Reinforcement Learning. ***arxiv.org/pdf/2501.04870***