

Kernel-Matrix Determinant Estimates from stopped Cholesky Decomposition

Simon Bartels

*University of Copenhagen
Universitetsparken 1
2100 København, Denmark*

BARTELS@DI.KU.DK

Wouter Boomsma

*University of Copenhagen
Universitetsparken 1
2100 København, Denmark*

WB@DI.KU.DK

Jes Frellsen

*Technical University of Denmark
Richard Petersens Plads
2800 Kgs. Lyngby, Denmark*

JEFR@DTU.DK

Damien Garreau

*Université Côte d'Azur, Inria, CNRS, LJAD
Parc Valrose
06108 Nice Cedex 2, France*

DAMIEN.GARREAU@UNICE.FR

Editor: Isabelle Guyon

Abstract

Algorithms involving Gaussian processes or determinantal point processes typically require computing the determinant of a kernel matrix. Frequently, the latter is computed from the Cholesky decomposition, an algorithm of cubic complexity in the size of the matrix. We show that, under mild assumptions, it is possible to estimate the determinant from only a sub-matrix, with probabilistic guarantee on the relative error. We present an augmentation of the Cholesky decomposition that stops under certain conditions before processing the whole matrix. Experiments demonstrate that this can save a considerable amount of time while rarely exceeding an overhead of more than 5% when not stopping early. More generally, we present a probabilistic stopping strategy for the approximation of a sum of known length where addends are revealed sequentially. We do not assume independence between addends, only that they are bounded from below and decrease in conditional expectation.

Keywords: Gaussian Processes, Optimal Stopping, Kernel Methods, Kriging

1. Introduction

Gaussian processes are a popular probabilistic model in the machine learning community, and a core element of many other methods such as Bayesian optimization (Moćkus, 1975), Bayesian quadrature (Diaconis, 1988), probabilistic numerics (Hennig et al., 2015) or the *Automatic Statistician* (Steinruecken et al., 2019). Typically, inference with a Gaussian

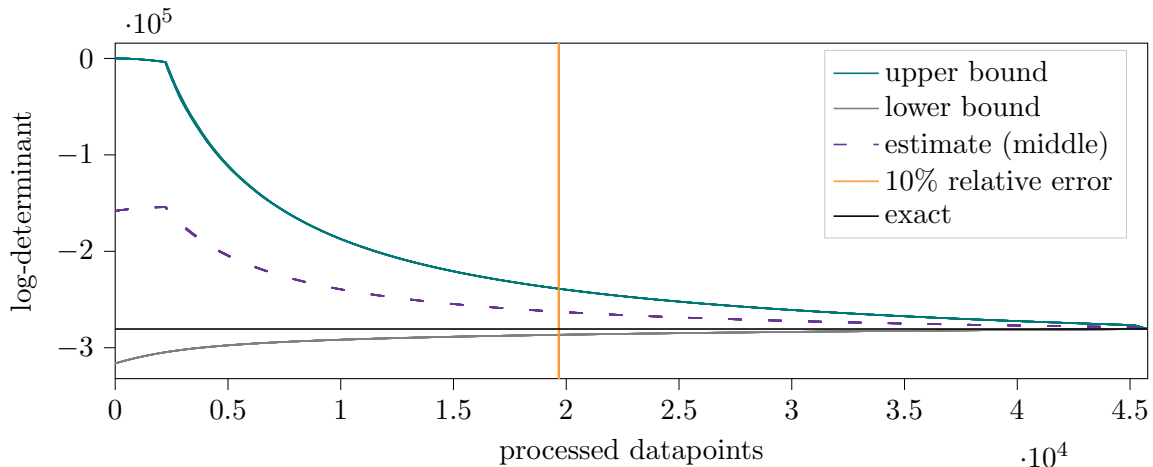


Figure 1: Estimating the log-determinant of a kernel matrix (Section 3). While sequentially processing the dataset, our approach is to monitor upper and lower bounds on the log-determinant, and to stop the computation when they are sufficiently close. In this example, less than half of the dataset is sufficient for the estimate to be within one order of magnitude (10% relative error) of the exact solution. Main strength of our approach: unprocessed datapoints remain untouched—we achieve sub-linear complexity. This speed-gain is obtained in exchange for a certain, controllable failure probability of the bounds, assuming that datapoints are i.i.d.

process requires the computation of a Cholesky decomposition of a kernel matrix. For most datasets, this is computationally feasible despite the cubic worst-case complexity of the Cholesky decomposition in the number of samples. Nevertheless, when this computation has to be performed often, *e.g.*, to optimize kernel parameters, the computational cost of this decomposition becomes paramount.

When a kernel’s parameters do not fit well with the data, our observation is that the log-determinant of the kernel matrix can often be predicted from a subset. This situation frequently occurs in particular at the beginning of the kernel-parameter optimization process. In the following, we will demonstrate that it is possible (i) to recognize this situation while computing the Cholesky decomposition, and (ii) to stop the computation prematurely, which can save a considerable amount of time. When we are not in a situation that justifies stopping the computation early, we propose to simply continue the computation of the log-determinant until the end. Thus the additional computational cost of our method is just that of keeping track of some simple numerical indicators. The main benefit of our method is that it provides an “almost-free lunch” since **the overhead when not stopping early is relatively small** (on average less than five percent). To make this idea practical, we modified the OpenBLAS (Wang et al., 2013) implementation and made our code¹ available.

More generally, we will see that our optional stopping strategy can be used to estimate a sum of random variables that are decreasing in expectation. In this general setting, we

1. https://github.com/SimonBartels/pac_kernel_matrix_determinant_estimation

prove that our stopped Cholesky decomposition returns an estimate of a desired relative error r with respect to the full computation, with probability $1 - \delta$, where δ is a user-defined probability threshold. Under the assumption that the dataset inputs are independent and identically distributed, the probability threshold δ refers to the distribution of the inputs. For a given level of accuracy that is satisfactory for the problem at hand, the user can then pick a level of confidence in the result and obtain a gain in computational cost with provable guarantees. The level of confidence is **the only parameter** of our method.

2. Problem Setup, Related Work and Background

2.1 Problem setup

Given a $\sigma^2 \in \mathbb{R}^+$, a set of inputs $\mathbf{x}_1, \dots, \mathbf{x}_N \in \mathbb{X}$ and a kernel function $k : \mathbb{X} \times \mathbb{X} \rightarrow \mathbb{R}$, we define the *kernel matrix* $\mathbf{A} := \mathbf{K}_N + \sigma^2 \mathbf{I}_N$, where

$$\mathbf{K}_N := \begin{bmatrix} k(\mathbf{x}_1, \mathbf{x}_1) & k(\mathbf{x}_1, \mathbf{x}_2) & \dots & k(\mathbf{x}_1, \mathbf{x}_N) \\ k(\mathbf{x}_2, \mathbf{x}_1) & k(\mathbf{x}_2, \mathbf{x}_2) & & \vdots \\ \vdots & & \ddots & \\ k(\mathbf{x}_N, \mathbf{x}_1) & \dots & & k(\mathbf{x}_N, \mathbf{x}_N) \end{bmatrix}.$$

The main focus of this article is the efficient computation of $\log \det(\mathbf{A})$, which is typically achieved via Cholesky decomposition of \mathbf{A} , if N is not too large. That is, find the unique, lower triangular matrix $\mathbf{L} \in \mathbb{R}^{N \times N}$ satisfying $\mathbf{L}\mathbf{L}^\top = \mathbf{A}$. Given the Cholesky decomposition of \mathbf{A} , one subsequently computes the log-determinant using the formula

$$\log \det(\mathbf{A}) = 2 \sum_{n=1}^N \log \mathbf{L}_{nn}.$$

2.2 Related work

Approximation methods for the log-determinant have been studied extensively—often the more general case of symmetric and positive definite matrices (Skilling, 1989; Seeger, 2000; Dorn and Enßlin, 2015; Ubaru et al., 2017; Fitzsimons et al., 2017a,b; Saibaba et al., 2017; Boutsidis et al., 2017; Dong et al., 2017; Gardner et al., 2018). All of the aforementioned methods are conceptually similar in that they rely on stochastic trace estimators: the kernel matrix is multiplied with random (probe) vectors and the inner products of the results are used to construct an estimate for the log-determinant. The theoretical performance analysis of these methods often requires knowledge or an upper bound on expensive-to-compute quantities such as the largest eigenvalue, the condition number or eigenvalue gaps of \mathbf{A} (Ubaru et al., 2017; Boutsidis et al., 2017; Saibaba et al., 2017; Gardner et al., 2018). An advantage of our approach is that we only require knowledge of the largest diagonal entry on \mathbf{A} and a lower bound on the smallest eigenvalue which is given by σ^2 .

Most related to our work are Ubaru et al. (2017); Boutsidis et al. (2017); Gardner et al. (2018) in the sense that for a desired relative error and confidence, they prove how to set the parameters of their algorithms accordingly. Though, a noteworthy distinction to our work is the choice of the probability measure which the desired confidence refers to. In our case,

this probability measure is the law of the inputs \mathbf{x}_i . For the stochastic trace estimators the confidence refers to the source of randomness of the probe vectors. For the problems we consider in our experiments in Section 5, none of the theorems by Boutsidis et al. (2017); Ubaru et al. (2017); Gardner et al. (2018) that guarantee relative error are applicable. Lemma 8 by Boutsidis et al. (2017) assumes that all eigenvalues are bounded from above by 1. This assumption can be established by dividing \mathbf{A} by $\text{trace}[\mathbf{A}]$, but this would no longer provide a relative approximation error guarantee on $\log \det(\mathbf{A})$. Theorem 4.1 by Ubaru et al. (2017) is not applicable, since the log of the eigenvalues of the kernel matrix can be of different sign. Theorem 2 by Gardner et al. (2018) is a consequence of Theorem 4.1 by Ubaru et al. (2017) and therefore also not applicable. Gardner et al. (2018) recommend certain default parameter values, though we observed experimentally that this configuration yields estimates whose relative errors are *more often than not* worse than 0.1 and may vary over two orders of magnitude (see Fig. 12 in Appendix E). We therefore did not compare our approach to their method. To nevertheless allow the reader to assess the difficulty of the numerical problems considered in Section 5, we compare our method to the pivoted Cholesky decomposition of Harbrecht et al. (2012) (see Section 5.3).

Most related to our Theorem 2 is the work by Mnih et al. (2008) and references therein. They propose an algorithm called *EBStop* that returns an estimate of the mean of a sum of i.i.d. random variables. Theorem 2 is more general and assumes only a (non-strict) decrease in conditional expectation. Their approach is in a sense more sophisticated as they also monitor the empirical variance of the addends, which is also an interesting future direction for our approach.

2.3 Cholesky decomposition

In the following, we will focus on an implementation of the Cholesky decomposition that proceeds *row-wise* over the elements of the matrix, Algorithm 1. As opposed to a column-wise or submatrix implementation, the number of floating operations increases with each iteration of the outer loop (George et al., 1986). Hence, this version can benefit the most from early stopping. Algorithm 1 is useful to express and motivate our idea. To exploit blocking and parallel computation resources requires some modifications which we describe in Appendix A. Note that computing \mathbf{L}_{jj} requires access only to the first $\mathbf{x}_1, \dots, \mathbf{x}_j$ datapoints.

3. Stopped Cholesky Decomposition

This section is a high-level description of our algorithm. The formal proof of our claims is deferred to Section 4 and the supplementary material. The main idea of the algorithm is the following: each time a new diagonal element of the Cholesky decomposition is computed, we compute an upper bound and a lower bound of $\log \det(\mathbf{A})$. If the two bounds are sufficiently close to each other and sufficiently far away from zero, a certain relative error can be guaranteed. We first introduce the bounds used by our algorithm, and then define more precisely what we mean by “close.”

Denote by n the number of diagonal elements that have been computed so far. Define $D_n := 2 \sum_{j=1}^n \log \mathbf{L}_{jj}$ which is the log-determinant of the principal sub-matrix of \mathbf{A} up to index n and observe that $\log \det(\mathbf{A}) = D_N = D_n + 2 \sum_{j=n+1}^N \log \mathbf{L}_{jj}$ for any n . We will use

Algorithm 1 Augmented row-wise Cholesky decomposition with optional stopping. Highlighted are our modifications to the original algorithm. See Algorithm 3 on page 19 for a practical implementation.

Input \mathbf{A} : matrix for which to compute the log-determinant
Input N : matrix size
Input r : desired relative error
Input δ : desired confidence
Input σ^2 : lower bound on $\min_j \mathbf{A}_{jj}$
Input C^+ : upper bound on $\log(\max_j \mathbf{A}_{jj})$

- 1: $D \leftarrow 0, c_\delta \leftarrow (C^+ - \log(\sigma^2))H_N^{-1}(\delta/2)$ constant described in Section 3
- 2: **for** $j = 1, \dots, N$ **do**
- 3: **for** $i = 1, \dots, j - 1$ **do**
- 4: **for** $k = 1, \dots, j - 1$ **do**
- 5: $\mathbf{A}_{ij} \leftarrow \mathbf{A}_{ij} - \mathbf{A}_{ik}\mathbf{A}_{jk}$
- 6: **end for**
- 7: $\mathbf{A}_{ij} \leftarrow \mathbf{A}_{ij}/\mathbf{A}_{jj}$ now $\mathbf{A}_{ij} = \mathbf{L}_{ij}$
- 8: **end for**
- 9: **for** $k = 1, \dots, j - 1$ **do**
- 10: $\mathbf{A}_{jj} \leftarrow \mathbf{A}_{jj} - \mathbf{A}_{jk}\mathbf{A}_{jk}$
- 11: **end for**
- 12: $\mathbf{A}_{jj} \leftarrow \sqrt{\mathbf{A}_{jj}}$ now $\mathbf{A}_{jj} = \mathbf{L}_{jj}$
- 13: $D \leftarrow D + 2 \cdot \log(\mathbf{A}_{jj})$ track log-determinant of the submatrix up to index n
- 14: $\hat{D} \leftarrow \text{EvaluateConditionsAndEstimator}(r, N, n, D, \sigma^2, C^+, c_\delta)$
- 15: **if** $\hat{D} \neq 0$ **then**
- 16: **return** \hat{D} bounds are close enough, return estimate for log-determinant
- 17: **end if**
- 18: **end for**
- 19: **return** D Now the lower-triangular part of \mathbf{A} contains \mathbf{L} .

in the following that for kernel matrices, one can write (Lemma 5)

$$\mathbf{L}_{nn}^2 = k(\mathbf{x}_n, \mathbf{x}_n) + \sigma^2 - \mathbf{k}_n^\top (\mathbf{K}_{n-1} + \sigma^2 \mathbf{I}_{n-1})^{-1} \mathbf{k}_n, \quad (1)$$

where $\mathbf{k}_n^\top := [k(\mathbf{x}_n, \mathbf{x}_1), \dots, k(\mathbf{x}_n, \mathbf{x}_{n-1})]^\top \in \mathbb{R}^{n-1}$, that is: the diagonal elements of the Cholesky, squared, correspond to the posterior variance of a Gaussian process given observations disturbed by Gaussian noise (see Rasmussen and Williams (2006, p. 16)).

Our lower bound \mathcal{L}_n is deterministic. It is simply the sum of log of the elements computed so far plus a linear extrapolation in σ^2 . That is,

$$\mathcal{L}_n := D_n + (N - n) \log \sigma^2,$$

which uses that Eq. (1) is bounded from below by σ^2 .

On the other hand, the upper bound is probabilistic. We show in Section 4 how we can achieve the control of the failure probability. The key observation is that the diagonal elements of the Cholesky **decrease in (conditional) expectation**, under the assumption that $\mathbf{x}_1, \dots, \mathbf{x}_N$ are independent and identically distributed. (This assumption is not always fulfilled, *e.g.* when the inputs are sorted. However, in practice, the assumption can be considered established, after a random shuffle of the dataset.) With increasing n , this variance can only decrease (see Rasmussen and Williams (2006, Question 2.9.4) or the proof of Theorem 7). Thus, the mean of all \mathbf{L}_{nn} is likely to be an overestimate of the expected value of $\mathbf{L}_{n+1, n+1}$. Therefore, we use as an upper bound, the sum of the elements computed so far, plus a linear extrapolation of their mean:

$$\mathcal{U}'_n := D_n + (N - n) \frac{D_n + c_\delta}{n} + c_\delta,$$

where $c_\delta := (\log(\sigma^2 + \max_{j \in \{1, \dots, N\}} k(\mathbf{x}_j, \mathbf{x}_j)) - \log \sigma^2) H_N^{-1}(\delta/2)$ and $H_N(\delta)$ is defined later in Eq. (9). We defer the motivation of c_δ to Section 4. The constant $H_N^{-1}(\delta)$ can be found with negligible overhead, for example using the SCIPY method `MINIMIZE_SCALAR(F, BOUNDS=[0, 1], METHOD='BOUNDED')` where $F(x) = (H_N(x) - H_N(\delta))^2$.

A *deterministic* upper bound to $\log \det(\mathbf{A})$ is

$$\mathcal{U}''_n := D_n + (N - n) \log \left(\sigma^2 + \max_{j \in \{1, \dots, N\}} k(\mathbf{x}_j, \mathbf{x}_j) \right)$$

which is a consequence of Lemma 6. To make sure that our bound is never worse than this deterministic bound we set

$$\mathcal{U}_n := \min(\mathcal{U}'_n, \mathcal{U}''_n). \quad (2)$$

Fig. 1 shows a visualization of our bounds using an RBF kernel (Eq. (13) with $\theta = 1$ and $\ell = \exp(-1)$) for ten random permutation of the TAMILNADU dataset (see Table 1).

Now we are nearly ready to write our algorithm. The only missing piece is to decide whether \mathcal{U}_n and \mathcal{L}_n are close enough. Suppose we believe that $\log \det(\mathbf{A}) \in [\mathcal{L}_n, \mathcal{U}_n]$, then $\hat{D}_n := \frac{1}{2}(\mathcal{U}_n + \mathcal{L}_n)$ is a natural estimate. We will show (Lemma 15) that it is possible to guarantee the *relative error*

$$\left| \frac{\log \det(\mathbf{A}) - \hat{D}_n}{\log \det(\mathbf{A})} \right| \leq r, \quad (3)$$

when $\log \det(\mathbf{A})$ cannot be zero, and

$$\frac{\mathcal{U}_n - \mathcal{L}_n}{2 \min(|\mathcal{U}_n|, |\mathcal{L}_n|)} \leq r. \quad (4)$$

To exclude $\log \det(\mathbf{A}) = 0$, we check in addition that

$$\text{sign}(\mathcal{L}_n) = \text{sign}(\mathcal{U}_n) \neq 0. \quad (5)$$

Algorithm 2 describes above elaborations in pseudo code. Algorithm 1 shows our modifications to a Cholesky decomposition algorithm with new statements highlighted. Importantly, the computation of the bounds and checks are inexpensive in comparison to an outer-loop iteration of the Cholesky decomposition. Figs. 2 and 3 show the left-hand side of Eq. (4) for two examples. Note that when \mathbb{X} is bounded and the kernel is differentiable, with a sufficient amount of data, the upper bound gets arbitrarily close to the lower bound.

Algorithm 2 EvaluateConditionsAndEstimator. At a given step, this routine computes the lower and upper bounds, and proceeds to check if they are close enough.

Input r : desired relative error

Input N : matrix size

Input n : size of the processed subset

Input D_n : sum of the log diagonal elements up to index n

Input σ^2 : lower bound on $\min_j \mathbf{A}_{jj}$

Input C^+ : upper bound on $\log(\max_j \mathbf{A}_{jj})$

Input $c_\delta = H_N^{-1}(\delta/2)(C^+ - \log(\sigma^2))$: constant described in Section 3

- 1: $\mathcal{L}_n \leftarrow D_n + (N - n) \log \sigma^2$ lower bound estimate
 - 2: $\mathcal{U}_n \leftarrow \min(D_n + (N - n) \frac{D_n + c_\delta}{n} + c_\delta, D_n + (N - n) C^+)$ upper bound estimate
 - 3: **if** $\text{sign}(\mathcal{U}_n) = \text{sign}(\mathcal{L}_n) \neq 0$ and $\mathcal{U}_n - \mathcal{L}_n < 2r \min(|\mathcal{U}_n|, |\mathcal{L}_n|)$ **then**
 - 4: **return** $\frac{1}{2}(\mathcal{U}_n + \mathcal{L}_n)$ bounds are close enough, return estimate
 - 5: **end if**
 - 6: **return** 0 bounds are not close enough
-

4. Theoretical Justification

We now turn to the theoretical analysis of our algorithm. Our main goal in this section is to explain how the expressions of the lower and upper bounds are obtained. Note that we consider, in fact, a more general problem: stopping the computation of a sum of random variables that decrease in expectation. To the best of our knowledge, this is the first result obtained in this setting, where the addends are not independent and identically distributed (the \mathbf{x}_i are not the addends). Theorem 2 states that the stopping condition described in the following is a solution to this problem, and Theorem 4 states that Theorem 2 can be applied to estimate determinants of kernel matrices.

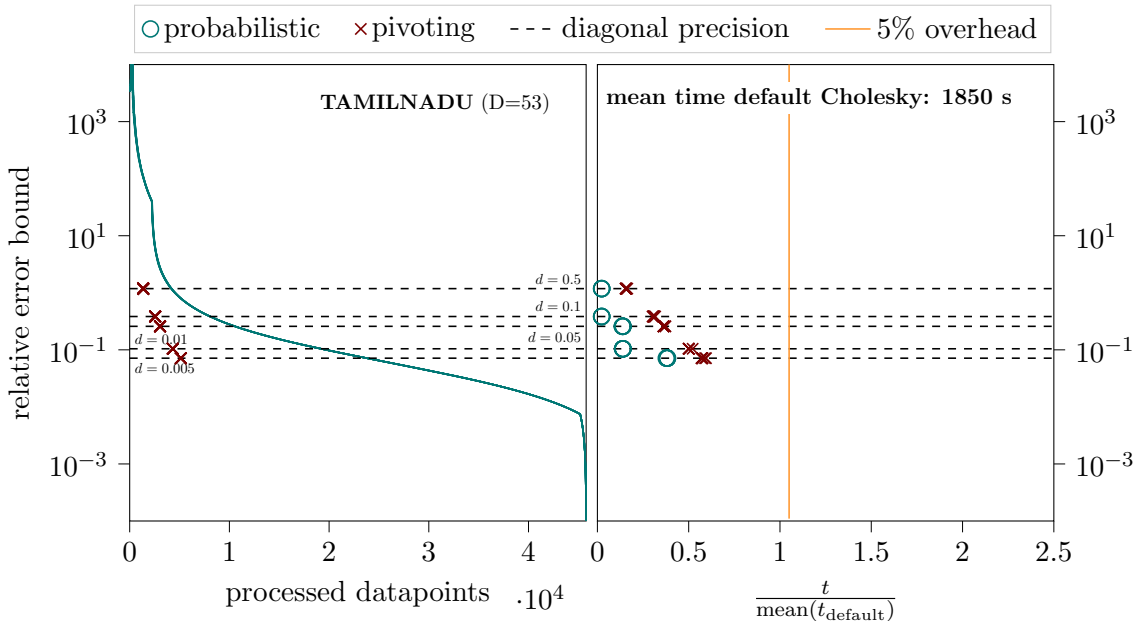


Figure 2: An early stopping scenario. We compute the log-determinant of a kernel matrix using the RBF kernel (with $\theta = 1$, $\ell = \exp(-1)$ in Eq. (13)) for ten random permutations of the TAMILNADU dataset (see Table 1) using our algorithm (\circ) and Cholesky decomposition with pivoting (\times , see Section 5.3).

The y-axis represents the relative error each algorithm can guarantee for (left panel) number of (fully) processed datapoints or (right panel) a given computation time. Even for such a short length-scale ℓ and a desired relative error $r = 0.1$, our algorithm touches only half the dataset before stopping with a confidence of 90% ($\delta = 0.1$) with respect to the data distribution.

Left panel: the solid, green lines show the relative error our algorithm can guarantee. At the same time these lines display our stopping condition Eq. (4). The variance between repetitions is so small such that only one line is visible to the eye. The horizontal lines (---) mark the mean relative error corresponding to an absolute approximation error on the diagonal elements (denoted with d , see Eq. (15)) which is the pivoted Cholesky’s stopping criterion. The singularity in the beginning of the stopping condition stems from the denominator crossing 0 which demonstrates the necessity of the second stopping condition Eq. (5). The reason for the slope changes are switches from the deterministic bound \mathcal{U}_n'' to \mathcal{U}_n' and back in Eq. (2).

Right panel: for each repetition, fraction of both algorithm’s CPU time over the mean time of the default Cholesky. Since S steps of the approximate Cholesky with pivoting cost $\mathcal{O}(NS^2)$ operations, it stops earlier in the left panel, but our algorithm (\circ) scaling as $\mathcal{O}(S^3)$ is faster in practice.

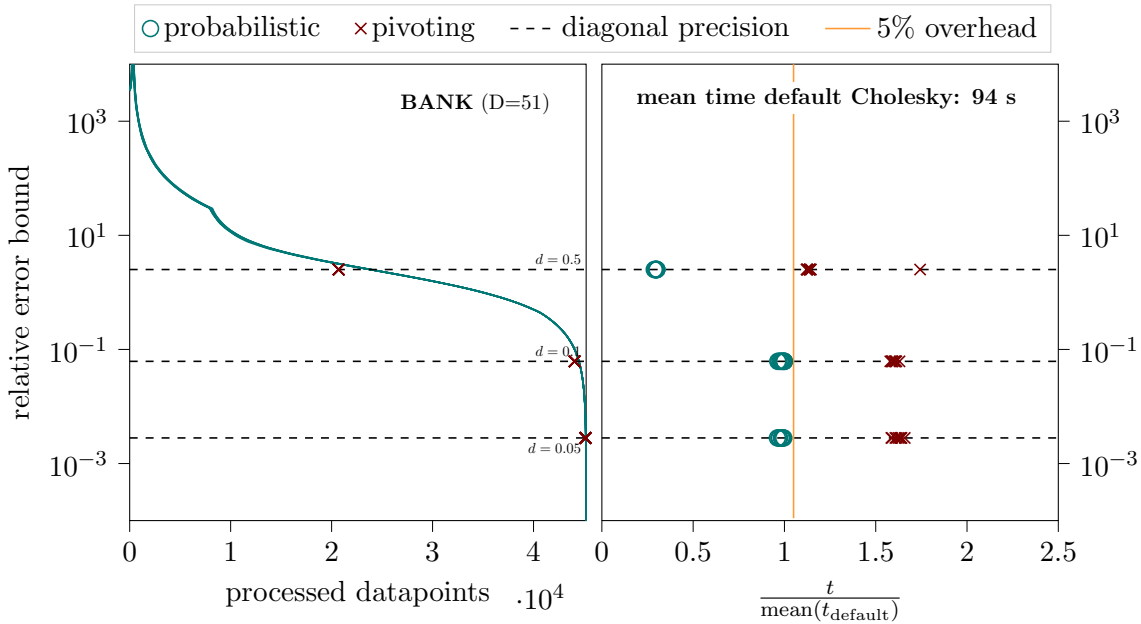


Figure 3: A disadvantageous scenario. We compute the log-determinant of a kernel matrix using the OU kernel ($\theta = 1$, $\ell = \exp(1)$ in Eq. (14)) on the BANK dataset for ten random permutations.

Left panel: same setup as in Fig. 2. On this dataset, even using a long length-scale, requires processing more than 90% of the data to achieve a relative error r of at least 0.1.

Right panel: same setup as in Fig. 2. When our algorithm is not stopping early, that is, it returns the result of the default Cholesky, the overhead is on average less than 5%. The Cholesky with pivoting on the other hand may require more than 150% of the time of the default Cholesky. The extreme difference in absolute runtime between this figure and Fig. 2 is investigated in Section 5.5.

4.1 Notation

Since we are considering an optional stopping problem, we use the terminology of stochastic processes. This section is a quick reminder of the most important concepts, we refer to Grimmett and Stirzaker (2001), and Davidson (1994) for a more thorough introduction. For a monotonically increasing function $f : \mathbb{R} \rightarrow \mathbb{R}$ and $\delta \in \mathbb{R}$, define $f^{-1}(\delta) := \arg \sup_{\varepsilon \in \mathbb{R}} \{f(\varepsilon) \leq \delta\}$. A *filtration* is a sequence $(\mathcal{F}_j)_{j \in \mathbb{N}}$ of increasing σ -algebras, *i.e.*, $\mathcal{F}_j \subseteq \mathcal{F}_{j+1}$ for all $j \in \mathbb{N}$. For random variables X_1, \dots, X_N , we denote by $\sigma(X_1, \dots, X_N)$ the σ -algebra generated by (X_1, \dots, X_N) . A sequence of random variables $(X_j)_{j \in \mathbb{N}}$ is called *adapted* to a filtration, if X_j is \mathcal{F}_j -measurable for all $j \in \mathbb{N}$. A random variable τ is called a *stopping time* (w.r.t. a filtration), if it takes values in \mathbb{N} and $\{\tau = j\} \in \mathcal{F}_j$ for all $j \in \mathbb{N}$.

4.2 Problem Setting

Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space and $(\mathcal{F}_j)_{j \in \{1, \dots, N\}}$ be a filtration. Furthermore, let $(f_j)_{j \in \{1, \dots, N\}} \in [C^-, C^+]$ be a sequence of random variables such that for $j \in \{1, \dots, N-1\}$: f_j is \mathcal{F}_j -measurable and the **conditional expectation is decreasing**, formally:

$$\mathbb{E}[f_{j+1} \mid \mathcal{F}_j] \leq \mathbb{E}[f_j \mid \mathcal{F}_{j-1}], \quad (*)$$

with $\mathcal{F}_0 := \{\emptyset, \mathbb{R}\}$. For this sequence, we want to estimate its sum

$$D_N := \sum_{j=1}^N f_j.$$

Given a desired upper bound on the relative error $r \in (0, 1)$ and a probability of failure $\delta \in (0, 1)$, our goal is to devise a strategy that, being presented sequentially with the f_1, f_2, \dots , decides in each step whether to continue or to stop, and if stopping, provides an estimator \hat{D}_τ , such that its relative error is less than r with probability $1 - \delta$. Formally, the goal is to devise a stopping time τ and an estimator \hat{D}_τ , such that,

$$\mathbb{P} \left(\left| \frac{D_N - \hat{D}_\tau}{D_N} \right| > r \right) \leq \delta.$$

Remark 1 *A trivial solution is to define $\tau := N$ and $\hat{D}_\tau := D_N$, which simply consists in doing the whole computation.*

4.3 Stopping Condition

We now define precisely the quantities introduced in Section 3: the lower bounds \mathcal{L}_n and the upper bounds \mathcal{U}_n . Recall that the lower bounds \mathcal{L}_n are deterministic, whereas $D_N \leq \mathcal{U}_n$ holds only with a certain probability. The stopping time τ will monitor these bounds and stop if they are large in magnitude (away from zero) and close enough that the relative error cannot exceed the desired tolerance $r \in (0, 1)$.

As in Section 3, set

$$\mathcal{L}_n := D_n + (N - n)C^-, \quad (6)$$

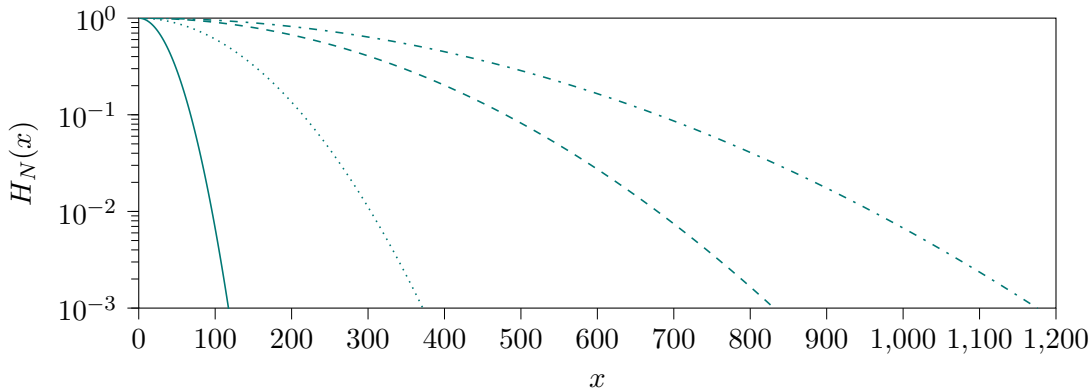


Figure 4: Visualization of the function H_N from Eq. (9). With growing dataset size N , the error guarding constant c_δ grows too, yet at a lower rate. For our experiments in Section 5, we consider datasets of roughly 50000 datapoints and a failure tolerance of $\delta := 0.1/2$, which corresponds to a constant of approximately 547.3.

$$\mathcal{U}_n := D_n + \min \left(c_\delta + (N - n) \frac{D_n + c_\delta}{n}, (N - n)C^+ \right), \quad (7)$$

$$\hat{D}_n := \frac{1}{2}(\mathcal{L}_n + \mathcal{U}_n), \quad (8)$$

where $c_\delta := (C^+ - C^-)H_N^{-1}(\delta/2)$ and

$$H_N(x) := \mathbf{1}_{\{x \leq N\}} \sqrt{\left(\frac{N}{N+x} \right)^{N+x} \left(\frac{N}{N-x} \right)^{N-x}}. \quad (9)$$

The function H_N is derived from a theorem by Fan et al. (2012) which our proofs rely on. Fig. 4 shows visualizations of that function.

Finally, we define the stopping time as

$$\tau = \min\{N\} \cup \{n < N \text{ s.t. } C_n^s \text{ and } C_n^p \text{ hold}\}, \quad (10)$$

where C_n^s is the *sign condition*

$$C_n^s \text{ true if } \text{sign}(\mathcal{U}_n) = \text{sign}(\mathcal{L}_n) \neq 0, \quad (11)$$

and C_n^p is the *relative error condition*

$$C_n^p \text{ true if } \frac{\mathcal{U}_n - \mathcal{L}_n}{2 \min(|\mathcal{U}_n|, |\mathcal{L}_n|)} \leq r. \quad (12)$$

Note that the quantities in the stopping conditions are all \mathcal{F}_n -measurable, thus τ is indeed a stopping time. We can now state our main result.

Theorem 2 *Assume that D_N is a sum of random variables decreasing conditionally in expectation as in Section 4.2. Then, for any $r, \delta \in (0, 1)$, the relative error of the estimator*

\hat{D}_τ defined by Eqs. (6), (7) and (10) to (12) is bounded by r with probability at least $1 - \delta$, formally:

$$\mathbb{P}\left(\left|\frac{D_N - \hat{D}_\tau}{D_N}\right| > r\right) \leq \delta.$$

Intuitively, Theorem 2 guarantees that stopping early in the computation makes sense for any given r and δ . The less precision is required (corresponding to larger r) the easier the second stopping condition in Eq. (12) can be satisfied. The less confidence is necessary (corresponding to larger δ), the smaller the term c_δ in Eq. (7), which also increases chances to satisfy Eq. (12) earlier. On the other hand, when $r = 0$, Eq. (7) can only be true, if upper and lower bounds coincide. The latter can only be the case if $c_\delta = 0$ (requires $\delta = 2$) and $D_n = nC^-$. This means: if we were to desire absolute precision, the theorem would recommend to compute the full sum.

The proof of Theorem 2, and the proof the following lemma are part of the supplementary material. Let us give a sketch of the proof. The design of the stopping condition is based on the following Lemma 3.

Lemma 3 *Let $D \in [\mathcal{L}, \mathcal{U}]$, and assume $\text{sign}(\mathcal{L}) = \text{sign}(\mathcal{U}) \neq 0$. Then*

$$\frac{|D - (\mathcal{U} + \mathcal{L})/2|}{|D|} \leq \frac{\mathcal{U} - \mathcal{L}}{2 \min(|\mathcal{L}|, |\mathcal{U}|)}.$$

The proof of Theorem 2 first bounds $\mathbb{P}\left(\left|\frac{D_N - \hat{D}_\tau}{D_N}\right| > r\right)$ by $\mathbb{P}\left(\left|\frac{D_N - \hat{D}_\tau}{D_N}\right| > r, D_N \leq \mathcal{U}_\tau\right) + \mathbb{P}(D_N > \mathcal{U}_\tau)$. By using Lemma 3 and the stopping conditions, we can show that the probability of the left addend is 0. Finally, we bound $\mathbb{P}(D_N > \mathcal{U}_\tau)$ by applying Fan et al. (2012)'s *Hoeffding's inequality for martingales* twice. Once, to show that D_N is probably not much larger than its expected value, and a second time, to show that \mathcal{U}_τ is probably not much smaller.

4.4 Application to Kernel-Matrix Determinant Estimation

We now specialize Theorem 2 to the situation at hand.

Theorem 4 *Assume $\mathbf{x}_1, \dots, \mathbf{x}_N \in \mathbb{X}$ are independent and identically distributed. Denote with \mathbb{P} the law of the $\mathbf{x}_1, \dots, \mathbf{x}_N$ and with \mathbf{C} the Cholesky decomposition of \mathbf{A} . Define the probability space $(\mathbb{X}, \sigma(\mathbf{x}_1, \dots, \mathbf{x}_N), \mathbb{P})$ and the canonical filtration $\mathcal{F}_j := \sigma(\mathbf{x}_1, \dots, \mathbf{x}_j)$ for $j = 1, \dots, N$. Further, define*

$$\begin{aligned} f_j &:= 2 \log \mathbf{C}_{jj}, \\ C^- &:= \log \sigma^2, \end{aligned}$$

and assume there exists a constant C^+ such that

$$\max_{j=1, \dots, N} \log(k(\mathbf{x}_j, \mathbf{x}_j) + \sigma^2) \leq C^+ \quad \text{almost surely.}$$

Then, using the definitions of Theorem 2,

$$\mathbb{P} \left(\frac{|\log \det(\mathbf{A}) - \hat{D}_\tau|}{|\log \det(\mathbf{A})|} > r \right) \leq \delta.$$

As stated before, the i.i.d. assumption is not too stringent. Finding the deterministic upper bound C^+ is also given in most use-cases, for example when \mathbb{X} is bounded, or when the kernel is normalized or stationary. For instance, $C^+ = \theta$ in the case of the RBF and OU kernels in Eqs. (13) and (14) respectively.

The proof of Theorem 4 is part of the supplementary material. Essentially, to apply Theorem 2 for the estimation of kernel-matrix determinants, one has to show that the summands are decreasing in expectation. As stated before, the key observation is that the diagonal elements of the Cholesky correspond to the posterior variance of a Gaussian process given observations disturbed by independent Gaussian noise. With each observation, the posterior variance can only decrease, which in turn allows to show that the diagonal elements of the Cholesky decrease conditionally in expectation.

5. Experiments

5.1 Intuition on the stopping strategy

One application of our implementation is to probe *bad* kernel parameters quickly. For example, consider the case of a kernel matrix generated from a radial basis function (RBF) kernel

$$k_{RBF}(\mathbf{x}, \mathbf{z}) := \theta \exp \left(-\frac{\|\mathbf{x} - \mathbf{z}\|^2}{2\ell^2} \right) \quad (13)$$

with a lengthscale ℓ far too large with respect to the data. In that case, the diagonal elements of the Cholesky then come quickly close to σ^2 , which implies that upper and lower bounds become close enough to stop the computation earlier. We examine this hypothesis for the RBF on different datasets increasing the length scale exponentially. Furthermore, to also explore the limitations of our approach, we run the same experiments for the Ornstein-Uhlenbeck (OU) kernel

$$k_{OU}(\mathbf{x}, \mathbf{z}) := \theta \exp \left(-\frac{\|\mathbf{x} - \mathbf{z}\|}{\ell} \right). \quad (14)$$

Both kernels are (in the limit) members of the Matérn class of covariance functions (Rasmussen and Williams, 2006, p. 85). Whereas samples from a Gaussian process with RBF covariance are the smoothest in this class, samples from the OU are the roughest. It is therefore not a surprise that our approach is less successful when using the OU kernel. Hence, one advantage of our approach is that one can qualitatively distinguish settings in which stopping occurs earlier or later. Furthermore, the overhead compared to the exact computation is negligible when not stopping.

The other parameters, σ^2 and θ , influence stopping primarily via the error guarding constant $c_\delta \propto \log(\sigma^2 + \theta) - \log \sigma^2$ in Eq. (7) (see Theorem 4). A smaller σ^2 results in a

Key	N	D	Source & URL
BANK	45211	51	Moro et al. (2014) Bank+Marketing
METRO	48204	66	no citation request Metro+Interstate+Traffic+Volume
PM2.5	43824	79	Liang et al. (2015) Beijing+PM2.5+Data
PROTEIN	45730	9	no citation request Physicochemical+Properties+of+Protein+Tertiary+Structure
PUMADYN	8192	32	Snelson and Ghahramani (2006) www.cs.toronto.edu/~delve/data/pumadyn/desc.html
TAMILNADU ²	45781	53	no citation request Tamilnadu+Electricity+Board+Hourly+Readings

Table 1: Overview over all datasets used for the experiments in Section 5. Key refers to the title, we gave a dataset in this article. The letter N refers to the number of instances (training and testing) and D refers to the dimensionality after one-hot encoding. The URL is a suffix for <http://archive.ics.uci.edu/ml/datasets/>. The reference in Source acknowledges a citation request, if any.

smaller deterministic lower-bound \mathcal{L}_n and a larger c_δ , both factors potentially preventing early stopping. On the other hand, we will see that a smaller σ^2 also implies smaller addends $\log \mathbf{L}_{jj}$, such that the exact log-determinant is further away from zero which allows faster stopping. In our case, where $\sigma^2 = 10^{-3}$, $\theta = 1$, $\delta = 0.1$, and dataset size $N \approx 50000$, one can infer with Fig. 4 that c_δ is approximately $7 \cdot 600 = 4200$ which is negligible in comparison to the determinants in the following which are of order 10^7 .

5.2 Experiment setup

From the UCI machine learning repository (Dua and Graff, 2019), we took all multivariate datasets in matrix format with 40000 to 50000 instances without missing values. Furthermore, we included the frequently used PUMADYN dataset (Snelson and Ghahramani, 2006) as a small-scale example of only 8000 instances. Categorical variables were one-hot encoded and each dataset was then standardized. Table 1 provides an overview of all the datasets that we use.

All large-scale experiments (≥ 40000 datapoints) were executed on machines running Ubuntu 18.04 with 32 Gigabytes of RAM and two Intel Xeon E5-2670 v2 CPUs. The experiments for the PUMADYN dataset were run on a laptop running Ubuntu 20.04 with 16 Gigabytes of RAM and an Intel i7-8665U CPU, to demonstrate the usefulness of our approach on more standard hardware. We remark again that we do not use Algorithm 1 but Algorithm 3 in Appendix A which is a more practical implementation capable of exploiting blocking and parallelization.

5.3 Baseline

As described in Section 2.2, none of the referenced stochastic trace estimators can guarantee a desired relative error, and they are therefore not included in the following comparison. The performance for the algorithm by Gardner et al. (2018) is shown in Fig. 12 when using the RBF kernel. As baseline, we compare against the Cholesky decomposition with full pivoting (Harbrecht et al., 2012). Denote with \mathbf{L}_n the $N \times n$ matrix for the intermediate approximation of the Cholesky decomposition in step n . In each step n , the algorithm keeps track of the approximation error of all remaining diagonal elements i —that is how much \mathbf{K}_{ii} differs from $[\mathbf{L}_n \mathbf{L}_n^\top]_{ii}$ —and processes the element inducing the most error next. The algorithm stops when an absolute error tolerance on the diagonal elements, denoted with d , can be guaranteed, that is when

$$d \geq \max_i |\mathbf{K}_{ii} - [\mathbf{L}_n \mathbf{L}_n^\top]_{ii}|. \quad (15)$$

Note that S iterations of the pivoted Cholesky require $\mathcal{O}(NS^2)$ operations whereas Algorithm 1 scales as $\mathcal{O}(S^3)$. For this algorithm, we can set³

$$\mathcal{U}_n^P := D_n + \sum_{j=n+1}^N \log(\mathbf{K}_{jj} - [\mathbf{L}_n \mathbf{L}_n^\top]_{jj}) \quad (16)$$

and $\mathcal{L}_n^P := \mathcal{L}_n$, and apply the same stopping strategy which allows to compare this algorithm with our proposed approach. In the next paragraph, we describe how to compare both algorithms without modifying the Fortran implementation of the Cholesky with pivoting.

5.4 Parameters and performance metric

We set $\sigma^2 := 0.001$ and $\theta := 1$, and increased the lengthscale as $\ell := \exp(i)$ for $i = -1, \dots, 3$. In the appendix we report results where we set $\ell := 1$ and $\theta := 1$, and vary the observation noise as $\sigma^2 := \exp(i)$ for $i = -4, \dots, 0$. The Cholesky decomposition with full pivoting takes as input parameter a desired relative error on the diagonal elements d (instead of a relative error on the log-determinant, see Eq. (15)). We ran this algorithm for $d \in \{0.001, 0.005, 0.01, 0.05, 0.1, 0.5\}$. After the pivoted Cholesky stopped, we computed the relative error on the log-determinant that this algorithm could guarantee in that step. Then we ran our algorithm trying to achieve the same error for $\delta = 0.1$. Occasionally, the desired relative error is larger than 1. In that case, $\hat{D}_N := 0$ is an estimator satisfying this requirement which would allow stopping before even starting. However, we did not check for this condition, to instead observe when the algorithm would stop in such situations. We repeated each configuration for ten random permutations of the dataset. We measured the performance of our method in terms of CPU time t saved over the average CPU time used for the default Cholesky t_{default} :

$$m := \frac{t}{\text{mean}(t_{\text{default}})}.$$

Thus, small values of m are better.

3. One can show that $\mathbf{K}_{jj} - [\mathbf{L}_n \mathbf{L}_n^\top]_{jj}$ is the posterior variance of a GP in \mathbf{x}_j conditioned on the first n datapoints. The claim follows from Rasmussen and Williams (2006, Question 2.9.4).

5.5 Results

As an example, Fig. 5 shows our results for the PM2.5 dataset. For all other datasets, similar figures (Figs. 6 to 9 and 11) can be found in Appendix E. In all experiments, the returned estimate of our modified Cholesky decomposition had indeed the desired error.

For the easy cases, our algorithm needs less than 10% of the average time of the default Cholesky. Here, with easy cases we mean that the relative error can be larger than 0.1 and using an RBF kernel with $\ell \geq \exp(1)$ (there is one exception: the BANK dataset and $\ell = \exp(1)$). The Cholesky decomposition with pivoting also saves time in these settings, yet less. The difference between the algorithms becomes more apparent the harder the problem. Except for three cases, which we will elaborate below, our algorithm needs never longer than 105% of the time of the default Cholesky. In contrast, the Cholesky with pivoting may take more than twice as long.

In three cases our approach crosses the 105% mark: using an RBF kernel with $\ell = 1$ on PM2.5 and METRO, and $\ell = \exp(1)$ on METRO. In these scenarios, the kernel matrix contains many extremely small entries of less than 10^{-65} . Floating-point multiplication is not a constant-time operation and we observed that a large number of such entries significantly prolongs the runtime of our experiments. It is the reason why for $\ell = \exp(-1)$, the run time for the default Cholesky can take up to ten times longer than for larger length-scales. Our row-wise implementation of the Cholesky decomposition suffers *more* from this phenomenon than the original OpenBLAS version. One can circumvent this problem by eliminating such small entries or by increasing the block-size in Algorithm 3. However, we deliberately did not apply these strategies to showcase possible downsides of this implementation. Furthermore, note that the absolute overhead is less than 30s for these three cases and that the effect becomes more negligible the longer the absolute running time. Importantly, the additional run-time does not stem from checking our stopping conditions which can be seen in Fig. 17 where we also compare to the original BLAS implementation endowed with our stopping condition. The take-away is that the row-wise Cholesky implementation allows higher speed gains but may sometimes induce more than 5% overhead. In future work, we aim to investigate middle-grounds between these two Cholesky implementations.

For the experiments where we varied σ^2 the results are similar. The plots can be found in Appendix E.3. The smaller σ^2 , the further is the exact log-determinant away from zero, which allows to stop earlier. An additional problem with larger σ^2 is that our deterministic lower bound is too conservative. This can be seen in Appendix E.5 where we show case the development of the bounds for both experiments. That a better lower bound is conceivable is demonstrated by Artemev et al. (2021); Bartels et al. (2023).

6. Conclusion

6.1 Summary

We presented a stopping strategy for the Cholesky decomposition that allows to obtain estimates for the log-determinant of a kernel matrix of desired error r , before completing the decomposition. The stopping strategy has only one parameter: a failure probability δ . We showed that the returned estimate indeed has the desired error with probability $1 - \delta$, under the mild assumptions that the dataset inputs are independent and identically distributed

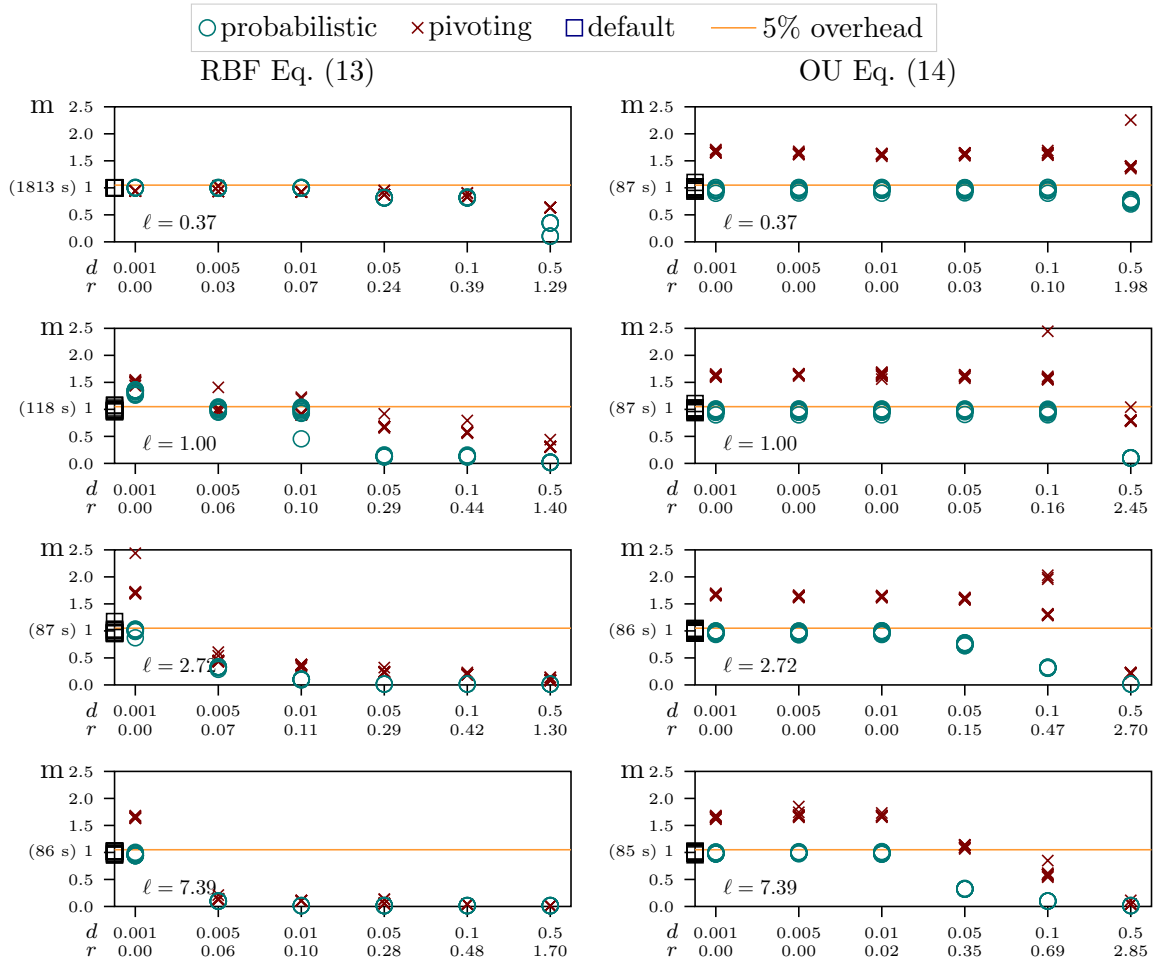


Figure 5: Relative execution times to compute the log-determinant using RBF (**left panel**) and OU (**right panel**) kernels on the PM2.5 dataset for $\theta = 1$, $\log \ell = -1, \dots, 3$ and $\delta = 0.1$ for ten repetitions. The number next to one on the y -axis displays the absolute execution times of the default Cholesky. The solid, horizontal, orange line (—) visualizes the 105% mark. The x -axis displays a desired absolute error on the diagonal elements d (top) and the average corresponding desired relative error r (bottom) on the log-determinant. The longer the length-scale, the earlier it is possible to stop and the higher the speed-up. The speed-up is generally higher for the RBF than for the OU. Even though the Cholesky decomposition with pivoting (\times) needs to compute less diagonal elements (see Figs. 2 and 3) compared with our methods it is slower in practice and may even take more than twice as long as simply running the default Cholesky decomposition (\square). Our method (\circ) on the other hand is faster, and when approximation is hard, the overhead is negligible. One exception can be seen for the RBF kernel and using a length-scale of $\ell = 1$. The reason for this exception is described in Section 5.5.

and a boundedness assumption that is met if the kernel or the domain is bounded. We demonstrated that there exists settings in which it is possible to save considerable amounts of time when stopping the Cholesky decomposition before completion. Importantly, when not stopping early, the induced overhead is less than five percent on average.

As part of their concluding remarks, Chalupka et al. (2013) wrote that

...the results presented above point to the very simple Subset of Data method (or the Hybrid variant) as being the leading contender. We hope this will act as a rallying cry to those working on GP approximations to beat this “dumb” method.

In essence, the presented idea makes a virtue of necessity. Algorithm 2 can be viewed as an estimate for how much data is necessary to identify a kernel model for a particular dataset distribution. The claim that kernel machines do not scale well with large datasets becomes brittle, when the overall dataset size matters little.

6.2 Future work

Early stopping for lower error values r , closer to numerical precision would be desirable. One way to achieve this goal could be to find a less conservative, *probabilistic* lower bound on the log determinant. A direction to investigate are concentration inequalities for self-bounding functions (Boucheron et al., 2013, p. 60). Some concentration inequalities for self-bounding functions allow to reason about the probability of the function falling *below* its expectation. One can show, that the log-determinant of a kernel matrix is such a function (Bartels, 2020, Lemma 33, p. 94).

In the long run, we hope to lift our experiments to hyper-parameter optimization for Gaussian processes. For that, our analysis needs to be extended to the term $\mathbf{y}^\top \mathbf{A}^{-1} \mathbf{y}$. This analysis is similar, but not trivial (Bartels et al., 2023). Furthermore, we will then require an extension of our approach to control the quality of the gradient approximation, since it can be quite different from the accuracy of the function approximation.

Acknowledgments

This work was funded in part by the Novo Nordisk Foundation through the Center for Basic Machine Learning Research in Life Science (NNF20OC0062606). It also received funding from the Danish Ministry of Education and Science, Digital Pilot Hub and Skylab Digital. Damien Garreau acknowledges the support of the French government, through the NIM-ML project (ANR-21-CE23-0005-01).

Simon is grateful for patient listening and fruitful discussions to Gabriele Abbati, Philipp Hennig, Motonobu Kanagawa, Hans Kersting, Jonas Kübler, Simon Lacoste-Julien, Krikamol Muandet, Alexander Neitz, Giambatista Parascandolo, Michaël Perrot, Carl Rasmussen, Luca Rendsburg, Maja Rudolph, Michael Schober, Sebastian Weichwald and Inna Zeitler.

Algorithm 3 Blocked and recursive formulation of Algorithm 1. Highlighted are our modifications to the original algorithm.

Input \mathbf{A} : matrix for which to compute the log-determinant
Input N : matrix size
Input b : block size (number of elements to process in parallel)
Input r : desired relative error
Input δ : desired confidence
Input σ^2 : lower bound on $\min_j \mathbf{A}_{jj}$
Input C^+ : upper bound on $\log(\max_j \mathbf{A}_{jj})$

- 1: $D \leftarrow 0, c_\delta \leftarrow (C^+ - \log(\sigma^2))H_N^{-1}(\delta/2)$ constant described in Section 3
- 2: $\mathbf{A}_{1:b,1:b} \leftarrow \text{chol}(\mathbf{A}_{1:b,1:b})$
- 3: $D \leftarrow D + 2 \cdot \sum_{l=1}^b \log(\mathbf{A}_{ll})$ track log-determinant of the submatrix up to index b
- 4: $\hat{D} \leftarrow \text{EvaluateConditionsAndEstimator}(N, b, D, \sigma^2, C^+, c_\delta)$
- 5: **if** $\hat{D} \neq 0$ **then**
- 6: **return** \hat{D}
- 7: **end if**
- 8: $i \leftarrow b + 1, j \leftarrow \min(i + b, N)$
- 9: **while** $i < N$ **do**
- 10: $\mathbf{A}_{i:j,1:i} \leftarrow \mathbf{A}_{i:j,1:i} \mathbf{A}_{1:i,1:i}^{-\top}$
- 11: $\mathbf{A}_{i:j,i:j} \leftarrow \mathbf{A}_{i:j,i:j} - \mathbf{A}_{i:j,1:i} \mathbf{A}_{1:i,1:i}^\top$
- 12: $\mathbf{A}_{i:j,i:j} \leftarrow \text{chol}(\mathbf{A}_{i:j,i:j})$
- 13: $D \leftarrow D + 2 \cdot \sum_{l=i}^j \log(\mathbf{A}_{ll})$ track log-determinant of the submatrix up to index j
- 14: $\hat{D} \leftarrow \text{EvaluateConditionsAndEstimator}(N, j, D, \sigma^2, C^+, c_\delta)$
- 15: **if** $\hat{D} \neq 0$ **then**
- 16: **return** \hat{D} bounds are close enough, return estimate for log-determinant
- 17: **end if**
- 18: $i \leftarrow i + b, j \leftarrow \min(i + b, N)$
- 19: **end while**
- 20: **return** D Now the lower-triangular part of \mathbf{A} contains \mathbf{L} .

Appendix A. A practical implementation of Cholesky decomposition with stopping

Algorithm 3 is a blocked and recursive version of Algorithm 1. Our OpenBLAS implementation uses the above algorithm with a block size of $b := \#CPUS \cdot \text{BLOCK_SIZE}$, where BLOCK_SIZE is the internal OpenBLAS block size. Furthermore, the call to `chol` is a call to the default OpenBLAS Cholesky. Algorithm 3 is easy to employ in or on top of any library.

Appendix B. Proof of Theorem 4

Proof By Lemma 8: $\log \det(\mathbf{A}) = \sum_{j=1}^N \mathbf{L}_{jj}$, and one can see that the problem already has the right form for (main paper) Theorem 2. To apply the theorem, we need to show that for all $j = 1, \dots, N$, the \mathbf{L}_{jj} are functions of $\mathbf{x}_1, \dots, \mathbf{x}_j$ (Lemma 5), that $f_j := 2 \log \mathbf{L}_{jj} \in [C^-, C^+]$ (Lemma 6), and that $\mathbb{E}[f_{j+1} | \mathcal{F}_j] \leq \mathbb{E}[f_j | \mathcal{F}_{j-1}]$ (Lemma 7). \blacksquare

We now proceed just as in the proof above. We are going to show that the j -th diagonal element of the Cholesky is bounded and can be computed from $\mathbf{x}_1, \dots, \mathbf{x}_j$ only. Then, we conclude that the elements must decrease in (conditional) expectation. To proof the following lemmata, define

$$\begin{aligned} \mathbf{k}_n(\mathbf{x}) &:= [k(\mathbf{x}, \mathbf{x}_1), \dots, k(\mathbf{x}, \mathbf{x}_n)]^\top \in \mathbb{R}^n, \\ \mathbf{k}_{n+1} &:= \mathbf{k}_n(\mathbf{x}_{n+1}) \in \mathbb{R}^n \text{ and} \\ v_n &:= k(\mathbf{x}_n, \mathbf{x}_n) + \sigma^2 - \mathbf{k}_n^\top (\mathbf{K}_{n-1} + \sigma^2 \mathbf{I}_{n-1})^{-1} \mathbf{k}_n. \end{aligned}$$

The first term $\mathbf{k}_n(\mathbf{x})$ denotes the covariance between an arbitrary input \mathbf{x} and the first n datapoints from the dataset. In particular, this definition will be used in the proof of Lemma 7, which states the decrease in expectation. The term v_n is the posterior variance of a Gaussian process f conditioned on observations $\mathbf{y} \in \mathbb{R}^n$, perturbed by Gaussian white noise⁴: $p(\mathbf{y} | \mathbf{f}) = \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I})$. Lemma 5 establishes a link between v_n and the n -th diagonal element of the Cholesky, which is then used in the proof of Lemma 7.

Lemma 5 (Link between the Cholesky and Gaussian process regression) *Denote with \mathbf{L}_N the Cholesky decomposition of $\mathbf{A} := \mathbf{K}_N + \sigma^2 \mathbf{I}_N$, so that $\mathbf{L}_N \mathbf{L}_N^\top = \mathbf{A}$. The n -th diagonal element of \mathbf{L}_N , squared, is equivalent to v_n :*

$$[\mathbf{L}_N]_{nn}^2 = v_n.$$

Proof By a slight abuse of notation, let us define

$$\begin{aligned} \mathbf{L}_1 &:= \sqrt{k(\mathbf{x}_1, \mathbf{x}_1) + \sigma^2}, \quad \text{and} \\ \mathbf{L}_N &:= \begin{bmatrix} \mathbf{C}_{N-1} & \mathbf{0} \\ \mathbf{k}_N^\top \mathbf{L}_{N-1}^{-\top} & \sqrt{v_N} \end{bmatrix}. \end{aligned}$$

We will show that the lower triangular matrix \mathbf{L}_N satisfies $\mathbf{L}_N \mathbf{L}_N^\top = \mathbf{K}_N + \sigma^2 \mathbf{I}_N$. Since the Cholesky decomposition is unique (Golub and Van Loan, 2013, Theorem 4.2.7), \mathbf{L}_N

4. see for example Rasmussen and Williams (2006, p. 16)

must be the Cholesky decomposition of $\mathbf{K}_N + \sigma^2 \mathbf{I}_N$. Furthermore, by definition of \mathbf{L}_N , $[\mathbf{L}_N]_{NN}^2 = v_N$. The statement then follows by the recursive definition of \mathbf{L}_N .

We want to show that $\mathbf{L}_N \mathbf{L}_N^\top = \mathbf{K}_N + \sigma^2 \mathbf{I}_N$. The proof follows by induction. To show the beginning, note that

$$\mathbf{L}_1 \mathbf{L}_1^\top = k(\mathbf{x}_1, \mathbf{x}_1) + \sigma^2 = \mathbf{K}_1 + \sigma^2 \mathbf{I}_1.$$

For the induction step, let us assume that the proposition holds up to $N - 1$, that is, $\mathbf{L}_{N-1} \mathbf{L}_{N-1}^\top = \mathbf{K}_{N-1} + \sigma^2 \mathbf{I}_{N-1}$, then, by definition of \mathbf{L}_N ,

$$\begin{aligned} \mathbf{L}_N \mathbf{L}_N^\top &= \begin{bmatrix} \mathbf{L}_{N-1} & \mathbf{0} \\ \mathbf{k}_N^\top \mathbf{L}_{N-1}^{-\top} & \sqrt{v_N} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{L}_{N-1}^\top & \mathbf{L}_{N-1}^{-1} \mathbf{k}_N \\ \mathbf{0}^\top & \sqrt{v_N} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{L}_{N-1} \mathbf{L}_{N-1}^\top & \mathbf{L}_{N-1} \mathbf{L}_{N-1}^{-1} \mathbf{k}_N \\ \mathbf{k}_N^\top \mathbf{L}_{N-1}^{-\top} \mathbf{L}_{N-1}^\top & \mathbf{k}_N^\top \mathbf{L}_{N-1}^{-\top} \mathbf{L}_{N-1}^{-1} \mathbf{k}_N + v_N \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{K}_{N-1} + \sigma^2 \mathbf{I}_{N-1} & \mathbf{k}_N \\ \mathbf{k}_N^\top & \mathbf{k}_N^\top (\mathbf{K}_{N-1} + \sigma^2 \mathbf{I})^{-1} \mathbf{k}_N + v_N \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{K}_{N-1} + \sigma^2 \mathbf{I}_{N-1} & \mathbf{k}_N \\ \mathbf{k}_N^\top & k(\mathbf{x}_N, \mathbf{x}_N) + \sigma^2 \end{bmatrix}. \end{aligned}$$

■

Lemma 6 (Bounding the f_j s) *Denote by \mathbf{L}_N the Cholesky decomposition of $\mathbf{K}_N + \sigma^2 \mathbf{I}_N$. Define $C^- := \log \sigma^2$ and take $C^+ \geq \max_{j=1, \dots, N} \log(k(\mathbf{x}_j, \mathbf{x}_j) + \sigma^2)$. Then, for all $j \in \{1, \dots, N\}$,*

$$C^- \leq f_j \leq C^+ \quad a.s..$$

Proof By Lemma 5,

$$\mathbf{L}_{nn}^2 = k(\mathbf{x}_n, \mathbf{x}_n) + \sigma^2 - \mathbf{k}_n^\top (\mathbf{K}_{n-1} + \sigma^2 \mathbf{I}_{n-1})^{-1} \mathbf{k}_n.$$

The term $\mathbf{k}_n^\top (\mathbf{K}_{n-1} + \sigma^2 \mathbf{I}_{n-1})^{-1} \mathbf{k}_n$ is always positive since $(\mathbf{K}_{n-1} + \sigma^2 \mathbf{I}_{n-1})^{-1}$ is a symmetric positive definite matrix. Hence, $k(\mathbf{x}_n, \mathbf{x}_n) + \sigma^2$ is an upper bound to \mathbf{L}_{nn}^2 . On the other hand, since k is a kernel, $k(\mathbf{x}_n, \mathbf{x}_n) - \mathbf{k}_n^\top (\mathbf{K}_{n-1} + \sigma^2 \mathbf{I}_{n-1})^{-1} \mathbf{k}_n$ cannot be negative and σ^2 is therefore a lower bound to \mathbf{L}_{nn}^2 . Since both values are positive and the logarithm is an increasing function on the positive real axis, the proof is complete. ■

Equipped with the link between the diagonal elements of the Cholesky and Gaussian process regression stated in Lemma 5, we can now show that the diagonal elements of the Cholesky must decrease in (conditional) expectation, when treating the $\mathbf{x}_1, \dots, \mathbf{x}_N$ as random variables. This follows intuitively from the fact that the posterior variance of a Gaussian process in a fixed location \mathbf{x}_* can only decrease with more observations.

Lemma 7 (The f_j s are decreasing in expectation) *Assume $\mathbf{x}_1, \dots, \mathbf{x}_N \in \mathbb{X}$ are independent and identically distributed. Denote with \mathbb{P} the law of the $\mathbf{x}_1, \dots, \mathbf{x}_N$ and with \mathbf{L}*

the Cholesky decomposition of \mathbf{A} . Define the probability space $(\mathbb{X}, \sigma(\mathbf{x}_1, \dots, \mathbf{x}_N), \mathbb{P})$ and the canonical filtration $\mathcal{F}_j := \sigma(\mathbf{x}_1, \dots, \mathbf{x}_j)$ for $j = 1, \dots, N$. Then the f_j decrease in conditional expectation, that is,

$$\mathbb{E}[f_{j+1} \mid \sigma(\mathbf{x}_1, \dots, \mathbf{x}_j)] \leq \mathbb{E}[f_j \mid \sigma(\mathbf{x}_1, \dots, \mathbf{x}_{j-1})].$$

Proof Denote with $\mathbb{Q}_j(d\mathbf{x}) := \mathbb{P}(d\mathbf{x} \mid \mathbf{x}_1, \dots, \mathbf{x}_j)$, the regular conditional probability. Define the shorthand $q_j(\mathbf{x}) := \mathbf{k}_j(\mathbf{x})^\top (\mathbf{K}_j + \sigma^2 \mathbf{I})^{-1} \mathbf{k}_j(\mathbf{x})$. We will show later in the proof, in Eq. (18), that $q_j(\mathbf{x}) = q_{j-1}(\mathbf{x}) + r_j(\mathbf{x})$ where $r_j(\mathbf{x}) \geq 0$. Taking Eq. (18) as granted for now, we can show the claim as follows.

$$\begin{aligned} \mathbb{E}[f_{j+1} \mid \sigma(\mathbf{x}_1, \dots, \mathbf{x}_j)] &= \mathbb{E}[\log \mathbf{L}_{j+1, j+1}^2 \mid \sigma(\mathbf{x}_1, \dots, \mathbf{x}_j)] \\ &\quad (\text{definition of } f_j) \\ &= \int \log (k(\mathbf{x}, \mathbf{x}) + \sigma^2 - \mathbf{k}_j(\mathbf{x})^\top (\mathbf{K}_j + \sigma^2 \mathbf{I})^{-1} \mathbf{k}_j(\mathbf{x})) \mathbb{Q}_j(d\mathbf{x}) \\ &\quad (\text{property of conditional expectation}) \\ &= \int \log (k(\mathbf{x}, \mathbf{x}) + \sigma^2 - q_j(\mathbf{x})) \mathbb{Q}_j(d\mathbf{x}) \\ &\quad (\text{definition of } q_j(\mathbf{x})) \\ &= \int \log (k(\mathbf{x}, \mathbf{x}) + \sigma^2 - q_{j-1}(\mathbf{x}) - r_j(\mathbf{x})) \mathbb{Q}_j(d\mathbf{x}) \\ &\quad (\text{using Eq. (17)}) \\ &\leq \int \log (k(\mathbf{x}, \mathbf{x}) + \sigma^2 - q_{j-1}(\mathbf{x})) \mathbb{Q}_j(d\mathbf{x}) \\ &\quad (\text{using Eq. (18) and monotonicity of the logarithm}) \\ &= \int \log (k(\mathbf{x}, \mathbf{x}) + \sigma^2 - q_{j-1}(\mathbf{x})) \mathbb{Q}_{j-1}(d\mathbf{x}) \\ &\quad (\text{with Fubini's theorem}) \\ &= \mathbb{E}[\log \mathbf{L}_{jj}^2 \mid \sigma(\mathbf{x}_1, \dots, \mathbf{x}_{j-1})] \\ &\quad (\text{property of conditional expectation}) \\ &= \mathbb{E}[f_j \mid \sigma(\mathbf{x}_1, \dots, \mathbf{x}_{j-1})] \\ &\quad (\text{definition of } f_j) \end{aligned}$$

It remains to show $q_j(\mathbf{x}) = q_{j-1}(\mathbf{x}) + r_j(\mathbf{x})$ where $r_j(\mathbf{x}) \geq 0$. For readability, we define $\mathbf{v}_{\mathbf{x}} := (\mathbf{K}_{j-1} + \sigma^2 \mathbf{I})^{-1} \mathbf{k}_{j-1}(\mathbf{x})$ and $c := v_j^{-1}$. First note, that using block-matrix inversion we can write

$$(\mathbf{K}_j + \sigma^2 \mathbf{I}_j)^{-1} = \begin{bmatrix} (\mathbf{K}_{j-1} + \sigma^2 \mathbf{I}_{j-1})^{-1} + \mathbf{v}_{\mathbf{x}_j} c \mathbf{v}_{\mathbf{x}_j}^\top & -\mathbf{v}_{\mathbf{x}_j} c \\ -\mathbf{v}_{\mathbf{x}_j}^\top c & c \end{bmatrix}.$$

Using above observation, we can transform $q_j(\mathbf{x})$.

$$q_j(\mathbf{x}) = [\mathbf{k}_{j-1}(\mathbf{x})^\top \quad k(\mathbf{x}_j, \mathbf{x})]$$

$$\begin{aligned}
 & \cdot \begin{bmatrix} (\mathbf{K}_{j-1} + \sigma^2 \mathbf{I})^{-1} + \mathbf{v}_{\mathbf{x}_j} \mathbf{c} \mathbf{v}_{\mathbf{x}_j}^\top & -\mathbf{v}_{\mathbf{x}_j} \mathbf{c} \\ -\mathbf{v}_{\mathbf{x}_j}^\top \mathbf{c} & c \end{bmatrix} \\
 & \cdot \begin{bmatrix} \mathbf{k}_{j-1}(\mathbf{x}) \\ k(\mathbf{x}_j, \mathbf{x}) \end{bmatrix} \\
 & \quad (\text{definition of } q_j(\mathbf{x}) \text{ and using above observation}) \\
 = & \begin{bmatrix} \mathbf{k}_{j-1}(\mathbf{x})^\top & k(\mathbf{x}, \mathbf{x}_j) \end{bmatrix} \\
 & \cdot \begin{bmatrix} \mathbf{v}_{\mathbf{x}} + \mathbf{v}_{\mathbf{x}_j} \mathbf{c} \mathbf{v}_{\mathbf{x}_j}^\top \mathbf{k}_{j-1}(\mathbf{x}) - \mathbf{v}_{\mathbf{x}_j} \mathbf{c} k(\mathbf{x}, \mathbf{x}_j) \\ -\mathbf{v}_{\mathbf{x}_j}^\top \mathbf{k}_{j-1}(\mathbf{x}) \mathbf{c} + c k(\mathbf{x}, \mathbf{x}_j) \end{bmatrix} \\
 & \quad (\text{evaluating the RHS matrix-vector multiplication}) \\
 = & \mathbf{k}_{j-1}(\mathbf{x})^\top \mathbf{v}_{\mathbf{x}} + c(\mathbf{v}_{\mathbf{x}_j}^\top \mathbf{k}_{j-1}(\mathbf{x}))^2 \\
 & - 2\mathbf{v}_{\mathbf{x}_j}^\top \mathbf{k}_{j-1}(\mathbf{x}) c k(\mathbf{x}, \mathbf{x}_j) + c k(\mathbf{x}, \mathbf{x}_j)^2 \\
 & \quad (\text{evaluating the vector product}) \\
 = & \mathbf{k}_{j-1}(\mathbf{x})^\top \mathbf{v}_{\mathbf{x}} + c(k(\mathbf{x}, \mathbf{x}_j) - \mathbf{v}_{\mathbf{x}_j}^\top \mathbf{k}_{j-1}(\mathbf{x}))^2 \\
 & \quad (\text{rearranging terms into a quadratic}) \\
 = & q_{j-1}(\mathbf{x}) + c(k(\mathbf{x}, \mathbf{x}_j) - \mathbf{v}_{\mathbf{x}_j}^\top \mathbf{k}_{j-1}(\mathbf{x}))^2 \\
 & \quad (\text{definition of } q_{j-1}(\mathbf{x}))
 \end{aligned}$$

This shows that

$$q_j(\mathbf{x}) = q_{j-1}(\mathbf{x}) + r_j(\mathbf{x}), \text{ where} \quad (17)$$

$$r_j(\mathbf{x}) := c(k(\mathbf{x}, \mathbf{x}_j) - \mathbf{v}_{\mathbf{x}_j}^\top \mathbf{k}_{j-1}(\mathbf{x}))^2 \geq 0. \quad (18)$$

■

The claim of Lemma 8 can for example be found in Rasmussen and Williams (2006, p. 203).

Lemma 8 (Computing the log determinant from the Cholesky decomposition)

Denote with \mathbf{L} the Cholesky decomposition of a symmetric and positive definite matrix \mathbf{A} . Then

$$\log \det(\mathbf{A}) = 2 \sum_{j=1}^N \log L_{jj}.$$

Proof

$$\begin{aligned}
 \log |\mathbf{A}| &= \log |\mathbf{L}\mathbf{L}^\top| \\
 & \quad (\text{using } \mathbf{K} = \mathbf{L}\mathbf{L}^\top) \\
 &= \log(|\mathbf{L}| \cdot |\mathbf{L}^\top|) \\
 & \quad (\text{property of the determinant}) \\
 &= \log(|\mathbf{L}|^2) \\
 & \quad (\text{transposition does not affect the determinant})
 \end{aligned}$$

$$\begin{aligned}
 &= \log \left(\prod_{j=1}^N L_{jj} \right)^2 \\
 &\quad \text{(property of triangular matrices)} \\
 &= 2 \sum_{j=1}^N \log L_{jj} \\
 &\quad \text{(property of the logarithm)}
 \end{aligned}$$

■

Appendix C. Background Material for the Proof of Theorem 2

Before we state the proof of Theorem 2, we provide here the tools that we are going to use.

Our main tool will be the following theorem by Fan et al. (2012). Essentially, this is a generalization of Hoeffding’s inequality to martingales. It states that for a sum of random variables that decrease in (conditional) expectation, the probability of exceeding a certain threshold is low, when at the same time the (conditional) variance is bounded by another constant. Importantly, this probability holds *simultaneously* for all partial sums starting in 1 and ending in $n = 1$ to $n = N$.

Theorem 9 (Hoeffding’s inequality for supermartingales (Fan et al., 2012))

Assume that $(\xi_j, \mathcal{F}_j)_{j=1, \dots, N}$ are supermartingale differences satisfying $\xi_j \leq 1$. Then, for any $x \geq 0$ and $v > 0$,

$$\begin{aligned}
 &\mathbb{P} \left(\text{for some } n \in [1, N] \right. \\
 &\quad \left. \sum_{j=1}^n \xi_j \geq x \text{ and } \sum_{j=1}^n \mathbb{V}[\xi_j \mid \mathcal{F}_{j-1}] \leq v \right) \leq H_N(x, v),
 \end{aligned}$$

where

$$H_N(x, v) := \mathbf{1}_{\{x \leq N\}} \left\{ \left(\frac{v}{v+x} \right)^{v+x} \left(\frac{N}{N-x} \right)^{N-x} \right\}^{\frac{N}{N+v}}.$$

The following theorem will give us the upper bound on the conditional variance, necessary for Theorem 9. The theorem below applies to empirical variance estimates, but the remark thereafter shows that it provides a bound on the true variance as well.

Theorem 10 (Popoviciu’s inequality (Popoviciu, 1935; Sharma et al., 2010)) For a sequence of real numbers $x_1, \dots, x_n \in [m, M]$, define $\mu := \frac{1}{n} \sum_{j=1}^n x_j$ and $\sigma^2 := \frac{1}{n} \sum_{j=1}^n (x_j - \mu)^2$, then

$$\sigma^2 \leq 1/4(M - m)^2.$$

Remark 11 *Theorem 10 can be used to obtain a bound on the conditional variance as well. Let $x_1, \dots, x_n \sim P(\cdot | \mathcal{F})$ be independent. Then,*

$$\begin{aligned} \mathbb{V}[X | \mathcal{F}] &= \mathbb{E}[(X - \mathbb{E}[X | \mathcal{F}])^2 | \mathcal{F}] \\ &\quad \text{(definition of conditional variance)} \\ &= \frac{n}{n-1} \mathbb{E}[\sigma^2 | \mathcal{F}] \\ &\quad \text{(using Bessel's correction)} \\ &\leq \frac{n}{4(n-1)} (M - m)^2 \\ &\quad \text{(by Theorem 10)} \end{aligned}$$

which holds for all $n \in \mathbb{N}$. Hence, $\mathbb{V}[X | \mathcal{F}] \leq 1/4(M - m)^2$.

The martingale differences that we will be analyzing have random indices from our stopping time. Doob's Optional Sampling Theorem (see for example Grimmett and Stirzaker (2001, p. 489)) and the remark below provide us with the mathematical justification.

Theorem 12 (Doob's Optional Sampling Theorem) *Let $(X_j, \mathcal{F}_j)_{j \in \mathbb{N}}$ be a submartingale and $\tau_1 \leq \tau_2 \leq \dots$ be a sequence of stopping times s.t. $P(\tau_j \leq n_j) = 1$ for some deterministic real sequence n_j , then the stopped process $(X_{\tau_j}, \mathcal{F}_{\tau_j})_{j \in \mathbb{N}}$ is also a submartingale.*

Remark 13 *By exchanging X_j for $-X_j$ the theorem can be shown to hold for supermartingales as well.*

Corollary 14 (Stopped submartingale differences) *Let $(\xi_j, \mathcal{F}_j)_{j \in \mathbb{N}}$ be a submartingale-difference and let τ be a stopping time, then the stopped process $(\xi_{\min(j, \tau)}, \mathcal{F}_{\min(j, \tau)})_{j \in \mathbb{N}}$ is also a submartingale-difference.*

Proof Define $X_l := \sum_{j=1}^l \xi_j$ and observe that this defines a submartingale. By Theorem 12 $(X_{\min(j, \tau)}, \mathcal{F}_{\min(j, \tau)})_{j \in \mathbb{N}}$ is a submartingale. Then $X_{\min(j, \tau)} - X_{\min(j, \tau)-1} = \xi_{\min(j, \tau)}$ is again a submartingale-difference. \blacksquare

Appendix D. Proof of Theorem 2

The proof can be split into two parts. Lemma 16 shows by using the stopping conditions that if the bound holds, the relative error of the estimator is indeed less than r with probability 1. The second part is to show that $\mathbb{P}(\mathcal{U}_\tau < D_N) \leq \delta$, which is the purpose of Lemma 17 and which makes use of the assumption stated in Eq. (*).

Proof

$$\begin{aligned} &\mathbb{P}\left(\left|\frac{D_N - \hat{D}_\tau}{D_N}\right| > r\right) \\ &= \mathbb{P}\left(\frac{|D_N - \hat{D}_\tau|}{|D_N|} > r, D_N \leq \mathcal{U}_\tau\right) \end{aligned}$$

$$\begin{aligned}
 & + \mathbb{P} \left(\frac{|D_N - \hat{D}_\tau|}{|D_N|} > r, D_N > \mathcal{U}_\tau \right) \\
 & \quad \text{(sum rule)} \\
 & \leq \mathbb{P} \left(\frac{|D_N - \hat{D}_\tau|}{|D_N|} > r, D_N \leq \mathcal{U}_\tau \right) + \mathbb{P}(D_N > \mathcal{U}_\tau) \\
 & \quad \text{(upper-bounding joint by marginal)} \\
 & \leq 0 + \delta \\
 & \quad \text{(by Lemma 16 and Lemma 17)}
 \end{aligned}$$

■

The following lemma gives an upper bound on the relative error of an estimator in terms of upper and lower bounds for the quantity of interest. The bound is minimized if the estimator is chosen to be the average of upper and lower bound. The lemma can also be found in Mnih (2008) but has been developed independently.

Lemma 15 (Bounding the relative error) *Let $D, \hat{D} \in [\mathcal{L}, \mathcal{U}]$, and assume $\text{sign}(\mathcal{L}) = \text{sign}(\mathcal{U}) \neq 0$. Then the relative error of the estimator \hat{D} can be bounded as*

$$\frac{|D - \hat{D}|}{|D|} \leq \frac{\max(\mathcal{U} - \hat{D}, \hat{D} - \mathcal{L})}{\min(|\mathcal{L}|, |\mathcal{U}|)}.$$

Proof First observe that if $D_N > \hat{D}$ then $|D_N - \hat{D}| = D_N - \hat{D} \leq \mathcal{U} - \hat{D}$. If $D_N \leq \hat{D}$, then $|D_N - \hat{D}| = \hat{D} - D_N \leq \hat{D} - \mathcal{L}$. Hence,

$$|D_N - \hat{D}| \leq \max(\mathcal{U} - \hat{D}, \hat{D} - \mathcal{L}).$$

Case $\mathcal{L} > 0$: In this case $|D_N| = D_N \geq \mathcal{L} = |\mathcal{L}|$, and we obtain for the relative error:

$$\frac{\max(\mathcal{U} - \hat{D}, \hat{D} - \mathcal{L})}{|D_N|} \leq \frac{\max(\mathcal{U} - \hat{D}, \hat{D} - \mathcal{L})}{|\mathcal{L}|}.$$

Case $\mathcal{U} < 0$: In that case $|\mathcal{L}| \geq |D_N| \geq |\mathcal{U}|$, and the relative error can be bounded as follows.

$$\frac{\max(\mathcal{U} - \hat{D}, \hat{D} - \mathcal{L})}{|D_N|} \leq \frac{\max(\mathcal{U} - \hat{D}, \hat{D} - \mathcal{L})}{|\mathcal{U}|}$$

Since we assumed $\text{sign}(\mathcal{L}) = \text{sign}(\mathcal{U})$ these were all cases that required consideration. Combining all observations yields

$$\begin{aligned}
 \frac{|D_N - \hat{D}|}{|D_N|} & \leq \max(\mathcal{U} - \hat{D}, \hat{D} - \mathcal{L}) \max \left(\frac{1}{|\mathcal{U}|}, \frac{1}{|\mathcal{L}|} \right) \\
 & = \frac{\max(\mathcal{U} - \hat{D}, \hat{D} - \mathcal{L})}{\min(|\mathcal{U}|, |\mathcal{L}|)}.
 \end{aligned}$$

■

Lemma 16 (Controlling the relative error when $D_N \leq \mathcal{U}_\tau$) *With the definitions of Section 4.2, the probability that the relative error of the estimator is larger than some $r > 0$ and at the same time the bound holds, is zero. Formally,*

$$\mathbb{P}\left(\frac{|D_N - \hat{D}_\tau|}{|D_N|} > r, D_N \leq \mathcal{U}_\tau\right) = 0.$$

Proof As a preliminary observation note that

$$\begin{aligned} D_N &= \sum_{j=1}^N f_j \\ &\quad \text{(by definition)} \\ &= D_n + \sum_{j=n+1}^N f_j \text{ for all } n = 0, \dots, N \\ &\quad \text{(definition of } D_n) \\ &\geq D_n + \sum_{j=n+1}^N C^- \text{ for all } n = 0, \dots, N \\ &\quad \text{(since } f_j \in [C^-, C^+]) \\ &= \mathcal{L}_n \text{ for all } n = 0, \dots, N \\ &\quad \text{(using the definition of } \mathcal{L}_n) \end{aligned} \tag{19}$$

and hence, for all $n = 0, \dots, N$, \mathcal{L}_n is an almost sure lower bound to D_N .

$$\begin{aligned} &\mathbb{P}\left(\frac{|D_N - \hat{D}_\tau|}{|D_N|} > r, D_N \leq \mathcal{U}_\tau\right) \\ &= \mathbb{P}\left(\frac{|D_N - \hat{D}_\tau|}{|D_N|} > r, D_N \leq \mathcal{U}_\tau, \tau < N\right) \\ &\quad + \mathbb{P}\left(\frac{|D_N - \hat{D}_\tau|}{|D_N|} > r, D_N \leq \mathcal{U}_\tau, \tau = N\right) \end{aligned}$$

Recall that $\hat{D}_\tau = 1/2(\mathcal{L}_\tau + \mathcal{U}_\tau)$. In case $\tau = N$, we have that $\mathcal{U}_N = \mathcal{L}_N = D_N$, and hence, $\mathbb{P}\left(\frac{|D_N - \hat{D}_\tau|}{|D_N|} > r, D_N \leq \mathcal{U}_\tau, \tau = N\right) = 0$.

For brevity, define the event $A := \{D_N \leq \mathcal{U}_\tau, \tau < N\}$, that is, the upper bound holds and the stopping conditions are fulfilled at a time before N .

$$\begin{aligned} &\mathbb{P}\left(\frac{|D_N - \hat{D}_\tau|}{|D_N|} > r, D_N \leq \mathcal{U}_\tau, \tau < N\right) \\ &= \mathbb{P}\left(\frac{|D_N - \hat{D}_\tau|}{|D_N|} > r, A\right) \\ &\quad \text{(definition of } A) \end{aligned}$$

$$\begin{aligned}
 &= \mathbb{P} \left(\frac{|D_N - \hat{D}_\tau|}{|D_N|} > r, \mathcal{L}_\tau \leq D_N, A \right) \\
 &\quad \text{(since } \mathcal{L}_\tau \text{ is an almost sure lower bound to } D_N \text{ by Eq. (19))} \\
 &\leq \mathbb{P} \left(\frac{\max(\mathcal{U}_\tau - \hat{D}_\tau, \hat{D}_\tau - \mathcal{L}_\tau)}{\min(|\mathcal{L}_\tau|, |\mathcal{U}_\tau|)} > r, \mathcal{L}_\tau \leq D_N, A \right) \\
 &\quad \text{(by Lemma 15, using the first condition of } \tau, \text{ Eq. (11))} \\
 &= \mathbb{P} \left(\frac{\mathcal{U}_\tau - \mathcal{L}_\tau}{2 \min(|\mathcal{L}_\tau|, |\mathcal{U}_\tau|)} > r, \mathcal{L}_\tau \leq D_N, A \right) \\
 &\quad \text{(definition of } \hat{D}_\tau) \\
 &= 0 \\
 &\quad \text{(by the second condition of } \tau, \text{ Eq. (12))}
 \end{aligned}$$

■

Lemma 17 (Upper bound control) *With the definitions of Section 4.2, the probability that the upper bound fails is less than δ . Formally,*

$$\mathbb{P}(D_N > \mathcal{U}_\tau) \leq \delta.$$

Proof The following parts of the proof rely on Theorem 9 by Fan et al. (2012). To apply Theorem 9, define $Z'_j := f_j - \mathbb{E}[f_j \mid \mathcal{F}_{j-1}]$ and $Z_j := Z'_{\tau+j}$.

For brevity, we define $\varepsilon := (C^+ - C^-)H_N^{-1}(\delta/2)$, $\varepsilon_n := \varepsilon \left(\frac{1}{N-n} + \frac{1}{n} \right)$, and $\hat{\mu}_n := \frac{D_n}{n} + \varepsilon_n$ such that we can write

$$\mathcal{U}_n = D_n + (N - n) \min(\hat{\mu}_n, C^+). \quad (20)$$

$$\begin{aligned}
 &\mathbb{P}(D_N > \mathcal{U}_\tau) \\
 &= \mathbb{P} \left(D_\tau + \sum_{j=\tau+1}^N f_j > D_\tau + (N - \tau) \min(\hat{\mu}_\tau, C^+) \right) \\
 &\quad \text{(using the definition of } D_n \text{ and Eq. (20))} \\
 &= \mathbb{P} \left(\sum_{j=\tau+1}^N f_j > (N - \tau) \min(\hat{\mu}_\tau, C^+) \right) \\
 &\quad \text{(simplifying)} \\
 &= \mathbb{P} \left(\sum_{j=\tau+1}^N f_j > (N - \tau) \hat{\mu}_\tau \text{ or } \sum_{j=\tau+1}^N f_j > (N - \tau) C^+ \right) \\
 &\quad \text{(exchanging min for logical or)}
 \end{aligned}$$

$$\begin{aligned}
 &= \mathbb{P} \left(\sum_{j=\tau+1}^N f_j > (N - \tau) \hat{\mu}_\tau \right) \\
 &\quad \text{(since } f_j \leq C^+ \text{)} \\
 &= \mathbb{P} \left(\sum_{j=1}^{N-\tau} [Z_j + \mathbb{E}[f_{\tau+j} \mid \mathcal{F}_{\tau+j-1}]] > (N - \tau) \hat{\mu}_\tau \right) \\
 &\quad \text{(definition of } Z_j \text{)} \\
 &\leq \mathbb{P} \left(\sum_{j=1}^{N-\tau} Z_j + \sum_{j=\tau+1}^N \mathbb{E}[f_j \mid \mathcal{F}_{j-1}] > (N - \tau) \hat{\mu}_\tau, \right. \\
 &\quad \left. \sum_{j=\tau+1}^N \mathbb{E}[f_j \mid \mathcal{F}_{j-1}] \leq \frac{N - \tau}{\tau} (D_\tau + \varepsilon) \right) \\
 &+ \mathbb{P} \left(\sum_{j=\tau+1}^N \mathbb{E}[f_j \mid \mathcal{F}_{j-1}] > \frac{N - \tau}{\tau} (D_\tau + \varepsilon) \right) \\
 &\quad \text{(sum rule and upper-bounding joint by marginal)}
 \end{aligned} \tag{21}$$

Consider the first addend in Eq. (21).

$$\begin{aligned}
 &\mathbb{P} \left(\sum_{j=1}^{N-\tau} Z_j + \sum_{j=\tau+1}^N \mathbb{E}[f_j \mid \mathcal{F}_{j-1}] > (N - \tau) \hat{\mu}_\tau, \right. \\
 &\quad \left. \sum_{j=\tau+1}^N \mathbb{E}[f_j \mid \mathcal{F}_{j-1}] \leq \frac{N - \tau}{\tau} (D_\tau + \varepsilon) \right) \\
 &\leq \mathbb{P} \left(\sum_{j=1}^{N-\tau} Z_j + \frac{N - \tau}{\tau} (D_\tau + \varepsilon) > (N - \tau) \hat{\mu}_\tau \right) \\
 &\quad \text{(combining the two events)} \\
 &= \mathbb{P} \left(\sum_{j=1}^{N-\tau} Z_j + \frac{N - \tau}{\tau} (D_\tau + \varepsilon) > \right. \\
 &\quad \left. (N - \tau) \left(\frac{D_\tau}{\tau} + \varepsilon \left(\frac{1}{N - \tau} + \frac{1}{\tau} \right) \right) \right) \\
 &\quad \text{(definition of } \hat{\mu}_\tau \text{ and } \varepsilon_n \text{)} \\
 &= \mathbb{P} \left(\sum_{j=1}^{N-\tau} Z_j > \varepsilon \right) \\
 &\quad \text{(simplifying)}
 \end{aligned}$$

$$\begin{aligned}
 &= \mathbb{P} \left(\sum_{j=1}^{N-\tau} \frac{Z_j}{C^+ - C^-} > H_N^{-1}(\delta/2) \right) \\
 &\quad \text{(definition of } \varepsilon \text{ and dividing by } C^+ - C^- \text{)} \\
 &\leq \mathbb{P} \left(\sum_{j=1}^n \frac{Z_j}{C^+ - C^-} > H_N^{-1}(\delta/2) \right. \\
 &\quad \left. \text{for some } n \in \{1, \dots, N\} \right) \\
 &\quad \text{(enlarging the event)}
 \end{aligned} \tag{22}$$

We are now ready to use Theorem 9. Since $(Z'_j, \mathcal{F}_j)_{j \in \{1, \dots, N\}}$ is a martingale difference,

$$(Z_{\min(j, N)}, \mathcal{F}_{\min(\tau+j, N)})_{j \in \mathbb{N}_0}$$

is a martingale difference as well (Corollary 14). Further note, that the random variables $\frac{Z_j}{C^+ - C^-}$ are bounded from above by 1. Hence, there is only one ingredient missing to apply Theorem 9, which is a bound on the conditional variance. To this end, we use Popoviciu's inequality. The latter is applicable, since the $\frac{Z_j}{C^+ - C^-}$ are also bounded from below by -1 .

$$\begin{aligned}
 &\mathbb{P} \left(\sum_{j=1}^n \frac{Z_j}{C^+ - C^-} > H_N^{-1}(\delta/2) \text{ for some } n \in \{1, \dots, N\} \right) \\
 &= \mathbb{P} \left(\sum_{j=1}^n \frac{Z_j}{C^+ - C^-} > H_N^{-1}(\delta/2), \right. \\
 &\quad \left. \sum_{j=1}^n \mathbb{V} \left[\frac{Z_j}{C^+ - C^-} \middle| \mathcal{F}_{j-1} \right] \leq N \right. \\
 &\quad \left. \text{for some } n \in \{1, \dots, N\} \right) \\
 &\quad \text{(by Popoviciu's inequality (Theorem 10))} \\
 &\leq H(H_N^{-1}(\delta/2), N) \\
 &\quad \text{(by Theorem 9, where } H \text{ is defined in that theorem)} \\
 &= H_N(H_N^{-1}(\delta/2)) \leq \delta/2. \\
 &\quad \text{(definition of } H_N \text{)}
 \end{aligned}$$

Now we will take care of the second addend in Eq. (21), using the assumption that the f_j decrease in expectation: Eq. (*). We will again apply Theorem 9.

$$\mathbb{P} \left(\sum_{j=\tau+1}^N \mathbb{E}[f_j | \mathcal{F}_{j-1}] > \frac{N-\tau}{\tau} (D_\tau + \varepsilon) \right)$$

$$\begin{aligned}
 &\leq \mathbb{P} \left(\sum_{j=\tau+1}^N \mathbb{E}[f_{\tau+1} \mid \mathcal{F}_\tau] > \frac{N-\tau}{\tau} (D_\tau + \varepsilon) \right) \\
 &\quad \text{(using Eq. (*))} \\
 &= \mathbb{P} (\tau \mathbb{E}[l_{\tau+1} \mid \mathcal{F}_\tau] > D_\tau + \varepsilon) \\
 &\quad \text{(dividing by } N - \tau \text{ and multiplying by } \tau) \\
 &= \mathbb{P} \left(\sum_{j=1}^{\tau} (\mathbb{E}[l_{\tau+1} \mid \mathcal{F}_\tau] - f_j) > \varepsilon \right) \\
 &\quad \text{(definition of } D_\tau) \\
 &\leq \mathbb{P} \left(\sum_{j=1}^{\tau} (\mathbb{E}[l_{j+1} \mid \mathcal{F}_j] - f_j) > \varepsilon \right) \\
 &\quad \text{(using again Eq. (*))} \\
 &= \mathbb{P} \left(\sum_{j=1}^{\tau} \frac{\mathbb{E}[l_{j+1} \mid \mathcal{F}_j] - f_j}{C^+ - C^-} > H_N^{-1} \left(\frac{\delta}{2} \right) \right) \\
 &\quad \text{(definition of } \varepsilon \text{ and dividing by } C^+ - C^-) \\
 &= \mathbb{P} \left(\sum_{j=1}^{\tau} -\frac{Z'_j}{C^+ - C^-} > H_N^{-1} \left(\frac{\delta}{2} \right) \right) \\
 &\quad \text{(definition of } Z'_j)
 \end{aligned}$$

Changing the sign does not change the martingale difference property and hence, $(-Z'_j, \mathcal{F}_j)_{j \in \{1, \dots, N\}}$ is a martingale difference as well. We can apply the same argument as in Eq. (22).

$$\begin{aligned}
 &\mathbb{P} \left(\sum_{j=1}^n -\frac{Z'_j}{C^+ - C^-} > H_N^{-1} \left(\frac{\delta}{2} \right) \right) \\
 &\leq H \left(H_N^{-1} \left(\frac{\delta}{2} \right), N \right) \\
 &\quad \text{(using the same argument as in Eq. (22))} \\
 &\leq \frac{\delta}{2}
 \end{aligned}$$

■

Appendix E. Results

This section contains the complete results from the experiments described in Section 5. In Appendix E.1 we show the results for varying lengthscale ℓ in Figs. 6 to 11. Considering the same datasets, Fig. 12 in Appendix E.2 shows the relative error when using the method of Gardner et al. (2018) with default parameters for the RBF kernel. Appendix E.3 shows the results for fixed lengthscale and varying observation noise in Figs. 13 to 17. Appendix E.5 shows with Figs. 23 to 27 how varying the noise affects the evolution of the bounds.

E.1 Varying length-scale

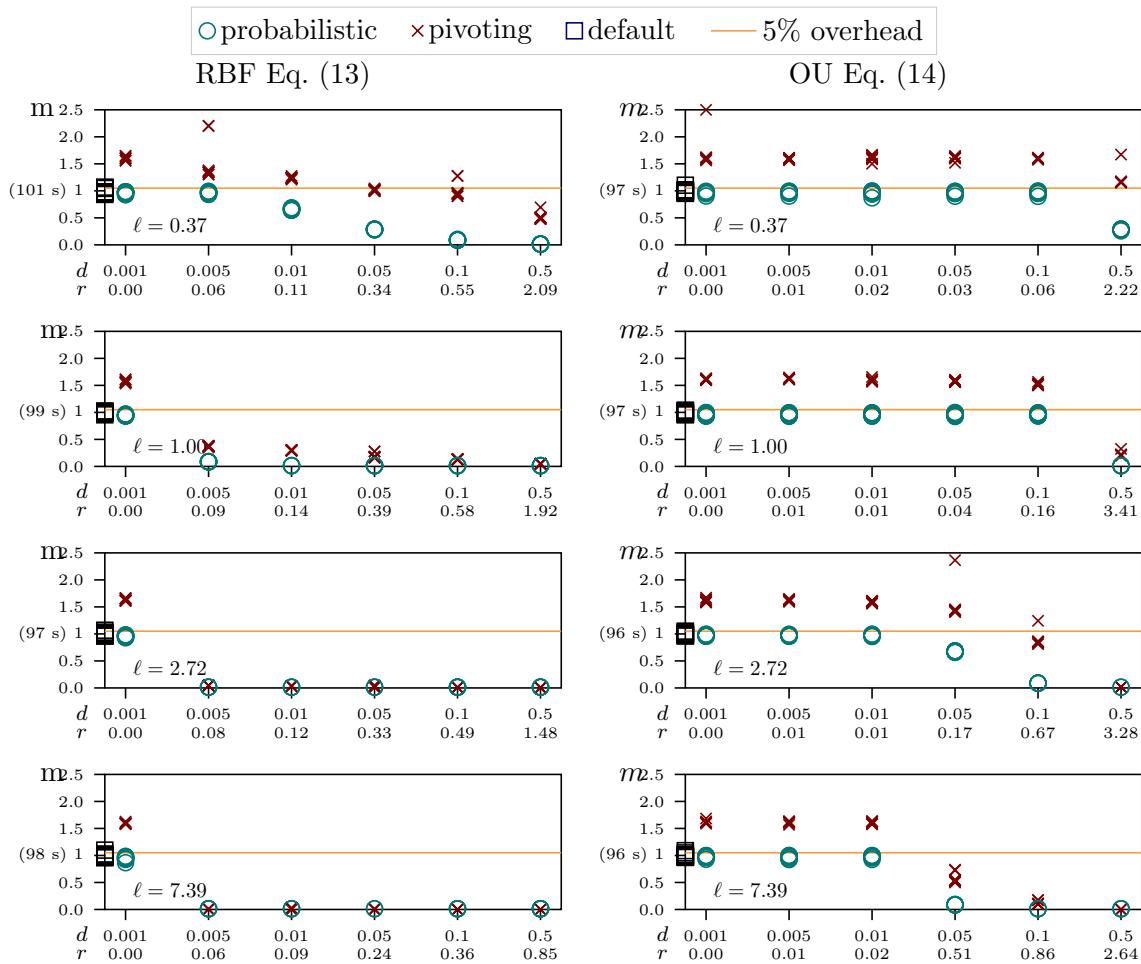


Figure 6: Relative execution times to compute the log-determinant using RBF (**left panel**) and OU (**right panel**) kernels on the PROTEIN dataset for $\theta = 1$, $\log \ell = -1, \dots, 3$ and $\delta = 0.1$ for ten repetitions. The number next to one on the y -axis displays the absolute execution times of the default Cholesky. The solid, horizontal, orange line (—) visualizes the 105% mark. The x -axis displays a desired absolute precision on the diagonal elements d (top) and the average corresponding desired relative precision r (bottom) on the log-determinant.

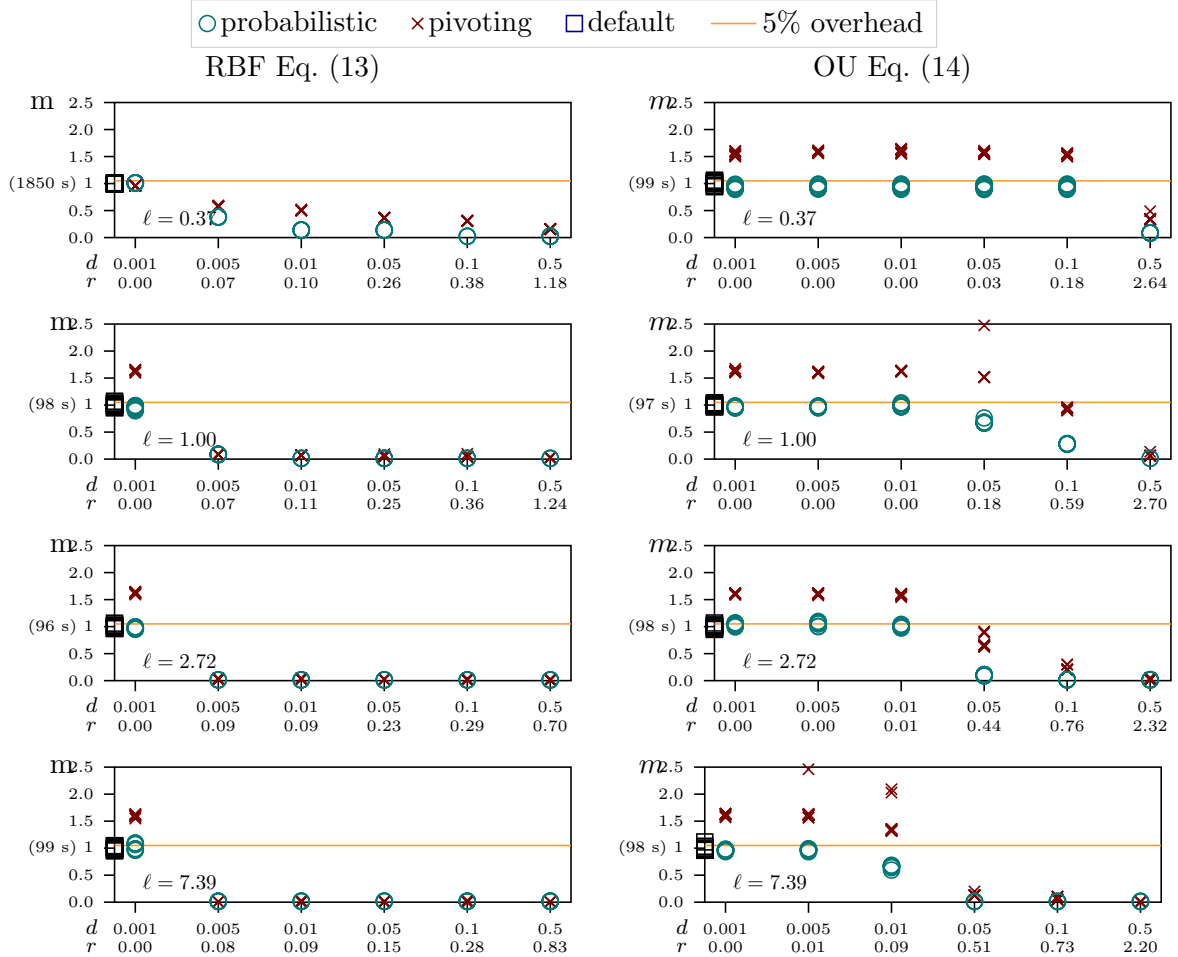


Figure 7: Relative execution times to compute the log-determinant using RBF (**left panel**) and OU (**right panel**) kernels on the TAMILNADU dataset for $\theta = 1$, $\log \ell = -1, \dots, 3$ and $\delta = 0.1$ for ten repetitions. The number next to one on the y -axis displays the absolute execution times of the default Cholesky. The solid, horizontal, orange line (—) visualizes the 105% mark. The x -axis displays a desired absolute precision on the diagonal elements d (top) and the average corresponding desired relative precision r (bottom) on the log-determinant.

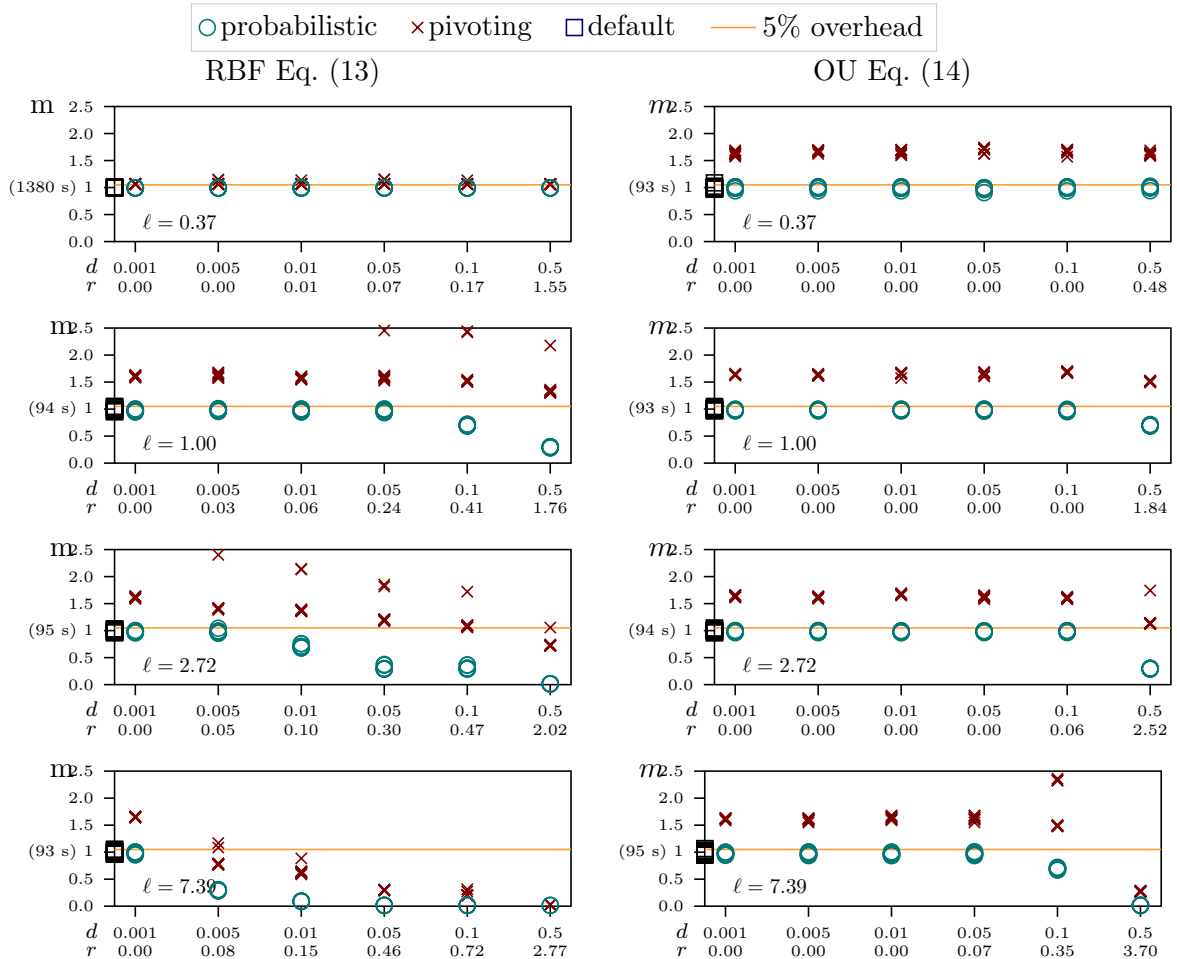


Figure 8: Relative execution times to compute the log-determinant using RBF (**left panel**) and OU (**right panel**) kernels on the BANK dataset for $\theta = 1$, $\log \ell = -1, \dots, 3$ and $\delta = 0.1$ for ten repetitions. The number next to one on the y -axis displays the absolute execution times of the default Cholesky. The solid, horizontal, orange line (—) visualizes the 105% mark. The x -axis displays a desired absolute precision on the diagonal elements d (top) and the average corresponding desired relative precision r (bottom) on the log-determinant.

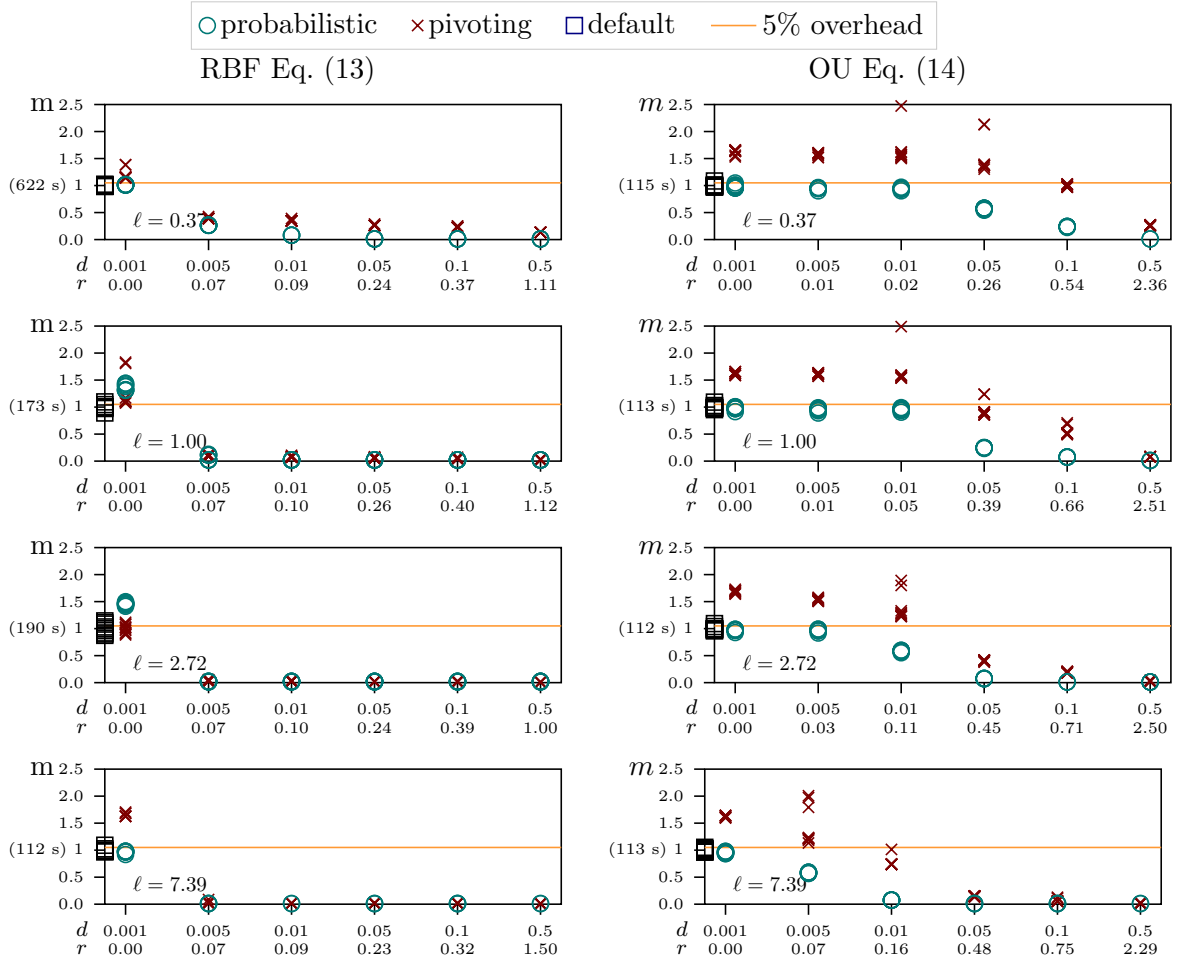


Figure 9: Relative execution times to compute the log-determinant using RBF (**left panel**) and OU (**right panel**) kernels on the METRO dataset for $\theta = 1$, $\log \ell = -1, \dots, 3$ and $\delta = 0.1$ for ten repetitions. The number next to one on the y -axis displays the absolute execution times of the default Cholesky. The solid, horizontal, orange line (—) visualizes the 105% mark. The x -axis displays a desired absolute precision on the diagonal elements d (top) and the average corresponding desired relative precision r (bottom) on the log-determinant.

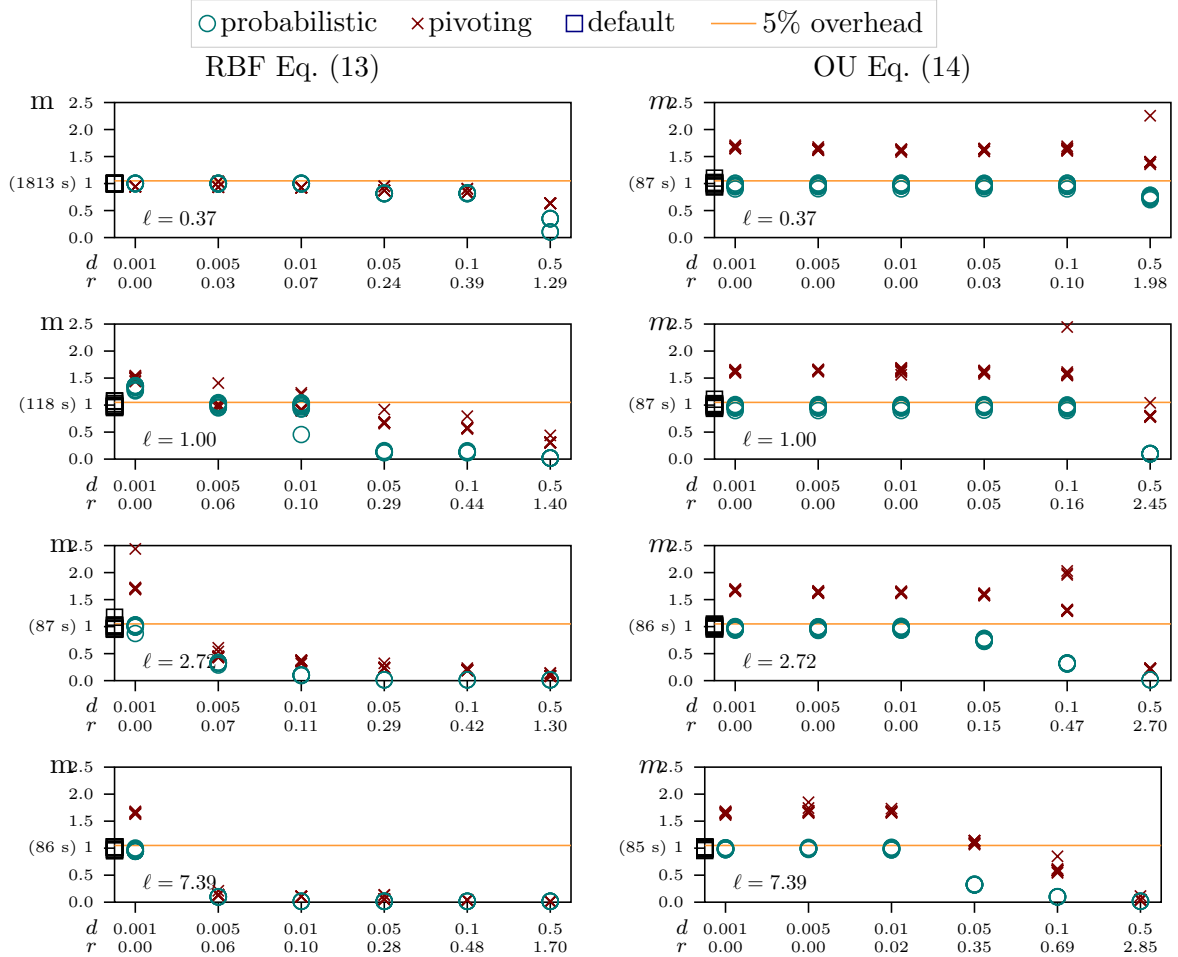


Figure 10: Relative execution times to compute the log-determinant using RBF (**left panel**) and OU (**right panel**) kernels on the PM2.5 dataset for $\theta = 1$, $\log \ell = -1, \dots, 3$ and $\delta = 0.1$ for ten repetitions. The number next to one on the y -axis displays the absolute execution times of the default Cholesky. The solid, horizontal, orange line (—) visualizes the 105% mark. The x -axis displays a desired absolute precision on the diagonal elements d (top) and the average corresponding desired relative precision r (bottom) on the log-determinant.

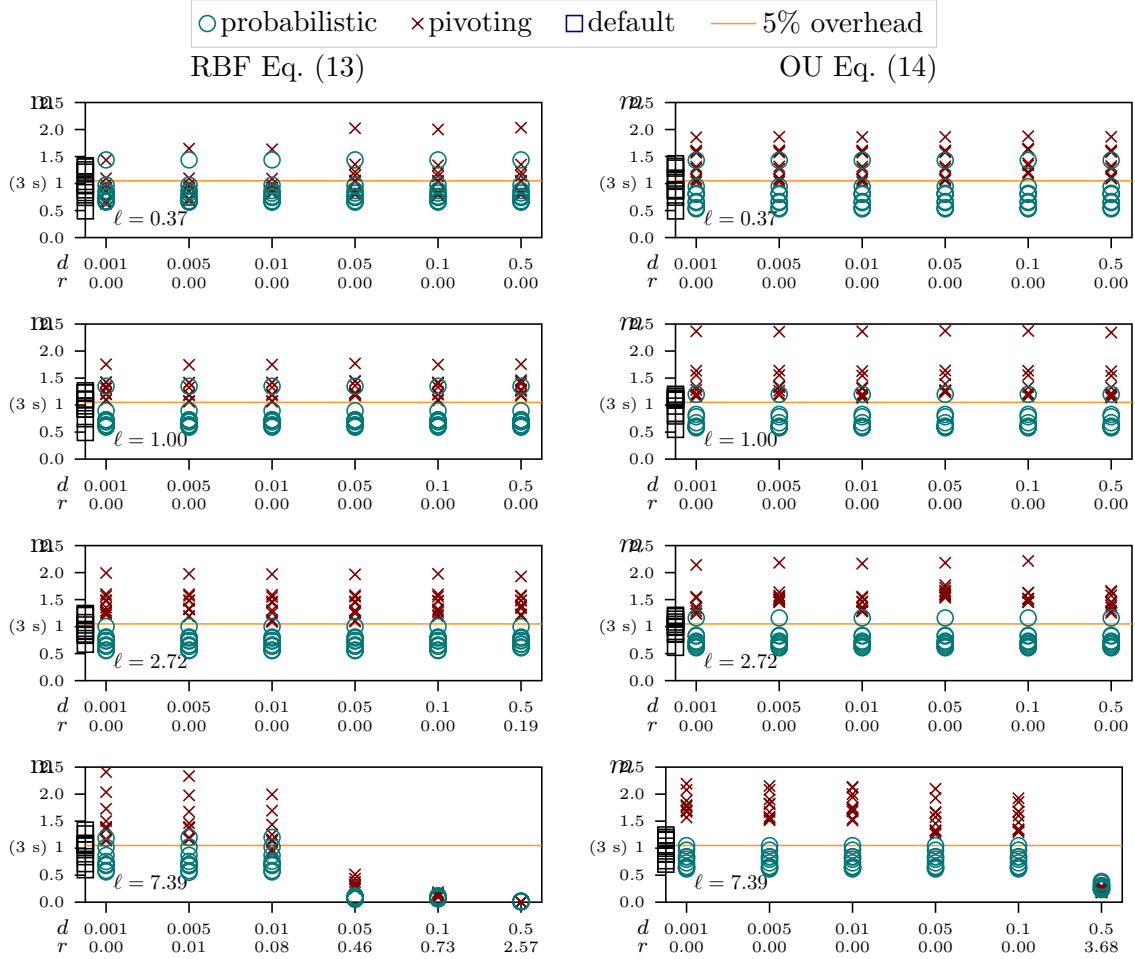


Figure 11: Relative execution times to compute the log-determinant using RBF (**left panel**) and OU (**right panel**) kernels on the PUMADYN dataset for $\theta = 1$, $\log \ell = -1, \dots, 3$ and $\delta = 0.1$ for ten repetitions. The number next to one on the y -axis displays the absolute execution times of the default Cholesky. The solid, horizontal, orange line (—) visualizes the 105% mark. The x -axis displays a desired absolute precision on the diagonal elements d (top) and the average corresponding desired relative precision r (bottom) on the log-determinant.

E.2 Stochastic Trace Estimator

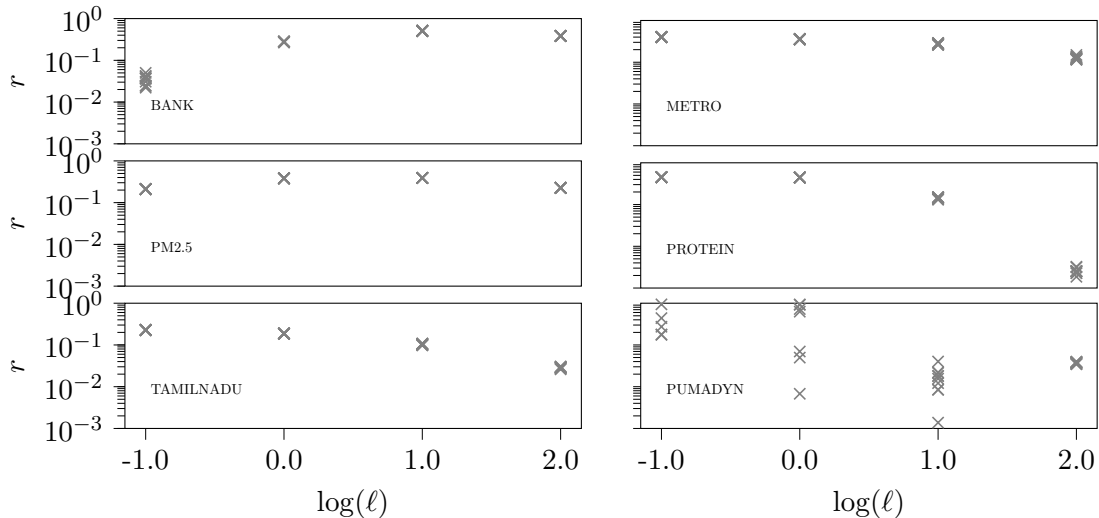


Figure 12: The need for theoretical guarantees. Of the related work described in Section 2.2 only Gardner et al. (2018) provide publicly accessible code. The figure shows the achieved relative error r , Eq. (3), when using default parameters, for the RBF kernel, Eq. (13), with $\theta := 1$ and different length-scales ℓ on all our considered datasets (see Table 1). The relative error is *more often than not* worse than 0.1 and can differ over two orders of magnitude. Theorem 2 in Gardner et al. (2018) which could describe how to set the parameters of their method to achieve a desired precision is not applicable in this setting (see Section 2.2).

E.3 Varying observation noise

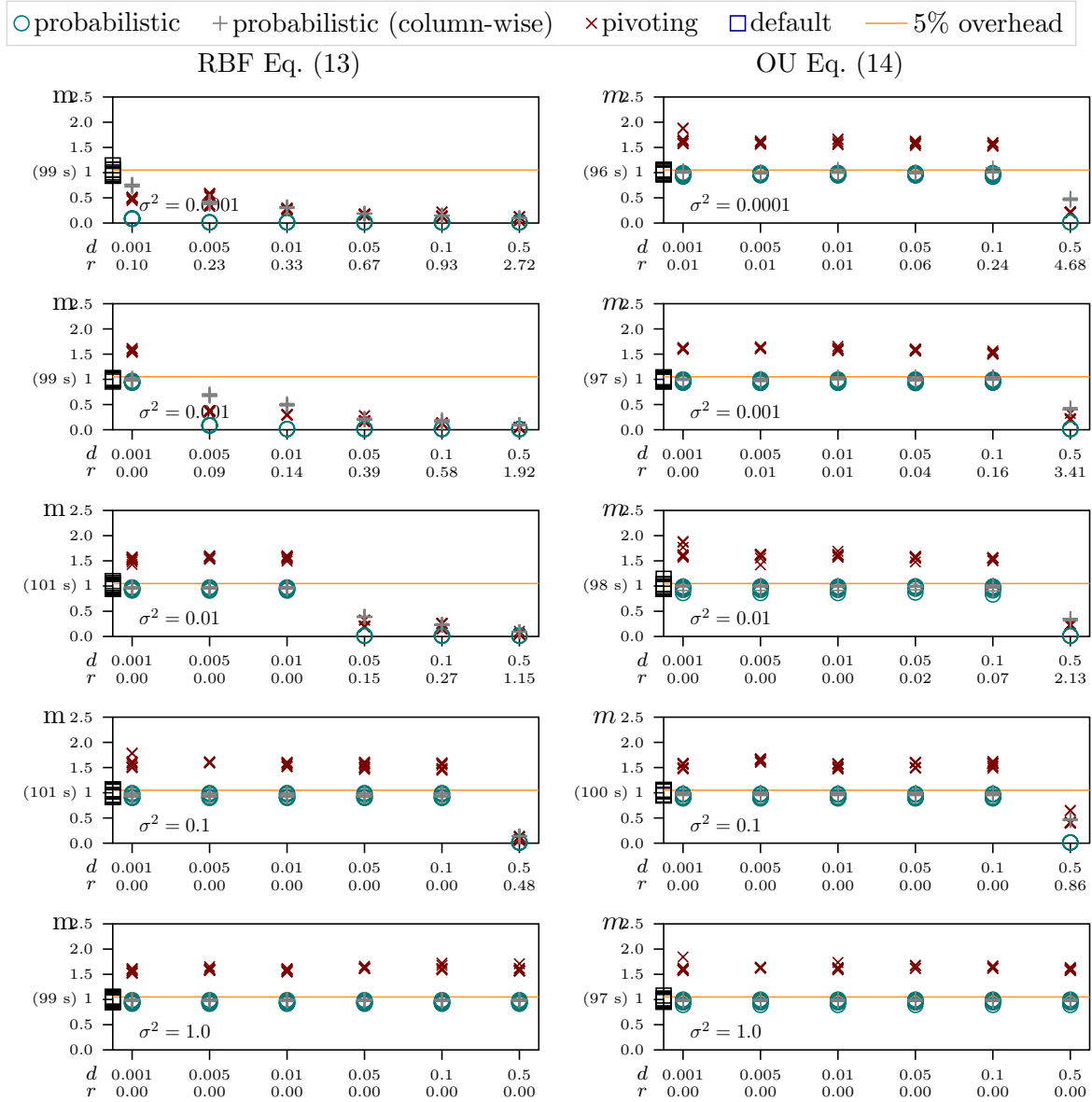


Figure 13: Relative execution times to compute the log-determinant using RBF (**left panel**) and OU (**right panel**) kernels on the PROTEIN dataset for $\theta = 1$, $\ell = 1$, $\log_{10} \sigma^2 = -4, \dots, 0$ and $\delta = 0.1$ for ten repetitions. The number next to one on the y -axis displays the absolute execution times of the default Cholesky. The solid, horizontal, orange line (—) visualizes the 105% mark. The x -axis displays a desired absolute precision on the diagonal elements d (top) and the average corresponding desired relative precision r (bottom) on the log-determinant.

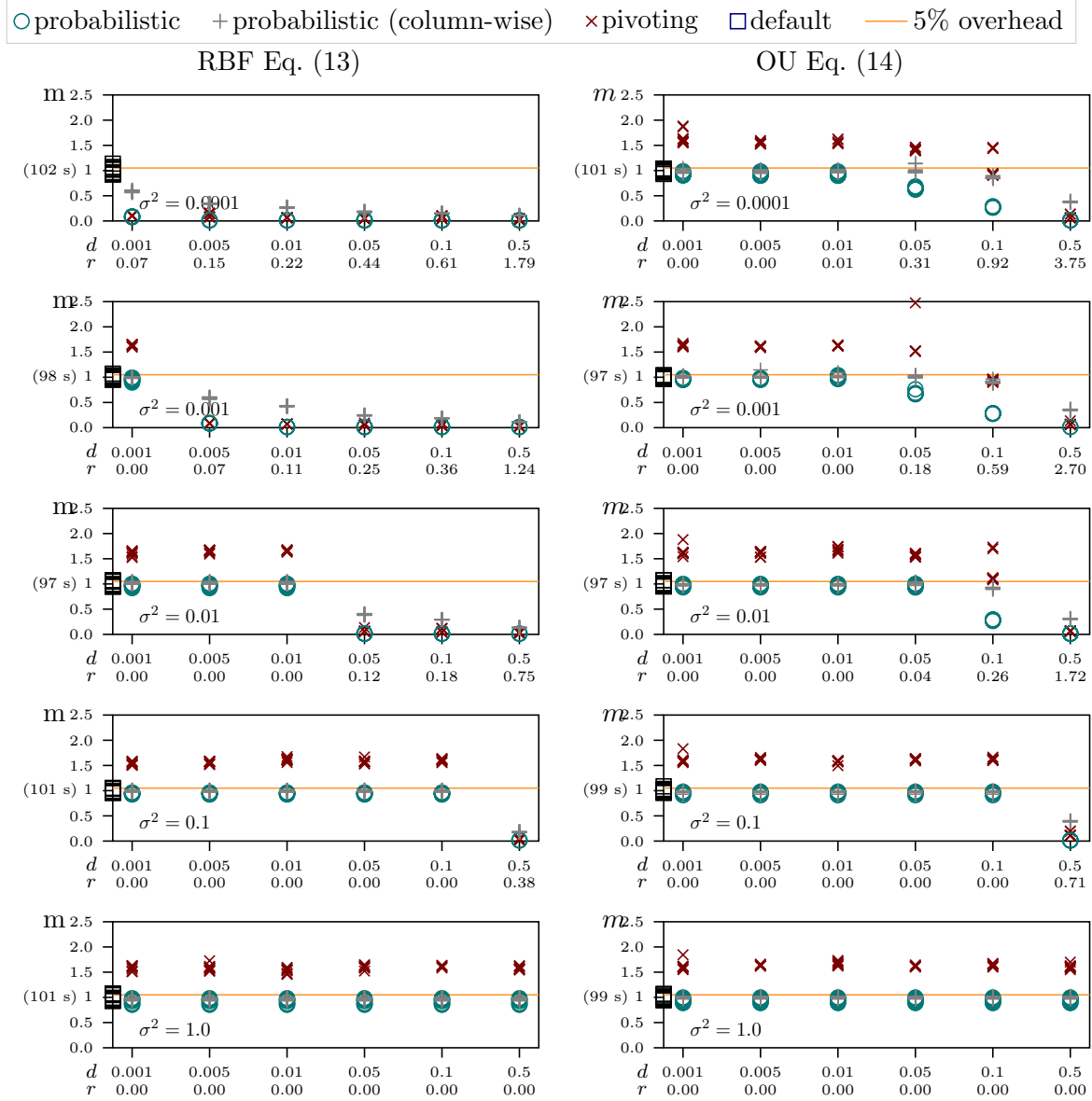


Figure 14: Relative execution times to compute the log-determinant using RBF (**left panel**) and OU (**right panel**) kernels on the TAMILNADU dataset for $\theta = 1$, $\ell = 1$, $\log_{10} \sigma^2 = -4, \dots, 0$ and $\delta = 0.1$ for ten repetitions. The number next to one on the y -axis displays the absolute execution times of the default Cholesky. The solid, horizontal, orange line (—) visualizes the 105% mark. The x -axis displays a desired absolute precision on the diagonal elements d (top) and the average corresponding desired relative precision r (bottom) on the log-determinant.

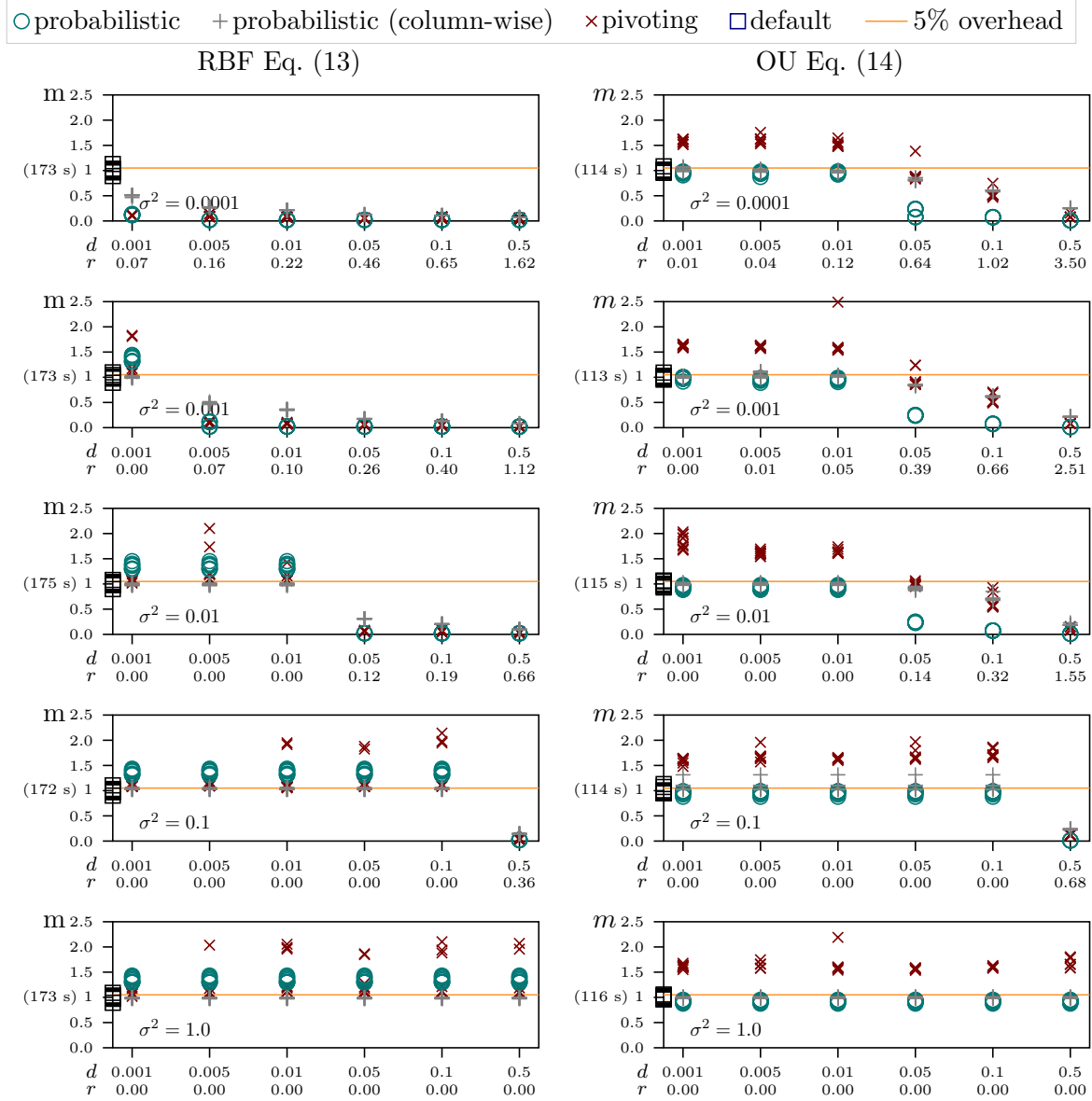


Figure 15: Relative execution times to compute the log-determinant using RBF (**left panel**) and OU (**right panel**) kernels on the METRO dataset for $\theta = 1$, $\ell = 1$, $\log_{10} \sigma^2 = -4, \dots, 0$ and $\delta = 0.1$ for ten repetitions. The number next to one on the y -axis displays the absolute execution times of the default Cholesky. The solid, horizontal, orange line (—) visualizes the 105% mark. The x -axis displays a desired absolute precision on the diagonal elements d (top) and the average corresponding desired relative precision r (bottom) on the log-determinant.

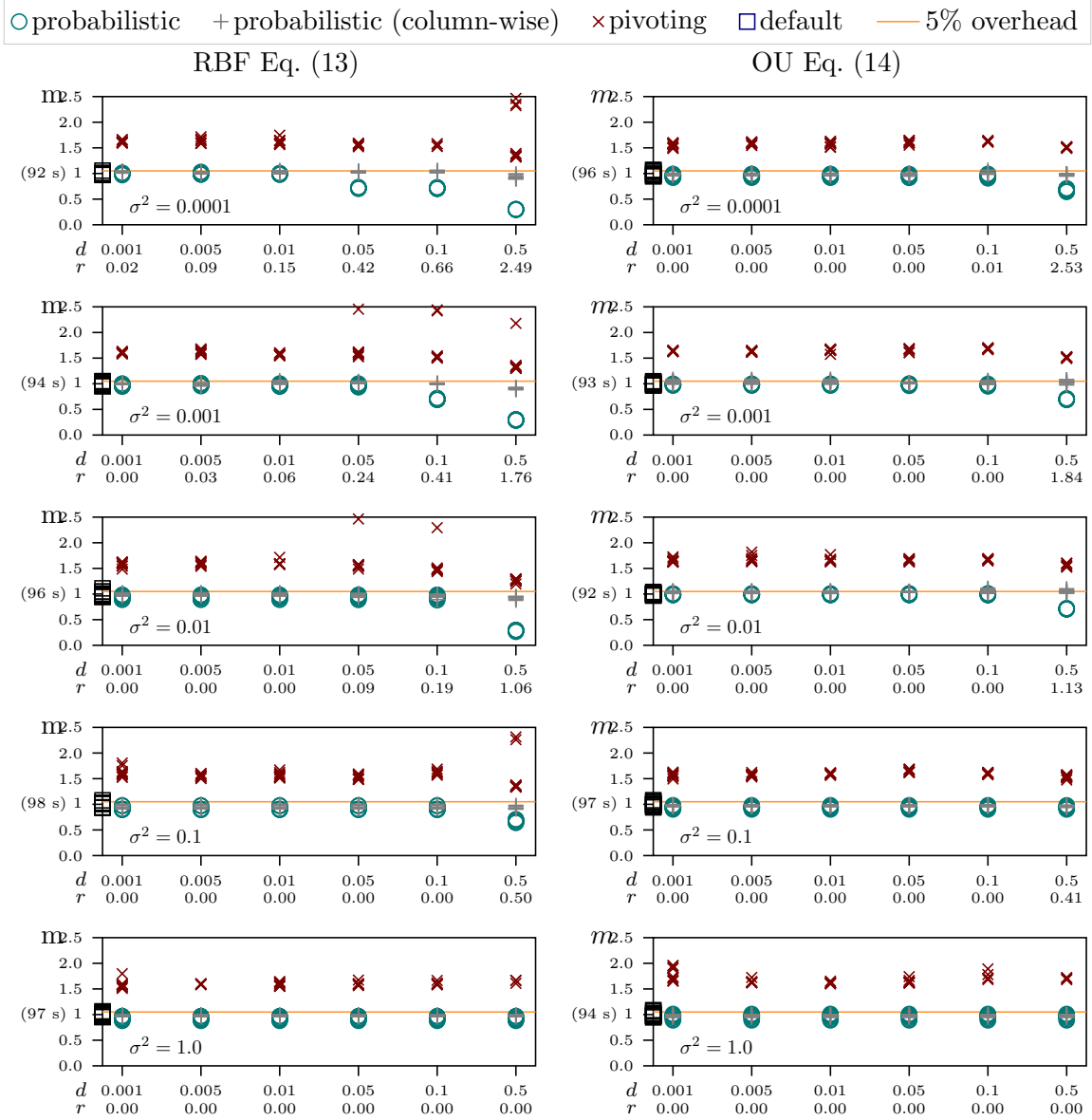


Figure 16: Relative execution times to compute the log-determinant using RBF (**left panel**) and OU (**right panel**) kernels on the BANK dataset for $\theta = 1$, $\ell = 1$, $\log_{10} \sigma^2 = -4, \dots, 0$ and $\delta = 0.1$ for ten repetitions. The number next to one on the y -axis displays the absolute execution times of the default Cholesky. The solid, horizontal, orange line (—) visualizes the 105% mark. The x -axis displays a desired absolute precision on the diagonal elements d (top) and the average corresponding desired relative precision r (bottom) on the log-determinant.

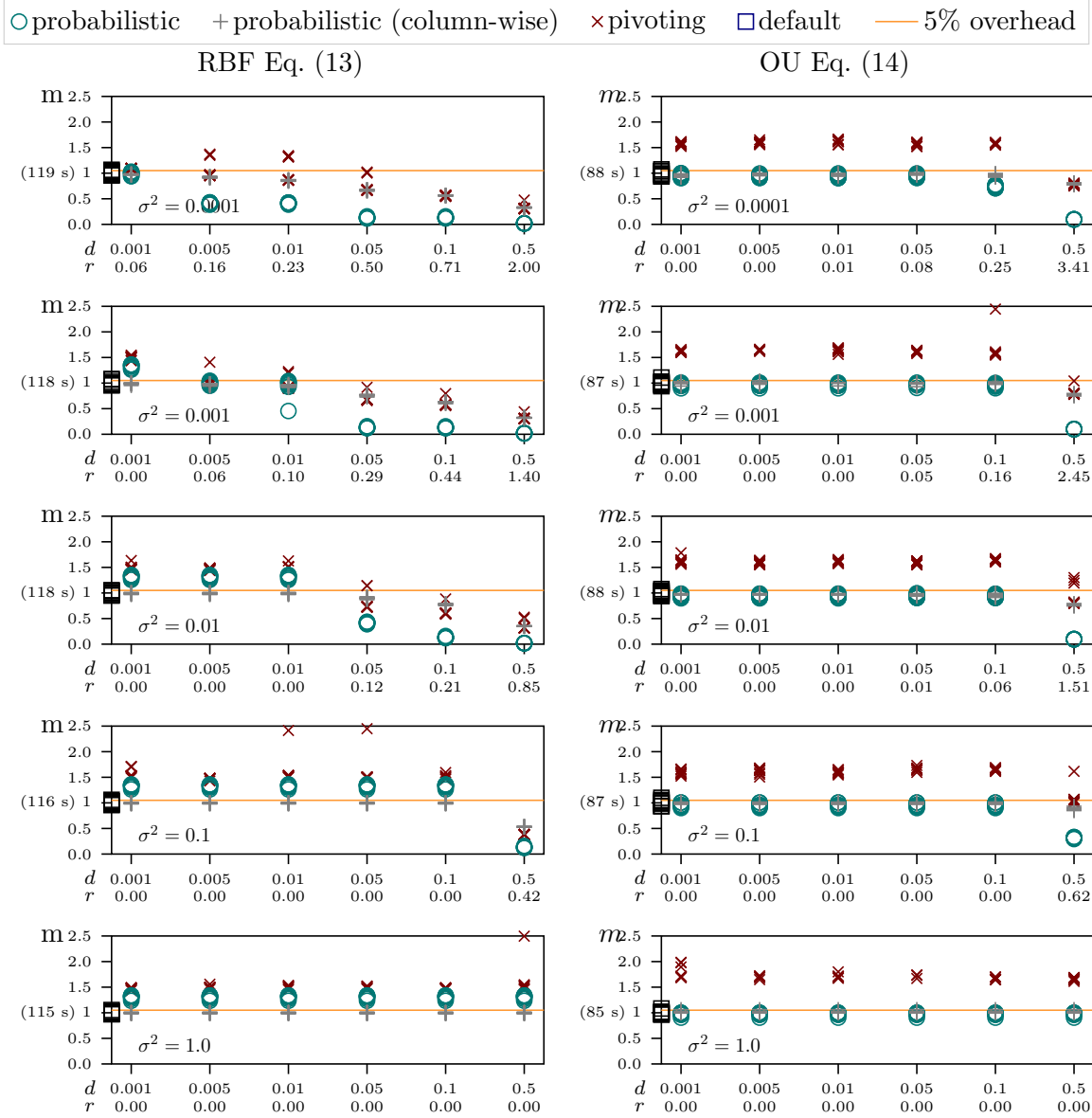


Figure 17: Relative execution times to compute the log-determinant using RBF (**left panel**) and OU (**right panel**) kernels on the PM2.5 dataset for $\theta = 1$, $\ell = 1$, $\log_{10} \sigma^2 = -4, \dots, 0$ and $\delta = 0.1$ for ten repetitions. The number next to one on the y -axis displays the absolute execution times of the default Cholesky. The solid, horizontal, orange line (—) visualizes the 105% mark. The x -axis displays a desired absolute precision on the diagonal elements d (top) and the average corresponding desired relative precision r (bottom) on the log-determinant.

E.4 Bound progression for varying lengthscale

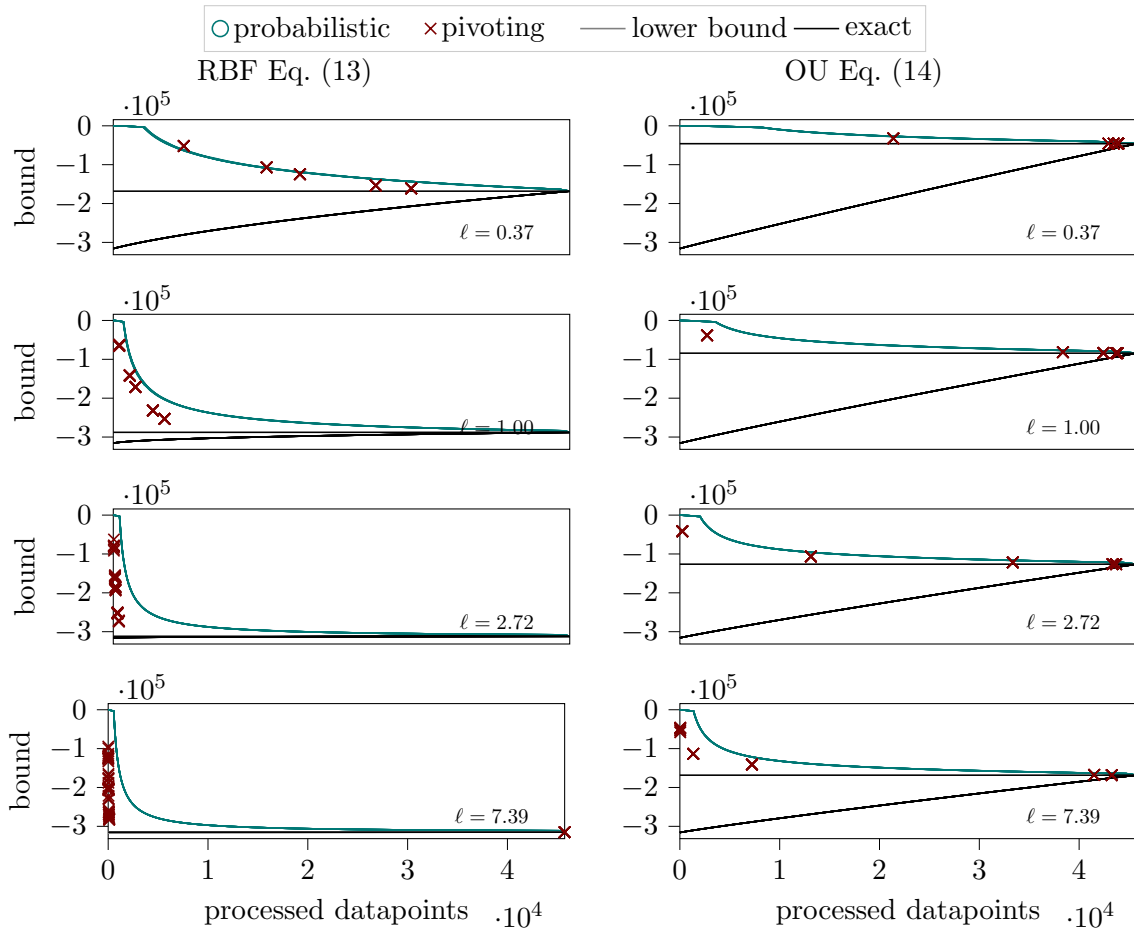


Figure 18: Progression of the bounds from Eqs. (6), (7) and (16) over the number of processed datapoints using RBF (**left panel**) and OU (**right panel**) kernels on the PROTEIN dataset for $\sigma^2 = 10^{-3}$, $\theta = 1$, $\log \ell = -1, \dots, 3$ and $\delta = 0.1$ for ten repetitions. The variance between repetitions is so small such that only one line is visible to the eye. For the Cholesky decomposition with pivoting (\times , continuous data is not available). The stopping points correspond to the ones described in Section 5.4.

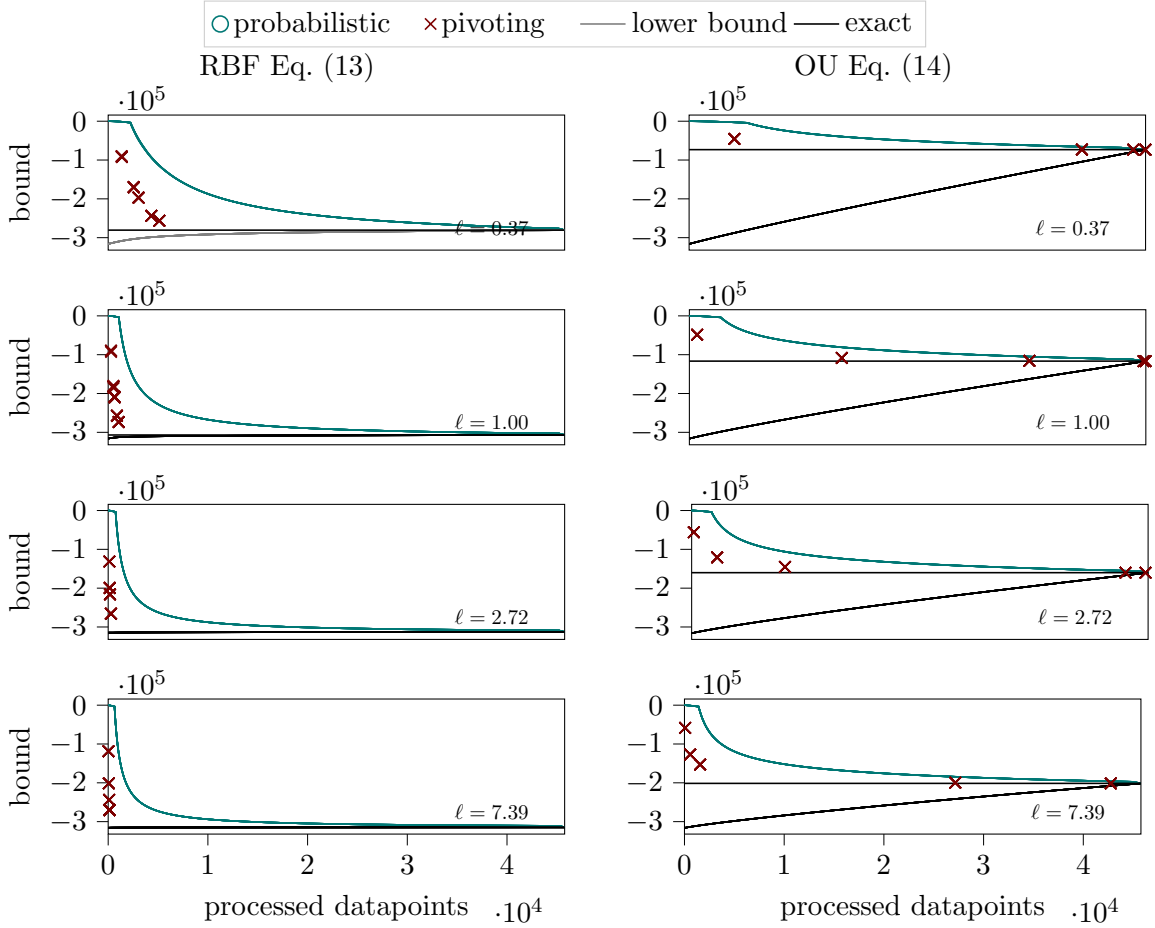


Figure 19: Progression of the bounds from Eqs. (6), (7) and (16) over the number of processed datapoints using RBF (**left panel**) and OU (**right panel**) kernels on the TAMILNADU dataset for $\sigma^2 = 10^{-3}$, $\theta = 1$, $\log \ell = -1, \dots, 3$ and $\delta = 0.1$ for ten repetitions. The variance between repetitions is so small such that only one line is visible to the eye. For the Cholesky decomposition with pivoting (\times , continuous data is not available). The stopping points correspond to the ones described in Section 5.4.

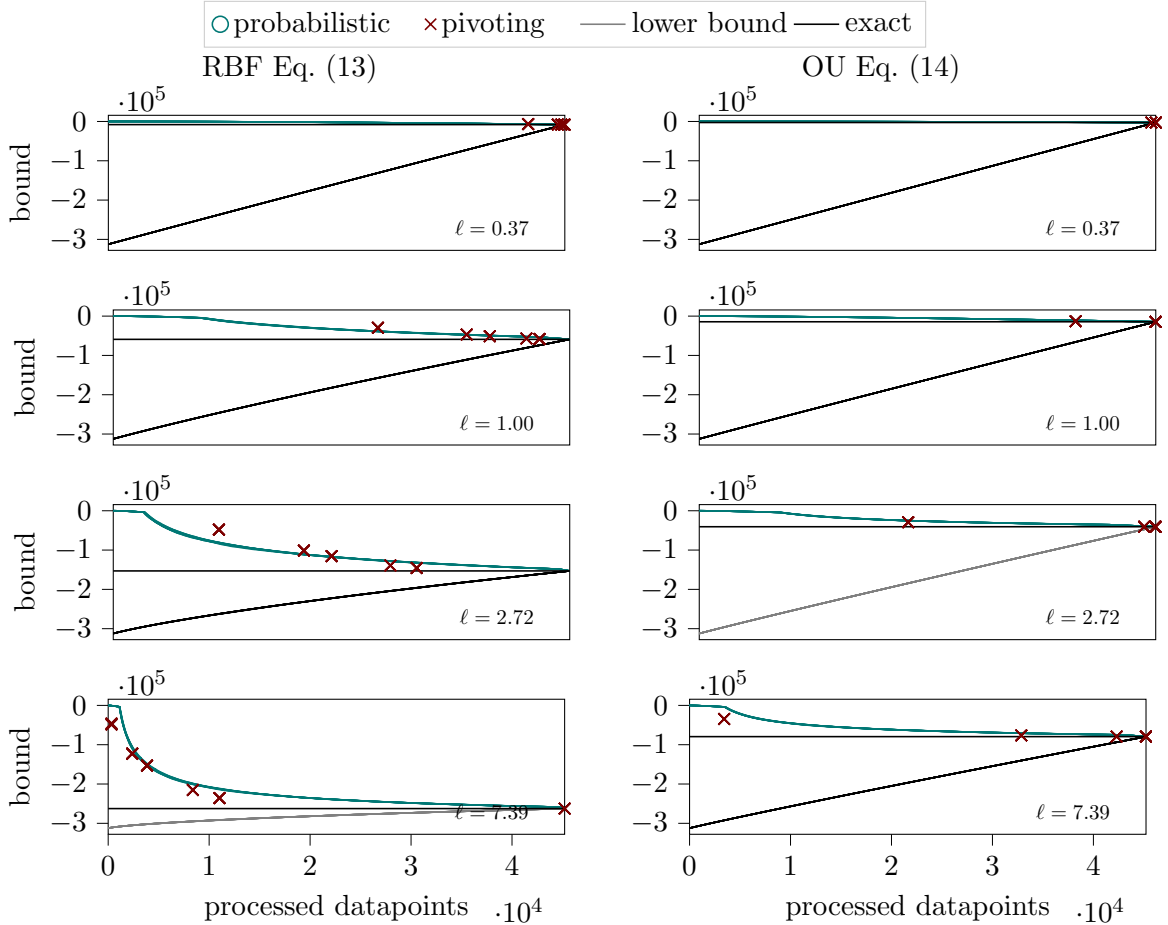


Figure 20: Progression of the bounds from Eqs. (6), (7) and (16) over the number of processed datapoints using RBF (**left panel**) and OU (**right panel**) kernels on the BANK dataset for $\sigma^2 = 10^{-3}$, $\theta = 1$, $\log \ell = -1, \dots, 3$ and $\delta = 0.1$ for ten repetitions. The variance between repetitions is so small such that only one line is visible to the eye. For the Cholesky decomposition with pivoting (\times , continuous data is not available). The stopping points correspond to the ones described in Section 5.4.

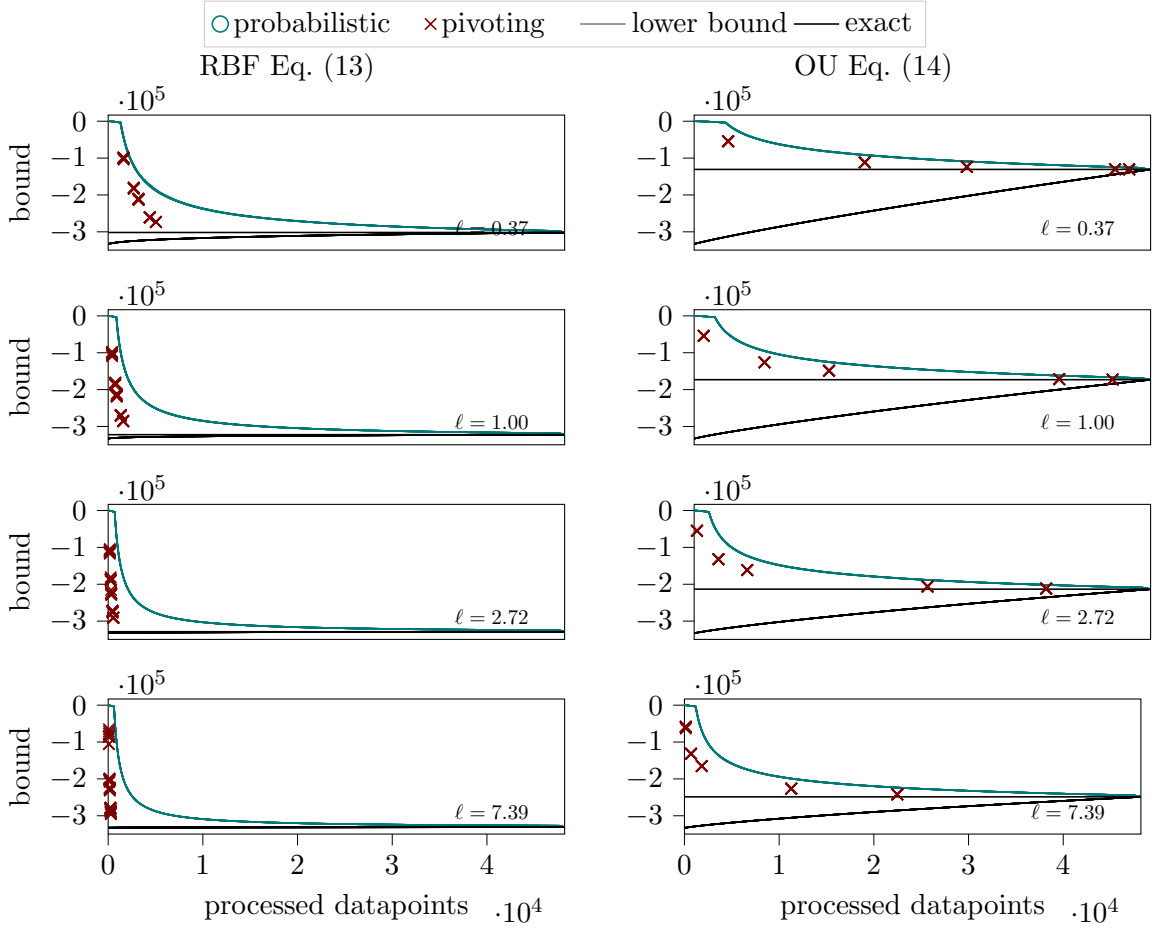


Figure 21: Progression of the bounds from Eqs. (6), (7) and (16) over the number of processed datapoints using RBF (**left panel**) and OU (**right panel**) kernels on the METRO dataset for $\sigma^2 = 10^{-3}$, $\theta = 1$, $\log \ell = -1, \dots, 3$ and $\delta = 0.1$ for ten repetitions. The variance between repetitions is so small such that only one line is visible to the eye. For the Cholesky decomposition with pivoting (\times , continuous data is not available). The stopping points correspond to the ones described in Section 5.4.

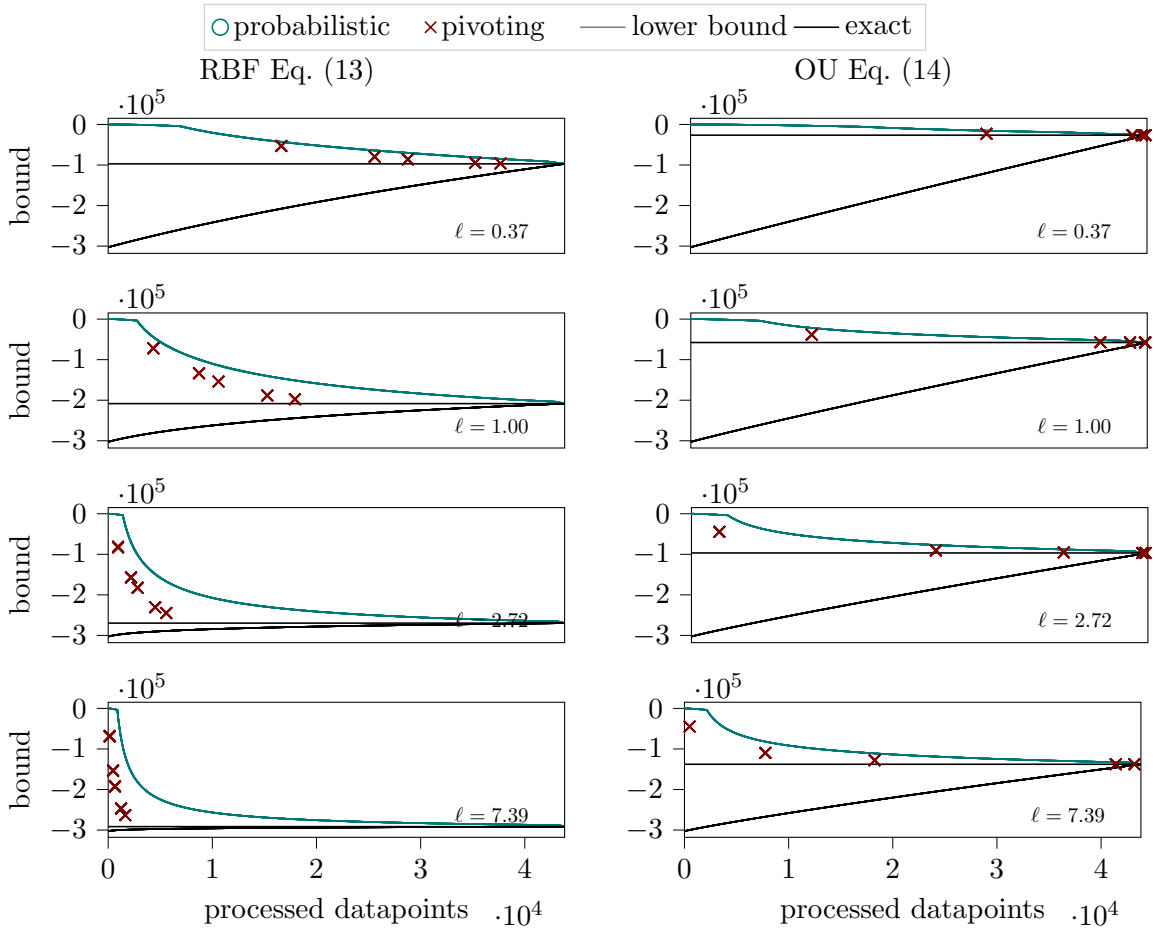


Figure 22: Progression of the bounds from Eqs. (6), (7) and (16) over the number of processed datapoints using RBF (**left panel**) and OU (**right panel**) kernels on the PM2.5 dataset for $\sigma^2 = 10^{-3}$, $\theta = 1$, $\log \ell = -1, \dots, 3$ and $\delta = 0.1$ for ten repetitions. The variance between repetitions is so small such that only one line is visible to the eye. For the Cholesky decomposition with pivoting (\times , continuous data is not available). The stopping points correspond to the ones described in Section 5.4.

E.5 Bound progression for varying observation noise

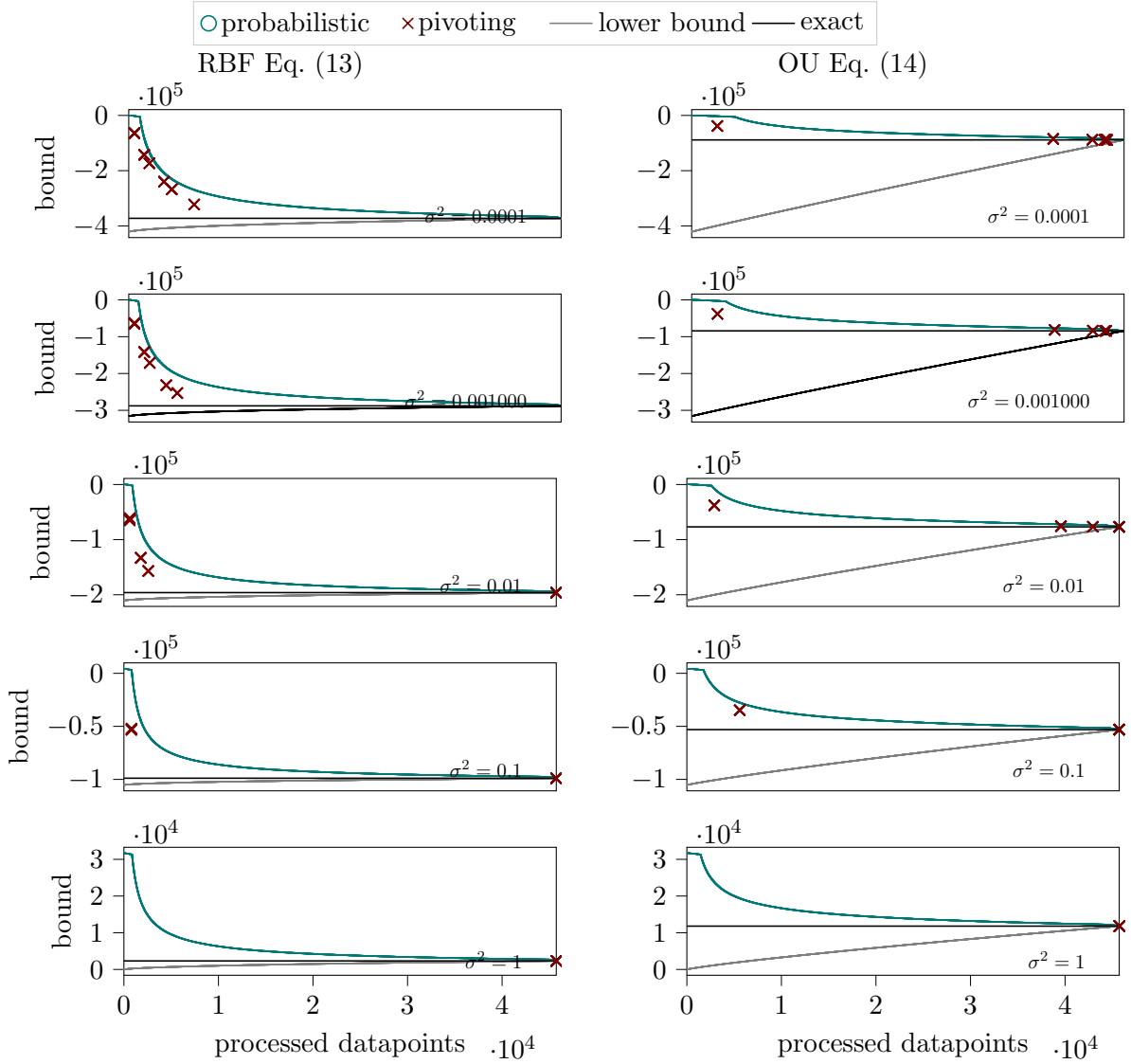


Figure 23: Progression of the bounds from Eqs. (6), (7) and (16) over the number of processed datapoints using RBF (**left panel**) and OU (**right panel**) kernels on the PROTEIN dataset for $\log \sigma^2 = -4, \dots, 0, \theta = 1, \log \ell = 0$ and $\delta = 0.1$ for ten repetitions. The variance between repetitions is so small such that only one line is visible to the eye. For the Cholesky decomposition with pivoting (\times , continuous data is not available). The stopping points correspond to the ones described in Section 5.4.

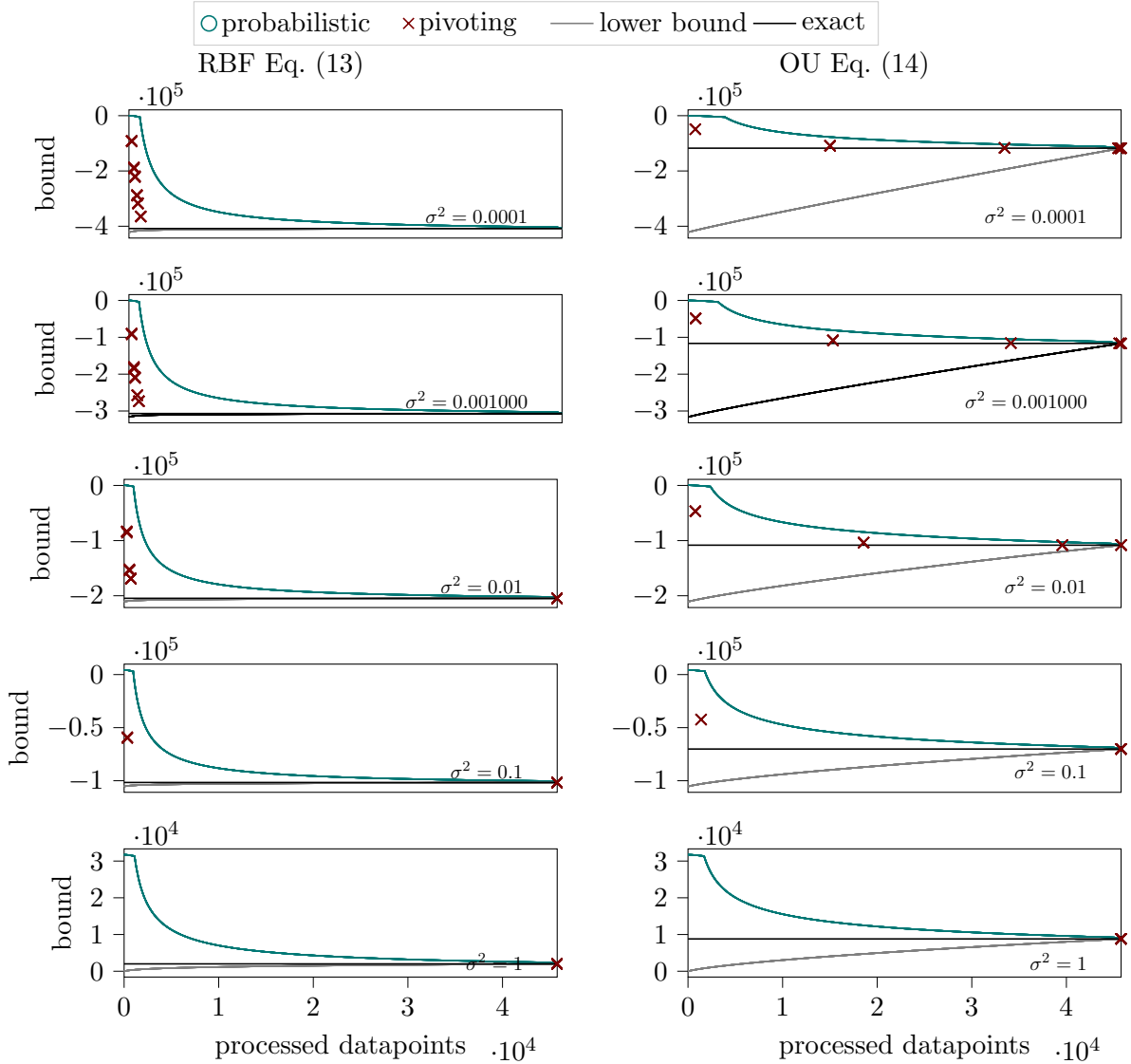


Figure 24: Progression of the bounds from Eqs. (6), (7) and (16) over the number of processed datapoints using RBF (**left panel**) and OU (**right panel**) kernels on the TAMILNADU dataset for $\log \sigma^2 = -4, \dots, 0, \theta = 1, \log \ell = 0$ and $\delta = 0.1$ for ten repetitions. The variance between repetitions is so small such that only one line is visible to the eye. For the Cholesky decomposition with pivoting (\times , continuous data is not available). The stopping points correspond to the ones described in Section 5.4.

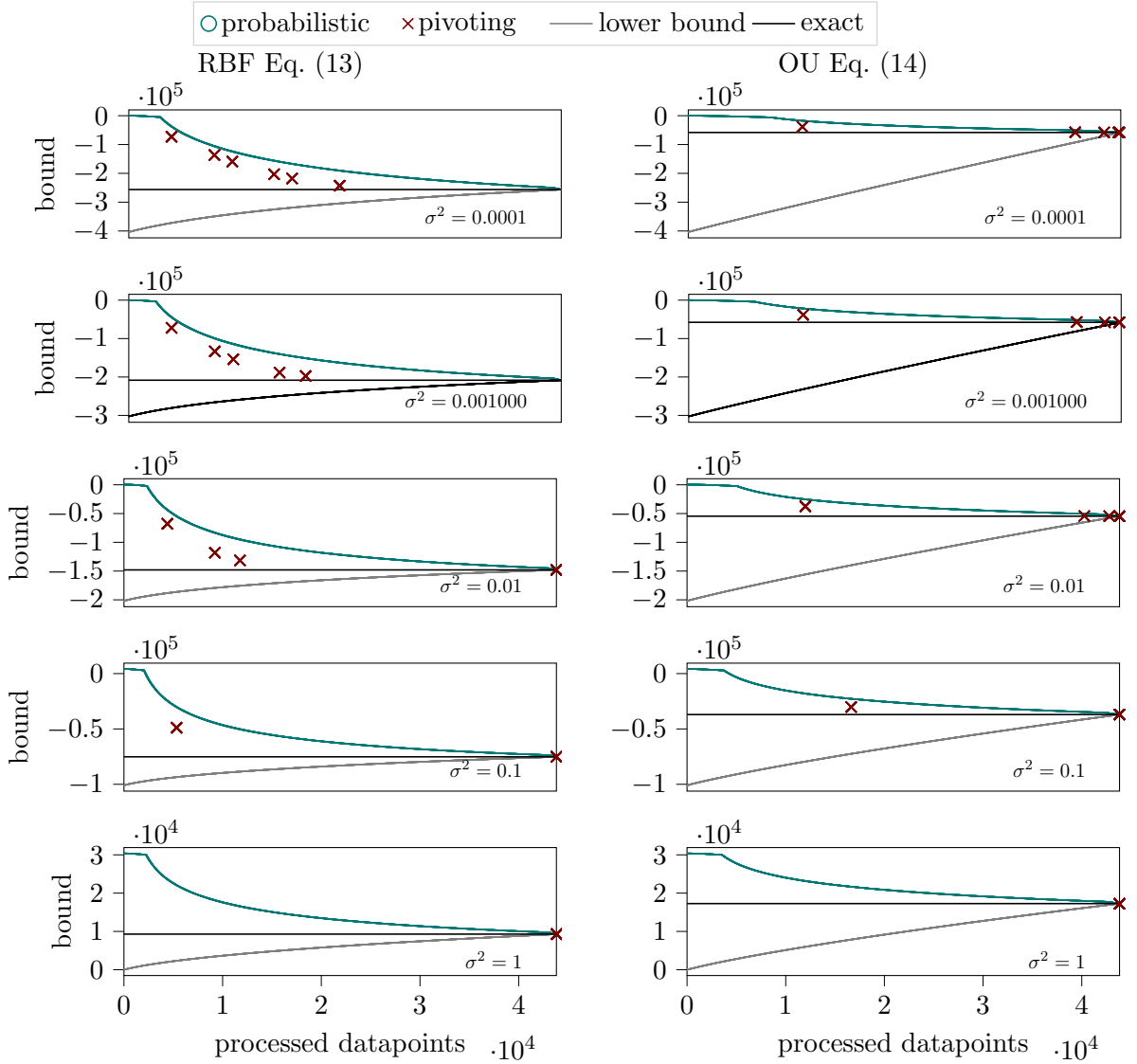


Figure 25: Progression of the bounds from Eqs. (6), (7) and (16) over the number of processed datapoints using RBF (**left panel**) and OU (**right panel**) kernels on the PM2.5 dataset for $\log \sigma^2 = -4, \dots, 0, \theta = 1, \log \ell = 0$ and $\delta = 0.1$ for ten repetitions. The variance between repetitions is so small such that only one line is visible to the eye. For the Cholesky decomposition with pivoting (\times , continuous data is not available). The stopping points correspond to the ones described in Section 5.4.

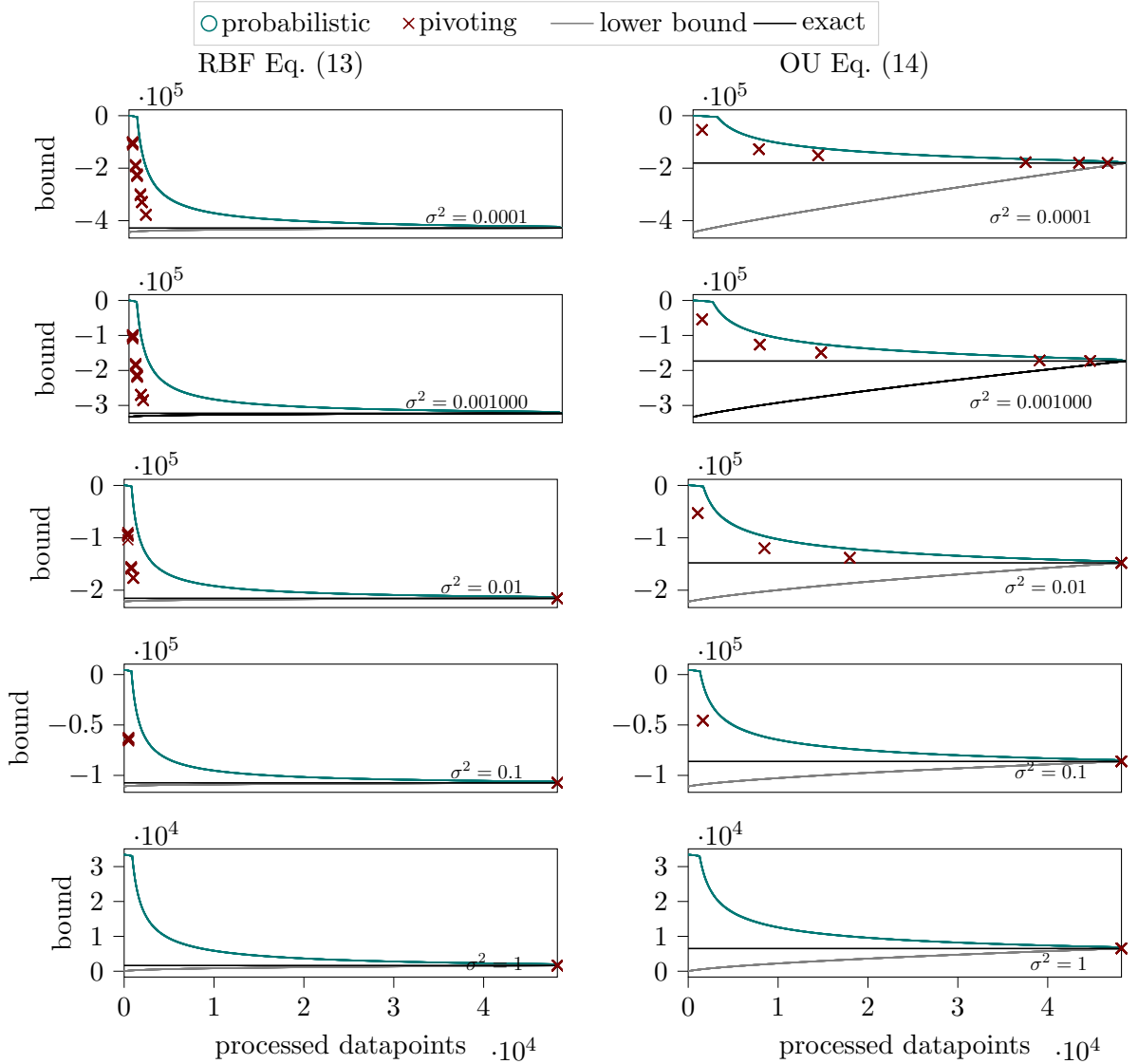


Figure 26: Progression of the bounds from Eqs. (6), (7) and (16) over the number of processed datapoints using RBF (**left panel**) and OU (**right panel**) kernels on the METRO dataset for $\log \sigma^2 = -4, \dots, 0$, $\theta = 1$, $\log \ell = 0$ and $\delta = 0.1$ for ten repetitions. The variance between repetitions is so small such that only one line is visible to the eye. For the Cholesky decomposition with pivoting (\times , continuous data is not available). The stopping points correspond to the ones described in Section 5.4.

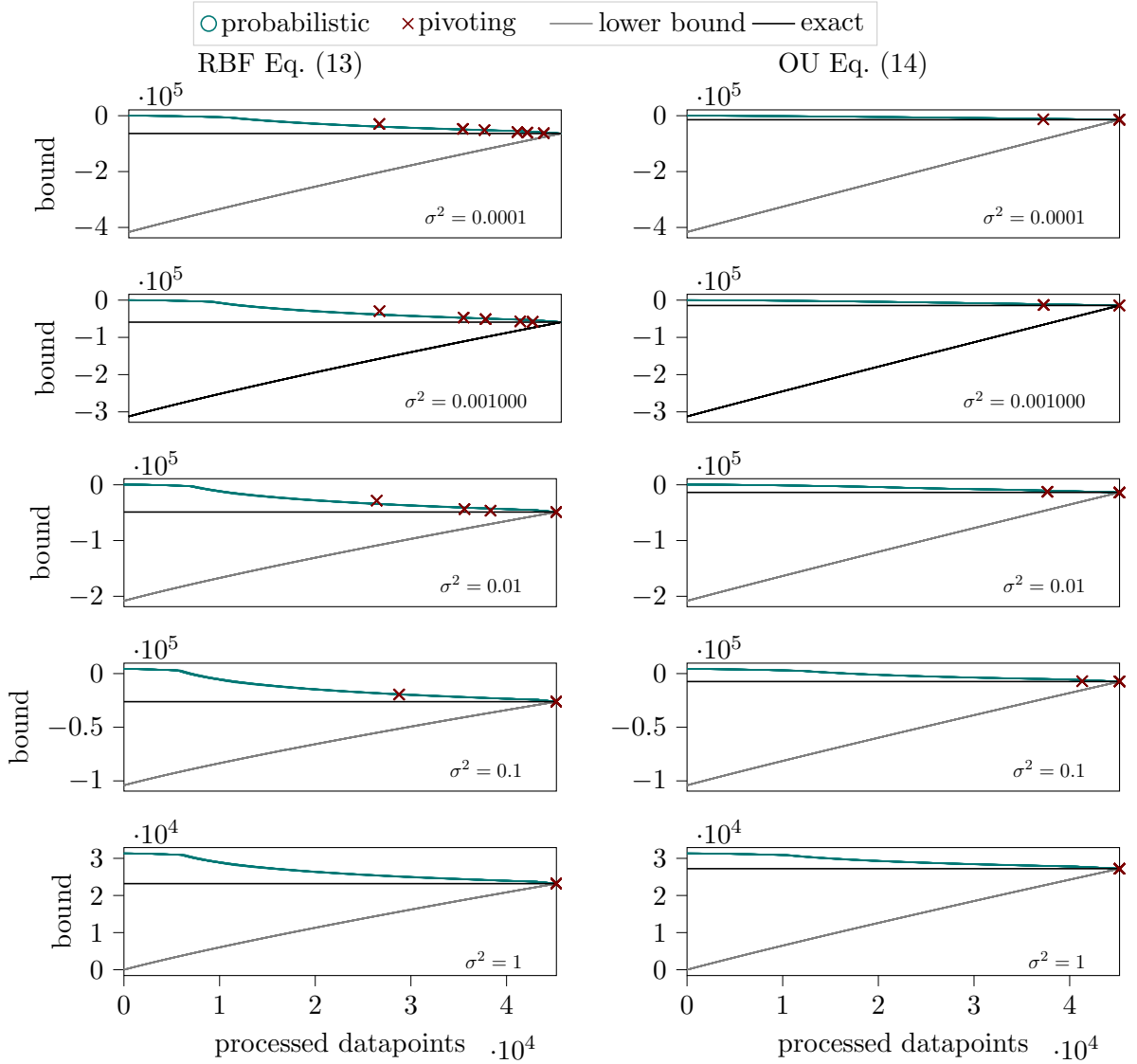


Figure 27: Progression of the bounds from Eqs. (6), (7) and (16) over the number of processed datapoints using RBF (**left panel**) and OU (**right panel**) kernels on the BANK dataset for $\log \sigma^2 = -4, \dots, 0, \theta = 1, \log \ell = 0$ and $\delta = 0.1$ for ten repetitions. The variance between repetitions is so small such that only one line is visible to the eye. For the Cholesky decomposition with pivoting (\times , continuous data is not available). The stopping points correspond to the ones described in Section 5.4.

References

- Artem Artemev, David R. Burt, and Mark van der Wilk. Tighter bounds on the log marginal likelihood of gaussian process regression using conjugate gradients. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 362–372, 2021.
- Simon Bartels. *Probabilistic Linear Algebra*. PhD thesis, University of Tübingen, 2020.
- Simon Bartels, Kristoffer Stensbo-Smidt, Pablo Mureño-Munoz, Wouter Boomsma, Jes Frellsen, and Søren Hauberg. Adaptive Cholesky Gaussian Processes. *Proceedings of Artificial Intelligence and Statistics (AISTATS)*, to appear, 2023. <http://arxiv.org/abs/2202.10769>.
- Stéphane Boucheron, Gábor Lugosi, and Pascal Massart. *Concentration Inequalities: A Nonasymptotic Theory of Independence*. Oxford University Press, 1st edition, 2013.
- Christos Boutsidis, Petros Drineas, Prabhanjan Kambadur, Eugenia-Maria Kontopoulou, and Anastasios Zouzias. A randomized algorithm for approximating the log determinant of a symmetric positive definite matrix. *Linear Algebra and its Applications*, 533:95 – 117, 2017.
- Krzysztof Chalupka, Williams, C. K. I., and Iain Murray. A framework for evaluating approximation methods for Gaussian process regression. *Journal of Machine Learning Research*, 14(1):333–350, 2013.
- James Davidson. *Stochastic Limit Theory: An Introduction for Econometricians*. Oxford University Press, 1994.
- P. Diaconis. Bayesian numerical analysis. *Statistical decision theory and related topics*, IV (1):163–175, 1988.
- Kun Dong, David Eriksson, Hannes Nickisch, David Bindel, and Andrew G. Wilson. Scalable log determinants for gaussian process kernel learning. In *Advances in Neural Information Processing Systems*, pages 6330–6340, 2017.
- Sebastian Dorn and Torsten A. Enßlin. Stochastic determination of matrix determinants. *Physical Review E*, 92:013302, 2015.
- Dheeru Dua and Casey Graff. UCI machine learning repository, 2019. URL <http://archive.ics.uci.edu/ml>.
- Xiequan Fan, Ion Grama, and Quansheng Liu. Hoeffding’s inequality for supermartingales. *Stochastic Processes and their Applications*, 122(10):3545–3559, 2012.
- Jack Fitzsimons, Kurt Cutajar, Michael Osborne, Stephen Roberts, and Maurizio Filippone. Bayesian inference of log determinants. In Gal Elidan, Kristian Kersting, and Alexander T. Ihler, editors, *Thirty-Third Conference on Uncertainty in Artificial Intelligence, UAI 2017, August 11-15, 2017, Sydney, Australia*, 2017a.

- Jack Fitzsimons, Diego Granziol, Kurt Cutajar, Michael Osborne, Maurizio Filippone, and Stephen Roberts. Entropic trace estimates for log determinants. In Michelangelo Ceci, Jaakko Hollmén, Ljupčo Todorovski, Celine Vens, and Sašo Džeroski, editors, *Machine Learning and Knowledge Discovery in Databases*, pages 323–338, 2017b.
- Jacob R. Gardner, Geoff Pleiss, David Bindel, Kilian Q. Weinberger, and Andrew G. Wilson. Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration. In *Advances in Neural Information Processing Systems*, 2018.
- Alan George, Michael T. Heath, and Joseph Liu. Parallel cholesky factorization on a shared-memory multiprocessor. *Linear Algebra and its Applications*, 77:165–187, 1986.
- Gene H. Golub and Charles F. Van Loan. *Matrix computations*. Johns Hopkins University Press, 4 edition, 2013.
- Geoffrey Grimmett and David Stirzaker. *Probability and Random Processes*. Oxford University Press, 3rd edition, 2001.
- Helmut Harbrecht, Michael Peters, and Reinhold Schneider. On the low-rank approximation by the pivoted Cholesky decomposition. *Applied Numerical Mathematics*, 62(4):428–440, 2012.
- P. Hennig, M.A. Osborne, and M. Girolami. Probabilistic numerics and uncertainty in computations. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 471(2179), 2015.
- Xuan Liang, Tao Zou, Bin Guo, Shuo Li, Haozhe Zhang, Shuyi Zhang, Hui Huang, and Song Xi Chen. Assessing beijing’s $pm_{2.5}$ pollution: severity, weather impact, apec and winter heating. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 471(2182):20150257, 2015.
- Volodymyr Mnih. Efficient stopping rules. Master’s thesis, University of Alberta, Canada, 2008.
- Volodymyr Mnih, Csaba Szepesvári, and Jean-Yves Audibert. Empirical Bernstein stopping. pages 672–679, 2008.
- Jonas Močkus. On Bayesian methods for seeking the extremum. In Gury I. Marchuk, editor, *Optimization Techniques IFIP Technical Conference*, volume 27 of *Lecture Notes in Computer Science*, pages 400–404, 1975.
- Sérgio Moro, Paulo Cortez, and Paulo Rita. A data-driven approach to predict the success of bank telemarketing. *Decision Support Systems*, 62:22–31, 2014.
- Tiberiu Popoviciu. Sur les équations algébriques ayant toutes leurs racines réelles. *Mathematica*, 9:129–145, 1935.
- Carl E. Rasmussen and Christopher K. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2006.

- Arvind K. Saibaba, Alen Alexanderian, and Ilse C. F. Ipsen. Randomized matrix-free trace and log-determinant estimators. *Numerische Mathematik*, 137(2):353–395, Oct 2017.
- Matthias Seeger. Skilling techniques for Bayesian analysis. 2000.
- Rajesh Sharma, Madhu Gupta, and Girish Kapoor. Some better bounds on the variance with applications. *Journal of Mathematical Inequalities*, 4:355–363, 2010.
- John Skilling. *The Eigenvalues of Mega-dimensional Matrices*, pages 455–466. 1989.
- Edward Snelson and Zoubin Ghahramani. Sparse Gaussian processes using pseudo-inputs. In Y. Weiss, B. Schölkopf, and J. C. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 1257–1264. 2006.
- Christian Steinruecken, Emma Smith, David Janz, James Lloyd, and Zoubin Ghahramani. The Automatic Statistician. In Frank Hutter, Lars Kotthoff, and Joaquin Vanschoren, editors, *Automated Machine Learning*, Series on Challenges in Machine Learning. 2019.
- Shashanka Ubaru, Jie Chen, and Yousef Saad. Fast estimation of $tr(f(a))$ via stochastic lanczos quadrature. *SIAM Journal on Matrix Analysis and Applications*, 38(4):1075–1099, 2017.
- Qian Wang, Xianyi Zhang, Yunquan Zhang, and Qing Yi. AUGEM: Automatically generate high performance Dense Linear Algebra kernels on x86 CPUs. In *SC '13: Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, pages 1–12, 2013.