Efficient Heteroscedastic Bayesian Optimization for Risk-Aware AutoRL

Mingxuan Che, Tsung-Yuan Tseng, Theresa Eimer, Marius Lindauer, Alexander von Rohr

Keywords: Bayesian optimization, reinforcement learning, AutoRL

Summary

Reinforcement learning (RL) learning outcomes can be highly stochastic, and both expected performance and variability often depend strongly on hyperparameter (HP) configurations. This makes HP optimization (HPO) brittle: outcome variability translates into a practical risk of unsuccessful training after HPO.

We propose risk-aware HPO through a mean-variance objective, dubbed efficient and risk-averse heteroscedastic Bayesian optimization (ERAHBO). It models both mean and variance, and improves sample efficiency via adaptive re-sampling rather than a fixed replication budget per HP configuration. We show that our adaptive strategy improves efficiency and HPO performance across a wide range of RL environments. We additionally provide an offline dataset of RL learning outcomes mapping HP configurations to post-training policy returns across 50 repeated runs per HP configuration to evaluate risk-aware HPO methods.

Contribution(s)

1. We propose efficient and risk-averse heteroscedastic Bayesian optimization (ERAHBO), an adaptive re-sampling formulation of RAHBO (Makarova et al., 2021) that improves practical sample efficiency in HPO for RL. We provide sublinear regret guarantees along with an empirical comparison to common Bayesian optimization (BO) baselines.
   Context: To integrate risk into the optimization process, we need to repeatedly evaluate each hyperparameter configuration. RAHBO sets a fixed number of evaluations for each configuration, which makes it computationally expensive on a runtime-intense domain like reinforcement learning.
2. We provide a dataset of RL learning outcomes mapping hyperparameters to post-training policy returns for 50 runs, enabling reproducible benchmarking of risk-aware hyperparameter optimization methods.
   Context: Recent HPO performance datasets such as those provided in HPO-RL-Bench (Shala et al., 2024) and ARLBench (Becktepe et al., 2026) include repeated evaluations for each configuration, but the number of seeds (up to ten) per hyperparameter configuration can be insufficient for a reliable estimation of heteroscedastic variance across the hyperparameter space. Our dataset addresses this by providing 50 seeds per configuration.

Efficient Heteroscedastic Bayesian Optimization for Risk-Aware AutoRL

Mingxuan Che[1,*], **Tsung-Yuan Tseng**[3,4,*], **Theresa Eimer**[1,2], **Marius Lindauer**[1,2], **Alexander von Rohr**[5]
{m.che,t.eimer,m.lindauer}@ai.uni-hannover.de,
tsung.yuan.tseng@cariad.technology, alexander.von.rohr@utn.de

[1]**Leibniz University Hannover** [2]**L3S Research Center** [3]**CARIAD SE**
[4]**University of Wuppertal** [5]**University of Technology Nuremberg**

[*] Equal contribution; order decided by a fair coin flip

Abstract

Reinforcement learning (RL) has shown remarkable success across a wide range of complex tasks. However, RL outcomes can be highly stochastic, and both expected performance and variability often depend on hyperparameter (HP) configurations. We propose efficient and risk-averse heteroscedastic Bayesian Optimization (ERAHBO), a Bayesian optimization method that models both the mean and variance of learning outcomes as functions of the HP configurations. ERAHBO aims to identify HP configurations that achieve high average return while reducing variability across training runs, and it improves the sample efficiency of the HP optimization via adaptive re-sampling rather than a fixed budget per HP. Empirical evaluations across diverse RL algorithms and environments demonstrate that ERAHBO generally outperforms both risk-neutral and risk-averse baselines, delivering improved sample efficiency for risk-averse returns.

1 Introduction

The performance of reinforcement learning (RL) algorithms is often highly dependent on the choice of hyperparameters (HPs) (Berkenkamp et al., 2016; Islam et al., 2017; Henderson et al., 2018; Zhang et al., 2021; Eimer et al., 2023; Obando-Ceron et al., 2024). Moreover, in most RL experiments, inherent randomness arises from e.g., stochastic task behavior, exploratory choices, or hardware nondeterminism (Zhuang et al., 2022). Recent work shows that even very good HP configurations may still lead to a high degree of randomness in learning outcomes (Dierkes et al., 2025).

Automated RL (AutoRL) aims to make RL work reliably out-of-the-box by automating, for example, the setting of HPs. However, the stochasticity of RL training makes automated hyperparameter optimization brittle: the configuration that appears best during HPO can fail to reproduce its performance in subsequent runs, and high-variance HP configurations can yield poor policies with nontrivial probability, requiring costly retraining. In practice, we would like HPO to return configurations that *reliably* achieve strong performance across runs, supporting reproducible and cost-effective RL experimentation. Therefore, it is an important research question how we can efficiently extract HP configurations that consistently lead to good performance, i.e., high average return with low variability across repeated training runs.

Bayesian optimization (BO) is a sample-efficient, model-based approach to HPO that is well suited to expensive black-box objectives and is widely used as an automated alternative to manual tuning and search heuristics (Bischl et al., 2023). However, classical BO approaches typically optimize

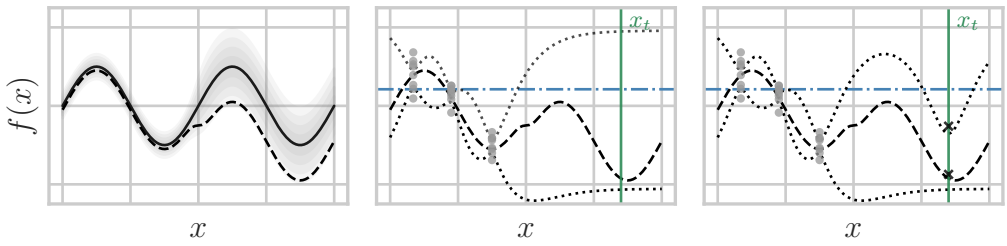

Figure 1: Adaptive re-sampling in ERHABO. **Left:** Example objective with two maxima at different noise levels, and the MV objective (dashed). **Middle:** Upper and lower confidence bounds for the MV objective (dotted) together with the incumbent LCB (blue dash-dot) after three configurations; the next query x_t (chosen via UCB) is shown by the green vertical line. **Right:** After two additional samples at x_t, the UCB falls below the incumbent LCB, so ERHABO does not need to re-sample.

the mean of the objective function, in RL typically the expected final return. This ignores training stability across runs for each hyperparameter configuration, i.e., its *risk*.

A natural way to incorporate this notion into HPO is to use *risk-aware* Bayesian optimization methods that explicitly model both the mean performance and the input-dependent variability of outcomes, and optimize a risk-sensitive objective. In particular, risk-averse heteroscedastic Bayesian optimization (RAHBO; Makarova et al. (2021)) provides a principled sequential BO approach that maintains separate surrogate models for the mean and variance of the return, and uses a risk-aware acquisition function. RAHBO can therefore be used for risk-aware HP tuning. However, in its original form, estimating the variance model relies on a *static* number of evaluations (replications) per HP configuration, which can be inefficient in RL where each run is computationally costly.

We propose a novel adaptive replication strategy for RAHBO that determines when to acquire more samples of a configuration. The key idea is to allocate runs where they matter most: configurations that are promising under a risk-aware objective but still ambiguous due to uncertainty, rather than spending a uniform replication budget everywhere (see Fig. 1). Our resulting method is an **efficient risk-averse heteroscedastic Bayesian optimization (ERAHBO)** algorithm, designed for the high degrees of variance and simultaneously high computational loads we see in RL.

To support the study of risk in AutoRL, we curate a new dataset of RL learning outcomes for different algorithms and environments which enables systematic analysis of heteroscedasticity in RL hyperparameters and benchmarking of risk-aware HPO methods.

2 Problem formulation

We consider a discounted Markov decision process (MDP). A (stochastic) policy $\pi(a \mid s)$ induces a distribution over trajectories $\tau \sim \pi(\tau)$. The performance of a fixed policy π is its expected discounted return $J(\pi) := \mathbb{E}_{a\sim\pi}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)\right]$, where $\gamma \in (0, 1)$ is the discount factor and $r(s, a)$ is the reward function.

In HPO for RL, the decision variable is a hyperparameter configuration $x \in \mathcal{X} \subset \mathbb{R}^d$ of a learning algorithm $\mathcal{A}$, and the outcome is the trained policy produced by running $\mathcal{A}$ with x. Because RL training is inherently stochastic (e.g., random initialization, exploration, and stochasticity encountered during training), running $\mathcal{A}$ with a fixed configuration x induces a distribution over learned policies. We model the learning outcome as $\pi \sim \mathcal{A}(\pi; x)$. Crucially, the randomness we consider is the randomness of the *learning outcome* π (and thus of $J(\pi)$) induced by training. We formulate risk-aware HPO as maximizing expected learning outcome while penalizing variability across

learning outcomes:

$$x^* \in \operatorname*{argmax}_{x \in \mathcal{X}} \mathbb{E}_{\pi \sim \mathcal{A}(\pi;x)}\big[J(\pi)\big] - \alpha \mathbb{V}ar_{\pi \sim \mathcal{A}(\pi;x)}\big[J(\pi)\big], \tag{1}$$

where $\alpha \geq 0$ defines the trade-off between mean and variance. For notational simplicity, define the mean performance

$$f(x) := \mathbb{E}_{\pi \sim \mathcal{A}(\pi;x)}\big[J(\pi)\big]. \tag{2}$$

In Bayesian optimization (BO), we consider a sequential interaction problem with unknown objective $f : \mathcal{X} \to \mathbb{R}$. In each BO round t, the algorithm chooses $x_t \in \mathcal{X}$, runs training once to obtain $\pi_t \sim \mathcal{A}(\pi; x_t)$, and observes one realization of the learning outcome

$$y_t := J(\pi_t) = f(x_t) + \xi(x_t), \tag{3}$$

where $\xi(x_t)$ is a zero-mean noise term capturing the stochasticity of the learning algorithm $\mathcal{A}$ when trained with hyperparameters x_t (i.e., variability across independent executions of the RL training with the same HPs x_t), and is independent across rounds t. In the following, we drop the subscript t when referring to a general x or when its meaning is clear from the context. Consequently,

$$\mathbb{V}ar[\xi(x)] = \mathbb{V}ar_{\pi \sim \mathcal{A}(\pi;x)}\big[J(\pi)\big]. \tag{4}$$

Following Makarova et al. (2021), we assume that f belongs to a reproducing kernel Hilbert space (RKHS) $\mathcal{H}_\kappa$ induced by a kernel $\kappa(\cdot, \cdot)$ satisfying $\kappa(x, x') \leq 1$ for all $x, x' \in \mathcal{X}$, and that $\|f\|_\kappa \leq \mathcal{B}_f$ for some $\mathcal{B}_f > 0$. We further assume $\xi(x)$ is sub-Gaussian with a (possibly input-dependent) variance proxy $\rho^2(x)$ uniformly bounded as $\rho(x) \in [\underline{\rho}, \overline{\rho}]$ for constants $\overline{\rho} \geq \underline{\rho} > 0$.

Following Sani et al. (2012), Makarova et al. (2021) and Dai et al. (2023), the mean-variance objective with variance proxy $\rho^2(x)$ is

$$\mathrm{MV}(x) = f(x) - \alpha \rho^2(x), \tag{5}$$

where a fixed $\alpha \geq 0$ expresses a user defined risk aversion.

We seek a sequence of evaluations $\{x_t\}_{t=1}^T$ that attains high values of the risk-averse objective $\mathrm{MV}(x)$ over T BO rounds. Since each round t collects k_t independent samples (training runs) using x_t, we evaluate performance via the sample-count cumulative regret

$$R_T := \sum_{t=1}^{T} k_t \Big(\mathrm{MV}(x^*) - \mathrm{MV}(x_t)\Big), \quad \text{where} \quad x^* \in \arg\max_{x \in \mathcal{X}} \mathrm{MV}(x). \tag{6}$$

3 Related work

This section gives a brief overview of BO for hyperparameter optimization (HPO), focusing on settings with stochastic and configuration-dependent outcomes, and then connects these ideas to HPO for RL, where training variability is often substantial.

Grid search and random search are simple baselines for HPO, but they become inefficient as the dimensionality of the HP space grows (Bergstra & Bengio, 2012). Bayesian optimization (Garnett, 2023) improves sample efficiency for costly black-box objectives by fitting a surrogate model and selecting evaluations sequentially (Bergstra et al., 2011; Snoek et al., 2012; Springenberg et al., 2016). It is the backbone of many widely used HPO approaches (Akiba et al., 2019; Lindauer et al., 2022; Song et al., 2024).

The substantial run-to-run variability of RL training has been documented in prior work (Henderson et al., 2018; Obando-Ceron et al., 2024) and motivates systematic and efficient approaches to HPO.

HPO for RL is an emerging field (see Parker-Holder et al. (2022) for an overview), but poses challenges due to high computational overhead and stochastic learning outcomes (Agarwal et al., 2021; Eimer et al., 2023) arising from training instabilities (Dierkes et al., 2025). In this context, optimizing expected performance alone, often approximated via a small number of training runs (Parker-Holder et al., 2020; Wan et al., 2022), may not capture reliability across runs. Our goal is to provide an objective and optimization strategy that better targets reliable HP configurations in RL.

Many classical BO methods assume homoscedastic noise and mismatched noise assumptions can degrade BO performance. Heteroscedastic Gaussian process models capture input-dependent noise explicitly (Binois et al., 2018; 2019), but modeling variance alone does not prevent BO from selecting configurations with undesirable high variability in the learning outcomes. To address this, risk-aware BO methods optimize objectives that explicitly trade off mean performance and variability. Prior work considers mean-variance objectives for heteroscedastic BO (Makarova et al., 2021; Dai et al., 2023). In particular, RAHBO (Makarova et al., 2021) uses a fixed evaluation budget per configuration to estimate variability, which is inefficient since it allocates significant resources to evaluate under-performing configurations. Dai et al. (2023) addresses adaptive evaluation in a batch setting. In this paper, we evaluate RAHBO for HPO for RL problems and extend it with an adaptive evaluation strategy that speeds up the optimization process across many HPO tasks.

4 Method

This section describes our approach for risk-aware HPO under heteroscedastic training outcomes. We begin with the essential modeling and acquisition components of RAHBO (Makarova et al., 2021), and then present ERAHBO, an adaptive resampling strategy, together with a corresponding sublinear regret bound.

4.1 RAHBO: risk-averse heteroscedastic Bayesian optimization

We briefly summarize the components of risk-averse heteroscedastic Bayesian optimization (RAHBO) (Makarova et al., 2021) that we build on. The key idea of RAHBO is to model both the performance f and variance proxy ρ^2 as separate Gaussian processes (GPs) to derive confidence bounds for a risk-averse variant of the Gaussian process upper confidence bound (GP-UCB) algorithm (Srinivas et al., 2010).

RAHBO obtains mean and variance estimation to condition the GPs by repeatedly evaluating a configuration. In iteration t, it selects a configuration x_t and performs a fixed number k of independent evaluations, yielding observations $\{y_t^i\}_{i=1}^k$. The sample mean and variance are estimated as

$$\hat{m}(x_t) = \frac{1}{k}\sum_{i=1}^{k} y_t^{(i)}, \quad \hat{s}^2(x_t) = \frac{1}{k-1}\sum_{i=1}^{k}(y_t^{(i)} - \hat{m}(x_t))^2, \tag{7}$$

which serve as noisy observations of $f(x_t)$ and $\rho^2(x_t)$ to condition the two GPs.

Let $\mathrm{UCB}_t^f(x)$ denote an upper confidence bound for $f(x)$, and $\mathrm{LCB}_t^{\mathrm{var}}(x)$ a lower confidence bound for $\rho^2(x)$, constructed from the two GP posteriors with appropriately chosen confidence parameters as in Makarova et al. (2021). RAHBO selects the next query by maximizing an upper bound on $\mathrm{MV}(x)$,

$$x_t \in \arg\max_{x\in\mathcal{X}} \ \mathrm{UCB}_t^f(x) \ - \ \alpha\,\mathrm{LCB}_t^{\mathrm{var}}(x). \tag{8}$$

Under standard regularity assumptions, RAHBO achieves sublinear regret for the mean-variance objective with a fixed sample budget k per iteration. For more details, see Appendix A.

Our method retains the same modeling and risk-aware acquisition structure, but replaces the fixed budget k with an adaptive re-sampling strategy that allocates evaluations based on estimated uncertainty, improving practical efficiency in high-cost settings such as RL.

4.2 ERAHBO: efficient risk-averse heteroscedastic Bayesian optimization

To resolve the limitation of RAHBO, we propose a confidence-based adaptive rule to automatically determine replications for $k \in [k_{min}, k_{max}]$ in each BO iteration.

4.2.1 Confidence-based resampling

While repeated sampling improves the estimation accuracy of both the mean and variance, it is not necessary to improve the estimate of the configuration once it can be certified as suboptimal. We construct the high-probability confidence bounds for the MV objective as

$$\mathrm{UCB}_t^{\mathrm{MV}}(x) := \mathrm{UCB}_t^{f}(x) - \alpha\,\mathrm{LCB}_t^{\mathrm{var}}(x), \quad \mathrm{LCB}_t^{\mathrm{MV}}(x) \quad := \mathrm{LCB}_t^{f}(x) - \alpha\,\mathrm{UCB}_t^{\mathrm{var}}(x). \tag{9}$$

The core idea of ERAHBO is to stop re-sampling a configuration once we know it is worse than the incumbent with high probability.

Incumbent lower bound We define the incumbent lower bound at iteration t as

$$B_t := \max_{x \in \mathcal{D}_{t-1}} \mathrm{LCB}_t^{\mathrm{MV}}(x), \tag{10}$$

where $\mathcal{D}_{t-1}$ denotes the set of previously evaluated configurations. The quantity B_t represents a high-confidence lower bound on the best MV value observed so far and serves as a conservative benchmark against which newly queried configurations are compared.

Stopping criterion At iteration t, the algorithm selects a configuration

$$x_t \in \arg\max_{x \in \mathcal{X}} \mathrm{UCB}_t^{\mathrm{MV}}(x), \tag{11}$$

and begins collecting repeated evaluations at x_t. Let k denote the current number of samples collected at x_t. As k increases, the uncertainty of the mean-variance estimate decreases due to the effective observation noise scaling as $\rho^2(x_t)/k$, leading to progressively tighter confidence bounds.

Sampling at x_t is terminated once the following condition is satisfied:

$$\mathrm{UCB}_t^{\mathrm{MV}}(x_t; k) \le B_t, \tag{12}$$

or once a predefined maximum number of evaluations $k_{\max}$ is reached. This stopping rule admits a clear interpretation. If even the optimistic estimate of the MV objective at x_t is no larger than the pessimistic estimate of the best competing configuration, then with high probability, x_t cannot outperform the incumbent and further replication is unnecessary. Conversely, as long as the confidence interval of $\mathrm{MV}(x_t)$ overlaps with the incumbent lower bound, additional evaluations may be informative and are therefore allocated. Appendix B shows that if a configuration is suboptimal relative to the incumbent lower bound, then ERAHBO will stop sampling it after finitely many samples.

4.2.2 Sublinear regret guarantee

We show that the sublinear regret guarantees of RAHBO (Makarova et al., 2021) carry over to a setting with varying replication counts. Under the standard RKHS smoothness and sub-Gaussian noise assumptions used in RAHBO, ERAHBO preserves the sublinear regret guarantees of RAHBO when the number of replications per BO round is allowed to vary but remains uniformly bounded, $k_t \in [k_{\min}, k_{\max}]$. In particular, with high probability, the sample-count cumulative regret satisfies

$$R_T = \tilde{O}\Big(k_{\max}\sqrt{T\gamma_T} + \alpha k_{\max}\sqrt{T\Gamma_T}\Big), \tag{13}$$

where γ_T and Γ_T are the information-gain terms for the mean and variance surrogate models, respectively. Thus, bounded adaptive replication does not change the no-regret behavior: whenever the corresponding RAHBO information-gain terms are sublinear, ERAHBO has vanishing average sample-count regret. Appendix A gives the full theorem statement, assumptions, and proof.

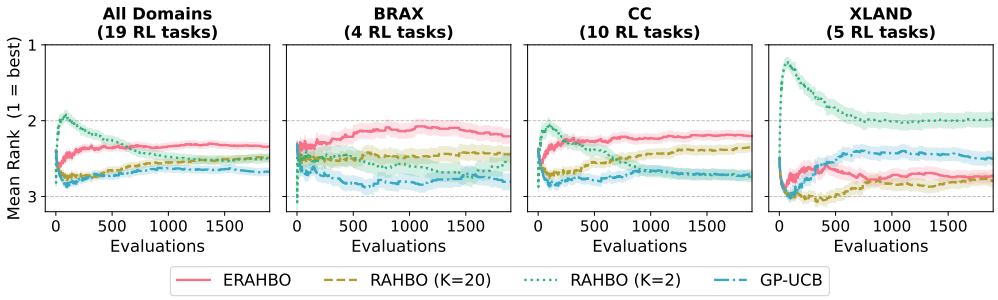

Figure 2: Mean-variance regret ranks aggregated by domain with lines and shaded area showing the mean and standard error, respectively. ERAHBO performs best overall. We observe different optimal values of k in RAHBO across domains.

5 Experiments

We validate our approach on three popular RL algorithms (DQN (Mnih et al., 2015), SAC (Haarnoja et al., 2018), and PPO (Schulman et al., 2017)) on Brax (Freeman et al., 2021), XLand-Minigrid (Nikulin et al., 2023) and Classic Control environments (Lange, 2022).

BO baselines. We compare our method against two baselines. GP-UCB (Srinivas et al., 2010) implemented by the work of Makarova et al. (2021) is our risk-neutral baseline. RAHBO (Makarova et al., 2021) is the state-of-the-art risk-averse BO algorithm in our problem setting. For GP-UCB, we use $k = 20$ observations per configuration; for RAHBO, we use two variants, $k = 2$ and $k = 20$; and ERAHBO starts with $k_{\min} = 2$ and resamples each configuration until (12) is satisfied or $k_{\max} = 20$ is reached. For RAHBO and ERAHBO, we use a heteroscedastic GP for modeling $f(x)$ and a homoscedastic GP for $\rho^2(x)$. We follow common practices by setting $\beta_t = 2.5$. We preprocess the dataset using the outcome processing proposed in Song et al. (2024) and choose $\alpha = 1$ for the processed data to have a meaningful mean–variance trade-off. Generally, the MV objective is scale sensitive (multiplying the performance by a positive scalar will change the optimal configuration) which makes online outcome processing more difficult. For confidence-based resampling, we set $\beta_{\text{stop}} = 1$ for (12) to allow optimistic stopping. Before the BO procedure, we determine the GP hyperparameters by maximizing the marginal likelihood over 5 Sobol-sampled initial points that are the same for all experiments. We repeat each experiment 20 times with new initial points for every repetition. To tackle the high-dimensional search space, we apply the GP kernel hyperparameter prior that scales with the dimensionality of the search space, as proposed in Hvarfner et al. (2024).

RL Tasks. We use ARLBench (Becktepe et al., 2026) to produce performance datasets on which we run our experiments. In total, we use 5 environments for DQN, 10 for PPO and 4 for SAC. For each environment, we compute a dataset consisting of 512 configurations with 50 seeds each for more repeatable experimentation.[1] The full algorithm-environment combination is documented in Appendix C.1. We use the configuration spaces proposed in ARLBench and its default network architectures. As the performance objective, we use the mean evaluation return across 128 evaluation episodes at the end of training. Since environments differ in reward scales, we report the normalized regret to the average score of BO's initial points in each environment's dataset. We report additional empirical results in Appendix C.

5.1 Can ERAHBO find reliable configurations?

We optimize the hyperparameters of PPO, DQN and SAC to see if ERAHBO can outperform RAHBO through its adaptive sampling mechanism. Figure 2 shows that both RAHBO and ERAHBO outrank the GP-UCB baseline, but that the best value for RAHBO's k varies across environments. On Brax for example, $k = 20$ outperforms $k = 2$ while the opposite is true in XLand. The

[1]The code and the dataset can be found at https://github.com/LUH-AI/Efficient-Risk-Averse-BO.

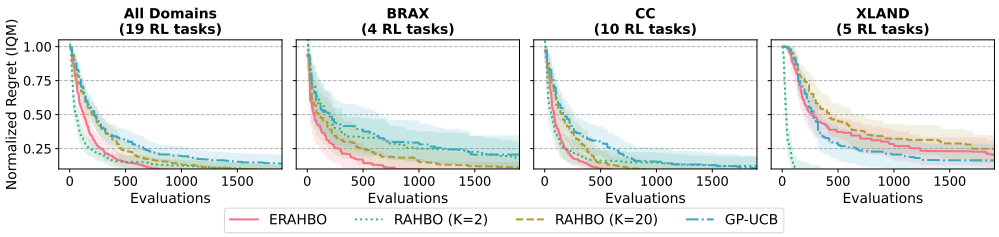

Figure 3: Domain-aggregated normalized regret over evaluation budgets. Lines and shaded areas show the inter-quantile mean (IQM) and 95% bootstrapped confidence interval (CI), respectively.

environments ERAHBO does not perform well on are predominantly from XLand, which contains environments for which we see only very few good performances overall. This makes it hard to find a good incumbent and leads to many repetitions preventing exploration.

Overall, ERAHBO outperforms both RAHBO variants in a total of 7 out of 19 algorithm-environment pairs, and at least one RAHBO variation 16 out of 19 times. In Table 1, the Wilcoxon signed-rank test shows a statistically significant improvement of ERAHBO over RAHBO with $k = 20$ and GP-UCB. This means the adaptive sampling can automatically strike a balance between the different versions of RAHBO and eliminates the need for manually selecting k. We document all rank metrics-related results in Appendix C.2, including a comparison of ERAHBO with RAHBO using different repetitions k.

Table 1: Mean $\pm$ standard error of algorithm rank across experiments (lower is better). Bold values are not statistically significantly worse than best.

Metric	ERAHBO	RAHBO (k=20)	RAHBO (k=2)	GP-UCB
Simple Regret MV	**1.79 $\pm$ 0.77**	2.42 $\pm$ 1.04	**2.37 $\pm$ 1.09**	3.42 $\pm$ 0.88
Cumulative Regret MV	**1.79 $\pm$ 0.77**	2.58 $\pm$ 0.82	**2.32 $\pm$ 1.34**	3.32 $\pm$ 0.86

5.2 Is ERAHBO making HPO more sample efficient?

ERAHBO not only balances additional evaluations for the best performance tradeoff, but it is also more efficient in terms of evaluations than RAHBO or GP-UCB. In Table 2, we report the evaluation number where the algorithms achieve a regret threshold measured as the percent of the initial design's regret, and we count the tasks where all algorithms reach this threshold for easier comparability. ERAHBO achieves sample efficiency comparable to RAHBO with $k = 2$, reaching the regret thresholds after a similar number of evaluations. However, as discussed above, RAHBO with $k = 2$ yields worse configurations after the full evaluation budget. Compared with RAHBO with $k = 20$ and GP-UCB, ERAHBO reaches almost all the regret thresholds significantly faster.

Table 2: Mean $\pm$ standard error number of evaluations needed to improve over regret thresholds. We average tasks where all methods reach the threshold (count per method in brackets).

Threshold	ERAHBO	RAHBO (k=20)	RAHBO (k=2)	GP-UCB
75% ($n_{\text{common}} = 15$)	40.8 $\pm$ 5.9 (18)	62.3 $\pm$ 9.1 (17)	80.6 $\pm$ 43.2 (17)	83.7 $\pm$ 25.1 (17)
50% ($n_{\text{common}} = 12$)	107.6 $\pm$ 14.9 (16)	132.7 $\pm$ 15.7 (16)	139.2 $\pm$ 91.6 (15)	197.7 $\pm$ 52.5 (14)
25% ($n_{\text{common}} = 11$)	193.6 $\pm$ 32.7 (13)	304.6 $\pm$ 41.5 (13)	147.9 $\pm$ 72.2 (12)	359.2 $\pm$ 98.5 (12)

Furthermore, Figure 3 shows normalized mean regret for all domains. Compared to GP-UCB, both ERAHBO and RAHBO can have better anytime performance. Therefore, they are better suited to low-budget HPO, as is often necessary in RL. As in the overall ranks, we observe k being an essential

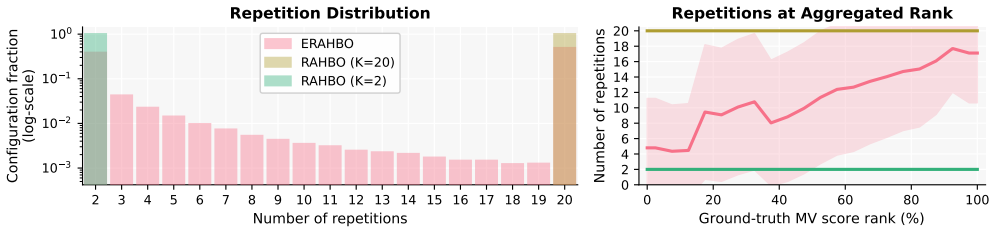

Figure 4: Repetition numbers in dataset experiments. **Left**: Compared with RAHBO using fixed sampling repetition, ERAHBO rejects the majority of configurations without spending the full budget. **Right**: ERAHBO invests more repetition in promising configurations with higher MV scores, demonstrating adaptivity through the confidence-based resampling behavior.

HP for RAHBO. With a lower k, RAHBO is very efficient for small budgets, but cannot effectively utilize larger optimization budgets. With $k = 20$, on the other hand, RAHBO becomes inefficient with small budgets. ERAHBO can adapt to different budgets without HPO for its own HPs, making it a more flexible method. We note that the pattern depends on the risk-sensitivity parameter α, since large values can prevent BO from exploring high-mean performance configurations. More detailed results are available in Appendix C.3 and C.4.

5.3 Does the adaptive sampling focus on well-performing configurations?

Beyond being more efficient than RAHBO, we want to verify that ERAHBO actually prioritizes promising configurations and spends very few evaluations on poorly performing ones. Figure 4 shows the distribution of repetitions across all evaluations. These results indicate that ERAHBO allocates more budget to better configurations. Additionally, we note that scheduling β in the incumbent lower bound in (10) further improves the performance of ERAHBO (see Appendix C.5).

6 Conclusion

We identify variability between repeated runs as a key difficulty factor for HPO in RL and propose to use risk-aware optimization to find an ideal variability-performance tradeoff. To do so efficiently for computationally expensive RL algorithms, we introduce ERAHBO, an adaptive sampling mechanism that automatically adds more samples only in regions of the search space that show high uncertainty. We provide sublinear regret guarantees and demonstrate empirically that ERAHBO improves upon its baselines in terms of overall performance and efficiency. Furthermore, the ERAHBO successfully identifies which configurations to prioritize and which to discard early.

We show that ERAHBO can improve the practical efficiency of risk-aware hyperparameter optimization by adaptively allocating samples to configurations where uncertainty matters most. However, our approach optimizes a mean-variance objective, which is only one notion of risk: the risk parameter α is scale-sensitive and does not directly target tail events or failure probabilities. Moreover, our regret guarantee is a conservative worst-case bound over all sample-count schedules $k_t \in [k_{\min}, k_{\max}]$ rather than a tight characterization of the specific adaptive rule. Finally, our empirical results rely on offline datasets of learning outcomes; while this enables broad and reproducible comparisons, it is not identical to fully online HPO with fresh training runs.

While we demonstrate the benefits of risk-aware HPO and ERAHBO's adaptive budget allocation for RL algorithms, our empirical evaluation shows that not all environments benefit equally, as we see in the XLand environments. The dataset we release with this paper will facilitate further research into the factors that influence risk in different RL settings. This will help the community and us to further develop adaptive risk-aware HPO methods for the wide range of RL task settings. We see ERAHBO's sampling strategy as a first step in this direction.

A Regret bound for ERAHBO

We follow the setup of Makarova et al. (2021). Throughout, let $k_t := k(x_t) \in [k_{\min}, k_{\max}]$ with $k_{\min} \geq 2$, so that the unbiased sample variance is well-defined. We assume normalized kernels, $\kappa(x,x) \leq 1$ and $\kappa^{\text{var}}(x,x) \leq 1$ for all $x \in \mathcal{X}$. We consider the BO observation model (Eq. (3)) with heteroscedastic noise, and denote by $\mu^f_{t-1}(x \mid \Sigma)$ and $\sigma^f_{t-1}(x \mid \Sigma)$ the GP posterior mean and standard deviation of the mean model with noise matrix Σ.

Lemma 1 (Lemma 7 in Kirschner & Krause (2018))**.** *Let $f \in \mathcal{H}_\kappa$, and let $\mu_t(\cdot)$ and $\sigma^2_t(\cdot)$ be the GP posterior mean and variance $\lambda > 0$. Assume that the observations $(x_t, y_t)_{t\geq 1}$ satisfy Eq. (3). Then for all $t \geq 1$ and $x \in \mathcal{X}$, with probability at least $1-\delta$,*

$$|\mu_{t-1}(x) - f(x)| \leq \underbrace{\left(\sqrt{2 \ln \left(\frac{\det(\lambda \Sigma_t + K_t)^{1/2}}{\delta \, \det(\lambda \Sigma_t)^{1/2}} \right)} + \sqrt{\lambda} \|f\|_\kappa \right)}_{:=\beta_t} \sigma_{t-1}(x). \tag{14}$$

In each BO round t, ERAHBO collects k_t independent samples at x_t, denoted $\{y_t^{(i)}\}_{i=1}^{k_t}$, and forms $\hat{m}(x_t) := k_t^{-1} \sum_{i=1}^{k_t} y_t^{(i)}$. Using Lemma 1 for the mean GP posterior, define

$$\text{UCB}^f_t(x) := \mu^f_{t-1}(x \mid \widehat{\Sigma}_{t-1}) + \beta^f_t \, \sigma^f_{t-1}(x \mid \widehat{\Sigma}_{t-1}), \tag{15}$$

$$\text{LCB}^f_t(x) := \mu^f_{t-1}(x \mid \widehat{\Sigma}_{t-1}) - \beta^f_t \, \sigma^f_{t-1}(x \mid \widehat{\Sigma}_{t-1}). \tag{16}$$

We make the same smoothness assumption on the variance proxy as in Makarova et al. (2021).

Assumption 1 (Assumption 1 in Makarova et al. (2021))**.** *The variance-proxy $\rho^2(x)$ belongs to an RKHS induced by some kernel κ^{var}, i.e., $\rho^2 \in \mathcal{H}_{\kappa^{var}}$, and its RKHS norm is bounded $\|\rho^2\|_{\kappa^{var}} \leq B_{var}$ for some finite $B_{var} > 0$. Moreover, the noise $\xi(x)$ in Eq. (3) is strictly $\rho(x)$-sub-Gaussian, i.e., $Var[\xi(x)] = \rho^2(x)$ for every $x \in \mathcal{X}$.*

Assumption 2 (Analogue of Assumption 2 in Makarova et al. (2021))**.** *Let $\mathcal{F}_{t-1}$ denote the BO history before round t. The noise $\eta(x_t, k_t)$ in $\hat{s}^2(x_t) = \rho^2(x_t) + \eta(x_t, k_t)$ is conditionally $\rho_\eta(x_t, k_t)$-sub-Gaussian given $\mathcal{F}_{t-1}$, x_t, and k_t, with known $\rho^2_\eta(x_t, k_t)$. The realizations $\{\eta(x_t, k_t)\}_{t\geq 1}$ are conditionally independent across t given the corresponding histories and selected pairs (x_t, k_t).*

In the variable-sample setting, the variance-observation proxy may depend on both the query point and the replication count, i.e., $\rho^2_\eta(x_t, k_t)$. The regret analysis only requires a uniform bound on these proxies. Thus, it suffices to assume that there exists $\mathcal{R}^2$ such that

$$\mathcal{R}^2 \geq \sup_{x \in \mathcal{X},\ k \in [k_{\min}, k_{\max}]} \rho^2_\eta(x,k), \qquad \mathcal{R}^2 = \frac{2\bar{\rho}^4}{k_{\min} - 1} \text{ is valid under strict sub-Gaussianity.}$$

Let μ^{var}_{t-1} and $\sigma^{\text{var}}_{t-1}$ denote the variance-GP posterior mean and standard deviation built with $\Sigma^{\text{var}}_{t-1} := \text{diag}(\rho^2_\eta(x_1, k_1), \ldots, \rho^2_\eta(x_{t-1}, k_{t-1}))$. Using Lemma 1 for the variance GP posterior, define

$$\text{UCB}^{\text{var}}_t(x) := \mu^{\text{var}}_{t-1}(x) + \beta^{\text{var}}_t \, \sigma^{\text{var}}_{t-1}(x), \tag{17}$$

$$\text{LCB}^{\text{var}}_t(x) := \mu^{\text{var}}_{t-1}(x) - \beta^{\text{var}}_t \, \sigma^{\text{var}}_{t-1}(x). \tag{18}$$

Following Makarova et al. (2021), we construct

$$\widehat{\Sigma}_{t-1} := \text{diag}_{s<t} \left(\frac{\min\{\text{UCB}^{\text{var}}_{t-1}(x_s), \bar{\rho}^2\}}{k_s} \right). \tag{19}$$

On the simultaneous variance-confidence event, $\rho^2(x_s) \leq \text{UCB}^{\text{var}}_{t-1}(x_s)$ for all $s \leq t-1$, and by assumption $\rho^2(x_s) \leq \bar{\rho}^2$. Hence the construction above is used as a conservative noise matrix for

the sample-mean observations. Equivalently, throughout the proof we condition on the event that

$$\frac{\rho^2(x_s)}{k_s} \le \frac{\min\{\mathrm{UCB}^{\mathrm{var}}_{t-1}(x_s), \overline{\rho}^2\}}{k_s}, \qquad s < t,$$

so Lemma 1 applies to the mean GP with noise matrix $\widehat{\Sigma}_{t-1}$.

For convenience, define $N_T := \sum_{t=1}^{T} k_t$ and $\bar{\beta}^{\ell}_T := \max_{1\le s\le T} \beta^{\ell}_s$, $\ell \in \{f, \mathrm{var}\}$.

We use the sequence-based information gain terms

$$\overline{\gamma}_T := \max_{\substack{x_1,\dots,x_T\in\mathcal{X} \\ k_1,\dots,k_T\in[k_{\min},k_{\max}]}} \sum_{t=1}^{T} \frac{1}{2} \ln\left(1 + \frac{\left(\sigma^f_{t-1}\right)^2(x_t \mid D_{t-1})}{\overline{\rho}^2/k_t}\right), \tag{20}$$

$$\Gamma_T := \max_{\substack{x_1,\dots,x_T\in\mathcal{X} \\ k_1,\dots,k_T\in[k_{\min},k_{\max}]}} \sum_{t=1}^{T} \frac{1}{2} \ln\left(1 + \frac{\left(\sigma^{\mathrm{var}}_{t-1}\right)^2(x_t)}{\rho^2_{\eta,t}}\right). \tag{21}$$

Here $D_{t-1} := \mathrm{diag}(\overline{\rho}^2/k_1, \dots, \overline{\rho}^2/k_{t-1})$.

We now adapt the regret analysis of Makarova et al. (2021) to variable sample counts.

Theorem 1. *Consider any $f \in \mathcal{H}_\kappa$ with $\|f\|_\kappa \le B_f$, and the BO observation model in Eq. (3) with an unknown variance proxy $\rho^2(x)$ that satisfies Assumptions 1 and 2. Let $\{x_t\}_{t=1}^T$ denote the sequence of query points selected by ERAHBO over T BO rounds, where in round t the algorithm collects $k_t \in [k_{\min}, k_{\max}]$ samples at x_t. We assume that the aggregated observations used to update the mean and variance GPs satisfy the observation models stated above; in particular, this holds when k_t is fixed before observing the samples in round t. Let $\{\beta^f_t\}_{t=1}^T$ and $\{\beta^{\mathrm{var}}_t\}_{t=1}^T$ be defined according to Lemma 1 (with $\lambda = 1$). Set $\rho(\cdot) \in [\underline{\rho}, \overline{\rho}]$ and let $\mathcal{R}^2$ be any uniform bound on the variance-GP observation proxies. Then, with probability at least $1 - 2\delta$, for all $T \ge 1$,*

$$R_T \;\le\; 2\,\bar{\beta}^f_T\, k_{\max} \sqrt{\frac{2T\,\overline{\gamma}_T}{\ln\!\left(1 + k_{\max}/\overline{\rho}^2\right)}} \;+\; 2\,\alpha\,\bar{\beta}^{\mathrm{var}}_T\, k_{\max} \sqrt{\frac{2T\,\Gamma_T}{\ln\!\left(1 + \mathcal{R}^{-2}\right)}}, \tag{22}$$

where $\bar{\beta}^f_T := \max_{1\le s\le T} \beta^f_s$, $\bar{\beta}^{\mathrm{var}}_T := \max_{1\le s\le T} \beta^{\mathrm{var}}_s$, and $\overline{\gamma}_T$ and Γ_T are defined in (20)–(21).

Proof. The proof follows the same steps as Appendix A.4 in Makarova et al. (2021), with the modifications needed for variable sample counts.

Step 1 (Confidence bounds for MV). By the same argument as in Makarova et al. (2021), with probability at least $1 - 2\delta$,

$$\mathrm{LCB}^{\mathrm{MV}}_t(x) \le \mathrm{MV}(x) \le \mathrm{UCB}^{\mathrm{MV}}_t(x), \qquad \forall x \in \mathcal{X},\ t \ge 1,$$

where

$$\mathrm{UCB}^{\mathrm{MV}}_t(x) = \mathrm{UCB}^f_t(x) - \alpha\,\mathrm{LCB}^{\mathrm{var}}_t(x), \tag{23}$$

$$\mathrm{LCB}^{\mathrm{MV}}_t(x) = \mathrm{LCB}^f_t(x) - \alpha\,\mathrm{UCB}^{\mathrm{var}}_t(x). \tag{24}$$

Step 2 (Instantaneous and cumulative regret). Let $r_t := \mathrm{MV}(x^*) - \mathrm{MV}(x_t)$. As in Makarova et al. (2021),

$$\begin{aligned} r_t &\le \mathrm{UCB}^{\mathrm{MV}}_t(x_t) - \mathrm{LCB}^{\mathrm{MV}}_t(x_t) \\ &= 2\beta^f_t\,\sigma^f_{t-1}(x_t \mid \widehat{\Sigma}_{t-1}) + 2\alpha\beta^{\mathrm{var}}_t\,\sigma^{\mathrm{var}}_{t-1}(x_t). \end{aligned} \tag{25}$$

Hence

$$R_T = \sum_{t=1}^{T} k_t r_t$$
$$\leq 2\bar{\beta}_T^f \sum_{t=1}^{T} k_t \, \sigma_{t-1}^f(x_t \mid \widehat{\Sigma}_{t-1}) + 2\alpha\bar{\beta}_T^{\mathrm{var}} \sum_{t=1}^{T} k_t \, \sigma_{t-1}^{\mathrm{var}}(x_t). \tag{26}$$

Step 3 (Bounding the mean-model term). Let $v_t := \overline{\rho}^2/k_t$ and $u_t := \big(\sigma_{t-1}^f\big)^2(x_t \mid D_{t-1})/v_t$. Since

$$\frac{\min\{\mathrm{UCB}_{t-1}^{\mathrm{var}}(x_s), \overline{\rho}^2\}}{k_s} \leq \frac{\overline{\rho}^2}{k_s} = v_s \qquad \forall s < t,$$

we have $\widehat{\Sigma}_{t-1} \preceq D_{t-1}$, and therefore $\sigma_{t-1}^f(x_t \mid \widehat{\Sigma}_{t-1}) \leq \sigma_{t-1}^f(x_t \mid D_{t-1})$. By Cauchy–Schwarz,

$$\sum_{t=1}^{T} k_t \, \sigma_{t-1}^f(x_t \mid \widehat{\Sigma}_{t-1}) \leq \sum_{t=1}^{T} k_t \, \sigma_{t-1}^f(x_t \mid D_{t-1})$$
$$= \sum_{t=1}^{T} k_t \sqrt{v_t}\sqrt{u_t} = \overline{\rho} \sum_{t=1}^{T} \sqrt{k_t}\sqrt{u_t}$$
$$\leq \overline{\rho}\sqrt{N_T}\sqrt{\sum_{t=1}^{T} u_t}. \tag{27}$$

Moreover, since $\lambda = 1$ and $\kappa(x, x) \leq 1$, $0 \leq u_t \leq k_t/\overline{\rho}^2 \leq k_{\max}/\overline{\rho}^2$. Thus,

$$u_t \leq \frac{k_{\max}/\overline{\rho}^2}{\ln\big(1 + k_{\max}/\overline{\rho}^2\big)} \ln(1 + u_t),$$

and summing over t gives

$$\sum_{t=1}^{T} u_t \leq \frac{2k_{\max}/\overline{\rho}^2}{\ln\big(1 + k_{\max}/\overline{\rho}^2\big)} \sum_{t=1}^{T} \frac{1}{2} \ln(1 + u_t)$$
$$\leq \frac{2k_{\max}\,\overline{\gamma}_T/\overline{\rho}^2}{\ln\big(1 + k_{\max}/\overline{\rho}^2\big)}. \tag{28}$$

Substituting into (27), we obtain

$$\sum_{t=1}^{T} k_t \, \sigma_{t-1}^f(x_t \mid \widehat{\Sigma}_{t-1}) \leq \sqrt{\frac{2k_{\max} N_T \, \overline{\gamma}_T}{\ln\big(1 + k_{\max}/\overline{\rho}^2\big)}}. \tag{29}$$

Step 4 (Bounding the variance-model term). Define $w_t := \big(\sigma_{t-1}^{\mathrm{var}}\big)^2(x_t)/\rho_{\eta,t}^2$. By Cauchy–Schwarz,

$$\sum_{t=1}^{T} k_t \, \sigma_{t-1}^{\mathrm{var}}(x_t) \leq \sqrt{\sum_{t=1}^{T} k_t} \sqrt{\sum_{t=1}^{T} k_t \big(\sigma_{t-1}^{\mathrm{var}}\big)^2(x_t)}$$
$$\leq \sqrt{k_{\max} N_T} \sqrt{\sum_{t=1}^{T} \big(\sigma_{t-1}^{\mathrm{var}}\big)^2(x_t)}. \tag{30}$$

Since $\lambda = 1$ and $\kappa^{\mathrm{var}}(x, x) \leq 1$, $0 \leq w_t \leq \rho_{\eta,t}^{-2}$. Hence

$$\rho_{\eta,t}^2 w_t \leq \frac{1}{\ln\big(1 + \rho_{\eta,t}^{-2}\big)} \ln(1 + w_t) \leq \frac{1}{\ln\big(1 + \mathcal{R}^{-2}\big)} \ln(1 + w_t),$$

where the second inequality uses $\rho_{\eta,t}^2 \le \mathcal{R}^2$. Summing over t yields

$$\begin{aligned}\sum_{t=1}^{T}\big(\sigma_{t-1}^{\mathrm{var}}\big)^2(x_t) &= \sum_{t=1}^{T}\rho_{\eta,t}^2 w_t \\ &\le \frac{1}{\ln\big(1+\mathcal{R}^{-2}\big)}\sum_{t=1}^{T}\ln(1+w_t) \\ &= \frac{2\Gamma_T}{\ln\big(1+\mathcal{R}^{-2}\big)}. \end{aligned} \tag{31}$$

Substituting into (30), we obtain

$$\sum_{t=1}^{T} k_t\,\sigma_{t-1}^{\mathrm{var}}(x_t) \le \sqrt{\frac{2k_{\max}N_T\,\Gamma_T}{\ln\big(1+\mathcal{R}^{-2}\big)}}. \tag{32}$$

Step 5 (Combine bounds). Substituting (29) and (32) into (26) yields (22) with $N_T \le Tk_{\max}$. □

Remark on adaptive resampling. The bound is stated for bounded sample counts $k_t \in [k_{\min}, k_{\max}]$ under the aggregate observation model used by RAHBO. ERAHBO's practical stopping rule chooses k_t by repeatedly checking the MV-UCB at the current point. Since the confidence bounds are uniform over points and rounds, and since the inner loop contains only finitely many checks $k \in [k_{\min}, k_{\max}]$, this provides a valid high-confidence stopping certificate whenever the intermediate GP updates satisfy the stated concentration assumptions, after applying a union or anytime bound over the inner-loop checks if needed. A fully optional-stopping analysis of the stopped sample mean and sample variance is beyond the scope of this work; the theorem should therefore be read as showing that bounded variable replication itself does not degrade the RAHBO regret rate, while our stopping rule is the practical mechanism used to choose such bounded replications.

Acknowledgments

Mingxuan Che acknowledges funding by the European Union (ERC, "ixAutoML", grant no.101041029). Theresa Eimer and Marius Lindauer acknowledge funding by the German Research Foundation (DFG) under LI 2801/7-1 and LI 2801/10-1. Alexander von Rohr is funded by the Deutsche Forschungsgemeinschaft (DFG) under project 468806714 of the Emmy Noether Programme. Alexander von Rohr also gratefully acknowledges funding from the European Union (ERC, ConSequentIAL, 101165883). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council. Neither the European Union nor the granting authority can be held responsible for them.

References

R. Agarwal, M. Schwarzer, P. Castro, A. Courville, and M. Bellemare. Deep reinforcement learning at the edge of the statistical precipice. In *Proceedings of the 35th International Conference on Advances in Neural Information Processing Systems (NeurIPS'21)*, 2021.

T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama. Optuna: A next-generation Hyperparameter Optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD'19)*, pp. 2623–2631, 2019.

J. Becktepe, J. Dierkes, C. Benjamins, A. Mohan, D. Salinas, R. Rajan, F. Hutter, H. Hoos, M. Lindauer, and T. Eimer. ARLBench: Flexible and efficient benchmarking for hyperparameter optimization in reinforcement learning. *Journal of Data-Centric Machine Learning*, 2026.

J. Bergstra and Y. Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13:281–305, 2012.

J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl. Algorithms for hyper-parameter optimization. In *Proceedings of the 25th International Conference on Advances in Neural Information Processing Systems (NeurIPS'11)*, pp. 2546–2554, 2011.

F. Berkenkamp, A. Schoellig, and A. Krause. Safe controller optimization for quadrotors with gaussian processes. In *2016 IEEE international conference on robotics and automation (ICRA)*, pp. 491–496, 2016.

M. Binois, R. Gramacy, and M. Ludkovski. Practical heteroscedastic gaussian process modeling for large simulation experiments. *Journal of Computational and Graphical Statistics*, 27(4):808–821, 2018.

M. Binois, J. Huang, R. Gramacy, and M. Ludkovski. Replication or exploration? sequential design for stochastic simulation experiments. *Technometrics*, 61(1):7–23, 2019.

B. Bischl, M. Binder, M. Lang, T. Pielok, J. Richter, S. Coors, J. Thomas, T. Ullmann, M. Becker, A.-L. Boulesteix, D. Deng, and M. Lindauer. Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, pp. e1484, 2023.

Z. Dai, Q. Nguyen, S. Tay, D. Urano, R. Leong, B. Low, and P. Jaillet. Batch bayesian optimization for replicable experimental design. In *Proceedings of the 37th International Conference on Advances in Neural Information Processing Systems (NeurIPS'23)*, 2023.

J. Dierkes, T. Eimer, M. Lindauer, and H. Hoos. Performance prediction in reinforcement learning: The bad and the ugly. In *18th European Workshop on Reinforcement Learning (EWRL)*, 2025.

T. Eimer, M. Lindauer, and R. Raileanu. Hyperparameters in reinforcement learning and how to tune them. In *Proceedings of the 40th International Conference on Machine Learning (ICML'23)*, 2023.

C. Freeman, E. Frey, A. Raichuk, S. Girgin, I. Mordatch, and O. Bachem. Brax - A differentiable physics engine for large scale rigid body simulation. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, 2021.

R. Garnett. *Bayesian Optimization*. Cambridge University Press, 2023.

T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proceedings of the 35th International Conference on Machine Learning (ICML'18)*, 2018.

P. Henderson, R. Islam, P. Bachman, J. Pineau, D. Precup, and D. Meger. Deep reinforcement learning that matters. In *Proceedings of the Thirty-Second Conference on Artificial Intelligence (AAAI'18)*, 2018.

C. Hvarfner, E. Hellsten, and L. Nardi. Vanilla Bayesian optimization performs great in high dimensions. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 20793–20817, 2024.

R. Islam, P. Henderson, M. Gomrokchi, and D. Precup. Reproducibility of benchmarked deep reinforcement learning tasks for continuous control. *arXiv:1708.04133*, 2017.

J. Kirschner and A. Krause. Information directed sampling and bandits with heteroscedastic noise. In *Conference On Learning Theory*, pp. 358–384, 2018.

R. Lange. gymnax: A JAX-based reinforcement learning environment library, 2022.

M. Lindauer, K. Eggensperger, M. Feurer, A. Biedenkapp, D. Deng, C. Benjamins, T. Ruhkopf, R. Sass, and F. Hutter. SMAC3: A versatile bayesian optimization package for Hyperparameter Optimization. *Journal of Machine Learning Research*, 23(54):1–9, 2022.

A. Makarova, I. Usmanova, I. Bogunovic, and A. Krause. Risk-averse heteroscedastic Bayesian optimization. In *Proceedings of the 35th International Conference on Advances in Neural Information Processing Systems (NeurIPS'21)*, 2021.

V. Mnih, K. Kavukcuoglu, D. Silver, A. Rusu, J. Veness, M. Bellemare, A. Graves, M. Riedmiller, A. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 02 2015.

A. Nikulin, V. Kurenkov, I. Zisman, V. Sinii, A. Agarkov, and S. Kolesnikov. XLand-minigrid: Scalable meta-reinforcement learning environments in JAX. In *Intrinsically-Motivated and Open-Ended Learning Workshop, NeurIPS*, 2023.

J. Obando-Ceron, J. Araújo, A. Courville, and P. Castro. On the consistency of hyper-parameter selection in value-based deep reinforcement learning. *RLJ*, 3:1037–1059, 2024.

J. Parker-Holder, V. Nguyen, and S. J. Roberts. Provably efficient online Hyperparameter Optimization with population-based bandits. In *Proceedings of the 34th International Conference on Advances in Neural Information Processing Systems (NeurIPS'20)*, 2020.

J. Parker-Holder, R. Rajan, X. Song, A. Biedenkapp, Y. Miao, T. Eimer, B. Zhang, V. Nguyen, R. Calandra, A. Faust, F. Hutter, and M. Lindauer. Automated reinforcement learning (AutoRL): A survey and open problems. *Journal of Artificial Intelligence Research (JAIR)*, 74:517–568, 2022.

A. Sani, A. Lazaric, and R. Munos. Risk-aversion in multi-armed bandits. In *Proceedings of the 26th International Conference on Advances in Neural Information Processing Systems (NeurIPS'12)*, 2012.

J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *arXiv:1707.06347 [cs.LG]*, 2017.

G. Shala, S. P. Arango, A. Biedenkapp, F. Hutter, and J. Grabocka. Hpo-rl-bench: A zero-cost benchmark for hpo in reinforcement learning. In *Proceedings of the Third International Conference on Automated Machine Learning*, 2024.

J. Snoek, H. Larochelle, and R. Adams. Practical Bayesian optimization of machine learning algorithms. In *Proceedings of the 26th International Conference on Advances in Neural Information Processing Systems (NeurIPS'12)*, pp. 2960–2968, 2012.

X. Song, Q. Zhang, C. Lee, E. Fertig, T. Huang, L. Belenki, G. Kochanski, S. Ariafar, S. Vasudevan, and S. Perel. The vizier gaussian process bandit algorithm. *arXiv:2408.11527*, 2024.

J. Springenberg, A. Klein, S. Falkner, and F. Hutter. Bayesian optimization with robust Bayesian neural networks. In *Proceedings of the 30th International Conference on Advances in Neural Information Processing Systems (NeurIPS'16)*, 2016.

N. Srinivas, A. Krause, S. Kakade, and M. Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In *Proceedings of the 27th International Conference on Machine Learning (ICML'10)*, pp. 1015–1022, 2010.

X. Wan, C. Lu, J. Parker-Holder, P. Ball, V. Nguyen, B. Ru, and M. Osborne. Bayesian generational population-based training. In *Proceedings of the First International Conference on Automated Machine Learning*, 2022.

B. Zhang, R. Rajan, L. Pineda, N. Lambert, A. Biedenkapp, K. Chua, F. Hutter, and R. Calandra. On the importance of Hyperparameter Optimization for model-based reinforcement learning. In *Proceedings of the 24th International Conference on Artificial Intelligence and Statistics (AISTATS'21)*, pp. 4015–4023, 2021.

D. Zhuang, X. Zhang, S. Song, and S. Hooker. Randomness in neural network training: Characterizing the impact of tooling. *Proceedings of Machine Learning and Systems*, 4:316–336, 2022.

Supplementary Materials

The following content was not necessarily subject to peer review.

B Finite-sample certificates

This section provides a local certificate for ERAHBO's stopping rule in the case where the currently queried configuration can be shown to be suboptimal relative to the incumbent lower bound. Concretely, we derive a sufficient condition under which, after collecting enough samples at the current configuration x_t, its optimistic mean-variance estimate drops below the incumbent lower bound with the current confidence bounds, so that additional sampling at x_t is unnecessary.

Fix a BO round t and define

$$B_t := \max_{x \in \mathcal{D}_{t-1}} \mathrm{LCB}_t^{\mathrm{MV}}(x),$$

computed once at the start of round t and kept fixed during the inner sampling loop at x_t. For $k \geq 2$, let $\mu_{t,k}^{\mathrm{MV}}(x_t)$, $\sigma_{t,k}^{f}(x_t)$, and $\sigma_{t,k}^{\mathrm{var}}(x_t)$ denote the corresponding posterior quantities after incorporating the first k samples collected at x_t, and define

$$m_t(k) := B_t - \mu_{t,k}^{\mathrm{MV}}(x_t).$$

Also define

$$\mathrm{UCB}_{t,k}^{\mathrm{MV}}(x_t) := \mu_{t,k}^{\mathrm{MV}}(x_t) + \beta_t^f \sigma_{t,k}^{f}(x_t) + \alpha \beta_t^{\mathrm{var}} \sigma_{t,k}^{\mathrm{var}}(x_t).$$

Lemma 2 (Finite-sample certificate of suboptimality)**.** *Assume* $m_t(k) > 0$ *and*

$$\sigma_{t,k}^{f}(x_t) \leq \frac{\rho(x_t)}{\sqrt{k}}, \qquad \sigma_{t,k}^{\mathrm{var}}(x_t) \leq \frac{c_{\mathrm{var}}}{\sqrt{k}},$$

for some constants $\rho(x_t) > 0$ *and* $c_{\mathrm{var}} > 0$*. If*

$$k \geq \frac{\left(\beta_t^f \rho(x_t) + \alpha \beta_t^{\mathrm{var}} c_{\mathrm{var}}\right)^2}{m_t(k)^2}, \tag{33}$$

then

$$\mathrm{UCB}_{t,k}^{\mathrm{MV}}(x_t) \leq B_t.$$

Proof. By definition of $\mathrm{UCB}_{t,k}^{\mathrm{MV}}(x_t)$, it suffices that

$$\beta_t^f \sigma_{t,k}^{f}(x_t) + \alpha \beta_t^{\mathrm{var}} \sigma_{t,k}^{\mathrm{var}}(x_t) \leq m_t(k).$$

Using the assumed bounds on $\sigma_{t,k}^{f}(x_t)$ and $\sigma_{t,k}^{\mathrm{var}}(x_t)$, this is implied by

$$\frac{\beta_t^f \rho(x_t) + \alpha \beta_t^{\mathrm{var}} c_{\mathrm{var}}}{\sqrt{k}} \leq m_t(k),$$

which is equivalent to (33). □

Takeaway. If the current posterior mean $\mu_{t,k}^{\mathrm{MV}}(x_t)$ lies below the incumbent lower bound B_t by a margin $m_t(k) > 0$, then the stopping rule is triggered once the confidence radius becomes smaller than this margin. In particular, the sufficient sample size in (33) scales as $1/m_t(k)^2$: candidates that are closer to the incumbent require substantially more samples to certify suboptimality. The threshold also increases with the uncertainty multipliers β_t^f and β_t^{var}, with the local noise level $\rho^2(x_t)$, and with the variance-model constant c_{var}, formalizing that ERAHBO allocates more samples only when uncertainty could plausibly change the mean-variance decision.

C Additional empirical results

This section contains additional empirical results for ERAHBO and the full experiment details for the results presented in Section 5.

C.1 Algorithm-environment combination

Table 3 shows the RL algorithms and environments used in the experiment, and each pair of them corresponds to a dataset for risk-aware BO tasks. By 'task', we refer to a dataset for the algorithm-environment pair. The dataset is generated using ARLbench (Becktepe et al., 2026), including all available RL tasks at the time of publication. Each dataset contains 512 configurations generated by a Sobol sequence, and each configuration includes policy returns across 50 random seeds.

Some datasets contain configurations that return `nan`, a common issue in RL benchmarks. In our experiments, we filter out these `nan` configurations, since failure handling is beyond the scope of this work and is left for future work. In detail, there are 18 infeasible configurations from `ppo_cc_pendulum` dataset, 48 from `ppo_cc_continuous_mountain_car`, and 182 from `ppo_brax_halfcheetah`. In addition, we exclude the dataset `dqn_xland_doorkey` because for this task, all configurations sampled by the Sobol sequence yield zero performance. Both datasets, before and after feasibility filtering, are available online at https://github.com/LUH-AI/Efficient-Risk-Averse-BO.

Table 3: RL algorithms and environments from ARLBench used in the experiment by domain

Algorithm	Classic Control (CC)	Brax	XLand
DQN	`acrobot`, `cartpole`, `mountain_car`	—	`empty_random`,`four_rooms`
PPO	`acrobot`, `cartpole`, `continuous_mountain_car`, `mountain_car`, `pendulum`	`fast`, `halfcheetah`	`door_key`, `empty_random`, `four_rooms`
SAC	`continuous_mountain_car`, `pendulum`	`fast`, `halfcheetah`	—

C.2 Ranking statistics

Tables 1 and 4 show the aggregated mean and median rank for final simple and cumulative regret of the mean-variance objective. We see similar trends in performance over time: RAHBO and ERAHBO outperform GP-UCB, with RAHBO with $k = 2$ ranking similarly to ERAHBO but with a larger inter-quantile range.

Table 4: Median algorithm rank [25th percentile, 75th percentile] aggregated across experiments; lower is better.

Metric	ERAHBO	RAHBO ($k = 20$)	RAHBO ($k = 2$)	GP-UCB
Simple Regret MV	2.00 [1.00, 2.00]	3.00 [1.50, 3.00]	2.00 [1.50, 3.00]	4.00 [3.00, 4.00]
Cumulative Regret MV	2.00 [1.00, 2.00]	3.00 [2.00, 3.00]	2.00 [1.00, 4.00]	4.00 [3.00, 4.00]

Figures 5 shows the mean ranks over time per environment, aggregated across RL algorithms. As in the main paper, we see clear differences across domains, especially in XLand, but ERAHBO ranks best overall.

Figure 6 shows the rank distribution. In this experiment, the rank distributions across algorithms are similar; however, the fraction colormap reveals that the RAHBO $k = 2$ rank distribution is bimodal, with a concentration at the best and worst ranks.

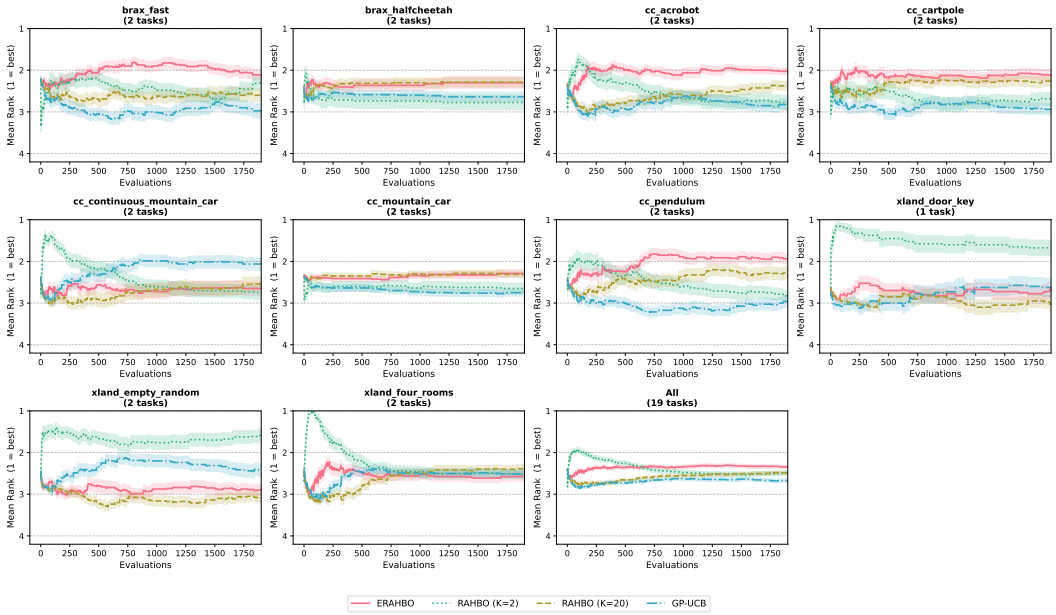

Figure 5: Mean algorithm rank aggregated by RL environments. The line and shaded area show the mean and $1\times$ standard error, respectively.

In addition to the rank result with RAHBO repetitions $k = 2$ and $k = 20$ shown in Figure 2, we run RAHBO with $k \in \{5, 8, 11, 14, 17\}$ and present the domain-aggregated rank over time in Figures 7. After around 750 evaluations, ERAHBO outperforms all the RAHBO repetition variants.

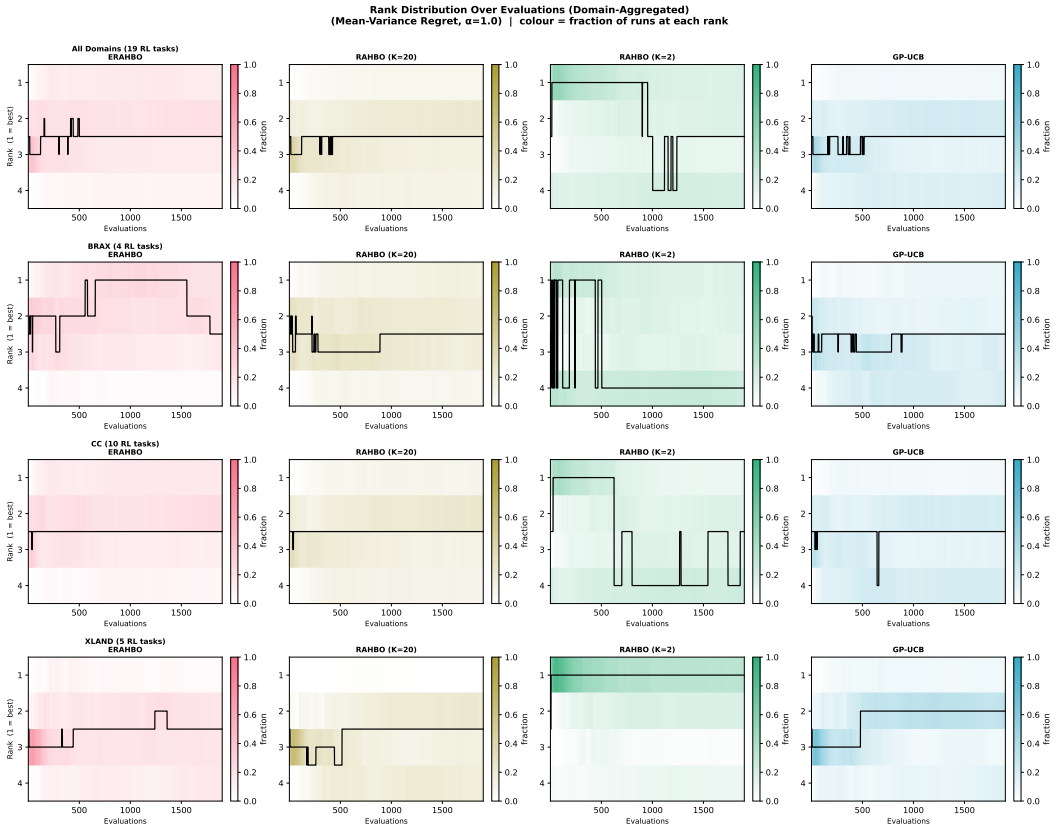

Figure 6: Rank distribution over time. Rows and columns correspond to the aggregation domain and method, respectively, as compared in the experiment (e.g., Figure 2). The solid line and the colormap show the rank mode and the fraction of rank appearances.

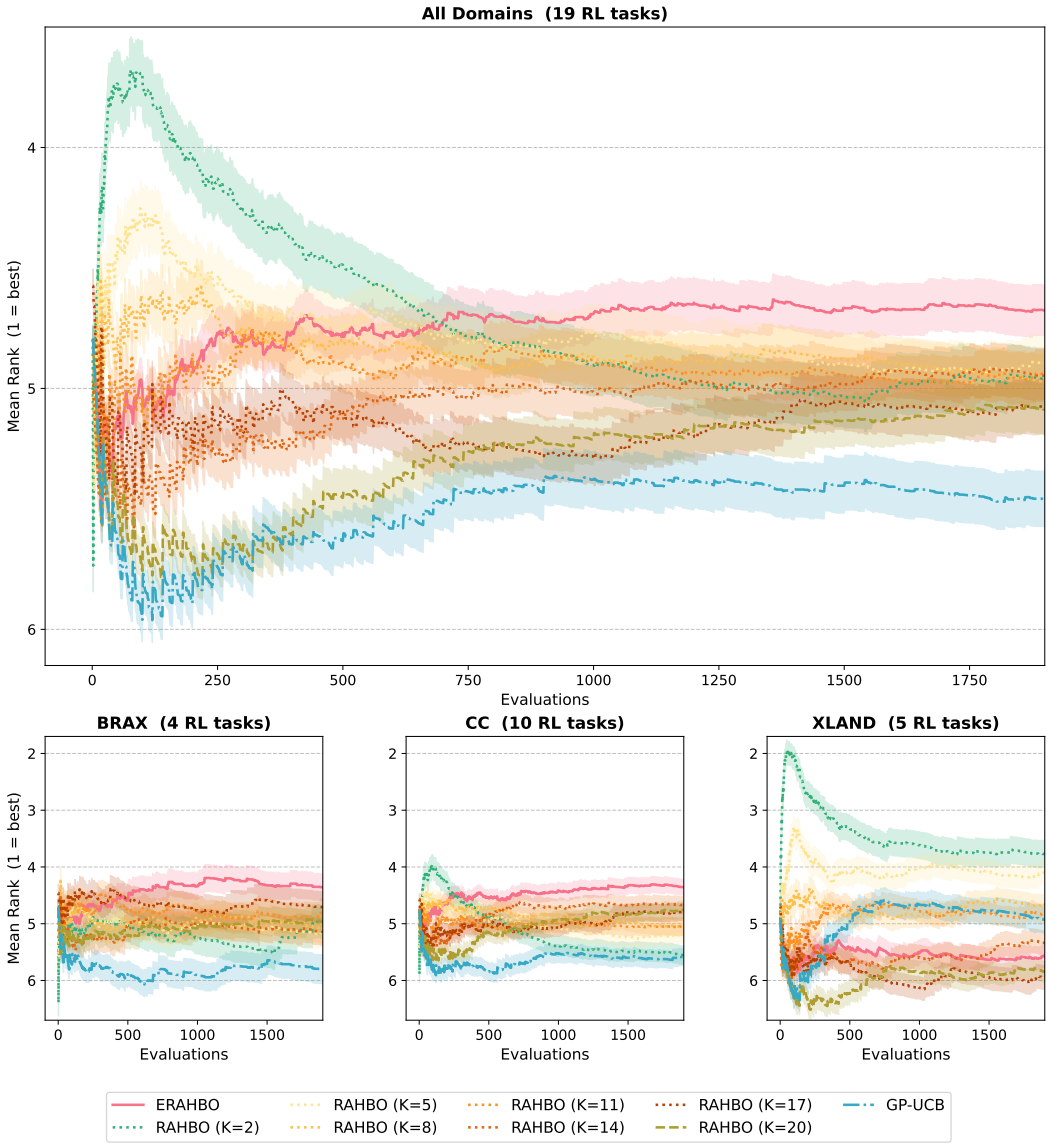

Figure 7: Mean algorithm rank aggregated by domain with RAHBO $k \in \{2, 5, 8, 11, 14, 17, 20\}$. The line and shaded area show the mean and $1\times$ standard error, respectively.

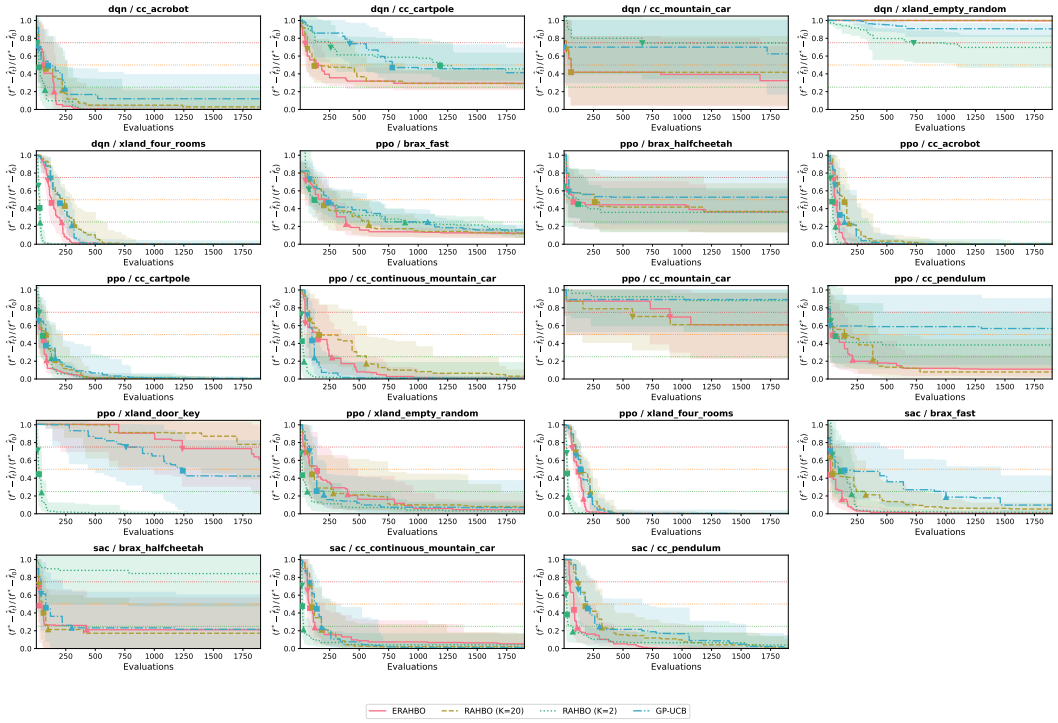

Figure 8: Algorithm normalized regret for individual RL task. The line and shaded area show the IQM and 95% bootstrapped CI, respectively. The colored marker indicates the point at which the algorithm's normalized regret exceeds the threshold (% of initial-design regret).

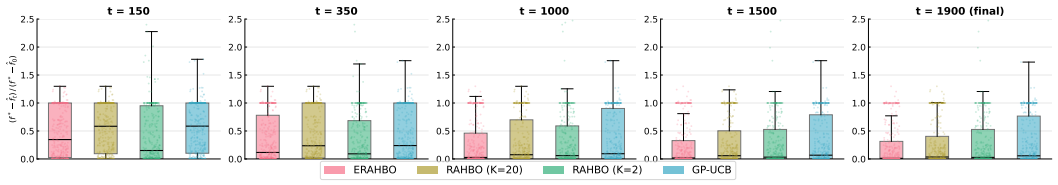

Figure 9: Normalized mean-variance regret at different evaluation budget t. Mean regret is normalized by the initial design's mean regret.

C.3 Aggregated regret over time

Figure 9-12 show the normalized mean-variance regret at different evaluation budgets t for domains CC, Brax, and XLand. Figures 8 shows the mean regret over time per environment and algorithm. Overall, the tasks are quite different: for some, we see large improvements over the initial design (e.g., on acrobot), whereas in others, not much can be gained (e.g., on mountain car). While the curves look slightly different for different algorithms, the overall trend seems to be environment-dependent.

C.4 Sample efficiency

Tables 5 shows how quickly ERAHBO, RAHBO and GP-UCB improve over the initial design. We clearly see that ERAHBO can be more efficient than RAHBO in most cases due to discarding poor configurations early on.

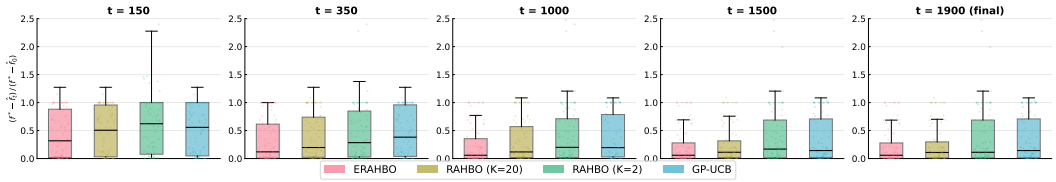

Figure 10: Normalized mean-variance regret at different evaluation budget t in domain Brax. Mean regret is normalized by the initial design's mean regret.

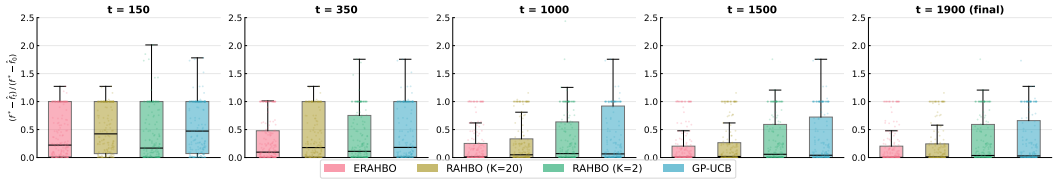

Figure 11: Normalized mean-variance regret at different evaluation budget t in domain CC. Mean regret is normalized by the initial design's mean regret.

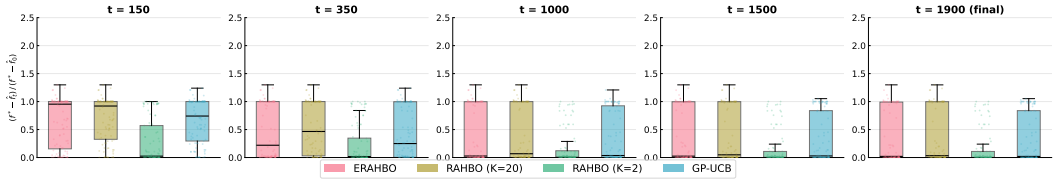

Figure 12: Normalized mean-variance regret at different evaluation budget t in domain Xland. Mean regret is normalized by the initial design's mean regret.

Table 5: First evaluation steps where the IQM of normalized MV regret drops below threshold (% of initial-design regret). "—" means not reached.

	ERAHBO			RAHBO (K=20)			RAHBO (K=2)			GP-UCB		
Combo	75%	50%	25%	75%	50%	25%	75%	50%	25%	75%	50%	25%
dqn/cc_acrobot	21	65	151	21	81	221	7	25	75	21	101	241
dqn/cc_cartpole	43	135	—	61	121	—	259	1185	—	421	781	—
dqn/cc_mountain_car	4	61	—	21	61	—	663	—	—	1	—	—
dqn/xland_empty_random	—	—	—	—	—	—	727	—	—	—	—	—
dqn/xland_four_rooms	97	130	217	121	241	321	19	29	33	121	201	301
ppo/brax_fast	48	223	389	81	201	581	75	121	889	81	241	1081
ppo/brax_halfcheetah	23	81	—	41	261	—	3	121	—	41	—	—
ppo/cc_acrobot	48	63	92	81	141	181	21	37	65	61	101	141
ppo/cc_cartpole	21	63	87	21	81	161	25	57	129	21	81	161
ppo/cc_continuous_mountain_car	43	151	270	81	161	561	11	15	33	61	101	121
ppo/cc_mountain_car	898	—	—	581	—	—	—	—	—	—	—	—
ppo/cc_pendulum	21	43	210	21	141	381	25	65	—	21	—	—
ppo/xland_door_key	1238	—	—	—	—	—	13	23	45	761	1241	—
ppo/xland_empty_random	41	145	403	61	101	281	9	19	61	81	141	201
ppo/xland_four_rooms	69	123	165	101	141	221	23	27	37	101	141	221
sac/brax_fast	21	25	120	21	41	321	37	111	201	21	141	1001
sac/brax_halfcheetah	23	27	427	21	61	101	—	—	—	41	81	301
sac/cc_continuous_mountain_car	63	83	125	81	101	181	15	19	29	81	141	181
sac/cc_pendulum	49	85	111	121	181	321	17	25	75	121	201	301

C.5 Schedule β_{stop}

It can be observed, e.g., in Figure 2, that RAHBO (k=2) ranks best for small budgets, indicating that exploring with minimal repetition can be efficient at this stage. Inspired by this, we propose to increase the LCB of the incumbent. Recall (10) as

$$B_t := \max_{x \in \mathcal{D}_{t-1}} \mathrm{LCB}_t^{\mathrm{MV}}(x). \tag{34}$$

We set the confidence parameter β in the $\mathrm{LCB}_t^{\mathrm{MV}}(x)$ as $\beta_{\text{stop},t}$ and increase it with an exponential schedule

$$\beta_{\text{stop},t} = \frac{e^{\gamma\tau} - 1}{e^{\gamma} - 1} \tag{35}$$

or a sigmoid schedule

$$\beta_{\text{stop},t} = \frac{\sigma_s(\tau) - \sigma_s(0)}{\sigma_s(1) - \sigma_s(0)} \tag{36}$$

where $\tau = t/T$ is the normalized progress, with t being the current evaluation and T being the total evaluation budget, $\sigma_s(\tau) = 1/(1 + e^{s(\tau - 0.5)})$ is a sigmoid function, γ and s are hyperparameter of the schedule. We choose different schedule hyperparameters and illustrate them in Figure 13. Moreover, we apply the same $\beta_{\text{stop},t}$ to the UCB in (12). We note that it is independent of the confidence parameter β in the acquisition function.

Figure 14 shows the domain-aggregated mean rank over time for ERAHBO with different schedules. We see that most schedules can outperform RAHBO $k = 2$ after 200-500 evaluations, while some consistently outperform ERAHBO throughout the budget. Notably, on XLand, an exponential schedule with $\gamma = 50$ can drastically improve ERAHOBO, as it not only outperforms a constant schedule but also ranks comparably close to RAHBO $k = 2$. Even though the schedule introduces one additional hyperparameter, we see the performance is at least comparable with ERAHBO with a constant β_{stop} as long as the initial $\beta_{\text{stop},t}$ value of the schedule is small enough.

A small initial β_{stop} benefits ERAHBO from a more optimistic triggering of the confidence-based resampling rule (12) early on; therefore, ERAHBO with such a schedule will start with investing fewer repetitions for rejection, however, still resamples thoroughly if the configuration is promising enough. As the β_{stop} increases with the schedule, promising configurations are exploited with more repetitions.

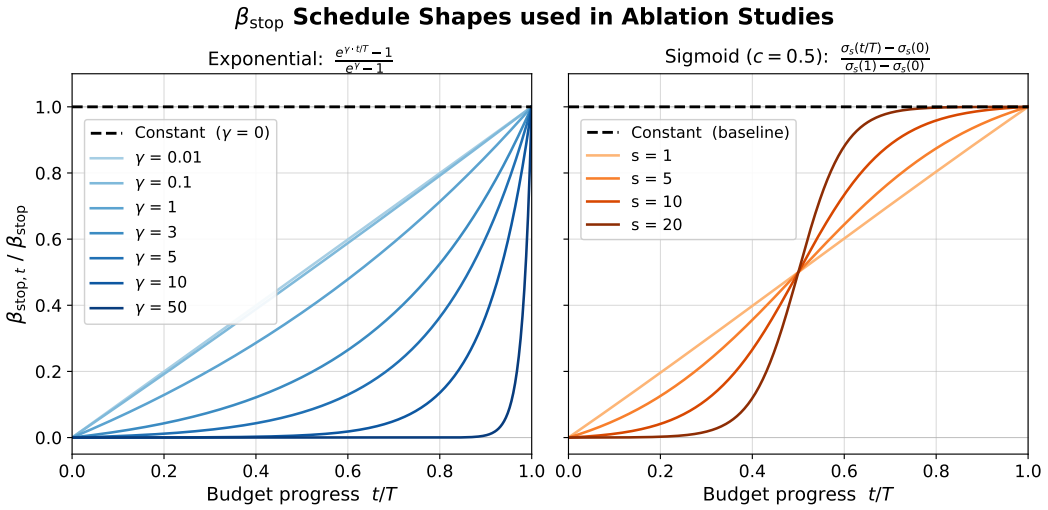

Figure 13: The β_{stop} schedule used in the ablation study. The horizontal axis is the evaluation progress t normalized by the total evaluation budget T, and the vertical axis is the instantaneous value $\beta_{\text{stop},t}$, normalized by the maximum β_{stop}, at normalized progress t/T.

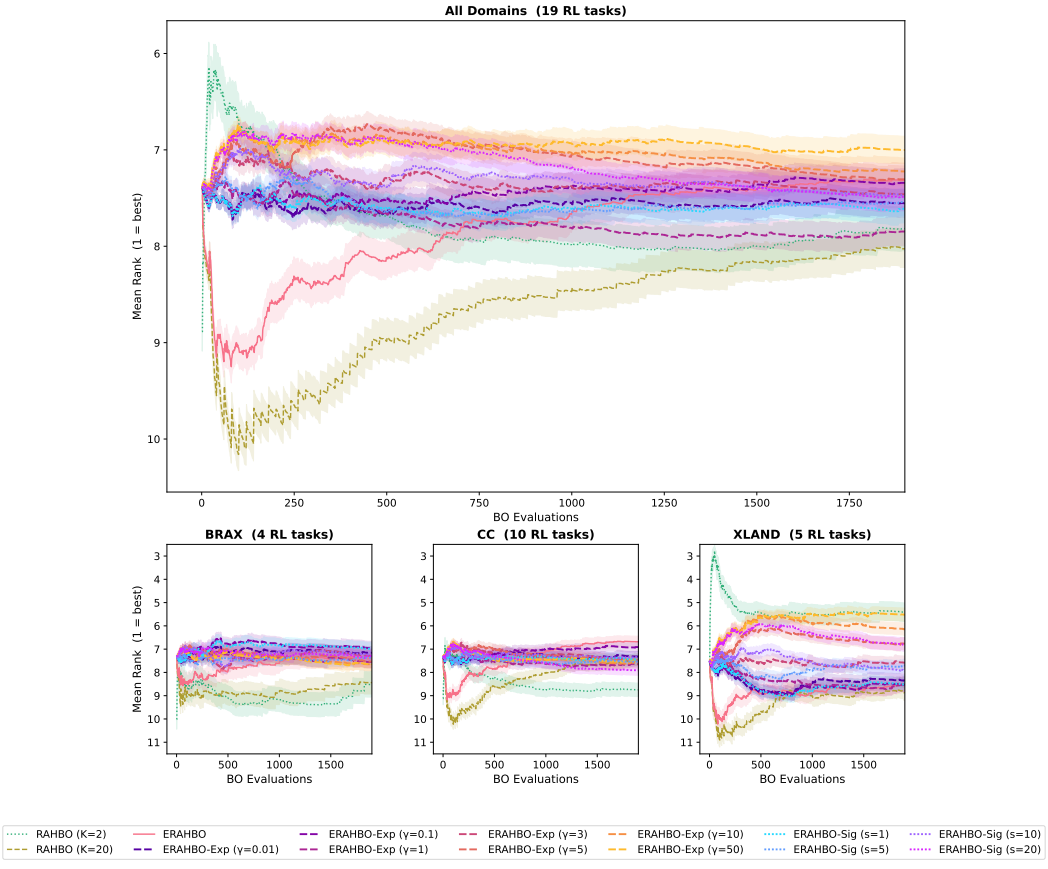

Figure 14: Domain aggregated mean rank over evaluation. The line and shaded area show the mean and $1\times$ standard error, respectively. Schedules with steep rises can outperform constant schedules.