One-Step Bellman Alignment Enables Provably Efficient Transfer in Online RL

Elynn Chen[♯*] Enpei Zhang[♮] Jinhang Chai[◇] Yujun Yan[†*]

[♯]Department of Technology, Operations, & Statistics, New York University
[◇]Department of Operations Research & Financial Engineering, Princeton University
[♮†] Department of Computer Science, Dartmouth College

May 26, 2026

Abstract

We study online transfer reinforcement learning (RL) in episodic Markov decision processes, where experience from related source tasks is available during learning on a target task. A fundamental difficulty is that task similarity is typically defined in terms of rewards or transitions, whereas online RL algorithms operate on Bellman regression targets. As a result, naively reusing source Bellman updates introduces systematic bias and invalidates regret guarantees.

We identify one-step Bellman alignment as the correct abstraction for transfer in online RL and propose re-weighted targeting (RWT), an operator-level correction that retargets continuation values and compensates for transition mismatch via a change of measure. RWT reduces task mismatch to a fixed one-step correction and enables statistically sound reuse of source data.

This alignment yields a two-stage RWT Q-learning framework that separates variance reduction from bias correction. Under RKHS function approximation, we establish regret bounds that scale with the complexity of the task shift rather than the target MDP. Empirical results in both tabular and neural network settings demonstrate consistent improvements over single-task learning and naïve pooling, highlighting Bellman alignment as a model-agnostic transfer principle for online RL.

Keywords: Transfer reinforcement learning; Q-learning, Optimistic under uncertainty; Regret analysis; RKHS;

*Correspondence to E. Chen at elynn.chen@stern.nyu.edu and Y. Yan at yujun.yan@dartmouth.edu

1 Introduction

Reinforcement learning agents are increasingly deployed in settings where experience from related tasks is available, such as simulation-to-real transfer, personalization across users, and decision-making across markets. Leveraging such source experience to accelerate learning on a new target task is natural in practice, yet providing principled guarantees for *online* transfer under exploration remains challenging. While many transfer heuristics perform well empirically, existing theory often treats the target task in isolation or relies on assumptions misaligned with how online RL algorithms actually learn Taylor and Stone [2009], Brunskill and Li [2013].

A central obstacle is that *task similarity is rarely defined at the level where online reinforcement learning operates*. Modern RL algorithms learn by regressing *one-step Bellman targets*, which depend jointly on the continuation value and the transition distribution (e.g. Yang et al. [2020], Jin et al. [2020]. As a result, similarity assumptions stated in terms of rewards, transitions, or value functions do not directly translate into reusable Bellman samples. Naïvely pooling source Bellman updates therefore introduces systematic bias even when tasks are intrinsically close, breaking optimism-based exploration and invalidating regret guarantees Srinivas et al. [2009].

We show that this failure is structural rather than statistical. The object governing transfer in online RL is the *one-step Bellman mismatch* between tasks. Under the standard Bellman operator, this mismatch depends intrinsically on the continuation value and thus varies across dynamic programming iterations, making it impossible to represent as a fixed state-action correction. Consequently, without explicit alignment, there is no principled way to reuse source Bellman samples in online reinforcement learning.

To resolve this obstruction, we introduce *re-weighted targeting (RWT)*, an operator-level Bellman alignment that retargets continuation values to the target task and corrects transition mismatch via a change of measure. This alignment removes continuation-value dependence and reduces task mismatch to a *fixed one-step correction*. As a result, source Bellman samples become Bellman-

consistent for the target task up to a simple residual, making transfer statistically well-posed.

Bellman alignment has direct algorithmic and theoretical consequences. Once mismatch is reduced to a fixed one-step correction, value estimation naturally decomposes into two stages: a *source-based baseline* that leverages aligned Bellman targets for variance reduction, and a *target-based correction* that learns the structured task shift. Under RKHS function approximation, we establish regret bounds that scale with the complexity of the task shift rather than the target MDP, yielding strict improvements over single-task learning when the shift is structured. Empirically, we observe consistent improvements over both single-task learning and naïve pooling across tabular and neural network Q-learning, highlighting Bellman alignment as a principled, model-agnostic transfer mechanism.

Contributions. Our main contributions are:

(i) *Conceptual:* We identify *one-step Bellman alignment* as the correct abstraction for *online* transfer and show that naive Bellman reuse is structurally biased;

(ii) *Algorithmic:* We propose *re-weighted targeting (RWT)*, an operator-level alignment that enables Bellman-consistent reuse of source data and yields a two-stage Q-learning framework for online learning with exploration.

(iii) *Theoretical:* We establish regret bounds under RKHS function approximation that scale with the complexity of the task shift rather than the target MDP;

(iv) *Empirical:* We demonstrate consistent gains over single-task learning and naïve pooling in both tabular and neural network settings.

1.1 Related Work and Our Distinction

Meta and multitask RL. A large literature studies how experience from related tasks can improve sample efficiency in RL, including multitask and meta-RL approaches that learn shared representations, priors, or initializations [Taylor and Stone, 2009, Brunskill and Li, 2013, Calandriello

et al., 2014, Zhou et al., 2025]. Recent theory establishes benefits of representational transfer under shared structure, such as linear or low-rank representations and shared latent dynamics, typically by reducing the effective dimension of the target problem [Hu et al., 2021, Agarwal et al., 2023, Sam et al., 2024, Lee et al., 2025]. These works focus on representation sharing across tasks. In contrast, we identify a distinct obstruction specific to *online* value-based RL: transfer must be compatible with Bellman regression targets under exploration, which is not addressed by representation-level similarity alone.

Transfer under task shift. A more closely related line studies transfer under structured reward/transition differences, often in finite-horizon settings. For example, Chen et al. [2025b], Zhang et al. [2025], Chen et al. [2026] analyze transfer Q-learning across related MDPs, and Chen et al. [2025a] develop data-driven transfer in *batch* Q-learning. Recent extensions consider composite MDP structures and transition transfer [Chai et al., 2025]. However, existing treatments either operate in offline/batch regimes or rely on forms of reuse that do not explicitly resolve the *continuation-value dependence* of Bellman mismatch in online learning. Our work differs by formalizing transfer at the *Bellman-operator level* and providing regret guarantees for online learning under exploration after alignment.

Kernelized RL and optimism. Our analysis builds on kernelized RL with optimism-based exploration, which yields regret bounds scaling with RKHS complexity measures such as information gain rather than the size of the state–action space [Yang et al., 2020, Vakili and Olkhovskaya, 2023, 2024]. Prior kernel RL work treats the target task in isolation. In contrast, we decompose value estimation into a source-based baseline and a residual correction with distinct RKHS complexity, and design upper confidence bounds that jointly capture uncertainty from both components.

2 Transferability via Bellman Alignment

A central difficulty in transfer reinforcement learning is that *task similarity is typically defined at a level misaligned with how online RL algorithms learn.* Although tasks may be similar in rewards or dynamics, episodic value-based methods learn by regressing one-step Bellman targets, which depend jointly on the continuation value and the transition distribution. We show that naive Bellman transfer fails due to continuation-value dependence, introduce re-weighted targeting (RWT) to align Bellman operators at one step, and show that this alignment reduces task mismatch to a fixed, learnable one-step correction.

2.1 Problem Setup

We consider episodic Markov decision processes (MDPs) with finite horizon H and discount factor $\gamma \in (0, 1]$. There is a target task, indexed by $m = 0$, and a collection of source tasks, indexed by $m \in [M]$. Each task m is specified by an MDP: $\mathcal{M}^{(m)} = \big(\mathcal{S}, \mathcal{A}, H, \{P_h^{(m)}\}_{h=1}^H, \{R_h^{(m)}\}_{h=1}^H, \gamma\big)$ with the same state and action spaces $(\mathcal{S}, \mathcal{A})$ but may differ in reward functions $R_h^{(m)}$ and transition dynamics $P_h^{(m)}$.

The learner interacts online only with the target task $\mathcal{M}^{(0)}$, collecting one trajectory per episode and incurring regret relative to the optimal target policy.

Source tasks may provide offline datasets or auxiliary interaction (e.g., simulators), collected under arbitrary, possibly non-optimistic policies. Our goal is to leverage such source data to accelerate online learning on the target task without compromising exploration or regret guarantees.

2.2 Why Naïve Bellman Transfer Fails

A natural approach to transfer is to reuse Bellman updates from source tasks when learning the target task. If source and target MDPs are "close," one might expect that pooling Bellman samples reduces variance and accelerates learning. We show that this intuition is incorrect in online RL.

Let $V_{h+1}^{(m)}(s) = \max_{a'} Q_{h+1}^{(m)}(s, a')$ where $Q_{h+1}^{(m)}(s, a')$ is the action-value function at stage $h{+}1$ for task m. The standard Bellman operator for task m at stage h is

$$(\mathcal{B}_h^{(m)} V_{h+1}^{(m)})(s, a) = R_h^{(m)}(s, a) + \gamma \mathbb{E}_{s' \sim P_h^{(m)}(\cdot|s,a)}[V_{h+1}^{(m)}(s')].$$

There are two distinct sources of misalignment between $\mathcal{B}_h^{(m)}$ and the target Bellman operator $\mathcal{B}_h^{(0)}$. First, the continuation value $V_{h+1}^{(m)}$ is task-dependent. Even if transitions were identical, replacing $V_{h+1}^{(0)}$ with $V_{h+1}^{(m)}$ introduces bias unless the value functions coincide. Second, expectations are taken under the wrong transition distribution $P_h^{(m)}$ instead of $P_h^{(0)}$.

Under the standard Bellman operator, the discrepancy between source and target Bellman backups, $(\mathcal{B}_h^{(m)} V_{h+1}^{(m)})(s, a) - (\mathcal{B}_h^{(0)} V_{h+1}^{(0)})(s, a)$, depends intrinsically on the continuation values through both the future-value term and the transition distribution. Without imposing strong assumptions on direct similarity between value functions – which are generally untestable in practice – pooling source Bellman targets therefore produces persistent bias that does not vanish with additional target data. This bias is structural, arising from Bellman-operator misalignment, and is incompatible with optimism-based exploration and regret guarantees in online reinforcement learning.

2.3 Re-Weighted Targeting for Bellman Alignment

We now introduce *re-weighted targeting (RWT)*, an operator-level correction that removes the continuation-value dependence identified above.

Fix stage h and source task m. Assume that the target transition kernel $P_h^{(0)}$ is absolutely continuous with respect to $P_h^{(m)}$, and define the density ratio

$$\omega_h^{(m)}(s' \mid s, a) := p_h^{(0)}(s' \mid s, a) / p_h^{(m)}(s' \mid s, a).$$

The *RWT-aligned Bellman operator* is defined as

$$\big(\mathcal{B}_h^{(m\to 0)} V_{h+1}^{(m)}\big)(s,a) := R_h^{(m)}(s,a) + \gamma\, \mathbb{E}_{s'\sim P_h^{(m)}(\cdot|s,a)}\Big[\omega_h^{(m)}(s' \mid s,a)\, V_{h+1}^{(0)}(s')\Big]. \tag{1}$$

Crucially, the continuation value V_{h+1} is evaluated under the target task, while the density ratio corrects the transition mismatch.

For any bounded continuation value V_{h+1}, define the *RWT-aligned Bellman difference*

$$\Delta_h^{(m)}(s,a) \;:=\; \big(\mathcal{B}_h^{(0)} V_{h+1}^{(0)}\big)(s,a) - \big(\mathcal{B}_h^{(m\to 0)} V_{h+1}^{(m)}\big)(s,a),$$

A direct calculation shows that

$$\Delta_h^{(m)}(s,a) \equiv R_h^{(0)}(s,a) - R_h^{(m)}(s,a), \tag{2}$$

which depends only on (s,a) and is independent of V_{h+1}. Thus, re-weighted targeting transforms the task-to-task Bellman difference from a continuation-value-dependent object into a fixed one-step correction that is invariant across dynamic programming iterations.

2.4 Transferability via Structured Reward Differences

After Bellman alignment, task mismatch is fully characterized by the one-step reward difference

$$\Delta_{r,h}^{(m)}(s,a) := R_h^{(0)}(s,a) - R_h^{(m)}(s,a). \tag{3}$$

We therefore formalize transferability by imposing structure directly on this reward-level discrepancy.

Assumption 2.1 (**Structured Reward Difference).** For each stage h and source task m, the reward difference $\Delta_{r,h}^{(m)}$ belongs to a function class $\mathcal{F}_\Delta$ whose complexity is strictly smaller than that of the class $\mathcal{F}_R$ used to model the target Bellman backup.

This assumption applies exclusively to the *one-step* reward difference, which is intrinsic to the task definition and independent of policies or value functions. We do not assume similarity of value functions across tasks, nor do we impose structure on transition kernels or multi-step value differences, which may remain arbitrarily complex. The assumption is meaningful precisely because RWT renders the Bellman mismatch invariant across iterations: learning $\Delta_{r,h}^{(m)}$ suffices to correct aligned source Bellman targets.

Under Assumption 2.1, source data can be used to construct Bellman-consistent pseudo-labels via re-weighted targeting, while target data is required only to estimate the simpler reward correction. This separation, i.e. variance reduction from sources and bias correction from targets, forms the basis of the algorithmic design in Section 3 and enables regret bounds that depend on the complexity of the task shift rather than the full target problem.

3 RWT Q-Learning: Algorithmic Framework

This section presents a *model-agnostic algorithmic framework for online transfer reinforcement learning* based on the Bellman alignment developed in Section 2. We refer to the resulting method as **RWT Q-learning**. Algorithm 1 gives an optimism-based instantiation, termed **OFU-RWT Q-learning**. While OFU is used for theoretical analysis, the Bellman alignment mechanism is independent of the exploration strategy and can be combined with alternatives such as ε-greedy, as used in our empirical evaluation.

Learning proceeds episodically: in each episode, the agent rolls out a policy on the target task, then performs Bellman-aligned updates that combine a source-based baseline for variance reduction with a target-based correction for the residual task shift, before deploying the updated policy in the next episode.

Notation. We denote by N the total number of episodes of interaction with the target task, each of horizon H (so $T = NH$ total steps). We further denote by κ the source-to-target sampling

ratio, meaning that by episode n the m-th source task provides $n^{(m)} = \kappa^{(m)} n$ samples, and we write $\kappa = \sum_{m=1}^{M} \kappa^{(m)}$.

3.1 Synchronous Bellman Updates with RWT

After episode n, the learner performs *synchronous backward updates* of the stage-wise Q-functions $\{Q_{n+1,h}\}_{h=1}^{H}$, starting from $h = H$ down to 1. Let the continuation value be $V_{n,h+1}(s) := \max_{a \in \mathcal{A}} Q_{n,h+1}(s, a)$. With transfer, Bellman updates are constructed using RWT, which enables principled reuse of source data while preserving Bellman correctness for the target task.

Stage I. (Source-based Bellman alignment). For each source task m, given the density-ratio estimators $\widehat{\omega}_h^{(m)}$, we construct *RWT-aligned Bellman pseudo-labels*

$$y_{i,h}^{(m\to 0)} := r_{i,h}^{(m)} + \gamma\, \widehat{\omega}_h^{(m)}(s_{i,h+1}^{(m)} \mid s_{i,h}^{(m)}, a_{i,h}^{(m)})\, V_{h+1}^{(0)}(s_{i,h+1}^{(m)}). \tag{4}$$

By construction, these pseudo-labels are Bellman-consistent with the target task up to the one-step reward difference identified in Section 2. Pooling aligned pseudo-labels across source tasks yields a *baseline estimator*

$$Q_{n,h}^{\text{base}} \;\approx\; \mathcal{B}_h^{(0)} V_{n,h+1}^{(0)} - \Delta_{r,h}^{(m)},$$

which leverages source data to reduce estimation variance without introducing Bellman bias.

Stage II. (Target-based correction). Using target-task samples, we form residual labels

$$y_{i,h}^{(0)} = r_{i,h}^{(0)} + \gamma V_{h+1}^{(0)}(s_{i,h+1}^{(0)}),\; z_{i,h} = y_{i,h}^{(0)} - Q_{n,h}^{\text{base}}(s_{i,h}^{(0)}, a_{i,h}^{(0)}), \tag{5}$$

and estimate a correction $\delta_{n,h}$ that captures the structured one-step reward difference. The resulting transfer update is

$$Q_{n,h}^{\text{trans}} = Q_{n,h}^{\text{base}} + \delta_{n,h}. \tag{6}$$

This two-stage update directly mirrors the Bellman-level decomposition in Section 2: *source data contributes primarily to variance reduction, while target data is used to learn the structured task shift.*

3.2 Exploration and Policy Deployment

To ensure exploration, we adopt the principle of optimism in the face of uncertainty. After the transfer update in Section 3.1, we construct an optimistic estimate

$$Q_{n,h}^{\mathrm{ucb}}(s,a) \;:=\; Q_{n,h}^{\mathrm{trans}}(s,a) + b_{n,h}(s,a),$$

where the exploration bonus $b_{n,h}$ accounts for uncertainty from both stages of estimation: the source-based baseline (including finite source data and density-ratio estimation) and the target-based correction. An explicit form of $b_{n,h}$ for the RKHS instantiation is given in Section 4.2.

We then set $Q_{n+1,h} = Q_{n,h}^{\mathrm{ucb}}$. In episode $n+1$, the agent follows the greedy policy:

$$a_{n+1,h} \in \arg\max_{a\in\mathcal{A}} Q_{n+1,h}(s_{n+1,h}, a),\ h = 1, \ldots, H.$$

While optimism underpins our theoretical analysis, RWT Q-learning itself is exploration-agnostic and can be paired with alternative strategies, such as ε-greedy, when upper confidence bounds are difficult to derive (e.g., under neural network function approximation). We adopt this approach in our empirical evaluation with DQN.

3.3 Algorithm Summary

Algorithm 1 summarizes OFU-RWT Q-learning, an optimism-based instantiation of RWT Q-learning. Learning proceeds episodically on the target task. In each episode, the agent rolls out a policy on the target task, collects a single trajectory, and then performs synchronous backward Bellman

Algorithm 1 OFU-RWT Q-Learning (Model-Agnostic)

Input: Horizon H, discount γ; target task $\mathcal{M}^{(0)}$ (online); source datasets $\{\mathcal{D}^{(m)}\}_{m=1}^{K}$ (online or offline); density-ratio estimators $\{\widehat{\omega}_h^{(m)}\}$ (or known $\omega_h^{(m)}$); function classes $\mathcal{F}_R$ (baseline) and $\mathcal{F}_\Delta$ (correction).
Output: Policies $\{\pi_n\}_{n=1}^{N}$ and Q-functions $\{Q_{n,h}\}$.
Initialization:
Set $Q_{1,h}(s,a) \leftarrow H$ for all $h \in [H]$ and all (s,a).
Set $V_{1,h}(s) \leftarrow \max_a Q_{1,h}(s,a)$.
for $n = 1, 2, \ldots, N$ **do**
  (A) Rollout on target task.
  Observe initial state $s_{n,1}$.
  for Each stage $h = 1, \cdots, H$ **do**
    Select $a_{n,h}(s) = \arg\max_a Q_{n,h}(s,a)$.
    Execute $a_{n,h}(s)$ in $\mathcal{M}^{(0)}$, and observe reward $r_{n,h}$ and next state $s_{n,h+1}$.
  end for
  Store the trajectory $\tau_n = \{(s_{n,h}, a_{n,h}, r_{n,h}, s_{n,h+1})\}_{h=1}^{H}$.
  (B) Synchronous backward Bellman updates with RWT.
  Set $V_{n,H+1} \equiv 0$.
  for $h = H, H-1, \ldots, 1$ **do**
    Stage I (Source-based Bellman alignment):
    Construct RWT-aligned pseudo-labels $y_h^{(m\to 0)}$ from $\{\mathcal{D}^{(m)}\}_{m=1}^{K}$ using $V_{n,h+1}$ via (4), and fit a baseline estimator $Q_{n,h}^{\text{base}} \in \mathcal{F}_R$.
    Stage II (Target-based correction).
    Using target data $\{\tau_i\}_{i\le n}$ and residual labels via (5), estimate correction $\delta_{n,h} \in \mathcal{F}_\Delta$.
    Set $Q_{n,h}^{\text{trans}} \leftarrow Q_{n,h}^{\text{base}} + \delta_{n,h}$.
    (Optimistic Q-update).
    Set $Q_{n+1,h} \leftarrow Q_{n,h}^{\text{trans}} + b_{n,h}$.
    Set $V_{n,h}(s) \leftarrow \max_a Q_{n,h}(s,a)$.
  end for
  (C) Policy update.
  For each $h \in [H]$, set $V_{n+1,h}(s) \leftarrow \max_a Q_{n+1,h}(s,a)$ and π_{n+1} greedy with respect to $Q_{n+1,h}$.
end for

updates.

At each stage, RWT-aligned source pseudo-labels are first used to fit a source-based baseline, which leverages source data for variance reduction while remaining Bellman-consistent with the target task. Using target-task data, a correction term is then estimated to capture the structured one-step task shift. The baseline and correction are combined into a transfer estimate, which is augmented with an exploration bonus to form the updated Q-function for the next episode.

Section 4 instantiates this procedure under RKHS function approximation and establishes regret guarantees for the optimism-based variant. Section 5 evaluates RWT Q-learning under both tabular and neural network implementations, using ε-greedy exploration when explicit confidence bounds are difficult to derive.

4 RKHS Instantiation and Regret Analysis

This section instantiates RWT Q-learning under reproducing kernel Hilbert space (RKHS) function approximation and establish regret guarantees for the optimism-based variant. Building on the RWT Bellman alignment framework of Section 2 and the algorithmic structure in Section 3, we formalize the structured one-step task shift in RKHS, derive explicit estimators and exploration bonuses, and analyze the resulting regret. The analysis highlights how Bellman alignment localizes statistical complexity to the task shift, yielding regret bounds that depend on the complexity of the shift rather than that of the target MDP.

4.1 RKHS Modeling of Structured Task Shift

We now specify an RKHS instantiation that formalizes the structured one-step task shift identified in Section 2. Throughout, let $\mathcal{K}$ be an RKHS over $\mathcal{S} \times \mathcal{A}$ with kernel k and feature map ϕ.

We assume that the *target Bellman backup* admits an RKHS representation: for each stage h,

$$(\mathcal{B}_h^{(0)} V_{h+1})(\cdot, \cdot) \in \mathcal{K},$$

with uniformly bounded RKHS norm. This assumption is standard in kernelized value-based RL and places no restriction on policy structure or value-function similarity across tasks [Yang et al., 2020, Vakili and Olkhovskaya, 2023].

Sufficient conditions. As a concrete sufficient condition, if the target reward $R_h^{(0)} \in \mathcal{K}$ and the

transition kernel admits a bounded linear operator representation in $\mathcal{K}$, then the Bellman backup belongs to $\mathcal{K}$ for any bounded continuation value. We emphasize, however, that our analysis does not rely on explicitly modeling transitions; this condition is provided only for intuition.

Following the Bellman alignment framework of Section 2, *transferability is imposed exclusively on the aligned one-step mismatch*, not on value functions or multi-step Bellman differences.

Structured one-step reward difference. For each stage h and source task m, recall the aligned one-step reward difference

$$\Delta_{r,h}^{(m)}(s,a) := R_h^{(0)}(s,a) - R_h^{(m)}(s,a).$$

We assume this discrepancy lies in a lower-complexity RKHS. Further details are left in Appendix A,

Assumption 4.1 (RKHS-Structured Task Shift)**.** For each stage h and source task m, there exists an RKHS $\widetilde{\mathcal{K}} \subseteq \mathcal{K}$ with feature map $\widetilde{\phi}$ such that

$$\Delta_{r,h}^{(m)} \in \widetilde{\mathcal{K}}, \qquad \|\Delta_{r,h}^{(m)}\|_{\widetilde{\mathcal{K}}} \leq B_\Delta.$$

The inclusion $\widetilde{\mathcal{K}} \subseteq \mathcal{K}$ captures the assumption that the *task shift is simpler than the ambient problem*, e.g., smoother, lower-rank, or lower-dimensional. Crucially, we *do not* assume similarity of value functions or multi-step Bellman differences across tasks, which are policy-dependent and generally unverifiable.

4.2 OFU-RWT Q-Learning under RKHS Approximation

We now instantiate OFU-RWT Q-learning under the RKHS assumptions. This subsection makes explicit how Algorithm 1 is realized with kernel ridge regression and how uncertainty from the two learning stages is quantified.

Stage I. (Source-based Bellman alignment). At each episode n and stage h, RWT-aligned

pseudo-labels constructed via (4) are used to estimate a *source-based baseline* $Q_{n,h}^{\text{base}} \in \mathcal{K}$. Concretely, we perform kernel ridge regression

$$Q_{n,h}^{\text{base}} \arg\min_{f \in \mathcal{K}} \sum_{m=1}^{M} \sum_{i=1}^{n^{(m)}} \left(f(s_{i,h}^{(m)}, a_{i,h}^{(m)}) - y_{i,h}^{(m \to 0)} \right)^2 + \lambda \|f\|_{\mathcal{K}}^2. \tag{7}$$

This estimator leverages abundant source data to reduce variance while remaining Bellman-consistent with the target task up to the structured one-step correction. No structural assumptions are imposed beyond the RKHS representability of the target Bellman backup.

Stage II. (Target-based correction). Using target-task data collected up to episode n, we estimate a *correction term* $\delta_{n,h} \in \widetilde{\mathcal{K}}$ that captures the structured one-step task shift. Given residual labels defined in (5), we solve

$$\delta_{n,h} = \arg\min_{f \in \widetilde{\mathcal{K}}} \sum_{i=1}^{n} \left(f(s_{i,h}^{(0)}, a_{i,h}^{(0)}) - z_{i,h} \right)^2 + \widetilde{\lambda} \|f\|_{\widetilde{\mathcal{K}}}^2. \tag{8}$$

The resulting *transfer estimate* $Q_{n,h}^{\text{trans}} = Q_{n,h}^{\text{base}} + \delta_{n,h}$ combines variance reduction from source data with bias correction from target data, reflecting the Bellman-aligned decomposition of Section 2.

Optimistic Q-update. To support exploration, we augment the transfer estimate with an exploration bonus, $Q_{n,h}^{\text{ucb}}(s,a) = Q_{n,h}^{\text{trans}} + b_{n,h}(s,a)$. The exploration bonus $b_{n,h}(s,a)$ accounts for uncertainty from both stages: finite source data (including density-ratio estimation) and online learning of the correction. For RKHS regression, it admits the form (Lemma B.1):

$$\begin{aligned} b_{n,h}(s,a) =& \beta_{n,m} \sqrt{\phi(s,a)^\top (\Lambda_{n,h}^{(m)} + \lambda I)^{-1} \phi(s,a)} \\ &+ \beta_{n,0} \sqrt{\widetilde{\phi}(s,a)^\top (\Lambda_{n,h}^{(0)} + \widetilde{\lambda} I)^{-1} \widetilde{\phi}(s,a)} \end{aligned} \tag{9}$$

with confidence parameters $\beta_{n,m}$ and $\beta_{n,0}$ independent of (s,a). Here $\Lambda_{n,h}^{(m)}$ and $\Lambda_{n,h}^{(0)}$ denote the empirical covariance operators induced by the baseline and correction feature maps ϕ and $\widetilde{\phi}$,

respectively, using source samples from task m and target samples up to episode n.

4.3 Regret Analysis

We now state the regret guarantee for the RKHS instantiation of Algorithm 1. Proofs are deferred to Appendix B.

4.3.1 Complexity Measures and Their Roles

The regret bound depends on standard RKHS complexity quantities, reflecting the distinct statistical roles of the source-based baseline and the target-based correction.

Definition 4.2 (Baseline RKHS complexity (source stage))**.** Let $T_{\mathcal{K}}$ denote the integral operator associated with kernel $\mathcal{K}$. The effective dimension at regularization level $\lambda > 0$,

$$\mathcal{N}_0(\lambda) = \mathrm{Tr}\left(T_{\mathcal{K}}(T_{\mathcal{K}} + \lambda I)^{-1}\right),$$

controls the estimation error of kernel ridge regression with i.i.d. samples. We assume $\mathcal{N}_0(\lambda) \lesssim \lambda^{-2\beta_0}$, which holds for common kernels with polynomial eigenvalue decay.

The $N_0(\lambda)$ controls estimation error for kernel ridge regression with i.i.d. source trajectories. This quantity appears when bounding the baseline estimation error terms in Lemma B.1, where source samples are independent of the target trajectory and standard RKHS concentration applies.

Definition 4.3 (Correction RKHS complexity (target stage))**.** For the correction RKHS $\widetilde{\mathcal{K}}$, target data are collected online and are adaptively dependent. Two complexity measures are therefore required.

- **Maximal information gain:** For the correction kernel $\widetilde{\mathcal{K}}$, define the maximal information gain

$$\Gamma_1(N, \widetilde{\lambda}) = \sup_{G \in \mathcal{G}_{\widetilde{\mathcal{K}}, N}} \log\det(I + G/\widetilde{\lambda}),$$

where $\mathcal{G}_{\widetilde{\mathcal{K}},N}$ is the set of all $N \times N$ Gram matrices induced by $\widetilde{\mathcal{K}}$. We assume $\Gamma_1(N, \widetilde{\lambda}) \leq N^{\beta_1}/\widetilde{\lambda}^2$ for $\widetilde{\lambda} \lesssim 1$, which holds for common kernels such as squared exponential or Matérn kernels.

- **Uniform covering number:** We further assume that the L_∞-covering number of $\widetilde{\mathcal{K}}$ satisfies $\log \mathcal{N}_\infty(\widetilde{\mathcal{K}}, \epsilon) \leq \epsilon^{-\alpha_1}$. For the induced quadratic-form class $\mathcal{F}_{\widetilde{\mathcal{K}}}(\epsilon, \widetilde{\lambda}) := \{\widetilde{\phi}^\top Q \widetilde{\phi} : \|Q\|_2 \leq 1/\widetilde{\lambda}\}$, we assume $\log \mathcal{N}_\infty(\mathcal{F}_{\widetilde{\mathcal{K}}}, \epsilon, \widetilde{\lambda}) \leq (\widetilde{\lambda}\epsilon)^{-\alpha_1}$.

These conditions enter the proof when controlling self-normalized martingale terms arising from online target data (Appendix B). As is standard in kernelized RL with optimism, effective dimension alone is insufficient in the online setting Yang et al. [2020], Vakili and Olkhovskaya [2023].

Here we define the Maximal Information Gain as the maximum over all N-points Gram matrices G, here $G_{\widetilde{\mathcal{K}},N}$ is the class of all N by N gram matrices with respect to kernel $\widetilde{\mathcal{K}}$. This is the online analog of effective dimension

$$\Gamma_1(N, \widetilde{\lambda}) = \sup_{G \in G_{\widetilde{\mathcal{K}},N}} \log\det(I + G/\widetilde{\lambda})$$

Assume $\Gamma_1(N, \widetilde{\lambda}) \leq N^{\beta_1}/\widetilde{\lambda}^2$ for $\widetilde{\lambda} \lesssim 1$. This holds for squared exponential kernel or Matern kernel. For details, we refer the reader to Theorem 5 in Srinivas et al. [2009].

4.3.2 Source Data and Density-Ratio Conditions

We next formalize conditions governing how effectively source data can be exploited.

Assumption 4.4 (Source coverage)**.** We assume a uniform source coverage condition:

$$|\phi(s,a)^\top (\Lambda_{n,h}^{(m)} + \lambda)^{-1} \phi(s,a)| \leq C_{\mathrm{cov}}/n^{(m)}.$$

holds for some coverage parameter C_{cov} and every source m and stage h.

This ensures that source data reduce variance rather than induce ill-conditioned estimates.

The condition is used to bound the baseline confidence radius in Lemma B.1 and in the regret decomposition.

Assumption 4.5 (Density-ratio estimation error)**.** Let $\widehat{\omega}_{i,h}$ be the estimated density ratio. Define the per-episode averaged ratio error at stage h as $\mathcal{E}_{n,h}^2 := \frac{1}{\kappa n}\sum_{m=1}^{M}\sum_{i=1}^{n^{(m)}}(\hat{\omega}_{i,h}^{(m)} - \omega_{i,h}^{(m)})^2$. We assume the cumulative error satisfies

$$\sum_{n=1}^{N}\sqrt{\mathcal{E}_{n,h}^2} \lesssim N^{\frac{2\alpha_0+1}{2(\alpha_0+1)}}\kappa^{-\frac{1}{2(\alpha_0+1)}},$$

where N the total number of episodes of interaction with the target task and κ is the source-target sampling ratio defined in Section 3.

This term appears explicitly in the baseline error decomposition (term $T_2^{(\omega)}$ in Lemma B.1) and captures *imperfect Bellman alignment* due to density-ratio estimation. When κ is large or ratios are accurate, this contribution becomes lower order. Estimation strategies are discussed in Appendix C.

4.3.3 Main Regret Theorem

We are now ready to state the regret bound.

Theorem 4.6 (Regret under RKHS Instantiation)**.** *Under the baseline and correction RKHS complexity conditions (Definition 4.2–4.3), together with source coverage (Assumption 4.4) and density-ratio estimation (Assumption 4.5), with probability at least* $1 - 2/(NH)$*, the regret satisfies*

$$\begin{aligned}\mathrm{Regret}(N) \le H\sqrt{(N^{\beta_1})\left[N^{\beta_1+1} + N^{\alpha_1+1}\right]} \\ + H\sqrt{NC_{\mathrm{cov}}}N^{\frac{\alpha_0}{2(\alpha_0+1)}}\kappa^{-\frac{1}{2(\alpha_0+1)}}\end{aligned}$$

The regret bound decomposes naturally into two terms.

- **Target exploration term.** The *first* term depends only on the complexity of task-shift RKHS

$\widetilde{\mathcal{K}}$. This reflects the fact that, after Bellman alignment, exploration is required solely to learn the structured one-step task shift.

- **Source estimation term.** The *second* residual uncertainty from the source-based baseline, including coverage and density-ratio error. This term decreases with the source-to-target ratio κ and becomes negligible when source data are sufficiently abundant.

As a sanity check, when $\widetilde{\mathcal{K}}$ is finite-dimensional (e.g., linear MDPs), the bound recovers the standard $O(H\sqrt{N})$ regret.

4.3.4 Comparison and Regimes

Single-task RL. Without transfer, regret is governed by the ambient RKHS $\mathcal{K}$, leading to regret of order $\widetilde{\mathcal{O}}\big(HN^{1/2+\beta_0}\big)$.

Naïve pooled learning. Pooling Bellman updates without alignment induces persistent bias (Section 2) and admits no comparable regret guarantee unless tasks are identical; see also Section 5 for empirical evidence.

OFU-RWT transfer. By contrast, re-weighted targeting removes Bellman bias at one step and isolates uncertainty to the smaller space $\widetilde{\mathcal{K}}$. When $\alpha_1 \leq \beta_1$ (a mild condition; see Vakili and Olkhovskaya [2023]) and κ is large enough for the source term to saturate, the regret simplifies to $\mathrm{Regret}(N) = \widetilde{\mathcal{O}}\big(HN^{1/2+\beta_0}\big)$, which is strictly smaller than the single-task rate whenever the task shift is simpler than the ambient problem.

Overall, the theorem shows that Bellman alignment does more than reduce constants: it changes the effective statistical complexity of online RL, replacing dependence on the target MDP with dependence on the structured one-step task difference.

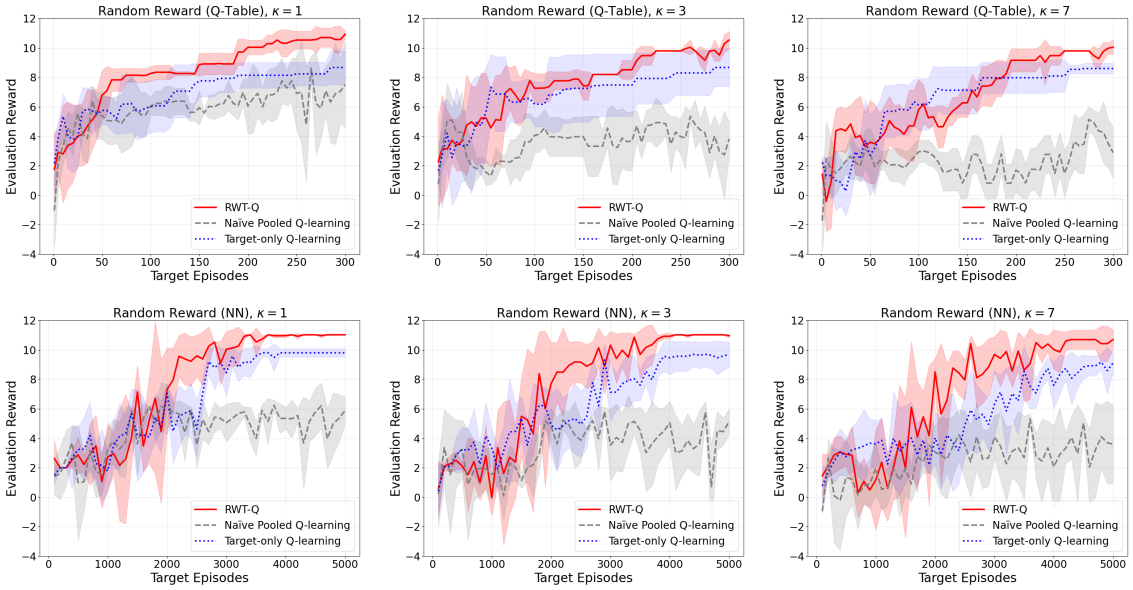

Figure 1: Learning curves comparing RWT-Q, naïve pooled Q-learning, and target-only Q-learning. **Top**: RandomRewardGridEnv with tabular Q-learning. **Bottom**: RandomRewardGridEnv with DQN function approximation. RWT-Q consistently improves sample efficiency, while naïve pooling often degrades performance due to Bellman misalignment.

5 Empirical Experiments

We evaluate *RWT Q-learning* on controlled grid-world benchmarks designed to test transfer under structured one-step reward shifts and to expose the failure of naïve Bellman reuse. All methods use identical exploration strategies and differ only in how source data are incorporated.

5.1 Experimental Setup

RandomRewardGridEnv. To control task mismatch, we construct environments with synthetic reward surfaces. The target reward is generated as a fixed Q-table with i.i.d. Gaussian entries, while the source reward is defined as

$$R_{\text{source}}(s,a) = R_{\text{target}}(s,a) + \Delta(s,a),$$

where $\Delta(s,a) \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,\sigma_\Delta^2)$ is a zero-mean Gaussian perturbation, where the standard deviation σ_Δ controls the complexity of the task shift. This environment directly instantiates the structured

one-step reward-difference model in Section 2.

Benchmarks. We compare the following methods:

• **RWT-Q (ours).** Implements the two-stage Bellman-aligned update: a source-based baseline Q-function combined with a target-only correction term. This corresponds exactly to the RWT decomposition analyzed in Sections 3-4.

• **Naïve pooled Q-learning.** Trains a single Q-function on pooled source and target data, implicitly assuming tasks are identical. This baseline reflects common heuristic transfer approaches and directly violates Bellman alignment.

• **Target-only Q-learning.** Standard Q-learning trained solely on target-task data, ignoring source information.

All methods use identical exploration strategies (ϵ-greedy) and differ only in how source data are incorporated.

Implementations. We run experiments with both *tabular Q-learning* and *neural network* function approximation. For neural experiments, all methods share the same architecture. We vary the source-target data ratio κ. Results are averaged over multiple random seeds. See more implementation details in Section D.

5.2 Results

Figure 1 reports undiscounted episode return averaged over seeds. Across all setting, *RWT-Q consistently outperforms both baselines*. In particular:

• Naïve pooling often degrades performance relative to target-only learning, confirming the structural Bellman bias predicted in Section 2.

• RWT-Q reliably improves sample efficiency, especially when the reward shift is structured but nontrivial.

• Performance gains persist under neural network approximation, indicating that Bellman

alignment provides benefits beyond the tabular setting.

Overall, these results empirically validate the central theoretical message of the paper: *transfer in online RL should respect Bellman alignment*, and naïve reuse of Bellman updates can be harmful even when tasks are closely related.

6 Conclusion

We showed that naïve transfer in online reinforcement learning fails due to continuation-value-dependent Bellman bias, and identified one-step Bellman alignment as the correct abstraction for principled transfer. We introduced re-weighted targeting (RWT), an operator-level correction that removes this dependence and reduces task mismatch to a fixed one-step correction, yielding a two-stage learning framework that separates variance reduction from bias correction.

Instantiated as RWT Q-learning and analyzed via an optimism-based variant, our theory establishes regret bounds that scale with the complexity of the task shift rather than the target MDP. Empirical results with both tabular and neural network implementations further demonstrate that Bellman alignment provides consistent benefits beyond the specific settings covered by our analysis.

References

Arpit Agarwal, Yandong Song, Wen Sun, Kevin Wang, Mengdi Wang, and Xin Zhang. Provable benefits of representational transfer in reinforcement learning. In *Proceedings of the Thirty Sixth Annual Conferenceon Learning Theory*, pages 2114–2187. PMLR, 2023.

Emma Brunskill and Lihong Li. Sample complexity of multi-task reinforcement learning. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence*, pages 122–131, 2013.

Daniele Calandriello, Alessandro Lazaric, and Marcello Restelli. Sparse multi-task reinforcement learning. *Advances in neural information processing systems*, 27, 2014.

Jinhang Chai, Elynn Chen, and Lin Yang. Transition transfer Q-learning with composite mdp structures. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 7089–7106. PMLR, 2025. URL `https://proceedings.mlr.press/v267/chai25b.html`.

Elynn Chen, Xi Chen, and Wenbo Jing. Data-driven knowledge transfer in batch Q-learning. *Journal of the American Statistical Association*, 2025a. doi: 10.1080/01621459.2025.2603731.

Elynn Chen, Sai Li, and Michael I Jordan. Transfer Q-learning for finite-horizon markov decision processes. *Electronic Journal of Statistics*, 19(2):5289–5312, 2025b. doi: 10.1214/25-EJS2459.

Elynn Chen, Xi Chen, and Yi Zhang. Transfer learning for contextual joint assortment-pricing under cross-market heterogeneity. *arXiv preprint arXiv:2603.18114*, 2026.

Jiachen Hu, Xiaoyu Chen, Chi Jin, Lihong Li, and Liwei Wang. Near-optimal representation learning for linear bandits and linear rl. In *International Conference on Machine Learning*, pages 4349–4358. PMLR, 2021.

Chi Jin, Zhuoran Yang, Zhaoran Wang, and Michael I Jordan. Provably efficient reinforcement learning with linear function approximation. In *Conference on Learning Theory*, pages 2137–2143, 2020.

Donghwan Lee, Kexin Zhang, Zhuoran Yang, and Mengdi Wang. Provably efficient multi-task reinforcement learning with shared linear structure. In *Proceedings of the Thirty-Ninth AAAI Conference on Artificial Intelligence*, 2025.

Tyler Sam, Yudong Chen, and Christina L Yu. The limits of transfer reinforcement learning with

latent low-rank structure. *Advances in Neural Information Processing Systems*, 37:108262–108330, 2024.

Niranjan Srinivas, Andreas Krause, Sham M Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. *arXiv preprint arXiv:0912.3995*, 2009.

Matthew E Taylor and Peter Stone. Transfer learning for reinforcement learning domains: A survey. *Journal of Machine Learning Research*, 10(7), 2009.

Sattar Vakili and Julia Olkhovskaya. Kernelized reinforcement learning with order optimal regret bounds. *Advances in Neural Information Processing Systems*, 36:4225–4247, 2023.

Sattar Vakili and Julia Olkhovskaya. Kernel-based function approximation for average reward reinforcement learning: an optimist no-regret algorithm. *Advances in Neural Information Processing Systems*, 37:25401–25425, 2024.

Zhuoran Yang, Chi Jin, Zhaoran Wang, Mengdi Wang, and Michael I Jordan. On function approximation in reinforcement learning: optimism in the face of large state spaces. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, pages 13903–13916, 2020.

Yi Zhang, Elynn Chen, and Yujun Yan. Transfer faster, price smarter: Minimax dynamic pricing under cross-market preference shift. In *Advances in Neural Information Processing Systems*, 2025. URL `https://arxiv.org/abs/2505.17203`.

Runlin Zhou, Chixiang Chen, and Elynn Chen. Prior-aligned meta-rl: Thompson sampling with learned priors and guarantees in finite-horizon mdps. *arXiv preprint arXiv:2510.05446*, 2025.