Adaptive Cholesky Gaussian Processes Supplementary

Anonymous Author(s)
Affiliation
Address
`email`

Contents

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

A Experimental details

Table 1: Overview over all datasets used for the experiments in Section 4. The total dataset size (training and testing) is denoted N and D denotes the dimensionality.

Key	N	D	Source
bike	17 379	17	Fanaee-T & Gama (2013). Available at this UCI page.
elevators	16 599	18	Camachol (1998).
kin40k	40 000	8	Schwaighofer & Tresp (2002).
metro	48 204	66	No citation request. Available at this UCI page.
pm25	43 824	79	Liang et al. (2015). Available at this UCI page.
poletelecomm	15 000	26	Weiss & Indurkhya (1995).
protein	45 730	9	No citation request. Available at this UCI page.
pumadyn	8192	32	No citation request. Available at this website.

For an overview of the datasets we use, see Table 1. The datasets are all normalized to have zero mean and unit variance for each feature. We explore two different computing environments. For datasets smaller than 20 000 data points, we ran our experiments on a single GPU. This is the same setup as in Artemev et al. (2021) with the difference that we use a TITAN RTX whereas they have used a TESLA V100. For datasets larger than 20 000 datapoints, our setup differs from Artemev et al. (2021). We use only CPUs on machines where the kernel matrix still fits fully into memory. Specifically, we used machines running Ubuntu 18.04 with 50 Gigabytes of RAM and two INTEL XEON E5-2670 V2 CPUs.

A.1 Bound quality experiments

For CGLB, we compute the bounds with varying number of inducing inputs $M := \{512, 1024, 2048, 4096\}$ and measure the time it takes to compute the bounds. For ACGP, we define the blocksize $m := 256 \cdot 40 = 10\,192$ which is the default OPENBLAS block size on our machines times the number of cores. This ensures that the sample size for our bounds is sufficiently large for accurate estimation, and at the same time the number of page-faults should be comparable to the default Cholesky implementation. We measure the elapsed time every time a block of data points is added to the processed dataset and the bounds are recomputed.

We compare both methods using squared exponential kernel (SE) and the Ornstein-Uhlenbeck kernel (OU).

$$k_{\text{SE}}(\boldsymbol{x}, \boldsymbol{z}) := \theta \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{z}\|^2}{2\ell^2}\right) \tag{1}$$

$$k_{\text{OU}}(\boldsymbol{x}, \boldsymbol{z}) := \theta \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{z}\|}{\ell}\right). \tag{2}$$

where we fix $\theta := 1$ and we vary ℓ as $\log \ell \in \{-1, 0, 1, 2\}$. We use a Gaussian likelihood and fix the noise to $\sigma^2 := 10^{-3}$.

A.2 Hyper-parameter tuning

In this section, we describe our experimental setup for the hyper-parameter optimization experiments, which closely follows that of Artemev et al. (2021). We randomly split each dataset into a training set consisting of 2/3 of examples, and a test set consisting of the remaining third. We use a Matérn$\frac{3}{2}$ kernel function and L-BFGS-B as the optimizer with SCIPY (Virtanen et al., 2020) default parameters if not specified otherwise. All algorithms are stopped the latest after 2000 optimization steps, after 12 hours of compute time, or when optimization has failed three times. We repeat each experiment five times with a different shuffle of the dataset and report the results in Tables 2 and 3.

For CGLB, it is necessary to decide on a number of inducing inputs. From the results reported by Artemev et al. (2021), it appears that using $M = 2048$ inducing inputs yields the best trade-off in terms of speed and performance, hence we use this value in our experiments. For the exact Cholesky and CGLB, the L-BFGS-B convergence criterion “relative change in function value” (`ftol`) is set to 0.

For ACGP, we need to decide on both the desired relative error, r, as well as the block size m. We successively decrease the optimizer’s tolerance `ftol` as $(2/3)^{\text{restart}+1}$ and we set the same value for r. That is, regardless of whether the optimization of ACGP stopped successfully or for abnormal reasons, the optimization restarts aiming for higher precision. The effect of this is that, early in the hyper-parameter optimization, ACGP will stop early, thus providing only an approximation to the optimal hyper-parameter values, but also saving computations. With each restart, ACGP increases the precision, ensuring that we get closer and closer to the optimal hyper-parameter values at the expense of approaching the computational demand of an exact GP. The block size m is set to the same value as for the bound quality experiments, Section 4.1, $40 \cdot 256 = 10\,192$, which is the number of cores times the OpenBLAS block size. This ensures that the sample size for our bounds is sufficiently large for accurate estimation, and at the same time the number of page-faults should be comparable to the default Cholesky implementation. Note that m is a global parameter, independent of the dataset. Hence, natural choices for both r and m are determined by parameters of standard software, which have sensible, machine-dependent default values. ACGP can therefore be considered parameter-free.

Differing from the previous section, we use for ACGP the biased estimator $(N-M)\log p(\boldsymbol{y}_{:M})/M$ instead of $\mathcal{U}/2+\mathcal{L}/2$ to approximate $\log p(\boldsymbol{y})$ when stopping. Since stopping occurs when log-determinant and quadratic form evolve roughly linearly, the two estimators are not far off each other. The main reason for using the biased estimator is of technical nature: for auto-differentiation, it is easier and faster to implement a custom backward function which can handle the in-place operations of our Cholesky implementation. This custom backward function needs roughly a factor two of the computation of $\log p(\boldsymbol{y})$ whereas the Torch-default needs a factor six. This shows that when comparing to exact inference, auto-differentiation can be disadvantageous and make the Cholesky appear slower than it is. Regarding CGLB, computation time is not dominated by the gradient but only the function evaluation itself.

B Additional results

In this section, we report additional results for both the hyper-parameter tuning experiments (section B.1) as well as plots to show the quality of the bounds on both the log-determinant term and the quadratic term (Appendices B.3.1 to B.3.4). Appendix B.4 shows how the bounds evolve when aggregated.

B.1 Additional results for hyper-parameter tuning

Denote with N_* the number of test instances, and with μ and σ^2 the mean and variance approximations of a method. As performance metrics we use root mean square error (RMSE)

$$\sqrt{\frac{1}{N^*}\sum_{n=1}^{N^*}(y_n^* - \mu(\boldsymbol{x}_n^*))^2}\ ,$$

negative log predictive density (NLPD)

$$\frac{1}{2N^*}\sum_{n=1}^{N^*}\frac{(y_n^* - \mu(\boldsymbol{x}_n^*))^2}{\sigma^2(\boldsymbol{x}_n^*)} + \log\left(2\pi\sigma^2(\boldsymbol{x}_n^*)\right)\ ,$$

and the negative marginal log likelihood $-\log p(\boldsymbol{y})$. Tables 2 and 3 summarize the results reported for each dataset, averaging over the outcomes of the final optimization step of each repetition. For each metric, we indicate whether a higher (↑) or lower (↓) value indicates a better result.

The results for the exact GP regression are marked in italics to emphasize that these are results we are trying to approach, not to beat. As the other methods are all approximations to the exact GP, there is little hope of achieving better performance. The best result among the approximation methods for each dataset is highlighted in bold.

Table 2: Summary of the CPU hyper-parameter tuning results from Section 4.2. For each metric, we report its final value over the course of optimization. For SVGP, we did not compute the exact marginal log-likelihoods, to save cluster time.

Dataset	**Model**	**RMSE** / 10^{-2} (↓)	**NLPD** / 10^{-1} (↓)	$\log p(\boldsymbol{y})$ / 10^4 (↑)
`metro`	*Exact*	*31.07 ± 13.90*	*−8.10 ± 9.55*	*−0.3247 ± 1.8943*
	ACGP	**37.99 ± 27.82**	**−7.64 ± 10.59**	**−0.3894 ± 1.9981**
	CGLB	38.10 ± 7.15	7.09 ± 0.92	−2.4484 ± 0.1567
	SVGP (512)	94.29 ± 0.55	13.61 ± 0.06	−4.4089 ± 0.0150
	SVGP (1024)	93.29 ± 0.32	13.50 ± 0.03	−4.3881 ± 0.0049
	SVGP (2048)	92.34 ± 0.33	13.40 ± 0.03	−4.3685 ± 0.0055
`pm25`	*Exact*	*42.70 ± 0.00*	*2.78 ± 0.00*	*−1.9396 ± 0.0000*
	ACGP	44.45 ± 1.40	**3.24 ± 0.42**	**−1.9243 ± 0.0307**
	CGLB	**43.96 ± 4.82**	7.07 ± 0.90	−2.2813 ± 0.2266
	SVGP (512)	81.20 ± 4.19	11.92 ± 0.27	−3.3763 ± 0.0798
	SVGP (1024)	73.60 ± 7.21	11.16 ± 0.97	−3.1765 ± 0.2001
	SVGP (2048)	60.25 ± 7.45	9.24 ± 1.08	−2.7404 ± 0.2288
`kin40k`	*Exact*	*7.41 ± 0.12*	*−12.36 ± 0.07*	*2.0837 ± 0.0063*
	ACGP	**7.41 ± 0.12**	**−12.36 ± 0.07**	**2.0837 ± 0.0063**
	CGLB	8.69 ± 0.15	−8.25 ± 0.03	1.6218 ± 0.0071
	SVGP (512)	16.60 ± 0.16	−2.56 ± 0.03	0.3064 ± 0.0085
	SVGP (1024)	14.01 ± 0.17	−4.17 ± 0.03	0.6147 ± 0.0079
	SVGP (2048)	12.12 ± 0.18	−5.64 ± 0.04	0.8884 ± 0.0073
`protein`	*Exact*	*55.76 ± 0.61*	*6.51 ± 0.46*	*−2.3686 ± 0.0355*
	ACGP	**55.82 ± 0.58**	**6.53 ± 0.45**	**−2.3663 ± 0.0349**
	CGLB	56.86 ± 0.47	8.33 ± 0.07	−2.7662 ± 0.0119
	SVGP (512)	64.86 ± 0.34	9.85 ± 0.04	−3.0941 ± 0.0110
	SVGP (1024)	62.21 ± 0.35	9.41 ± 0.04	−2.9974 ± 0.0112
	SVGP (2048)	60.04 ± 0.38	9.00 ± 0.05	−2.9043 ± 0.0104

Table 3: Summary of the GPU hyper-parameter tuning results from Section 4.2. For each metric, we report its final value over the course of optimization. We did not compute the exact marginal log-likelihoods, to save cluster time.

Dataset	**Model**	**RMSE** / 10^{-2} (↓)	**NLPD** / 10^{-1} (↓)	$\log p(\boldsymbol{y})$ / 10^4 (↑)
bike	*Exact*	*0.06 ± 0.03*	*−50.53 ± 0.06*	*4.9364 ± 0.0076*
	ACGP	0.90 ± 1.71	**−50.51 ± 0.16**	**4.9168 ± 0.0144**
	CGLB	**0.50 ± 0.32**	−38.06 ± 0.36	3.8664 ± 0.0526
	SVGP (512)	1.64 ± 0.22	−24.11 ± 0.49	2.5054 ± 0.0286
	SVGP (1024)	1.23 ± 0.23	−27.19 ± 0.59	2.7762 ± 0.0507
	SVGP (2048)	1.03 ± 0.23	−30.39 ± 0.21	3.0345 ± 0.0098
poletelecomm	*Exact*	*8.07 ± 0.56*	*−9.99 ± 2.01*	*0.8801 ± 0.1089*
	ACGP	**7.37 ± 0.09**	**−12.30 ± 0.18**	**1.0164 ± 0.0063**
	CGLB	7.74 ± 0.15	−11.41 ± 0.11	0.9262 ± 0.0057
	SVGP (512)	46.22 ± 43.71	1.10 ± 10.67	−0.1927 ± 1.0013
	SVGP (1024)	9.23 ± 0.21	−9.08 ± 0.11	0.7395 ± 0.0072
	SVGP (2048)	8.29 ± 0.19	−10.47 ± 0.13	0.8490 ± 0.0068
elevators	*Exact*	*35.12 ± 0.15*	*3.77 ± 0.07*	*−0.4671 ± 0.0018*
	ACGP	**35.10 ± 0.14**	**3.76 ± 0.07**	**−0.4671 ± 0.0020**
	CGLB	35.29 ± 0.22	3.81 ± 0.07	−0.4677 ± 0.0019
	SVGP (512)	37.79 ± 2.30	4.26 ± 0.21	−0.5060 ± 0.0284
	SVGP (1024)	35.66 ± 0.30	3.91 ± 0.07	−0.4724 ± 0.0064
	SVGP (2048)	35.48 ± 0.31	3.86 ± 0.08	−0.4701 ± 0.0065
pumadyn	*Exact*	*21.89 ± 0.96*	*−0.97 ± 0.50*	*0.0342 ± 0.0312*
	ACGP	**22.32 ± 1.01**	**−0.75 ± 0.52**	**0.0224 ± 0.0310**
	CGLB	40.65 ± 30.42	4.19 ± 6.92	−0.2480 ± 0.3776
	SVGP (512)	99.70 ± 1.44	14.16 ± 0.14	−0.7749 ± 0.0000
	SVGP (1024)	99.70 ± 1.44	14.16 ± 0.14	−0.7749 ± 0.0000
	SVGP (2048)	99.70 ± 1.44	14.16 ± 0.14	−0.7749 ± 0.0000

B.2 Additional plots for hyper-parameter tuning

The plots for the hyper-parameter optimization are shown in figures 1–12. Each point in the plots corresponds to one accepted optimization step for the given methods. Each point thus corresponds to a particular set of hyper-parameters during the optimization. In figures 5–12, we show the root-mean-square error, RMSE, that each methods obtains on the test set at each optimisation step. In figures 1–4, we show the log-marginal likelihood, $\log p(\boldsymbol{y})$, that an exact GP would have achieved with the specific set of hyper-parameters at each optimization step for each method.

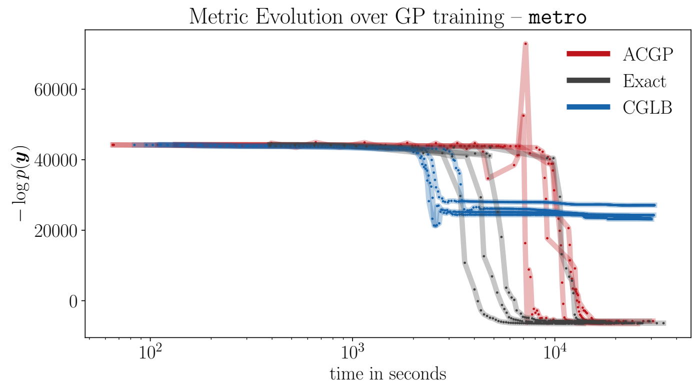


Figure 1: Log-marginal likelihood over time while optimizing hyper-parameters for the `metro` dataset.

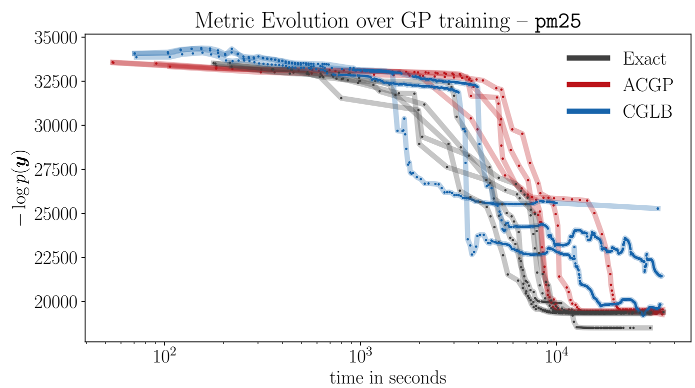


Figure 2: Log-marginal likelihood over time while optimizing hyper-parameters for the `pm25` dataset.

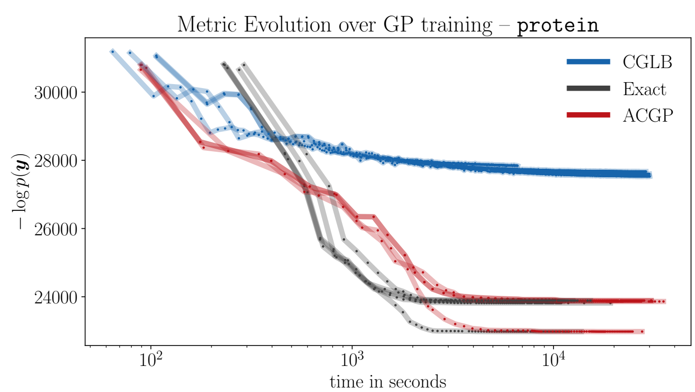


Figure 3: Log-marginal likelihood over time while optimizing hyper-parameters for the `protein` dataset.

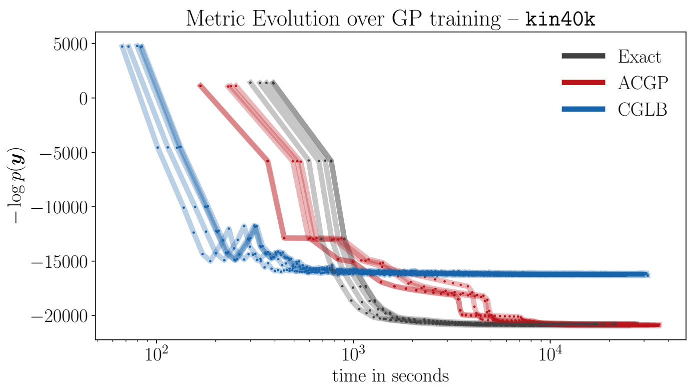


Figure 4: Log-marginal likelihood over time while optimizing hyper-parameters for the `kin40k` dataset.

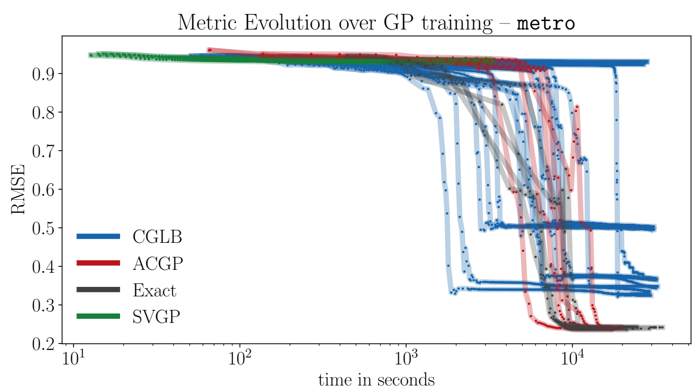


Figure 5: RMSE over time while optimizing hyper-parameters for the `metro` dataset.

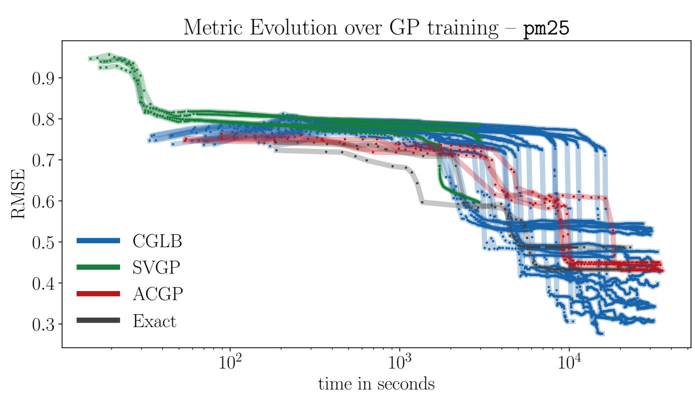


Figure 6: RMSE over time while optimizing hyper-parameters for the `pm25` dataset.

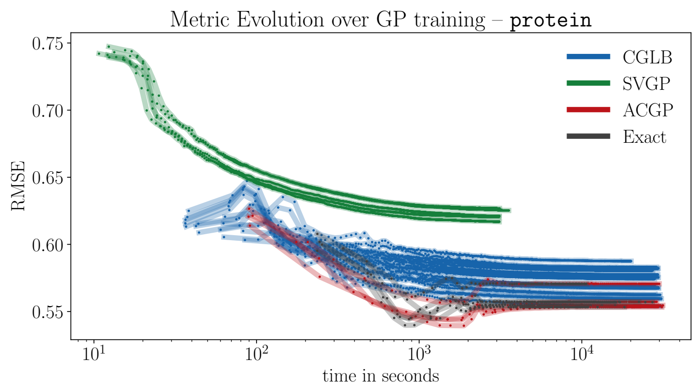


Figure 7: RMSE over time while optimizing hyper-parameters for the `protein` dataset.

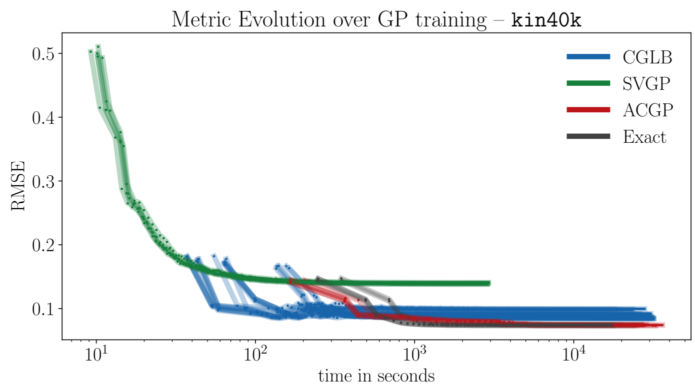


Figure 8: RMSE over time while optimizing hyper-parameters for the `kin40k` dataset.

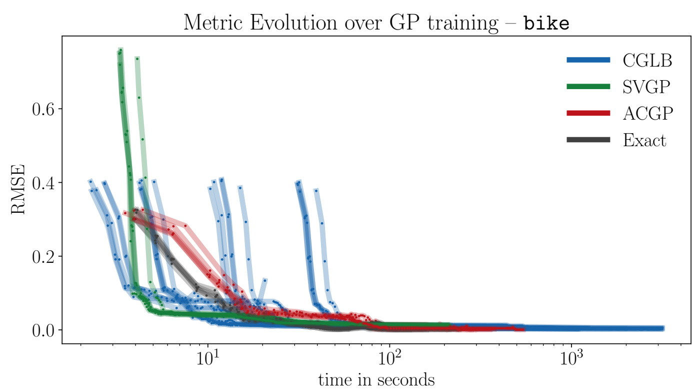


Figure 9: RMSE over time while optimizing hyper-parameters for the `bike` dataset.

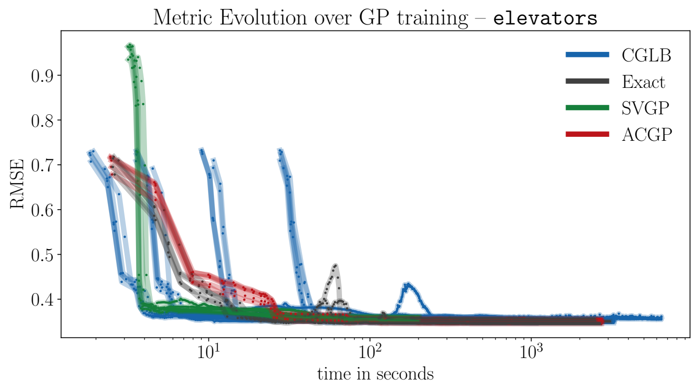


Figure 10: RMSE over time while optimizing hyper-parameters for the `elevators` dataset.

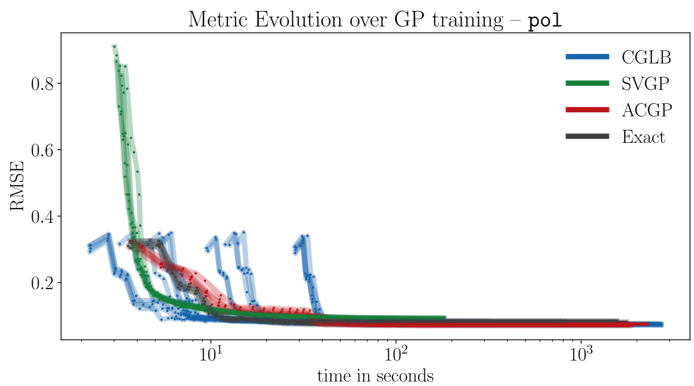


Figure 11: RMSE over time while optimizing hyper-parameters for the `pole` dataset.

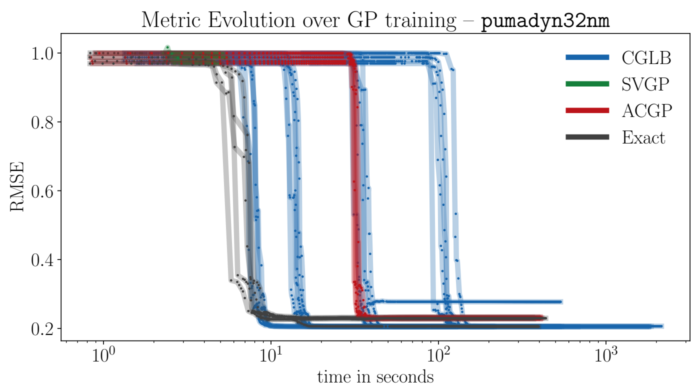
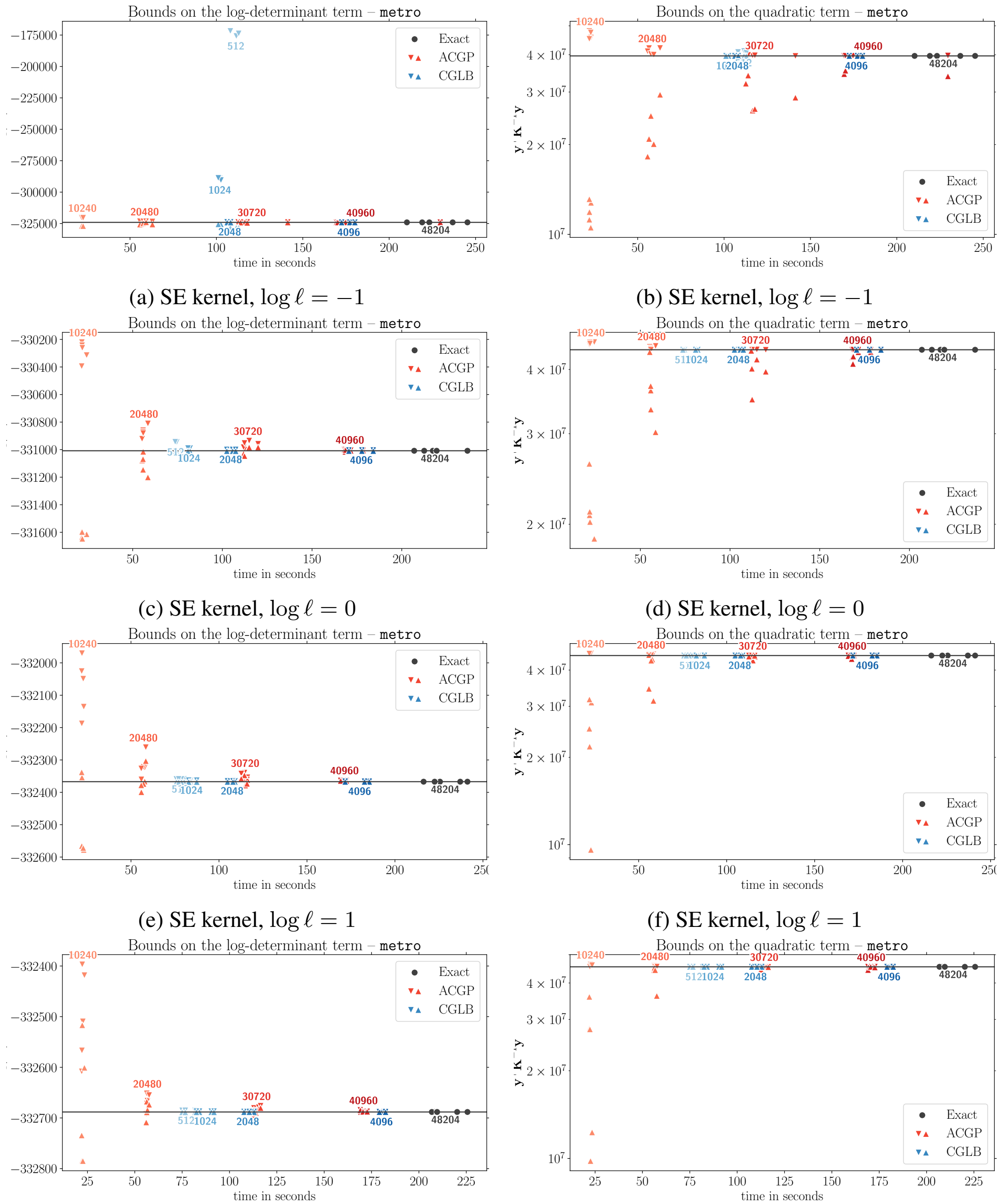


Figure 12: RMSE over time while optimizing hyper-parameters for the `pumadyn32nm` dataset.

B.3 Additional plots for the bound quality experiments

B.3.1 Bounds for experiments on `metro`



(a) SE kernel, $\log \ell = -1$

(b) SE kernel, $\log \ell = -1$

(c) SE kernel, $\log \ell = 0$

(d) SE kernel, $\log \ell = 0$

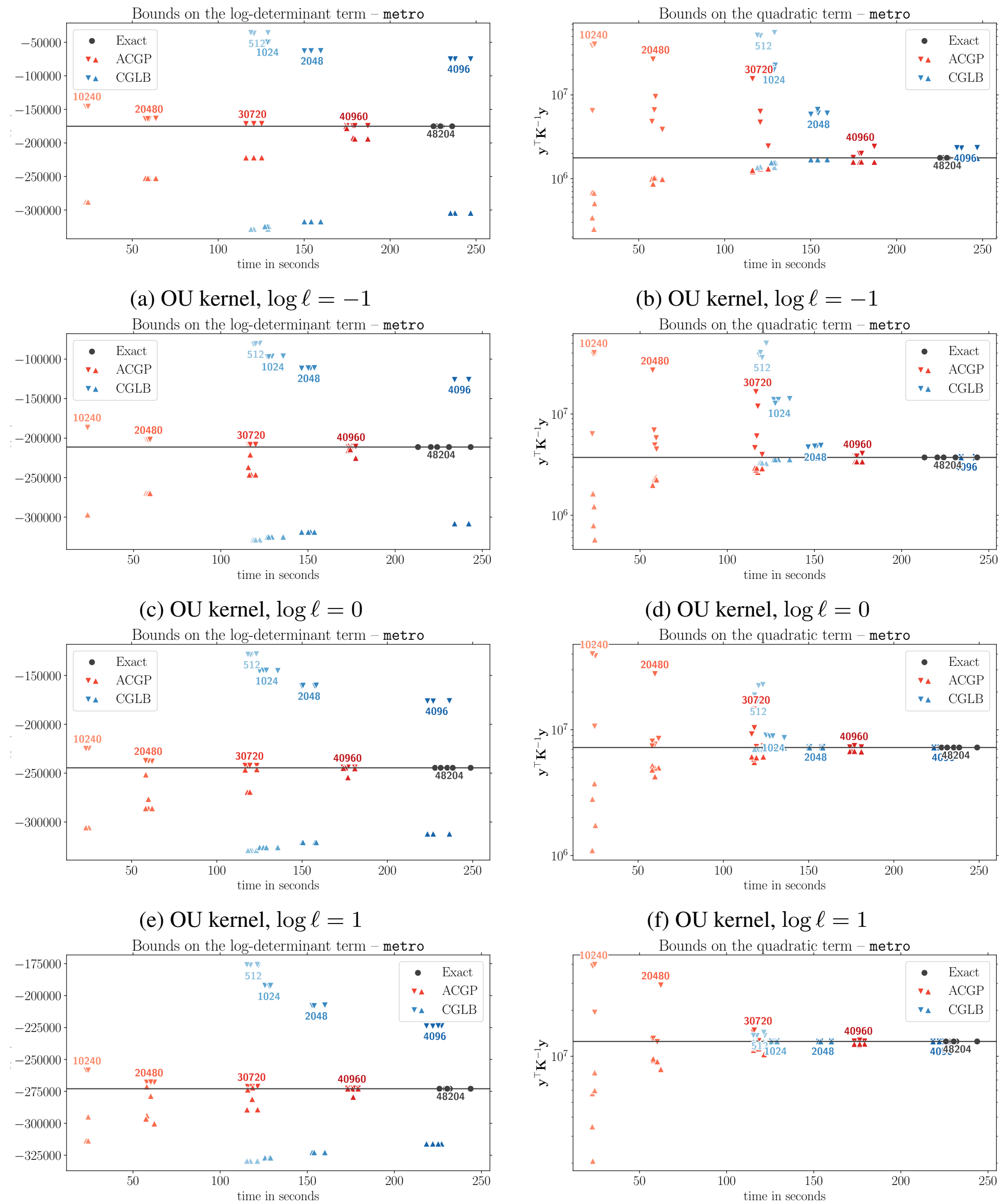
(e) SE kernel, $\log \ell = 1$

(f) SE kernel, $\log \ell = 1$

(g) SE kernel, $\log \ell = 2$

(h) SE kernel, $\log \ell = 2$

Figure 13: Upper and lower bounds on the log-determinant term (left column) and the quadratic term (right column) for the `metro` dataset when using a squared exponential (SE) kernel.



(a) OU kernel, $\log \ell = -1$

(b) OU kernel, $\log \ell = -1$

(c) OU kernel, $\log \ell = 0$

(d) OU kernel, $\log \ell = 0$

(e) OU kernel, $\log \ell = 1$

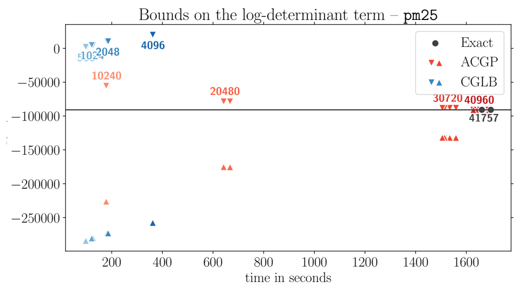
(f) OU kernel, $\log \ell = 1$

(g) OU kernel, $\log \ell = 2$

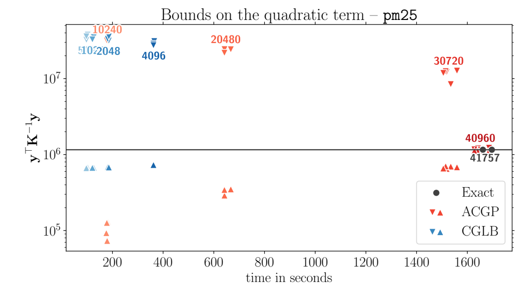
(h) OU kernel, $\log \ell = 2$

Figure 14: Upper and lower bounds on the log-determinant term (left column) and the quadratic term (right column) for the `metro` dataset using an Ornstein-Uhlenbeck (OU) kernel.

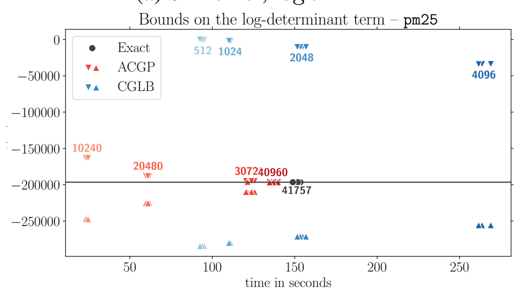
B.3.2 Bounds for experiments on pm25



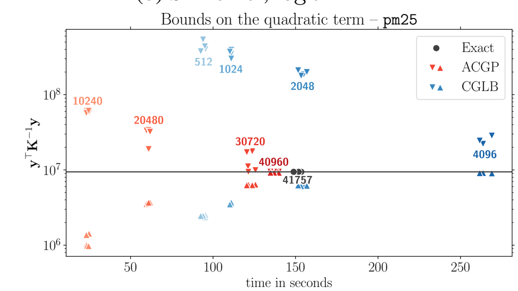
(a) SE kernel, $\log \ell = -1$



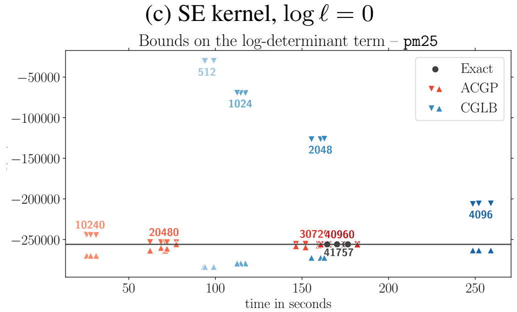
(b) SE kernel, $\log \ell = -1$



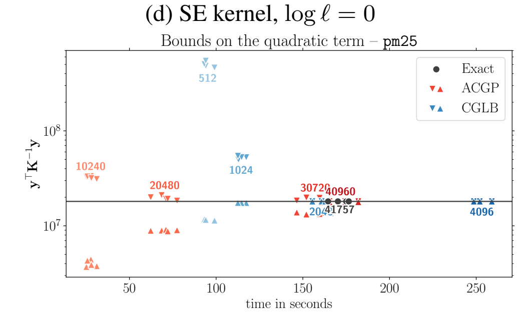
(c) SE kernel, $\log \ell = 0$



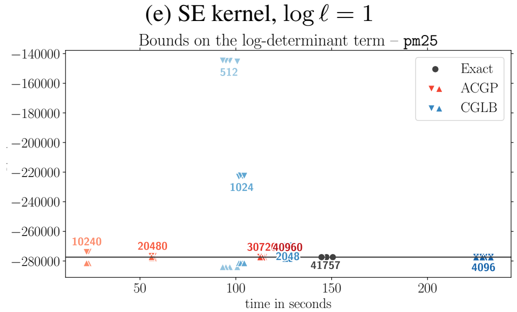
(d) SE kernel, $\log \ell = 0$



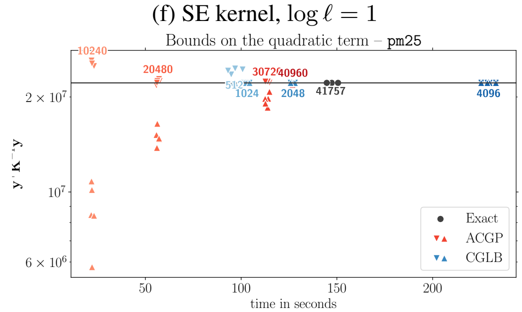
(e) SE kernel, $\log \ell = 1$



(f) SE kernel, $\log \ell = 1$

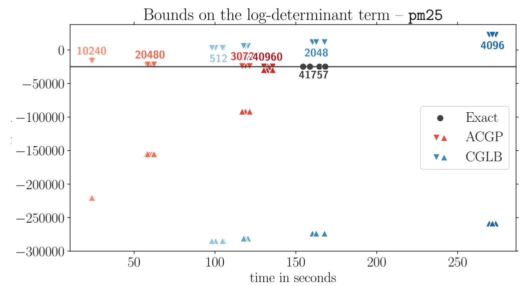


(g) SE kernel, $\log \ell = 2$

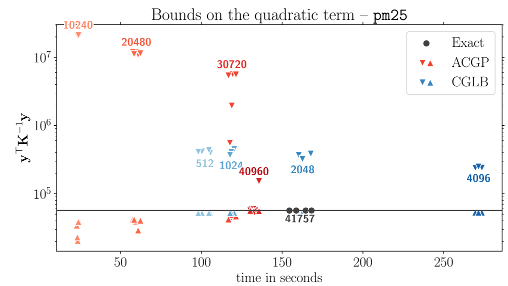


(h) SE kernel, $\log \ell = 2$

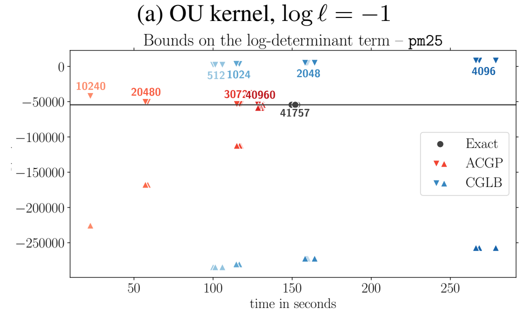
Figure 15: Upper and lower bounds on the log-determinant term (left column) and the quadratic term (right column) for the pm25 dataset.



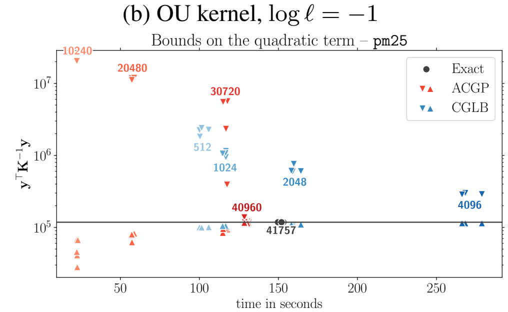
(a) OU kernel, $\log \ell = -1$



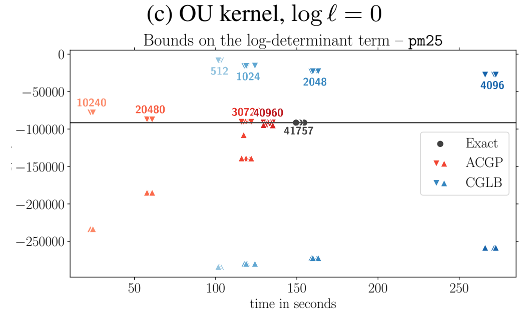
(b) OU kernel, $\log \ell = -1$



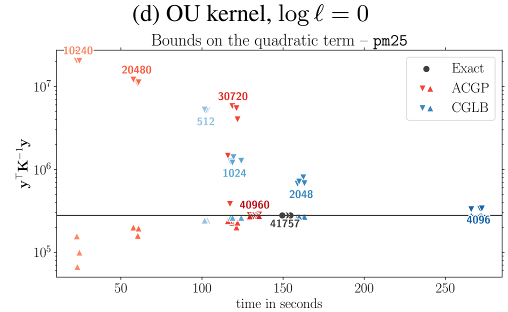
(c) OU kernel, $\log \ell = 0$



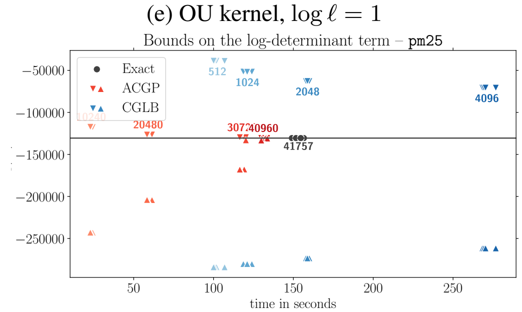
(d) OU kernel, $\log \ell = 0$



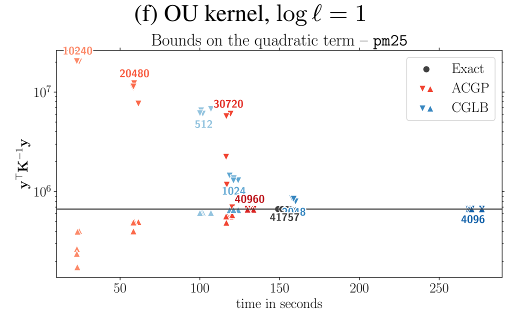
(e) OU kernel, $\log \ell = 1$



(f) OU kernel, $\log \ell = 1$



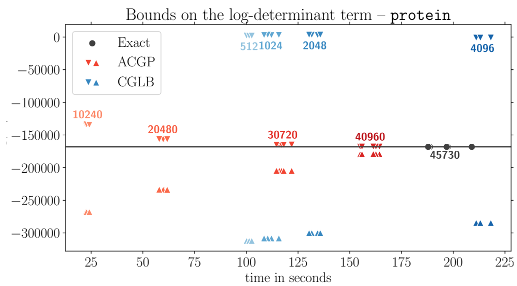
(g) OU kernel, $\log \ell = 2$



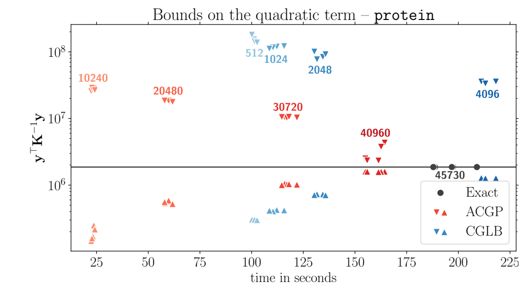
(h) OU kernel, $\log \ell = 2$

Figure 16: Upper and lower bounds on the log-determinant term (left column) and the quadratic term (right column) for the pm25 dataset using an Ornstein-Uhlenbeck (OU) kernel.

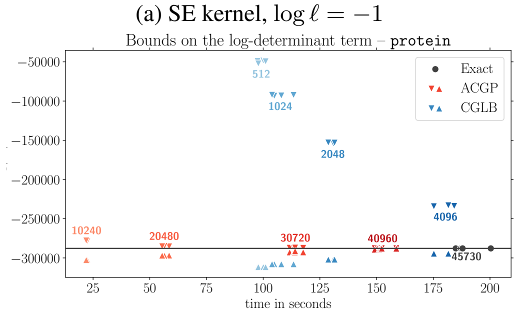
B.3.3 Bounds for experiments on `protein`



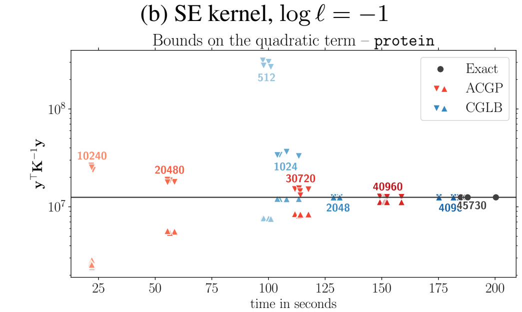
(a) SE kernel, $\log \ell = -1$



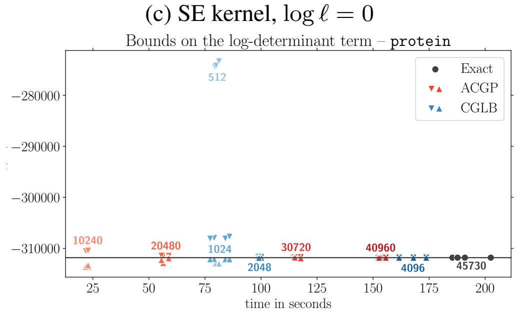
(b) SE kernel, $\log \ell = -1$



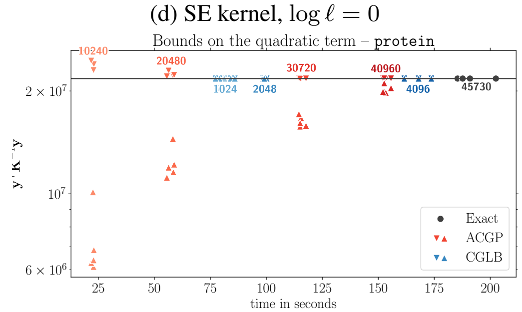
(c) SE kernel, $\log \ell = 0$



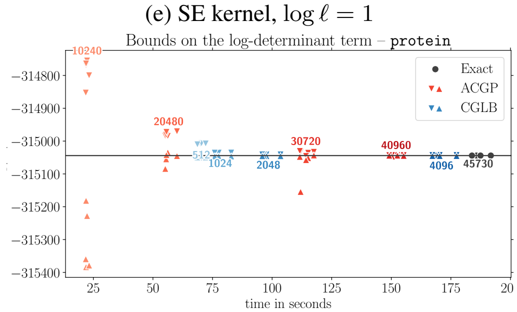
(d) SE kernel, $\log \ell = 0$



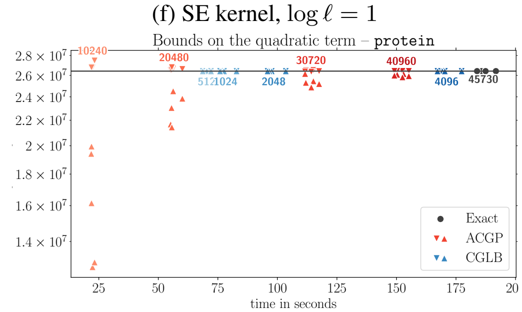
(e) SE kernel, $\log \ell = 1$



(f) SE kernel, $\log \ell = 1$

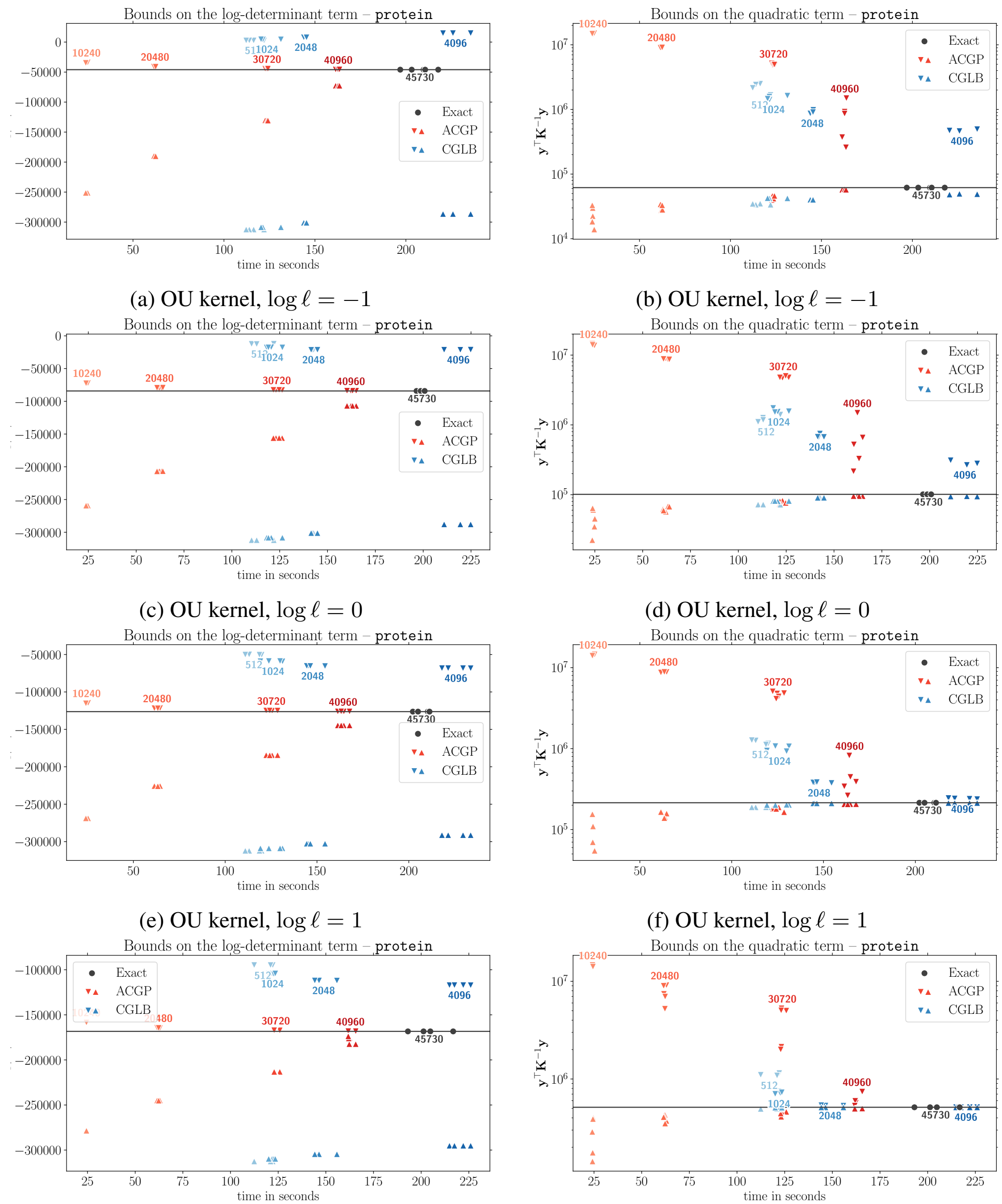


(g) SE kernel, $\log \ell = 2$



(h) SE kernel, $\log \ell = 2$

Figure 17: Upper and lower bounds on the log-determinant term (left column) and the quadratic term (right column) for the `protein` dataset.



(a) OU kernel, $\log \ell = -1$

(b) OU kernel, $\log \ell = -1$

(c) OU kernel, $\log \ell = 0$

(d) OU kernel, $\log \ell = 0$

(e) OU kernel, $\log \ell = 1$

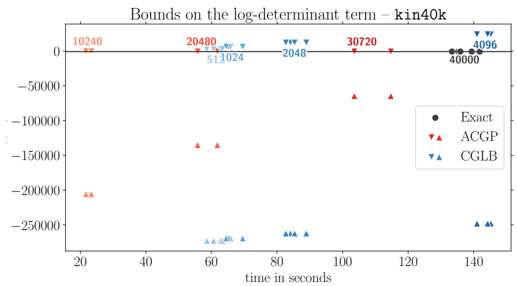
(f) OU kernel, $\log \ell = 1$

(g) OU kernel, $\log \ell = 2$

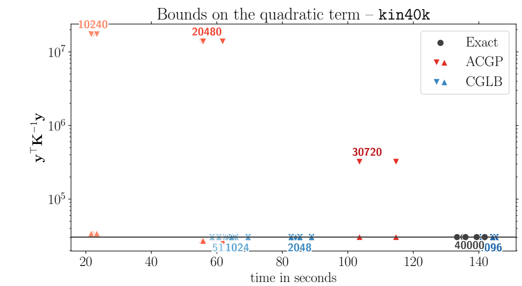
(h) OU kernel, $\log \ell = 2$

Figure 18: Upper and lower bounds on the log-determinant term (left column) and the quadratic term (right column) for the `protein` dataset using an Ornstein-Uhlenbeck (OU) kernel.

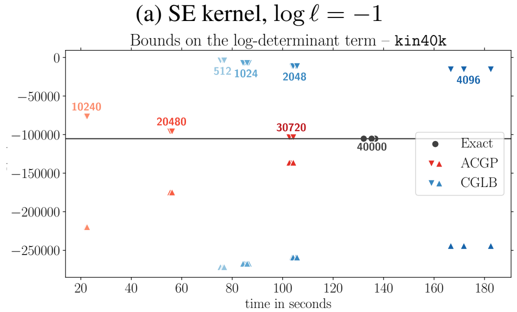
114 B.3.4 Bounds for experiments on `kin40k`



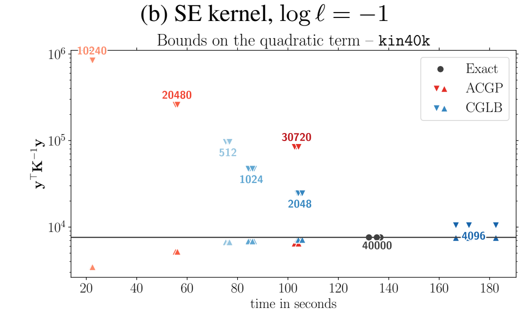
(a) SE kernel, $\log \ell = -1$



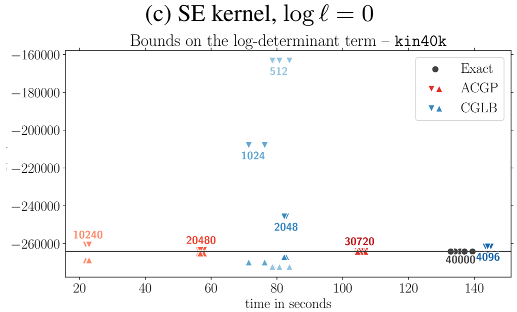
(b) SE kernel, $\log \ell = -1$



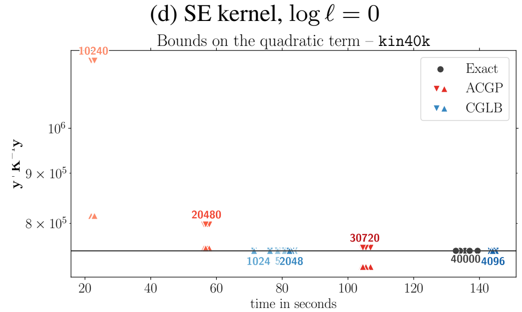
(c) SE kernel, $\log \ell = 0$



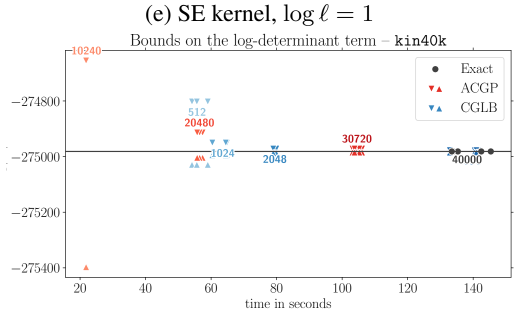
(d) SE kernel, $\log \ell = 0$



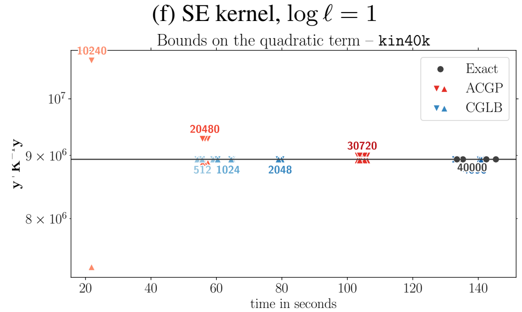
(e) SE kernel, $\log \ell = 1$



(f) SE kernel, $\log \ell = 1$

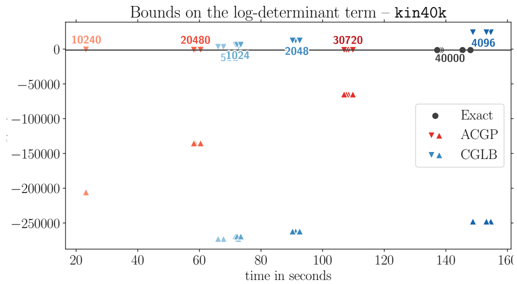


(g) SE kernel, $\log \ell = 2$

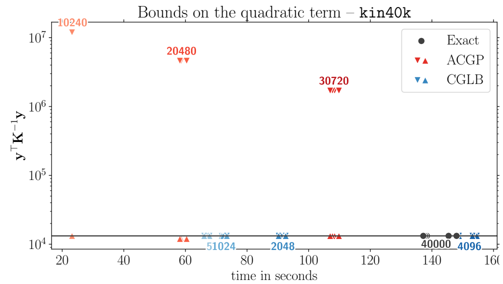


(h) SE kernel, $\log \ell = 2$

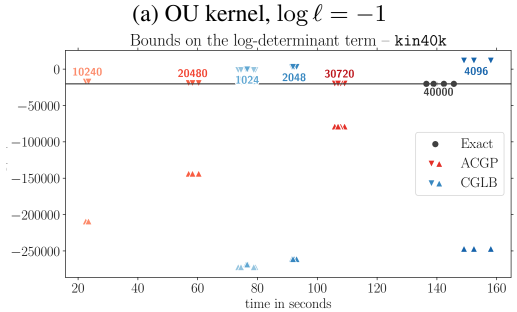
Figure 19: Upper and lower bounds on the log-determinant term (left column) and the quadratic term (right column) for the `kin40k` dataset.



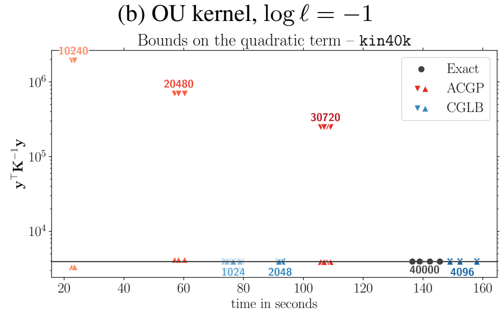
(a) OU kernel, $\log \ell = -1$



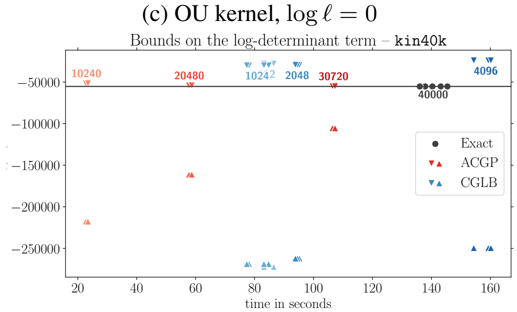
(b) OU kernel, $\log \ell = -1$



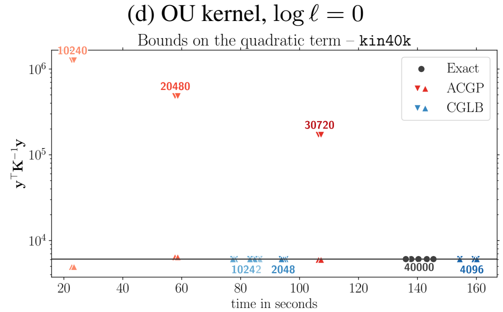
(c) OU kernel, $\log \ell = 0$



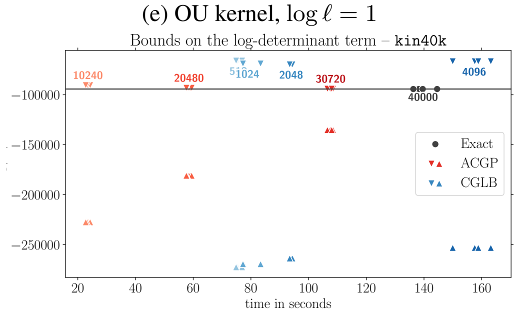
(d) OU kernel, $\log \ell = 0$



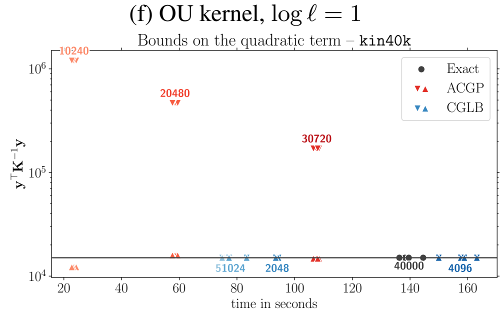
(e) OU kernel, $\log \ell = 1$



(f) OU kernel, $\log \ell = 1$



(g) OU kernel, $\log \ell = 2$

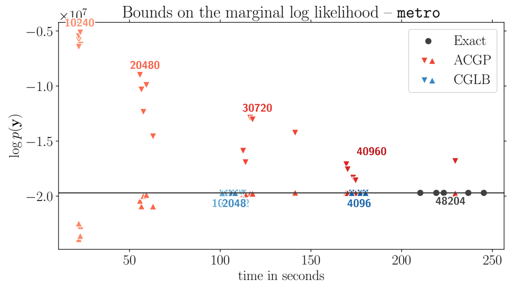


(h) OU kernel, $\log \ell = 2$

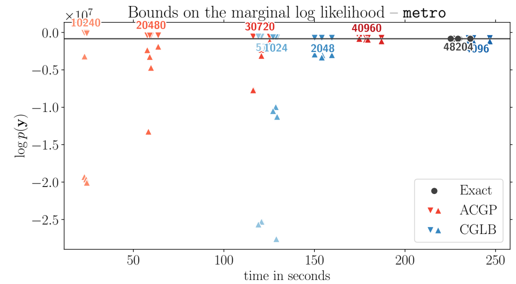
Figure 20: Upper and lower bounds on the log-determinant term (left column) and the quadratic term (right column) for the `kin40k` dataset using an Ornstein-Uhlenbeck (OU) kernel.

B.4 Aggregated plots for the bound quality experiments

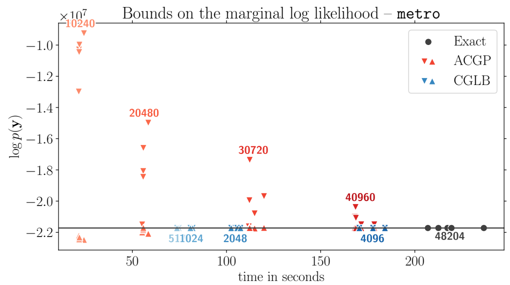
B.4.1 Bounds for experiments on `metro`



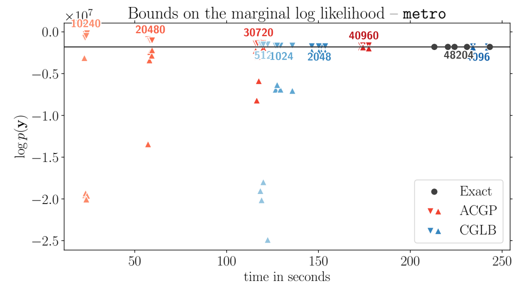
(a) SE kernel, $\log \ell = -1$



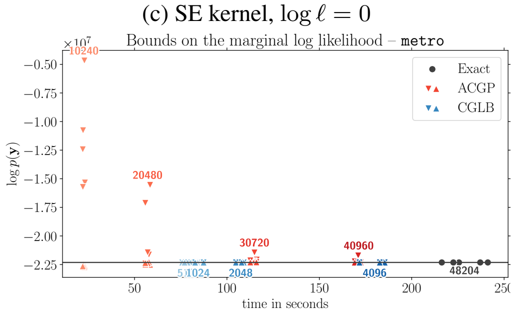
(b) OU kernel, $\log \ell = -1$



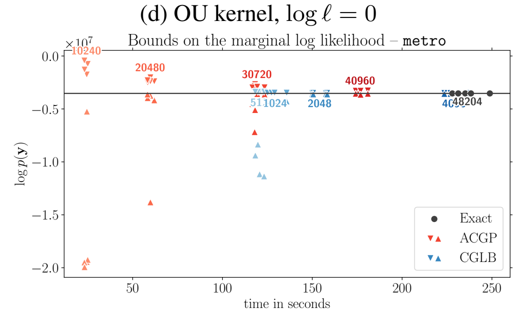
(c) SE kernel, $\log \ell = 0$



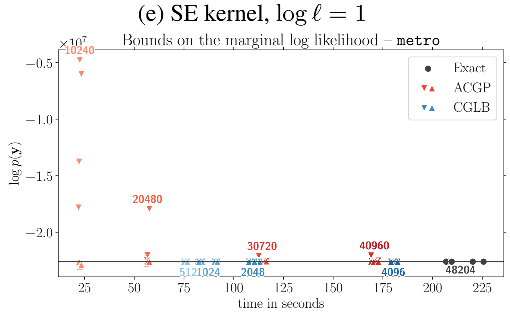
(d) OU kernel, $\log \ell = 0$



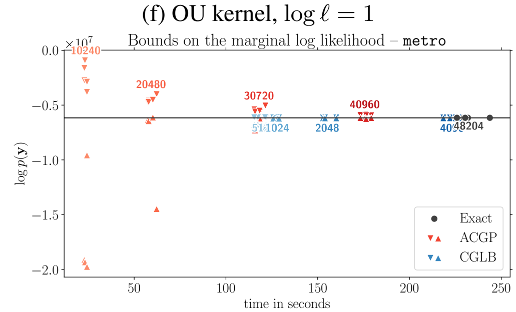
(e) SE kernel, $\log \ell = 1$



(f) OU kernel, $\log \ell = 1$



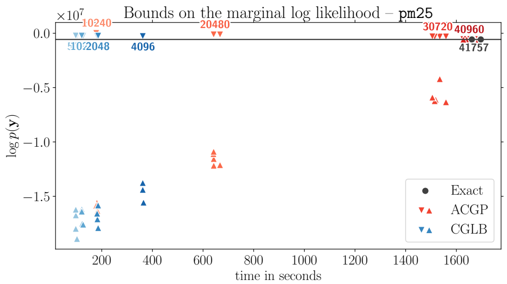
(g) SE kernel, $\log \ell = 2$



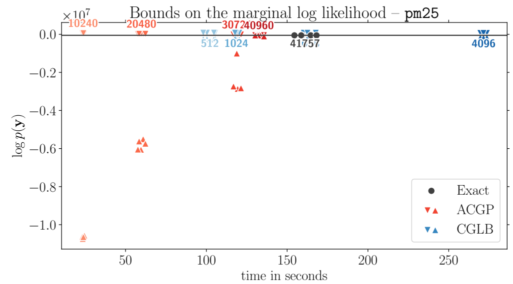
(h) OU kernel, $\log \ell = 2$

Figure 21: Upper and lower bounds on the marginal log-likeihood for the `metro` dataset when using a squared exponential (SE) kernel (left column) and the Ornstein-Uhlenbeck (OU) kernel (right column).

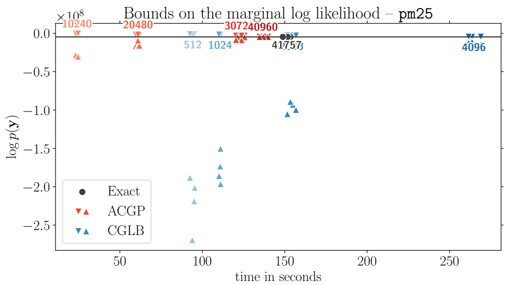
B.4.2 Bounds for experiments on pm25



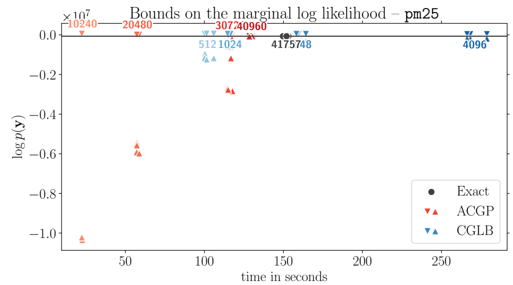
(a) SE kernel, $\log \ell = -1$



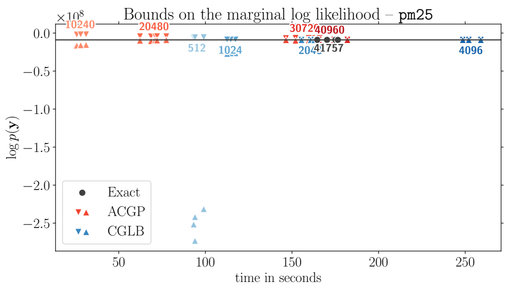
(b) OU kernel, $\log \ell = -1$



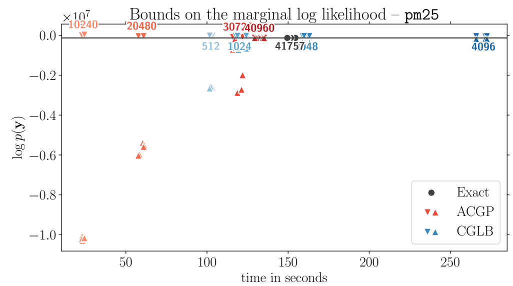
(c) SE kernel, $\log \ell = 0$



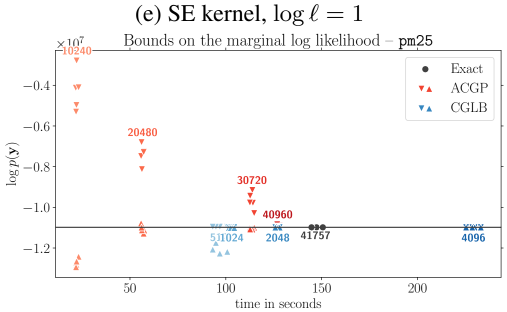
(d) OU kernel, $\log \ell = 0$



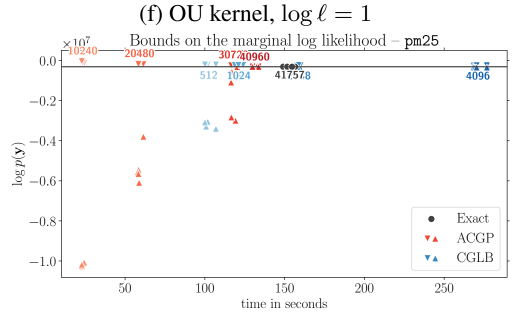
(e) SE kernel, $\log \ell = 1$



(f) OU kernel, $\log \ell = 1$



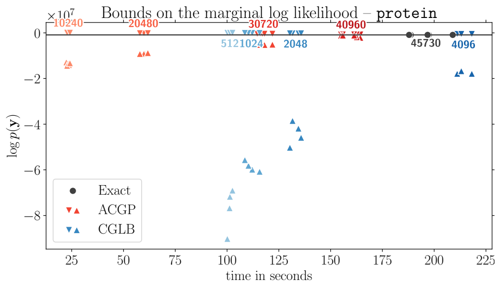
(g) SE kernel, $\log \ell = 2$



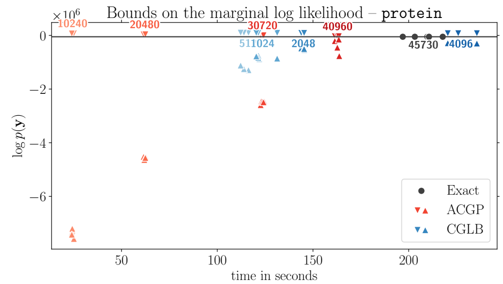
(h) OU kernel, $\log \ell = 2$

Figure 22: Upper and lower bounds on the marginal log-likeihood for the pm25 dataset when using a squared exponential (SE) kernel (left column) and the Ornstein-Uhlenbeck (OU) kernel (right column).

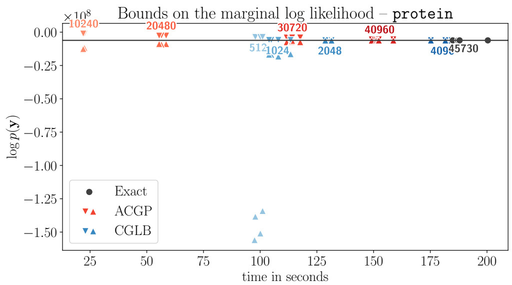
B.4.3 Bounds for experiments on `protein`



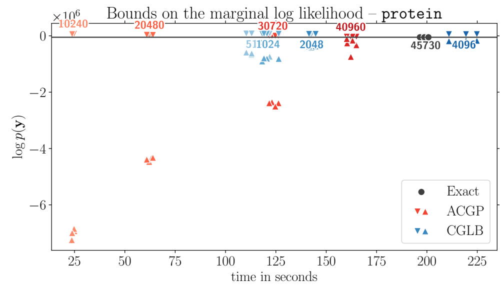
(a) SE kernel, $\log \ell = -1$



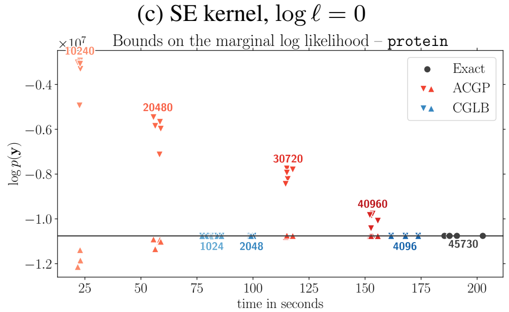
(b) OU kernel, $\log \ell = -1$



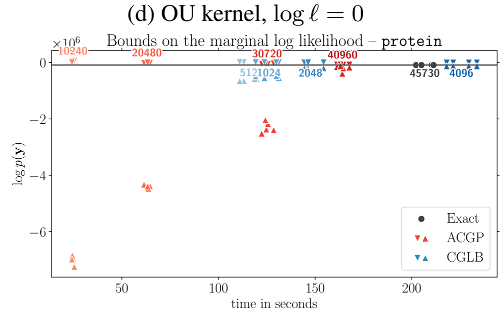
(c) SE kernel, $\log \ell = 0$



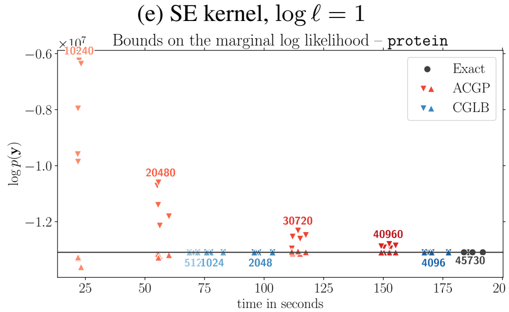
(d) OU kernel, $\log \ell = 0$



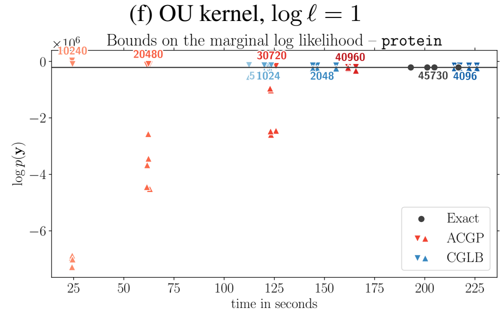
(e) SE kernel, $\log \ell = 1$



(f) OU kernel, $\log \ell = 1$



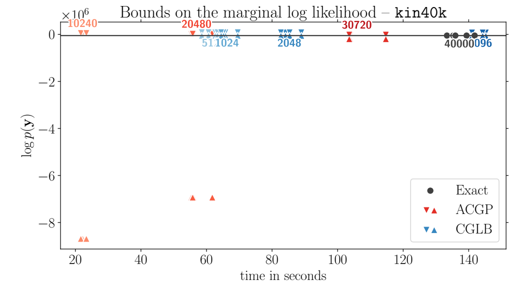
(g) SE kernel, $\log \ell = 2$



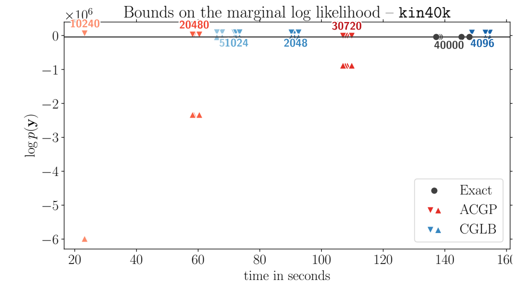
(h) OU kernel, $\log \ell = 2$

Figure 23: Upper and lower bounds on the marginal log-likeihood for the `protein` dataset when using a squared exponential (SE) kernel (left column) and the Ornstein-Uhlenbeck (OU) kernel (right column).

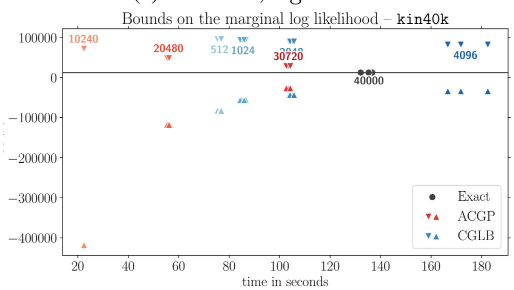
B.4.4 Bounds for experiments on `kin40k`



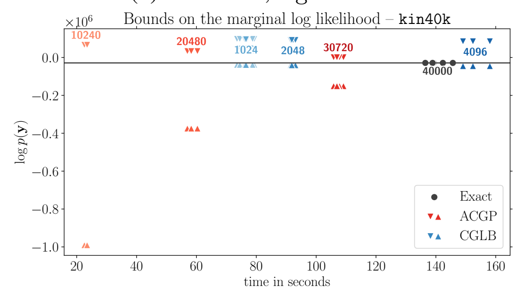
(a) SE kernel, $\log \ell = -1$



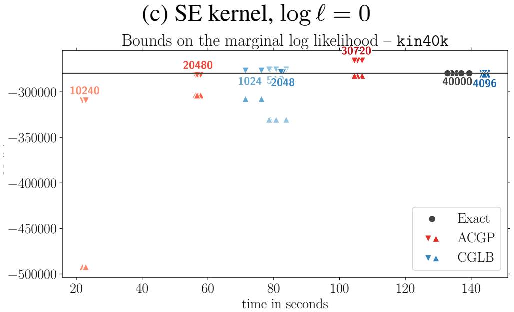
(b) OU kernel, $\log \ell = -1$



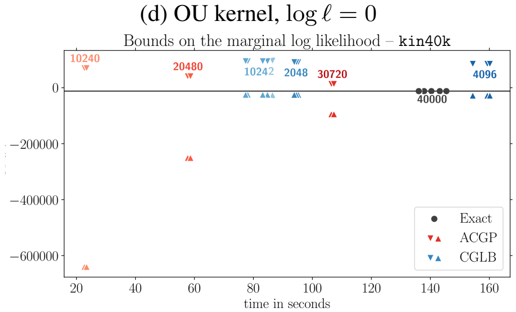
(c) SE kernel, $\log \ell = 0$



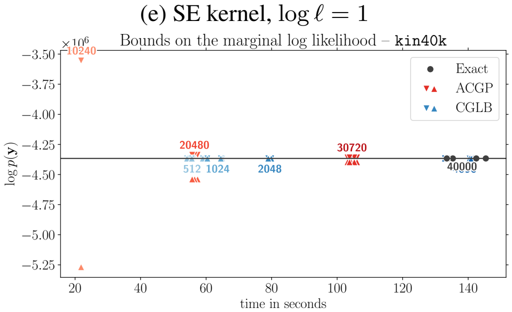
(d) OU kernel, $\log \ell = 0$



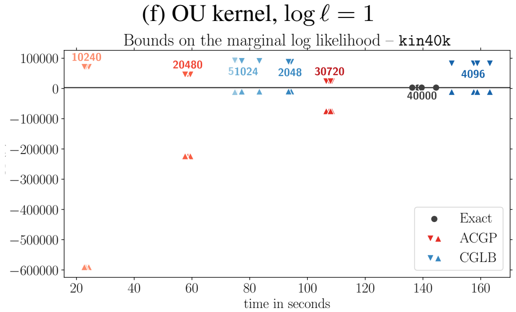
(e) SE kernel, $\log \ell = 1$



(f) OU kernel, $\log \ell = 1$



(g) SE kernel, $\log \ell = 2$



(h) OU kernel, $\log \ell = 2$

Figure 24: Upper and lower bounds on the marginal log-likeihood for the `kin40k` dataset when using a squared exponential (SE) kernel (left column) and the Ornstein-Uhlenbeck (OU) kernel (right column).

C Notation

We use a PYTHON-inspired index notation, abbreviating for example $[y_1, \ldots, y_n]^\top$ as $\boldsymbol{y}_{:n}$—observe that the indexing starts at 1. Indexing binds before any other operation such that $\boldsymbol{K}^{-1}_{:s,:s}$ is the inverse of $\boldsymbol{K}_{:s,:s}$ and *not* all elements up to s of $\boldsymbol{K}^{-1}$. Differing from the main paper, we assume a heteroskedastic noise model such that we exchange $\sigma^2 \in \mathbb{R}^+$ for a function of the inputs, $\sigma^2 : \mathbb{X} \to \mathbb{R}^+$. With σ^2 we will refer to $\inf_{\boldsymbol{x}\in\mathbb{X}} \sigma^2(\boldsymbol{x})$, which we assume to be strictly larger than 0. For $s \in \{1, \ldots, N\}$ define $\mathcal{F}_s := \sigma(\boldsymbol{x}_1, y_1, \ldots, \boldsymbol{x}_s, y_s)$ to be the σ-algebra generated by $\boldsymbol{x}_1, y_1, \ldots, \boldsymbol{x}_s, y_s$. With respect to the main article, we change the letter M to t. The motivation for the former notation is to highlight the role of the variable as a subset size, whereas in this part, the focus is on M as a stopping time.

D Proof Sketch

In this section of the appendix, we provide additional details, proofs and theorems on the proposed formulation. The principal equations included in Sec. 3 of the main manuscript are also included here for a better comprehension.

D.1 The cumulative perspective

The key issue this paper is concerned with is how to estimate the full marginal likelihood, $p(\boldsymbol{y})$, given only a subset of n observations and their combined marginal likelihood, $p(\boldsymbol{y}_{:n})$. In particular, we will derive bounds, which are functions of seen observations, on this estimate. These bounds will allow us to decide, on the fly, when we have seen enough observations to accurately estimate the full marginal likelihood.

We can write $\log p(\boldsymbol{y})$ equivalently as

$$\log p(\boldsymbol{y}) = \sum_{n=1}^{N} \log p(y_n \mid \boldsymbol{y}_{:n-1}). \tag{3}$$

With this equation in hand, the phenomena shown in Fig. 1 of the main manuscript becomes much clearer: The figure shows the value of Equation (3) for an increasing number of observations N. When the plot exhibits a linear trend it is because the summands $\log p(y_n \mid \boldsymbol{y}_{:n-1})$ become approximately constant, implying that the model is not gaining additional knowledge after the nth observation.[1] From this perspective, we can craft an approximation by an *optimal stopping problem*: after processing observation n, we may decide whether to continue processing the sum or whether to stop and to estimate the remaining $N - n$ terms.

D.2 Extrapolation

For each potential stopping point t we can decompose Equation (3) into a sum of terms which have already been computed and a remaining sum

$$\log p(\boldsymbol{y}) = \underbrace{\sum_{n=1}^{t} \log p(y_n \mid \boldsymbol{y}_{:n-1})}_{A\text{: processed}} + \underbrace{\sum_{n=t+1}^{N} \log p(y_n \mid \boldsymbol{y}_{:n-1})}_{B\text{: remaining}}.$$

It is tempting to estimate B as $\frac{N-t}{t}A$, yet this is estimator is biased. In the following, we will derive lower and upper bounds, $\mathcal{L}_t$ and $\mathcal{U}_t$, such that conditioned on the points already processed, B can be sandwiched,

$$\mathbb{E}[\mathcal{L}_t \mid \boldsymbol{x}_1, y_1, \ldots \boldsymbol{x}_t, y_t] \leq \mathbb{E}[B \mid \boldsymbol{x}_1, y_1, \ldots \boldsymbol{x}_t, y_t] \leq \mathbb{E}[\mathcal{U}_t \mid \boldsymbol{x}_1, y_1, \ldots \boldsymbol{x}_t, y_t]. \tag{4}$$

These bounds tighten as we increase the number of observations, which allow us to monitor convergence of the approximation. We can then detect when the upper and lower bounds are sufficiently

[1] An alternative way of understanding the linear trend is that the spectrum of the covariance matrix $\boldsymbol{K}$ typically drop to σ^2 at some point; since the log-determinant is the sum of the log-eigenvalues then the linear trend comes from additional $2\log\sigma$ terms in the sum.

near each other, and stop computations early when the approximation is sufficiently good. This is in contrast to other approximations, where one specifies a computational budget, rather than a desired accuracy.

D.3 General bounds

A more practical-minded reader may safely skip this section and continue in Appendix D.6 where we show how to use the bounds to obtain a stopped Cholesky decomposition. Recall that we want to detect the case when the log-marginal likelihood starts to behave linearly with more processed datapoints as shown in Fig. 1 in the main manuscript. That is to say, the bounds presented in the following are valid in general but useful only in the linear setting.

The posterior of the nth observation conditioned on the previous is Gaussian with

$$\begin{aligned} p(y_n \mid \boldsymbol{y}_{:n-1}) =& \mathcal{N}(m_{n-1}(\boldsymbol{x}_n), k_{n-1}(\boldsymbol{x}_n, \boldsymbol{x}_n) + \sigma^2(\boldsymbol{x}_n)) \\ m_{n-1}(\boldsymbol{x}_n) :=& k(\boldsymbol{x}_n, \boldsymbol{X}_{:n-1})\boldsymbol{K}_{n-1}^{-1}\boldsymbol{y}_{:n-1} \\ k_{n-1}(\boldsymbol{x}_n, \boldsymbol{x}_n) :=& k(\boldsymbol{x}_n, \boldsymbol{x}_n) - k(\boldsymbol{x}_n, \boldsymbol{X}_{:n-1})\boldsymbol{K}_{n-1}^{-1}k(\boldsymbol{X}_{:n-1}, \boldsymbol{x}_n), \end{aligned}$$

where we assumed (w.l.o.g) that $\mu_0(\boldsymbol{x}) := 0$. Inspecting these expressions one finds that

$$\log\det \boldsymbol{K}_N = \sum_{n=1}^{N} \log\left(k_{n-1}(\boldsymbol{x}_n, \boldsymbol{x}_n) + \sigma^2(\boldsymbol{x}_n)\right), \tag{5}$$

$$\boldsymbol{y}^\top \boldsymbol{K}_N^{-1}\boldsymbol{y} = \sum_{n=1}^{N} \frac{(y_n - m_{n-1}(\boldsymbol{x}_n))^2}{k_{n-1}(\boldsymbol{x}_n, \boldsymbol{x}_n) + \sigma^2(\boldsymbol{x}_n)}. \tag{6}$$

These expressions are permutation invariant. This allows us to prove that these terms cannot be too far from their expected values using a *Hoeffding's inequality for super-martingales* by Fan et al. (2012). This observation also holds for the conditional case, that is, after having observed $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_t, y_t)$. Taking for granted that we can bound $\mathbb{P}\left(|\log p(\boldsymbol{y}) - \mathbb{E}[\log p(\boldsymbol{y}) \mid \boldsymbol{x}_1, y_1, \ldots, \boldsymbol{x}_\tau, y_\tau]| > \epsilon\right)$ for stopping times τ, we proceed with the estimation of the expectation.

Recall that $\frac{N-t}{t}\log p(\boldsymbol{y}_{:t})$ is *not* an unbiased estimator for $\mathbb{E}[\log p(\boldsymbol{y}) \mid \boldsymbol{x}_1, y_1, \ldots, \boldsymbol{x}_t, y_t]$, due to the interaction of $\boldsymbol{x}_{t+1}, y_{t+1}, \ldots, \boldsymbol{x}_N, y_N$. Our strategy is to find function families u (and l) which upper (and lower) bound the expectation

$$\begin{aligned} l^d_{n,t} \leq_E \log k_{n-1}(\boldsymbol{x}_n, \boldsymbol{x}_n) &\leq_E u^d_{n,t} \\ l^q_{n,t} \leq_E \frac{(y_n - m_{n-1}(\boldsymbol{x}_n))^2}{k_{n-1}(\boldsymbol{x}_n, \boldsymbol{x}_n) + \sigma^2(\boldsymbol{x}_n)} &\leq_E u^q_{n,t}, \end{aligned}$$

where $\leq_E$ denotes that the inequality holds in expectation. We will choose the function families such that the unseen variables interact only in a *controlled* manner. More specifically,

$$f^x_{n,t}(\boldsymbol{x}_n, y_n, \ldots, \boldsymbol{x}_1, y_1) = \sum_{j=s+1}^{n} g_t^{f,x}(\boldsymbol{z}_n, \boldsymbol{z}_j; \boldsymbol{z}_1, \ldots \boldsymbol{z}_s),$$

with $f \in \{u, l\}$ and $x \in \{d, q\}$. The effect of this restriction becomes apparent when taking the expectation. The sum over the bounds becomes the sum of only two terms: variance and covariance, formally:

$$\mathbb{E}\left[\sum_{n=s+1}^{N} f^x_{n,t}(\boldsymbol{z}_n, \ldots, \boldsymbol{z}_1) \mid \sigma(\boldsymbol{z}_1, \ldots, \boldsymbol{z}_s)\right] \tag{7}$$

$$\begin{aligned} &= (N-s)\,\mathbb{E}\left[g(\boldsymbol{z}_{s+1}, \boldsymbol{z}_{s+1}, \boldsymbol{z}_1 \ldots, \boldsymbol{z}_n) \mid \sigma(\boldsymbol{z}_1, \ldots, \boldsymbol{z}_s)\right] \\ &+ (N-s)\,\frac{N-s+1}{2}\mathbb{E}\left[g(\boldsymbol{z}_{s+1}, \boldsymbol{z}_{s+2}, \boldsymbol{z}_1 \ldots, \boldsymbol{z}_n) \mid \sigma(\boldsymbol{z}_1, \ldots, \boldsymbol{z}_s)\right]. \end{aligned} \tag{8}$$

We can estimate this expectation from the observations we obtained between s and t.

$$\approx \frac{N-t}{t-s} \sum_{n=s+1}^{t} g(\boldsymbol{z}_n, \boldsymbol{z}_n, \boldsymbol{z}_1 \dots, \boldsymbol{z}_s) \tag{9}$$
$$+ \frac{2(N-t)}{t-s} \frac{N-s+1}{2} \sum_{i=1}^{\frac{t-s}{2}} g(\boldsymbol{z}_{s+2i}, \boldsymbol{z}_{s+2i-1}, \boldsymbol{z}_1 \dots, \boldsymbol{z}_s).$$

As mentioned before, we will present a way of choosing t in Appendix D.6.

D.4 Bounds on the log-determinant

Since the posterior variance of a Gaussian process can never increase with more data, the average of the (log) posterior variances is an estimator for an upper bound on the log-determinant (Bartels, 2020, Part V). Hence in this case, we simply ignore the interaction between the remaining variables. We set $g(\boldsymbol{x}_n, \boldsymbol{x}_i) := \delta_{ni} \log\left(k_s(\boldsymbol{x}_n, \boldsymbol{x}_n) + \sigma^2(\boldsymbol{x}_n)\right)$ where δ_{ni} denotes Kronecker's δ.

To obtain a lower bound we use that for $c > 0$ and $a \geq b \geq 0$, one can show that $\log(c + a - b) \geq \log(c + a) - \frac{b}{c}$ where the smaller b the better the bound. In our case $c = \sigma^2(\boldsymbol{x}_n)$, $a = k_s(\boldsymbol{x}_n, \boldsymbol{x}_n)$ and $b = k_s(\boldsymbol{x}_n, \boldsymbol{X}_{s+1:n-1}) \left(k_s(\boldsymbol{X}_{s+1:n-1}, \boldsymbol{X}_{s+1:n-1}) + \sigma^2(\boldsymbol{X}_{s+1:n-1})\right)^{-1} k_s(\boldsymbol{X}_{s+1:n-1}, \boldsymbol{x}_n)$. Underestimating the eigenvalues of $k_s(\boldsymbol{X}_{s+1:n-1}, \boldsymbol{X}_{s+1:n-1})$ by 0 we obtain a lower bound, where each quantity can be estimated. Formally, for any $s \leq t$,

$$\log\left(k_{n-1}(\boldsymbol{x}_n, \boldsymbol{x}_n) + \sigma^2(\boldsymbol{x}_n)\right) \geq \left(\log\left(k_s(\boldsymbol{x}_n, \boldsymbol{x}_n) + \sigma^2(\boldsymbol{x}_n)\right) - \sum_{i=s+1}^{n-1} \frac{k_s(\boldsymbol{x}_n, \boldsymbol{x}_i)^2}{\sigma^2(\boldsymbol{x}_n)\sigma^2(\boldsymbol{x}_i)}\right). \tag{10}$$

This bound can be worse than the deterministic lower bound $\min_{n'} \log \sigma^2(\boldsymbol{x}_{n'})$. It depends on how large n is, how large the average correlation is and how small $\sigma^2(\cdot)$ is. We can determine the number of steps $n - s$ that this bound is better by solving a quadratic equation. Denote with μ the estimator for the left addend and with ρ the estimator for the second addend. The tipping point ψ is the solution of $(\psi - s)\left(\mu - \frac{\psi-s+1}{2}\rho\right) \leq (\psi - s) \min \log \sigma^2(\cdot)$. One solution is $\psi = s$, the other is

$$\psi := \lfloor s - 1 + \frac{2}{\rho}\left(\mu - \min \log \sigma^2(\cdot)\right) \rfloor. \tag{11}$$

Hence, for $n > \psi$ we set $u_n^d := \min \log \sigma^2(\cdot)$.

Observe that, the smaller $k_s(\boldsymbol{x}_j, \boldsymbol{x}_{j+1})^2$ the closer the bounds. This term represents the correlation of datapoints conditioned on the s datapoints observed before. Thus, our bounds come together, when incoming observations become independent conditioned on what was already observed. Essentially, that $k_s(\boldsymbol{x}_j, \boldsymbol{x}_{j+1})^2 = 0$ is the basic assumption of inducing input approximations (Quiñonero-Candela & Rasmussen, 2005).

D.5 Bounds on the quadratic form

For an upper bound on the quadratic form we apply a similar trick:

$$\frac{x}{c + a - b} \leq \frac{x(c + b)}{c(c + a)}, \tag{12}$$

where $x \geq 0$. Further we assume that in expectation the mean square error improves with more data. Formally,

$$\frac{(y_j - m_{j-1}(\boldsymbol{x}_j))^2}{k_{j-1}(\boldsymbol{x}_j, \boldsymbol{x}_j) + \sigma^2(\boldsymbol{x}_j)} \leq_E \frac{(y_j - m_s(\boldsymbol{x}_j))^2}{\sigma^2(\boldsymbol{x}_j)\left(k_s(\boldsymbol{x}_j, \boldsymbol{x}_j) + \sigma^2(\boldsymbol{x}_j)\right)} \left(\sigma^2(\boldsymbol{x}_j) + \sum_{i=s+1}^{j-1} \frac{(k_s(\boldsymbol{x}_j, \boldsymbol{x}_i))^2}{\sigma^2(\boldsymbol{x}_i)}\right) \tag{13}$$

For a lower bound observe that

$$\boldsymbol{y}^\top \boldsymbol{K}^{-1} \boldsymbol{y} = \boldsymbol{y}_{:s}^\top \boldsymbol{K}_s^{-1} \boldsymbol{y}_{:s} + \left(\boldsymbol{y}_{s+1:N} - m_s(\boldsymbol{X}_{t+1:N})\right)^\top \boldsymbol{Q}_{s+1:N}^{-1} \left(\boldsymbol{y}_{s+1:N} - m_s(\boldsymbol{X}_{t+1:N})\right) \tag{14}$$

where $\boldsymbol{Q}_{s+1:j} := k_s(\boldsymbol{X}_{s+1:j}, \boldsymbol{X}_{s+1:j}) + \sigma^2(\boldsymbol{X}_{s+1:j})$ with $j \geq s+1$ for the posterior covariance matrix of $\boldsymbol{X}_{s+1:j}$ conditioned on $\boldsymbol{X}_{1:s}$. We use a trick we first encountered in Kim & Teh (2018): $\boldsymbol{y}^\top \boldsymbol{A}^{-1}\boldsymbol{y} \geq 2\boldsymbol{y}^\top \boldsymbol{b} - \boldsymbol{b}^\top \boldsymbol{A}\boldsymbol{b}$, for any $\boldsymbol{b}$. Applying this inequality directly would result in a poor lower bound. For brevity introduce $\boldsymbol{e} := \boldsymbol{y}_{s+1:N} - m_s(\boldsymbol{X}_{t+1:N})$. We rewrite the second term as

$$\boldsymbol{e}^\top \operatorname{diag}(\boldsymbol{e}) \left(\operatorname{diag}(\boldsymbol{e})\boldsymbol{Q}_{s+1:N} \operatorname{diag}(\boldsymbol{e})\right)^{-1} \operatorname{diag}(\boldsymbol{e})\boldsymbol{e} \tag{15}$$

Now applying the inequality with $\boldsymbol{b} := \alpha \mathbf{1}$, we obtain

$$2\alpha \sum_{n=s+1}^{N} (y_n - m_s(\boldsymbol{x}_n))^2 - \alpha^2 \sum_{n,n'=s+1}^{N} (y_n - m_s(\boldsymbol{x}_n)) \, [\boldsymbol{Q}_{s+1:N}]_{nn'} \, (y_{n'} - m_s(\boldsymbol{x}_{n'})) \tag{16}$$

which is now in the form of Equation (7). After taking the expectation, the optimal value of α is

$$\frac{\mathbb{E}[(y_{s+1} - m_s(\boldsymbol{x}_{s+1}))^2 \mid \mathcal{F}_s]}{\alpha_{\text{den}}} \tag{17}$$

where

$$\begin{aligned}\alpha_{\text{den}} &= \mathbb{E}\left[(y_{s+1} - m_s(\boldsymbol{x}_{s+1}))^2 \, k_s(\boldsymbol{x}_{s+1}, \boldsymbol{x}_{s+1}) + \sigma^2(\boldsymbol{x}_{s+1})\right] \\ &\quad + \mathbb{E}\left[(N-s-1)\,(y_{s+1} - m_s(\boldsymbol{x}_{s+1}))\,(y_{s+2} - m_s(\boldsymbol{x}_{s+2}))\,(k_s(s+1, s+2)) \mid \mathcal{F}_s\right],\end{aligned} \tag{18}$$

which is $\mathcal{F}_s$-measureable and we can estimate it.

Observe that, the smaller the square error $(y_j - m_s(\boldsymbol{x}_j))^2$, the closer the bounds. That is, if the model fit is good, the quadratic form can be easily identified.

D.6 Using the Bounds for Stopping the Cholesky

We will use the same stopping strategy as Mnih et al. (2008); Bartels (2020): when the difference between bounds becomes sufficiently small and their absolute value is far away from zero. More precisely, when having deterministic bounds $\mathcal{L} \leq x \leq \mathcal{U}$ on a number x, with

$$\frac{\mathcal{U} - \mathcal{L}}{2\min(|\mathcal{U}|, |\mathcal{L}|)} \leq r \text{ and} \tag{19}$$

$$\operatorname{sign} \mathcal{U} = \operatorname{sign} L, \tag{20}$$

then the relative error of the estimate $\frac{1}{2}(\mathcal{U} + \mathcal{L})$ is less than r, that is $|\frac{\frac{1}{2}(\mathcal{U}+\mathcal{L})-x}{x}| \leq r$.

Remark 1. *In our experiments, we do* ***not*** *use $\frac{1}{2}(\mathcal{U} + \mathcal{L})$ as estimator, and instead use the* ***biased*** *estimator $(N - \tau)\frac{1}{\tau} \log p(\boldsymbol{y}_{:\tau})$. Since stopping occurs when log-determinant and quadratic form evolve roughly linear, the two estimators are not far off each other. The main reason for using the biased estimator is of technical nature: it is easier and faster to implement a custom backward function which can handle the in-place operations of our Cholesky implementation.*

Remark 2. *Another ingredient where we deviate from the theory before is the estimation of the average correlation. In theory, the estimator is allowed to take only every second entry of the off-diagonal. Yet, we could use the same estimator using entries with an offset of* 1 *and still would get a valid estimator. Hence, we the average of the two should also be a decent estimator. We believe that it should be possible to prove that the average of the two estimators is also a decent estimator. Therefore, in practice, take the average over all indices.*

The question remains how to use bounds and stopping strategy to derive an approximation algorithm. We transform the exact Cholesky decomposition for that purpose. For brevity denote $\boldsymbol{L}_s := \operatorname{chol}[k(\boldsymbol{X}_{:s}, \boldsymbol{X}_{:s}) + \sigma^2(\boldsymbol{X}_{:s})]$ and $\boldsymbol{T}_s := k(\boldsymbol{X}_{s+1:}, \boldsymbol{X})\boldsymbol{L}_s{}^{-\top}$. For any $s \in \{1, \ldots N\}$:

$$\boldsymbol{L}_N = \begin{bmatrix} \boldsymbol{L}_s & \mathbf{0} \\ \boldsymbol{T} & \operatorname{chol}\left[k(\boldsymbol{X}_{s+1:,s+1:}) - \boldsymbol{T}\boldsymbol{T}^\top\right] \end{bmatrix} \tag{21}$$

One can verify that $\boldsymbol{L}_N$ is indeed the Cholesky of $\boldsymbol{K}_N$ by evaluating $\boldsymbol{L}_N \boldsymbol{L}_N^\top$. Observe that $k(\boldsymbol{X}_{s+1:,s+1:}) - \boldsymbol{T}\boldsymbol{T}^\top$ is the posterior covariance matrix of the $\boldsymbol{y}_{s+1:}$ conditioned on $\boldsymbol{y}_{:s}$. Hence, in the step before the Cholesky of the posterior covariance matrix is computed, we can estimate our log-determinant bounds.

Similar reasoning applies for solving the linear equation system. We can write

$$\boldsymbol{\alpha}_N = \begin{bmatrix} \boldsymbol{\alpha}_s \\ \operatorname{chol}\left[k(\boldsymbol{X}_{s+1:,s+1:}) - \boldsymbol{T}\boldsymbol{T}^\top\right]^{-1} \left(\boldsymbol{y}_{s+1:} - \boldsymbol{T}_s\boldsymbol{\alpha}_s\right) \end{bmatrix} \tag{22}$$

Now observe that $\boldsymbol{T}_s\boldsymbol{\alpha}_s = m_s(\boldsymbol{X}_{s+1:})$. Hence, before the solving the lower equation system (and before computing the posterior Cholesky), we can compute our bounds for the quadratic form. There are different options to implement the Cholesky decomposition. We use a blocked, row-wise implementation (George et al., 1986). For a practical implementation see Algorithm 1 and Algorithm 2.

Algorithm 1 blocked and recursive formulation of Cholesky decomposition and Gaussian elimination, augmented with our stopping conditions.

---

procedure ACGP($k(\cdot,\cdot)$, $\mu(\cdot)$, $\sigma^2(\cdot)$, $\boldsymbol{X}$, $\boldsymbol{y}$, m, $N_{\max}$)
  $\boldsymbol{A} \leftarrow \boldsymbol{0}^{N_{\max}\times N_{\max}}$, $\boldsymbol{\alpha} \leftarrow \boldsymbol{0}^{N_{\max}}$ // allocate memory
  $\boldsymbol{A}_{1:m,1:m} \leftarrow k(\boldsymbol{X}_{1:m}) + \sigma^2(\boldsymbol{X}_{1:m})$ // initialize kernel matrix
  $\boldsymbol{\alpha}_{1:m} \leftarrow \boldsymbol{y}_{1:m} - \mu(\boldsymbol{X}_{1:m})$ // evaluate mean function for the same datapoints
  $\boldsymbol{A}_{1:m,1:m} \leftarrow \operatorname{chol}(\boldsymbol{A}_{1:m,1:m})$ // call to low-level Cholesky
  $\boldsymbol{\alpha}_{1:m} \leftarrow \boldsymbol{A}_{1:m,1:m}^{-1}\boldsymbol{\alpha}_{1:m}$ // second back-substitution step
  $i \leftarrow m+1$, $j \leftarrow \min(i+m, N)$
  while $i < N_{\max}$ **do**
    $\boldsymbol{A}_{i:j,1:i} \leftarrow k(\boldsymbol{X}_{i:j}, \boldsymbol{X}_{1:i})$ // evaluate required block-off-diagonal part of the kernel matrix
    $\boldsymbol{A}_{i:j,1:i} \leftarrow \boldsymbol{A}_{i:j,1:i}\boldsymbol{A}_{1:i,1:i}^{-\top}$ // solve triangular linear equation system
    $\boldsymbol{A}_{i:j,i:j} \leftarrow k(\boldsymbol{X}_{i:j}) + \sigma^2(\boldsymbol{X}_{i:j})$ // evaluate required block-diagonal part of the kernel matrix
    $\boldsymbol{\alpha}_{i:j} \leftarrow \boldsymbol{y}_{i:j} - \mu(\boldsymbol{X}_{i:j})$ // evaluate mean function for the same datapoints
    $\boldsymbol{A}_{i:j,i:j} \leftarrow \boldsymbol{A}_{i:j,i:j} - \boldsymbol{A}_{i:j,1:i}\boldsymbol{A}_{i:j,1:i}^\top$ // down-date
    // now $\boldsymbol{A}_{i:j,i:j} = \boldsymbol{Q}_{s+1:j}$
    $\boldsymbol{\alpha}_{i:j} \leftarrow \boldsymbol{\alpha}_{i:j} - \boldsymbol{A}_{i:j,1:i}\boldsymbol{\alpha}_{1:i}$ // now $\boldsymbol{\alpha}_{i:j}$ contains $\boldsymbol{y}_{i:j} - m_i(\boldsymbol{X}_{i:j})$
    $\mathcal{L}, \mathcal{U} \leftarrow$ **EvaluateBounds**(i, j) // costs $\mathcal{O}(j-i)$
    if Equations (19) and (20) fulfilled **then**
      return estimator
    end if
    $\boldsymbol{A}_{i:j,i:j} \leftarrow \operatorname{chol}(\boldsymbol{A}_{i:j,i:j})$ // finish computing Cholesky for data-points up to index j
    $\boldsymbol{\alpha}_{i:j} \leftarrow \boldsymbol{A}_{i:j,i:j}^{-1}\boldsymbol{\alpha}_{i:j}$ // finish solving linear equation system for index up to j
    $i \leftarrow i+m$, $j \leftarrow \min(i+m, N_{\max})$
  end while // now $\boldsymbol{A} = \boldsymbol{L}$ and $\boldsymbol{\alpha} = \boldsymbol{L}^{-1}(\boldsymbol{y} - \mu(\boldsymbol{X}))$
  return estimator
end procedure

E Assumptions

Assumption 3. *Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space and let $(\boldsymbol{x}_j, y_j)_{j=1}^N$ be a sequence of independent and identically distributed random vectors with $\boldsymbol{x} : \Omega \to \mathbb{R}^D$ and $y : \Omega \to \mathbb{R}$.*

Assumption 4. *For all s, i, j, t with $s < i \leq j \leq N$ and functions $f(\boldsymbol{x}_j, \boldsymbol{x}_i; \boldsymbol{x}_1, \ldots \boldsymbol{x}_s) \geq 0$*

$$\mathbb{E}\left[f(\boldsymbol{x}_j,\boldsymbol{x}_i)\left(y_j - m_{j-1}(\boldsymbol{x}_j)\right)^2 \mid \mathcal{F}_s\right] \leq \mathbb{E}\left[f(\boldsymbol{x}_j,\boldsymbol{x}_i)\left(y_j - m_s(\boldsymbol{x}_j)\right)^2 \mid \mathcal{F}_s\right] \tag{23}$$

where $f(\boldsymbol{x}_j, \boldsymbol{x}_i) \in \left\{\frac{1}{k_s(\boldsymbol{x}_j,\boldsymbol{x}_j)+\sigma^2(\boldsymbol{x}_j)}, \frac{k_s(\boldsymbol{x}_j,\boldsymbol{x}_i)^2}{(k_s(\boldsymbol{x}_j,\boldsymbol{x}_j)+\sigma^2(\boldsymbol{x}_j))\sigma^2(\boldsymbol{x}_j)\sigma^2(\boldsymbol{x}_i)}\right\}$.

That is, we assume that in expectation the estimator improves with more data. Note that, f can not depend on any entries of $\boldsymbol{y}$.

Algorithm 2 bound algorithm as used in our experiments. The algorithm deviates slightly from our theory. We use Equation (39) for the upper bound in the quadratic form, and we use all off-diagonal entries (instead of only every second).

procedure EVALUATEBOUNDS(s, t)
  $D \leftarrow \sum_{j=1}^{s} 2\log \boldsymbol{A}_{jj}$ // in practice we reuse the sum from the last iteration
  $Q \leftarrow \sum_{j=1}^{s} \boldsymbol{\alpha}_j^2$
  $\mu \leftarrow \frac{1}{t-s}\sum_{j=s+1}^{t} 2\log \boldsymbol{A}_{jj}$ // average variance of the new points conditioned on all points processed until s
  $\mathcal{U}_D \leftarrow D + (N-s)\mu$
  $\rho \leftarrow \frac{1}{t-s-1}\sum_{j=s+1}^{t-1} \boldsymbol{A}_{j,j+1}^2$ // average square correlation (deviating from theory!)
  $\psi \leftarrow \lfloor s - 1 + \frac{2}{\rho}\left(\mu - \log\sigma^2\right)\rfloor$ // number of steps the probabilistic bound is better than the deterministic
  $\mathcal{L}_D \leftarrow D + (\psi - s)\left(\mu - \frac{\psi-s-1}{2}\rho\right) + (N-\psi)\log\sigma^2$
  $\mathcal{U}_Q \leftarrow Q + \frac{N-s}{t-s-1}\sum_{j=s+1}^{t-1} \frac{\boldsymbol{\alpha}_j^2}{\boldsymbol{A}_{j,j}\sigma^2(\boldsymbol{x}_j)}\left(\sigma^2(\boldsymbol{x}_j) + \frac{\boldsymbol{A}_{j,j+1}^2}{\sigma^2(\boldsymbol{x}_{j+1})}\right)$
  $m \leftarrow \frac{1}{t-s}\sum_{j=s+1}^{t} \boldsymbol{\alpha}_j^2$ // mean square error
  $\mu \leftarrow \frac{1}{t-s}\sum_{j=s+1}^{t} \boldsymbol{\alpha}_j^2 \boldsymbol{A}_{j,j}$
  $\rho \leftarrow \frac{1}{t-s-1}\sum_{j=s+1}^{t-1} \boldsymbol{\alpha}_j \boldsymbol{\alpha}_{j+1} \boldsymbol{A}_{j,j+1}$
  $\alpha \leftarrow \frac{m}{\mu + (N-s-1)\rho}$
  $\mathcal{L}_Q \leftarrow Q + \alpha(N-s)(2m - \alpha(\mu + \rho(N-s-1)))$
  return $\mathcal{L}_D + \mathcal{L}_Q, \mathcal{U}_D + \mathcal{U}_Q$
end procedure

F Main Theorem

This section restates Theorem 2 and connects the different proofs in the sections to follow.

Theorem 5. *Assume that Assumption 3 and Assumption 4 hold. For any even $m \in \{2, 4, \ldots, N-2\}$ and any $s \in \{1, \ldots, N-m\}$, the bounds defined in Equations (6), (7), (9) and (10) hold in expectation:*

$$\mathbb{E}[\mathcal{L}_D \mid \mathcal{F}_s] \leq \mathbb{E}[\log(\det[\boldsymbol{K}]) \mid \mathcal{F}_s] \leq \mathbb{E}[\mathcal{U}_D \mid \mathcal{F}_s] \text{ and}$$
$$\mathbb{E}[\mathcal{L}_Q \mid \mathcal{F}_s] \leq \mathbb{E}[\boldsymbol{y}^\top \boldsymbol{K}^{-1}\boldsymbol{y} \mid \mathcal{F}_s] \leq \mathbb{E}[\mathcal{U}_Q \mid \mathcal{F}_s]\,.$$

Proof. Follows from Theorems 6, 9 and 13, and Theorem 28 in Bartels (2020).

□

G Proof for the Lower Bound on the Determinant

Theorem 6. *Assume that Assumption 3 holds, and that $m \in \{2, 4, \ldots, N\}$ is an even number. Set $t := s + m$, then, for all $s \in \{1, \ldots, N-m\}$*

$$\mathbb{E}\left[\mathcal{L}_D \mid \mathcal{F}_s\right] \leq \mathbb{E}\left[D_N \mid \mathcal{F}_s\right].$$

$$\mathcal{L}_D := \log\det \boldsymbol{K}_{:s,:s} + (\psi_t - s)\left(\log \underline{\mu}_t - \frac{\psi_t - s - 1}{2}\tilde{\rho}_t\right) + (N - \psi_t)\log\sigma^2 \tag{24}$$

// the lower bound

$$\log \underline{\mu}_t := \frac{1}{m}\sum_{j=s+1}^{t} \log\left(\sigma^2(\boldsymbol{x}_j) + k_s(\boldsymbol{x}_j, \boldsymbol{x}_j)\right) \tag{25}$$

// (under-)estimate of the posterior variance conditioned on s points

$$\tilde{\rho}_t := \frac{2}{m}\sum_{j=\frac{s+1}{2}}^{\frac{t-1}{2}} \frac{k_s(\boldsymbol{x}_{2j+1}, \boldsymbol{x}_{2j})^2}{\sigma^2(\boldsymbol{x}_{2j+1})\sigma^2(\boldsymbol{x}_{2j})} \tag{26}$$

$$
\begin{aligned}
&\text{// (over-)estimate of the correlation conditioned on } s \text{ points} \\
\psi_t &:= s + \max p \textit{ where } p \textit{ is such that} \qquad (27) \\
&p \left(\log \underline{\mu}_t - \frac{p-1}{2} \tilde{\rho}_t \right) \geq p \log \sigma^2 \qquad (28) \\
&\text{// number of steps that we suspect the decrease in variance to be controllable} \\
&= \max(N, \lfloor s - 1 + \frac{2}{\rho_D} \left(\mu_D - \log \sigma^2 \right) \rfloor) \qquad (29)
\end{aligned}
$$

Proof.

$$
\begin{aligned}
&\mathbb{E}\left[\mathcal{L}_D \mid \mathcal{F}_s\right] - \mathbb{E}\left[\log\det \boldsymbol{K} \mid \mathcal{F}_s\right] = \mathbb{E}\left[\mathcal{L}_D - \log\det \boldsymbol{K} \mid \mathcal{F}_s\right] \\
&= \mathbb{E}\left[(\psi_t - s) \left(\log \underline{\mu}_t - \frac{\psi_t - s - 1}{2} \tilde{\rho}_t \right) + (N - \psi_t) \log \sigma^2 - \sum_{j=s+1}^{N} \log \left(k_s(\boldsymbol{x}_j, \boldsymbol{x}_j) + \sigma^2(\boldsymbol{x}_j) \right) \mid \mathcal{F}_s \right] \\
&\textit{// using the definition of } \mathcal{L}_t \textit{ and slightly simplifying using Lemma 17} \\
&\leq \mathbb{E}\left[(\psi_t - s) \left(\log \underline{\mu}_t - \frac{\psi_t - s - 1}{2} \tilde{\rho}_t \right) - \sum_{j=s+1}^{\psi_t} \log \left(k_s(\boldsymbol{x}_j, \boldsymbol{x}_j) + \sigma^2(\boldsymbol{x}_j) \right) \mid \mathcal{F}_s \right] \\
&\textit{// using that } \log \sigma^2 \leq \log \left(k_s(\boldsymbol{x}_j, \boldsymbol{x}_j) + \sigma^2(\boldsymbol{x}_j) \right) \textit{ for all } j \\
&= (\psi_t - s) \left(\mathbb{E}\left[\log \left(\sigma^2 + k_s(\boldsymbol{x}_{t+1}, \boldsymbol{x}_{t+1}) \mid \mathcal{F}_s \right) \right] - \frac{\psi_t - s - 1}{2} \mathbb{E}\left[\frac{k_s(\boldsymbol{x}_{t+1}, \boldsymbol{x}_{t+2})^2}{\sigma^2(\boldsymbol{x}_{t+1}) \sigma^2(\boldsymbol{x}_{t+2})} \mid \mathcal{F}_s \right] \right) \\
&\quad - \mathbb{E}\left[\sum_{j=s+1}^{\psi_t} \log \left(k_s(\boldsymbol{x}_j, \boldsymbol{x}_j) + \sigma^2(\boldsymbol{x}_j) \right) \mid \mathcal{F}_s \right] \\
&\textit{// using Assumption 3} \\
&\leq 0 \\
&\textit{// Lemma 8}
\end{aligned}
$$

□

Lemma 7. *For $c > 0$ and $b \geq a \geq 0$:*

$$
\log(c + b - a) \geq \log(c + b) - \frac{a}{c}
$$

Proof. For $a = 0$, the statement is true with equality. We rewrite the inequality as

$$
\frac{a}{c} \geq \log \left(\frac{c + b}{c + b - a} \right) = \log \left(1 + \frac{a}{c + b - a} \right).
$$

For the case $a = b$, apply the exponential function on both sides, and the statement follows from $e^x \geq x + 1$ for all x. For $a \in (0, b)$, consider $f(a) := \log(c + b - a) + \frac{a}{c} - \log(c + b)$. The first derivative of this function is $f'(a) = -\frac{1}{c+b-a} + \frac{1}{c}$, which is always positive for $a \in (0, b)$. Since $f(0) = 0$, we must have $f(a) \geq 0$ for all $a \in (a, b)$. □

Lemma 8. *For all $n \geq t \geq s$:*

$$
\begin{aligned}
\mathbb{E}\left[\sum_{j=t+1}^{n} \log \left(k_s(\boldsymbol{x}_j, \boldsymbol{x}_j) + \sigma^2(\boldsymbol{x}_j) \right) \mid \mathcal{F}_s \right] \geq (n - t) \Bigg(& \mathbb{E}\left[\log \left(\sigma^2(\boldsymbol{x}_{t+1}) + k_s(\boldsymbol{x}_{t+1}, \boldsymbol{x}_{t+1}) \right) \mid \mathcal{F}_s \right] \\
& - \frac{n - t - 1}{2\sigma^4} \mathbb{E}\left[k_s(\boldsymbol{x}_{t+1}, \boldsymbol{x}_{t+2})^2 \mid \mathcal{F}_s \right] \Bigg)
\end{aligned}
$$

Proof. Introduce $\overline{\boldsymbol{X}}_j := [\boldsymbol{x}_{s+1}, \ldots, \boldsymbol{x}_{j-1}]$ with the convention $k_s(\boldsymbol{x}_{s+1}, \overline{\boldsymbol{X}}_{s+1}) := 0$.

$$\mathbb{E}\left[\sum_{j=t+1}^{n} \log k_{j-1}(\boldsymbol{x}_j, \boldsymbol{x}_j) + \sigma^2(\boldsymbol{x}_j) \mid \mathcal{F}_s\right] \tag{30}$$

$$= \mathbb{E}\left[\sum_{j=t+1}^{n} \log\left(\sigma^2(\boldsymbol{x}_j) + k_s(\boldsymbol{x}_j, \boldsymbol{x}_j) - k_s(\boldsymbol{x}_j, \overline{\boldsymbol{X}}_j)\left(k_s(\overline{\boldsymbol{X}}_j, \overline{\boldsymbol{X}}_j) + \sigma^2(\overline{\boldsymbol{X}})\right)^{-1} k_s(\overline{\boldsymbol{X}}_j, \boldsymbol{x}_j)\right) \mid \mathcal{F}_s\right] \tag{31}$$

// *Lemma 19*

$$\geq \mathbb{E}\left[\sum_{j=t+1}^{n}\left(\log\left(\sigma^2(\boldsymbol{x}_j) + k_s(\boldsymbol{x}_j, \boldsymbol{x}_j)\right) - \frac{1}{\sigma^2(\boldsymbol{x}_j)} k_s(\boldsymbol{x}_j, \overline{\boldsymbol{X}}_j)\left(k_s(\overline{\boldsymbol{X}}_j, \overline{\boldsymbol{X}}_j) + \sigma^2(\overline{\boldsymbol{X}}_j)\right)^{-1} k_s(\overline{\boldsymbol{X}}_j, \boldsymbol{x}_j)\right) \mid \mathcal{F}_s\right] \tag{32}$$

// *Lemma 7*

$$\geq \mathbb{E}\left[\sum_{j=t+1}^{n}\left(\log\left(\sigma^2(\boldsymbol{x}_j) + k_s(\boldsymbol{x}_j, \boldsymbol{x}_j)\right) - \frac{1}{\sigma^2(\boldsymbol{x}_j)} k_s(\boldsymbol{x}_j, \overline{\boldsymbol{X}}_j)\left(\sigma^2(\overline{\boldsymbol{X}}_j)\right)^{-1} k_s(\overline{\boldsymbol{X}}_j, \boldsymbol{x}_j)\right) \mid \mathcal{F}_s\right] \tag{33}$$

// *underestimating $k_s(\overline{\boldsymbol{X}}_j, \overline{\boldsymbol{X}}_j)$ by* $\mathbf{0}$

$$\geq \mathbb{E}\left[\sum_{j=t+1}^{n}\left(\log\left(\sigma^2(\boldsymbol{x}_j) + k_s(\boldsymbol{x}_j, \boldsymbol{x}_j)\right) - \frac{1}{\sigma^2(\boldsymbol{x}_j)} \sum_{i=t+1}^{j-1} \frac{k_s(\boldsymbol{x}_j, \boldsymbol{x}_j)^2}{\sigma^2(\boldsymbol{x}_i)}\right) \mid \mathcal{F}_s\right] \tag{34}$$

// *writing the vector multiplication as sum*

$$\begin{aligned} &= (n-t)\mathbb{E}\left[\log\left(\sigma^2 + k_s(\boldsymbol{x}_{t+1}, \boldsymbol{x}_{t+1})\right) \mid \mathcal{F}_s\right] \\ &\quad + \frac{(n-t)(n-t-1)}{2}\mathbb{E}\left[\frac{k_s(\boldsymbol{x}_{t+1}, \boldsymbol{x}_{t+2})^2}{\sigma^2(\boldsymbol{x}_{t+1})\sigma^2(\boldsymbol{x}_{t+2})} \mid \mathcal{F}_s\right] \end{aligned} \tag{35}$$

// *using Assumption 3 and then applying Lemma 20*

□

H Proof for the Upper Bound on the Quadratic Form

Theorem 9. *Assume that Assumption 3 and Assumption 4 hold. Let $m \in \mathbb{N}$ be even, then for all $s \in \{1, \ldots, N-m\}$*

$$\mathbb{E}[\boldsymbol{y}^\top \boldsymbol{K}^{-1}\boldsymbol{y} \mid \mathcal{F}_s] \leq \mathbb{E}[\mathcal{U}_Q \mid \mathcal{F}_s]\,,$$

where

$$\mathcal{U}_Q := \boldsymbol{y}_{:s}^\top \boldsymbol{K}_{:s,:s}^{-1} \boldsymbol{y}_{:s} + (N-s)\left(\mu_t + \frac{N-s-1}{2}\tilde{\rho}_t\right) \tag{36}$$

// the upper bound

$$\mu_t := \frac{1}{t-s}\sum_{j=s+1}^{t} \frac{(y_j - m_s(\boldsymbol{x}_j))^2}{k_s(\boldsymbol{x}_j, \boldsymbol{x}_j) + \sigma^2(\boldsymbol{x}_j)} \tag{37}$$

$$\tilde{\rho}_t := \frac{2}{t-s}\sum_{j=\frac{s+2}{2}}^{\frac{t}{2}} \frac{(y_{2j} - m_s(\boldsymbol{x}_{2j}))^2 k_s(\boldsymbol{x}_{2j}, \boldsymbol{x}_{2j-1})^2}{\left(k_s(\boldsymbol{x}_{2j}, \boldsymbol{x}_{2j}) + \sigma^2(\boldsymbol{x}_{2j})\right)\sigma^2(\boldsymbol{x}_{2j})\sigma^2(\boldsymbol{x}_{2j-1})}\,. \tag{38}$$

Proof.

$$\begin{aligned}
&\mathbb{E}\left[\boldsymbol{y}^\top \boldsymbol{K}^{-1}\boldsymbol{y} \mid \mathcal{F}_s\right] - \mathbb{E}\left[\mathcal{U}_Q \mid \mathcal{F}_s\right] \\
&= \mathbb{E}\left[\sum_{j=s+1}^{N} \frac{(y_j - m_{j-1}(\boldsymbol{x}_j))^2}{k_{j-1}(\boldsymbol{x}_j, \boldsymbol{x}_j) + \sigma^2(\boldsymbol{x}_j)} - (N-s)\left(\mu_t + \frac{N-s-1}{2}\tilde{\rho}_t\right) \mid \mathcal{F}_s\right] \\
&\text{// using the definition of } \mathcal{U}_Q \text{ and slightly simplifying with Lemma 18} \\
&= \mathbb{E}\left[\sum_{j=s+1}^{\psi_t} \frac{(y_j - m_{j-1}(\boldsymbol{x}_j))^2}{k_{j-1}(\boldsymbol{x}_j, \boldsymbol{x}_j) + \sigma^2(\boldsymbol{x}_j)} \mid \mathcal{F}_s\right] \\
&- (N-s)\left(\mathbb{E}\left[\frac{(y_{s+1} - m_s(\boldsymbol{x}_{s+1}))^2}{k_s(\boldsymbol{x}_{s+1}, \boldsymbol{x}_{s+1}) + \sigma^2(\boldsymbol{x}_{s+1})} \mid \mathcal{F}_s\right]\right) \\
&- (N-s)\frac{N-s-1}{2}\mathbb{E}\left[\frac{(y_{s+1} - m_s(\boldsymbol{x}_{s+1}))^2 k_s(\boldsymbol{x}_{s+1}, \boldsymbol{x}_{s+2})^2}{(k_s(\boldsymbol{x}_{s+1}, \boldsymbol{x}_{s+1}) + \sigma^2(\boldsymbol{x}_{s+1}))\,\sigma^2(\boldsymbol{x}_{s+2})} \mid \mathcal{F}_s\right] \\
&\text{// using Assumption 3} \\
&\leq 0 \\
&\text{// Lemma 11}
\end{aligned}$$

$\square$

Lemma 10. *For $c > 0$, $b, x \geq 0$ and $a \geq b$:*

$$\frac{x}{c+a-b} \leq \frac{x}{c}\left(1 - \frac{a-b}{c+a}\right) = \frac{x(c+b)}{c(c+a)}$$

Proof.

$$\begin{aligned}
\frac{x}{c+a-b} &= \frac{x}{c}\left(\frac{c}{c+a-b}\right) \\
&= \frac{x}{c}\left(1 - \frac{a-b}{c+a-b}\right) \\
&\leq \frac{x}{c}\left(1 - \frac{c+a-b}{c+a}\frac{a-b}{c+a-b}\right) \\
&\text{// since } \frac{c+a-b}{c+a} \leq 1 \\
&= \frac{x}{c}\left(1 - \frac{a-b}{c+a}\right) \\
&\text{// cancelling terms}
\end{aligned}$$

$\square$

Lemma 11. *For all s, t and n with $n \geq t \geq s$:*

$$\begin{aligned}
&\mathbb{E}\left[\sum_{j=t+1}^{n} \frac{(y_j - m_{j-1}(\boldsymbol{x}_j))^2}{k_{j-1}(\boldsymbol{x}_j, \boldsymbol{x}_j) + \sigma^2(\boldsymbol{x}_j)} \mid \mathcal{F}_s\right] \\
&\leq (n-t)\left(\mathbb{E}\left[\frac{(y_{s+1} - m_s(\boldsymbol{x}_{s+1}))^2}{k_s(\boldsymbol{x}_{s+1}, \boldsymbol{x}_{s+1}) + \sigma^2(\boldsymbol{x}_{s+1})} \mid \mathcal{F}_s\right]\right) \\
&+ (n-t)\left(\left(\frac{n+t+1}{2} - s\right)\mathbb{E}\left[\frac{(y_{s+1} - m_s(\boldsymbol{x}_{s+1}))^2 k_s(\boldsymbol{x}_{s+1}, \boldsymbol{x}_{s+2})^2}{(k_s(\boldsymbol{x}_{s+1}, \boldsymbol{x}_{s+1}) + \sigma^2(\boldsymbol{x}_{s+1}))\,\sigma^2(\boldsymbol{x}_{s+2})} \mid \mathcal{F}_s\right]\right)
\end{aligned}$$

Proof. Introduce $\overline{\boldsymbol{X}}_j := [\boldsymbol{x}_{s+1}, \ldots, \boldsymbol{x}_{j-1}]$ with the convention $k_s(\boldsymbol{x}_{s+1}, \overline{\boldsymbol{X}}_{s+1}) := 0$.

$$
\begin{aligned}
&\mathbb{E}\left[\sum_{j=t+1}^{n} \frac{(y_j - m_{j-1}(\boldsymbol{x}_j))^2}{k_{j-1}(\boldsymbol{x}_j, \boldsymbol{x}_j) + \sigma^2(\boldsymbol{x}_j)} \mid \mathcal{F}_s\right] \\
&= \mathbb{E}\left[\sum_{j=t+1}^{n} \frac{(y_j - m_{j-1}(\boldsymbol{x}_j))^2}{\sigma^2(\boldsymbol{x}_j) + k_s(\boldsymbol{x}_j, \boldsymbol{x}_j) - k_s(\boldsymbol{x}_j, \overline{\boldsymbol{X}}_j)\left(k_s(\overline{\boldsymbol{X}}_j, \overline{\boldsymbol{X}}_j) + \sigma^2(\overline{\boldsymbol{X}})\right)^{-1} k_s(\overline{\boldsymbol{X}}_j, \boldsymbol{x}_j)} \mid \mathcal{F}_s\right] \\
&\text{// \textit{Lemma 19}} \\
&\leq \mathbb{E}\left[\sum_{j=t+1}^{n} \frac{(y_j - m_{j-1}(\boldsymbol{x}_j))^2 \left(\sigma^2(\boldsymbol{x}_j) + k_s(\boldsymbol{x}_j, \overline{\boldsymbol{X}}_j)\left(k_s(\overline{\boldsymbol{X}}_j, \overline{\boldsymbol{X}}_j) + \sigma^2(\overline{\boldsymbol{X}})\right)^{-1} k_s(\overline{\boldsymbol{X}}_j, \boldsymbol{x}_j)\right)}{\sigma^2(\boldsymbol{x}_j)\left(\sigma^2(\boldsymbol{x}_j) + k_s(\boldsymbol{x}_j, \boldsymbol{x}_j)\right)} \mid \mathcal{F}_s\right] \\
&\text{// \textit{Lemma 10}} \\
&\leq \mathbb{E}\left[\sum_{j=t+1}^{n} \frac{(y_j - m_{j-1}(\boldsymbol{x}_j))^2 \left(\sigma^2(\boldsymbol{x}_j) + k_s(\boldsymbol{x}_j, \overline{\boldsymbol{X}}_j)\left(\sigma^2(\overline{\boldsymbol{X}})\right)^{-1} k_s(\overline{\boldsymbol{X}}_j, \boldsymbol{x}_j)\right)}{\sigma^2(\boldsymbol{x}_j)\left(\sigma^2(\boldsymbol{x}_j) + k_s(\boldsymbol{x}_j, \boldsymbol{x}_j)\right)} \mid \mathcal{F}_s\right] \\
&\text{// \textit{underestimating the eigenvalues of} } k_s(\overline{\boldsymbol{X}}, \overline{\boldsymbol{X}}) \text{ \textit{by} } 0 \\
&= \mathbb{E}\left[\sum_{j=t+1}^{n} \frac{(y_j - m_{j-1}(\boldsymbol{x}_j))^2 \left(\sigma^2(\boldsymbol{x}_j) + \sum_{i=s+1}^{j-1} \frac{k_s(\boldsymbol{x}_j, \boldsymbol{x}_i)^2}{\sigma^2(\boldsymbol{x}_i)}\right)}{\sigma^2(\boldsymbol{x}_j)\left(\sigma^2(\boldsymbol{x}_j) + k_s(\boldsymbol{x}_j, \boldsymbol{x}_j)\right)} \mid \mathcal{F}_s\right] \\
&\text{// \textit{writing the vector-product explicitly as a sum}} \\
&= \sum_{j=t+1}^{n}\left(\mathbb{E}\left[\frac{(y_j - m_{j-1}(\boldsymbol{x}_j))^2}{\sigma^2(\boldsymbol{x}_j) + k_s(\boldsymbol{x}_j, \boldsymbol{x}_j)} \mid \mathcal{F}_s\right] + \sum_{i=s+1}^{j-1} \mathbb{E}\left[\frac{(y_j - m_{j-1}(\boldsymbol{x}_j))^2 k_s(\boldsymbol{x}_j, \boldsymbol{x}_i)^2}{\left(k_s(\boldsymbol{x}_j, \boldsymbol{x}_j) + \sigma^2(\boldsymbol{x}_j)\right)\sigma^2(\boldsymbol{x}_i)\sigma^2(\boldsymbol{x}_j)} \mid \mathcal{F}_s\right]\right) \\
&\text{// \textit{linearity of expectation}} \\
&= \sum_{j=t+1}^{n}\left(\mathbb{E}\left[\frac{(y_j - m_s(\boldsymbol{x}_j))^2}{\sigma^2(\boldsymbol{x}_j) + k_s(\boldsymbol{x}_j, \boldsymbol{x}_j)} \mid \mathcal{F}_s\right] + \sum_{i=s+1}^{j-1} \mathbb{E}\left[\frac{(y_j - m_s(\boldsymbol{x}_j))^2 k_s(\boldsymbol{x}_j, \boldsymbol{x}_i)^2}{\left(k_s(\boldsymbol{x}_j, \boldsymbol{x}_j) + \sigma^2(\boldsymbol{x}_j)\right)\sigma^2(\boldsymbol{x}_i)\sigma^2(\boldsymbol{x}_j)} \mid \mathcal{F}_s\right]\right) \\
&\text{// \textit{by assumption Equation} (23)} \\
&= \sum_{j=t+1}^{n}\left(\mathbb{E}\left[\frac{(y_{s+1} - m_s(\boldsymbol{x}_{s+1}))^2}{\sigma^2(\boldsymbol{x}_{s+1}) + k_s(\boldsymbol{x}_{s+1}, \boldsymbol{x}_{s+1})} \mid \mathcal{F}_s\right] + \sum_{i=s+1}^{j-1} \mathbb{E}\left[\frac{(y_{s+1} - m_s(\boldsymbol{x}_{s+1}))^2 k_s(\boldsymbol{x}_{s+1}, \boldsymbol{x}_{s+2})^2}{\left(k_s(\boldsymbol{x}_{s+1}, \boldsymbol{x}_{s+1}) + \sigma^2(\boldsymbol{x}_{s+1})\right)\sigma^2(\boldsymbol{x}_{s+2})\sigma^2(\boldsymbol{x}_{s+1})} \mid \mathcal{F}_s\right]\right) \\
&\text{// \textit{using Assumption 3}} \\
&= (n-t)\left(\mathbb{E}\left[\frac{(y_{s+1} - m_s(\boldsymbol{x}_{s+1}))^2}{\sigma^2(\boldsymbol{x}_{s+1}) + k_s(\boldsymbol{x}_{s+1}, \boldsymbol{x}_{s+1})} \mid \mathcal{F}_s\right]\right) \\
&+ (n-t)\left(\left(\frac{n+t+1}{2} - s\right)\mathbb{E}\left[\frac{(y_{s+1} - m_s(\boldsymbol{x}_{s+1}))^2 k_s(\boldsymbol{x}_{s+1}, \boldsymbol{x}_{s+2})^2}{\left(k_s(\boldsymbol{x}_{s+1}, \boldsymbol{x}_{s+1}) + \sigma^2(\boldsymbol{x}_{s+1})\right)\sigma^2(\boldsymbol{x}_{s+2})\sigma^2(\boldsymbol{x}_{s+1})} \mid \mathcal{F}_s\right]\right) \\
&\text{// \textit{by Lemma 20}}
\end{aligned}
$$

□

Remark 12. *Similar to the proof of Theorem 6, we can improve the bound by monitoring how many steps the sum of average correlations is below the average variance. More precisely, we solve for the largest $\psi \leq N$ such that*

$$
\mu_t + \frac{N - \psi - 1}{2}\tilde{\rho}_t \leq \frac{1}{t-s}\sum_{j=s+1}^{t} \frac{(y_j - m_s(\boldsymbol{x}_{\boldsymbol{x}_j}))^2}{\sigma^2(\boldsymbol{x}_j)},
$$

and replace the upper bound by

$$
\mathcal{U}_t := \boldsymbol{y}_{:s}^{\top}\boldsymbol{K}_{:s,:s}^{-1}\boldsymbol{y}_{:s} + (\psi - s)\left(\mu_t + \frac{\psi - s - 1}{2}\tilde{\rho}_t\right) + \frac{N - \psi}{t - s}\sum_{j=s+1}^{t} \frac{(y_j - m_s(\boldsymbol{x}_{\boldsymbol{x}_j}))^2}{\sigma^2(\boldsymbol{x}_j)}. \quad (39)
$$

I Proof for the Lower Bound on the Quadratic Form

Theorem 13. *Assume that Assumption 3 and Assumption 4 hold. Let $m \in \{2, \dots, N-2\}$ be an even number less than N. For $s \in \{1, \dots, N-m\}$, let α be $\mathcal{F}_s$-measurable, then*

$$\mathbb{E}[\mathcal{L}_Q \mid \mathcal{F}_s] \leq \mathbb{E}[\boldsymbol{y}^\top \boldsymbol{K}^{-1} \boldsymbol{y} \mid \mathcal{F}_s]$$

where

$$\mathcal{L}_Q := \boldsymbol{y}_{:s}^\top \boldsymbol{K}_{:s,:s}^{-1} \boldsymbol{y}_{:s} + \alpha(N-s)\left(2\mu_t - \alpha\mu_t' - \alpha\frac{N-s}{2}\rho_t\right) \tag{40}$$

$$\mu_t := \frac{1}{t-s}\sum_{j=s+1}^{t}(y_j - m_s(\boldsymbol{x}_j))^2 \tag{41}$$

$$\mu_t' := \frac{1}{t-s}\sum_{j=s+1}^{t}(y_j - m_s(\boldsymbol{x}_j))^2(k_s(\boldsymbol{x}_j, \boldsymbol{x}_j) + \sigma^2(\boldsymbol{x}_j)) \tag{42}$$

$$\rho_t := \frac{2}{t-s}\sum_{j=\frac{s+2}{2}}^{\frac{t}{2}}(y_{2j} - m_s(\boldsymbol{x}_{2j}))(y_{2j-1} - m_s(\boldsymbol{x}_{2j-1}))k_s(\boldsymbol{x}_{2j}, \boldsymbol{x}_{2j-1}) \tag{43}$$

Remark 14. *In our implementation we choose $\alpha := \frac{\mu_t}{\mu_t' + (N-s)\rho_t}$ which maximizes the lower bound, though violates the assumption that α is $\mathcal{F}_s$-measurable.*

Proof.

$$\begin{aligned}
&\mathbb{E}\left[\mathcal{L}_Q \mid \mathcal{F}_s\right] - \mathbb{E}\left[\boldsymbol{y}^\top \boldsymbol{K}^{-1}\boldsymbol{y} \mid \mathcal{F}_s\right] \\
&= \mathbb{E}\left[\alpha(N-s)\left(2\mu_t - \alpha\mu_t' - \alpha\frac{N-s}{2}\rho_t\right) - \sum_{j=s+1}^{N}\frac{(y_j - m_{j-1}(\boldsymbol{x}_j))^2}{k_{j-1}(\boldsymbol{x}_j, \boldsymbol{x}_j) + \sigma^2(\boldsymbol{x}_j)} \mid \mathcal{F}_s\right] \\
&\text{// using the definition of } \mathcal{L}_Q \text{ and slightly simplifying} \\
&= 2(N-s)\alpha\mathbb{E}\left[(y_{s+1} - m_s(\boldsymbol{x}_{s+1}))^2 \mid \mathcal{F}_s\right] \\
&\quad - (N-s)\alpha^2\mathbb{E}\left[(y_{s+1} - m_s(\boldsymbol{x}_{s+1}))^2(k_s(\boldsymbol{x}_{s+1}, \boldsymbol{x}_{s+1}) + \sigma^2(\boldsymbol{x}_{s+1})) \mid \mathcal{F}_s\right] \\
&\quad - \frac{(N-s)^2}{2}\alpha^2\mathbb{E}\left[(y_{s+1} - m_s(\boldsymbol{x}_{s+1}))(y_{s+2} - m_s(\boldsymbol{x}_{s+2}))k_s(\boldsymbol{x}_{s+1}, \boldsymbol{x}_{s+2}) \mid \mathcal{F}_s\right] \\
&\quad - \sum_{j=s+1}^{N}\mathbb{E}\left[\frac{(y_j - m_{j-1}(\boldsymbol{x}_j))^2}{k_{j-1}(\boldsymbol{x}_j, \boldsymbol{x}_j) + \sigma^2(\boldsymbol{x}_j)} \mid \mathcal{F}_s\right] \\
&\text{// using Assumption 3} \\
&\leq 0 \\
&\text{// using Lemma 15}
\end{aligned}$$

□

Lemma 15. *For all $\mathcal{F}_s$-measurable $\alpha \in \mathbb{R}$:*

$$\begin{aligned}
\mathbb{E}\left[\boldsymbol{y}^\top \boldsymbol{K}^{-1}\boldsymbol{y} \mid \mathcal{F}_s\right] \geq\ & \boldsymbol{y}_{:s}^\top \boldsymbol{K}_{:s,:s}^{-1}\boldsymbol{y}_{:s} + 2\alpha(N-s)\mathbb{E}[(y_{s+1} - m_s(\boldsymbol{x}_{s+1}))^2 \mid \mathcal{F}_s] \\
& - \alpha^2(N-s)\mathbb{E}[(y_{s+1} - m_s(\boldsymbol{x}_{s+1}))^2(k_s(\boldsymbol{x}_{s+1}, \boldsymbol{x}_{s+1}) + \sigma^2(\boldsymbol{x}_{s+1})) \mid \mathcal{F}_s] \\
& - \alpha^2\frac{(N-s)^2}{2}\mathbb{E}[(y_{s+1} - m_s(\boldsymbol{x}_{s+1}))(y_{s+2} - m_s(\boldsymbol{x}_{s+2}))k_s(\boldsymbol{x}_{s+1}, \boldsymbol{x}_{s+2}) \mid \mathcal{F}_s]
\end{aligned} \tag{44}$$

Proof. Using Lemma 18, we can write the quadratic form as a sum of two quadratic forms:

$$\boldsymbol{y}^\top \boldsymbol{K}^{-1}\boldsymbol{y} = \boldsymbol{y}_{:s}^\top \boldsymbol{K}_{:s,:s}^{-1}\boldsymbol{y}_{:s} + (\boldsymbol{y}_{s+1:} - m_s(\boldsymbol{X}_{s+1:}))^\top\left(k_s(\boldsymbol{X}_{s+1:}, \boldsymbol{X}_{s+1:}) + \sigma^2(\boldsymbol{X}_{s+1:})\right)^{-1}(\boldsymbol{y}_{s+1:} - m_s(\boldsymbol{X}_{s+1:})). \tag{45}$$

Define $\boldsymbol{e} := (\boldsymbol{y}_{s+1:} - m_s(\boldsymbol{X}_{s+1:}))$ to rewrite the second addend as

$$\boldsymbol{e}^\top \operatorname{Diag}(\boldsymbol{e}) \left(\operatorname{Diag}(\boldsymbol{e}) \left(k_s(\boldsymbol{X}_{s+1:}, \boldsymbol{X}_{s+1:}) + \sigma^2(\boldsymbol{X}_{s+1:})\right) \operatorname{Diag}(\boldsymbol{e})\right)^{-1} \operatorname{Diag}(\boldsymbol{e})\boldsymbol{e} \tag{46}$$

We use a trick we first encountered in Kim & Teh (2018): $\boldsymbol{y}^\top \boldsymbol{A}^{-1}\boldsymbol{y} \geq 2\boldsymbol{y}^\top \boldsymbol{b} - \boldsymbol{b}^\top \boldsymbol{A}\boldsymbol{b}$, for any $\boldsymbol{b}$. Choose $\boldsymbol{b} := \alpha\mathbf{1}$ and observe that $\boldsymbol{b}$ is $\mathcal{F}_s$-measurable.

$$\mathbb{E}\left[\boldsymbol{y}^\top \boldsymbol{K}^{-1}\boldsymbol{y} \mid \mathcal{F}_s\right] \tag{47}$$

$$= \boldsymbol{y}_{:s}^\top \boldsymbol{K}_{:s,:s}^{-1}\boldsymbol{y}_{:s} + \mathbb{E}\left[(\boldsymbol{y}_{s+1:} - m_s(\boldsymbol{X}_{s+1:}))^\top \left(k_s(\boldsymbol{X}_{s+1:}, \boldsymbol{X}_{s+1:}) + \sigma^2(\boldsymbol{X}_{s+1:})\right)^{-1} (\boldsymbol{y}_{s+1:} - m_s(\boldsymbol{X}_{s+1:})) \mid \mathcal{F}_s\right] \tag{48}$$

// *since $\boldsymbol{y}_{:s}^\top \boldsymbol{K}_{:s,:s}^{-1}\boldsymbol{y}_{:s}$ is $\mathcal{F}_s$-measurable*

$$\geq \boldsymbol{y}_{:s}^\top \boldsymbol{K}_{:s,:s}^{-1}\boldsymbol{y}_{:s} + 2\alpha\mathbf{1}^\top\mathbb{E}\left[\operatorname{Diag}(\boldsymbol{e})\boldsymbol{e}\right] - \alpha^2\mathbf{1}^\top\mathbb{E}\left[\operatorname{Diag}(\boldsymbol{e})\left(k_s(\boldsymbol{X}_{s+1:}, \boldsymbol{X}_{s+1:}) + \sigma^2(\boldsymbol{X}_{s+1:})\right)\operatorname{Diag}(\boldsymbol{e}) \mid \mathcal{F}_s\right]\mathbf{1} \tag{49}$$

// *applying the inequality for quadratic forms and using the $\mathcal{F}_s$-measurabilituy of α*

$$\begin{aligned}
&= \boldsymbol{y}_{:s}^\top \boldsymbol{K}_{:s,:s}^{-1}\boldsymbol{y}_{:s} + 2\alpha(N-s)\mathbb{E}[(y_{s+1} - m_s(\boldsymbol{x}_{s+1}))^2 \mid \mathcal{F}_s] \\
&\quad - \alpha^2(N-s)\mathbb{E}[(y_{s+1} - m_s(\boldsymbol{x}_{s+1}))^2(k_s(\boldsymbol{x}_{s+1}, \boldsymbol{x}_{s+1}) + \sigma^2(\boldsymbol{x}_{s+1})) \mid \mathcal{F}_s] \\
&\quad - \alpha^2\left((N-s)\frac{N-s+1}{2} - (N-s)\right)\mathbb{E}[(y_{s+1} - m_s(\boldsymbol{x}_{s+1}))(y_{s+2} - m_s(\boldsymbol{x}_{s+2}))k_s(\boldsymbol{x}_{s+1}, \boldsymbol{x}_{s+2}) \mid \mathcal{F}_s]
\end{aligned} \tag{50}$$

// *using Assumption 3, grouping variance and covariance terms separately*

$$\begin{aligned}
&= \boldsymbol{y}_{:s}^\top \boldsymbol{K}_{:s,:s}^{-1}\boldsymbol{y}_{:s} + 2\alpha(N-s)\mathbb{E}[(y_{s+1} - m_s(\boldsymbol{x}_{s+1}))^2 \mid \mathcal{F}_s] \\
&\quad - \alpha^2(N-s)\mathbb{E}[(y_{s+1} - m_s(\boldsymbol{x}_{s+1}))^2(k_s(\boldsymbol{x}_{s+1}, \boldsymbol{x}_{s+1}) + \sigma^2(\boldsymbol{x}_{s+1})) \mid \mathcal{F}_s] \\
&\quad - \alpha^2(N-s)\frac{N-s-1}{2}\mathbb{E}[(y_{s+1} - m_s(\boldsymbol{x}_{s+1}))(y_{s+2} - m_s(\boldsymbol{x}_{s+2}))k_s(\boldsymbol{x}_{s+1}, \boldsymbol{x}_{s+2}) \mid \mathcal{F}_s]
\end{aligned} \tag{51}$$

// *simplifying*

□

J Utility Proofs

Lemma 16 (Bounding the relative error)**.** *Let $D, \hat{D} \in [\mathcal{L}, \mathcal{U}]$, and assume* $\operatorname{sign}(\mathcal{L}) = \operatorname{sign}(\mathcal{U}) \neq 0$. *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 $\mathrm{sign}(\mathcal{L}) = \mathrm{sign}(\mathcal{U})$ 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 17. *The* log*-determinant of a kernel matrix can be written as a sum of conditional variances.*

$$\log\det \boldsymbol{K} = \sum_{j=1}^{N} \log(k_{j-1}(\boldsymbol{x}_j, \boldsymbol{x}_j) + \sigma^2(\boldsymbol{x}_j)) \tag{52}$$

Proof. Denote with $\boldsymbol{L}$ the Cholesky decomposition of $\boldsymbol{K}$. Then we obtain

$$\begin{aligned}\log\det \boldsymbol{K} &= \log\det(\boldsymbol{L}\boldsymbol{L}^\top) && (53)\\ &\text{// using } \boldsymbol{A} = \boldsymbol{L}\boldsymbol{L}^\top \\ &= \log\left(\det(\boldsymbol{L})\det(\boldsymbol{L}^\top)\right) && (54)\\ &\text{// for square matrices } \boldsymbol{B}, \boldsymbol{C}\text{: } \det(\boldsymbol{B}\boldsymbol{C}) = \det(\boldsymbol{B})\det(\boldsymbol{C}) \\ &= \log\left(\prod_{j=1}^{N} \boldsymbol{L}_{jj}^2\right) && (55)\\ &\text{// for triangular matrices the determinant is the product of the diagonal elements} \\ &= \sum_{j=1}^{N} 2\log \boldsymbol{L}_{jj} && (56)\\ &\text{// property of } \log\end{aligned}$$

With Lemma 21 the result follows. □

Lemma 18. *The term $\boldsymbol{y}^\top\boldsymbol{K}^{-1}\boldsymbol{y}$ can be written as*

$$\boldsymbol{y}^\top\boldsymbol{K}^{-1}\boldsymbol{y} = \sum_{n=1}^{N} \frac{(y_n - m_{n-1}(\boldsymbol{x}_n))}{k_{n-1}(\boldsymbol{x}_n, \boldsymbol{x}_n) + \sigma^2(\boldsymbol{x}_n)}. \tag{57}$$

Proof. Define

$$\begin{aligned}\boldsymbol{k}_j(\boldsymbol{x}) &:= [k(\boldsymbol{x}, \boldsymbol{x}_1), ..., k(\boldsymbol{x}, \boldsymbol{x}_j)]^\top \in \mathbb{R}^j \\ \boldsymbol{k}_{j+1} &:= \boldsymbol{k}_j(\boldsymbol{x}_{j+1}) \in \mathbb{R}^j \\ p_j &:= k(\boldsymbol{x}_j, \boldsymbol{x}_j) + \sigma^2 - \boldsymbol{k}_j^\top(\boldsymbol{K}_{j-1} + \sigma^2\boldsymbol{I})^{-1}\boldsymbol{k}_j \\ \boldsymbol{\alpha} &:= (\boldsymbol{K}_n + \sigma^2\boldsymbol{I})^{-1}\boldsymbol{k}_{n+1}\end{aligned}$$

First note, that using block-matrix inversion we can write

$$(\boldsymbol{K}_{n+1} + \sigma^2\boldsymbol{I})^{-1} = \begin{bmatrix}(\boldsymbol{K}_n + \sigma^2\boldsymbol{I})^{-1} + \boldsymbol{\alpha}p_{n+1}^{-1}\boldsymbol{\alpha}^\top & -\boldsymbol{\alpha}p_{n+1}^{-1} \\ -\boldsymbol{\alpha}^\top p_{n+1}^{-1} & p_{n+1}^{-1}\end{bmatrix}.$$

This allows to write

$$
\begin{aligned}
&\boldsymbol{y}_{n+1}^\top(\boldsymbol{K}_{n+1}+\sigma^2\boldsymbol{I})^{-1}\boldsymbol{y}_{n+1}\\
&=\begin{bmatrix}\boldsymbol{y}_n^\top & y_{n+1}\end{bmatrix}\begin{bmatrix}(\boldsymbol{K}_n+\sigma^2\boldsymbol{I})^{-1}+\boldsymbol{\alpha}p_{n+1}^{-1}\boldsymbol{\alpha}^\top & -\boldsymbol{\alpha}p_{n+1}^{-1}\\ -\boldsymbol{\alpha}^\top p_{n+1}^{-1} & p_{n+1}^{-1}\end{bmatrix}\begin{bmatrix}\boldsymbol{y}_n\\ y_{n+1}\end{bmatrix}\\
&\text{// \textit{using above observation}}\\
&=\begin{bmatrix}\boldsymbol{y}_n^\top & y_{n+1}\end{bmatrix}\begin{bmatrix}(\boldsymbol{K}_n+\sigma^2\boldsymbol{I})^{-1}\boldsymbol{y}_n+\boldsymbol{\alpha}p_{n+1}^{-1}\boldsymbol{\alpha}^\top\boldsymbol{y}_n-\boldsymbol{\alpha}p_{n+1}^{-1}y_{n+1}\\ -\boldsymbol{\alpha}^\top p_{n+1}^{-1}\boldsymbol{y}_n+p_{n+1}^{-1}y_{n+1}\end{bmatrix}\\
&\text{// \textit{simplifying from the right}}\\
&=\boldsymbol{y}_n^\top(\boldsymbol{K}_n+\sigma^2\boldsymbol{I})^{-1}\boldsymbol{y}_n+\boldsymbol{y}_n^\top\boldsymbol{\alpha}p_{n+1}^{-1}\boldsymbol{\alpha}^\top\boldsymbol{y}_n-\boldsymbol{y}_n^\top\boldsymbol{\alpha}p_{n+1}^{-1}y_{n+1}-y_{n+1}\boldsymbol{\alpha}^\top p_{n+1}^{-1}\boldsymbol{y}_n+y_{n+1}p_{n+1}^{-1}y_{n+1}\\
&\text{// \textit{simplifying from the left}}\\
&=\boldsymbol{y}_n^\top(\boldsymbol{K}_n+\sigma^2\boldsymbol{I})^{-1}\boldsymbol{y}_n+p_{n+1}^{-1}(\boldsymbol{y}_n^\top\boldsymbol{\alpha}\boldsymbol{\alpha}^\top\boldsymbol{y}_n-\boldsymbol{y}_n^\top\boldsymbol{\alpha}y_{n+1}-y_{n+1}\boldsymbol{\alpha}^\top\boldsymbol{y}_n+y_{n+1}y_{n+1})\\
&\text{// \textit{pulling out} } p_{n+1}^{-1}\\
&=\boldsymbol{y}_n^\top(\boldsymbol{K}_n+\sigma^2\boldsymbol{I})^{-1}\boldsymbol{y}_n+p_{n+1}^{-1}((\boldsymbol{y}_n^\top\boldsymbol{\alpha})^2-2\boldsymbol{y}_n^\top\boldsymbol{\alpha}y_{n+1}+y_{n+1}^2)\\
&\text{// \textit{simplifying}}\\
&=\boldsymbol{y}_n^\top(\boldsymbol{K}_n+\sigma^2\boldsymbol{I})^{-1}\boldsymbol{y}_n+p_{n+1}^{-1}(\boldsymbol{y}_n^\top\boldsymbol{\alpha}-y_{n+1})^2\\
&\text{// \textit{simplifying}}
\end{aligned}
$$

Now observe that the last addend is indeed the mean square error divided by the posterior variance. By induction the result follows. □

Lemma 19. *For all $t, m \in \mathbb{N}$ with $1 \le t+m \le N$*

$$
k_{t+m}(\boldsymbol{x}_a,\boldsymbol{x}_b)=k_t(\boldsymbol{x}_a,\boldsymbol{x}_b)-k_t(\boldsymbol{x}_a,\overline{\boldsymbol{X}})\left(k_t(\overline{\boldsymbol{X}})+\sigma^2\boldsymbol{I}_m\right)^{-1}k_t(\overline{\boldsymbol{X}},\boldsymbol{x}_b)
$$

where $k_t(\boldsymbol{x}_a,\boldsymbol{x}_b) := k(\boldsymbol{x}_a,\boldsymbol{x}_b)-k(\boldsymbol{x}_a,\boldsymbol{X}_t)\left(k(\boldsymbol{X}_{:t},\boldsymbol{X}_{:t})+\sigma^2(\boldsymbol{X}_{:t})\right)^{-1}k(\boldsymbol{X}_t,\boldsymbol{x}_b)$ and $\overline{\boldsymbol{X}} := \boldsymbol{X}_{t:t+m}$.

Proof.

$$
\begin{aligned}
&k_{t+m}(\boldsymbol{x}_a,\boldsymbol{x}_b)\\
&=k(\boldsymbol{x}_a,\boldsymbol{x}_b)-k(\boldsymbol{x}_a,\boldsymbol{X}_{t+m})\boldsymbol{A}_{t+m}^{-1}k(\boldsymbol{X}_{t+m},\boldsymbol{x}_b)\\
&\text{// \textit{by definition}}\\
&=k(\boldsymbol{x}_a,\boldsymbol{x}_b)-\begin{bmatrix}k(\boldsymbol{x}_a,\boldsymbol{X}_t) & k(\boldsymbol{x}_a,\overline{\boldsymbol{X}})\end{bmatrix}\begin{bmatrix}k(\boldsymbol{X}_t)+\sigma^2\boldsymbol{I}_t & k(\boldsymbol{X}_t,\overline{\boldsymbol{X}})\\ k(\overline{\boldsymbol{X}},\boldsymbol{X}_t) & k(\overline{\boldsymbol{X}})+\sigma^2\boldsymbol{I}_t\end{bmatrix}^{-1}\begin{bmatrix}k(\boldsymbol{X}_t,\boldsymbol{x}_b)\\ k(\overline{\boldsymbol{X}},\boldsymbol{x}_b)\end{bmatrix}\\
&\text{// \textit{in block notation}}\\
&=k(\boldsymbol{x}_a,\boldsymbol{x}_b)-\begin{bmatrix}k(\boldsymbol{x}_a,\boldsymbol{X}_t) & k(\boldsymbol{x}_a,\overline{\boldsymbol{X}})\end{bmatrix}\begin{bmatrix}\boldsymbol{A}_t & k(\boldsymbol{X}_t,\overline{\boldsymbol{X}})\\ k(\overline{\boldsymbol{X}},\boldsymbol{X}_t) & k(\overline{\boldsymbol{X}})+\sigma^2\boldsymbol{I}_t\end{bmatrix}^{-1}\begin{bmatrix}k(\boldsymbol{X}_t,\boldsymbol{x}_b)\\ k(\overline{\boldsymbol{X}},\boldsymbol{x}_b)\end{bmatrix}\\
&\text{// \textit{using the definition of} } \boldsymbol{A}_t\\
&=k(\boldsymbol{x}_a,\boldsymbol{x}_b)-\begin{bmatrix}k(\boldsymbol{x}_a,\boldsymbol{X}_t) & k(\boldsymbol{x}_a,\overline{\boldsymbol{X}})\end{bmatrix}\cdot\\
&\begin{bmatrix}\boldsymbol{A}_t^{-1}+\boldsymbol{A}_t^{-1}k(\boldsymbol{X}_t,\overline{\boldsymbol{X}})\left(k(\overline{\boldsymbol{X}})+\sigma^2\boldsymbol{I}_t-k(\overline{\boldsymbol{X}},\boldsymbol{X}_t)\boldsymbol{A}_t^{-1}k(\boldsymbol{X}_t,\overline{\boldsymbol{X}})\right)^{-1}k(\overline{\boldsymbol{X}},\boldsymbol{X}_t)\boldsymbol{A}_t^{-1} & -\boldsymbol{A}_t^{-1}k(\boldsymbol{X}_t,\overline{\boldsymbol{X}})\left(k(\overline{\boldsymbol{X}})+\sigma^2\boldsymbol{I}_t-k(\overline{\boldsymbol{X}},\boldsymbol{X}_t)\boldsymbol{A}_t^{-1}k(\boldsymbol{X}_t,\overline{\boldsymbol{X}})\right)^{-1}\\ -\left(k(\overline{\boldsymbol{X}})+\sigma^2\boldsymbol{I}_t-k(\overline{\boldsymbol{X}},\boldsymbol{X}_t)\boldsymbol{A}_t^{-1}k(\boldsymbol{X}_t,\overline{\boldsymbol{X}})\right)^{-1}k(\overline{\boldsymbol{X}},\boldsymbol{X}_t)\boldsymbol{A}_t^{-1} & \left(k(\overline{\boldsymbol{X}})+\sigma^2\boldsymbol{I}_t-k(\overline{\boldsymbol{X}},\boldsymbol{X}_t)\boldsymbol{A}_t^{-1}k(\boldsymbol{X}_t,\overline{\boldsymbol{X}})\right)^{-1}\end{bmatrix}\cdot\\
&\begin{bmatrix}k(\boldsymbol{X}_t,\boldsymbol{x}_b)\\ k(\overline{\boldsymbol{X}},\boldsymbol{x}_b)\end{bmatrix}\\
&\text{// \textit{applying block-matrix inversion}}\\
&=k(\boldsymbol{x}_a,\boldsymbol{x}_b)-\begin{bmatrix}k(\boldsymbol{x}_a,\boldsymbol{X}_t) & k(\boldsymbol{x}_a,\overline{\boldsymbol{X}})\end{bmatrix}\cdot\\
&\begin{bmatrix}\boldsymbol{A}_t^{-1}+\boldsymbol{A}_t^{-1}k(\boldsymbol{X}_t,\overline{\boldsymbol{X}})\left(k_t(\overline{\boldsymbol{X}})+\sigma^2\boldsymbol{I}_t\right)^{-1}k(\overline{\boldsymbol{X}},\boldsymbol{X}_t)\boldsymbol{A}_t^{-1} & -\boldsymbol{A}_t^{-1}k(\boldsymbol{X}_t,\overline{\boldsymbol{X}})\left(k_t(\overline{\boldsymbol{X}})+\sigma^2\boldsymbol{I}_t\right)^{-1}\\ -\left(k_t(\overline{\boldsymbol{X}})+\sigma^2\boldsymbol{I}_t\right)^{-1}k(\overline{\boldsymbol{X}},\boldsymbol{X}_t)\boldsymbol{A}_t^{-1} & \left(k_t(\overline{\boldsymbol{X}})+\sigma^2\boldsymbol{I}_t\right)^{-1}\end{bmatrix}\cdot
\end{aligned}
$$

$$\begin{bmatrix} k(\boldsymbol{X}_t, \boldsymbol{x}_b) \\ k(\overline{\boldsymbol{X}}, \boldsymbol{x}_b) \end{bmatrix}$$

// *applying the definition of* k_t

$$= k(\boldsymbol{x}_a, \boldsymbol{x}_b) - \begin{bmatrix} k(\boldsymbol{x}_a, \boldsymbol{X}_t) & k(\boldsymbol{x}_a, \overline{\boldsymbol{X}}) \end{bmatrix} \cdot$$

$$\begin{bmatrix} \boldsymbol{A}_t^{-1} k(\boldsymbol{X}_t, \boldsymbol{x}_b) + \boldsymbol{A}_t^{-1} k(\boldsymbol{X}_t, \overline{\boldsymbol{X}}) \left(k_t(\overline{\boldsymbol{X}}) + \sigma^2 \boldsymbol{I}_t\right)^{-1} k(\overline{\boldsymbol{X}}, \boldsymbol{X}_t) \boldsymbol{A}_t^{-1} k(\boldsymbol{X}_t, \boldsymbol{x}_b) - \boldsymbol{A}_t^{-1} k(\boldsymbol{X}_t, \overline{\boldsymbol{X}}) \left(k_t(\overline{\boldsymbol{X}}) + \sigma^2 \boldsymbol{I}_t\right)^{-1} k(\overline{\boldsymbol{X}}, \boldsymbol{x} \\ - \left(k_t(\overline{\boldsymbol{X}}) + \sigma^2 \boldsymbol{I}_t\right)^{-1} k(\overline{\boldsymbol{X}}, \boldsymbol{X}_t) \boldsymbol{A}_t^{-1} k(\boldsymbol{X}_t, \boldsymbol{x}_b) + \left(k_t(\overline{\boldsymbol{X}}) + \sigma^2 \boldsymbol{I}_t\right)^{-1} k(\overline{\boldsymbol{X}}, \boldsymbol{x}_b) \end{bmatrix}$$

// *evaluating multiplication with right-most vector*

$$= k(\boldsymbol{x}_a, \boldsymbol{x}_b) - \begin{bmatrix} k(\boldsymbol{x}_a, \boldsymbol{X}_t) & k(\boldsymbol{x}_a, \overline{\boldsymbol{X}}) \end{bmatrix} \cdot$$

$$\begin{bmatrix} \boldsymbol{A}_t^{-1} k(\boldsymbol{X}_t, \boldsymbol{x}_b) - \boldsymbol{A}_t^{-1} k(\boldsymbol{X}_t, \overline{\boldsymbol{X}}) \left(k_t(\overline{\boldsymbol{X}}) + \sigma^2 \boldsymbol{I}_t\right)^{-1} \left(k(\overline{\boldsymbol{X}}, \boldsymbol{x}_b) - k(\overline{\boldsymbol{X}}, \boldsymbol{X}_t) \boldsymbol{A}_t^{-1} k(\boldsymbol{X}_t, \boldsymbol{x}_b)\right) \\ \left(k_t(\overline{\boldsymbol{X}}) + \sigma^2 \boldsymbol{I}_t\right)^{-1} \left(k(\overline{\boldsymbol{X}}, \boldsymbol{x}_b) - k(\overline{\boldsymbol{X}}, \boldsymbol{X}_t) \boldsymbol{A}_t^{-1} k(\boldsymbol{X}_t, \boldsymbol{x}_b)\right) \end{bmatrix}$$

// *rearranging*

$$= k(\boldsymbol{x}_a, \boldsymbol{x}_b) - \begin{bmatrix} k(\boldsymbol{x}_a, \boldsymbol{X}_t) & k(\boldsymbol{x}_a, \overline{\boldsymbol{X}}) \end{bmatrix} \cdot$$

$$\begin{bmatrix} \boldsymbol{A}_t^{-1} k(\boldsymbol{X}_t, \boldsymbol{x}_b) - \boldsymbol{A}_t^{-1} k(\boldsymbol{X}_t, \overline{\boldsymbol{X}}) \left(k_t(\overline{\boldsymbol{X}}) + \sigma^2 \boldsymbol{I}_t\right)^{-1} k_t(\overline{\boldsymbol{X}}, \boldsymbol{x}_b) \\ \left(k_t(\overline{\boldsymbol{X}}) + \sigma^2 \boldsymbol{I}_t\right)^{-1} k_t(\overline{\boldsymbol{X}}, \boldsymbol{x}_b) \end{bmatrix}$$

// *applying the definition of* k_t

$$= k(\boldsymbol{x}_a, \boldsymbol{x}_b) - k(\boldsymbol{x}_a, \boldsymbol{X}_t) \boldsymbol{A}_t^{-1} k(\boldsymbol{X}_t, \boldsymbol{x}_b)$$
$$+ k(\boldsymbol{x}_a, \boldsymbol{X}_t) \boldsymbol{A}_t^{-1} k(\boldsymbol{X}_t, \overline{\boldsymbol{X}}) \left(k_t(\overline{\boldsymbol{X}}) + \sigma^2 \boldsymbol{I}_t\right)^{-1} k_t(\overline{\boldsymbol{X}}, \boldsymbol{x}_b) - k(\boldsymbol{x}_a, \overline{\boldsymbol{X}}) \left(k_t(\overline{\boldsymbol{X}}) + \sigma^2 \boldsymbol{I}_t\right)^{-1} k_t(\overline{\boldsymbol{X}}, \boldsymbol{x}_b)$$

// *evaluating the vector product*

$$= k(\boldsymbol{x}_a, \boldsymbol{x}_b) - k(\boldsymbol{x}_a, \boldsymbol{X}_t) \boldsymbol{A}_t^{-1} k(\boldsymbol{X}_t, \boldsymbol{x}_b) - \left(k(\boldsymbol{x}_a, \overline{\boldsymbol{X}}) - k(\boldsymbol{x}_a, \boldsymbol{X}_t) \boldsymbol{A}_t^{-1} k(\boldsymbol{X}_t, \overline{\boldsymbol{X}})\right) \left(k_t(\overline{\boldsymbol{X}}) + \sigma^2 \boldsymbol{I}_t\right)^{-1} k_t(\overline{\boldsymbol{X}}, \boldsymbol{x}_b)$$

// *rearranging*

$$= k_t(\boldsymbol{x}_a, \boldsymbol{x}_b) - k_t(\boldsymbol{x}_a, \overline{\boldsymbol{X}}) \left(k_t(\overline{\boldsymbol{X}}) + \sigma^2 \boldsymbol{I}_t\right)^{-1} k_t(\overline{\boldsymbol{X}}, \boldsymbol{x}_b)$$

// *applying the definition of* k_t

□

Lemma 20.

$$\sum_{j=t+1}^{n} \sum_{i=t_0+1}^{j-1} 1 = (n-t)\left(\frac{n+t+1}{2} - t_0\right) \tag{58}$$

Proof.

$$\begin{aligned} \sum_{j=t+1}^{n} \sum_{i=t_0+1}^{j-1} 1 &= \sum_{j=t+1}^{n} (j-1-t_0) && (59) \\ &= \sum_{j=0}^{n-t-1} (j-1-t_0+t+1) && (60) \\ &= (t-t_0)(n-t) + \sum_{j=0}^{n-t-1} j && (61) \\ &= (t-t_0)(n-t) + \frac{(n-t-1)(n-t)}{2} && (62) \\ &= (n-t)\left(\frac{n-t-1}{2} + t - t_0\right) && (63) \\ &= (n-t)\left(\frac{n+t-1}{2} - t_0\right) && (64) \end{aligned}$$

□

Lemma 21 (Link between the Cholesky and Gaussian process regression)**.** *Denote with $\boldsymbol{C}_N$ the Cholesky decomposition of $\boldsymbol{K}$, so that $\boldsymbol{C}_N\boldsymbol{C}_N^\top = \boldsymbol{K}$. The n-th diagonal element of $\boldsymbol{C}_N$, squared, is equivalent to $k_{n-1}(\boldsymbol{x}_n, \boldsymbol{x}_n) + \sigma^2(\boldsymbol{x}_n)$:*

$$[\boldsymbol{C}_N]^2_{nn} = k_{n-1}(\boldsymbol{x}_n, \boldsymbol{x}_n) + \sigma^2(\boldsymbol{x}_n)\,.$$

Proof. With abuse of notation, define $\boldsymbol{C}_1 := \sqrt{k(\boldsymbol{x}_1, \boldsymbol{x}_1)}$ and

$$\boldsymbol{C}_N := \begin{bmatrix} \boldsymbol{C}_{N-1} & \boldsymbol{0} \\ \boldsymbol{k}_N^\top \boldsymbol{C}_{N-1}^{-\top} & \sqrt{k(\boldsymbol{x}_N, \boldsymbol{x}_N) + \sigma^2 - \boldsymbol{k}_N^\top (\boldsymbol{K}_{n-1} + \sigma^2 \boldsymbol{I}_{n-1})^{-1} \boldsymbol{k}_N} \end{bmatrix}.$$

We will show that the lower triangular matrix $\boldsymbol{C}_N$ satisfies $\boldsymbol{C}_N\boldsymbol{C}_N^\top = \boldsymbol{K}_N + \sigma^2\boldsymbol{I}_N$. Since the Cholesky decomposition is unique (Golub & Van Loan, 2013, Theorem 4.2.7), $\boldsymbol{C}_N$ must be the Cholesky decomposition of $\boldsymbol{K}$. Furthermore, by definition of $\boldsymbol{C}_N$, $[\boldsymbol{C}_N]^2_{NN} = k(\boldsymbol{x}_N, \boldsymbol{x}_N) + \sigma^2 - \boldsymbol{k}_N^\top (\boldsymbol{K}_{n-1} + \sigma^2 \boldsymbol{I}_{n-1})^{-1} \boldsymbol{k}_N$. The statement then follows by induction.

To remain within the text margins, define

$$x := \boldsymbol{k}_N^\top \boldsymbol{C}_{N-1}^{-\top} \boldsymbol{C}_{N-1}^{-1} \boldsymbol{k}_N + k(\boldsymbol{x}_N, \boldsymbol{x}_N) + \sigma^2 - \boldsymbol{k}_N^\top (\boldsymbol{K}_{n-1} + \sigma^2 \boldsymbol{I}_{n-1})^{-1} \boldsymbol{k}_N.$$

We want to show that $\boldsymbol{C}_N\boldsymbol{C}_N^\top = \boldsymbol{K}_N + \sigma^2\boldsymbol{I}_N$.

$$\begin{aligned}
\boldsymbol{C}_N\boldsymbol{C}_N^\top &= \begin{bmatrix} \boldsymbol{C}_{N-1} & \boldsymbol{0} \\ \boldsymbol{k}_N^\top \boldsymbol{C}_{N-1}^{-\top} & \sqrt{k(\boldsymbol{x}_N, \boldsymbol{x}_N) + \sigma^2 - \boldsymbol{k}_N^\top (\boldsymbol{K}_{n-1} + \sigma^2 \boldsymbol{I}_{n-1})^{-1} \boldsymbol{k}_N} \end{bmatrix} \\
&\quad \cdot \begin{bmatrix} \boldsymbol{C}_{N-1}^\top & \boldsymbol{C}_{N-1}^{-1}\boldsymbol{k}_N \\ \boldsymbol{0}^\top & \sqrt{k(\boldsymbol{x}_N, \boldsymbol{x}_N) + \sigma^2 - \boldsymbol{k}_N^\top (\boldsymbol{K}_{n-1} + \sigma^2 \boldsymbol{I}_{n-1})^{-1} \boldsymbol{k}_N} \end{bmatrix} \\
&= \begin{bmatrix} \boldsymbol{C}_{N-1}\boldsymbol{C}_{N-1}^\top & \boldsymbol{C}_{N-1}\boldsymbol{C}_{N-1}^{-1}\boldsymbol{k}_N \\ \boldsymbol{k}_N^\top \boldsymbol{C}_{N-1}^{-\top}\boldsymbol{C}_{N-1}^\top & x \end{bmatrix} \\
&= \begin{bmatrix} \boldsymbol{K}_{N-1} + \sigma^2 \boldsymbol{I}_{N-1} & \boldsymbol{k}_N \\ \boldsymbol{k}_N^\top & x \end{bmatrix}
\end{aligned}$$

Also x can be simplified further.

$$\begin{aligned}
x &= \boldsymbol{k}_N^\top \boldsymbol{C}_{N-1}^{-\top} \boldsymbol{C}_{N-1}^{-1} \boldsymbol{k}_N + k(\boldsymbol{x}_N, \boldsymbol{x}_N) + \sigma^2 - \boldsymbol{k}_N^\top (\boldsymbol{K}_{n-1} + \sigma^2 \boldsymbol{I}_{n-1})^{-1} \boldsymbol{k}_N \\
&= \boldsymbol{k}_N^\top (\boldsymbol{K}_{n-1} + \sigma^2 \boldsymbol{I}_{n-1})^{-1} \boldsymbol{k}_N + k(\boldsymbol{x}_N, \boldsymbol{x}_N) + \sigma^2 - \boldsymbol{k}_N^\top (\boldsymbol{K}_{n-1} + \sigma^2 \boldsymbol{I}_{n-1})^{-1} \boldsymbol{k}_N \\
&= k(\boldsymbol{x}_N, \boldsymbol{x}_N) + \sigma^2.
\end{aligned}$$

□

References

Artemev, A., Burt, D. R., and van der Wilk, M. Tighter bounds on the log marginal likelihood of gaussian process regression using conjugate gradients. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 362–372, 2021.

Bartels, S. *Probabilistic Linear Algebra*. PhD thesis, University of Tübingen, 2020.

Camachol, R. Inducing models of human control skills. In Nédellec, C. and Rouveirol, C. (eds.), *Machine Learning: ECML-98*, pp. 107–118, 1998.

Fan, X., Grama, I., and Liu, Q. Hoeffding's inequality for supermartingales. *Stochastic Processes and their Applications*, 122(10):3545–3559, 2012.

Fanaee-T, H. and Gama, J. Event labeling combining ensemble detectors and background knowledge. *Progress in Artificial Intelligence*, pp. 1–15, 2013.

George, A., Heath, M. T., and Liu, J. Parallel cholesky factorization on a shared-memory multiprocessor. *Linear Algebra and its Applications*, 77:165–187, 1986.

Golub, G. and Van Loan, C. *Matrix computations*. Johns Hopkins Univ Pr, 4 edition, 2013.

Kim, H. and Teh, Y. W. Scaling up the automatic statistician: Scalable structure discovery using gaussian processes. In Storkey, A. and Perez-Cruz, F. (eds.), *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, pp. 575–584, 2018.

Liang, X., Zou, T., Guo, B., Li, S., Zhang, H., Zhang, S., Huang, H., and Chen, S. X. 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.

Mnih, V., Szepesvári, C., and Audibert, J.-Y. Empirical Bernstein stopping. pp. 672–679, 2008.

Quiñonero-Candela, J. and Rasmussen, C. A unifying view of sparse approximate Gaussian process regression. *J of Machine Learning Research*, 6:1939–1959, 2005.

Schwaighofer, A. and Tresp, V. Transductive and inductive methods for approximate gaussian process regression. In Becker, S., Thrun, S., and Obermayer, K. (eds.), *Advances in Neural Information Processing Systems*, volume 15. MIT Press, 2002.

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., Carey, C. J., Polat, İ., Feng, Y., Moore, E. W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E. A., Harris, C. R., Archibald, A. M., Ribeiro, A. H., Pedregosa, F., van Mulbregt, P., and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.

Weiss, S. M. and Indurkhya, N. Rule-based machine learning methods for functional prediction. *Journal of Artificial Intelligence Research*, 3(1):383–403, 1995.