Transfer Q-learning for finite-horizon Markov decision processes

Electronic Journal of Statistics
Vol. 19 (2025) 5289–5312
ISSN: 1935-7524
https://doi.org/10.1214/25-EJS2459

Transfer Q-learning for finite-horizon Markov decision processes

Elynn Chen[1], **Sai Li**[2] **and Michael I. Jordan**[3]

[1]*Stern School of Business, New York University, NY, USA,*
e-mail: elynn.chen@stern.nyu.edu

[2]*Department of Statistics, Renmin University, Beijing, China,*
e-mail: saili@ruc.edu.cn

[3]*Department of EECS, University of California, Berkeley, CA, USA,*
e-mail: jordan@cs.berkeley.edu

Abstract: Time-inhomogeneous finite-horizon Markov decision processes (MDP) are frequently employed to model decision-making in dynamic treatment regimes and other statistical reinforcement learning (RL) scenarios. These fields, especially healthcare and business, often face challenges such as high-dimensional state spaces and time-inhomogeneity of the MDP process, compounded by insufficient sample availability which complicates informed decision-making. To overcome these challenges, we investigate knowledge transfer within time-inhomogeneous finite-horizon MDP by leveraging data from both a target RL task and several related source tasks. We have developed transfer learning (TL) algorithms that are adaptable for both batch and online Q-learning, integrating valuable insights from offline source studies. The proposed transfer Q-learning algorithm contains a novel *re-targeting* step that enables *cross-stage transfer* along multiple stages in an RL task, besides the usual *cross-task transfer* for supervised learning. We establish the first theoretical justifications of TL in RL tasks by showing a faster rate of convergence of the Q^*-function estimation in the offline RL transfer, and a lower regret bound in the offline-to-online RL transfer under stage-wise reward similarity and mild design similarity across tasks. Empirical evidence from both synthetic and real datasets is presented to evaluate the proposed algorithm and support our theoretical results.

Keywords and phrases: Dynamic treatment regimes, transfer learning, time-inhomogeneous finite-horizon Markov decision process, backward inductive Q-learning, high-dimensional estimation.

Received June 2024.

1. Introduction

Transfer learning aims at accelerating the learning process of a target task by leveraging knowledge acquired from different but similar source tasks. Recently, it has garnered substantial attention in statistical machine learning, due to its empirical successes in domains such as natural language processing, computer vision, game playing, and climate modeling. On the theoretical front, transfer learning has been investigated within a decision-theoretic framework across a spectrum of supervised learning problems, including classification [2], high-dimensional linear regression [23, 13], and generalized linear models [25], and there are also applications to problems in unsupervised learning scenarios [24].

Our focus is on transfer learning (TL) in the domain of time-inhomogeneous finite-horizon reinforcement learning. Reinforcement learning (RL) is a sequential decision-making framework that has natural applications in data-driven decision-making settings like robotics, business, healthcare, and education [22, 30, 31, 18]. One of the main challenges in RL is that the state space is often of high dimensionality, given that the goal is often that of providing for personalized services or treatments, and given the overall complexity of these domains [11, 26, 10, 9]. Thus there is a significant data scarcity problem in RL. Transfer learning is a promising framework for addressing data scarcity in RL, with the potential of expediting learning in a target RL task by leveraging related sources tasks from observational, simulated, or offline datasets. This motivates our interest in integrating of TL perspectives into the realm of RL.

As a first cautionary note, we will demonstrate that a straightforward application of TL algorithms designed for supervised learning (SL) does not yield minimax optimal performance within the RL context. The underlying problem has to do with the distinctive nature of multi-stage optimization and delayed responses in RL, and the cautionary note is equally pertinent in the offline RL setting as it is to the online RL setting. To address this challenge, we carefully design a novel framework based on backward inductive Q-learning [28, 9] and leverages pseudo-responses with re-targeting. It exploits a similarity structure for the reward function, incorporating offline transfer and online Q-learning with source offline data while effectively handling high-dimensional features. We provide a thorough theoretical analysis of this framework for TL in RL, specifically addressing the high dimensionality, and highlighting the distinctive nature of the RL setting compared to TL in SL.

We describe the contribution of our work in three aspects: new model formulation (for transferable MDPs), novel solution methodology (the re-targeting pseudo response to handle intermediate misalignment), and theoretical guarantees and practical application. Firstly, under general function approximations, we rigorously define the formulation of TL under finite-horizon MDPs, and unveil a significant distinction between TL in the realms of RL and SL. While the concept of *"cross-task transfer"* is present in both SL and RL, allowing knowledge exchange across source and target tasks within the same stage, the notion of *"cross-stage transfer"* is uniquely pertinent to the RL domain. This idea taps into future-stage information from source tasks to expedite the learning process in the current stage of the target task. While the multi-stage nature of RL tasks is well-acknowledged, its manifestation within TL has remained unexplored. In practice, due to the accumulation of differences between source and target tasks over stages, careful consideration is necessary to ensure the transfer's minimax performance.

This leads us to our second contribution, which involves two novel components integrated into the proposed transfer Q-learning algorithm. These components are designed to guarantee minimax optimality and enable both "cross-task transfer" and "cross-stage transfer." The first component establishes *pseudo-responses* for source tasks through a process called *"re-targeting,"* aligning the future state values of source samples with the corresponding states in the target model. The second component capitalizes on these *re-targeted pseudo-responses*, leveraging techniques from supervised transfer learning to estimate the target's optimal Q^*-function at each stage. This estimation process utilizes data from both the target task and the pseudo-source data, thus enabling "cross-task transfer."

Lastly, our contributions are substantiated through a combination of rigorous theoretical analysis and empirical evaluation. The theory uncovers conditions for effective transfer, offering practical insights. Empirical results affirm the potency of our methods in refining optimal Q^*-function estimation for offline RL tasks under reasonable assumptions. For online RL tasks, our framework demonstrates its value by trimming exploration phases and elevating cumulative rewards, substantiated by empirical findings. This collective evidence underscores the robust practicality of our approach.

Organization. The paper is organized as follows: Section 2 presents the model for Q-learning and transfer Q-learning. In Section 3, we introduce our proposed methods for offline and online Q-learning with knowledge transfer. Theoretical guarantees for our proposals are provided in Section 4. Section 5 presents an extensive analysis of the numerical performance of our methods in diverse simulation settings and a real-world medical data application. Finally, Section 6 concludes the paper, summarizing the key findings and highlighting avenues for future research.

2. Formulation of transfer between MDP's

The mathematical model of an time-inhomogeneous episodic RL task is a finite-horizon MDP defined as a tuple $\mathcal{M} = \{\mathcal{S}, \mathcal{A}, \Pr, r, \gamma, T\}$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, r is the reward function, $\gamma \in [0, 1]$ is the discount factor, and T is the finite horizon. At time t, for the i-th individual, an agent observes the current system state $\boldsymbol{s}_{t,i} \in \mathcal{S}$, chooses a decision $a_{t,i}$ from a finite action set $\mathcal{A} = \{1, \cdots, M\}$, and transits to the next state $\boldsymbol{s}_{t+1,i}$ according to the system transition probability $\Pr\left(\boldsymbol{s}_{t+1,i}|\boldsymbol{s}_{t,i}, a_{t,i}\right)$. At the same time, she receives an immediate reward $r_{t,i}$, which serves as a partial signal of the goodness of her action $a_{t,i}$. The reward function $r(\cdot)$ is the expected reward at stage t for the observation $(\boldsymbol{s}_{t,i}, a_{t,i})$, that is, $r(\boldsymbol{s}_{t,i}, a_{t,i}) = \mathbb{E}[r_{t,i}|\boldsymbol{s}_{t,i}, a_{t,i}]$.

An agent's decision-making rule is denoted by a policy function $\pi\left(a_{t,i}|\boldsymbol{s}_{t,i}\right)$ that maps the covariate space $\mathcal{S}$ to probability mass functions on the action space $\mathcal{A}$. For each step $t \in [T]$ and a policy π, the *state-value function* is the expectation of the cumulative discounted reward starting from a state $\boldsymbol{s}$ at the t-th step:

$$V_t^{\pi}(\boldsymbol{s}) = \mathbb{E}^{\pi}\left[\sum_{s=t}^{T} \gamma^{s-t} r(\boldsymbol{s}_{s,i}, a_{s,i}) \middle| \boldsymbol{s}_{t,i} = \boldsymbol{s}\right]. \tag{1}$$

The expectation $\mathbb{E}^{\pi}$ is taken under the trajectory distribution, assuming that the dynamic system follows the given policy π afterwards. Accordingly, the *action-value function* or Q^{π}-*function* of a given policy π at step t is the expectation of the accumulated discounted reward starting from a state $\boldsymbol{s}$ and taking action a:

$$Q_t^{\pi}(\boldsymbol{s}, a) = \mathbb{E}^{\pi}\left[\sum_{s=t}^{T} \gamma^{s-t} r(\boldsymbol{s}_{s,i}, a_{s,i}) \middle| \boldsymbol{s}_{t,i} = \boldsymbol{s}, a_{t,i} = a\right]. \tag{2}$$

For any given action-value function $Q_t^{\pi} : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$, the *greedy policy* π_t^{Q} is defined as,

$$\pi_t^{Q}(a|\boldsymbol{s}) = \begin{cases} 1 & \text{if } a = \arg\max_{a' \in \mathcal{A}} Q_t^{\pi}(\boldsymbol{s}, a'), \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

The overall goal of RL is to learn an optimal policy, $\pi_t^*, t \in [T]$, that maximizes the discounted accumulative reward. To characterize optimality, we define the optimal action-value function Q_t^* as

$$Q_t^*(\boldsymbol{s}, a) = \sup_{\pi \in \Pi} Q_t^{\pi}(\boldsymbol{s}, a), \quad \forall (\boldsymbol{s}, a) \in \mathcal{S} \times \mathcal{A}. \tag{4}$$

The optimal policy π_t^* can be derived as any policy that is greedy with respect to Q_t^*. The *Bellman optimal equation* holds that

$$\mathbb{E}\left[r_{t,i} + \gamma \max_{a' \in \mathcal{A}} Q_{t+1}^*\left(\boldsymbol{s}_{t+1,i}, a'\right) - Q_t^*\left(\boldsymbol{s}_{t,i}, a_{t,i}\right) \middle| \boldsymbol{s}_{t,i}, a_{t,i}\right] = 0. \tag{5}$$

The Q-learning algorithms estimate the optimal Q_t^* directly and then obtain the optimal π^* as the greedy policy (3) according to Q_t^*. The MDP considered in this paper is time-inhomogeneous since the value functions, specifically $Q_t^{\pi}(\boldsymbol{s}, a)$, $Q_t^*(\boldsymbol{s}, a)$, and $V_t^{\pi}(\boldsymbol{s})$, depend on the stage index t. But they are fixed across episodes; we do not assume stochastic drift over episodes.

2.1. Target and source MDPs

Knowledge transfer aims to leverage data from source RL tasks that are *similar* to the target task to improve learning of the target RL task. We consider source data from offline observational data or simulated data. The target task can be an offline or online RL task. We consider K source tasks from a set $\mathcal{K}$. The target RL task of interest is referred to as the 0-th task and written with a superscript "(0)," while the source RL tasks are written with a superscript "(k)," for $k \in \mathcal{K}$.

Denote the optimal Q^*-functions for the target and all source tasks by $Q_t^{*(k)}(\boldsymbol{s}, a)$ for $t \in [T]$, $k \in \{0\} \cup \mathcal{K}$, with the convention that $Q_{T+1}^*(\boldsymbol{s}, a) = 0$. The random trajectories for the k-th source task are generated from the MDP $\mathcal{M}^{(k)} = \left\{\mathcal{S}, \mathcal{A}, \Pr^{(k)}, r^{(k)}, \gamma, T\right\}$. Without loss of generality, we assume the horizon length of all tasks are the same and we denote that length as T. For each task $0 \le k \le K$, we collect n_k trajectories of length T, denoted $\{\boldsymbol{s}_{t,i}^{(k)}, a_{t,i}^{(k)}, r_{t,i}^{(k)}\}_{t \in [T], i \in [n_k]}$. We also assume that the trajectories in different tasks are independent and n_k does not depend on stage t, i.e., none of the tasks have missing data.

Single-task RL considers each task $k \in \{0\} \cup \mathcal{K}$ separately and denote the underlying *true response* at step t as

$$y_{t,i}^{(k)} := r_{t,i}^{(k)} + \gamma \cdot \max_{a \in \mathcal{A}} Q_{t+1}^{*(k)}(\boldsymbol{s}_{t+1,i}^{(k)}, a). \tag{6}$$

According to (5), we have

$$\mathbb{E}\left[y_{t,i}^{(k)} - Q_t^{*(k)}\left(\boldsymbol{s}_{t,i}^{(k)}, a_{t,i}^{(k)}\right) \middle| \boldsymbol{s}_{t,i}^{(k)}, a_{t,i}^{(k)}\right] = 0, \quad \text{for} \quad k \in \{0\} \cup \mathcal{K}, \tag{7}$$

which provides a moment condition for the estimation of function $Q_t^{*(k)}$. If $y_{t,i}^{(0)}$ is directly observable, then $Q_t^{*(0)}$ can be estimated via regression. However, what we observe is a "partial response" $r_{t,i}^{(0)}$. The other component of $y_{t,i}^{(0)}$, as shown in the second term on the RHS of (6), depends on the unknown Q^*-function and future observations. As will be discussed in detail in Section 3, we estimate $Q_t^{*(0)}$ in a backward fashion for $t = T, \cdots, 1$.

Remark 1. In this paper we focus on the case where task similarity is captured primarily through *reward differences*. Although transition dynamics may also differ across tasks ($\Pr^{(k)} \neq \Pr^{(0)}$), we do not develop full theoretical guarantees for transition transfer here. Instead, we sketch several methodological ideas in Section 3.5 and leave a comprehensive treatment to future work.

2.2. Similarity characterization on reward functions

We leverage the similarity of the rewards for the target and source tasks at each stage. Defining RL task difference through the reward function has several advantages. Firstly, it is widely applicable to health care, business, and marketing scenarios where individuals may respond slightly differently (in terms of rewards) to the same treatment. For instance, in many applications, reward functions represent the revenue generated in a given time period. Discrepancies in revenue across different markets at the same time period are a significant source of divergence in total discounted revenue over a finite time horizon.

Secondly, the similarity assumptions introduced on equation (8) can be practically validated, as rewards for all stages are *directly observable*. Lastly, by permitting non-zero difference functions $D^{(k)}(\boldsymbol{s}, a)$, our task similarity definition is more generalized compared to that in [21] and [27]. In these works, they assume that optimal Q^*-functions are identical between different MDPs, which is a more restrictive condition.

Specifically, we define the difference between the reward functions of the target and the k-th source task via

$$\mathbb{E}\left[r_t^{(0)}(\boldsymbol{s}_{t,i}, a_{t,i}) - r_t^{(k)}(\boldsymbol{s}_{t,i}, a_{t,i}) \middle| \boldsymbol{s}_{t,i} = \boldsymbol{s}, a_{t,i} = a\right] = D_t^{(k)}(\boldsymbol{s}, a), \tag{8}$$

for $t \in [T]$ and $k \in [K]$ where $D_t^{(k)}(\boldsymbol{s}, a)$ is the the discrepancy function between reward functions. When this difference, denoted as $D_t^{(k)}(\boldsymbol{s}, a)$, is deemed "small," we have the capability to consolidate all source and target data to estimate the "large" common component across the data. At the same time, we can utilize only the target data to estimate the "small" differences. We will formally define the magnitude of difference $D_t^{(k)}(\boldsymbol{s}, a)$ in Section 4.

The similarity quantification (8) also has a potential outcome point of view. Specifically, for a realized state-action pair $(\boldsymbol{s}, a)$, the difference of reward values when switching its participation from the k-th task to the target task is $D_t^{(k)}(\boldsymbol{s}, a)$. Because one individual only participates one study in the current setting, the "switching" describes unobserved counterfactual facts [17]. Nevertheless, empowered by the potential response framework, we are able to generate counterfactual estimates using samples from k-th study and estimated coefficients of the target study.

3. Transfer Q-learning algorithms for time-inhomogeneous MDPs

In Section 3.1, we delineate the novel challenges and phenomena inherent in TL for RL, while also presenting the rationale behind our proposed methods. In Section 3.2, we introduce a Q-learning method for offline-to-offline transfer. In Section 3.3, we generalize our proposal to deal with offline-to-online transfer. All discussions are under a general function setting.

3.1. New challenge and phenomena in TL for RL

We utilize the Bellman optimal equations (5) and (7) to estimate the optimal Q^*-function for the target task. A fundamental distinction between RL and SL lies in the unavailability of the true response $y_{t,i}^{(0)}$, as defined in (6).

In the context of single-task Q-learning, pseudo-responses $\widehat{y}_{t,i}^{(k)}$ for $k \in 0 \cup [K]$ are typically constructed using the observed $r_{t,i}^{(k)}$ and an estimator of $Q_{t+1}^*(\boldsymbol{s}, a)$. Thus, an immediate extension of transfer learning (TL) for SL involves augmenting target pseudo-samples $\{\boldsymbol{s}_{t,i}^{(0)}, \widehat{y}_{t,i}^{(0)}\}$ with pseudo-samples $\{\boldsymbol{s}_{t,i}^{(k)}, \widehat{y}_{t,i}^{(k)}\}$ from the source tasks $k \in \mathcal{K}$. Here, *pseudo-response* $\widehat{y}_{t,i}^{(k)}$ is defined as:

$$\widehat{y}_{t,i}^{(k)} := r_{t,i}^{(k)} + \gamma \cdot \max_{a \in \mathcal{A}} Q_{t+1}^{*(k)}(\boldsymbol{s}_{t+1,i}^{(k)}, a), \quad \text{for} \quad k \in \{0\} \cup \mathcal{K}. \tag{9}$$

However, this straightforward construction of pseudo-responses introduces an additional bias due to the mismatch between the source $\max_a Q_{t+1}^{*(k)}(\boldsymbol{s}, a)$ and the target $\max_a Q_{t+1}^{*(0)}(\boldsymbol{s}, a)$ on a population level. Specifically, by combining equations (6) and (8), we the following moment equations:

$$\mathbb{E}\left[y_{t,i}^{(0)} \mid \boldsymbol{s}_{t,i} = \boldsymbol{s}, a_{t,i} = a\right] = Q_t^{*(0)}(\boldsymbol{s}, a), \tag{10}$$

$$\mathbb{E}\left[y_{t,i}^{(k)} \mid \boldsymbol{s}_{t,i}^{(k)} = \boldsymbol{s}, a_{t,i}^{(k)} = a\right] = Q_t^{*(0)}(\boldsymbol{s}, a) + D_t^{(k)}(\boldsymbol{s}, a) + \text{Bias}(\boldsymbol{s}, a), \tag{11}$$

where $D_t^{(k)}(\boldsymbol{s}, a)$ is typically assumed to be small because of the similarity between tasks, but the additional error

$$\begin{aligned}\text{Bias}(\boldsymbol{s}, a) = &\mathbb{E}\left[\gamma \cdot \max_{a \in \mathcal{A}} Q_{t+1}^{*(0)}(\boldsymbol{s}_{t+1,i}^{(0)}, a) \mid \boldsymbol{s}_{t,i}^{(0)} = \boldsymbol{s}, a_{t,i}^{(0)} = a\right] \\ &- \mathbb{E}\left[\gamma \cdot \max_{a \in \mathcal{A}} Q_{t+1}^{*(k)}(\boldsymbol{s}_{t+1,i}^{(k)}, a) \mid \boldsymbol{s}_{t,i}^{(k)} = \boldsymbol{s}, a_{t,i}^{(k)} = a\right]\end{aligned}$$

resulting from task differences is not inherently minor.

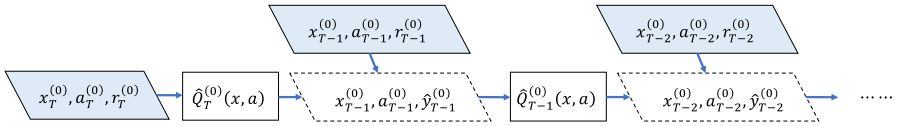
Our innovative approach involves constructing pseudo-responses for the source tasks through a process of *"re-targeting"* them to the target model. To grasp the rationale behind this approach, let's consider the population level. At any time step t, if we possess the true value of $Q_{t+1}^{*(0)}(\cdot, \cdot)$, we can design a *"re-targeted response"* $y_{t,i}^{(tl-k)}$, using source samples, as follows:

$$y_{t,i}^{(tl-k)} := r_{t,i}^{(k)} + \gamma \cdot \max_{a \in \mathcal{A}} Q_{t+1}^{*(0)}(\boldsymbol{s}_{t+1,i}^{(k)}, a). \tag{12}$$

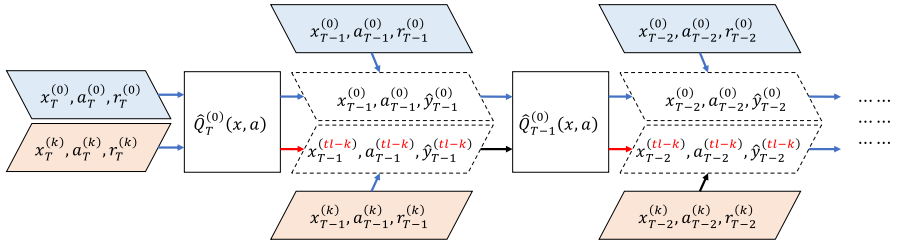
Remarkably, $Q_{t+1}^{*(0)}(\cdot, \cdot)$ is not the true Q^* function of the k-th task, but rather that of the target task. Notably, $y_{t,i}^{(tl-k)}$ is generated by aligning its future state with the future state of the target model. By combining equations (6), (8), and (12), we derive the following moment equation:

$$\mathbb{E}\left[y_t^{(tl-k)} \mid \boldsymbol{s}_{t,i}^{(k)} = \boldsymbol{s}, a_{t,i}^{(k)} = a\right] = Q_t^{*(0)}(\boldsymbol{s}, a) + D_t^{(k)}(\boldsymbol{s}, a). \tag{13}$$

It is worth-noting that $Q_t^{*(0)} + D_t^{(k)} \neq Q_t^{*(k)}$ and we do not impose any assumption on $Q_t^{*(k)} - Q_t^{*(0)}$. Comparing (10), (11), and (13), we can see that, through the re-targeting, the discrepancy



(a) Single-task Q-learning exclusively employs data from the target task (depicted by blue blocks). The calculation occurs in a reverse manner, and $\widehat{y}_t^{(0)}$ signifies pseudo-responses that are not directly observed.



(b) Transfer Q-learning utilizes data from both the target task (represented by blue blocks) and the k-th source task (depicted by orange blocks). $\widehat{Q}_t^{(0)}(x,a)$ stands for the estimated optimal Q^*-function at step t for the target task. The re-targeting step (highlighted in red) prevents errors caused by the disparity between $Q_t^{*(0)}$ and $Q_t^{*(k)}$ from accumulating across stages.

Fig 1. *Illustration of single-task Q-learning and transfer Q-learning. Naive application of transfer algorithms for regression without re-targeting will incur extra accumulated bias along the red arrows.*

between the expected values of $y_{t,i}^{(0)}$ and $y_{t,i}^{(tl-k)}$ solely arises from the inconsistency between their reward functions at stage t.

This rationale underlies the concept of the *"re-targeted pseudo responses"* $\widehat{y}_t^{(tl-k)}$ for the source tasks $k \in [K]$, which is defined by

$$\widehat{y}_{t,i}^{(tl-k)} := r_{t,i}^{(k)} + \gamma \cdot \max_{a \in \mathcal{A}} \widehat{Q}_{t+1}^{(0)}(\boldsymbol{s}_{t+1,i}^{(k)}, a), \tag{14}$$

where $\widehat{Q}_{t+1}^{(0)}$ is an estimate of $Q_{t+1}^{*(0)}$. For the target task, we use simply the pseudo responses $\widehat{\boldsymbol{y}}_t^{(0)}$, defined as

$$\widehat{y}_{t,i}^{(0)} := r_{t,i}^{(0)} + \gamma \cdot \max_{a \in \mathcal{A}} \widehat{Q}_{t+1}^{(0)}(\boldsymbol{s}_{t+1,i}^{(0)}, a). \tag{15}$$

The premise of transfer learning is the presence of abundant source data and relatively minor differences. Consequently, we can effectively learn the term $Q_t^{(0)}(\boldsymbol{s},a) + D_t^{(k)}(\boldsymbol{s},a)$ in (13) by leveraging a substantial amount of source data. Then, due to the small differences, we can also achieve accurate learning of the difference, or bias term, $D_t^{(k)}(\boldsymbol{s},a)$, between (10) and (13), even with a limited amount of target data. In summary, the transferred estimation for $Q_t^{(0)}(\boldsymbol{s},a)$ surpasses single-task estimation due to these factors.

Figure 1 visually outlines the process of our proposed transfer Q-learning algorithm in contrast to the conventional single-task Q-learning approach. It is evident from the illustration that our method employs information aggregation through two key mechanisms. Firstly, during the estimation of $\widehat{Q}_t^{(0)}$ at stage t, we leverage information from the t-th stage of source tasks, represented by $\{\boldsymbol{r}_{t,i}^{(k)}\}_{k=1}^K$. This process is referred to as "cross-task transfer," which is similar to the phenomena of transfer learning for supervised regression.

Secondly, the estimated pseudo response for the target model, denoted as $\widehat{\boldsymbol{y}}_{t,i}^{(0)}$, is constructed based on the transfer learning estimator $\widehat{Q}_{t+1}^{(0)}$. This introduces a positive cascading effect across stages: improved accuracy in estimating $\widehat{Q}_s^{(0)}$, for $t < s \leq T$, enhances the estimation of the second term in (6). Consequently, this enhancement positively influences the estimation of $\widehat{Q}_t^{(0)}$ in subsequent steps. As a direct result, the accuracy of the current-step estimator $\widehat{Q}_t^{(0)}$ benefits, given that the current response estimation $\widehat{\boldsymbol{y}}_{t,i}^{(0)}$ exhibits lower error compared to its single-task counterpart.

This phenomenon, distinct to RL transfer learning, is termed "cross-stage transfer," and it serves as an additional layer of information exchange that enhances the algorithm's performance.

3.2. Offline-to-offline transfer

Building upon the foundational principle outlined in the previous section, we introduce a transfer Q-learning algorithm that follows a backward inductive approach, designed specifically for offline-to-offline transfer.

The algorithm's flow is summarized in Algorithm 1. Starting from the final stage T where we have full observations of $\{\boldsymbol{s}_{T,i}^{(k)}, a_{T,i}^{(k)}, y_{T,i}^{(k)}\}_{i\in[n_k]}$ for each task, we apply supervised transfer learning algorithms to estimate the target function $\widehat{Q}_T^{(0)}$. Proceeding backwardly to stages $t = T-1, \cdots, 1$, we calculate the pseudo response $y_{t,i}^{(0)}$ in (15) using $\widehat{Q}_{t+1}^{(0)}$, thereby estimating the Q^*-function's observations for the target task at stage t. To leverage the source data, we calculate the re-targeted pseudo response $y_{t,i}^{(tl-k)}$ in (14) with the help of $\widehat{Q}_{t+1}^{(0)}$.

The general transfer learning procedure involves two main steps: (i) It begins by estimating a pooled estimator from the combined source re-targeted pseudo data $y_{t,i}^{(tl-k)}$ for $k \in [K]$; (ii) It then estimates the bias using the target pseudo data $y_{t,i}^{(0)}$ and estimated pooled $\widehat{Q}$. The final estimate $\widehat{Q}_{t+1}^{(0)}$ is obtained by debiasing the pooled estimator with the estimated bias.

The penalty term $\lambda_0 \cdot p(D)$ serves to enforce the "small" difference assumption, while the penalty term $\lambda_{\text{src}} \cdot p(Q)$ prevents overfitting of the pooled Q^* function.

Adaptability of the supervised regression transfer block. The component of the supervised regression transfer block is designed to be flexible, allowing for seamless substitution with various transfer learning methodologies from supervised learning, contingent upon their assumptions and objectives [36, 1, 13, 25]. An instance of transfer Q-learning with linear Q^* and TransLASSO [23] is detailed in Section 4 and Algorithm 3.

3.3. Offline-to-online transfer

Now, we introduce the offline-to-online transfer Q-learning algorithm. In this context, transfer learning can provide an efficient way to leverage the offline data, including demonstrations for the target task, suboptimal policies and demonstrations for the source tasks. Algorithm 2 outlines the online transfer Q-learning framework, grounded in the exploration and then commit (ETC) paradigm.

Algorithm 2 first generates n_e trajectories from the target task in the exploration phase. Then in the learning phase, Algorithm 1 is called to combine the target data with the offline

Algorithm 1: Transfer Q-learning (General)

Input : Target data $\{\{s_{t,i}^{(0)}, a_{t,i}^{(0)}, r_{t,i}^{(0)}\}_{t\in[T]}\}_{i\in[n_0]}$,
source data $\{\{\{s_{t,i}^{(k)}, a_{t,i}^{(k)}, r_{t,i}^{(k)}\}_{t\in[T]}\}_{i\in[n_k]}\}_{k=1}^{K}$, and
discount factor $\gamma \in [0,1]$.

Let $\widehat{Q}_{T+1}^{(0)}(\cdot) = 0$ to deal with pseudo-response construction at last stage T.

for $t = T, \ldots, 1$; `// Backward calculation from the last stage.`

do

Construct re-targeted pseudo-response $\widehat{y}_{t,i}^{(tl-k)}$ (14) and $\widehat{y}_{t,i}^{(0)}$ (15).

`/* Supervised regression transfer block: aggregate and debias. */`

Transfer learning:

$$\widehat{Q}_t = \operatorname{argmin}_Q \left\{ \sum_{k=1}^{K} \sum_{i=1}^{n_k} \ell\left(\widehat{y}_{t,i}^{(tl-k)} - Q(s_{t,i}^{(k)}, a_{t,i}^{(k)})\right) + \lambda_{\text{src}} \cdot p(Q) \right\}.$$

$$\widehat{D}_t = \operatorname{argmin}_D \left\{ \sum_{i=1}^{n_0} \ell\left(\widehat{y}_{t,i}^{(0)} - \widehat{Q}_t(s_{t,i}^{(0)}, a_{t,i}^{(0)}) - D(s_{t,i}^{(0)}, a_{t,i}^{(0)})\right) + \lambda_0 \cdot p(D) \right\}.$$

where $\ell(\cdot)$ and $p(\cdot)$ are any suitable loss and regularization functions chosen for desired scenarios.

Set $\widehat{Q}_t^{(0)} = \widehat{Q}_t + \widehat{D}_t$.

Output: $\widehat{Q}_t^{(0)}$ for all stage $t \in [T]$.

source data to estimate the target parameters. Lastly in the exploitation phase, it executes a greedy action to maximize the estimated Q^*-function at each stage afterwards. The longer the exploration phase, the larger the regret since the estimated optimal policy is not employed in the exploration phase. We will show in theory that n_e can be set much smaller by leveraging offline data under certain conditions. Specifically, it suffices to take $n_e \geq c \log p$ in Algorithm 2 when having a large amount of offline data. In contrast, without offline data, we show that one needs to take $n_e \geq cs \log p$. This is because one can learn a near-optimal policy based on the offline data when the offline tasks are sufficiently similar to the target task.

3.4. Streaming retargeting for online sources

The re-targeting step in Algorithm 1 requires only the observed transition $(s_t^{(k)}, a_t^{(k)}, r_t^{(k)}, s_{t+1}^{(k)})$ and the current estimate of the target value $\widehat{Q}_{t+1}^{(0)}$:

$$\widehat{y}_{t,i}^{tl-k} = r_{t,i}^{(k)} + \gamma \max_{a\in\mathcal{A}} \widehat{Q}_{t+1}^{(0)}(s_{t+1,i}^{(k)}, a).$$

Thus, re-targeting never requires observing actions beyond $t+1$. This observation allows us to adapt the method efficiently to streaming settings.

Episode-buffered approach. A straightforward strategy is to buffer full episodes from each source and, once an episode terminates, run a backward pass to re-target labels and update estimates. While statistically valid, this strategy is computationally expensive, as it requires waiting for complete episodes.

One-step re-targeting. In streaming sources, a more efficient approach is to form re-targeted labels online. As soon as a transition is completed, we immediately compute the re-targeted

Algorithm 2: Explore-Transfer-Then-Commit: Offline to Online Transfer

Input : Offline source data $\{\{\{s_{t,i}^{(k)}, a_{t,i}^{(k)}, r_{t,i}^{(k)}\}_{t\in[T]}\}_{i\in[n_k]}\}_{k=1}^{K}$.
Exploration phase length n_e, an exploration policy π_e.

# Exploration phase
for $i = 1, \ldots, n_e$ **do**
  for $t = 1, \ldots, T$ **do**
    Take action $a_{t,i}^{(0)} = \pi_e(\cdot|s_{t,i}^{(0)})$, get $r_{t,i}^{(0)}$ and $s_{t+1,i}^{(0)}$.
Transfer phase
Run Algorithm 1 with target data $\{\{s_{t,i}^{(0)}, a_{t,i}^{(0)}, r_{t,i}^{(0)}\}_{t\in[T]}\}_{i\in[n_e]}$ and offline source data $\{\{\{s_{t,i}^{(k)}, a_{t,i}^{(k)}, r_{t,i}^{(k)}\}_{t\in[T]}\}_{i\in[n_k]}\}_{k=1}^{K}$.
Output $\{\widehat{Q}_t^{(0)}\}_{t=1}^{T}$.
# Exploitation phase
for $i = n_e + 1, \ldots,$ **do**
  for $t = 1, \ldots, T$ **do**
    Take greedy action $\widehat{a}_{t,i} = \arg\max_{a\in[M]} \widehat{Q}_t^{(0)}(s_{t,i}^{(0)}, a)$ and transit to $s_{t+1,i}^{(0)}$.

Output: Online actions $\{\widehat{a}_{t,i}^{(0)}\}_{t\in[T]}$, $i = 1, \ldots$

label using the current $\widehat{Q}_{t+1}^{(0)}$ and add it to a per-stage buffer. This requires only a one-step delay (until s_{t+1} is observed), not the full episode.

Mini-batch backward updates. We maintain stage-wise buffers $\mathcal{B}_t^{\text{src}}$ and $\mathcal{B}_t^{\text{tgt}}$ of recent tuples and their re-targeted labels. Periodically (e.g., every H new target episodes or when $\widehat{Q}^{(0)}$ changes materially), we perform a short backward sweep over $t = T, \ldots, 1$: (i) re-label only the right-hand sides (since feature matrices W_t remain fixed), and (ii) update the pooled and de-bias regressions using warm starts (Algorithm 3 in the linear case). This procedure preserves the "cross-stage" backward structure while avoiding full-episode relabeling, yielding substantial computational savings.

In summary, one-step streaming re-targeting ensures that our framework extends naturally to continuous data arrival, while maintaining both statistical validity and computational efficiency.

3.5. Beyond reward similarity

Our current framework focuses on stage-wise reward similarity with mild design similarity across tasks, and does not provide guarantees under general transition shifts. While this setting is broad and practically verifiable, important extensions remain. In particular, when transition dynamics differ significantly across tasks, re-targeting alone may leave residual bias.

We briefly outline several promising directions for handling such transition shifts:

- **Importance-weighted re-targeting.** By estimating conditional transition density ratios $\omega_t^{(k)}(s' \mid s, a) = \frac{p_t^{(0)}(s'|s,a)}{p_t^{(k)}(s'|s,a)}$, source transitions can be re-weighted so that their conditional distribution aligns with the target's. This corrects for covariate shift in next-state features and provides unbiased re-targeted samples when transitions differ.

- **Representation alignment.** In linear and feature-based MDPs, one can estimate or learn successor features that capture long-run transition dynamics. Aligning these representations across tasks, through penalization or constrained regression, allows pooled estimation to remain valid even under transition heterogeneity.
- **Robust transfer via distributional penalties.** Rather than assuming exact similarity, one can add penalties based on integral probability metrics (IPMs) such as Maximum Mean Discrepancy (MMD) or Wasserstein distance to discourage reliance on sources whose next-state feature distributions diverge from the target [33, 12]. This yields error bounds with explicit additive terms reflecting the magnitude of transition shift.

These extensions call for new estimation tools and theoretical analysis, but they provide a principled roadmap for expanding our framework beyond the reward-similarity regime studied here. Recent papers in transfer learning in RL has explored some of those directions [6, 3, 38, 4, 39]. A full development of these methods is beyond the scope of this paper and will be pursued in future work.

4. Theoretical results

For our theoretical analysis we focus on the *linear function approximation* setting (Algorithm 3), which is widely adopted in applications across health, economics, and business. Recall that Algorithm 1 specifies the general transferred Q-learning framework; in this section we analyze its *linear instantiation*, Algorithm 3. All formal guarantees, including Theorem 4.2, are therefore stated for Algorithm 3. While the overall proof strategy extends in spirit to semi-parametric, nonparametric, or other general function approximation classes, such cases require substantially different technical tools. We view these extensions as important directions for future research.

The optimal action-value and the reward difference function are parameterized as

$$Q_t^{*(k)}(\boldsymbol{s}, a) = Q_t^*(\boldsymbol{s}, a;\ \boldsymbol{\theta}^{(k)}) = \boldsymbol{w}(\boldsymbol{s}, a)^\top \boldsymbol{\theta}_t^{(k)}, \tag{16}$$

$$D_t^{(k)}(\boldsymbol{s}, a) = D_t(\boldsymbol{s}, a;\ \boldsymbol{\delta}^{(k)}) = \boldsymbol{w}(\boldsymbol{s}, a)^\top \boldsymbol{\delta}_t^{(k)}, \tag{17}$$

for $k \in \{0\} \cup [K]$ where $\boldsymbol{w}(\boldsymbol{s}, a) = [\boldsymbol{\phi}(\boldsymbol{s})^\top \mathbb{1}(a = 1), \cdots, \boldsymbol{\phi}(\boldsymbol{s})^\top \mathbb{1}(a = M)]$, $\phi(\cdot)$ are fixed nonlinear feature functions and can be viewed as the state representation. Linear Q^*-functions on features have been widely used in literature of dynamic treatment regime [32, 26] and machine learning [37, 15]. For example, the Q^*-function of a linear MDP is a linear function of features defined on $\boldsymbol{s}$ and a [37, 15, 14]. In different contexts, $\boldsymbol{s}$ can be high-dimensional, such as the feature vector constructed by basis functions [34, 26, 15] as an approximation to nonparametric Q^*-functions. The feature $\phi(\boldsymbol{s})$ can also be taken as a given representation learned from kernel methods or neural networks.

Accordingly, the task similarity under linear approximation (16) and (17) is defined by the ℓ_q-sparsity of $\boldsymbol{\delta}_t^{(k)}$ for $q \in [0, 1]$:

$$\max_{t \in [T], k \in [K]} \|\boldsymbol{\delta}_t^{(k)}\|_q \le h. \tag{18}$$

Let $\boldsymbol{W}_t^{(k)}$ be a $n_k \times p$ matrix whose i-th row is $\boldsymbol{w}(\boldsymbol{s}_{i,t}^{(k)}, a_{i,t}^{(k)})$. In the linear case, we employ the Trans-Lasso algorithm [23], a representative of transferred linear regression methods,

Algorithm 3: Transfer Q-learning algorithm (Linear)

Input : Target data $\{\{s_{t,i}^{(0)}, a_{t,i}^{(0)}, r_{t,i}^{(0)}\}_{t\in[T]}\}_{i\in[n_0]}$,
source data $\{\{\{s_{t,i}^{(k)}, a_{t,i}^{(k)}, r_{t,i}^{(k)}\}_{t\in[T]}\}_{i\in[n_k]}\}_{k=1}^{K}$, and
discount factor $\gamma \in [0,1]$.

1 Construct features $\boldsymbol{W}_t^{(k)}$ as a $n_k \times p$ matrix whose i-th row is $\boldsymbol{w}(s_{i,t}^{(k)}, a_{i,t}^{(k)})$ for $k \in \{0\} \cup [\mathcal{K}]$ where $\boldsymbol{w}(s,a) = [\boldsymbol{\phi}(s)^\top \mathbb{1}(a=1), \cdots, \boldsymbol{\phi}(s)^\top \mathbb{1}(a=M)]$.

2 Let $\widehat{\boldsymbol{\theta}}_{T+1}^{(0)} = 0$ to deal with pseudo-response construction at last stage T.

3 **for** $t = T, \ldots, 1$ **do**

4 Construct vectors of pseudo-response $\widehat{y}_{t,i}^{(tl-k)}$ (14) and $\widehat{y}_{t,i}^{(0)}$ (15) for the linear setting by

$$\widehat{\boldsymbol{y}}_t^{(tl-k)} = \boldsymbol{r}_t^{(k)} + \gamma \boldsymbol{W}_t^{(k)} \widehat{\boldsymbol{\theta}}_{t+1}^{(0)}, \quad k = 1, \ldots, K$$
$$\widehat{\boldsymbol{y}}_t^{(0)} = \boldsymbol{r}_t^{(0)} + \gamma \boldsymbol{W}_t^{(0)} \widehat{\boldsymbol{\theta}}_{t+1}^{(0)}.$$

5 Apply the transfer learning algorithm for supervised regression:

$$\widehat{\boldsymbol{b}}_t = \operatorname{argmin}_{\boldsymbol{w}\in\mathbb{R}^p} \left\{ \sum_{k=1}^{K} \|\widehat{\boldsymbol{y}}_t^{(tl-k)} - \boldsymbol{W}_t^{(k)} \boldsymbol{b}\|_2^2 + \lambda_{\mathrm{src}} \|\boldsymbol{b}\|_1 \right\}.$$
$$\widehat{\boldsymbol{\delta}}_t = \operatorname{argmin}_{\boldsymbol{\delta}\in\mathbb{R}^p} \left\{ \|\widehat{\boldsymbol{y}}_t^{(0)} - \boldsymbol{W}_t^{(0)} (\boldsymbol{\delta} + \widehat{\boldsymbol{b}}_t)\|_2^2 + \lambda_0 \|\boldsymbol{\delta}\|_1 \right\}.$$

Threshold $\widehat{\boldsymbol{\delta}}_t$ such that $(\check{\boldsymbol{\delta}}_t)_j = (\widehat{\boldsymbol{\delta}}_t)_j \mathbb{1}(|(\widehat{\boldsymbol{\delta}}_t)_j| \geq \lambda_0)$. Threshold $\widehat{\boldsymbol{b}}_t$ such that $(\check{\boldsymbol{b}}_t)_j = (\widehat{\boldsymbol{b}}_t)_j \mathbb{1}(|(\widehat{\boldsymbol{b}}_t)_j| \geq \lambda_{\mathrm{src}})$.

6 Calculate $\widehat{\boldsymbol{\theta}}_t^{(0)} = \check{\boldsymbol{b}}_t + \check{\boldsymbol{\delta}}_t$.

Output: $\{\widehat{\boldsymbol{\theta}}_t^{(0)}\}_{t=[T]}$.

owing to its simplicity and rate-optimal properties. Specifically, line 5 - 6 in Algorithm 1 is instantiated with:

$$\widehat{\boldsymbol{b}}_t = \operatorname{argmin}_{\boldsymbol{b}\in\mathbb{R}^p} \left\{ \sum_{k=1}^{K} \|\widehat{\boldsymbol{y}}_t^{(tl-k)} - \boldsymbol{W}_t^{(k)} \boldsymbol{b}\|_2^2 + \lambda_{\mathrm{src}} \|\boldsymbol{b}\|_1 \right\}.$$
$$\widehat{\boldsymbol{\delta}}_t = \operatorname{argmin}_{\boldsymbol{\delta}\in\mathbb{R}^p} \left\{ \|\widehat{\boldsymbol{y}}_t^{(0)} - \boldsymbol{W}_t^{(0)} (\boldsymbol{\delta} + \widehat{\boldsymbol{b}}_t)\|_2^2 + \lambda_0 \|\boldsymbol{\delta}\|_1 \right\}.$$

We threshold $\widehat{\boldsymbol{\delta}}_t$ such that $(\check{\boldsymbol{\delta}}_t)_j = (\widehat{\boldsymbol{\delta}}_t)_j \mathbb{1}(|(\widehat{\boldsymbol{\delta}}_t)_j| \geq \lambda_0)$ and threshold $\widehat{\boldsymbol{b}}_t$ such that $(\check{\boldsymbol{b}}_t)_j = (\widehat{\boldsymbol{b}}_t)_j \mathbb{1}(|(\widehat{\boldsymbol{b}}_t)_j| \geq \lambda_{\mathrm{src}})$. The final output is $\widehat{\boldsymbol{\theta}}_t^{(0)} = \check{\boldsymbol{b}}_t + \check{\boldsymbol{\delta}}_t$ and $\widehat{Q}_t^{(0)}(s,a) = [\boldsymbol{\phi}(s)^\top \mathbb{1}(a = 1), \cdots, \boldsymbol{\phi}(s)^\top \mathbb{1}(a = M)] \widehat{\boldsymbol{\theta}}_t^{(0)}$ for all stage $t \in [T]$.

4.1. Theoretical guarantees under linear function approximation

In this section, we provide rigorous error and regret analyses of both offline and online transfer Q-learning under the similarity characterization (18). Under such a characterization, the parameter space we consider is

$$\Omega_q(s,h) = \left\{ \{\boldsymbol{\theta}_t^{(0)}, \{\boldsymbol{\delta}_t^{(k)}\}_{k\in[K]}\}_{t\in[T]} : \max_{t\in[T]} \|\boldsymbol{\theta}_t^{(0)}\|_0 \leq s, \max_{t\in[T], k\in[K]} \|\boldsymbol{\delta}_t^{(k)}\|_q \leq h \right\}.$$

We write $N_{\mathrm{src}} = \sum_{k=1}^{K} n_k$ as the total number of trajectories from the source tasks.

Assumption 4.1 (Conditions on the design). For $k = 0, \ldots, K$ and $t = 1, \ldots, T$, $\boldsymbol{s}_{t,i}^{(k)}$ are sub-Gaussian with mean zero and independent across tasks indexed by k and across trajectories indexed by i. Let $(\boldsymbol{w}_{t,i}^{(k)})^\top = [\boldsymbol{\phi}(\boldsymbol{s}_{t,i}^{(k)})^\top \mathbb{1}(a_{t,i}^{(k)} = 1), \cdots, \boldsymbol{\phi}(\boldsymbol{s}_{t,i}^{(k)})^\top \mathbb{1}(a_{t,i}^{(k)} = M)]$ for some known basis functions $\boldsymbol{\phi}(\cdot)$ and denote the dimension of $\boldsymbol{w}_{t,i}^{(k)}$ as p. The covariance matrices $\boldsymbol{\Sigma}_t^{(k)} = \mathbb{E}[\boldsymbol{w}_{t,i}^{(k)}(\boldsymbol{w}_{t,i}^{(k)})^\top]$ are positive definite with bounded eigenvalues, $t = 1, \ldots, T$, $k = 1, \ldots, K$. Moreover,

$$\max_{t\in[T]} \max_{k\in[K]} \left\| \boldsymbol{\Sigma}_t^{(k)} (\boldsymbol{\Sigma}_t^{(0)})^{-1} - I_{2p} \right\|_{\infty,1} \le C_\Sigma < 1. \tag{19}$$

Assumption 4.1 requires sub-Gaussian designs for all the tasks. The positive definiteness of $\boldsymbol{\Sigma}_t^{(k)}$ assumes regularity of the covariance of $\boldsymbol{s}_{t,i}^{(k)}$ but also requires $a_{t,i}^{(k)}$ has constant variance across samples. Moreover, we require in (19) that the covariance matrices for the source and target tasks are sufficiently similar.

4.1.1. Convergence rate for the linear setting

We provide theoretical guarantees for the offline-to-offline transfer in this subsection. Theorem 4.2 establishes the convergence rate for $\{\widehat{\boldsymbol{\theta}}_t^{(0)}\}_{t\in[T]}$ in this setting.

Theorem 4.2 (Convergence rate of Algorithm 3)**.** *Suppose that Assumptions 4.1 hold and* (18) *holds with* $q = 1$. *Let* N_{src} *be the total number of samples in source tasks. We take the tuning parameters to be*

$$\lambda_{\text{src}} = c_1\sqrt{\frac{\log p}{N_{\text{src}}}} + c_1\sqrt{\frac{h}{s}}\left(\frac{\log p}{n_0}\right)^{1/4} \text{ and } \lambda_0 = c_1\sqrt{\frac{\log p}{n_0}}.$$

Under the sample size condition that $s\sqrt{\log p/N_{\text{src}}} + h + sh\sqrt{\log p/n_0} \le C$, *for any* $t = 1, \ldots, T$, *we have*

$$\frac{1}{n_0}\left\|\boldsymbol{W}_t^{(0)}(\widehat{\boldsymbol{\theta}}_t^{(0)} - \boldsymbol{\theta}_t^{(0)})\right\|_2^2 \vee \left\|\widehat{\boldsymbol{\theta}}_t^{(0)} - \boldsymbol{\theta}_t^{(0)}\right\|_2^2 \lesssim \left\|\widehat{\boldsymbol{\theta}}_{t+1}^{(0)} - \boldsymbol{\theta}_{t+1}^{(0)}\right\|_2^2 + \frac{s\log p}{N_{\text{src}}} + h\sqrt{\frac{\log p}{n_0}}.$$

with probability at least $1 - \exp(-c_2 \log p)$.

Theorem 4.2 establishes the estimation accuracy of the target Q^*-function at each stage. The rate for the final stage ($t = T$) is the minimax optimal rate in supervised regression with transfer learning [23] since the final-stage reward equals the value of the state-action pair. For the t-th stage ($t < T$), the estimation error from $\widehat{\boldsymbol{\theta}}_{t+1}^{(0)} - \boldsymbol{\theta}_{t+1}^{(0)}$ accumulates, as a consequence of using pseudo-responses. Thus, empirically one may expect that the estimation errors become larger in the earlier stages, even though the convergence rate of each stage are the same. In the upper bound, the sparsity s of the target parameter is weighted with $1/N_{\text{src}}$ and the maximum sparsity h of the contrast vectors is weighted with $1/\sqrt{n_0}$. This shows the improvement of transfer learning, where the s-sparse component is learned based on all the studies but the task-specific contrast vectors can only be identified based on the target study with n_0 samples. When the similarity is sufficiently high, then $h \ll s\sqrt{\log p/n_0}$ and transfer learning can lead to improvements in this case. In the theorem, we our choice of λ_{src} depends on the unknown h/s, which is for a convenient proof of desirable convergence rate. In practice, λ_{src} can be chosen by cross-validation.

Remark 2 (Convergence rate of single-task Q-learning). Let $\widehat{\boldsymbol{\theta}}_t^{(st)}$, $t \in [T]$, be the single-task Q-learning estimator studied in [32]. For any $t = 1, \ldots, T$,

$$\|\widehat{\boldsymbol{\theta}}_t^{(st)} - \boldsymbol{\theta}_t^{(0)}\|_2^2 \lesssim \|\widehat{\boldsymbol{\theta}}_{t+1}^{(st)} - \boldsymbol{\theta}_{t+1}^{(0)}\|_2^2 + \frac{s \log p}{n_0}.$$

For the final stage, the transfer learning estimator $\widehat{\boldsymbol{\theta}}_T^{(0)}$ has faster convergence rate than the single-task estimator $\widehat{\boldsymbol{\theta}}_t^{(st)}$ as long as $N_{\text{src}} \gg n_0$ and $h \ll s\sqrt{\log p / n_0}$. Furthermore, the convergence rate of $\widehat{\boldsymbol{\theta}}_T^{(0)}$ is minimax optimal in the parameter space of interest. For the t-th stage for $t < T$, the gain of transfer learning has two aspects. The first is the error inherited from the next stage, $\|\widehat{\boldsymbol{\theta}}_{t+1}^{(0)} - \boldsymbol{\theta}_{t+1}^{(0)}\|_2^2$, which is smaller than its single-task counterpart $\|\widehat{\boldsymbol{\theta}}_{t+1}^{(st)} - \boldsymbol{\theta}_{t+1}^{(0)}\|_2^2$. The second gain comes from aggregating the data at the current stage.

Remark 3 (Similarity characterization with $q \in [0, 1)$). In Theorem 4.2, we require $\boldsymbol{\delta}_t^{(k)}$ to be approximately sparse ($q = 1$). For $q \in [0, 1)$, transferred Q-learning algorithms can be analogously developed based on Algorithm 1 in the supplement of [23], which is a minimax optimal approach for transfer learning in linear models for $q \in [0, 1)$.

Remark 4 (Limitations under transition shift). Our theoretical guarantees rely on stage-wise reward similarity together with mild design similarity across tasks. If transition dynamics differ substantially, thereby violating the covariance-similarity condition, or if the reward discrepancies h are large, the advantages of transfer learning may diminish or even disappear. These extensions call for new estimation tools and theoretical analysis, but they provide a principled roadmap for expanding our framework beyond the reward-similarity regime studied here. A full development of these methods is beyond the scope of this paper and will be pursued in future work.

4.1.2. Regret bound of offline-to-online transfer Q-learning

In this subsection, we provide theoretical guarantees for the online Algorithm 2 with knowledge transferred from offline data for $M = 2$. Results for a finite M can be derived similarly, with the final outcomes differing only by a constant factor dependent on M. In the online setting, the learner aims to minimize the cumulative regret that measures the expected loss of following the estimated optimal policy instead of the oracle optimal policy. Mathematically, the cumulative regret over T episodes is defined as

$$\text{Regret}_{NT} = \sum_{i=1}^{N} \sum_{t=1}^{T} \gamma^t \left(\mathbb{E}[r_{t,i}^{(0)} | \boldsymbol{s}_{t,i}^{(0)}, a_{t,i}^*] - \mathbb{E}[r_{t,i}^{(0)} | \boldsymbol{s}_{t,i}^{(0)}, \widehat{a}_{t,i}^{(0)}] \right), \tag{20}$$

where the estimated optimal policy is $\widehat{a}_{t,i}^{(0)} = \arg\max_{a' \in \{-1,1\}} Q(\boldsymbol{s}_{t,i}^{(0)}, a'; \widehat{\boldsymbol{\theta}}_t^{(0)})$ and the oracle optimal policy is $a_{t,i}^* = \arg\max_{a' \in \{-1,1\}} Q(\boldsymbol{s}_{t,i}^{(0)}, a'; \boldsymbol{\theta}_t^{(0)})$. Under the linear Q^*-function (16) with $M = 2$, they can be further simplified to

$$\widehat{a}_{t,i}^{(0)} = sgn((\boldsymbol{s}_{t,i}^{(0)})^\top \widehat{\boldsymbol{\psi}}_t), \quad \text{and} \quad a_{t,i}^* = sgn((\boldsymbol{s}_{t,i}^{(0)})^\top \boldsymbol{\psi}_t),$$

where $sgn(\cdot)$ is the sign function. The regret bound of online learning with offline transfer is given in Theorem 4.3.

Theorem 4.3 (Cumulative regret of Algorithm 2 after T rounds)**.** *Suppose that Assumptions 4.1 hold, (18) holds with $q = 1$. and $s\sqrt{\log p/N_{\text{src}}} + h \le C$, $n_e \gtrsim s^2h^2 \log p + \log p$. We take tuning parameters to be*

$$\lambda_{\text{src}} = \sqrt{\frac{\log p}{N_{\text{src}}}} + \sqrt{\frac{h}{s}}\left(\frac{\log p}{n_e}\right)^{1/4} \quad \text{and} \quad \lambda_0 = \sqrt{\frac{\log p}{n_e}}.$$

For any $N > n_e$,

$$\text{Regret}_{NT} \lesssim \frac{n_e\gamma}{1-\gamma} + (N - n_e)\left(\sqrt{\frac{s\log p}{N_{\text{src}}}} + h^{1/2}\left(\frac{\log p}{n_e}\right)^{1/4}\right) \tag{21}$$

with probability at least $1 - \exp(-c_1 \log p)$.

We now find the optimal choice of n_e, which minimizes the RHS of (21). To simplify the analysis, we parameterize $h = N_{\text{src}}^{-\alpha}$ for some $\alpha \ge 0$. In the supplementary files [7], we show that if we take

$$n_e \asymp \max\left\{\frac{N^{4/5}(\log p)^{1/5}}{N_{\text{src}}^{2\alpha/5}}, \frac{s^2 \log p}{N_{\text{src}}^{2\alpha}}\right\},$$

then with probability at least $1 - \exp(-c_1 \log p)$

$$\text{Regret}_{NT} \lesssim \max\left\{\frac{N^{4/5}(\log p)^{1/5}}{N_{\text{src}}^{2\alpha/5}}, \frac{s^2 \log p}{N_{\text{src}}^{2\alpha}}\right\} + N\sqrt{\frac{s\log p}{N_{\text{src}}}}. \tag{22}$$

We see that the larger the N_{src}, i.e., more source data, the smaller the cumulative regret; and the larger the value of α, i.e., higher the similarity, the smaller the cumulative regret.

Remark 5 (Cumulative regret of single-task Q-learning). Without offline data, we denote the estimated optimal policy by $\widehat{a}^{(st)}$. It is easy to show that for $n_e \gg (s\log p)^2$, then with probability at least $1 - \exp(-c_1 \log p)$,

$$\text{Regret}_{NT}^{(st)} \lesssim \frac{n_e\gamma}{1-\gamma} + (N - n_e)\sqrt{\frac{s\log p}{n_e}}. \tag{23}$$

with probability at least $1 - \exp(-c_1 \log p)$. If we take $n_e \asymp N^{2/3}(s\log p)^{1/3} \vee (s\log p)^2$, then

$$\text{Regret}_{NT}^{(st)} \lesssim N^{2/3}(s\log p)^{1/3} + (s\log p)^2, \tag{24}$$

with probability at least $1 - \exp(-c_1 \log p)$.

Comparing (21) with (23), we see that the cumulative regret of transfer Q-learning is always smaller when $h = o(s\sqrt{\log p/n_e})$. Comparing (22) with (24), we see that as long as $N_{\text{src}} > N^{1/(3\alpha)} + N^{2/3}$, the regret of transfer learning policy is always no larger than the single-task policy. This comparison further implies that if N is very large, then the optimal choice of n_e can be larger than N_{src}. In this case, transfer learning may not yield a significant improvement and it suffices to consider single-task Q-learning.

5. Numerical studies

In this section, we demonstrate the advantages of the proposed transferred Q-learning algorithm on synthetic and real data sets. Code and implementation details for this section are available online.

5.1. Two-stage MDP with analytical optimal Q^* function

We first consider a simple MDP model which has an analytical form for the optimal Q^* function. In such a setting we can explicitly compare the estimated Q^* function with the ground truth. The generative model is designed based on those in [5] and [32]. The underlying model is a two-stage MDP with two possible actions $\mathcal{A} = \{-1, 1\}$ and two states $\mathcal{S} = \{-1, 1\}$. The binary states S_t and the binary treatment A_t are generated as follows:

$$\begin{aligned}
&\Pr(S_1 = -1) = \Pr(S_0 = 1) = 0.5,\\
&\Pr(A_t = -1) = \Pr(A_t = 1) = 0.5, \quad t = 1, 2,\\
&\Pr(S_2|S_1, A_1) = 1 - \Pr(S_2|S_1, A_1) = \operatorname{expit}(b_1 S_1 + b_2 A_1),
\end{aligned}$$

where $\operatorname{expit}(x) = \exp(x)/(1 + \exp(x))$. The immediate reward $R_1 = 0$ and R_2 is given by

$$R_2 = \kappa_1 + \kappa_2 S_1 + \kappa_3 A_1 + \kappa_4 S_1 A_1 + \kappa_5 A_2 + \kappa_6 S_2 A_2 + \kappa_7 A_1 A_2 + \varepsilon_2,$$

where $\varepsilon_2 \sim \mathcal{N}(0, 1)$. Under this setting, the true Q^* functions for time $t = 1, 2$ are

$$\begin{aligned}
Q_2(S_2, A_2; \boldsymbol{\theta}_2) &= \theta_{2,1} + \theta_{2,2} S_1 + \theta_{2,3} A_1 + \theta_{2,4} S_1 A_1\\
&\quad + \theta_{2,5} A_2 + \theta_{2,6} S_2 A_2 + \theta_{2,7} A_1 A_2\\
Q_1(S_1, A_1; \boldsymbol{\theta}_1) &= \theta_{1,1} + \theta_{1,2} S_1 + \theta_{1,3} A_1 + \theta_{1,4} S_1 A_1,
\end{aligned} \tag{25}$$

where the true coefficients $\boldsymbol{\theta}_t$ are explicitly functions of b_1, b_2, $\kappa_1, \ldots, \kappa_7$ given in (6) in the supplemental material. At each state, the observed covariates $X_t \in \mathbb{R}^p$, $p = 100$, consist of first eight elements 1, S_1, A_1, $S_1 A_1$, A_2, S_2, $S_2 A_2$, $A_1 A_2$ and the remaining elements that are randomly sampled from standard normal.

We consider transfer between two different but similar MDPs from the above model. The target and source MDPs are different in the coefficients κ's and therefore θ's in (25). Specifically, we set $\theta_{2,j} = 1$, $1 \le j \le 7$ for the target MDP. The second-stage coefficients of the source task are the same except that the second element $\theta^{(1)}_{2,2} = 1.2$. Therefore, according to equation (6) in the supplemental material, the true coefficients of stage-one Q^* functions are $\theta_{1,1}, \theta_{1,2}, \theta_{1,3}, \theta_{1,4} \approx 2.69, 1.19, 1.69, 1.19$ for the target MDP, and $\theta^{(1)}_{1,1}, \theta^{(1)}_{1,2}, \theta^{(1)}_{1,3}, \theta^{(1)}_{1,4} \approx 2.69, 1.39, 1.69, 1.19$ for the source MDP. In summary, the true Q_1 functions of the target and source MDPs are different only in $\theta_{1,2}$ and the Q_2 functions are different only in $\theta_{2,2}$. The coefficients of the remaining elements in X_t are all set to zero. We generate trajectories of the form $(\boldsymbol{x}_{1,i}, a_{1,i}, r_{1,i}, \boldsymbol{x}_{2,i}, a_{2,i}, r_{2,i})$ from both target and source MDPs. The target task consists of n_0 trajectories while the source task consists of n_1 trajectories.

We compare $\left\|\widehat{\boldsymbol{\theta}}_t - \boldsymbol{\theta}_t\right\|_2^2$, $t = 1, 2$, with or without transfer under combinations of n_0 and $n_{\text{src}} = n_1$. The boxplots are shown in Figure 2. We also generate a new dataset and compare predicted $\widehat{Q}^*(\boldsymbol{x}, a)$ obtained by transferred Q^* learning and its vanilla counterpart. The boxplots of $\left|\widehat{Q}^*(\boldsymbol{x}, a) - Q^*(\boldsymbol{x}, a)\right| / |Q^*(\boldsymbol{x}, a)|$ (averaged over all state-action pairs in the new dataset) are presented in Figure 3. It is clear that transfer Q-learning performs much better than the vanilla Q-learning without transfer in terms of both coefficient estimation and prediction. The advantage is more prominent in earlier stages since the transfer benefit in the latter stage positively cascade to the earlier stages through the second term in (12).

Table 1 compares the frequency of correct optimal actions chosen by single-task Q-learning and transfer Q-learning with the new dataeset. We observe that transfer Q-learning achieves

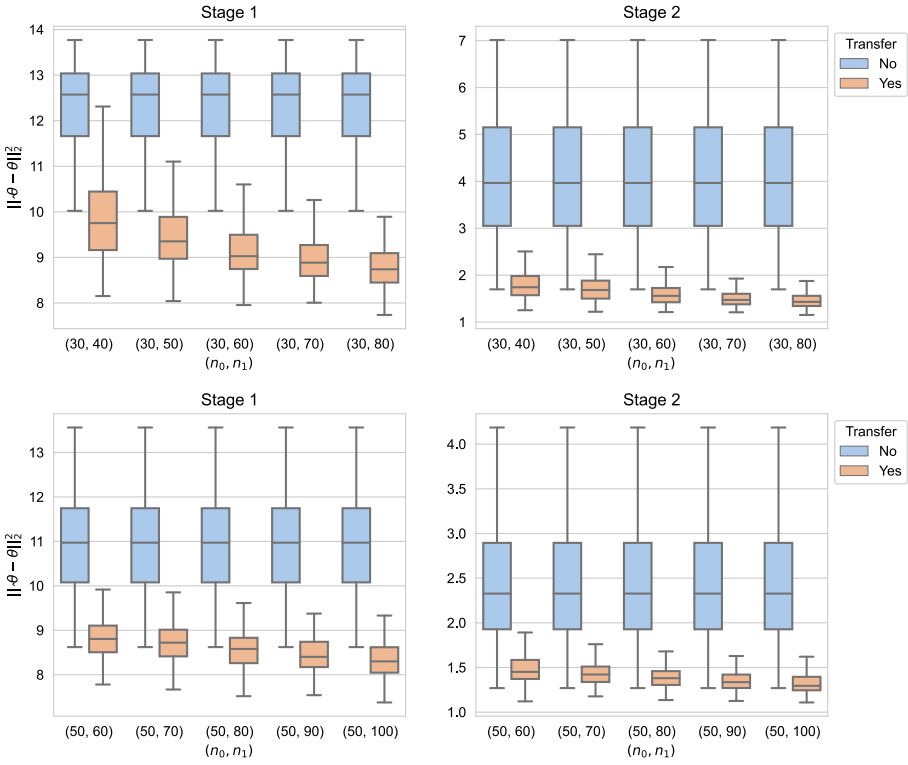


Fig 2. *Comparison of the estimated coefficients of the optimal Q^* function. The y-axis represents $\left\|\widehat{\boldsymbol{\theta}}_t - \boldsymbol{\theta}_t\right\|_2^2$ for $t = 1$ (left) and $t = 2$ (right). The dimension is $p = 100$ and sparsity $s = 7$.*

higher accuracy in choosing the optimal actions across all combinations of n_0 and n_1, which are the number of trajectories of the target task and the source task, respectively. The amplitude of accuracy increase is higher when n_0 is small. This again shows the advantage of transfer Q-learning, especially when the number of trajectories of the target task is small.

5.2. *Offline-to-online transfer*

In this section, we study the empirical performance of online transfer Q-learning using an offline source dataset. The generative MDP is the same as that defined in (25). We have access to an offline source dataset of trajectories $(X_1, A_1, R_1, X_2, A_2, R_2)$ generated from an MDP with $\theta_{2j}^{(1)} = \kappa_j^{(1)} = 1$, $1 \le j \le 3$, $5 \le j \le 7$ and $\theta_{2j}^{(1)} = \kappa_j^{(1)} = 2$, $j = 4$. The online target RL task is modeled by an MDP with $\theta_{2j}^{(1)} = \kappa_j^{(1)} = 1$, $1 \le j \le 7$. The dimension of covariates is $p = 100$.

We first study the cumulative regret of transferred and vanilla Q-learning with different lengths of exploration, $n_e \in \{1, 2, \cdots, 20\}$. The size of the offline source dataset is $n_1 = 100$. At the exploration stage, n_e trajectories are generated by random actions. The coefficients $\widehat{\boldsymbol{\theta}}_t$ are estimated using Q-learning with or without transfer. The average cumulative regret at the exploitation stage versus different length of exploration are presented in Figure 4 (a). The length of exploitation stage is 100. For each n_e, the mean of the cumulative regret at exploitation stage is reported since the values of $\widehat{\boldsymbol{\theta}}_t^{(0)}$ are not updated during exploitation

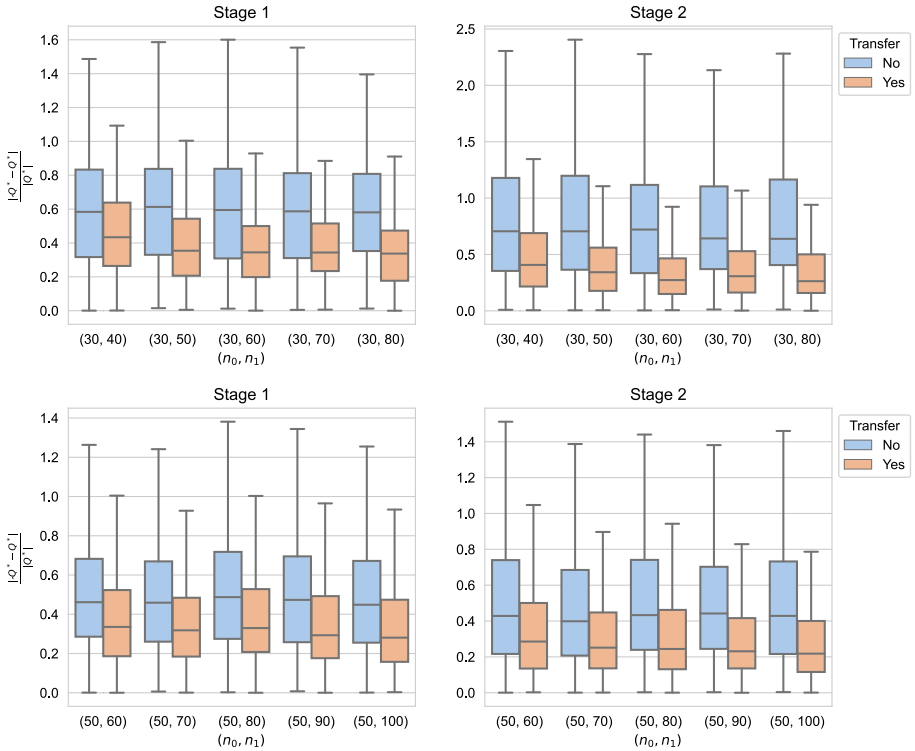


Fig 3. *Comparison of the prediction of the optimal Q^* function with different sample size configurations. The y-axis represents $\left|\widehat{Q}^* - Q^*\right|/|Q^*|$. The dimension is $p = 100$ and sparsity $s = 7$.*

Table 1

The frequency of correct optimal actions. The comparison is between the optimal action $A^{*,(st)}$ chosen by single-task Q-learning and the optimal action $A^{*,(tl)}$ chosen by transfer Q-learning. We consider different combinations of $n_0 = 30, 50, 70$ and $n_1 = n_0 + 10, \cdots, n_0 + 50$, where n_0 and n_1 denote the number of trajectories of the target task and the source task, respectively

n_0	$A^{*,(st)}$	$A^{*,(tl)}$ with different n_1				
		$n_0 + 10$	$n_0 + 20$	$n_0 + 30$	$n_0 + 40$	$n_0 + 50$
30	0.555	0.965	0.985	0.990	0.970	0.985
50	0.860	0.990	0.995	1.000	1.000	1.000
70	0.905	1	1	1	1	1

stage. We observe that under the condition that $h \ll s\sqrt{\log p/n_e}$, the regret of transfer Q-learning is much smaller than that of vanilla Q-learning, which is consistent with the result in Theorem 4.3. Since Algorithm 2 is of the *explore-then-commit* (ETC) type, the advantage of transferred Q^* learning shown in the left panel of Figure 4 can be viewed as the jumpstart improvement—one of the three main objectives of transfer learning defined in [20] and [35].

We also empirically study the cumulative regret of online Q-learning where the values of $\widehat{\boldsymbol{\theta}}_t$, $t = 1, 2$ are updated during the exploitation stage. This phase-based ETC online Q-learning algorithm is a natural extension of Algorithm 2 and it goes as follows. At the first phase, vanilla Q-learning initializes $\widehat{\boldsymbol{\theta}}_t$ to zero, while transfer Q-learning initializes with $\widehat{\boldsymbol{\theta}}_t$ that are estimated using offline trajectories from the source task. Then at each phase, a batch of 100 trajectories are generated using greedy actions based on estimated $\widehat{\boldsymbol{\theta}}_t$. Using the extra

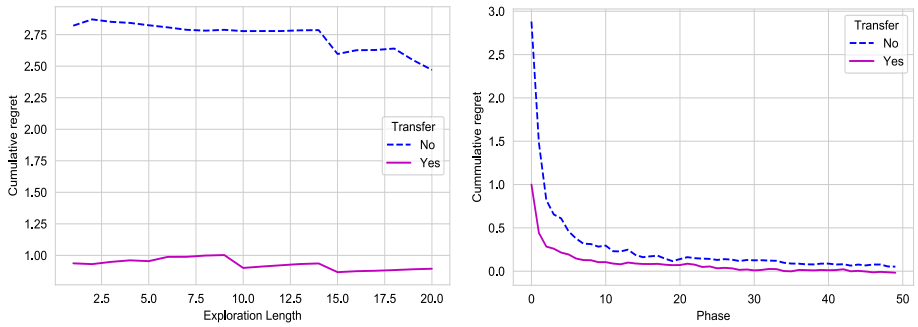


Fig 4. *Cumulative regret of online Q-learning with or without transfer. Left panel: The* explore-then-commit *(ETC) online Q-learning; Right panel: phase-based ETC online Q-learning with parameter updates.*

generated trajectories at the current phase, the values $\widehat{\boldsymbol{\theta}}_t$ are updated and will be used to generate greedy actions for the next phase. The right panel of Figure 4 shows the mean of cumulative regret as phases proceed online. It shows all three main advantages of transferred learning, namely, jumpstart, learning speed, and asymptotic improvements [20, 35].

5.3. Medical data application: MIMIC-III *sepsis cohort*

In this section, we illustrate an application of the transfer Q-learning in the Medical Information Mart for Intensive Care version III (MIMIC-III) Database [16], which is a freely available source of de-identified critical care data from 2001 – 2012 in six ICUs at a Boston teaching hospital.

We consider a cohort of sepsis patients, following the same data processing procedure as that in [8, 19]. Each patient in the cohort is characterized by a set of 47 variables, including demographics, Elixhauser premorbid status, vital signs, and laboratory values. Patients' data were coded as multidimensional discrete time series $\boldsymbol{x}_{i,t} \in \mathbb{R}^{47}$ for $1 \leq i \leq N$ and $1 \leq t \leq T_i$ with 4-hour time steps. The actions of interests are the total volume of intravenous (IV) fluids and maximum dose of vasopressors administrated over each 4-hour period. We discretize two actions into three levels (i.e., low, medium, and high), respectively. In our setting, the low-level corresponds to level 1 - 2, the medium-level corresponds to level 3 and the high-level corresponds to level 4 - 5 in [19]. The combination of the two drugs makes $M = 3 \times 3 = 9$ possible actions in total. The final processed dataset contains 20943 unique adult ICU admissions, among which 11704 (55.88%) are female (0) and 9239 (44.11%) are male (1).

The reward signal is important and is crafted carefully in real applications. For the final reward, we follow [19] and use hospital mortality or 90-day mortality. Specifically, when a patient survived after 90 days out of hospital, a positive reward was released at the end of each patient's trajectory; a negative reward was issued if the patient died in hospital or within 90 days out of hospital. In our dataset, the mortality rate is 24.21% for female and 22.71% for male. For the intermediate rewards, we follow [29] and associates reward to the health measurement of a patient. The detailed description of the data pre-processing is presented in Section 5 of the supplemental material.

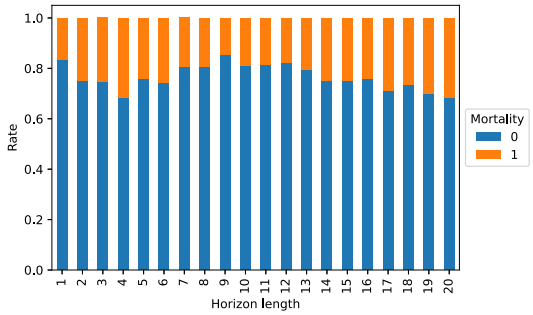


Fig 5. *Mortality rates across different horizon lengths.*

The trajectory horizons are different in the dataset, with the maximum being 20 and minimum being 1. We use 20 as the total length of horizon. The trajectories are aligned at the last steps while allowing the starting steps vary. For examples, the trajectories with length 20 start at step 1 while the trajectories with length 10 start at step 11. But they all end at step 20. We adopt this method because the distribution of final status are similar across trajectories. Figure 5 presents the mortality rates of different lengths. We see that while the numbers of trajectories differ a lot, the mortality rates do not vary much across trajectories with different horizons. On the contrary, the starting status of patients may be very different. The one with trajectory length 20 may be in a worse status and needs 10 steps to reach the status similar to the starting status of the one with length 10. We believe this is a reasonable setup to illustrate our method. A rigorous medical analysis is beyond the scope of this paper and is a worthwhile topic for future research.

We consider transfer Q-learning across genders. The analytical model for optimal $Q_t^{(k)}$ function for each step $t \in [20]$ is modeled by

$$Q_t^{(k)}(\boldsymbol{x}, a) \approx \boldsymbol{x}^\top \sum_{a'=1}^{9} \boldsymbol{\theta}_{t,a}^{(k)} \mathbf{1}(a = a'), \tag{26}$$

where the covariates $\boldsymbol{x}$ contains 44 variables detailed in Table 1 in the supplements. Even though the total number of trajectories of gender 0 is large, estimating (26) is still a high-dimensional problem since we allow $\boldsymbol{\theta}_{t,a}^{(k)}$ be different across step t, action a and gender subgroup $k = 0, 1$ and the number of trajectories corresponding to a specific combination of t, a, and g is small. For example, there are only 19 samples available to estimate $\boldsymbol{\theta}_{t,a}^{(k)}$ for gender $k = 1$, step $t = 1$, and action a corresponding to the combination (IV, Vaso) $= (0, 1)$. Table 3 in the supplemental material shows that the least ten samples sizes are all under 30. We observe that gender $k = 1$ (male) has fewer samples so we consider transfer Q-learning with target task for gender $k = 1$ and auxiliary source task from gender $k = 0$.

The estimation procedure of transfer Q-learning follows Algorithm 1. We set the discount factor as $\gamma = 0.98$. We also estimate the Q^* function by the vanilla Q-learning which is the counterpart of Algorithm 1 without transfer. The Lasso tuning parameters are chosen according to Theorem 4.2 and by a linear search for the value of c_1 that maximizes the objective function. We calculate the optimal aggregated values of transferred and Vanilla Q-learning, denoted by $V^{*,(tl)}$ and $V^{*,(st)}$, respectively. Figure 6 plot the average of the ratio $V^{*,(tl)}/V^{*,(st)}$ and its standard deviation. The mean ratio is above one, indicating that the optimal value of transfer Q-learning is larger.

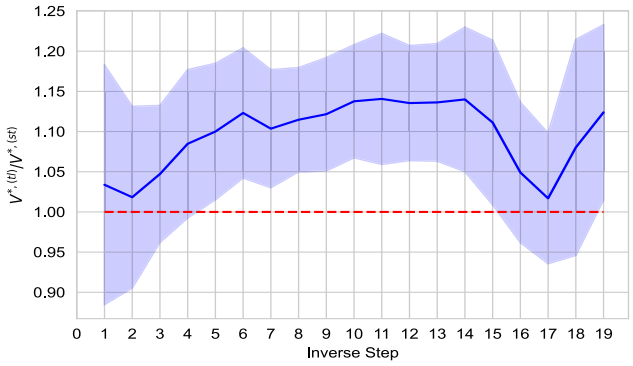


Fig 6. *Ratio between optimal aggregated values V^* of transfer Q-learning ($V^{*,(tl)}$) and those of Vanilla Q-learning ($V^{*,(st)}$) applied on the target data. Shaded area covers one standard deviation from the mean. Inverse step is $20 - t$ where $1 \leq t \leq 20$ is the natural step in a trajectory.*

6. Conclusions

In this work, we envisage Q-learning with knowledge transfer in the form of an algorithm that exploits the availability of samples from different but similar RL tasks. The *similarity* between target and source RL tasks is characterized by their corresponding reward functions which can be checked directly in practice. We note that a salient feature of RL is its multi-stage learning and, accordingly, we design a novel *re-targeting* step to enable "cross-stage transfer" along multiple stages in an RL task, in addition to the usual "cross-task transfer" in TL for supervised learning. For both offline and online RL tasks, our approach demonstrates improvements over single-task Q-learning, both theoretically and empirically.

The similarity definition among multi-stage RL tasks can be diverse and more flexible. For instance, some related tasks may have different action spaces or different number of stages. It is also important to study transfer learning for such situations. As the similarity level is always unknown, it is likely that we include some source tasks which are far away from the target task. As a future research direction, one can perform a model selection aggregation step and aggregation in multi-stage models, especially the online setting, is largely an open problem and requires further study.

Acknowledgments

The authors would like to thank the anonymous referees, an Associate Editor and the Editor for their constructive comments that improved the quality of this paper.

Funding

Elynn Chen was supported by NSF Grant DMS-2412577.

Supplementary Material

Supplementary Material of "Transfer Q-Learning for Finite-Horizon Markov Decision Processes"

(doi: 10.1214/25-EJS2459SUPP; .pdf). The supplementary material provides detailed mathematical proofs for the main theoretical results, including the convergence rate and regret bounds, together with additional lemmas and intermediate derivations. It also contains implementation details and extended results from the numerical studies, such as full simulation settings, additional tables, and figures. For the real-data application on the MIMIC-III sepsis cohort, the supplement describes the data preprocessing procedures, variable definitions, reward construction, and cohort statistics. These materials ensure full reproducibility and transparency of the theoretical and empirical findings reported in the main paper.

References

[1] Bastani, H. (2021). Predicting with proxies: Transfer learning in high dimension. *Management Science* **67** 2964–2984.

[2] Cai, T. T. and Wei, H. (2021). Transfer learning for nonparametric classification: Minimax rate and adaptive classifier. *The Annals of Statistics* **49** 100–128. MR4206671

[3] Chai, J., Chen, E. and Fan, J. (2025). Deep Transfer Q-Learning for Offline Non-Stationary Reinforcement Learning. *arXiv preprint arXiv:2501.04870*.

[4] Chai, J., Chen, E. and Yang, L. (2025). Transfer Q-Learning with Composite MDP Structures. In *The Forty-second International Conference on Machine Learning (ICML 2025)*.

[5] Chakraborty, B., Murphy, S. and Strecher, V. (2010). Inference for non-regular parameters in optimal dynamic treatment regimes. *Statistical Methods in Medical Research* **19** 317–343. MR2757118

[6] Chen, E., Chen, X. and Jing, W. (2024). Data-Driven Knowledge Transfer in Batch Q-Learning. *arXiv preprint arXiv:2404.15209, 2024*.

[7] Chen, E., Li, S. and Jordan, M. I. (2025). Supplement to "Transfer Q-learning for finite-horizon Markov decision processes". https://doi.org/10.1214/25-EJS2459SUPP.

[8] Chen, E., Song, R. and Jordan, M. I. (2022). Reinforcement Learning in Latent Heterogeneous Environments. *arXiv preprint arXiv:2202.00088*. MR4833941

[9] Clifton, J. and Laber, E. (2020). Q-learning: Theory and applications. *Annual Review of Statistics and Its Application* **7** 279–301. MR4104194

[10] Ertefaie, A., McKay, J. R., Oslin, D. and Strawderman, R. L. (2021). Robust Q-learning. *Journal of the American Statistical Association* **116** 368–381. MR4227700

[11] Ertefaie, A. and Strawderman, R. L. (2018). Constructing dynamic treatment regimes over indefinite time horizons. *Biometrika* **105** 963–977. MR3877877

[12] Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B. and Smola, A. (2012). A kernel two-sample test. *The journal of machine learning research* **13** 723–773. MR2913716

[13] Gu, T., Han, Y. and Duan, R. (2022). Robust angle-based transfer learning in high dimensions. *arXiv preprint arXiv:2210.12759*.

[14] Hao, B., Lattimore, T., Szepesvári, C. and Wang, M. (2021). Online Sparse Reinforcement Learning. In *International Conference on Artificial Intelligence and Statistics* 316–324. PMLR.

[15] JIN, C., YANG, Z., WANG, Z. and JORDAN, M. I. (2020). Provably efficient reinforcement learning with linear function approximation. In *Conference on Learning Theory* 2137–2143.

[16] JOHNSON, A. E., POLLARD, T. J., SHEN, L., LI-WEI, H. L., FENG, M., GHASSEMI, M., MOODY, B., SZOLOVITS, P., CELI, L. A. and MARK, R. G. (2016). MIMIC-III, a freely accessible critical care database. *Scientific Data* **3** 160035.

[17] KALLUS, N. (2020). More efficient policy learning via optimal retargeting. *Journal of the American Statistical Association* 1–13. MR4270011

[18] KOLM, P. N. and RITTER, G. (2020). Modern perspectives on reinforcement learning in finance. *Modern Perspectives on Reinforcement Learning in Finance (September 6, 2019). The Journal of Machine Learning in Finance* **1**.

[19] KOMOROWSKI, M., CELI, L. A., BADAWI, O., GORDON, A. C. and FAISAL, A. A. (2018). The artificial intelligence clinician learns optimal treatment strategies for sepsis in intensive care. *Nature Medicine* **24** 1716–1720.

[20] LANGLEY, P. (2006). Transfer of knowledge in cognitive systems. In *Talk, workshop on Structural Knowledge Transfer for Machine Learning at the Twenty-Third International Conference on Machine Learning*.

[21] LAZARIC, A. (2012). Transfer in reinforcement learning: A framework and a survey. In *Reinforcement Learning* 143–173. Springer.

[22] LEVINE, S., KUMAR, A., TUCKER, G. and FU, J. (2020). Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*.

[23] LI, S., CAI, T. T. and LI, H. (2022). Transfer learning for high-dimensional linear regression: Prediction, estimation and minimax optimality. *Journal of the Royal Statistical Society Series B: Statistical Methodology* **84** 149–173. MR4400393

[24] LI, S., CAI, T. T. and LI, H. (2022). Transfer learning in large-scale gaussian graphical models with false discovery rate control. *Journal of the American Statistical Association* 1–13. MR4646634

[25] LI, S., ZHANG, L., CAI, T. T. and LI, H. (2023). Estimation and inference for high-dimensional generalized linear models with knowledge transfer. *Journal of the American Statistical Association* 1–12. MR4765987

[26] LUCKETT, D. J., LABER, E. B., KAHKOSKA, A. R., MAAHS, D. M., MAYER-DAVIS, E. and KOSOROK, M. R. (2019). Estimating dynamic treatment regimes in mobile health using v-learning. *Journal of the American Statistical Association*. MR4107673

[27] MOUSAVI, A., NADJAR ARAABI, B. and NILI AHMADABADI, M. (2014). Context transfer in reinforcement learning using action-value functions. *Computational intelligence and neuroscience* **2014**.

[28] MURPHY, S. (2005). A Generalization Error for Q-Learning. *Journal of Machine Learning Research: JMLR* **6** 1073–1097. MR2249849

[29] PRASAD, N., CHENG, L. F., CHIVERS, C., DRAUGELIS, M. and ENGELHARDT, B. E. (2017). A reinforcement learning approach to weaning of mechanical ventilation in intensive care units. In *33rd Conference on Uncertainty in Artificial Intelligence, UAI 2017*.

[30] SCHULTE, P. J., TSIATIS, A. A., LABER, E. B. and DAVIDIAN, M. (2014). Q- and A-learning methods for estimating optimal dynamic treatment regimes. *Statistical Science: A Review Journal of the Institute of Mathematical Statistics* **29** 640. MR3300363

[31] SINGLA, A., RAFFERTY, A. N., RADANOVIC, G. and HEFFERNAN, N. T. (2021). Reinforcement Learning for Education: Opportunities and Challenges. *arXiv preprint arXiv:2107.08828*.

[32] SONG, R., WANG, W., ZENG, D. and KOSOROK, M. R. (2015). Penalized *Q*-learning for dynamic treatment regimens. *Statistica Sinica* **25** 901. MR3409730

[33] SRIPERUMBUDUR, B. K., FUKUMIZU, K., GRETTON, A., SCHÖLKOPF, B. and LANCKRIET, G. R. (2009). On integral probability metrics,\phi-divergences and binary classification. *arXiv preprint arXiv:0901.2698*.

[34] SUTTON, R. S. and BARTO, A. G. (2018). *Reinforcement Learning: An Introduction*. MIT press. MR3889951

[35] TAYLOR, M. E. and STONE, P. (2009). Transfer learning for reinforcement learning domains: A survey. *Journal of Machine Learning Research* **10**. MR2534874

[36] TRIPURANENI, N., JIN, C. and JORDAN, M. I. (2020). Provable meta-learning of linear representations. *arXiv preprint arXiv:2002.11684*.

[37] YANG, L. and WANG, M. (2019). Sample-optimal parametric Q-learning using linearly additive features. In *International Conference on Machine Learning* 6995–7004. PMLR.

[38] ZHANG, Y., CHEN, E. and YAN, Y. (2025). Transfer Faster, Price Smarter: Minimax Dynamic Pricing under Cross-Market Preference Shift. In *The 39th Conference on Neural Information Processing Systems (NeurIPS 2025), arXiv:2505.17203*.

[39] ZHOU, R., CHEN, C. and CHEN, E. (2025). Prior-Aligned Meta-RL: Thompson Sampling with Learned Priors and Guarantees in Finite-Horizon MDPs. *arXiv preprint arXiv:2510.05446*.