The Annals of Applied Statistics
2025, Vol. 19, No. 1, 191–211
https://doi.org/10.1214/24-AOAS1951

CONTOUR LOCATION FOR RELIABILITY IN AIRFOIL SIMULATION EXPERIMENTS USING DEEP GAUSSIAN PROCESSES

BY ANNIE S. BOOTH[1,a], S. ASHWIN RENGANATHAN[2,b] AND ROBERT B. GRAMACY[3,c]

[1]*Department of Statistics, North Carolina State University,* [a]*annie_booth@ncsu.edu*
[2]*Department of Aerospace Engineering, Pennsylvania State University,* [b]*ashwin.renganathan@psu.edu*
[3]*Department of Statistics, Virginia Tech,* [c]*rbg@vt.edu*

Bayesian deep Gaussian processes (DGPs) outperform ordinary GPs as surrogate models of complex computer experiments when response surface dynamics are nonstationary, which is especially prevalent in aerospace simulations. Yet DGP surrogates have not been deployed for the canonical downstream task in that setting: reliability analysis through contour location (CL). In that context we are motivated by a simulation of an RAE-2822 transonic airfoil which demarcates efficient and inefficient flight conditions. Level sets separating passable vs. failable operating conditions are best learned through strategic sequential designs. There are two limitations to modern CL methodology which hinder DGP integration in this setting. First, derivative-based optimization underlying acquisition functions is thwarted by sampling-based Bayesian (i.e., MCMC) inference, which is essential for DGP posterior integration. Second, canonical acquisition criteria, such as entropy, are famously myopic to the extent that optimization may even be undesirable. Here we tackle both of these limitations at once, proposing a hybrid criterion that explores along the Pareto front of entropy and (predictive) uncertainty, requiring evaluation only at strategically located "triangulation" candidates. We showcase DGP CL performance in several synthetic benchmark exercises and on the RAE-2822 airfoil.

1. Introduction. Computer simulations are increasingly utilized for experimentation when direct manipulation is impossible or infeasible. Although diverse applications abound (e.g.,Tietze (2015), Bremer (2019), Lippe et al. (2019)), here we are motivated by one in particular: physics-based simulations in aeronautics. These involve computational fluid dynamics (CFD[1]) simulation requiring the numerical solution of a set of stiffly coupled nonlinear partial differential equations satisfying conservation laws (e.g., Pamadi et al. (2004), Vassberg et al. (2008), Mehta et al. (2014)). A specific example is aerodynamic modeling of the interaction between an aircraft, aircraft wing, or wing section (a.k.a.,"airfoils") and the surrounding airflow at transonic speeds (i.e., flight at ∼70–80% of the speed of sound). Transonic flow straddles the line between subsonic (flying slower than the speed of sound) and supersonic (faster than sound) flow that makes it particularly complex to model. When the computational costs of obtaining runs severely limits how many may be collected, a "surrogate model" or "emulator," which is a fitted statistical model trained on data from a computer experiment, may be essential for downstream tasks requiring model evaluation at unobserved inputs.

The canonical surrogate is a Gaussian process (GP; Santner, Williams and Notz (2018), Gramacy (2020)), but in their traditional form, GPs suffer from the assumption of stationarity. They must impart similar dynamics across the entire input space which limits their

Received August 2023; revised August 2024.
Key words and phrases. Active learning, emulator, entropy, Pareto front, sequential design, surrogate, tricands, uncertainty quantification.

[1]See Supplementary Material A (Booth, Renganathan and Gramacy (2025)) for a complete list of acronyms.

ability to cope with regime shifts, as are common in aerospace simulations. Aircraft experience "shocks" when the sound barrier is broken. Several adaptations have been considered to allow for nonstationary GP flexibility (see Sauer, Cooper and Gramacy (2023b) for a review), but there are drawbacks. Nonstationary kernels suit spatial applications with low input dimension (Higdon, Swall and Kern (1999), Paciorek and Schervish (2003)). Partition—or treed—GPs divvy up the input space (Gramacy and Lee (2008), Bitzer, Meister and Zimmer (2023)), sacrificing global scope. Locally approximate GPs are data hungry (Gramacy and Apley (2015)).

The deep Gaussian process (DGP; Damianou and Lawrence (2013)) shares many of these same ingredients but has fewer drawbacks. DGPs use stationary GPs to spatially warp the original inputs. Warped inputs were first proposed in the spatial statistics community (Sampson and Guttorp (1992), Schmidt and O'Hagan (2003)) but have recently been popularized by machine learners who showcased DGP prowess on large-scale classification and regression tasks (Damianou and Lawrence (2013), Dunlop et al. (2018)). The challenge in DGP inference lies in learning high-dimensional latent warping variables. With large training data sizes and a penchant for computational thrift, many in machine learning embraced approximate variational inference (VI; Bui et al. (2016), Salimbeni and Deisenroth (2017)).

Yet the supposed thriftiness of approximate VI often disappoints because the algorithms require careful tuning. Moreover, optimization in lieu of full posterior integration sacrifices uncertainty quantification (UQ), which is essential to safety/reliability tasks, especially in data-poor settings. We instead embrace the fully-Bayesian inferential scheme of Sauer, Gramacy and Higdon (2023), leveraging modern advances in Markov chain Monte Carlo (MCMC). This fully-Bayesian DGP has been shown to outperform VI-based approximate DGP competitors (Sauer, Cooper and Gramacy (2023a, 2023b)), and to excel in surrogate modeling settings where training data is limited (Sauer, Gramacy and Higdon (2023)).

1.1. *RAE-2822 airfoil computer experiment.* Here we are motivated by a computer simulation of transonic flow past an RAE-2822 airfoil. The model, utilizing Reynolds Averaged Navier–Stokes equations, is solved via SU2 (Economon et al. (2016)), a public simulation suite. A spherical domain of 100 airfoil chord length radius is used to model the fluid domain, which is discretized with 27,857 triangular and hexahedral mesh elements. The hexahedral elements are employed closer to the surface of the airfoil to capture the boundary layer. Additionally, the mesh density is made deliberately higher in the near-field of the airfoil to resolve the shock better; see Figure 1. Conservation laws (mass, momentum, and energy) are enforced in each mesh element forming a system of nonlinear equations which are solved iteratively. The computational cost of the solution scales cubically with the number of mesh elements. All simulations seek the steady-state solution, with pseudo time-stepping. In serial execution each simulation takes 20–30 minutes of wall-clock time for convergence and about 150 seconds with eight parallel MPI processes.

We consider seven input variables: three freestream conditions and four shape parameters. Freestream conditions include angle of attack with bounds $[0, 10]$, Mach number ranging in $[0.7, 0.9]$, and Reynolds number with bounds $[5 \times 10^6, 15 \times 10^6]$. The airfoil shape and surrounding mesh are modified by displacing free-form deformation control points around the airfoil. The quantity of interest is the lift-drag ratio $L/D \equiv C_l/C_d$, where C_l and C_d are the lift and drag coefficients, respectively. L/D is a measure of the aerodynamic efficiency of an aircraft and directly impacts the amount of fuel required. Strict airworthiness standards from the Federal Aviation Administration limit the amount of fuel an aircraft can burn per passenger mile, since more jet fuel burned results in more greenhouse gas emissions; thus, aircraft designers aim to meet a minimum required L/D. Our current goal is to identify where $L/D < 3$, which is considered failure.

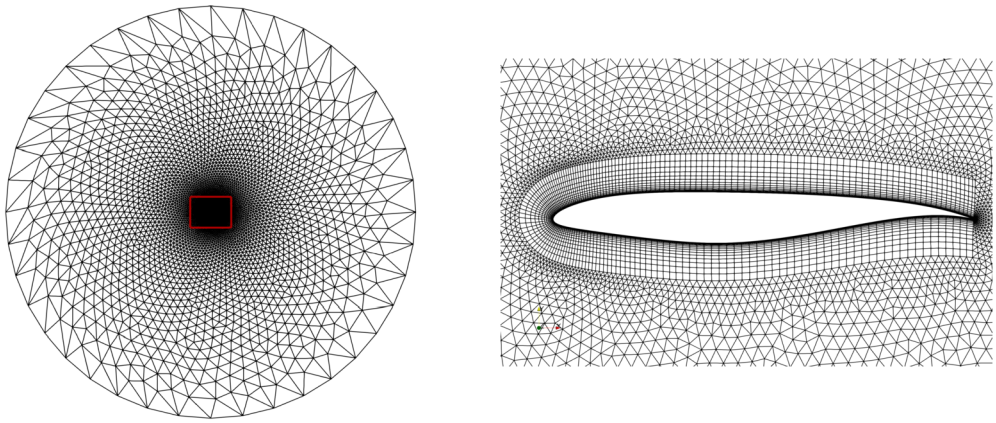

FIG. 1. *Computational mesh. Left: Full domain showing all* 27,857 *mesh elements. Right: Near-field mesh refinement for the RAE-*2822 *airfoil to capture the boundary layer and shock.*

DGPs are relatively new to this area, but expert knowledge suggests there is nonstationarity in the response surfaces of these simulations. To test this, we fit both shallow and deep GP surrogates to simulations obtained from a 500-run space-filling Latin hypercube sample (LHS; McKay, Beckman and Conover (1979)) in the 7d input space, specifically: the Bayesian DGP using elliptical slice sampling (DGP ESS) of Sauer, Gramacy and Higdon (2023), a traditional stationary GP via maximum likelihood estimates (GP MLE), and the approximate VI DGP (DGP VI) of Salimbeni and Deisenroth (2017). Further details about these comparators will be revealed as the paper progresses. Predictive performance is assessed on a hold-out 4500-point LHS via root mean squared error (RMSE, measuring accuracy) and continuous ranked probability score (CRPS, measuring UQ; Gneiting and Raftery (2007)); see the left panels of Figure 2. Lower is better for both (Supplementary Material B).

The Bayesian DGP is both more accurate and offers better UQ, but improved prediction is just the tip of an iceberg. Surrogate models are most often a means to another end; they are used for downstream tasks such as Bayesian optimization (BO; e.g., Jones, Schonlau and Welch (1998)), active learning (AL; e.g., Cohn (1994)), and calibration (e.g., Kennedy and O'Hagan (2001)). Here we are interested in contour location (CL), a common task in aeronautic reliability analysis (e.g., Stanford et al. (2022), Renganathan, Rao and Navon (2022, 2023)). In CL the surrogate's task is to identify a specific level set in the response surface, one that separates system "passes" from "failures." A surrogate must accurately locate failure regions in order to provide precise quantification of failure probabilities.

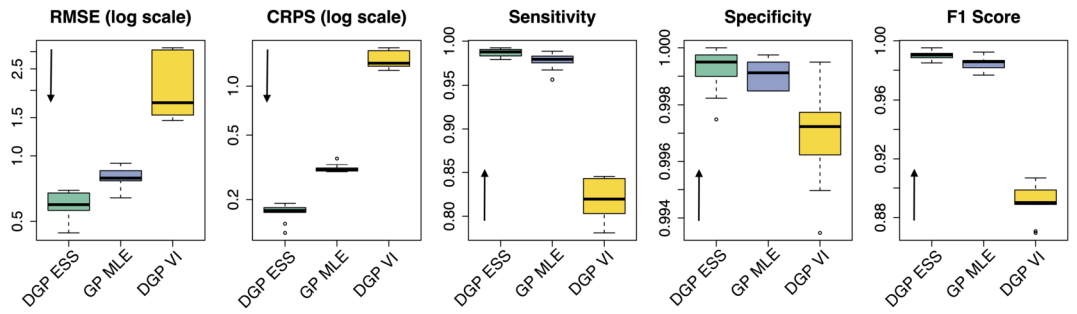

FIG. 2. *Out-of-sample prediction metrics for surrogate fits to static LHS designs of size* $n = 500$ *from the RAE-*2822 *simulation. Failures defined at* $L/D < 3$. *Arrows along* y*-axes indicate which direction is preferred. Boxplots show* 10 *MC repetitions.*

With our 7d airfoil simulator, we seek to identify efficient vs. inefficient ($L/D < 3$) flight conditions. The right panels of Figure 2 provide out-of-sample classification metrics measuring sensitivity, specificity, and F1 scores (Supplementary Material B) on surrogates fit to RAE-2822 simulations. Observe that the Bayesian DGP also performs well on these metrics (higher is better for all). Yet there is further scope for improvement; space-filling designs are, of course, inefficient for finding lower-dimensional manifolds like level sets. Furthermore, the computing time required to evaluate the computer model at these 500 points was nearly 20 hours. With a strategic design, we seek to obtain superior predictive performance with fewer evaluations of the expensive model.

1.2. *Sequential design for contour location.* We are interested in finding inputs that separate "pass" and "fail" regimes under a limited simulation budget. In this context it makes sense to build up the design sequentially to strategically hone in on this target, that is, CL—a form of AL. There are two keys to effective AL: (a) training a good surrogate with the data you already have and (b) using the surrogate to solve an acquisition criterion to determine the next run of the simulator which will be used to augment the data. These steps are repeated until the simulation budget is exhausted or until some performance metric is satisfied. The (Bayesian) DGP provides (a), but it is less amenable to (b) compared to the ordinary GP.

For CL, acquisition criteria target the contour through uncertainty in the pass/fail prediction, which is quantifiable through surrogate posterior predictions. The most popular criterion, based on entropy (Gray (2011)), converts classification probabilities into a metric which is maximized for 50/50 probabilities, where classification uncertainty is highest. But entropy and its variants have two downsides. One is that finding the input location with highest entropy requires a multistart numerical optimizer in order to circumvent disparate regions (local optima) of high entropy, separated by wide gulfs of numerically zero entropy. This difficulty is compounded when there are disconnected level sets separating regimes. MCMC-based inferential schemes make things more challenging still: the surrogate (i.e., a DGP) doesn't provide a single predictive surface but thousands, precluding library-based optimization. Second, and perhaps more importantly, the level set is a continuum, not a singleton as in BO and other AL variants. Although entropy does indeed target the contour, it is famously myopic (e.g., Cole et al. (2022)) as an acquisition function: it targets the most probable boundary crossing point, not exploration of the entire manifold. The result is clumped acquisitions.

These drawbacks have inspired a cottage industry of workarounds. One simple solution involves evaluating entropy on a discrete set of space-filling candidates, like an LHS. This is straightforward, and has some advantages over continuous numerics, including parallel evaluation and simple modular implementation. (Evaluating entropy on candidates is a post-processing step, whereas numerical optimization must be embedded within the surrogate.) Yet after an initial handful of acquisitions, most candidates reside in zero-entropy regions, resulting in essentially random selection and a coarse view of the target contour. Alternatively, some have adapted expected improvement ideas from BO to fit the CL setting, attempting to balance both exploration and exploitation (Ranjan, Bingham and Michailidis (2008), Bichon et al. (2008), Picheny et al. (2010)). Others have found success within stepwise uncertainty reduction (SUR) BO frameworks in which entropy (and other criteria) are numerically integrated over the input domain (Chevalier et al. (2014), Marques, Lam and Willcox (2018)). However, the requisite numerical quadrature can be difficult, making SUR-based strategies computationally daunting in moderate input dimension. Cole et al. (2022) suggested a hybrid-candidate-local numerical optimization to favor local optima in the entropy surface, a sensible but ultimately ad hoc enterprise: deliberately doing worse at one thing (solving for high entropy) to do better at another (avoiding myopia). Nevertheless, it out-performs the methods cited above in many exercises. But it requires derivatives, making it ill-suited to MCMC.

Here we explore an idea from the BO literature that strategically allocates candidates between existing design points through the use of Delaunay triangulation (Gramacy, Sauer and Wycoff (2022)). So-called "tricands" are space-filling in an adaptive way, guaranteeing that there are candidates in promising regions of the input space for acquisition. Optimization on surrogate predictive quantities, cumbersome with MCMC-based surrogates, is replaced with a geometric solver (Barber, Dobkin and Huhdanpaa (1996), Habel et al. (2019)) that doesn't consult the surrogate at all. As candidates, tricands enjoy parallel evaluation downstream with any surrogate, even MCMC-based ones. For BO they are superior to LHS candidates *and* numerically optimized acquisitions. For CL well-chosen candidates provide an opportunity to address multiple criteria simultaneously, for example, to correct the myopia in pure entropy-based acquisition. Toward that end, we propose acquisitions on the "Pareto front" of tricands evaluations of CL entropy and ordinary predictive uncertainty. This is both effective and enjoys a simple implementation as a postprocessing step on surrogate prediction. While similar to ideas of Bryan et al. (2005), who utilize the product of posterior variance and entropy, our Pareto acquisitions are agnostic to the relative scale of predictive variance.

1.3. *Roadmap*. Each ingredient in this scheme (DGP, tricands, Pareto front), separately offers a performance bump over its state-of-the-art analogue (GP, derivative-based acquisition, entropy-only). Over the course of this paper, we will offer intuitive explanations of these improvements and demonstrate superiority empirically with realistic benchmarking exercises. We acknowledge that these ingredients, taken separately, may not comprise a substantial novel contribution. But their application in the CL context is novel, both separately and together, and they have never been used in an airfoil reliability context. DGPs, whether VI- or MCMC-based, have never been entertained for CL, and both tricands and Pareto acquisitions are new to CL. To ease use downstream and support reproducibility, we provide a fully open source implementation as a supplement to this document and in public repositories.[2]

The remainder of this paper is laid out as follows. Section 2 provides a review of the surrogate-CL state-of-the-art: GPs and entropy-based acquisition. Section 3 details the main, modern ingredients—DGP surrogates, triangulation candidates, and Pareto front acquisition—and how they weave together to form our contribution. Section 4 offers implementation details and empirical results on synthetic examples. Section 5 returns to our motivating airfoil simulation. We offer a brief discussion in Section 6.

2. Stationary GP contour location. Here we review the state-of-the-art in GP-based entropy contour location. The more modern ingredients of our approach (DGPs, tricands, etc.) will be reviewed, as they are adapted to suit CL in Section 3.

2.1. *Gaussian process surrogates*. Let $f : \mathbb{R}^d \to \mathbb{R}$ denote a blackbox computer simulator; let X_n be an $n \times d$ matrix of simulation inputs, with rows $x_i^\top$, and let $y_n = f(X_n)$ represent an n-vector of outputs. A traditional GP prior assumes $y_n \sim \mathcal{N}_n(\mu, \Sigma(X_n))$, where

$$\Sigma(X_n)^{ij} = \Sigma(x_i, x_j) = \tau^2\left(k\left(\sum_{h=1}^{d}\frac{(x_{ih}-x_{jh})^2}{\theta_h}\right)+\eta\mathbb{I}_{i=j}\right). \tag{1}$$

While μ may be linear in columns of X_n, we specify $\mu = 0$ without loss of generality. Popular choices of kernel $k(\cdot)$ are the squared exponential or Matèrn (Stein (1999)), but any positive definite function will suffice. Hyperparameters τ^2, $\theta = [\theta_1, \ldots, \theta_d]$, and η govern

[2]Code to reproduce all examples and figures may be found in Supplementary Material (Booth, Renganathan and Gramacy (2025)) and at https://bitbucket.org/gramacylab/deepgp-ex/. DGP surrogate entropy calculations are provided as an update to the `deepgp` package on CRAN (Booth (2023)).

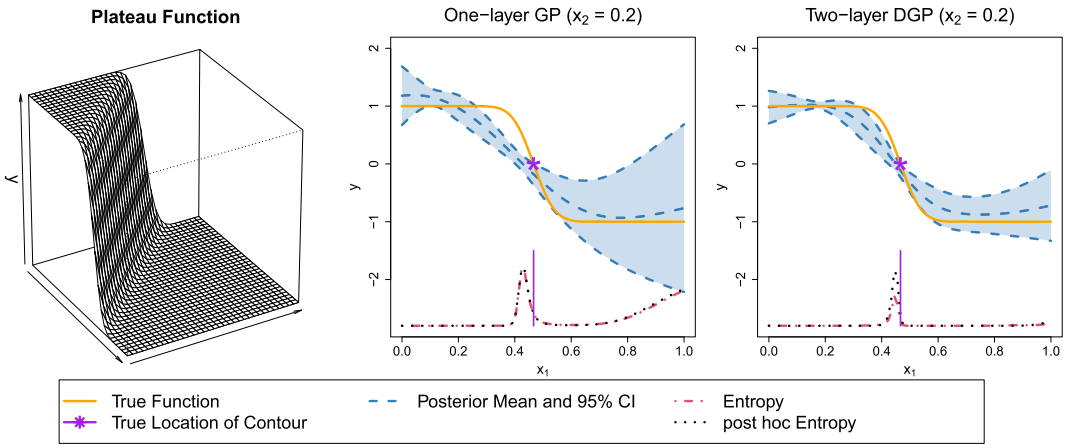

FIG. 3. Left: *Plateau function in* 2*d*. Middle and right: *Predictive distribution along a slice in* x_2 *with entropy calculations for an ordinary* (*left*) *and deep* (*right*) *GP.*

the scale, lengthscale, and nugget/noise, respectively. We focus here on deterministic computer simulations, fixing $\eta = 1 \times 10^{-6}$ throughout, to interpolate observations. Vectorized θ allows for coordinatewise anisotropic modeling. Fixing $\theta = \theta_1 = \cdots = \theta_d$ enforces isotropy. These unknown hyperparameters may be estimated through the likelihood by maximization or MCMC; see Santner, Williams and Notz (2018), Gramacy (2020) for review.

Given $\{X_n, y_n\}$, posterior predictions at an $n_p \times d$ matrix of unobserved inputs $\mathcal{X}$ follow

$$\mathcal{Y}|X_n, y_n \sim \mathcal{N}_{n_p}(\mu_Y, \Sigma_Y), \quad \text{where } \mu_Y = \Sigma(\mathcal{X}, X_n)\Sigma(X_n)^{-1} y_n, \\ \Sigma_Y = \Sigma(\mathcal{X}) - \Sigma(\mathcal{X}, X_n)\Sigma(X_n)^{-1}\Sigma(X_n, \mathcal{X}). \tag{2}$$

As a visual, consider the 2d "plateau" function displayed in the left panel of Figure 3 and featured later in Section 4.2. We trained a GP surrogate to data obtained from a uniformly random design of size $n = 15$; a 1d slice of the predictive surface (with fixed $x_2 = 0.2$) is displayed in the center panel. For now, focus on just the dashed lines/shading. The GP offers an adequate nonlinear fit to the surface, considering the sparsity of the training data in 2d, but it struggles to balance the disparity between the flat regions and the steep drop. This GP is restricted by the assumption of *stationarity*, a byproduct of $\Sigma(\cdot)$ being solely a function of relative distances. A stationary GP will struggle to balance steep drops with flat regions; it must compromise between oversmoothing the drop and undersmoothing the flats. A DGP, previewed in the right panel, is more flexible in handling this transition. Surrogate predicted mean and variance estimates are essential components of AL enterprises like CL.

2.2. *Entropy contour location.* The most popular acquisition criterion for CL is entropy (e.g., Oakley (2004), Marques, Lam and Willcox (2018), Cole et al. (2022)). Let g represent a threshold level such that $f(x) > g$ indicates failure. Denote $p_x = \mathbb{P}(f(x) > g)$ as the probability of observing a failure at location x. The entropy criterion

$$H(x) = -p_x \log(p_x) - (1 - p_x)\log(1 - p_x) \tag{3}$$

will be high in regions of pass/fail uncertainty (i.e., $p_x \approx 0.5$). Under a GP surrogate, failure probabilities may be evaluated through a Gaussian CDF Φ,

$$p_x = \mathbb{P}\big(f(x) > g\big) = 1 - \Phi\left(\frac{g - \mu_Y(x)}{\sigma_Y(x)}\right), \tag{4}$$

where $\mu_Y(x)$ and $\sigma_Y(x) = \sqrt{\Sigma_Y(x)}$ are provided in equation (2). Consider again the center panel of Figure 3. Define $g = 0$ such that $y > 0$ represents failure. The true location of the failure contour (where the true line crosses $g = 0$) is marked by the star and vertical line. The dotted/dashed lines along the x-axis show entropy $H(x)$ evaluated along a dense grid. Entropy is highest where the surrogate mean prediction crosses $g = 0$, but it is high in all areas where the 95% interval captures $g = 0$ as well. This is entropy's desirable property—it hones in on the failure contour. Yet as we acquire more points, entropy will become peakier and will still favor acquisitions around the vertical bar.

Notice, there is a local optima in the entropy surface at the right-most boundary where $x_1 = 1$. This arises primarily because the GP predictive equations indicate higher uncertainty at the edges of the input space. Many AL criteria are coupled with theory indicating that eventually such disconnected "high uncertainty" regions will recruit acquisitions because the criteria will eventually be maximized there, after targeted sampling elsewhere saturates. This is not true with entropy; it will never "in-fill." A consequence of this is that it will not explore the entire level set. Once it finds a highly probable crossing, like the one along the slice in the figure, it will only explore near its current best estimate. In the parlance of AL/BO, it is too exploitative, too myopic (Renganathan, Larson and Wild (2021)).

There are several suggested solutions in the literature. For example, Marques, Lam and Willcox (2018) integrate entropy over the input space to target the largest reduction in global entropy rather than simply the largest value. This is a good idea in principle, but unfortunately, the integral is not available in closed form, and the quadrature required is numerically fraught beyond 2d. This approach does not compare favorably to simpler, more ad hoc methods. For example, Cole et al. (2022) suggest deliberately coarsening the entropy search by only finding local rather than global optima of the acquisition surface to encourage exploration. This works well and represents the state-of-the-art as far as we are aware. However, it has the downside of requiring a tuning parameter (trading off local vs. global search) and derivative-based optimization, which is not amenable to MCMC-based surrogates.

3. Nonstationary DGP contour location. Here we describe our contribution: contour location (CL) with deep Gaussian process (DGP) surrogates, triangulation candidates, and Pareto front acquisitions.

3.1. *Deep Gaussian process surrogates.* DGPs are formed by layering stationary GPs. While this may be accomplished with kernel convolutions (Dunlop et al. (2018)), we prefer the lens of functional composition. A two-layer DGP prior is defined as

$$\begin{aligned} y_n | W &\sim \mathcal{N}_n(\mu_y, \Sigma(W)), \\ w_i &\overset{\text{ind}}{\sim} \mathcal{N}_n(\mu_w, \Sigma(X_n)) \quad \text{for } i = 1, \ldots, p, \end{aligned} \qquad \text{where } W = \begin{bmatrix} w_1 & w_2 & \ldots & w_p \end{bmatrix}, \tag{5}$$

and $\Sigma(\cdot)$ is provided in equation (1) with potentially unique hyperparameters. Each w_i is a conditionally independent stationary GP over the inputs X_n. Together these "nodes" form the latent W layer, which serves as a warped version of the inputs and feeds into an outer stationary GP connecting to the response y_n. The dimension of the latent layer need not match the dimension of the input space, but $p = d$ has been shown to work well (Sauer, Gramacy and Higdon (2023)). Prior means are adjustable, but common choices are $\mu_y = 0$ and either $\mu_w = 0$ or $\mu_w = X_n$ (the latter only being applicable when $p = d$). We have found settings where both choices are advantageous, but here we opt for the simplest option $\mu_y = \mu_w = 0$. Although additional GPs may be layered to create deeper models, such complexity has been shown to provide marginal, if any, benefit for surrogate modeling tasks (Radaideh and Kozlowski (2020), Sauer, Gramacy and Higdon (2023)) while requiring far more computation. We thus focus on the two-layer case (although we entertain a three-layer DGP in Figure 6).

Latent W drives nonstationary flexibility. Its nodes may "stretch" inputs in regions of high signal and "squish" inputs in regions of low complexity. Yet the functional and multidimensional nature of this unknown quantity poses an inferential challenge. One solution is to approximate the intractable DGP posterior with a "close" one (in terms of Kullback–Leibler divergence) from a known target family; this is known as approximate variational inference (VI; Blei, Kucukelbir and McAuliffe (2017)) and is popular in machine learning applications where signal-to-noise ratios are low and data are abundant (Damianou and Lawrence (2013), Salimbeni and Deisenroth (2017)). However, in our deterministic and data-sparse surrogate modeling setting, we opt for full posterior integration to prioritize UQ. We accomplish this using the Bayesian DGP set-up of Sauer, Gramacy and Higdon (2023). Latent W are sampled through elliptical slice sampling (ESS; Murray, Adams and MacKay (2010)). Kernel hyperparameters are also sampled through MCMC, although these are more of a fine-tuning, and their sampling may be replaced with MLE-based alternatives (e.g., Ming, Williamson and Guillas (2023)); see Sauer (2023) for a thorough review of Bayesian DGP surrogates.

After burn-in, ESS provides posterior samples $w_i^{(t)}$ for $i = 1, \ldots, p$ and $t \in \mathcal{T}$ (we shall drop reliance on kernel hyperparameters here, which may also be indexed by t). Samples in hand, prediction at inputs $\mathcal{X}$ requires first "warping" the predictive locations, $\mathcal{X}$ to $\mathcal{W}_i^{(t)}$, for each node (i) and each MCMC sample (t). Each component layer is conditionally Gaussian (5), so this means applying equation (2) with sampled $w_i^{(t)}$ in place of y_n. We follow Sauer, Gramacy and Higdon (2023) in mapping through the posterior predictive mean,

$$\mathcal{W}_i^{(t)} = \Sigma(\mathcal{X}, X_n)\Sigma(X_n)^{-1} w_i^{(t)} \tag{6}$$

instead of drawing from the full Gaussian posterior. Second, we column bind $\mathcal{W}_i^{(t)}$ into the $n_p \times p$ matrix $\mathcal{W}^{(t)}$ and again apply equation (2) to obtain posterior moments

$$\begin{aligned} \mu_Y^{(t)} &= \Sigma\big(\mathcal{W}^{(t)}, W^{(t)}\big)\Sigma\big(W^{(t)}\big)^{-1} y_n, \\ \Sigma_Y^{(t)} &= \Sigma\big(\mathcal{W}^{(t)}\big) - \Sigma\big(\mathcal{W}^{(t)}, W^{(t)}\big)\Sigma\big(W^{(t)}\big)^{-1}\Sigma\big(W^{(t)}, \mathcal{W}^{(t)}\big). \end{aligned} \tag{7}$$

We thus obtain $|\mathcal{T}|$ samples for each posterior moment (typically in the thousands). It is often beneficial to summarize these quantities using the law of total variance,

$$\mu_Y = \frac{1}{|\mathcal{T}|}\sum_{t\in\mathcal{T}} \mu_Y^{(t)} \quad \text{with } \Sigma_Y = \frac{1}{|\mathcal{T}|}\sum_{t\in\mathcal{T}} \Sigma_Y^{(t)} + \mathbb{C}\text{ov}(\mu_Y^{(t)}) \text{ and } \sigma_Y = \sqrt{\text{diag}(\Sigma_Y)}. \tag{8}$$

To provide a concrete visual, we fit a two-layer Bayesian DGP to the data from Figure 3. The right panel of that figure shows the DGP fit over the 1d slice. Compared to the GP (middle panel), the DGP is able to more accurately capture the plateau shape of the surface and the sloping drop between the two flat regions. The full 2d DGP mean surface is shown in the left panel of Figure 4, with training data locations indicated by open circles. The estimated failure contour is indicated by the dashed line; the true failure contour is overlayed for reference. The center panel displays a heat map of the predicted standard deviation, σ_Y from equation (8). Notice that uncertainty is high in areas near the transition and far from training data. The equivalent figure for the one-layer GP is relegated to Supplementary Material C since posterior uncertainty for a stationary GP is less interesting, being solely a function of distance to training data.

Although the full DGP posterior is no longer strictly Gaussian, its components are conditionally Gaussian, allowing us to utilize the failure probability calculation of equation (4) on component pieces to evaluate entropy for CL. Using equations (6)–(7) with $\mathcal{X} = x$ and with

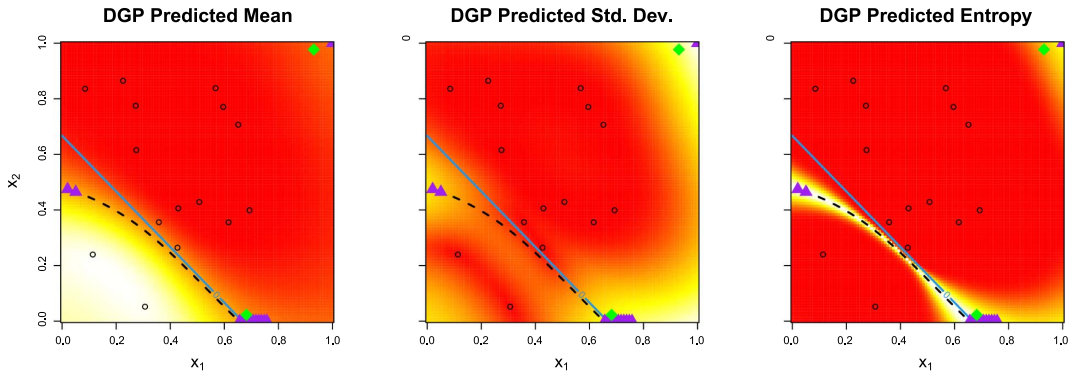

FIG. 4. *Two-layer DGP mean* (*left*), *standard deviation* (*center*), *and post hoc entropy* (*right*) *for the plateau function* (10): *light/high*, *dark/low. Open circles indicate training data. Triangles and diamonds represent Pareto front acquisitions* (*Section* 3.3) *under gridded candidates and "tricands"* (*Section* 3.2), *respectively.*

MCMC-based expectation over t, we obtain

$$H(x) = \frac{1}{|\mathcal{T}|}\sum_{t\in\mathcal{T}}[-p_x^{(t)}\log(p_x^{(t)}) - (1-p_x^{(t)})\log(1-p_x^{(t)})],$$
$$p_x^{(t)} = 1 - \Phi\left(\frac{g-\mu_Y^{(t)}(x)}{\sigma_Y^{(t)}(x)}\right). \tag{9}$$

It may be more expedient to plug the summarized moments μ_Y and Σ_Y from equation (8) directly into the Gaussian CDF of equation (4). We refer to this as post hoc entropy since it uses postprocessed DGP moments. The right panel of Figure 3 shows both entropy calculations along the 1d slice as dotted/dashed lines. They are equivalent for the GP since there was no warping at play. For the DGP the shapes of the curves are very similar. Absolute magnitudes do not feature in our acquisitions; entropy values are only considered relative to each other. Both acquisition surfaces identify a single high-entropy region where the predicted surface crosses the failure boundary. We have noticed this in all of our experiments (not shown here), and so for simplicity and ease-of-use (the post hoc calculation is more "plug-and-play"), we utilize post hoc entropy from here on out. However, our software supports both options.

The right panel of Figure 4 displays a heat map of the post hoc entropy surface from the two-layer DGP fit to the plateau function. Notice how it is very peaky, with a ridge of maxima along the entire predicted contour. For now, ignore the triangles and diamonds, which are discussed later in Section 3.3. Suppose we sought a new acquisition utilizing the entropy criterion, based on this DGP surrogate. There are two clear obstacles. First, finding optima in the entropy surface comes with a hefty computational price-tag. The ridge highlighted in Figure 4 was found by evaluating the DGP surrogate on a dense grid, a method that is infeasible in higher dimensions. Numerical optimizers, on the other hand, require many successive evaluations of singleton predictive locations. In a DGP each predictive location must be mapped through the latent layer (6), then to posterior moments (7), with this entire process repeated $|\mathcal{T}|$-many times before final aggregate predictive mean and variances are available (8). Doing this evaluation one-at-a-time (without closed-form gradients) is incredibly inefficient. A common workaround in AL (e.g., Gramacy and Lee (2009)) and BO (e.g., Eriksson et al. (2019)) involves deploying a limited, discrete set of candidates. With candidates the loop over $t \in \mathcal{T}$ need only be done once. Although simple to implement, the fidelity of such candidate searches is low in situations where the goal is to identify a small target (e.g., a contour) in a big input space. Space-filling candidates, like LHS's, are likely to miss peaky areas of high entropy, unless the size of the candidate set is prohibitively large.

The second obstacle involves the entropy criterion itself; it is too myopic for CL (see, e.g., Marques, Lam and Willcox (2018), Cole et al. (2022)). Even if we are able to find optima in the entropy surface, this criterion only minimally accounts for predictive uncertainty. In Figure 4 a portion of the high entropy ridge is actually very close to existing training data locations. Even though uncertainty is low here, entropy is still high. Acquisitions near the outer edge of the predicted contour, where uncertainty is higher, or in the upper-right corner, where there is no training data, would be better. Although there is a local maximum in the entropy surface in the upper-right corner, this local maximum will never overcome the global maximum near the predicted contour. This is why the hybrid-local-optimization scheme of Cole et al. (2022), which successfully identifies local optima in the entropy surface, outperforms globally-optimized competitors. But further mitigation of entropy's limitations is warranted.

3.2. *Triangulation candidates.* Here we address the first issue, and to a lesser extent the second (which is more squarely the subject of Section 3.3), by adapting a new candidate scheme from the BO literature: triangulation candidates (Gramacy, Sauer and Wycoff (2022)), or "tricands." Rather than allocating candidates at-random or in a space-filling scheme, we allocate candidates strategically by "in-filling" the gaps in the existing design X_n. Start with a *Delaunay triangulation* of X_n: a set of line segments between points that divvy up the space so no lines cross. The left panel of Figure 5 provides a visual of this in 2d (for the training data from the 2d plateau function example). The other panels will be narrated in Section 3.3. Open circles indicate observed training data locations, and solid lines form the Delaunay triangulation. In two dimensions this triangulation forms triangles; in higher dimension it forms tetrahedra, although we will still refer to them as triangles. Each triangle is defined by a $(d+1) \times d$ matrix. Denote each triangle by T_j for $j = 1, \ldots, n_T$. The number of triangles (n_T) is affected by the size of the training data (n) and the dimension of the input space (d).

Tricands are allocated based on the triangulation in two realms: internal and fringe. Internal candidates are located at the *barycenter* of each triangle: $\tilde{x}_j = \frac{1}{d+1} \sum_{i=1}^{d+1} T_j[i, \ ]$. Notice in Figure 5 how they spread out between existing design points. For fringe candidates, denote F_j for $j = 1, \ldots, n_F$ as the $d \times d$ matrix representing a facet (outer edge of the triangulation). Calculate the midpoint of each facet as $\bar{F}_j = \frac{1}{d} \sum_{i=1}^{d} F_j[i, \ ]$ for $j = 1, \ldots, n_F$. Then take the normal vector $\vec{v}_j$ extending perpendicularly from F_j at the midpoint $\bar{F}_j$ (dashed arrows in Figure 5, left). Allocate a fringe candidate along each $\vec{v}_j$, with the exact placement determined by a tuning parameter $\alpha \in [0, 1]$ representing the proportional distance from the

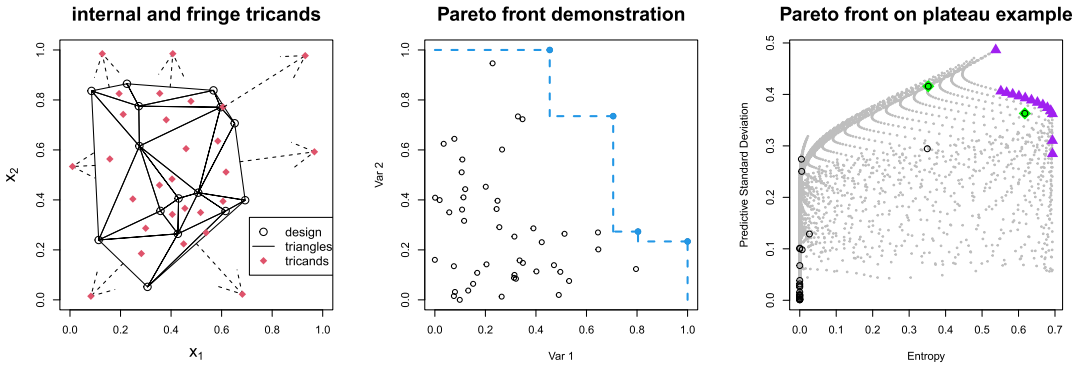

FIG. 5. *Left*: *Delaunay triangulation with internal and fringe* ($\alpha = 0.9$) *"tricands." Middle*: *A Pareto front set* (*dashed lines*) *for two hypothetical variables. Right*: *Entropy vs. standard deviation for the DGP of Figure 4 two ways*: *on a grid in x-space* (*dots*), *with the Pareto front as triangles*; *via "tricands"* (*open circles*) *with Pareto front as diamonds.*

facet midpoint to the boundary of the space. For BO, Gramacy, Sauer and Wycoff (2022) found $\alpha = 0.5$ worked well. For CL we choose $\alpha = 0.9$ to encourage acquisitions near the boundary—contours are often found near boundaries, especially in reliability applications. Fringe candidates are represented by the tips of dashed arrows in the figure. Crucially, tricands (both internal and fringe) are agnostic to surrogate models; they rely only on the geometry depicted by existing training locations, spreading out to encourage exploration while still offering a higher concentration of candidates in more densely observed areas to allow for exploitation.

Denote the combined set of internal and fringe tricands as $\mathcal{X}_N$, where $N = n_T + n_F$. As input dimension (d) and data size (n) increase, the total number of tricands will grow. When N is large, it is helpful to limit the computational burden by setting a maximum size of the candidate set, $N_{\max} \ll N$. While a rudimentary approach would simply subsample $N_{\max}$ points from all N tricands, we find it helpful to employ a targeted subsampling approach to guarantee retention of some exploitative candidates near the failure contour. This approach is inspired by the BO-based subsampling of Gramacy, Sauer and Wycoff (2022), but it requires modification to suit the CL setting. Rather than retaining candidates around a *single* best-observed point, as in BO, we seek to retain candidates around the entire contour. Since the true contour is unknown, we leverage the absolute difference between the observed responses and the limit state ($|y_i - g|$) to rank the observations by closeness to the contour. Then, starting with the observation closest to g, we identify all triangles T_j and/or facets F_j that were generated with this point and randomly select one of their candidates to retain. We repeat this process, working down the ordering and retaining one candidate *adjacent* to each location, until we have collected at least 10% of $N_{\max}$. Then we select remaining candidates, filling out $N_{\max}$, completely at random. By iteratively working down the ordering of closeness to the contour, we are able to target the entire contour set. As with ordinary tricands, this targeted subsampling does not rely on any surrogate information.

3.3. *Pareto front acquisitions*. Tricands circumvent a continuous, derivative-based optimization (which is troublesome with MCMC-based surrogates) by replacing it with a geometric calculation. They also, to a degree, encourage exploration of the acquisition space by placing candidates where predictive uncertainty is likely to be high. However, we have found that this is not enough to correct the myopia of entropy. Toward that end, we propose an acquisition criterion that strikes a deliberate balance between entropy (exploitation) and predictive uncertainty (exploration). Although approaches similar to the ones we describe have been suggested in other AL contexts, we are not aware of any such attempts for CL, or with tricands, or similar candidate-based schemes.

Consider entropy $H(x)$ and predictive standard deviation $\sigma_Y(x)$ evaluated over $x \in \mathcal{X}$ in the input space. When using either as a lone acquisition criterion, higher is better. Now, let $c(x) = (H(x), \sigma_Y(x)) \in \mathbb{R}^2$ represent a two-dimensional "criterion," and write $c(x) \prec c(x')$, for $x' \in \mathcal{X}$, if $c(x')$ *strictly dominates* $c(x)$, meaning that it is better in both dimensions simultaneously: $H(x) < H(x')$ and $\sigma_Y(x) < \sigma_Y(x')$. The *Pareto frontier*, defined as $P(\mathcal{X}) = \{x \in \mathcal{X} : \{x' \in \mathcal{X} : c(x) \prec c(x'),\ x \neq x'\} = \varnothing\}$, is the subset of the input space $\mathcal{X}$ separating dominated from nondominated points. The definition is similar for higher-dimensional criteria; for more details, see Goodarzi, Ziaei and Hosseinipour ((2014), Chapter 4).[3]

The Pareto frontier is a compact set, and in surrogate modeling/AL contexts, it is possible to estimate and quantify uncertainty around it (e.g., Binois, Ginsbourger and Roustant (2015),

[3]Note that most textbook definitions of Pareto fronts/nondominated sets do not involve an input, x-variable, working instead on generic c-criterion values. However, we have introduced x to make an explicit link to criteria evaluated over an input space $\mathcal{X}$. To connect to classical definitions, one could instead write $c(P(\mathcal{X}))$.

Luo et al. (2018)) for arbitrary $x \in \mathcal{X}$. When restricting to a discrete subset of the input space, like to tricands $\mathcal{X}_N$, the Pareto front $P(\mathcal{X}_N)$ is also discrete and is known as the *nondominated set*. For example, consider random samples on $c(x) \in \mathbb{R}^2$, displayed as `Var1` and `Var2` in the middle panel of Figure 5. We generated these directly in c-space, without generating x's first, for a simple illustration, but we shall return to x-space momentarily. The filled circles connected by the dashed staircase indicate the points on the Pareto front, P (assuming larger is preferred). These are not beaten by any other point with respect to both variables: other points may be higher on one or the other, but not both.

Identifying the Pareto front from a discrete sample in 2d is straightforward: (a) order the observations from highest to lowest for one variable (points that yield the maximum of either variable are always in the Pareto front set), and (b) work down the ordering one-by-one, adding points to P only if they are higher on the second variable than all preceding points (which were higher on the first variable). Our implementation is five to six lines of R code, available in our Git repository. However we note that the `rPref` R-package on CRAN (Roocks (2016)) offers faster computations for larger datasets in higher dimensions.

For a particular example in our CL setting, let us return to the plateau surface of Figure 4, which involves two input dimensions, and now two acquisition criteria $c(x) = (H(x), \sigma_Y(x))$. The right panel of Figure 5 combines the visuals from the center and right panels of Figure 4 by plotting H vs. σ_Y, evaluated over a dense grid of locations (dots), and also over tricands $\mathcal{X}_N$ (open circles). An ideal acquisition would be in the upper right corner of this plot, with both high entropy and high uncertainty. In both Figures 4 and 5, candidates along the Pareto front are highlighted by triangles for the candidate grid and diamonds for tricands. Figure 4 is providing the x-location of the Pareto front(s), whereas Figure 5 shows the corresponding $c(x)$-value(s).

Observe in the figure(s) that tricands provide similar information at a fraction of the cost compared to a dense grid. Any of the points on either Pareto front (diamonds or triangles) represents a sensible acquisition. Observe in Figure 4 that all of these choices are far from the training data X_n, thus correcting the mypoia of entropy. They are either very close to the estimated contour, but away from X_n, or if they are not near the contour, they are *very* far from X_n. We prefer to select an acquisition completely at random from $P(\mathcal{X}_N)$. Going forward, we use only tricands because grids that are dense enough for good resolution become prohibitively large in higher input dimension.

It is worth noting that tricands have an extra advantage over grids—in x-space, the members of $P(\mathcal{X}_N)$ (diamonds) are located far from the training design X_N *and* far from one another by virtue of their geometric nature. This is not true for Pareto fronts constructed from a dense grid. This means a tricands–Pareto set could be used to select a batch of new runs at once (Cole et al. (2022)). We provide further discussion of this idea in Section 6.

To summarize, our candidate-based contour location scheme boils down to three steps: (1) build tricands $\mathcal{X}_N$ from X_n; (2) fit a DGP surrogate to evaluate posterior predictive standard deviation $\sigma_Y(\mathcal{X}_N)$ and predictive entropy $H(\mathcal{X}_N)$ for these candidates; (3) select an acquisition uniformly at random on the Pareto front of uncertainty and entropy: $x_{n+1} \sim \text{Unif}[P(\mathcal{X}_N)]$. Supplementary Material D provides a summary of this algorithm. While this Pareto–tricands CL scheme is technically independent of surrogate modeling choices and may be deployed with any surrogate—and also any multidimensional acquisition criteria—we find DGPs are crucial for the nonstationary response surfaces that motivate this work. Empirical evidence is presented in Sections 4–5.

4. Implementation and benchmarking. In this section we describe the publicly available implementation of our method, then validate it on a variety of synthetic test problems.

4.1. *Implementation details*. Code to reproduce all results (Sections 4.2–5) is provided in our public Git repository.[4] We utilize the `deepgp` package for R on CRAN (Booth (2023)) for Bayesian DGP surrogate training and prediction. That package was recently updated to provide entropy calculations following equation (9). Even though full entropy calculation is offered in the package, we opt for the post hoc calculations using summarized posterior moments. All experiments utilize package defaults aside from fixing the noise parameter $\eta = 1 \times 10^{-6}$ for interpolation of deterministic computer models.

We use the `geometry` (Habel et al. (2019)) R-package for Delaunay triangulation, which is based on a legacy C library called `quickhull` (Barber, Dobkin and Huhdanpaa (1996)). We offer convenient R and python wrappers, which include calculation of internal and fringe "tricands" with the targeted subsampling approach described in Section 3.2. Identification of the Pareto front requires a simple `for` loop over the ordered observations and an `if` statement to check values of the second variable. Our main performance metric is sensitivity (Supplementary Material B) on a hold-out LHS testing set over iterations of acquisition or for a single space-filling LHS of fixed size. Higher sensitivity is better, with 1.0 indicating detection of all failures. Discussion of computation times is reserved for Supplementary Material E.

We entertain the following surrogates as competitors:

- DGP ESS: Two-layer Bayesian ESS-based DGP of Sauer, Gramacy and Higdon (2023) using `deepgp` in R;
- GP MCMC: Stationary GP with MCMC-sampled separable lengthscales, also via `deepgp`
- GP MLE: Stationary GP with maximum likelihood estimated separable lengthscales using `scikit-learn` in Python (Pedregosa et al. (2011));
- DGP VI: Doubly stochastic VI DGP of Salimbeni and Deisenroth (2017) using `gpytorch` in Python (Gardner et al. (2018)).

All utilize the Matèrn $\nu = 5/2$ kernel, and all training data are observed without noise. For each model we compare a static space-filling LHS to a strategic CL sequential design (when feasible). We deploy our tricands-Pareto-front acquisition scheme with the MCMC-based surrogates (DGP ESS pareto and GP MCMC pareto). Since these two differ only in the choice of surrogate (two-layer DGP vs. stationary GP), they offer a glimpse into the direct impact of nonstationary flexibility on CL performance. We deploy the hybrid entropy-based numerical optimization scheme of Cole et al. (2022) with the MLE-based GP (GP MLE hyb ent), which is the state-of-the-art at the time of publication and thus our main competitor.

The DGP VI surrogate employs approximate VI through a stochastic optimization of the evidence lower bound. This optimization entails several tuning parameters, including the learning rate, number of optimization steps (epochs), and the number of Gaussian mixtures in the target posterior. There exist automated tuning methods involving inner-optimizations (e.g., Zimmer, Lindauer and Hutter (2021)); however, these approaches incur a significant computational expense. In the context of sequential design for CL, having an internal sequential design (i.e., a BO) to fine-tune optimization parameters seems to us like a rabbit hole. Rather, we fix these values at what we consider to be good defaults (mimicking the other surrogates which all use package defaults): 1×10^{-3} learning rate, 2500 epochs, and 32 Gaussian mixtures. We include the DGP VI competitor as an alternative DGP benchmark but admit this comparison is tangential to our main objective. In light of this and DGP VI's poor performance, we only entertained it in the static settings.

[4] https://bitbucket.org/gramacylab/deepgp-ex/.

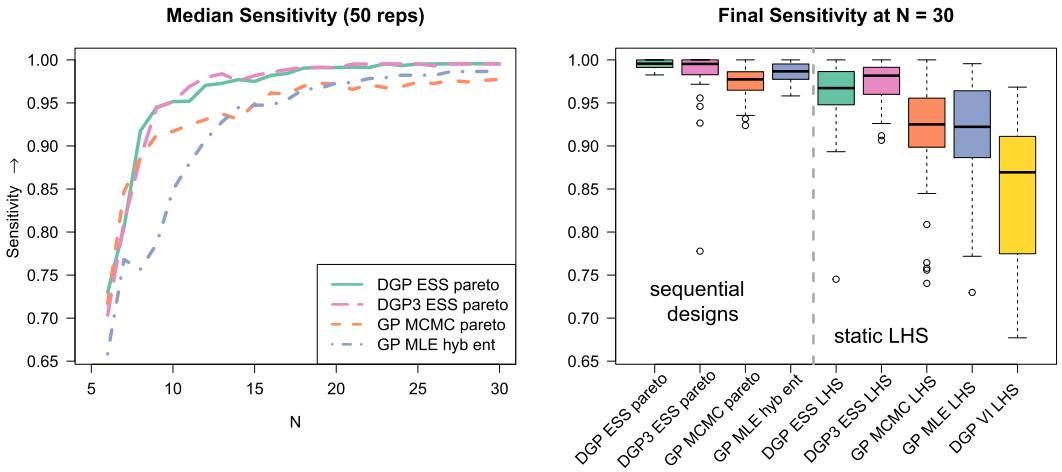

FIG. 6. *Sensitivity (higher is better) for the 2d plateau function.*

4.2. *Synthetic examples.*

4.2.1. *Plateau function.*

Consider the following function,

$$f(x) = 2 \cdot \Phi\left(\sqrt{2}\left(-4 - 3\sum_{i=1}^{d} x_i\right)\right) - 1 \quad \text{for } x \in [-2, 2]^d \text{ (scaled to } x \in [0, 1]^d\text{)}, \tag{10}$$

which we have adapted from Izzaturrahman et al. (2022) to be defined in arbitrary dimension. This function formed the basis of our earlier illustrations in Figures 3–5. We define the contour at $g = 0$ such that $f(x) > 0$ indicates a failure. Starting with $d = 2$ and an initial LHS design of size $n_0 = 5$, we conduct sequential designs targeting this contour up to an ending design size of $n = 30$ (25 acquisitions). Median sensitivity across 50 Monte Carlo (MC) repetitions with re-randomized starting designs is displayed in the left panel of Figure 6. Sensitivities after the last acquisition are displayed in the left interior panel. As an additional benchmark, we fit each surrogate on static LHS designs of equivalent size ($n = 30$) and reported the sensitivities in the right interior panel. For this example we also entertained a three-layer DGP (DGP3) using `deepgp` with package defaults.

Although the DGP surrogate starts at a similar place as the MCMC-based GP, it quickly jumps into the top position and maintains this superiority for the extent of the design. Median performance of the three-layer DGP matched that of the two-layer, but the additional flexibility of the deeper model led to poorer worst-case performance (which we attribute to over-fitting). The three-layer DGP required twice the computing time of the two-layer (Supplementary Material E), prompting us to drop it from further consideration. All methods benefited from CL acquisition when compared to their static LHS counterparts. The DGP VI model struggled to cope with this small-data setting; it was designed for much larger problems.

Next, we bumped the dimension up to $d = 5$ and repeated this exercise, starting with $n_0 = 20$ and acquiring up to $n = 150$. Results are shown in Figure 7. Performance trends are similar to those of Figure 6, although it appears the DGP has an even bigger edge in this higher-dimensional setting. Also noteworthy, entropy-based acquisitions behind "GP MLE hyb ent" occasionally led the GP astray and resulted in poorer performance than their static design counterparts, a byproduct of clustered acquisitions. DGP VI LHS performs better, comparatively, than in the 2d setting. We attribute this to the larger training data size.

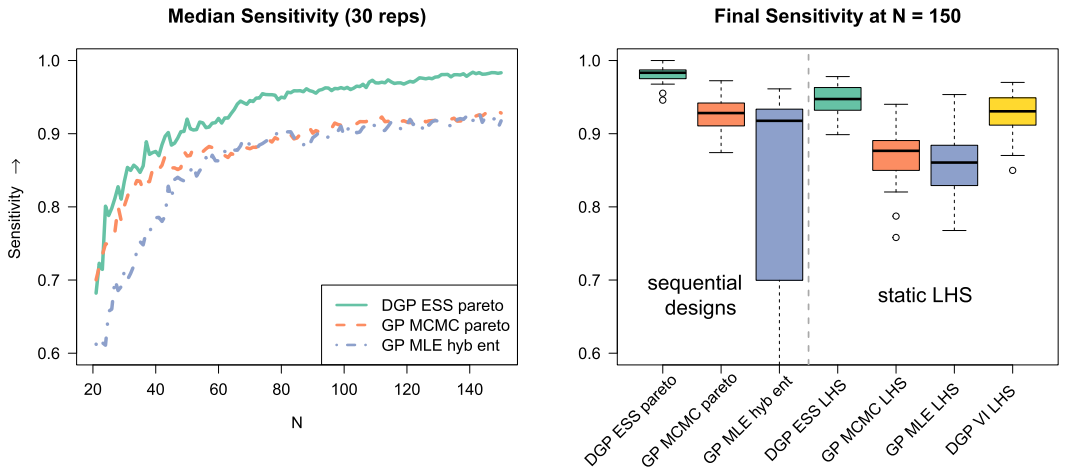

FIG. 7. *Sensitivity for the 5d plateau function.*

4.2.2. *Cross-in-tray function.* Next, consider the "cross-in-tray" function,

$$f(x_1, x_2) = -0.001\left(\left|\sin(x_1)\sin(x_2)\exp\left(\left|100 - \frac{\sqrt{x_1^2 + x_2^2}}{\pi}\right|\right)\right| + 1\right)^{0.1}$$

found on the pages of the Virtual Library of Simulation Experiments (Surjanovic and Bingham (2013)). We use the domain $x \in [-2, 2]^2$ (similarly to $[0, 1]^2$). A visual of the surface is provided in Figure 8 (left) with a contour defined at $g = 2$. We follow the same sequential design procedures, but we increase training data sizes to $n_0 = 50$ with a final design size of $n = 300$ to account for the increased complexity in the surface and the contour. Results over 20 MC repetitions are displayed in Figure 9. Again, all methods benefit from sequential design, but the DGP has a clear advantage due to its nonstationary flexibility. The poor performance of DGP VI LHS here is intriguing; perhaps it could be improved with fine-tuning. But we suspect that the built-in inducing point approximations (Snelson and Ghahramani (2006)), which have been shown to hinder approximation fidelity (Sauer, Cooper and Gramacy (2023a)), are restricting the model's ability to handle the finer scale of this failure region.

In addition, the center and right panels of Figure 8 display the design at the end of a single MC exercise for the "DGP ESS pareto" and the "GP MLE hyb ent" schemes. Circles

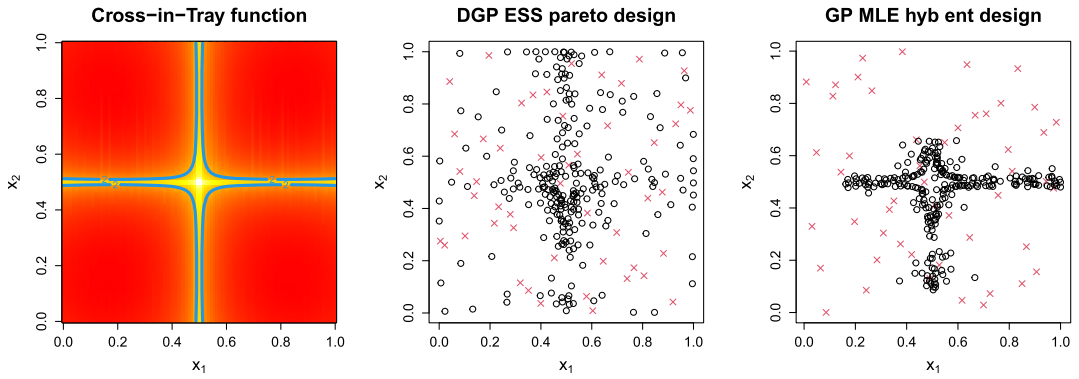

FIG. 8. *Left*: *Heat map of 2d cross-in-tray function with contour at $g = 2$. Center*: *Sequential design from two-layer DGP with Pareto–tricands acquisitions. Right*: *Sequential design from stationary GP with hybrid–entropy acquisitions. Circles indicate acquired points*; *"x" indicate starting LHS.*

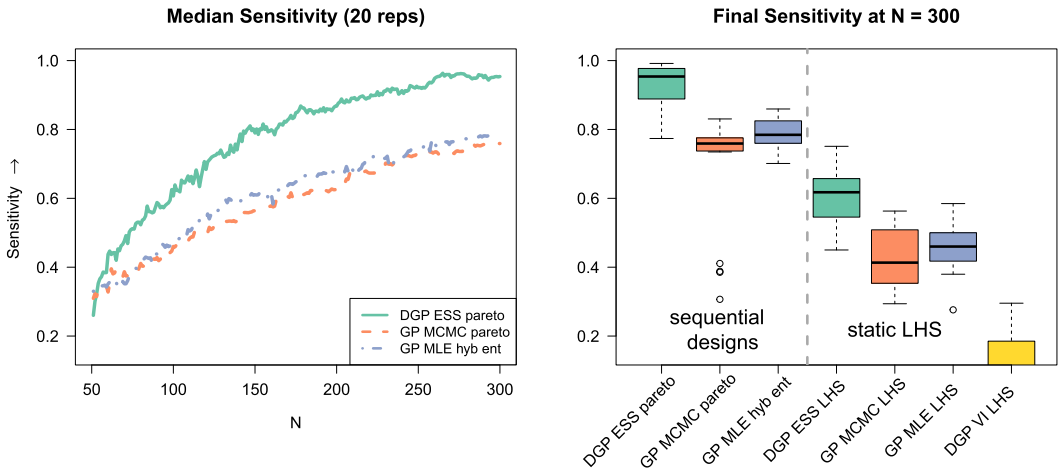

FIG. 9. *Sensitivity for the 2d cross-in-tray function.*

indicate acquired points, and "x" indicate the starting LHS design. As expected, acquisitions that rely solely on entropy tend to cluster (right panel). Pareto front acquisitions do a better job of exploring the contour. The real "secret sauce" though is the combination of the Pareto–tricands acquisition scheme with the flexible DGP model.

5. RAE-2822 airfoil contour location. In Section 1.1 we previewed surrogate predictive prowess on space-filling LHS designs of size $n = 500$. Results were shown earlier in Figure 2, where we observed that the Bayesian DGP indeed outperformed the stationary GP. The performance of DGP VI was underwhelming; again, we suspect fine-tuning may help, but the main culprit is a blurry inducing point approximation.

To improve performance with fewer simulations, we deployed our Pareto–tricands AL scheme for CL with the DGP ESS surrogate. We initialized with a 100-point LHS and acquired 400 points for a resulting design of size $n = 500$. Sensitivity, specificity, and F1 scores (Supplementary Material B, higher is better) for 10 MC repetitions (with re-randomized initial and testing LHS sets) are shown in Figure 10. The left interior panels show progress across the sequential designs, and the right interior panels show the results from static designs of equivalent size (copied from Figure 2 for reference). DGP-tricands-Pareto sequential designs outperformed static fits with as few as 200 evaluations of the simulator, a savings of nearly 12 hours of computing time compared to the 500-point LHS.

To help explain/visualize how the DGP ESS Pareto–tricands scheme is gaining an advantage, Figure 11 shows one of the resulting sequential designs, visualized as a projection over

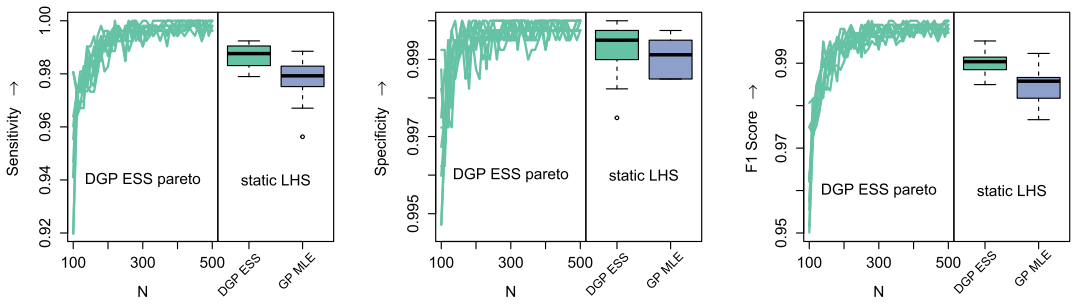

FIG. 10. *Sensitivity, specificity, and F1 score for "DGP ESS pareto" sequential designs of the 7d airfoil simulation (left interior panels). Right interior panels show results from static LHS designs (DGP VI omitted).*

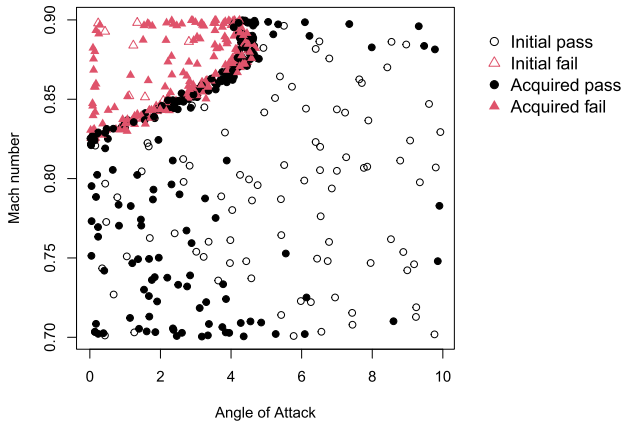

FIG. 11. *2d projection of a "DGP ESS pareto" sequential design* ($n = 500$). *Failures at* $L/D < 3$.

the two most impactful inputs: angle of attack and Mach number. Pareto–tricands acquisitions (filled circles and triangles) occasionally explore but overwhelmingly focus on low angles of attack. All failures occurred at low angles of attack and high speeds. This makes sense, as this is where lift is generally low. Acquisitions were often placed near this boundary in the upper left corner, an indication that the AL procedure honed in on the contour.

6. Discussion. We proposed a sequential design scheme for contour location with DGPs that relies on triangulation candidates and selects acquisitions on the Pareto front of entropy and posterior predictive uncertainty. Our scheme circumvents cumbersome numerical optimizations, avoids clustered acquisitions, and allows for CL with Bayesian DGPs (which has not been done before). While we were motivated by the application of DGPs in aeronautic simulations, our sequential design scheme is not limited to these cases. Pareto–tricands sequential designs may be useful for other hefty surrogate models which are incompatible with numerical optimization of acquisition functions, such as an MCMC-based treed GP (Gramacy and Lee (2008)). We believe that the Pareto front acquisition procedure is applicable to any combination of multiple criteria. One advantage over other hybrid schemes, such as those based on aggregation, is that it is independent of the relative scales of the component criteria. Combining any criteria with predictive uncertainty and selecting acquisitions on the Pareto front has potential to encourage exploration while retaining exploitative behavior.

Our DGP-Pareto-tricands approach excels in settings where the failure contour itself exhibits complex or nonstationary behavior. Our synthetic examples were designed with this aim in mind: the contour of the plateau function is near the steep drop between regimes, and the contour of the cross-in-tray function surrounds the nonstationary peaks of the surface. If the response surface is stationary, then the complexity of a DGP is not warranted and could even lead to over-fitting. GP diagnostics (Bastos and O'Hagan (2009)) may help identify these situations. Furthermore, if the failure region is very small and/or the contour itself is rather simple, a stationary GP may suffice, even if there is nonstationarity away from the contour. In these scenarios the Pareto–tricands acquisitions designed to promote exploration may underperform simpler alternatives, like the hybrid entropy optimization of Cole et al. (2022).

While tricands proved useful in our modest dimensional cases, the computational costs of the Delaunay triangulation may become prohibitive in higher dimensions. In our experience, eight dimensions is pushing the boundary of tractability for the underlying `quickhull` library, unless the design size n is kept low. There are some opportunities for improvement, such as only updating the portion of the triangulation near a newly acquired point. But we

suspect that new methodology will be needed to replace the Delaunay triangulation in higher dimension while still mimicking its behavior. One advantage to tricands/Pareto acquisition, which was not explored in this paper, involves batch acquisition. We observed in Section 3.3 that it would be easy to take a batch whose size matched $|P(\mathcal{X})|$, the size of the nondominated set. We note that one could always get more, to augment the batch, by removing those points and recalculating the Pareto front. Exploring the extent to which this is a productive use of resources would be an interesting subject of future research.

Reliability analysis involves quantifying the probability of failure, given uncertain inputs (e.g., Allen and Maute (2004)). Our goal has been to train a surrogate to accurately identify an entire failure contour. The surrogate can then be utilized with any probability distribution over the input variables to estimate probabilities of failure. Nevertheless, it may be advantageous to incorporate information about a given input distribution into the sequential design procedure, so the surrogate can prioritize learning around certain regions of the contour. Abdelmalek-Lomenech et al. (2022) have shown such methods to be advantageous with GPs, albeit at great computational expense. We suspect similar strategies will work with DGPs, as long as the computational burden can be kept in check.

Acknowledgments. SAR acknowledges the allocation on Pennsylvania State University's Institute for Computational and Data Sciences' Roar supercomputer to run the simulator experiments.

Funding. RBG acknowledges partial support from NSF 2318861 and 2152679.

SUPPLEMENTARY MATERIAL

Supplement to "Contour location for reliability in airfoil simulation experiments using deep Gaussian processes" (DOI: 10.1214/24-AOAS1951SUPPA; .pdf). *A*: List of all acronyms used in the manuscript. *B*: Definitions of performance metrics (sensitivity, specificity, F1 score, RMSE, and CRPS). *C*: Typical one-layer stationary GP fit to the 2d plateau function. *D*: Summary of the proposed sequential design scheme. *E*: Compute times for the synthetic examples of Section 4.2.

Zipped code files (DOI: 10.1214/24-AOAS1951SUPPB; .zip). Code to reproduce all results and figures.

REFERENCES

ABDELMALEK-LOMENECH, R. A., BECT, J., CHABRIDON, V. and VAZQUEZ, E. (2022). Bayesian sequential design of computer experiments to estimate reliable sets. Available at arXiv:2211.01008.

ALLEN, M. and MAUTE, K. (2004). Reliability-based design optimization of aeroelastic structures. *Struct. Multidiscip. Optim.* **27** 228–242.

BARBER, C. B., DOBKIN, D. P. and HUHDANPAA, H. (1996). The quickhull algorithm for convex hulls. *ACM Trans. Math. Software* **22** 469–483. MR1428265 https://doi.org/10.1145/235815.235821

BASTOS, L. S. and O'HAGAN, A. (2009). Diagnostics for Gaussian process emulators. *Technometrics* **51** 425–438. MR2756478 https://doi.org/10.1198/TECH.2009.08019

BICHON, B. J., ELDRED, M. S., SWILER, L. P., MAHADEVAN, S. and MCFARLAND, J. M. (2008). Efficient global reliability analysis for nonlinear implicit performance functions. *AIAA J.* **46** 2459–2468.

BINOIS, M., GINSBOURGER, D. and ROUSTANT, O. (2015). Quantifying uncertainty on Pareto fronts with Gaussian process conditional simulations. *European J. Oper. Res.* **243** 386–394. MR3315549 https://doi.org/10.1016/j.ejor.2014.07.032

BITZER, M., MEISTER, M. and ZIMMER, C. (2023). Hierarchical-hyperplane kernels for actively earning Gaussian process models of nonstationary systems. In *International Conference on Artificial Intelligence and Statistics* 7897–7912. PMLR, Valencia, Spain.

BLEI, D. M., KUCUKELBIR, A. and MCAULIFFE, J. D. (2017). Variational inference: A review for statisticians. *J. Amer. Statist. Assoc.* **112** 859–877. MR3671776 https://doi.org/10.1080/01621459.2017.1285773

BOOTH, A. S. (2023). deepgp: Deep Gaussian processes using MCMC. R package version 1.1.2.

BOOTH, A. S., RENGANATHAN, S. A. and GRAMACY, R. B. (2025). Supplement to "Contour location for reliability in airfoil simulation experiments using deep Gaussian processes." https://doi.org/10.1214/24-AOAS1951SUPPA, https://doi.org/10.1214/24-AOAS1951SUPPB

BREMER, S. A. (2019). *The GLOBUS Model*: *Computer Simulation of Worldwide Political and Economic Developments*. Routledge, London.

BRYAN, B., NICHOL, R. C., GENOVESE, C. R., SCHNEIDER, J., MILLER, C. J. and WASSERMAN, L. (2005). Active learning for identifying function threshold boundaries. *Adv. Neural Inf. Process. Syst.* **18**.

BUI, T., HERNÁNDEZ-LOBATO, D., HERNANDEZ-LOBATO, J., LI, Y. and TURNER, R. (2016). Deep Gaussian processes for regression using approximate expectation propagation. In *International Conference on Machine Learning* 1472–1481. PMLR, New York, USA.

CHEVALIER, C., GINSBOURGER, D., BECT, J., VAZQUEZ, E., PICHENY, V. and RICHET, Y. (2014). Fast parallel kriging-based stepwise uncertainty reduction with application to the identification of an excursion set. *Technometrics* **56** 455–465. MR3290615 https://doi.org/10.1080/00401706.2013.860918

COHN, D. (1994). Neural network exploration using optimal experiment design. *Adv. Neural Inf. Process. Syst.* **6** 679–686.

COLE, D. A., GRAMACY, R. B., WARNER, J. E., BOMARITO, G. F., LESER, P. E. and LESER, W. P. (2022). Entropy-based adaptive design for contour finding and estimating reliability. *J. Qual. Technol.* 1–18.

DAMIANOU, A. and LAWRENCE, N. D. (2013). Deep Gaussian processes. In *Artificial Intelligence and Statistics* 207–215. PMLR, Arizona, USA.

DUNLOP, M. M., GIROLAMI, M. A., STUART, A. M. and TECKENTRUP, A. L. (2018). How deep are deep Gaussian processes? *J. Mach. Learn. Res.* **19** Paper No. 54, 46. MR3874162

ECONOMON, T. D., PALACIOS, F., COPELAND, S. R., LUKACZYK, T. W. and ALONSO, J. J. (2016). SU2: An open-source suite for multiphysics simulation and design. *AIAA J.* **54** 828–846.

ERIKSSON, D., PEARCE, M., GARDNER, J., TURNER, R. D. and POLOCZEK, M. (2019). Scalable global optimization via local Bayesian optimization. In *Advances in Neural Information Processing Systems* **32**. Curran Associates, Red Hook.

GARDNER, J., PLEISS, G., WEINBERGER, K. Q., BINDEL, D. and WILSON, A. G. (2018). Gpytorch: Blackbox matrix–matrix Gaussian process inference with gpu acceleration. *Adv. Neural Inf. Process. Syst.* **31**.

GNEITING, T. and RAFTERY, A. E. (2007). Strictly proper scoring rules, prediction, and estimation. *J. Amer. Statist. Assoc.* **102** 359–378. MR2345548 https://doi.org/10.1198/016214506000001437

GOODARZI, E., ZIAEI, M. and HOSSEINIPOUR, E. Z. (2014). *Introduction to Optimization Analysis in Hydrosystem Engineering*. Springer, Cham.

GRAMACY, R. B. (2020). *Surrogates—Gaussian Process Modeling, Design, and Optimization for the Applied Sciences*. *Chapman & Hall/CRC Texts in Statistical Science Series*. CRC Press, Boca Raton, FL. MR4283556 https://doi.org/10.1201/9780367815493

GRAMACY, R. B. and APLEY, D. W. (2015). Local Gaussian process approximation for large computer experiments. *J. Comput. Graph. Statist.* **24** 561–578. MR3357395 https://doi.org/10.1080/10618600.2014.914442

GRAMACY, R. B. and LEE, H. K. H. (2008). Bayesian treed Gaussian process models with an application to computer modeling. *J. Amer. Statist. Assoc.* **103** 1119–1130. MR2528830 https://doi.org/10.1198/016214508000000689

GRAMACY, R. B. and LEE, H. K. H. (2009). Adaptive design and analysis of supercomputer experiments. *Technometrics* **51** 130–145. MR2668170 https://doi.org/10.1198/TECH.2009.0015

GRAMACY, R. B., SAUER, A. and WYCOFF, N. (2022). Triangulation candidates for Bayesian optimization. *Adv. Neural Inf. Process. Syst.* **35** 35933–35945.

GRAY, R. M. (2011). *Entropy and Information Theory*, 2nd ed. Springer, New York. MR3134681 https://doi.org/10.1007/978-1-4419-7970-4

HABEL, K., GRASMAN, R., GRAMACY, R. B., MOZHAROVSKYI, P. and STERRATT, D. C. (2019). geometry: Mesh generation and surface tessellation. R package version 0.4.5.

HIGDON, D., SWALL, J. and KERN, J. (1999). Non-stationary spatial modeling. *Bayesian Stat.* **6** 761–768.

IZZATURRAHMAN, M. F., PALAR, P. S., ZUHAL, L. and SHIMOYAMA, K. (2022). Modeling non-stationarity with deep Gaussian processes: Applications in aerospace engineering. In *AIAA Scitech Forum* 1096.

JONES, D. R., SCHONLAU, M. and WELCH, W. J. (1998). Efficient global optimization of expensive black-box functions. *J. Global Optim.* **13** 455–492. MR1673460 https://doi.org/10.1023/A:1008306431147

KENNEDY, M. C. and O'HAGAN, A. (2001). Bayesian calibration of computer models. *J. R. Stat. Soc. Ser. B. Stat. Methodol.* **63** 425–464. MR1858398 https://doi.org/10.1111/1467-9868.00294

LIPPE, M., BITHELL, M., GOTTS, N., NATALINI, D., BARBROOK-JOHNSON, P., GIUPPONI, C., HALLIER, M., HOFSTEDE, G. J., LE PAGE, C. et al. (2019). Using agent-based modelling to simulate social-ecological systems across scales. *GeoInformatica* **23** 269–298.

LUO, J., GUPTA, A., ONG, Y.-S. and WANG, Z. (2018). Evolutionary optimization of expensive multiobjective problems with co-sub-Pareto front Gaussian process surrogates. *IEEE Trans. Cybern.* **49** 1708–1721.

MARQUES, A., LAM, R. and WILLCOX, K. (2018). Contour location via entropy reduction leveraging multiple information sources. *Adv. Neural Inf. Process. Syst.* **31**.

MCKAY, M. D., BECKMAN, R. J. and CONOVER, W. J. (1979). A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics* **21** 239–245. MR0533252 https://doi.org/10.2307/1268522

MEHTA, P. M., WALKER, A., LAWRENCE, E., LINARES, R., HIGDON, D. and KOLLER, J. (2014). Modeling satellite drag coefficients with response surfaces. *Adv. Space Res.* **54** 1590–1607.

MING, D., WILLIAMSON, D. and GUILLAS, S. (2023). Deep Gaussian process emulation using stochastic imputation. *Technometrics* **65** 150–161. MR4580865 https://doi.org/10.1080/00401706.2022.2124311

MURRAY, I., ADAMS, R. P. and MACKAY, D. J. C. (2010). Elliptical slice sampling. In *The Proceedings of the* 13*th International Conference on Artificial Intelligence and Statistics*. *JMLR*: *W&CP* **9** 541–548. PMLR, Sardinia, Italy.

OAKLEY, J. (2004). Estimating percentiles of uncertain computer code outputs. *J. R. Stat. Soc. Ser. C. Appl. Stat.* **53** 83–93. MR2043762 https://doi.org/10.1046/j.0035-9254.2003.05044.x

PACIOREK, C. J. and SCHERVISH, M. J. (2003). Nonstationary covariance functions for Gaussian process regression. In *Proceedings of the* 16*th International Conference on Neural Information Processing Systems*. *NIPS'*03 273–280. MIT Press, Cambridge, MA, USA.

PAMADI, B., COVELL, P., TARTABINI, P. and MURPHY, K. (2004). Aerodynamic characteristics and glide-back performance of langley glide-back booster. In 22*nd Applied Aerodynamics Conference and Exhibit* 5382.

PEDREGOSA, F., VAROQUAUX, G., GRAMFORT, A., MICHEL, V., THIRION, B., GRISEL, O., BLONDEL, M., PRETTENHOFER, P., WEISS, R. et al. (2011). Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **12** 2825–2830. MR2854348

PICHENY, V., GINSBOURGER, D., ROUSTANT, O., HAFTKA, R. T. and KIM, N.-H. (2010). Adaptive designs of experiments for accurate approximation of a target region. *J. Mech. Des.* **132**.

RADAIDEH, M. I. and KOZLOWSKI, T. (2020). Surrogate modeling of advanced computer simulations using deep Gaussian processes. *Reliab. Eng. Syst. Saf.* **195** 106731.

RANJAN, P., BINGHAM, D. and MICHAILIDIS, G. (2008). Sequential experiment design for contour estimation from complex computer codes. *Technometrics* **50** 527–541. MR2655651 https://doi.org/10.1198/004017008000000541

RENGANATHAN, A., RAO, V. and NAVON, I. (2022). Multifidelity Gaussian processes for failure boundary and probability estimation. In *AIAA Scitech Forum* 0390.

RENGANATHAN, S. A., LARSON, J. and WILD, S. M. (2021). Lookahead acquisition functions for finite-horizon time-dependent Bayesian optimization and application to quantum optimal control. Available at arXiv:2105.09824.

RENGANATHAN, S. A., RAO, V. and NAVON, I. M. (2023). CAMERA: A method for cost-aware, adaptive, multifidelity, efficient reliability analysis. *J. Comput. Phys.* **472** Paper No. 111698, 25. MR4502218 https://doi.org/10.1016/j.jcp.2022.111698

ROOCKS, P. (2016). Computing Pareto frontiers and database preferences with the rPref package. *R J.* **8** 393–404.

SALIMBENI, H. and DEISENROTH, M. (2017). Doubly stochastic variational inference for deep Gaussian processes. Available at arXiv:1705.08933.

SAMPSON, P. D. and GUTTORP, P. (1992). Nonparametric estimation of nonstationary spatial covariance structure. *J. Amer. Statist. Assoc.* **87** 108–119.

SANTNER, T. J., WILLIAMS, B. J. and NOTZ, W. I. (2018). *The Design and Analysis of Computer Experiments*. *Springer Series in Statistics*. Springer, New York. 2nd ed. of [MR2160708]. MR3887662

SAUER, A., COOPER, A. and GRAMACY, R. B. (2023a). Vecchia-approximated deep Gaussian processes for computer experiments. *J. Comput. Graph. Statist.* **32** 824–837. MR4641462 https://doi.org/10.1080/10618600.2022.2129662

SAUER, A., COOPER, A. and GRAMACY, R. B. (2023b). Non-stationary Gaussian process surrogates. Available at arXiv:2305.19242.

SAUER, A., GRAMACY, R. B. and HIGDON, D. (2023). Active learning for deep Gaussian process surrogates. *Technometrics* **65** 4–18. MR4543056 https://doi.org/10.1080/00401706.2021.2008505

SAUER, A. E. (2023). Deep Gaussian process surrogates for computer experiments. Ph.D. thesis, Virginia Tech.

SCHMIDT, A. M. and O'HAGAN, A. (2003). Bayesian inference for non-stationary spatial covariance structure via spatial deformations. *J. R. Stat. Soc. Ser. B. Stat. Methodol.* **65** 743–758. MR1998632 https://doi.org/10.1111/1467-9868.00413

SNELSON, E. and GHAHRAMANI, Z. (2006). Sparse Gaussian processes using pseudo-inputs. *Adv. Neural Inf. Process. Syst.* **18** 1259–1266.

STANFORD, B., SAUER, A., JACOBSON, K. and WARNER, J. (2022). Gradient-enhanced reliability analysis of transonic aeroelastic flutter. In *AIAA Scitech Forum* 0632.

STEIN, M. L. (1999). *Interpolation of Spatial Data*: *Some Theory for Kriging*. *Springer Series in Statistics*. Springer, New York. MR1697409 https://doi.org/10.1007/978-1-4612-1494-6

SURJANOVIC, S. and BINGHAM, D. (2013). Virtual library of simulation experiments: Test functions and datasets. Available at http://www.sfu.ca/~ssurjano.

TIETZE, N. (2015). Model-based calibration of engine control units using Gaussian process regression. Ph.D. thesis, Technische Universität.

VASSBERG, J., DEHAAN, M., RIVERS, M. and WAHLS, R. (2008). Development of a common research model for applied CFD validation studies. In 26*th AIAA Applied Aerodynamics Conference* 6919.

ZIMMER, L., LINDAUER, M. and HUTTER, F. (2021). Auto-pytorch: Multi-fidelity MetaLearning for efficient and robust AutoDL. *IEEE Trans*. *Pattern Anal*. *Mach*. *Intell*. **43** 3079–3090. https://doi.org/10.1109/TPAMI.2021.3067763