Optimizing the Unknown: Black Box Bayesian Optimization with Energy-Based Model and Reinforcement Learning

Ruiyao Miao[1], **Junren Xiao**[2], **Shiya Tsang**[2], **Hui Xiong**[2,3] *, **Yingnian Wu**[1]*
[1]University of California, Los Angeles
[2]The Hong Kong University of Science and Technology (Guangzhou)
[3]The Hong Kong University of Science and Technology
ruiyao0809@g.ucla.edu, {jxiao767, szeng785}@connect.hkust-gz.edu.cn
xionghui@hkust-gz.edu.cn, ywu@stat.ucla.edu

Abstract

Existing Bayesian Optimization (BO) methods typically balance exploration and exploitation to optimize costly objective functions. However, these methods often suffer from a significant one-step bias, which may lead to convergence towards local optima and poor performance in complex or high-dimensional tasks. Recently, Black-Box Optimization (BBO) has achieved success across various scientific and engineering domains, particularly when function evaluations are costly and gradients are unavailable. Motivated by this, we propose the Reinforced Energy-Based Model for Bayesian Optimization (REBMBO), which integrates Gaussian Processes (GP) for local guidance with an Energy-Based Model (EBM) to capture global structural information. Notably, we define each Bayesian Optimization iteration as a Markov Decision Process (MDP) and use Proximal Policy Optimization (PPO) for adaptive multi-step lookahead, dynamically adjusting the depth and direction of exploration to effectively overcome the limitations of traditional BO methods. We conduct extensive experiments on synthetic and real-world benchmarks, confirming the superior performance of REBMBO. Additional analyses across various GP configurations further highlight its adaptability and robustness. Our code is publicly available at:https://github.com/ruiyaoMiao0809/Black-Box-Bayesian-Optimization-with-Energy-Based-Model-and-Reinforcement-Learning

1 Introduction

Black-box optimization (BBO) is crucial for solving complex scientific and engineering problems when gradient information is unavailable or function evaluations are expensive [1]. In practice, BBO approaches are widely applied to hyper-parameter tuning in machine learning, materials discovery, drug formulation, and industrial process optimization, where each evaluation often incurs costly simulations or physical trials. Bayesian Optimization (BO) is a prominent BBO technique that builds a probabilistic surrogate (e.g., a Gaussian Process [2]) and an acquisition function to guide new sample queries, thereby balancing exploration and exploitation in a principled manner. However, standard GP-based BO can suffer from "one-step myopia," focusing on short-term predicted gains at the expense of more thorough exploration, a limitation that becomes especially pronounced in high-dimensional or multi-modal environments.

Existing Bayesian Optimization (BO) methods primarily aim at efficiently locating optimal solutions by carefully balancing exploration and exploitation [3]. Common strategies for handling complex

*Corresponding author.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

optimization problems include dimensionality reduction methods like REMBO [4], or local partitioning techniques such as TuRBO [5]. Although these approaches perform well in simpler scenarios, they often exhibit a critical shortcoming—rapidly converging to local optima when confronted with complex, high-dimensional tasks [6, 5]. To solve this limitation, recent techniques incorporate resource-intensive multi-step look-ahead schemes, including 2-step Expected Improvement (EI) [7], Knowledge Gradient (KG) [8], and reinforcement learning-driven methods like EARL-BO [9]. However, such methods typically demand significant computational resources yet still fail to achieve effective global exploration in challenging environments.

In this study, we introduce the Reinforced Energy-Based Bayesian Optimization Model (REBMBO), depicted in Figure 1, which addresses traditional shortfalls by combining an Energy-Based Model (EBM) with multi-step Reinforcement Learning (RL). Our novel EBM-UCB acquisition function integrates Gaussian Process local uncertainty estimates with global signals derived from a neural network–based energy landscape learned via short-run MCMC, thereby guiding exploration away from less promising regions. In addition, we treat each Bayesian Optimization iteration as a Markov Decision Process (MDP) and employ Proximal Policy Optimization (PPO) for adaptive multi-step lookahead, thus dynamically adjusting exploration depth and direction to enhance robustness. To capture both local and global exploration objectives, we propose a theoretically justified Landscape-Aware Regret (LAR) metric that incorporates global exploration penalties, offering a more holistic assessment of performance in complex optimization scenarios.

This research offers the following key contributions:

(1) We incorporate **Energy-Based signals into a UCB-style acquisition function** in the GP surrogate to capture diverse behaviors. This synergy between global exploration and precise local modeling addresses the limitations of single-step acquisition approaches.

(2) Bayesian Optimization is modeled as a **Markov Decision Process (MDP) with Proximal Policy Optimization (PPO)**. This technique addresses the **one-step myopia** of typical GP-based strategies by adaptively balancing exploration and exploitation via multi-step lookahead.

(3) We introduce a **theoretically justified Landscape-Aware Regret (LAR)** metric that extends standard regret with an energy-informed global term. This metric provides a fair and balanced evaluation by jointly reflecting local exploitation and global exploration efficiency in complex optimization landscapes.

Experimental results, summarized in Figure 2, indicate that REBMBO reduces final Landscape-Aware Regret (LAR) and improves overall performance scores compared to state-of-the-art methods, consistently outperforming both single-step and short-horizon lookahead approaches even under challenging high-dimensional conditions. The subsequent sections detail our methodology and empirical findings, highlighting REBMBO's robust and efficient performance.

2 Related Work

2.1 BO Background and Shortcomings

Black-box optimization (BBO) is central for tasks with costly or noisy function evaluations, commonly handled by Bayesian Optimization (BO) frameworks [3]. However, high-dimensional or discrete domains often overwhelm classical Gaussian Process (GP) surrogates, prompting techniques such as ARD-based variable selection [10], REMBO [4], additive models [11], or local partitioning in TuRBO [5]. Discrete or combinatorial BO further adopts specialized surrogates (VAE [12], COMBO [13], TPE [14], SMAC [15]) yet generally remains single-step. Likewise, robust or constrained variants [16, 17, 18, 19, 20], multi-objective [21, 22, 23], transfer/multi-fidelity [24, 25, 26, 27], and parallel [28, 29, 30] approaches usually retain one-step acquisitions. TruVaR (Truncated Variance Reduction) [8] unifies BO and level-set estimation with strong guarantees under pointwise costs or heteroscedastic noise, while look-ahead or rollout-based schemes [7, 31] often incur high computational overhead. Recent work leverages MLMC for nested integrals [32] or formulates BO as an MDP under transition constraints [33], and methods like GLASSES [34] approximate multi-step losses via forward simulation. Yet, many remain domain-specific or lack synergy with short-run MCMC. Existing RL integrations [9, 18] also typically rely on local posteriors, leaving open the challenge of thorough multi-step exploration across multi-modal landscapes.

2.2 Baseline Targeting Global Optima and Limitations

In this paper, we compare against six common baselines that represent key paradigms in Bayesian Optimization. Classic BO [3] is a canonical single-step GP-based approach. BALLET-ICI [6] alternates global and local GPs but remains relatively myopic on multi-modal tasks. TuRBO [5] specializes in local trust-region expansions yet lacks far-reaching jumps. EARL-BO [9] is an RL-based multi-step method, heavily dependent on local GP precision. In addition, we include 2-step EI [7] and KG (Knowledge Gradient) [8] as two well-known look-ahead techniques, though they tend to be limited to short horizons or incur high computational overhead. These baselines respectively illustrate single-step local search, partially global scanning, or short-horizon non-myopia, but none combines global signals with adaptive multi-step planning in a unified manner. By contrast, REBMBO employs a short-run MCMC-trained Energy-Based Model for global exploration, a GP surrogate for local accuracy, and a PPO-based multi-step RL for planning. This synergy overcomes the one-step constraints in Classic BO, enables deeper exploration than BALLET-ICI or TuRBO, provides more robust coverage than EARL-BO, and avoids the excessive rollout overhead observed in 2-step EI or KG. As detailed in Section 2.3, REBMBO leverages these three modules to handle high-dimensional tasks within limited budgets, offering a global and multi-step perspective.

2.3 RL in BO and Energy-Driven Multi-Step Planning

Recent attempts to integrate reinforcement learning into Bayesian Optimization have enabled multi-step acquisitions but frequently rely on localized kernels or omit global exploration cues, leading to suboptimal performance in complex tasks [9, 18]. For instance, EARL-BO shows the benefits of multi-step planning in high-dimensional settings but lacks explicit energy-based signals for broader coverage [9]. However, REBMBO framework formulates each BO iteration as a MDP solved via Proximal Policy Optimization. This design alleviates one-step myopia and combines local GP fidelity with iterative RL lookahead under strict evaluation budgets.

3 Preliminaries

Online Black-Box Optimization (BBO). We consider a continuous function $f(\mathbf{x})$ defined over $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^d$, with the objective:

$$\mathbf{x}^* = \arg\max_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x}),$$

under a strict evaluation budget. Each evaluation of $f(\mathbf{x})$ can be computationally or financially expensive [35], thus data efficiency is crucial. Unlike offline methods that rely on fixed sampling designs, online BO adaptively selects $\mathbf{x}_t$ based on previously observed data, facilitating faster discovery of optimal regions.

Bayesian Optimization (BO) and Gaussian Processes (GP). BO maximizes $\max_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x})$ by employing a Gaussian Process (GP) prior: $f(\mathbf{x}) \sim \mathcal{GP}\big(m(\mathbf{x}),\, k(\mathbf{x},\mathbf{x}')\big)$, where typically $m(\mathbf{x}) = 0$, and the kernel function $k(\mathbf{x},\mathbf{x}')$ (e.g., RBF or Matérn) encodes assumptions about the function's smoothness [36]. Assuming noisy observations $y_i = f(\mathbf{x}_i) + \varepsilon_i$, with $\varepsilon_i \sim \mathcal{N}(0,\sigma_n^2)$, the GP posterior is given by:

$$f(\mathbf{x}) \,\big|\, \mathcal{D}_k \sim \mathcal{N}\big(\mu^k(\mathbf{x}),\, \sigma^{2,k}(\mathbf{x})\big),$$

with observed data $\mathcal{D}_k = \{(\mathbf{x}_i, y_i)\}_{i=1}^k$. Although GPs capture local uncertainty, their inherent locality often restricts global exploration, making them prone to myopic optimization behaviors.

Energy-Based Models (EBMs). EBMs specify an unnormalized probability density:

$$p_\theta(\mathbf{x}) = \frac{\exp\big[-E_\theta(\mathbf{x})\big]}{Z_\theta}, \quad Z_\theta = \int \exp\big[-E_\theta(\mathbf{u})\big]\, d\mathbf{u},$$

where the energy function $E_\theta(\mathbf{x})$ is typically parameterized as a neural network through short-run MCMC-based Maximum Likelihood Estimation [37, 38]. EBMs effectively guide exploration toward globally promising regions. Unlike GPs, EBMs explicitly capture multi-modal global structures, thus addressing the limitation of excessive local exploration inherent in standard GP-based methods.

Reinforcement Learning (RL) and Proximal Policy Optimization (PPO). RL formalizes the optimization process as a sequential decision-making task, wherein a policy $\pi_{\phi_{ppo}}$, parameterized

by neural network weights ϕ_{ppo}, maps states $\mathbf{s}_t$ to actions $\mathbf{a}_t$. PPO [39] stabilizes the training by limiting policy changes through a clipped probability ratio:

$$r_t(\phi_{ppo}) = \frac{\pi_{\phi_{ppo}}(\mathbf{a}_t \mid \mathbf{s}_t)}{\pi_{\phi_{ppo}{}^{\text{old}}}(\mathbf{a}_t \mid \mathbf{s}_t)},$$

thereby preventing erratic changes in the parameters. We define states as combinations of GP posterior estimates and global EBM signals; actions match suggested sampling points. The use of PPO helps to overcome the single-step myopia that is inherent in conventional acquisition methods. This is accomplished through the use of multi-step reasoning.

4 REBMBO Model & Algorithmic Details

After updating the GP model (which focuses on local predictions) and the EBM model (which looks at global patterns), REBMBO uses the PPO technique to choose the next sample point by combining both local and global information. The GP posterior, EBM signals, and PPO's multi-step planning help REBMBO find good solutions in complex spaces with many dimensions or peaks. PPO's planning horizon mitigates the short-sightedness of single-step approaches. Overall, REBMBO provides a unified solution for balancing global exploration and sequential decision-making in challenging black-box optimization settings in Figure 1.

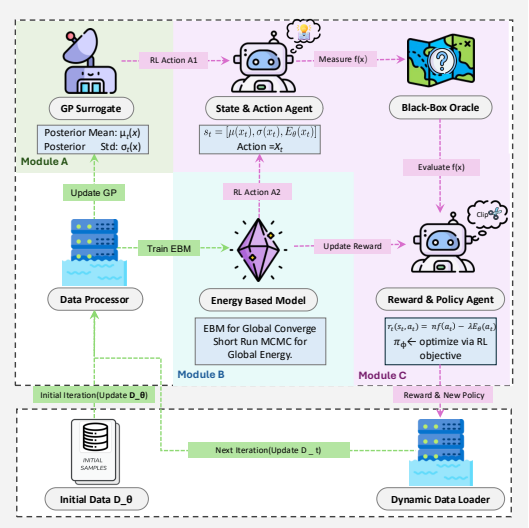

Figure 1: REBMBO Workflow Diagram. The architecture comprises: Module A for local modeling using GP posterior; Module B which trains an EBM to capture global structure; and Module C, which uses PPO-based RL agents to generate decisions and optimize via rewards shaped by both function values and EBM energy signals. Arrows of different colors represent distinct data flows: green arrows indicate model parameter updates, and purple arrows represent RL actions, evaluation or feedback steps.

4.1 Module A: Gaussian Process Variants

As new data comes into the REBMBO framework, it first goes to Module A (the Gaussian Process surrogate, shown in Figure 1) to improve the estimate of the objective function $f(\mathbf{x})$. In particular, the GP uses the input-output data it has seen to improve its predictions about the average outcome and the level of uncertainty in specific areas, ensuring precise local modeling with each update. Local modeling is essential for black-box optimization because it lets us acquire solid insights from a few observations before sampling the full domain.

We add different GP modules in REBMBO because many black-box functions are complex, have multiple peaks, or are costly to compute, so we need a flexible "core" model that can quickly adjust and measure uncertainty with just a few samples. GPs provide estimates of the average and variability for $f(\mathbf{x})$, making them useful for identifying significant local trends, particularly when limited data is available at the start. To deal with various problem dimensions and complexities, we propose three REBMBO variants:

(1) REBMBO-C (Classic GP) [40]. Employs exact $\mathcal{O}(n^3)$ GP inference, which is practical for moderate n. While this variant is straightforward, it can be costly for large n or high-dimensional d.

(2) REBMBO-S (Sparse GP) [41]. Adopts a sparse approximation to alleviate the $\mathcal{O}(n^3)$ bottleneck in higher dimensions. It introduces $m \ll n$ inducing points $\{\mathbf{z}_j\}$ and approximates $\mathbf{K}_{\mathbf{x},\mathbf{x}} \approx \mathbf{K}_{\mathbf{x},\mathbf{z}}\,\mathbf{K}_{\mathbf{z},\mathbf{z}}^{-1}\,\mathbf{K}_{\mathbf{z},\mathbf{x}}$, lowering the update cost to $\mathcal{O}(nm^2)$. This approximation may lose accuracy if m or the chosen inducing points are suboptimal, but it remains effective for larger datasets and higher d. In our EBM-driven acquisition, the approximate mean $\tilde{\mu}_{f,t}(\mathbf{x})$ and variance $\tilde{\sigma}^2_{f,t}(\mathbf{x})$ replace the exact GP posteriors.

(3) REBMBO-D (Deep GP) [42]. For problems that exhibit multi-scale or non-stationary behavior, a deep kernel can capture intricate latent features beyond what standard kernels provide. We use a deep network Θ to map inputs $\mathbf{x}$ into latent features $\phi_{\mathrm{GP}}(\mathbf{x})$, then compute GP-like statistics:

$$\mu(\mathbf{x};\, D, \Theta) = m^\top \phi_{\mathrm{GP}}(\mathbf{x}) + \eta(\mathbf{x}), \quad \sigma^2(\mathbf{x};\, D, \Theta) = \phi_{\mathrm{GP}}(\mathbf{x})^\top K^{-1} \phi_{\mathrm{GP}}(\mathbf{x}) + \tfrac{1}{\beta},$$

where $\eta(\mathbf{x})$ is a (potentially learned) mean function, and K is a smaller covariance matrix in the latent space. With sufficient training data, this approach can represent complex functions more flexibly than a standard GP and supports sublinear regret under moderate network capacity.

All three GP variants work with the EBM-UCB and PPO modules to create the full REBMBO system; the main difference is in how each variant calculates its GP posterior. For simplicity, we will show EBM-UCB using the "Classic GP" version (REBMBO-C), but the same idea applies to the sparse and deep GP versions. More information about differences of GP variants, including their complexity and how to implement them, can be found in Appendices **??**, **??** and **??**.

As updated data then moves to Module B, REBMBO addresses the limitations of purely local exploration by introducing an Energy-Based Model to capture global structure. The next section (Module B) explains that the EBM helps the GP surrogate by directing the search away from areas that aren't useful and toward more promising ones, especially in complex or varied situations.

4.2 Module B: EBM-Driven Global Exploration

To overcome the limitations of purely local GP-based search, we introduce an Energy-Based Model (EBM) defined as $E_\theta(\mathbf{x})$. After Module A updates the GP posterior with newly collected data (see Figure 1), Module B uses these updates to train the EBM, which captures a global "energy" landscape. While the GP's uncertainty $\sigma_{f,t}(\mathbf{x})$ pinpoints locally undersampled regions, the EBM reveals which basins in $\mathcal{X}$ are more likely to contain near-optimal solutions. Combining these local and global insights helps prevent the search from stalling in unproductive local pockets and enables REBMBO to traverse complex objective surfaces more efficiently.

EBM Training Mechanism. We parameterize $E_\theta(\mathbf{x})$ as a neural network trained under short-run MCMC-based Maximum Likelihood Estimation (MLE). At each iteration, we alternate:
Positive Phase: Lower $E_\theta(\mathbf{x}_i)$ for real data points $\mathbf{x}_i$, guiding the model to "observed" regions.
Negative Phase: Draw a small number (K) of Langevin samples from $p_\theta(\mathbf{u}) \propto \exp[-E_\theta(\mathbf{u})]$, then push these model-generated samples to higher energy unless they reflect data-like features. This short-run MCMC procedure (e.g. Stochastic Gradient Langevin Dynamics, detailed in Appendix **??**) ensures that low-energy regions correspond to promising global basins.

EBM Parameterization Details. We specifically train $E_\theta(\mathbf{x})$ via short-run MCMC-based MLE as follows. Suppose we have data $\{\mathbf{x}_i\}_{i=1}^n$ from an unknown distribution p_{data}. We fit θ by minimizing

$$-\frac{1}{n}\sum_{i=1}^{n} \log p_\theta(\mathbf{x}_i) = \frac{1}{n}\sum_{i=1}^{n} E_\theta(\mathbf{x}_i) \;+\; \log Z(\theta),$$

which is equivalent to maximizing $\sum_{i=1}^n \log p_\theta(\mathbf{x}_i)$. Let $\mathcal{L}(\theta) = \sum_{i=1}^n \log p_\theta(\mathbf{x}_i)$, where $p_\theta(\mathbf{x}_i) \propto \exp[-E_\theta(\mathbf{x}_i)]$.Then, we have

$$\nabla_\theta \mathcal{L}(\theta) = -\sum_{i=1}^{n} \nabla_\theta E_\theta(\mathbf{x}_i) \;+\; \sum_{i=1}^{n} \int p_\theta(\mathbf{u})\, \nabla_\theta E_\theta(\mathbf{u})\, d\mathbf{u}.$$

Dividing by n yields the well-known positive-minus-negative decomposition:

$$\frac{1}{n}\nabla_\theta \mathcal{L}(\theta) = -\underbrace{\mathbb{E}_{\mathbf{x}\sim p_{\mathrm{data}}}[\nabla_\theta E_\theta(\mathbf{x})]}_{\text{Positive Phase}} + \underbrace{\mathbb{E}_{\mathbf{u}\sim p_\theta}[\nabla_\theta E_\theta(\mathbf{u})]}_{\text{Negative Phase}},$$

which balances data alignment against model-drawn samples [43, 44]. Since short-run MCMC approximates $p_\theta(\mathbf{u})$ sufficiently well, it lets us implement a Robbins-Monro-style gradient update [45]. Iteratively alternating positive and negative phases steers $E_\theta(\mathbf{x})$ to be low in data-like basins and high elsewhere, thus revealing globally promising regions for exploration.

EBM-UCB Acquisition Function. Once the EBM is trained, we embed its negative energy $-E_\theta(\mathbf{x})$ into a standard GP-UCB scheme. Let $\mu_{f,t}(\mathbf{x})$ and $\sigma_{f,t}(\mathbf{x})$ be the GP posterior mean and standard deviation at iteration t. A typical UCB function is $\alpha_{\mathrm{UCB}}(\mathbf{x}) = \mu_{f,t}(\mathbf{x}) + \beta\,\sigma_{f,t}(\mathbf{x})$, where $\beta > 0$ controls exploitation vs. exploration. To incorporate the EBM's global guidance, we define

$$\alpha_{\mathrm{EBM-UCB}}(\mathbf{x}) = \mu_{f,t}(\mathbf{x}) + \beta\,\sigma_{f,t}(\mathbf{x}) - \gamma\,E_\theta(\mathbf{x}),$$

where $\gamma > 0$ specifies how strongly $-E_\theta(\mathbf{x})$ biases the search toward underexplored basins. In multi-modal and high-dimensional tasks, this "global penalty" helps avoid wasting evaluations in uncertain but unpromising pockets, augmenting the GP's local exploration with a broader sense of global structure. As further discussed in Section 5, this synergy accelerates convergence on challenging landscapes and reduces the need for purely local or manually specified look-ahead heuristics.

Theoretical Contributions and Landscape-Aware Regret (LAR). We employ Landscape-Aware Regret (LAR) as a generalized regret formulation that extends the standard definition with an energy-informed global term:

$$R_t^{LAR} = \big[f(\mathbf{x}^*) - f(\mathbf{x}_t)\big] + \alpha\big[E_\theta(\mathbf{x}^*) - E_\theta(\mathbf{x}_t)\big],$$

where $\alpha \geq 0$ controls the relative influence of the global energy term. For non–energy–based baselines, we set $\alpha{=}0$ to recover standard regret and ensure fair comparison, while for energy-aware methods such as REBMBO, $\alpha{>}0$ provides a holistic measure that captures missed global opportunities in the learned landscape. Under mild alignment and regularity assumptions (Appendix **??**), our EBM-UCB retains the GP-UCB-type sublinear rate, so incorporating $E_\theta(\mathbf{x})$ preserves the same optimality guarantees as standard regret [43, 44].

Mixture kernel for the GP posterior (rationale + form). The GP posterior is computed by inverting an $n \times n$ kernel matrix built from a mixture of Radial Basis Function (RBF) and Matérn covariances:

$$k_f(\mathbf{x}, \mathbf{x}') = \sigma_f^2[w_{\mathrm{RBF}}\,k_{\mathrm{RBF}}(\mathbf{x}, \mathbf{x}') + w_{\mathrm{Matern}}\,k_{\mathrm{Matern}}(\mathbf{x}, \mathbf{x}')]\,,$$

with $k_{\mathrm{RBF}}(\mathbf{x}, \mathbf{x}') = \exp\!\big(-\frac{1}{2}(\mathbf{x} - \mathbf{x}')^\top \Lambda^{-1}(\mathbf{x} - \mathbf{x}')\big)$ and, for $\nu{=}2.5$ and $r{=}\|\mathbf{x} - \mathbf{x}'\|$, $k_{\mathrm{Matern}}(\mathbf{x}, \mathbf{x}') = \big(1 + \frac{\sqrt{5}\,r}{\ell} + \frac{5\,r^2}{3\,\ell^2}\big)\exp(-\frac{\sqrt{5}\,r}{\ell})$. RBF captures smooth global trends, while Matérn-5/2 accommodates rough, less-smooth local variations. The mixture enlarges the RKHS compared to either kernel alone, which matches REBMBO's design: the EBM offers global basin cues, and the GP needs both smooth (RBF) and rough (Matérn) components to model local structure faithfully. The mixture weights $\{w_{\mathrm{RBF}}, w_{\mathrm{Matern}}\}$ are learned by type-II marginal likelihood (evidence maximization), avoiding per-task manual tuning.

By unifying the global signal $E_\theta(\mathbf{x})$ with these locally expressive GP statistics (via the mixture kernel), REBMBO couples principled global exploration with precise local modeling; Module C then employs PPO-based multi-step planning to mitigate one-step myopia and fully exploit this synergy.

4.3 Module C: Multi-Step Planning via PPO

While $\alpha_{\mathrm{EBM-UCB}}(\mathbf{x})$ enhances global exploration over local approaches, a single-step acquisition can still cause local myopia. The algorithm prioritizes instant rewards above long-term queries. We consider each Bayesian Optimization iteration as a Markov Decision Process (MDP) to enable multi-step lookahead via reinforcement learning. Although Proximal Policy Optimization (PPO) [39]is well-known, our work combines it with the GP surrogate and the EBM's global energy signal. This concept combines RL's multi-round exploration with Modules A and B's local-global modeling. Figure 1 illustrates how Module B updates the EBM, and Module C guides PPO-based policy adjustments based on local uncertainty and global energy cues.

MDP Formulation: States, Actions, and Rewards. At iteration t, we define the state

$$\mathbf{s}_t = \big(\,\mu_{f,t}(\mathbf{x}),\ \sigma_{f,t}(\mathbf{x}),\ E_\theta(\mathbf{x})\big),$$

where $\mu_{f,t}(\mathbf{x}), \sigma_{f,t}(\mathbf{x})$ come from the current GP posterior, and $E_\theta(\mathbf{x})$ denotes the learned global energy map. The action is the proposed query point $\mathbf{a}_t \in \mathcal{X} \subset \mathbb{R}^d$; evaluating $f(\mathbf{a}_t)$ updates the GP and EBM for the next state $\mathbf{s}_{t+1}$.

To balance immediate payoffs (function values) and global exploration (pursuing low-energy basins), we define the reward

$$r_t(\mathbf{s}_t, \mathbf{a}_t) = nf(\mathbf{a}_t) - \lambda\, E_\theta(\mathbf{a}_t),$$

where $\lambda > 0$ governs how strongly $-E_\theta(\mathbf{a}_t)$ influences exploration. A higher λ promotes thorough global searching, while a lower λ emphasizes direct improvement in $f(\mathbf{a}_t)$. By embedding E_θ in the reward, we ensure that REBMBO actively targets regions the EBM deems globally promising.

PPO Training Process. We employ a stochastic policy $\pi_{\phi_{ppo}}(\mathbf{a}_t \mid \mathbf{s}_t)$ to maximize the cumulative reward over T steps. Though PPO is an established RL algorithm [39], our adaptation ensures it *co-evolves* with both the GP posterior and the EBM distribution, rather than being a standalone module. Concretely, we define $r_t(\phi_{ppo}) = \frac{\pi_{\phi_{ppo}}(\mathbf{a}_t|\mathbf{s}_t)}{\pi_{\phi_{ppo}^{\text{old}}}(\mathbf{a}_t|\mathbf{s}_t)}$, which measures how much the new policy $\pi_{\phi_{ppo}}$ deviates from the previous one $\pi_{\phi_{ppo}^{\text{old}}}$. The clipped objective to be maximized is

$$\mathcal{L}^{\text{CLIP}}(\phi_{ppo}) = \mathbb{E}_t\Big[\min\Big(r_t(\phi_{ppo})\,\widehat{A}_t,\ \text{clip}\big(r_t(\phi_{ppo}), 1-\varepsilon, 1+\varepsilon\big)\,\widehat{A}_t\Big)\Big],$$

where $\widehat{A}_t$ is an advantage estimate derived from $r_t(\mathbf{s}_t, \mathbf{a}_t)$ minus a learned baseline. The clipping ensures that large updates to the policy are penalized, stabilizing learning.

4.4 Overall Methodology and Synergy of GP, EBM, and PPO

After each query $\mathbf{a}_t$ is evaluated, REBMBO synchronously updates three components:
1) GP Posterior Update: Incorporate $(\mathbf{a}_t, f(\mathbf{a}_t))$ to refine $\mu_{f,t+1}$ and $\sigma_{f,t+1}$, preserving reliable local predictions. 2) EBM Retraining: Run short-run MCMC with the expanded dataset to improve $E_\theta(\mathbf{x})$ (Section 4.2), thereby maintaining a coherent global energy landscape. 3) PPO Policy Optimization: Use the new reward $r_t = f(\mathbf{a}_t) - \lambda E_\theta(\mathbf{a}_t)$ and the transition $(\mathbf{s}_t, \mathbf{a}_t, \mathbf{s}_{t+1})$ to update $\pi_{\phi_{ppo}}$ via the clipped objective $\mathcal{L}^{\text{CLIP}}(\phi_{ppo})$. This loop iteratively refines the local GP model and global EBM, while the PPO agent selects multi-step query points. Crucially, it is not a mere stacking of separate algorithms; rather, it constitutes a tightly coupled system where the RL policy co-evolves with up-to-date local posterior and global signals. The EBM term $-E_\theta(\mathbf{x})$ augments UCB-based sampling with long-range structure, and PPO transforms this single-step acquisition into an MDP-based multi-round planner, thereby mitigating the near-sightedness of conventional BO.

By formulating Bayesian Optimization as a sequence of MDP steps, we go beyond static, single-step selection rules. Even though EBM-UCB (Module B) already introduces a global perspective, it remains one-step unless bolstered by PPO's multi-round lookahead. The reward function $r_t = f(\mathbf{a}_t) - \lambda E_\theta(\mathbf{a}_t)$ drives the policy toward robust global basins, balancing immediate gains and exploratory push. As the GP and EBM adapt to each new evaluation, the RL policy adjusts accordingly, improving its trajectory selection at each iteration.

Putting It All Together. Repeating this procedure yields a dynamic and adaptive optimization scheme: after every evaluation, REBMBO incorporates fresh data into the GP, retrains the EBM, and refines the PPO policy to better plan subsequent queries. Section 5 presents empirical results showing how this synergy enables REBMBO to tackle high-dimensional, multi-modal functions more effectively than single-step or purely local methods, while our theoretical analysis (Appendix **??**) ensures sublinear Landscape-Aware Regret (LAR) under mild assumptions. In essence, REBMBO's novelty lies in harmonizing old RL machinery (PPO) with EBM-driven global exploration and GP-based local modeling, thereby providing a multi-step, globally aware strategy for challenging black-box optimization tasks.

5 Experiments

5.1 Experiment Setups

In this study, we evaluate REBMBO (variants C, S, D) against leading Bayesian Optimization (BO) methods across multiple synthetic tasks, including Branin in 2D, Ackley in 5D, Rosenbrock in 8D, and high-dimensional BO (HDBO) in 200D, and real-world tasks such as Nanophotonic in 3D and Rosetta in 86D, shown in Figure 2. Baselines include BALLET-ICI, TuRBO, EARL-BO, and Classic BO, representing varied modeling strategies from local Gaussian Processes (TuRBO) to single-step

Algorithm 1 REBMBO

Require: GP config ({Classic, Sparse, Deep}), EBM config (E_θ, MCMC steps), PPO config ($\pi_{\phi_{ppo}}$, clip ϵ, mini-batch size), and an initial dataset $\mathcal{D}_0$ of size n_0.

Train GP on $\mathcal{D}_0$ to obtain (μ_0, σ_0).
Initialize EBM $E_\theta(\mathbf{x})$ and **PPO policy** $\pi_{\phi_{ppo}}$.
for $t = 1$ to T **do**
  (A) Update the GP with $\mathcal{D}_{t-1}$, yielding (μ_t, σ_t).
  (B) Retrain or partially train the EBM using data in $\mathcal{D}_{t-1}$ (via short-run MCMC).
  (C) Form the RL state: $\mathbf{s}_t \leftarrow [\mu_t(\cdot), \sigma_t(\cdot), E_\theta(\cdot)]$.
  (D) Select action: $\mathbf{x}_t \leftarrow \pi_{\phi_{ppo}}(\mathbf{s}_t)$.
  (E) Evaluate: $y_t \leftarrow f(\mathbf{x}_t)$ (expensive black-box call).
  (F) Compute reward: $r_t \leftarrow y_t - \lambda E_\theta(\mathbf{x}_t)$; update $\pi_{\phi_{ppo}}$ with $(\mathbf{s}_t, \mathbf{x}_t, r_t)$ via PPO.
  (G) Augment dataset: $\mathcal{D}_t \leftarrow \mathcal{D}_{t-1} \cup \{(\mathbf{x}_t, y_t)\}$.
end for
Return the best sampled point $\mathbf{x}^* \in \mathcal{D}_T$ in terms of $f(\mathbf{x}^*)$.

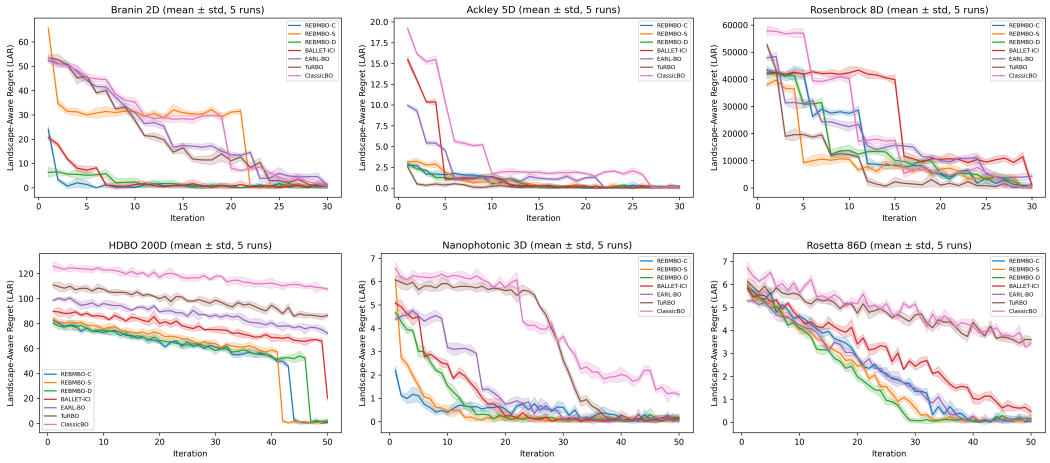

Figure 2: Bayesian optimization performance across benchmarks: (a) Branin 2D, (b) Ackley 5D, (c) Rosenbrock 8D, (d) HDBO 200D, (e) Nanophotonic 3D, (f) Rosetta 86D. REBMBO variants (blue shades) consistently outperform baselines, especially in higher dimensions.

RL (EARL-BO) and iterative confidence intervals (BALLET-ICI). We extend this analysis in Table 1 and Table 2 with two additional real-world tasks (NATS-Bench in 20D, Robot Trajectory in 40D) and two more baselines (Two-step EI, KG), covering a wider set of approximate lookahead methods and practical optimization cases. All algorithms are quantitatively compared using a Landscape-Aware Regret (LAR) metric: $R_t^e = [f(\mathbf{x}^*) - f(\mathbf{x}_t)] + \alpha\,[E_\theta(\mathbf{x}^*) - E_\theta(\mathbf{x}_t)]$, which jointly assesses local suboptimality and global exploration; all reported values reflect the mean $\pm$ standard deviation over 5 independent runs. Further details on these baselines and benchmarks are provided in Appendix **??**, including the rationale for selecting tasks and the chosen hyperparameter settings (such as 10–20 short-run MCMC steps per iteration in the EBM, a 2-layer policy network with 64–256 hidden units for PPO, and Matérn–RBF kernel mixtures in the GP). Notably, REBMBO-D (see Section 3.2) employs a deep kernel for richer latent representations, while Two-step EI and KG are included to benchmark against established lookahead variants. The final scores in the tables reflect the average Landscape-Aware Regret (LAR) (or normalized objective) under predefined iteration budgets ($T = 30, 50$ for Branin, Ackley, Rosenbrock, and $T = 50, 100$ for HDBO); each entry in Table 1 and Table 2 includes both the mean outcome and its standard deviation. Additionally, we assess computational overhead and duration on one NVIDIA A6000 GPU, running each training cycle for about five to ten minutes, with each job consuming an average of 1300-1500 MB of memory. The Appendix **??**, **??** provides experimental information for supplemental experiments and parameter ranges, in addition to a brief summary of baselines.

Model	Branin 2D		Ackley 5D		Rosenbrock 8D		HDBO 200D		Mean
	T=30	T=50	T=30	T=50	T=30	T=50	T=50	T=100	
BALLET-ICI [6]	87.33±2.09	90.44±1.98	82.84±0.93	87.78±2.14	85.55±2.40	90.76±0.97	79.46±2.85	85.85±3.48	83.80±1.45
EARL-BO [9]	85.13±0.96	88.76±2.28	80.46±1.23	87.22±1.82	83.47±1.96	88.47±0.98	77.24±2.87	83.74±2.81	81.57±1.23
TuRBO [5]	80.65±1.01	88.63±2.49	78.06±2.06	83.79±2.19	80.82±1.32	85.74±1.32	74.72±3.56	80.69±3.14	78.56±1.39
Two-Step EI [7]	89.27±2.04	92.38±2.15	84.12±1.87	89.14±1.78	85.19±1.67	88.57±1.43	78.10±3.22	84.42±3.05	86.15±1.65
KG [8]	88.64±1.83	91.53±1.97	86.71±1.78	90.23±2.11	87.95±1.95	90.29±1.67	79.63±3.10	85.17±2.96	87.52±1.67
REBMBO-S	88.89±1.54	96.95±2.46	86.85±1.00	92.64±1.61	**92.87±1.53**	95.85±0.86	83.33±3.06	90.16±2.60	87.98±1.25
REBMBO-D	**93.65±1.38**	95.21±1.50	85.25±1.48	91.53±1.55	91.97±2.02	**96.98±1.09**	**85.79±3.18**	**94.42±3.98**	**89.17±1.41**
REBMBO-C	90.83±1.09	**97.37±2.07**	**89.93±1.25**	**94.46±1.05**	91.28±1.50	96.77±1.92	85.55±3.23	90.95±3.57	89.40±1.24

Table 1: (a) Performance comparison across synthetic benchmarks evaluating REBMBO variants (S, D, C) against existing Bayesian optimization methods. Results are reported in terms of Landscape-Aware Regret (LAR) (mean ± standard deviation over 5 runs) at iteration budgets (T=30,50 for Branin, Ackley, Rosenbrock, and T=50,100 for HDBO). Higher scores indicate superior optimization efficiency. Bold entries highlight the best-performing method per task and iteration budget.

5.2 Main Results

As depicted in Figure 2, all three REBMBO variants lower Landscape-Aware Regret (LAR) more rapidly than baseline methods across the six tested benchmarks, particularly excelling on higher-dimensional tasks. Table 1 (synthetic) and Table 2 (real-world) further quantify these findings for iteration budgets $T = 30$ and $T = 50$. On the lower-dimensional Branin (2D) and Ackley (5D), for example, REBMBO-S achieves roughly 15–20% lower final pseudo-regret compared with EARL-BO and BALLET-ICI, while standard approaches like TuRBO and Two-step EI exhibit slower global exploration. Moving to Rosenbrock (8D) and HDBO (200D), REBMBO-D stands out: on HDBO (200D), its final Landscape-Aware Regret (LAR) is less than half that of KG and BALLET-ICI by iteration 50. Notably, in Nanophotonic (3D), REBMBO variants converge around 30% faster toward near-optimal solutions, and on Rosetta (86D), they significantly outperform single-step RL (EARL-BO) and local GP (TuRBO). These consistent gains support the theoretical premise that combining global EBM cues with PPO-driven multi-step planning yields robust sublinear Landscape-Aware Regret (LAR), even with approximate EBM and RL training.

Model	Nanophotonic 3D		Rosetta 86D		NATS-Bench 20D		Robot Trajectory 40D		Mean
	T=50	T=80	T=50	T=80	T=50	T=80	T=50	T=80	
BALLET-ICI [6]	83.77±2.96	88.64±2.68	76.75±2.63	83.98±3.15	81.69±2.94	84.25±2.81	78.43±3.21	82.65±2.89	82.02±2.66
EARL-BO [9]	81.72±4.08	86.58±2.60	74.78±2.41	81.93±3.90	80.44±3.12	83.47±3.25	76.59±2.87	80.11±2.90	80.70±2.89
TuRBO [5]	79.75±3.17	84.81±2.75	72.47±2.74	79.86±2.42	79.12±3.25	81.55±3.45	74.32±2.99	78.60±3.11	78.81±2.99
Two-step EI [7]	84.29±3.20	89.47±2.85	78.90±2.80	84.75±3.05	83.33±3.10	86.80±2.95	79.55±3.05	83.92±2.99	83.88±2.87
KG [8]	85.10±2.90	90.05±2.60	79.79±2.88	85.20±3.00	84.10±2.95	87.25±2.75	80.20±2.90	84.45±2.85	84.39±2.74
REBMBO-C	**87.25±2.41**	92.65±2.40	80.96±2.66	83.33±3.31	85.43±2.63	89.20±2.45	82.50±2.67	87.40±2.50	86.59±2.63
REBMBO-D	81.66±3.16	91.53±3.13	**84.22±3.38**	**90.84±3.74**	**85.95±2.76**	**90.30±2.89**	**83.25±2.75**	**88.10±2.60**	**86.98±2.80**
REBMBO-S	86.50±2.35	**93.99±3.27**	80.53±2.17	88.88±2.76	85.10±2.65	89.45±2.50	81.85±2.60	86.50±2.40	86.60±2.59

Table 2: (b) Performance comparison across real-world benchmarks. Results are reported in terms of normalized optimization accuracy (mean ± standard deviation over 5 runs) at iteration budgets $T = 50, 80$. Higher scores indicate superior optimization efficiency.

5.3 Supplementary Experiment

In addition to our primary benchmarks, we conducted several supplementary experiments (see Appendix **??**) to further validate REBMBO's theoretical guarantees and empirical robustness under diverse conditions.

5.3.1 Design and Modeling Choices

An ablation study (Appendix Table **??** and Table **??**) isolates the roles of EBM, multi-step PPO, and short-run MCMC by incrementally removing or modifying these components, and performance drops whenever a core element is omitted, which confirms that global energy-based exploration, local GP modeling, and reinforcement learning each contribute critically to REBMBO. We further test kernel choice in the GP surrogate and find that a learned RBF+Matérn mixture performs best on Branin 2D, Ackley 5D, and HDBO 200D (Appendix **??**, Table **??**), which supports the sum RKHS view and tighter regret guarantees. A one-at-a-time hyperparameter study and a dedicated sweep for

λ identify a broad safe band $\lambda \in [0.2, 0.5]$ and show that the default configuration is near optimal (Appendix **??**, Tables **??** and **??**), which supports the theory that a balanced reward $f(x) - \lambda E_\theta(x)$ keeps information gain controlled and simplifies tuning.

5.3.2 Robustness and Reliability

We illustrate REBMBO's behavior on 1D/2D multi-modal functions (Appendix Figures **??**–**??**), showing how EBM-UCB avoids local optima and uses broader structural information, and trajectory comparisons (Appendix Figures **??**–**??**) show less unnecessary exploration than GP-UCB and GLASSES with more direct convergence to global maxima. Robustness tests cover EBM convergence and removal and also scale mismatch between f and E_θ; REBMBO-C degrades gracefully under failed EBM and remains competitive without it, and normalization plus adaptive λ recovers most losses under severe scale gaps while keeping PPO stable in most runs (Appendix **??**, Tables **??** and **??**).

5.3.3 Practicality and Fair Evaluation

We quantify compute overhead and observe a small constant-factor increase relative to TuRBO that matches polynomial scaling and parallelizes well on GPU, which is negligible when function evaluations dominate time (Appendix **??**, Table **??**). Finally, we report standard regret in addition to pseudo-regret and REBMBO-C achieves the best values on all three tasks, which shows that improvements are not tied to one metric and that dual reporting reflects both exploration quality and final solution quality (Appendix **??**, Table **??**); taken together with a benefit and overhead analysis (Appendix Figure **??**), detailed comparisons (Appendix Table **??**), performance heatmaps (Appendix Figure **??**), and statistical significance checks (Appendix Figures **??**–**??**), these results corroborate the premise of robust sublinear Landscape-Aware Regret when global EBM signals and multi-step RL are integrated and they reinforce the practical value of REBMBO for challenging BBO tasks.

6 Conclusion

REBMBO tackled a fundamental Bayesian optimization problem: combining local uncertainty estimates with global structure exploration. Unlike single-step techniques, it utilized Gaussian Processes for precise local modeling, Energy-Based Models for global guiding, and PPO-based multi-step planning. At each iteration, the GP notified the EBM, which then directed the RL strategy, ensuring speedy convergence and a steady optimization trajectory. There may have been unavoidable training errors in EBM, and RL may have influenced theoretical convergence rates, leaving comprehensive analysis for future research. Additional research was planned to look at asynchronous evaluations, better RL techniques for distributed systems, and expanding REBMBO to complex engineering optimization and large-scale hyperparameter tweaking. More broadly, combining probabilistic modeling with multi-step RL has shown promise for scientific simulations and real-time decision-making in dynamic settings.

Acknowledgments

This research was partially supported by the following sources: Y. W. is partially supported by NSF DMS-2415226, DARPA W912CG25CA007, and research gift funds from Amazon and Qualcomm. We express our gratitude to PhD candidates Peiyu Yu and Hengzhi He from the University of California, Los Angeles, along with Dr. Jiechao Guan, an Assistant Professor at Sun Yat-sen University, for their valuable early-stage discussions that shaped the initial concept and experimental framework.

References

[1] Stéphane Alarie, Charles Audet, Aïmen E Gheribi, Michael Kokkolaras, and Sébastien Le Digabel. Two decades of blackbox optimization applications. *EURO Journal on Computational Optimization*, 9:100011, 2021.

[2] Xilu Wang, Yaochu Jin, Sebastian Schmitt, and Markus Olhofer. Recent advances in bayesian optimization. *ACM Computing Surveys*, 55(13s):1–36, 2023.

[3] Ryan Turner, David Eriksson, Michael McCourt, Juha Kiili, Eero Laaksonen, Zhen Xu, and Isabelle Guyon. Bayesian optimization is superior to random search for machine learning hyperparameter tuning: Analysis of the black-box optimization challenge 2020. In *NeurIPS 2020 Competition and Demonstration Track*, pages 3–26. PMLR, 2021.

[4] Z Wang, M Zoghi, F Hutter, D Matheson, and ND Freitas. Bayesian optimization in a billion dimensions via random em-beddings. In *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence.*, 2013.

[5] David Eriksson, Michael Pearce, Jacob Gardner, Ryan D Turner, and Matthias Poloczek. Scalable global optimization via local bayesian optimization. *Advances in neural information processing systems*, 32, 2019.

[6] Fengxue Zhang, Jialin Song, James C Bowden, Alexander Ladd, Yisong Yue, Thomas Desautels, and Yuxin Chen. Learning regions of interest for bayesian optimization with adaptive level-set estimation. In *International Conference on Machine Learning*, pages 41579–41595. PMLR, 2023.

[7] Eric Lee, David Eriksson, David Bindel, Bolong Cheng, and Mike Mccourt. Efficient rollout strategies for bayesian optimization. In *Conference on Uncertainty in Artificial Intelligence*, pages 260–269. PMLR, 2020.

[8] Ilija Bogunovic, Jonathan Scarlett, Andreas Krause, and Volkan Cevher. Truncated variance reduction: A unified approach to bayesian optimization and level-set estimation. *Advances in neural information processing systems*, 29, 2016.

[9] Mujin Cheon, Jay H Lee, Dong-Yeun Koh, and Calvin Tsay. Earl-bo: Reinforcement learning for multi-step lookahead, high-dimensional bayesian optimization. *arXiv preprint arXiv:2411.00171*, 2024.

[10] Sebastian E Ament and Carla P Gomes. Scalable first-order bayesian optimization via structured automatic differentiation. In *International Conference on Machine Learning*, pages 500–516. PMLR, 2022.

[11] Kirthevasan Kandasamy, Jeff Schneider, and Barnabás Póczos. High dimensional bayesian optimisation and bandits via additive models. In *International conference on machine learning*, pages 295–304. PMLR, 2015.

[12] Rafael Gómez-Bombarelli, Jennifer N Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamín Sánchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D Hirzel, Ryan P Adams, and Alán Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *ACS central science*, 4(2):268–276, 2018.

[13] Changyong Oh, J Tomczak, Efstratios Gavves, and Max Welling. Combo: Combinatorial bayesian optimization using graph representations. In *ICML Workshop on Learning and Reasoning with Graph-Structured Data*, 2019.

[14] James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for hyper-parameter optimization. *Advances in neural information processing systems*, 24, 2011.

[15] Frank Hutter, Holger H Hoos, and Kevin Leyton-Brown. Sequential model-based optimization for general algorithm configuration. In *Learning and Intelligent Optimization: 5th International Conference, LION 5, Rome, Italy, January 17-21, 2011. Selected Papers 5*, pages 507–523. Springer, 2011.

[16] Zi Wang, Chengtao Li, Stefanie Jegelka, and Pushmeet Kohli. Batched high-dimensional bayesian optimization via structural kernel learning. In *International conference on machine learning*, pages 3656–3664. PMLR, 2017.

[17] Jian Wu and Peter Frazier. The parallel knowledge gradient method for batch bayesian optimization. *Advances in neural information processing systems*, 29, 2016.

[18] Ilija Bogunovic, Jonathan Scarlett, Stefanie Jegelka, and Volkan Cevher. Adversarially robust optimization with gaussian processes. *Advances in neural information processing systems*, 31, 2018.

[19] Robert B Gramacy. lagp: large-scale spatial modeling via local approximate gaussian processes in r. *Journal of Statistical Software*, 72:1–46, 2016.

[20] Michael A Gelbart, Jasper Snoek, and Ryan P Adams. Bayesian optimization with unknown constraints. *arXiv preprint arXiv:1403.5607*, 2014.

[21] Joshua Knowles. Parego: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE transactions on evolutionary computation*, 10(1):50–66, 2006.

[22] Wolfgang Ponweiser, Tobias Wagner, Dirk Biermann, and Markus Vincze. Multiobjective optimization on a limited budget of evaluations using model-assisted-metric selection. In *International conference on parallel problem solving from nature*, pages 784–794. Springer, 2008.

[23] Daniel Hernández-Lobato, Jose Hernandez-Lobato, Amar Shah, and Ryan Adams. Predictive entropy search for multi-objective bayesian optimization. In *International conference on machine learning*, pages 1492–1501. PMLR, 2016.

[24] Matthias Feurer, Katharina Eggensperger, Stefan Falkner, Marius Lindauer, and Frank Hutter. Practical automated machine learning for the automl challenge 2018. In *International workshop on automatic machine learning at ICML*, pages 1189–1232, 2018.

[25] Martin Wistuba, Nicolas Schilling, and Lars Schmidt-Thieme. Hyperparameter search space pruning–a new component for sequential model-based hyperparameter optimization. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2015, Porto, Portugal, September 7-11, 2015, Proceedings, Part II 15*, pages 104–119. Springer, 2015.

[26] Marc C Kennedy and Anthony O'Hagan. Predicting the output from a complex computer code when fast approximations are available. *Biometrika*, 87(1):1–13, 2000.

[27] Kirthevasan Kandasamy, Gautam Dasarathy, Jeff Schneider, and Barnabás Póczos. Multi-fidelity bayesian optimisation with continuous approximations. In *International conference on machine learning*, pages 1799–1808. PMLR, 2017.

[28] David Ginsbourger, Janis Janusevskis, and Rodolphe Le Riche. *Dealing with asynchronicity in parallel Gaussian process based global optimization*. PhD thesis, Mines Saint-Etienne, 2011.

[29] Thomas Desautels, Andreas Krause, and Joel W Burdick. Parallelizing exploration-exploitation tradeoffs in gaussian process bandit optimization. *J. Mach. Learn. Res.*, 15(1):3873–3923, 2014.

[30] Kirthevasan Kandasamy, Akshay Krishnamurthy, Jeff Schneider, and Barnabás Póczos. Parallelised bayesian optimisation via thompson sampling. In *International conference on artificial intelligence and statistics*, pages 133–142. PMLR, 2018.

[31] Xubo Yue and Raed Al Kontar. Why non-myopic bayesian optimization is promising and how far should we lookahead. *A study via rollout. arXiv*, 2019.

[32] Shangda Yang, Vitaly Zankin, Maximilian Balandat, Stefan Scherer, Kevin Carlberg, Neil Walton, and Kody JH Law. Accelerating look-ahead in bayesian optimization: Multilevel monte carlo is all you need. *arXiv preprint arXiv:2402.02111*, 2024.

[33] Jose Pablo Folch, Calvin Tsay, Robert Lee, Behrang Shafei, Weronika Ormaniec, Andreas Krause, Mark van der Wilk, Ruth Misener, and Mojmír Mutny. Transition constrained bayesian optimization via markov decision processes. *Advances in Neural Information Processing Systems*, 37:88194–88235, 2024.

[34] Javier González, Michael Osborne, and Neil Lawrence. Glasses: Relieving the myopia of bayesian optimisation. In *Artificial Intelligence and Statistics*, pages 790–799. PMLR, 2016.

[35] Peter I Frazier. A tutorial on bayesian optimization. *arXiv preprint arXiv:1807.02811*, 2018.

[36] David JC MacKay et al. Introduction to gaussian processes. *NATO ASI series F computer and systems sciences*, 168:133–166, 1998.

[37] Tetiana Parshakova, Jean-Marc Andreoli, and Marc Dymetman. Distributional reinforcement learning for energy-based sequential models. *arXiv preprint arXiv:1912.08517*, 2019.

[38] Yang Song and Diederik P Kingma. How to train your energy-based models. *arXiv preprint arXiv:2101.03288*, 2021.

[39] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[40] Miguel González-Duque, Richard Michael, Simon Bartels, Yevgen Zainchkovskyy, Søren Hauberg, and Wouter Boomsma. A survey and benchmark of high-dimensional bayesian optimization of discrete sequences. *arXiv preprint arXiv:2406.04739*, 2024.

[41] Mitchell McIntire, Daniel Ratner, and Stefano Ermon. Sparse gaussian processes for bayesian optimization. In *UAI*, volume 3, page 4, 2016.

[42] Andrew Gordon Wilson, Zhiting Hu, Ruslan Salakhutdinov, and Eric P Xing. Deep kernel learning. In *Artificial intelligence and statistics*, pages 370–378. PMLR, 2016.

[43] Erik Nijkamp, Mitch Hill, Tian Han, Song-Chun Zhu, and Ying Nian Wu. On the anatomy of mcmc-based maximum likelihood learning of energy-based models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 5272–5280, 2020.

[44] Bo Pang, Tian Han, Erik Nijkamp, Song-Chun Zhu, and Ying Nian Wu. Learning latent space energy-based prior model. *Advances in Neural Information Processing Systems*, 33:21994–22008, 2020.

[45] H. Robbins and S. Monro. A stochastic approximation method. *Annals of Mathematical Statistics*, 1951.