How Diverse Initial Samples Help and Hurt Bayesian Optimizers

Eesh Kamrah
Dept. of Mechanical Engineering
University of Maryland
College Park, Maryland 20742
Email: kamrah@umd.edu

Seyede Fatemeh Ghoreishi
Dept. of Civil and Environmental Engineering
& Khoury College of Computer Sciences
Northeastern University
Boston, Massachusetts 02115
Email: f.ghoreishi@northeastern.edu

Zijian “Jason” Ding
College of Information Studies
University of Maryland
College Park, Maryland 20742
Email: ding@umd.edu

Joel Chan
College of Information Studies
University of Maryland
College Park, Maryland 20742
Email: joelchan@umd.edu

Mark Fuge[*]
Dept. of Mechanical Engineering
University of Maryland
College Park, Maryland 20742
Email: fuge@umd.edu

Design researchers have struggled to produce quantitative predictions for exactly why and when diversity might help or hinder design search efforts. This paper addresses that problem by studying one ubiquitously used search strategy—Bayesian Optimization (BO)—on a 2D test problem with modifiable convexity and difficulty. Specifically, we test how providing diverse versus non-diverse initial samples to BO affects its performance during search and introduce a fast ranked-DPP method for computing diverse sets, which we need to detect sets of highly diverse or non-diverse initial samples.

We initially found, to our surprise, that diversity did not appear to affect BO, neither helping nor hurting the optimizer's convergence. However, follow-on experiments illuminated a key trade-off. Non-diverse initial samples hastened posterior convergence for the underlying model hyper-parameters—a Model Building advantage. In contrast, diverse initial samples accelerated exploring the function itself—a Space Exploration advantage. Both advantages help BO, but in different ways, and the initial sample diversity directly modulates how BO trades those advantages. Indeed, we show that fixing the BO hyper-parameters removes the Model Building advantage, causing diverse initial samples to always outperform models trained with non-diverse samples. These findings shed light on why, at least for BO-type optimizers, the use of diversity has mixed effects and cautions against the ubiquitous use of space-filling initializations in BO. To the extent that humans use explore-exploit search strategies similar to BO, our results provide a testable conjecture for why and when diversity may affect human-subject or design team experiments.

1 INTRODUCTION AND RELATED WORK

One open question within design research is when or under what conditions providing diverse stimuli or starting solutions to either humans or algorithms can improve their designs' final performance. Researchers have struggled to produce quantitative predictions or explanations for exactly why and when diversity might help or hinder design search efforts. In studies of human designers or teams, there have been numerous empirical results on the effect of diverse stimuli or sets of stimuli on designers, typically referred to under the topic of *Design Fixation* (for recent reviews, see [1] and [2]). In general, available empirical results are mixed and it is difficult to quantitatively predict, for a new problem or person, whether or not diversity in problem stimuli will or will not help. For instance, there are a number of empirical demonstrations of positive effects of example diversity on novelty and diversity of ideas [3–5], but substantially more mixed results on the effects of diversity on solution *quality*,

[*] Address all correspondence to this author.

with some observations of positive effects [6–8], some null or contingent effects [4, 9–14], and even some negative effects on solution quality [15, 16].

Likewise, in research focused purely on optimization, common academic and industrial practice initializes search algorithms with different strategies like Latin Hypercube Sampling (LHS) [17] and others in an attempt to "fill" or "cover" a space as uniformly as possible [18] or via quasi-random methods [19–21]. Some methods build diversity-encouraging loss functions directly into their core search algorithms, such as in common meta-heuristic optimizers [22] such as Particle Swarm Optimization (PSO), Simulated Annealing (SA), and Genetic Algorithms (GA), with one of the most well-known diversity-inducing ones being NSGA-II [23]. For BO specifically, a common strategy is to build diversity directly into the acquisition function used in sampling new points from the Gaussian Process posterior [24]. As with human-subjects experiments, the precise effect of diversity on optimization performance is often problem dependent [22] and difficult to predict apriori. Nevertheless, optimization practitioners take these steps to improve initial sample diversity with the hope that the optimizer will converge faster or find better global optima.

But is encouraging initial diversity in this way always a good idea? If so, when and why is it good? Are there any times or conditions when diversity might hurt rather than help our search for good designs?
(Spoiler Alert: Yes, it can—see **S**4 for how and **S**6 for why.)

To address the above questions, this paper studies one type of commonly used search strategy—Bayesian Optimization (BO)—and how the diversity of its initialization points affects its performance on a search task. We uncover a fascinating dance that occurs between two competing advantages that initial samples endow upon BO—a *Model Building* versus *Space Exploration* advantage that we define later—and how the initial samples' diversity directs the choreography. While the fundamental reason for this interplay will later appear straightforward (and perhaps even discernible through thought experiments rather than numerical experiments), it nevertheless flies in the face of how most practitioners initialize their BO routines or conduct Optimal Experimental Design studies. It also posits a testable prediction about how to induce greater effects of diversity on novice human designers or the conditions under which there may be mixed or even negative effects (see **S**6).

Before describing our particular experiment and results, we will first review why BO is a meaningful and generalizable class of search algorithm to use, as well as past work that has tried to understand how diversity affects search processes such as optimization.

Why model design search as Bayesian optimization? While this paper addresses only BO, this is an important algorithm in that it plays an out-sized role within the design research and optimization community. For example, BO underlies a vast number of industrially-relevant gradient-free surrogate modeling approaches implemented in major design or analysis packages, where it is referred to under a variety of names, including Kriging methods or meta-modeling [25, 26]. Its use in applications of computationally expensive multidisciplinary optimization problems is, while not unilateral [27], quite widespread. Likewise, researchers studying human designers often use BO as a proxy model [28] to understand human search, due to the interplay between exploration and exploitation that lies at the heart of most BO acquisition functions like Expected Improvement. More generally, there is a robust history of fruitful research in cognitive science modeling human cognition as Bayesian processing [29], such as concept learning in cognitive development [30], causal learning [31], and analogical reasoning [32].

While the bulk of BO-related papers focus on new algorithms or acquisition functions, few papers focus on how BO is initialized, preferring instead the general use of space-filling initializations that have a long history in the field of Optimal Experiment Design [27]. In contrast, this paper shows that in certain situations that faith in space-filling designs might be misplaced, particularly when the BO kernel hyper-parameters are adjusted or fit during search.

What does it even mean for samples to be diverse? As a practical matter, if we wish to study how diverse samples impact BO, we face a subtle but surprisingly non-trivial problem: how exactly do you quantify whether one set of samples is more or less diverse than another? This is a set-based (*i.e.*, combinatorially large) problem with its own rich history too large to cover extensively here, however our past work on diversity measurement [33–35], computation [36], and optimization [37, 38] provides further pointers for interested readers, and in particular the thesis of Ahmed provides a good starting point for the broader literature and background in this area [39].

For the purposes of understanding how this paper relates to existing approaches, it suffices to know the following regarding common approaches to quantifying diversity: (1) most diversity measurement approaches focus on some variant of a hyper-volume objective spanned by the set of selected points; (2) since this measure depends on *a set* rather than individual points, it becomes combinatorially expensive, necessitating fast polynomial-time approximation, one common tool for which is a Determinantal Point Process (DPP) [40]; however, (3) while sampling the most diverse set via DPPs is easy, sampling percentile sets from the DPP distribution to get the top 5%, median, or lowest 5% of diverse sets becomes exceedingly slow for a large sample pool.

In contrast, for this paper, we created a faster DPP-type sampling method to extract different percentiles of the distribution without actually needing to observe the entire DPP distribution and whose sampling error we can bound using concentration inequalities. Section 2 provides further mathematical background, including information on DPP hyperparameters and how to select them intelligently, and the Supplemental Material provides further algorithmic details. With an understanding of diversity distribution measures in hand, we can now address diversity's specific effects on optimization more generally.

How does diversity in initial inputs affect optimizers? While there are a number of papers that propose either different initialization strategies or benchmarking of existing strategies for optimization, there is limited prior work addressing the direct effect of initial sample diversity.

For general reviews and benchmarking on how to initialize optimizers and the effects of different strategies, papers such as [20, 22] compare initialization strategies for particular optimizers and quantify performance differences. An overall observation across these contributions is the inability of a single initialization method to improve performance across functions of varying complexity. These studies also do not directly measure or address the role of sample diversity directly, only noting such behavior as it correlates indirectly with the sampling strategy.

A second body of work tries to customize initialization strategies on a per-problem basis, often achieving faster convergence on domain-specific problems [18, 19, 41–43]. While useful in their designed domain, these studies do not directly address the role of diversity either. In contrast, this paper addresses diversity directly using properties of BO that are sufficiently general to apply across multiple domains and applications.

Lastly, how to initialize optimizers has garnered new interest from the machine learning community, for example in the initial settings of weights and biases in a Neural Network and the downstream effects on network performance [44,45]. There is also general interest in how to collect diverse samples during learning, either in an Active Learning [46] or Reinforcement Learning context [47, 48]; however, those lines of work address only diversity throughout data collection, rather than the impact of initial samples considered in this paper.

What does this paper contribute beyond past work? This paper's specific contributions are:

1. To compute diversity: we describe a fast DPP-based diversity scoring method for selecting diverse initial examples with a fixed size k. Any set of size k with these initial examples can be then used to approximate the percentile of diversity that the set belongs to. This method requires selecting a hyper-parameter relating to the DPP measure. We describe a principled method for selecting this parameter in Section 2.1, and provide numerical evidence of the improved sampling performance in the Supplemental Material. Compared to prior work, this makes percentile sampling of DPP distributions computationally tractable.
2. To study effects on BO: we empirically evaluate how diverse initial samples affect the convergence rate of a Bayesian Optimizer. Section 4 finds that low diversity samples provide a *Model Building* advantage to BO while diverse samples provide a *Space Exploration* advantage that helps BO converge faster. Section 5 shows that removing the model building advantage makes having diverse initial samples uniformly better than non-diverse samples.[1]

We will next describe our overall experimental approach and common procedures used across all three of our main experiments. We will introduce individual experiment-specific methods only when relevant in each separate experiment section.

2 OVERALL EXPERIMENTAL APPROACH

This section will first describe how we compute diverse initial samples, including how we set a key hyper-parameter that controls the DPP kernel needed to measure sample set diversity. It then briefly describes the controllable 2D test problem that we use in our experiments. It ends with a description of how we set up the BO search process and the hyper-parameters that we study more deeply in each individual experiment.

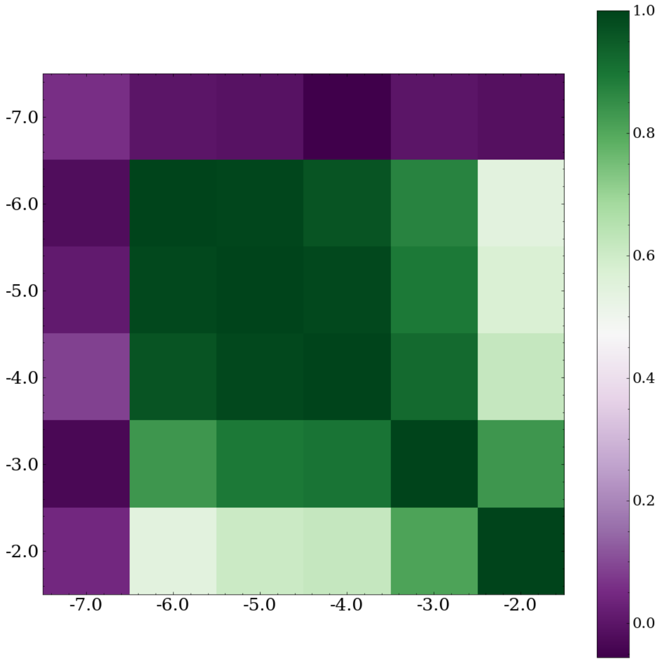


Fig. 1: Correlation matrix showing the relative correlation between two gammas by comparing the way our DPP approach ranks 10,000 sampled sets of cardinality k=10. The gamma values in both axes here are logarithmic values with base 10.

2.1 Measuring and Sampling from Diverse Sets using Determinantal Point Processes

As mentioned above, we measure diversity of a set of points using Determinantal Point Processes (DPP), which get their name from the fact that they compute the Determinant

[1] For grammatical simplicity and narrative flow, we will use the phrase "non-diverse" throughout the paper to refer to cases where samples are taken from the 5th percentile of diverse sets—these are technically "low-diversity" rather than being absolutely "non-diverse" which would occur when all points in the set are identical, but we trust that readers can keep this minor semantic distinction in mind.

of a matrix referred to as an *L-ensemble* (as seen in Eq. 1) that correlates with the volume spanned by a collection or set of samples (Y) taken from all possible sets ($\mathcal{Y}$) given a diversity/similarity (feature) metric.

$$P(\mathbb{L}_Y) \propto det(K(\mathbb{L}_Y)) \tag{1}$$

Here $\mathbb{L}$ is the ensemble defined by any positive semi-definite matrix [40], and K is the kernel matrix. For sampling diverse examples, this positive semi-definite matrix is typically chosen as a kernel matrix (K) that defines the similarity measure between pairs of data points. For this paper, we use a standard and commonly used similarity measure defined using a Radial Basis Function (RBF) kernel matrix [49]. Specifically, each entry in $\mathbb{L}_Y$ for two data points with index i and j is:

$$[\mathbb{L}_Y]_{i,j} = \exp\left(-\gamma \cdot ||\mathbf{x}_i - \mathbf{x}_j||^2\right) \tag{2}$$

The hyper-parameter γ in the DPP kernel can be set in the interval $(0, \infty)$ and will turn out to be quite important in how well we can measure diversity. The next section explores this choice in more depth, but to provide some initial intuition: set γ too high and any selection of points looks equally diverse compared to any other set, essentially destroying the discriminative power of the DPP, while setting γ too low causes the determinant of $\mathbb{L}$ to collapse to zero for any set of cardinality greater than the feature-length of $\mathbf{x}$.

With $\mathbb{L}$ in hand, we can now turn Eq. 1 into an equality by using the fact that $\sum_{Y \subset \mathcal{Y}} det(\mathbb{L}_Y) = det(\mathbb{L} + I)$, where I is an identity matrix of the same shape as the ensemble matrix $\mathbb{L}$. Then, using Theorem 2.2 from [40], we can write the $P(Y \in \mathcal{Y})$ as follows:

$$P(Y) = \frac{det(\mathbb{L}_{\mathbb{Y}})}{det(\mathbb{L} + I)} \tag{3}$$

This is the probability that a given set of points (Y) is highly diverse compared to other possible sets ($\mathcal{Y}$)—that is, the higher $P(Y)$ the more diverse the set. The popularity of DPP-type measures is due to their ability to efficiently sample diverse samples of fixed size k. Sampling a set of k samples from a DPP is done using a conditional DPP called k-DPP [50]. k-DPP are able to compute marginal and conditional probabilities with polynomial complexity, in turn allowing sampling from the DPP in polynomial complexity. k-DPPs are also well researched and there exists several different methods to speed up the sampling process using a k-DPP [51, 52]. Our approach allows sampling in constant complexity however there is a trade-off in complexity in generating the DPP distribution. The complexity for generating traditional DPP distributions is independent of 'k', while our approach has linear dependence on 'k'. Since, existing k-DPP approaches lack the ability to efficiently sample from different percentiles of diversity and thus make it computationally expensive to regenerate the distribution to alternatively sample from different percentiles.

To tackle this problem, our approach is designed to sample efficiently from different percentiles of diversity. This is made possible by creating an absolute diversity score. This score is generated by taking a *logdeterminant* of the kernel matrix defined over the set Y. Randomly sampling the k-DPP space allows us to bound errors in generating this absolute score through the use of concentration inequalities. The details about how to sample from this distribution and calculate the score are mentioned in the supplementary material, so as not to disrupt the paper's main narrative. Additionally, the supplementary material provides empirical results to support our earlier claims regarding efficient sampling from our approach vs the traditional k-DPP approach, as well as the trade-off in complexity when generating the DPP distribution. Figure 2 shows example sets of five points and their corresponding DPP score, where the diversity score is monotonic and a positive score corresponds to a more diverse subset.

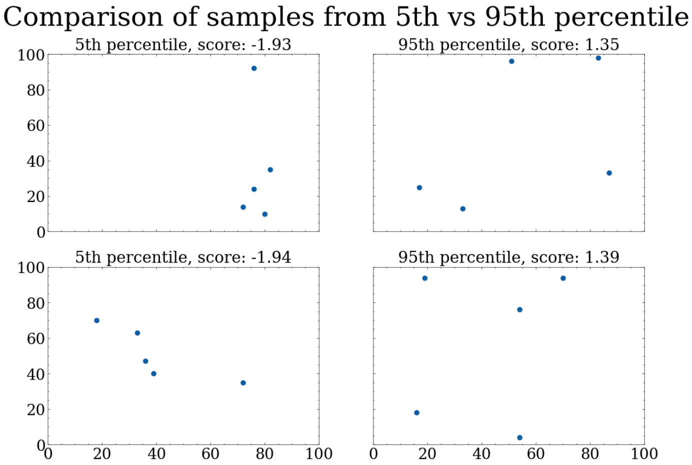


Fig. 2: Scatter plots showing randomly chosen sets with k=5 High Diversity and Low Diversity samples with their diversity score on top of each of the chosen set.

2.1.1 Selecting the hyper-parameter for the DPP kernel

As mentioned above, the choice of γ impacts the accuracy of the DPP score, and when we initially fixed γ to $\frac{|Y_i|}{10}$, where Y_i is the set of data points over which the RBF kernel is calculating the DPP score as suggested by [53], the DPP seemed to be producing largely random scores. To select an appropriate γ we designed a kernel-independent diagnostic method for assessing the DPP kernel with four steps.

First, we randomly generate M samples of size k sets (think of these as random k-sized samples from $\mathcal{Y}$). Second, we compute their DPP scores for different possible γ values and then sort those M sets by that score. Third, we compute the rank correlation among these sets for different pairs of γ—intuitively, if the rank correlation is high (toward 1) then either choice of γ would produce the same rank orders of which points were considered diverse, meaning the (relative) DPP scores are insensitive to γ. In contrast, if the rank correlation is 0, then the two γ values produce essentially random orderings. This rank correlation between two different γ settings is the color/value shown in each cell of the matrix in Fig. 1. Large ranges of γ with high-rank correlation mean that the rankings of DPP scores are stable or robust to small perturbations in γ. Lastly, we use this "robust γ" region by choosing the largest range of γ values that have a relative correlation index of 0.95 or higher. We compute the mean of this range and use that as our selected γ in our later experiments. We should note that the functional range of γ is dependent on sample size (k), and so this "robust γ" needs to be recomputed for different initialization sizes.

The detailed settings for the results as seen in Figure 1 are as follows: the $M = 10000$; $k = 10$; $\gamma \in [e-7, e-2]$. The correlation matrix shows a range of γ with strongly correlating relative ordering of the test sets. All γ within this range provide a consistent ranking.

2.2 A Test Function with Tunable Complexity

A problem that is common across the study of initialization methods is their inconsistency across problems of varying difficulty, motivating the need to test BO's search behavior on a problem class with variable complexity. Synthetic objective functions are often used to test the efficiency of different optimizers and there are several libraries online to choose these functions from [54], though these functions are largely static, in the sense that there is only a single test function definition. There has been research into developing objective function *generators*; for example, in [55], the author uses a mixture of four features to generate synthetic objective functions. These have been well categorized and the relative performance of different optimizers documented on each landscape. Similar to this, [56] looks at using a mixture of different sinusoidal functions to create a noisy 1-D function. Both the generators discussed are capable of generating complicated landscapes, but the complexity arises from mixing different randomly generated sinusoids and thus are unable to control or quantify a measure of complexity of the generated landscapes.

To address this controllable complexity problem directly, we created a simple 2D test function generator with tunable complexity parameters that allow us to instantiate multiple random surfaces of similar optimization difficulty. We modified this function from the one used in [57] where it was referred to as "Wildcat Wells", though the landscape is functionally just a normal distribution with additive noise of different frequency spectra. We used four factors to control the synthetic objective functions: 1) the number of peaks, 2) noise amplitude, 3) smoothness, and 4) distance between peaks and a seed. The number of peaks control the number of layers of multivariate normal with single peaks. The noise amplitude in the range of [0,1] controls the relative height of the noise compared to the height of the peaks. Setting this to 1 would essentially make the noise in the function as tall as the peaks and give the function infinite peaks. Smoothness in the range of [0,1] controls the weighted contribution of the smooth Gaussian function compared to the rugged noise to the wildcat-wells landscape. Setting this to 1 would remove the noise from the function because then the normal distribution completely controls and dominates the function. The last parameter, the distance between peaks, can be tuned in the range of [0,1]. This parameter prevents overlap of peaks when the function is generated with more than 1 peak.

Some of these parameters overlap in their effects. For example, N controls the number of peaks, and ruggedness amplitude controls the height of the noise in the function, so increasing the noise automatically increases the peaks in the function thus we will only look at varying the ruggedness amplitude. Apart from this, ruggedness frequency (the distance between peaks) plays the same role as smoothness (radius of influence of each individual on its neighbor). Thus, for the numerical experiments presented in Sections 3–5 only the ruggedness amplitude and smoothness will be varied between $[0.2, 0.8]$ with increments of 0.2. To provide some visual examples of the effect of these parameters on the generated functions, Fig. 3 visualizes an example random surface generated with different smoothness and ruggedness amplitude parameters.

2.3 Bayesian optimization

Bayesian optimization (BO) has emerged as a popular sample-efficient approach for optimization of these expensive black-box (BB) functions. BO models the black-box function using a surrogate model, typically a Gaussian process (GP). The next design to evaluate is then selected according to an acquisition function. The acquisition function uses the GP posterior and makes the next recommendation for function evaluation by balancing between exploration and exploitation. It allows exploration of regions with high uncertainty in the objective function, and exploitation of regions where the mean of the objective function is optimum. At each iteration, the GP gets updated according to the selected sample, and this process continues iteratively according to the available budget.

Each data point in the context of Bayesian optimization is extremely expensive; thus, there is a need for selection of an informative set of initial samples for the optimization process. Toward this, this paper investigates the effect of level of initial diverse coverage of the input space on convergence of Bayesian optimization policies.

For the purpose of numerical experiments, the optimizer used is from the BOTorch Library [58]. The optimizer uses a Single Task GP Model with Expected Improvement; the kernel used is a Matérn kernel.

A GP is specified by its mean and covariance functions, as:

$$f(\mathbf{x}) \sim \mathcal{GP}\left(\mu(\mathbf{x}), k(\mathbf{x}, \mathbf{x})\right), \tag{4}$$

where $\mu(.)$ and $k(.,.)$ are the mean function and a real-valued kernel function encoding the prior belief on the correlation among the samples in the design space. In Gaussian process regression, the kernel function dictates the structure of the surrogate model we can fit. An important kernel for

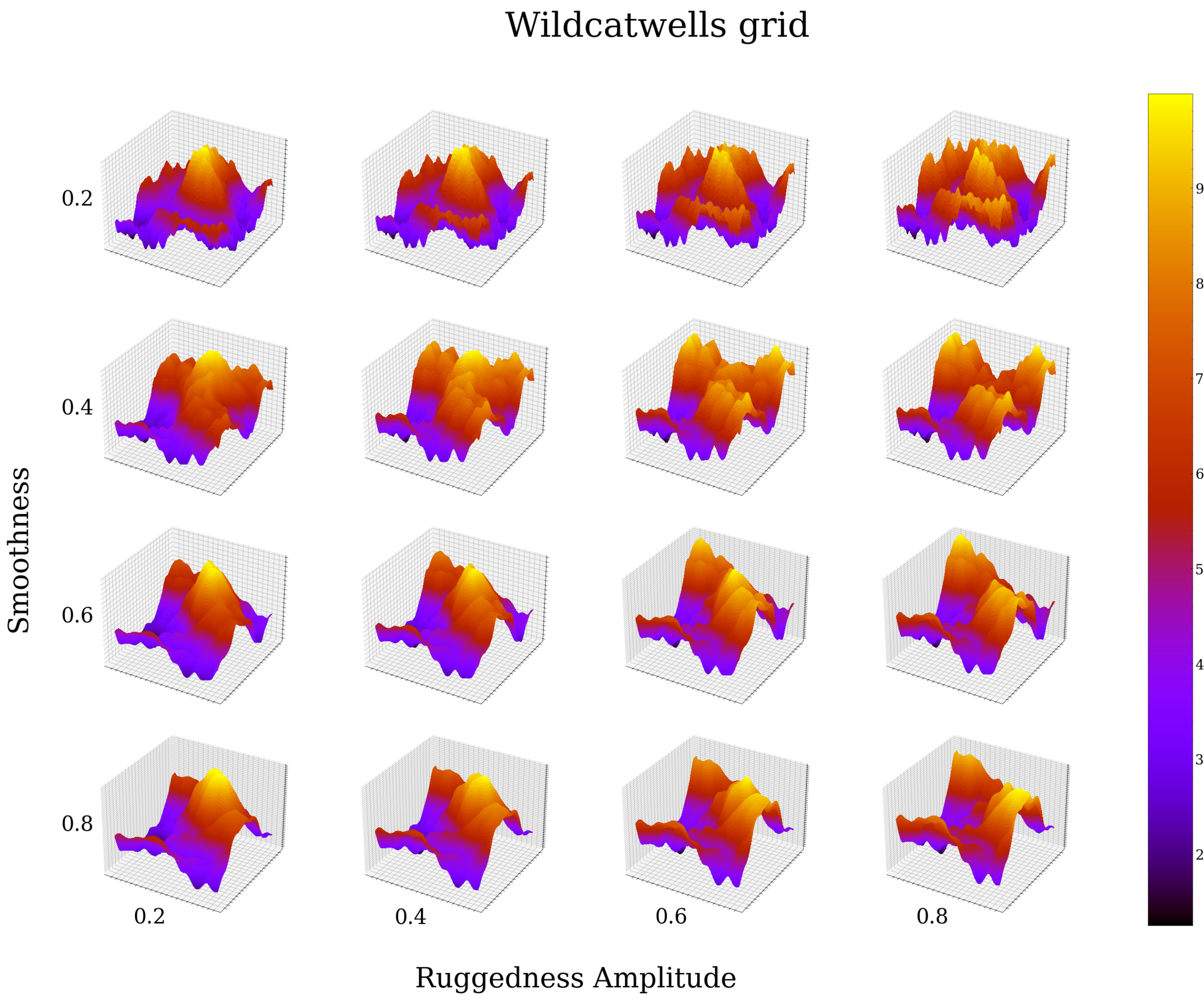


Fig. 3: A Grid plot showing how the landscape of wildcat wells changes with smoothness and ruggedness amplitude.

Bayesian optimization is the Matérn kernel, which incorporates a smoothness parameter ν to permit greater flexibility in modeling functions:

$$k_{\text{Matérn}}(\mathbf{x}_1, \mathbf{x}_2) = \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\sqrt{2\nu} \frac{||\mathbf{x}_1 - \mathbf{x}_2||}{\theta} \right)^{\nu} H_\nu \left(\sqrt{2\nu} \frac{||\mathbf{x}_1 - \mathbf{x}_2||}{\theta} \right), \tag{5}$$

where $\Gamma(.)$ and $H_\nu(.)$ are the Gamma function and the Bessel function of order ν, and θ is the length-scale hyper-parameter which denotes the correlation between the points within each dimension and specifies the distance that the points in the design space influence one another. Here, we use a constant mean for the mean function. The *Model Building* advantage that we refer to in this paper corresponds to learning these hyper-parameters. The hyper-parameters of the Gaussian process, namely, the parameters of the kernel function and the mean function are:

Lengthscale of the Matérn Kernel In Eq. 5, where θ is the lengthscale parameter of the kernel. This parameter controls the ruggedness expected by the Bayesian optimizer in the black box function being studied.

The effects of the parameter are similar to ν, but ν is not learned during the optimization process while lengthscale is. So, ν is not studied as a parameter that influences the modeling behavior but rather studied as an additional parameter for sensitivity.

Output scale of Scale Kernel Output scale is used to control how the Matérn kernel is scaled for each batch. Since our Bayesian optimizer uses a single task GP, we do not use batch optimization. Thus, this parameter is unique for us and the way it's implemented using BoTorch can be seen Equation 6.

$$K_{scaled} = \theta_{scale} K_{orig} \tag{6}$$

Noise for likelihood calculations The noise parameter is used to model measurement error or noise in the data. So, as the Gaussian Process gets more data the noise term decreases. So, ideally, this term should converge to 0 when the

Bayesian optimizer has found an optimal value since our test functions did not have any added noise.

Constant for Mean Module This constant is used as the mean for the Normal distribution that forms the prior of the Gaussian Process as shown in Equation 4.

Further studies and results regarding the effects of the hyper-parameters are available in the Supplemental Material.

We now describe the first experiment where we explore the effects of diversity of initial samples on the convergence of Bayesian Optimizers.

3 EXPERIMENT 1: DOES DIVERSITY AFFECT OPTIMIZATION CONVERGENCE?

3.1 Methods

To test the effects of diversity of initial samples on optimizer convergence, we first generated a set of initial training samples of size (k) 10 either from low (5^{th} percentile of diversity) or high diversity (95^{th} percentile of diversity) using our procedure in **S**2.1. Next, we created 100 different instances of the wildcat wells function with different randomly generated seeds for each cell in a 4x4 factor grid of 4 values each of the smoothness and ruggedness amplitude parameters of the wildcat wells function (ranging from 0.2 to 0.8, in steps of 0.2). For simplicity here, we refer to these combinations as families of the wildcat wells function. This resulted in 1600 function instances.

Our experiment consisted of 200 runs of the Bayesian Optimizer within each of the smoothness-ruggedness function families, where each run consisted of 100 iterations, and half of the runs were initialized with a low-diversity training sample, and half were initialized with a high-diversity training sample.

We then compared the cumulative optimality gap across the iterations for the runs with low-diverse initializations and high-diverse initalizations within each smoothness-ruggedness combination family. We did this by computing bootstrapped mean and confidence intervals within each low-diverse and high-diverse sets of runs within each family. Given the full convergence data, we compute a Cumulative Optimality Gap (COG) which is just the area under the Optimality Gap curve for both the 5^{th} and 95^{th} diversity curves. Intuitively, a larger COG corresponds to a worse overall performance by the optimizer. Using these COG values we can numerically calculate the improvement of the optimizer in the 95^{th} percentile. The net improvement of COG value while comparing the 5^{th} and 95^{th} percentile is also presented as text in each subplot in Figure 4.

3.2 Results

As Figure 4 shows, the Cumulative Optimality Gap does not seem to have a consistent effect across the grid. Diversity produces a positive convergence effect for some cells, but is negative in others. Moreover, there are wide empirical confidence bounds on the mean effect overall, indicating that should an effect exist at all, it likely does not have a large effect size. Changing the function ruggedness or smoothness did not significantly modulate the overall effect. As expected, given sufficient samples (far right on the x-axis) both diverse and non-diverse initializations have the same optimality gap, since at that point the initial samples have been crowded out by the new samples gathered by BO during its search.

3.3 Discussion

Overall, the results from Fig. 4 seem to indicate that diversity helps in some cases and hurts in others, and regardless has a limited impact one way or the other. This seems counter to the widespread practice of diversely sampling the initial input space using techniques like LHS. Figure 4 shows that it has little effect.

Why would this be? Given decades of research into initialization schemes for BO and Optimal Experiment Design, we expected diversity to have at least some (perhaps small but at least consistent) positive effect on convergence rates, and not the mixed bag that we see in Fig. 4. How were the non-diverse samples gaining such an upper hand when the diverse samples had a head start on exploring the space—what we call a *Space Exploration* advantage?

The next section details an experiment we conducted to test a hypothesis regarding a potential implicit advantage that non-diverse samples might endow to BO that would impact the convergence of BO's hyper-parameter posteriors. As we will see next, this accelerated hyper-parameter posterior convergence caused by non-diverse initialization is the Achilles' heel of diversely initialized BO that allows the non-diverse samples to keep pace and even exceed diverse BO.

4 EXPERIMENT 2: DO LOWER DIVERSITY SAMPLES IMPROVE HYPER-PARAMETER POSTERIOR CONVERGENCE?

After reviewing the results from Fig. 4, we tried to determine why the Space Exploration advantage of diversity was not helping BO as we thought it should. We considered as a thought experiment the one instance where a poorly initialized BO model with the same acquisition function might outperform another: if one model's kernel hyper-parameter settings were so grossly incorrect that the model would waste many samples exploring areas that it did not need to if it had the correct hyper-parameters.

Could this misstep be happening in the diversely sampled BO but not in the non-diverse case? If so, this might explain how non-diverse BO was able to keep pace: while diverse samples might give BO a head start, it might be unintentionally blindfolding BO to the true function posteriors, making it run ragged in proverbial directions that it need not. If this hypothesis was true, then we would see this reflected in the comparative accuracy of the kernel hyper-parameters learned by the diverse versus non-diverse BO samples. This experiment set out to test that hypothesis.

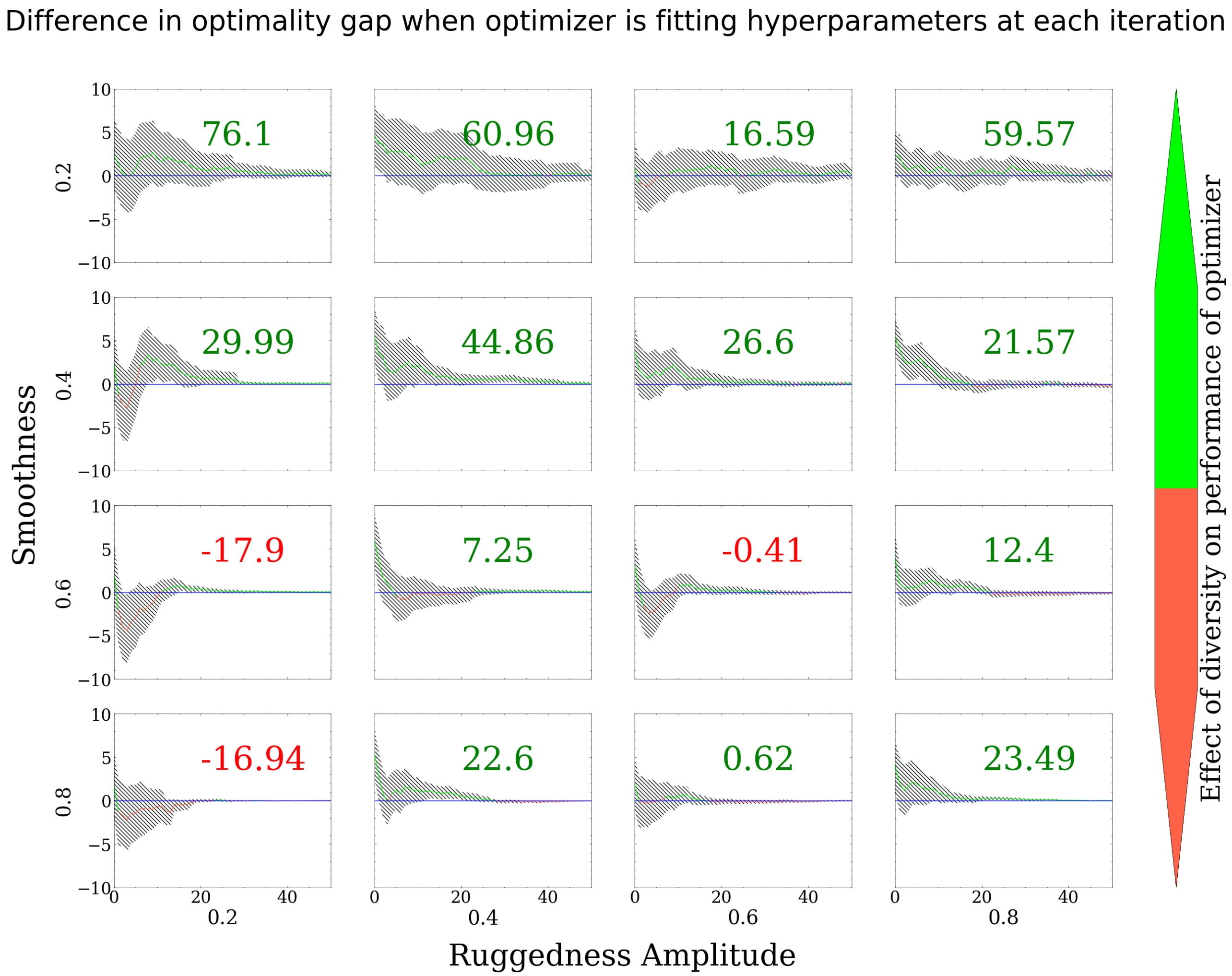


Fig. 4: Experiment 1: Optimality gap grid plot showing the difference in current Optimality Gap between optimizers initialized with 5th vs 95th percentile diverse sample (y-axis) as a function of optimization iteration (x-axis). The different factors in the factor grid plot the effects of diversity as the noise amplitude and smoothness are varied in the range [0.2,0.8]. Each plot also has text indicating the Net Cumulative Optimality Gap (NCOG), a positive value corresponds to a better performance by high diversity samples compared to the low diversity samples. The plot shows that BO benefits from diversity in some cases but not others. There is no obvious trends in how the NCOG values change in the grid. The results are further discussed in **S**3

4.1 Methods

The key difference from Experiment 1 is that, rather than comparing the overall optimization convergence, we instead focus on how the initial samples' diversity affects BO's hyper-parameter posterior convergence, and compare how far each is from the "ground truth" optimal hyperparameters.

As with Experiment 1, we used the same smoothness and ruggedness amplitude families of the wildcat wells function. To then generate the data for each instance in one of these families, we sampled 20 sets of initial samples. Half of the sampled 20 sets were low (5^{th} percentile of diversity) and the other half from high diversity (95^{th} percentile of diversity) percentiles.

For each initial sample, we then maximized the GP's kernel Marginal Log Likelihood (via BOTorch's GP fit method). We then recorded the hyper-parameters obtained for all 20 initial samples. The mean of the 10 samples from low diversity was then used as one point in the box plot's low diversity distribution as seen in Fig. 5. We then repeated this process for the high diversity initial samples. Each point in the box plot can be then understood as the mean hyper-parameter learned by BOTorch given just the initial sample of size (k) 10 points. To get the full box plot distribution for each family the above process is repeated over 100 seeds and Fig. 5 provides the resulting box plot for both diverse and non-diverse initial samples for all the 16 families of wildcat wells function as described in Experiment 1.

To provide a ground truth for the true hyper-parameter settings, we ran a Binary search to find the size of the sample ($k_{optimal}$) for which BO's kernel hyper-parameters converged for all families. The hyper-parameter found by providing $k_{optimal}$ amount of points for each instance in the family was then plotted as a horizontal line in each box plot. An interesting observation is that some families have non-overlapping horizontal lines. This is because for some families there are more than one modes of 'optimal hyper-parameters'. The mode chosen as the 'optimal hyper-parameter' is the more observed mode. The process for finding the 'optimal hyper-parameter' and which mode is chosen as the optimal hyper-parameter has been described in the Supplemental Material.

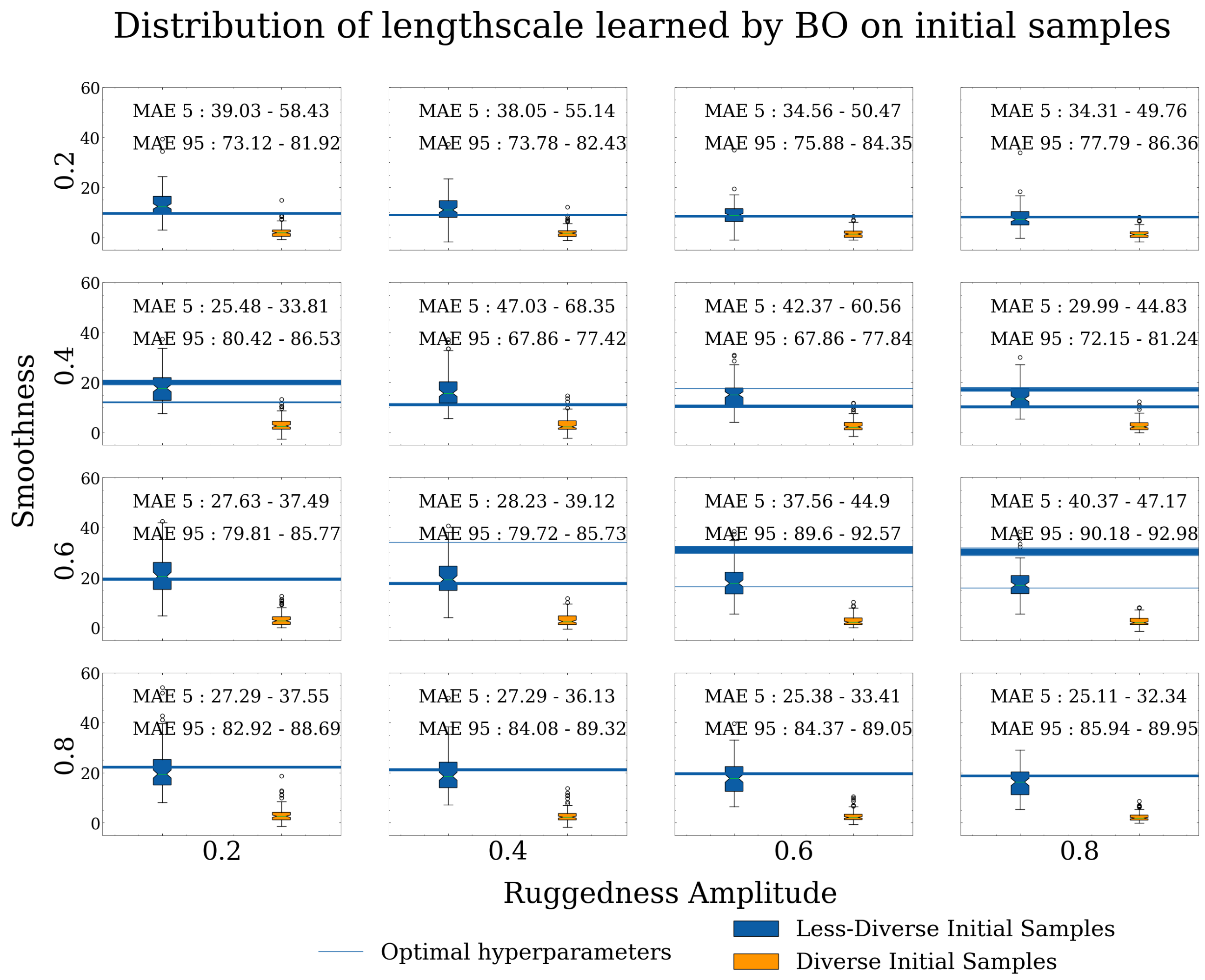


Fig. 5: Experiment 2: Box plot showing distribution of 'Lengthscale' hyper-parameter learned by BO when initiated with diverse (orange) and non-diverse samples (blue) for 16 different families of wildcat wells functions of the same parameters but 100 different seeds. The optimal hyper-parameter for each of the 100 wildcat wells instances from each family is also plotted as horizontal (blue) lines—in many but not all cases these overlap. Each cell in the plot also has the 95th percentile confidence bound on Mean Absolute Error (MAE) for both diverse and non-diverse samples. The results show that MAE confidence bounds for non-diverse samples are smaller compared to diverse samples for all the families of wildcat wells function. Thus, indicating a presence of Model Building advantage for non-diverse initial samples. The results of this figure are further discussed in **S**4

If an initial sample provides a good initial estimate of the kernel hyper-parameter posterior, then the box plot should align well or close to the horizontal lines of the true posterior. Figure 5 only shows the results for the Matérn Kernel's Lengthscale parameter, given its out-sized importance in controlling the GP function posteriors compared to the other hyper-parameters (*e.g.*, output scale, noise, *etc.*), which we do not plot here for space reasons. We provide further details and plots for all hyper-parameters in the Supplementary Material for interested readers.

To quantify the average distance between the learned and true hyper-parameters, we also plot on Fig. 5 the Mean Absolute Error (MAE) for both highly diverse (95th) and less diverse (5th) points. The MAE is the sum of the absolute distance of each predicted hyper-parameter from the optimal hyper-parameter for the particular surface of each wildcat wells function. The range as seen in each cell in Figure 5 corresponds to a 95th percentile confidence bound on the Mean absolute error across all the 100 runs.

4.2 Results and Discussion

The results in Figure 5 show that the MAE values for low diversity samples are always lower compared to the MAE for high diversity samples. This general behavior is also qualitatively visible in the box plot. This means that after only the initial samples, the non-diverse samples provided much more accurate estimates of the kernel hyper-parameters compared to diverse samples. Moreover, BO systematically *underestimates* the correct lengthscale with diverse samples—this corresponds to the diverse BO modeling function posteriors that have higher frequency components than the true function actually does (as shown via the pedagogical examples in the Supplemental Material).

This provides evidence for the *Model Building* advantage of non-diverse samples that we defined in Sec. 2.3. It also confirms our previous conjecture from the thought ex-

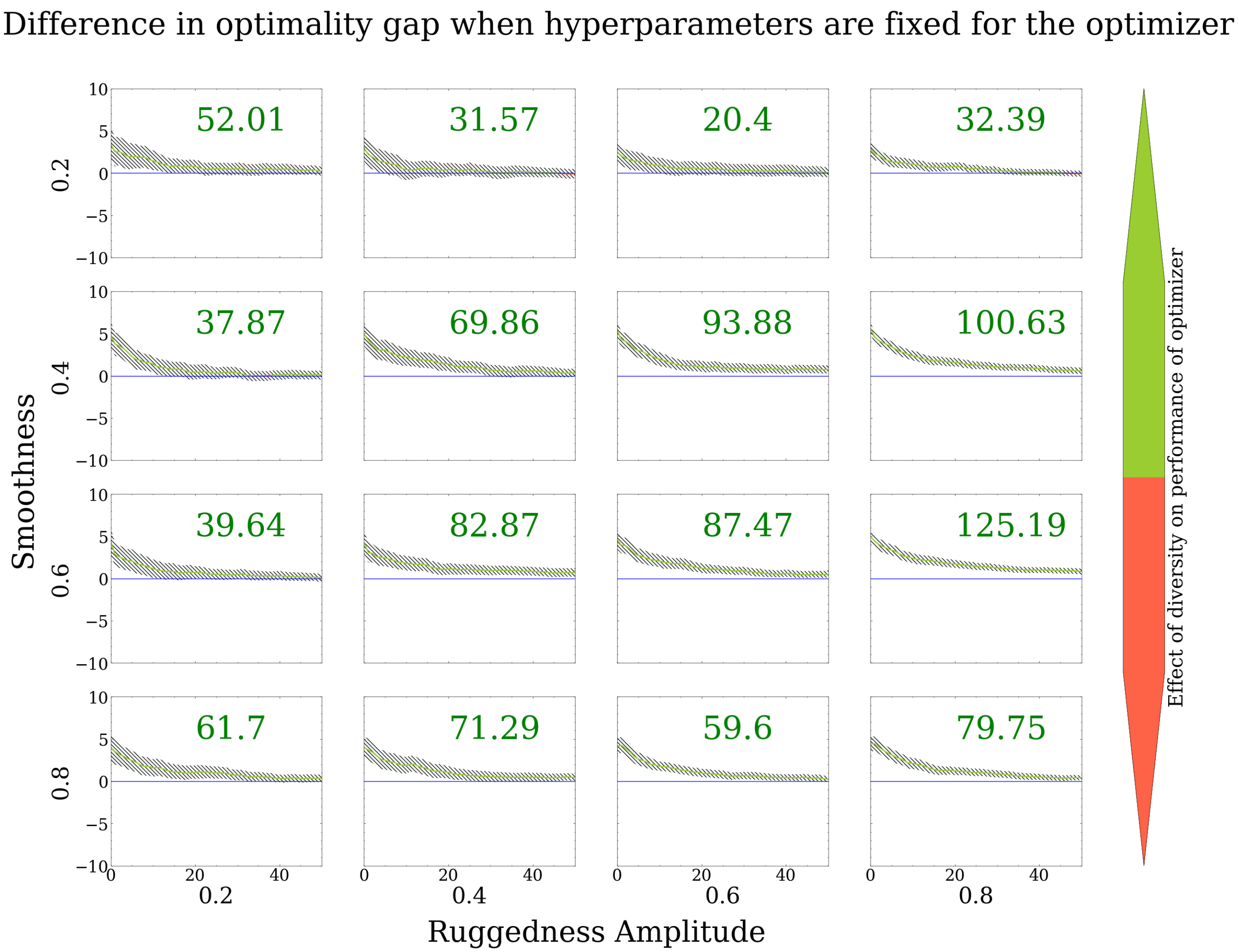


Fig. 6: Experiment 3: Optimality gap plot showing effects of diversity when the optimizer is not allowed to fit the hyper-parameters for the Gaussian Process and the hyper-parameters are instead fixed to the values found in Experiment 2. The results from this plot show positive NCOG values for all families of wildcat wells function, showing that once the Model Building advantage' is taken away the diverse samples outperform non-diverse samples. Further discussion on this plot can be read in **S**5

periment that diverse samples might be impacting BO by causing slower or less accurate convergence to the right BO hyper-parameters. The Space Exploration advantage of the diverse samples helps it compensate somewhat for its poor hyper-parameters, but BO trained with non-diverse samples can leverage the better hyper-parameters to make more judicious choices about what points to select next.

We did not see major differences in the other three kernel hyper-parameters such as Output Scale, Noise, or the Mean Function (see Supplemental Material); however, this is not surprising, since BO is not highly sensitive to any of these parameters and the lengthscale parameter dominates large changes in BO behavior.

Comparing the different smoothness and ruggedness settings, when the function is more complex (the top right of the grid at low smoothness and high ruggedness amplitude values) the function's lengthscale is lower and closer to the value learned by the diverse samples. Looking at the low diversity MAE values ('MAE 5'), we can see they are much closer to those of the high diversity samples ('MAE 95'), in contrast to when the function is less complex (bottom left side of the grid). Under such conditions, low diversity samples lose some of the relative Model Building advantage they have over high diversity samples. This conjecture aligns with Experiment 1 (Fig 4) where the COG values on the top right part are positive while those on the bottom left are negative.

Figure 5 demonstrated our hypothesized Model Building advantage that non-diverse initial samples confer to BO. But how do we know that this is the actual causal factor that accelerates BO convergence, and not just correlated with some other effect? If correct, our conjecture posits a natural testable hypothesis: if we fix the values of the hyperparameter posteriors to identical values between the non-diverse and diverse samples and do not allow the BO to update or optimize them, then this should effectively eliminate the Model Building advantage, and diverse samples should always outperform non-diverse samples. Metaphorically, if we were to take away the arrow that Paris used against Achilles, would the Battle of Troy have ended differently? Our next experiment finds this out.

5 EXPERIMENT 3: DOES DIVERSITY AFFECT OPTIMIZATION CONVERGENCE IF HYPER-PARAMETERS ARE FIXED TO OPTIMAL VALUES?

5.1 Methods

This experiment is identical to Experiment 1, with two key differences: (1) we now fix the kernel hyper-parameters to the 'optimal hyper-parameter' values we found in Experiment 2 for all the instances in each family of the wildcat wells function, (2) and we do not allow either BO model to further optimize the kernel hyper-parameters. This should remove the hypothesized Model Building advantage of non-diverse samples without altering any other aspects of Experiment 1 and the results in Fig. 4.

5.2 Results and Discussion

Figure 6 shows that once the kernel hyper-parameters are fixed—removing the Model Building advantage of non-diverse samples—diverse samples consistently and robustly outperform non-diverse initial samples. This holds for both the initial Optimality Gap at the beginning of the search as well as the Cumulative Optimality Gap and is not qualitatively affected by the function smoothness or roughness amplitude. Unlike in Experiment 1 where diversity could either help or hurt the optimizer, once we remove the Model Building advantage, diversity only helps.

6 GENERAL DISCUSSION AND CONCLUSIONS

6.1 Summary and Interpretation of Findings

This paper's original goal was to investigate how and when diverse initial samples help or hurt Bayesian Optimizers. Overall, we found that the initial diversity of the provided samples created two competing effects. First, Experiment 2 showed that non-diverse samples improved BO's abilities to quickly converge to optimal hyper-parameters—we called this a *Model Building* advantage. Second, Experiment 3 showed that conditioned on the same fixed hyper-parameters diverse samples improved BO's convergence to the optima through faster exploration of the space—we called this a *Space Exploration* advantage. In Experiment 1, diversity had mixed-to-negligible effects since both of these advantages were in play and competed with one another. This interaction provides insight for academic or industrial BO users since common practice recommends initializing BO with space-filling samples (to take advantage of the Space Exploration advantage), and ignores the Model Building advantage of non-diverse samples.

Beyond our main empirical result, our improvements to existing diverse sampling approaches (Sec. 2.1) provide new methods for studying how different percentile diversity sets affect phenomena. Researchers may find this contribution of separate technical and scientific interest for related studies that investigate the impact of diversity.

6.2 Implications and Future Work

Beyond the individual results we observed and summarized in each experiment, there are some overall implications and limitations that may guide future work or interpretation of our results more broadly, which we address below.

Where does this Model Building advantage induced by non-diverse samples come from? As we conjectured in Experiment 2 (**S**4), and confirmed in Experiment 3 (**S**5), the key advantage of using non-diverse initial samples lies in their ability to induce faster and more accurate posterior convergence when inferring the optimal kernel hyper-parameters, such as length scale and others. This allowed the BO to make more judicious and aggressive choices about what points to sample next, so while the diversely initialized models might get a head start on exploring the space, non-diversely initialized models needed to explore less of the space overall, owing to tighter posteriors of possible functions under the Gaussian Process.

While we do not have space to include it in the main paper, the supplemental material document's section 5 shows how this model building advantage occurs as we provide BO with a greater number of initial samples. Briefly, there are three "regimes": (1) sample-deficient, where there are too few samples to induce a modeling advantage regardless of how diversely we sample the initial points; (2) the "modeling advantage" region, where low-diversity samples induce better hyperparameter convergence than high-diversity samples; and (3) sample-saturated, where there are enough initial samples to induce accurate hyper-parameter posteriors regardless of how diversely we sample initial points. We direct interested readers to Section 5 of the supplemental material for a deeper discussion on this.

What this behavior implies more broadly is that non-diverse samples, whether given to an algorithm or a person, have a unique and perhaps underrated value in cases where we have high entropy priors over the Gaussian Process hyper-parameters or kernel. In such cases, sacrificing a few initial non-diverse points to better infer key length scales in the GP model may well be a worthwhile trade.

We also saw that in cases where the BO hyper-parameters were not further optimized (as in Experiment 3 where hyper-parameters were fixed to optimal values), using diverse points only helped BO. Researchers or practitioners using BO would benefit from carefully reviewing what kernel optimization strategy their library or implementation of choice actually does since that will affect whether or not the Model Building advantage of non-diverse samples is actually in play.

What if Hyper-parameters are fixed to non-optimal values? We showed in Experiment 3 that fixing BO hyper-parameters to their optimal values ahead of time using an oracle allowed diverse initial samples to unilaterally outperform non-diverse samples. An interesting avenue of future work that we did not explore here for scope reasons would be to see if this holds when hyper-parameters are fixed to non-optimal values. In practical problems, we will not often know the optimal hyper-parameters ahead of time as we did in Experiment 3 which caused diversity's unilateral advantage, so we do not have evidence to generalize beyond this. However, our explanation of the Model Building advantage would predict that, so long as the hyper-parameters remain fixed (to any value), BO would not have a practical mechanism to benefit much from non-diverse samples, on average.

What are the implications for how we currently initialize BO? One of our result's most striking implications is how it might influence BO initialization procedures that are often considered standard practice. For example, it is common to initialize a BO procedure with a small number of initial space-filling designs, using techniques like Latin Hypercube Sampling (LHS) before allowing BO to optimize its acquisition function for future samples. In cases where the BO hyper-parameters will remain fixed, Experiment 3 implies that this standard practice is excellent advice and far better than non-diverse samples. However, in cases where you plan to optimize the BO kernel during search, using something like LHS becomes more suspect.

In principle, from Experiment 1 we see that diverse samples may help or hurt BO, depending on how much leverage the Model Building advantage of the non-diverse samples can provide. For example, in the upper right of Fig. 4 the function is effectively random noise, and so there is not a strong Model Building advantage to be gained. In contrast, in the lower left, the smooth and well-behaved functions allowed non-diverse initialization to gain an upper hand.

Our results propose a perhaps now obvious initialization strategy: if you plan on optimizing the BO hyper-parameters, use some non-diverse samples to strategically provide an early Model Building advantage, while leveraging the rest of the samples to diversely cover the space.

How might other acquisition functions modulate diversity's effect? While we have been referring to BO as though it is a single method throughout this paper, individual BO implementations can vary, both in terms of their kernel structure and their choice of acquisition function—or how BO uses information about the underlying fitted Gaussian Process to select subsequent points. In this paper's experiments, we used Expected Improvement (EI) since it is one of the most widespread choices, and behaves qualitatively like other common improvement-based measures like Probability of Improvement, Posterior Mean, and Upper Confidence Bound functions. Indeed, we hypothesize that part of the reason non-diverse initial samples are able to gain a Model Building advantage over diverse samples is due to a faster collapse in the posterior distribution of possible GP functions which serves as strong input to EI methods and related variants.

Yet EI and its cousins are only one class of acquisition function; would our results hold if we were to pick an acquisition function that directly attacked the GP's posterior variance? For example, either Entropy-based or Active Learning based acquisition functions? This paper did not test this and it would be a logical and valuable future study. Our experimental results and proposed explanation would predict the following: the Model Building advantage seen by non-diverse samples should reduce or disappear in cases where the acquisition function explicitly samples new points to minimize the posterior over GP function classes since in such cases BO itself would try to select samples that reduced overall GP variance, reducing its dependence on what the initial samples provide.

To what extent should we expect these results to generalize to other types of problems? We selected a simple 2D function with controllable complexity in this paper to aid in experimental simplicity, speed, replicability, and ease of visualization; however, this does raise the question of whether or not these results would truly transfer to more complex problems of engineering interest. While future work would have to address more complex problems, we performed two additional experiments studying how the above phenonmena change as we (1) increased the wildcat wells function from two to three dimensions, and (2) how this behavior changes for other types of common optimization test functions—specifically, we chose the N-Dimensional Sphere, Rastrigin, and Rosenbrock functions from two to five dimensions. While the existing paper length limits did not allow us to include all of these additional results in the paper's main body, we direct interested readers to Sections 6 and 7 of the supplemental material document. Briefly, our results align overall with what we described above for the 2D wildcat wells function, and we do not believe that the phenomena we observed are restricted to only our chosen test function or dimension, although clearly future research would need to conduct further tests on other problems to say this with any certainty. Beyond these supplemental results, we can also look at a few critical problem-specific factors and ask what our proposed explanatory model would predict.

For higher dimensional problems, standard GP kernel choices like RBF or Matérn begin to face exponential cost increases due to how hyper-volumes expand. In such cases, having strong constraints (via hyper-parameter priors or posteriors) over possible GP functions becomes increasingly important for fast BO convergence. Our results would posit that any Model Building advantages from non-diverse sampling would become increasingly important or impactful in cases where it helped BO rapidly collapse the hyper-parameter posteriors.

For discontinuous functions (or GP kernels that assumed as much), the Model Building advantage of non-diverse samples would decrease since large sudden jumps in the GP posterior mean and variance would make it harder for BO to exploit a Model Building advantage. However, in discontinuous cases where there were still common global smoothness parameters that governed the continuous portions the Model Building advantage would still accelerate advantages for BO convergence.

How might the results guide human subject experiments or understanding of human designers? One possible implication of our results for human designers is that the effects of example diversity on design outcomes may vary as a function of designer's prior knowledge of the design problem. More specifically, the Model Building advantage observed in Experiment 2 (and subsequent removal in Experiment 3) suggests that when designers have prior knowledge of how quickly the function changes in a local area of the design space, they can more reliably benefit from the Space Exploration advantage of diverse examples. This leads to a potentially counter-intuitive prediction that domain experts may benefit more from diverse examples compared to domain novices since domain experts would tend to have prior knowledge of the nature of the design problem (a Model Building advantage). Additionally, perhaps under conditions of uncertainty about the nature of the design problem, it would be useful to combine the strengths of diverse and non-diverse examples; this could be accomplished with a cluster-sampling approach, where we sample diverse points of the design space, but include local non-diverse clusters of examples that are nearby, to facilitate learning of the shape of the design function.

While these implications might be counter-intuitive in that common guidance suggests that the most informative method is to only diversely sample initial points, the crux of our paper's argument is that non-diverse points *can*, surprisingly, be informative to Bayesian Optimization due to their ability to quickly concentrate the posterior distribution of the kernel hyper-parameters, and thus accelerate later optimization. Given this tension, a natural question is "how many non-diverse samples do I really need to take advantage of the modeling advantage without giving up the space exploration advantage?" If I have, say, a budget of ten experiments, should I spend only one low-diversity sample? Or do I need two? Half of my budget? We did not explore these practical questions in this work, due to space constraints, but we think this would be an excellent avenue for continued study.

Acknowledgements

This research was supported in part by funding from the National Science Foundation through award #1826083.

References

[1] Sio, U. N., Kotovsky, K., and Cagan, J., 2015. "Fixation or inspiration? A meta-analytic review of the role of examples on design processes". *Design Studies,* **39**, July, pp. 70–99. 00174.

[2] Crilly, N., and Cardoso, C., 2017. "Where next for research on fixation, inspiration and creativity in design?". *Design Studies,* **50**, May, pp. 1–38.

[3] Baruah, J., and Paulus, P. B., 2011. "Category assignment and relatedness in the group ideation process". *Journal of Experimental Social Psychology,* **47**(6), pp. 1070–1077. 00000.

[4] Siangliulue, P., Arnold, K. C., Gajos, K. Z., and Dow, S. P., 2015. "Toward Collaborative Ideation at Scale: Leveraging Ideas from Others to Generate More Creative and Diverse Ideas". In Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing, CSCW '15, ACM, pp. 937–945.

[5] Nijstad, B. A., Stroebe, W., and Lodewijkx, H. F. M., 2002. "Cognitive stimulation and interference in groups: Exposure effects in an idea generation task". *Journal of Experimental Social Psychology,* **38**(6), pp. 535–544.

[6] Taylor, A., and Greve, H. R., 2006. "Superman or the Fantastic Four? Knowledge Combination and Experience in Innovative Teams.". *Academy of Management Journal,* **49**(4), pp. 723–740. 00000.

[7] Zeng, L., Proctor, R. W., and Salvendy, G., 2011. "Fostering creativity in product and service development: validation in the domain of information technology.". *Hum Factors,* **53**(3), pp. 245–70.

[8] Howard-Jones, P. A., Blakemore, S.-J. . J., Samuel, E. A., Summers, I. R., and Claxton, G., 2005. "Semantic divergence and creative story generation: An fMRI investigation". *Cognitive Brain Research,* **25**(1), pp. 240–250.

[9] Chan, J., and Schunn, C. D., 2015. "The importance of iteration in creative conceptual combination". *Cognition,* **145**, Dec., pp. 104–115.

[10] Gielnik, M. M., Frese, M., Graf, J. M., and Kampschulte, A., 2011. "Creativity in the opportunity identification process and the moderating effect of diversity of information". *Journal of Business Venturing,* **27**(5), pp. 559–576. 00000.

[11] Althuizen, N., and Wierenga, B., 2014. "Supporting Creative Problem Solving with a Case-Based Reasoning System". *Journal of Management Information Systems,* **31**(1), pp. 309–340.

[12] Yuan, H., Lu, K., Jing, M., Yang, C., and Hao, N., 2022. "Examples in creative exhaustion: The role of example features and individual differences in creativity". *Personality and Individual Differences,* **189**, Apr., p. 111473. 00000.

[13] Doboli, A., Umbarkar, A., Subramanian, V., and Doboli, S., 2014. "Two experimental studies on creative concept combinations in modular design of electronic embedded systems". *Design Studies,* **35**(1), pp. 80–109.

[14] Jang, S., 2014. "The Effect of Image Stimulus on Conceptual Combination in the Design Idea Generation Process". *Archives of Design Research,* **112**(4), p. 59.

[15] Mobley, M. I., Doares, L. M., and Mumford, M. D., 1992. "Process analytic models of creative capacities: Evidence for the combination and reorganization process". *Creativity Research Journal,* **5**(2), pp. 125–155.

[16] Baughman, W. A., and Mumford, M. D., 1995. "Process-analytic models of creative capacities: Operations influencing the combination-and-reorganization process.". *Creativity Research Journal,* **8**(1), pp. 37–62.

[17] Kamath, C., 2021. Intelligent sampling for surrogate modeling, hyperparameter optimization, and data anal-

ysis, Dec.

[18] Yang, X.-S., 2014. “Swarm intelligence based algorithms: a critical analysis”. *Evol. Intel.,* **7**(1), Apr., pp. 17–28.

[19] Ma, Z., and Vandenbosch, G. A. E., 2012. “Impact of Random Number Generators on the performance of particle swarm optimization in antenna design”. In 2012 6th European Conference on Antennas and Propagation (EUCAP), pp. 925–929. ISSN: 2164-3342.

[20] Kazimipour, B., Li, X., and Qin, K., 2014. “Effects of Population Initialization on Differential Evolution for Large Scale Optimization”.

[21] Maaranen, H., Miettinen, K., and Mäkelä, M. M., 2004. “Quasi-random initial population for genetic algorithms”. *Computers & Mathematics with Applications,* **47**(12), June, pp. 1885–1895.

[22] Li, Q., Liu, S.-Y., and Yang, X.-S., 2020. “Influence of Initialization on the Performance of Metaheuristic Optimizers”. *Applied Soft Computing,* **91**, June, p. 106193. arXiv:2003.03789 [cs, math].

[23] Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T., 2002. “A fast and elitist multiobjective genetic algorithm: Nsga-ii”. *IEEE transactions on evolutionary computation,* **6**(2), pp. 182–197.

[24] Shu, L., Jiang, P., Shao, X., and Wang, Y., 2020. “A New Multi-Objective Bayesian Optimization Formulation With the Acquisition Function for Convergence and Diversity”. *Journal of Mechanical Design,* **142**(9), Mar.

[25] Simpson, T. W., Poplinski, J., Koch, P. N., and Allen, J. K., 2001. “Metamodels for computer-based engineering design: survey and recommendations”. *Engineering with computers,* **17**(2), pp. 129–150.

[26] Queipo, N. V., Haftka, R. T., Shyy, W., Goel, T., Vaidyanathan, R., and Tucker, P. K., 2005. “Surrogate-based analysis and optimization”. *Progress in aerospace sciences,* **41**(1), pp. 1–28.

[27] Jin, R., Chen, W., and Simpson, T. W., 2001. “Comparative studies of metamodelling techniques under multiple modelling criteria”. *Structural and multidisciplinary optimization,* **23**(1), pp. 1–13.

[28] Sexton, T., and Ren, M. Y., 2017. “Learning an optimization algorithm through human design iterations”. *Journal of Mechanical Design,* **139**(10).

[29] Tauber, S., Navarro, D. J., Perfors, A., and Steyvers, M., 2017. “Bayesian models of cognition revisited: Setting optimality aside and letting data drive psychological theory.”. *Psychological Review,* **124**(4), July, pp. 410–441.

[30] Kemp, C., and Tenenbaum, J. B., 2008. “The discovery of structural form”. *Proceedings of the National Academy of Sciences,* **105**(31), pp. 10687–10692.

[31] Lu, H., Yuille, A. L., Liljeholm, M., Cheng, P. W., and Holyoak, K. J., 2008. “Bayesian generic priors for causal learning”. *Psychological Review,* **115**(4), pp. 955–984. Place: US Publisher: American Psychological Association.

[32] Lu, H., Chen, D., and Holyoak, K. J., 2012. “Bayesian analogy with relational transformations.”. *Psychological Review,* **119**(3), p. 617.

[33] Fuge, M., Stroud, J., and Agogino, A., 2013. “Automatically inferring metrics for design creativity”. In International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Vol. 55928, American Society of Mechanical Engineers, p. V005T06A010.

[34] Ahmed, F., Ramachandran, S. K., Fuge, M., Hunter, S., and Miller, S., 2021. “Design variety measurement using sharma–mittal entropy”. *Journal of Mechanical Design,* **143**(6).

[35] Miller, S. R., Hunter, S. T., Starkey, E., Ramachandran, S., Ahmed, F., and Fuge, M., 2021. “How should we measure creativity in engineering design? a comparison between social science and engineering approaches”. *Journal of Mechanical Design,* **143**(3).

[36] Ahmed, F., Ramachandran, S. K., Fuge, M., Hunter, S., and Miller, S., 2019. “Interpreting idea maps: Pairwise comparisons reveal what makes ideas novel”. *Journal of Mechanical Design,* **141**(2), p. 021102.

[37] Ahmed, F., and Fuge, M., 2018. “Ranking ideas for diversity and quality”. *Journal of Mechanical Design,* **140**(1), p. 011101.

[38] Ahmed, F., and Fuge, M., 2017. “Ranking ideas for diversity and quality”. *arXiv:1709.02063 [cs]*, Sept. arXiv: 1709.02063.

[39] Ahmed, F., 2019. “Diversity and novelty: Measurement, learning and optimization”. PhD thesis.

[40] Kulesza, A., and Taskar, B., 2012. “Determinantal point processes for machine learning”. *Foundations and Trends® in Machine Learning,* **5**(2-3), pp. 123–286. arXiv: 1207.6083.

[41] Li, C., Chu, X., Chen, Y., and Xing, L., 2015. “A knowledge-based initialization technique of genetic algorithm for the travelling salesman problem”. In 2015 11th International Conference on Natural Computation (ICNC), pp. 188–193. ISSN: 2157-9563.

[42] Dong, N., Wu, C.-H., Ip, W.-H., Chen, Z.-Q., Chan, C.-Y., and Yung, K.-L., 2012. “An opposition-based chaotic GA/PSO hybrid algorithm and its application in circle detection”. *Computers & Mathematics with Applications,* **64**(6), Sept., pp. 1886–1902.

[43] Eskandar, H., Sadollah, A., Bahreininejad, A., and Hamdi, M., 2012. “Water cycle algorithm – A novel metaheuristic optimization method for solving constrained engineering optimization problems”. *Computers & Structures,* **110-111**, Nov., pp. 151–166.

[44] Mishkin, D., and Matas, J., 2016. All you need is a good init, Feb. arXiv:1511.06422 [cs].

[45] Yuan, W., Han, Y., Guan, D., and Lee, S., 2011. “Initial training data selection for active learning”. p. 5.

[46] Settles, B., 2012. “Active learning”. *Synthesis lectures on artificial intelligence and machine learning,* **6**(1), pp. 1–114.

[47] Yoon, J., Arik, S., and Pfister, T., 2020. “Data Valuation using Reinforcement Learning”. In Proceedings of the 37th International Conference on Machine Learn-

ing, PMLR, pp. 10842–10851. ISSN: 2640-3498.

[48] Eysenbach, B., Gupta, A., Ibarz, J., and Levine, S., 2018. "Diversity is all you need: Learning skills without a reward function". *arXiv preprint arXiv:1802.06070*.

[49] Schölkopf, B., and Smola, A. J., 2018. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. June.

[50] Kulesza, A., and Taskar, B. "k-DPPs: Fixed-Size Determinantal Point Processes". p. 8.

[51] Calandriello, D., Dereziński, M., and Valko, M., 2020. "Sampling from a k-DPP without looking at all items".

[52] Li, C., Jegelka, S., and Sra, S., 2016. "Efficient Sampling for k-Determinantal Point Processes". *arXiv:1509.01618 [cs]*, May. arXiv: 1509.01618.

[53] Mariet, Z. E., 2016. "Learning and enforcing diversity with Determinantal Point Processes". Master's thesis, MASSACHUSETTS INSTITUTE OF TECHNOLOGY.

[54] Hansen, N., Finck, S., Ros, R., and Auger, A. "Real-Parameter Black-Box Optimization Benchmarking 2010: Noisy Functions Definitions". p. 12.

[55] Rönkkönen, J., Li, X., Kyrki, V., and Lampinen, J., 2011. "A framework for generating tunable test functions for multimodal optimization". *Soft Comput.,* **15**, Sept., pp. 1689–1706.

[56] Mo, H., Li, Z., and Zhu, C., 2017. "Epistasis-tunable test functions with known maximum constructed with sinusoidal bases". In 2017 12th International Conference on Intelligent Systems and Knowledge Engineering (ISKE), pp. 1–6.

[57] Mason, W., and Watts, D. J., 2012. "Collaborative learning in networks". *Proceedings of the National Academy of Sciences,* **109**(3), Jan., pp. 764–769.

[58] 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". In Advances in Neural Information Processing Systems, Vol. 33, Curran Associates, Inc., pp. 21524–21538.

Supplemental Material for How Diverse Initial Samples Help and Hurt Bayesian Optimizers

Eesh Kamrah
Dept. of Mechanical Engineering
University of Maryland
College Park, Maryland 20742
Email: kamrah@umd.edu

Seyede Fatemeh Ghoreishi
Dept. of Civil and Environmental Engineering
& Khoury College of Computer Sciences
Northeastern University
Boston, Massachusetts 02115
Email: f.ghoreishi@northeastern.edu

Zijian "Jason" Ding
College of Information Studies
University of Maryland
College Park, Maryland 20742
Email: ding@umd.edu

Joel Chan
College of Information Studies
University of Maryland
College Park, Maryland 20742
Email: joelchan@umd.edu

Mark Fuge[*]
Dept. of Mechanical Engineering
University of Maryland
College Park, Maryland 20742
Email: fuge@umd.edu

1 Fast sampling DPP Method

Our idea seeks to reduce the complexity of the sampling method and the construction time for DPP as well as investigate a Diverse sampling method that can generate both low-diversity and high-diversity samples. To do this we build on the work from [1] to rank and compare the diversity of the two sets. To define our diversity measure, let's assume $X \subset \mathbb{R}^{\mathcal{F}}$, where $|\mathcal{F}|$ is the number of features of X. Then we can define a set as $S_Y^k \subset X$ of size k. This means $S_{Y_i}^k \in \mathbb{R}^{\mathcal{F}} \times \mathbb{R}^k$, then using a similarity measure (RBF kernel) W on this set, we can define the DPP score for a set $S_{Y_i}^k$ as follows:

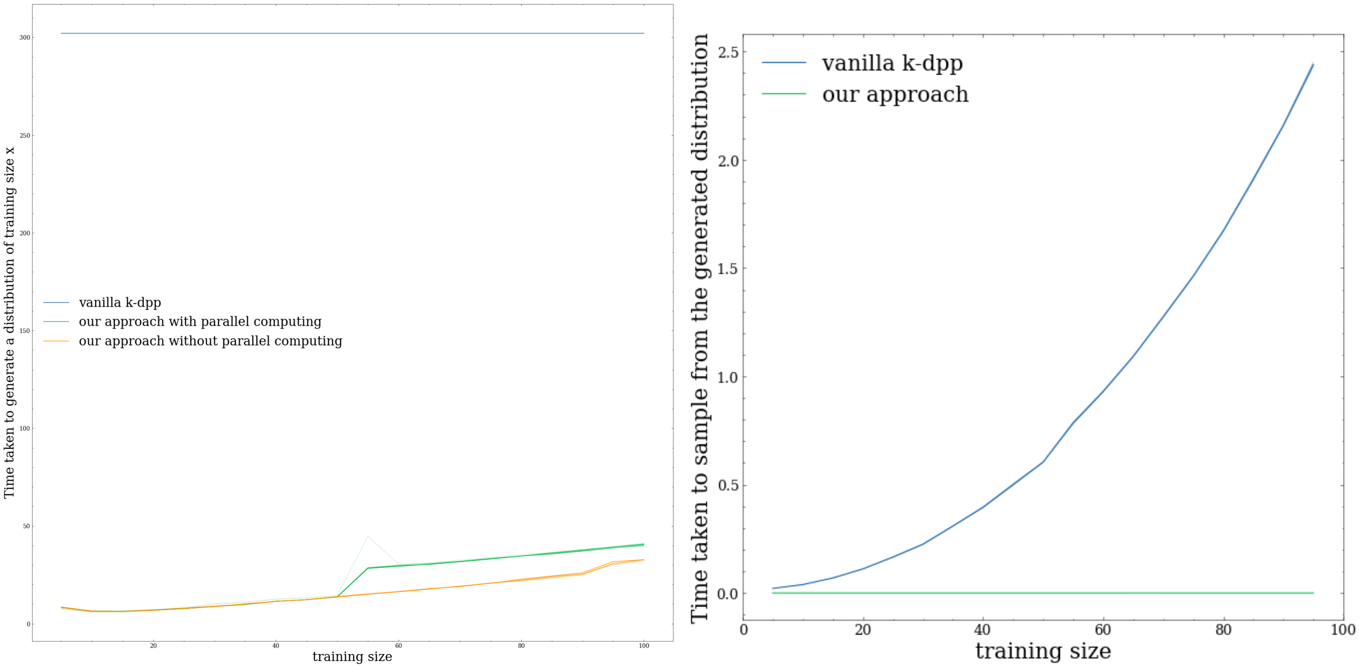
$$f(W_{Y_i}) = \frac{\log(det(K(W_{Y_i}))) - \left(\sum_i^{|S^k|} log(det(K(W_{Y_i})))\right)}{\sqrt{\sum_i^{|S^k|} (\log(det(K(W_{Y_i}))) - \left(\sum_i^{|S^k|} log(det(K(W_{Y_i})))\right))^2}} \\ \text{, where } |S^k| = (\binom{\prod_i^{dim(X)} dim(\mathcal{F}_i)}{k}) \tag{1}$$

As we can see in Eq. 1, the number of sets or cardinality of the distribution $|S^k|$ needed to be sampled grows combinatorially with the changes in the size of the sample space for Xs, and the size of the set k. For example for a $X \in \mathbb{Z}^2$ where each feature $\mathbb{Z}_i \in [0, 100]$. Then, the number of possible sets of size k is given by $\binom{100\times100}{k}$, thus normalizing the distribution

[*] Address all correspondence to this author.

using a mean and standard distribution is an expensive task. We can re-write Eq. 1 in words as follows:

$$\text{DPP Score}(S_{Y_i}^k) = \frac{DPPScore(K(W_{Y_i}) - \text{mean score}}{\text{s.d. of DPP scores for the k-HDPP}}$$



(a) Compares the construction time for regular k-DPP distribution with different size of samples

(b) Compares the sampling time time for regular k-DPP vs our approach as the size of k is increased from 5 to 100.

Fig. 1: Compares the relative performance/speed-up of our method over the traditional k-dpp methods. The figure contains two plots showing the tradeoff between the two methods. In the traditional method constructing the DPP distribution is costly but generating a distribution is only dependent on the number of points in X, and independent of training size (k). While, sampling from a k-DPP has a polynomial complexity on the training size (k), while both these facts are inverted for our approach.

1.1 Sampling method

The sampling method for our DPP approach is straightforward. Based on the constructed DPP, our approach samples randomly from either above a certain percentile or below a certain percentile. As shown in Fig. 1(b), our approach's sampling time is faster than that of a regular k-DPP, where the cost of sampling increases as a function of training size (k). Conversely, generating the distribution for our approach is dependent on 'k', while the same distribution can be used for different k(s) with a traditional k-DPP approach. Our approach's biggest benefit is the ability to draw samples of different diversity. Using our approach this is as simple as sampling from different percentiles of the distribution.

Algorithm 1 Constructing the DPP sub distribution

```
1: for $i \in range(M)$ do
2:     Sample $S_{Y_i}^k \sim \mathbb{IID}(S^k)$
3:     Calculate $g(S_{Y_i}^k) = g_{y_i}$ and append this to $Scores_{S^k}$
4: end for
5: Return DPP Score= $\frac{Score_{S^k} - mean(Score_{S^k})}{s.d.(score_{S^k})}$.
```

The uniqueness of our approach lies in an easy trick to upper bound the error on the generated DPP scores, and thus our approach can provide certain guarantees on whether the sampled S_Y^k is in fact from the percentile that the method claims it is from.

1.2 Upper bound on errors

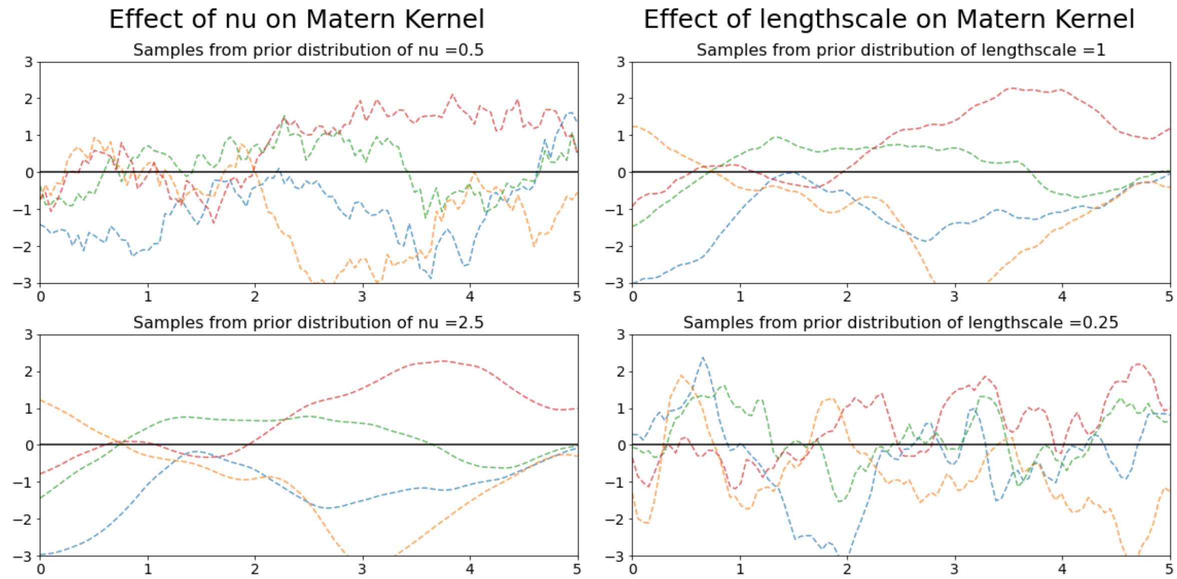
The guarantee is based on method's independence of choosing the S_Y^K from a combinatorially large set. For IID sampling each set, $S_{Y_i}^K$, needs to be sampled independent of the other and the sampling should be done with replacement. But since the distribution of S^k needs to mirror that of a k-DPP, all the sets in the space are sampled over X without replacement and are unordered because DPP scores for two S^k with the same points (Y) will always correspond to the same score. Thus, sampling IID on S^k means identically sampling unordered sets of X without replacement.

If we sample the sets $S_{Y_i}^k$ such that they are Independent Identical Distributed (I.I.D.) sets, then we can upper bound the Expected Value of population mean through the use of Hoeffding's inequality: Eq. 2 as discussed in [2]. The inequality states that if a distribution is sampled using i.i.d random variables, we can then put a bound on the Error for estimating Expected Values of the population mean ($|\mathbb{M}_n = \frac{1}{n}\sum_i^n [M_i]|$), where $\mathbb{M}_n$ is the mean of the sample of size n.

$$\mathbb{P}\{|\mathbb{M}_n - \mathbb{E}(S^k)| \leq \varepsilon\} \geq 1 - 2 \cdot exp\left\{\frac{-2 \cdot n^2 \varepsilon^2}{\sum_{i=1}^n (b_i - a_i)^2}\right\} \tag{2}$$

Using Eq. 2 we can guarantee the probability of this error to be some $1-\delta$, where the δ term is given by the exponential. This allows us to limit the cardinality of the $|S^k|$ to M given we choose an ε. Based on this guarantee a schematic explanation for the construction of our sub-distribution using the approach detailed till now is then documented well in Algorithm 1. This approach is extensively discussed and proved in an upcoming paper.

A clear shortcoming of this approach is the need to generate the distribution whenever the k is changed. But, because of the faster construction speed for our approach, this cost outweighs using a k-DPP. Another, shortcoming our approach faces is the limited number of samples that can be drawn from the distribution, which requires us to construct a new distribution if more than M samples need to be drawn.



(a) Effect of changing the ν hyper-parameter on the Gaussian Process.

(b) Effect of changing the lengthscale hyper-parameter of the Matern Kernel. The figure has 2 similar GPs with shorter (bottom) and longer (top) lengthscales.

Fig. 2: Effect of ν and lengthscale on Gaussian Process.

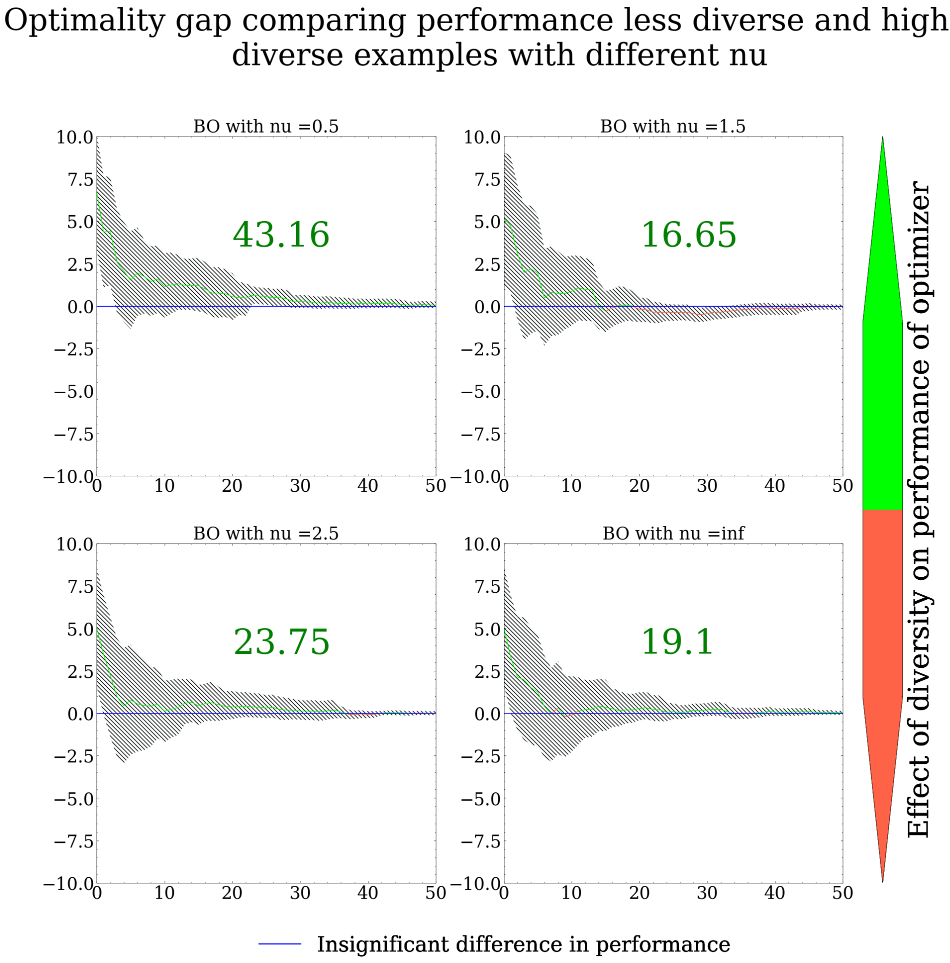


Fig. 3: Grid plot showing how changing ν affects the relative performance of diverse and non-diverse initialization on Bayesian optimizers. To understand the plot better quantitatively, each subplot also has the Net Cumulative Optimality Gap (NCOG) for each value of ν. No trends are seen when relative performance of the diverse and non-diverse samples.

2 Effects of additional hyper-parameters on performance of the optimizer

As described in the Methods section in the main paper Bayesian Optimizer that uses a Matern kernel has several hyper-parameters. This section will serve to further explore the effects that each parameter has on the Gaussian Process (GP). The main paper provides a brief introduction to each hyper-parameter apart from ν. So, let's begin this section with a brief look into the hyper-parameter ν.

ν of the Matérn Kernel The kernel used with the Gaussian Process is the Matern kernel which essentially is a scaled RBF kernel controlled by the parameter ν [3] as shown in Eq. 3.

$$k_{\text{Matérn}}(\mathbf{x}_1, \mathbf{x}_2) = \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\sqrt{2\nu} \frac{||\mathbf{x}_1 - \mathbf{x}_2||}{\theta} \right)^{\nu} H_{\nu} \left(\sqrt{2\nu} \frac{||\mathbf{x}_1 - \mathbf{x}_2||}{\theta} \right), \tag{3}$$

The hyper-parameter (ν) dictates how smooth or differentiable the function is. Changes in this parameter then influence the expectation of the Gaussian Process in terms of its acquisition function. A more differentiable function or a higher ν means that the acquisition function samples assuming a smoother Gaussian Process function. It can be seen in Fig. 2(a) how changes in ν changes the prior of the GP.

While, ν controls the prior μ and 'lengthscale' control how the data is scaled and thus indirectly control the expectations of the GP. The effects of lengthscale on GP can be seen in Fig. 2(c). The effects are similar to that of the parameter ν. Thus, we can conclude that 'lengthscale' can be used to control the expectations of the GP. Since, ν is not a parameter that is learned during the optimization process it does not have significant effect on "*Model Building advantage*". This can be seen in Fig. 3, even as ν is changed there is no significant change in the performance of the optimizer, and thus we can conclude that ν is an insignificant factor in studying “*Model Building advantage*”.

To provide some empirical evidence to the importance of ‘lengthscale’ as a hyper-parameter. Let us look at results from additional plots that were generated while working on experiment 2.

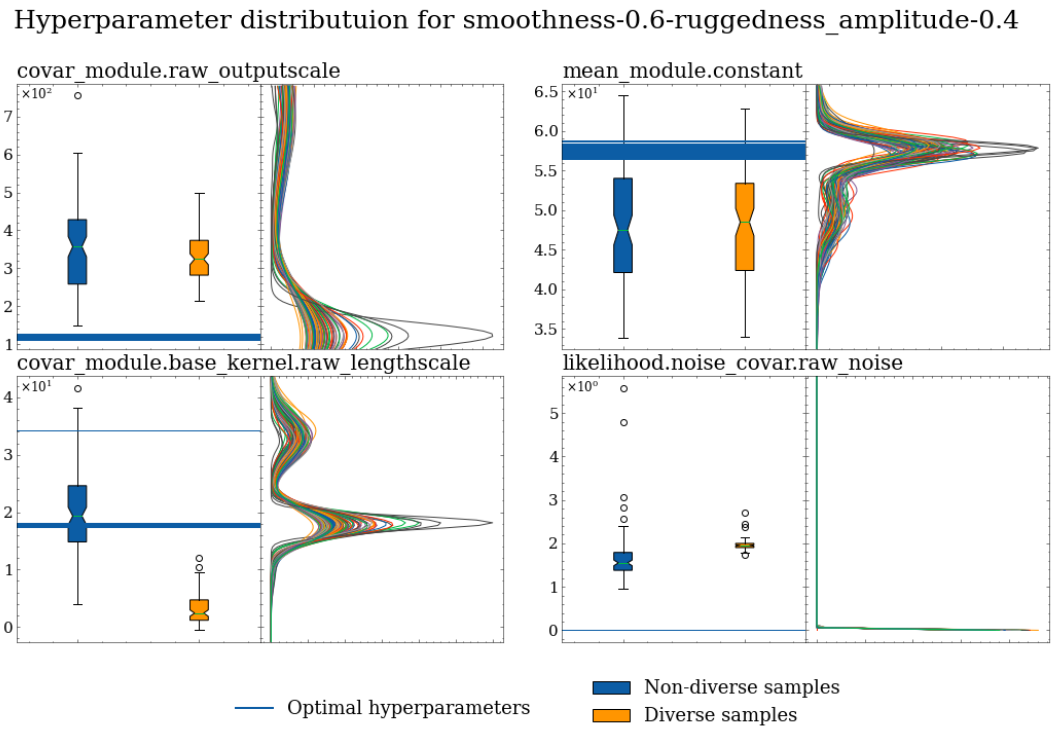


Fig. 4: Box plots showing the distribution of different hyper-parameters of the Gaussian Process as learned by Bayesian optimizer when fitted with just the initial examples as training data. The shown hyper-parameters are specific to Wildcatwells configuration with smoothness = 0.2 and ruggedness amplitude =0.2. The data is collected over 100 seeds. The horizontal lines across the boxplot indicate the optimal hyper-parameters learned over 100 different seeds.

3 Further plots for Experiment 2

While studying “*Model Building advantage*” for Gaussian Processes, we looked at not only at ‘lengthscale’ but all hyper-parameters as it can be seen in Fig. 4. The box-plot for each hyper-parameter is constructed in the same way as the steps detailed in Methods section of Experiment 2 in the main paper. To the right of each box-plot in Fig. 4 is also 100 kernel density functions that have been used to estimate the ‘optimal hyper-parameter’ for a particular instance of that family (smoothness =0.6, ruggedness amplitude =0.4) of wildcat wells function.

Now, as it can be seen in Fig. 4 the optimal noise hyper-parameter is close to 0 for all the instances in the family. While, the one’s estimated using a sample size (k) of 10, in the box-plot, are not. The performance for both diverse and non-diverse is relatively similar for this hyperparameter. This can be seen as the case for both the ‘Mean function’ (μ) and the ‘Outputscale’ as well. While, ‘lengthscale’ is the only hyper-parameter that has varying performance across diverse and non-diverse samples.

An important factor while quantifying the “*Model Building advantage*” is learning the ‘optimal hyper-parameter’ for an instance of wildcat wells function, which is described in the next section.

4 Finding optimal hyper-parameters for a given objective function

To compute the ‘optimal hyper-parameter’ we first use a Binary search method to discern a robust range (of 200 points) over which all families of wildcat wells functions has a noise parameter value of $< 10^{-5}$. This essentially means that Bayesian optimizer has found an optimal set of hyper-parameters for the Gaussian Process that accurately imitates the given black-box function.

This robust range for all the families of wildcat wells function used in the experiment was determined as 1000-1200 points.

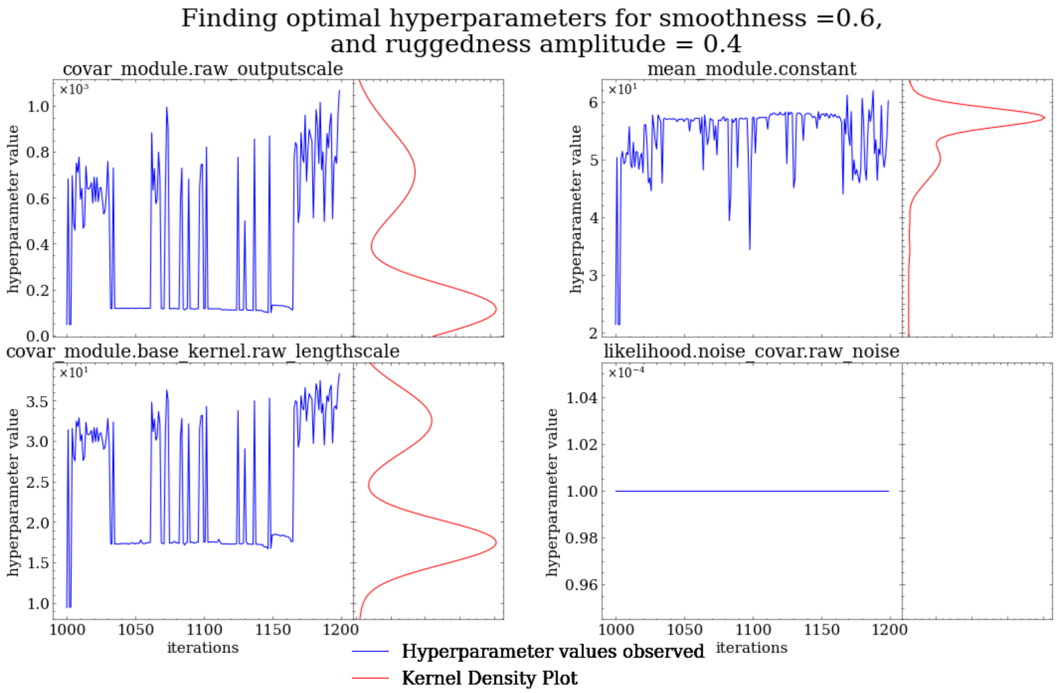


Fig. 5: Figure depicting the first step in determining optimal hyperparameters for wildcat wells function with smoothness 0.6 and ruggedness amplitude 0.4 and seed 88. Each hyperparameter in the grid plot has subsequent two adjacent plots. The observed hyperparameter values over when BOTorch is used to maximize the Marginal Log Likelihood given 1000 to 1200 random points (left), the kernel density function derived from this data (right).

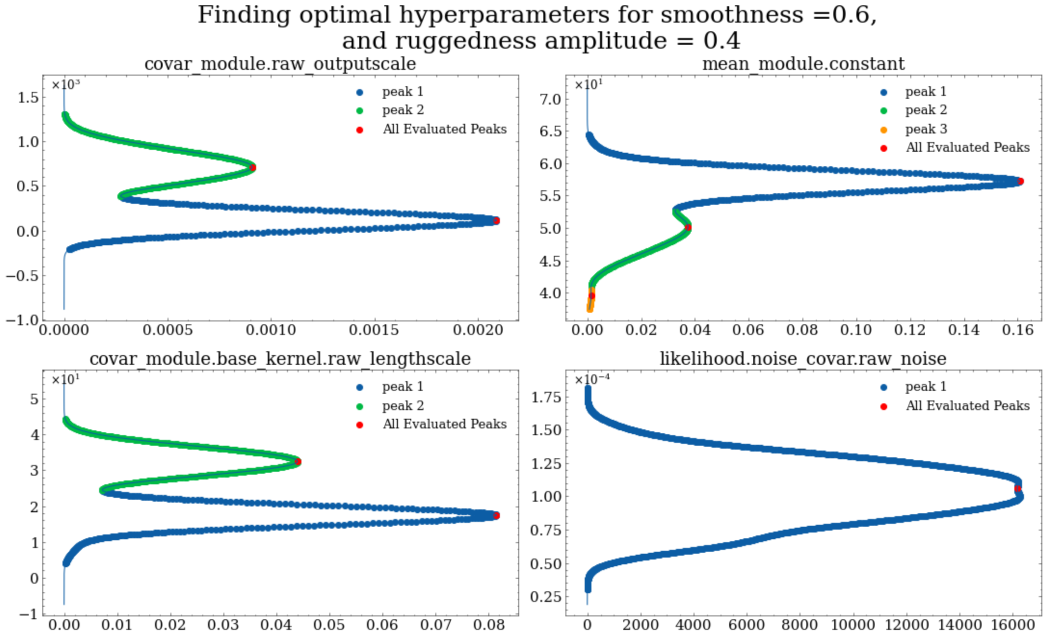


Fig. 6: Figure depicting the second step in determining optimal hyperparameters. Figure shows the peaks evaluated as potential optimal hyperparameter, and the shaded points that are used to calculate the area under the corresponding peak.

Once, this range is determined the data is collected over the 200 points by maximizing the Marginal Log Likelihood for the a Single Task GP model using BOTorch's '*fit-gpytorch-model*' [4] method. The resulting data is the hyperparameters that BoTorch learns using the given data points. This data is then used to build a kernel density function as indicated by the red line-plot (right side of every subplot) next to the data observed over the 200 points in Fig. 5. Then using '*scipy.signal.find-peaks*' [5], peaks are found in the density function labeled by red dots in Fig. 6. Sometimes more than one peak is observed this is because there are multiple modes of hyperparameters that provide a stable solution for the problem. For the purpose of this paper we only focus on extracting the most observed mode as our optimal hyperparameter.

To find the most observed mode, we use the width of the peaks in the kernel density function. The width of the peak is estimated by calculating a numerical gradient on the density function as seen in Fig. 6. The width of the whole peak can be seen highlighted/labeled in each subplot for each peak using a different color. The peak with the largest area is selected as the optimal hyper-parameter for the particular instance of wildcat wells function.

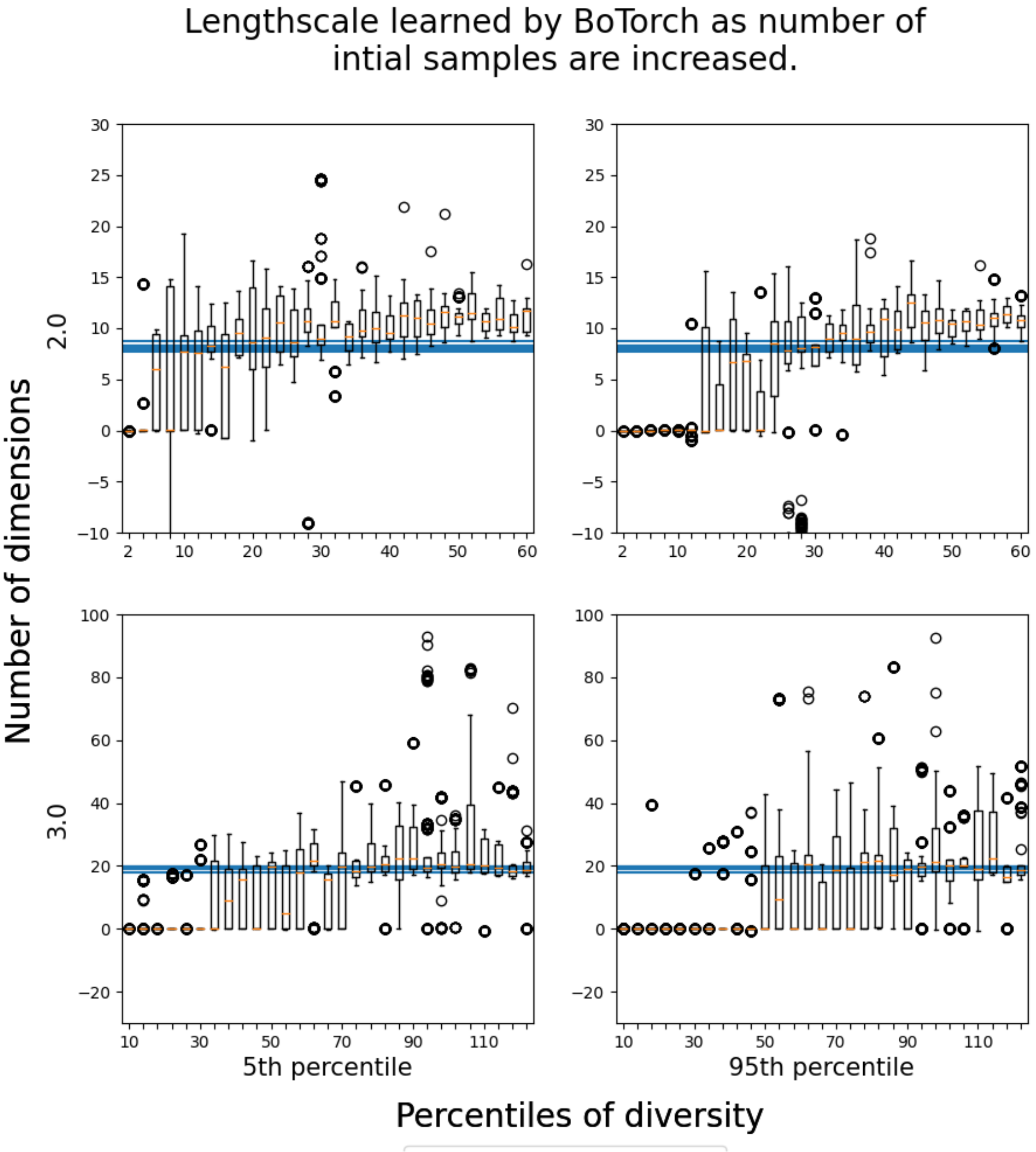


Fig. 7: Box plot showing the lengthscale parameter as learned by wildcatwells with ‘high’ level of ruggedness in 2D and 3D as the training samples are increased. The plot also confirms the existence of a “modeling advantage” for training samples of a particular size. The results are further discussed in **S**5

5 Effect of increasing training size on hyperparameter learning

While trying to replicate the results for the 3D case we observed that the ‘modeling advantage’ we observed for less diverse examples was also influenced by the number of examples in the initial set. This was because if we initialized the 3D case with the same number of initial samples as the 2D case, the optimizer in the 3D case would not be able to accurately estimate the appropriate hyperparameters regardless of the sampling method and would just set the hyperparameters to zero. This is perhaps obvious if we think about how space coverage degrades for a fixed number of samples as we increase the dimensionality of a design space. What we observed, and show below in Fig. 7, is that there are essentially three “initial sample size regimes” that determine whether or not non-diverse sampling can use its ‘modeling advantage’, although this advantage exists in both the 2D and 3D case:

1. Sample-deficient: This is when we provide each optimizer with too few initial examples, such that irrespective of that set's diversity the BO will not be able to meaningfully learn hyperparameters and will instead set them to zero. For example, in Fig. 7 bottom, with fewer than 26 initial samples, both the 5th and 95th percentile samples cannot provide good estimates of the kernel hyper-parameters
2. The 'modeling advantage' region: With this number of samples, the 5th percentile is able to reasonably estimate the hyperparameter values but the 95th struggles to do so. For example, in Fig. 7 top (2D), we can observe this at 10 samples, which, by coincidence, was the original setting for our 2D example in our initial manuscript. We see that in Fig. 7 bottom (3D) this transitions somewhere between 35 to 75 initial samples. In this region, 5th percentile sampling can exercise its modeling advantage while the 95th percentile still does not have enough initial samples to consistently and accurately estimate the kernel hyper-parameters.
3. Sample-saturated: In this region, the shear number of initial points we provide BO is sufficiently high such that it can estimate the kernel hyper-parameters well, regardless of whether the initial points are diverse or not. For example, in Fig. 7 top, this occurs after around 40 initial samples. In Fig. 7 bottom this occurs after around 100 initial samples. In this 'sample-saturated' case, the modeling advantage of non-diverse sampling disappears, often because this is a sufficient number of points that the optima become easy to find at that point (see Fig. 8 where the BO often converges at those same number of samples).

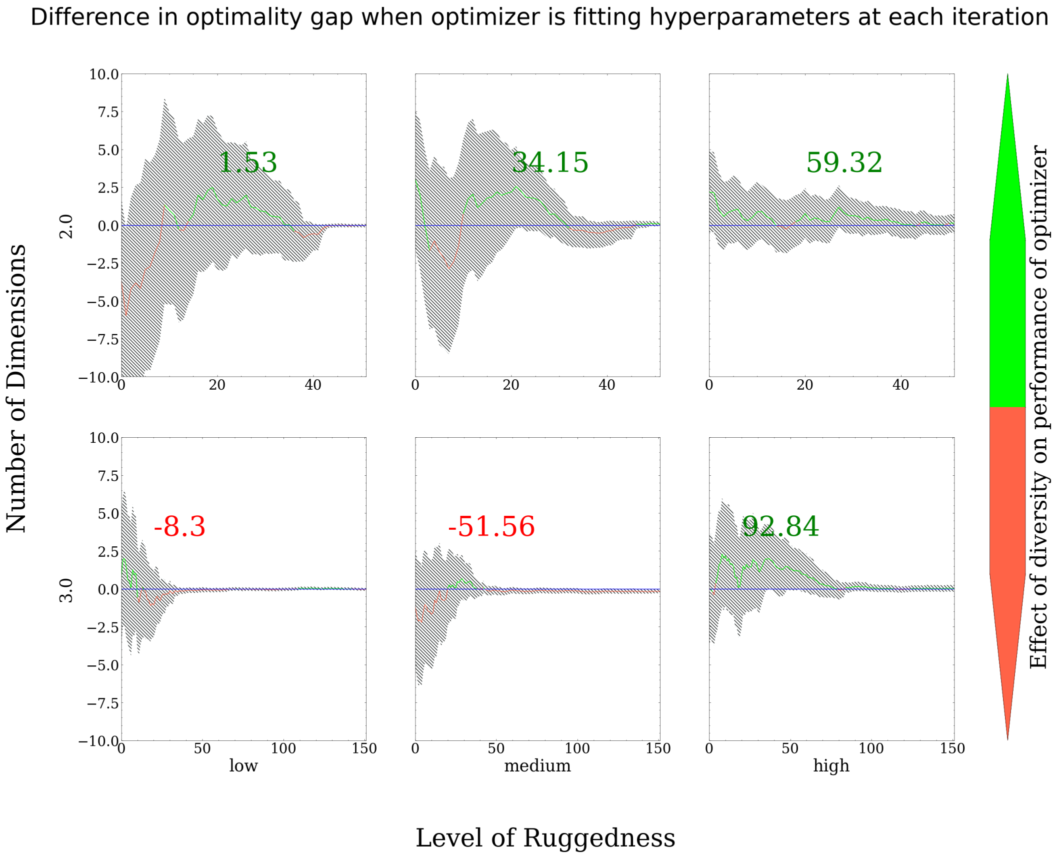


Fig. 8: Optimality gap grid plot showing the difference in current Optimality Gap between optimizers initialized with 5^{th} vs 95^{th} percentile diverse sample (y-axis) as a function of optimization iteration (x-axis). The different factors in the factor grid plot are the dimensions across the rows and the ruggedness level across the columns. Each plot also has text indicating the Net Cumulative Optimality Gap (NCOG), a positive value corresponds to a better performance by high diversity samples compared to the low diversity samples. The plot shows that BO benefits from diversity in some cases but not others. There is no obvious trends in how the NCOG values change in the grid. The results are further discussed in **S**6

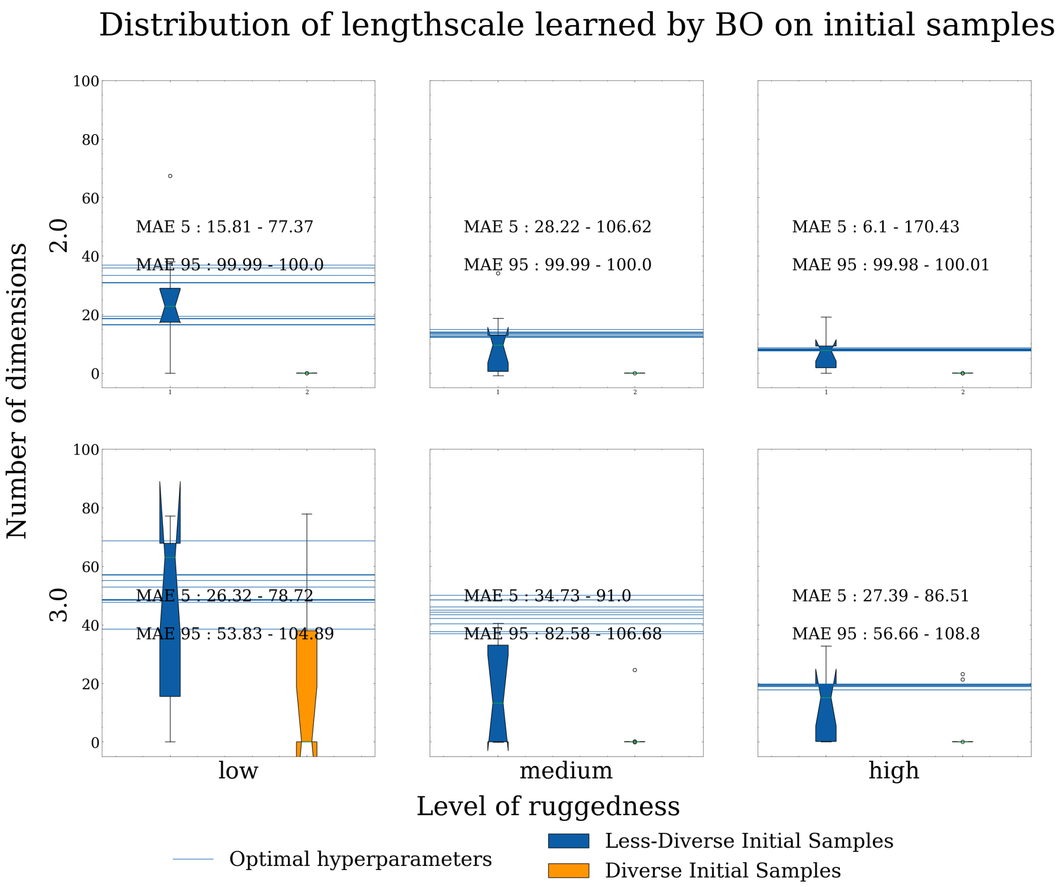


Fig. 9: Box plot showing distribution of 'Lengthscale' hyper-parameter learned by BO when initiated with diverse (orange) and less-diverse samples (blue) for 3 different families of wildcat wells functions of the same parameters but 100 different seeds in each dimension. The optimal hyper-parameter for each of the 100 wildcat wells instances from each family is also plotted as horizontal (blue) lines—in many but not all cases these overlap. Each cell in the plot also has the 95^{th} percentile confidence bound on Mean Absolute Error (MAE) for both diverse and non-diverse samples. The results show that MAE confidence bounds for non-diverse samples are smaller compared to diverse samples for all the families of wildcat wells function. Thus, indicating a presence of Model Building advantage for non-diverse initial samples. The results of this figure are further discussed in **S**6

6 Effect of increasing problem dimensions on the results

To confirm what we observed was not limited to 2 dimensions we decided to run Experiment 1, 2, and 3 with wildcatwells in 3 dimensions. To make the results comparable in a single figure for both 2D and 3D case it was necessary to limit the variability of ruggedness from a 4x4 grid to 3 levels of 'ruggedness'. These 'levels of ruggedness' are 'low', 'medium' and 'high', which correspond to (smoothness : 0.8, ruggedness amplitude : 0.2), (smoothness : 0.4, ruggedness amplitude : 0.4) and (smoothness : 0.2, ruggedness amplitude : 0.8) respectively.

Further, to see the 'model building' advantage for the 3D case we changed the experiment set-up slightly by initializing all the plots generated in 3D with 40 examples instead of the 10 used to initialize BO in 2D. The intuition behind this is further explained in **S**5. Figure 8 shows the results of Experiment A1, which is a modification of Experiment 1 from the main paper, where we compare 2D and 3D behavior. The results in 3D mirror our observations in 2D.

As with Experiment 2 in the main paper, Fig. 9 shows Experiment A2 that compares with a third dimension. Here, we can see that much like in 2 dimensions, in 3 dimensions the 5^{th} percentile performs better than 95^{th} in estimating the lengthscale, hence confirming the 'modeling advantage'.

Lastly, we can use Fig. 10 to see that when the modeling advantage is taken away the 95^{th} percentile performs better compared to the 5^{th} percentile. These results mirror our original results in 2D.

7 Do these results hold on alternative test functions?

A natural question is whether our results are limited to just our choice of the wildcat-wells class of function generators, or do they transfer across different functions? To test this, we repeated the experiments described in **S**6 for three different but commonly used N-Dimensional optimization test functions: the Sphere, Rosenbrock and Rastrigin functions as seeen in

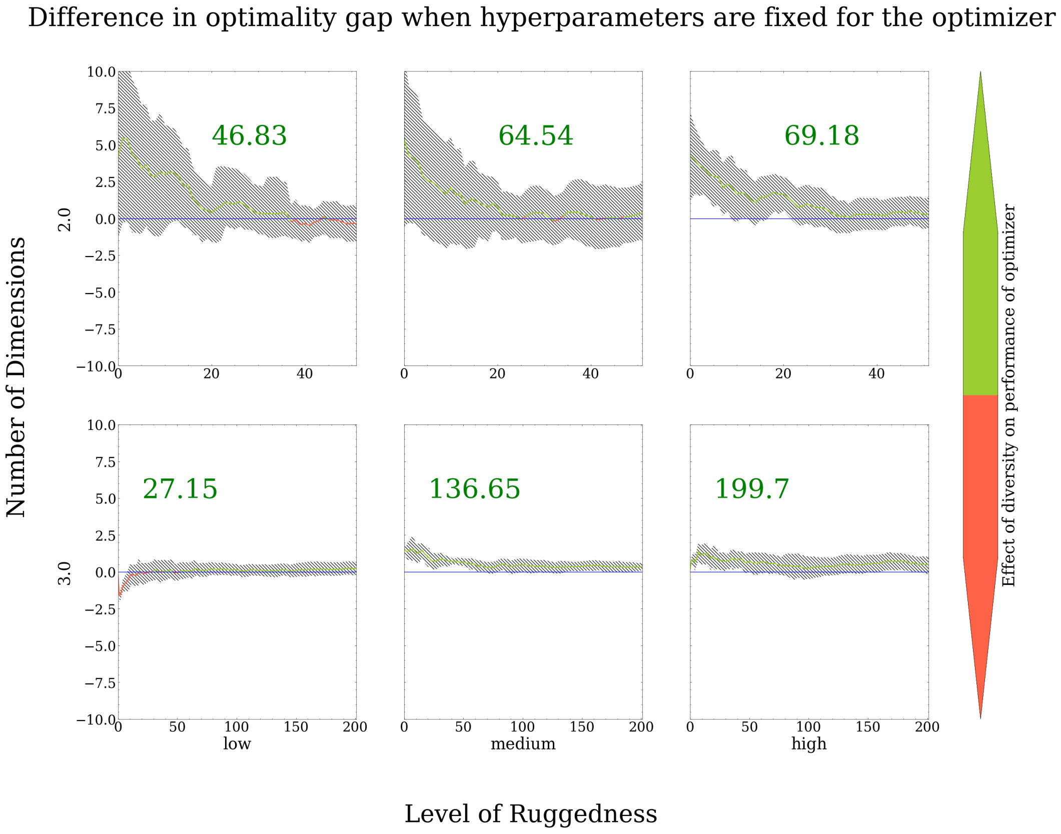


Fig. 10: Optimality gap plot showing effects of diversity when the optimizer is not allowed to fit the hyper-parameters for the Gaussian Process and the hyper-parameters are instead fixed to the values found in Experiment A2. The results from this plot show positive NCOG values for all families of wildcat wells function even as dimensions increase, showing that once the ‘Model Building advantage’ is taken away the diverse samples outperform non-diverse samples. Further discussion on this plot can be read in **S**6

Eq. 4. The only major difference with the previous experiments is that instead of plotting the optimality gap directly, we instead plot the Percentage difference in the optimality gap in Fig. 13. This was done to bring the plot to a comparable scale since the absolute difference in raw optimality gap can be, at certain points, on the order of millions, and at some points less than 1.

$$
\begin{aligned}
\text{Sphere}(X) &= \sum_{i=1}^{\text{dims}} x_i^2 \\
\text{Rastrigin}(X) &= 10 \times \text{dims} + \sum_{i=1}^{\text{dims}} \left[x_i^2 - 10\cos(2\pi x_i)\right] \\
\text{Rosenbrock}(X) &= \sum_{i=1}^{\text{dims-1}} \left[100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2\right]
\end{aligned} \tag{4}
$$

As seen in Fig. 11, when the hyperparameters are allowed to be optimized, in general low-diversity samples led to faster convergence than high-diversity initial samples. This is not always that case, as the 4D and 5D Rastrigin functions cases shows—in such cases non-diverse samples have comparatively marginal improvement in the longer term. For reference, this plot is designed to be a replication of Experiment 1 in the main paper, but just for different test functions.

Fig. 12 shows that 5^{th}-percentile diversity (low diversity) initial samples learns the kernel hyperparameter more accurately using fewer samples compared to 95^{th}-percentile diversity initial samples in two dimensions and that this holds true irrespective of the choice of test function. However, as the function dimension increases this effect diminishes since the number of initial samples needed to activate this “modeling advantage” regime increases (See earlier Fig. 7). With this additional set of data, samples from from the 95^{th}-percentile of diversity learn the hyperparameters as well as 5^{th}-percentile samples. For reference, like with Fig. 9 above, this plot was designed to be a replication of Experiment 2 in the main paper, but just

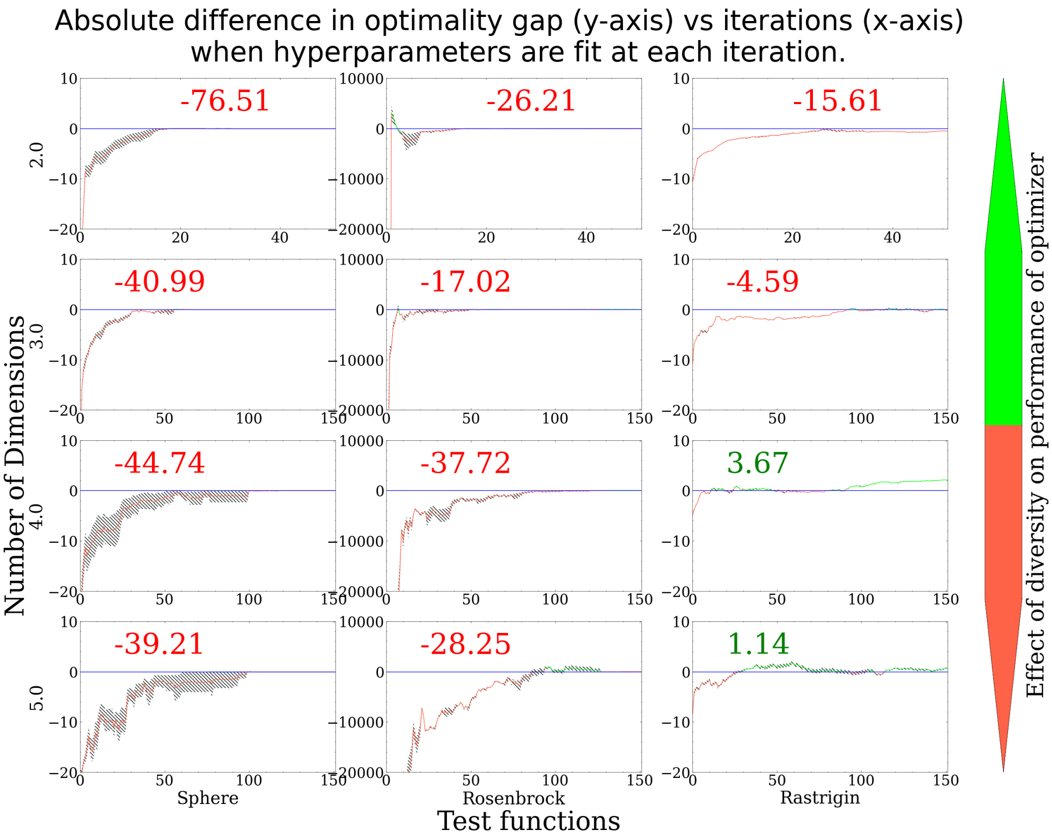


Fig. 11: Optimality gap grid plot showing the absolute difference in current Optimality Gap between optimizers initialized with 5^{th} vs 95^{th} percentile diverse sample (y-axis) as a function of optimization iteration (x-axis). The different factors in the factor grid plot are the dimensions across the rows and the different test functions across the columns. Each plot also has text indicating the Percentage Cumulative Optimality Gap (PCOG), a positive value corresponds to a better performance by high diversity samples compared to the low diversity samples. The plot shows that BO benefits from diversity in some cases but not others. There are no obvious trends in how the PCOG values change in the grid. The results are further discussed in **S**7

for different test functions and across increased dimensions. Unlike in Fig. 9 here we see that our proposed causal explanation for the "modeling advantage" is less clear since for certain functions the high-diversity samples have better posterior convergence than the 5^{th}-percentile samples, and vice versa depending on the specific function and dimension.

In Fig. 13 where the kernel hyper-parameters are fixed to what should be optimal values, (compared to Fig. 11 where the kernel hyper-parameters are learned) we can see several effects. First, we see that the low diversity initial samples had, on average, better initial starting points on these test functions as seen by the PCOG values on the x-axis at "0". This could largely be luck or a peculiarity with the three test functions, since common optimization test functions often have their optimal points toward the center of the domain, which non-diverse starting points are likely to sample with higher frequency compared to diverse starting points. (Note in our wildcat wells function this was not the case and the optimal point was likely to occur at any point in the domain depending on the seed of the random function generator.) Second, we see compared to Fig. 11 that high diversity initial samples appear to be able to benefit from the 'Space Exploration' advantage we hypothesized in the main paper and do catch-up almost instantaneously compared to the lower-diversity samples. For reference, this plot is designed to be a replication of Experiment 3 in the main paper, but just for different test functions. We still see a similar effect, in the sense that fixing the BO hyper-parameters aids the diverse initial sample condition, on average, which mirrors qualitatively the phenomenon we observed on the wildcat wells function (compare this supplemental material document's Fig. 11 with Fig. 13).

In Figs. 14, 15, and 16 we can see how increasing the number of initial training samples induces convergence on the learned kernel hyper-parameters for the Rastrigin, Rosenbrock, and Sphere functions, respectively. We used these plots to choose the number of training samples to be used in Figs. 11, 12, and 13 by selecting the number of samples within the "model building advantage" regime (as opposed to the sample deficient or sample saturated regime). The specific number of training samples used for each function at each dimension can be seen in Table 1. We can see that the performance of high diversity samples is significantly better when compared to the performance in Fig. 11. The high diversity samples still struggle to improve performance for 'Rosenbrock' function, our hypothesis is that because the number of samples needed to learn the hyperparameters is exceedingly large for the Rosenbrock function (see Fig. 15) our proposed "modeling-advantage" is not that helpful to the optimizer, since it has already found a reasonable optimum by the time it has collected sufficient samples to converge to reasonable kernel estimates.

Dimension	Sphere	Rosenbrock	Rastrigin
2	8	4	5
3	12	5	7
4	38	8	30
5	75	20	60

Table 1: Table showing the different training size/number of examples used to initialize BO for different test functions in Figs. 11,12,13.

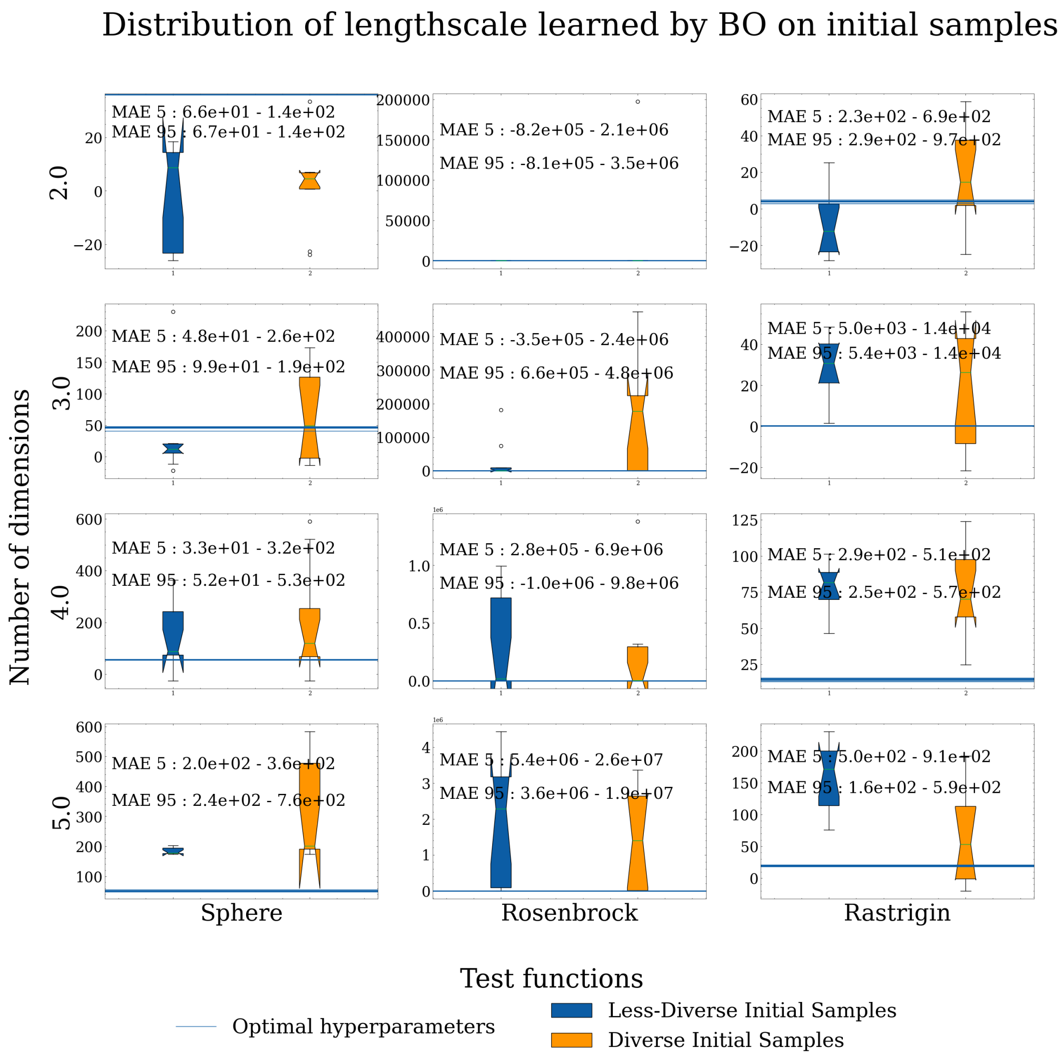


Fig. 12: Box plot showing distribution of ‘Lengthscale’ hyper-parameter learned by BO when initiated with diverse (orange) and less-diverse samples (blue) for Sphere, Rosenbrock and Rastrigin test functions over 200 different seeds in each dimension. For refernce to how many training samples were used pleae check Table. 1. The optimal hyper-parameter for each test function over 10 different runs is also plotted as horizontal (blue) lines—in many but not all cases these overlap. Each cell in the plot also has the 95^{th} percentile confidence bound on Mean Absolute Error (MAE) for both diverse and non-diverse samples. The results show that MAE confidence bounds for non-diverse samples are smaller compared to diverse samples for most test functions but at least does as well as the 95^{th}. Thus, indicating a presence of Model Building advantage for non-diverse initial samples. The results of this figure are further discussed in **S**7

References

[1] Ahmed, F., and Fuge, M., 2017. “Ranking ideas for diversity and quality”. *arXiv:1709.02063 [cs]*, Sept. arXiv: 1709.02063.

[2] Serfling, R. J., 1974. “Probability Inequalities for the Sum in Sampling without Replacement”. *The Annals of Statistics,* **2**(1), Jan., pp. 39–48. Publisher: Institute of Mathematical Statistics.

[3] Garnett, R. Bayesian Optimization Book.

[4] 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”. In Advances in Neural Information Processing Systems, Vol. 33, Curran Associates, Inc., pp. 21524–21538.

[5] Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., Carey, C. J., Polat, , Feng, Y., Moore, E. W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E. A., Harris, C. R., Archibald, A. M., Ribeiro, A. H., Pedregosa, F., van Mulbregt, P., SciPy 1.0 Contributors, Vijaykumar, A., Bardelli, A. P., Rothberg, A., Hilboll, A., Kloeckner, A., Scopatz, A., Lee, A., Rokem, A., Woods, C. N., Fulton, C., Masson, C., Häggström, C., Fitzgerald, C., Nicholson, D. A., Hagen, D. R., Pasechnik,

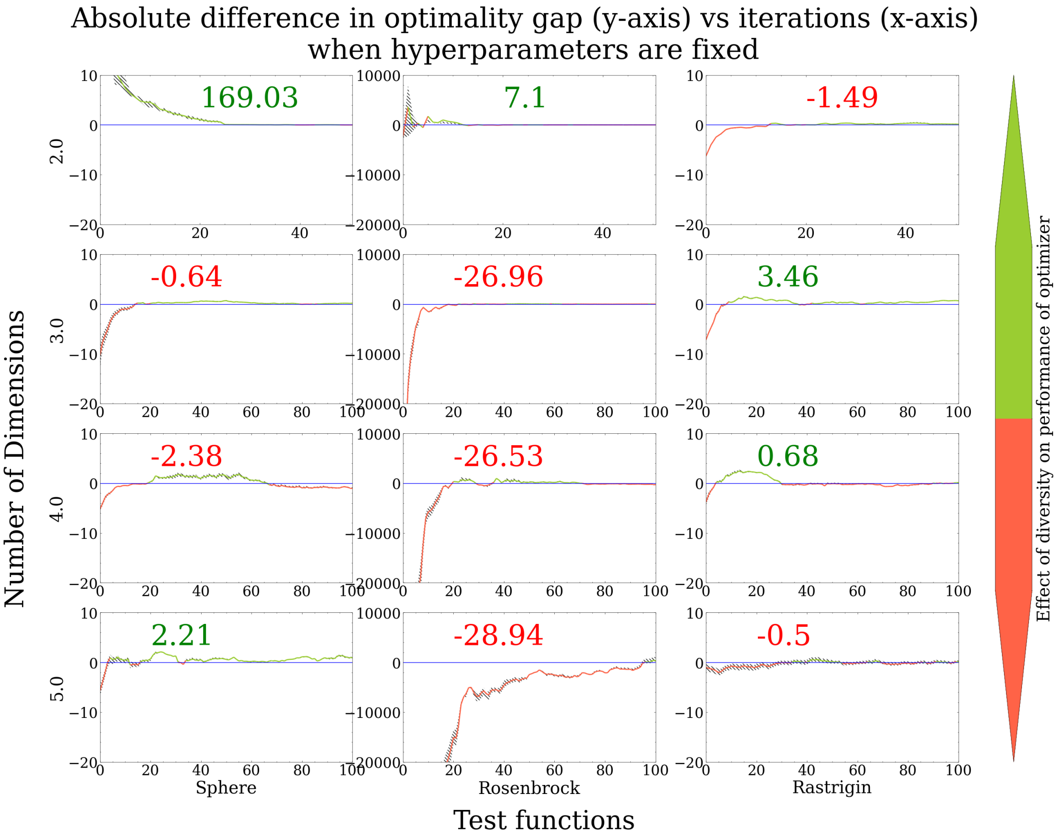


Fig. 13: Optimality gap plot showing effects of diversity when the optimizer is not allowed to fit the hyper-parameters for the Gaussian Process and the hyper-parameters are instead fixed to the values found in Experiment A2. The results from this plot show signficantly improved PCOG values compared to Fig. 11. ‘Rosenbrock’ is the only test function that does not benefit from the diverse samples, its performance remains the same as it was when hyperparameters were optimized, Further discussion on this plot can be read in **S**7

D. V., Olivetti, E., Martin, E., Wieser, E., Silva, F., Lenders, F., Wilhelm, F., Young, G., Price, G. A., Ingold, G.-L., Allen, G. E., Lee, G. R., Audren, H., Probst, I., Dietrich, J. P., Silterra, J., Webber, J. T., Slavič, J., Nothman, J., Buchner, J., Kulick, J., Schönberger, J. L., de Miranda Cardoso, J. V., Reimer, J., Harrington, J., Rodríguez, J. L. C., Nunez-Iglesias, J., Kuczynski, J., Tritz, K., Thoma, M., Newville, M., Kümmerer, M., Bolingbroke, M., Tartre, M., Pak, M., Smith, N. J., Nowaczyk, N., Shebanov, N., Pavlyk, O., Brodtkorb, P. A., Lee, P., McGibbon, R. T., Feldbauer, R., Lewis, S., Tygier, S., Sievert, S., Vigna, S., Peterson, S., More, S., Pudlik, T., Oshima, T., Pingel, T. J., Robitaille, T. P., Spura, T., Jones, T. R., Cera, T., Leslie, T., Zito, T., Krauss, T., Upadhyay, U., Halchenko, Y. O., and Vázquez-Baeza, Y., 2020. “SciPy 1.0: fundamental algorithms for scientific computing in Python”. *Nature Methods,* **17**(3), Mar., pp. 261–272.

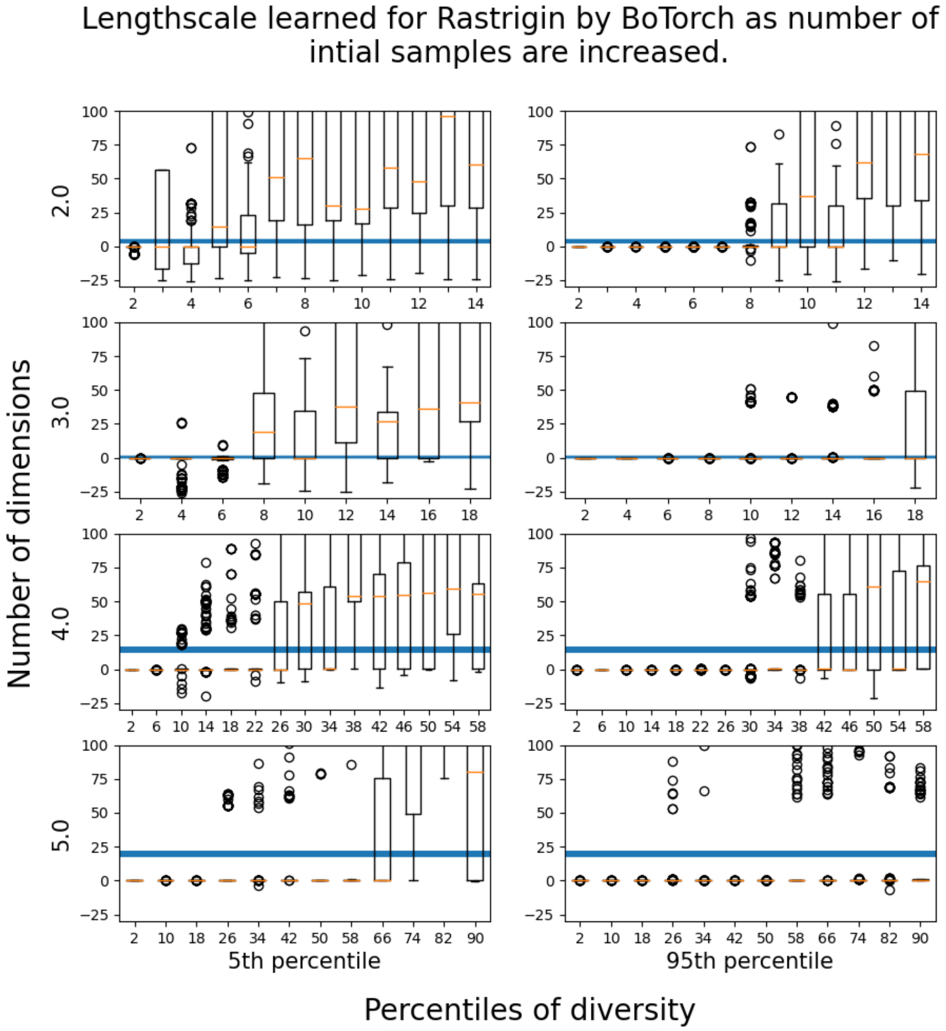


Fig. 14: Box plot showing the lengthscale parameter as learned by Rastrigin test function in 2D and 3D as the training samples are increased. The plot also confirms the existence of a ‘modeling advantage’ for training samples of a particular size. The results are further discussed in **S**7

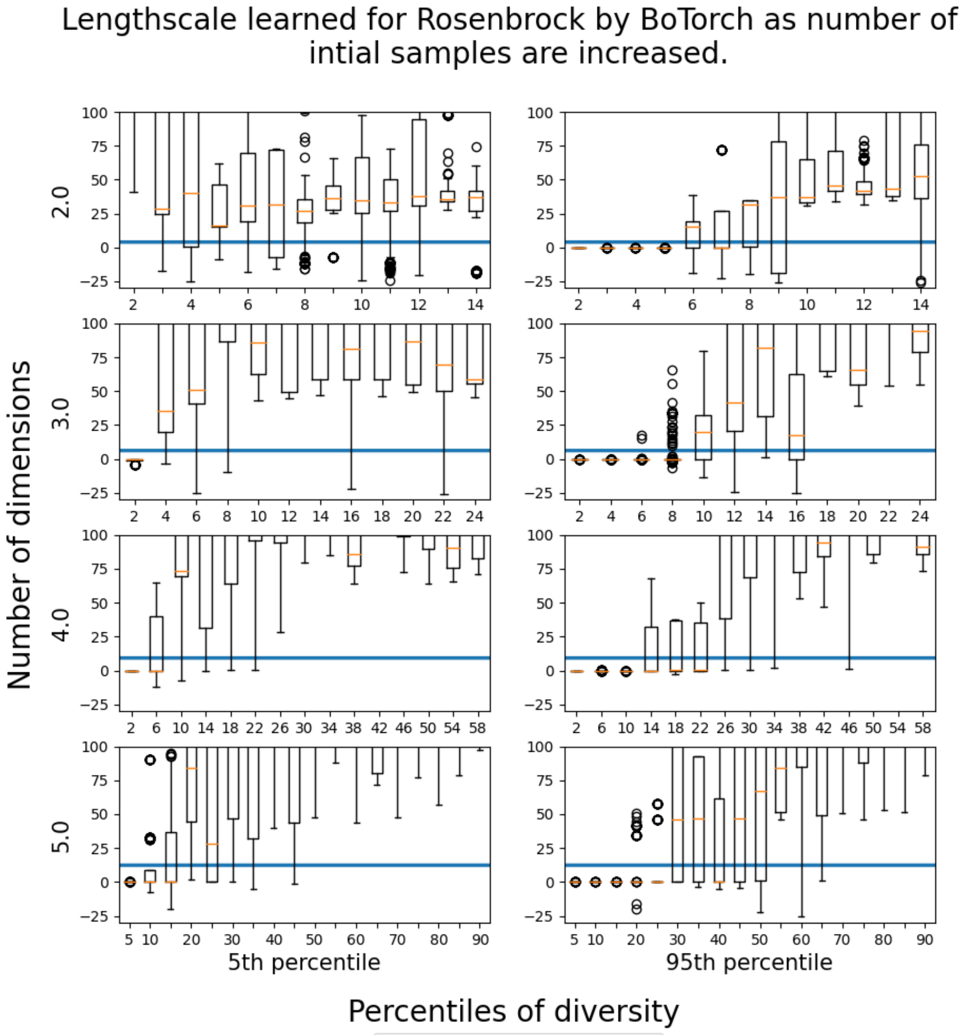


Fig. 15: Box plot showing the lengthscale parameter as learned by Rosenbrock test function in 2D and 3D as the training samples are increased. The plot also confirms the existence of a 'modeling advantage' for training samples of a particular size. The results are further discussed in **S**7

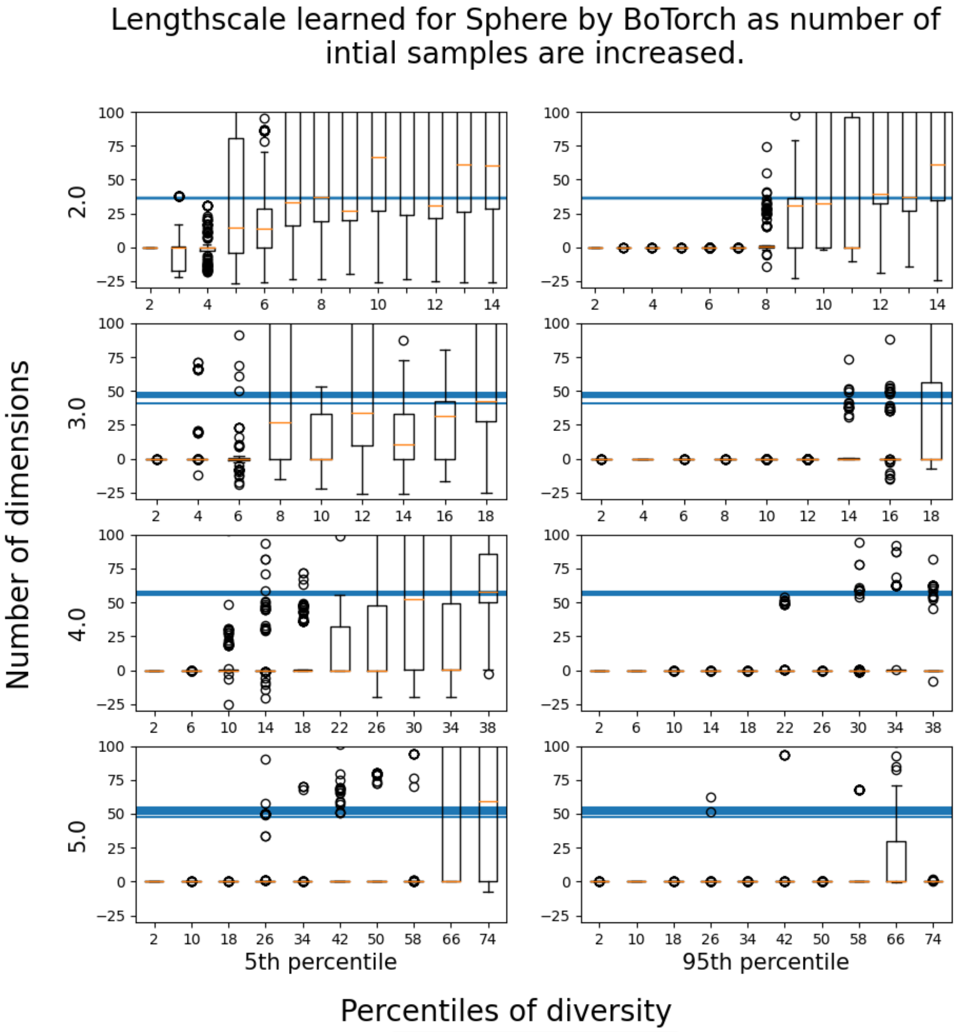


Fig. 16: Box plot showing the lengthscale parameter as learned by Sphere test function in 2D and 3D as the training samples are increased. The plot also confirms the existence of a ‘modeling advantage’ for training samples of a particular size. The results are further discussed in **S**7