We Still Don't Understand High-Dimensional Bayesian Optimization

Colin Doumont*
Tübingen AI Center
c.doumont@uni-tuebingen.de

Donney Fan
University of British Columbia
Vector Institute

Natalie Maus
University of Pennsylvania

Jacob R. Gardner
University of Pennsylvania

Henry Moss
Lancaster University
University of Cambridge

Geoff Pleiss
University of British Columbia
Vector Institute

https://github.com/colmont/linear-bo

Abstract

Existing high-dimensional Bayesian optimization (BO) methods aim to overcome the curse of dimensionality by carefully encoding structural assumptions, from locality to sparsity to smoothness, into the optimization procedure. Surprisingly, we demonstrate that these approaches are outperformed by arguably the simplest method imaginable: Bayesian linear regression. After applying a geometric transformation to avoid boundary-seeking behavior, Gaussian processes with linear kernels match state-of-the-art performance on tasks with 60- to 6,000-dimensional search spaces. Linear models offer numerous advantages over their non-parametric counterparts: they afford closed-form sampling and their computation scales linearly with data, a fact we exploit on molecular optimization tasks with $> 20{,}000$ observations. Coupled with empirical analyses, our results suggest the need to depart from past intuitions about BO methods in high-dimensions.

1 INTRODUCTION

High-dimensional search spaces have historically been a challenge for Bayesian optimization (BO). Classic theoretical results demonstrate that regret increases exponentially with dimensionality (Srinivas et al., 2010; Bull, 2011) and naïve BO implementations can even perform worse than random search on problems exceeding 10 dimensions (Wang et al., 2016; Santoni et al., 2024). The most successful high-dimensional BO algorithms have exploited or imposed structural assumptions on the problem's objective function, such as *additive decompositions* (Kandasamy et al., 2015; Gardner et al., 2017; Mutny and Krause, 2018; Wang et al., 2018), *locality* (Eriksson et al., 2019; Müller et al., 2021; Wu et al., 2023), *geometric warpings* (Oh et al., 2018; Kirschner et al., 2019), or *sparsity* (Eriksson and Jankowiak, 2021; Papenmeier et al., 2022). These approaches achieve better performance than naïve methods and even offer asymptotic reductions in regret under appropriate structural conditions (Kandasamy et al., 2015; Wu et al., 2023). Nevertheless, these algorithms are often complicated, introduce many hyperparameters, and may require many observations for meaningful optimization.

Recently, Hvarfner et al. (2024) and Xu et al. (2025) achieved breakthrough results on high-dimensional problems without any structural assumptions, relying only on Gaussian process priors that favor simple functions as dimensionality D increases. Specifically, after scaling kernel lengthscale hyperparameters proportionally to $\sqrt{D}$ to encourage smoothness, they show that the decades-old "Vanilla" BO recipe—expected improvement using a Gaussian process with a squared-exponential kernel (Močkus, 1974; Jones et al., 1998)—achieves competitive performance across problems up to $D > 6000$. Perhaps most remarkably, they demonstrate empirical success in a seemingly intractable regime, where the observation budget N is

Proceedings of the 29th International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300. Copyright 2026 by the author(s).

*Work done while at the Vector Institute.

on the same order as the search space dimensionality D (which we denote by $N \approx D$). With $N \approx D$, one cannot even build a first-order Taylor approximation around a single point (i.e. a locally linear model), and yet Vanilla BO outperforms methods specifically engineered for high-dimensional problems.

Recognizing that true global optimization is all but impossible when $N \approx D$, we push this smoothness approach to its logical extreme. Specifically, we restrict the surrogate model to only have support for *linear functions*, the smoothest functions under most mathematical definitions. With linear models, we can globally model linear functions or locally model nonlinear functions, but not both. Nevertheless, we hypothesize that the simplicity of these models may be beneficial, particularly in the $N \approx D$ regime. Specifically, if observation budgets cannot support learning beyond locally-linear approximations, we hypothesize that simple Bayesian linear surrogate models are viable alternatives to non-parametric Gaussian processes. Linear models could also prove valuable in the $N \gg D$ setting: their $O(D)$ parametric representation affords closed-form sampling and $O(ND^2)$ computational complexity, whereas their non-degenerate Gaussian process counterparts require $O(N^3)$ computation and do not admit exact pathwise samples.

Standard BO references do not recommend the use of linear models, or equivalently Gaussian processes with linear kernels (Frazier, 2018; Garnett, 2023). Indeed, our own experiments show that linear kernels in their most naïve form perform extremely poorly on most high-dimensional BO tasks (see Figure 2). However, our analyses reveal that this poor performance is largely attributable to boundary-seeking behavior of linear models, which we address through simple geometric modifications to the kernel. Our most significant proposal is to bijectively map the input space (typically a hypercube in $\mathbb{R}^D$) onto a hypersphere in $\mathbb{R}^{D+1}$, yielding provable immunity from boundary-seeking behavior and resulting in a kernel that is a function of cosine similarity between pairs of inputs.

Surprisingly, while standard linear kernels fail to yield meaningful optimization, our spherically-projected linear kernels match or exceed state-of-the-art performance across benchmarks from $D = 60$ to $D > 6000$. Most notably, our sphere-domain linear models excel when $N \gg D$ and exact Gaussian process inference becomes computationally prohibitive, providing a significant performance improvement over existing scalable approaches like stochastic variational Gaussian processes (Hensman et al., 2013; Vakili et al., 2021; Moss et al., 2023). Beyond demonstrating the state-of-the-art performance of our spherically-projected linear kernels, especially in large-N settings, these results further question our understanding of high-dimensional Bayesian optimization. The fact that the simplest possible surrogate model (after correcting for geometric pathologies) can match or outperform all other methods suggests the need for a radical departure from the conventional wisdom and theory in the field.

2 BACKGROUND & RELATED WORK

2.1 Bayesian Optimization

The goal of Bayesian optimization (Garnett, 2023) is to find the global optimum $\mathbf{x}^* \in \mathcal{X}$ of a black-box function $f : \mathcal{X} \to \mathbb{R}$, where f can only be evaluated point-wise and those evaluations are considered "expensive" (e.g. in time or money). Depending on the setting, these evaluations of f can also be noisy, i.e. $f(\mathbf{x}) + \varepsilon_i$ with $\varepsilon_i \sim \mathcal{N}\left(0, \sigma_\varepsilon^2\right)$. Although $\mathcal{X}$ can represent different spaces, the typical space in BO is the unit hypercube, i.e. $\mathcal{X} = [0,1]^D$. In this paper, we consider the *centered* hypercube $[-1,1]^D$, which can easily be mapped to or from $[0,1]^D$.

To find $\mathbf{x}^*$, BO iteratively selects new points at which to evaluate f, where these points are chosen by maximizing an acquisition function $\alpha_t(\cdot)$, i.e. $\mathbf{x}_{t+1} = \arg\max_{\mathbf{x}\in\mathcal{X}} \alpha_t(\mathbf{x})$, which balances exploration and exploitation by relying on a (probabilistic) surrogate model of f, for example a Gaussian process (GP).

Gaussian Processes. GPs define distributions over functions that are specified by a mean function $\mu(\mathbf{x})$ (typically a constant) and a covariance function or kernel $k(\mathbf{x}, \mathbf{x}') : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. GPs can be equivalently viewed as Bayesian linear regression with a (potentially infinite-dimensional) basis expansion, where $k(\mathbf{x}, \mathbf{x}')$ corresponds to the inner product under this featurization. It is common to use kernels that correspond to infinite-dimensional basis expansions, such as the Radial Basis Function (RBF):

$$k_{\text{RBF}}(\mathbf{x}, \mathbf{x}') := \exp\left(-\tfrac{1}{2}\left\|\mathbf{z} - \mathbf{z}'\right\|^2\right),$$
$$\mathbf{z} := \begin{bmatrix} \frac{x_1}{\ell_1} & \cdots & \frac{x_D}{\ell_D} \end{bmatrix}, \quad \mathbf{z}' := \begin{bmatrix} \frac{x'_1}{\ell_1} & \cdots & \frac{x'_D}{\ell_D} \end{bmatrix}.$$

Here, the lengthscales $\ell_i > 0$ are hyperparameters that re-scale the search space $\mathcal{X}$ before applying the kernel. Recent work has shown the necessity of scaling these lengthscales with $\sqrt{D}$, so that the expected norms of the re-scaled inputs $\mathbf{z}$, $\mathbf{z}'$ are independent of dimensionality (Hvarfner et al., 2024; Xu et al., 2025; Papenmeier et al., 2025b).

Degenerate Kernels and Linear Models. While kernels corresponding to infinite basis expansions (e.g. the RBF kernel) often yield universally approximating

functions, kernels corresponding to finite basis expansions have limited representational capacity. In particular, the *linear kernel*, defined as

$$k_{\text{linear}}(\mathbf{x}, \mathbf{x}') := b_0 + b_1 \mathbf{x}^\top \mathbf{x}' \tag{1}$$

for some hyperparameters $b_0, b_1 > 0$, only corresponds to the $(D+1)$-dimensional basis expansion $[\sqrt{b_0}, \sqrt{b_1}x_1, \ldots, \sqrt{b_1}x_D]$. Similarly, polynomial kernels of the form $\left(b_0 + b_1 \mathbf{x}^\top \mathbf{x}'\right)^m$ with $m \in \mathbb{N}$ correspond to $O(D^m)$-dimensional basis expansions. Usually, these kernels are avoided for BO in favor of their universal approximating counterparts (Garnett, 2023).

2.2 High-Dimensional Bayesian Optimization

Classic results show that the regret of BO with universally approximating kernels increases exponentially with D (Srinivas et al., 2010; Bull, 2011), and thus problems with ≥ 20 dimensions can prove challenging for BO (Frazier, 2018). To circumvent this curse of dimensionality, a wide range of methods have been designed specifically for high-dimensional Bayesian optimization (HDBO), often relying on complex algorithms and strong assumptions on the objective function. We keep a more detailed literature review of these methods for Appendix B, but present the main high-level ideas in this paragraph: common approaches either impose structural assumptions on the objective function, such as an additive decomposition over the input features (e.g. Kandasamy et al., 2015) or sparsity over projected features (e.g. Papenmeier et al., 2023), or reconfigure the BO algorithm to perform local rather than global optimization (e.g. Eriksson et al., 2019). All of these algorithms can be seen as approaches towards reducing the complexity of the objective function or search space.

Dimensionality-Based Smoothness. A recent line of work by Hvarfner et al. (2024) demonstrates that neither structural assumptions nor explicitly local algorithms are necessary for successful HDBO. Instead, they propose increasing the kernel lengthscale hyperparameters at a rate of $\sqrt{D}$—equivalent to rescaling the $\mathcal{X}$ hypercube by $1/\sqrt{D}$—to yield an objective function prior that favors smoothness as D increases. This "Vanilla BO" recipe, alongside contemporaneous methods with similar lengthscale scalings (Xu et al., 2025; Papenmeier et al., 2025b), significantly outperforms the methods described above on benchmarks with up to $D = 6000$ dimensions and as few as $N = 1000$ observations.

Geometric Transformations. Few works propose geometric warpings of the inputs beyond element-wise scaling through lengthscales. A notable exception is `BOCK` (Oh et al., 2018), who, like us, bijectively map the inputs into a different space before applying a kernel. We note several differences between this method and ours. Superficially, our proposed method maps inputs to a hypersphere, whereas `BOCK` maps to a hypercylinder. More substantially, our geometric warping is motivated to obtain meaningful performance from linear kernels, whereas `BOCK` applies this transformation to kernels with high representational capacity.

3 METHOD

The recent works of Hvarfner et al. (2024) and Xu et al. (2025) have demonstrated the necessity of scaling lengthscales with dimensionality, effectively creating a prior that favors increasing smoothness as dimensionality increases. In this work, we investigate if further improvements are possible if we take this smoothness idea to its logical extreme. Importantly, we cannot simply consider a faster rate of lengthscale scaling, as this would lead to $\|\mathbf{x}/\ell - \mathbf{x}'/\ell\| \to 0$ as $D \to \infty$, effectively leading to a prior over constant functions. Instead, we turn to what is (under most definitions) the smoothest model of non-constant functions: Gaussian processes with linear kernels.

The idea of using linear kernels (i.e. Bayesian linear regression) for HDBO may appear surprising, especially given that standard BO references discourage its use (Shahriari et al., 2015; Garnett, 2023): the black-box objective functions modeled during BO are often of unknown complexity, necessitating non-parametric and universal approximating surrogates rather than simple parametric models. However, we argue that the limited fidelity of linear models is not a limitation in $N \approx D$ settings, where there are barely enough observations to "fill up" the capacity of linear models. Instead, we hypothesize that linear kernels can be competitive for HDBO tasks after accounting for pathologies in the optimization geometry.

Below, we propose a modified linear kernel for Bayesian optimization. Like the standard linear kernel, our proposed kernel is rank $O(D)$ and yields an extremely simple prior over functions. However, as we will demonstrate, our kernel significantly surpasses the BO performance of the standard linear kernel and even rivals the most sophisticated methods.

3.1 Proposed Linear Model

We propose the following linear kernel for GPs (equivalent to Bayesian linear regression):

$$k_{\text{linear}}\left(\mathbf{x}, \mathbf{x}'\right) := b_0 + b_1 \, P\left(\mathbf{z}\right)^\top P\left(\mathbf{z}'\right), \tag{2}$$

where $b_i > 0$ are hyperparameters, $P : \mathbb{R}^D \to \mathbb{S}^D$ is a bijective mapping onto the unit sphere, and $\mathbf{z}, \mathbf{z}'$ are

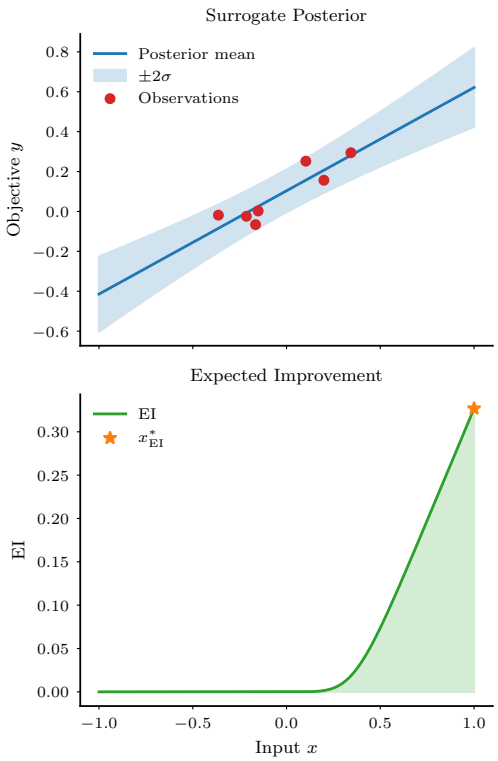

Figure 1: **Boundary-seeking behavior of linear models (Theorem 1).** *(Top)* A Bayesian linear model's posterior mean and variance strictly increase away from the origin, i.e. $x = 0$. *(Bottom)* This causes most well-known acquisition functions (e.g. expected improvement) to be maximized exactly at the boundaries, discarding most of the search space. We resolve this pathology via a spherical projection of the inputs, allowing simple linear models to achieve state-of-the-art performance on high-dimensional problems.

scaled versions of $\mathbf{x}, \mathbf{x}'$, i.e.

$$\mathbf{z} := \frac{1}{a}\begin{bmatrix}\frac{x_1}{\ell_1} & \cdots & \frac{x_D}{\ell_D}\end{bmatrix}, \quad \mathbf{z}' := \frac{1}{a}\begin{bmatrix}\frac{x'_1}{\ell_1} & \cdots & \frac{x'_D}{\ell_D}\end{bmatrix} \tag{3}$$

for hyperparameters $a \in \mathbb{R}^+$, $\boldsymbol{\ell} \in \mathbb{R}^D$. Of all the modifications, the spherical mapping is the most impactful (confirmed by ablations in Appendix D.3). We now define each of these changes more rigorously.

Spherical Mapping. Linear models induce a boundary-seeking behavior during optimization, which we hypothesize is the primary factor affecting their performance. Specifically, the posterior mean and variance of linear models increase away from the center of the hypercube, resulting in maximal acquisition values on the search space boundary (implicitly discarding most of $\mathcal{X}$). We formalize this idea in the following theorem (see Appendix A.1 for a proof), and provide a visualization for $D = 1$ in Figure 1.

Theorem 1. *For acquisition functions increasing in posterior mean and variance (e.g. expected improvement), Bayesian linear models will maximize acquisition on the boundary of the search space. That is, for* $\mathbf{x}_{t+1} = \arg\max_{\mathbf{x}\in[-1,1]^D} \alpha_t(\mathbf{x})$ *at any timestep* t*, we have that* $\mathbf{x}_{t+1}$ *contains at least one dimension equal to* -1 *or* 1 *(i.e.* $\|\mathbf{x}_{t+1}\|_\infty = 1$*).*

However, when points lie on the unit hypersphere $\mathbb{S}^D \subset \mathbb{R}^{D+1}$ instead of the hypercube $[-1, 1]^D \subset \mathbb{R}^D$, the acquisition function cannot increase by scaling inputs, since all points are equally far from the origin (i.e. on a sphere) and thus Theorem 1 does not apply (see counterexample in Appendix A.2). Thus, we propose mapping our inputs $\mathbf{x}$ from $\mathcal{X}$ to a unit sphere (via P in Equation 2) before applying the linear kernel.

While numerous mappings from $\mathbb{R}^D \to \mathbb{S}^D$ exist, we choose P to be the *inverse stereographic projection*:

$$P(\mathbf{z}) := \frac{1}{||\mathbf{z}||^2+1}\begin{bmatrix}2z_1 & \cdots & 2z_D & ||\mathbf{z}||^2 - 1\end{bmatrix}. \tag{4}$$

As can be seen from the equation, the projection scales all dimensions of the original D-dimensional vector by their norm (with some constants), and only introduces one additional dimension, resulting in $D + 1$ features for our linear model. This mapping possesses a number of attractive characteristics: it is smooth, bijective over $\mathbb{R}^D$, and conformal, meaning that it preserves angles at which curves meet (Hilbert and Cohn-Vossen, 2021). Most notably, it is a (near) identity mapping for unit norm $\mathbf{z}$ (i.e. $P(\mathbf{z}) = [\mathbf{z}, 0]$ when $\|\mathbf{z}\| = 1$), a fact we will explore in depth in Section 5.

Decoupled Magnitude and Direction for ℓ_i. To ensure a meaningful distance is maintained between all points as the dimensionality D grows, recent methods have scaled the lengthscales ℓ_i by $\sqrt{D}$, either through the hyperprior (Hvarfner et al., 2024; Xu et al., 2025) or the initialization value (Papenmeier et al., 2025b). As a simplified alternative to impose this scaling, we decouple the magnitude and direction of the lengthscale vector. Specifically, we introduce a *global* lengthscale a in Equation (3), initialized to be $O(\sqrt{D})$, which we multiply by an (unconstrained) vector $\boldsymbol{\ell}$ that models varying sensitivities of each dimension. In the context of our kernel, the $O(\sqrt{D})$ scaling on a ensures that the entries of $\mathbf{z}$ are $O(1/\sqrt{D})$, ensuring our scaled points have $O(1)$ norm before projecting them onto the unit sphere. We place a log-normal prior on the vector $\boldsymbol{\ell}$, namely $\ell_i \sim \mathcal{LN}\left(\sqrt{2}, \sqrt{3}\right)$, though we find that performance is largely unaffected by this choice (see Appendix D.1).

No Implicit Outputscale. A useful property of RBF kernels is that they are bounded above, i.e.

$k(\mathbf{x}, \mathbf{x}') \leq 1$, achieving the maximum if and only if $\mathbf{x} = \mathbf{x}'$. The latter property does not hold for standard linear kernels; however, it is naturally enforced for our spherically-projected inputs. Since P bijectively maps to the unit hypersphere, we have $P(\mathbf{z})^\top P(\mathbf{z}') \leq 1$ and thus $k_{\text{linear}}(\mathbf{x}, \mathbf{x}') \leq b_0 + b_1$ with equality if and only if $\mathbf{x} = \mathbf{x}'$. Inspired by Hvarfner et al. (2024), we further enforce that $b_0 + b_1 = 1$, removing any implicit outputscale that could affect optimization. Since $b_0, b_1 > 0$, we enforce this constraint by learning unconstrained parameters, and then mapping them to the simplex through the softmax function.

3.2 Polynomial Extension

We can naturally extend the logic of our linear kernel to higher-order polynomial kernels via:

$$k_{\text{poly}}\left(\mathbf{x}, \mathbf{x}'\right) := \sum_{i=0}^{m} b_i \left[P\left(\mathbf{z}\right)^\top P\left(\mathbf{z}'\right)\right]^i, \quad (5)$$

where $m \in \mathbb{N}$ indicates the polynomial order. We again constrain the coefficients b_i to the simplex through a softmax function. Kernels of this form are trivially positive semi-definite through a standard application of kernel composition rules (e.g. Genton, 2001).

Equation (5) can approximate many kernels on $P(\mathbf{z}), P(\mathbf{z}') \in \mathbb{S}^D$ to an arbitrary degree of precision. Schoenberg (1942) proves that any dot-product kernel on the unit sphere (i.e. where the kernel is a function of $P(\mathbf{z}), P(\mathbf{z}')$) takes the form of Equation (5) for some value of m. Moreover, any kernel that is a function of Euclidean distance (e.g. the Matérn or rational quadratic kernels) can be approximated by Equation (5) for inputs $P\left(\mathbf{z}\right), P\left(\mathbf{z}'\right) \in \mathbb{S}^D$ (see Appendix A.3). For example, Appendix A.4 shows that the RBF kernel restricted to the unit sphere has the Taylor expansion

$$k_{\text{RBF}}\left(P\left(\mathbf{z}\right), P\left(\mathbf{z}'\right)\right) = \sum_{i=0}^{\infty} \frac{1}{i!\, e} \left[P\left(\mathbf{z}\right)^\top P\left(\mathbf{z}'\right)\right]^i. \quad (6)$$

Therefore, increasing m, our proposed model can approximate the RBF kernel (on a hypersphere) with any arbitrary precision. Surprisingly, as will become clear in Section 4.3, there is rarely a need to go beyond $m = 1$ to match state-of-the-art HDBO performance.

3.3 Advantages of Linear Models

Before considering optimization performance, it is worth remarking on the practical benefits afforded by our kernel (or any kernel corresponding to an $O(D)$ feature expansion), where these benefits cannot be obtained by standard (e.g. RBF or Matérn) kernels.

Computational Scalability. Since GPs with linear kernels correspond to an $O(D)$ feature map, we can perform exact posterior inference over the $O(D)$-dimensional parameters in $O(ND^2)$ time (Williams and Rasmussen, 2006). In contrast, RBF and Matérn kernels, which correspond to infinite-dimensional basis expansions, require $O(N^3)$ computation for exact posterior inference. As a result, linear kernels can be used to scale to optimization problems where $N \gg D$ (e.g. molecular optimization in Section 4.2), whereas standard kernels require posterior approximations.

Exact Thompson Sampling. Thompson sampling (Thompson, 1933), a common acquisition function for high-dimensional problems (e.g. Eriksson et al., 2019) and parallel acquisitions (e.g. Hernández-Lobato et al., 2017), requires sampling a realization of the GP posterior. While standard kernels necessitate approximations or discretizations of $\mathcal{X}$ for sampling, linear kernels yield exact posterior function samples. Functions under our linear model posterior can be represented as $f(\mathbf{x}) = \theta_0 + \boldsymbol{\theta}^\top P(\mathbf{z})$, where θ_0 and $\boldsymbol{\theta}$ are (finite-dimensional) Gaussian random variables. Hence, we can exactly sample functions by sampling θ_0 and $\boldsymbol{\theta}$.

4 EXPERIMENTS

To showcase the performance and scalability of our proposed linear kernel, we evaluate it on a range of optimization problems (both $N \approx D$ and $N \gg D$). Moreover, we ablate the design choices that differentiate our kernel from the standard linear kernel.

Benchmarks. To test performance in the $N \approx D$ setting, we evaluate our method on a range of standard benchmarks from the HDBO literature ranging from $D = 60$ all the way to $D = 6392$: `Rover` (60D; Wang et al., 2018), `MOPTA08` (124D; Eriksson and Jankowiak, 2021), `Lasso-DNA` (180D; Nardi et al., 2022), `SVM` (388D; Eriksson and Jankowiak, 2021), `Ant` (888D; Wang et al., 2020), and `Humanoid` (6392D; Wang et al., 2020). We evaluate these tasks under a budget of $N = 1000$ observations. For the $N \gg D$ setting, we benchmark on multiple molecular tasks from the `GuacaMol` benchmark (Brown et al., 2019), which typically require large optimization budgets for meaningful progress. We perform BO in the 256D latent space of a pre-trained variational auto-encoder (Maus et al., 2022, 2024) with a $N = 20{,}000$ budget.

Baselines. For the $N \approx D$ tasks, we compare against popular HDBO methods: `BOCK` (Oh et al., 2018), `TuRBO` (Eriksson et al., 2019), `SAASBO` (Eriksson and Jankowiak, 2021), `BAxUS` (Papenmeier et al., 2022), and "Vanilla BO" (Hvarfner et al., 2024), as

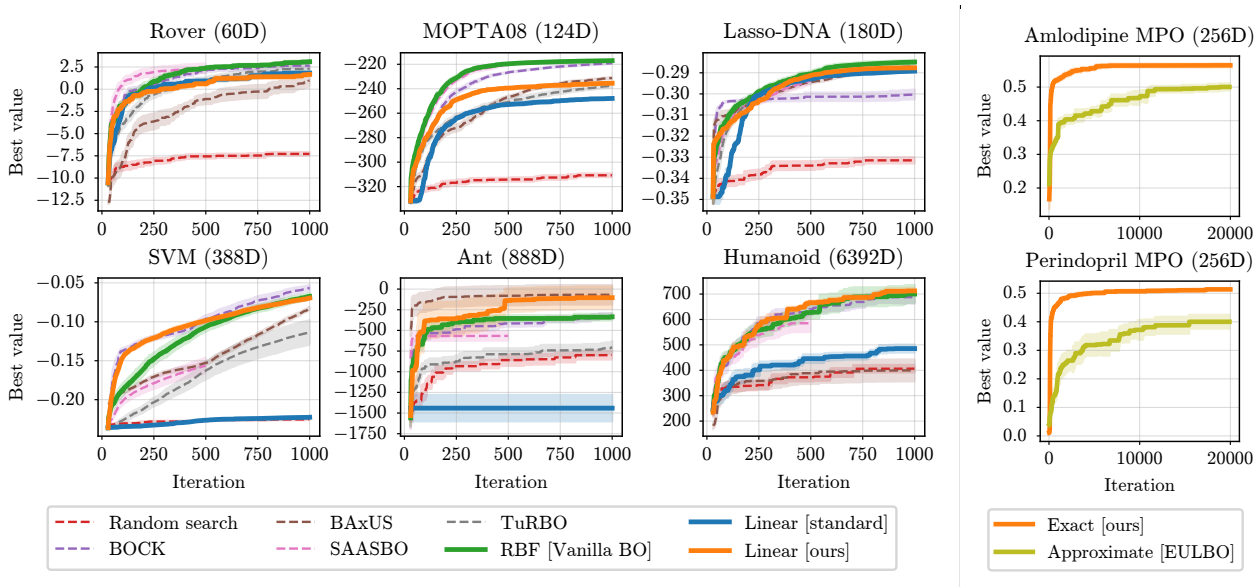

Figure 2: **Our linear kernel on spherically-mapped inputs matches state-of-the-art high-dimensional BO performance.** (Left): benchmarks with $N \approx D$ evaluation budgets. While standard linear kernels can fail to make any optimization progress, our modified kernel matches or exceeds competitive methods. (Right): benchmarks with $N \gg D$. The natural scalability of linear kernels, coupled with the improved optimization performance afforded by our spherical mapping, yields new state-of-the-art results on large-N tasks.

well as a standard linear kernel as in Equation (1). The $N \gg D$ tasks necessitate the use of scalable GP approximations (except in the case of linear kernels), and therefore we compare against the `EULBO` method (Maus et al., 2024), which relies on variational approximations. For the "random search" baseline, we perform quasi-random Sobol sampling (Sobol, 1967) of $\mathcal{X}$.

Experimental Set-Up. Our set-up largely follows Hvarfner et al. (2024): we use the `LogEI` acquisition function (Ament et al., 2023), initialize all methods with 30 quasi-random observations (100 points for $N \gg D$ tasks), and follow their other modeling protocols. We run each method for at least 10 different random seeds, and the plots in this paper display the mean and standard error of the mean (with respect to the seeds) as a solid/dotted line and shaded area, respectively. Details of our experimental set-up can be found in Appendix C.

4.1 $N \approx D$ Optimization Problems

Figure 2 (left) demonstrates optimization performance of all methods on $N \approx D$ tasks. As expected, the standard linear kernel fails to optimize many benchmarks, though it obtains decent performance on the lower-dimensional tasks. In contrast, our modified linear model matches the current state-of-the-art on all $N \approx D$ benchmarks. In fact, the optimization trajectories from our linear kernel and Vanilla BO are statistically indistinguishable on the `Lasso-DNA` and `Humanoid` benchmarks. We note the stark performance difference between the modified and standard linear kernels on the `SVM` and `Ant` datasets. Moreover, in Appendix E.1, we compare against Vanilla BO on a number of latent-space molecular tasks with a similar observation budget. Spherical linear kernels strongly outperform Vanilla BO on all these benchmarks, potentially suggesting our method is particularly advantageous for latent-space BO.

4.2 $N \gg D$ Optimization Problems

For $N \gg D$ molecular tasks, we observe significant performance gains over the scalable `EULBO` method (Figure 2, right). We hypothesize that our linear kernel, unencumbered by the need for approximations, is better suited for large-N tasks, despite its lack of representational capacity.

4.3 Ablation Studies

Higher-Order Polynomials. In Figure 3 (and its extended version, Figure 14, in Appendix E.2), we replicate the $N \approx D$ benchmark results for order-m polynomial kernels of the form in Equation (5). (Recall that our proposed linear kernel is the special case of this form for $m = 1$.) Surprisingly, the polynomial order has almost no effect on optimization performance, with our linear kernel matching the $m = 5^{\text{th}}$ order

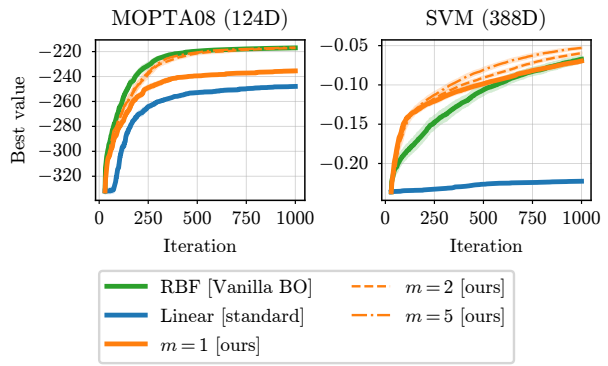

Figure 3: Extension of our spherical linear kernel to order-m polynomial kernels. Higher-order polynomials do not improve much upon our linear model ($m = 1$). See Figure 14 for additional datasets.

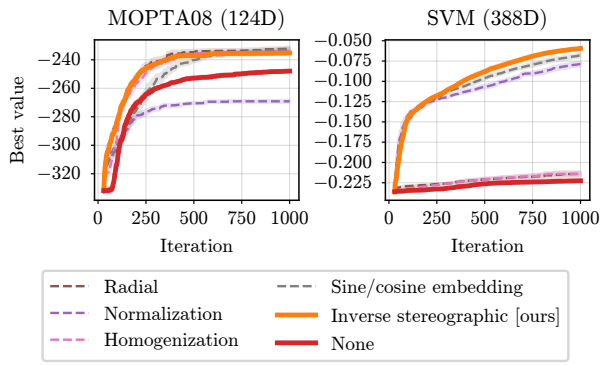

Figure 4: Ablation over the spherical mapping function used by our modified linear kernel. While most spherical mappings improve upon the unmodified inputs (red line), the inverse stereographic projection (orange line) outperforms all other mappings. See Figure 15 for additional datasets.

polynomial on nearly all benchmarks. From a practical perspective, higher order polynomials are net detrimental, as they lose scalability and exact Thompson sampling (the $O(D^m)$ feature representation of these kernels is too large to afford these benefits), without improving optimization performance.

Choice of Spherical Mapping. Our biggest modification to the standard linear kernel is the spherical mapping (see Section 3.1), i.e. the inverse stereographic projection defined in Equation (4). In Figure 4 (and its extended version, Figure 15, in Appendix E.3), we compare a range of other common mappings from $\mathcal{X}$ to the hypersphere (see Appendix E.3 for descriptions). We also compare against the case of no projection (None), but where we keep our other proposed modifications to the linear kernel (decoupled lengthscales and no implicit outputscale). While we can conclude that a spherical projection is necessary to improve performance, the choice of spherical projection itself can also have a drastic impact on performance.

We note that the inverse stereographic projection, which consistently outperforms all other mappings, is the only mapping that does not modify unit-norm inputs (i.e. $P(\mathbf{z}) = [\mathbf{z}, 0]$ for all $\|\mathbf{z}\| = 1$). We hypothesize that this property may be instrumental in its performance (see Section 5). Interestingly, while a spherical mapping is crucial for linear BO, it has almost no effect for non-parametric models. In Appendix D.2, we demonstrate that Vanilla BO performance is largely unchanged under the inverse stereographic projection.

Other Ablations. In Appendix D.3, we ablate over our other modifications (decoupled lengthscale, centering of the search space, implicit outputscale, etc.), showing that our decoupled parameterization makes our kernel less sensitive to any prior placed on the lengthscale vector $\boldsymbol{\ell}$. In contrast, priors on standard (coupled) lengthscales have a significant impact on BO performance (Hvarfner et al., 2024).

5 ANALYSIS

By conventional wisdom, linear models should not work as well as they do in Section 4. Their success challenges our understanding of what makes effective high-dimensional surrogate models. While our analysis cannot resolve this puzzle, it reveals a tension that may be central to explaining their efficacy. Reconciling the following contradictory observations highlights the non-intuitive nature of HDBO, suggesting that our understanding is fundamentally incomplete:

1. after applying our spherical mapping, the linear kernel exhibits BO behavior that resembles more traditional non-parametric GPs; and yet
2. the spherical mapping is geometrically minor for most points in high dimensional spaces, preserving local structure except at the boundary.

5.1 Spherical Linear Kernels Avoid the Pathologies of Standard Linear Kernels

Standard linear kernels suffer from a pathological limitation in BO: as we prove in Section 3.1, they exclusively acquire points on the boundary of $\mathcal{X}$, implicitly discarding the interior of the search space. We empirically validate this in Figure 5 (left), which plots the percentage of entries of $\mathbf{x}_t$ equal to ± 1 for each BO iteration t under various kernels (0% indicates an interior

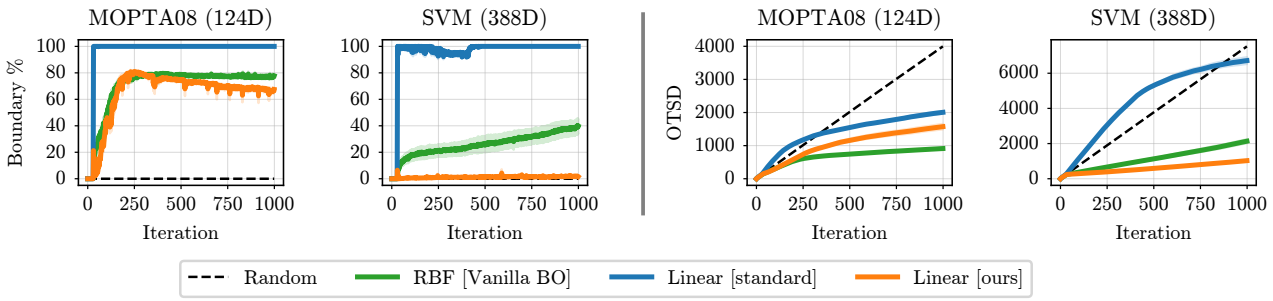

Figure 5: **Differences in the acquired inputs of standard versus modified linear models.** (Left): the "boundary %" depicts, for each acquisition $\mathbf{x}_t$, how many dimensions of the vector lie on the boundary of $\mathcal{X}$ (i.e. ± 1). Standard linear models lead to acquisitions with points nearly in the corners of the hypercube (i.e. 100% of dimensions on the boundary). Our linear model acquires non-corner points ($\approx 75\%$ boundary on `MOPTA08`) and interior points ($\approx 0\%$ boundary on `SVM`). See Figure 16, Appendix E.4 for additional datasets. (Right): length of the shortest path connecting all acquired observations, or Observation Traveling Salesman Distance (OTSD). The corner-searching behavior of the standard linear model leads to acquisitions that are spread out over $\mathcal{X}$, whereas our modified linear model yields locality that is comparable to RBF-based models. See Figure 17, Appendix E.5 for additional datasets.

point and 100% indicates a corner). On the `MOPTA08` and `SVM` tasks, we observe that the standard linear kernel acquires points with 100% of dimensions equal to ± 1, i.e. corners of $\mathcal{X}$. This represents an even more extreme pathology than simple boundary-seeking: the model restricts itself to the 2^D corners, a measure-zero subset of both the boundary and the search space (i.e. a finite instead of infinite set of points).

In contrast, RBF kernels acquire points with $\leq 50\%$ of entries equal to ± 1. While these points still lie on the boundary, they are less extreme and cover a broader region of $\mathcal{X}$. Our spherically-mapped linear kernel exhibits similar behavior to the RBF kernel, and even yields points entirely in the interior (0% of ± 1 elements) on the `SVM` task. (See Appendix E.4 for plots on additional datasets.) Again, the spherical mapping prevents the monotonic growth in posterior statistics that drives standard linear models toward extremal points (see counterexample in Appendix A.2).

The Observation Traveling Salesman Distance (OTSD; Papenmeier et al., 2025a) measures the minimum path length connecting all acquired points, where lower OTSD values suggest more local search. In Figure 5 (right), standard linear kernels yield large OTSD on `MOPTA08` and `SVM`—often exceeding random search—because corner points (i.e. 100% of dimensions on the boundary) are maximally far apart. In contrast, both our spherically-projected linear kernel and RBF kernels yield similarly low OTSD values, suggesting more local search strategies. (See Appendix E.5 for plots on additional datasets.) Collectively, these results provide preliminary evidence that spherically-mapped linear kernels exhibit exploration patterns more similar to RBF kernels than to their unmapped linear counterparts, though complete characterization remains future work.

5.2 Spherical and Standard Linear Kernels are Asymptotically The Same

While the spherical mapping $P : \mathbb{R}^D \to \mathbb{S}^D$ is crucial for avoiding boundary-seeking pathology, it is a relatively minor transformation for most points in $\mathcal{X}$. The inverse stereographic projection is an identity mapping for unit-norm inputs: $P(\mathbf{z}) = [\mathbf{z}, 0]$ for all $\|\mathbf{z}\| = 1$. When lengthscales are set to $\sqrt{3/D}$ as suggested by recent work (Hvarfner et al., 2024; Xu et al., 2025), any $\mathbf{x}$ drawn uniformly from $\mathcal{X}$ yields a scaled vector $\mathbf{z} = [x_1/\ell_1, \ldots, x_D/\ell_D]$ with $\|\mathbf{z}\| \to 1$ as $D \to \infty$—a direct consequence of the law of large numbers (see Appendix A.5). Thus, for most points in $\mathcal{X}$, the spherically-mapped linear model behaves nearly identically to a standard linear model and has no additional representational capacity.

Moreover, we emphasize that the inverse stereographic projection transforms the D-dimensional function space into a $(D+1)$-dimensional function space, where this one additional dimension represents less than a 1% increase in D for most considered benchmarks. In contrast, polynomial kernels rely on a $\mathcal{O}(D^m)$-dimensional function space at a minimum, and RBF or Matérn kernels rely on an ∞-dimensional function space. As a result, although our linear model is *technically* not linear in $\mathbf{x}$ (though it is in $P(\mathbf{x})$), we argue it is *approximately* linear in $\mathbf{x}$.

To validate that the spherical mapping does not mean-

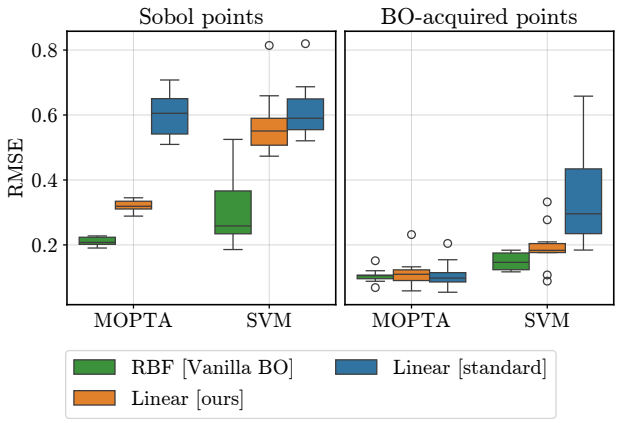

Figure 6: **Spherical mappings affect BO performance, but not supervised regression performance.** (Left): predictive RMSE of GPs trained and tested on *quasi-random data*. The standard and modified linear models make worse predictions than the RBF model. (Right): predictive RMSE on *adaptively-chosen BO acquisitions*. Unlike with random data, linear models match the ability of RBF models for predictions at future BO acquisitions from prior ones. See Figure 18, Appendix E.6 for additional benchmarks.

ingfully increase model expressiveness on a typical regression task, we fit both models to 400 points generated by a Sobol sequence over $\mathcal{X}$ and evaluate predictions on 100 held-out Sobol points (concentrated around unit norm after scaling). In Figure 6 (left), both the spherically-mapped and standard linear models achieve nearly identical predictive RMSE on most benchmarks, while the RBF model generally achieves lower error due to its higher representational capacity (see Figure 18, Appendix E.6 for results on other benchmarks).

5.3 Model Expressiveness Does Not Predict BO Performance

This similarity on standard regression tasks makes the performance gap in BO (see Figure 2) all the more puzzling: if the spherical and standard linear models have similar predictive capabilities, why do spherical mappings improve optimization so dramatically? The answer is that model expressiveness and predictive accuracy on *random test points* do not necessarily translate to optimization performance with *adaptively chosen data*.

To further illustrate this point, Figure 6 (right) repeats the above experiment on adaptively chosen BO acquisitions rather than Sobol points. We fit models to 400 points from a BO trajectory (generated by our spherical linear model) and predict the objective values at the next 100 acquired points. In Figure 6 (right) and Figure 18 (right), all three models—spherical linear, standard linear, and RBF—achieve similar predictive accuracy, despite their vastly different performance as BO surrogates (see Figure 18, right, in Appendix E.6 for results on other benchmarks). These results contrast the supervised learning setting where RBF substantially outperforms linear models on a number of problems.

Why are spherical linear models able to accurately predict future (adaptive) acquisitions but not random test data? One possible explanation is that acquisition functions like EI tend to select points locally (Garnett, 2023), where objectives may be well-approximated by linear models (i.e. a first-order Taylor approximation) even when globally complex. However, fully characterizing which properties actually matter for HDBO success—beyond standard notions of expressiveness or generalization—remains an important open question.

6 CONCLUSION

In this work, we show that linear models—arguably the simplest of all models—match state-of-the-art performance on high-dimensional BO tasks after mapping inputs to the unit sphere. From a **practical perspective**, our modified linear kernel also provides state-of-the-art performance on large-N tasks and promising performance on latent-space tasks. We thus recommend its adoption in these settings. From a **theoretical perspective**, our findings call into question long-standing intuitions and practices. The fact that linear kernels, largely dismissed by the BO community, achieve such strong performance provides a compelling case against model complexity in high-dimensional settings. We hope that our initial analysis provides the seeds for radically new theories that unlock a new and more robust understanding of high-dimensional BO.

Acknowledgments

Resources used in preparing this research were provided, in part, by the Province of Ontario, the Government of Canada through CIFAR, and companies sponsoring the Vector Institute. DF and GP acknowledge the support of the Natural Sciences and Engineering Research Council and the Social Sciences and Humanities Research Council of Canada (NSERC: RGPIN-2024-06405, NFRFE-2024-00830). NM was supported by NSF Graduate Research Fellowship (NSF GRFP). JRG was supported by NSF grants DBI-2400135 and IIS-2145644. HM was supported by Schmidt Sciences and Research England under the Expanding Excellence in England (E3) funding stream. GP is supported by the Canada CIFAR AI Chairs program.

References

Ament, S., Daulton, S., Eriksson, D., Balandat, M., and Bakshy, E. (2023). Unexpected improvements to expected improvement for Bayesian optimization. *Advances in Neural Information Processing Systems*. Cited on page 6.

Balandat, M., Karrer, B., Jiang, D., Daulton, S., Letham, B., Wilson, A. G., and Bakshy, E. (2020). BoTorch: A framework for efficient Monte-Carlo Bayesian optimization. *Advances in Neural Information Processing Systems*. Cited on page 16.

Brown, N., Fiscato, M., Segler, M. H., and Vaucher, A. C. (2019). GuacaMol: benchmarking models for de novo molecular design. *Journal of Chemical Information and Modeling*. Cited on pages 5, 20, and 21.

Bull, A. D. (2011). Convergence rates of efficient global optimization algorithms. *Journal of Machine Learning Research*. Cited on pages 1 and 3.

Eriksson, D. and Jankowiak, M. (2021). High-dimensional Bayesian optimization with sparse axis-aligned subspaces. In *Conference on Uncertainty in Uncertainty in Artificial Intelligence*. Cited on pages 1, 5, and 16.

Eriksson, D., Pearce, M., Gardner, J., Turner, R. D., and Poloczek, M. (2019). Scalable global optimization via local Bayesian optimization. *Advances in Neural Information Processing Systems*. Cited on pages 1, 3, 5, and 16.

Fan, Z., Wang, W., Ng, S. H., and Hu, Q. (2024). Minimizing UCB: a better local search strategy in local Bayesian optimization. *Advances in Neural Information Processing Systems*. Cited on page 16.

Frazier, P. I. (2018). Bayesian optimization. In *Recent Advances in Optimization and Modeling of Contemporary Problems*. Informs. Cited on pages 2 and 3.

Gardner, J., Guo, C., Weinberger, K., Garnett, R., and Grosse, R. (2017). Discovering and exploiting additive structure for Bayesian optimization. In *International Conference on Artificial Intelligence and Statistics*. Cited on page 1.

Gardner, J., Pleiss, G., Weinberger, K. Q., Bindel, D., and Wilson, A. G. (2018). GPyTorch: Black-box matrix-matrix Gaussian process inference with GPU acceleration. *Advances in Neural Information Processing Systems*. Cited on page 16.

Garnett, R. (2023). *Bayesian optimization*. Cambridge University Press. Cited on pages 2, 3, and 9.

Genton, M. G. (2001). Classes of kernels for machine learning: a statistics perspective. *Journal of Machine Learning Research*. Cited on page 5.

Han, E., Arora, I., and Scarlett, J. (2021). High-dimensional Bayesian optimization via tree-structured additive models. In *Proceedings of the AAAI Conference on Artificial Intelligence*. Cited on page 16.

Hartman, J. K. (1973). Some experiments in global optimization. *Naval Research Logistics Quarterly*. Cited on page 18.

Hellsten, E., Hvarfner, C., Papenmeier, L., and Nardi, L. (2025). Leveraging axis-aligned subspaces for high-dimensional Bayesian optimization with group testing. *arXiv preprint arXiv:2504.06111*. Cited on page 16.

Hensman, J., Fusi, N., and Lawrence, N. D. (2013). Gaussian processes for big data. In *Conference on Uncertainty in Artificial Intelligence*. Cited on page 2.

Hernández-Lobato, J. M., Requeima, J., Pyzer-Knapp, E. O., and Aspuru-Guzik, A. (2017). Parallel and distributed Thompson sampling for large-scale accelerated exploration of chemical space. In *International Conference on Machine Learning*. Cited on page 5.

Hilbert, D. and Cohn-Vossen, S. (2021). *Geometry and the Imagination*, volume 87. American Mathematical Soc. Cited on page 4.

Hvarfner, C., Hellsten, E. O., and Nardi, L. (2024). Vanilla Bayesian optimization performs great in high dimensions. In *International Conference on Machine Learning*. Cited on pages 1, 2, 3, 4, 5, 6, 7, 8, 16, 17, 18, and 20.

Jones, D. R., Schonlau, M., and Welch, W. J. (1998). Efficient global optimization of expensive black-box functions. *Journal of Global optimization*. Cited on page 1.

Kandasamy, K., Schneider, J., and Póczos, B. (2015). High dimensional Bayesian optimisation and bandits via additive models. In *International Conference on Machine Learning*. Cited on pages 1, 3, and 16.

Kirschner, J., Mutny, M., Hiller, N., Ischebeck, R., and Krause, A. (2019). Adaptive and safe Bayesian optimization in high dimensions via one-dimensional subspaces. In *International Conference on Machine Learning*. Cited on page 1.

Letham, B., Calandra, R., Rai, A., and Bakshy, E. (2020). Re-examining linear embeddings for high-dimensional Bayesian optimization. *Advances in Neural Information Processing Systems*. Cited on page 16.

Levy, A. V. and Montalvo, A. (1985). The tunneling algorithm for the global minimization of functions.

SIAM Journal on Scientific and Statistical Computing. Cited on page 18.

Li, C., Gupta, S., Rana, S., Nguyen, V., Venkatesh, S., and Shilton, A. (2017). High dimensional Bayesian optimization using dropout. In *International Joint Conference on Artificial Intelligence*. Cited on page 16.

Maus, N., Jones, H., Moore, J., Kusner, M. J., Bradshaw, J., and Gardner, J. (2022). Local latent space Bayesian optimization over structured inputs. *Advances in Neural Information Processing Systems*. Cited on pages 5, 16, 20, and 21.

Maus, N., Kim, K., Pleiss, G., Eriksson, D., Cunningham, J. P., and Gardner, J. R. (2024). Approximation-aware Bayesian optimization. *Advances in Neural Information Processing Systems*. Cited on pages 5, 6, and 16.

Močkus, J. (1974). On Bayesian methods for seeking the extremum. In *IFIP Technical Conference on Optimization Techniques*. Cited on page 1.

Moriconi, R., Deisenroth, M. P., and Sesh Kumar, K. (2020). High-dimensional Bayesian optimization using low-dimensional feature spaces. *Machine Learning*. Cited on page 16.

Moss, H. B., Ober, S. W., and Picheny, V. (2023). Inducing point allocation for sparse Gaussian processes in high-throughput Bayesian optimisation. In *International Conference on Artificial Intelligence and Statistics*. Cited on page 2.

Müller, S., von Rohr, A., and Trimpe, S. (2021). Local policy search with Bayesian optimization. *Advances in Neural Information Processing Systems*. Cited on pages 1 and 16.

Mutny, M. and Krause, A. (2018). Efficient high dimensional Bayesian optimization with additivity and quadrature Fourier features. *Advances in Neural Information Processing Systems*, 31. Cited on page 1.

Nardi, L., Gramfort, A., Salmon, J., and Sehic, K. (2022). LassoBench: A high-dimensional hyperparameter optimization benchmark suite for lasso. In *International Conference on Automated Machine Learning*. Cited on page 5.

Nayebi, A., Munteanu, A., and Poloczek, M. (2019). A framework for Bayesian optimization in embedded subspaces. In *International Conference on Machine Learning*. Cited on page 16.

Nguyen, Q., Wu, K., Gardner, J., and Garnett, R. (2022). Local Bayesian optimization via maximizing probability of descent. *Advances in Neural Information Processing Systems*. Cited on page 16.

Oh, C., Gavves, E., and Welling, M. (2018). BOCK: Bayesian optimization with cylindrical kernels. In *International Conference on Machine Learning*. Cited on pages 1, 3, 5, and 16.

Papenmeier, L., Cheng, N., Becker, S., and Nardi, L. (2025a). Exploring exploration in Bayesian optimization. In *Conference on Uncertainty in Artificial Intelligence*. Cited on page 8.

Papenmeier, L., Nardi, L., and Poloczek, M. (2022). Increasing the scope as you learn: Adaptive Bayesian optimization in nested subspaces. *Advances in Neural Information Processing Systems*. Cited on pages 1, 5, and 16.

Papenmeier, L., Nardi, L., and Poloczek, M. (2023). Bounce: Reliable high-dimensional Bayesian optimization for combinatorial and mixed spaces. *Advances in Neural Information Processing Systems*. Cited on pages 3 and 16.

Papenmeier, L., Poloczek, M., and Nardi, L. (2025b). Understanding high-dimensional Bayesian optimization. In *International Conference on Machine Learning*. Cited on pages 2, 3, 4, and 16.

Rashidi, B., Johnstonbaugh, K., and Gao, C. (2024). Cylindrical Thompson sampling for high-dimensional Bayesian optimization. In *International Conference on Artificial Intelligence and Statistics*. Cited on page 16.

Rolland, P., Scarlett, J., Bogunovic, I., and Cevher, V. (2018). High-dimensional Bayesian optimization via additive models with overlapping groups. In *International Conference on Artificial Intelligence and Statistics*. Cited on page 16.

Santoni, M. L., Raponi, E., Leone, R. D., and Doerr, C. (2024). Comparison of high-dimensional Bayesian optimization algorithms on BBOB. *ACM Transactions on Evolutionary Learning*. Cited on page 1.

Schoenberg, I. (1942). Positive definite functions on spheres. *Duke Mathematical Journal*. Cited on page 5.

Schoenberg, I. J. (1938). Metric spaces and completely monotone functions. *Annals of Mathematics*. Cited on page 15.

Shahriari, B., Swersky, K., Wang, Z., Adams, R. P., and De Freitas, N. (2015). Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE*. Cited on page 3.

Sobol, I. M. (1967). Distribution of points in a cube and approximate evaluation of integrals. *USSR Computational mathematics and mathematical physics*. Cited on page 6.

Song, L., Xue, K., Huang, X., and Qian, C. (2022). Monte Carlo tree search based variable selection for high dimensional Bayesian optimization. *Advances in Neural Information Processing Systems*. Cited on page 16.

Srinivas, N., Krause, A., Kakade, S. M., and Seeger, M. (2010). Gaussian process optimization in the bandit setting: No regret and experimental design. In *International Conference on Machine Learning*. Cited on pages 1 and 3.

Thompson, W. R. (1933). On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*. Cited on page 5.

Vakili, S., Moss, H., Artemev, A., Dutordoir, V., and Picheny, V. (2021). Scalable Thompson sampling using sparse Gaussian process models. *Advances in Neural Information Processing Systems*. Cited on page 2.

Vershynin, R. (2025). *High-dimensional probability: An introduction with applications in data science*. Cambridge University Press, second edition. Cited on page 16.

Wan, X., Nguyen, V., Ha, H., Ru, B., Lu, C., and Osborne, M. A. (2021). Think global and act local: Bayesian optimisation over high-dimensional categorical and mixed search spaces. In *International Conference on Machine Learning*. Cited on page 16.

Wang, L., Fonseca, R., and Tian, Y. (2020). Learning search space partition for black-box optimization using Monte Carlo tree search. *Advances in Neural Information Processing Systems*. Cited on page 5.

Wang, Z., Gehring, C., Kohli, P., and Jegelka, S. (2018). Batched large-scale Bayesian optimization in high-dimensional spaces. In *International Conference on Artificial Intelligence and Statistics*. Cited on pages 1 and 5.

Wang, Z., Hutter, F., Zoghi, M., Matheson, D., and De Feitas, N. (2016). Bayesian optimization in a billion dimensions via random embeddings. *Journal of Artificial Intelligence Research*. Cited on pages 1 and 16.

Williams, C. K. and Rasmussen, C. E. (2006). *Gaussian processes for machine learning*. MIT press. Cited on page 5.

Wilson, J., Borovitskiy, V., Terenin, A., Mostowsky, P., and Deisenroth, M. (2020). Efficiently sampling functions from Gaussian process posteriors. In *International Conference on Machine Learning*. Cited on page 18.

Wu, K., Kim, K., Garnett, R., and Gardner, J. (2023). The behavior and convergence of local Bayesian optimization. *Advances in Neural Information Processing Systems*. Cited on pages 1 and 16.

Xu, Z., Wang, H., Phillips, J. M., and Zhe, S. (2025). Standard Gaussian process is all you need for high-dimensional Bayesian optimization. In *International Conference on Learning Representations*. Cited on pages 1, 2, 3, 4, 8, and 16.

Ziomek, J. K. and Ammar, H. B. (2023). Are random decompositions all we need in high dimensional Bayesian optimisation? In *International Conference on Machine Learning*. Cited on page 16.

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, see Section 3.]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes, see Section 3.]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes, source code in the supplementary material.]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes, see Section 3 and Appendix A.]
   (b) Complete proofs of all theoretical results. [Yes, see Appendix A.]
   (c) Clear explanations of any assumptions. [Yes, see Section 3 and Appendix A.]

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, source code in the supplementary material.]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes, see Appendix C.]
   (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, see Appendix C.]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes, see Appendix C.]

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, see Section 4.]
   (b) The license information of the assets, if applicable. [Yes, see Appendix C.]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Yes, source code in the supplementary material.]
   (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]

Appendix

A Proofs, examples and derivations

A.1 Proof of Theorem 1

Theorem 1. *For acquisition functions increasing in posterior mean and variance (e.g. expected improvement), Bayesian linear models will maximize acquisition on the boundary of the search space. That is, for* $\mathbf{x}_{t+1} = \arg\max_{\mathbf{x}\in[-1,1]^D} \alpha_t(\mathbf{x})$ *at any timestep* t*, we have that* $\mathbf{x}_{t+1}$ *contains at least one dimension equal to* -1 *or* 1 *(i.e.* $\|\mathbf{x}_{t+1}\|_\infty = 1$*).*

Proof. Most common acquisition functions, including Expected Improvement (`EI`) and Upper Confidence Bound (`UCB`), can be shown to be monotonically increasing in the posterior mean $\mu(\mathbf{x})$ and standard deviation $\sigma(\mathbf{x})$ of the underlying surrogate. Now, consider the following proof by contradiction: the acquisition optimum $\mathbf{x}$ does not lie on the boundary. If we parametrize $\mathbf{x}$ as a direction $\mathbf{z}$ and magnitude $c \geq 0$, we get

$$\mathbf{x} = \underbrace{\|\mathbf{x}\|}_{c} \cdot \underbrace{\frac{\mathbf{x}}{\|\mathbf{x}\|}}_{\mathbf{z}}, \quad \text{with} \quad \mu(\mathbf{x}) = \mathbf{x}^\top \hat{\boldsymbol{\beta}} = c \cdot \mathbf{z}^\top \hat{\boldsymbol{\beta}} \quad \text{and} \quad \sigma(\mathbf{x}) = \sqrt{\sigma_\varepsilon^2 + \mathbf{x}^\top \mathbf{S} \mathbf{x}} = \sqrt{\sigma_\varepsilon^2 + c^2 \cdot \mathbf{z}^\top \mathbf{S} \mathbf{z}}.$$

As a result, for any $\mathbf{x}$, increasing c monotonically increases $\mu(\mathbf{x})$ and $\sigma(\mathbf{x})$, and therefore also $\alpha(\mathbf{x})$. Thus, the interior point $\mathbf{x}$ cannot be the (constrained) optimum, since increasing c will result in a higher acquisition value, which is a contradiction and concludes the proof.

Here, we have assumed $\mathbf{x}^\top \hat{\boldsymbol{\beta}} \geq 0$, since assuming otherwise would lead to a better optimum, namely $-\mathbf{x}$, which is also a contradiction. Moreover, unless all previous data points $\mathbf{x}_{1:t}$ obtain exactly the same objective value y (which is highly unlikely), we can safely assume $\hat{\boldsymbol{\beta}} \neq (0, \ldots, 0)$, meaning that $\mu(\mathbf{x})$ will be strictly increasing in c. As a consequence, not only will there be an optimum on the boundary, but this optimum will also be unique, meaning there is no optimum in the interior of the hypercube.

Adding an Intercept To extend the proof to a Bayesian linear model with an intercept, we follow a similar approach as before, but need to pay closer attention to the parameterization of $\mathbf{x}$. Specifically, we write $\mu(\mathbf{x}) = \hat{\beta}_0 + \mathbf{x}^\top \hat{\boldsymbol{\beta}}$ using the augmented feature map $\varphi(\mathbf{x}) = [1; \mathbf{x}]$. The posterior variance can then be expressed as

$$\sigma^2(\mathbf{x}) = \sigma_\varepsilon^2 + \varphi(\mathbf{x})^\top \mathbf{S}\, \varphi(\mathbf{x}) = \sigma_\varepsilon^2 + S_{00} + 2\,\mathbf{x}^\top S_{0x} + \mathbf{x}^\top S_{xx} \mathbf{x}, \quad \text{with} \quad \mathbf{S} = \begin{bmatrix} S_{00} & S_{0x}^\top \\ S_{0x} & S_{xx} \end{bmatrix}.$$

Completing the square, we choose a shift point $\mathbf{x}_0$ that removes the linear term, i.e. any solution of $S_{xx}\mathbf{x}_0 = -S_{0x}$ (unique when S_{xx} is invertible). Then

$$\sigma^2(\mathbf{x}) \;=\; \underbrace{\sigma_\varepsilon^2 + S_{00} - \mathbf{x}_0^\top S_{xx} \mathbf{x}_0}_{\text{constant in } \mathbf{x}} + (\mathbf{x} - \mathbf{x}_0)^\top S_{xx} (\mathbf{x} - \mathbf{x}_0),$$

where $S_{xx} \succeq 0$ is the slope block of the posterior covariance.

Now, we parametrize any interior point relative to $\mathbf{x}_0$ as $\mathbf{x} = \mathbf{x}_0 + c\,\mathbf{z}$ with $c \geq 0$ and $\|\mathbf{z}\| = 1$. If necessary, we flip $\mathbf{z}$ so that $\mathbf{z}^\top \hat{\boldsymbol{\beta}} \geq 0$. Along this ray, we obtain

$$\mu(\mathbf{x}) = \hat{\beta}_0 + \mathbf{x}_0^\top \hat{\boldsymbol{\beta}} + c\,\mathbf{z}^\top \hat{\boldsymbol{\beta}}, \quad \text{and} \quad \sigma(\mathbf{x}) = \sqrt{A + c^2\,\mathbf{z}^\top S_{xx} \mathbf{z}},$$

with $A = \sigma_\varepsilon^2 + S_{00} - \mathbf{x}_0^\top S_{xx} \mathbf{x}_0$ independent of c. Because $\mathbf{z}^\top \hat{\boldsymbol{\beta}} \geq 0$ and $S_{xx} \succeq 0$, both $\mu(\mathbf{x})$ and $\sigma(\mathbf{x})$ are nondecreasing in c, and are strictly increasing unless simultaneously $\mathbf{z}^\top \boldsymbol{\beta} = 0$ and $\mathbf{z}^\top S_{xx} \mathbf{z} = 0$ (a nongeneric degeneracy). □

A.2 Counterexample to Theorem 1 (For Spherical Projections)

We provide a specific example to support the following statement from Section 3.1: "when points are projected to the hypersphere, Theorem 1 does not apply anymore." That is, for

$$\mathbf{x}_{t+1} = \arg\max_{\mathbf{x}\in[-1,1]^D} \alpha_t\left(P(\mathbf{x})\right),$$

where $P : \mathbb{R}^D \to \mathbb{S}^D$, the optimum $\mathbf{x}_{t+1}$ can be any point in the hypercube (i.e. $\|\mathbf{x}_{t+1}\|_\infty \in [0,1]$), instead of only boundary points (i.e. $\|\mathbf{x}_{t+1}\|_\infty = 1$) as stated in Theorem 1.

For simplicity, consider $D = 1$ and `UCB` with an exploration factor of $\lambda = 0$, giving us

$$\alpha_t(P(x)) = P(x)^\top \hat{\boldsymbol{\beta}}, \quad \text{where we assume} \quad \hat{\boldsymbol{\beta}} = \begin{pmatrix} 1/2 \\ -1 \end{pmatrix}.$$

As spherical mapping $P : \mathbb{R}^D \to \mathbb{S}^D$, we consider the inverse stereographic projection from Equation (4), giving us

$$\alpha(x) = \frac{1}{2} \cdot \frac{2x}{x^2+1} - \frac{x^2-1}{x^2+1}, \quad \text{with} \quad \alpha(-1) = -\tfrac{1}{2}, \quad \alpha\left(\tfrac{1}{2}\right) = 1, \quad \alpha(1) = \tfrac{1}{2}.$$

Here, the interior point $x = 1/2$ leads to a higher acquisition value than the two boundary points $x = -1$ and $x = 1$, thus representing a clear example for which Theorem 1 does not hold (because of the non-linear projection $P(x)$ from the real line $\mathbb{R}$ to the circle $\mathbb{S}$).

A.3 Isotropic Kernels Become Dot-Product Kernels (After Spherical Projection)

Any isotropic kernel becomes a dot product kernel when applied to unit norm inputs.

Let $k : \mathbb{R}^\infty \times \mathbb{R}^\infty \to \mathbb{R}$ be an isotropic kernel, where $\mathbb{R}^\infty \times \mathbb{R}^\infty$ denotes that the kernel is valid on any possible input dimensionality. By the result of Schoenberg (1938, Theorem 2), this kernel can be expressed as:

$$k(\mathbf{x}, \mathbf{x}') = \int \exp\left(-\frac{\ell}{2}\left\|\mathbf{x}-\mathbf{x}'\right\|_2^2\right) d\mu(\ell) = \int \exp\left(-\frac{\ell\left(\|\mathbf{x}\|^2 + \|\mathbf{x}'\|^2\right)}{2}\right) \exp\left(\ell \mathbf{x}^\top \mathbf{x}'\right) d\mu(\ell) \tag{7}$$

where $\mu(\ell)$ is some positive finite measure over $[0, \infty)$. Without loss of generality, we assume that μ is a probability measure—alternatively, we can normalize the measure by $1/\mu([0,\infty))$—as the value of $\mu([0,\infty))$ can be viewed as an implicit outputscale on the kernel.

Applying this kernel to sphere-projected inputs $P(\mathbf{z}), P(\mathbf{z}')$ (which are both unit norm), we have:

$$k(P(\mathbf{z}), P(\mathbf{z}')) = \int \exp(-\ell) \exp\left(\ell P(\mathbf{z})^\top P(\mathbf{z}')\right) d\mu(\ell)$$

Taking a Taylor expansion of the right exponential and applying Fubini-Tonelli, we have:

$$k(P(\mathbf{z}), P(\mathbf{z}')) = \sum_{i=0}^{\infty} \frac{\int \ell^i \exp(-\ell)\, d\mu(\ell)}{i!} \left[P(\mathbf{z})^\top P(\mathbf{z}')\right]^i. \tag{8}$$

As $\ell^i \exp(-\ell)$ is trivially bounded for any $i \in \mathbb{N}$ and $\ell \geq 0$, the integrals in Equation (8) are all finite and well defined. Furthermore, by Equation (7) we have that $k(P(\mathbf{z}), P(\mathbf{z})) = 1$ for all $\mathbf{z}$; combining this fact with Equation (8) we have that

$$k(P(\mathbf{z}), P(\mathbf{z})) = \sum_{i=0}^{\infty} \frac{\int \ell^i \exp(-\ell)\, d\mu(\ell)}{i!} = 1. \tag{9}$$

In other words, the polynomial coefficients sum to 1, and thus k restricted to the unit sphere can be arbitrarily approximated by the higher order polynomial kernel in Equation (5).

A.4 Derivation of Equation (6)

The polynomial expression of the sphere-based RBF kernel can be derived from Equations (7) and (8), under the atomic measure $d\mu(\ell) = \delta(\ell - 1)$, where δ is the Dirac delta function.

A.5 Thin Shell Phenomenon

Assume $X = (X_1, X_2, \ldots, X_D)$ is generated uniformly over the D-dimensional hypercube $[-1, 1]^D$. Moreover, define the scaled variables Z as $Z = \begin{bmatrix} X_1/\ell_1 & \cdots & X_D/\ell_D \end{bmatrix} = X\sqrt{3/D}$, where the lengthscales have been scaled according to $O(\sqrt{D})$ as suggested in recent works (Hvarfner et al., 2024; Xu et al., 2025; Papenmeier et al., 2025b). Then, we have

$$\mathbb{E}\left(X_i^2\right) = \frac{1}{2}\int_{-1}^{1} x_i^2\, dx = \frac{1}{3}, \quad \text{and} \quad \text{var}\left(X_i^2\right) = \frac{1}{2}\int_{-1}^{1} x_i^4\, dx - \mathbb{E}\left(X_i^2\right)^2 = \frac{4}{45},$$

which gives us

$$\mathbb{E}\left(Z_i^2\right) = \frac{1}{D}, \quad \text{and} \quad \text{var}\left(Z_i^2\right) = \frac{4}{5D^2}.$$

Using the above, we have

$$\mathbb{E}\left(\|Z\|^2\right) = \sum_{i=1}^{D} \mathbb{E}\left(Z_i^2\right) = 1, \quad \text{and} \quad \text{var}\left(\|Z\|^2\right) = \sum_{i=1}^{D} \text{var}\left(Z_i^2\right) = \frac{4}{5D}.$$

Therefore, as $D \to \infty$, it follows that

$$\mathbb{E}\left(\|Z\|^2\right) = 1, \quad \text{and} \quad \text{var}\left(\|Z\|^2\right) = 0,$$

meaning $\|Z\|$ converges to 1 (in probability, and almost surely). This phenomenon is also known more broadly as the "thin shell phenomenon", and we refer the reader to Vershynin (2025) for a more complete treatment of the topic.

B Extended Related Work

Low-Dimensional Subspaces Instead of searching in a high-dimensional space, a major stream of HDBO methods instead map $\mathcal{X}$ to a low-dimensional subspace, on which the search is more feasible. This mapping can be: *(i) linear*, with methods such as `REMBO` (Wang et al., 2016), `HeSBO` (Nayebi et al., 2019), `ALEBO` (Letham et al., 2020), `BAxUS` (Papenmeier et al., 2022), and `Bounce` (Papenmeier et al., 2023), or *(ii) non-linear*, with methods such as `DMGPC-BO` (Moriconi et al., 2020) and `LOL-BO` (Maus et al., 2022). Additionally, the linear embeddings can be focused on *variable selection* only, as done in `Dropout-Mix` (Li et al., 2017), `SAASBO` (Eriksson and Jankowiak, 2021), `MCTS-VS` (Song et al., 2022), and `GTBO` (Hellsten et al., 2025).

Local Search Alternatively, instead of performing global search in a low-dimensional subspace, another stream of methods instead perform local search in the full-dimensional space. This can be done using: *(i) trust regions*, such as `TuRBO` (Eriksson et al., 2019), `CASMOPOLITAN` (Wan et al., 2021), and `CTS-TuRBO` (Rashidi et al., 2024), or *(ii) approximate gradients*, such as `GIBO` (Müller et al., 2021; Wu et al., 2023), `MPD` (Nguyen et al., 2022), and `LA-MinUCB` (Fan et al., 2024).

Additive Decompositions Rather than change the search space, additive-decomposition methods impose simplifying assumptions on the structure of the objective function. Examples include `Add-GP-UCB` (Kandasamy et al., 2015), `G-Add-GP-UCB` (Rolland et al., 2018), `Tree-GP-UCB` (Han et al., 2021), and `RDUCB` (Ziomek and Ammar, 2023).

C Implementation Details

We run `EULBO` (Maus et al., 2024) and `BAxUS` (Papenmeier et al., 2022) using the authors' original implementations. For `TuRBO` (Eriksson et al., 2019), `SAASBO` (Eriksson and Jankowiak, 2021), and Vanilla BO (Hvarfner et al., 2024), we use implementations present in `BoTorch` (Balandat et al., 2020). For `BOCK` (Oh et al., 2018), we replicate the authors' original implementation as closely as possible using the provided kernel in `GPyTorch` (Gardner et al., 2018), allowing for an easy integration with `BoTorch`.

Due to the high computational costs associated with `SAASBO`, we only run the model for 500 iterations, whereas all other methods are run for 1000 iterations. We run each method for at least 10 different random seeds, and the plots in this paper display the mean and standard error of the mean (with respect to the seeds) as a solid line and shaded area, respectively. We run our linear kernel as well other baselines on a cluster containing both CPUs and GPUs, where the latter were used to speed up computation for slower methods or datasets. Vanilla BO and our proposed linear kernel delivered similar run times ($\pm 10\%$): given an evaluation budget of $N = 1000$, the pipeline takes between 1h and 12h on 1 NVIDIA GPU (e.g. L40S) depending on the benchmark. For the molecular experiments with $N > 20{,}000$ (on a similar GPU), the pipeline takes between four and six days.

D Additional Ablation Studies

D.1 Effect of Hyperprior

As mentioned in Section 3.1, performance of our linear model is largely unaffected by the hyperprior placed on the lengthscales. In Figure 7, we evaluate three different hyperpriors:

1. `Gamma`: $\ell_i \sim \Gamma(3, 6)$, which used to be the default in `BoTorch` (Hvarfner et al., 2024),
2. `LogNormal`: $\ell_i \sim \mathcal{LN}\left(\sqrt{2} + \frac{\log(D)}{2}, \sqrt{3}\right)$, which is used in Vanilla BO (often referred to as `DSP`), and
3. `LogNormal`*: $\ell_i \sim \mathcal{LN}\left(\sqrt{2} + \frac{\log(1)}{2}, \sqrt{3}\right)$, which is equivalent to `DSP` without the dimensionality D, since scaling by D is not needed when combined with our global lengthscale a (see Section 3.1).

As can be seen, using `Gamma` instead of `LogNormal`* leads to identical performance for our linear model (orange and red lines), whereas it results in large differences when applied to standard RBF kernels (green and purple lines). In the following section, we discuss how our proposed spherical projection and global lengthscale can be combined with RBF kernels (brown and pink lines).

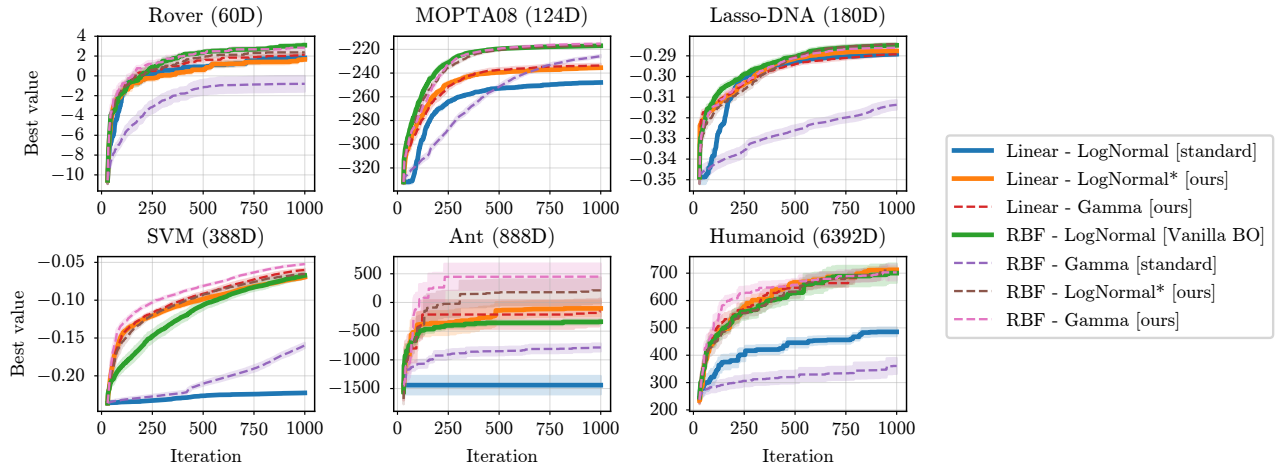

Figure 7: When applied to our spherical mapping with global lengthscale, both linear and RBF kernels are robust to the choice of (lengthscale) hyperprior, and match the performance of Vanilla BO.

D.2 Effect of Our Spherical Mapping on an RBF Kernel

In Section 4.3, we claim that the inverse stereographic projection has almost no effect for non-parametric models. To support this claim, we evaluate the RBF kernel on points projected to the unit hypersphere; see the left-hand side of Equation (6) for a more explicit formulation. In Figure 7, this RBF kernel applied to the hypersphere (brown line) leads to similar performance as the RBF kernel from Vanilla BO (green line), supporting our claim above. Moreover, replacing the `LogNormal` prior from Vanilla BO by the `Gamma` prior (pink line) does not degrade performance for our model (thanks to our projection and global lengthscale a), whereas it does lead to significantly worse results for the standard RBF kernel (as presented in Hvarfner et al., 2024).

D.3 Effects of Projection, Centering, ARD, and Global Lengthscale

In Figure 8, we modify certain components of our model proposed in Section 3.1 and assess their impact on performance. We consider four modifications: *(i)* no spherical projection P (i.e. P is an identity map), *(ii)* a non-centered search space $[0, 1]^D$ (instead of $[-1, 1]^D$), *(iii)* no Automatic Relevance Determination (ARD) for the lengthscales (i.e. replacing ℓ_i by ℓ), and *(iv)* removing the global lengthscale a (i.e. $a = 1$).

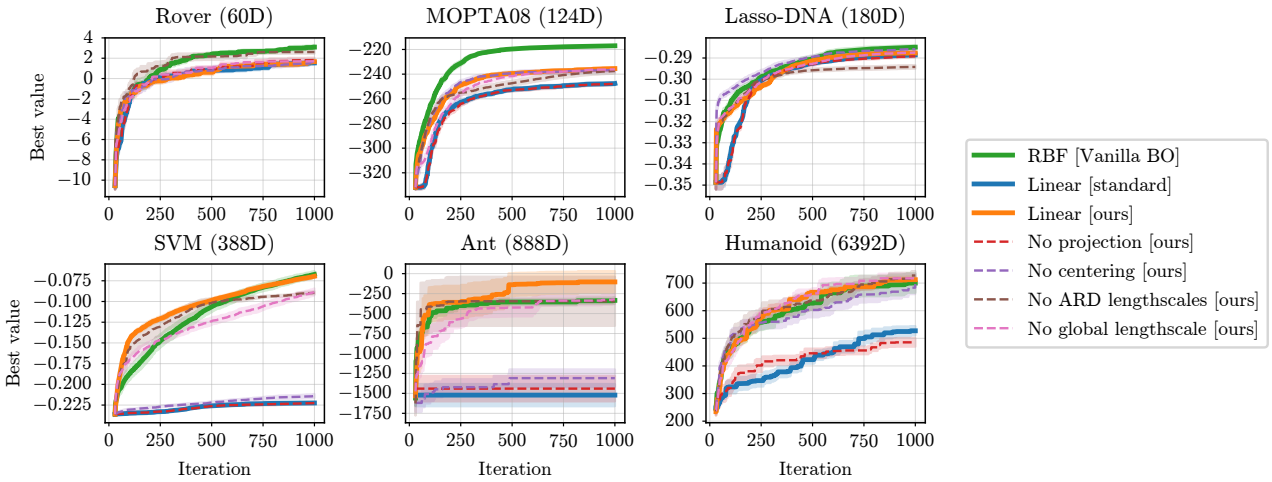

Figure 8: The spherical projection and centering of the hypercube search-space play a key role in our proposed linear model, whereas ARD and our global lengthscale only provide modest improvements.

As confirmed by Figures 4 and 15, the spherical projection is the most important element among our different modifications, and removing it significantly affects performance. Similarly, going from a centered hypercube $[-1, 1]^D$ to a non-centered hypercube $[0, 1]^D$ leads to a noticeable drop in performance. We hypothesize that this is related to the inverse stereographic projection, where the "0-boundary" and the "1-boundary" are treated differently if we don't center, but exactly the same if we do. Specifically, Equation (4) heavily relies on the norm of $\mathbf{x}$, and both boundaries will have different norms if no centering is applied. Interestingly, though both our global lengthscale and ARD bring performance improvements, these improvements are rather minor in contrast to centering and spherical projection of the search space.

D.4 Effect of Acquisition Function

Our results do not change significantly if we replace EI with UCB (Figure 9): Vanilla BO and our proposed linear model obtain performance similar to that in Figure 2. If we replace EI with TS (Figure 10), both methods incur a noticeable performance drop. However, our spherical linear model still matches, and sometimes exceeds, the performance of an RBF GP with TS (i.e. Vanilla BO). For the generation of the Thompson samples, we rely on Wilson et al. (2020), which means we can optimize the acquisition function using gradient-based methods.

D.5 Effect of Problem Structure

Here we test our linear models versus Vanilla BO models on synthetic problems with known objective functions. In particular, we aim to assess the relative efficacy of our method as we increase the degree of sparsity and nonlinearity in the objective.

Effect of Sparsity. Following the setup in Hvarfner et al. (2024, Figure 5), we embed the `Levy` ($D = 4$) and `Hartmann` ($D = 6$) functions into a range of higher dimensionalities ($D = 25$, $D = 100$, $D = 300$, and $D = 1000$), thus leading to objective functions with increasing amounts of sparsity (Levy and Montalvo, 1985; Hartman, 1973). For both test functions, we observe a clear trend across dimensionality (Figure 11): as the embedding dimension grows (i.e. sparsity increases), the gap between Vanilla BO and our proposed linear model vanishes. From these results, we conclude that our proposed linear model might be more suited to functions with a higher

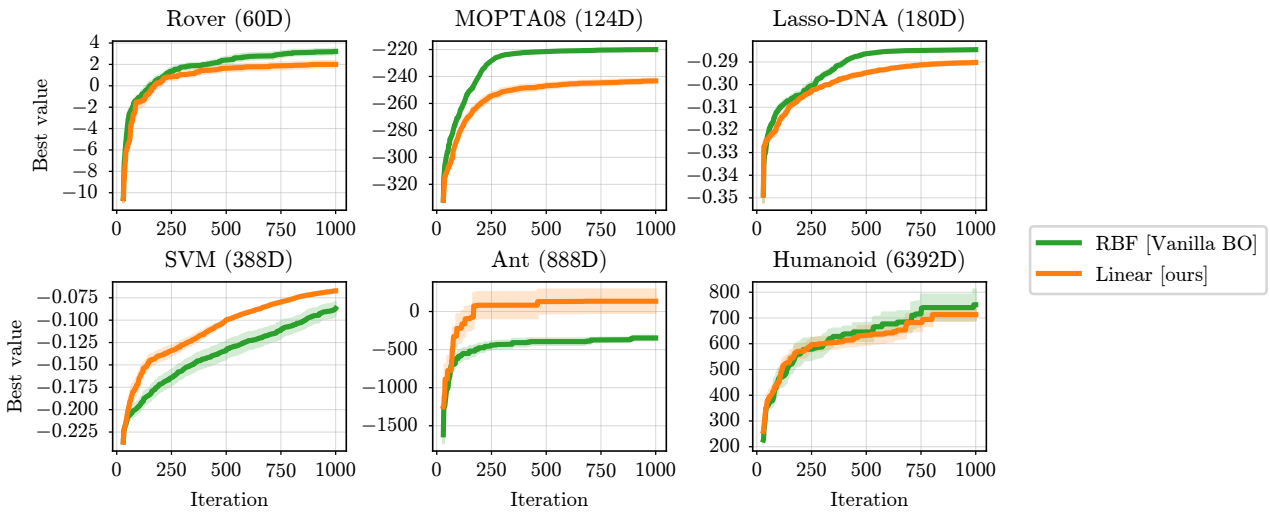

Figure 9: Replacing EI with UCB does not significantly affect performance of Vanilla BO and our proposed linear model.

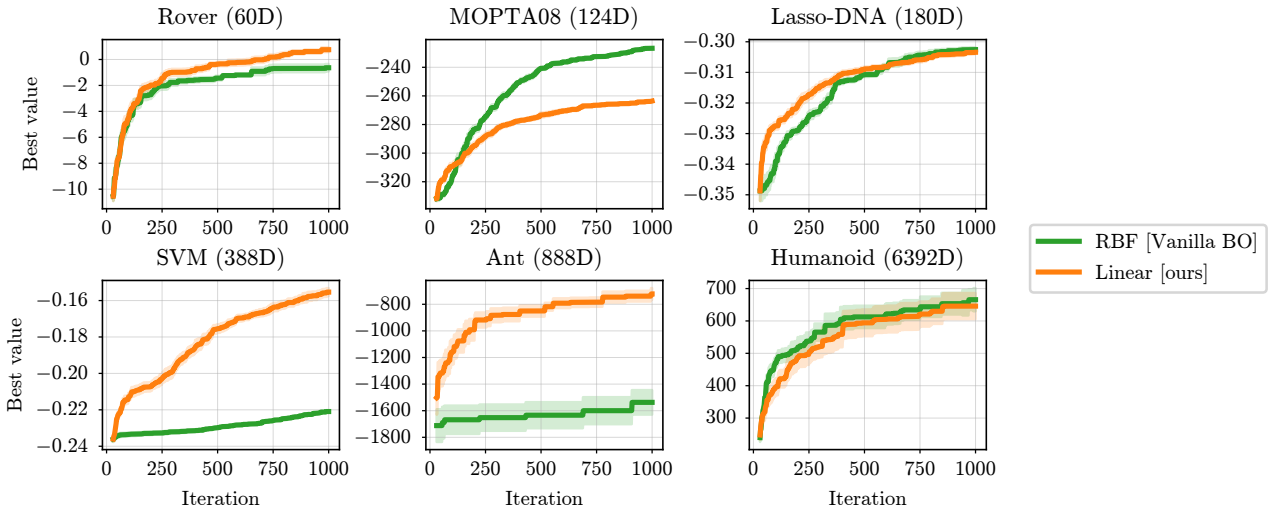

Figure 10: Replacing EI with TS leads to a noticeable performance drop for both methods, but overall our proposed linear model is still on par with (or maybe slightly better than) Vanilla BO.

dimensionality or a higher degree of sparsity.

Effect of Non-Linearity in the Objective Function. We consider objective functions that are varying convex combinations of linear and non-linear objective functions. Specifically, we draw the objective function from a Gaussian process with the following kernel:

$$k_{\text{test}} = \alpha\, k_{\text{linear}} + (1-\alpha)\, k_{\text{RBF}}, \quad \text{with} \quad \alpha \in [0,1]. \tag{10}$$

Small values of α correspond to highly nonlinear functions; large values correspond to nearly linear functions. In Figure 12, as α goes from zero to one, the difference between our proposed linear model and Vanilla BO vanishes (30% gap to 5% gap). Nevertheless, we observe that our spherical linear model still yields substantial optimization progress even when the objective function is entirely nonlinear ($\alpha = 0$).

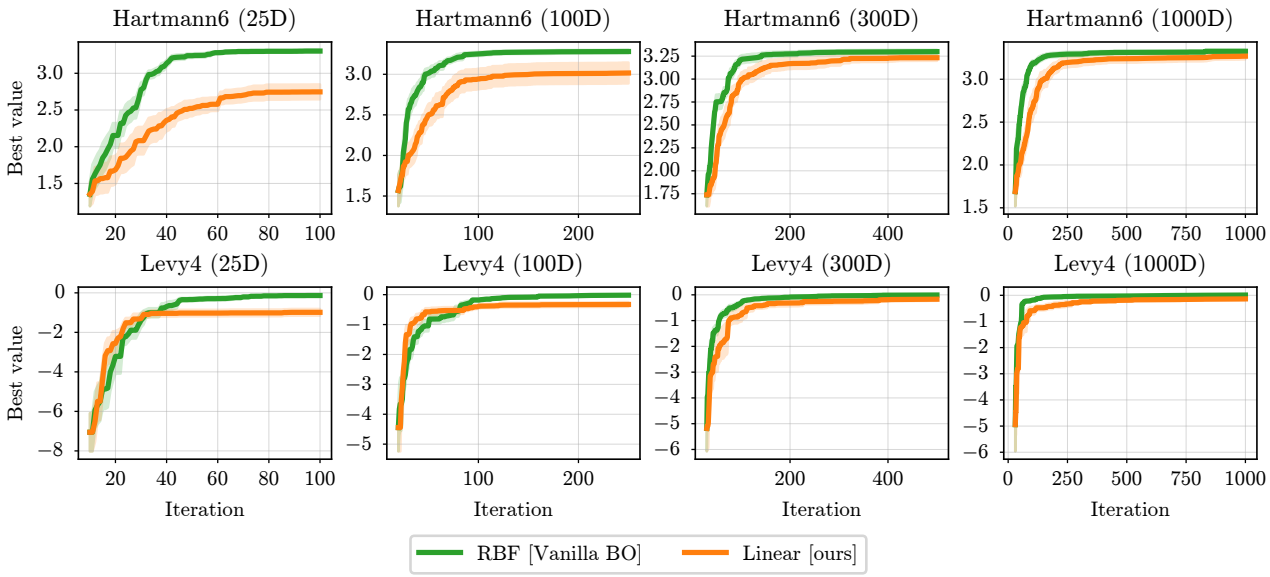

Figure 11: **Impact of objective function sparsity on optimization results.** The Hartmann6 (top) and Levy4 (bottom) test functions are embedded in high-dimensional spaces, following the protocol of Hvarfner et al. (2024). Increasing the ambient dimensionality reduces the performance gap between our proposed linear model and Vanilla BO; however both methods are comparable in most settings.

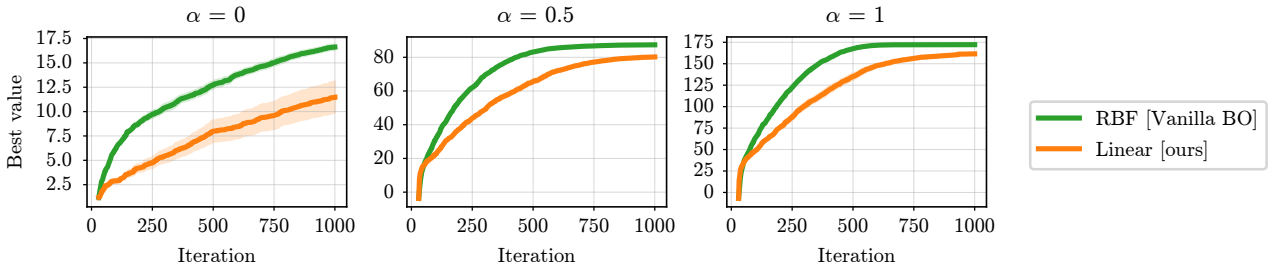

Figure 12: **Impact of objective function nonlinearity on optimization results.** The objective function is drawn from a convex combination of linear and non-linear Gaussian processes, where α is the combination parameter. $\alpha = 0$ (left) is entirely non-linear; $\alpha = 1$ (right) is entirely linear. The difference between our proposed linear model and Vanilla BO decreases as $\alpha \to 1$, though performance is arguably comparable at all combination levels.

E Complete Experimental Results

E.1 `GuacaMol` Datasets for $N = 1{,}000$

We compare our linear kernel from Equation (2) against Vanilla BO on nine different tasks from the `GuacaMol` benchmark (Brown et al., 2019). Specifically, we run both methods in the 256D latent space of the `SELFIES-VAE` (Maus et al., 2022), and plot the results in Figure 13. On all nine datasets, our linear model significantly outperforms Vanilla BO, potentially indicating (a part of) its design is better suited for (molecular) latent spaces. We leave more thorough analysis of this strong performance to further work, but believe it to be a fruitful direction.

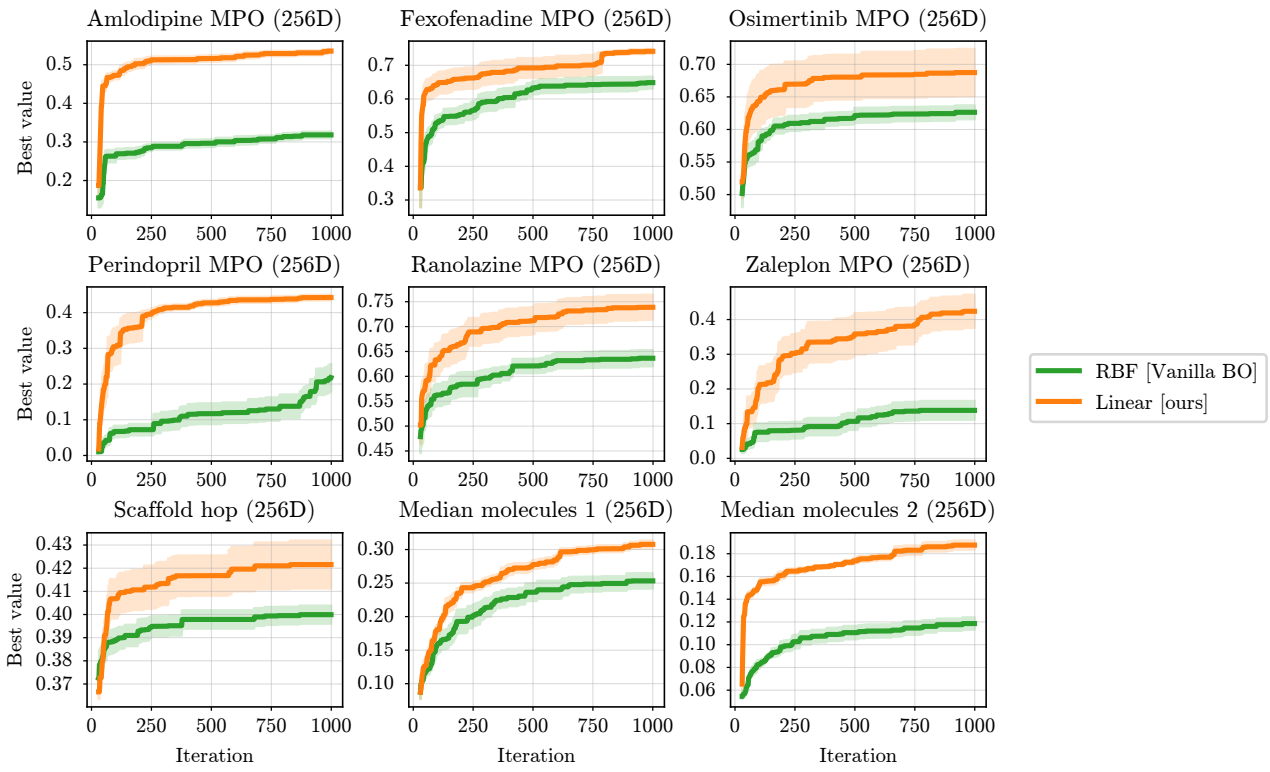

Figure 13: Our proposed linear kernel from Equation (2) significantly outperforms Vanilla BO on molecular tasks from the `GuacaMol` benchmark (Brown et al., 2019), where BO takes place in the 256D latent space of the `SELFIES-VAE` (Maus et al., 2022).

E.2 Higher-Order Polynomials

We replicate the results of Figure 3 in Figure 14, where the latter now contains six datasets instead of two. The conclusions for this extended figure are the same as in Section 4.3: higher-order polynomials ($m > 1$) from Equation (5) all obtain near-identical performance compared to the linear kernel ($m = 1$). The only (minor) exception to this rule is the `MOPTA08` dataset, where higher-order polynomials slightly outperform the linear kernel.

E.3 Choice of Spherical Mapping

We extend the findings from Figure 4 in Figure 15, which expands the analysis to encompass six datasets rather than two. In both figures, we rely on our linear kernel from Equation (2), and only modify the spherical projection P. Here, `None` means no projection is used (i.e. P is an identity map), the `Inverse stereographic` projection is defined in Equation (4), and the remaining projections can be written as:

$$
\begin{aligned}
P_{\text{radial}}(\mathbf{z}) &= \tfrac{1}{\gamma}\begin{bmatrix} z_1 & \cdots & z_D & \sqrt{\gamma^2 - ||\mathbf{z}||^2} \end{bmatrix}, \\
P_{\text{norm.}}(\mathbf{z}) &= \tfrac{1}{||\mathbf{z}||}\begin{bmatrix} z_1 & \cdots & z_D \end{bmatrix}, \\
P_{\text{homo.}}(\mathbf{z}) &= \tfrac{1}{||\mathbf{z}||+D\gamma}\begin{bmatrix} z_1 & \cdots & z_D & \gamma_1 & \cdots & \gamma_D \end{bmatrix}, \\
P_{\text{(co)sine}}(\mathbf{z}) &= \tfrac{1}{||\mathbf{z}||}\begin{bmatrix} z_1\cos(||\mathbf{z}||) & \cdots & z_D\cos(||\mathbf{z}||) & z_1\sin(||\mathbf{z}||) & \cdots & z_D\sin(||\mathbf{z}||) \end{bmatrix},
\end{aligned}
$$

where $\gamma = \gamma_1 = \ldots = \gamma_D$ represents the largest norm possible for $\mathbf{z}$ (given that $\mathbf{z}$ is a scaled version of $\mathbf{x} \in [-1, 1]^D$).

This broader comparison of datasets yields consistent conclusions with those presented in Section 4.3: a spherical projection is necessary to improve performance, but the choice of spherical projection can have a drastic impact

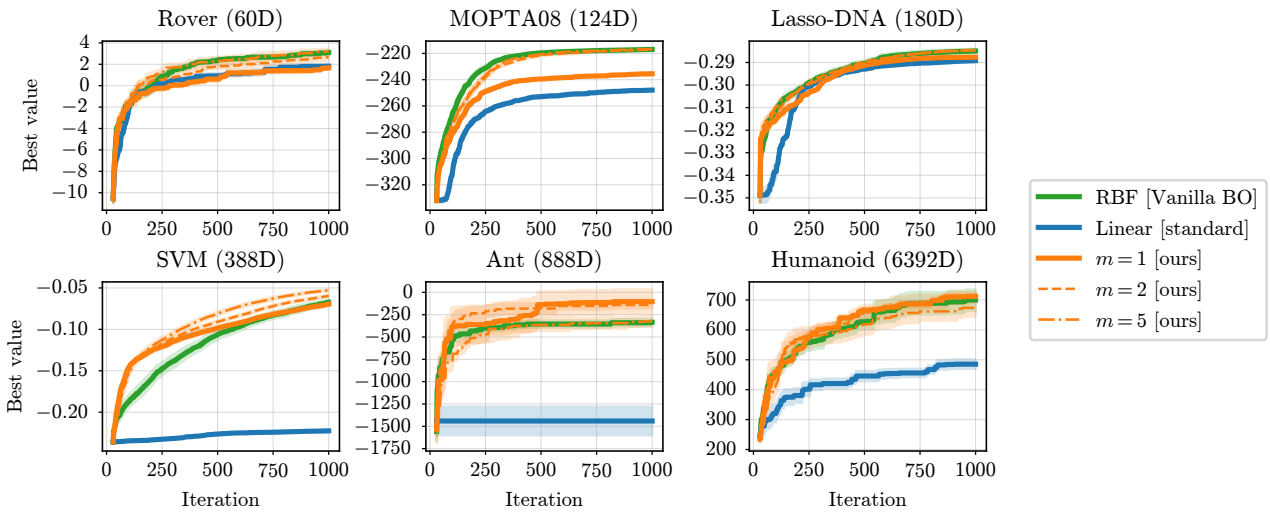

Figure 14: Extension of Figure 3 (high-order polynomials do not improve upon our linear model).

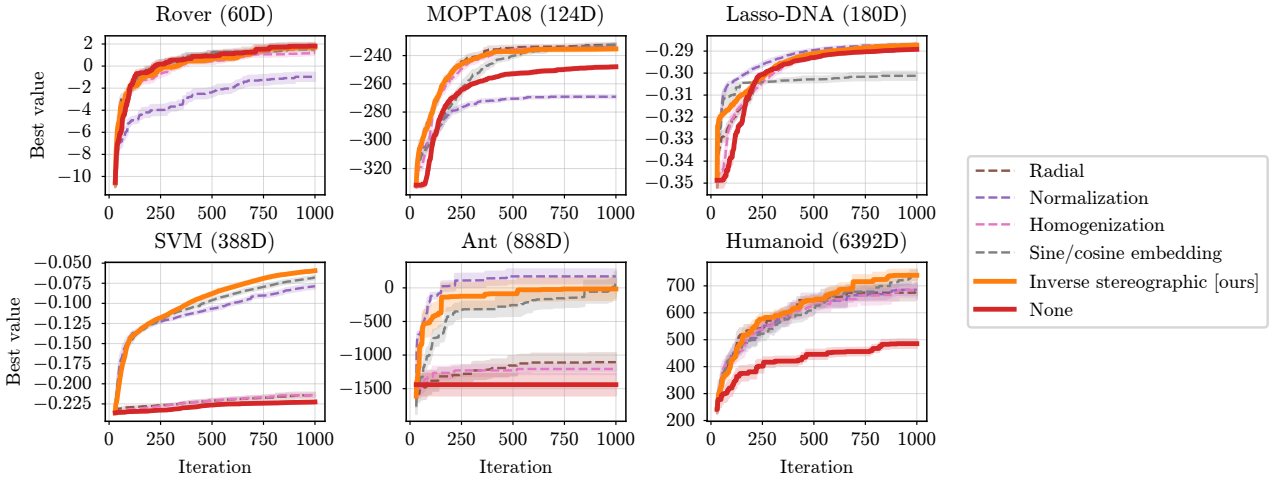

Figure 15: Extension of Figure 4 (the choice of spherical projection can have a drastic impact on performance).

on performance. For example, simple `Normalization` achieves state-of-the-art performance on the `Ant` dataset, but is outperformed by all other projections—including no projection at all—on the `MOPTA08` dataset. For the final version of our linear kernel, we opt for the `Inverse stereographic` projection, which consistently delivers superior performance across datasets.

E.4 Percentage of Dimensions on the Boundary

In Figure 16, we complement the results of Figure 5 by adding four more datasets (for a total of six). As described in Section 5, we can clearly observe a corner-seeking behavior for the standard linear kernel, with almost 100% of the dimensions (of acquired points) lying on the boundary. For our linear kernel and Vanilla BO, this is not the case, but we note a surprisingly high diversity of acquired points across datasets: from almost 0% dimensions on the boundary for `SVM` and `Ant` to more than 50% for the other four datasets.

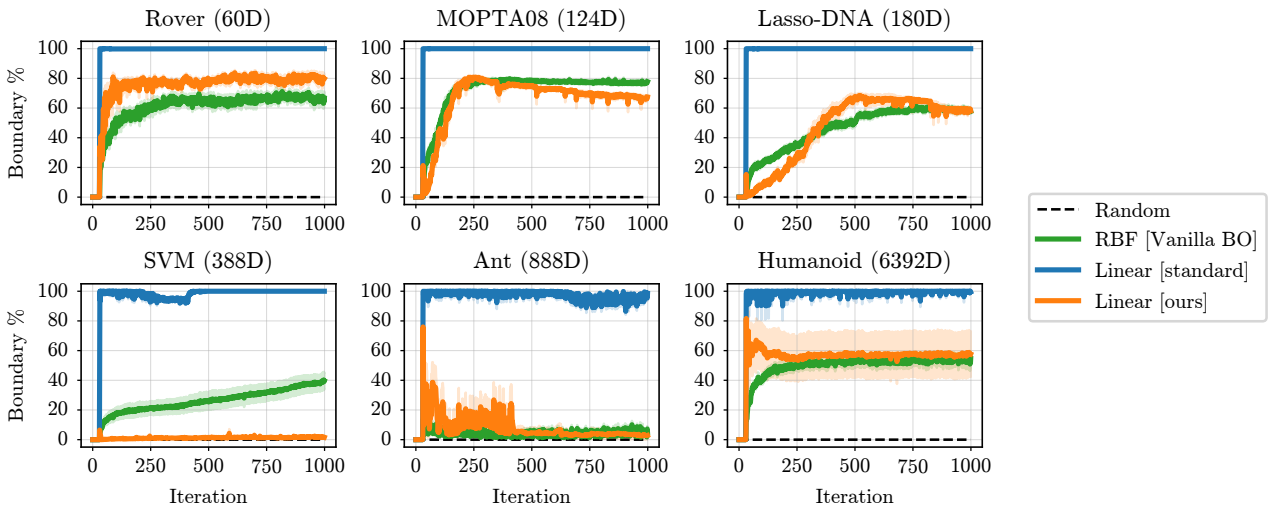

Figure 16: Extension of Figure 5 (standard linear kernels lead to acquisitions in the corners of the hypercube, whereas our spherically-mapped linear kernels acquire points similar to Vanilla BO).

E.5 Observation Traveling Salesman Distance (OTSD)

Similar to the above paragraph, we expand the results of Figure 5 in Figure 17 to include four more datasets. Here, we can observe a similar pattern as described in Section 5: the corner-seeking behavior of the standard linear kernel leads to an OTSD metric often exceeding that of random search, whereas our spherically-mapped linear kernel displays patterns of "locality" closer to those of Vanilla BO. Additionally, we can see that the OTSD metric for the standard linear kernel keeps increasing with dimensionality D. This result is not surprising: as D grows large, corners in a high-dimensional hypercube are increasingly far away from each other compared to randomly-sampled points.

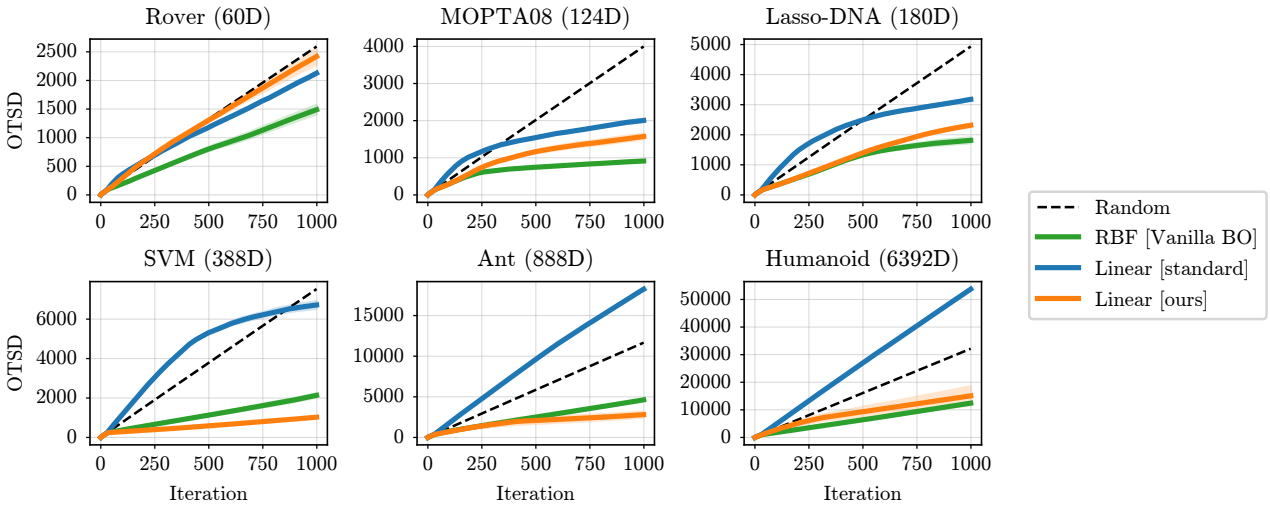

Figure 17: Extension of Figure 5 (our linear kernel exhibits "locality", similar to Vanilla BO and unlike the standard linear kernel).

E.6 Regression Performance

In Figure 18, we add four more datasets (for a total of six) to the Box plots of Figure 6. Based on these six datasets, we draw the same conclusions as in Section 5:

1. the points chosen during the BO loop are easier to predict than evenly-spread Sobol points (perhaps because of the "local" behavior of our model), and
2. our spherically-mapped linear model does not significantly improve the regression performance of the standard linear model (thus discarding the idea that the performance of our linear model is due to strong, non-linear features induced by the spherical projection).

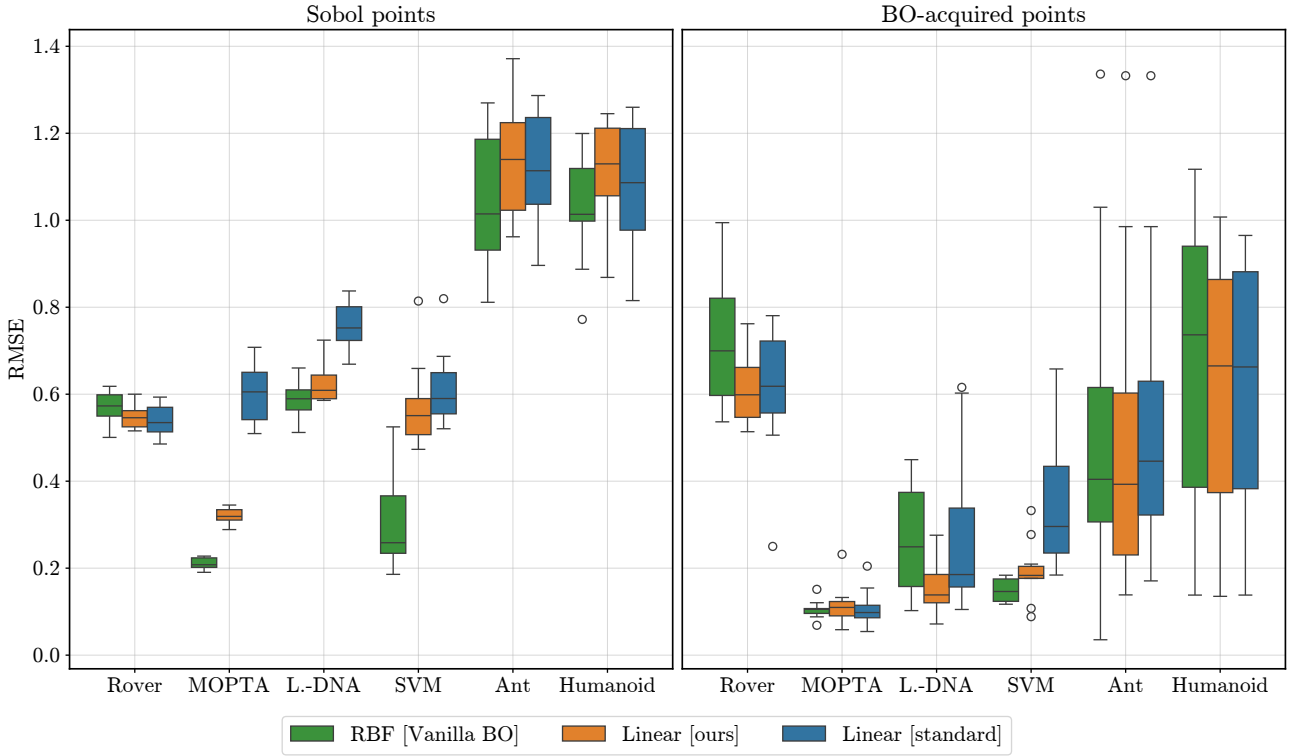

Figure 18: Extension of Figure 6 (our spherical projection does not lead to better predictive performance for the standard linear kernel).