Scaled Vecchia Approximation for Fast Computer-Model Emulation[*]

Matthias Katzfuss[†], Joseph Guinness[‡], and Earl Lawrence[§]

Abstract. Many scientific phenomena are studied using computer experiments consisting of multiple runs of a computer model while varying the input settings. Gaussian processes (GPs) are a popular tool for the analysis of computer experiments, enabling interpolation between input settings, but direct GP inference is computationally infeasible for large datasets. We adapt and extend a powerful class of GP methods from spatial statistics to enable the scalable analysis and emulation of large computer experiments. Specifically, we apply Vecchia's ordered conditional approximation in a transformed input space, with each input scaled according to how strongly it relates to the computer-model response. The scaling is learned from the data by estimating parameters in the GP covariance function using Fisher scoring. Our methods are highly scalable, enabling estimation, joint prediction, and simulation in near-linear time in the number of model runs. In several numerical examples, our approach substantially outperformed existing methods.

1. Introduction. At the cutting edge of science, computationally intensive simulations are used to make predictions of complex phenomena, such as the distribution of matter in the universe (Lawrence et al., 2017), the behavior of materials under high pressure (Walters et al., 2018), or the composition of rocks on Mars (Bhat et al., 2020). These simulations are simply too slow for use in data analysis (Higdon et al., 2004) or real-time applications (Mehta et al., 2014), so the statistics discipline known as computer experiments has grown to address this computational challenge. The key ingredient in much of this work is an emulator, a statistical approximation to the computer simulation. Emulators can predict the output of a simulation many orders of magnitude faster than the simulation itself, at the cost of additional error. Emulation is achieved by building a regression model from the inputs to the outputs.

[†]Corresponding author. Department of Statistics, Texas A&M University, College Station, TX 77843 USA (katzfuss@gmail.com).

[‡]Department of Statistics and Data Science, Cornell University, Ithaca, NY 14850 USA (guinness@cornell.edu).

[§]Los Alamos National Laboratory, Los Alamos, NM 87545 USA (earl@lanl.gov).

Gaussian processes (GPs) are popular emulators and have emerged as indispensable tools for design, analysis, and calibration of computer experiments (e.g., Sacks et al., 1989; Kennedy and O'Hagan, 2001). GPs are accurate, flexible, interpretable, and probabilistic, thus providing natural quantification of uncertainty. For the analysis of n computer-model runs, GP inference typically requires working with a dense $n \times n$ covariance matrix. Thus, direct GP inference is infeasible for many present and future computer experiments, as new supercomputers enable increasingly large numbers of increasingly detailed simulations to be carried out. Scalability improvements for computer-experiment methods are vital to handle the expected increase in simulation output.

Many approaches have been proposed to enable scalable GP inference. Heaton et al. (2019) review and compare approaches from spatial statistics, and Liu et al. (2020) review approaches in machine learning. In the context of large computer experiments, scalable GP approaches include: compactly supported covariances (Kaufman et al., 2011); sparse grid-based GPs (Plumlee, 2014); and the local approximate GP (laGP) of Gramacy and Apley (2015), which makes independent predictions at different input values using nearby observations in the input space. In spatial statistics, the Vecchia approximation (Vecchia, 1988) and its extensions (e.g., Stein et al., 2004; Datta et al., 2016a; Guinness, 2018; Katzfuss and Guinness, 2021; Katzfuss et al., 2020a) are very popular GP approximations. Similar to the laGP, the Vecchia approximation considers nearest neighbors, but it does so from an ordered conditional perspective; as a result, Vecchia approximations imply a valid joint distribution for the data, resulting in straightforward global likelihood-based parameter inference and joint predictions at a set of input values, which allows proper uncertainty quantification in downstream applications.

Here, we adapt and extend the powerful class of Vecchia GP approximations from spatial statistics to enable the scalable analysis and emulation of large computer experiments. Specifically, we apply Vecchia's ordered conditional approximation in a transformed input space, for which each input is scaled according to how strongly it is related to the computer-model response. The scaling of the input space is learned from the data by estimating parameters in the GP covariance function using Fisher scoring (Guinness, 2021). Our scaled Vecchia methods are highly scalable, enabling ordering, neighbor-search, estimation, joint prediction, and simulation in near-linear time in the number of model runs. Thus, our methods can handle large numbers of model runs, joint sampling of paths through the input space, and relatively high input dimensions, assuming that only some of the inputs have a strong effect on the output, while others are less important. Recently, Shi et al. (2017) also applied a Vecchia-type approximation in a GP-emulation setting, but their focus was on infinitely smooth covariances (i.e., squared exponential) and change-of-support problems. Datta et al. (2016b) proposed a Bayesian Vecchia-type approximation for spatio-temporal processes, in which the neighbors (but not the ordering) were selected based on the spatio-temporal correlations; for a specific choice of covariance function, their correlation-based neighbor-selection procedure is similar to our neighbor-selection based on distances between scaled inputs.

Relative to recent work on Vecchia approximations of spatial processes by the authors (Guinness, 2018; Katzfuss and Guinness, 2021; Katzfuss et al., 2020a; Guinness, 2021), the present paper makes several contributions that are crucial to addressing challenges with computer-model emulation. We extend the Vecchia approximation to nonspatial inputs and

to anisotropic covariance functions. The Vecchia ordering and neighbor sets are determined based on scaled inputs; this greatly improves the accuracy in the high-dimensional input spaces common in computer experiments (as opposed to the usual two-dimensional space in spatial statistics). When estimating the (unknown) scaling parameters in an iterative fashion, the scaling of the inputs changes along with the parameter estimates over the course of the Fisher-scoring iterations; hence, as we refine the estimates of the parameters, we refine our Vecchia approximation of the implied anisotropic covariance. As a stationary GP may be less appropriate for modeling some computer-model surfaces than many geospatial fields, model misspecification and resulting underestimation of prediction uncertainty may be an issue; to address this, we developed a simple, computationally cheap, and effective variance-correction approach, resulting in well-calibrated and sharp predictive distributions.

The remainder of this document is organized as follows. In section 2, we describe GP emulation of computer models, and we review existing Vecchia approximations from spatial statistics. In section 3, we introduce our new scaled Vecchia methods for fast emulation of large computer experiments. In section 4, we provide numerical comparisons to existing approaches. Section 5 concludes and discusses future work. R code to run our method and reproduce all results is available at https://github.com/katzfuss-group/scaledVecchia.

2. Review.

2.1. Computer-model emulation using Gaussian processes.
Let $y(\mathbf{x})$ be the response of a computer model at a d-dimensional input vector $\mathbf{x}$ on the input domain $\mathcal{X}$. In GP emulation, $y(\cdot) \sim \mathcal{GP}(\mu, K)$ is assumed to be a GP with mean function $\mu : \mathcal{X} \to \mathbb{R}$ and a positive-definite covariance or kernel function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. Then, the vector $\mathbf{y} = (y(\mathbf{x}_1), \ldots, y(\mathbf{x}_n))^\top$ of responses at n input values $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ follows an n-variate Gaussian distribution with covariance matrix $\mathbf{K} = (K(\mathbf{x}_i, \mathbf{x}_j))_{i,j=1,\ldots,n}$, whose (i,j)th entry describes the covariance between the responses of simulations i and j as a function of the corresponding input values $\mathbf{x}_i$ and $\mathbf{x}_j$.

For simplicity, we henceforth make some additional assumptions, although most of our methodology is also applicable in more general situations. Specifically, we assume that the mean function $\mu(\mathbf{x}) = \boldsymbol{\psi}(\mathbf{x})^\top \boldsymbol{\beta}$ is linear in a number of covariate parameters, $\boldsymbol{\beta}$; a typical assumption is $\boldsymbol{\psi}(\mathbf{x}) = 1$ or $\boldsymbol{\psi}(\mathbf{x}) = (1, \mathbf{x}^\top)^\top$.

We also assume an anisotropic covariance function K with a separate range parameter λ_l for each input dimension l, also referred to as automatic relevance determination: $K(\mathbf{x}_i, \mathbf{x}_j) = \tilde{K}(q(\mathbf{x}_i, \mathbf{x}_j))$, where

$$(1) \qquad q(\mathbf{x}_i, \mathbf{x}_j) = \left(\sum_{l=1}^{d} \left(\frac{x_{i,l} - x_{j,l}}{\lambda_l} \right)^2 \right)^{1/2},$$

and $\tilde{K}$ can be any covariance function that is valid (i.e., strictly positive definite) in $\mathbb{R}^d$, such as the squared exponential or Matérn covariance function.

While GPs are indispensable tools for computer-model emulation due to their flexibility and natural uncertainty quantification, direct GP inference requires an $\mathcal{O}(n^3)$ factorization of the covariance matrix, which is not feasible for large computer experiments. Thus, we propose an approximation that reduces computational complexity and hence improves scalability.

2.2. Vecchia approximations in spatial statistics. Vecchia's approximation (Vecchia, 1988) is a powerful GP approximation that is popular in spatial statistics. Motivated by the exact decomposition of the joint density $p(\mathbf{y}) = \prod_{i=1}^{n} p(y_i | y_1, \ldots, y_{i-1})$ as a product of univariate conditional densities, Vecchia (1988) proposed the approximation

$$(2) \qquad \widehat{p}(\mathbf{y}) = \prod_{i=1}^{n} p(y_i | \mathbf{y}_{c(i)}),$$

where $c(i) \subset \{1, \ldots, i-1\}$ is a conditioning index set of size $|c(i)| = \min(m, i-1)$ for all $i = 2, \ldots, n$ (and $c(1) = \emptyset$). Even with relatively small conditioning-set size $m \ll n$, the approximation (2) with appropriate choice of the $c(i)$ can often be very accurate due to the screening effect (e.g., Stein, 2011). The $p(y_i | \mathbf{y}_{c(i)})$ in (2) are all Gaussian distributions that can be computed in parallel using standard formulas, each using $\mathcal{O}(m^3)$ operations based on $\mathcal{O}(m)$ data.

The Vecchia approximation has many useful properties. For example, the implied joint distribution $\widehat{p}(\mathbf{y}) = \mathcal{N}_n(\boldsymbol{\mu}, \widehat{\mathbf{K}})$ is also multivariate Gaussian, and the Cholesky factor of $\widehat{\mathbf{K}}^{-1}$ is highly sparse with fewer than nm off-diagonal nonzero entries (e.g., Datta et al., 2016a; Katzfuss and Guinness, 2021). Further, under the sparsity constraint implied by the choice of the $c(i)$, the Vecchia approximation results in the optimal inverse Cholesky factor $\widehat{\mathbf{K}}^{-1/2}$, as measured by the Kullback–Leibler (KL) divergence, $KL(p(\mathbf{y}) \| \widehat{p}(\mathbf{y}))$ (Schäfer et al., 2021a). Enlarging the conditioning sets $c(i)$ never increases the KL divergence (Guinness, 2018); for $m = n - 1$, the approximation is exact, $\widehat{p}(\mathbf{y}) = p(\mathbf{y})$. In contrast to local GP approximations, the Vecchia approximation to the underlying model is global; thus, for example, model parameters can be estimated (see section 3.2) from a subsample of the data, and then the estimated parameters can be used to make predictions (section 3.3) using all of the data.

The approximation accuracy of the Vecchia approach depends on the choice of ordering of the variables $y_1, \ldots, y_n$ and on the choice of the conditioning sets $c(i)$. A general Vecchia framework (Katzfuss and Guinness, 2021) obtained by varying these choices unifies many popular GP approximations (e.g., Quiñonero-Candela and Rasmussen, 2005; Snelson and Ghahramani, 2007; Banerjee et al., 2008; Katzfuss, 2017; Katzfuss and Gong, 2020). In practice, high accuracy can be achieved using a maximum-minimum distance (maximin) ordering and nearest-neighbor (NN) conditioning, which are illustrated in Figure 1a. Maximin ordering picks the first variable arbitrarily and then chooses each subsequent variable in the ordering as the one that maximizes the minimum distance to previous variables in the ordering. For NN conditioning, each $c(i)$ then consists of the indices corresponding to the m nearest previously ordered variables. (For $1 < i < m + 2$, we simply have $c(i) = \{1, \ldots, i-1\}$.) Due to the maximin ordering, all previously ordered inputs are far from the ith input for small i, and so many possible distances are represented in the conditioning sets for different i. For both ordering and conditioning, distance between two variables y_i and y_j is typically defined as the Euclidean distance $\|\mathbf{x}_i - \mathbf{x}_j\|$ between their corresponding inputs. In addition, we employ a grouping strategy (Guinness, 2018) that combines conditioning sets $c(i)$ and $c(j)$ when doing so is computationally advantageous. When using maximin ordering and NN conditioning, recent results (Schäfer et al., 2021a) imply that, for increasing n, a specific accuracy for certain isotropic Matérn kernels can be guaranteed using conditioning sets of size $m = \mathcal{O}(\log^d(n))$,
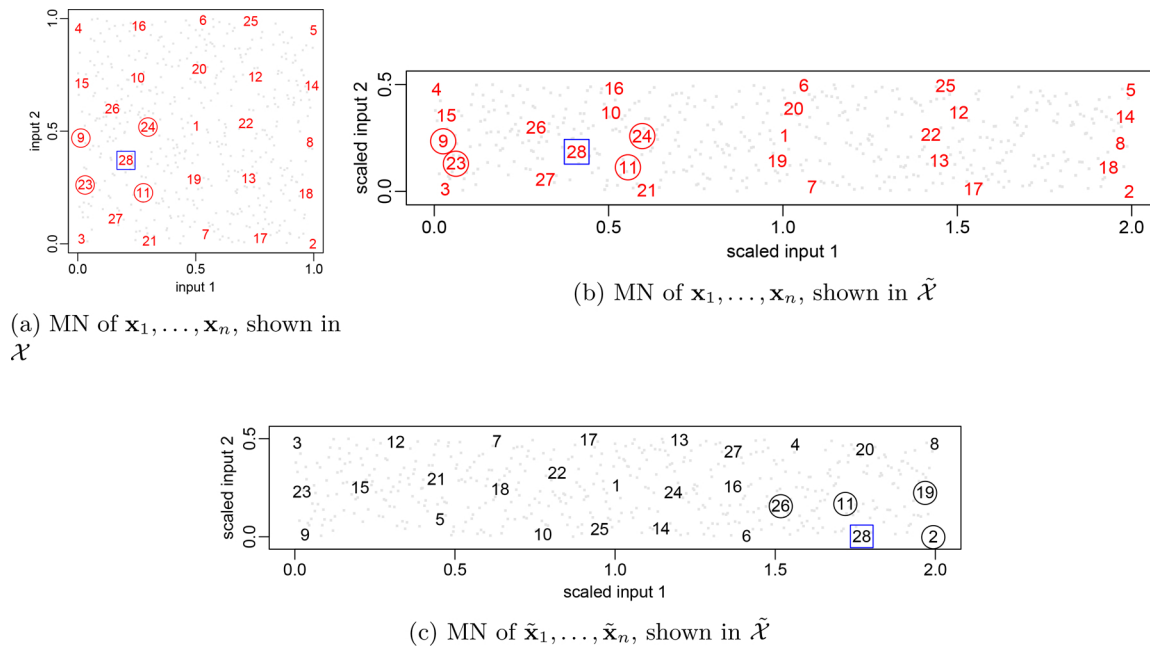
(a) MN of $\mathbf{x}_1, \ldots, \mathbf{x}_n$, shown in $\mathcal{X}$

(b) MN of $\mathbf{x}_1, \ldots, \mathbf{x}_n$, shown in $\tilde{\mathcal{X}}$

(c) MN of $\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_n$, shown in $\tilde{\mathcal{X}}$

Figure 1. *Maximin ordering and nearest-neighbor conditioning (MN) for $n = 500$ inputs (small grey points) generated using Latin hypercube sampling on $\mathcal{X} = [0,1]^2$ in $d = 2$ dimensions, assuming an anisotropic covariance* (1) *with range parameters $\boldsymbol{\lambda} = (1/2, 2)$. MN is carried out on the original inputs $\mathbf{x}_1, \ldots, \mathbf{x}_n$ (top row, red) or the scaled inputs $\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_n$ (bottom row, black). The first $i = 28$ ordered inputs are numbered, with emphasis on the ith input ($\square$) and its nearest $m = 4$ previously ordered neighbors with indices $c(i)$ ($\bigcirc$). (a) MN of original inputs viewed on original space $\mathcal{X} = [0,1] \times [0,1]$: The first i inputs are spread out over input space; $c(i) = (9, 11, 23, 24)$ are nearby. (b) Same MN on scaled space $\tilde{\mathcal{X}} = [0,2] \times [0, 1/2]$: The first i inputs are irregularly spaced; $c(i)$ missed near 26 and 27. (c) MN of scaled inputs on scaled space: The first i inputs are spread out over input space; $c(i) = (2, 11, 19, 26)$ are nearby, as desired.*

under regularity conditions and ignoring edge effects. The resulting near-linear time complexity is the best known complexity for problems of this type.

3. Methodology. Several issues arise when applying Vecchia approximations for spatial GPs to emulation of computer experiments (section 2.1). While physical distance between spatial locations is usually meaningful, Euclidean distance between inputs to a computer experiment depends heavily on the arbitrary scaling of each input dimension. In addition, while spatial fields are typically two- or three-dimensional, computer experiments often consider $d \approx 10$ inputs; as the asymptotics discussed at the end of section 2.2 imply that $m = \mathcal{O}(\log^d(n))$ is required to achieve a certain accuracy, a very large m might be required for large d, resulting in a prohibitive computational cost (which scales cubically in m).

3.1. Scaled Vecchia approximation for computer experiments. We propose a scaled Vecchia approximation that exploits that the input variables can vary widely in the magnitude of their effect on the response; this is sometimes referred to as factor sparsity. Specifically, for known range parameters $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_d)^\top$, the anisotropic distance in (1) can be viewed as a Euclidean distance between scaled inputs,

(3) $$q(\mathbf{x}_i, \mathbf{x}_j) = \|\tilde{\mathbf{x}}_i - \tilde{\mathbf{x}}_j\|,$$

where $\tilde{\mathbf{x}} = (x_1/\lambda_1, \ldots, x_d/\lambda_d)$ are the scaled inputs, and we call $1/\lambda_l$ the relevance of the lth input dimension or variable x_l (assuming standardized input space $\mathcal{X} = [0,1]^d$). Similar scaling ideas have been considered for other GP approximations of computer experiments (e.g., Gramacy, 2016).

Our scaled Vecchia approximation is defined as in (2), except based on a maximin ordering and NN conditioning of the scaled inputs $\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_n$, assuming known $\boldsymbol{\lambda}$ for now; more precisely, we define the distance between variables y_i and y_j as the Euclidean distance $\|\tilde{\mathbf{x}}_i - \tilde{\mathbf{x}}_j\|$ between the corresponding scaled inputs (see Figure 1c), instead of $\|\mathbf{x}_i - \mathbf{x}_j\|$ in the standard Vecchia approximation. Note that this scaled Vecchia approximation can be viewed as a special case of correlation-based Vecchia (Kang and Katzfuss, 2021). The ordering and conditioning can be computed in quasilinear time in n (Schäfer et al., 2021b,a).

The resulting scaled Vecchia approximation of the GP $y(\cdot)$ with anisotropic kernel K, can be viewed as a standard Vecchia approximation of a GP with isotropic kernel $\tilde{K}$ with scaled inputs $\tilde{\mathbf{x}}$ in the scaled input space $\tilde{\mathcal{X}}$. Importantly, Euclidean distance is only meaningful in $\tilde{\mathcal{X}}$, not in $\mathcal{X}$. Figure 1b shows that maximin ordering of $\mathbf{x}_1, \ldots, \mathbf{x}_n$ can be highly irregular in $\tilde{\mathcal{X}}$, and nearest-neighbor conditioning of $\mathbf{x}_1, \ldots, \mathbf{x}_n$ may miss important and nearby inputs in $\tilde{\mathcal{X}}$. In contrast, scaled Vecchia (Figure 1c) is directly carried out in $\tilde{\mathcal{X}}$; the resulting ordering is more regular, and the conditioning set $c(i)$ contains the nearest previously ordered neighbors, as desired to achieve good screening properties in the conditional distributions in (2).

Many computer codes contain input variables x_l that only weakly affect the response y; this can be captured in our model by a large λ_l, so that changes in x_l only result in small changes in $\tilde{x}_l = x_l/\lambda_l$, and thus only minor changes in position in $\tilde{\mathcal{X}}$. In the extreme case of $\lambda_l \to \infty$, the input variable x_l is effectively eliminated from the model, and the dimension $\tilde{d}$ of $\tilde{\mathcal{X}}$ is smaller than the dimension d of the original input space, and thus a smaller $m = \mathcal{O}(\log^{\tilde{d}}(n))$ is required to achieve a certain approximation accuracy. But even for large but finite range parameters, Figure 2a shows that scaled Vecchia can achieve a certain accuracy with much smaller m than standard Vecchia (see section 4.2 for more details).

3.2. Estimation of parameters. In practice, the parameters $\boldsymbol{\beta}$ in the mean function μ and parameters $\boldsymbol{\theta}$ in the covariance function K are unknown, including the range or scaling parameters $\boldsymbol{\lambda}$. We estimate these parameters by maximizing the logarithm of the Vecchia likelihood in (2). This is challenging due to the potentially large number of parameters. Hence, we use a Fisher scoring algorithm (Guinness, 2021), which exploits first- and second-derivative information for fast convergence but preserves the $\mathcal{O}(nm^3)$ scaling of the Vecchia approximation. We briefly review this algorithm here, but refer to Guinness (2021) for details.

Let $\ell(\boldsymbol{\beta}, \boldsymbol{\theta}) = \log \hat{p}_{\boldsymbol{\beta}, \boldsymbol{\theta}}(\mathbf{y})$, where $\hat{p}_{\boldsymbol{\beta}, \boldsymbol{\theta}}(\mathbf{y})$ is the Vecchia approximation from (2) with $m = m_{\text{est}}$, except that we have now made explicit the dependence of the density on the parameters. Taking derivatives of the conditional densities in (2) is challenging; replacing them by joint distributions,

(4) $$\ell(\boldsymbol{\beta}, \boldsymbol{\theta}) = \sum_{i=1}^{n} \left(\log p_{\boldsymbol{\beta}, \boldsymbol{\theta}}(y_i, \mathbf{y}_{c(i)}) - \log p_{\boldsymbol{\beta}, \boldsymbol{\theta}}(\mathbf{y}_{c(i)}) \right),$$

enables the use of well-known formulas for the gradient and Fisher information of the Gaussian distributions $p_{\boldsymbol{\beta}, \boldsymbol{\theta}}(y_i, \mathbf{y}_{c(i)})$ and $p_{\boldsymbol{\beta}, \boldsymbol{\theta}}(\mathbf{y}_{c(i)})$. Because $\boldsymbol{\beta}$ appears linearly in the mean of the
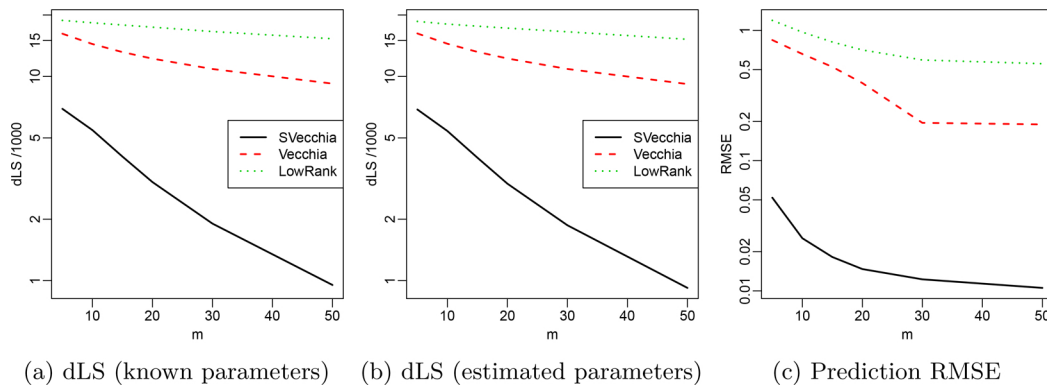
(a) dLS (known parameters) (b) dLS (estimated parameters) (c) Prediction RMSE

Figure 2. *For data simulated from a Matérn GP in $d = 10$ input dimensions, comparison of our proposed scaled Vecchia (SVecchia) approach to two existing GP approximations, in terms of average difference in log score (dLS), which approximates KL divergence, and in terms of prediction root mean square error (RMSE)—see* section 4.2 *for details.*

Gaussian distributions, we can profile out $\boldsymbol{\beta}$ using the closed-form expression for the generalized least squares estimator $\hat{\boldsymbol{\beta}}(\boldsymbol{\theta})$. Then, starting with an initial value $\boldsymbol{\theta}^{(0)}$, Fisher scoring for $\boldsymbol{\theta}$ proceeds for $k = 0, 1, 2, \ldots$ as

$$(5) \qquad \boldsymbol{\theta}^{(k+1)} = \boldsymbol{\theta}^{(k)} + (\mathbf{M}^{(k)})^{-1}\mathbf{g}^{(k)},$$

where $\mathbf{g}^{(k)} = \frac{\partial \ell(\hat{\boldsymbol{\beta}}(\boldsymbol{\theta}), \boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\big|_{\boldsymbol{\theta} = \boldsymbol{\theta}^{(k)}}$ and $\mathbf{M}^{(k)} = -\mathbb{E}\frac{\partial^2 \ell(\hat{\boldsymbol{\beta}}(\boldsymbol{\theta}), \boldsymbol{\theta})}{\partial \boldsymbol{\theta}\partial \boldsymbol{\theta}'}\big|_{\boldsymbol{\theta} = \boldsymbol{\theta}^{(k)}}$ can be computed based on (4) as the sum of n log-densities that are at most of dimensions $m+1$. The algorithm is terminated when the dot product between the step and the gradient $\mathbf{g}^{(k)}$ is less than 10^{-4}, obtaining the estimates $\hat{\boldsymbol{\theta}} = \boldsymbol{\theta}^{(k+1)}$ and $\hat{\boldsymbol{\beta}} = \hat{\boldsymbol{\beta}}(\hat{\boldsymbol{\theta}})$. In practice, a mild penalization term (e.g., to discourage variance parameters that are much larger than the sample variance of the training data) is added to (4) to improve convergence. Also, when the Fisher-scoring step fails to increase the loglikelihood, the step is replaced by a line search along the gradient. This concludes the review of Guinness (2021).

In our scaled Vecchia approach, over the course of the Fisher-scoring iterations, the estimate of $\boldsymbol{\theta}$ will change, and along with it, the scaled inputs $\tilde{\mathbf{x}} = (x_1/\lambda_1, \ldots, x_d/\lambda_d)$, the resulting maximin ordering and NN conditioning, and the implied approximate density $\widehat{p}_{\boldsymbol{\beta},\boldsymbol{\theta}}(\mathbf{y})$. For the purpose of computing the derivatives and the resulting parameter update in (5), we ignore the dependence of the ordering and conditioning on $\boldsymbol{\theta}$; instead, we update the ordering and conditioning separately given the current estimate $\boldsymbol{\theta}^{(k)}$, but only at certain iterations, say $k = 2, 4, 8, 16, \ldots$ to avoid slowing the algorithm unnecessarily. Hence, as we refine the estimates of the parameters, we refine our Vecchia approximation of the implied covariance. If it is of interest, we can carry out crude variable selection and eliminate inactive input dimensions by setting $\lambda_l = \infty$ if $\lambda_l^{(k)}$ is over a certain threshold (e.g., 10^3).

Figure 2b shows that our scaled Fisher-scoring approach can be much more accurate than using standard Vecchia (see section 4.2 for details). The estimation algorithm converged quickly, requiring only around ten Fisher-scoring iterations to estimate eleven parameters.

3.3. Prediction. Given the estimated parameters $\hat{\boldsymbol{\theta}}$ and $\hat{\boldsymbol{\beta}}$, we would like to predict the response $y(\cdot)$ at unobserved inputs, $\mathbf{x}_1^*, \ldots, \mathbf{x}_{n_*}^*$. This is equivalent to obtaining the posterior distribution of $\mathbf{y}^* = (y_1^*, \ldots, y_{n_*}^*)^\top$, where $y_i^* = y(\mathbf{x}_i^*)$. To be able to compute this distribution even for large n or n_*, we apply a Vecchia approximation to the joint density $p(\mathbf{y}^{\mathrm{all}})$, where $\mathbf{y}^{\mathrm{all}} = (\mathbf{y}, \mathbf{y}^*)$. To do so, we employ a maximin ordering of the scaled inputs corresponding to $\mathbf{y}^{\mathrm{all}}$, under the restriction that the entries of $\mathbf{y}$ are ordered before those in $\mathbf{y}^*$ (Guinness, 2018; Katzfuss et al., 2020a). As a result, we can write $\hat{p}(\mathbf{y}, \mathbf{y}^*) = \hat{p}(\mathbf{y})\hat{p}(\mathbf{y}^*|\mathbf{y})$, where $\hat{p}(\mathbf{y})$ is as before in (2), and the desired posterior predictive distribution is

$$(6) \qquad \hat{p}(\mathbf{y}^*|\mathbf{y}) = \prod_{i=1}^{n_*} p(y_i^*|\mathbf{y}_{g^*(i)}^{\mathrm{all}}),$$

and $g^*(i)$ contains the indices of the m_* variables that are closest to y_i^* (in terms of scaled distance) among those that are previously ordered in $\mathbf{y}^{\mathrm{all}}$. It is possible for few or even none of the indices in a particular set $g^*(i)$ to correspond to observations, so that y_i^* only conditions on other unobserved prediction variables; however, because these prediction variables may, in turn, condition on observations, the resulting predictions can be as good or better marginally (and much more accurate jointly) than predictions using only observations (e.g., compare the methods LF-full and LF-ind in the first two rows of Fig. 5 in Katzfuss et al., 2020a).

As in (2), all the univariate conditionals, $p(y_i^*|\mathbf{y}_{g^*(i)}^{\mathrm{all}})$, are Gaussian and can be computed in $\mathcal{O}(m_*^3)$ time. It is straightforward to, for example, compute the mean of or draw *joint* samples from $\hat{p}(\mathbf{y}^*|\mathbf{y})$ using the expression (6). In addition, $\hat{p}(\mathbf{y}, \mathbf{y}^*)$ is jointly Gaussian with a sparse inverse Cholesky factor, from which any distributional summary of interest can be computed (Katzfuss et al., 2020a). These properties enable joint simulation and uncertainty quantification for a set of unobserved input values, such as a path through the input space.

3.4. Variance correction. Our SVecchia approach uses the GP model in section 2.1, which (like virtually all statistical models) is misspecified, in that computer models are not truly realizations of such a GP. As the predictions in section 3.3 do not account for this model misspecification, the predictive distributions may sometimes be poorly calibrated with variances that are too small. To address this, we developed a variance-correction approach that is simple, computationally cheap, and highly effective in the examples we have studied.

Specifically, we first estimate the parameters $\hat{\boldsymbol{\theta}}$ and $\hat{\boldsymbol{\beta}}$ as described in section 3.2. We then randomly split the original training set into an "inner" training and test set. Computing SVecchia predictions (section 3.3) at the inner test set given the inner training set, the predictive distribution for the ith inner test observation y_i° is Gaussian, say $\mathcal{N}(\mu_i, \sigma_i^2)$. We modify this distribution to $\mathcal{N}(\mu_i, b\sigma_i^2)$, where the variance correction factor b is chosen to optimize a scoring rule (e.g., Gneiting and Katzfuss, 2014) involving y_i° and $\mathcal{N}(\mu_i, b\sigma_i^2)$. In our implementation, we use the log score, which means that b is chosen to minimize the sum of the negative log predictive densities at the inner test points, $-\sum_i \log \mathcal{N}(y_i^\circ|\mu_i, b\sigma_i^2)$. This minimization problem consists of a simple line search, whose computational cost is negligible relative to that of estimating $\hat{\boldsymbol{\theta}}$ and $\hat{\boldsymbol{\beta}}$. Given b, we then make predictions at unobserved test inputs as described in section 3.3, multiplying each prediction variance by b.

In our numerical experiments in section 4 below, the estimates of b ranged from around 1 (for the piston function) to the hundreds (for the very simplistic borehole function). The

resulting corrected predictive distributions were sharp and well calibrated, with the empirical coverage of 95% prediction intervals close to 95%.

3.5. Implementation. We implemented our methods in R, building on top of the R package GpGp (Guinness and Katzfuss, 2018). We provide the anisotropic covariance function matern_scaleDim as in (1), where $\tilde{K}$ is the isotropic Matérn covariance (e.g., Stein, 1999). We also provide its special cases for half-integer smoothness values 0.5, 1.5, 2.5, 3.5, 4.5; when the smoothness is fixed by the user at one of these values, expensive Bessel function evaluations are avoided. Parameter estimation is based on the Fisher-scoring procedure in GpGp; at iterations $k = 2, 4, 8, 16, \ldots$ we update the ordering and conditioning of the current scaled inputs $\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_n$, using the exact maximin ordering algorithm implemented in GPvecchia (Katzfuss et al., 2020b). Each ordering and conditioning can be computed in quasilinear time in n (Schäfer et al., 2021a), and in practice the added time is negligible relative to a standard Vecchia approximation that keeps the ordering and conditioning fixed. We also provide an efficient implementation for our variance-correction procedure and for Vecchia predictions based on scaled inputs.

Due to the global nature of the Vecchia approximation (see section 2.2), it is possible to separate training of our emulator (i.e., parameter estimation) from prediction. As parameter estimation requires multiple Fisher-scoring iterations, we recommend using relatively small conditioning sets of size $m = m_{\text{est}}$ for the Vecchia density (2) used for the parameter estimation described in section 3.2, and of larger size $m = m_*$ for the Vecchia approximation of the predictive distribution in (6). In addition, our numerical experiments below showed that a random subsample of the training data of size n_{est} in the low thousands was enough to estimate the small number of unknown mean and covariance parameters for the $d \leq 10$ considered here. For this SVecchia procedure, the computational cost is independent of the full training size n, aside from negligible preprocessing costs. For parameter estimation, each Fisher-scoring iteration scales roughly as $\mathcal{O}(n_{\text{est}} m_{\text{est}}^3)$; given the estimated parameters, prediction at n_* input values based on the full training set of size n scales as $\mathcal{O}(n_* m_*^3)$. The computations can be carried out in parallel across the n_{est} terms for estimation. Keeping in mind these computational costs, we recommend setting the tuning parameters $n_{\text{est}}, m_{\text{est}}, m_*$ as large as possible (to maximize accuracy) within given computational constraints. The default values in our implementation are $n_{\text{est}} = 5{,}000$, $m_{\text{est}} = 30$, and $m_* = 140$.

The code is available at https://github.com/katzfuss-group/scaledVecchia. Using default settings, scaled-Vecchia estimation and prediction is as simple as

```
fit <- fit_scaled( y.train, inputs.train )
preds <- predictions_scaled( fit, inputs.test ).
```

3.6. Design. Our methods can also be extended straightforwardly for the design of computer experiments. For example, consider the following two-stage design of total size n. In the first stage, we obtain a small number of runs, say $n_1 = n/10$, with input values chosen by a space-filling design, such as a Latin hypercube (LH). Then, we apply our estimation method from section 3.2 to the n_1 responses to obtain an estimate of $\boldsymbol{\theta}$, including the estimated ranges $\hat{\boldsymbol{\lambda}}_1$. In the second stage, we "oversample," say, $N = 20n$ inputs values using a LH design, and then we choose the first $n_2 = n - n_1$ inputs in a maximin ordering of the scaled space determined by the range estimates $\hat{\boldsymbol{\lambda}}_1$ from the first stage. Finally, based on the resulting

full dataset of size $n = n_1 + n_2$, we can re-estimate the parameters and make predictions at unobserved input values as described in section 3.3. Note that such a "sensitivity-weighted distance" has previously been considered for small sequential designs in Williams et al. (2011).

In addition, our methods can be used for designs based on optimization criteria (e.g., Mockus, 1989; Jones et al., 1998), sometimes referred to as Bayesian or model-based optimization. These sequential designs at each stage require re-estimation of parameters and predictions at large numbers of inputs (e.g., to compute the expected improvement), which can be carried out rapidly using our methods.

4. Numerical comparisons.

4.1. General information.
We carried out numerical studies comparing the following methods:

SVecchia: Our proposed scaled Vecchia approximation, as described in section 3

Vecchia: Existing standard Vecchia approximation, with maximin ordering and nearest-neighbor conditioning based on Euclidean distance $\|\mathbf{x}_i - \mathbf{x}_j\|$ between inputs.

LowRank: Modified predictive process (Finley et al., 2009), equivalent to Vecchia, except that all variables simply condition on the first m variables in the (Euclidean) maximin ordering: $c(i) = (1, \ldots, m)$ for $i > m$.

laGP: Local approximate GP (Gramacy and Apley, 2015; Gramacy, 2016).

H-laGP: Hybrid global-local extension of laGP (Sun et al., 2019, Sect. 3) with prescaling based on a random subsample of size 1,000.

For SVecchia, Vecchia, and LowRank, we assumed zero mean $\mu(\mathbf{x}) = 0$, and $\tilde{K}$ above (1) was a Matérn covariance with smoothness 3.5 and zero nugget. For each comparison, n training input values were generated using LH sampling using the R package lhs (Carnell, 2019), and n_* test inputs were sampled uniformly at random on $\mathcal{X}$.

4.2. Matérn simulations.
We considered $n = 5,000$ responses simulated from a GP with mean zero and Matérn covariance function with smoothness 3.5 in $d = 10$ dimensions. We assumed two "relevant" input dimensions with range parameters $\lambda_1 = \lambda_2 = .05$, and eight less relevant inputs with range parameters $\lambda_3 = \cdots = \lambda_{10} = 5$. Only squared-exponential covariances are implemented in laGP, and so laGP was not included in this comparison. For the other three methods, we considered the average difference in log scores (dLS; Gneiting and Katzfuss, 2014) or loglikelihoods, $\log p(\mathbf{y}) - \log \widehat{p}(\mathbf{y})$, over ten datasets $\mathbf{y} \sim p(\mathbf{y})$ simulated from the true model; this score approximates the KL divergence between the true and approximated model. For each of the ten datasets and each of the approximation methods with different values of m, we estimated the parameters using $m_{\text{est}} = m$, made predictions at $n_* = 2,000$ unobserved test inputs using $m_* = 2m$, and computed the root mean square error (RMSE) between the true test responses $\mathbf{y}^*$ and the corresponding predictive means (averaged over the ten datasets).

Figure 2a shows the dLS when assuming that the covariance function (including its parameters) was known. Vecchia was more accurate than LowRank, but SVecchia resulted in additional, substantial improvement. For example, SVecchia with $m = 5$ was more accurate than Vecchia (or LowRank) with $m = 50$; due to the cubic scaling in m, this implies a 1,000-fold decrease in computational cost for a given accuracy. For Figure 2b, the parameters $\boldsymbol{\theta}$

were assumed unknown and estimated from the data, but the resulting dLS were very similar to the known-parameter case. Figure 2c shows that SVecchia predictions were much more accurate than those using Vecchia or LowRank.

4.3. Borehole function. We carried out a simulation study comparing prediction accuracy for the Vecchia-based methods (i.e., SVecchia, Vecchia, and LowRank) using the popular borehole-function example (Morris et al., 1993), which models the water-flow rate through a borehole as a function of $d = 8$ input variables. For various training-data sizes n and different values of m, we estimated parameters based on the training data using $m_{est} = m$ and made predictions at $n_* = 2{,}000$ unobserved test inputs using $m_{est} = m$; for SVecchia, a training subsample of size $n_{est} = 3{,}000$ was used for estimation if $n > n_{est}$. We computed the resulting RMSE values, averaged over ten datasets.

The results are shown in Figure 3. For scale, the trivial predictor given by the average of the training data had an RMSE around 45, while the exact GP had an RMSE around 0.24 for $n = 100$ and 0.06 for $n = 400$, which was similar to the RMSE for SVecchia with $m = 50$ (0.24 and 0.07). SVecchia outperformed the other approximation methods for every combination of n and m shown in the plots. Note that RMSE is plotted on a log-scale. Thus, for example, for $n = 10{,}000$ and $m = 50$, the seemingly small improvement of SVecchia over Vecchia actually corresponds roughly to a 50% reduction in RMSE. Figure 3a shows that LowRank's accuracy did not improve much with n, and so this method was not considered for the large-n comparisons below. There is a trade-off with the tuning parameter m, which determines the size of the conditioning or neighbor sets: For all methods, increasing m resulted in higher accuracy (Figure 3b), but the computational cost also increases roughly cubically with m (see section 3.5 for a discussion of SVecchia's cost).

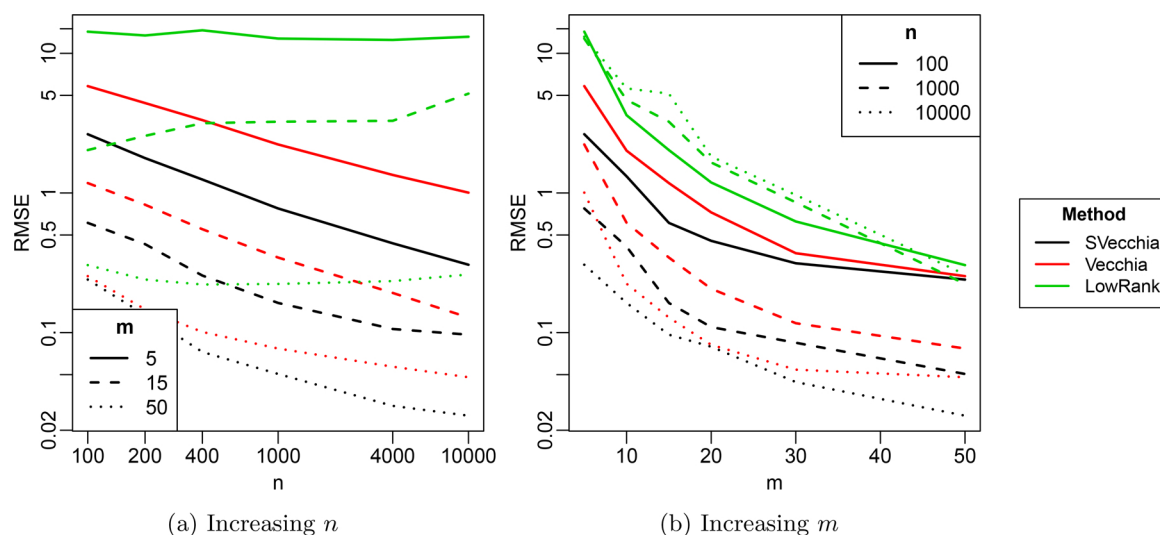(a) Increasing n (b) Increasing m

Figure 3. *Root mean square error (RMSE, on a log scale) for prediction at unobserved inputs using different GP approximations for the borehole example (see section 4.3 for more details).*
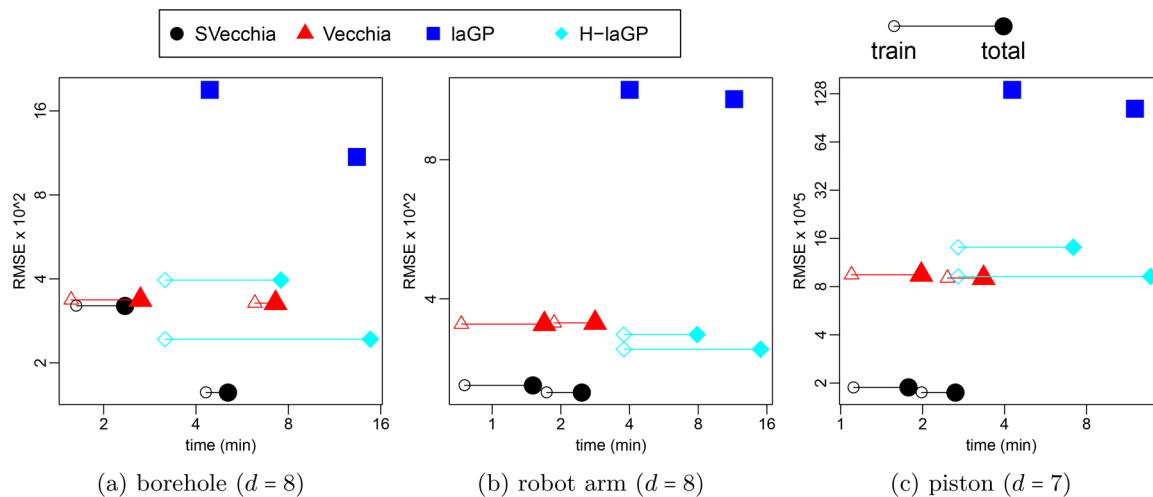
Figure 4. *Comparison of root mean square error (RMSE) versus computing time (both on a log scale) for test functions in section 4.4, with two different tuning-parameter settings for each function and method. We provide training and total (i.e., training plus prediction) times, on a single core for (S)Vecchia and on four cores for (H-)laGP.*

4.4. Test functions.
We then considered larger datasets generated using three physical models from the Virtual Library of Simulation Experiments (Surjanovic and Bingham, 2013), including the borehole function from section 4.3. We generated $n = 100{,}000$ training inputs and $n_* = 20{,}000$ test inputs, and averaged the results over five datasets for each test function. For (S)Vecchia, we used $m_{\mathrm{est}} \in \{30, 50\}$ and a subsample of the training data of size $n_{\mathrm{est}} = 3{,}000$ for parameter estimation, and $m_* = 140$ and all training data for prediction. For (H-)laGP, we used 30 or 50 neighbors for both training and prediction, and we manually specified a much smaller nugget (10^{-7}) than the default value to obtain more accurate predictions. Timing results were obtained on a basic desktop computer (3.4 GHz Intel Quad Core i5-3570), using one core (single-threaded) for (S)Vecchia and using all four cores for (H-)laGP.

The results are summarized in Figure 4, and the detailed numerical results are given in Table 3. For all three test functions, SVecchia was the most accurate, despite having the lowest computational cost. The cost of recomputing the ordering and conditioning sets for SVecchia at certain Fisher-scoring iterations was negligible, as it only took a fraction of a second.

Table 1

Scores evaluating the accuracy of the uncertainty quantification of the predictive distributions for the piston test function. Lower is better for all scores except ICov. All scores except ICov and LogScore were multiplied by 10^5. ICov = empirical coverage of 95% prediction intervals; IWidth = average interval width; IScore = interval score; CRPS = continuous ranked probability score; Energy = energy score.

	ICov (%)	IWidth	IScore	LogScore	CRPS	Energy
SVecchia	95.4	2.9	9.8	-97.8	0.7	182.0
Vecchia	95.4	14.7	51.5	-81.8	3.7	977.0
laGP	81.3	81.3	742.8	-57.7	47.5	
H-laGP	98.5	31.0	70.0	-76.7	6.6	

To assess the accuracy of the uncertainty quantification, we also computed several scores for the predictive distributions and the implied 95% prediction intervals for the piston test function, using the $m_{\text{est}} = 30$ and 30-neighbor setting from above. The interval score, log score, and continuous ranked probability score (CRPS) each simultaneously quantify calibration and sharpness of the marginal predictive distributions (see, e.g., Gneiting and Katzfuss, 2014, for details). The energy score (Gneiting et al., 2008) assesses the quality of the joint predictive distribution at all $n_* = 20{,}000$ test inputs, which were only available for (S)Vecchia. The (S)Vecchia intervals were well calibrated due to variance correction. SVecchia performed by far the best in terms of all other scores.

In general, it is difficult to set a comparison in which all methods are placed on perfectly equal footing. The Vecchia approaches used a subsample for parameter estimation, and the full training set with a larger conditioning-set size m_* for prediction. H-laGP also uses data for global pre-estimation, but in a different manner. Both laGP methods do some estimation on the fly. For all methods, increasing the size of the conditioning or neighbor sets improves the accuracy but also increases the computational cost. Timing results will also depend heavily on a number of other factors, including n_{est}, n_*, implementation, parallelization, and the computing environment. Due to the good parallelization properties of the laGP implementation, laGP prediction times could potentially be pushed below those of single-core SVecchia by using enough cores for laGP.

While the test functions are smooth, deterministic functions without noise, we also tested estimating a noise variance using SVecchia. For the piston function, we artificially added observation noise with variance τ^2, with $\tau = .02$. When including the noise variance as an unknown parameter to estimate in the Fisher scoring algorithm, we obtained a highly accurate estimate of $\hat{\tau} = .0198$, even with a small $m_{\text{est}} = 30$.

4.5. Computer model for satellite drag. Finally, we carried out comparisons using a computer simulator for atmospheric drag coefficients of satellites in low Earth orbit under varying input conditions. A detailed description of the computer model and a previous analysis using state-of-the-art GP emulators can be found in Sun et al. (2019), with data and results available at https://bitbucket.org/gramacylab/tpm/src/master/. In short, we considered simulations of drag coefficients for the Hubble space telescope with $d = 8$ inputs. The simulation runs consist of $n = 2 \times 10^6$ responses for each of six pure chemical species, which can be combined into actual drag coefficients by computing a weighted average of the species.

As in Sun et al. (2019, Sect. 6.1), we carried out 10-fold cross-validation (CV), separately for each of the six species. For the Vecchia-based methods, we used $m_* = 140$ for prediction, and we used $m_{\text{est}} = 30$ and a randomly selected subset of size $n_{\text{est}} = 10{,}000$ for parameter estimation. We also tried estimation using the full dataset (i.e., $n_{\text{est}} = n$) and larger m_{est}, but the increase in predictive accuracy was small relative to the increase in computational cost. The parameter estimates were quite stable between different CV folds. One example of the estimated relevance $1/\hat{\lambda}_l$ is shown in Figure 5b for each input variable x_l and each species; the highest and lowest relevance differed by two to three orders of magnitude, indicating that SVecchia's corresponding scaling of the input dimensions should be useful for emulating this simulator.
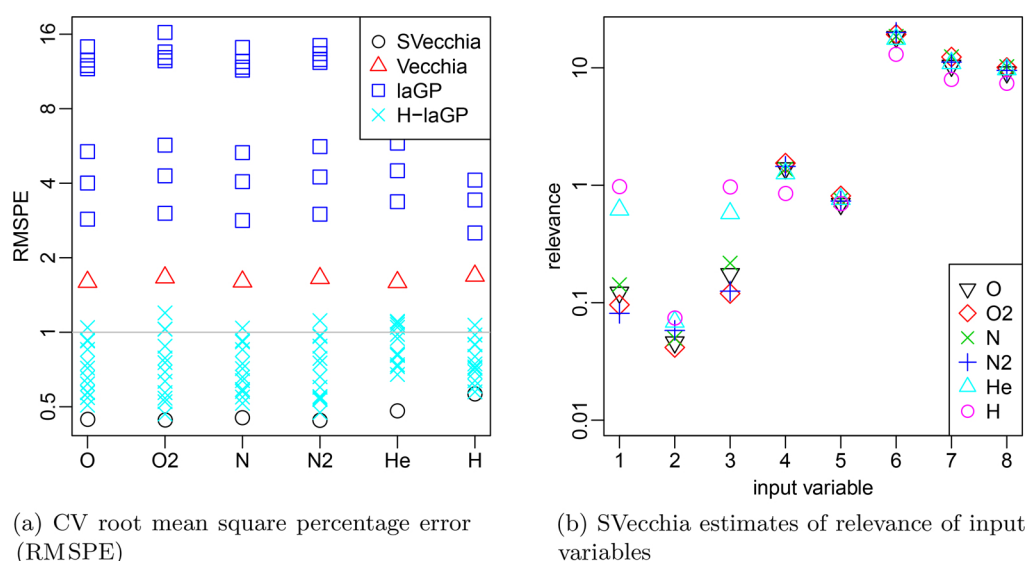
(a) CV root mean square percentage error (RMSPE)

(b) SVecchia estimates of relevance of input variables

Figure 5. *Results for the six chemical species in the satellite-drag simulator. H-laGP: hybrid global-local extensions of laGP. Relevance: $1/\hat{\lambda}_l$ (see details below (3)).*

Figure 5a shows a comparison of CV prediction accuracy in terms of root mean square percentage error (RMSPE). We compared the Vecchia-based methods to the 19 laGP variants considered and described in Sun et al. (2019), seven of which are versions of the basic, local-only laGP, and twelve of which are hybrid global-local laGP (H-laGP) extensions. Vecchia was more accurate than the basic laGP methods, but none of these approaches was able to achieve the standard benchmark of a 1% relative error, indicated by the horizontal line. In contrast, SVecchia met the benchmark and was the most accurate method for all six chemical species. While the accuracy improvement might look small on the log scale of Figure 5a, note that the RMSPE of the best-performing H-laGP method ("alcsep2.sb") in Figure 5a was considerably higher than the SVecchia RMSPE for several species, ranging from roughly 2% higher for H, to around 14% for O and N, up to 40% for He. This is especially remarkable when considering that the total time for estimation and prediction for SVecchia was only around 13 to 14 minutes (4–5 min for estimation and roughly 9 min for prediction) per species and fold, on a single core on a basic desktop computer; the best-performing laGP method took up to 45 core hours according to Sun et al. (2019), which is around 200 times as long.

We also examined predictions along likely trajectories in low Earth orbit, which corresponds to paths in input space. Sun et al. (2019, Sect. 6.2) consider two trajectories, for a quiet and an active regime, each for $n_* = 8{,}600$ ten-second intervals (i.e., about one day). Predictions are made for each of the six pure chemical species, which are then averaged according to weights corresponding to the actual chemical compositions for each of the two regimes. Given estimated parameters, joint prediction using SVecchia scales linearly in n_*, the number of test inputs. Thus, SVecchia can produce joint predictions (e.g., samples from the joint predictive distribution) for the day-long trajectory with $n_* = 8{,}641$ in less than one minute

Table 2

For the O species in the satellite-drag application, scores (same as in Table 1) for joint predictions at 100 pseudo-trajectories of size 100 each. LogScore was multiplied by 10, and all other scores were multiplied by 100. The scores for the other five chemical species were very similar.

	RMSPE	ICov	IWidth	IScore	LogScore	CRPS	Energy
SVecchia	38.3	95.8	1.9	5.2	-32.9	0.5	6.7
Vecchia	137.7	96.8	9.5	25.9	-18.5	2.0	28.7

Table 3

Comparison for test functions in section 4.4. E: root mean square error. Computing times for training + prediction = total, on one core for (S)Vecchia and on four cores for (H-)laGP. Numbers after method names are (m_{est}/m_*) for (S)Vecchia, and (neighborhood size) for (H-)laGP. Smallest errors and computing times are highlighted in red for each test function and for the first and last four rows, respectively.

	Borehole ($d=8$)		Robot arm ($d=8$)		Piston ($d=7$)	
Method	E $\times 10^2$	Time (min)	E $\times 10^2$	Time (min)	E $\times 10^5$	Time (min)
SVecchia (30/140)	3.2	$1.6+0.7=2.3$	2.6	$0.8+0.8=1.6$	1.9	$1.1+0.7=1.8$
Vecchia (30/140)	3.4	$1.6+1.1=2.7$	3.5	$0.7+1.0=1.7$	9.5	$1.1+0.9=2.0$
laGP (30)	19.0	$0+4.4=4.4$	11.3	$0+4.0=4.0$	135.4	$0+4.3=4.3$
H-laGP (30)	4.0	$3.2+4.3=7.5$	3.3	$3.9+4.0=7.9$	14.1	$3.1+4=7.1$
SVecchia (50/140)	1.6	$4.3+0.8=5.1$	2.5	$1.7+0.7=2.4$	1.7	$2+0.7=2.7$
Vecchia (50/140)	3.3	$6.2+1=7.2$	3.5	$1.9+1.0=2.9$	9.1	$2.5+0.9=3.4$
laGP (50)	10.9	$0+13.4=13.4$	10.8	$0+11.5=11.5$	103.2	$0+12.0=12.0$
H-laGP (50)	2.4	$3.1+11.7=14.8$	3.1	$3.7+11.3=15.0$	9.3	$2.3+11.5=13.8$

on a single core; this is less time than it takes the most accurate laGP method ("ALC-ex") to compute predictions for small subsets of size $n_* = 100$. The RMSPE for ALC-ex was about 39% and 8% higher than for SVecchia for the quiet and active regimes, respectively. However, the trajectories traverse only a small fraction of the input space, so that comparing prediction scores for only two such trajectories is not statistically meaningful. Vecchia even happened to have a smaller RMSPE than SVecchia for the active regime.

For a more statistically meaningful comparison, we created 100 pseudo-trajectories, each of which was of size 100 and obtained by randomly selecting one of the n inputs and then sequentially selecting the nearest not-yet-selected input. We computed joint predictions from (S)Vecchia with variance correction at each pseudo-trajectory, using the remaining $n - 100^2$ observations as training data. The resulting scores, averaged over the 100 pseudo-trajectories, are shown in Table 2; SVecchia strongly outperformed Vecchia. We did not have predictions for the (H-)laGP methods from Figure 5a for this experiment; we instead considered a comparison to the (H-)laGP methods from section 4.4, but these methods were not tuned to this satellite-drag application and their scores were not competitive.

5. Conclusions and future work. We have introduced a fast and accurate scaled-Vecchia approximation for Gaussian-process emulation of large computer experiments. The Vecchia approach relies on an ordered conditional approximation, which results in a joint global likelihood and natural joint prediction and uncertainty quantification. Maximin ordering ensures that high accuracy can be achieved by simply conditioning on (previously ordered) nearest neighbors. For the high input dimensions prevalent in computer experiments, our approach

applies the Vecchia approximation in a scaled input space, for which the scaling parameters are automatically determined from the data using a fast parameter-estimation procedure. For fixed conditioning-set sizes, this estimation procedure requires linear time in the number of estimation data, while joint prediction scales linearly in the number of prediction points.

In several numerical comparisons, our proposed method substantially outperformed existing approximations, in that it was able to produce more accurate results in less computational time. For example, for the satellite-drag computer simulator, even a basic version of scaled Vecchia was more accurate and several orders of magnitude faster than the state-of-the-art laGP approaches. As it can produce highly accurate joint predictions with a few lines of code in minutes on modest computers even for big datasets, we consider scaled Vecchia to be a good candidate for a default approach for emulating large computer experiments.

Additional improvements in prediction accuracy may be possible for our method by considering nonstationary covariance functions, such as a Matérn covariance whose parameters vary over input space (Paciorek and Schervish, 2006); ordering and conditioning should then be correlation-based (Kang & Katzfuss, 2021). Such a correlation-based approach would also be possible for joint emulation for multivariate or functional computer-model output.

More sophisticated frequentist variable (i.e., input-dimension) selection could be achieved by adding a lasso-type L1 penalty for the inverse range parameters to (4). MCMC-based Bayesian inference can also be accurately approximated using Vecchia approaches (Finley et al., 2019; Katzfuss and Guinness, 2021, App. E); straightforward extensions include scaling the input space at certain MCMC iterations, and variable selection (Linkletter et al., 2006).

Non-Gaussian computer-model responses could be analyzed by combining scaled Vecchia with the Vecchia-Laplace approximation of generalized GPs (Zilber and Katzfuss, 2021). Finally, it would be interesting to investigate the use and extension of our methods in the context of computer-model calibration (Kennedy and O'Hagan, 2001).

Appendix A. Detailed test-function comparison results. Table 3 provides the specific accuracy and timing results underlying Figure 4, as discussed in section 4.4.

Acknowledgments. We would like to thank Bobby Gramacy and Furong Sun for providing help, data, and laGP results for the application in section 4.5.

REFERENCES

S. BANERJEE, A. E. GELFAND, A. O. FINLEY, AND H. SANG (2008), *Gaussian predictive process models for large spatial data sets*, J. R. Stat. Soc. Ser. B. Stat. Methodol., 70, pp. 825–848.

K. S. BHAT, K. MYERS, E. LAWRENCE, J. COLGAN, AND E. JUDGE (2020), *Estimating Scale Discrepancy in Bayesian Model Calibration for ChemCam on the Mars Curiosity Rover*, preprint, https://arxiv.org/abs/2004.04301.

R. CARNELL (2019), *lhs: Latin Hypercube Samples*, R package version 1.0.1.

A. DATTA, S. BANERJEE, A. O. FINLEY, AND A. E. GELFAND (2016a), *Hierarchical nearest-neighbor Gaussian process models for large geostatistical datasets*, J. Amer. Statist. Assoc., 111, pp. 800–812.

A. DATTA, S. BANERJEE, A. O. FINLEY, N. A. S. HAMM, AND M. SCHAAP (2016b), *Non-separable dynamic nearest-neighbor Gaussian process models for large spatio-temporal data with an application to particulate matter analysis*, Ann. Appl. Stat., 10, pp. 1286–1316.

A. O. FINLEY, A. DATTA, B. C. COOK, D. C. MORTON, H. E. ANDERSEN, AND S. BANERJEE (2019), *Efficient algorithms for Bayesian nearest neighbor Gaussian processes*, J. Comput. Graph. Statist., 28, pp. 401–414.

A. O. Finley, H. Sang, S. Banerjee, and A. E. Gelfand (2009), *Improving the performance of predictive process modeling for large datasets*, Comput. Statist. Data Anal., 53, pp. 2873–2884.

T. Gneiting and M. Katzfuss (2014), *Probabilistic forecasting*, Annu. Rev. Stat. Appl., 1, pp. 125–151.

T. Gneiting, L. I. Stanberry, E. P. Grimit, L. Held, and N. A. Johnson (2008), *Assessing probabilistic forecasts of multivariate quantities, with an application to ensemble predictions of surface winds*, Test, 17, pp. 211–235.

R. B. Gramacy (2016), *LaGP: Large-scale spatial modeling via local approximate Gaussian processes in R*, J. Stat. Softw., 72, pp. 1–46.

R. B. Gramacy and D. W. Apley (2015), *Local Gaussian process approximation for large computer experiments*, J. Comput. Graph. Statist., 24, pp. 561–578.

J. Guinness (2018), *Permutation and grouping methods for sharpening Gaussian process approximations*, Technometrics, 60, pp. 415–429.

J. Guinness (2021), *Gaussian process learning via Fisher scoring of Vecchia's approximation*, Stat. Comput., 31, 25.

J. Guinness and M. Katzfuss (2018), *GpGp: Fast Gaussian Process Computation Using Vecchia's Approximation*, R package version 0.2.2.

M. J. Heaton, A. Datta, A. O. Finley, R. Furrer, J. Guinness, R. Guhaniyogi, F. Gerber, R. B. Gramacy, D. Hammerling, M. Katzfuss, F. Lindgren, D. W. Nychka, F. Sun, and A. Zammit-Mangion (2019), *A case study competition among methods for analyzing large spatial data*, J. Agric. Biol. Environ. Stat., 24, pp. 398–425.

D. Higdon, M. Kennedy, J. C. Cavendish, J. A. Cafeo, and R. D. Ryne (2004), *Combining field data and computer simulations for calibration and prediction*, SIAM J. Sci. Comput., 26: pp. 448–466, https://doi.org/10.1137/S1064827503426693.

D. R. Jones, M. Schonlau, and W. J. Welch (1998), *Efficient global optimization of expensive black-box functions*, J. Global Optim., 13, pp. 455–492.

M. Kang and M. Katzfuss (2021), Correlation-Based Sparse Inverse Cholesky Factorization for Fast Gaussian-Process Inference, arXiv:2112.14591.

M. Katzfuss (2017), *A multi-resolution approximation for massive spatial datasets*, J. Amer. Statist. Assoc., 112, pp. 201–214.

M. Katzfuss and W. Gong (2020), *A class of multi-resolution approximations for large spatial datasets*, Stat. Sinica, 30, pp. 2203–2226.

M. Katzfuss and J. Guinness (2021), *A general framework for Vecchia approximations of Gaussian processes*, Stat. Sci., 36, pp. 124–141.

M. Katzfuss, J. Guinness, W. Gong, and D. Zilber (2020a), *Vecchia approximations of Gaussian-process predictions*, J. Agric. Biol. Environ. Stat., 25, pp. 383–414.

M. Katzfuss, M. Jurek, D. Zilber, W. Gong, J. Guinness, J. Zhang, and F. Schäfer (2020b), *GPvecchia: Fast Gaussian-Process Inference Using Vecchia Approximations*, R package version 0.1.3.

C. G. Kaufman, D. Bingham, S. Habib, K. Heitmann, and J. A. Frieman (2011), *Efficient emulators of computer experiments using compactly supported correlation functions, with an application to cosmology*, Ann. Appl. Stat., 5, pp. 2470–2492.

M. C. Kennedy and A. O'Hagan (2001), *Bayesian calibration of computer models*, J. R. Stat. Soc. Ser. B. Stat. Methodol., 63, pp. 425–464.

E. Lawrence, K. Heitmann, J. Kwan, A. Upadhye, D. Bingham, S. Habib, D. Higdon, A. Pope, H. Finkel, and N. Frontiere (2017), *The Mira-Titan universe. II. Matter power spectrum emulation*, Astrophys. J., 847, p. 50.

C. Linkletter, D. Bingham, N. Hengartner, D. Higdon, and K. Q. Ye (2006), *Variable selection for Gaussian process models in computer experiments*, Technometrics, 48, pp. 478–490.

H. Liu, Y.-S. Ong, X. Shen, and J. Cai (2020), *When Gaussian process meets big data: A review of scalable GPs*, IEEE Trans. Neural Netw. Learn. Syst., 31, pp. 4405–4423.

P. M. Mehta, A. Walker, E. Lawrence, R. Linares, D. Higdon, and J. Koller (2014), *Modeling satellite drag coefficients with response surfaces*, Adv. Space Res., 54, pp. 1590–1607.

J. Mockus (1989), Bayesian Approach to Global Optimization, Kluwer, Dordrecht, the Netherlands.

M. Morris, T. Mitchell, and D. Ylvisaker (1993), *Bayesian design and analysis of computer experiments: Use of derivatives in surface prediction*, Technometrics, 35, pp. 243–255.

C. PACIOREK AND M. SCHERVISH (2006), *Spatial modelling using a new class of nonstationary covariance functions*, Environmetrics, 17, pp. 483–506.

M. PLUMLEE (2014), *Fast prediction of deterministic functions using sparse grid experimental designs*, J. Amer. Statist. Assoc., 109, pp. 1581–1591.

J. QUIÑONERO-CANDELA AND C. E. RASMUSSEN (2005), *A unifying view of sparse approximate Gaussian process regression*, J. Mach. Learn. Res., 6, pp. 1939–1959.

J. SACKS, W. WELCH, T. MITCHELL, AND H. WYNN (1989), *Design and analysis of computer experiments*, Stat. Sci., 4, pp. 409–435.

F. SCHÄFER, M. KATZFUSS, and H. OWHADI (2021a), *Sparse Cholesky factorization by Kullback–Leibler minimization*, SIAM J. Sci. Comput., 43, pp. A2019–A2046, https://doi.org/10.1137/20M1336254.

F. SCHÄFER, T. J. SULLIVAN, and H. OWHADI (2021b), *Compression, inversion, and approximate PCA of dense kernel matrices at near-linear computational complexity*, Multiscale Model. Simul., 19, pp. 688–730, https://doi.org/10.1137/19M129526X.

H. SHI, E. L. KANG, B. A. KONOMI, K. VEMAGANTI, and S. MADIREDDY (2017), *Uncertainty quantification using the nearest neighbor Gaussian process*, in New Advances in Statistics and Data Science, D.-G. CHEN, Z. JIN, G. LI, Y. LI, A. LIU, and Y. ZHAO, eds., Springer, Berlin, pp. 89–107.

E. SNELSON and Z. GHAHRAMANI (2007), *Local and global sparse Gaussian process approximations*, in Proceedings of Artificial Intelligence and Statistics 11 (AISTATS), pp. 524–531.

M. L. STEIN (1999), *Interpolation of Spatial Data: Some Theory for Kriging*, Springer, New York.

M. L. STEIN (2011), 2010 *Rietz lecture: When does the screening effect hold?*, Ann. Statist., 39, pp. 2795–2819.

M. L. STEIN, Z. CHI, AND L. WELTY (2004), *Approximating likelihoods for large spatial data sets*, J. R. Stat. Soc. Ser. B. Stat. Methodol., 66, pp. 275–296.

F. SUN, R. B. GRAMACY, B. HAALAND, E. LAWRENCE, and A. WALKER (2019), *Emulating satellite drag from large simulation experiments*, SIAM/ASA J. Uncertain. Quantif., 7, pp. 720–759, https://doi.org/10.1137/18M1170157.

S. SURJANOVIC AND D. BINGHAM (2013), *Virtual Library of Simulation Experiments: Test Functions and Datasets*, http://www.sfu.ca/~ssurjano.

A. VECCHIA (1988), *Estimation and model identification for continuous spatial processes*, J. R. Stat. Soc. Ser. B. Stat. Methodol., 50, pp. 297–312.

D. J. WALTERS, A. BISWAS, E. C. LAWRENCE, D. C. FRANCOM, D. J. LUSCHER, D. A. FREDENBURG, K. R. MORAN, C. M. SWEENEY, R. L. SANDBERG, J. P. AHRENS, ET AL. (2018), *Bayesian calibration of strength parameters using hydrocode simulations of symmetric impact shock experiments of Al-5083*, J. Appl. Phys., 124, 205105.

B. J. WILLIAMS, J. L. LOEPPKY, L. M. MOORE, AND M. S. MACKLEM (2011), *Batch sequential design to achieve predictive maturity with calibrated computer models*, Reliability Engrg. System Safety, 96, pp. 1208–1219.

D. ZILBER AND M. KATZFUSS (2021), *Vecchia-Laplace approximations of generalized Gaussian processes for big non-Gaussian spatial data*, Comput. Statist. Data Anal., 153, 107081.