Model-Based Reinforcement Learning for Control under Time-Varying Dynamics

Klemens Iten, Bruce Lee, Chenhao Li, Lenart Treven, Andreas Krause, Bhavya Sukhija

***Abstract*— Learning-based control methods typically assume stationary system dynamics, an assumption often violated in real-world systems due to drift, wear, or changing operating conditions. We study reinforcement learning for control under time-varying dynamics. We consider a continual model-based reinforcement learning setting in which an agent repeatedly learns and controls a dynamical system whose transition dynamics evolve across episodes. We analyze the problem using Gaussian process dynamics models under frequentist variation-budget assumptions. Our analysis shows that persistent non-stationarity requires explicitly limiting the influence of outdated data to maintain calibrated uncertainty and meaningful dynamic regret guarantees. Motivated by these insights, we propose a practical optimistic model-based reinforcement learning algorithm with adaptive data buffer mechanisms and demonstrate improved performance on continuous control benchmarks with non-stationary dynamics.**

I. INTRODUCTION

A. Motivation and Setting

Learning-based control methods have shown strong performance in complex dynamical systems [1]–[5]. In particular, model-based reinforcement learning (MBRL), where a dynamics model is learned from data and used for planning, has emerged as a sample-efficient approach for learning in the real-world [4]–[7]. Furthermore, MBRL methods enjoy strong theoretical guarantees for general nonlinear systems in the episodic [8]–[10], non-episodic [10], and safe learning [11] settings. However, a common assumption across these algorithms is that the system dynamics remain constant during learning. This assumption is often violated in real-world systems due to drift, wear, changing operating conditions, or environmental variation. Examples include robotic systems with hardware degradation, vehicles under varying loads, or systems operating across different regimes.

In such settings, naively applying existing MBRL techniques results in suboptimal performance (Section VI). In particular, with time-varying dynamics, data collected in the past may no longer be representative of the current system. As a result, reusing all historical data can lead to biased models and degraded control performance. This raises a fundamental question: *how should RL algorithms adapt to time-varying dynamics?*

All authors are with ETH Zürich, Rämistrasse 101, 8092 Zürich, Switzerland {kiten,trevenl,krausea,sukhijab}@ethz.ch

B. Lee, C. Li, and A. Krause are with the ETH AI Center at ETH Zürich {bruce.lee,chenhao.li}@ai.ethz.ch

We address this question and propose two MBRL algorithms, R-OMBRL and SW-OMBRL, for sample-efficient learning in time-varying nonlinear systems with continuous state-action spaces. Fundamental to both algorithms is the use of Bayesian models to learn an uncertainty-aware dynamics representation of the true system.

Moreover, we use the epistemic uncertainty of our learned dynamics model as an intrinsic reward to direct exploration. Furthermore, we address the challenge of time-varying dynamics by carefully selecting the data used for training the agent. In particular, we consider periodic resets of the data buffer (R-OMBRL) and sliding windows (SW-OMBRL) that retain only recent data.

Our key contributions are

- **Formulation:** We introduce an episodic model-based reinforcement learning framework for control under time-varying dynamics, together with a variation-budget model that captures temporal drift.
- **Theory:** We derive dynamic regret bounds for optimistic MBRL with restricted data buffers, showing how performance depends on the retained data horizon and the total variation in the dynamics.
- **Insight:** Our analysis reveals that limiting the influence of stale data is necessary to maintain calibrated uncertainty and achieve sublinear dynamic regret under non-stationarity.
- **Algorithms:** Based on the theoretical insights, we propose practical modifications that enable the use of NNs for learning in high-dimensional systems.
- **Experiments:** We demonstrate improved performance of the proposed methods over MBRL baselines on continuous control tasks with non-stationary dynamics.

II. RELATED WORK

A. Model-Based Reinforcement Learning

MBRL algorithms are commonly applied for learning in the real-world [5], [12], [13]. Crucially, model-based RL algorithms learn a dynamics model of the underlying true system, and use the learned model for planning. However, to avoid overexploitation of an inaccurate model, [8], [9] learn an uncertainty-aware model of the dynamics and use the uncertainty to enforce optimistic exploration. Recently, [10] propose a scalable and efficient optimistic exploration strategy where the model epistemic uncertainty is used as an intrinsic reward to direct exploration. Under common regularity assumptions on the true system, they provide sublinear regret bounds for the algorithm in the finite-horizon, discounted infinite-horizon, and average reward RL settings.

Nonetheless, crucial to their theoretical analysis is that the true system remains constant during learning. In this work, we go beyond this setting and tackle the more realistic problem of learning under non-stationarity.

B. Learning under Non-Stationarity

Learning in non-stationary environments has been extensively studied in various contexts. The most developed theory exists in bandit and Bayesian optimization settings. Time-varying GP bandits capture non-stationarity via stochastic drift in function space [14], [15], or using variation budgets [16], leading to dynamic regret guarantees. However, these algorithms maximize and observe immediate reward and do not tackle the more general RL setting, where high-dimensional policies are learned using trajectories acquired through rollouts on the true system.

In RL, non-stationarity has mainly been studied in tabular settings by controlling cumulative changes in rewards and transitions [17], or even without prior knowledge of the variation in model-free settings [18]. Notably, [19] propose a sliding-window based MBRL algorithm that builds on top of the seminal UCRL [20] algorithm for optimistic exploration. While similar to our approach in spirit, these methods tackle the finite state-action settings and do not consider the continuous case, which is ubiquitous in real-world systems.

III. PROBLEM FORMULATION

A. Control Objective

We consider finite-horizon control of a dynamical system with state space $\mathcal{X} \subset \mathbb{R}^{d_x}$ and action space $\mathcal{U} \subset \mathbb{R}^{d_u}$. The performance of a policy $\boldsymbol{\pi} : \mathcal{X} \to \mathcal{U}$ under dynamics $\boldsymbol{f}$ is measured by the finite-horizon return

$$J(\boldsymbol{\pi}, \boldsymbol{f}) = \mathbb{E}^{\boldsymbol{\pi}, \boldsymbol{f}}_{\boldsymbol{w}_{0:T-1}} \left[\sum_{t=0}^{T-1} r(\boldsymbol{x}_t, \boldsymbol{u}_t) \right], \tag{1}$$

where the expectation is taken over the Gaussian noise $\boldsymbol{w}_{0:T-1} \overset{iid}{\sim} \mathcal{N}(0, \sigma^2 \boldsymbol{I})$, the superscript $\boldsymbol{\pi}$ denotes that the inputs are selected under the policy $\boldsymbol{u}_t = \boldsymbol{\pi}(\boldsymbol{x}_t)$, and $\boldsymbol{f}$ denotes that the state evolves according to $\boldsymbol{x}_{t+1} = \boldsymbol{f}(\boldsymbol{x}_t, \boldsymbol{u}_t) + \boldsymbol{w}_t$ starting from initial state $\boldsymbol{x}_0$. The summand is given by the reward function $r : \mathcal{X} \times \mathcal{U} \to [0, R_{\max}] \subset \mathbb{R}$ evaluated at the states and inputs along this trajectory.

B. Time-Varying Dynamics

We consider an unknown discrete-time, time-varying dynamical system. For an episode $n \in \{1, \dots, N\}$, the system dynamics are given by

$$\boldsymbol{x}_{n,t+1} = \boldsymbol{f}^*_n(\boldsymbol{x}_{n,t}, \boldsymbol{u}_{n,t}) + \boldsymbol{w}_{n,t}, \quad t = 0, \dots, T-1, \tag{2}$$

where $\boldsymbol{f}^*_n$ denotes the (unknown) dynamics at episode n. We study the *episodic* RL setting, where at the beginning of each episode the initial state is sampled from an initial state distribution ρ, i.e., $\boldsymbol{x}_0 \sim \rho$. Then we rollout a policy $\boldsymbol{\pi} \in \Pi$ for T steps in the environment and collect the resulting trajectory for learning. The goal of the RL agent is to find the policy that maximizes the cumulative sum of rewards, i.e., solve (1) for the dynamics $\boldsymbol{f}^*_n$. For simplicity of notation, consider the case where the dynamics $\boldsymbol{f}^*_n$ are fixed within an episode.[1]

At the beginning of each episode n, the agent selects a policy $\boldsymbol{\pi}_n$, which is executed for T steps under $\boldsymbol{f}^*_n$. The corresponding optimal policy is defined as $\boldsymbol{\pi}^*_n = \arg\max_{\boldsymbol{\pi} \in \Pi} J(\boldsymbol{\pi}, \boldsymbol{f}^*_n)$. Since the dynamics vary across episodes, the optimal policy also generally varies and depends on the current episode.

C. Performance Metric

To evaluate performance in the presence of time-varying dynamics, we consider *dynamic regret*, defined as

$$R_N = \sum_{n=1}^{N} \left[J(\boldsymbol{\pi}^*_n, \boldsymbol{f}^*_n) - J(\boldsymbol{\pi}_n, \boldsymbol{f}^*_n) \right]. \tag{3}$$

Dynamic regret compares the learner against the sequence of episode-wise optimal policies $(\boldsymbol{\pi}^*_n)_{n \geq 1}$. This is in contrast to *static regret*, which compares against a single fixed policy and is inappropriate in non-stationary environments where the optimal policy may change over time.

D. Non-Stationarity Model

To quantify temporal variation in the dynamics, we adopt a frequentist model based on reproducing kernel Hilbert spaces (RKHS). We assume that for all episodes n, each component of the dynamics $\boldsymbol{f}^*_n$ lies in a common RKHS $\mathcal{H}_k$ with bounded norm:

$$\mathcal{H}^{d_x}_{k,B} \overset{\text{def}}{=} \{ \boldsymbol{f} \mid \|f_j\|_{\mathcal{H}_k} \leq B, j = 1, \dots, d_x \}$$

for some kernel satisfying $k(\boldsymbol{z}, \boldsymbol{z}) \leq \sigma_{\max}$ for all $\boldsymbol{z} \in \mathcal{Z}$.

To capture non-stationarity, we impose a *variation budget* on the sequence of dynamics:

$$\sum_{n=1}^{N-1} \|\boldsymbol{f}^*_{n+1} - \boldsymbol{f}^*_n\|_{\mathcal{H}^{d_x}_k} \leq P_N, \tag{4}$$

where $\|\cdot\|_{\mathcal{H}^{d_x}_k}$ denotes the d_x-dimensional RKHS norm. In particular, for $\boldsymbol{f} \in \mathcal{H}^{d_x}_k$, let

$$\|\boldsymbol{f}\|_{\mathcal{H}^{d_x}_k} = \sum_{j=1}^{d_x} \|f_j\|_{\mathcal{H}_k}.$$

This assumption allows the dynamics to evolve over time while controlling the total amount of variation. Generally, P_N will increase with N as the dynamics evolve.

IV. GAUSSIAN PROCESS DYNAMICS MODELS

A. GP Model

For our theoretical analysis, we model the unknown dynamics with Gaussian processes (GPs). GPs are a particularly promising class of models because they are expressive, Bayesian, and also enable thorough theoretical analysis of

[1] Our results can be extended to the setting where the dynamics also change during the episode.

the algorithm. Furthermore, they have a closed-form solution for the posterior mean and epistemic uncertainty.

Let $\boldsymbol{z}_{n,t} \stackrel{\text{def}}{=} (\boldsymbol{x}_{n,t}, \boldsymbol{u}_{n,t}) \in \mathcal{Z} \stackrel{\text{def}}{=} \mathcal{X} \times \mathcal{U}$. For each state dimension $j \in \{1, \ldots, d_x\}$, we model the j-th component of the dynamics, $f^*_{n,j} : \mathcal{Z} \to \mathbb{R}$ using an independent GP with kernel k. We assume that this kernel satisfies $k(\boldsymbol{z}, \boldsymbol{z}) \leq \sigma_{\max}$ for all $\boldsymbol{z} \in \mathcal{Z}$. Consider then the trajectory at episode n as follows

$$\boldsymbol{\tau}_n \stackrel{\text{def}}{=} \{(\boldsymbol{z}_{n,t}, \boldsymbol{y}_{n,t}) \mid 0 \leq t < T-1\},$$

where the state-action tuples and the observations,

$$\boldsymbol{z}_{n,t} \stackrel{\text{def}}{=} (\boldsymbol{x}_{n,t}, \boldsymbol{\pi}_n(\boldsymbol{x}_{n,t})) \text{ and } \boldsymbol{y}_{n,t} \stackrel{\text{def}}{=} \boldsymbol{x}_{n,t+1},$$

come from the rollout of $\boldsymbol{\pi}_n$ on $\boldsymbol{f}^*_n$. Fitting the GP to the data collected in episodes $m, m+1, \ldots, \ell$ results in the posterior mean and covariance functions at arbitrary $\boldsymbol{z} \in \mathcal{Z}$ given by

$$\mu_{m:\ell,j}(\boldsymbol{z}) = \boldsymbol{k}_{m:\ell}(\boldsymbol{z})^\top (\boldsymbol{K}_{m:\ell} + \sigma^2 \boldsymbol{I})^{-1} \mathbf{y}^j_{m:\ell}, \tag{5}$$
$$\sigma^2_{m:\ell,j}(\boldsymbol{z}) = k(\boldsymbol{z}, \boldsymbol{z}) - \boldsymbol{k}_{m:\ell}(\boldsymbol{z})^\top (\boldsymbol{K}_{m:\ell} + \sigma^2 \boldsymbol{I})^{-1} \boldsymbol{k}_{m:\ell}(\boldsymbol{z}),$$

where $\mathbf{y}^j_{m:\ell}$ denotes the vector of j-th components of the observed next states over the episodes m to ℓ, while $\boldsymbol{k}_{m:\ell}(\boldsymbol{z}) = \left[k(\boldsymbol{z}, \boldsymbol{z}_{i,t})\right]_i$ and $\boldsymbol{K}_{m:\ell} = \left[k(\boldsymbol{z}_{h,t}, \boldsymbol{z}_{i,t})\right]_{h,i}$ are the data kernel vector and matrix over the transitions $\boldsymbol{z}_{h,t}, \boldsymbol{z}_{i,t}$ in the dataset recorded over episodes $m : \ell$, i.e., $\mathcal{D}_{m:\ell} \stackrel{\text{def}}{=} \bigcup_{s=m}^{\ell} \boldsymbol{\tau}_s$.

B. Maximum Information Gain

Crucial to our theoretical results is the *maximum information gain* of kernel k [21],

$$\gamma_N(k) = \max_{\mathcal{A} \subset \mathcal{X} \times \mathcal{U}; |\mathcal{A}| \leq NT} \frac{1}{2} \log\det\left(\boldsymbol{I} + \sigma^{-2} \boldsymbol{K}(\mathcal{A})\right). \tag{6}$$

where $\boldsymbol{K}(\mathcal{A}) = [k(\boldsymbol{z}_{h,t}, \boldsymbol{z}_{i,t})]_{h,i}$ for all pairs $\boldsymbol{z}_{h,t}, \boldsymbol{z}_{i,t} \in \mathcal{A}$. The maximum information gain γ_N is a measure of the complexity for learning $\boldsymbol{f}^*$ from N episodes and is sublinear for many kernels.[2]

C. Stale Data Problem

In stationary settings, it is natural to fit the GP using all previously observed data. In the present non-stationary case, doing so introduces a systematic mismatch: older samples were generated by past dynamics $\boldsymbol{f}^*_s$, whereas control at episode n depends on the current dynamics $\boldsymbol{f}^*_n$.

If the model is fitted to all past data, then the posterior mean $\boldsymbol{\mu}_{1:n-1}$ over $n-1$ episodes is biased toward outdated dynamics. Importantly, the GP variance $\boldsymbol{\sigma}^2_{1:n-1}$ does not capture this temporal mismatch, since it reflects only statistical uncertainty under a stationary-function assumption. Thus, even if $\boldsymbol{\sigma}_{1:n-1}$ is small, the true prediction error $\|\boldsymbol{f}^*_n(\boldsymbol{z}) - \boldsymbol{\mu}_{1:n-1}(\boldsymbol{z})\|$ may remain large due to drift. This motivates explicitly restricting the retained data horizon.

D. Forgetting Mechanisms

We consider two mechanisms to control the influence of stale data.

[2]e.g., $\mathcal{O}(\log^{d_x+d_u+1}(NT))$ for the squared exponential (RBF) kernel, $\mathcal{O}((d_x+d_u)\log(NT))$ for the linear kernel; see [21] for more detail.

Full resets: Fix a reset period $H \in \mathbb{N}$. Let

$$n_0(n) \stackrel{\text{def}}{=} H \left\lfloor \frac{n-1}{H} \right\rfloor + 1$$

such that at $n_0(n)-1$ the data buffer is emptied, i.e., "stale" data is removed from the buffer. Therefore, at episode n, the model is fitted only on data collected since the last reset, i.e., the dataset $\mathcal{D}_{n_0(n):n-1}$.

Sliding window: Fix a window size $w \in \mathbb{N}$. At episode n, the model is fitted only on the most recent w episodes, i.e. on the dataset $\mathcal{D}_{n-w:n-1}$.

E. Calibration Under Drift

In the stationary case, where the dynamics do not change across episodes, the true dynamics $\boldsymbol{f}^*_j$ is fixed. Standard GP concentration results [21]–[23] then imply that, there exists $\beta_n(\delta) = B + \sigma\sqrt{2(\gamma_n + d_x \log(1/\delta))}$ such that with probability at least $1-\delta$

$$|f^*_j(\boldsymbol{z}) - \mu_{1:n-1,j}(\boldsymbol{z})| \leq \beta_n(\delta)\, \sigma_{1:n-1,j}(\boldsymbol{z}) \;\forall \boldsymbol{z} \in \mathcal{Z}. \tag{7}$$

Here, $\beta_n(\delta)$ represents the width of the confidence interval around the mean within which the true system is captured. Effectively, in this setting $\beta_n(\delta)\,\sigma_{1:n-1,j}(\boldsymbol{z})$ is a proxy for the error of our model estimate $\mu_{1:n-1,j}(\boldsymbol{z})$.

However, these results hold for the stationary setting. For time-varying dynamics, the temporal drift introduces an additional bias term that is not captured by the inequality above. In the following Lemma, we extend these confidence bounds to the non-stationary case.

Lemma 1 (Lemmas 1 and 2 of [16] adapted for vector function $\boldsymbol{f}^*$)**.** *Assume the dynamics satisfy the RKHS regularity assumptions of Section III. Consider the GP mean and variance estimates $(\mu_{m:n-1,j}, \sigma_{m:n-1,j})$ with either the full resetting mechanism $m = n_0(n)$ or the sliding window $m = n-w$. It holds with probability at least $1-\delta$ that for any episode $n \in \{1, \ldots, N\}$, any $\boldsymbol{z} \in \mathcal{Z}$, and any $j = 1, \ldots, d_x$*

$$\left|f^*_{n,j}(\boldsymbol{z}) - \mu_{m:n-1,j}(\boldsymbol{z})\right| \leq \beta_n(\delta, n-m)\, \sigma_{m:n-1,j}(\boldsymbol{z}) + \xi_{n-m} \sum_{s=m}^{n-1} \left\|f^*_{s+1,j} - f^*_{s,j}\right\|_{\mathcal{H}_k}, \tag{8}$$

where the confidence parameter is given by

$$\beta_n(\delta, n-m) = \begin{cases} B + \sigma\sqrt{2\left(\gamma_{n-m} + d_x \log(\frac{1}{\delta})\right)} & \text{if full reset} \\ B + \sigma\sqrt{2\left(\gamma_{n-m} + d_x \log(\frac{nT}{\delta})\right)} & \text{if sliding window.} \end{cases}$$

and the coefficient on the drift term is

$$\xi_{n-m} = \frac{2\sigma_{\max}\sqrt{(n-m)(1+\sigma^2)\gamma_{n-m}}}{\sigma^2}.$$

Equation (8) reveals two distinct sources of error:

1) the standard epistemic uncertainty term $\beta\sigma$, and
2) an additive temporal bias term caused by the accumulated drift in the dynamics over the past $n-m$ episodes, i.e., since the last reset at episode $n_0(n)$ or within the sliding window of size w.

V. OPTIMISTIC MODEL-BASED RL UNDER NON-STATIONARY DYNAMICS

The analysis in Section IV shows that, under time-varying dynamics, the calibrated uncertainty depends explicitly on the retained data horizon. Motivated by this insight, we adapt an optimistic model-based reinforcement learning framework [10] to the non-stationary setting by restricting the data buffer used for model learning and policy optimization. To this end, we first demonstrate how the model error propagates through the control cost, and use this to propose an optimistic synthesis algorithm.

A. Performance Difference Bound

We now relate model error to control performance akin to [24]. To unify the reset and sliding-window cases, let $\boldsymbol{\mu}_{m:n-1}$ and $\boldsymbol{\sigma}_{m:n-1}$ denote an arbitrary dynamics model and uncertainty model constructed using (5) from the most recent retained data horizon from episodes $m : n-1$, with m as defined in Lemma 1.

For a policy $\boldsymbol{\pi}$, let $J(\boldsymbol{\pi}, \boldsymbol{f}_n^*)$ and $J(\boldsymbol{\pi}, \boldsymbol{\mu}_{m:n-1})$ denote the expected finite-horizon returns from (1) under the true dynamics and the learned model, respectively. We further define the trajectory-wise accumulated epistemic uncertainty for a range of episodes $m, \dots, \ell$ as

$$\Sigma_{m:\ell}(\boldsymbol{\pi}, \boldsymbol{f}) \overset{\text{def}}{=} \mathbb{E}^{\boldsymbol{\pi}, \boldsymbol{f}}_{\boldsymbol{w}_{0:T-1}} \left[\sum_{t=0}^{T-1} \left\| \boldsymbol{\sigma}_{m:\ell}(\boldsymbol{x}_{n,t}, \boldsymbol{\pi}(\boldsymbol{x}_{n,t})) \right\|_2 \right], \tag{9}$$

where the expectation is taken over trajectories generated by policy $\boldsymbol{\pi}$ under the dynamics $\boldsymbol{f}$ with the same process noise as in (2). The bound is then as follows.

Lemma 2 (Performance difference bound)**.** *Assume the dynamics satisfy the regularity assumptions of Section III. Let $(\mu_{m:n-1}, \sigma_{m:n-1})$ be the GP model fit with either resetting or forgetting. Conditioned on the success event of Lemma 1, it holds for every episode n and every policy $\boldsymbol{\pi}$ that*

$$\begin{aligned} \left| J(\boldsymbol{\pi}, \boldsymbol{f}_n^*) - J(\boldsymbol{\pi}, \boldsymbol{\mu}_{m:n-1}) \right| &\le B_{n,m} \\ &+ \lambda_{n,m} \min\{\Sigma_{m:n-1}(\boldsymbol{\pi}, \boldsymbol{f}_n^*), \Sigma_{m:n-1}(\boldsymbol{\pi}, \boldsymbol{\mu}_{m:n-1})\}. \end{aligned} \tag{10}$$

where

$$\lambda_{n,m} \overset{\text{def}}{=} \frac{R_{\max} T}{\sigma} \beta_n(\delta, n-m) \tag{11}$$

scales the contribution of epistemic uncertainty, and

$$B_{n,m} \overset{\text{def}}{=} \xi_{n-m} \frac{R_{\max} T^2}{\sigma} \sum_{s=m}^{n-1} \left\| \boldsymbol{f}_{s+1}^* - \boldsymbol{f}_s^* \right\|_{\mathcal{H}_k} \tag{12}$$

collects the temporal drift over the recorded $n-m$ episodes.

Thus, the policy performance difference decomposes into two terms: an uncertainty term, governed by the accumulated posterior standard deviation along the trajectory, and a temporal bias term, governed by the amount of drift contained in the retained dataset. Most importantly, the term $B_{n,m}$ is independent of the policy. We use this key insight to integrate the optimistic MBRL framework in the non-stationary case.

B. Optimistic MBRL Framework

At each episode n, the algorithm maintains a policy $\boldsymbol{\pi}_n$, a dynamics model $\mathcal{M}_n = (\boldsymbol{\mu}_{m:n-1}, \boldsymbol{\sigma}_{m:n-1})$, and a data buffer $\mathcal{D}_{m:n-1}$. Given the current model, the policy is updated by solving an optimistic planning problem under the learned mean dynamics:

$$\boldsymbol{\pi}_n = \arg\max_{\boldsymbol{\pi}} J(\boldsymbol{\pi}, \boldsymbol{\mu}_{m:n-1}) + \lambda_{n,m} \Sigma_{m:n-1}(\boldsymbol{\pi}, \boldsymbol{\mu}_{m:n-1}). \tag{13}$$

Here, the uncertainty term $\Sigma_{m:n-1}(\boldsymbol{\pi}, \boldsymbol{\mu}_{m:n-1})$ acts as an intrinsic reward and directs exploration toward poorly modeled regions of the state-action space, and $\lambda_{n,m}$ from (11) is a positive constant which is used to trade off maximizing the reward and model uncertainty.

In stationary settings, such optimism-based methods train both the policy and the dynamics model on all data collected so far. In the present non-stationary setting, however, using all previously collected data can induce substantial bias due to stale data. Our main algorithmic modification is therefore to restrict the data buffer to recent data only.

C. Algorithms

We consider two mechanisms for limiting stale data.

R-OMBRL (reset-based, Algorithm 1): The data buffer is reset every H episodes. Hence, at episode n, both the dynamics model and the policy are trained only on data collected since the most recent reset.

SW-OMBRL (sliding window, Algorithm 2): The data buffer retains only the most recent w episodes, discarding older samples. Hence, at episode n, both the dynamics model and the policy are trained only on transitions from this window.

In both cases, restricting the data buffer limits the temporal bias identified in (8,10).

We empirically validate our theoretical results for Gaussian Processes in Fig. 1. We compare R-OMBRL and SW-OMBRL against SOMBRL from [10], which is a MBRL algorithm that optimizes for the objective in (13), but without restricting the data buffer to train the GP for the model $(\boldsymbol{\mu}_{1:n-1}, \boldsymbol{\sigma}_{1:n-1})$. We use the Pendulum environment from Gym [25], adapted to a non-stationary setting where at one point, the maximum applicable action $\boldsymbol{u}_t$ decays rapidly.

In this application with GP dynamics, we find that R-OMBRL and SW-OMBRL both outperform SOMBRL: They recover performance and adapt to the changed dynamics, whereas the baseline model $(\boldsymbol{\mu}_{1:n-1}, \boldsymbol{\sigma}_{1:n-1})$, trained on all past data, remains biased toward outdated dynamics and fails to adapt.

VI. THEORETICAL ANALYSIS

A. Main Regret Bound

Combining the calibration result with the performance difference bound yields a dynamic regret bound. We state an informal version here and defer the full theorem and proof to the extended manuscript.[3]

[3]Available under: `https://arxiv.org/abs/2604.02260`

Algorithm 1 *R-OMBRL:* Reset-based optimistic MBRL

Require: Horizon T, reset period H

Initialize $\boldsymbol{\pi}_\theta$, $\mathcal{M}_0$, empty data buffer $\mathcal{D} = \emptyset$
for episode $n = 1, 2, \ldots, N$ **do**
  if $n \bmod H = 0$ **then**
    $\mathcal{D} \leftarrow \emptyset$
  end if
  Collect trajectory $\boldsymbol{\tau}_n$ using $\boldsymbol{\pi}_{\boldsymbol{\theta}}$
  $\mathcal{D} \leftarrow \mathcal{D} \cup \boldsymbol{\tau}_n$
  Update $\mathcal{M}_{\boldsymbol{\phi}}$ using $\mathcal{D}$
  Update $\boldsymbol{\pi}_{\boldsymbol{\theta}}$ by (13) using $(\mathcal{M}_{\boldsymbol{\phi}}, \mathcal{D})$
end for

Algorithm 2 *SW-OMBRL:* Sliding-window optimistic MBRL

Require: Horizon T, window size w

Initialize $\boldsymbol{\pi}_{\boldsymbol{\theta}}$, $\mathcal{M}_0$, empty data buffer $\mathcal{D} = \emptyset$
for episode $n = 1, 2, \ldots, N$ **do**
  Collect trajectory $\boldsymbol{\tau}_n$ using $\boldsymbol{\pi}_{\boldsymbol{\theta}}$
  $\mathcal{D} \leftarrow \mathcal{D} \cup \boldsymbol{\tau}_n$
  Remove data older than w episodes from $\mathcal{D}$
  Update $\mathcal{M}_{\boldsymbol{\phi}}$ using $\mathcal{D}$
  Update $\boldsymbol{\pi}_{\boldsymbol{\theta}}$ by (13) using $(\mathcal{M}_{\boldsymbol{\phi}}, \mathcal{D})$
end for

Theorem 1 (Regret bound)**.** *Assume the dynamics satisfy the RKHS regularity and variation-budget assumptions of Section III. Then, the optimistic MBRL algorithm of* (13) *with either resetting or a sliding window achieves dynamic regret that satisfies*

$$R_N = \tilde{\mathcal{O}}\left(N\sqrt{\frac{\gamma_p^3}{p}} + \gamma_p p^{3/2} P_N\right), \tag{14}$$

where $p = H$ for resetting and $p = w$ for the sliding window, and $\tilde{\mathcal{O}}$ hides the dependence of the episode length and factors that are logarithmic in N.

B. Interpretation for R-OMBRL

In the following, we interpret the regret bound for the R-OMBRL algorithm. The analysis is analogous for SW-OMBRL.

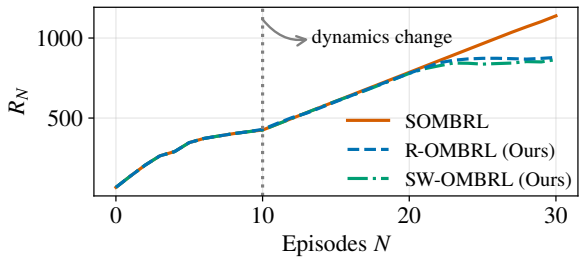

Fig. 1. Learning curves for the setting with GP dynamics on the Pendulum environment. We report the mean cumulative regret R_N over 5 random seeds. At episode $N = 10$, we induce a change in dynamics by limiting the maximum applicable action $\boldsymbol{u}_{N,t}$ to half its original value over time. This leads to a linear regret for the stationary SOMBRL baseline, while R-OMBRL and SW-OMBRL adapt to the change in dynamics.

The regret bound of Theorem 1 decomposes into a learning term and a drift term. For the reset case (R-OMBRL), $N\sqrt{\gamma_H^3/H}$ decreases with larger H, reflecting improved performance from reusing more data, whereas $\gamma_H H^{3/2} P_N$ increases with H, reflecting the growing influence of stale data if the reset period increases. Hence H controls a bias-variance-type trade-off:

- small H: strong adaptivity to changing dynamics, but less data per model fit,
- large H: more data for training the model but with a larger bias.

In the stationary case, OMBRL [10] – where $H = N$ – has a regret of

$$R_N = \mathcal{O}\left(\sqrt{N\gamma_N^3} + \gamma_N N^{3/2} P_N\right),$$

which is only sublinear if $P_N = 0$, i.e., no variation in the dynamics takes place. In our case, we can achieve sublinear regret for $P_N > 0$ by carefully selecting H. For instance, similar to [16], by selecting $H \propto \gamma_N^{1/4} N^{1/2}$, we get

$$R_N = \mathcal{O}\left(\gamma_N^{11/8}(1 + P_N) N^{3/4}\right).$$

For a specific total variation rate, e.g., $P_N \propto \log(N)$, this results in a regret $R_N = \mathcal{O}\left(\gamma_N^{11/8} N^{3/4} \log(N)\right)$, which is sublinear for common kernels such as the exponential and linear kernel [21]. Moreover, while the current algorithm requires H to be set *a priori* based on the duration N, for specific choices of γ_N and P_N (e.g., the exponential kernel example above), we can convert our regret to an anytime regret bound by applying the doubling trick [26].

C. Practical Modification

Our theoretical analysis focuses on GP models, where we can guarantee calibration of our learned model and bound the complexity of learning the dynamics, i.e., γ_n. However, GPs scale poorly to high-dimensional systems. Furthermore, they are also computationally expensive.[4] Accordingly, most MBRL algorithms use neural networks for representing the dynamics. In particular, works such as [9], [27] use Bayesian neural networks (BNNs), specifically deep ensembles [28], to learn an uncertainty-aware dynamics model.

To this end, we propose practical modifications to R-OMBRL and SW-OMBRL, which enable learning with BNNs. Moreover, we employ a scalable parametrized uncertainty-aware dynamics model $\mathcal{M}_{\boldsymbol{\phi}} = (\boldsymbol{\mu}_{\boldsymbol{\phi}}, \boldsymbol{\sigma}_{\boldsymbol{\phi}})$ and instantiate it using deep ensembles. The model is trained to maximize the data likelihood using stochastic gradient descent. Similar to [10], we train a parameterized policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}$ using the MBPO algorithm [29] on the optimistic objective (13). To handle non-stationarity, we restrict the data buffer through resets or sliding windows as described in Section V.

In contrast to the theoretically derived exploration weight $\lambda_{n,m}$ from (11), which depends explicitly on confidence bounds and the retained data horizon, it can be beneficial in

[4] Cubic in the number of data points.

practice to treat this coefficient as a tunable parameter. For instance, [30] propose an automatic tuning mechanism for this exploration weight and adapt it online. Moreover, [31] study this trade-off empirically in the model-based setting and demonstrate that adaptive or manually tuned exploration weights can perform well in practice.

Furthermore, we additionally employ soft resets of both model and policy parameters. In particular, at predefined intervals, we update using randomly sampled new parameters $(\boldsymbol{\theta}_0, \boldsymbol{\phi}_0)$ by

$$\boldsymbol{\phi} \leftarrow (1-\alpha_1)\boldsymbol{\phi} + \alpha_1 \boldsymbol{\phi}_0, \quad \boldsymbol{\theta} \leftarrow (1-\alpha_2)\boldsymbol{\theta} + \alpha_2 \boldsymbol{\theta}_0. \quad (15)$$

We ablate the effects of (α_1, α_2) in the extended manuscript.[5]

VII. Experiments

A. Setup

We evaluate the proposed methods on continuous control benchmarks from Gym [25] and MuJoCo [32]. To study learning under changing dynamics, we introduce controlled non-stationarity via parameter drift.

We evaluate R-OMBRL and SW-OMBRL, which restrict the data buffer through resets and sliding windows, respectively. As a baseline, we use a stationary optimistic model-based RL method (OMBRL) [10], a SOTA MBRL method for stationary dynamics, which trains on all collected data without any forgetting mechanism. Additional implementation details and ablations are provided in the extended manuscript.[5]

B. Non-Stationary Environments

We evaluate on the Pendulum, HalfCheetah [33], and Hopper [34] environments by modifying the implementation from [25] by introducing time-varying dynamics via an episode-dependent decay of actuator strength. Specifically, we scale the maximum admissible control as

$$\bar{u}_n = \exp(-an)\,(u_{\max} - u_{\min}) + u_{\min}, \quad (16)$$

where $a \geq 0$ controls the rate of change. Here, we treat the reset period H and window size w as tunable hyperparameters and tune the internal reward weight $\lambda_{n,m}$ using the method from [30]. For the policy and model parameters $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$, we employ soft resets as described in (15) with $\alpha_1 = \alpha_2 = 0.2$.

Fig. 2 shows results across environments for three decay rates. We report dynamic cumulative regret with respect to an estimate of the optimal policy, obtained by running an RL algorithm (SAC [35]), independently for fixed actuator strengths until convergence.

Across all settings, the stationary baseline fails to track the evolving dynamics, leading to rapidly increasing regret. In contrast, both R-OMBRL and SW-OMBRL significantly reduce regret accumulation by limiting the influence of stale data. Overall, we conclude that restricting the data buffer improves robustness under non-stationarity and leads to consistently better tracking of the underlying time-varying system.

[5]Available under: `https://arxiv.org/abs/2604.02260`

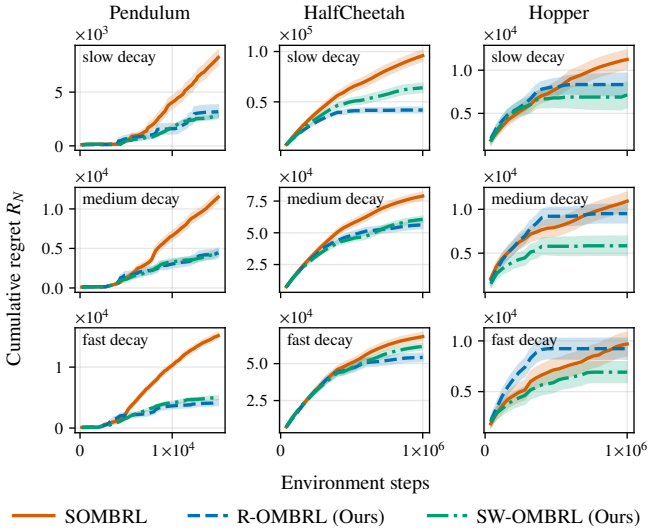

Fig. 2. Dynamic regret under time-varying dynamics for different decay rates. The stationary baseline SOMBRL accumulates large regret due to stale data, while R-OMBRL and SW-OMBRL improve tracking by restricting the data buffer. We report the mean regret compared to an estimate of the optimal performance over five seeds with standard error.

C. Different Environments

We evaluate the methods across multiple MuJoCo environments [32], comparing the stationary baseline (SOMBRL) with R-OMBRL and SW-OMBRL. The setup follows the previous experiment.

Fig. 3 shows training under initially stationary dynamics, followed by a transition to time-varying dynamics induced by the decay in (16). The top row illustrates the evolution of the environment parameter, while the bottom rows report dynamic regret averaged over five seeds.

Under stationary dynamics, all methods perform similarly. After the onset of non-stationarity, the baseline accumulates substantially higher regret, while both R-OMBRL and SW-OMBRL adapt effectively by restricting the data buffer.

These results demonstrate that data buffer adaptation improves performance across environments and scales to higher-dimensional control tasks.

D. Hardware Experiments

We evaluate the proposed methods on a real-world RC car platform following the setup of [36]. The system consists of a high-torque racecar capable of highly dynamic maneuvers, including drifting, with the state capturing position, orientation, and velocities, and control inputs given by steering and throttle.

The task is a dynamic parking maneuver, where the car must rotate and park at a target location. At high actuator strength, the optimal behavior involves aggressive sliding and drifting. To induce non-stationarity, we introduce an episode-dependent decay of the maximum throttle, gradually transforming the task from a drift-based maneuver into a standard parking problem.

We compare R-OMBRL against the stationary baseline (SOMBRL) in Fig. 4. The top row shows rollouts of the

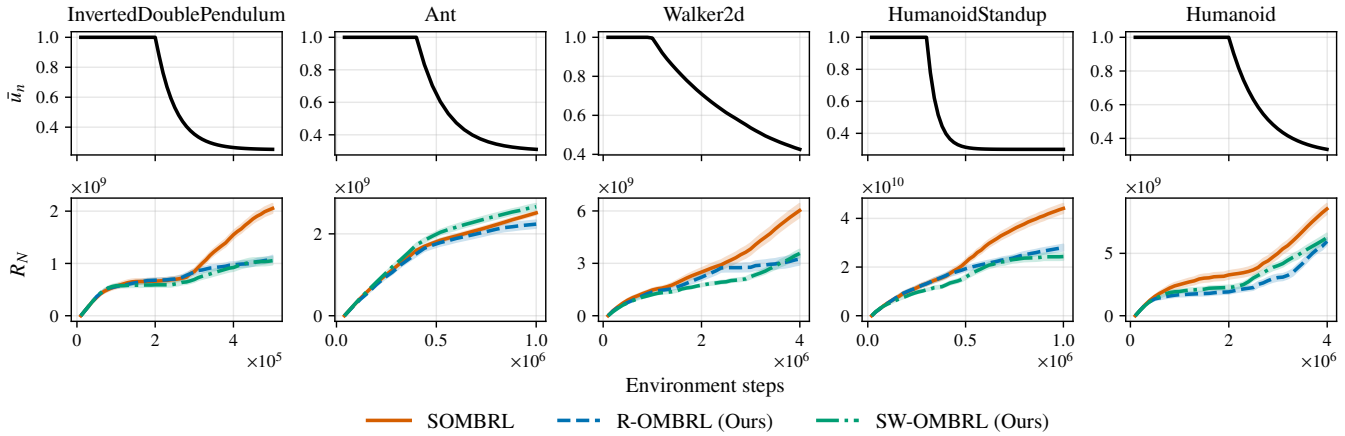

Fig. 3. Dynamic regret across multiple environments. The top row shows the evolution of the maximum admissible torque $\bar{u}_n$ and its decay over environment training steps, while the bottom rows report the cumulative regret R_n averaged over five seeds with standard error. Under stationary dynamics, all methods perform similarly. After the onset of non-stationarity, R-OMBRL and SW-OMBRL significantly reduce regret compared to the stationary baseline.

learned policies after 30 episodes. While SOMBRL fails to adapt and is unable to complete the parking maneuver, R-OMBRL successfully performs the task. The bottom row reports the return on the real system over episodes. Before the onset of non-stationarity, both methods perform similarly, whereas after the throttle decay, R-OMBRL adapts more effectively and achieves higher returns.

These results mirror the simulation findings: restricting the data buffer improves adaptation to evolving dynamics. The stationary baseline accumulates outdated transitions generated under earlier, high-throttle regimes, leading to degraded performance. In contrast, R-OMBRL discards stale data and adapts to the current system, resulting in more stable and consistent behavior.

VIII. CONCLUSIONS AND FUTURE WORKS

A. Conclusions

In this work, we study model-based reinforcement learning under time-varying dynamics. Our analysis shows that, under a variation-budget model of the dynamics, persistent non-stationarity requires explicitly limiting the influence of stale data to maintain calibrated uncertainty and achieve meaningful dynamic regret guarantees. Motivated by this insight, we propose practical optimistic MBRL algorithms, R-OMBRL and SW-OMBRL, based on data buffer resets and sliding windows. While similar ideas have been proposed by prior work, such as [19] for finite state-action spaces, to the best of our knowledge, we are the first to provide dynamic regret guarantees for the general setting of continuous spaces and non-linear dynamics. Furthermore, we also empirically validate the proposed methods on high-dimensional continuous control benchmarks as well as on real-world hardware. We show both in simulation and the real-world that R-OMBRL and SW-OMBRL significantly outperform the stationary MBRL baseline. This shows that data buffer adaptation is essential, in both theory and practice, for robust learning and control in a non-stationary system.

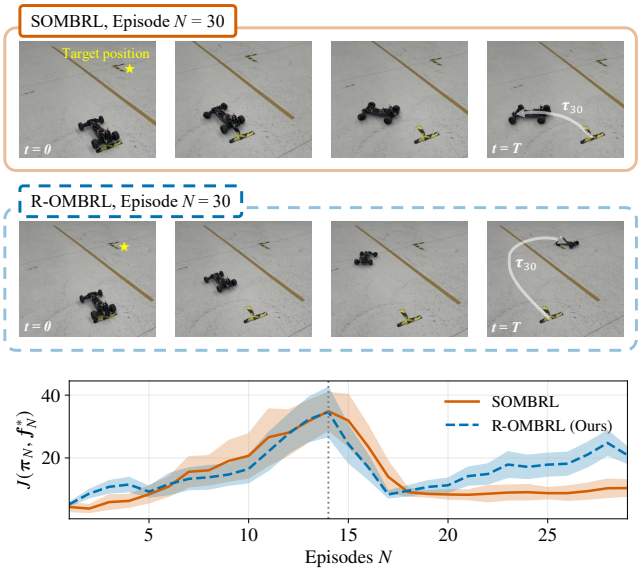

Fig. 4. Hardware experiments on a real RC car. The task is a parking maneuver that transitions from drift-based behavior to standard parking as the maximum throttle decays at $N = 14$ episodes. The top row shows rollouts after $N = 30$, where R-OMBRL successfully completes the task while SOMBRL fails to adapt. The bottom row shows the mean return and error bands along trajectories τ_N on the real system over episodes, averaged over 6 random seeds. Restricting the data buffer enables adaptation to changing dynamics and improves performance compared to the stationary baseline.

B. Future Works

An important direction concerns selecting the reset period or window size adaptively based on the observed data could further improve performance, as [15] do in the Bayesian optimization setting. In our experimental evaluation, we focus on parameter decay, which captures gradual performance degradation in many practical scenarios and satisfies the variation-budget assumptions. Our methods could be applied more broadly, evaluating them under alternative patterns such as abrupt shifts or cyclic variations. Finally, extending the framework to non-episodic settings with continuously evolving dynamics remains an important open problem.

IX. ACKNOWLEDGEMENTS

B. Lee was supported by a postdoctoral fellowship and C. Li by a doctoral fellowship from ETH AI Center. B. Sukhija was supported by ELSA (European Lighthouse on Secure and Safe AI) funded by the European Union under grant agreement No. 101070617. This project has received funding from the Swiss National Science Foundation under NCCR Automation, grant agreement 51NF40 180545. Numerical simulations were performed on the ETH Zürich Euler cluster. Parts of this text were revised with the assistance of a large language model to aid or polish writing and to improve grammar and clarity; the authors remain responsible for all content.

REFERENCES

[1] J. Hwangbo, J. Lee, A. Dosovitskiy, D. Bellicoso, V. Tsounis, V. Koltun, and M. Hutter, "Learning agile and dynamic motor skills for legged robots," *Science robotics*, 2019.

[2] A. Spiridonov, F. Buehler, M. Berclaz, V. Schelbert, J. Geurts, E. Krasnova, E. Steinke, J. Toma, J. Wuethrich, R. Polat, W. Zimmermann, P. Arm, N. Rudin, H. Kolvenbach, and M. Hutter, "Spacehopper: A small-scale legged robot for exploring low-gravity celestial bodies," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 3464–3470.

[3] K. Zakka, B. Tabanpour, Q. Liao, M. Haiderbhai, S. Holt, J. Y. Luo, A. Allshire, E. Frey, K. Sreenath, L. A. Kahrs *et al.*, "Mujoco playground," *arXiv preprint arXiv:2502.08844*, 2025.

[4] D. Hafner, J. Pasukonis, J. Ba, and T. Lillicrap, "Mastering diverse control tasks through world models," *Nature*, 2025.

[5] H. Zheng, B. Sukhija, C. Li, K. Iten, A. Krause, and R. K. Katzschmann, "Learning soft robotic dynamics with active exploration," *arXiv preprint arXiv:2510.27428*, 2025.

[6] T. M. Moerland, J. Broekens, A. Plaat, and C. M. Jonker, "Model-based reinforcement learning: A survey," *Foundations and Trends in Machine Learning*, 2023.

[7] B. Hoffman, J. Cheng, C. Li, and S. Coros, "Learning more with less: Sample efficient model-based rl for loco-manipulation," 2025. [Online]. Available: https://arxiv.org/abs/2501.10499

[8] S. Kakade, A. Krishnamurthy, K. Lowrey, M. Ohnishi, and W. Sun, "Information theoretic regret bounds for online nonlinear control," *NeurIPS*, 2020.

[9] S. Curi, F. Berkenkamp, and A. Krause, "Efficient model-based reinforcement learning through optimistic policy search and planning," *NeurIPS*, 2020.

[10] B. Sukhija, L. Treven, C. Sferrazza, F. Dörfler, P. Abbeel, and A. Krause, "SOMBRL: Scalable and optimistic model-based RL," *Conference on Neural Information Processing Systems (NeurIPS)*, 2025. [Online]. Available: https://openreview.net/forum?id=eGfi5k7RP6

[11] Y. As, B. Sukhija, L. Treven, C. Sferrazza, S. Coros, and A. Krause, "Actsafe: Active exploration with safety constraints for reinforcement learning," *ICLR*, 2025.

[12] P. Wu, A. Escontrela, D. Hafner, P. Abbeel, and K. Goldberg, "Daydreamer: World models for physical robot learning," in *CORL*. PMLR, 2023.

[13] N. Hansen, Y. Lin, H. Su, X. Wang, V. Kumar, and A. Rajeswaran, "Modem: Accelerating visual model-based reinforcement learning with demonstrations," *arXiv preprint arXiv:2212.05698*, 2022.

[14] I. Bogunovic, J. Scarlett, and V. Cevher, "Time-Varying Gaussian Process Bandit Optimization," Jan. 2016, arXiv:1601.06650 [stat]. [Online]. Available: http://arxiv.org/abs/1601.06650

[15] P. Brunzema, A. v. Rohr, F. Solowjow, and S. Trimpe, "Event-Triggered Time-Varying Bayesian Optimization," *Transactions on Machine Learning Research*, 2025. [Online]. Available: https://openreview.net/forum?id=WEYMCLu8u7

[16] X. Zhou and N. Shroff, "No-regret algorithms for time-varying bayesian optimization," in *CISS*, 2021, pp. 1–6.

[17] R. Ortner, P. Gajane, and P. Auer, "Variational Regret Bounds for Reinforcement Learning," in *Proceedings of The 35th Uncertainty in Artificial Intelligence Conference*. PMLR, Aug. 2020, pp. 81–90. [Online]. Available: https://proceedings.mlr.press/v115/ortner20a.html

[18] C.-Y. Wei and H. Luo, "Non-stationary Reinforcement Learning without Prior Knowledge: an Optimal Black-box Approach," in *Proceedings of Thirty Fourth Conference on Learning Theory*. PMLR, Jul. 2021, pp. 4300–4354. [Online]. Available: https://proceedings.mlr.press/v134/wei21b.html

[19] W. C. Cheung, D. Simchi-Levi, and R. Zhu, "Reinforcement learning for non-stationary markov decision processes: The blessing of (more) optimism," in *ICML*. PMLR, 2020, pp. 1843–1854.

[20] P. Auer, T. Jaksch, and R. Ortner, "Near-optimal regret bounds for reinforcement learning," *NeurIPS*, vol. 21, 2008.

[21] N. Srinivas, A. Krause, S. M. Kakade, and M. Seeger, "Gaussian Process Optimization in the Bandit Setting: No Regret and Experimental Design," *IEEE Transactions on Information Theory*, vol. 58, no. 5, pp. 3250–3265, May 2012, arXiv:0912.3995 [cs]. [Online]. Available: http://arxiv.org/abs/0912.3995

[22] S. R. Chowdhury and A. Gopalan, "On Kernelized Multi-armed Bandits," *ICML*, 2017. [Online]. Available: https://proceedings.mlr.press/v70/chowdhury17a.html

[23] J. Rothfuss, B. Sukhija, T. Birchler, P. Kassraie, and A. Krause, "Hallucinated adversarial control for conservative offline policy evaluation," in *UAI*, 2023, pp. 1774–1784. [Online]. Available: https://proceedings.mlr.press/v216/rothfuss23a.html

[24] S. Kakade and J. Langford, "Approximately Optimal Approximate Reinforcement Learning," in *Proceedings of the Nineteenth International Conference on Machine Learning*, ser. ICML '02. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., Jul. 2002, pp. 267–274.

[25] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "OpenAI Gym," 2016. [Online]. Available: https://arxiv.org/abs/1606.01540

[26] L. Besson and E. Kaufmann, "What doubling tricks can and can't do for multi-armed bandits," *arXiv preprint arXiv:1803.06971*, 2018.

[27] K. Chua, R. Calandra, R. McAllister, and S. Levine, "Deep reinforcement learning in a handful of trials using probabilistic dynamics models," *NeurIPS*, 2018.

[28] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," *NIPS*, 2017. [Online]. Available: https://arxiv.org/abs/1612.01474

[29] M. Janner, J. Fu, M. Zhang, and S. Levine, "When to trust your model: Model-based policy optimization," *Conference on Neural Information Processing Systems (NeurIPS)*, 2019. [Online]. Available: https://arxiv.org/abs/1906.08253

[30] B. Sukhija, S. Coros, A. Krause, P. Abbeel, and C. Sferrazza, "MaxinfoRL: Boosting exploration in reinforcement learning through information gain maximization," in *The Thirteenth International Conference on Learning Representations*, 2025. [Online]. Available: https://openreview.net/forum?id=R4q3cY3kQf

[31] K. Iten, L. Treven, B. Sukhija, F. Dorfler, and A. Krause, "Sample-efficient and scalable exploration in continuous-time RL," in *The Fourteenth International Conference on Learning Representations*, 2026. [Online]. Available: https://openreview.net/forum?id=PJdMrK79Mo

[32] E. Todorov, T. Erez, and Y. Tassa, "MuJoCo: A physics engine for model-based control," *International Conference on Intelligent Robots and Systems (IROS)*, 2012. [Online]. Available: https://ieeexplore.ieee.org/document/6386109

[33] P. Wawrzyński, "A cat-like robot real-time learning to run," *International Conference on Adaptive and Natural Computing Algorithms (ICANNGA)*, 2009. [Online]. Available: https://doi.org/10.1007/978-3-642-04921-7_39

[34] T. Erez, Y. Tassa, and E. Todorov, "Infinite-horizon model predictive control for periodic tasks with contacts," *Robotics: Science and systems (RSS)*, 2012.

[35] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *International conference on machine learning*. Pmlr, 2018, pp. 1861–1870. [Online]. Available: https://proceedings.mlr.press/v80/haarnoja18b

[36] J. Rothfuss*, B. Sukhija*, L. Treven*, F. Dörfler, S. Coros, and A. Krause, "Bridging the sim-to-real gap with bayesian inference," *IEEE Int. Conf. on Intelligent Robots and Systems (IROS)*, 2024.