Scalable Bayesian Optimization via Focalized Sparse Gaussian Processes

Yunyue Wei[1], **Vincent Zhuang**[2], **Saraswati Soedarmadji**[1], **Yanan Sui**[1]
[1] Tsinghua University
[2] Google DeepMind
weiyy20@mails.tsinghua.edu.cn
vincentzhuang@google.com
chenxuying24@mails.tsinghua.edu.cn
ysui@tsinghua.edu.cn

Abstract

Bayesian optimization is an effective technique for black-box optimization, but its applicability is typically limited to low-dimensional and small-budget problems due to the cubic complexity of computing the Gaussian process (GP) surrogate. While various approximate GP models have been employed to scale Bayesian optimization to larger sample sizes, most suffer from overly-smooth estimation and focus primarily on problems that allow for large online samples. In this work, we argue that Bayesian optimization algorithms with sparse GPs can more efficiently allocate their representational power to relevant regions of the search space. To achieve this, we propose focalized GP, which leverages a novel variational loss function to achieve stronger local prediction, as well as FocalBO, which hierarchically optimizes the focalized GP acquisition function over progressively smaller search spaces. Experimental results demonstrate that FocalBO can efficiently leverage large amounts of offline and online data to achieve state-of-the-art performance on robot morphology design and to control a 585-dimensional musculoskeletal system.

1 Introduction

Bayesian Optimization (BO) is a powerful approach for solving black-box optimization problems, demonstrating notable success in hyperparameter tuning[1], reinforcement learning[2, 3], and scientific discovery[4]. The efficacy of BO is attributed to its ability to model the unknown objective function using a surrogate model and to strategically select the next sample position by optimizing an acquisition function. Among the surrogate models, Gaussian Processes (GPs)[5] are usually favored due to their flexibility and robust uncertainty quantification. However, the computation of the posterior GP covariance matrix scales as $\mathcal{O}(n^3)$ with the number of data points n, which can severely restrict the applicability of BO in handling large datasets. This poses a significant challenge for real-world applications with high-dimensional and heterogeneous function landscapes such as those in robot control, which often necessitate a substantial amount of data to adequately explore the vast search space. To extend the scope of BO to accommodate larger datasets (from long-horizon online trials and/or pre-collected offline datasets[6]), it is imperative to employ surrogate models that offer enhanced computational efficiency.

Using sparse GP models is a popular method for reducing the computational cost of BO. Sparse GPs accomplish this by learning an approximation of the full GP, either by using a

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

subset of data[7], ensemble of local models[8], or variational inference[9]. However, classical sparse GP models are typically tailored for regression tasks, and therefore are designed to fit to the entire function landscape. Given limited representational resources, the resulting posterior is likely to be overly smooth, which may negatively impact the performance of BO. This issue is exacerbated in the high-dimensional setting, in which accurately fitting the entire domain is a far more challenging task. As such, several works have proposed strategies to improve BO performance with sparse GP models by focusing promising regions[10, 11] or advanced sparse GP models[12]. However, most of their empirical evaluations are only conducted under large online sample setting in low-dimensional problems with fewer than 20 variables. It is unclear whether existing methods can be generalized to large offline data or high-dimensional setting.

In this work, we explore the application of sparse Gaussian processes for optimizing high-dimensional problems with large offline (and optionally large online) datasets. We argue that by iteratively identifying key sub-regions of the input space and focusing the modeling capacity on these areas, we can enhance the modeling fidelity of the sparse GP in regions that are most relevant, thereby improving the overall performance of the Bayesian optimization algorithm. To this end, we propose a novel loss function to train a variational sparse GP model (focalized GP) that emphasizes the fitting of local functional landscapes through weighting the training data. Along with focalized GP, we design a hierachical algorithm, `FocalBO`, to propose sample points via acquisition function optimization across varying scales of the search space. Experimental results demonstrate that `FocalBO` can improve upon commonly used acquisition functions in optimizing heterogeneous functions and can effectively utilize large offline datasets for efficient high-dimensional optimization. Furthermore, we showcase that `FocalBO` can efficiently optimize a policy with 585 parameters to control a musculoskeletal system, leveraging both offline and online data. To the best of our knowledge, `FocalBO` is the first sparse GP-based Bayesian optimization algorithm capable of efficiently optimizing high-dimensional problems under both large online sample and large offline data settings.

Our main contributions: 1) We design `FocalBO`, which employs a hierarchical acquisition optimization strategy to achieve efficient optimization over high-dimensional problems with heterogeneous structure with limited representation capability. 2) Experimental results demonstrate the superior performance of `FocalBO` in leveraging large offline datasets for online optimization, and its capability to optimize high-dimensional musculoskeletal system control problems involving over 500 variables.

2 Related Work

2.1 Sparse Gaussian processes

Scaling Gaussian processes to large datasets is an important topic [13]. It can be broadly divided into global approximation strategies and local approximation strategies.

Global sparse GPs perform distillation over the whole dataset to approximate the expensive full covariance matrix with a sparse representation. Several methods aim to choose a subset of representative training points from the whole dataset, and use the corresponding covariance matrix in place of the full covariance[14, 7, 15, 16]. Sparse kernels aim at removing uncorrelated entries in the full covariance to obtain a compact matrix[17, 18, 19, 20]. Sparse approximation methods use inducing variables to learn a low-rank representation of full covariance matrix[21, 22, 23, 24, 8, 9, 25, 26, 27]. Stochastic variational Gaussian process (SVGP) is a popular sparse GP method which employs variational inference to learn inducing variables and kernel hyperparameters jointly and enable training using stochastic gradient descent from mini-batch data[25]. Recently, nearest neighbor information has also been used to further improve the scalability of sparse GP over massive amount of data[28, 29].

In contrast, local sparse GPs divide the entire dataset and employ local GPs trained from different data subsets to approximate the full GP. For a given test set, the prediction can be extracted from one of the local GPs[30, 31], mixture of GPs[32, 33] or product of GPs[34, 35].

2.2 Scalable Bayesian optimization

Recent works have proposed modifications to sparse GPs for Bayesian optimization. Sparse GP has been used to determine the search region where local GPs are used to determine the next samples[36]. Weighted-update online Gaussian processes (WOGP) was developed to select a subset of training points to approximate high performing regions of the input space[10]. IMP-DPP is motivated by a similar observation and uses a weighted Determinantal Point Process to select training points as inducing variables for the SVGP[11]. However, their proposed selection strategies require sequentially evaluating every training point, which can be computationally very expensive with large offline datasets. Combining SVGP with Thompson sampling has the same order of regret as standard Thompson sampling method[37]. Online variational conditioning (OVC) was proposed to efficiently conditioning SVGPs in an online setting, enabling using look-ahead acquisition functions[38]. Vecchia approximation of GP was also applied[39] for Bayesian optimization, with improved performance compared to prior works[12]. A concurrent work [40] aims at improving the acquisition optimization performance based on target-aware Bayesian inference [41].

Besides sparse GPs, Neural network[42, 43] and random forest[44] can also be used as BO surrogate model to circumvent the cubic complexity of GP. Ensemble Bayesian optimization utilizes the addictive function structure and uses ensembles of addictive GPs in parallel to achieve scalability[45]. Trust Region Bayesian optimization (TuRBO) and its variants uses exact GP to optimize over local regions, and employs a restart mechanism to achieve large number of evaluation, which is a representative line of works in high-dimensional Bayesian optimization[46, 47, 48]. TuRBO can also be combined with sparse GP models to further enhance the scalability[49, 50].

3 Background

3.1 Bayesian optimization

For an unknown objective function f, Bayesian optimization aims to solve $\max_{\boldsymbol{x}\in\mathcal{X}} f(\boldsymbol{x})$ over input space $\mathcal{X} \in [0,1]^d$. BO mainly consists of two components: a surrogate model to approximate the objective function, and an acquisition function a to decide the next sample position based on surrogate model.

Gaussian process is a commonly used surrogate model. Consider a given dataset $D = (\boldsymbol{X}, \boldsymbol{y})$ where $\boldsymbol{X} = (\boldsymbol{x}_1, ..., \boldsymbol{x}_t)$ are input locations and $\boldsymbol{y} = (y_1, ..., y_t)$ are associated noisy observations of $f(\boldsymbol{X})$. We assume the observation noise to be independent Gaussian, i.e. $y_i = f(\boldsymbol{x}_i) + \eta, \eta \sim \mathcal{N}(0, \sigma^2)$. Using GP with kernel function K, the function distribution $\boldsymbol{f}_*$ at test positions $\boldsymbol{X}_* = (\boldsymbol{x}_{*,1}, \ldots, \boldsymbol{x}_{*,t_*})^T$ is a multivariate Gaussian:

$$\begin{aligned} p(\boldsymbol{f}_* \mid \boldsymbol{X}, \boldsymbol{y}) = \mathcal{N}(\boldsymbol{f}_* \mid & K_{\boldsymbol{X}_*\boldsymbol{X}}[K_{\boldsymbol{X}\boldsymbol{X}} + \sigma I]^{-1}\boldsymbol{y}, \\ & K_{\boldsymbol{X}_*\boldsymbol{X}_*} - K_{\boldsymbol{X}_*\boldsymbol{X}}[K_{\boldsymbol{X}\boldsymbol{X}} + \sigma I]^{-1}K_{\boldsymbol{X}\boldsymbol{X}_*}), \end{aligned} \tag{1}$$

where K is the covariance matrix between subscript inputs. With the posterior distribution given D, the next sample point is the maximum position of the acquisition function: $\boldsymbol{x}_{t+1} = \max_{\boldsymbol{x}\in\mathcal{X}} a(\boldsymbol{x}|\mathcal{M}_t)$, where $\mathcal{M}_t$ is the GP model fitted on dataset collected at time step t. Common-used choice of a includes upper confidence bound(UCB, [51]), expected improvement (EI, [52]) and Thompson sampling (TS, [53]). The inner optimization problem is usually solved by grid search, evolutionary algorithms[54], or gradient-based methods[55]. When the online sample budget is large, batch optimization is commonly used to evaluate multiple inputs in parallel[56].

3.2 Variational Gaussian process

For predictive distribution conditioned on given dataset of size t, the computational complexity of exact Gaussian process is $\mathcal{O}(t^3)$ for each test position due to the inverse of the covariance matrix $K_{\boldsymbol{X}\boldsymbol{X}}$, which is expensive for large scale datasets with more than a few thousand points. A common used strategy is to approximate full GP regression using sparse GPs. In sparse GP, $m \ll t$ inducing variables $\boldsymbol{u} = (u_1, \ldots, u_m)^T$ characterized by inducing inputs

$\boldsymbol{Z} = (\boldsymbol{z}_1, \ldots, \boldsymbol{z}_m)$ are introduced to approximate the covariance matrix of the full GP. In this section, we focus on sparse GP derived from variational inference.

Variational GP [9] considers the joint latent prior

$$p(\boldsymbol{f}, \boldsymbol{u}) = (\begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{u} \end{bmatrix} \mid 0, \begin{bmatrix} K_{\boldsymbol{XX}} & K_{\boldsymbol{XZ}} \\ K_{\boldsymbol{ZX}} & K_{\boldsymbol{ZZ}} \end{bmatrix}), \tag{2}$$

where $\boldsymbol{f} = (f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_t))^T$. A variational distribution $q(\boldsymbol{u}) = \mathcal{N}(\boldsymbol{u} \mid \boldsymbol{m}, \boldsymbol{S})$ is used to approximate the posterior over inducing variables using the exact conditional distribution of $\boldsymbol{f}$ given $\boldsymbol{u}$, that is, $q(\boldsymbol{f}, \boldsymbol{u}) = p(\boldsymbol{f} \mid \boldsymbol{u})q(\boldsymbol{u})$. The posterior of $\boldsymbol{f}$ can be computed by marginalizing $\boldsymbol{u}$ with analytic form:

$$q(\boldsymbol{f}) = \int p(\boldsymbol{f} \mid \boldsymbol{u})q(\boldsymbol{u})d\boldsymbol{u} = \mathcal{N}(\boldsymbol{f} \mid \boldsymbol{Am}, K_{\boldsymbol{XX}} - \boldsymbol{A}^T(K_{\boldsymbol{ZZ}} - \boldsymbol{S})\boldsymbol{A}), \tag{3}$$

where $\boldsymbol{A} = K_{\boldsymbol{ZZ}}^{-1} K_{\boldsymbol{ZX}}$. The variational parameters $\boldsymbol{Z}, \boldsymbol{m}, \boldsymbol{S}$ are optimized by maximizing the Evidence Lower Bound (ELBO) which can be written in the following formulation[25]:

$$\mathcal{L}_1 = \sum_{i=1}^{t} \mathbb{E}_{q(f(\boldsymbol{x}_i))}[\log p(y_i \mid f(\boldsymbol{x}_i))] - \mathrm{KL}[q(\boldsymbol{u}) \parallel p(\boldsymbol{u})] = \mathcal{L}_{\mathrm{LL}} + \mathcal{L}_{\mathrm{KL}} \tag{4}$$

where $\mathrm{KL}[\cdot \parallel \cdot]$ is the KL divergence between two distributions. The ELBO breaks into a data likelihood term which factorized over training data, and a KL divergence term which can be computed in closed form. The factorization over data allows optimization via stochastic gradient descent (SGD), reducing the computational complexity to $\mathcal{O}(m^3)$.

4 Focalized Gaussian Process for Bayesian Optimization

Prior studies about variational sparse GPs are mainly designed for regression tasks, where the goal is to fit global training data distribution. In Bayesian optimization, the next sample is determined by the predictive function distribution over test positions. Gradient-based and evolutionary-based acquisition function optimization methods employ local search from random starting points to find a local optimal of the acquisition function. Recent works also scale grid search-based optimization to high dimensional space by restricting the search space within local sub-regions[46, 47]. All the above procedure would benefit from an accurate estimation over sub-region of the input space. Therefore, a sensible way to improve BO performance is to allocate limited computational resources to obtain better prediction over specific search regions instead of the entire input domain.

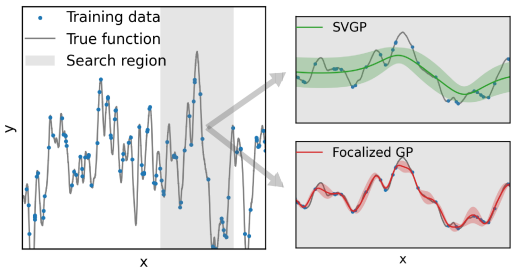

Figure 1: Performance comparison of focalized GP and SVGP over 1d GP functions. Posteriors are shown as mean ± 1 standard deviation.

We define the search region as the region where the acquisition function is optimized on, which is an axis-aligned hypercube with length $\boldsymbol{l} = (l_1, \cdots, l_d)^T$ centered at $\boldsymbol{c}$:

$$\mathcal{S}_{\boldsymbol{c},\boldsymbol{l}} = \{\boldsymbol{x} \mid \boldsymbol{c} - \frac{1}{2}\boldsymbol{l} \leq \boldsymbol{x} \leq \boldsymbol{c} + \frac{1}{2}\boldsymbol{l}\}. \tag{5}$$

When $\boldsymbol{l} = (1, \cdots, 1)^T$ and $\boldsymbol{c} = (0.5, \cdots, 0.5)^T$, the acquisition optimization is performed over the entire input space $\mathcal{X}$, as commonly-used in vanilla BO algorithms. In the rest of this section, we first present the derivation of focalized loss function to improve GP prediction over the search region. Then we demonstrate how to incorporate our proposed GP model into Bayesian optimization.

4.1 Focalized evidence lower bound

We recall eq.1 and rewrite the mean estimation $\mu_t(\boldsymbol{x}_*)$ and variance estimation $\sigma_t(\boldsymbol{x}_*)$ for each test position $\boldsymbol{x}_*$:

$$\begin{aligned}\mu_t(\boldsymbol{x}_*) &= \sum_{i=1}^{t} k(\boldsymbol{x}_*, \boldsymbol{x}_i)[K_{\boldsymbol{XX}} + \sigma I]^{-1} y_i, \\ \sigma_t(\boldsymbol{x}_*) &= k(\boldsymbol{x}_*, \boldsymbol{x}_*) - \sum_{i=1}^{t}\sum_{j=1}^{t} k(\boldsymbol{x}_*, \boldsymbol{x}_i)\bar{k}_{ij}k(\boldsymbol{x}_*, \boldsymbol{x}_j),\end{aligned} \tag{6}$$

where $\bar{k}_{ij}$ is the (i, j)-th entry of $[K_{\boldsymbol{XX}} + \sigma I]^{-1}$. From eq. 6 we can observe that the mean estimation at $\boldsymbol{x}_*$ is a linear combination of observation $\boldsymbol{y}$ multiplied by $k(\boldsymbol{x_*}, \boldsymbol{x})$, and the reduction of variance is a quadratic form of the covariance between $\boldsymbol{x}_*$ and training points. Both estimation can be written as linear summations of constant values with kernel function as weight. As mentioned in prior works[57], data points far from the test positions have a vanishingly small influence on the predictive distribution with commonly used kernel functions. Utilizing this observation, we propose to weight the data likelihood term using the kernel function to focus training over points that contribute to the prediction of the search region:

$$\mathcal{L}_{\text{WLL}} = \sum_{i=1}^{t} w_i \mathbb{E}_{q(f(\boldsymbol{x}_i))}[\log p(y_i \mid f(\boldsymbol{x}_i))], \quad w_i = \max_{\boldsymbol{x}_* \in \mathcal{S}_{\boldsymbol{c},\boldsymbol{l}}} k(\boldsymbol{x}_i, \boldsymbol{x}_*). \tag{7}$$

We use the maximum covariance of $\boldsymbol{x}_i$ to positions in the search region as the corresponding weight to filter out points that have marginally influence to the search region during GP training. In this way, the model can selectively utilize the training data to achieve good local prediction. When using a popular kernel functions such as RBF or Matern kernel, the maximum kernel value is equivalent to finding the nearest point in the search region, which can be easily calculated when the region boundary is axis-aligned as defined in eq. 5.

We additionally regularize the sum of weights to make the model focus on improving prediction over search region:

$$\mathcal{L}_{\text{reg}} = \frac{|\boldsymbol{X} \notin \mathcal{S}_{\boldsymbol{c},\boldsymbol{l}}|}{|\boldsymbol{X} \in \mathcal{S}_{\boldsymbol{c},\boldsymbol{l}}|} = \left(\frac{\sum_{i=1}^{t} w_i}{|\boldsymbol{X} \in \mathcal{S}_{\boldsymbol{c},\boldsymbol{l}}|} - 1\right), \tag{8}$$

where $|\boldsymbol{X} \in \mathcal{S}_{\boldsymbol{c},\boldsymbol{l}}| = \sum_{i=1}^{t} \mathbb{1}_{\boldsymbol{x}_i \in \mathcal{S}_{\boldsymbol{c},\boldsymbol{l}}}$ is the number of training points in the search region. The proposed regularization term $\mathcal{L}_{\text{reg}}$ encourages accurate local prediction instead of blurred global estimation, avoiding getting stuck on suboptimal of large kernel lengthscale. Combined with KL loss, our finalized new ELBO is as follows:

$$\mathcal{L}_2 = \mathcal{L}_{\text{WLL}} + \mathcal{L}_{\text{KL}} - \mathcal{L}_{\text{reg}}. \tag{9}$$

Compared to the original ELBO loss in SVGP, our proposed function maintains the same computational complexity and does not introduce additional hyperparameters. Our ELBO also reproduces eq. 4 when considering to predict the entire input space $\mathcal{X}$. During the model training, both GP hyperparameters and variational parameters are jointly optimized to obtain focalized GP for Bayesian optimization. Figure 1 shows a comparison of focalized GP and SVGP over 1d functions sampled from GP. While SVGP can only able to vaguely predict the function, focalized GP accurately delineate the function landscape within search region by training with the focalized loss. Our proposed GP model is sensitive to high-performing positions within the search space which contribute to better acquisition optimization. We also systematically compare the GP prediction performance in Appendix B.3, where our GP model trained from focalized ELBO consistently achieves good prediction on small size of search space.

Theoretical implications of focalized ELBO. Our focalized ELBO can be interpreted as a soft variant of training a local approximation over datapoints that lie within the search region. Here, we illustrate how local approximations can substantially reduce the KL divergence of the approximate posterior over the search region, and discuss the effects of tighter

approximations on BO regret bounds. We focus on providing general theoretical intuition rather than deriving precise bounds due to the lack of existing convergence guarantees for ELBO maximization in the general setting.

Suppose that we know the optimal point lies in some small sub-region of $\boldsymbol{X}$ that contains $N' << N$ training points. Corollary 19 in [58] shows that given a squared exponential kernel and some assumptions on the inducing point selection, for a fixed number of inducing points the KL-divergence upper bound scales super-quadratically in the number of training points. Hence, fitting locally can yield much tighter approximations than fitting globally (e.g. SVGP).

Next, we consider the impact of the KL approximation error on the optimization regret. Proposition 1 in [58] states that the gap between the means of the approximate and exact posteriors is upper bounded by $\mathcal{O}(\sigma\sqrt{\gamma})$, where γ is an upper-bound on the approximation KL-divergence. This has an immediate impact on the regret - for example, when GP-UCB [51] is combined with sparse GPs, the confidence bounds must be enlargened by an additive $\sqrt{\gamma}$ factor to account for the approximation error. Because the regret bound scales with $\sqrt{\beta_T}$ where β_T is the maximum confidence interval coefficient, having a large approximation error can arbitrarily scale the regret incurred by the algorithm. In order to achieve no additional regret order, the additional approximation error noise must be uniformly bounded (Assumption 4 in [37]). Although focalized GP cannot guarantee a constant bound, it still directly reduces the regret of the algorithm, where we empirically investigate in Appendix B.1.

4.2 Bayesian optimization with focalized GP

One advantage of focalized GP is that it can be easily integrated into existing BO algorithms. To further leverage the strong local modeling properties of focalized GP, we design FocalBO, a hierachical acquisition optimization framework described in Algorithm 1.

Algorithm 1 FocalBO

Input Initial Dataset $\mathcal{D}_0$, Inducing Variable Size m, Batch Size B

1: $H \leftarrow 1$
2: **for** $t = 1, 2, \cdots$ **do**
3: $\quad \{\boldsymbol{x}_{t,i}\}_{i=1}^{B}, \{h_{t,i}\}_{i=1}^{B} \leftarrow$ FocalAcq$(\mathcal{D}_{t-1}, H, m, B)$
4: $\quad$ Observe $\{y_{t,i}\}_{i=1}^{B} = \{f(\boldsymbol{x}_{t,i}) + \eta\}_{i=1}^{B}$
5: $\quad \mathcal{D}_{t+1} \leftarrow \mathcal{D}_t \cup \{(\boldsymbol{x}_{t,i}, y_{t,i})\}_{i=1}^{B}$
6: $\quad i_{\text{best}} \leftarrow \text{argmax}_{i \in 1, \cdots, B} y_i$
7: $\quad$ **if** $h_{i_{\text{best}}} < H$ **then**
8: $\quad\quad H \leftarrow H - 1$
9: $\quad$ **else**
$\quad\quad H \leftarrow H + 1$
10: $\quad$ **end if**
11: **end for**

At each BO iteration, FocalBO iteratively optimizes the acquisition function over a progressively smaller search region via focalized acquisition function (FocalAcq) as shown in Algorithm 2. The first depth of acquisition optimization starts with the entire input space $\mathcal{X}$ with $\boldsymbol{l} = (1, \cdots, 1)^T$ and $\boldsymbol{c} = (0.5, \cdots, 0.5)^T$ (line 1). We train specific focalized GP base on the search region at each round of acquisition optimization (line 4-5). Our framework is compatible with any acquisition function that extracts instant posterior information from the GP and is optimized within pre-defined search region. After one round of acquisition function optimization, the search space length $\boldsymbol{l}$ is halved to focus on a smaller search region centered at current best position $\boldsymbol{x}_{\text{best}}$(line 6-7). In this way we can obtain a more accurate model for decision making, and also relieve the over-exploration problem when the problem dimension is high[59]. One batch of inputs is proposed at each round of optimization, and the final decision is sampled from all proposed inputs via Softmax distribution over their corresponding acquisition function values (line 9). Our hierarchical optimization strategy enables collecting candidates from both global sparse estimation and local focalized prediction, achieving balance between exploration and exploitation with constrained computation power.

The optimization depth H in FocalAcq controls the degree of utilizing local information from current best position, where the GP estimate variance decreases with the shrinkage of search space. The best-performing optimization depth is likely problem-dependent (e.g. high-dimensional functions may require higher optimization depths). Therefore in FocalBO, we

propose to automatically adjust the optimization depth according to the instant optimization performance. At the beginning of the optimization, we initialize the optimization depth as 1, indicating global search of the input space (Algorithm 1, line 1). Then we keep track of the depth where the proposed positions are sampled from. If the depth of the best point in this round is less than the current optimization depth H, we reduce H to encourage exploration of the input space, otherwise we increase H for better exploitation of $\boldsymbol{x}_{\text{best}}$ (line 6-10).

Our proposed framework is orthogonal to TuRBO-M [46], but bears some similarities in searching over multiple sub-regions and adaptively adjusting the search region. Our algorithm differs in that TuRBO-M constructs equal-sized trust regions and fits independent Exact GP using separated dataset, aiming at searching for different local optima in the search space. By contrast, the search region in FocalBO is constructed with different sizes to make decision based on both global and local information. Our framework allows data sharing across search regions, and the use of focalized GP helps to accurately estimate local region with limited representation. Additionally, FocalBO does not introduce extra hyperparameters. Finally, we demonstrate in Section 5 that TuRBO is complementary to FocalBO in optimizing high-dimensional problems.

Algorithm 2 FocalAcq

Input Dataset $\mathcal{D}_{t-1}$, Optimization Depth H, Inducing Variable Size m, Batch Size B

1: $\boldsymbol{l} \leftarrow (1, \cdots, 1)^T, \boldsymbol{c} \leftarrow (0.5, \cdots, 0.5)^T$
2: Select current best point $\boldsymbol{x}_{\text{best}}$ from $\mathcal{D}_{t-1}$
3: **for** $h = 1, \cdots, H$ **do**
4: Train $\mathcal{M}_t^h$ using $\mathcal{L}_2$ given $\mathcal{S}_{\boldsymbol{c},\boldsymbol{l}}$
5: $\{\boldsymbol{x}_{t,i}^h\}_{i=1}^B \leftarrow \text{argmax}_{\boldsymbol{x}\in\mathcal{X}_*} a(\boldsymbol{x}|\mathcal{M}_t^h)$
6: $\boldsymbol{l} \leftarrow \boldsymbol{l}/2$
7: $\boldsymbol{c} \leftarrow \boldsymbol{x}_{\text{best}}$
8: **end for**
9: **return** $\{\boldsymbol{x}_{t,i}\}_{i=1}^B, \{h_{t,i}\}_{i=1}^B \sim P(i = i') \propto \frac{\exp^{a(\boldsymbol{x}_{t,i'}^{h'}|\mathcal{M}_t^{h'})}}{\sum_{h=1}^H \sum_{j=1}^B \exp^{a(\boldsymbol{x}_{t,j}^h|\mathcal{M}_t^h)}}$

5 Experiments

In this section, we extensively evaluate FocalBO over a variety of tasks. We first use synthetic functions to showcase the compatibility of FocalBO in improving commonly-used acquisition functions. Next, we consider the online optimization of robot morphology design that is additionally given a large offline dataset. We also show that FocalBO is able to optimize very high-dimensional musculoskeletal system control with both a large offline dataset and a large number of online budget. Finally we dig deeper into FocalBO to analyze how each of its components contributes to superior optimization performance.

We compare FocalBO with representative sparse GP models used for Bayesian optimization, including SVGP[25], WOGP[10], and Vecchia GP[12]. We only run WOGP on synthetic functions due to its extremely low speed in dealing with the datasets in the remaining tasks. The number of inducing variables in sparse GP models is set as 50 for synthetic functions and as 200 for other tasks. The optimization performances are shown as mean $\pm$ 1 standard error for all considered problems over 10 independent trials.

5.1 Synthetic functions

We select Shekel and Michalewicz as the test functions, which are heterogeneous with both smooth and rigid regions. We also sample functions directly from Gaussian processes to evaluate algorithm performance under full BO assumption. For each function, we choose to use different acquisition functions to optimize: TS optimized by grid search, EI optimized by analytic gradient, and probability of improvement (PI) optimized by Monte Carlo gradient[55]. Optimization performances are shown in Figure 2. We observe that FocalBO significantly improves the performance of all acquisition functions compared to SVGP, and is able to consistently achieve top-tier performance over all problems. In Michalewicz function where a large fraction of the input space is flat, all baselines tend to increase the noise estimation to maintain a stationary prediction, while focalized GP is able to focus on the local search region and successfully optimize the function. Additional experiment with online samples as major data source is shown in Appendix B.2, where FocalBO still maintains comparable or better performance against baselines.

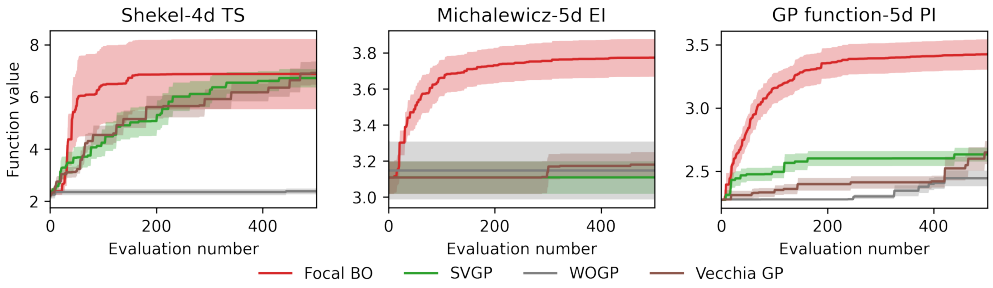

Figure 2: Optimization performance under different synthetic function and acquisition function. Sparse GP models are trained with 50 inducing variables. The offline dataset contains 2000 random data points and the online budget is 500 with batch size of 10.

5.2 Robot morphology design

We compare FocalBO to several baselines over robot morphology design task from Design-Bench, which provides large offline dataset with an exact function oracle[6]. The goal of the task is to optimize the morphological structure of D'Kitty robot[60] to improve the simulation performance under RL controller. While the benchmark is initially designed for offline model-based optimization (MBO), it can also be used as an offline-to-online BO benchmark. In this task, we use the training dataset with 10,000 points and additionally evaluate 128 points on-the-fly with batch size of 4. EI is used as the base acquisition function for better optimizing with small batch size. We also try to combine FocalBO with TuRBO to optimize over the high-dimensional space, with the results shown in Figure 3. We observe that FocalBO achieves significant improvement from the initial data while other baselines struggle to obtain performance gain, even combined with TuRBO. FocalBO with TuRBO effectively extracts information from large offline dataset and is the first GP-based method to achieve top-tier performance reported by prior MBO works[61].

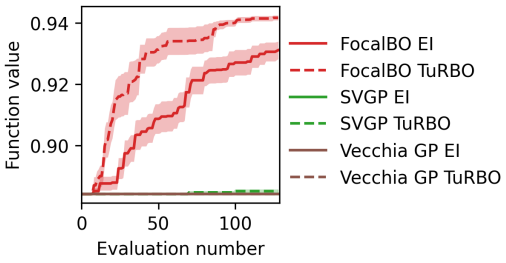

Figure 3: Optimization on robot morphology design. Function values are normalized by best and worst values in the unseen full dataset.

5.3 Human musculoskeletal system control

We further apply FocalBO to control a human arm musculoskeletal system[62] for the task of pouring liquid into a cup, as shown in Figure 4(a). To control the musculoskeletal system, we optimize a linear policy $\pi \in |A| \times |O|$, where $|A| = 5$ and $|O| = 117$ are the corresponding action and observation dimensions. The action dimension has been reduced from individual muscles to synergetic groups of muscles by applying principled component analysis to sampled action data from an RL agent (Appendix A.6). Although the original control dimension is reduced, the remaining 585-dimensional input space is still very high for existing high-dimensional BO algorithms. Therefore we consider a large offline-online setting, where we randomly sample 2000 points from the input space to serve as the offline dataset, and set the online budget as 3000 with batch size of 100. We use Thompson sampling as the base acquisition function. Figure 4(b) demonstrates that FocalBO outperforms other baselines, achieving higher maximum reward and faster convergence speed. Our supplementary video shows that the optimized policy is able to perform well on the task, demonstrating the successful application of FocalBO to high-dimensional control problems.

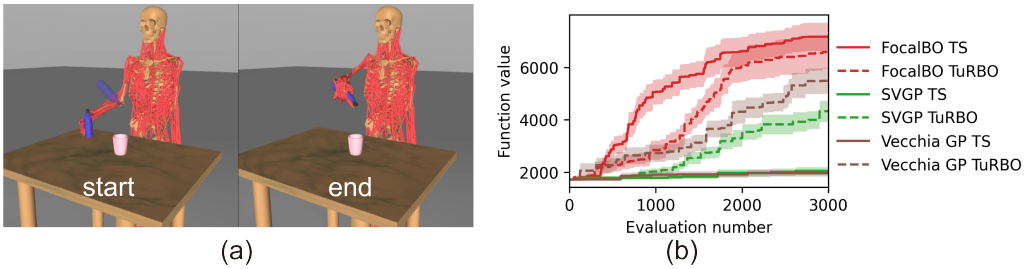

Figure 4: Optimization of musculoskeletal system control. (a) Task illustration of initial and target state. Full video in supplementary. (b) Optimization performance of algorithms.

5.4 Algorithm analysis

To understand the reasons behind FocalBO's superior optimization performance, we investigate the optimization depth in FocalBO, which is the central component of the method. Figure 5(a) shows the evolution of optimization depth over different problems, where FocalBO is able to adapt the optimization depth according to different function structure. For Shekel and musculoskeletal model control where the promising regions are distinct, the optimization exhibits an increasing trend to exploit current best points, while for other problems the depth tends to converge at a fixed level. Figure 5(b) shows the sources of proposed batches during the optimization of musculoskeletal system control. Overall the samples exhibits clear trend from exploration to exploitation over high-dimensional input space. Our hierarchical optimization strategy enables flexibility between exploration and exploitation.

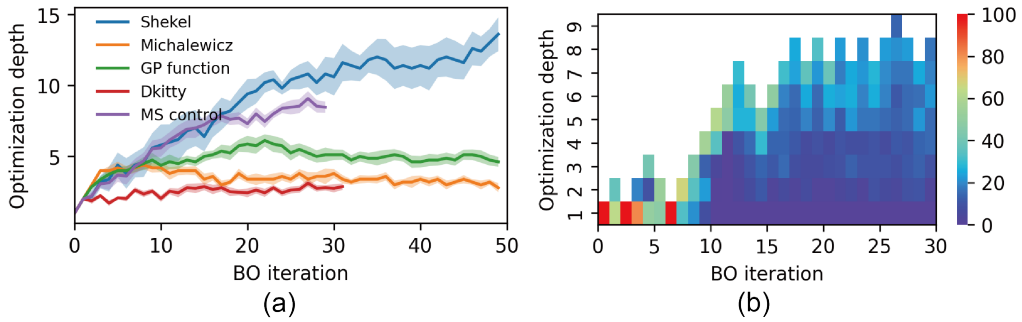

Figure 5: Algorithm analysis over optimization depth. (a) Depth evolution during optimization. (b) Samples source of each BO iteration during one trial of musculoskeletal system control optimization. Color bar indicates the number of samples proposed by corresponding optimization depth.

6 Conclusion

In this paper, we propose FocalBO, which uses a hierarchical acquisition optimization strategy equipped with focalized GP model to scale Bayesian optimization to problems with large offline datasets and/or a large number of online samples. Despite limited representation capability, FocalBO consistently improves various acquisition functions in optimizing heterogeneous functions, and adeptly leverages large offline dataset for efficient optimization over robot morphology. Under the large offline-to-online optimization setting, FocalBO achieves stable high-dimensional control of human musculoskeletal model with over 500 parameters. Ablation studies over the algorithm components further verify the principled design of FocalBO. Future work may include theoretically analyzing FocalBO, and applying the method to more complex problems, such as large-scale parameter tuning and whole-body human musculoskeletal system control.

References

[1] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. *Advances in neural information processing systems*, 25, 2012.

[2] Roberto Calandra, André Seyfarth, Jan Peters, and Marc Peter Deisenroth. Bayesian optimization for learning gaits under uncertainty: An experimental comparison on a dynamic bipedal walker. *Annals of Mathematics and Artificial Intelligence*, 76:5–23, 2016.

[3] Yutian Chen, Aja Huang, Ziyu Wang, Ioannis Antonoglou, Julian Schrittwieser, David Silver, and Nando de Freitas. Bayesian optimization in alphago. *arXiv preprint arXiv:1812.06855*, 2018.

[4] 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.

[5] Carl Edward Rasmussen. Gaussian processes in machine learning. In *Summer school on machine learning*, pages 63–71. Springer, 2003.

[6] Brandon Trabucco, Xinyang Geng, Aviral Kumar, and Sergey Levine. Design-bench: Benchmarks for data-driven offline model-based optimization. In *International Conference on Machine Learning*, pages 21658–21676. PMLR, 2022.

[7] Neil Lawrence, Matthias Seeger, and Ralf Herbrich. Fast sparse gaussian process methods: The informative vector machine. *Advances in neural information processing systems*, 15, 2002.

[8] Edward Snelson and Zoubin Ghahramani. Sparse gaussian processes using pseudo-inputs. *Advances in neural information processing systems*, 18, 2005.

[9] Michalis Titsias. Variational learning of inducing variables in sparse gaussian processes. In *Artificial intelligence and statistics*, pages 567–574. PMLR, 2009.

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

[11] Henry B Moss, Sebastian W Ober, and Victor Picheny. Inducing point allocation for sparse gaussian processes in high-throughput bayesian optimisation. In *International Conference on Artificial Intelligence and Statistics*, pages 5213–5230. PMLR, 2023.

[12] Felix Jimenez and Matthias Katzfuss. Scalable bayesian optimization using vecchia approximations of gaussian processes. In *International Conference on Artificial Intelligence and Statistics*, pages 1492–1512. PMLR, 2023.

[13] Haitao Liu, Yew-Soon Ong, Xiaobo Shen, and Jianfei Cai. When gaussian process meets big data: A review of scalable gps. *IEEE transactions on neural networks and learning systems*, 31(11):4405–4423, 2020.

[14] Kohei Hayashi, Masaaki Imaizumi, and Yuichi Yoshida. On random subsampling of gaussian process regression: A graphon-based analysis. In *International Conference on Artificial Intelligence and Statistics*, pages 2055–2065. PMLR, 2020.

[15] Matthias Seeger. Pac-bayesian generalisation error bounds for gaussian process classification. *Journal of machine learning research*, 3(Oct):233–269, 2002.

[16] Sathiya Keerthi and Wei Chu. A matching pursuit approach to sparse gaussian process regression. *Advances in neural information processing systems*, 18, 2005.

[17] Tilmann Gneiting. Compactly supported correlation functions. *Journal of Multivariate Analysis*, 83(2):493–508, 2002.

[18] Arman Melkumyan and Fabio Tozeto Ramos. A sparse covariance function for exact gaussian process inference in large datasets. In *Twenty-first international joint conference on artificial intelligence*, 2009.

[19] Martin Buhmann. A new class of radial basis functions with compact support. *Mathematics of Computation*, 70(233):307–318, 2001.

[20] Holger Wendland. *Scattered data approximation*, volume 17. Cambridge university press, 2004.

[21] Joaquin Quinonero-Candela and Carl Edward Rasmussen. A unifying view of sparse approximate gaussian process regression. *The Journal of Machine Learning Research*, 6:1939–1959, 2005.

[22] Alex Smola and Peter Bartlett. Sparse greedy gaussian process regression. *Advances in neural information processing systems*, 13, 2000.

[23] Matthias W Seeger, Christopher KI Williams, and Neil D Lawrence. Fast forward selection to speed up sparse gaussian process regression. In *International Workshop on Artificial Intelligence and Statistics*, pages 254–261. PMLR, 2003.

[24] Lehel Csató and Manfred Opper. Sparse on-line gaussian processes. *Neural computation*, 14(3):641–668, 2002.

[25] James Hensman, Nicolo Fusi, and Neil D Lawrence. Gaussian processes for big data. *arXiv preprint arXiv:1309.6835*, 2013.

[26] Lehel Csató and Manfred Opper. Sparse representation for gaussian process models. *Advances in neural information processing systems*, 13, 2000.

[27] Andrew Wilson and Hannes Nickisch. Kernel interpolation for scalable structured gaussian processes (kiss-gp). In *International conference on machine learning*, pages 1775–1784. PMLR, 2015.

[28] Luhuan Wu, Geoff Pleiss, and John P Cunningham. Variational nearest neighbor gaussian process. In *International Conference on Machine Learning*, pages 24114–24130. PMLR, 2022.

[29] Gia-Lac Tran, Dimitrios Milios, Pietro Michiardi, and Maurizio Filippone. Sparse within sparse gaussian processes using neighbor information. In *International Conference on Machine Learning*, pages 10369–10378. PMLR, 2021.

[30] Hyoung-Moon Kim, Bani K Mallick, and Chris C Holmes. Analyzing nonstationary spatial data using piecewise gaussian processes. *Journal of the American Statistical Association*, 100(470):653–668, 2005.

[31] Abhirup Datta, Sudipto Banerjee, Andrew O Finley, and Alan E Gelfand. On nearest-neighbor gaussian process models for massive spatial data. *Wiley Interdisciplinary Reviews: Computational Statistics*, 8(5):162–171, 2016.

[32] Seniha Esen Yuksel, Joseph N Wilson, and Paul D Gader. Twenty years of mixture of experts. *IEEE transactions on neural networks and learning systems*, 23(8):1177–1193, 2012.

[33] Saeed Masoudnia and Reza Ebrahimpour. Mixture of experts: a literature survey. *Artificial Intelligence Review*, 42:275–293, 2014.

[34] Geoffrey E Hinton. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800, 2002.

[35] Samuel Cohen, Rendani Mbuvha, Tshilidzi Marwala, and Marc Deisenroth. Healing products of gaussian process experts. In *International Conference on Machine Learning*, pages 2068–2077. PMLR, 2020.

[36] Tipaluck Krityakierne and David Ginsbourger. Global optimization with sparse and local gaussian process models. In *International Workshop on Machine Learning, Optimization and Big Data*, pages 185–196. Springer, 2015.

[37] Sattar Vakili, Henry Moss, Artem Artemev, Vincent Dutordoir, and Victor Picheny. Scalable thompson sampling using sparse gaussian process models. *Advances in neural information processing systems*, 34:5631–5643, 2021.

[38] Wesley J Maddox, Samuel Stanton, and Andrew G Wilson. Conditioning sparse variational gaussian processes for online decision-making. *Advances in Neural Information Processing Systems*, 34:6365–6379, 2021.

[39] Matthias Katzfuss, Joseph Guinness, Wenlong Gong, and Daniel Zilber. Vecchia approximations of gaussian-process predictions. *Journal of Agricultural, Biological and Environmental Statistics*, 25:383–414, 2020.

[40] Natalie Maus, Kyurae Kim, Geoff Pleiss, David Eriksson, John P Cunningham, and Jacob R Gardner. Approximation-aware bayesian optimization. *arXiv preprint arXiv:2406.04308*, 2024.

[41] Tom Rainforth, Adam Golinski, Frank Wood, and Sheheryar Zaidi. Target–aware bayesian inference: how to beat optimal conventional estimators. *Journal of Machine Learning Research*, 21(88):1–54, 2020.

[42] Jasper Snoek, Oren Rippel, Kevin Swersky, Ryan Kiros, Nadathur Satish, Narayanan Sundaram, Mostofa Patwary, Mr Prabhat, and Ryan Adams. Scalable bayesian optimization using deep neural networks. In *International conference on machine learning*, pages 2171–2180. PMLR, 2015.

[43] Zhongkai Shangguan, Lei Lin, Wencheng Wu, and Beilei Xu. Neural process for black-box model optimization under bayesian framework. *arXiv preprint arXiv:2104.02487*, 2021.

[44] Balaji Lakshminarayanan, Daniel M Roy, and Yee Whye Teh. Mondrian forests for large-scale regression when uncertainty matters. In *Artificial Intelligence and Statistics*, pages 1478–1487. PMLR, 2016.

[45] Zi Wang, Clement Gehring, Pushmeet Kohli, and Stefanie Jegelka. Batched large-scale bayesian optimization in high-dimensional spaces. In *International Conference on Artificial Intelligence and Statistics*, pages 745–754. PMLR, 2018.

[46] 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.

[47] Linnan Wang, Rodrigo Fonseca, and Yuandong Tian. Learning search space partition for black-box optimization using monte carlo tree search. *Advances in Neural Information Processing Systems*, 33:19511–19522, 2020.

[48] David Eriksson and Matthias Poloczek. Scalable constrained bayesian optimization. In *International Conference on Artificial Intelligence and Statistics*, pages 730–738. PMLR, 2021.

[49] Natalie Maus, Haydn Jones, Juston Moore, Matt J Kusner, John Bradshaw, and Jacob Gardner. Local latent space bayesian optimization over structured inputs. *Advances in Neural Information Processing Systems*, 35:34505–34518, 2022.

[50] Saulius Tautvaišas and Julius Žilinskas. Scalable bayesian optimization with generalized product of experts. *Journal of Global Optimization*, 88(3):777–802, 2024.

[51] Niranjan Srinivas, Andreas Krause, Sham M Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. *arXiv preprint arXiv:0912.3995*, 2009.

[52] Donald R Jones, Matthias Schonlau, and William J Welch. Efficient global optimization of expensive black-box functions. *Journal of Global optimization*, 13:455–492, 1998.

[53] 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.

[54] Nikolaus Hansen. The cma evolution strategy: a comparing review. *Towards a new evolutionary computation: Advances in the estimation of distribution algorithms*, pages 75–102, 2006.

[55] Maximilian Balandat, Brian Karrer, Daniel R. Jiang, Samuel Daulton, Benjamin Letham, Andrew Gordon Wilson, and Eytan Bakshy. BoTorch: A Framework for Efficient Monte-Carlo Bayesian Optimization. In *Advances in Neural Information Processing Systems 33*, 2020.

[56] Javier González, Zhenwen Dai, Philipp Hennig, and Neil Lawrence. Batch bayesian optimization via local penalization. In *Artificial intelligence and statistics*, pages 648–657. PMLR, 2016.

[57] Robert B Gramacy and Daniel W Apley. Local gaussian process approximation for large computer experiments. *Journal of Computational and Graphical Statistics*, 24(2):561–578, 2015.

[58] David R Burt, Carl Edward Rasmussen, and Mark Van Der Wilk. Convergence of sparse variational inference in gaussian processes regression. *Journal of Machine Learning Research*, 21(131):1–63, 2020.

[59] ChangYong Oh, Efstratios Gavves, and Max Welling. Bock: Bayesian optimization with cylindrical kernels. In *International Conference on Machine Learning*, pages 3868–3877. PMLR, 2018.

[60] Michael Ahn, Henry Zhu, Kristian Hartikainen, Hugo Ponte, Abhishek Gupta, Sergey Levine, and Vikash Kumar. Robel: Robotics benchmarks for learning with low-cost robots. In *Conference on robot learning*, pages 1300–1313. PMLR, 2020.

[61] Brandon Trabucco, Aviral Kumar, Xinyang Geng, and Sergey Levine. Conservative objective models for effective offline model-based optimization. In *International Conference on Machine Learning*, pages 10358–10368. PMLR, 2021.

[62] Kaibo He, Chenhui Zuo, Jing Shao, and Yanan Sui. Self model for embodied intelligence: Modeling full-body human musculoskeletal system and locomotion control with hierarchical low-dimensional representation. *arXiv preprint arXiv:2312.05473*, 2023.

[63] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[64] Il'ya Meerovich Sobol'. On the distribution of points in a cube and the approximate evaluation of integrals. *Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki*, 7(4):784–802, 1967.

[65] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 5026–5033. IEEE, 2012.

[66] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. PMLR, 2018.

A Implementation Details

A.1 Implementation of FocalBO

We implement FocalBO with BoTorch[1], which is a popular library for BO implementation with GPU acceleration. For acquisition optimization, we directly use acquisition function implementation and corresponding optimizers from BoTorch. Our code for fully reproducing all experimental results is in the: `https://github.com/yunyuewei/FocalBO`. Our musculoskeletal model will be released soon. In the meantime, the model can be accessed for research purposes upon request (ysui@tsinghua.edu.cn).

A.2 Implementation of baselines

SVGP. We directly use approximated GP class in Gpytorch example[2].

WOGP. We refer to the original implementation[3], and write a Botorch GP wrapper with inducing point kernel to enable acquisition optimization using BoTorch. As the hyperparameter are unknown to the GP model, we first warm up WOGP using random set of inducing points for 100 epochs, then perform weighted training point selection and continue hyperparameter fitting with the selected WOGP model.

Vecchia GP. We directly use the original implementation[4] without much modification, as it is also implemented in BoTorch.

TuRBO. We refer to the implementation in BoTorch tutorials[5], and use the default setting in trust region length and success/failure thresholds.

A.3 GP training details

For all GP, we use Matern $\frac{5}{2}$ kernel with automatic relevance determination, and do not restrict the lengthscale or noise range. For each round of GP training, we fit GP hyperparameters (and variational parameters for focalized GP and SVGP) for 1000 epochs via Adam optimizer[63] with learning rate as 0.01. For focalized GP and SVGP, we initialize the inducing points using Sobol sampler[64] over input space. all experiment are conducted on a server with Intel(R) Xeon(R) Gold 6348 CPU @ 2.60GHz, NVIDIA-A100 and 512Gb memory.

A.4 Synthetic functions

For GP function, we directly sample from a exact 5d GP using Matern $\frac{5}{2}$ kernel with lengthscale as 0.5. For other synthetic functions, we directly use the test function implementation from BoTorch.

A.5 Robot morphology design

We use the dataset and function oracle from Design Bench[6]. We choose D'Kitty morphology design for its consistency in function values between offline dataset and online function oracle, and its compatibility with python 3.8+.

A.6 Human musculoskeletal system control

We use the musculoskeletal system from [62], which enables foward simlation with Mujoco[65] and environment customization. We design the following reward for each environment step:

[1] `https://botorch.org/`
[2] `https://gpytorch.ai/`
[3] `https://github.com/ermongroup/bayes-opt`
[4] `https://github.com/feji3769/VecchiaBO/tree/master/code/pyvecch`
[5] `https://botorch.org/tutorials/turbo_1`
[6] `https://github.com/brandontrabucco/design-bench`

$$r = 50r_{\text{pos}} * 10r_{\text{ori}} + 10r_{\text{reach}} + r_{\text{lift}} - r_{\text{act}} - 5r_{\text{done}} \tag{10}$$

where r_{pos} encourages the bottle near the target position, r_{ori} encourages the bottle near the target orientation, r_{reach} encourages the hand to grab the bottle, r_{lift} encourages the hand to lift the bottle, r_{act} penalize the overall muscle activation, r_{done} penalize the early ended episode due to dropped bottle or hand outside of pre-defined range.

We trained a Soft Actor-Critic (SAV) [66] agent for 6M timesteps to collect task-related muscle activation data, and use principled component analysis to reduce the action dimension from 81 to 5.

B Additional Experiments

B.1 Theoretical implications of sparse GP approximation.

In Figure 6, we also empirically measure our claim that Focalized GP can significantly reduce approximation error on the search region. We sampled 8000 training points from 2d GP functions to train focalized GP and SVGP. Over different size of the search region, we compare the KL divergence of the GP posterior prediction over search region between sparse GPs and the exact GP. We observe that the KL divergence between focalized GP and exact GP is consistently smaller than that between SVGP and exact GP, implying tighter approximation to the exact GP over local region.

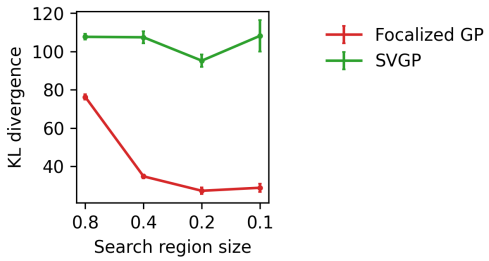

Figure 6: KL divergence between sparse GPs and exact GP. Results shows the mean and one standard error, averaged over 50 independent trials.

While a rigorous regret bound is hard to derive, we conduct an empirical study where we directly compare the optimization performance between focalized GP and SVGP when combining with TuRBO. In this way we can eliminate the influence of hierarchical acquisition optimization. The optimization performances are shown in Figure 7. We observe that focalized GP outperforms SVGP on both high-dimensional problems, which empirically demonstrates our theoretical implications that Focalized GP contributes to reducing regret.

Different way of centering the search region

We empirically investigate this in Figure 8 (a), which compares different ways of selecting the search region center by measuring the distance from the search region center to the global optima. We observe that current best point consistently is the closest to the global optimum, which validates this design choice.

For the experiment above, we sampled 2d functions from GPs with Matern $\frac{5}{2}$ kernel and lengthscale of 0.05 (representing rigid functions), and selected the best point over unifromly sampled 10,000 points as the global optima.

A sparse GP is already more explorative than using the full GP, since the smaller representational capacity leads to smoother posteriors. In Figure 8 (b), We demonstrate this empirically below, where we measure the pair-wise distance of 100 Thompson sampling points under exact and SVGP (with 50 inducing points). We observe that sparse GP actually samples more diverse sets compared to exact GPs, i.e. exhibiting more exploration. Therefore,

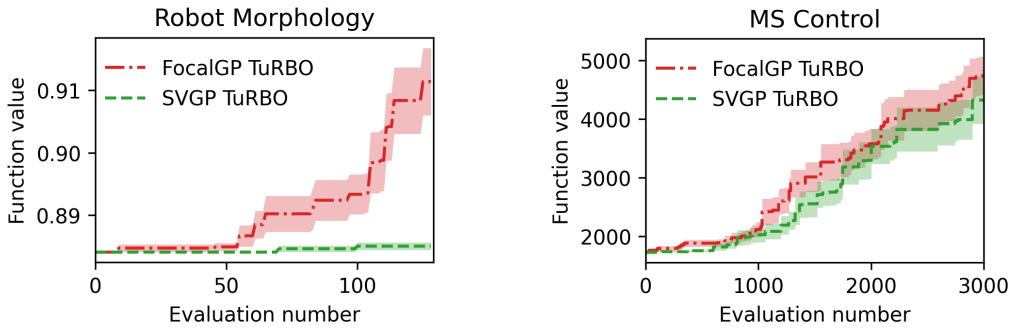

Figure 7: Optimization performance of focalized GP and SVGP when combining with TuRBO.

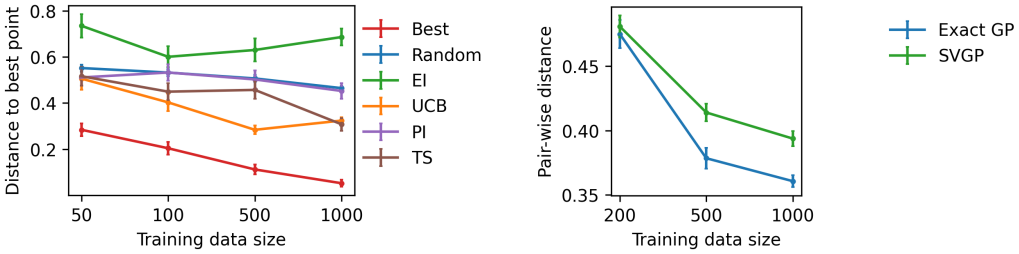

Figure 8: (a) Distance of search region center to the global optima. (b) Pair-wise distance of Thompson sampling samples. Results shows the mean and one standard error, averaged over 50 independent trials.

using focalized GPs does not sacrifice exploration, and significantly helps exploitation by performing acquisition function optimization over smaller search regions.

B.2 Optimization on synthetic functions with large online data

We choose Ackley and Hartmann, which are common-used test functions for BO community. We use the similar optimization setting in [12]. The optimization performances are shown in Figure 9, where FocalBO is still able to achieve comparable or better performance when online samples dominates the data source.

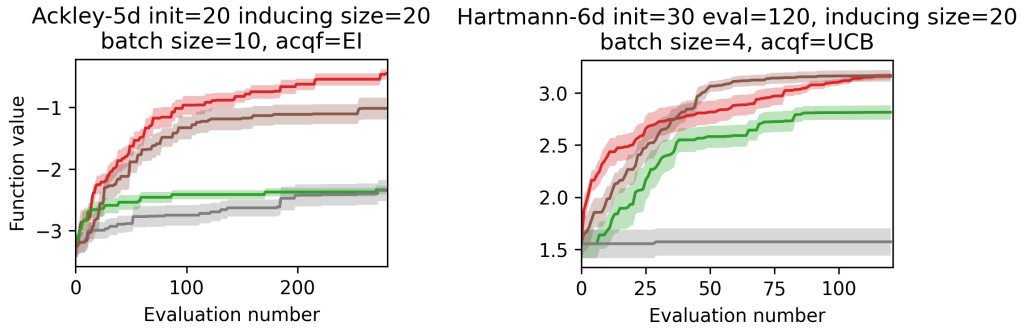

Figure 9: GP predictive performance of specific search region on 2d Ackley and Rastrigin function. Results show mean ± one standard deviation over 10 random search regions.

B.3 GP predictive performance

We use two common-used synthetic functions, Ackley and Rastrigin, to analyze the the GP predictive performance of focalized GP compared with Exact GP and SVGP under different search region size l and different inducing variables number m. We show the negative log likelihood (NLL) and root mean squared error (RMSE) in Figure 10. The results shows that focalized GP outperforms both Exact GP and SVGP in terms of both NLL and MSE when the search space size is lower than 0.5. In Rastrigin function where Exact GP achieves similar performance as SVGP, focalized GP is still able to accurately predict the local search region over different choice of inducing variable numbers. We also show in Figure 11 that the regularization term $\mathcal{L}_{\text{reg}}$ is indispensable to the training of focalized GP to achieve good local prediction.

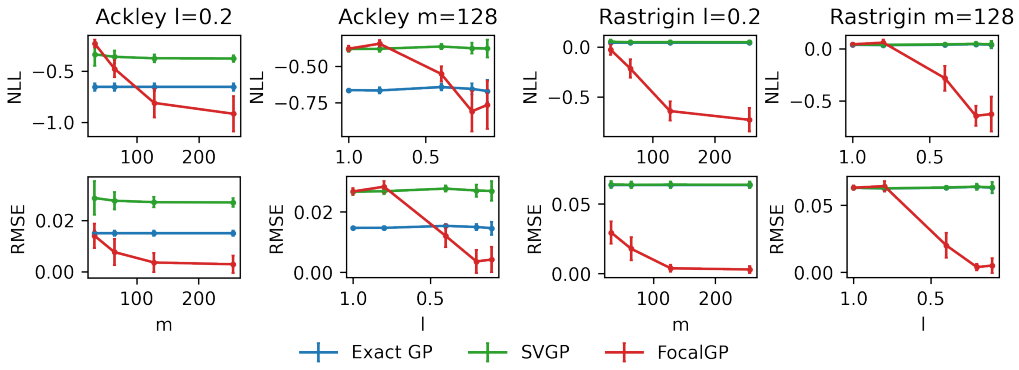

Figure 10: GP predictive performance of specific search region on 2d Ackley and Rastrigin function. Results show mean ± one standard deviation over 10 random search regions.

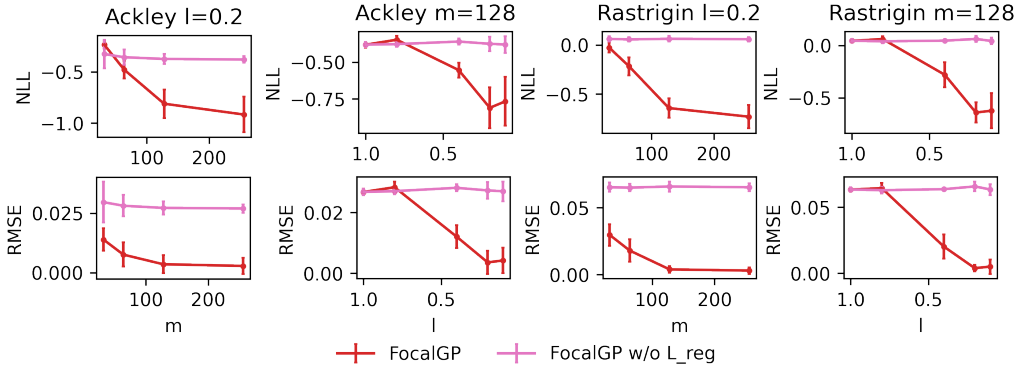

Figure 11: Ablations on the reguralization loss $\mathcal{L}_{reg}$. Results show mean ± one standard deviation over 10 random search regions.

B.4 Comparison with TuRBO

We run the original TuRBO implementation (with exact GP and Thompson sampling) and TuRBO with nearest neighbor GO model on both robot morphology design and human musculoskeletal system control task (Figure 12). We observed that FocalBO outperforms TuRBO on both tasks with smaller computational cost. The reason of TuRBO's poor performance may be that it cannot quickly adapt over the search space when the online evaluation budget is small.

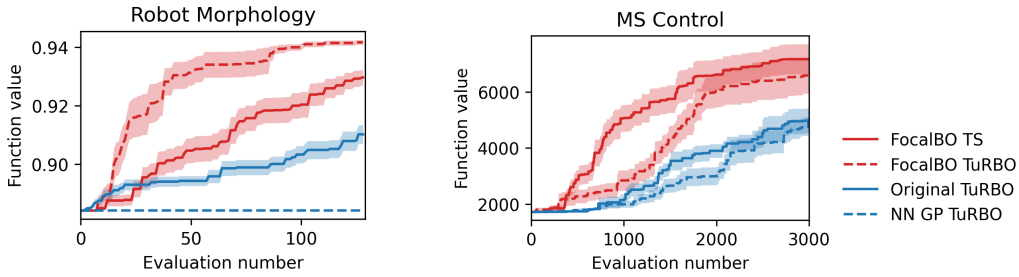

Figure 12: Optimization performance of FocalBO and TuRBO.

C Lemmas Used for Theoretical Implications of Focalized ELBO

Lemma 1. *(Corollary 19 in [58]). Let k be a squared exponential kernel. Suppose that N real-valued (onedimensional) covariates are observed, with identical Gaussian marginal distributions. Suppose the conditions of Theorem 13 are satisfied for some $R > 0$. Fix any $\gamma \in (0, 1]$. Then there exists an $M = \mathcal{O}(\log(N^3/\gamma))$ and an $\epsilon = \Theta(\gamma/N^2)$ such if inducing points are distributed according to an ϵ-approximate M-DPP with kernel matrix K_{ff},*

Lemma 2. *(Proposition 1 in [58]). Suppose $2KL[Q \parallel P] \leq \gamma \leq \frac{1}{5}$. For any $x^* \in \mathcal{X}$, let μ_1 denote the posterior mean of the variational approximation at x^* and μ_2 denote the mean of the exact posterior at x^*. Similarly, let σ_1^2, σ_2^2 denote the variances of the approximate and exact posteriors at x^*. Then,*

$$|\mu_1 - \mu_2| \leq \sigma_2\sqrt{\gamma} \leq \frac{\sigma_1\sqrt{\gamma}}{\sqrt{1-\sqrt{3\gamma}}} and |1 - \sigma_1^2/\sigma_2^2| < \sqrt{3\gamma} \tag{11}$$

Lemma 3. *(Assumption 4 in [37]). (quality of the approximate prediction). For the approximate $\tilde{\mu}_t$, the exact μ_t and σ_t, and for all $x \in \mathcal{X}$,*

$$|\tilde{\mu}_t(x) - \mu_t(x)| \leq c_t\sigma_t(x), \tag{12}$$

where $0 \leq c_t \leq c$ for all $t > 1$ and some constant $c \in \mathbb{R}$

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and precede the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS paper checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We conduct comprehensive evaluations on the proposed methods on both offline and online data setting, and compare with existing sparse GP-based BO baselines to support our main claims.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.

- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We mention the limitations.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: This paper does not involve rigorous theoretical analysis about the proposed method.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.

- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Code to fully reproduce all experimental results has been attached.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Code to fully reproduce all experimental results has been attached.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.

- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: All related experimental setting is stated in the main paper or the appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: All of the results are plotted with averaged performance with errorbar, and from the plot FocalBO significantly outperforms baselines.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.

- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The related content has been stated in Appendix A.3

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We propose `FocalBO`, which is able to optimize high-dimensional data with large offline/online sample budgets. It can be used in high-dimensional robot control.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: There is no risk about the possible misuse.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [NA]

    Justification: The paper does not use existing assets.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: he paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This paper does not include related topic in this question.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This paper does not include related topic in this question.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.