Cylindrical Thompson Sampling for High-Dimensional Bayesian Optimization

Bahador Rashidi[†]
Huawei Canada

Kerrick Johnstonbaugh[†]
Huawei Canada

Chao Gao
Huawei Canada

Abstract

Many industrial and scientific applications require optimization of one or more objectives by tuning dozens or hundreds of input parameters. While Bayesian optimization has been a popular approach for the efficient optimization of blackbox functions, its performance decreases drastically as the dimensionality of the search space increases (i.e., above twenty). Recent advancements in high-dimensional Bayesian optimization (HDBO) seek to mitigate this issue through techniques such as adaptive local search with *trust regions* or dimensionality reduction using *random embeddings*. In this paper, we provide a close examination of these advancements and show that sampling strategy plays a prominent role and is key to tackling the curse-of-dimensionality. We then propose cylindrical Thompson sampling (CTS), a novel strategy that can be integrated into single- and multi-objective HDBO algorithms. We demonstrate this by integrating CTS as a modular component in state-of-the-art HDBO algorithms. We verify the effectiveness of CTS on both synthetic and real-world high-dimensional problems, and show that CTS largely enhances existing HDBO methods.

1 INTRODUCTION

The need for optimization of high-dimensional blackbox functions with unknown gradients is pervasive in real-world applications, ranging from hyper-parameter optimization (Kandasamy et al., 2018), camera image signal processor tuning (Mosleh et al., 2020), policy parameter optimization in robotic control (Calandra et al., 2016), vehicle design optimization (Kohira et al., 2018), and drug discovery (Negoescu et al., 2011).

Bayesian optimization (BO) stands out for its exceptional sample efficiency in the optimization of blackbox functions. Typically, these BO algorithms encompass two vital components. The first employs learning methodologies to construct a surrogate model of the unknown objective function(s). The second component utilizes this model, combined with a sample acquisition strategy, to judiciously select query point(s) for evaluation. By leveraging the uncertainty quantification provided by the surrogate model, BO algorithms balance exploration and exploitation, resulting in highly sample-efficient optimization. While BO has emerged as a versatile approach with applications in diverse problem domains, a fundamental challenge remains — the "curse of dimensionality", which poses a significant obstacle to traditional BO (Wang et al., 2018b). Presumably due to increased real-world demand, there is an increased interest in overcoming this challenge by enhancing the performance of high-dimensional Bayesian optimization (HDBO). Different approaches to HDBO have been explored, including transformation of the search space geometry (Oh et al., 2018; Jaquier et al., 2020), limiting search to local trust regions (Eriksson et al., 2019; Daulton et al., 2022), and simultaneous discovery and optimization of low-dimensional active subspaces (Wang et al., 2016a; Munteanu et al., 2019; Raponi et al., 2020; Papenmeier et al., 2022). However, there is a lack of systematic study on how these independent design features affect HDBO performance, and which of them is most critical.

In this paper, we aim to bridge this gap by first reviewing recent HDBO methods, where each can be viewed as a combination of design choices to effectively tackle the curse-of-dimensionality. From there, we show that the candidate generation strategy is arguably the most important component in Thompson sampling-based HDBO algorithms. Upon this observation, we develop *Cylindrical Thompson Sampling (CTS)*, which

Proceedings of the 27[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2024, Valencia, Spain. PMLR: Volume 238. († Equal contributions, alphabetical order)

can improve high-dimensional exploration in existing HDBO frameworks. Experiments in which CTS is integrated as a modular component in state-of-the-art algorithms show that CTS enhances HDBO performance on both public benchmarks and real-world domains. The source code for CTS-BO is available at `https://github.com/HW-AI-Research/CTS-HDBO`.

2 SUMMARY OF PREVIOUS WORK

We begin by examining the curse-of-dimensionality in HDBO from the sampling strategy point of view. After that, we provide an anatomy of HDBO by extensively reviewing recent HDBO algorithms. We then summarize their algorithmic differences by identifying a list of design choices where they differ by employing disparate choices to mitigate the dimensionality issue.

2.1 Curse-of-Dimensionality From the Perspective of Sampling Strategies

The curse-of-dimensionality manifests in many ways, and in Bayesian optimization it is particularly sinister (Binois and Wycoff, 2022). At the root of several issues is the fact that uniformly sampled points in high dimensions will almost certainly be distant from one another (see Figure. 1). In the course of optimization, a candidate point randomly sampled from a high-dimensional search space is likely to be distant from all previously observed points. This is problematic because the commonly used Gaussian Process (GP) model relies heavily on the assumption that nearby points exhibit similar function values. If a candidate point has no nearby neighbors, the posterior will thus exhibit high uncertainty. This high uncertainty yields large scores according to the acquisition functions, causing the BO algorithm to become trapped in a cycle of constant exploration. This phenomenon is dubbed the "over-exploration problem" Siivola et al. (2018). Due to the over-exploration problem, standard GP surrogates without any dedicated strategies for dealing with high-dimensional spaces (e.g., (Snoek et al., 2012)) are usually limited to BO with functions over spaces with less than twenty dimensions (Letham et al., 2020).

2.2 Anatomy of HDBO

Existing HDBO solutions tackle the curse of dimensionality by innovations from majorly three aspects: *parameter range reduction*, *parameter dimensionality reduction* and *effective strategies for acquisition*. In below, we review each of them and then provide a focused discussion on the acquisition strategy.

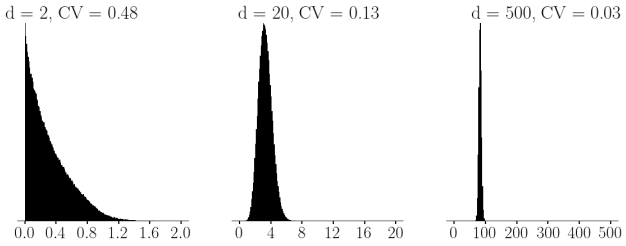


Figure 1: Squared interpoint distances of points sampled uniformly from unit hypercubes of increasing dimensionality. The coefficient of variation (CV) decreases as d increases.

(1) Parameter range reduction as proposed in LaMoo(Zhao et al., 2022), utilizes a support vector machine (SVM) within a Monto-Carlo Tree Search (MCTS) defined over search space to truncate the parameter range space, distinguishing good (i.e., Pareto-front areas likely to yield optimum solutions) from bad regions. While this approach offers improvements over standard BO (Daulton et al., 2021), its computational complexity restricts its practicality. In (Perrone et al., 2019), it is proposed to utilize transfer learning to design BO search space automatically. This approach outperforms its predecessor sequential model-based optimization (SMBO) (Wistuba et al., 2015) which also leverages knowledge transfer to reduce search space volume.

(2) Parameter dimensionality reduction effectively handles the challenge of high dimensionality in Bayesian Optimization (BO) by employing linear embeddings (Munteanu et al., 2019; Wang et al., 2016a; Letham et al., 2020; Raponi et al., 2020). These embeddings transform the high-dimensional parameter space into a more manageable lower-dimensional equivalent for GP-based BO. However, active-subspace methods often necessitate user-guessed dimensions for the active subspace, which is impractical. In contrast, BAxUS (Papenmeier et al., 2022) introduces a distinct random embedding approach, transforming the parameter space into a lower-dimension latent space, progressively increasing them towards a theoretical upper limit. On the other hand, SAASBO (Eriksson and Jankowiak, 2021) deactivates redundant parameter dimensions in the BO surrogate model using priors on the length scale hyperparameters, and achieves rapid convergence to competitive solutions, albeit with high computational complexity and limited scalability to high budgets.

(3) Acquisition strategy is pivotal for HDBO, since in high-dimensional spaces, effective exploration becomes very difficult. In comparison to standard BO, it requires specialization in **Sampling Method**, *Sur-*

rogate Model Choice, *Surrogate Model Domain*, and *Surrogate Model Sampling Scope*. Figure. 2 summarizes the diverse design choices by recent advancements in HDBO.

(3.I) Sampling Methods can have a significant impact on the performance of BO algorithms. In the case of Monte Carlo acquisition functions, the choice of sampler can affect approximation variance. Several common acquisition functions can be expressed as the **expectation** of some function of the model output (Daulton et al., 2021; Frazier et al., 2009). Evaluating these integrals is often intractable, especially in the batched/parallel acquisition setting. As such, the expectations are typically replaced with Monte Carlo approximations. Quasi-Monte Carlo sampling approaches (e.g., scrambled Sobol sequences) are typically used to reduce the variance of these approximations (Balandat et al., 2020).

As an alternative to Monte Carlo acquisition functions, Thompson sampling (TS) (Russo et al., 2020) has been embraced as a versatile acquisition strategy. In Thompson sampling, the surrogate model posterior is sampled over a discrete set of candidate points. The candidate point with the best function value according to the posterior sample is then selected for evaluation. Several HDBO algorithms, namely TuRBO (Eriksson et al., 2019) and its multi-objective extension MORBO (Daulton et al., 2022), rely on TS with a specific approach to generating candidate points (Figure 2 **I-c**). In these algorithms, candidates for TS are generated by perturbing promising points along random axis-aligned subspaces (i.e., by perturbing random subsets of the dimensions). Although this sampling approach was unnamed in the original publications, we found it crucial to TuRBO/MORBO's performance in high-dimensional problems. We will henceforth refer to this sampling strategy as **R**andom **A**xis-**A**ligned **S**ubspace **P**erturbations (**RAASP**)[1].

RAASP Sampling reduces over-exploration by limiting the distance between the generated candidate points and previously observed points. By perturbing a small number (say 20) of coordinates in a high-dimensional (e.g., $d = 500$) space, RAASP effectively yields interpoint distances distributed as in the central plot of Figure 1. Naïvely sampling in the original space would result in much larger interpoint distances, according to the distribution depicted on the right-hand side of Figure 1. The smaller interpoint distances achieved through RAASP sampling enable GP surrogates to predict the local function values with a much higher degree of confidence. In practice, this leads to more efficient optimization.

[1]BAxUS, which builds upon TuRBO, also inherits RAASP sampling.

While RAASP sampling is effective at mitigating over-exploration, it can lead to under-exploration during optimization of functions in which only a few coordinates in the high-dimensional search space matter. To address this issue, we propose Cylindrical Thompson Sampling (CTS) as an alternative approach that scales to higher dimensions while striking a balance between over-exploration and under-exploration. The phenomena of over-exploration and under-exploration in HDBO are demonstrated in section 4.1.2.

(3.II) Surrogate Model Choice pertains to the selection of surrogate model type, surrogate kernel geometry, and distance metric. Common surrogate model types encompass non-parametric Gaussian Process (GP) regression (Snoek et al., 2012), Tree-based uncertainty models such as Random Forest (RF) (Kim and Choi, 2022), Gradient Boosting (van Hoof and Vanschoren, 2021), and Neural Network Ensembles (Lim et al., 2021).

In general, the choice of surrogate geometry and distance metric is interlinked with the choice of surrogate type. In particular, for GP surrogates, the discussion focuses on design considerations concerning kernel geometry and distance metrics. For instance, the Matérn and squared exponential (SE) kernels, typically using Euclidean distance, are the most common kernels for GP surrogates. As a result, the Euclidean distance is commonly used in recent HDBO approaches (Daulton et al., 2021; Eriksson et al., 2019; Papenmeier et al., 2022; Konakovic Lukovic et al., 2020; Daulton et al., 2022). Alternatively, ALEBO (Letham et al., 2020) replaces Euclidean distance in the SE kernel with Mahalanobis distance and introduces a Mahalanobis kernel. The BOCK algorithm (Oh et al., 2018) transforms the search space geometry with cylindrical kernels, which have the effect of expanding the volume near the center of the search space.

(3.III) Surrogate Model Learning Domain. The domain over which surrogates are tasked with modeling the unknown function (Figure. 2III) directly influences sample efficiency and scalability in HDBO. Standard BO methods typically employ a single surrogate model (i.e., *Global*) to predict objective values across the entire parameter search space. Another intuitive approach utilizes a local surrogate models through Trust Regions (TRs), as in (Eriksson et al., 2019). In this strategy, trust regions are defined over promising regions of the parameter search space, and within these trust regions samples are used to train local models. As depicted in Figure. 2, TR-based BO methods rely on two additional design choices: TR shape (e.g., Rectangular and Spherical) and the number of TR.

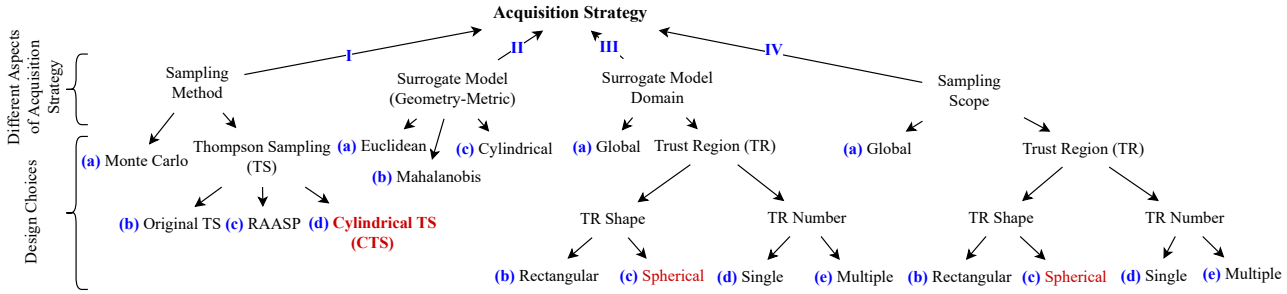


Figure 2: Different design choices for acquisition strategy within high-dimensional BO

Table 1: Summary of how recent HDBO methods tackle curse-of-dimensionality

Method	Dim Reduction	Scalability	Support # of Obj	Acquisition Strategy wr.t. Figure 2
CTS-BO (Ours)	✗	✓	Multi	I(d), II(a), III(a,b,c,d), IV(a,b,c,d)
BAxUS(Papenmeier et al., 2022)	✓	✓	Single	I(c), II(a), III(b,d,e), IV(b,d,e)
MORBO(Daulton et al., 2022)	✗	✓	Multi	I(c), II(a), III(b,d,e) ,IV(b,d,e)
LaMOO-qNEHVI(Zhao et al., 2022)	✗	✗	Multi	I(a), II(a), III(a), IV(a)
SAASBO(Eriksson and Jankowiak, 2021)	✓	✗	Single	I(a), II(a), III(a), IV(a)
qNEHVI(Daulton et al., 2021)	✗	✗	Multi	I(a), II(a),III(a), IV(a)
DGEMO(Konakovic Lukovic et al., 2020)	✗	✗	Multi	I(a), II(a), III(a), IV(a)
ALEBO(Letham et al., 2020)	✓	✗	Single	I(a), II(b), III(a), IV(a)
TuRBO(Eriksson et al., 2019)	✗	✓	Single	I(c), II(a), III(b,d,e) ,IV(b,d,e)
HESBO(Munteanu et al., 2019)	✓	✓	Single	I(a), II(a),III(a), IV(a)
BOCK(Oh et al., 2018)	✗	✗	Single	I(a), II(c), III(a), IV(a)
REMBO(Wang et al., 2016a)	✓	✗	Single	I(a), II(a),III(a), IV(a)
TSEMO(Bradford et al., 2018)	✗	✗	Single	I(b), II(a), III(a), IV(a)

(3.IV) Sampling Scope dictates whether candidate points are sampled globally from the entire search space, or locally from a specific region. Although the choice of sampling scope typically mirrors the surrogate model learning domain (Daulton et al., 2022), in principle these design choices are independent.

In summary, Table 1 offers a thorough comparison of recent HDBO approaches by presenting the corresponding choices made in the design tree depicted in Figure 2. We further outline their scalability, which is indicative of an approach's ability to effectively manage a relatively large evaluation budget (e.g., ≥ 1000) without encountering severe computational inefficiencies. Scalability can be achieved through the utilization of trust regions to regulate the surrogate model domain (reducing the number of samples used to compute the GP posterior) or by implementing a batch sampling strategy. Conversely, LaMOO (Zhao et al., 2022) and SAASBO (Eriksson and Jankowiak, 2021) do not scale to large evaluation budgets due to the introduction of additional computational complexity that becomes prohibitively expensive.

3 CYLINDRICAL THOMPSON SAMPLING

We propose a novel strategy for sampling in HDBO algorithms, namely cylindrical Thomson sampling (CTS). Algorithm 1 presents pseudocode for CTS-BO: a generic BO framework that includes CTS as a module. This module can also be integrated with SOTA trust-region-based algorithms (see section 4.2).

CTS transforms the geometry of the low-level sampler within the standard TS acquisition function. Samples take the form

$$\mathbf{x} = \mathbf{c} + r\mathbf{v}, \quad r \sim \mathcal{U}(0, R), \quad \|\mathbf{v}\| = 1, \tag{1}$$

where $\mathbf{c}$ is the **center of perturbation**, unit vector $\mathbf{v}$ is the **angular component**, and perturbation distance r is the **radial component** of the sample. In our experiments, we take the best solution achieved thus far (i.e. the incumbent solution) as the center of perturbation. Note that in a bounded search space, the maximum perturbation radius R depends on $\mathbf{v}$. Thus $\mathbf{v}$ must be sampled before r. As we will show, the choice of distribution from which to sample $\mathbf{v}$ is critical to the performance of CTS as an acquisition sampling method in Bayesian optimization.

3.1 Sampling the Angular Component

The CTS angular component (i.e. perturbation direction) in Eq. (1) can be sampled in various ways. A naive approach is to sample uniformly from the surface of a d-dimensional hypersphere. In practice, such samples are typically generated by normalizing sam-

Algorithm 1 CTS-BO: Bayesian optimization with Cylindrical Thompson Sampling

Input: Initial data $\mathcal{D}$, incumbent and lowest discovered function value $(\mathbf{c}, y^*)$, budget B
Output: Minimizer $\mathbf{x}^* \in \operatorname{argmin}_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x})$
$n_{\text{evals}} \leftarrow |\mathcal{D}|$
while $n_{\text{evals}} \leq B$ **do** ▷ Until budget expended
  Init set of eval candidates $\mathcal{X}_{\text{cand}} \leftarrow \emptyset$
  for $i = 1$ to n_{cand} **do**
    Sample $\mathbf{v}$ from TMVN ... (Eq. 4)
    Compute $r_{\max}(\mathbf{c}, \mathbf{v})$ ... (Eq. 5)
    $R \leftarrow \min\{r_{\max}(\mathbf{c}, \mathbf{v}), R_{\max}\}$
    Sample radial component $r \sim \mathcal{U}(0, R)$
    $\mathbf{x}_{\text{cand}} \leftarrow \mathbf{c} + r\mathbf{v}$
    $\mathcal{X}_{\text{cand}} \leftarrow \mathcal{X}_{\text{cand}} \cup \{\mathbf{x}_{\text{cand}}\}$
  end for
  $\mathbf{x}_{\text{eval}} \leftarrow \operatorname{argmin}_{\mathbf{x}\in\mathcal{X}_{\text{cand}}} y_{\text{pred}}$, where $y_{\text{pred}} \sim \mathcal{GP}(\mu(\mathbf{x}), \sigma(\mathbf{x}) | \mathcal{D})$ } CTS
  Collect y_{eval} by evaluating f at $\mathbf{x}_{\text{eval}}$
  if $y_{\text{eval}} < y^*$ **then**
    Update incumbent $\mathbf{c} \leftarrow \mathbf{x}_{\text{eval}}$
    Update best function value $y^* \leftarrow y_{\text{eval}}$
  end if
  Update dataset $\mathcal{D} \leftarrow \mathcal{D} \cup \{(\mathbf{x}_{\text{eval}}, y_{\text{eval}})\}$
  $n_{\text{evals}} \leftarrow n_{\text{evals}} + 1$
end while

ples from a d-dimensional isotropic multivariate normal distribution:

$$\mathbf{v} \leftarrow \frac{\mathbf{z}}{\|\mathbf{z}\|}, \quad \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d) \tag{2}$$

Angular components sampled according to Eq. (2) inherently exhibit the spherical symmetry characteristic of the multivariate normal distribution. Indeed, this approach can be effective when the center of perturbation lies near the center of the search space. However, when the perturbation center approaches the search space boundaries, the angular components from this distribution, as depicted in Figure 3, only result in minor perturbations.

To illustrate this phenomenon, let $\mathbf{c}$ represent a perturbation center near the vertex of a d-dimensional hypercube, constrained by $\mathbf{0}$ as the lower bound. Assuming $\mathbf{c} = \epsilon\mathbb{1}_d$ for a small $\epsilon > 0$, meaningful perturbations are achievable only when the samples point away from the vertex (i.e., $\mathbf{v}$ is positive in every coordinate). Given the independence of components in a standard normal random vector, and a 0.5 probability of selecting a positive value for each coordinate, the probability of selecting such a direction is $(0.5)^d$, diminishing exponentially with increasing dimensionality, d.

To avoid this problem, we instead proposed to sample from a truncated multivariate normal distribution (TMVN). The TMVN distribution is derived from the

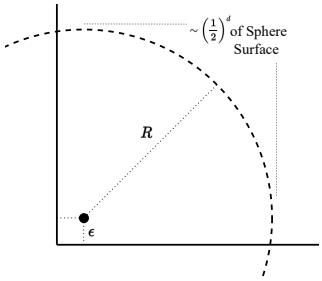


Figure 3: Near a vertex, the probability a uniformly sampled direction affords non-negligible perturbations shrinks exponentially as a function of the search space dimension (prob. $\sim (\frac{1}{2})^d$). This motivates usage of the TMVN angular sampler.

standard multivariate normal distribution, with density

$$f(\mathbf{z}) = \frac{1}{\gamma}\exp\left(-\frac{1}{2}\mathbf{z}^\top\mathbf{z}\right) \cdot \mathbb{I}\{\mathbf{l} \leq \mathbf{z} \leq \mathbf{u}\}, \quad \mathbf{z} = [z_1, \ldots, z_d]^\top, \quad \mathbf{u}, \mathbf{l} \in \mathbb{R}^d, \tag{3}$$

where $\mathbb{I}\{\cdot\}$ denotes the indicator function, and $\gamma = \mathbb{P}(\mathbf{l} \leq \mathbf{z} \leq \mathbf{u})$ is the probability that a random vector $\mathbf{Z}$ with standard normal distribution in d-dimensions falls in the hypercube defined by the lower and upper bounds $\mathbf{l}$ and $\mathbf{u}$. The TMVN distribution cuts off the density of the standard MVN that would lie outside of the search domain, biasing samples away from search domain boundaries. This ensures that meaningful perturbations are afforded with high probability. In CTS, the angular component is sampled according to

$$\mathbf{v} \leftarrow \frac{\mathbf{z}}{\|\mathbf{z}\|}, \quad \mathbf{z} \sim \text{TMVN}(\mathbf{0}, \sigma^2\mathbf{I}_d, \mathbf{l}, \mathbf{u}). \tag{4}$$

Sampling from the TMVN distribution is relatively more challenging and expensive than sampling from its univariate counterpart. We mitigate this cost by implementing a vectorized version of the algorithm for efficient simulation described in (Botev, 2017).

3.1.1 Variance of Angular Sampler

The variance of the truncated multivariate normal angular sampler in CTS has a significant impact on BO performance. When σ is set to a large value, it steers samples away from the search domain's boundaries. Conversely, small values of σ can yield angular components that offer only minor perturbations, resembling those from the untruncated multivariate normal (MVN) angular sampler, as seen in Figure 3. In general, higher σ values promote exploration towards the center of the search space, while smaller values of σ enable precise adjustments, particularly near the search domain's edges.

To balance these considerations, we decrease the value of σ following τ_{fail} consecutive "failures". Here, failure means that the sample selected for evaluation failed to decrease regret relative to the current incumbent. If the algorithm succeeds in improving the best solution thus far, the failure counter is reset to zero. After τ_{succ} consecutive successes, σ is doubled, i.e. $\sigma \leftarrow \min\{\sigma_{\max}, 2\sigma\}$. Similarly, after τ_{fail} consecutive failures, σ is halved, i.e. $\sigma \leftarrow \sigma/2$. This scheme enables the CTS sampler to initially explore the center of the search space, and then fine-tune solutions over time as progress slows.

3.2 Maximum Perturbation Magnitude

In general, we can write the linear constraints (e.g., lower and upper bounds) of the search space in the form $\mathbf{Ax} \leq \mathbf{b}$. Given some center of perturbation $\mathbf{c}$ and an angular component $\mathbf{v}$ s.t. $\|\mathbf{v}\| = 1$, we can compute the maximum perturbation $r_{\max}(\mathbf{c}, \mathbf{v})$ in the direction $\mathbf{v}$. Letting $\rho = r_{\max}(\mathbf{c}, \mathbf{v})$ to simplify notation: $\mathbf{A}(\mathbf{c} + \rho\mathbf{v}) \leq \mathbf{b}$. Then letting $\mathbf{v}' = \mathbf{Av}$ and $\mathbf{b}' = \mathbf{b} - \mathbf{Ac}$, we have $\rho\mathbf{v}' \leq \mathbf{b}'$. Since this vector inequality has to hold for every element, i.e., $\rho v_i' \leq b_i'$ $\forall i \in \{1, \ldots, d\}$, the maximum permissible value for ρ is:

$$r_{\max}(\mathbf{c}, \mathbf{v}) = \rho \leftarrow \min\{b_i'/v_i'\} \tag{5}$$

Finally, CTS limits the maximum perturbation magnitude to some largest radius $R_{\max}$. That is, $R \leftarrow \min\{r_{\max}(\mathbf{c}, \mathbf{v}), R_{\max}\}$. This R is the upper bound for the radial component in CTS (Eq. 1).

3.3 Geometry of Trust Regions with CTS

The choice of trust region (TR) geometry relies on the characteristics of the optimization problem. For problems with general linear or nonlinear constraints on parameter space, an ellipsoidal TR is commonly preferred. However, when dealing with unconstrained or bound-constrained parameter spaces, such as our focus here, a spherical trust region offers simplicity in computation and implementation. Hence, we adopt CTS with spherical TRs for our experimentation, leaving the adaptation of CTS for ellipsoidal TRs, that is a natural extension of rectangular TR used by TuRBO/BAXUS for future research. In our case, the problem constraints are represented as $Ax \leq b$, as discussed in Section 3.2.

A spherical TR geometry can be integrated with CTS by limiting the maximum radius of perturbations. For a CTS with a global scope, the maximum radius $R_{\max}$ is typically set to $\sqrt{d}$, where d is the dimension of the search space (assuming the search space is scaled to a unit hypercube). The adaptation of the spherical TR radius in CTS follows similar principles to the TR side-lengths in TuRBO. The TRs are centered on the incumbent, representing the best current solution. After τ_{succ} consecutive successes, $R_{\max}$ is doubled: $R_{\max} \leftarrow \min\sqrt{d}, 2R_{\max}$. Conversely, after τ_{fail} consecutive failures, $R_{\max}$ is halved: $R_{\max} \leftarrow R_{\max}/2$. If $R_{\max}$ falls below a predefined minimum threshold, the current TR is terminated, and a new TR is initialized. Throughout our experiments, we maintain consistency by using the same τ_{fail} and τ_{succ} values for adjusting σ and $R_{\max}$, ensuring they grow or shrink in tandem.

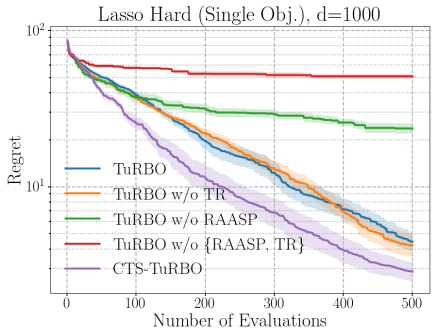


Figure 4: Results of TuRBO variants on 1000-dimension Lasso hard benchmark, with performance ranking `CTS-TuRBO` > `TuRBO` ≈ `TuRBO w/o TR` > `TuRBO w/o RAASP` > `TuRBO w/o {RAASP, TR}`.

4 EXPERIMENTS

To illustrate the strength of CTS, we first elaborate on how sampling strategies affect BO by performing a controlled empirical study with TuRBO, a representative HDBO algorithm. Then, we conduct end-to-end experiments on both synthetic and real-world problems by integrating CTS into three state-of-the-art HDBO algorithms as shown in Table 1.

4.1 Controlled Study with TuRBO

We motivate the development of cylindrical Thompson sampling by first demonstrating the significance of sampling strategies in an ablation of TuRBO. We then study in more detail the mechanistic consequences of each sampling strategy.

4.1.1 Significance of Sampling Strategy

To highlight the importance of the sampling strategy in Thompson sampling, we begin with an ablation of TuRBO, which has served as the core for several state-of-the-art HDBO algorithms (e.g., MORBO and BAxUS; see Table 1).

Using the super-high-dimensional 1000D Lasso-Hard

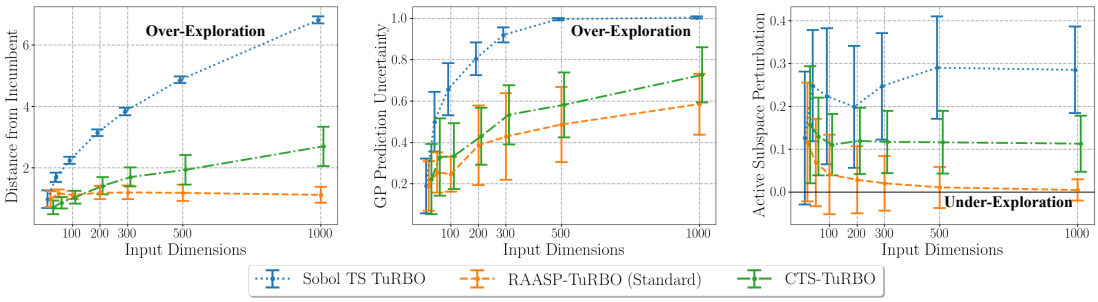


Figure 5: Sobol TS suffers from over-exploration (left and middle), tending to sample distant points with high uncertainty. TuRBO solves this problem by adopting RAASP. RAASP reduces over-exploration but perturbs the active subspace with decreasing probability (under-exploration) as dimensionality increases (right). CTS strikes a balance between these extremes, ensures significant perturbations in the active subspace (right), while reducing over-exploration (left and middle).

test problem (Šehić et al., 2022) as a benchmark, we collect experiment results of the following algorithm variants along with the original TuRBO.

(1) `TuRBO w/o RAASP`: Replace the RAASP candidate generation of TuRBO's TS acquisition with Sobol sampling. (2) `TuRBO w/o TR`: Turn off the use of TR from TuRBO, performing global optimization with RAASP sampling. (3) `TuRBO w/o {RAASP, TR}`: Turn off both RAASP sampling and TRs, resulting in global Thompson sampling with Sobol candidate generation. (4) `CTS-TuRBO`: Replace the RAASP sampler of TuRBO with cylindrical Thompson sampling.

As in Figure 4, experiment results indicate that in TuRBO, the use of sampling method RAASP had a greater impact than the adoption of trust regions. Removing the RAASP sampler from TuRBO while resorting to generating candidates in the global search space using scrambled Sobol sequences resulted in considerable performance degradation of TuRBO. Meanwhile, removing the use of TR only slightly harmed TuRBO on this benchmark. Additionally, replacing RAASP with CTS enhanced TuRBO, producing lower regret than original TuRBO.

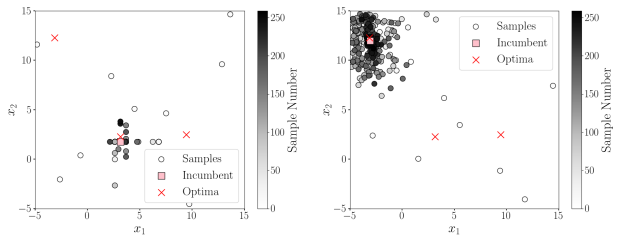
4.1.2 How Different Thompson Sampling Methods Affect TuRBO

To gain insight into the mechanics and limitations of different sampling strategies, we investigated how model uncertainty increases as a function of search space dimensionality for three TS strategies: `Sobol TS`, `RAASP`, and `CTS`. As the distance between a candidate and previously observed points (e.g., the incumbent) grows, the uncertainty of the GP surrogate model increases. We measure the *magnitude of perturbations* within the global search space, specifically examining the distance between samples and incumbent points. At the same time, we quantify the *GP surrogate uncertainty* for the selected samples. While consistent high uncertainty can lead to over-exploration, a successful sampling strategy must explore the active subspace of the test function. We track the *magnitude of perturbations in the active subspace* to quantify active subspace exploration. Data were collected by tracking the initial 50 points selected for evaluation during optimization of the embedded Branin2 function (Wang et al., 2016b). This function serves as an ideal benchmark for assessing the performance of HDBO algorithms on problems with low-dimensional active subspaces. The experiment was replicated across various dimensions, ranging from 20 to 1000.

Sobol TS succumbs to the curse-of-dimensionality, with perturbation magnitudes increasing significantly with the number of input dimensions (Figure 5, left). These large perturbations result in high uncertainty (Figure. 5, middle), which leads to over-exploration. On the other hand, RAASP solves the problem of over-exploration by limiting perturbations to low-dimensional random subspaces. However, this solution introduces a new problem; as the dimensionality of the search space increases, RAASP perturbs the active subspace with decreasing probability. When averaged across many samples, this results in a tiny value for the mean active subspace perturbation (Figure. 5, right). It is worth mentioning that under-exploration of the active subspace leads to slow optimization progress. As seen in Figure 5, CTS maintained a balance between these two extremes. Global perturbation magnitudes grow slower than Sobol Thompson sampling, and the uncertainty of the GP surrogate is comparable to that of RAASP sampling (Figure. 5, left & middle). At the same time, CTS managed to consistently produce perturbations in the active subspace (Figure. 5, right), which is likely to produce a more effective training set of samples to update the surrogate model.

As an auxiliary analysis, we compared the first 260

samples acquired using RAASP-TuRBO and CTS-TuRBO on the 500D embedded Branin function in Fig. 6a and 6b, respectively. Sample candidates generated by RAASP form a cross pattern and do not result in good coverage of the 2-D active subspace (Figure 6a). Many of the samples do not perturb the active subspace at all, causing the points in the plot to overlap (under the incumbent). When the incumbent *is* perturbed in the active subspace, it is usually perturbed along only one dimension, resulting in the formation of the observed cross pattern. Clearly, RAASP sampling struggled to explore the active subspace — this is reflected in the small averaged magnitude of active subspace perturbations in high-dimensional variants of the embedded Branin function (Fig. 5, right). On the other hand, candidates generated by CTS provide a local exploration of the active subspace (Figure. 6b). These results corroborate the quantitative measure of active subspace perturbations provided in the right-hand plot of Figure 5.



(a) TuRBO with RAASP. (b) TuRBO with CTS.

Figure 6: Candidates selected for evaluation projected to the 2D active subspace of the 500D embedded Branin function. RAASP generated samples that mostly overlap when projected onto the active subspace.

4.2 End-to-End Results with CTS

We investigate the effectiveness of CTS by comparing with more HDBO algorithms including `SAASBO` (Eriksson and Jankowiak, 2021), random embedding based `BAxUS` (Papenmeier et al., 2022), parallel BO by `qNEHVI` (Daulton et al., 2021) and `MORBO` (Daulton et al., 2022). We compare the performance on both single and multiple objective problems. The single objective benchmark set includes synthetic problems `Branin2`, `Hartmann6` (Balandat et al., 2020), `Lasso-Hard` (Šehić et al., 2022) and two real-world problems including `Mopta08` and `SVM` — both examined in (Eriksson and Jankowiak, 2021). The multi-objective setting covers benchmark problems including `Branin-Currin`, `DTLZ2` (Balandat et al., 2020) and two-objective benchmark of `Rover Trajectory Planning` (Daulton et al., 2022). The first two benchmarks are synthetic and last one derives from a real-world application. These problems have large input dimensions ranging from 60 to 1000.

4.2.1 Single-Objective Problems

For this evaluation, in addition to recent SOTA HDBO algorithms `BOCK` `SAASBO`, `TuRBO`, `BAxUS` and a standard `sobol` `TS`, we generate two new algorithms by incorporating CTS, resulting in `CTS-TuRBO` and `CTS-BAxUS`. We also ran `CTS-BO`, which can be viewed as a standard BO without using trust regions or embeddings for dimensionality reduction. In our implementation of TuRBO, we utilize a single trust region, as described in (Eriksson et al., 2019). This choice prioritizes sample efficiency, given that in this setup, TRs do not share data. Opting for a single TR allows us to emphasize the importance of our sampling strategy and to directly compare the performance impact of employing the RAASP versus our proposed CTS. For each run, optimization proceeded with a batch size of 1 and was terminated when the regret fell below the threshold of 0.001. To ensure fair comparisons, we adopted the baseline tuning parameters and configurations used by the authors in the original implementations.

As shown in Figure. 7(a, b, c, d, e), the results across all single objective benchmarks indicate that incorporating CTS into TuRBO led to improved performance. For Branin2 and Hartmann6, adding CTS to BAxUS (`CTS-BAxUS`) significantly accelerated the convergence. In the case of Lasso Hard, `CTS-BAxUS` obtained almost identical results in comparison to the baseline BAxUS with RAASP sampling.

4.2.2 Multi-Objective Problems

In multi-objective optimization, there is not a single optimal solution. Therefore, we take the standard approach of plotting the hypervolume of the empirical Pareto fronts as optimization progresses. We select MORBO as the primary baseline and test on DTLZ2 and Branin-Currin with two objectives and 300 input parameters, as well as the real-world 2-objective Rover Trajectory planning problem (Wang et al., 2018a), with settings as in (Daulton et al., 2022).

MORBO (Daulton et al., 2022) is essentially an extension of TuRBO into the multi-objective realm, utilizing RAASP Thompson sampling for acquisition. In our work, we enhance MORBO by replacing its RAASP sampler with CTS, leading to `CTS-MORBO`. Unlike TuRBO, the MORBO implementation features TRs that share data points within the sampling scope, addressing the sample inefficiency observed in TuRBO. Therefore, we incorporate five trust regions in both our MORBO and CTS-MORBO implementations. Besides Thompson sampling-based approaches,

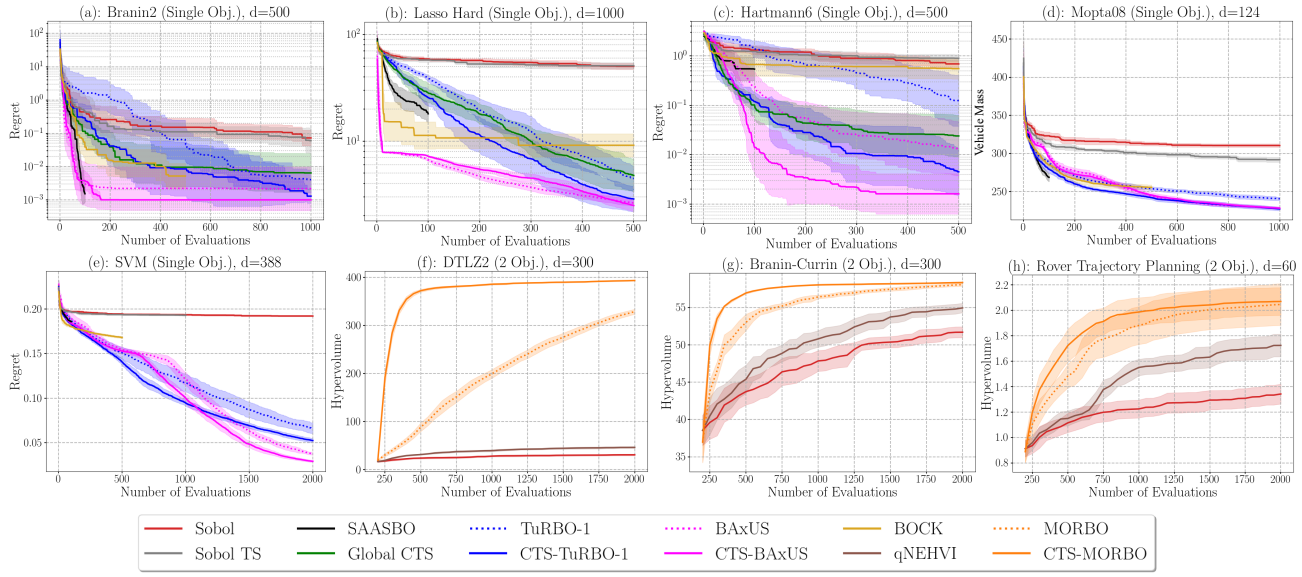


Figure 7: Optimization results on five single and three multi-objective benchmarks, with CTS integrated to TuRBO, BAxUS, and MORBO. Simple regret is plotted for single objective benchmarks (lower means better), while hypervolume is plotted for multi-objective benchmarks (higher means better). Curves plot mean performance with the shaded region representing 95% confidence interval.

we also include a comparison with the recent algorithm `qNEHVI` (Daulton et al., 2021). Following the experimental setup described in (Daulton et al., 2022), we perform multi-objective optimization with a budget of 2000 and a batch size of 50 for each benchmark.

As shown in Figure. 7(f,g,h), consistent to the single-objective case, on both synthetic and real-world benchmarks, `CTS-MORBO` improved optimization efficiency relative to the baselines including `MORBO`, `qNEHVI`, `Random Search`.

4.2.3 Performance of CTS in the Low-Dimensional Setting

CTS exhibits notable improvements for both TuRBO and BAXUS when handling high-dimensional problems. Yet, it is not immediately clear if CTS can retain its effectiveness in a lower-dimensional context: is it true that its usefulness is confined to $d \geq 20$? To investigate this, we selected two low-dimensional problems: *14D-Swimmer* and *14D-Robot Pushing* as in (Eriksson et al., 2019), and conduct comparisons between CTS-TuRBO, the original TuRBO (both with a single trust region), and Sobol (quasi-random) search. As shown in Figure 8, CTS-TuRBO is competitive with TuRBO in Swimmer, but performs slightly worse in Robot Pushing. We conjecture that TuRBO's capacity to adjust trust region side lengths according to GP length scales confers an advantage in the low-dimensional regime. This would suggest that utilizing adaptive ellipsoid trust regions for CTS could enhance performance in low-dimensional optimization.

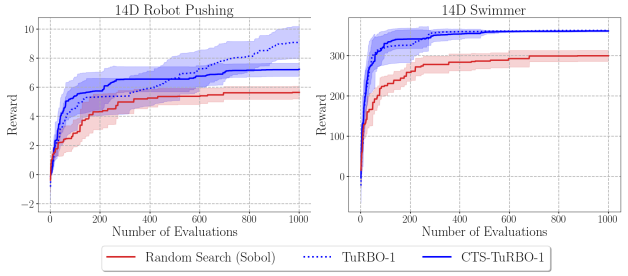


Figure 8: Comparison between TuRBO, CTS-TuRBO, and Sobol search for two low-dimensional problems. These findings suggest that CTS with spherical trust regions may underperform in certain low-dimensional benchmarks (e.g., Robot Pushing).

5 CONCLUSION

We have developed a modular algorithm, namely cylindrical Thompson sampling (CTS), for improving Bayesian optimization in high dimensional spaces. The utility and modularity of CTS has been demonstrated by incorporating it into multiple state-of-the-art algorithms. We have shown that CTS strikes a balance between over and under-exploration in high-dimensional spaces. Experiments on both single and multi-objective benchmarks confirm that CTS can improve the performance of existing HDBO algorithms.

References

Balandat, M., Karrer, B., Jiang, D. R., Daulton, S., Letham, B., Wilson, A. G., and Bakshy, E. (2020). BoTorch: A Framework for Efficient Monte-Carlo Bayesian Optimization. In *Advances in Neural Information Processing Systems 33*.

Binois, M. and Wycoff, N. (2022). A survey on high-dimensional gaussian process modeling with application to bayesian optimization. *ACM Trans. Evol. Learn. Optim.*, 2(2).

Botev, Z. I. (2017). The normal law under linear restrictions: simulation and estimation via minimax tilting. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 79(1):125–148.

Bradford, E., Schweidtmann, A. M., and Lapkin, A. (2018). Efficient multiobjective optimization employing Gaussian processes, spectral sampling and a genetic algorithm. In *J. of Global Optimization*.

Calandra, R., Seyfarth, A., Peters, J., and Deisenroth, M. P. (2016). Bayesian optimization for learning gaits under uncertainty: An experimental comparison on a dynamic bipedal walker. *Annals of Mathematics and Artificial Intelligence*, 76:5–23.

Daulton, S., Balandat, M., and Bakshy, E. (2021). Parallel bayesian optimization of multiple noisy objectives with expected hypervolume improvement. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W., editors, *Advances in Neural Information Processing Systems*, volume 34, pages 2187–2200. Curran Associates, Inc.

Daulton, S., Eriksson, D., Balandat, M., and Bakshy, E. (2022). Multi-objective bayesian optimization over high-dimensional search spaces. In *Proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence*, Proceedings of Machine Learning Research. PMLR.

Eriksson, D. and Jankowiak, M. (2021). High-dimensional bayesian optimization with sparse axis-aligned subspaces.

Eriksson, D., Pearce, M., Gardner, J., Turner, R. D., and Poloczek, M. (2019). Scalable global optimization via local bayesian optimization. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Eriksson, D. and Poloczek, M. (2021). Scalable constrained bayesian optimization. In Banerjee, A. and Fukumizu, K., editors, *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pages 730–738. PMLR.

Frazier, P., Powell, W., and Dayanik, S. (2009). The knowledge-gradient policy for correlated normal beliefs. *INFORMS Journal on Computing*, 21(4):599–613.

Jaquier, N., Rozo, L., Calinon, S., and Bürger, M. (2020). Bayesian optimization meets riemannian manifolds in robot learning. In Kaelbling, L. P., Kragic, D., and Sugiura, K., editors, *Proceedings of the Conference on Robot Learning*, volume 100 of *Proceedings of Machine Learning Research*, pages 233–246. PMLR.

Kandasamy, K., Neiswanger, W., Schneider, J., Poczos, B., and Xing, E. P. (2018). Neural architecture search with bayesian optimisation and optimal transport. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Kim, J. and Choi, S. (2022). On uncertainty estimation by tree-based surrogate models in sequential model-based optimization. In Camps-Valls, G., Ruiz, F. J. R., and Valera, I., editors, *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, pages 4359–4375. PMLR.

Kohira, T., Kemmotsu, H., Akira, O., and Tatsukawa, T. (2018). Proposal of benchmark problem based on real-world car structure design optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, GECCO '18, page 183–184, New York, NY, USA. Association for Computing Machinery.

Konakovic Lukovic, M., Tian, Y., and Matusik, W. (2020). Diversity-guided multi-objective bayesian optimization with batch evaluations. *Advances in Neural Information Processing Systems*, 33.

Letham, B., Calandra, R., Rai, A., and Bakshy, E. (2020). Re-examining linear embeddings for high-dimensional bayesian optimization. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1546–1558. Curran Associates, Inc.

Lim, Y.-F., Ng, C. K., Vaitesswar, U., and Hippalgaonkar, K. (2021). Extrapolative bayesian optimization with gaussian process and neural network ensemble surrogate models. *Advanced Intelligent Systems*, 3(11):2100101.

Mosleh, A., Sharma, A., Onzon, E., Mannan, F., Robidoux, N., and Heide, F. (2020). Hardware-in-the-loop end-to-end optimization of camera image processing pipelines. In *Proceedings of the IEEE/CVF*

Conference on Computer Vision and Pattern Recognition, pages 7529–7538.

Munteanu, A., Nayebi, A., and Poloczek, M. (2019). A framework for bayesian optimization in embedded subspaces. In *Proceedings of the 36th International Conference on Machine Learning, (ICML)*. Accepted for publication. The code is available at https://github.com/aminnayebi/HesBO.

Negoescu, D. M., Frazier, P. I., and Powell, W. B. (2011). The knowledge-gradient algorithm for sequencing experiments in drug discovery. *INFORMS Journal on Computing*, 23(3):346–363.

Oh, C., Gavves, E., and Welling, M. (2018). Bock: Bayesian optimization with cylindrical kernels. In *International Conference on Machine Learning*.

Papenmeier, L., Nardi, L., and Poloczek, M. (2022). Increasing the scope as you learn: Adaptive bayesian optimization in nested subspaces. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K., editors, *Advances in Neural Information Processing Systems*.

Perrone, V., Shen, H., Seeger, M. W., Archambeau, C., and Jenatton, R. (2019). Learning search spaces for bayesian optimization: Another view of hyperparameter transfer learning. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Raponi, E., Wang, H., Bujny, M., Boria, S., and Doerr, C. (2020). High Dimensional Bayesian Optimization Assisted by Principal Component Analysis. In *Parallel Problem Solving from Nature – PPSN XVI (PPSN 2020)*, volume 12269 of *Lecture Notes in Computer Science*, pages 169–183, Leiden, Netherlands. Springer.

Rasmussen, C. E. and Williams, C. K. I. (2005). *Gaussian Processes for Machine Learning*. The MIT Press.

Russo, D., Roy, B. V., Kazerouni, A., Osband, I., and Wen, Z. (2020). A tutorial on thompson sampling.

Siivola, E., Vehtari, A., Vanhatalo, J., Gonzalez, J., and Andersen, M. R. (2018). Correcting boundary over-exploration deficiencies in bayesian optimization with virtual derivative sign observations. In *2018 IEEE 28th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6.

Snoek, J., Larochelle, H., and Adams, R. P. (2012). Practical bayesian optimization of machine learning algorithms. In Pereira, F., Burges, C., Bottou, L., and Weinberger, K., editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc.

Solis, F. J. and Wets, R. J. B. (1981). Minimization by random search techniques. *Math. Oper. Res.*, 6(1):19–30.

Spall, J. C. (2003). *Introduction to Stochastic Search and Optimization*. John Wiley & Sons, Inc., USA, 1 edition.

van Hoof, J. and Vanschoren, J. (2021). Hyperboost: Hyperparameter optimization by gradient boosting surrogate models. *CoRR*, abs/2101.02289.

Wang, Z., Gehring, C., Kohli, P., and Jegelka, S. (2018a). Batched large-scale bayesian optimization in high-dimensional spaces. In Storkey, A. and Perez-Cruz, F., editors, *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, pages 745–754. PMLR.

Wang, Z., Hutter, F., Zoghi, M., Matheson, D., and De Freitas, N. (2016a). Bayesian optimization in a billion dimensions via random embeddings. *J. Artif. Int. Res.*, 55(1):361–387.

Wang, Z., Hutter, F., Zoghi, M., Matheson, D., and De Freitas, N. (2016b). Bayesian optimization in a billion dimensions via random embeddings. *J. Artif. Int. Res.*, 55(1):361–387.

Wang, Z., Jegelka, S., and Hennig, P. (2018b). A survey on high-dimensional gaussian process modeling with applications to bayesian optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(1):1–1.

Wistuba, M., Schilling, N., and Schmidt-Thieme, L. (2015). Hyperparameter search space pruning – a new component for sequential model-based hyperparameter optimization. In *Machine Learning and Knowledge Discovery in Databases*, pages 104–119, Cham. Springer International Publishing.

Zhao, Y., Wang, L., Yang, K., Zhang, T., Guo, T., and Tian, Y. (2022). Multi-objective optimization by learning space partitions. *arXiv*.

Šehić, K., Gramfort, A., Salmon, J., and Nardi, L. (2022). Lassobench: A high-dimensional hyperparameter optimization benchmark suite for lasso.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Not Applicable]
   (b) Complete proofs of all theoretical results. [Not Applicable]
   (c) Clear explanations of any assumptions. [Not Applicable]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Yes]
   (b) The license information of the assets, if applicable. [Not Applicable]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Yes]
   (d) Information about consent from data providers/curators. [Not Applicable]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

A Experimental Details

Here we report additional details, including hyperparameters, used for each algorithm in our experimental results. Unless otherwise stated, the default hyperparameter values from official author implementations were used for the baseline algorithms. Note that the only hyperparameter introduced by CTS is the initial standard deviation σ_{init} of the TMVN sampler. All experiments were run on a server with two 2.3 GHz Intel Xeon Gold 6140 Processors, and an NVIDIA Tesla V100 GPU.

Implementations for the baseline algorithms in our experiments were as follows:

- For TuRBO, we used the implementation at `https://github.com/uber-research/TuRBO`, license: `Uber`, last accessed: Oct 02, 2023.
- For BAxUS, we used the implementation at `https://github.com/LeoIV/BAxUS`, license: `Uber`, last accessed: Oct 02, 2023.
- For MORBO, we used the implementation at `https://github.com/facebookresearch/morbo`, license: `MIT (Copyright Meta)`, last accessed: Oct 02, 2023.
- For SAASBO, qNEI, and qNEHVI, we used the BoTorch (Balandat et al., 2020) implementations at `https://github.com/pytorch/botorch`, license: `MIT (Copyright Meta)`, last accessed: Oct 02, 2023.

A.1 Details Regarding Figure 5

Next, we provide additional experimental details regarding the generation of the exploration-related quantities plotted in Figure 5, section 4.1.2. For each input dimension d (i.e., number of input variables) plotted on the x-axis of Figure 5, the following process was repeated: We carried out three repetitions of Bayesian Optimization on the d-dimensional embedded Branin function. For each repetition, we collected data associated with the first 50 candidates selected for evaluation. The recorded quantities were (1) *Distance from the Incumbent*, (2) *GP Prediction Uncertainty*, and (3) *Active Subspace Perturbation*. For each of these quantities, we accumulated the respective values over the three BO repetitions, resulting in $3 \times (50 \text{ candidates}) = 150$ values for each. In Figure 5, we plot the mean of these 150 values, with the error bars indicating one standard deviation.

The quantities plotted in Figure 5 were computed as follows:

(1) *Distance from the Incumbent*: The L2 distance between the incumbent and the point selected for evaluation.

(2) *GP Prediction Uncertainty*: The variance of the GP posterior at the point selected for evaluation. This represents the models uncertainty about this point.

(3) *Active Subspace Perturbation*: The L2 distance between the incumbent and the point selected for evaluation after projecting both onto the active subspace. In the case of the embedded Branin function, this projection is accomplished by simply taking the first two components of the d-dimensional input vector.

A.2 Single-Objective Optimization

In the reported single-objective experiments, all algorithms used a batch size of 1. In both Figures 4 and 7, the shaded areas around each curve represent ± 1 standard error across 10 repetitions (with different random seeds). All single-objective variants that included CTS (CTS-BO, CTS-TuRBO, CTS-BAxUS) used $\sigma_{\text{init}} = 0.125$. CTS-TuRBO and CTS-BAxUS each sampled 5000 discrete candidate points during acquisition, which is the same as the baseline variants (as recommended by (Eriksson et al., 2019)).

Our implementation of CTS-TuRBO modifies the geometry of trust regions in TuRBO, changing them from rectangular to spherical as described in section 3.3. When fitting the local GP models associated with each trust region, we exclude points outside of a ball centered at the incumbent with radius $2R$. That is, CTS samples points within a sphere of radius R, and selects one of these candidates for evaluation given a GP fitted on data within radius $2R$.

In addition, we chose a more aggressive value for τ_{fail}, the number of "failures" before a trust region decreases in size. In CTS-TuRBO we set τ_{fail} such that the minimum trust region size will be reached in half of the budget, assuming no designs improve upon the current incumbent:

$$\tau_{\text{fail}} \leftarrow \min \left\{ \lceil d/q \rceil , \left\lceil \frac{B'}{2q\kappa} \right\rceil \right\} \tag{6}$$

where $B' = B - n_{\text{init}}$ is the budget remaining after the initial samples[2], q is the batch size (in our single-objective experiments, $q = 1$), and $\kappa = \left\lceil -\log_2(R_{\text{max}}^{(\text{min})}/R_{\text{max}}^{(\text{init})}) \right\rceil$ is the number of TR shrinking events that must occur before the minimum radius is reached. Note that $\lceil d/q \rceil$ is the setting recommended by the authors of TuRBO (Eriksson et al., 2019).

Our implementation of CTS-BAxUS used the dynamically computed value of τ_{fail} recommended by the authors of BAxUS (Papenmeier et al., 2022), until the input dimensionality (i.e., full underlying dimensionality of the problem) d is reached, at which point we set τ_{fail} according to Equation (6) with B' equal to the remaining budget.

For the SAASBO baseline in our experiments, we paired a SAAS GP surrogate with qNEI as the acquisition function. For the GP fitted with a fully Bayesian SAAS prior, we used the NUTS sampler (Eriksson and Jankowiak, 2021) with 128 warmup steps, and 128 samples, and a thinning parameter of 16. Optimization of qNEI used 5 random restarts and 5000 raw samples.

A.3 Multi-Objective Optimization

In the reported multi-objective experiments, all algorithms used a batch size of 50. In both Figure 7, the shaded areas around each curve represent ± 1 standard error across 10 repetitions (with different random seeds). The reported CTS results used $\sigma_{\text{init}} = 0.125$ in the Branin-Currin experiment, 1.0 for DTLZ2 and 0.0625 for Rover trajectory planning. CTS-MORBO sampled 4096 discrete candidate points during acquisition, which is the same as baseline MORBO (as recommended by (Daulton et al., 2022)). For the qNEHVI baseline in our experiments, we used 2 random restarts and 64 raw samples.

B Computational Cost of CTS

The angular sampler of CTS relies on samples from the truncated multivariate normal (TMVN) distribution with density f in equation 3. We simulate samples from this density according to the max-exponentially-tilted (MET) estimator described in (Botev, 2017). This estimator relies on an accept-reject sampling scheme and is *strongly efficient* in a technical sense. However, it uses a method called exponential tilting, and requires the solution to a minimax optimization problem to select the tilting parameters.

Computing these parameters for sampling with general linear constraints (i.e., of the form $\mathbf{Ax} \leq \mathbf{b}$ where $\mathbf{x} \in \mathbb{R}^d$) requires Cholesky decomposition, with complexity $\mathcal{O}(d^3)$, as an initial step. However, with standard hypercube boundary constraints (as in all the experiments reported in this work), this step can be bypassed.

[2]Here and in Algorithm 1, we use $B \in \mathbb{N}$ to represent the evaluation budget.

The cost of the MET estimator is then dominated by the need to solve the aforementioned minimax optimization problem. The solution is given by the system of nonlinear equations described in Eq. (8) of (Botev, 2017). Our implementation builds on the implementation at `https://github.com/brunzema/truncated-mvn-sampler` and uses the same method for this nonlinear optimization, namely, an implementation of the modified Powell method.

Luckily, given that many of the Thompson Sampling (TS) candidates share the same center of perturbation, the optimal tilting parameters need only be calculated once, when the incumbent is updated. The tilting parameters are then cached and reused for all subsequent samples, until the incumbent changes. Hence the cost of optimizing the tilting parameters is amortized across the 5000 TS candidates in each batch, and across batches if the incumbent remains unchanged.

Assuming standard Cholesky-based approaches, exact posterior sampling with all Thompson Sampling acquisition functions (e.g., including RAASP-TS; Sobol TS) have complexity that is cubic with respect to the number of test points (Rasmussen and Williams, 2005).

C Global Consistency of CTS-TuRBO and CTS-BAxUS

We show that CTS-TuRBO and CTS-BAxUS converge to a global optimum as the number of samples tends to infinity, under common assumptions in the Bayesian Optimization literature.

(Papenmeier et al., 2022) extended the proof by (Eriksson and Poloczek, 2021) that established the consistency of TuRBO, relaxing the assumption of a unique global minimizer. They went on to show that after a finite number of evaluations, BAxUS will behave like TuRBO, and thus inherits these convergence guarantees. Here we prove that CTS-TuRBO (and by extension, CTS-BAxUS) also enjoys global consistency. We restate the consistency theorem from (Papenmeier et al., 2022), including the relevant definitions and assumptions, and show that it also applies to CTS-TuRBO and CTS-BAxUS.

Theorem 1 (Consistency of CTS-TuRBO and CTS-BAxUS)**.** *With the following definitions:*

D1. $\{\mathbf{x}_k\}_{k=1}^{\infty}$ *is a sequence of points of decreasing function value;*

D2. $\mathbf{x}^* \in \operatorname{argmin}_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x})$ *is a minimizer in* $\mathcal{X}$*;*

and under the following assumptions:

A1. f is observed without noise;

A2. f is bounded in $\mathcal{X}$*, i.e.,* $\exists\, C \in \mathbb{R}_{++}$ *s.t.* $|f(\mathbf{x})| < C\ \forall \mathbf{x} \in \mathcal{X}$*;*

A3. CTS-TuRBO (CTS-BAxUS) considers an evaluated point an improvement only if it improves over the current best solution by at least some constant $\gamma \in \mathbb{R}_{++}$*;*

A4. At least one of the minimizers $\mathbf{x}_i^*$ *lies in a continuous region with positive measure;*

A5. The initial points $\{\mathbf{x}_i\}_{i=1}^{n_{\text{init}}}$ *after each trust region restart for CTS-TuRBO are chosen such that* $\forall \delta \in \mathbb{R}_{++}$ *and* $\mathbf{x} \in \mathcal{X}$*,* $\exists\, \nu(\mathbf{x}, \delta) > 0$ *s.t.* $\mathbb{P}(\exists\, i : \|\mathbf{x} - \mathbf{x}_i\| \leq \delta) \geq \nu(\mathbf{x}, \delta)$*, i.e., the probability that at least one point in* $\{\mathbf{x}_i\}_{i=1}^{n_{\text{init}}}$ *ends up in a ball centered at* $\mathbf{x}$ *with radius* δ *is at least* $\nu(\mathbf{x}, \delta)$*;*

A6. The dimension of the search space $d = |\mathcal{X}|$ *is finite;*

Then, CTS-TuRBO and CTS-BAxUS converge to a global optimum $f(\mathbf{x}^*)$ *with probability* 1*.*

Proof. We first show that consistency holds for CTS-TuRBO **(Case 1)**, then use this result to show it also holds for CTS-BAxUS **(Case 2)**.

Case 1 (CTS-TuRBO)**.**

Under the more restrictive assumption of a unique global optimum, it was shown by (Eriksson and Poloczek, 2021) that TuRBO converges with probability 1. We summarize their argument, showing that integration of CTS into TuRBO has no effect on it.

To begin, note that because f is bounded (A2.) and CTS-TuRBO only considers an evaluated point an improvement if it improves over the current best solution by some constant γ (A3.), CTS-TuRBO can only evaluate a finite number of points from a given trust region before failing to improve τ_{fail} times. That is, TR shrinking is guaranteed to occur after a finite number of samples. Since a finite number of shrinking events will trigger a trust region restart, there will be infinite restarts as the number of samples tends to infinity. Each restart involves uniform random sampling over $\mathcal{X}$ that satisfies assumption A5.. Thus, global convergence to a unique global optimum follows from the proof of global convergence for random search (e.g., see (Spall, 2003)).

Finally, as noted by (Papenmeier et al., 2022) (reproduced here for completeness), this argument can be extended to the setting with potentially multiple global optima. To do so, we must establish that CTS-TuRBO generates a sequence of points that adheres to Definition D1. We achieve this by considering the sequence of points of decreasing function values

$$\left\{ \mathbf{x}' \in \underset{\hat{\mathbf{x}} \in \{\mathbf{x}_k\}_{k=1}^{i}}{\operatorname{argmin}} f(\hat{\mathbf{x}}) \right\}_{i=1}^{\infty} \tag{7}$$

where $\{\mathbf{x}_k\}_{k=1}^{i}$ are the observations up to the i-th function evaluation. This sequence, in addition to the fact that CTS-TuRBO samples points with uniform probability on $\mathcal{X}$ upon trust region restarts, satisfies the assumptions of the theorem by (Solis and Wets, 1981), which in turn, along with assumption A4., implies convergence to $f(\mathbf{x}^*)$.

Case 2 (CTS-BAxUS)**.**

First, note that assumption A5. applies only to CTS-TuRBO, preventing us from directly applying **Case 1**. Now, (Papenmeier et al., 2022) have shown that under assumption A6., BAxUS must eventually arrive at an embedding equivalent to the input space, at which point BAxUS behaves in a way equivalent to TuRBO. Likewise, CTS-BAxUS relies on the same mechanics used in BAxUS to initialize and expand its random embeddings — and therefore also reaches an embedding equivalent to the input space after a finite number of function evaluations. Hence, after this transient period, CTS-BAxUS begins and continues to behave like CTS-TuRBO, and **Case 1** applies. □

D Advantage of Adaptive Trust Region Geometry in Low-Dimensional Problems

We found that CTS-TuRBO with spherical trust regions can under-perform compared to TuRBO in low-dimensional optimization problems, such as the 14D Robot Pushing problem (see figure 8). We hypothesize that TuRBO's advantage in this problem could be coming from its ability to modulate its trust region side lengths according to the GP length scales. To test this hypothesis, we ablated TuRBO's adaptive trust region geometry, resulting in a variant of TuRBO with strictly cubic trust regions. The results of this ablation are shown in figure 9.

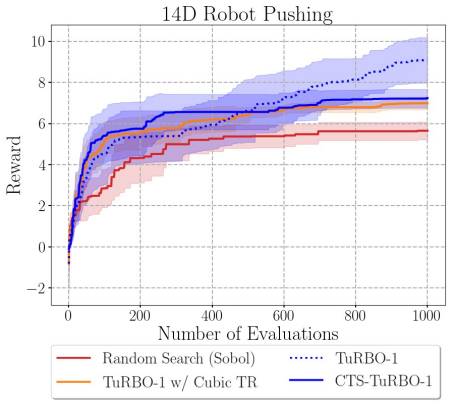


Figure 9: Comparison between TuRBO, CTS-TuRBO, and TuRBO with cubic trust regions on the 14D Robot Pushing problem. Each algorithm was run for ten repetitions. The shaded regions indicate 95% confidence intervals.

Removing TuRBO's adaptive trust region geometry yielded an algorithm with performance similar to that of CTS-TuRBO. This result serves as evidence that this feature may be important to TuRBO's performance on some low-dimensional benchmarks. Experiments on more benchmarks are needed to provide additional evidence. However, this result suggests that ellipsoidal trust regions with adaptive geometry may yield improved performance for CTS-TuRBO. This is left to future work.