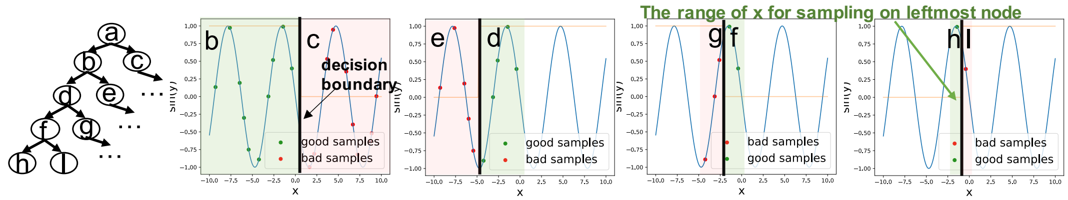


(a) search tree

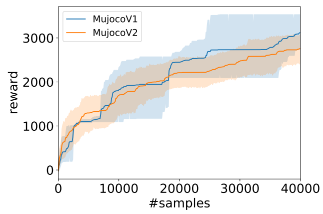
(b) split on node **a**
average sin(x): -0.234
average sin(x): 0.127

(c) split on node **b**
average sin(x): 0.05
average sin(x): 0.204

(d) split on node **d**
average sin(x): -0.123
average sin(x): 0.694

(e) split on node **f**
average sin(x): 0.398
average sin(x): 0.99

Rebuttal-Fig. 1: the visualization of partitioning 1d $\sin(x)$.



Rebuttal-Fig. 2: LA-MCTS on Walker2d

Task	Reward Threshold	#episodes needed by LA-MCTS to get threshold
Swimmer-v1	325	**126**
Hopper-v1	3120	2913
HalfCheetah-v1	3430	3967
Walker2d-v1	4390	N/A($r_{best} = 3523$)
Ant-v1	3580	N/A($r_{best} = 2871$)
Humanoid-v1	6000	N/A($r_{best} = 3202$)

Table 1: Averaged samples to reach the reward threshold on Mujoco-V1. Table. 2 in the main paper uses Mujoco-V2.

We sincerely thank reviewers R1 , R2 , R3 for their constructive feedbacks. We answer the questions as follows:

R1 Mujoco versions: Thanks for pointing out! We redo the experiment on Mujoco-V1 in Table. 1. LA-MCTS shows similar performance between V1 and V2 except for Walker2d, where LA-MCTS does slightly better (Rebuttal-Fig.2), consistent with previous reports[1].

R1 How to quantify regret reaching the plateau: For each node, a minimal number of samples are needed to establish a decent local model, while more samples do not help improve its performance substantially, due to the curse of dimensionality. This is when *a plateau of regret* happens. In this case, it is better to split the region so that the future sampling focuses on promising (and smaller) regions, yielding higher sample efficiency. To determine when to split, we introduced a hyper-parameter *splitting threshold* and provided ablation study in Fig. 6(c), which indeed shows there is a sweetspot of splitting threshold. We will clarify it in the paper.

R1 Partition of one-dimensional $\sin(x)$**. Will interleaving high/low function values cause problem?**: We cluster data points using $(\mathbf{x}, f(\mathbf{x}))$, which are $d+1$ dimensional vectors. Since it involves the input features $\mathbf{x}$, K-means will consider the vicinity of data points and group close points together, preventing the interleaving pattern from happening. As shown in Rebuttal-Fig. 1, for one-dimensional $\sin(x)$, for the splitting sequence $a \to b \to d \to f \to h$, LA-MCTS first groups local regions together, then gradually focuses on a particular peak and make refinements around it.

R1 Show the leftmost is the best leaf: By construction, the value of a left child $(v_l) >$ a right child (v_r) and by recursively applying this rule on the tree, the leftmost node *is expected to be* the highest value node. However, it is possible that the current local models at each nodes may not be correct, due to insufficient samples. Fig. 10 in appendix shows this behavior (iterations 0→3). The exploration in MCTS alleviates this issue by visiting different leaves to capture a global view of the search space and update the learned partition accordingly.

R1 Deterministic assumption in LA-MCTS: LA-MCTS can also be applied to stochastic black-box function. None of its components require the function to be deterministic, while it is possible that for stochastic function, more samples are needed to learn partition and to fit a local model at the leaves. In Mujoco, LA-MCTS, as well as all black-box optimization baselines we compare against, uses an average rewards from 5 different trajectories (or episodes) to mimic deterministic rewards when evaluating a sampled policy.

R3 Hyperparameters. *1) How to choose the length scale of RBF?* We used SVM in scikit-learn, and the length scale of a RBF kernel is decided by a hyper-parameter gamma with two choice of values *auto* and *scale*. We notice scale is better than auto in practice. *2) Choice of hyper-parameters for baselines?* We do have carefully chosen the hyper-parameters for baselines. For example, Shiwa is a meta-method that internally optimizes hyper-parameters for CMA-ES; we used suggested hyper-parameters from scikit-learn for Diff-Evo, Anneal and CMA-ES. The setting of TuRBO inside LA-MCTS is exactly the same as TuRBO used in baselines; We also tuned the embedding size for HesBO and used the suggested settings for BOHB.

R2 Is LA-MCTS maximizing? LA-MCTS is maximization; and we can change a minimization to a maximization by multiplying -1 in $f(\mathbf{x})$.

Other issues. We will correct typos in the next iteration. R2 The citation (line 225) is to point out the source of using a linear policy. "gibbson sampling" means "Gibbs sampling" and we will fix all "STOA" to be "SoTA".

[1] `https://github.com/openai/gym/pull/834`