We present an approach to Bayesian Optimization that allows for robust search strategies over a large class of challenging functions. Our method is motivated by the belief that the trends useful to exploit in search of the optimum typically are a subset of the characteristics of the true objective function. At the core of our approach is the use of a Latent Gaussian Process Regression model that allows us to modulate the input domain with an orthogonal latent space. Using this latent space we can encapsulate local information about each observed data point that can be used to guide the search problem. We show experimentally that our method can be used to significantly improve performance on challenging benchmarks.

1 Introduction

Many problems in machine learning are associated with finding the optimum of functions that are either explicitly unknown or expensive or slow to evaluate. A popular approach to such problems is Bayesian optimisation (BO) which formulates a sequential search based on a probabilistic model. Underpinning the search are two components, a probabilistic model of the objective function referred to as the surrogate model and an acquisition function describing the strategy of the sequential gathering of samples. A popular approach is to place a Gaussian process (GP) prior over the surrogate model. This is beneficial as it permits the formulation of rich, yet descriptive, distributions over functions such that prior knowledge of the objective can easily be incorporated. The deterministic acquisition function, then, identifies where the objective should be evaluated in the next stage of the sequence given the current beliefs about the objective. The BO loop alternates between fitting the surrogate model to the currently observed data and evaluating the objective function at the location where the acquisition is maximised. The acquisition function can be viewed as a transformation of the surrogate’s estimate of the objective function to a measure of utility in observation with respect to locating the optimum.

Each step of the BO loop is tasked with finding locations in the input domain that should be included into the surrogate model. Importantly, we want these locations to be as informative as possible about the location of the optimum to aid the search problem, as we have a limited evaluation budget. However, due to the greedy sequential structure of the BO loop, we can sometimes add observations which are detrimental to the search problem as a whole. As an example, consider a function with several local minima. A traditional BO loop over such a function is likely to first locate one of the minima during exploration and then switch to a local exploration of this structure. Eventually we will have sufficiently explored the local structure and wish to go back to a global exploration behaviour.
Global exploration become challenging, however, if the length scale used to explore (and model the structure of) the identified local minimum is fundamentally different from the length scale between the extrema. Another example, shown in [Figure 1], is where the objective function has nonsmooth or complicated local structure that the surrogate struggles to model while there is a global structure on a different scale. In a traditional setting, it becomes challenging to model both these structures as the surrogate needs to accommodate for the observations in the posterior belief about the function, making future search either local or uninformed.

The underlying problem in both scenarios is the discrepancy in the objective of supervised learning and surrogate modelling for sequential decision making. In the former, the goal is to find a model which describes every point in the domain equally well. In contrast, the latter’s predictions are never considered directly. Instead, the acquisition translates them to a belief about the utility of a location for finding the optimum. The surrogate, thus, only needs to describe the path towards and the location of the optimum well. The specification of a well-suited surrogate capturing the informative variations in the objective function is a critical challenge in BO [14].

One approach is to introduce a noise model to the surrogate that aims to explain away uninformative parts of the function. However, while a homoscedastic noise model makes a too simplistic global noise assumption, a heteroscedastic GP is challenging to learn from small amounts of data. Another approach is to assume a hierarchical model that warps the input space such that it fits the surrogate model [22, 15, 2]. For function inputs that are smoothly warped, such as machine-learning model hyperparameters set on a logarithmic scale, this has proven to be a successful approach [22]. While transforming the space allows for modelling non-stationary functions, it increases the influence of the possibly only local structure rather than reducing it. This can be detrimental in the presence of uninformative or nonsmooth, local structure.

Acknowledging the challenge of model specification, an approach which includes this as a part of the optimization was presented in [11]. Using a compositional kernel grammar from [6] to induce a space of GP models to choose from. Although this and other model selection procedures [12, 6, 8, 7] themselves have shown promise, however, in addition to the computational overhead the procedures are still reliant on the existence of suitable models in this space. It remains challenging to handle cases where the objective function contains structure that is both hard to specify a-priori and that is unhelpful in guiding the search to the optimum. In the cases of uninformative, local structure, a search for a prior that models those structures well can again be detrimental for the search.

In this paper we propose an approach to address the discrepancy between the objective of fitting the surrogate model and finding the optimum of the function. We adopt the idea of a Latent Gaussian Process regression model [25, 27, 1] for the surrogate model. By introducing a latent space orthogonal to the domain of the function, we can modulate the relevance of each data point individually. Using this formulation allows us to control the influence of each observation, effectively reducing influence of locations believed to be detrimental for the search while highlighting important structures.

2 Latent Gaussian process surrogate models

In a supervised learning setting, predictive distributions of a model take local and global effects into account. Away from the data, these considerations can lead to uninformed predictions as all possible local behaviours have to remain plausible. In contrast, it is not the goal in Bayesian optimization to model the objective function as faithfully as possible. Instead, predictions should propagate the information needed for the decision where to evaluate next.

In this paper we focus on modelling trends that are informative for the search problem while reducing the influence of uninformative variations. In a non-parametric setting this means that our model must have the ability to reduce the influence of uninformative observations. This requirement is closely related to the noise assumptions in supervised learning problems, where conditionally independent additive likelihoods can explain away local disturbance.

Let \( f : \mathcal{X} \to \mathbb{R} \) be the latent objective function and let \( \{x_n, y_n\}_{n=1}^N \) be observed data with \( x_n \in \mathcal{X} = \mathbb{R}^Q \). We aggregate the observations as \( \mathcal{D} = (\mathbf{X}, \mathbf{y}) \) with the matrix \( \mathbf{X} \) containing the stacked observations \( x_n \), and \( \mathbf{y} \) containing the respective noiseless observations \( y_n = f(x_n) \). In this work, we consider the modulated objective function \( \hat{f} : \mathcal{X} \times \mathcal{Z} \) where we augment \( f \) with a latent space \( \mathcal{Z} \).
Figure 1: An illustrative example of the posterior surrogate function density obtained given observations of a nonsmooth function using a GP, a homoscedastic noise GP and a GP latent modulation (ours). The posterior belief for the noise-free and homoscedastic GP surrogates results in the EI acquisition function, shown in blue, making poorly informed decisions for the next query. In contrast, our proposed latent GP is able to reduce the influence of the rapid oscillations that do not match the function prior by explaining part of the variation using the latent variables $Z$. As a result, the acquisition function can utilize a confidently discovered global trend to increase the efficiency of the search. In this example, $\sigma_z$ is set to $1/50$ of the domain range.

such that $f$ can be recovered via the marginalization,

$$p(f | \hat{f}) = \int_Z p(f | \hat{f}, z) p(z) \, dz. \quad (1)$$

We denote the unobserved variable associated with $x_n \in X$ as $z_n \in Z$, their concatenation as $[x_n, z_n]$ and place a GP prior $\hat{f} \sim GP(0, \hat{k})$ over the function and an isotropic Gaussian prior over the latent variable $z_n \sim N(0, \sigma^2_z I)$.

The choice of the kernel $\hat{k}$ governs how the latent dimensions can be used by the modulated model to explain variations in data. Imposing additive structure $\hat{k}([x_i, z_i], [x_j, z_j]) = k_1(x, x') + k_2(z_i, z_j)$, implies that $f$ is itself a GP. A (degenerate) bias kernel $k_2(z_i, z_j) = \delta_{ij} \cdot \sigma_z^2$ acting only on the diagonal of the covariance matrix yields a GP with a homoscedastic Gaussian likelihood. While this is a common modelling choice in supervised learning, a homoscedastic assumption can be problematic for surrogate models as objective functions tend to have different local behaviours in different parts of the input space, leading to uninformed predictive uncertainties. An extension to heteroscedastic likelihoods is also problematic as the low number of observations makes training impractical.

Instead of additive structure, we choose $Z = \mathbb{R}^L$ to be compatible with $X$ and place a single stationary kernel on the complete space $X \times Z$. Common choices for stationary kernels in Bayesian optimization include the squared exponential kernel and the Matérn family of kernels [20]. This choice of kernel allows our model to reduce the relevance of individual observations (and thus explain them) by moving them away from other data in $Z$. Figure 1 shows a comparison of posteriors of our proposed method with noiseless and noisy GPs. In special cases, our model behaves similarly to common model choices: (1) Setting all $z_n = 0$ is equivalent to noiseless GP regression over $X$. (2) Constraining the $z_n$ to a constant pairwise distance resembles homoscedastic GP regression. (3) Moving all $z_n$ orthogonally along their own dimension yields heteroscedastic regression.

A high-dimensional latent space $Z$ allows the modulated objective function many degrees of freedom to explain away data and conform to the GP prior yielding uninformed models. In supervised learning, this trade-off can be performed by placing priors on the amplitude of the effects which should be explained in the output space via the noise variance. This is often problematic for Bayesian optimization because amplitudes in the output space are unknown. Our stationary joint kernel allows us to formulate this trade-off in the input space, which is often well-understood. A low dimensionality of $Z$ together with a small prior variance $\sigma^2_z$ for the latent variables $z_n$ imposes more structure. Points close together in $X$-space can never be explained independently by moving them too far apart in $Z$-space with respect to a kernel length scale $l$. 

3
3 Approximate inference

When performing inference in the model we want to be robust with respect to the hyper-parameter settings in the surrogate model. To achieve this we perform full marginalisation via Markov Chain Monte Carlo (MCMC) algorithms [19]. In addition to the model hyper-parameters we also need to address the latent variable $z$.

In our model, the jointly dependent $z$ grows with the number of observations which makes MCMC posterior samples increasingly challenging to obtain as more observations are being gathered. Since we know that the choice of kernel parameters is critical, we retain inference using MCMC for the kernel parameters and approximate the latent variables via variational inference. The variational bound is given by

$$
\log p(y | X) = \log \int p(y | X, \theta, z) p(z, \theta) d\theta dz \\
\geq \int q(z) \log \frac{p(y | X, \theta, z) p(\theta | z)}{q(z)} dz d\theta \\
= \mathbb{E}_{q(z)} \left[ \log \int p(y | X, \theta, z) p(\theta | z) d\theta \right] - KL(q(z) \| p(z)) \\
= \mathcal{L}(m, s),
$$

where we use a fully factorized mean field approximation $q(z) = \prod_{n=1}^N q(z_n) = \prod_{n=1}^N \mathcal{N}(z_n | m_n, s_n)$ depending on $2NL$ variational parameters $m = (m_1, \ldots, m_N)$ and $s = (s_1, \ldots, s_N)$. This bound is optimized using the Adam optimizer [9] to obtain a variational belief for the latent variables $z$. While the KL-divergence can be evaluated in closed form, the expected log-likelihood of the data is intractable due to the marginalization of the hyperparameters. We approximate the expectation over $q(z)$ via sampling and the inner marginalization with MCMC using slice sampling [20].

Running independent MCMC for every iteration of the optimization is costly. However, since the belief about $z$ does not change much between iterations, we can continue on the same Markov chain between bound iterations and significantly increase the performance. We draw one sample from $q(z)$ per MCMC sample and initialize the MCMC algorithm with a burn-in of 200 samples and 100 initial posterior samples to initialize $q(z)$. For every following iteration of the optimizer, we use no burn-in and draw 10 joint samples.

**Informative predictions** In the modulated objective function $\hat{f}$, the latent space $\mathcal{Z}$ is used to explain effects which are not informative for decision making. We therefore evaluate the acquisition function on the GP at the hyperplane $z_\ast = 0$, so that

$$
q(y_\ast | x_\ast, D) = \int p(y_\ast | x_\ast, z_\ast = 0, \hat{f}) q(\hat{f} | D) d\hat{f} \\
= \int p(y_\ast | x_\ast, z_\ast = 0, D, z, \theta) q(\theta, z) d\theta dz \\
= \int p(y_\ast | x_\ast, z_\ast = 0, D, z, \theta) p(\theta | D, z) q(z) d\theta dz.
$$

We approximate the marginalizations of the kernel parameters $\theta$ and the latent variables via MCMC on $\theta$ with joint sampling from $q(z) = q(z_1, \ldots, z_N)$. Given a sample of $z$ and $\theta$, the predictive distribution is a standard GP marginal and can be computed in closed form. Note that marginalizing the prior belief of $z_\ast$ would yield a posterior belief about the (unmodulated) objective function following [1]. This is not desirable as the marginalization would reintroduce effects which have been explained as local disturbance.

4 Experiments

We now demonstrate the benefit of our surrogate latent variable extension via experiments on standard BO benchmarks with quantitative comparisons to baseline methods. In addition we propose a heuristic to add uncertainty to the utility estimate.
We first evaluate the benefit of extending the surrogate model to include a latent input variable that
When domain specific knowledge is available, we use the Matérn 5/2 kernel for all surrogates and the model-marginalised expected improvement.
We provide two quantitative metrics. Firstly, we follow standard practice to compare across datasets and provide the mean gap estimated over 20 runs \[11\]. The gap measure is defined as
\[
\frac{f(x_{\text{first}}) - f(x_{\text{best}})}{f(x_{\text{first}}) - f(x_{\text{optimum}})},
\]
where \(f(x_{\text{first}})\) is the minimum function value among the first initial random points, \(f(x_{\text{best}})\) is the best function value found within the evaluation budget and \(f(x_{\text{optimum}})\) is the function’s true optimum. Methods are judged to have very similar or equivalent performance to the best performing if not significantly different, determined by an one-sided paired Wilcoxon signed rank test at 5% significance \[11\]. Secondly, as the nonparametric Wilcoxon signed rank test requires the individual runs, we report mean and standard deviation on the simple regret, the difference between \(f(x_{\text{best}})\) and \(f(x_{\text{optimum}})\).

We use the Matérn 5/2 kernel for all surrogates and the model-marginalised expected improvement acquisition function evaluated via slice sampling \[21\]. The inference for our latent variable extension also includes variational approximations, as described in Section 3. For the maximization of the expected utility with respect to input location, we use \(\delta\)-cover sampling, as in \[5\], for all models where the expected utility in the experiments is the model-marginalised expected improvement. For further details, we refer to the supplementary material.

**Baselines and metric** We compare using our proposed latent variable extension against not using the extension on a popular GP model setup for BO. In addition, we compare the latent variable extension against other methods of handling challenging structure in the objective function, namely (i) a noiseless GP, (ii) a GP with homoscedastic noise and (iii) a non-stationary, Warped GP \[22\].

We compare using our proposed latent variable extension against not using the extension on a popular GP model setup for BO. In addition, we compare the latent variable extension against other methods of handling challenging structure in the objective function, namely (i) a noiseless GP, (ii) a GP with homoscedastic noise and (iii) a non-stationary, Warped GP \[22\].

Baseline and metric We perform the comparisons on function benchmarks from \[3, 17\] using the default domains provided by the benchmark detailed in the supplementary material. A number of problems are available in different dimensionalities which are denoted in with the results. We note that in experiments in other work, domains are sometimes restricted, which changes the characteristics and difficulty of the benchmark problem. In addition, problems are marked with the descriptive properties given in \[3, \text{evalset}\] that can reflect the relative difficulty of the task.

**Priors on the latent input variables** The prior \(p(z) = \mathcal{N}(z | 0, \sigma_z^2)\) can be parameterized in relation to the relative scale of the characteristics to be ignored. We specify the function prior over the product space \(X \times Z\) using a kernel with common parameters for \(x_n\) and \(z_n\). Thus, the standard deviation of the prior \(\sigma_z\) relates directly to distances in \(X\)-direction.

When domain specific knowledge is available, \(p(z)\) may be specified at an appropriate scale. However, we often do not have access to such knowledge. In all our experiments we adopt a hierarchical prior approach whereby \(\sigma_z\) is sampled uniformly from a small candidate set at each evaluation. Specifically, we set \(L = 1\) and \(\sigma_z \sim U(\{0.1d, 0.01d, 0\})\) where \(d = \sqrt{Q}\), the length of the diagonal of the unit Q-hypercube. A choice of \(\sigma_z \rightarrow 0\) corresponds to a noiseless GP without latent covariates. We found that this approach performed well empirically and is applied consistently across all our experiments where not otherwise specified.

**4.1 Latent Gaussian process surrogate model**

We first evaluate the benefit of extending the surrogate model to include a latent input variable that will alter the relevance of the observed data as illustrated in Figure 1. Recall that we seek to provide a more informed belief about the function far from data to produce sensible sampling decisions via the acquisition function.

**Surrogate function modelling in the presence of nonsmooth structure** Figure 2 shows the behaviour of the latent input model on the Holder Table benchmark function from \[3\]. We show two scenarios, one with and one without an added nonsmooth component in form of saw-tooth and square wave functions. The challenge is to solve the optimization adequately using a GP surrogate despite the nonsmooth, uninformative structure that is difficult to model with a GP. While showing comparatively little variance in the simple regret measure on the default Holder Table benchmark, a noiseless GP’s variance increases substantially for the corrupted version, as the surrogate is influenced heavily by the nonsmooth component.
Figure 2: A comparison of experiments on the Holder Table benchmark [3] evalset (left) and a corrupted version with added nonsmooth structure (right). We show plots of the respective functions on top and performance in terms of regret so far for 30 repetitions in the bottom. We show both mean regret (line) and standard deviation (shading). The nonsmooth structure is challenging for a noiseless GP to model and leads to a high variance in-between runs. Warped and noisy GPs explain away the corruption but their performance plateaus as no informative trends are can be identified. GP+LM reliably identifies these trends and reliably finds good solutions.

Adding a homoscedastic noise model (GP+HS) helps during early iterations, however, the BO loop is unable to locate and exploit minima as informative trends are explained as noise instead and the mean regret plateaus. Warped GPs show similar behaviour in terms of performance. In contrast, our surrogate (GP+LM) shows good performance for both the smooth and nonsmooth settings resulting in both an improved mean regret but also a low variance in the resulting regret. While the regret drops slower compared to GP+HS in the beginning due to GP+LM tendency to explore more in the start, given enough observations, a good solution is found reliably.

Evaluation on benchmark suite Table 1 presents results across a wide range of benchmark functions consisting of the SigOpt benchmark suite [3] and two additional hyper parameter tuning benchmarks [17]. The benchmark functions have challenging properties such as complicated and oscillatory local structure and high dimensionality. In general, the GP+HS and Warped GP tend to be outperformed by the noiseless GP surrogate and GP+LM reliably shows comparable or improved performance. On some benchmarks known to be difficult functions, the difference is particularly large, such as Ackley 6D, Deflected Corrugated Spring 10D and Shubert 2D. The Ackley 6D, exhibiting the largest increase in performance by our latent variable extension, is described as “technically oscillatory, but with such a short wavelength that its behavior probably seems more like noise” [3].

4.2 Limiting the overconfidence of surrogate models

Experiments above have shown that the use of GP+LM increases robustness against complicated and uninformative local structure in the objective function. Such structure can cause a surrogate to be overconfident about the function at particular locations and underconfident about the function as whole as it can require extreme parameter choices such as very short length scales to be explained. This can be particularly detrimental for the search due to the acquisition function’s role of balancing efficient exploration of the domain with the exploitation of current knowledge of the underlying function [18]. With an overconfidence around the currently predicted minimum this trade-off can easily be tilted too heavily towards exploitation. We address this problem further by investigating the application of a heuristic to reduce the impact of such overconfidence and provide an empirical
Table 1: Mean gap performance for various test functions, higher is better. The upper table shows the results after 50 objective function evaluations and the lower table after 100 evaluations. Due to computational cost, Warped GP results are only reported for 50 evaluations. Methods not significantly different from the best within a small region around the location proposed by the acquisition function as a result of the two functions, Powell Triple Log 12D and Deflected Corrugated Spring 10D, show different properties. For the former, BO using an auxiliary optimizer significantly outperforms random for reference. This experiment is performed using standard noiseless GP surrogates.

Evaluation of its performance. The heuristic is easy to implement in any surrogate-based search setting and can offer complementary benefit to GP+LM. Limiting the precision of the location selection We limit the effect of overconfidence by injecting noise into the location selection. First, we find the location with highest utility \( x^* \) with high precision, such as via an auxiliary optimizer [18] or a sufficiently dense grid. We then perturb this location choice with uniform noise, 

\[
\hat{x} = x^* + \epsilon, \quad \epsilon \sim \mathbb{U}([-\epsilon_N, \epsilon_N]^Q),
\]

and use \( \hat{x} \) as the next query location. This implies that we are indifferent towards sample locations within a small region around the location proposed by the acquisition function as a result of the surrogate belief. We refer to this heuristic as the \( N_\Delta \)-construction as we define \( \epsilon_N \) as the side length of \( N_\Delta \) hypercubes that fit densely in the search space, the size of the indifference regions. This parameterization is independent of the input dimensionality and can easily be interpreted with respect to grid sampling. There are two special cases: \( N_\Delta = 1 \) yields random search while \( N_\Delta \to \infty \) disables the heuristic.

Parameterization We need to select the \( N_\Delta \), which for a given input dimensionality \( Q \) implies each hypercube’s side \( \epsilon_N \) to be \( \epsilon_N = 1/\sqrt[Q]{N_\Delta} \) if the domain itself is a unit hypercube. Similarly to the case of \( \sigma_z \) for the latent variable extension, it can be difficult to parameterize \( N_\Delta \) without specific domain knowledge about the precision of the function to optimize. For this reason, we sample \( N_\Delta \sim \mathbb{U}(\{10^4, 10^6, \infty\}) \) at every BO iteration, where \( N_\Delta \to \infty \) corresponds to using the expected utility maximum directly.

Illustration of overconfident estimates of utility In Figure 3 we provide two examples of using the heuristic with constant \( N_\Delta \) and sampled uniformly from the set denoted as \( N_\Delta \sim \mathbb{U}(\phi_\Delta) \). We compare using the added heuristic to, with the same auxiliary optimizer, directly accepting the maximum utility location as the query location. We include BO on a grid as well as random search for reference. This experiment is performed using standard noiseless GP surrogates.

The two functions, Powell Triple Log 12D and Deflected Corrugated Spring 10D, show different properties. For the former, BO using an auxiliary optimizer significantly outperforms random
Figure 3: A comparison of different settings of the $N_\Delta$-construction compared to grid evaluation and limit cases based on noiseless GP surrogates. The upper row shows mean regret (line) and standard deviations (shading) over 30 repetitions, the lower row shows the utility values found at the respective iteration. While the auxiliary optimizer successfully finds locations with high utility in both experiments, it performs worse than random search in the Deflected Corrugated Spring 10D example due to the exploitation of overconfidence in the utility model. The $N_\Delta$-construction’s added robustness leads to good performance for both experiments, while performing BO on a grid fails due to high dimensionality.

\begin{table}[h]
\centering
\begin{tabular}{|l|c|c|c|c|c|c|c|c|c|}
\hline
\textbf{Benchmark} & \textbf{Evals} & \textbf{Dim} & \textbf{Properties} & \textbf{GP $\infty$} & \textbf{GP $N_\Delta$} & \textbf{GP + LM $\infty$} & \textbf{GP + LM $N_\Delta$} \\
\hline
Brannin01 & 50 & 2 & none & 0.00 ± 0.00 & 0.00 ± 0.00 & 0.00 ± 0.00 & 0.00 ± 0.00 \\
Powell Triple Log & 50 & 12 & none & 3.83 ± 0.80 & 3.96 ± 0.53 & 4.20 ± 0.62 & 4.06 ± 0.71 \\
Hartmann & 50 & 6 & boring & 0.13 ± 0.12 & 0.17 ± 0.31 & 0.19 ± 0.38 & 0.18 ± 0.28 \\
Griewank & 50 & 2 & oscillatory & 0.08 ± 0.18 & 0.06 ± 0.05 & 0.07 ± 0.06 & 0.06 ± 0.04 \\
Shubert & 50 & 2 & oscillatory & 84.7 ± 64.9 & 103 ± 56.4 & 90.9 ± 61.1 & 89.2 ± 60.8 \\
Cosine Mixture & 50 & 10 & oscillatory & 0.83 ± 0.38 & 0.65 ± 0.18 & 0.76 ± 0.36 & 0.82 ± 0.15 \\
Drop-Wave & 50 & 10 & oscillatory & 0.92 ± 0.03 & 0.86 ± 0.07 & 0.92 ± 0.04 & 0.84 ± 0.08 \\
Deflected Corrugated Spring & 50 & 10 & oscillatory & 3.92 ± 1.80 & 1.53 ± 0.68 & 2.99 ± 1.32 & 1.11 ± 0.39 \\
Weierstrass & 50 & 8 & complicated & 7.10 ± 1.64 & 6.50 ± 1.56 & 6.87 ± 1.20 & 6.33 ± 1.63 \\
Holder Table & 50 & 2 & complicated, oscillatory & 0.73 ± 1.21 & 0.56 ± 1.15 & 0.02 ± 0.03 & 0.19 ± 0.66 \\
Ackley $[-10, 30]^d$ & 50 & 6 & complicated, oscillatory & 11.3 ± 6.91 & 11.4 ± 7.44 & 6.60 ± 5.18 & 3.82 ± 0.78 \\
NN Climate Model Crashes & 50 & 9 & unknown & 0.03 ± 0.01 & 0.02 ± 0.01 & 0.03 ± 0.01 & 0.02 ± 0.01 \\
NN Climate Model Crashes & 50 & 9 & unknown & 0.05 ± 0.05 & 0.04 ± 0.03 & 0.03 ± 0.01 & 0.04 ± 0.01 \\
\hline
\end{tabular}
\caption{Simple regret performance for various test functions, lower is better. We report mean and standard deviation for 30 repetitions. Methods with a mean performance at most one standard deviation away from the best performing method are shown in bold.}
\end{table}

sampling while for the latter, random sampling performs better than BO and choosing the maximum utility location. This can have its explanation in that the former function is comparably slowly varying while the latter is oscillatory. Grid evaluation performs poorly in these high dimensional problems due to the curse of dimensionality. Notably, by only adding the proposed heuristic $N_\Delta$ to the GP which uses the auxiliary optimizer, good solutions can be found reliably in both cases.

**Empirical evaluation** We expand the experiment above across a diverse range of functions and show that the heuristic can be combined with GP+LM surrogate models to achieve the benefits of both contributions. Table 2 details these results showing the average regret at the end of 50 observations. We can confirm that the addition of the heuristic to a GP surrogate (GP $N_\Delta$) improves over the GP surrogate without the heuristic (GP $\infty$) in the majority of cases. Similarly, adding the heuristic to GP+LM improves performance in most cases. The best overall performance is achieved
by combining GP+LM with the $N_{\Delta}$-construction. The most notable performance increase is achieved for high-dimensional problems with complicated or oscillatory behaviour.

5 Conclusion

We have presented an approach to Bayesian optimisation which utilises a Gaussian process surrogate model with an additional latent input variable extension. The latent variable allows us to influence the function evaluations in an informed manner to handle structure in the objective function detrimental for finding the optimum. We show experimentally how our approach is able to reliably solve benchmarks with challenging local structure. Importantly our approach can be included into any Gaussian process based surrogate model used for Bayesian Optimization.

6 Acknowledgements

This project was supported by Engineering and Physical Sciences Research Council (EPSRC) Doctoral Training Partnership, EPSRC CAMERA project (EP/M023281/1), the Royal Society, and the Hans Werthén Fund at The Royal Swedish Academy of Engineering Sciences. We would like to thank Zhenwen Dai at Amazon for useful discussions.

References

[1] Erik Bodin, Neill D. F. Campbell, and Carl Henrik Ek. “Latent Gaussian Process Regression”. In: (July 18, 2017). arXiv: 1707.05534 [cs, stat].

[2] Roberto Calandra et al. “Manifold Gaussian processes for regression”. In: 2016 International Joint Conference on Neural Networks (IJCNN). IEEE, 2016, pp. 3338–3345.

[3] Scott Clark and Patrick Hayes. SigOpt Web page. http://www.sigopt.com 2019.

[4] Nando De Freitas, Alex J. Smola, and Masrour Zoghi. “Exponential Regret Bounds for Gaussian Process Bandits with Deterministic Observations”. In: Proceedings of the 29th International Conference on International Conference on Machine Learning. ICML ‘12. Edinburgh, Scotland: Omnipress, 2012, pp. 955–962. ISBN: 978-1-4503-1285-1.

[5] Nando De Freitas, Alex Smola, and Masrour Zoghi. “Exponential regret bounds for Gaussian process bandits with deterministic observations”. In: arXiv preprint arXiv:1206.6457 (2012).

[6] David Duvenaud et al. “Structure discovery in nonparametric regression through compositional kernel search”. In: arXiv preprint arXiv:1302.4922 (2013).

[7] Jacob Gardner et al. “Discovering and exploiting additive structure for Bayesian optimization”. In: Artificial Intelligence and Statistics. 2017, pp. 1311–1319.

[8] Roger Grosse et al. “Exploiting compositionality to explore a large space of model structures”. In: arXiv preprint arXiv:1210.4856 (2012).

[9] Diederik Kingma and Jimmy Ba. “Adam: A Method for Stochastic Optimization”. In: (Dec. 22, 2014). arXiv: 1412.6980 [cs].

[10] Gustavo Malkomes and Roman Garnett. “Automating Bayesian Optimization with Bayesian Optimization”. In: Advances in Neural Information Processing Systems 31. Ed. by S. Bengio et al. Curran Associates, Inc., 2018, pp. 5984–5994.

[11] Gustavo Malkomes and Roman Garnett. “Automating Bayesian optimization with Bayesian optimization”. In: Advances in Neural Information Processing Systems. 2018, pp. 5984–5994.

[12] Gustavo Malkomes, Charles Schaff, and Roman Garnett. “Bayesian optimization for automated model selection”. In: Advances in Neural Information Processing Systems. 2016, pp. 2900–2908.

[13] M. McLeod, M.A. Osborne, and S.J. Roberts. “Optimization, fast and slow: Optimally switching between local and Bayesian optimization”. In: vol. 8. 2018, pp. 5521–5530.

[14] Jonas Močkus. “On Bayesian methods for seeking the extremum”. In: Optimization Techniques IFIP Technical Conference. Springer. 1975, pp. 400–404.

[15] ChangYong Oh, Efstratios Gavves, and Max Welling. “BOCK : Bayesian Optimization with Cylindrical Kernels”. In: ICML. 2018.
[16] Tim Salimans, Diederik P. Kingma, and Max Welling. “Markov Chain Monte Carlo and Variational Inference: Bridging the Gap”. In: (Oct. 23, 2014). arXiv:1410.6460 [stat]

[17] scikit-optimize. DOI: 10.5281/zenodo.1207017

[18] Bobak Shahriari et al. “Taking the human out of the loop: A review of Bayesian optimization”. In: Proceedings of the IEEE 104.1 (2015), pp. 148–175.

[19] B. Shahriari et al. “Taking the Human Out of the Loop: A Review of Bayesian Optimization”. In: Proceedings of the IEEE 104.1 (Jan. 2016), pp. 148–175. ISSN: 0018-9219. DOI: 10.1109/JPROC.2015.2494218

[20] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. “Practical Bayesian Optimization of Machine Learning Algorithms”. In: Advances in Neural Information Processing Systems 25. Ed. by F. Pereira et al. Curran Associates, Inc., 2012, pp. 2951–2959.

[21] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. “Practical bayesian optimization of machine learning algorithms”. In: Advances in neural information processing systems. 2012, pp. 2951–2959.

[22] Jasper Snoek et al. “Input warping for bayesian optimization of non-stationary functions”. In: International Conference on Machine Learning. 2014, pp. 1674–1682.

[23] M. Todorović et al. “Bayesian inference of atomistic structure in functional materials”. In: npj Computational Materials 5.1 (2019). DOI: 10.1038/s41524-019-0175-2

[24] R.E. Turner and M. Sahani. “Demodulation as probabilistic inference”. In: IEEE Transactions on Audio, Speech and Language Processing 19.8 (2011), pp. 2398–2411. DOI: 10.1109/TASL.2011.2135852

[25] Chunyi Wang and Radford M. Neal. “Gaussian Process Regression With Heteroscedastic Or Non-Gaussian Residuals”. In: CoRR (2012). arXiv:1212.6246 [stat.ML]

[26] Florian Wenzel et al. “Fast Inference in Non-Conjugate Gaussian Process Models via Data Augmentation”. In: (). p. 6.

[27] Fariba Yousefi et al. “Unsupervised Learning with Imbalanced Data via Structure Consolidation Latent Variable Model”. In: (June 30, 2016). arXiv:1607.00067 [cs, stat]