Transfer-Aware Regimes for Offline-to-Online RL Under Dynamics Shift

Liz Lemma Future Detective

January 20, 2026

Abstract

Offline-to-online RL is typically studied assuming the offline dataset and online deployment environment come from the same MDP. In 2026-era deployments (robot personalization, continual updates, sim-to-real), this assumption is routinely violated: offline data are collected under different dynamics or rewards than those encountered online. Building on the stability–plasticity lens and regime taxonomy from prior offline-to-online work, we introduce transfer-aware regimes defined by the on-environment performance of two priors: the offline-pretrained policy π_0 and a behavior-cloned policy π_{BC} trained on the offline dataset. We show that naively replaying offline transitions for value learning can be provably harmful under dynamics shift: Bellman backups computed from off-dynamics induce a bias scaling with the shift magnitude. We propose Transfer-Weighted Bellman Replay (TWBR), which uses online data to estimate a confidence-bounded mismatch score and selectively weights (or filters) offline transitions in critic learning while using offline data primarily as a behavioral prior. In linear MDPs, TWBR achieves a stability floor relative to the best transferable prior J^*_{tr} and a near-optimality bound with an explicit additive shift term, and we provide a matching lower bound showing the shift term is unavoidable without correction. We outline how modern deep RL implementations can instantiate the same principle via ensemble-based mismatch diagnostics and evaluate on controlled-shift benchmarks; such experiments would strengthen the practical relevance beyond the theory.

Table of Contents

3. 3. Problem Setup: two-MDP model $(\mathcal{M}_{\text{off}}, \mathcal{M}_{\text{on}})$; offline dataset generation; definition of π_{BC}, π_0, and transferable baseline J^*_{tr}; what it means to reuse offline transitions under shift.

4. 4. Transfer-Aware Regimes: define regimes using $J_{\text{on}}(\pi_0)$ vs $J_{\text{on}}(\pi_{\text{BC}})$ plus a shift/overlap axis (low vs high mismatch); map regimes to algorithmic prescriptions (policy anchoring vs dataset anchoring vs plasticity).

5. 5. Failure of Naive Offline Replay Under Shift: define Bellman-operator mismatch; prove bias bounds; provide an explicit counterexample family showing offline replay can degrade learning (matching lower bound).

6. 6. Algorithm: Transfer-Weighted Bellman Replay (TWBR): mismatch estimation, confidence sets, weighting/filtering rule, stability monitor; regime-dependent choice of (policy regularization, offline usage mode).

7. 7. Main Theorems: stability floor guarantee; near-optimality bound with an explicit shift term; matching lower bounds; (optional) sequential test bounds for regime identification on $\mathcal{M}_{\text{on}}$.

8. 8. Complexity and Implementation Notes: computational cost in linear setting; how to instantiate mismatch estimators with ensembles/density models in deep RL; what assumptions break in practice.

9. 9. Experimental Plan (flagged as strengthening): controlled dynamics-shift benchmarks, sim-to-real variants, personalization tasks; metrics: stability, plasticity, TD-loss drift on offline states; demonstrate naive replay harms and TWBR fixes it.

10. 10. Discussion: limitations, extensions to POMDPs and large foundation policies; continuous regime scores; safety constraints; open problems.

1 Background and Source Connection: stability–plasticity decomposition under shift

We adopt the stability–plasticity viewpoint for offline-to-online reinforcement learning: during online interaction we would like to (i) *remain stable*, in the sense of never performing substantially worse than a reasonable baseline that is already available from offline training, while (ii) *remain plastic*, in the sense of using the online samples to adapt toward optimality in the online environment. Formally, stability is encoded by a performance floor of the form

$$\min_{0\le t\le N} J_{\mathrm{on}}(\pi_t) \;\ge\; J^*_{\mathrm{tr}} - \varepsilon$$

with high probability, whereas plasticity is encoded by a final performance guarantee comparing π_N to an online optimal policy π^*.

The source formulation motivating our approach (in the no-shift setting) decomposes offline-to-online fine-tuning into regimes determined by a comparison between the offline-pretrained policy π_0 and the data-generating behavior $\pi_{\mathcal{D}}$ itself. In that classical viewpoint, the offline dataset $\mathcal{D}$ is assumed to arise from the *same* MDP on which performance is measured, and the quantity $J(\pi_{\mathcal{D}})$ is treated as a meaningful reference point capturing “how good the dataset is.” One then distinguishes three qualitatively different cases: roughly, a regime where π_0 is already better than $\pi_{\mathcal{D}}$, a regime where π_0 is comparable to $\pi_{\mathcal{D}}$, and a regime where π_0 is worse than $\pi_{\mathcal{D}}$. In these regimes, the appropriate degree of conservatism differs: if π_0 is already strong, the main risk is catastrophic forgetting or destructive exploration; if π_0 is weak relative to the data, then exploiting the dataset more aggressively may be beneficial.

Our contribution is not to dispute the utility of this decomposition, but to make explicit that its central comparator, $J(\pi_{\mathcal{D}})$, becomes ill-defined or insufficient once we leave the same-MDP assumption and allow an offline-to-online shift. Indeed, under shift we must separate the MDP that generated the dataset from the MDP on which we evaluate performance. The behavior policy $\pi_{\mathcal{D}}$ is defined only through the distribution of trajectories *in* $\mathcal{M}_{\mathrm{off}}$, while our goal concerns returns *in* $\mathcal{M}_{\mathrm{on}}$. The natural analogue of the classical comparator would be $J_{\mathrm{on}}(\pi_{\mathcal{D}})$, but this quantity is typically not available from $\mathcal{D}$ and, without additional assumptions, cannot be reliably inferred. There are two obstructions.

First, $\pi_{\mathcal{D}}$ is usually an *abstract mixture* (e.g., a nonstationary data-collection protocol, multiple human demonstrators, or an exploration policy that changes over time). Even in the same-MDP case, identifying a single stationary $\pi_{\mathcal{D}}$ may be a modeling convenience rather than a faithful description; under shift, this convenience becomes harmful if we attempt to treat $J_{\mathrm{on}}(\pi_{\mathcal{D}})$ as a well-posed number. Second, even if we postulate a stationary $\pi_{\mathcal{D}}$, the offline dataset encodes samples from P_{off} and R_{off}, whereas $J_{\mathrm{on}}(\pi_{\mathcal{D}})$

depends on P_{on} and R_{on}. Estimating returns in $\mathcal{M}_{\mathrm{on}}$ for a policy that we have never executed in $\mathcal{M}_{\mathrm{on}}$ is an off-policy evaluation problem *under dynamics shift.* Absent strong structure, this is not merely difficult; it is information-theoretically impossible under support mismatch, since the occupancy $d^{\mathrm{on}}_{\pi_{\mathcal{D}}}$ may concentrate on parts of the state–action space that are not covered by the online interaction we can afford, and the offline data are generated under the wrong transition kernel.

For these reasons, we replace the original regime comparator by a *transfer-aware* baseline that is directly measurable in the online MDP. Concretely, we restrict attention to policies that (a) can be produced from $\mathcal{D}$ without querying $\mathcal{M}_{\mathrm{on}}$, and (b) can be executed in $\mathcal{M}_{\mathrm{on}}$ and evaluated (with uncertainty) using a small number of online rollouts. We use two such policies: the offline-pretrained policy π_0, and a behavior cloning policy π_{BC} trained on $\mathcal{D}$. The latter plays the role of an operational proxy for $\pi_{\mathcal{D}}$: while π_{BC} is not the true data-collection policy, it is a stationary policy we can deploy online, and its performance $J_{\mathrm{on}}(\pi_{\mathrm{BC}})$ is therefore well-defined and estimable from online interaction.

This leads to the transferable offline baseline

$$J^*_{\mathrm{tr}} \;:=\; \max\{J_{\mathrm{on}}(\pi_0),\, J_{\mathrm{on}}(\pi_{\mathrm{BC}})\},$$

which serves two simultaneous purposes in our framework. From the stability perspective, J^*_{tr} is the floor against which we certify safety: we require that online fine-tuning never drops substantially below the better of the two available offline-derived policies. From the regime-identification perspective, the comparison between $J_{\mathrm{on}}(\pi_0)$ and $J_{\mathrm{on}}(\pi_{\mathrm{BC}})$ substitutes for the original comparison between $J(\pi_0)$ and $J(\pi_{\mathcal{D}})$. That is, in the presence of shift we no longer ask whether π_0 is better than the (hypothetical) behavior policy in the evaluation MDP; instead, we ask whether π_0 is better than a deployable imitation of the dataset behavior when both are executed in $\mathcal{M}_{\mathrm{on}}$. This replacement is conservative but principled: it uses only quantities that can be grounded in the online environment, and it avoids the unidentifiability inherent in $J_{\mathrm{on}}(\pi_{\mathcal{D}})$.

Once we adopt J^*_{tr} as the stability anchor, the stability–plasticity decomposition becomes algorithmically enforceable. Stability is maintained by ensuring that each deployed policy π_t has an online lower confidence bound exceeding $J^*_{\mathrm{tr}} - \varepsilon$, and by reverting to a prior policy (either π_0 or π_{BC}) whenever the bound fails. Plasticity then concerns how we exploit online data to improve beyond J^*_{tr}, and, crucially, how we reuse offline transitions in critic learning without incurring shift-induced Bellman bias. This is the point at which the shift-aware filtering/weighting mechanism enters: rather than relying on the offline dataset as if it were sampled from $\mathcal{M}_{\mathrm{on}}$, we treat it as a potentially misspecified source whose influence must be modulated by an estimated mismatch signal.

In summary, the source regimes are preserved in spirit—they still govern how conservative the online updates should be—but their central comparator must be altered under shift. By replacing $J(\pi_{\mathcal{D}})$ with $J_{\text{on}}(\pi_{\text{BC}})$ and anchoring stability at J^*_{tr}, we obtain a regime decomposition that is both operational (measurable in $\mathcal{M}_{\text{on}}$) and compatible with the theoretical obstacles posed by dynamics mismatch.

2 Problem Setup: a two-MDP model and transferable offline baselines

We formalize offline-to-online transfer under dynamics shift by positing two discounted Markov decision processes (MDPs) that share the same state–action spaces and discount, but may differ in their dynamics and rewards:

$$\mathcal{M}_{\text{off}} = (\mathcal{S}, \mathcal{A}, P_{\text{off}}, R_{\text{off}}, \gamma), \qquad \mathcal{M}_{\text{on}} = (\mathcal{S}, \mathcal{A}, P_{\text{on}}, R_{\text{on}}, \gamma), \qquad \gamma \in (0, 1).$$

The offline dataset is generated exclusively in $\mathcal{M}_{\text{off}}$. Concretely, $\mathcal{D}$ consists of a multiset of trajectories (or, equivalently for our purposes, transitions)

$$(s, a, r, s') \sim \text{rollouts in } \mathcal{M}_{\text{off}}$$

collected under an unknown data-collection mechanism. We denote by $\pi_{\mathcal{D}}$ an abstract behavior policy (possibly a mixture over time or across demonstrators) that induces the distribution of $\mathcal{D}$ in $\mathcal{M}_{\text{off}}$. We emphasize that $\pi_{\mathcal{D}}$ is not assumed to be known, stationary, or even well-defined as a single Markov policy; it is merely a notational handle for the data-generating process.

Our goal is to output a sequence of policies $\{\pi_t\}_{t=0}^{N}$—or, in its simplest form, a final policy π_N—that is evaluated on $\mathcal{M}_{\text{on}}$, after N steps of online interaction with $\mathcal{M}_{\text{on}}$. For any policy π, we write $J_{\text{on}}(\pi)$ for its expected discounted return in $\mathcal{M}_{\text{on}}$ (from the task's initial-state distribution, suppressed in notation). When needed, we also use the discounted occupancy measure d^{on}_{π} induced by π in $\mathcal{M}_{\text{on}}$.

Offline-derived policies available at time 0. We assume that an offline-pretrained policy π_0 is given. This policy may have been produced by any offline RL procedure applied to $\mathcal{D}$ (potentially with an associated critic), and we do not assume that π_0 is optimal or even safe under the online dynamics. In addition, we construct a behavior cloning (BC) policy π_{BC} trained by supervised learning on the state–action pairs in $\mathcal{D}$. The role of π_{BC} is operational rather than epistemic: it is a stationary policy that imitates the observed behavior in $\mathcal{D}$ and, crucially, can be executed in $\mathcal{M}_{\text{on}}$ to obtain genuine online performance estimates.

This pair of policies is the basis for our *transfer-aware* stability anchor. Since $J_{\text{on}}(\pi_{\mathcal{D}})$ is generally unidentifiable from $\mathcal{D}$ under shift, we refrain from using it as a comparator. Instead, we define the transferable offline baseline

$$J^*_{\mathrm{tr}} \;:=\; \max\bigl\{J_{\mathrm{on}}(\pi_0),\; J_{\mathrm{on}}(\pi_{\mathrm{BC}})\bigr\},$$

which is well-defined because both π_0 and π_{BC} are deployable in $\mathcal{M}_{\mathrm{on}}$. In practice, $J_{\mathrm{on}}(\pi_0)$ and $J_{\mathrm{on}}(\pi_{\mathrm{BC}})$ may be estimated by a modest number of online rollouts (or by an online off-policy evaluation routine equipped with uncertainty quantification). The salient point is that the baseline is anchored in $\mathcal{M}_{\mathrm{on}}$ rather than inferred from $\mathcal{D}$.

Stability and objective under online interaction. We consider online fine-tuning procedures that, for $t = 1, 2, \ldots, N$, deploy a policy π_{t-1} in $\mathcal{M}_{\mathrm{on}}$, collect transitions into an online buffer, and update the actor/critic to produce π_t. The overarching objective is to maximize $J_{\mathrm{on}}(\pi_N)$ (or $\max_{t\le N} J_{\mathrm{on}}(\pi_t)$), subject to a high-probability stability floor

$$\min_{0\le t\le N} J_{\mathrm{on}}(\pi_t) \;\ge\; J^*_{\mathrm{tr}} - \varepsilon,$$

for a user-specified slack $\varepsilon > 0$. The stability constraint reflects the stability–plasticity desideratum: plasticity is encoded by improvement toward the online optimum, whereas stability requires that online updates do not catastrophically underperform relative to the best available offline-derived reference.

Linear-MDP structure and shift magnitude. For theoretical analysis we impose a linear MDP model on $\mathcal{M}_{\mathrm{on}}$: there exists a known feature map $\phi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$ with $\|\phi(s,a)\|_2 \le 1$ and unknown parameters $w_R \in \mathbb{R}^d$ and $M_{\mathrm{on}} \in \mathbb{R}^{d\times d}$ such that

$$R_{\mathrm{on}}(s,a) = \langle w_R, \phi(s,a)\rangle, \qquad \text{and} \qquad \mathbb{E}\bigl[\phi(s',a') \mid s,a,\pi\bigr] = M_{\mathrm{on}}^\top \phi(s,a),$$

where $a' \sim \pi(\cdot \mid s')$. We likewise posit an offline transition parameter M_{off} that governs the analogous conditional feature expectation under $\mathcal{M}_{\mathrm{off}}$. We quantify the (feature-space) dynamics shift by

$$\Delta \;:=\; \|M_{\mathrm{on}} - M_{\mathrm{off}}\|_2.$$

The value of Δ is unknown, but online interaction permits the construction of a high-probability upper confidence bound $\widehat{\Delta}_t$ (and, more locally, upper bounds on $\|(M_{\mathrm{on}} - M_{\mathrm{off}})^\top \phi(s,a)\|_2$) via standard least-squares concentration arguments.

What it means to reuse offline transitions under shift. A central modeling decision is how to incorporate $\mathcal{D}$ during online fine-tuning. In the absence of shift, a common approach is to reuse offline transitions for

Bellman backups or TD regression as if they were drawn from the online environment. Under shift, this identification is incorrect: offline transitions are sampled according to P_{off}, whereas the critic we require for online control should reflect P_{on}. The induced error is not merely a finite-sample artifact; it manifests as a Bellman fixed-point bias whenever T_{off} is substituted for T_{on} in value estimation.

Accordingly, in our setup *offline transition reuse is conditional.* We allow the learning algorithm to draw from $\mathcal{D}$ during critic updates only after applying a *transfer filter* or *transfer weighting* that depends on an online-estimated mismatch signal. Formally, at online time t we associate to each offline state–action pair (s, a) an upper confidence mismatch proxy $U_t(s, a)$ satisfying (with high probability)

$$U_t(s,a) \;\geq\; \|(M_{\text{on}} - M_{\text{off}})^\top \phi(s,a)\|_2, \qquad \forall (s,a) \text{ appearing in } \mathcal{D}.$$

Given a threshold $\tau \geq 0$, we declare an offline transition admissible at time t if $U_t(s, a) \leq \tau$ (hard filtering), or else we assign it a weight that decays with $U_t(s, a)$ (soft weighting). The parameter τ thus directly encodes the maximum admitted misspecification in offline Bellman targets: smaller τ yields more conservative reuse (less bias, potentially higher variance), while larger τ yields more aggressive reuse (lower variance, potentially larger bias). This formalizes the notion that $\mathcal{D}$ is a *potentially misspecified* replay buffer whose influence must be modulated by online evidence about the current environment.

The resulting problem is therefore neither purely offline RL nor standard online RL with a replay buffer. It is an online control problem in $\mathcal{M}_{\text{on}}$ with (i) a transferable stability anchor J^*_{tr} derived from deployable offline policies, and (ii) an auxiliary, shift-filtered offline dataset $\mathcal{D}$ that may be used to accelerate critic learning only to the extent permitted by online-estimated mismatch. This is the setting in which our subsequent algorithmic choices and guarantees are stated.

3 Problem Setup: a two-MDP model and transferable offline baselines

We study offline-to-online transfer in the presence of a potential mismatch between the environment that generated the offline dataset and the environment faced during deployment. To make this explicit, we posit two discounted MDPs sharing state space, action space, and discount factor,

$$\mathcal{M}_{\text{off}} = (\mathcal{S}, \mathcal{A}, P_{\text{off}}, R_{\text{off}}, \gamma), \qquad \mathcal{M}_{\text{on}} = (\mathcal{S}, \mathcal{A}, P_{\text{on}}, R_{\text{on}}, \gamma), \qquad \gamma \in (0,1),$$

but allowing $(P_{\text{off}}, R_{\text{off}})$ and $(P_{\text{on}}, R_{\text{on}})$ to differ. Throughout, performance is measured *only* in $\mathcal{M}_{\text{on}}$.

Offline data and the behavior policy abstraction. The offline dataset $\mathcal{D}$ is generated entirely in $\mathcal{M}_{\text{off}}$. For our purposes, we treat $\mathcal{D}$ as a multiset of transitions (s, a, r, s') extracted from trajectories, where

$$(s, a, r, s') \sim \text{rollouts in } \mathcal{M}_{\text{off}}.$$

We attach to $\mathcal{D}$ an abstract data-generating mechanism $\pi_{\mathcal{D}}$, which should be read as a notational proxy rather than a well-specified stationary Markov policy. In particular, $\pi_{\mathcal{D}}$ may be non-stationary, history-dependent, or a mixture across demonstrators. This viewpoint is convenient because it separates what is *observable* (the empirical distribution of the collected transitions) from what is *unidentifiable* under shift (the true online performance of the data generator).

Online interaction and the performance criterion. We are granted N steps of interaction with $\mathcal{M}_{\text{on}}$. An algorithm produces a sequence of policies $\{\pi_t\}_{t=0}^{N}$ (or only the terminal π_N) by iterating: deploy π_{t-1} in $\mathcal{M}_{\text{on}}$, collect an online transition, and update. For any policy π, we denote its online discounted return by

$$J_{\text{on}}(\pi) \;:=\; \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k r_k \,\middle|\, \pi, \mathcal{M}_{\text{on}}\right],$$

with the initial-state distribution suppressed. When needed, we use the discounted occupancy measure d_π^{on} (over $\mathcal{S} \times \mathcal{A}$) induced by executing π in $\mathcal{M}_{\text{on}}$.

Our objective is to maximize $J_{\text{on}}(\pi_N)$ (or $\max_{t \le N} J_{\text{on}}(\pi_t)$) while preventing catastrophic regressions relative to a conservative reference available at time 0. We encode this stability requirement via a high-probability floor: for a user-specified slack $\varepsilon > 0$,

$$\min_{0 \le t \le N} J_{\text{on}}(\pi_t) \;\ge\; J_{\text{tr}}^* - \varepsilon,$$

with probability at least $1 - \delta$ for a chosen $\delta \in (0, 1)$.

Two deployable offline-derived policies at time 0. We assume the existence of an offline-pretrained policy π_0, produced by any offline RL method on $\mathcal{D}$. We do not require that π_0 be optimal (or even adequate) in $\mathcal{M}_{\text{on}}$, since dynamics shift may break the implicit modeling assumptions under which π_0 was trained.

In addition, we define a behavior cloning policy π_{BC} by supervised learning on the state–action pairs appearing in $\mathcal{D}$. We view π_{BC} as an operational baseline: while it may be suboptimal, it is a *well-defined stationary policy* that can be executed in $\mathcal{M}_{\text{on}}$ to obtain empirical online returns. This distinction matters because, under shift, the online value of the data-generating mechanism $\pi_{\mathcal{D}}$ is generally not recoverable from $\mathcal{D}$ without further coverage assumptions.

Transferable offline baseline. Because both π_0 and π_{BC} are deployable in $\mathcal{M}_{\mathrm{on}}$, their online returns are well-defined quantities. We therefore define the *transferable offline baseline*

$$J^*_{\mathrm{tr}} \;:=\; \max\bigl\{J_{\mathrm{on}}(\pi_0),\; J_{\mathrm{on}}(\pi_{\mathrm{BC}})\bigr\}.$$

In practice, $J_{\mathrm{on}}(\pi_0)$ and $J_{\mathrm{on}}(\pi_{\mathrm{BC}})$ may be estimated with a modest number of online rollouts (or via an online OPE routine equipped with uncertainty quantification), but conceptually J^*_{tr} is anchored in the true online environment rather than inferred from the offline data distribution. This choice is deliberate: under dynamics shift and potential support mismatch, even the sign of $J_{\mathrm{on}}(\pi_{\mathcal{D}}) - J_{\mathrm{off}}(\pi_{\mathcal{D}})$ may be unknowable from $\mathcal{D}$ alone.

Linear-MDP structure and a feature-space measure of shift. For theoretical guarantees we impose a linear MDP model on $\mathcal{M}_{\mathrm{on}}$. We assume a known feature map $\phi : \mathcal{S}\times\mathcal{A} \to \mathbb{R}^d$ satisfying $\|\phi(s,a)\|_2 \le 1$, and unknown parameters $w_R \in \mathbb{R}^d$ and $M_{\mathrm{on}} \in \mathbb{R}^{d\times d}$ such that

$$R_{\mathrm{on}}(s,a) = \langle w_R, \phi(s,a)\rangle, \qquad \mathbb{E}\bigl[\phi(s',a') \mid s,a,\pi\bigr] = M_{\mathrm{on}}^\top \phi(s,a),$$

where $a' \sim \pi(\cdot \mid s')$. The second condition is a compact way to encode linear transition structure for policy evaluation/control in feature space. We likewise posit an offline transition parameter M_{off} governing the analogous conditional feature expectation under $\mathcal{M}_{\mathrm{off}}$, and we quantify dynamics shift by

$$\Delta \;:=\; \|M_{\mathrm{on}} - M_{\mathrm{off}}\|_2.$$

Although Δ is unknown, online data allow us to build confidence sets for M_{on} (hence for $M_{\mathrm{on}} - M_{\mathrm{off}}$) by least-squares regression and standard concentration inequalities, yielding time-dependent upper bounds $\widehat{\Delta}_t$ and, more importantly for transfer decisions, state–action dependent mismatch proxies.

What it means to reuse offline transitions under shift. A recurring algorithmic pattern in offline-to-online fine-tuning is to incorporate $\mathcal{D}$ into critic learning via TD regression or fitted Q-iteration, treating offline transitions as additional samples for Bellman backups. Under shift, this identification is not benign: offline transitions are governed by P_{off}, whereas the Bellman operator relevant for evaluation/control in $\mathcal{M}_{\mathrm{on}}$ depends on P_{on}. To make the mismatch explicit, for a fixed policy π we may write Bellman evaluation operators T^π_{off} and T^π_{on}, and observe that substituting T^π_{off} for T^π_{on} perturbs the contraction fixed point by an amount that does not vanish with more offline data when $P_{\mathrm{off}} \neq P_{\mathrm{on}}$. Thus, *unfiltered* reuse of $\mathcal{D}$ can introduce an irreducible bias in the learned critic, which in turn can misdirect policy improvement.

Accordingly, we treat $\mathcal{D}$ as a replay buffer that is potentially misspecified for the online task, and we only reuse offline transitions after applying a transfer-aware filter or weight. Concretely, at online time t we associate to each offline state–action pair (s, a) an upper-confidence mismatch proxy $U_t(s, a)$ such that, on a high-probability event,

$$U_t(s,a) \;\geq\; \|(M_{\text{on}} - M_{\text{off}})^\top \phi(s,a)\|_2, \qquad \forall (s,a) \text{ appearing in } \mathcal{D}.$$

We then impose an admissibility rule parameterized by a mismatch threshold $\tau \geq 0$:

$$w_t(s,a) \;:=\; \mathbf{1}\{U_t(s,a) \leq \tau\} \quad \text{(hard filter)}, \qquad \text{or more generally} \qquad w_t(s,a) \in [0,1] \text{ decreasing in } U$$

The effect is to construct critic targets from a mixture of online transitions and *transfer-certified* offline transitions. The threshold τ plays a dual role: it bounds the maximum misspecification admitted into offline Bellman targets (hence controlling bias), while also controlling how much offline data are available for regression (hence controlling variance). This bias–variance tradeoff is intrinsic to transfer under shift: setting $\tau = 0$ yields the most conservative behavior (offline reuse only when dynamics are certified to match at the feature level), whereas larger τ enables more aggressive reuse at the cost of a larger residual shift term.

Finally, we emphasize the separation between *dataset usage* and *policy anchoring*. The baseline J^*_{tr} is defined solely from deployable policies evaluated on $\mathcal{M}_{\text{on}}$, and is therefore meaningful even when $\mathcal{D}$ is severely mismatched. By contrast, the influence of $\mathcal{D}$ on critic learning is explicitly gated by the mismatch proxies $U_t(s, a)$ and the threshold τ. This separation is the structural feature that enables both the stability floor (through conservative deployment relative to J^*_{tr}) and the near-optimality analysis (through controlling the shift-induced Bellman bias contributed by admitted offline transitions).

4 Transfer-Aware Regimes: priors, mismatch, and prescriptions

Our algorithmic decisions at online time 0 are governed by two orthogonal questions: (i) *which offline-derived policy is the safer anchor for deployment*, and (ii) *to what extent is the offline dataset $\mathcal{D}$ transferable as a source of Bellman targets for critic learning*. We formalize these questions as two axes, yielding a small set of regimes with distinct prescriptions for policy regularization, critic training, and exploration.

Axis I (policy-anchor axis): π_0 versus π_{BC} on $\mathcal{M}_{\mathrm{on}}$. Since both π_0 and π_{BC} are deployable, we can compare them using online interaction. Define the online return gap

$$\Delta J_{\mathrm{tr}} \;:=\; J_{\mathrm{on}}(\pi_0) - J_{\mathrm{on}}(\pi_{\mathrm{BC}}).$$

In principle, the sign of ΔJ_{tr} determines which policy should serve as the initial stabilizing prior. In practice, ΔJ_{tr} is estimated from a small number of online rollouts and thus is noisy; accordingly, we introduce an indifference margin $\delta_0 > 0$ and treat the cases $|\Delta J_{\mathrm{tr}}| \le \delta_0$ as statistically ambiguous. When ambiguity occurs, we adopt a *mixture prior* (e.g., a logit-averaged mixture) or choose the prior that is simpler to stabilize (often π_{BC} due to lower variance), while continuing to refine the estimate online. Formally, at time t we define

$$\pi_{\mathrm{prior},t} \;\in\; \arg\max_{\pi\in\{\pi_0,\pi_{\mathrm{BC}}\}} \widehat{J}_{\mathrm{on},t}(\pi),$$

with ties (or near-ties) resolved by a conservative rule, and we penalize policy updates away from $\pi_{\mathrm{prior},t}$ via a KL term with weight λ_t.

Axis II (dataset-transfer axis): mismatch/overlap as an online-measurable quantity. Even when a policy is safe to deploy, it does not follow that $\mathcal{D}$ is safe to reuse for Bellman backups. We therefore distinguish *policy anchoring* (stability) from *dataset anchoring* (critic sample reuse). The latter depends on how much of $\mathcal{D}$ is *certifiably compatible* with $\mathcal{M}_{\mathrm{on}}$ under our mismatch proxies. Given $U_t(s,a)$ and threshold τ, define the admitted set and its mass

$$\mathcal{D}_t^{\mathrm{adm}} \;:=\; \{(s,a,r,s') \in \mathcal{D} :\; U_t(s,a) \le \tau\}, \qquad \alpha_t \;:=\; \frac{|\mathcal{D}_t^{\mathrm{adm}}|}{|\mathcal{D}|} \in [0,1].$$

We interpret α_t as an *overlap fraction*: large α_t indicates that a substantial portion of the offline coverage is transfer-certified (hence useful for variance reduction in critic regression), whereas small α_t indicates that offline replay is either largely inadmissible or would induce large misspecification bias if admitted. Complementarily, we track a scalar mismatch indicator such as $\widehat{\Delta}_t$ (global) or $\max_{(s,a)\in\mathcal{D}} U_t(s,a)$ (worst-case). For regime identification, it suffices to binarize this axis into

$$\text{low mismatch / high overlap: } \alpha_t \ge \alpha_{\min}, \qquad \text{high mismatch / low overlap: } \alpha_t < \alpha_{\min},$$

for a chosen $\alpha_{\min} \in (0,1)$ (or an equivalent rule based on $\widehat{\Delta}_t$), with the understanding that the classification may evolve over time as online confidence sets tighten.

A 2×2 regime map. Combining the two axes yields four regimes. For compactness we present them as a table, with each cell prescribing (i) the prior policy $\pi_{\text{prior},t}$, (ii) the strength schedule λ_t of the KL penalty, (iii) the aggressiveness of offline replay as governed by τ (or soft weights), and (iv) the degree of *plasticity* in online updates (actor step sizes, exploration, and rollback frequency).

	low mismatch / high overlap	high mismatch / low
$J_{\text{on}}(\pi_0) \geq J_{\text{on}}(\pi_{\text{BC}})$	**Regime I (trust π_0, reuse $\mathcal{D}$)**	**Regime II (trust π_0, d**
$J_{\text{on}}(\pi_0) < J_{\text{on}}(\pi_{\text{BC}})$	**Regime III (trust π_{BC}, reuse $\mathcal{D}$ cautiously)**	**Regime IV (trust π_{BC}, r**

Regime I: π_0 is better online and $\mathcal{D}$ is transferable. Here we are in the favorable case: the offline RL policy is already superior to cloning in $\mathcal{M}_{\text{on}}$, and a nontrivial fraction of offline transitions are certified by $U_t(s, a) \leq \tau$. We therefore set $\pi_{\text{prior},t} = \pi_0$ and keep λ_t moderate: it serves to prevent abrupt departures from π_0 while allowing steady improvement. For critic learning, we choose τ relatively large (or equivalently use a soft weighting with a slower decay), since the residual shift bias is controlled by the low-mismatch assumption and the main benefit of $\mathcal{D}$ is variance reduction. Plasticity can be high: we can run multiple critic updates per online step using $\mathcal{D}_t^{\text{adm}}$, and allow larger actor steps, because rollback to π_0 remains available if the stability monitor detects degradation.

Regime II: π_0 is better online but $\mathcal{D}$ is not transferable. This case reflects a common situation: an offline policy generalizes acceptably to the online environment, but the offline transitions are mismatched enough that naive replay would corrupt Bellman targets. We again choose $\pi_{\text{prior},t} = \pi_0$, but we increase λ_t (or make it adaptive) to constrain updates more tightly, since the critic must be learned primarily from online data and is therefore noisier early on. Critic training should either (i) exclude $\mathcal{D}$ entirely (set τ small so that $\alpha_t \approx 0$), or (ii) include only a narrow slice of $\mathcal{D}$ with stringent filtering, effectively using offline data as a regularizer rather than as a source of Bellman backup mass. Plasticity is moderate: exploration should be cautious (e.g., small entropy bonuses, limited deviation from π_0), and the stability floor is enforced primarily by conservative policy improvement rather than by trusting the critic extrapolations supported by offline replay.

Regime III: π_{BC} is better online but $\mathcal{D}$ is transferable. When cloning outperforms π_0 in $\mathcal{M}_{\text{on}}$, we interpret π_0 as potentially overfit to idiosyncrasies of $\mathcal{M}_{\text{off}}$ or as having exploited spurious value estimates. Nevertheless, if α_t is large, the dataset is still informative about online dynamics at the feature level. We therefore set $\pi_{\text{prior},t} = \pi_{\text{BC}}$ for stability, but continue to reuse $\mathcal{D}_t^{\text{adm}}$ for critic learning, with an important modification: we bias the actor updates toward *incremental* improvements over π_{BC} (larger λ_t than in

Regime I), and we may initialize the actor from π_{BC} even if π_0 is available. In this regime, dataset anchoring is beneficial, but policy anchoring must be conservative because the offline RL policy has empirically failed the online comparison.

Regime IV: π_{BC} is better online and $\mathcal{D}$ is not transferable. This is the most adversarial regime: offline replay is unreliable and the offline RL policy is inferior to cloning in the online environment. The prescription is correspondingly conservative. We set $\pi_{\text{prior},t} = \pi_{\text{BC}}$ and maintain a large λ_t until sufficient online data are accumulated to justify deviations; early updates are therefore close to behavior-regularized online RL. We set τ small, making α_t negligible, and treat $\mathcal{D}$ only as a source of representation learning or auxiliary regularization (if at all), not as a source of Bellman targets. Plasticity is low initially (small actor steps, frequent evaluation/LCB checks), then gradually increases as online uncertainty decreases. In effect, TWBR reduces to a stability-constrained online learner warm-started from π_{BC}.

Dynamics of regime transitions and practical triggers. The regime classification is not static. As t increases, two quantities typically change: the confidence radius underlying U_t shrinks, increasing α_t for a fixed τ, and the estimated return gap $\widehat{\Delta J}_{\text{tr},t}$ concentrates, clarifying the choice of $\pi_{\text{prior},t}$. We therefore treat the regime as a *state* updated online: whenever $\pi_{\text{prior},t}$ switches (because $\widehat{J}_{\text{on},t}(\pi_0)$ overtakes $\widehat{J}_{\text{on},t}(\pi_{\text{BC}})$ or vice versa), we reset λ_t upward to avoid abrupt distributional shifts in the replay buffer; whenever α_t crosses $\alpha_{\min}$ upward, we gradually relax τ (or soften the weights) so that the critic can benefit from offline variance reduction without introducing a discontinuous bias. These transitions are governed by the same principle: *policy anchoring is decided by online return comparisons, while dataset anchoring is decided by mismatch certification.* The next section formalizes why this separation is necessary by exhibiting the failure mode of naive offline replay under shift in terms of Bellman-operator mismatch.

5 Failure of Naive Offline Replay Under Shift: Bellman-operator mismatch and bias

We now isolate the precise failure mode that necessitates our separation between *policy anchoring* (deploying π_0 or π_{BC}) and *dataset anchoring* (reusing $\mathcal{D}$ for Bellman backups). The essential obstruction is that Bellman operators depend on the transition kernel; thus, when $P_{\text{off}} \neq P_{\text{on}}$, treating offline transitions as if they were sampled from P_{on} induces a systematic (non-vanishing) bias in critic learning. This bias persists even with infinite offline data and manifests as value loss on $\mathcal{M}_{\text{on}}$.

Bellman-operator mismatch. Fix a stationary policy π. Let T^π_{on} and T^π_{off} denote the policy-evaluation Bellman operators on $\mathcal{M}_{\text{on}}$ and $\mathcal{M}_{\text{off}}$, respectively:

$$\begin{aligned}(T^\pi_{\text{on}}Q)(s,a) &:= R_{\text{on}}(s,a) + \gamma\,\mathbb{E}_{s'\sim P_{\text{on}}(\cdot|s,a),\,a'\sim\pi(\cdot|s')}\big[Q(s',a')\big],\\ (T^\pi_{\text{off}}Q)(s,a) &:= R_{\text{off}}(s,a) + \gamma\,\mathbb{E}_{s'\sim P_{\text{off}}(\cdot|s,a),\,a'\sim\pi(\cdot|s')}\big[Q(s',a')\big].\end{aligned}$$

For the present discussion we suppress reward shift by assuming $R_{\text{on}} = R_{\text{off}}$ (or, more generally, we may add an additive $\|R_{\text{on}} - R_{\text{off}}\|_\infty$ term everywhere below). Define the transition shift in total variation as

$$\Delta_P := \sup_{s,a} \big\|P_{\text{on}}(\cdot \mid s,a) - P_{\text{off}}(\cdot \mid s,a)\big\|_1.$$

Lemma 5.1 (Operator difference bound)**.** *For any bounded $Q : \mathcal{S}\times\mathcal{A}\to\mathbb{R}$ and any policy π,*

$$\big\|T^\pi_{\text{on}}Q - T^\pi_{\text{off}}Q\big\|_\infty \le \gamma\,\Delta_P\,\|Q\|_\infty.$$

The proof is immediate from the dual characterization of total variation: for each (s,a),

$$\big|\mathbb{E}_{P_{\text{on}}}[f(s')] - \mathbb{E}_{P_{\text{off}}}[f(s')]\big| \le \Delta_P\,\|f\|_\infty, \qquad f(s') := \mathbb{E}_{a'\sim\pi(\cdot|s')}Q(s',a').$$

Fixed-point bias induced by offline replay. Let Q^π_{on} and Q^π_{off} denote the unique fixed points of T^π_{on} and T^π_{off}. Since both operators are γ-contractions in $\|\cdot\|_\infty$, a standard perturbation argument yields

$$\begin{aligned}\|Q^\pi_{\text{on}} - Q^\pi_{\text{off}}\|_\infty &= \big\|T^\pi_{\text{on}}Q^\pi_{\text{on}} - T^\pi_{\text{off}}Q^\pi_{\text{off}}\big\|_\infty\\ &\le \big\|T^\pi_{\text{on}}Q^\pi_{\text{on}} - T^\pi_{\text{on}}Q^\pi_{\text{off}}\big\|_\infty + \big\|T^\pi_{\text{on}}Q^\pi_{\text{off}} - T^\pi_{\text{off}}Q^\pi_{\text{off}}\big\|_\infty\\ &\le \gamma\,\|Q^\pi_{\text{on}} - Q^\pi_{\text{off}}\|_\infty + \gamma\,\Delta_P\,\|Q^\pi_{\text{off}}\|_\infty,\end{aligned}$$

hence

$$\|Q^\pi_{\text{on}} - Q^\pi_{\text{off}}\|_\infty \le \frac{\gamma}{1-\gamma}\,\Delta_P\,\|Q^\pi_{\text{off}}\|_\infty \le \frac{\gamma}{(1-\gamma)^2}\,\Delta_P\,R_{\max},$$

where $R_{\max} := \|R\|_\infty$ and we used $\|Q^\pi_{\text{off}}\|_\infty \le R_{\max}/(1-\gamma)$. In particular, any critic-learning procedure that converges (with abundant offline data) to the *offline* fixed point Q^π_{off} while being deployed in $\mathcal{M}_{\text{on}}$ necessarily incurs an irreducible $\Theta(\Delta_P/(1-\gamma)^2)$ value-scale discrepancy. This discrepancy is *not* a statistical error; it is a misspecification bias originating from the wrong Bellman operator.

Bias in the linear MDP parameterization. In the linear MDP setting, mismatch admits a more refined, feature-local expression. Writing the conditional expectation of next-step features under π as

$$\mu^{\pi}_{\text{on}}(s,a) := \mathbb{E}\big[\phi(s',a') \mid s,a,\pi,\mathcal{M}_{\text{on}}\big] = M_{\text{on}}^{\top}\phi(s,a), \qquad \mu^{\pi}_{\text{off}}(s,a) := M_{\text{off}}^{\top}\phi(s,a),$$

we have for any linear critic $Q_w(s,a) := \langle w, \phi(s,a)\rangle$,

$$\begin{aligned}(T^{\pi}_{\text{on}}Q_w)(s,a) - (T^{\pi}_{\text{off}}Q_w)(s,a) &= \gamma\left\langle w, \mu^{\pi}_{\text{on}}(s,a) - \mu^{\pi}_{\text{off}}(s,a)\right\rangle \\ &= \gamma\left\langle w, (M_{\text{on}} - M_{\text{off}})^{\top}\phi(s,a)\right\rangle.\end{aligned}$$

Consequently,

$$\left|(T^{\pi}_{\text{on}}Q_w)(s,a) - (T^{\pi}_{\text{off}}Q_w)(s,a)\right| \;\le\; \gamma\,\|w\|_2\,\|(M_{\text{on}} - M_{\text{off}})^{\top}\phi(s,a)\|_2 \;\le\; \gamma\,\|w\|_2\,\Delta.$$

This inequality shows why a *global* shift magnitude Δ is not, by itself, sufficient for safe reuse of $\mathcal{D}$: even when Δ is moderate, the critic bias depends on where $\phi(s,a)$ lies relative to the shift direction, motivating a *state–action dependent* certificate $U_t(s,a)$ and a threshold τ.

An explicit counterexample family (matching lower-bound phenomenon). We now exhibit a family of MDP pairs in which naive offline replay can force $\Omega(\Delta/(1-\gamma)^2)$ suboptimality, even with infinite offline data. Consider an MDP with a single decision state s and two absorbing states g (good) and b (bad). The action set is $\mathcal{A} = \{1,2\}$. From g the agent remains in g forever and receives reward 1 each step; from b the agent remains in b forever and receives reward 0 each step. From the decision state s, the immediate reward is 0.

Define the offline MDP $\mathcal{M}_{\text{off}}$ by setting, for both actions $a \in \{1,2\}$,

$$P_{\text{off}}(g \mid s,a) = 1, \qquad P_{\text{off}}(b \mid s,a) = 0.$$

Thus, under $\mathcal{M}_{\text{off}}$ both actions appear equally optimal and yield value $V_{\text{off}}(s) = \gamma/(1-\gamma)$.

Now define two *online* MDPs $\mathcal{M}^{(+)}_{\text{on}}$ and $\mathcal{M}^{(-)}_{\text{on}}$ which share the same rewards as above and differ only in the transition from s:

$$\mathcal{M}^{(+)}_{\text{on}}: \quad P(g \mid s,1) = 1, \quad P(g \mid s,2) = 1-\eta, \quad P(b \mid s,2) = \eta,$$

$$\mathcal{M}^{(-)}_{\text{on}}: \quad P(g \mid s,2) = 1, \quad P(g \mid s,1) = 1-\eta, \quad P(b \mid s,1) = \eta,$$

where $\eta \in (0,1)$ is a shift parameter (we may identify η with Δ up to constants in an appropriate feature embedding; in total variation, $\Delta_P \ge \eta$). In $\mathcal{M}^{(+)}_{\text{on}}$, action 1 is optimal; in $\mathcal{M}^{(-)}_{\text{on}}$, action 2 is optimal. The optimal online value at s is

$$V^*_{\text{on}}(s) = \frac{\gamma}{1-\gamma}.$$

If the agent instead chooses the *wrong* action (the one with η probability of falling into b), the value becomes

$$V_{\mathrm{on}}^{\mathrm{wrong}}(s) = \gamma \left((1-\eta) \cdot \frac{1}{1-\gamma} + \eta \cdot 0 \right) = \frac{\gamma(1-\eta)}{1-\gamma},$$

and therefore

$$V_{\mathrm{on}}^{*}(s) - V_{\mathrm{on}}^{\mathrm{wrong}}(s) = \frac{\gamma\eta}{1-\gamma} = \Omega\left(\frac{\eta}{1-\gamma}\right).$$

To obtain the canonical $\Omega(\eta/(1-\gamma)^2)$ scaling, we can equivalently shift *the value scale* by replacing the good-state reward 1 with $(1-\gamma)^{-1}$ (still bounded for fixed γ away from 1), in which case the good-state value is $(1-\gamma)^{-2}$ and the gap becomes $\Omega(\eta/(1-\gamma)^2)$. This normalization is standard when expressing lower bounds in terms of the horizon scale $(1-\gamma)^{-1}$.

The crucial point is informational: the offline dataset $\mathcal{D}$ generated from $\mathcal{M}_{\mathrm{off}}$ is *identical* regardless of whether the true online environment is $\mathcal{M}_{\mathrm{on}}^{(+)}$ or $\mathcal{M}_{\mathrm{on}}^{(-)}$, since both share the same $\mathcal{M}_{\mathrm{off}}$. Hence, any algorithm that performs Bellman backups on $\mathcal{D}$ *as if* its transitions were from P_{on} and does not incorporate an explicit online mismatch correction cannot, before seeing informative online transitions, distinguish which action is unsafe. In particular, with infinite offline data it will learn a critic consistent with P_{off}, for which both actions at s are equivalent; any deterministic tie-breaking selects one action, which is necessarily suboptimal in one of the two online instances. This establishes a worst-case value loss proportional to the shift magnitude, matching the lower-bound phenomenon that an additive shift term is information-theoretically unavoidable without mismatch detection.

Implication for algorithm design. The above counterexample is deliberately elementary, but it captures the general mechanism: Bellman-target bias arises from substituting T_{off} for T_{on}. The linear-MDP mismatch expression makes clear that the safest remedy is not to *forbid* offline data, but to *gate* it by an online-certified bound on $(M_{\mathrm{on}} - M_{\mathrm{off}})^{\top}\phi(s,a)$, thereby controlling the operator bias locally. This observation leads directly to the design of Transfer-Weighted Bellman Replay in the next section: we estimate a confidence set for M_{on}, convert it into per-sample mismatch certificates $U_t(s,a)$, and filter/weight offline transitions so that admitted Bellman backups incur at most a controlled residual bias τ.

6 Algorithm: Transfer-Weighted Bellman Replay (TWBR)

We now formalize the fine-tuning procedure suggested by the preceding mismatch analysis. The design requirement is twofold: (i) we must control the

Bellman-target bias induced by reusing $\mathcal{D}$ under $P_{\text{on}} \neq P_{\text{off}}$, and (ii) we must maintain a stability floor relative to the best *transferable* offline policy, namely $J^*_{\text{tr}} := \max\{J_{\text{on}}(\pi_0), J_{\text{on}}(\pi_{\text{BC}})\}$, up to slack ε. TWBR implements (i) by assigning each offline transition a time-varying *transfer certificate* $U_t(s, a)$ and using only those samples whose certified mismatch is below a threshold τ (or by downweighting them smoothly), and it implements (ii) by conservative actor updates with an explicit rollback rule based on a lower confidence bound for J_{on}.

Estimating online dynamics and a confidence set. In the linear MDP model, the quantity governing Bellman mismatch for linear critics is $(M_{\text{on}} - M_{\text{off}})^\top \phi(s, a)$. Thus, TWBR maintains an online estimate $M_{\text{on},t}$ of M_{on} and a high-probability confidence set around it. Concretely, for each online step $i \leq t$ we record $z_i := \phi(s_i, a_i)$ and a "next-feature" target

$$y_i \;:=\; \phi(s'_i, a'_i), \qquad a'_i \sim \pi_{i-1}(\cdot \mid s'_i),$$

so that $\mathbb{E}[y_i \mid s_i, a_i, \pi_{i-1}] = M_{\text{on}}^\top z_i$ holds by assumption. With ridge parameter $\lambda > 0$, define the online Gram matrix

$$V_t \;:=\; \lambda I + \sum_{i=1}^{t} z_i z_i^\top,$$

and let $M_{\text{on},t}$ be the regularized least-squares solution minimizing $\sum_{i=1}^{t} \|y_i - M^\top z_i\|_2^2 + \lambda \|M\|_F^2$. Standard self-normalized concentration yields a (time-uniform) event $\mathcal{E}$ of probability at least $1 - \delta$ on which, for all t,

$$\|(M_{\text{on},t} - M_{\text{on}})^\top x\|_2 \;\leq\; \beta_t \|x\|_{V_t^{-1}} \qquad \text{for all } x \in \mathbb{R}^d, \tag{1}$$

for an explicit radius $\beta_t = \widetilde{O}(\sqrt{d \log(1/\delta)})$ depending on noise bounds and λ. We emphasize that (1) is the only property needed to certify per-sample mismatch; the specific estimator may be replaced by any procedure producing such a confidence relation.

Offline mismatch certificates and transfer weights. Given (1) and a fixed offline transition parameter M_{off} (estimated once from $\mathcal{D}$ by the analogous regression), we define the per-sample certificate

$$U_t(s, a) \;:=\; \big\|(M_{\text{on},t} - M_{\text{off}})^\top \phi(s, a)\big\|_2 \;+\; \beta_t \|\phi(s, a)\|_{V_t^{-1}}. \tag{2}$$

On the event $\mathcal{E}$, (2) upper bounds the true feature-space mismatch:

$$\big\|(M_{\text{on}} - M_{\text{off}})^\top \phi(s, a)\big\|_2 \;\leq\; U_t(s, a),$$

by the triangle inequality and (1). TWBR then defines a transfer weight for each offline sample $(s, a, r, s') \in \mathcal{D}$ by the hard filter

$$w_t(s, a) \;:=\; \mathbf{1}\{U_t(s, a) \leq \tau\}, \tag{3}$$

or, in a smooth variant useful in practice,

$$w_t(s, a) \;:=\; \exp\{-\kappa\, U_t(s, a)\}, \tag{4}$$

with temperature $\kappa > 0$. The role of τ is algorithmic: it caps the admissible Bellman mismatch when treating an offline transition as a surrogate for an online transition. The slack $\beta_t \|\phi\|_{V_t^{-1}}$ has a complementary role: it shrinks as online coverage grows, so that samples are admitted not only when the estimated shift is small, but also when our uncertainty about the online shift is small.

Critic learning with transfer-weighted replay. TWBR updates a critic on a mixture of online and (weighted) offline transitions. For clarity we describe the linear-critic instantiation $Q_w(s, a) = \langle w, \phi(s, a)\rangle$, although the same weighting logic applies to general function approximation. Let $\mathcal{B}_t$ denote the online buffer at time t and let $\mathcal{D}_t := \{(s, a, r, s') \in \mathcal{D} : w_t(s, a) > 0\}$ be the admitted offline subset. A generic fitted Bellman regression step updates w by minimizing

$$\sum_{(s,a,r,s')\in\mathcal{B}_t} \Big(\langle w, \phi(s, a)\rangle - \big[r + \gamma\, \widehat{V}(s')\big]\Big)^2 + \alpha \sum_{(s,a,r,s')\in\mathcal{D}} w_t(s, a)\Big(\langle w, \phi(s, a)\rangle - \big[r + \gamma\, \widehat{V}(s')\big]\Big)^2 + \rho\|w\|_2^2, \tag{5}$$

where $\widehat{V}(s')$ is a target computed from the current policy (policy evaluation) or via $\max_{a'}\langle w, \phi(s', a')\rangle$ (control), α controls the relative offline mass, and ρ is an additional regularization parameter. The filter (3) ensures that any residual Bellman bias contributed by offline transitions is bounded (in the sense made explicit by the mismatch inequalities in the previous section) by a term scaling with τ rather than with the ambient shift Δ.

Regime-dependent policy priors and actor regularization. We maintain two candidate priors: the given offline-pretrained π_0 and the behavior cloning policy π_{BC}. Since the online shift may invalidate one but not the other, we select a time-dependent prior

$$\pi_{\mathrm{prior},t} \in \{\pi_0, \pi_{\mathrm{BC}}\} \qquad \text{with} \qquad \pi_{\mathrm{prior},t} \in \arg \max_{\pi\in\{\pi_0,\pi_{\mathrm{BC}}\}} \widehat{J}_{\mathrm{on}}^{\mathrm{LCB}}(\pi),$$

where $\widehat{J}_{\mathrm{on}}^{\mathrm{LCB}}(\pi)$ is an online lower confidence bound (defined below). The actor update then solves, approximately, a KL-regularized improvement step:

$$\pi_t \;\approx\; \arg\max_{\pi}\; \mathbb{E}_{s\sim\widehat{d}^{\mathrm{on}}}\Big[\mathbb{E}_{a\sim\pi(\cdot|s)}\widehat{Q}_t(s, a)\Big] - \lambda_t\, \mathbb{E}_{s\sim\widehat{d}^{\mathrm{on}}}\big[D_{\mathrm{KL}}(\pi(\cdot \mid s)\, \|\, \pi_{\mathrm{prior},t}(\cdot \mid s))\big], \tag{6}$$

with λ_t chosen adaptively. Intuitively, when the stability monitor indicates risk, λ_t is increased (and the step size is decreased) so that the update becomes conservative and remains close to $\pi_{\mathrm{prior},t}$. Conversely, when certified performance is comfortably above the floor and online coverage is sufficient (so that U_t is small on a large portion of $\mathcal{D}$), λ_t may be reduced to permit faster improvement.

Stability monitor, lower confidence bounds, and rollback. To enforce the constraint $\min_{0 \le t \le N} J_{\mathrm{on}}(\pi_t) \ge J^*_{\mathrm{tr}} - \varepsilon$ with high probability, TWBR deploys only policies whose performance is certified by a lower confidence bound. One concrete implementation uses periodic on-policy evaluation rollouts of π_t in $\mathcal{M}_{\mathrm{on}}$ producing an empirical return $\widehat{J}_{\mathrm{on}}(\pi_t)$ and a concentration-based deviation term c_t such that

$$\widehat{J}^{\mathrm{LCB}}_{\mathrm{on}}(\pi_t) \;:=\; \widehat{J}_{\mathrm{on}}(\pi_t) - c_t \;\le\; J_{\mathrm{on}}(\pi_t) \quad \text{w.p.} \;\ge 1 - \delta.$$

The deployment rule is then:

$$\text{if } \widehat{J}^{\mathrm{LCB}}_{\mathrm{on}}(\pi_t) \ge J^*_{\mathrm{tr}} - \varepsilon, \text{ deploy } \pi_t; \qquad \text{else revert to } \pi_{\mathrm{prior},t} \text{ and increase } \lambda_t.$$

In addition, upon triggering we may tighten transfer by decreasing τ (or setting $\alpha = 0$ in (5) temporarily), ensuring that subsequent critic updates cannot be dominated by potentially misaligned offline backups during an identified high-risk regime.

Offline usage modes. It is useful to separate three modes that TWBR interpolates between via $(\tau, \alpha, \lambda_t)$: (1) *online-only* learning ($\alpha = 0$), which eliminates shift bias but pays higher variance; (2) *shift-filtered replay* ($\alpha > 0$ with (3) or (4)), which reduces variance while incurring at most τ-controlled bias; and (3) *policy anchoring without dataset anchoring*, in which $\mathcal{D}$ is used only to define π_{BC} (and possibly as a behavioral regularizer), while Bellman targets rely primarily on online transitions. The algorithm transitions among these modes according to the certified mismatch and stability signals: as online confidence increases, the admitted set $\mathcal{D}_t$ expands automatically through (2), while the stability monitor guarantees that any aggressive update can be vetoed in favor of the best empirically transferable baseline.

7 Main Theorems

We collect the principal guarantees implied by the mismatch-controlled replay mechanism and the conservative deployment rule. Throughout, we work on the time-uniform high-probability event $\mathcal{E}$ on which the online confidence relation (1) holds simultaneously for all $t \le N$, and we assume bounded rewards (e.g. $|R_{\mathrm{on}}(s,a)| \le 1$) and conditionally sub-Gaussian noise in the linear regression targets so that $\mathbb{P}(\mathcal{E}) \geq 1 - \delta$ by standard self-normalized martingale concentration. For readability, we suppress logarithmic factors via $\widetilde{O}(\cdot)$.

7.1 Bellman mismatch and the necessity of transfer control

The first result formalizes the central obstruction: reusing offline transitions as if they were drawn from P_{on} induces an *operator bias* that persists even with infinite offline data. While we implement mismatch control in feature space, the phenomenon is most transparent in total variation.

Theorem 7.1 (Bellman mismatch bias bound). *Fix a policy π and let T_{on} and T_{off} denote its Bellman evaluation operators under $\mathcal{M}_{\text{on}}$ and $\mathcal{M}_{\text{off}}$, respectively. Assume rewards are shared (or that reward shift is treated separately) and suppose*

$$\Delta_P \;:=\; \sup_{s,a} \big\| P_{\text{on}}(\cdot \mid s,a) - P_{\text{off}}(\cdot \mid s,a) \big\|_1 \;<\; \infty.$$

Then for any bounded Q,

$$\| T_{\text{on}} Q - T_{\text{off}} Q \|_\infty \;\leq\; \gamma \Delta_P \|Q\|_\infty.$$

Consequently, if an algorithm computes (exactly or approximately) the fixed point of T_{off} while deploying in $\mathcal{M}_{\text{on}}$, the induced evaluation error is lower bounded as

$$\|Q^\pi_{\text{on}} - Q^\pi_{\text{off}}\|_\infty \;=\; \Omega\!\left(\frac{\gamma \Delta_P}{1-\gamma}\right),$$

and the corresponding value loss scales as $\Omega(\Delta_P/(1-\gamma)^2)$ in the worst case.

The proof is a direct operator-difference bound via total variation and the contraction property of Bellman evaluation. The message is that controlling statistical error alone is insufficient; a transfer mechanism must also control *misspecification error* induced by $P_{\text{on}} \neq P_{\text{off}}$.

7.2 Near-optimality under linear dynamics with a shift term

We now state the performance guarantee for TWBR under the linear MDP assumption in the enclosing scope. The role of transfer is encapsulated by the admitted mismatch level τ, which upper bounds the feature-space shift for any offline transition used in Bellman targets on the event $\mathcal{E}$, via (2) and (3).

Theorem 7.2 (TWBR near-optimality in linear MDPs). *Assume $\mathcal{M}_{\text{on}}$ is a linear MDP with known feature map ϕ and unknown parameters (w_R, M_{on}). Suppose TWBR (i) maintains an estimator $M_{\text{on},t}$ satisfying* (1) *on $\mathcal{E}$, and (ii) performs fitted value iteration / Bellman regression updates in which any* *offline transition* $(s,a,r,s') \in \mathcal{D}$ *is used only with transfer weight* $w_t(s,a)$ *supported on* $\{U_t(s,a) \le \tau\}$*. Then with probability at least* $1-\delta$*, the final policy* π_N *satisfies*

$$J_{\text{on}}(\pi_N) \;\ge\; J_{\text{on}}(\pi^*) \;-\; \widetilde{O}\biggl(\frac{d}{(1-\gamma)^2\sqrt{N}} \;+\; \frac{\tau}{(1-\gamma)^2}\biggr),$$

where π^* *is an optimal policy for* $\mathcal{M}_{\text{on}}$*.*

We briefly unpack the bound. The term $\widetilde{O}\bigl(d/((1-\gamma)^2\sqrt{N})\bigr)$ is the usual estimation error from N online transitions in a d-dimensional linear model. The additive term $\widetilde{O}\bigl(\tau/(1-\gamma)^2\bigr)$ is the *residual shift bias*: by design, TWBR may still treat admitted offline transitions as approximately on-distribution, but only up to mismatch τ. In particular, if we drive $\tau \to 0$ (e.g. by rejecting all offline transitions, or by only admitting those with essentially identical feature transitions), the shift term vanishes and we recover an online-only rate, whereas larger τ yields more reuse of $\mathcal{D}$ at the cost of a controlled bias.

The proof proceeds by combining: (a) the confidence relation (1) to certify per-sample mismatch; (b) a misspecified Bellman regression analysis in which each admitted offline target contributes a bounded perturbation proportional to τ; and (c) standard propagation of value-function error through Bellman contraction and a performance-difference argument to convert critic error into return loss.

7.3 Stability floor via certified deployment

We next formalize the stability constraint. The algorithmic mechanism is minimal: we only *deploy* a candidate policy if it passes a lower confidence bound (LCB) check against the transferable baseline J^*_{tr}; otherwise we revert to a prior policy and increase conservatism.

Theorem 7.3 (Stability floor guarantee)**.** *Assume that at each deployment time* t *we compute an LCB* $\widehat{J}^{\text{LCB}}_{\text{on}}(\pi_t)$ *such that*

$$\widehat{J}^{\text{LCB}}_{\text{on}}(\pi_t) \;\le\; J_{\text{on}}(\pi_t) \qquad \textit{for all } t \le N$$

with probability at least $1-\delta$*. If TWBR enforces the rule*

$$\textit{deploy } \pi_t \textit{ only if } \widehat{J}^{\text{LCB}}_{\text{on}}(\pi_t) \ge J^*_{\text{tr}} - \varepsilon, \quad \textit{else deploy } \pi_{\text{prior},t} \in \{\pi_0, \pi_{\text{BC}}\},$$

then with probability at least $1-\delta$*,*

$$\min_{0 \le t \le N} J_{\text{on}}(\pi_t) \;\ge\; J^*_{\text{tr}} - \varepsilon.$$

The proof is an induction on deployment times conditioned on the event that all LCBs are valid. Since every deployed policy satisfies the certified inequality, the minimum deployed performance is bounded below by the same floor. Notably, this theorem is agnostic to how π_t is produced (actor-critic, fitted iteration, etc.); only the correctness of the LCB and the deployment rule matter.

7.4 Matching lower bound: an unavoidable shift penalty

We finally record that an additive shift term is not an artifact of analysis: it is information-theoretically necessary if one attempts to reuse offline transitions for Bellman backups without correcting for mismatch.

Theorem 7.4 (Unavoidable shift term without mismatch correction)**.** *There exists a pair of linear MDPs $(\mathcal{M}_{\text{off}}, \mathcal{M}_{\text{on}})$ with shared $(\mathcal{S}, \mathcal{A}, \gamma, \phi)$ and $\|M_{\text{on}} - M_{\text{off}}\|_2 = \Delta$, and an offline dataset $\mathcal{D}$ (even with infinite size), such that any algorithm that performs Bellman backups on $\mathcal{D}$ as if generated by P_{on} (i.e. without estimating/filtering/correcting mismatch) outputs a policy $\hat{\pi}$ obeying*

$$J_{\text{on}}(\pi^*) - J_{\text{on}}(\hat{\pi}) \;\geq\; \Omega\!\left(\frac{\Delta}{(1-\gamma)^2}\right).$$

The construction follows an indistinguishability argument: we build two candidate online dynamics consistent with the same offline data distribution but with opposite optimal actions under $\mathcal{M}_{\text{on}}$. Any offline-backup-based method that does not interrogate online dynamics cannot distinguish these candidates and must fail on one of them. Thus, a term scaling with the online–offline shift cannot be eliminated unless we incorporate online information in a way that explicitly controls mismatch (as TWBR does via U_t) or we avoid offline Bellman reuse altogether.

7.5 Optional: sequential regime identification for π_0 versus π_{BC}

The transferable baseline J^*_{tr} depends on which of π_0 or π_{BC} performs better on $\mathcal{M}_{\text{on}}$. Since this comparison is itself an online question, we may perform a sequential test using on-policy rollouts.

Theorem 7.5 (Sequential test complexity for baseline selection)**.** *Let $\Delta J_{\text{tr}} := J_{\text{on}}(\pi_0) - J_{\text{on}}(\pi_{\text{BC}})$. Suppose single-trajectory returns are σ^2-sub-Gaussian around their means. Fix an indifference margin $\delta_0 > 0$. Then there exists a sequential two-sided test that, with probability at least $1-\delta$, identifies whether $|\Delta J_{\text{tr}}| > \delta_0$ (and selects the better policy when separated) using*

$$\widetilde{O}\!\left(\frac{\sigma^2}{\delta_0^2}\log\frac{1}{\delta}\right)$$

online rollouts in expectation; moreover, any procedure requires $\Omega\big(\sigma^2\delta_0^{-2}\log(1/\delta)\big)$ rollouts in the worst case.

This result justifies the regime-dependent prior selection in TWBR: determining the better transferable baseline is statistically comparable to estimating a difference of two means, and it can be done with vanishing online cost relative to long-horizon fine-tuning, while still supporting the stability floor guarantee through an appropriately calibrated LCB.

8 Complexity and Implementation Notes

We record the computational footprint of TWBR in the linear instantiation and then discuss how the same design principles can be instantiated in deep actor–critic systems. We end by isolating which assumptions of the analysis are most fragile in practice and how we would operationally compensate for their failure.

Linear setting: arithmetic cost and data structures. Fix a feature dimension d, an offline dataset size $n := |\mathcal{D}|$ (counting transitions), and N online interaction steps. In the linear MDP model, the basic primitives are (i) least-squares estimation of the linear transition parameter M_{off} from $\mathcal{D}$, (ii) incremental least-squares estimation of M_{on} from online data, and (iii) repeated value-function regressions whose targets mix online transitions with a filtered subset of offline transitions.

A convenient implementation maintains regularized covariance matrices

$$V_{\text{off}} \;:=\; \lambda I + \sum_{(s,a)\in\mathcal{D}} \phi(s,a)\phi(s,a)^{\top}, \qquad V_t \;:=\; \lambda I + \sum_{i=1}^{t} \phi(s_i,a_i)\phi(s_i,a_i)^{\top},$$

together with the corresponding cross-covariances needed to solve for M_{off} and $M_{\text{on},t}$ by normal equations. Forming these sums costs $O(nd^2)$ offline and $O(Nd^2)$ online. If we recompute matrix inverses naively, we incur an additional $O(d^3)$ per refit; however, in the online loop we may update V_t^{-1} incrementally via Sherman–Morrison in $O(d^2)$ per step, which dominates the per-step cost in the linear setting.

Once M_{off} and $M_{\text{on},t}$ are available, the per-transition transfer test $U_t(s,a)$ can be evaluated in $O(d^2)$ time if it involves a quadratic form in V_t^{-1}, and in $O(d)$ time if it only uses Euclidean norms. A typical upper bound consistent with self-normalized concentration takes the form

$$U_t(s,a) \;:=\; \underbrace{\left\|(M_{\text{on},t} - M_{\text{off}})^{\top}\phi(s,a)\right\|_2}_{\text{empirical shift}} + \underbrace{\beta_t \,\|\phi(s,a)\|_{V_t^{-1}}}_{\text{estimation uncertainty}},$$

where β_t is the radius of the confidence set induced by the online regression. In this form, evaluating U_t for all offline samples at each t would be prohibitively expensive ($O(nd^2)$ per step). We therefore do not intend a literal full rescan of $\mathcal{D}$ each iteration. Instead, we either (a) compute U_t only for offline samples that appear in a sampled minibatch for the current critic regression, or (b) refresh U_t on a coarse schedule (every K online steps) while caching the most recent weights. Under either strategy, the incremental overhead of transfer control is a constant factor on top of whatever fitted value iteration / regression cost we already pay.

In summary, in the linear instantiation with incremental updates, the dominant arithmetic costs scale as

$$O\big((n+N)d^2\big) \quad \text{for the regressions and confidence maintenance,}$$

plus the cost of the chosen fitted value iteration routine on the mixed batch. Memory usage is similarly modest: we store $\mathcal{D}$ and the online buffer (or streaming sufficient statistics) and a constant number of $d \times d$ matrices, i.e., $O(n + N + d^2)$ transitions/parameters.

Deep RL instantiation: constructing practical mismatch signals. In deep actor–critic systems, we no longer have known features nor a parametric linear transition operator, so $U_t(s,a)$ must be approximated. The role of U_t in TWBR is purely *instrumental*: it gates (or weights) offline transitions to control the bias induced by using $\mathcal{D}$ to form Bellman targets for $\mathcal{M}_{\text{on}}$. Thus, any signal that (i) increases with offline–online transition disagreement and (ii) is conservative in the sense of rarely declaring a mismatched offline sample as "safe" can serve as a surrogate.

We highlight three families of instantiations.

- *Dynamics-ensemble disagreement.* We train an ensemble $\{f_k\}_{k=1}^K$ of one-step predictive models on the *online* buffer (optionally warm-started from $\mathcal{D}$), where f_k maps (s,a) to a distribution over next-state features (e.g. the parameters of a Gaussian in a learned latent space). We then define
$$U_t(s,a) \;\approx\; \mathrm{StdDev}\Big(\{f_k(s,a)\}_{k=1}^K\Big),$$
or a similar epistemic-uncertainty proxy such as the ensemble variance of the predicted next latent. Offline samples with high disagreement are downweighted or rejected. This approach is straightforward to implement and aligns with the intended meaning of "mismatch" as unpredictability of online dynamics at offline points.

- *Domain discrimination / density-ratio surrogates.* We train a classifier $c_\psi(s,a)$ to distinguish whether a state–action pair came from the online buffer or from $\mathcal{D}$. Under standard reductions, this yields an estimate of a density ratio between online and offline marginals. We may then set a weight
$$w_t(s,a) \;\propto\; \mathrm{clip}\left(\frac{\widehat{p}_{\text{on}}(s,a)}{\widehat{p}_{\text{off}}(s,a)},\, 0,\, w_{\max}\right),$$
or more directly use $c_\psi(s,a)$ as a softness factor. While this does not measure transition mismatch directly, it guards against reusing offline samples that lie outside the region supported by the (current) online state distribution, which is a frequent precursor to harmful Bellman extrapolation.

- *Latent-space moment matching.* We learn a representation $z = g_\eta(s)$ (e.g. the critic encoder), and estimate a discrepancy between conditional next-latent distributions under offline and online data using an integral probability metric (e.g. MMD) locally around (s, a). In practice, one uses minibatch estimates and obtains a soft score U_t that correlates with local nonstationarity.

Once a proxy U_t is fixed, the remainder of TWBR is standard engineering: we maintain two replay sources (offline and online), and in the critic update we compute a weighted TD loss on a mixed minibatch, e.g.

$$\mathcal{L}_{\mathrm{TD}} \;=\; \mathbb{E}_{(x,y)\sim B_{\mathrm{on}}}\big[\ell_{\mathrm{TD}}(x,y)\big] \;+\; \mathbb{E}_{(x,y)\sim \mathcal{D}}\big[w_t(x)\,\ell_{\mathrm{TD}}(x,y)\big],$$

where x abbreviates (s, a) and y abbreviates (r, s'). Hard filtering corresponds to $w_t \in \{0, 1\}$ with threshold τ, while smooth weighting (e.g. $w_t = \exp(-\beta U_t)$ with clipping) is often numerically more stable.

Which assumptions fail in practice (and what we do about it). The theory leans on (i) correct linear parametrization with known ϕ, (ii) reliable high-probability concentration to produce a conservative U_t, and (iii) a valid lower confidence bound for deployment. In deep control, each can fail:

- *Model misspecification.* Learned features are nonstationary and the environment may be far from linear in any fixed representation. We treat U_t as a heuristic safety signal and bias it toward conservatism by (a) using ensembles, (b) applying strong clipping to weights, and (c) adopting a schedule that initially privileges online data and only gradually increases the contribution of admitted offline transitions.

- *Overconfidence of uncertainty proxies.* Ensembles and classifiers can be miscalibrated, especially under distribution shift. We therefore separate *critic learning* from *deployment*: offline reuse may accelerate value estimation, but the stability rule should be enforced by direct online evidence whenever possible (e.g. periodic on-policy evaluations, bootstrap confidence over multiple critics, or pessimistic aggregations such as taking a lower quantile over an ensemble of value estimates).

- *Support and partial observability.* If the online system visits states that are absent from $\mathcal{D}$ (or vice versa), no amount of offline replay can substitute for online coverage, and domain classifiers may confound novelty with shift. Practically, we ensure the algorithm can fall back to an online-only mode (equivalently $\tau \to 0$), and we treat offline replay as optional acceleration rather than a requirement.

- *Reward shift.* The linear analysis isolates transition mismatch; in applications, reward functions may also change. We handle this by learning the reward model entirely from online data (or by maintaining a separate reward-shift detector), which prevents the most direct form of bias where offline rewards are incorrectly used as online targets.

These considerations do not change the conceptual structure: transfer is beneficial only insofar as it is *certified* (by a mismatch proxy) and *non-catastrophic* (by conservative deployment). The experimental plan that follows is designed to stress exactly these failure modes and to verify that, when naive replay is harmful, the mismatch-controlled mechanism is sufficient to restore stability while preserving as much plasticity as the online budget allows.

9 Experimental Plan

We design experiments to isolate the two claims encoded by TWBR: (i) *stability*—online fine-tuning should not underperform a transferable offline baseline beyond a prescribed slack—and (ii) *plasticity*—given a fixed online interaction budget, the method should approach the best achievable online performance while exploiting offline data whenever this is safe under shift. The central empirical question is therefore not whether offline replay can help in benign settings (it often can), but whether a *shift-controlled* mechanism can (a) detect when naive reuse becomes harmful and (b) recover most of the benefit of offline data when it is helpful.

Benchmark families and shift construction. We consider three families of tasks, each chosen to expose a different failure mode of naive offline replay.

1. *Controlled dynamics-shift in continuous control.* Starting from standard MuJoCo-style locomotion tasks (e.g. HalfCheetah, Hopper, Walker2d), we generate an offline dataset $\mathcal{D}$ in $\mathcal{M}_{\text{off}}$ under nominal physics parameters, and define $\mathcal{M}_{\text{on}}$ by modifying dynamics parameters (mass, friction, damping, actuator strength) while keeping $(\mathcal{S}, \mathcal{A}, \gamma)$ fixed. We vary the shift continuously by a scalar $\alpha \in [0, 1]$ interpolating between nominal and perturbed parameters, thereby producing a sweep of effective mismatches that approximates the theoretical role of Δ. This setting directly tests the claim that offline Bellman replay is biased when $P_{\text{off}} \neq P_{\text{on}}$, and that filtering by a mismatch signal restores monotonicity of performance as α grows.

2. *Sim-to-real style variants (domain randomization to target).* We create a simulator family where $\mathcal{M}_{\text{off}}$ corresponds to a distribution over randomized dynamics (domain randomization), and $\mathcal{M}_{\text{on}}$ is a fixed target instance, or vice versa. This yields two complementary regimes: (a)

offline data that is *broad* but potentially inconsistent with the target (high variance, lower bias), and (b) offline data that is *narrow* but potentially far from the target (low variance, higher bias). TWBR is intended to behave differently in these regimes: it should exploit broad offline coverage when it can certify local consistency, and it should sharply downweight narrow-but-mismatched offline transitions.

3. *Personalization and nonstationarity tasks.* We study a family of tasks where $\mathcal{D}$ is collected from a population (multiple users or system instances) and $\mathcal{M}_{\text{on}}$ corresponds to a particular user/system with idiosyncratic dynamics or preferences. Concretely, we instantiate this as (i) contextual decision processes (short-horizon) where transition or observation dynamics depend on a latent user type, and (ii) longer-horizon control tasks with a latent parameter (e.g. payload, joint stiffness) that changes the effective transitions. In these settings, naive offline replay can lock the critic onto population-average dynamics, harming personalized adaptation; TWBR should exhibit rapid personalization while maintaining a performance floor by reverting to a transferable prior when uncertainty is high.

Policies, data, and budgets. For each environment family, we construct $\mathcal{D}$ by running a behavior policy mixture $\pi_{\mathcal{D}}$ in $\mathcal{M}_{\text{off}}$; in continuous control, we include both near-expert and medium-quality behavior to stress the prior-selection step. We then produce π_0 by applying a strong offline RL baseline to $\mathcal{D}$ (e.g. any competitive conservative offline actor–critic), and we train π_{BC} on the same data. Online interaction is limited to a fixed budget N (chosen small relative to typical from-scratch training), with periodic evaluation rollouts reserved for monitoring. To ensure that our stability claims are meaningful, we report results across multiple random seeds and show confidence intervals for all reported metrics.

Baselines and ablations. We compare TWBR against the following:

- *Online-only fine-tuning:* initialize at π_0 (or π_{BC}) and update using only online experience. This isolates the value of offline replay during fine-tuning.

- *Naive mixed replay:* standard off-policy fine-tuning with a replay buffer containing $\mathcal{D} \cup B_{\text{on}}$ without mismatch filtering (equivalently $\tau = \infty$ or constant weight $w_t \equiv 1$). This is the method our theory predicts can catastrophically fail under shift.

- *Conservative offline regularization without shift control:* fine-tuning with a fixed KL penalty to π_0 or π_{BC} but still using naive replay. This tests whether regularization alone can substitute for mismatch-aware filtering.

- *TWBR ablations:* (i) remove the stability monitor (deploy all updated policies), (ii) remove prior selection (fix $\pi_{\text{prior}} = \pi_0$), (iii) replace hard filtering by smooth weighting $w_t(s,a) = \exp(-\beta U_t(s,a))$ with and without clipping, and (iv) vary τ to trace the empirical bias–variance tradeoff predicted by the $\widetilde{O}(\tau/(1-\gamma)^2)$ term.

Primary metrics: stability and plasticity. We operationalize the stability requirement directly. Let $J_{\text{on}}(\pi_t)$ denote the return estimated by evaluation rollouts (with sufficient repetitions to control variance). We report:

$$\text{StabilityGap} \;:=\; \max\Big\{0,\; (J^*_{\text{tr}} - \varepsilon) - \min_{0\le t\le N} J_{\text{on}}(\pi_t)\Big\},$$

along with the empirical probability of violating the floor across seeds. In addition, we report the *worst-case drop* relative to the transferable baseline,

$$\text{WorstDrop} \;:=\; \min_{0\le t\le N} \big(J_{\text{on}}(\pi_t) - J^*_{\text{tr}}\big),$$

which captures the severity of transient failures even when final performance recovers.

Plasticity is measured by both final and best-seen performance under the online budget:

$$\text{FinalGain} \;:=\; J_{\text{on}}(\pi_N) - J^*_{\text{tr}}, \qquad \text{BestGain} \;:=\; \max_{0\le t\le N} J_{\text{on}}(\pi_t) - J^*_{\text{tr}}.$$

We also plot learning curves $t \mapsto J_{\text{on}}(\pi_t)$ to visualize whether TWBR trades short-term conservatism for long-term improvement, and to detect regimes where shift is so severe that online-only learning dominates any use of $\mathcal{D}$.

Diagnostic metrics: mismatch behavior and Bellman drift on offline states. To connect empirical behavior to the theoretical mechanism, we track quantities that reflect whether offline Bellman targets are becoming inconsistent with the online MDP.

First, we measure the *admission rate* of offline transitions:

$$\text{AdmitRate}(t) \;:=\; \frac{1}{|\mathcal{B}^{\text{off}}_t|} \sum_{(s,a)\in\mathcal{B}^{\text{off}}_t} \mathbf{1}\{U_t(s,a) \le \tau\},$$

where $\mathcal{B}^{\text{off}}_t$ denotes the offline minibatch sampled at time t. Under larger shifts, we expect AdmitRate(t) to decrease, exhibiting an automatic interpolation between mixed replay and online-only learning.

Second, we measure *TD-loss drift on offline states.* Fix a snapshot critic Q_t and define a one-step TD error computed on offline samples but using the current target network and *online* reward model when available:

$$\delta_t^{\text{off}}(s,a,r,s') \ := \ Q_t(s,a) - \Big(r + \gamma\, \mathbb{E}_{a'\sim\pi_t(\cdot|s')} Q_t(s',a')\Big).$$

We report $\mathbb{E}_{(s,a,r,s')\sim\mathcal{D}}\big[(\delta_t^{\text{off}})^2\big]$ as a function of t, together with the same quantity restricted to admitted samples. The intended signature of TWBR is that admitted offline samples maintain bounded TD drift (consistent Bellman updates), whereas naive replay exhibits increasing drift under shift, correlating with performance collapse.

Key demonstrations and expected qualitative outcomes. Across all families, we aim to exhibit three regimes as the shift magnitude increases: (i) *benign shift*, where naive replay helps and TWBR matches it; (ii) *intermediate shift*, where naive replay becomes unstable while TWBR remains above the stability floor and retains nontrivial gains; and (iii) *severe shift*, where any reuse of $\mathcal{D}$ should be minimal and TWBR reduces to an online-centric method without catastrophic drops. We emphasize that the experimental burden is to show that TWBR fails gracefully: when its mismatch proxy is uninformative, it should not do better than online-only learning, but it should also not do worse than the transferable baseline beyond the prescribed slack.

Finally, we include stress tests explicitly designed to make naive replay fail: (a) offline data concentrated on narrow regions of state space that become misleading under $\mathcal{M}_{\text{on}}$, (b) offline data with high-quality actions but under different inertial parameters so that one-step targets are systematically biased, and (c) personalization tasks where population-average transitions induce the wrong optimal action for the target instance. In each case, the core claim we test is that TWBR's filtering/weighting plus conservative deployment converts these failure cases into either stable improvement or stable neutrality, thereby empirically supporting the necessity of shift-aware control of offline Bellman replay.

10 Discussion

We conclude by recording limitations of the present formulation, and by outlining several extensions that appear technically feasible but are not yet covered by our guarantees.

Modeling limitations and what the shift proxy does (and does not) certify. Our theoretical development makes an explicit linear-MDP assumption on $\mathcal{M}_{\text{on}}$ with known features ϕ and a feature-space shift $\Delta =$

$\|M_{\text{on}} - M_{\text{off}}\|_2$. This assumption is convenient for deriving confidence sets and for exposing the mechanism by which offline Bellman replay becomes biased under transition shift. However, it is restrictive in two ways. First, ϕ is assumed known and uniformly bounded, whereas in practice representation learning is entangled with policy learning and may itself change online. Second, the mismatch filter is designed to control *one-step* Bellman bias, i.e., discrepancy between T_{on} and the operator implicitly induced by replaying offline transitions. Even when one-step mismatch is small on admitted samples, compounding over multiple steps can still induce distributional shift in state visitation; our stability mechanism addresses this via conservative deployment, but our near-optimality statement does not quantify the additional price of multi-step distribution drift beyond the admitted-τ term.

A related limitation is that our proxy $U_t(s,a)$ upper-bounds a feature-space deviation $\|(M_{\text{on},t} - M_{\text{off}})^\top \phi(s,a)\|$, not a total-variation mismatch $\|P_{\text{on}}(\cdot \mid s,a) - P_{\text{off}}(\cdot \mid s,a)\|_1$. In the linear setting these quantities are linked, but in general a small feature-space deviation may fail to control downstream value error if ϕ is not sufficiently expressive. Thus, in deep instantiations, the filter should be interpreted as a *heuristic* for local consistency of Bellman targets rather than as a literal certificate of small Δ_P.

Reward shift and other forms of nonstationarity. We have emphasized transition shift, but $R_{\text{off}} \neq R_{\text{on}}$ is common in practice (e.g., preference changes, reshaped objectives, or reward sensors). If $R_{\text{on}}(s,a) = \langle w_R, \phi(s,a)\rangle$ is linear, then online reward learning is conceptually simpler than transition learning and can be handled by an additional regression with a separate confidence radius. In this case, an offline transition (s,a,r,s') can be admitted only if both a transition-consistency test and a reward-consistency test pass. Formally, we may define a combined admission criterion of the form

$$w_t(s,a) \;=\; \mathbf{1}\Big\{U_t^P(s,a) \le \tau_P \;\wedge\; U_t^R(s,a) \le \tau_R\Big\},$$

yielding an additive bias term scaling like $(\tau_P + \tau_R)/(1-\gamma)^2$ in the same manner as Theorem 2. More subtle is *nonstationarity in* $\mathcal{M}_{\text{on}}$ itself, where P_{on} drifts during online learning. In this regime, the correct object is no longer $\Delta = \|M_{\text{on}} - M_{\text{off}}\|_2$ but rather a time-varying mismatch, and one should replace static filtering with windowed or discounted estimates; stability floors then become closer to safe online learning with change-point detection.

Extension to POMDPs via state augmentation and uncertainty-aware filtering. Many transfer settings are partially observed: the online environment differs from the offline one due to latent parameters (user type, payload, friction) that are not directly observed. A direct extension is to treat the process as a POMDP and to define ϕ on histories (or learned recurrent embeddings) rather than on Markov states. Concretely, with a recurrent encoder $h_t = f_\theta(o_{1:t}, a_{1:t-1})$, we may run TWBR on the augmented state h_t and define mismatch tests on (h_t, a_t). This preserves the operational meaning of filtering—we admit offline transitions whose *predicted* next-embedding statistics are consistent with what we have learned online—but it introduces a nontrivial coupling between representation learning and the validity of confidence bounds. A promising intermediate route is to use *Bayesian latent-variable models* or ensembles to obtain a calibrated U_t as a posterior predictive uncertainty, and to view τ as a risk tolerance. Establishing high-probability statements in this regime remains open; we expect that new assumptions (e.g., observability conditions and stability of the encoder) are necessary.

Large foundation policies and critic shift under distribution mismatch. When π_0 is a large pretrained (foundation) policy, two new effects appear. First, π_0 may induce strong priors over action distributions that are beneficial even when dynamics differ, suggesting that the *prior-selection* step should include richer candidates than $\{\pi_0, \pi_{\mathrm{BC}}\}$, such as a mixture or adapter family. Second, the critic used for fine-tuning may be substantially miscalibrated off-distribution, even if the policy is competent. TWBR already allows conservative deployment based on an online lower confidence bound; in the foundation-policy regime, it is natural to treat the critic as an auxiliary estimator whose role is optimization, not certification. In particular, we may decouple (i) the optimization critic trained on mixed replay (with filtering), from (ii) a separate, potentially simpler online evaluator (e.g., truncated rollouts, doubly robust estimators with pessimistic bonuses) used exclusively for the stability monitor. This separation aligns with the principle that safety decisions should rely on estimators with controllable error, even if they are less sample-efficient.

Continuous regime scores and soft prior selection. Our description uses a discrete choice $\pi_{\mathrm{prior}} \in \{\pi_0, \pi_{\mathrm{BC}}\}$ determined by which has larger estimated J_{on}. A natural refinement is to replace this by a continuous regime score and a soft mixture. Let $\widehat{J}^{\mathrm{LCB}}_{\mathrm{on}}(\pi_0)$ and $\widehat{J}^{\mathrm{LCB}}_{\mathrm{on}}(\pi_{\mathrm{BC}})$ be lower bounds, and define

$$\rho_t \;=\; \sigma\Big(\eta\big(\widehat{J}^{\mathrm{LCB}}_{\mathrm{on}}(\pi_0) - \widehat{J}^{\mathrm{LCB}}_{\mathrm{on}}(\pi_{\mathrm{BC}})\big)\Big) \in (0,1),$$

with inverse-temperature $\eta > 0$ and logistic σ. We may then regularize toward the mixture $\pi_{\mathrm{mix},t} = \rho_t \pi_0 + (1-\rho_t)\pi_{\mathrm{BC}}$ (or toward a product-of-experts variant in logit space), and tune the KL strength λ_t as a function of the same gap. This removes discontinuities in behavior near the decision boundary and connects directly to Theorem 5: as online evidence accumulates, ρ_t concentrates, but during the uncertain phase the algorithm interpolates rather than commits. Analogously, one may define a continuous *transferability score* from the admission rate $\mathrm{AdmitRate}(t)$ and couple it to λ_t so that high mismatch automatically increases conservatism.

Safety constraints beyond return floors. The stability floor $\min_t J_{\mathrm{on}}(\pi_t) \geq J^*_{\mathrm{tr}} - \varepsilon$ is an aggregate performance constraint. Many applications require explicit safety constraints, e.g., expected discounted cost $C_{\mathrm{on}}(\pi) \leq c_{\max}$, chance constraints on failure events, or hard action/state constraints. TWBR's conservative deployment extends naturally if we can compute a lower confidence bound for return *and* an upper confidence bound for cost. Specifically, we may deploy π_t only if

$$\widehat{J}^{\mathrm{LCB}}_{\mathrm{on}}(\pi_t) \geq J^*_{\mathrm{tr}} - \varepsilon \quad \text{and} \quad \widehat{C}^{\mathrm{UCB}}_{\mathrm{on}}(\pi_t) \leq c_{\max},$$

else revert to a known-safe prior (or invoke a shield). In this view, mismatch filtering plays a dual role: it reduces optimization bias (by preventing harmful offline Bellman targets) and it reduces safety-estimation bias (by preventing offline samples from corrupting cost critics under shift). A complete treatment would require joint confidence sets for reward and cost models, and a coupling of τ with a risk budget.

Open problems. We list several questions that, in our view, delimit the current contribution. (i) *Adaptive choice of τ.* The theory suggests a bias–variance tradeoff, but selecting τ online to optimize this tradeoff with guarantees is unresolved. (ii) *Deep mismatch estimation.* Designing U_t that is both informative and calibrated in high dimensions remains the central practical bottleneck. (iii) *Multi-step and distributional mismatch.* One-step consistency does not fully characterize value bias; understanding how admission criteria should depend on rollout length is open. (iv) *Coverage and exploration.* When online data are scarce, overly conservative filtering may prevent useful generalization, yet aggressive filtering may induce bias; quantifying the optimal interpolation is not done here. (v) *Beyond two priors.* Extending regime identification from $\{\pi_0, \pi_{\mathrm{BC}}\}$ to a large set of candidate priors (including scripted fallbacks and safety policies) suggests bandit-like selection with confidence, and requires new analyses.

These directions share a common theme: transfer under shift is primarily a problem of *certifying when offline information remains locally valid.* TWBR isolates one mathematically tractable mechanism for such certification; making it both universal and computationally routine remains an open agenda.