RegimeFlow: Continuous, Uncertainty-Aware Control of Stability–Plasticity in Offline-to-Online RL

Liz Lemma Future Detective

January 20, 2026

Abstract

Offline-to-online RL fine-tuning exhibits regime-dependent behavior: sometimes retaining offline data is essential, and other times it slows learning. The source material formalizes this via a stability–plasticity decomposition and a three-regime taxonomy based on the relative returns of the pretrained policy π_0 and dataset behavior π_D. We push this idea into a 2026-ready direction: replace discrete regime assignment with a continuous, uncertainty-aware regime score and treat fine-tuning as a closed-loop control problem over stability and plasticity knobs. We propose RegimeFlow, a controller layer that adapts the offline replay ratio and conservative regularization during online fine-tuning using (i) uncertainty in $J(\pi_0) - J(\pi_D)$, (ii) distributional drift signals such as offline-vs-online TD-loss, and (iii) conservative value lower confidence bounds. In a stylized but clean model, we prove (a) stability-floor guarantees relative to the best offline baseline J^*_{off} and (b) oracle-competitive regret against the best fixed knob setting, with matching lower bounds. We outline implementations atop SAC/TD3 that preserve the source work's diagnostics, and we specify experiments that should strengthen the contribution: robustness across regimes, budgets, and near-tie settings where discrete classification is provably sample-inefficient.

Table of Contents

4. 4. RegimeFlow algorithm: regime score estimation, safety filter via value lower confidence bounds, and adaptive knob selection; relationship to the three regimes.

5. 5. Theory I (Upper bounds): stability-floor theorem and oracle-competitive regret bound; discretization error for continuous knobs.

6. 6. Theory II (Lower bounds): minimax regret lower bound vs best fixed knob; sample complexity lower bound for regime sign identification near ties.

7. 7. Practical instantiation: signals (offline/online TD-loss gap, ensemble disagreement), controller parameterization, and integration with SAC/TD3.

8. 8. Experimental design (to strengthen claims): benchmarks, ablations, stability-floor metrics, 'opposite mismatch' rate, and sensitivity to near-ties and dataset shift.

9. 9. Discussion and limitations: dependence on confidence bounds; beyond-same-MDP; toward partial resets and transfer-aware regime scores.

1 1. Introduction: inconsistent offline-to-online fine-tuning; from discrete regimes to continuous regime control; contributions and guarantees.

Offline-to-online reinforcement learning is routinely practiced under a tacit premise: given an offline dataset and a reasonable offline RL method, one may initialize from an offline-pretrained policy and then "safely" improve it online by continued learning. We take as the starting point that this premise is unreliable, and that the failure mode is structural rather than incidental. Even when the offline dataset is large, the offline-pretrained policy may be either (i) substantially better than the average behavior manifested in the dataset, or (ii) substantially worse, due to distribution shift, conservative bias, mis-specification, or simply because the dataset itself contains a mixture of behaviors whose best component is not recovered by the offline algorithm. Consequently, the most natural baseline available at deployment time is not a single policy but the better of two knowledge sources: the offline-pretrained policy and the dataset behavior as observed in the logged trajectories. This motivates an explicit stability floor relative to an offline benchmark, rather than an implicit promise of monotone improvement.

The corresponding control problem is not merely "how much to explore online," but rather how to configure the fine-tuning procedure so that it interpolates between stability and plasticity. In practice, fine-tuning exposes multiple knobs—most prominently the extent to which we continue to replay offline data versus prioritizing newly collected experience, and the strength of conservative regularization used to prevent out-of-distribution exploitation. These knobs interact in a nontrivial manner with the unknown sign and magnitude of the offline return gap

$$\Delta J \;:=\; J(\pi_0) - J(\pi_D),$$

where $\Delta J > 0$ corresponds to the regime in which the offline-pretrained policy is already superior to the dataset behavior, and $\Delta J < 0$ corresponds to the opposite regime. If ΔJ were known, a designer could prescribe an aggressive fine-tuning mode when $\Delta J < 0$ (to escape a poor initialization) and a conservative mode when $\Delta J > 0$ (to avoid "unlearning" a strong offline policy). The difficulty is that ΔJ is precisely what we do *not* know at deployment time, and the online budget may be too small to permit naive evaluation-and-commit strategies without violating a safety requirement.

A common simplification is to treat the problem as a discrete choice between a small number of qualitatively distinct regimes (e.g., "mostly offline replay" versus "mostly online learning"). We argue that such hard switching is intrinsically ill-suited to the indifference region in which $|\Delta J|$ is small. When π_0 and the dataset behavior are nearly tied, committing early to an extreme configuration can incur linear regret relative to a properly mixed configura-

tion, and can simultaneously create avoidable instability: small estimation errors in early online returns may trigger a switch that is not warranted, leading to oscillation or catastrophic degradation. This phenomenon is not an artifact of a particular implementation; it reflects the fact that identifying the sign of ΔJ is a hypothesis testing problem whose sample complexity scales as $\Omega(\tau^{-2}\log(1/\delta))$ for margin τ. Thus, near ties, one should not expect to resolve the regime quickly enough to justify discrete commitment.

We therefore replace discrete regime switching by *continuous regime control*. Concretely, we maintain a scalar regime score $r_t \in [0,1]$ which we interpret as a posterior probability that $\Delta J \geq 0$ given all information available up to episode t. This score is not itself a safety certificate, nor does it directly dictate the deployed policy. Rather, it serves as a soft inductive bias over a continuous family of fine-tuning configurations $\theta = (\alpha, \lambda)$: when r_t is large we prefer more conservative configurations (typically larger λ and smaller effective online adaptation), and when r_t is small we prefer more plastic configurations (typically smaller λ and/or larger reliance on online data), while still allowing the controller to adapt based on realized returns. The critical point is that the controller should *not* be forced to decide the regime sharply; it should instead allocate probability mass over configurations in a manner commensurate with uncertainty.

Safety is enforced by an explicit filter relative to the offline benchmark

$$J^*_{\text{off}} \;:=\; \max\big(J(\pi_0),\, J(\pi_D)\big),$$

together with a slack $\varepsilon \geq 0$. At each episode, we restrict the controller to configurations whose certified lower confidence bound satisfies $\mathrm{LCB}_t(\theta) \geq J^*_{\text{off}} - \varepsilon$. This yields a modular separation: (i) the LCB mechanism is responsible for preventing deployments that are plausibly unsafe; (ii) within the safe set, a standard experts/bandit strategy is responsible for tracking the best configuration; and (iii) the regime score r_t supplies a continuous prior that improves adaptation in the tie and near-tie regimes without sacrificing safety. The resulting controller, which we call RegimeFlow, is thus an offline-to-online procedure with an explicit safety floor and an explicit regret objective, rather than an ad hoc tuning rule.

Our contributions are threefold. First, we formalize knob selection for offline-to-online fine-tuning as a safe online learning problem over a discretized knob set, with stability constraints expressed directly in terms of J^*_{off}. Second, under a high-probability correctness assumption on $\mathrm{LCB}_t(\theta)$, we obtain a uniform-in-time stability guarantee: with probability at least $1-\delta$, every deployed episode satisfies $J(\pi_t) \geq J^*_{\text{off}} - \varepsilon$. Third, we show that, among configurations that are safe throughout, RegimeFlow attains oracle-competitive regret of order $\tilde{O}(\sqrt{T\log|\Theta|})$ (plus discretization error $O(T\varepsilon_\Theta)$ under Lipschitzness), and that this rate is unimprovable in general by a matching minimax lower bound. In addition, we isolate the fundamental

limitation underlying regime identification near ties: without sufficient evaluation, no method can reliably determine sign(ΔJ), justifying our insistence on continuous control rather than hard switching.

In the next section we recall the background decomposition into stability and plasticity and summarize the fine-tuning modules that instantiate the knobs (α, λ), together with the role of J^*_{off} as the operational offline baseline.

2 Background: stability–plasticity decomposition and fine-tuning modules

We recall a decomposition that is implicit in most offline-to-online pipelines, but which we will treat as an explicit design axis. When we deploy a policy derived from offline data and continue training online, we are simultaneously attempting to (i) preserve a baseline level of competence already present in the offline artifacts and (ii) incorporate new information from the online environment. We refer to the former requirement as *stability* and to the latter as *plasticity*. The tension is unavoidable: updates that are sufficiently plastic to correct an initially poor policy may also be sufficiently aggressive to destroy a good one.

Offline baseline and the role of J^*_{off}. At deployment time we typically have at least two policy-level sources of "knowledge" about the task. The first is the offline-pretrained policy π_0, obtained by running an offline RL algorithm (possibly with conservative bias) on D. The second is the data-generating process π_D, which we do not observe as a policy but whose empirical return can be estimated from the logged trajectories. Since neither source is uniformly dominant, the appropriate benchmark for stability is not $J(\pi_0)$ alone nor $J(\pi_D)$ alone, but rather

$$J^*_{\text{off}} \;:=\; \max\big(J(\pi_0),\, J(\pi_D)\big).$$

This choice is forced upon us by the possibility that π_0 underperforms the typical behavior in the dataset (e.g., due to pessimism, model misspecification, or mismatch between the offline objective and the online return), as well as by the opposite possibility that π_0 substantially improves upon the average logged behavior (e.g., by extracting the best component from a mixture). Consequently, stability should be expressed as the requirement that deployed online policies do not fall substantially below J^*_{off}, up to a slack ε that absorbs estimation error and desired conservatism.

A generic offline-to-online update template. Let π denote the actor (or policy) parameters and let $\widehat{Q}$ denote critic/value parameters where applicable. The fine-tuning procedures we consider can be summarized by

alternating updates on samples drawn from two sources: offline replay from D and newly collected online experience. The operative knob is an offline replay ratio $\alpha \in [0, 1]$ that determines the mixture used in the learning rule. Abstractly, one may view each episode-level update as minimizing an objective of the form

$$\mathcal{L}_t(\pi, \widehat{Q}; \theta) \;=\; \alpha(\theta)\, \mathcal{L}_{\text{off}}(\pi, \widehat{Q}; D) \;+\; \big(1-\alpha(\theta)\big)\, \mathcal{L}_{\text{on}}(\pi, \widehat{Q}; \mathcal{B}_t) \;+\; \lambda(\theta)\, \mathcal{R}(\pi, \widehat{Q}),$$

where $\mathcal{B}_t$ denotes the online buffer accumulated up to time t, $\mathcal{R}$ is a conservative regularizer, and $\theta = (\alpha, \lambda)$ collects the tunable coefficients. This template is intentionally broad: $\mathcal{L}_{\text{off}}$ may correspond to fitted Q iteration, actor–critic updates, behavior cloning, or value regression, and $\mathcal{L}_{\text{on}}$ may be any on-policy or off-policy RL loss. What matters for the controller is that α and λ implement the stability–plasticity trade-off at a coarse level and are exposed to selection online.

Module 1: offline replay and its schedules. Offline replay serves two distinct purposes. First, it anchors the agent to the support of D, which can mitigate the extrapolation error that arises when critics are trained on out-of-distribution actions. Second, it reduces the variance of gradient estimates early in online training when $\mathcal{B}_t$ is small. Both effects increase stability but can reduce plasticity when π_0 is poor, because heavy replay can trap learning near the behavior distribution represented in D. This suggests that a fixed α may be suboptimal, motivating episode-dependent schedules (e.g., warmup with large α followed by a decay). However, schedules are themselves knob choices; in our abstraction we fold them into θ by considering a discretized family of admissible update rules.

Module 2: warmup and delayed adaptation. A commonly used stabilization heuristic is a warmup period in which the agent either (i) does not update the actor, updating only the critic/value function, or (ii) updates with a strongly conservative objective (large λ) before relaxing it. From the stability–plasticity perspective, warmup is a mechanism that temporarily reduces plasticity until sufficient online data has been gathered to make online gradients informative. Warmup is particularly relevant when π_0 is strong ($\Delta J > 0$), since premature adaptation can cause regression due to critic miscalibration under the evolving state distribution. Conversely, when π_0 is weak ($\Delta J < 0$), warmup trades early opportunity for late safety, and its duration becomes a critical design decision.

Module 3: conservative regularization. The parameter $\lambda \geq 0$ represents the strength of a conservative mechanism designed to prevent the learner from exploiting errors in value estimation outside the data support.

Canonical instances include: penalties that push Q-values downward on out-of-distribution actions (as in conservative Q-learning), KL constraints that keep π close to a reference policy (often the behavior policy or π_0), explicit behavior cloning terms, pessimistic value iteration, and uncertainty-penalized objectives computed from ensembles. Increasing λ generally increases stability but may impede the discovery of improved behaviors when online data reveals advantageous actions not present in D. Importantly, λ interacts with α: heavy offline replay with weak regularization can still be unstable if the offline critic extrapolates poorly, while strong regularization with little replay can be overly inertial.

Module 4: reset and fallback mechanisms. Practical systems frequently incorporate resets, either hard (reverting parameters to π_0 or to a previous checkpoint) or soft (shrinking step sizes, increasing α, or increasing λ in response to degradation). Resets are an implicit admission that stability cannot be guaranteed by the base learning dynamics alone. In the framework we adopt, reset is treated as part of the admissible fine-tuning family $\{\mathsf{FT}(\theta)\}$: certain θ correspond to highly conservative modes that emulate "staying near π_0," while others allow more aggressive adaptation but must be vetted by a safety mechanism.

Summary. The above modules expose a multi-dimensional knob space in which stability and plasticity are traded continuously rather than by a binary choice. Since the sign and magnitude of $\Delta J = J(\pi_0) - J(\pi_D)$ determine which region of this space is desirable but are not known a priori, we require an online selection rule that (i) references J^*_{off} as the operational baseline, (ii) quantifies uncertainty in performance, and (iii) adapts (α, λ) without committing prematurely. This motivates the formal problem formulation in the next section, where we model knob selection as an online decision problem with an explicit stability floor.

3 Problem formulation: regime-adaptive fine-tuning with a stability floor

We formalize offline-to-online fine-tuning as an online control problem over a continuous knob space, in which each knob choice specifies an admissible update rule and induces a (random) deployed policy. The salient difficulty is that the appropriate stability–plasticity trade-off depends on the unknown comparison between the offline-pretrained policy and the dataset-level behavior.

MDP, value, and offline baselines. Let $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r, \gamma)$ be a discounted MDP with rewards bounded in $[0, 1]$. For any policy π, we denote

its discounted return by

$$J(\pi) \;:=\; \mathbb{E}\Big[\sum_{k\geq 0}\gamma^k r(s_k,a_k)\;\Big|\; a_k\sim\pi(\cdot|s_k),\; s_{k+1}\sim P(\cdot|s_k,a_k)\Big].$$

We are given an offline dataset D of trajectories generated by an unknown mixture behavior policy, which we abstract as a single policy π_D whose (empirical) return can be estimated from D:

$$J(\pi_D) \;\approx\; \frac{1}{|D|}\sum_{\tau\in D}\sum_{k\geq 0}\gamma^k r(s_k^\tau,a_k^\tau).$$

We are also given an offline-pretrained policy π_0 obtained from running an offline RL algorithm on D. Since neither π_0 nor the typical behavior in D is *a priori* dominant, we take as our operational offline baseline

$$J^*_{\text{off}} \;:=\; \max\big(J(\pi_0),\, J(\pi_D)\big), \qquad \Delta J \;:=\; J(\pi_0)-J(\pi_D).$$

The sign and magnitude of ΔJ encode the "regime" of the instance: if $\Delta J>0$, aggressive online adaptation risks degrading a strong initialization; if $\Delta J<0$, overly conservative updates delay needed improvement. The case $|\Delta J|$ small is an indifference region in which premature commitment to one extreme is undesirable.

Knob space and admissible fine-tuning procedures. We posit a family $\{\mathsf{FT}(\theta):\theta\in\Theta_{\text{cont}}\}$ of fine-tuning procedures parameterized by a continuous knob $\theta=(\alpha,\lambda)$, where $\alpha\in[0,1]$ is an offline replay ratio and $\lambda\in[0,\Lambda]$ is a conservative regularization strength. We emphasize that $\mathsf{FT}(\theta)$ is not a single gradient step but an episode-level (or epoch-level) update rule that, given the current learner state and data buffers, outputs an updated policy. In particular, if $\mathcal{B}_t$ denotes the online buffer available after episode t, then running $\mathsf{FT}(\theta)$ up to time t produces a deployed policy that we denote by π_t^θ.

To keep the decision problem finite while retaining the continuous semantics, we assume that Θ is a finite ε_Θ-net of $\Theta_{\text{cont}}=[0,1]\times[0,\Lambda]$ under an appropriate norm (e.g. ℓ_∞), and we restrict online decisions to $\theta_t\in\Theta$. Any approximation error due to discretization will be accounted for as an additive term of order $T\varepsilon_\Theta$ under standard Lipschitz assumptions (cf. Theorem 2).

Online interaction protocol and feedback. We consider an episodic interaction budget of T episodes. At each episode $t\in\{1,\dots,T\}$, a controller selects $\theta_t\in\Theta$, the underlying learner deploys the current policy produced under θ_t, and the environment returns an episode return R_t. The controller is not assumed to observe $J(\pi_t^\theta)$ for all θ, but only bandit feedback through R_t (and possibly auxiliary diagnostics such as TD-error gaps or ensemble

disagreement, which we treat as side information rather than as a separate feedback channel).

We model return noise by assuming that, conditional on the deployed policy, R_t is a sub-Gaussian observation of $J(\pi_t)$ with variance proxy σ^2; that is, for each t,

$$\mathbb{E}[R_t \mid \pi_t] = J(\pi_t), \qquad R_t - J(\pi_t) \text{ is sub-Gaussian with parameter } \sigma^2.$$

This abstraction covers both finite-horizon episodic returns and discounted continuing tasks under standard boundedness assumptions.

Stability floor as a high-probability constraint. The controller must enforce a stability floor relative to the best offline baseline. Fix a slack $\varepsilon \geq 0$ and failure probability $\delta \in (0, 1)$. The constraint is

$$\Pr\Big[\forall t \leq T : \; J(\pi_t) \geq J^*_{\text{off}} - \varepsilon\Big] \;\geq\; 1 - \delta.$$

We interpret ε as absorbing both the desired conservatism and the estimation error in $J(\pi_D)$ and $J(\pi_0)$ (e.g. if $J(\pi_0)$ is estimated from a finite number of evaluation rollouts).

Objective and regret. Among all controller strategies that satisfy the stability floor, we seek to maximize cumulative value. Since the environment is stationary but the learner state evolves with online data, we benchmark against the best fixed safe knob choice. Let θ^* denote a comparator in Θ that is safe throughout the process (formalized via confidence bounds below). We define regret as

$$\text{Regret}(T) \;:=\; \sum_{t=1}^{T} \Big(J(\pi_t^{\theta^*}) - J(\pi_t) \Big),$$

and our objective is to design a controller that achieves sublinear regret (in expectation) while maintaining stability with probability at least $1 - \delta$.

Uncertainty model via lower confidence bounds. The key structural assumption enabling safety is access to a value lower-confidence bound for each candidate knob. Concretely, for each episode t and knob $\theta \in \Theta$, we assume the existence of a computable quantity $\text{LCB}_t(\theta)$ such that

$$\Pr\big[J(\pi_t^{\theta}) \geq \text{LCB}_t(\theta)\big] \;\geq\; 1 - \frac{\delta}{T|\Theta|}. \tag{1}$$

The bound (1) is stated at the level of policy value rather than per-transition error; it can be realized by a variety of constructions (e.g. bootstrap/ensemble critics with pessimistic aggregation, or concentration bounds for fitted value

estimates) depending on the instantiated RL method. For our purposes, (1) is the interface between learning dynamics and the controller: it permits knob selection under a safety filter without requiring the controller to model the internal optimization trajectory of $\mathsf{FT}(\theta)$.

This completes the formal problem statement. In the next section we give an explicit controller, RegimeFlow, that couples (1) with regime-aware preferences over (α, λ) while maintaining the stability floor by construction.

4 RegimeFlow: safety-filtered regime-adaptive knob control

We now describe RegimeFlow, an episodic controller that selects $\theta_t = (\alpha_t, \lambda_t) \in \Theta$ while enforcing the stability floor by construction. The controller has two coupled responsibilities: (i) maintain a scalar *regime score* $r_t \in [0, 1]$ encoding how strongly the current evidence supports $\Delta J \geq 0$, and (ii) run an online experts/bandit procedure over knob choices, but only within a *safe set* defined by lower confidence bounds.

Safety filter from value lower bounds. At each episode t, we compute $\mathrm{LCB}_t(\theta)$ for each $\theta \in \Theta$ and define the (random) safe set

$$S_t \;:=\; \Big\{\theta \in \Theta :\; \mathrm{LCB}_t(\theta) \;\geq\; J^*_{\mathrm{off}} - \varepsilon\Big\}.$$

The controller is constrained to select $\theta_t \in S_t$. Operationally, this implements a one-step *accept/reject* rule: knobs whose pessimistic value estimate falls below the offline baseline (up to slack ε) are never deployed. We emphasize that the filter is agnostic to how $\mathrm{LCB}_t(\theta)$ is computed; it only requires the interface guarantee (1). In particular, the learner may use any mixture of offline replay and online experience under θ, provided it can expose a conservative estimate of the induced deployed value.

When S_t is empty, a reasonable implementation resorts to a predetermined "failsafe" knob (e.g. maximal regularization and/or high offline replay) together with an enlarged ε; however, our theoretical development will assume $S_t \neq \emptyset$ for all t (as in Theorem 2), which is the standard feasibility condition in safe bandits.

Regime score as a posterior over the sign of ΔJ. The regime score r_t is intended to approximate

$$r_t \;\approx\; \Pr\big[\Delta J \geq 0 \mid \mathcal{F}_t\big],$$

where $\mathcal{F}_t$ denotes the sigma-field generated by all offline information and online observations up to episode $t-1$ (including returns and diagnostics).

Concretely, we maintain an estimate $\widehat{\Delta J}_t$ and an uncertainty proxy $s_t > 0$ so that $\widehat{\Delta J}_t$ is interpreted as approximately normal with standard deviation s_t. This can be instantiated in several ways: (i) from an initial evaluation of π_0 together with the empirical mean of D (yielding r_1), and then (ii) updated online using additional rollouts under π_0 and/or using diagnostic evidence that correlates with distribution shift (e.g. increasing TD-error gap between online data and replayed offline data). Abstractly, given $(\widehat{\Delta J}_t, s_t)$ we set

$$r_t \;:=\; \Phi\left(\frac{\widehat{\Delta J}_t}{s_t}\right),$$

where Φ is the standard normal CDF (a logistic map could be used equivalently). The only property we exploit algorithmically is monotonicity: larger $\widehat{\Delta J}_t/s_t$ yields larger r_t, so the controller becomes progressively more confident that π_0 dominates the dataset behavior.

Regime-aware preferences over knobs. While safety is enforced solely by S_t, we use r_t to bias exploration *within* S_t toward knobs that match the inferred regime. We encode this via two fixed reference distributions over Θ, a "trust-π_0" prior q^+ and a "trust-D" prior q^-, chosen to reflect the intended stability–plasticity trade-off:

$$q^+(\theta) \text{ favors larger } \lambda \text{ and smaller } \alpha, \qquad q^-(\theta) \text{ favors smaller } \lambda \text{ and larger } \alpha.$$

The rationale is as follows. If $\Delta J > 0$ (so π_0 is strong relative to typical behavior in D), then heavy offline replay risks pulling updates toward suboptimal actions present in the dataset, hence we bias toward smaller α, while larger λ discourages abrupt departures from the current policy and mitigates overfitting to transient online noise. Conversely, if $\Delta J < 0$, then the dataset behavior is comparatively competent and can serve as an anchor for early improvement; thus we prefer larger α and weaker regularization to enable faster adaptation.

We combine these preferences into a time-varying prior

$$p_t(\theta) \;:=\; r_t\, q^+(\theta) \;+\; (1 - r_t)\, q^-(\theta),$$

which we treat as a soft suggestion rather than a constraint: safety is determined by S_t, and learning performance is determined by observed returns.

Adaptive knob selection by experts on the filtered set. Let $w_t(\theta)$ be nonnegative weights over Θ (initialized uniformly). Each episode, after computing S_t, we form a sampling distribution over admissible knobs by restricting to S_t and blending with the regime prior:

$$\mu_t(\theta) \;\propto\; \mathbf{1}\{\theta \in S_t\}\, w_t(\theta)\, p_t(\theta), \qquad \theta_t \sim \mu_t.$$

We then deploy the learner under θ_t for one episode, observe the episode return R_t, and update the weights w_t via an EXP3-style multiplicative rule based on an importance-weighted reward estimate. Writing $\widehat{R}_t(\theta_t) := R_t/\mu_t(\theta_t)$ and $\widehat{R}_t(\theta) := 0$ for $\theta \neq \theta_t$, one canonical update is

$$w_{t+1}(\theta) \;=\; w_t(\theta)\exp\big(\eta\,\widehat{R}_t(\theta)\big),$$

with learning rate $\eta > 0$. This produces an online trade-off between exploration over admissible knobs and exploitation of those that have yielded high returns. Importantly, all such updates are performed only over Θ; the controller does not require gradients through $\mathsf{FT}(\theta)$, nor does it need to predict how θ affects future learner states.

Connection to the three regimes. The controller's behavior interpolates smoothly between three qualitatively different situations. In the π_0-*dominant regime* ($\Delta J \gg 0$), evidence drives $r_t \to 1$, hence p_t concentrates toward knobs with stronger conservatism (larger λ) and reduced offline replay (smaller α), while the safety filter prevents excursions below $J^*_{\text{off}} - \varepsilon$. In the *dataset-dominant regime* ($\Delta J \ll 0$), we instead obtain $r_t \to 0$, so p_t favors higher replay and weaker regularization, accelerating improvement while still respecting the safety filter. In the *indifference regime* ($|\Delta J|$ small), r_t remains away from the extremes for a nontrivial period (cf. Theorem 4), and the prior p_t mixes the two extremes rather than committing early; combined with experts-style adaptation, this yields gradual knob adjustment driven by returns, rather than brittle hard switching between (α, λ) extremes.

In summary, RegimeFlow separates *safety* (a pointwise constraint implemented by S_t) from *adaptation* (an experts procedure on the remaining choices), and uses r_t only as a regime-dependent inductive bias within the safe region. The next section formalizes the resulting stability and regret guarantees.

5 Theory I: upper bounds (stability and oracle-competitive regret)

We analyze RegimeFlow under assumptions (A1)–(A3) stated in the enclosing scope. Throughout, rewards are bounded in $[0, 1]$, and $J(\pi)$ denotes the discounted return. We write $J^*_{\text{off}} := \max(J(\pi_0), J(\pi_D))$ and fix a slack $\varepsilon \geq 0$. For each episode t, the controller forms the safe set

$$S_t \;:=\; \Big\{\theta \in \Theta :\; \mathrm{LCB}_t(\theta) \;\geq\; J^*_{\text{off}} - \varepsilon\Big\},$$

and deploys only $\theta_t \in S_t$. Our first result shows that this pointwise filtering suffices to guarantee a uniform-in-time stability floor.

5.1 High-probability stability floor

Theorem 5.1 (High-probability stability floor)**.** *Assume (A1): for all $t \le T$ and $\theta \in \Theta$,*

$$\Pr\big[J(\pi_t^\theta) \;\ge\; \mathrm{LCB}_t(\theta)\big] \;\ge\; 1 - \frac{\delta}{T|\Theta|}.$$

Then, if RegimeFlow *always selects $\theta_t \in S_t$, we have*

$$\Pr\Big[\forall t \le T:\; J(\pi_t) \;\ge\; J^*_{\mathrm{off}} - \varepsilon\Big] \;\ge\; 1 - \delta.$$

Proof. Define the event

$$\mathcal{E} \;:=\; \bigcap_{t=1}^{T} \bigcap_{\theta\in\Theta} \Big\{J(\pi_t^\theta) \;\ge\; \mathrm{LCB}_t(\theta)\Big\}.$$

By (A1) and a union bound over the $T|\Theta|$ pairs (t,θ),

$$\Pr[\mathcal{E}] \;\ge\; 1 - \sum_{t=1}^{T}\sum_{\theta\in\Theta} \frac{\delta}{T|\Theta|} \;=\; 1-\delta.$$

On $\mathcal{E}$, for each episode t and each θ, we have $J(\pi_t^\theta) \ge \mathrm{LCB}_t(\theta)$. Since the controller chooses $\theta_t \in S_t$, it satisfies $\mathrm{LCB}_t(\theta_t) \ge J^*_{\mathrm{off}} - \varepsilon$. Therefore, on $\mathcal{E}$,

$$J(\pi_t) \;=\; J(\pi_t^{\theta_t}) \;\ge\; \mathrm{LCB}_t(\theta_t) \;\ge\; J^*_{\mathrm{off}} - \varepsilon \qquad \text{for all } t \le T,$$

which implies the desired probability statement. □

Remark (role of feasibility). Theorem 5.1 is conditional only on selecting $\theta_t \in S_t$. If $S_t = \emptyset$ can occur, then either a fallback mechanism is needed or the guarantee must be weakened (e.g., allow a small number of violations). In the remainder, we adopt the standard feasibility condition $S_t \neq \emptyset$ for all t.

5.2 Oracle-competitive regret over safe knobs

We compare to the best *fixed* knob in hindsight among those that are safe throughout the horizon. Let

$$\Theta_{\mathrm{safe}} \;:=\; \bigcap_{t=1}^{T} S_t, \qquad \theta^* \in \arg\max_{\theta\in\Theta_{\mathrm{safe}}} \sum_{t=1}^{T} J(\pi_t^\theta),$$

and define the (policy-value) regret

$$\mathrm{Regret}(T) \;:=\; \sum_{t=1}^{T}\Big(J(\pi_t^{\theta^*}) - J(\pi_t)\Big).$$

Because the learner state evolves over time, $J(\pi_t^\theta)$ should be interpreted as the expected return at episode t of the policy obtained by running the fine-tuning procedure under knob θ up to that episode; this is precisely the setting handled by adversarial experts, where payoffs may be nonstationary and adaptive.

Theorem 5.2 (Oracle-competitive regret on the filtered set)**.** *Assume (A2) sub-Gaussian return noise and that $\Theta_{\text{safe}} \neq \emptyset$. If RegimeFlow uses an EXP3-style multiplicative update on the admissible set (i.e., it samples θ_t from a distribution supported on S_t and updates weights using an importance-weighted return estimate), then for an appropriate learning rate η,*

$$\mathbb{E}\big[\text{Regret}(T)\big] \;\le\; \tilde{O}\big(\sqrt{T \log |\Theta|}\big),$$

where $\tilde{O}(\cdot)$ hides polylogarithmic factors and sub-Gaussian constants.

Proof sketch. We reduce to adversarial experts with a time-varying availability constraint. Define the episode payoff of knob θ at time t as $g_t(\theta) := J(\pi_t^\theta) \in [0, 1/(1-\gamma)]$ (or normalize to $[0,1]$ by scaling). The controller observes a single noisy sample R_t of $g_t(\theta_t)$ and forms an unbiased importance-weighted estimator $\widehat{g}_t(\theta)$ supported on the played arm θ_t. Standard EXP3 analysis yields, for any fixed comparator θ in the support of the sampling distribution at all times, that

$$\mathbb{E}\Big[\sum_{t=1}^{T} g_t(\theta) - g_t(\theta_t)\Big] \;\le\; O\Big(\sqrt{T \log |\Theta|}\Big)$$

(up to logarithmic factors arising from boundedness and variance control of the importance weights). The only additional work is to ensure the comparator is always admissible. This is achieved by restricting the comparator class to $\Theta_{\text{safe}} = \cap_t S_t$: for any $\theta^* \in \Theta_{\text{safe}}$, the algorithm's sampling distribution (by construction) is supported on S_t, hence θ^* is never removed by the filter and the standard EXP3 potential argument applies on each round after renormalization to S_t. Sub-Gaussian noise enters only to justify concentration/variance bounds for the importance-weighted estimates and to control the hidden polylogarithmic factors. □

5.3 Discretization error for continuous knobs

Assumption (A3) models Θ as a finite ε_Θ-net of the continuous knob space $[0,1] \times [0, \Lambda]$. To quantify the price of discretization, we impose a regularity condition on value as a function of knobs.

Lipschitz model. Assume that for each episode t, the mapping $\theta \mapsto J(\pi_t^\theta)$ is L-Lipschitz with respect to a norm $\|\cdot\|$ on $[0,1] \times [0,\Lambda]$:

$$\left|J(\pi_t^\theta) - J(\pi_t^{\theta'})\right| \;\le\; L\,\|\theta - \theta'\| \qquad \text{for all } \theta, \theta'.$$

Let $\theta^\star_{\text{cont}}$ be the best fixed *continuous* knob that is safe (in the analogous sense) and achieves maximal cumulative value. Let $\Pi(\theta^\star_{\text{cont}}) \in \Theta$ be a nearest grid point so that $\|\Pi(\theta^\star_{\text{cont}}) - \theta^\star_{\text{cont}}\| \le \varepsilon_\Theta$. Then, for each t,

$$J(\pi_t^{\theta^\star_{\text{cont}}}) \;\le\; J(\pi_t^{\Pi(\theta^\star_{\text{cont}})}) \;+\; L\varepsilon_\Theta,$$

and summing over t yields an approximation gap at most $LT\varepsilon_\Theta$. Consequently, combining Theorem 5.2 with this coupling argument, the regret to the best continuous safe knob is bounded as

$$\mathbb{E}\big[\mathrm{Regret}_{\text{cont}}(T)\big] \;\le\; \tilde{O}\big(\sqrt{T \log |\Theta|}\big) \;+\; O(LT\varepsilon_\Theta).$$

This is the only place where the finiteness of Θ is essential: the experts bound scales with $\log|\Theta|$, while the discretization term scales linearly in T and vanishes as the grid is refined.

The upper bounds above are information-theoretically tight up to logarithmic factors; we formalize this in the subsequent lower-bound section.

6 Theory II: lower bounds (minimax regret and regime identification)

We complement the upper bounds by two information-theoretic limitations inherent to the controller model: (i) a minimax regret lower bound for selecting knobs under bandit feedback (even ignoring safety), and (ii) a sample complexity lower bound for identifying the sign of $\Delta J := J(\pi_0) - J(\pi_D)$ near ties. Together, they justify that the rates in Section 5 are, up to logarithmic factors and discretization, the best one can hope for without additional structure beyond (A1)–(A3).

6.1 Minimax regret lower bound versus the best fixed knob

The regret guarantee in Theorem 5.2 scales as $\tilde{O}(\sqrt{T \log |\Theta|})$ for finite Θ. We now show that this dependence cannot be improved in general: for any controller that chooses θ_t adaptively based on past observed returns, there exist instances for which the expected regret is at least on the order of $\sqrt{T \log |\Theta|}$. The key point is that the controller observes only the realized return of the deployed knob, so knob selection is at least as hard as adversarial experts (or, by specialization, stochastic bandits).

Theorem 6.1 (Minimax lower bound for knob-selection regret)**.** *Fix any finite knob set Θ with $|\Theta| \geq 2$ and horizon T. Consider any (possibly randomized) algorithm that selects $\theta_t \in \Theta$ and observes only a noisy return sample R_t from the deployed policy. Then there exists a problem instance consistent with the interaction model (indeed, a one-step MDP embedded as $\mathcal{M}$) such that, even when* all *knobs are safe (i.e., $S_t = \Theta$ for all t), the algorithm satisfies*

$$\mathbb{E}\big[\mathrm{Regret}(T)\big] \;\geq\; c\sqrt{T \log |\Theta|}$$

for a universal constant $c > 0$, where regret is measured against the best fixed knob in hindsight.

Proof sketch. We reduce to the classical lower bound for adversarial experts. Construct an episodic MDP with a single nonterminal state s and a terminal state; taking any action terminates immediately and yields a reward in $[0, 1]$. Define a mapping $\varphi : \Theta \to \{1, \ldots, |\Theta|\}$ from knob settings to "arms." For each round t, an oblivious adversary chooses a reward vector $x_t \in [0, 1]^{|\Theta|}$; if the controller plays θ_t, the episode reward is $R_t := x_t(\varphi(\theta_t)) + \xi_t$ where ξ_t is mean-zero noise (or take $\xi_t \equiv 0$ in the purely adversarial construction). Since the MDP is one-step, the episode return equals the immediate reward, hence $J(\pi_t^\theta) = x_t(\varphi(\theta))$ for all θ, and the controller's observation model coincides with bandit feedback in experts. Standard minimax lower bounds (e.g., via a randomized hard instance over $\{0, 1\}^{|\Theta|}$ reward vectors with controlled KL divergence) imply that for any algorithm,

$$\sup_{(x_t)_{t=1}^T} \mathbb{E}\Big[\sum_{t=1}^{T} x_t(\varphi(\theta^*)) - x_t(\varphi(\theta_t))\Big] \;\geq\; c\sqrt{T \log |\Theta|},$$

where θ^* is the best fixed knob in hindsight. Interpreting $x_t(\varphi(\theta))$ as $J(\pi_t^\theta)$ yields the claim. □

Remark (safety does not help in the worst case). The reduction sets $S_t = \Theta$ so that the stability filter never removes any knob. Therefore, any improvement over $\sqrt{T \log |\Theta|}$ would contradict the minimax lower bound for bandit/experts. This formalizes the intuition that, absent additional structure (e.g., parametric reward models or smoothness enabling generalization across θ), the controller must pay the usual exploration cost to compete with the best fixed knob.

6.2 Lower bound for regime sign identification near ties

We next formalize the difficulty of determining whether π_0 is truly better than the dataset behavior level π_D when the gap ΔJ is small. This question

appears explicitly in the controller's regime score r_t, which we interpret as a posterior belief about $\Delta J \geq 0$. When $|\Delta J|$ is below a margin τ, any high-confidence decision requires on the order of τ^{-2} independent evaluation episodes, matching the familiar rate for mean estimation/hypothesis testing under sub-Gaussian noise.

Theorem 6.2 (Near-tie regime identification lower bound)**.** *Fix $\tau > 0$ and $\delta \in (0, 1)$. Consider testing*

$$H_+ : \ \Delta J \geq \tau \qquad \textit{versus} \qquad H_- : \ \Delta J \leq -\tau$$

from episode returns with sub-Gaussian noise. Any (possibly adaptive) procedure that outputs $\widehat{\sigma} \in \{+, -\}$ with $\Pr_{H_+}(\widehat{\sigma} = +) \geq 1 - \delta$ and $\Pr_{H_-}(\widehat{\sigma} = -) \geq 1 - \delta$ requires

$$n \ \geq \ c' \, \tau^{-2} \log\Big(\frac{1}{\delta}\Big)$$

evaluation episodes in the worst case, for a universal constant $c' > 0$.

Proof sketch. We apply Le Cam's method by constructing two instances whose induced return distributions are close in total variation. Consider an MDP family in which evaluating π_0 (respectively π_D) yields an episode return with distribution $\mathcal{N}(\mu_0, 1)$ (respectively $\mathcal{N}(\mu_D, 1)$), independent across episodes, with $\mu_0 - \mu_D = +\tau$ under H_+ and $\mu_0 - \mu_D = -\tau$ under H_-. Any algorithm observing n episodes (possibly interleaving evaluations and using adaptivity) induces two distributions $\mathbb{P}_+$ and $\mathbb{P}_-$ over transcripts. The KL divergence scales as $\mathrm{KL}(\mathbb{P}_+ \,\|\, \mathbb{P}_-) = O(n\tau^2)$ by additivity and the KL formula for Gaussians. Le Cam's inequality implies that the minimax testing error is bounded below in terms of $\mathrm{TV}(\mathbb{P}_+, \mathbb{P}_-)$, which in turn is controlled by KL. Ensuring error at most δ forces $\mathrm{KL}(\mathbb{P}_+ \,\|\, \mathbb{P}_-) \gtrsim \log(1/\delta)$, hence $n \gtrsim \tau^{-2} \log(1/\delta)$. The same argument holds for sub-Gaussian noise via a change of measure and standard KL upper bounds. □

Consequence for controllers. Theorem 6.2 implies that, in the indifference region $|\Delta J| \lesssim \tau$, no controller can rapidly drive r_t to 0 or 1 with high confidence; doing so would require $\Omega(\tau^{-2} \log(1/\delta))$ dedicated episodes. Consequently, strategies that *commit* early to a hard-switch extreme (e.g., $\alpha \in \{0, 1\}$) can incur linear regret in τ over the horizon, whereas maintaining a graded belief and selecting knobs conservatively is statistically aligned with what can be inferred from limited online interaction.

7 Practical instantiation: signals, controller parameterization, and integration with SAC/TD3

We describe a concrete instantiation of RegimeFlow in which the controller is driven by inexpensive online diagnostics and is coupled to a standard

actor–critic learner (SAC or TD3). The intent is not to strengthen the assumptions (A1)–(A3), but rather to indicate how one may approximate the oracle ingredients in a way that is faithful to the stability-floor design.

Replay-mixing knobs and fine-tuning procedure. We implement $\mathsf{FT}(\theta)$ by augmenting a base off-policy update with (i) a replay-mixture ratio $\alpha \in [0,1]$ and (ii) a conservative regularizer of strength $\lambda \geq 0$. Concretely, maintain an offline buffer $\mathcal{B}_{\text{off}} := D$ and an online buffer $\mathcal{B}_{\text{on}}$ that accumulates interaction. At episode t with chosen $\theta_t = (\alpha_t, \lambda_t)$, each gradient step samples a minibatch of size B by drawing $\lfloor \alpha_t B \rfloor$ transitions from $\mathcal{B}_{\text{off}}$ and the remainder from $\mathcal{B}_{\text{on}}$. This realizes the knob as an explicit control on distribution shift: large α_t anchors learning to the dataset support, while small α_t allows rapid adaptation to online data.

For the regularization knob, we use an additive penalty in the critic objective that is monotone in estimated extrapolation, in the spirit of conservative Q-learning. Writing the critic loss for parameters ϕ as $\mathcal{L}_{\text{TD}}(\phi)$ (SAC) or $\mathcal{L}_{\text{TD3}}(\phi)$ (TD3), we optimize

$$\mathcal{L}_Q(\phi; \theta_t) \;:=\; \mathcal{L}_{\text{TD}}(\phi) \;+\; \lambda_t \, \mathcal{R}(\phi),$$

where $\mathcal{R}(\phi)$ may be taken as a log-sum-exp penalty that lowers Q-values on actions not supported by the data, or as a squared deviation penalty to a behavior-cloned baseline. The actor update is standard SAC/TD3, but uses the regularized critic, hence inheriting the induced conservatism. In practice we couple λ_t only to the critic; this separation makes it easier to interpret λ_t as a safety knob rather than an exploration knob.

Online signals for regime tracking. The regime score r_t is meant to quantify whether the offline-pretrained policy is already at (or above) the dataset knowledge level, i.e., whether $\Delta J = J(\pi_0) - J(\pi_D) \geq 0$. While ΔJ is not directly observable in the fine-tuning stream, we can track surrogate signals that are predictive of "offline advantage" versus "need for adaptation."

(1) Offline/online TD-loss gap. Maintain running estimates of critic Bellman error on offline and online samples under the current networks:

$$\widehat{\mathcal{E}}_{\text{off}}(t) \;:=\; \mathbb{E}_{(s,a,r,s') \sim \mathcal{B}_{\text{off}}}\Big[(Q_\phi(s,a) - y(s,a,r,s'))^2\Big], \qquad \widehat{\mathcal{E}}_{\text{on}}(t) \;:=\; \mathbb{E}_{(s,a,r,s') \sim \mathcal{B}_{\text{on}}}\Big[(Q_\phi(s,a) - y)^2\Big],$$

with y the usual target. We then form the normalized gap

$$g_t \;:=\; \frac{\widehat{\mathcal{E}}_{\text{on}}(t) - \widehat{\mathcal{E}}_{\text{off}}(t)}{\widehat{\mathcal{E}}_{\text{off}}(t) + \eta},$$

for a small $\eta > 0$. A positive gap $g_t \gg 0$ indicates that the critic is substantially more inconsistent on online data than on offline data, suggesting either dataset shift or insufficient online coverage; this pushes the controller

toward smaller α_t (learn from online) but larger λ_t (guard against extrapolation). Conversely, $g_t \approx 0$ suggests that offline and online samples are similarly explained and that anchoring to D is not obviously harmful.

(2) Ensemble disagreement / epistemic uncertainty. We approximate uncertainty in Q-values by maintaining an ensemble $\{Q_{\phi^{(e)}}\}_{e=1}^{E}$ (or bootstrap heads). For states visited online, define the disagreement statistic

$$u_t \;:=\; \mathbb{E}_{s\sim d_{\pi_t}}\Big[\mathrm{Var}_{e\in[E]}\big(Q_{\phi^{(e)}}(s,\pi_t(s))\big)\Big].$$

Large u_t is interpreted as epistemic uncertainty under the current visitation distribution. Operationally, u_t serves two roles: (i) it tightens the safety filter by inflating uncertainty bonuses in $\mathrm{LCB}_t(\theta)$, and (ii) it encourages conservative regularization λ_t when disagreement is high.

(3) Return-based evidence. We also use the realized episode return R_t to update a running confidence interval for the current deployed configuration and, when explicit evaluations are available, for π_0 versus π_D. In particular, if we periodically deploy π_0 for evaluation episodes, we obtain a direct (noisy) estimate of $J(\pi_0)$ and can update r_t via a one-dimensional Bayesian model for ΔJ under sub-Gaussian noise. This optional evaluation mechanism respects the lower bound in Theorem 6.2: it does not "solve" near ties, but it makes explicit the cost of resolving them.

Controller parameterization over (α, λ). We discretize $[0,1]\times[0,\Lambda]$ into an ε_Θ-net Θ and run an experts algorithm on Θ restricted to the safe set S_t. To incorporate the regime score without overriding safety, we specify a prior preference distribution $p_t(\theta)$ that tilts weights before the EXP3-style update. One convenient choice is a separable log-linear model

$$p_t(\alpha,\lambda) \;\propto\; \exp\Big(\beta_\alpha\,(1-r_t)\,\alpha \;-\; \beta_\lambda\, r_t\,\lambda\Big),$$

interpreting $r_t \approx 1$ (offline already strong) as preferring smaller α (less need to replay) but larger λ (avoid degradation), and $r_t \approx 0$ as preferring larger α (more reliance on D) and smaller λ (allow improvement). The coefficients $\beta_\alpha, \beta_\lambda \geq 0$ merely shape the preference within the safe set and may be tuned coarsely; critically, the safety filter S_t is applied first.

Approximate construction of $\mathrm{LCB}_t(\theta)$. Assumption (A1) posits an oracle. Practically, we combine (i) an empirical return bound for configurations that have been deployed sufficiently often and (ii) an uncertainty penalty based on ensemble disagreement. For a fixed θ, let $\{R_i(\theta)\}_{i\leq t:\theta_i=\theta}$ be observed returns and $\widehat{\mu}_t(\theta)$ their mean. A simple bound is

$$\mathrm{LCB}_t(\theta) \;:=\; \widehat{\mu}_t(\theta) \;-\; b_t(\theta) \;-\; c_u\,\widehat{u}_t(\theta),$$

where $b_t(\theta)$ is a concentration term of order $\sqrt{\log(T|\Theta|/\delta)/n_t(\theta)}$ and $\widehat{u}_t(\theta)$ is the average disagreement along trajectories collected under θ. This form makes explicit the mechanism by which epistemic uncertainty shrinks the safe set early and gradually relaxes it as online evidence accumulates.

SAC/TD3 integration details. With SAC, we use the standard entropy-regularized objective and twin critics; the conservative regularizer is added to each critic loss. With TD3, we use clipped double Q, target policy smoothing, and delayed actor updates; again $\lambda_t \mathcal{R}(\phi)$ is added to each critic. In both cases, the replay mixture is implemented at the minibatch sampler, requiring no algorithmic modification to the base learner beyond access to two buffers. The net effect is that α_t controls how quickly the policy can move away from the dataset-induced fixed point, while λ_t controls how pessimistically the critic evaluates out-of-support actions; the controller then selects these knobs using only bandit feedback and the diagnostics above, subject to the stability-floor filter.

8 Experimental design: benchmarks, ablations, and stability diagnostics

We outline an experimental protocol intended to (i) validate the stability-floor claim empirically, (ii) isolate the contribution of each component of RegimeFlow, and (iii) stress-test the controller in regimes where $\Delta J := J(\pi_0) - J(\pi_D)$ is near zero and where the online visitation distribution departs from the dataset support.

Benchmarks and dataset families. We consider standard offline-to-online settings in which D is fixed and online interaction begins only after offline pretraining. For continuous control we use D4RL-style locomotion and manipulation tasks with multiple dataset qualities (random, medium, medium-replay, medium-expert, expert) to realize variation in coverage and in $J(\pi_D)$. To probe distribution shift more directly, we additionally include (when available) benchmarks with explicit environment parameters (e.g., friction, mass, actuation limits) so that we can hold $\mathcal{M}$ fixed for offline training and then modify a small subset of parameters for online fine-tuning while keeping reward scaling unchanged; we interpret this as "controlled dataset shift" rather than a different task. For high-dimensional observations, we include a limited set of pixel-based control domains with offline datasets that are known to be narrow-support; here the primary goal is to test whether the safe set S_t collapses appropriately early in online fine-tuning.

Protocols for estimating the offline baseline. We estimate $J(\pi_D)$ by averaging the trajectory returns in D (as in the global context) and esti-

mate $J(\pi_0)$ by M online evaluation rollouts prior to fine-tuning. We report $J^*_{\text{off}} = \max(J(\pi_0), J(\pi_D))$ and the empirical gap $\widehat{\Delta J}$. Since $J(\pi_D)$ can be sensitive to truncation and reward preprocessing, we enforce identical episode termination and reward normalization across (i) the logged dataset statistics, (ii) pretraining evaluation, and (iii) online deployment. We also report confidence intervals for $J(\pi_0)$ so that near-tie instances (where $0 \in \text{CI}(\Delta J)$) can be identified explicitly rather than implicitly.

Primary metrics: stability floor and performance. For stability, we report the *violation rate*

$$\text{Viol}(T) \;:=\; \frac{1}{T}\sum_{t=1}^{T} \mathbf{1}\{R_t < J^*_{\text{off}} - \varepsilon\}\,,$$

as well as the *maximum violation depth* $\max_{t\le T}(J^*_{\text{off}} - \varepsilon - R_t)_+$. Since the theoretical guarantee is on $J(\pi_t)$ rather than R_t, we additionally compute a smoothed version using windowed averages of returns, and we report how conclusions vary with the window length. For performance, we report cumulative return $\sum_{t=1}^{T} R_t$ and normalized area-under-curve (AUC) of the learning curve. When comparing to fixed-knob baselines, we also compute an empirical regret proxy relative to the best fixed safe knob selected in hindsight from the discretized net Θ:

$$\widehat{\text{Regret}}(T) \;:=\; \max_{\theta\in\Theta:\ \widehat{\text{Viol}}_\theta(T)=0} \sum_{t=1}^{T} \big(\widehat{\mu}_t(\theta) - R_t\big),$$

where $\widehat{\mu}_t(\theta)$ is estimated from repeated runs of the fixed-θ configuration.

Opposite-mismatch rate (regime/knob consistency). To strengthen the claim that the regime score r_t induces meaningful preferences without overriding safety, we introduce an *opposite-mismatch* diagnostic that measures how often the selected knobs oppose the direction suggested by the offline baseline. Fix thresholds $\underline{\alpha} < \overline{\alpha}$ and $\underline{\lambda} < \overline{\lambda}$. Using a ground-truth label $\text{sgn}(\Delta J)$ obtained from pre-fine-tuning evaluations (with a tie region declared when $|\widehat{\Delta J}| \le \tau$), we define

$$\text{Opp}(T) \;:=\; \frac{1}{T}\sum_{t=1}^{T} \mathbf{1}\left\{ \begin{array}{l} \widehat{\Delta J} > \tau \ \wedge\ \alpha_t \ge \overline{\alpha} \\ \quad \text{or} \\ \widehat{\Delta J} < -\tau \ \wedge\ \alpha_t \le \underline{\alpha} \end{array} \right\},$$

and analogously for λ_t with the direction reversed (when offline is strong we expect larger λ_t to prevent degradation). We report $\text{Opp}(T)$ both unconditionally and conditioned on $\theta_t \in S_t$ to separate "preference errors" from safety-filter effects. In addition, we plot $\mathbb{E}[\alpha_t \mid r_t]$ and $\mathbb{E}[\lambda_t \mid r_t]$ to verify monotonicity trends induced by the prior p_t.

Ablations. We include the following controlled ablations, each run under identical seeds and online budgets: (i) *No safety filter*: select θ_t by the same experts update but ignore S_t; (ii) *No regime prior*: set p_t uniform so that only bandit feedback drives selection; (iii) *No uncertainty penalty*: remove the disagreement term in $\mathrm{LCB}_t(\theta)$; (iv) *No ensemble*: use a single critic and only return-based concentration for LCB_t; (v) *Hard switching*: restrict $\alpha \in \{0, 1\}$ (and optionally a small discrete set of λ) to test Proposition 1 in near-tie instances; and (vi) *Discretization sensitivity*: vary ε_Θ (hence $|\Theta|$) to assess the $O(T\varepsilon_\Theta)$ effect empirically.

Sensitivity to near ties and dataset shift. To generate controlled near-tie instances, we construct behavior-mixture datasets $D(\rho)$ by mixing trajectories from two sources (e.g., medium and expert) with mixture weight ρ, and we select ρ so that $J(\pi_0)$ crosses $J(\pi_D)$ as ρ varies. We then evaluate how RegimeFlow behaves as $|\Delta J|$ shrinks relative to return noise, focusing on (a) the rate at which r_t concentrates and (b) whether the controller avoids committing prematurely to extreme α. For dataset shift, we use the same offline D but perturb online initial-state distributions and environment parameters, and we quantify shift by (i) increases in the TD-loss gap g_t and (ii) increases in disagreement u_t. We then test whether the safe set S_t contracts early (reducing risky choices) and whether the algorithm recovers performance without violating the stability floor.

Reporting. For each task family we report mean and standard error over seeds for AUC, Viol(T), maximum violation depth, Opp(T), and the evolution of $|S_t|/|\Theta|$. We also include empirical calibration plots for $\mathrm{LCB}_t(\theta)$ (coverage versus nominal $1 - \delta$) to diagnose when the stability-floor mechanism fails due to miscalibrated uncertainty rather than due to knob-selection noise.

9 Discussion and limitations

Dependence on value confidence bounds. Our stability statement ultimately reduces to the correctness of the lower-confidence bounds $\mathrm{LCB}_t(\theta)$ assumed in (A1). This dependence is not an artifact of our analysis: any mechanism that claims *high-probability* prevention of performance degradation must, implicitly or explicitly, certify that candidate updates are safe. In practice, constructing calibrated lower bounds for $J(\pi_t^\theta)$ in high-dimensional, nonlinear function approximation remains difficult. Common surrogates—bootstrap ensembles, disagreement penalties, and return-based concentration—can be systematically miscalibrated under distribution shift or when the critic is misspecified. When miscalibration occurs, the safety filter may admit unsafe θ (false positives) or, conversely, may reject nearly all θ (false

negatives), forcing the controller to behave overly conservatively.

Two concrete limitations follow. First, our guarantees require a union bound over both time and knob configurations; hence $\mathrm{LCB}_t(\theta)$ must be valid at level $\delta/(T|\Theta|)$. Achieving such a stringent simultaneous coverage typically induces wide intervals early in fine-tuning, which can shrink S_t dramatically. Second, the safe-set nonemptiness condition in Theorem 2 is not automatic: if the bound is overly pessimistic, we may have $S_t = \varnothing$ even when safe knobs exist. A practical fallback is to include a *do-nothing* configuration θ_{base} that freezes learning (or uses maximal regularization and maximal offline replay), together with a bound construction that certifies $\mathrm{LCB}_t(\theta_{\mathrm{base}}) \approx J^*_{\mathrm{off}}$ by direct online evaluation. This converts "empty safe set" into an explicit reversion to a baseline deployment, but it also highlights that some online evaluation is unavoidable if one wants an actionable safety filter.

What the stability floor does and does not protect. The constraint $J(\pi_t) \geq J^*_{\mathrm{off}} - \varepsilon$ is a *performance* guarantee, not a *behavioral* or *state-wise* safety guarantee. In particular, it does not preclude catastrophic states if their probability is small enough that the expected return remains above the floor. Moreover, since J^*_{off} can itself be low, the floor should be interpreted as "do not become worse than the best available offline reference" rather than as an absolute safety specification. In domains where safety is naturally expressed as a hard constraint (e.g., collision avoidance), one would need to incorporate explicit constraint costs and replace the floor by a constrained objective (e.g., CMDPs), together with confidence bounds for both reward and constraint value.

Beyond the same-MDP setting. Our formalization takes $\mathcal{M}$ to be fixed: offline and online interaction occur in the same discounted MDP, and only the visitation distribution changes due to learning. The experimental protocol in Section 8 already considers controlled parameter perturbations, but our theory does not cover the case where online fine-tuning occurs in $\mathcal{M}' \neq \mathcal{M}$ in a way that changes the optimality ordering of π_0 and π_D, or alters reward scaling. In such cases, the baseline J^*_{off} may cease to represent an attainable reference, and the regime score r_t—defined as a posterior over the sign of $\Delta J = J(\pi_0) - J(\pi_D)$—may be anchored to the wrong comparison.

A principled extension is to treat the online process as *nonstationary* and to replace (A1) by bounds that are robust to model drift, for example via distributionally robust MDP sets around empirical transition estimates, or via explicit change-point detection that triggers re-estimation of $J(\pi_0)$ and an updated baseline. At a minimum, if a shift detector indicates that the current return distribution under π_0 has changed, then the controller should regard ΔJ as time-dependent, ΔJ_t, and allow the prior over knobs to be reset rather than accumulated indefinitely.

Partial resets, continuing tasks, and intervention policies. Our controller is presented episodically, with full resets and an episode-level return R_t serving as the bandit feedback. Many practical systems instead operate in continuing time with only partial resets (or no resets), and performance is measured by average reward or by discounted return from a rolling start-state distribution. In such settings, a per-episode safety filter is not directly applicable: the deployed policy influences the future state distribution, and reverting to a baseline policy may not restore the system to the pre-intervention state distribution.

One direction is to equip RegimeFlow with an *intervention policy* π_{safe} that can be activated when diagnostics (e.g., disagreement u_t or TD-gap g_t) exceed thresholds, together with a notion of "partial reset" that restores a subset of state variables. Analytically, this suggests modeling the controller as operating over *segments* separated by interventions, where each segment has a bounded effective horizon and the stability constraint is required per segment. The corresponding confidence bounds must then track the value of knob choices under the evolving start-state distribution, which is a substantially harder problem than the stationary episodic case.

Toward transfer-aware regime scores. The regime score r_t was defined as a posterior probability that $\Delta J \geq 0$ and used only to induce a monotone preference over knobs. This is intentionally weak: safety is enforced solely via $\mathrm{LCB}_t(\theta)$. Nevertheless, when the online environment differs from the offline data-generating process, ΔJ is an incomplete descriptor of what should be preferred. A more informative regime variable would incorporate online evidence of *support mismatch*, e.g.,

$$\rho_t \;:=\; \Pr\big((s,a) \sim d^{\pi_t} \text{ lies in a low-density region under } D\big),$$

estimated via density models or representation-space distances. One may then define a transfer-aware posterior over latent regimes, such as "offline-competent but out-of-support" versus "offline-incompetent but in-support," and map these regimes to priors over (α, λ). Formally, this becomes a hierarchical model in which the knob prior depends on a latent variable z_t inferred from both returns and shift diagnostics; the bandit layer remains unchanged but may explore more efficiently by avoiding knobs that are predictably brittle under estimated mismatch.

Summary. The core message is that the controller viewpoint isolates two distinct burdens: (i) constructing reliable safety certificates (the LCB_t problem), and (ii) optimizing performance subject to those certificates (the safe experts problem). Our contribution addresses (ii) cleanly and admits transparent diagnostics; the principal remaining obstacles lie in (i) and in extending the formalism beyond stationary episodic MDPs to continuing, partially resettable, and transfer-shifted deployments.