Safe Active Learning for Multi-Output Gaussian Processes

Cen-You Li Cen-You.Li@de.bosch.com
Barbara Rakitsch Barbara.Rakitsch@de.bosch.com
Christoph Zimmer Christoph.Zimmer@de.bosch.com
Bosch Center for Artificial Intelligence
Robert-Bosch-Campus 1, 71272 Renningen, Germany

Abstract

Multi-output regression problems are commonly encountered in science and engineering. In particular, multi-output Gaussian processes have been emerged as a promising tool for modeling these complex systems since they can exploit the inherent correlations and provide reliable uncertainty estimates. In many applications, however, acquiring the data is expensive and safety concerns might arise (e.g. robotics, engineering). We propose a safe active learning approach for multi-output Gaussian process regression. This approach queries the most informative data or output taking the relatedness between the regressors and safety constraints into account. We prove the effectiveness of our approach by providing theoretical analysis and by demonstrating empirical results on simulated datasets and on a real-world engineering dataset. On all datasets, our approach shows improved convergence compared to its competitors.

1 Introduction

Active learning (AL) selects the most informative data sequentially according to previous measurements and an acquisition function (Krause et al., 2008; Houlsby et al., 2011; Zhang et al., 2016). The objective is to optimize a model without labeling unnecessary data. The problem setup is closely related to Bayesian optimization, i.e. BO (Brochu et al., 2010), which optimizes a black-box function with limited exploration. In various scenarios, safety concerns are also critical during the exploration phase. For instance, movements of a machine are not supposed to crash any objects. A system should avoid generating high pressure, high temperature, or explosion. Safe learning addresses this by incorporating and learning safety constraints (Sui et al., 2015). Schreiter et al. (2015) and Zimmer et al. (2018) combine safety considerations with AL so that the data selection is done only in the determined safe domain.

These works, however, rarely considered multi-output (MO) regression problems, despite them commonly encountered in science, engineering and medicine (Xu et al., 2019; Zhang and Yang, 2021; Liu et al., 2018). In such problems, it is possible to consider individual tasks or outputs independently, but the plausibly shared mechanisms are ignored, and the performances or data efficiency might be deteriorated. Zhang et al. (2016) dealt with AL on MO models but focused on efficient computation of AL with large datasets and safe exploration was not addressed.

We consider safe AL for MO regression models that exploit the correlations. In particular, we focus on problems in which different output components may not be synchronously observed (e.g. due to different measuring cost or difficulty). MO Gaussian processes (GPs) are natural candidates for these problems (Bonilla et al., 2008; Álvarez and Lawrence, 2011; Álvarez et al., 2012; van der Wilk et al., 2020), due to their capability of capturing the correlations among different outputs and of quantifying the uncertainty.

In our work, we consider as main model the Linear Model of Coregionalization (LMC, Journel and Huijbregts (1976)), in which each output is modeled as a weighted sum of shared latent functions. Each latent function is drawn from a GP. Later on, we extend the theoretical analysis also to the convolution process (Higdon, 2002; Álvarez and Lawrence, 2011) in which each latent function is additionally convolved by an output-specific smoothing kernel.

To the best of our knowledge, this is the first framework about safe AL for MOGP regression. Our con-

Proceedings of the 25$^{\text{th}}$ International Conference on Artificial Intelligence and Statistics (AISTATS) 2022, Valencia, Spain. PMLR: Volume 151.

tributions can be summarized as follows:

- We formulate an acquisition function for safe active learning in the MOGP framework that allows asynchronous measurements.
- We provide theoretical analysis of the safe AL algorithm in our framework, particularly we derive a convergence rate to the algorithm.
- We demonstrate the performance and superiority to state-of-the-art competitors on a real-world engineering dataset.

The overview of this papers is as follows. In section 2, we briefly review the related works. In section 3, we introduce our algorithm. We discuss the theory behind our algorithm in section 4, and validate empirically its usefulness in section 5. Finally, section 6 concludes our work.

2 Related Work

AL has been extensively investigated for classification tasks (Hoi et al., 2006; Joshi et al., 2009; Houlsby et al., 2011; Hahn et al., 2019; Shi and Yu, 2021), but less literature addresses AL in the regression setting (Krause et al., 2008; Garnett et al., 2014). The problem setup is closely related to BO (Brochu et al., 2010). While AL and BO both consider limited exploration, the goals are very different. AL aims to obtain a well performing model, usually with characteristic of overall precision, but BO only finds an optimum, e.g. a configuration of best performance or lowest cost. In a BO problem, the model quality for points far away from the optimum is not important and can be really bad. For a more general problem in this line of research, i.e. optimizing under uncertainty, GPs, which are capable of making predictions under uncertainty, are often used as surrogate models (Brochu et al., 2010; Srinivas et al., 2012).

In recent years, the importance of safety considerations has led to a novel line of research ranging from Safe Bayesian optimization (Sui et al., 2015; Berkenkamp et al., 2016, 2020) to safe AL in a static environment (Schreiter et al., 2015) or dynamic systems (Zimmer et al., 2018). None of these contributions, however, consider MO which is able to exploit correlations among outputs.

Exploiting MO correlations has been shown successful in various applications (Casale et al., 2017; Liu et al., 2018; Cheng et al., 2020). State of the art MO models, in particular with GPs (also refer to Álvarez et al. (2012) and van der Wilk et al. (2020) for an overview over MOGPs), include the Linear Model of Coregionalization (LMC), a simple yet effective model (Journel and Huijbregts, 1976; Bonilla et al., 2008; Teh et al., 2005), and one of its extensions, the convolution process, which further captures correlations in multiple outputs that vary in smoothness (Higdon, 2002; Álvarez and Lawrence, 2011).

Complexity of GPs and MOGPs scales cubically with the number of observations (Rasmussen and Williams, 2006). Existing works focus a lot on approximation methods for large datasets with GPs (Titsias, 2009; Hensman et al., 2013) and with MOGPs (Álvarez and Lawrence, 2011; Nguyen and Bonilla, 2014; van der Wilk et al., 2020). In contrast to Zhang et al. (2016), both our simulation and our real-world dataset can be modeled with very few data points. We thus consider MOGPs without any sparse approximations, even though these methods could be incorporated into our approach.

Very few works have tried to combine MO modeling with BO (Swersky et al., 2013) or AL (Zhang et al., 2016). Swersky et al. (2013) focused on transferring BO results between tasks, while Zhang et al. (2016) investigated efficient computation of AL for sparse MOGPs (Zhang et al., 2016). To the best of our knowledge, none of the literature addressed safe data query or safe AL for MOGPs.

3 Methods

We first provide background on GPs and MOGPs, and different inference strategies. In a second step, we show how safe active learning can be applied over multiple outputs.

3.1 GP Regression

Single-output A GP is a stochastic process where every finite subset follows a multivariate normal distribution. In GP regression, given observed data $\mathcal{D} = \{\boldsymbol{x}_n \in \mathbb{R}^D, y_n \in \mathbb{R}\}_{n=1}^N$, we specify a mean function $m : \mathbb{R}^D \to \mathbb{R}$ and a positive definite kernel function (covariance function) $k : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}$ as a GP prior for the function. The observations y_n are assumed to be the functional values blurred by i.i.d. Gaussian noise. The model is formulated as

$$g \sim \mathcal{GP}\left(m(\cdot), k(\cdot,\cdot)\right), y_n = g(\boldsymbol{x}_n) + \epsilon_n, \epsilon_n \sim \mathcal{N}\left(0, \sigma^2\right).$$

The goal is to predict $g(\boldsymbol{x}_*)$ and its uncertainty for a new input $\boldsymbol{x}_*$. Assuming for simplicity a zero mean prior, $m \equiv 0$, the posterior is $p\left(g(\boldsymbol{x}_*)|\boldsymbol{x}_*, \mathcal{D}\right) =$

$\mathcal{N}(\mu(\boldsymbol{x}_*), var(\boldsymbol{x}_*))$, with

$$\mu(\boldsymbol{x}_*) = K_{N*}^T \left(K_{NN} + \sigma^2 I\right)^{-1} (y_1, ..., y_N)^T, \quad (1)$$

$$var(\boldsymbol{x}_*) = k(\boldsymbol{x}_*, \boldsymbol{x}_*) - K_{N*}^T \left(K_{NN} + \sigma^2 I\right)^{-1} K_{N*}, \quad (2)$$

where $K_{N*} \in \mathbb{R}^{N\times 1}$ and $K_{NN} \in \mathbb{R}^{N\times N}$ are matrices with $[K_{N*}]_i = k(\boldsymbol{x}_i, \boldsymbol{x}_*)$ and $[K_{NN}]_{i,j} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$. For further details, please see Rasmussen and Williams (2006).

Multi-output (MO) We consider LMC as our main model (Journel and Huijbregts, 1976). Here we have $\boldsymbol{y}_n = \boldsymbol{f}(\boldsymbol{x}_n) + \boldsymbol{\epsilon}_n = W\boldsymbol{g}(\boldsymbol{x}_n) + \boldsymbol{\epsilon}_n \in \mathbb{R}^P$ with i.i.d. noise $[\boldsymbol{\epsilon}_n]_p \sim \mathcal{N}(0, \sigma_p^2)$ for $p = 1, 2, ..., P$, linear transformation $W \in \mathbb{R}^{P\times L}$, and latent GPs $g_l(\cdot) = [\boldsymbol{g}(\cdot)]_l \sim \mathcal{GP}(0, k_l(\cdot, \cdot))$ for $l = 1, ..., L$.

Throughout this paper, we further assume finite P, finite L, bounded $k_l(\cdot, \cdot)$, and each element of W bounded by a constant. Let $f_p(\cdot) = [\boldsymbol{f}(\cdot)]_p$. In this model, $\{f_p(\boldsymbol{x})\}_{p=1}^P$ is also a GP where every finite subset has zero mean and covariance $cov(f_p(\boldsymbol{x}), f_{p'}(\boldsymbol{x}')) = \sum_{l=1}^L W_{pl}W_{p'l}k_l(\boldsymbol{x}, \boldsymbol{x}') =: \eta_{p,p'}(\boldsymbol{x}, \boldsymbol{x}')$.

Let $\boldsymbol{Y}$ denote the collection of observations $\{\boldsymbol{y}_n \in \mathbb{R}^P\}_{n=1}^N$, $\boldsymbol{X}$ denote $\{\boldsymbol{x}_n\}_{n=1}^N$, and $\mathcal{D} = \{\boldsymbol{X}, \boldsymbol{Y}\}$. Let y_{pn} be the p-th component of the n-th observation, i.e. $y_{pn} = [\boldsymbol{y}_n]_p$. The posterior $p(\boldsymbol{f}(\boldsymbol{x}_*)|\boldsymbol{x}_*, \mathcal{D})$ is a multivariate Gaussian $\mathcal{N}(\mu(\boldsymbol{x}_*), \Sigma(\boldsymbol{x}_*))$ with

$$\mu(\boldsymbol{x}_*) = \Omega_{N*}^T \left(\Omega_{NN} + \text{diag}(\{\sigma_i^2\}_{p=1}^P) \otimes I_N\right)^{-1} \boldsymbol{Y}, \quad (3)$$

$$\Sigma(\boldsymbol{x}_*) = \Omega_{**} - \Omega_{N*}^T \left(\Omega_{NN} + \text{diag}(\{\sigma_i^2\}_{p=1}^P) \otimes I_N\right)^{-1} \Omega_{N*}, \quad (4)$$

where $\otimes$ denotes the Kronecker product, Ω_{**}, Ω_{N*} and Ω_{NN} are gram matrices of kernel $\eta_{p,p'}(\cdot, \cdot)$. See our supplementary section A for full expression of the matrices and for the derivation of this posterior.

Notice that $\boldsymbol{Y}$ can be ordered differently, but the corresponding permutation needs to be applied to the current Ω_{N*}, Ω_{NN} and $\text{diag}(\{\sigma_i^2\}_{p=1}^P) \otimes I_N$. As the permutation matrices cancel each other out, the posterior stays in the same form with only different indexing.

Partially observed MO In the previous section, each observation of $\boldsymbol{Y}$ has every component observed for every input $\boldsymbol{X}$. In the following, we assume that some components can be omitted to save measuring costs as well as computational costs due to smaller $\Omega_{...}$. Let N_p be the number of outputs with p-th component observed and $N_{sum} = \sum_{p=1}^P N_p$. If the output is fully observed, we can see that $N_1 = ... = N_P = N$ and $N_{sum} = PN$.

For clarification, we define a reindexing bijection that maps the original index pairs to scalar indices (i.e. the scheme concatenates the outputs over all components into an one-dimensional vector), and the non-observed components are assigned with negative or zero indices. With this bijection, we can consider all observed output components by looking only at the positive indices ranging from 1 to N_{sum}.

The notation of outputs now becomes $\boldsymbol{Y}_\phi = \{y_{p_k n_k}\}_{k=1}^{N_{sum}}$, where $\phi : (p, n) \to k$ is a re-indexing bijection with $(p_k, n_k) = \phi^{-1}(k)$. The output domain of ϕ is $\mathbb{Z} \cap [-NP + N_{sum} + 1, N_{sum}]$, where $\{1, ..., N_{sum}\}$ are the new indices of all observed output components. Notice that the notation is adapted from the fully observed scenario, so ϕ is dependent of N (i.e. we clearly have $N_p \leq N, \forall p$). However, we omit N in the notation for simplicity.

In addition, the corresponding rows and columns of the gram matrices are also omitted, and when we make predictions for one output component, the notation becomes as follows:

$$\mu(\boldsymbol{x}_*, p_*) = [\Omega_{N_{sum}*}]_{all,p*}^T \widehat{\Omega}_{N_{sum}N_{sum}}^{-1} \boldsymbol{Y}_\phi, \quad (5)$$

$$\Sigma(\boldsymbol{x}_*, p_*) = \eta_{p_*,p_*}(\boldsymbol{x}_*, \boldsymbol{x}_*) - [\Omega_{N_{sum}*}]_{all,p*}^T \widehat{\Omega}_{N_{sum}N_{sum}}^{-1} [\Omega_{N_{sum}*}]_{all,p*}, \quad (6)$$

where $\widehat{\Omega}_{N_{sum}N_{sum}} = \Omega_{N_{sum}N_{sum}} + diag(\{\sigma_{p_k}^2\}_{k=1}^{N_{sum}})$. Further notice that we omit the components without changing the order, so $\{\sigma_{p_k}^2\}_{k=1}^{N_{sum}}$ is actually $\{\sigma_1^2\}_{i=1}^{N_1}$ followed by $\{\sigma_2^2\}_{i=1}^{N_2}$ and so on.

MOGPs v.s. Multiple independent single-output GPs Concatenating P single output GPs without modeling the output correlations is equivalent to a MOGP with $W = I_p$, i.e. $\eta_{p,p} = k_p$ and $\eta_{p,p'} \equiv 0$ for $p \neq p'$. Notice that in this case, the gram matrix $\Omega_{N_{sum},N_{sum}}$ and the inverse with noise variances have only non-zero components on the diagonal subblocks corresponding to $\eta_{p,p}$. The cross-covariance $\Omega_{N_{sum},*}$ has only non-zero components at $(iN + p, p)$ entries, $i = 0, ..., P - 1$, and thus the posterior is identical to squeezing the posteriors of individual GPs into one vector/matrix. See supplementary section A for full matrix expression.

On the other hand, if $W \neq I_P$, information can flow between the components which can ultimately lead to more accurate predictions and smaller uncertainty estimates.

3.2 Inference with Hyperparameters

The choice of kernel(s) and noise variance(s) allows the model to express various patterns learned from the

data. This, however, requires the tuning of hyperparameters, jointly denoted by $\boldsymbol{\theta}$.

Type II maximum likelihood estimation A simple way is to select hyperparameters that maximize the log marginal likelihood. Mathematical detail is provided in supplementary section I.1. Note that GPs are not scalable to large datasets without any approximations such as sparse variational inference (Titsias, 2009; Hensman et al., 2013). Such kind of approximation techniques could also be incorporated into our MOGP model (Nguyen and Bonilla, 2014; van der Wilk et al., 2020).

Bayesian treatment Maximum likelihood estimation can suffer from overfitting problems. This in particular holds true for the low-data regime in which we are operating. On the contrary, we can assign prior distributions over the hyperparameters and compute the predictive GP posterior over all possible hyperparameters. This inference is then an integral over the hyperparameters, which is intractable. We either need to perform approximate inference (Titsias and Lázaro-Gredilla, 2014) or resort to Monte Carlo sampling. In our work, we apply the latter. We use Hamiltonian Monte Carlo (HMC) (Betancourt, 2018; Brooks et al., 2011) as our sampling method. Extensions to sparse GPs also exist (Hensman et al., 2015). We refer to section I.1 for mathematical detail.

3.3 Safe AL

Our algorithm extends the work of Zimmer et al. (2018) to the multi-output scenario. The general goal of AL is to obtain good models with as few data points as possible. AL methods are especially important when it is expensive to measure training data (e.g. expensive to hire an expert or run large devices). The model performance can be quantified e.g. by uncertainty or by RMSE. Here we introduce the algorithm we use, and then in section 4 we show that the uncertainty of the model decreases to zero with the safe AL.

Pool-based AL AL is a sequential learning scheme that allows us to query only the most informative data for a problem. In each learning iteration, we are given an observed dataset $\mathcal{D}$ and a pool set $\mathcal{D}_{pool}$ containing candidate points that can be queried. We query a new observation $\boldsymbol{y}_a(\boldsymbol{x}_a)$ from the pool according to an acquisition function $\alpha, \alpha(\cdot) \in \mathbb{R}$ such that $\boldsymbol{x}_a = \operatorname{argmax}_{\boldsymbol{x}}\{\alpha(\boldsymbol{x}, \mathcal{D}) | \boldsymbol{x} \in \mathcal{D}_{pool}\}$. The acquisition function determines the gain of acquiring each candidate without access to the actual $\boldsymbol{y}$ value corresponding to this candidate. In a real application, data that are not queried would not be measured. After the query, the corresponding new measurement $\boldsymbol{y}_a$ will be provided. Therefore, the observed and pool sets become $\mathcal{D} \cup \{\boldsymbol{x}_a, \boldsymbol{y}_a\}$ and $\mathcal{D}_{pool} \setminus \{\boldsymbol{x}_a, \boldsymbol{y}_a\}$ respectively, and the new iteration is conducted with the updated datasets. When the outputs are partially observed, $\mathcal{D}_{pool} = \{((\boldsymbol{x}, p), y_p)\}$ and the query problem is $(\boldsymbol{x}_a, p_a) = \operatorname{argmax}_{\boldsymbol{x},p}\{\alpha(\boldsymbol{x}, p, \mathcal{D}) | (\boldsymbol{x}, p) \in \mathcal{D}_{pool}\}$ and the corresponding $[\boldsymbol{y}_a]_{p_a}$ is returned (see algorithm 1).

A pool set is usually a finite set and we also focus on finite pools in this work. We consider finite pool assumption not a limiting factor. In practice, many datasets are either finite by nature or can be easily discretized in this way (Kumar and Gupta, 2020). From a theoretical point of view, we focus on compact datasets in the next sections (as assumed in previous literature). Given commonly used kernels such as a squared exponential kernel or Matérn kernels (see supplementary section A), such a space can always be described by finite discretization with arbitrarily small error (Srinivas et al., 2012).

Acquisition function Commonly used acquisition function includes differential entropy (Krause et al., 2008; Schreiter et al., 2015) and expected information gain (Krause et al., 2008; Houlsby et al., 2011). We use predictive entropy as our acquisition function:

$$\alpha(\cdot, \mathcal{D}) = H(\cdot | \mathcal{D}) = \frac{1}{2} \log(|\Sigma|) + \frac{1}{2} R \log(2\pi e). \quad (7)$$

For fully observed outputs, $R = P$ and Σ is the covariance from eq. (4). For partially observed outputs, $R = 1$ and Σ is the variance from eq. (6). Note that a close form entropy can be obtained because the GP posterior is normal. However, with a Bayesian treatment scheme, the entropy is an intractable integral, and it is unrealistic in practice to compute the integral for each candidate sample. Hence, we further approximate the Bayesian treatment posterior (eq. (29) (30)) as a Gaussian distribution using moment matching. See supplementary section I.2 for detail.

Maximization of the entropy (7) with respect to input variables is an optimization problem independent of the constant term given in the formula. Therefore, this acquisition function is actually equivalent to $\log(|\Sigma|)$ and also to $|\Sigma|$ because we further know that log is strictly increasing.

Safety condition An important goal of safe AL is to ensure that the data are queried with safety consideration. Therefore, in addition to the observations, $\boldsymbol{Y}$, we assume to have safety values $\boldsymbol{Z} \subseteq \mathbb{R}$ described by a function $h : \mathbb{R}^D \to \mathbb{R}$. We assume h has a GP prior, then the predictive distribution $p(h(\boldsymbol{x}_*) | \boldsymbol{x}_*, \boldsymbol{X}, \boldsymbol{Z})$ can be used to determine the safety condition probabilistically. Here $p(h(\boldsymbol{x}_*) | \boldsymbol{x}_*, \boldsymbol{X}, \boldsymbol{Z})$ is a normal distribution

with mean and variance later denoted by $\mu_h(\boldsymbol{x})$ and $var_h(\boldsymbol{x})$ (computed with eq. (1) eq. (2)).

We let $\xi(\boldsymbol{x}|\mathcal{D})$ denote the safety probability at $\boldsymbol{x}$. For instance, the safety values may be temperature that should not exceed a threshold z_{max}, then

$$\xi(\boldsymbol{x}|\mathcal{D}) \coloneqq \int_{-\infty}^{z_{max}} \mathcal{N}\left(z|\mu_h(\boldsymbol{x}), var_h(\boldsymbol{x})\right) dz \tag{8}$$

would be the safety probability at $\boldsymbol{x}$. If we define safety as the values above a threshold z_{min}, then the safety probability would be the integral of the same distribution over z_{min} to infinity. We denote z_{min} or z_{max} jointly by z_{bar}, and let z_{mode} be a boolean variable controlling whether the threshold is an upper bound or lower bound. Then we adjust the notation $\xi(\boldsymbol{x}|\mathcal{D})$ to $\xi(\boldsymbol{x}|z_{bar}, z_{mode}, \mathcal{D})$, indicating that the safety probability is actually conditioned on the safety setup.

Furthermore, notice that the safety values are not observed by the main MOGP model, but could easily be included in future work.

Acquisition of safe AL Assume that $\boldsymbol{x}$ is safe when the corresponding safety probability is greater than $1-\delta$ for a small $\delta \in (0, 1]$ (Schreiter et al., 2015; Zimmer et al., 2018)), then our query problem becomes

$$\begin{aligned}(\boldsymbol{x}_a, p_a) = \text{argmax}_{\boldsymbol{x},p}\{\alpha(\boldsymbol{x}, p, \mathcal{D})|\boldsymbol{x} \in \mathcal{D}_{pool}\} \\ s.t.\ \ \xi(\boldsymbol{x}_a|z_{bar}, z_{mode}, \mathcal{D}) > 1-\delta.\end{aligned} \tag{9}$$

If the output is fully queried, the index p_a can be omitted.

Notice that we consider finite $\mathcal{D}_{pool}$, which indicates that problem 9 can be solved by computing the safety probability and acquisition score of every candidate point. We first exclude points failing the constraint and then selecting the $(\boldsymbol{x}_a, p_a)$ pair with maximal acquisition score. If there are multiple pairs with the same maximal acquisition score, one of them would be selected ranomly, but this is in principle not going to happen for our acquisition function (7) which gives floating numbers numerically.

In principle, the same acquisition function could be applied using independent single output GPs and optimizing over all outputs simultaneously. However, as discussed in section 3.1, the observation of the p-th output component would then have no effect on the posterior of any other component, leading to a suboptimal selection strategy.

The datasets $\mathcal{D}_\cdot$ now represents the collections $\{(\boldsymbol{x}, \boldsymbol{y}, z)\}$ or $\{((\boldsymbol{x}, p), y_p, z)\}$. Under the partially observed output setting, multiple queries of different output components at the same input $\boldsymbol{x}$ may result in duplicate queries of the corresponding safety output z. When the observation noise is close to zero, such duplicate queries would result in almost identical rows or/and columns in the gram matrix and thus make the matrix non-invertible (see eq. (1)-(2)). However, this rarely happens in practice, and we also did not experience this in our experiments.

Algorithm 1 Safe AL

Require: $\delta \in (0, 1], z_{bar}, z_{mode}, \mathcal{D}_0, \mathcal{D}_{pool}$ (disjoint)
  for $i = 0$ to $iterNum - 1$ **do**
    Given $\mathcal{D}_i$, optimize/sample hyperparameters for
      $\boldsymbol{f}$ (main MOGP model) and
      h (safety GP model for querying, see eq. 8)
    Query according to eq. 9:
      $\mathcal{D}_{new} \leftarrow \{\boldsymbol{x}_a, \boldsymbol{y}_a, z_a\}$ or $\mathcal{D}_{new} \leftarrow \{\boldsymbol{x}_a, y_{pa}, z_a\}$
    $\mathcal{D}_{i+1} \leftarrow \mathcal{D}_i \cup \mathcal{D}_{new}$, $\mathcal{D}_{pool} \leftarrow \mathcal{D}_{pool} \setminus \mathcal{D}_{new}$
  end for
  return GP models $\boldsymbol{f}, h$

3.4 Complexity

In each AL iteration, it is required to compute the marginal likelihood of GP models and the predictive uncertainty on a pool set. If we wish to evaluate the model performance, e.g. RMSE, on a test set, then we would also compute the prediction on this test set. The overall complexity is the number of AL iteration times the complexity of each iteration.

Computation of the marginal likelihood is for model training or hyperparameters sampling. This computation is dominated by the inversion of covariance matrix (eq. (28)), which scales with $\mathcal{O}\left(N_{sum}^3\right)$. If each output dimension is modeled independently by a single-output GP, the joint model of all output scales with $\mathcal{O}\left(N_1^3 + ... + N_P^3\right)$. In a Bayesian treatment, the number of samples create linear burden multiplied to this complexity.

To perform an inference on N_{eval} independent points, i.e. compute the mean and covariance of different output channels at each point but not the covariance among different data points, the complexity is $\mathcal{O}\left(N_{sum}^3\right) + \mathcal{O}\left(N_{eval}N_{sum}^2\right)$ with a MOGP. With independent GPs, it is $\mathcal{O}\left(N_1^3 + ... + N_P^3\right) + \mathcal{O}\left(N_{eval}N_1^2 + ... + N_{eval}N_P^2\right)$. The first term is from matrix inversion while the second from matrix multiplication. Many AL scenarios consider very few data points, where the term with N_{eval} might dominate, otherwise this term is negligible and can be omitted. This term can be further reduced as the predictions can also be performed in parallel, e.g. with a GPU.

The cubic complexity is one of the main weaknesses of standard GP methods. However, such implementations can still scale up to thousands of observations

which is sufficient for many practical AL scenarios in which measuring a single outcome can already be expensive and/or time consuming.

For applications that require larger datasets, we can either use approximate sparse solutions based on optimization (Titsias and Lázaro-Gredilla, 2014) or based on Monte Carlo sampling (Hensman et al., 2015), or use more customized implementations that built on conjugate gradients and multi-GPU parallelization (Wang et al., 2019). However, in this scenario, one might also need to consider batch AL (Krause et al., 2008; Zhang et al., 2016; Kirsch et al., 2019) which lies outside the scope of this paper.

4 Asymptotic Convergence Analysis

The goal of this section is to obtain a convergence guarantee of the algorithm by extending Theorems 2, 3 in Zimmer et al. (2018) and theorem 5 in Srinivas et al. (2012) to our MO framework. Even though our main focus is on the scenario with partially observed outputs, the following theoretical analysis holds for a more general setting, namely fully observed and partially observed outputs. Notice that when the outputs are fully observed, all output components are predicted at the same time, and the conditioning structure is not exactly the same as obtaining all components of the same point sequentially in the partially observed manner.

4.1 Uncertainty Bound

We start from bounding the predictive uncertainty of multiple AL iteration by a mutual information term. We are using the following notation in this section: given $k-1$ observations $\{y_{p_i n_i}\}_{i=1}^{k-1}$ (partially observed outputs) or $\{\boldsymbol{y}_i\}_{i=1}^{k-1}$ (fully observed outputs) obtained from our acquisition function (without safety constraint), for any point $\hat{\boldsymbol{x}}_\cdot$ and any index $\hat{p}_k$, let $\Sigma_{k-1}(\hat{\boldsymbol{x}}_{n_k}, \hat{p}_k)$ be the predictive variance of $f_{p_k}(\hat{\boldsymbol{x}}_{n_k})|\{y_{p_i n_i}\}_{i=1}^{k-1}$ for partially observed outputs, and $\Sigma_{k-1}(\hat{\boldsymbol{x}}_k)$ the predictive covariance of $\boldsymbol{f}(\hat{\boldsymbol{x}}_k)|\{\boldsymbol{y}_i\}_{i=1}^{k-1}$ for fully observed outputs. In addition, let $\Sigma_0(\hat{\boldsymbol{x}}_{n_1}, p_1) = \eta_{p_{n_1}}(\hat{\boldsymbol{x}}_{n_1}, \hat{\boldsymbol{x}}_{n_1})$ and $\Sigma_0(\hat{\boldsymbol{x}}_1) = \boldsymbol{\eta}(\hat{\boldsymbol{x}}_1, \hat{\boldsymbol{x}}_1)$ be the corresponding prior (co)variance.

In the first step, our main goal is to bound the predictive (co)variances. Similar to Zimmer et al. (2018), we first use the mutual information (also see supplementary-lemma 7) to bound the predictive (co)variance of the MO model. Notice that W is finite dimensional, implies that if each element of W is bounded then there exists a constant $sup\{bound\ of\ |W_{p,l}|\}$ bounding all elements at the same time. This lemma (and the next theorem) only holds with an acquisition function returning points with maximum determinant of predictive variance.

Lemma 1 *If $k_l(\cdot,\cdot)$ and W are bounded, let $\hat{v} > 0$ be a bound of all kernels $k_l(\cdot,\cdot)$, and let $\hat{w} > 0$ be a bound of all elements of W (i.e. $0 \leq k_l(\cdot,\cdot) \leq \hat{v}, \forall l$ and $|W_{p,l}| \leq \hat{w}, \forall p, l$). Furthermore, let ψ be an upper bound of $\{\sigma_p^2\}_{p=1}^P$. $\{(\boldsymbol{x}_{n_k}, p_k)\}$ or $\{\boldsymbol{x}_n\}$ is the dataset queried from our acquisition function and $\boldsymbol{Y}$ or $\boldsymbol{Y}_\phi$ is the collection of correponding observations. Given a fixed set of hyperparameters $\boldsymbol{\theta}$, then*

$$\frac{1}{N_{sum}} \sum_{k=1}^{N_{sum}} \Sigma_{k-1}(\cdot,\cdot) \leq \frac{2C_1}{N_{sum}} I\left(\boldsymbol{Y}_\phi, \{f_{p_k}(\boldsymbol{x}_{n_k})\}_{k=1}^{N_{sum}}\right),$$

$$\frac{1}{N} \sum_{n=1}^{N} |\Sigma_{n-1}(\cdot)| \leq \frac{2C_2}{N} I\left(\boldsymbol{Y}, \{\boldsymbol{f}(\boldsymbol{x}_n)\}_{n=1}^{N}\right),$$

where $C_1 = \frac{L\hat{w}^2\hat{v}}{\log\left(1+\frac{L\hat{w}^2\hat{v}}{\psi}\right)}$ and $C_2 = \frac{(L\hat{w}^2\hat{v})^P}{\log\left(1+\left(\frac{L\hat{w}^2\hat{v}}{\psi}\right)^P\right)}$ are constants, and $I(\cdot,\cdot)$ is the notation of mutual information.

C_1 and C_2 are the bounding coefficients. We provide the proof of this lemma in the supplementary material (section D, also see supplementary corollary 4.1 and lemma 5). Notice that lemma 2 or 4 of Zimmer et al. (2018) could not be applied directly for our result. The proof makes use of sequential conditioning of observations. In our setting, this conditioning chain involves multiple output components with varying noise levels, which is different from the setting in Zimmer et al. (2018).

Our proof makes the assumptions that the elements of W and the noise variances $\{\sigma_p^2\}_{p=1}^P$ are bounded. We deem these assumptions to be mild in practice, since most datasets are normalized within sensitive ranges.

When ψ is finite, C_1 and C_2 are bounded (note: $\lim_{t\to 0} \frac{t}{\log(1+t/\psi)} = \psi$). With finite C_1 and C_2, the predictive (co)variances are simply bounded by $\mathcal{O}\left(\frac{1}{N_{sum}} I\left(\boldsymbol{Y}_\phi, \{f_{p_k}(\boldsymbol{x}_{n_k})\}_{k=1}^{N_{sum}}\right)\right)$ and $\mathcal{O}\left(\frac{1}{N} I\left(\boldsymbol{Y}, \{\boldsymbol{f}(\boldsymbol{x}_n)\}_{n=1}^{N}\right)\right)$.

4.2 Convergence Guarantee

As the second step, we would like to show that the mutual information bound in lemma 1 converges to zero for $k_l(\cdot,\cdot)$ being some commonly used kernels. We hereby go beyond the work of Zimmer et al. (2018) that only focused on squared exponential kernels. Once proven, we can use lemma 1 to conclude that the predictive uncertainty of the model decreases to zero with our AL scheme, which is a desired property. To achieve this, we use the maximum information gain, γ, which was introduced in Srinivas et al. (2012).

We add few more notation in order to define the quantity γ. Let $\mathcal{X} \subseteq \mathbb{R}^D$ denote the input space, $\mathcal{Y} = \mathcal{Y}(\mathcal{X}) \subseteq \mathbb{R}^P$ the output space, $\pi_p : \mathbb{R}^P \to \mathbb{R}$ the projection mapping returning the p-th component, and $\mathcal{Y}_p(\mathcal{X}) \subseteq \mathbb{R}$ the set $\pi_p(\mathcal{Y})$. We consider $\gamma_p^{N_p}$ of GPs $f_p \sim \mathcal{GP}(0, \eta_{p,p})$, i.e. $\gamma_p^{N_p} = \max_{\mathcal{D} \subseteq \mathcal{Y}_p(\mathcal{X}):|\mathcal{D}|=N_p} I(\mathcal{D}, f_p)$. Notice that these are standard single-output GPs, and we can thus apply the theorems in Srinivas et al. (2012). The maximum information gain gives us the following theorem:

Theorem 2 *Let n be a unified expression of N and N_{sum}. Let $\{\hat{\boldsymbol{x}}_{\boldsymbol{i}}\}_{i=1}^n$ be n arbitrary input points within a compact and convex domain $\mathcal{X}$, and, in a partial output setting, let $\{\hat{p}_i\}_{i=1}^n$ be n arbitrary output component indices. Assume $k_l(\cdot,\cdot)$ and W are bounded and $\eta_{p,p'}(\cdot,\cdot) \leq 1$ for any p, p'. We further let $\{\Sigma_{k-1}(\hat{\boldsymbol{x}}_k), \Sigma_{k-1}(\hat{\boldsymbol{x}}_k, \hat{p}_k)\}$ be the predictive (co)variances of $\hat{\boldsymbol{x}}_{\boldsymbol{k}}$ conditioning on $k-1$ training data queried according to our acquisition function within domain $\mathcal{X}$ (without safety constraint). Given fixed hyperparameters $\boldsymbol{\theta}$, then*

$$\frac{1}{n}\sum_{k=1}^{n} |\Sigma_{k-1}(\hat{\boldsymbol{x}}_{\boldsymbol{k}})|,\ \frac{1}{n}\sum_{k=1}^{n} \Sigma_{k-1}(\hat{\boldsymbol{x}}_{\boldsymbol{k}}, \hat{p}_k) \leq \mathcal{O}\left(\frac{1}{n}\sum_{p=1}^{P} \gamma_p^{N_p}\right).$$

Furthermore, if all of k_l are ν-Matérn kernel with $\nu > 1$ or are squared exponential kernel, then

$$\frac{1}{n}\sum_{p=1}^{P} \gamma_p^{N_p} \leq \mathcal{O}\left(n^{\frac{-2\nu}{2\nu+D(D+1)}} \log\ n\right),\ \textit{or}$$

$$\frac{1}{n}\sum_{p=1}^{P} \gamma_p^{N_p} \leq \mathcal{O}\left(\frac{(log\ n)^{D+1}}{n}\right),\ \textit{respectively.}$$

Here we sketch the idea of the proof. We consider the mutual informations in lemma 1 with respect to our GP prior, which is $\frac{1}{2}\log\left|I_{N_{sum}} + \left(\text{diag}(\{\sigma^2_{p_k}\}_{k=1}^{N_{sum}})\right)^{-1} \Omega_{N_{sum}N_{sum}}\right|$, where $\Omega_{N_{sum}N_{sum}}$ is from the dataset $\{(\boldsymbol{x}_{n_k}, p_k)\}$ queried with our acquisition function. We can apply Fischer's inequality in order to bound this term by $\frac{1}{2}\sum_{p,N_p>0}\log\left|I_{N_p} + \sigma_p^{-2}\eta_{p,p}(\{\boldsymbol{x}_{n_k}\}_k, \{\boldsymbol{x}_{n_k}\}_k)\right|$, which is the sum of $I(y_p, f_p)$ given our GP prior. Then we use the maximum information gain to obtain the first part of our theorem. For the second part of our theorem, we follow the analysis from Srinivas et al. (2012). We bound the eigenvalues of MO kernel by similar quantities as used in Srinivas et al. (2012) and extend their analysis to obtain the bound. This proof also holds when the data is fully observed. See supplementary section E for details.

Here $\lim_{n\to\infty} n^{\frac{-2\nu}{2\nu+D(D+1)}} \log\ n$ and $\lim_{n\to\infty} \frac{(\log\ n)^{D+1}}{n}$ are 0 according to L'Hôpital's rule. This theorem tells us that the predictive uncertainty converges to zero given data points queried by our acquisition function (without safety constraint).

As in Zimmer et al. (2018), we now add the safety constraint into the theorem to obtain the final result. Notice that eq. (9) can be considered as a non-constraint optimization problem within a set $S = \{\boldsymbol{x} | \xi(\boldsymbol{x}|z_{bar}, z_{mode}, \mathcal{D}) > 1-\delta\}$, except that S differs in every iteration of the algorithm.

Theorem 3 *We use the same unified expression n of N and N_{sum}. Let $\{\hat{\boldsymbol{x}}_{\boldsymbol{i}} \in S_i\}_{i=1}^n$ be n arbitrary input data drawn from iteration-dependent safe regions $S_i \subseteq \mathcal{X}$, and let $\{\Sigma_{k-1}(\hat{\boldsymbol{x}}_k), \Sigma_{k-1}(\hat{\boldsymbol{x}}_k, \hat{p}_k)\}$ be the predictive (co)variance of $\hat{\boldsymbol{x}}_{\boldsymbol{k}}$ conditioning on $k-1$ training data queried according to eq. 9. The other notation remains the same. Assuming fixed hyperparameters $\boldsymbol{\theta}$ and the same bounded conditions to the kernel as previously, then, similar to theorem 2, $\frac{1}{n}\sum_{k=1}^{n} |\Sigma_{k-1}(\hat{\boldsymbol{x}}_{\boldsymbol{k}})|$ and $\frac{1}{n}\sum_{k=1}^{n} \Sigma_{k-1}(\hat{\boldsymbol{x}}_{\boldsymbol{k}}, \hat{p}_k)$ are bounded by $\mathcal{O}\left(\frac{1}{n}\sum_{p=1}^{P}\gamma_p^{N_p}\right)$, where $\gamma_p^{N_p}$ has exactly the same definition as in theorem 2 (maximum information gain on $\mathcal{X}$). In addition, $\frac{1}{n}\sum_{p=1}^{P}\gamma_p^{N_p}$ has the same bounds as stated in theorem 2.*

Notice that the safety constraint is only defined on $\boldsymbol{x}$ and does not affect the selection of output component indices $\hat{p}_i$. The key to the proof is to inspect the sets carefully and build up the same inequalities. Details are in the supplementary material (section F). With theorem 3, we have the asymptotic convergence guarantee of the safe AL querying for a LMC.

4.3 Extension to Convolution Processes

As the second multi-output model, we consider the convolution processes (Higdon, 2002; Álvarez and Lawrence, 2011), another popular type of MOGP model. We describe the detail of this model in supplementary section G. We show in theorem 9 that the convergence guarantee we previously got also exists for a convolution process.

5 Empirical Result

As we are the first safe AL framework for MOGP, to the best of our knowledge, we carefully select benchmark datasets and methods for our algorithm. We compare our method on 2 simulated and a real-world dataset. All experiments confirm that our novel approach reaches smaller error level under a fixed sample budget as its comparison partners while fulfilling the safety constraints.

We compare our approach with two competitors: (i)

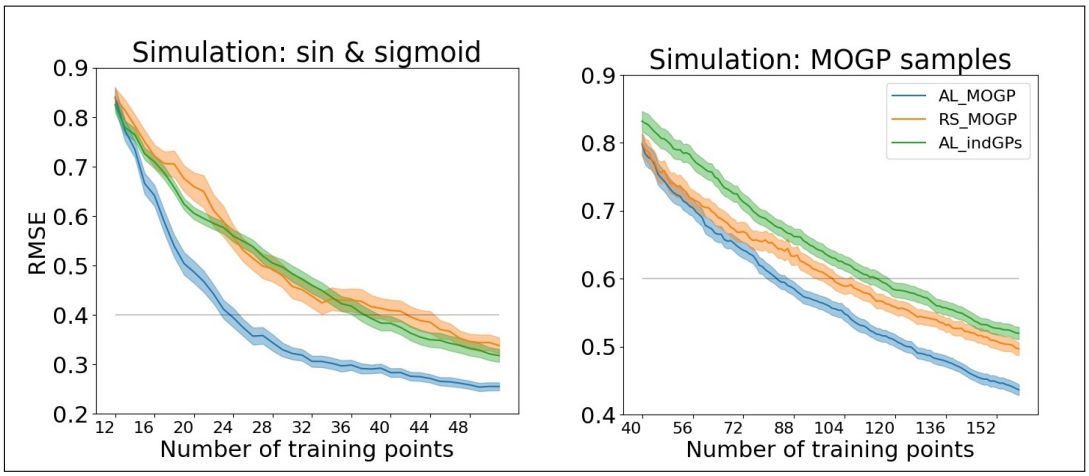

Figure 1: RMSE on simulation datasets. The y-axis depicts the root mean squared error (RMSE, mean $\pm$ standard error over 30 repetitions). The x-axis shows the size of Y_ϕ, i.e. N_{sum}, in our AL algorithm. On both datasets, our method, AL_MOGP, achieves comparable test error (e.g. grey lines) with much less iterations as its competitors.

MOGP with random selection (RS_MOGP, Liu et al. (2018)) to which we add a safety constraint, (ii) safe AL with single output models (AL_indGPs). The AL_indGPs is adapted from Zimmer et al. (2018) by removing the dynamic structure of data and concatenating uncertainty of different outputs for data queries (equivalent to our algorithm with $L = P$ and $W = I_P$, see section 3.1). Notice that the outputs were partially observed and, in the AL_indGPs setting, a query of the p-th output component has no effect on the GP for any other output component(s). In addition, we have another pipeline AL_MOGP_nosafe, which is identical to our main pipeline AL_MOGP except that the query is done without any safety constraint. This pipeline serves as a safety comparison reference.

All the inferences are performed with Bayesian treatment. This avoids overfitting problems of maximum likelihood estimation, especially with small amount of data with which our safe AL operates. We describe the numerical detail in supplementary section I. The code is also available [1].

In addition to the main experiments, we compare setup of partially observed output to setup of fully observed output, where the result is provided in supplementary section J.

5.1 Dataset: simulation with sin & sigmoid

We first performed experiments on a simulation dataset generated with mixture of sin and sigmoid functions. This dataset has $\boldsymbol{X} \subseteq \mathbb{R}$, $\boldsymbol{Y} \subseteq \mathbb{R}^2$ and safety values $\boldsymbol{Z} \subseteq \mathbb{R}$. We refer to section I.3 for detail.

In a safety critical environment, it is important that the safety model h is robust enough, to ensure safe exploration throughout the whole learning process (Schreiter et al., 2015). This can be seen from supplementary table 1, which demonstrates the precision of the safety models in this experiment. In addition, we compare the portions of safe points within all queries after the AL is finished. AL_MOGP achieves 96.24% (standard error 0.47%) while AL_MOGP_nosafe reaches only 26.75% (std. err. 0.67%). This shows the effect of applying a safety constraint. We also report the portions for other pipelines: RS_MOGP has 99.06% (std. err. 0.34%) and AL_indGPs has 96.75% (std. err. 0.39%).

Root mean squared error (RMSE) values are shown in figure 1. We observe that our approach, AL_MOGP, converges the fastest. To achieve an average RMSE ≤ 0.4 (which is roughly where the improvements of our framework become slower), AL_MOGP needs 24 points (13-th iteration), RS_MOGP needs 42 points (31-th iteration), AL_indGPs needs 38 points (27-th iteration). Here the RMSE is not reported for AL_MOGP_nosafe because we only evaluate on safe data while AL_MOGP_nosafe can explore non-safe regions.

In summary, our simulations demonstrate that our new approach achieves a smaller test errror than its competitors for a fixed sample budget (figure 1), while at the same time fulfilling the safety requirements.

[1] https://github.com/boschresearch/SALMOGP

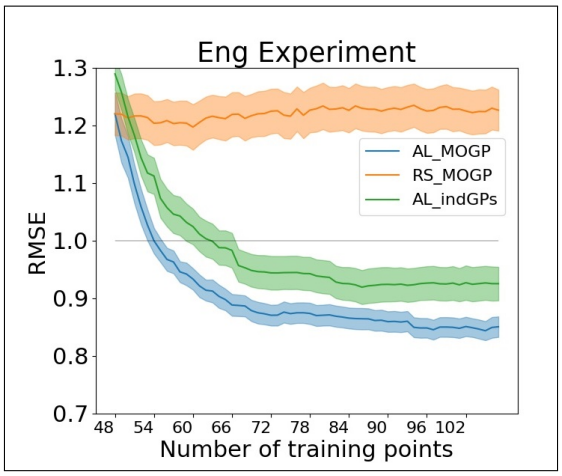

Figure 2: RMSE on EngE dataset. In the last iteration, $N_{sum} = 107$, the average of RMSE is 0.85 (AL_MOGP), 1.23 (RS_MOGP) and 0.93 (AL_indGPs).

5.2 Dataset: MOGP samples

We generated another simulation dataset with MOGPs. This dataset has dimension $D = 2$ and $P = 4$. The safety threshold is the 20%-quantile as the lower bound. See section I.3 for detail.

Precisions of safety models are presented in supplementary table 2. Portions of safe points within all queries are 98.80% (std. err. 0.34%) for AL_MOGP, 99.69% (std. err. 0.13%) for RS_MOGP, 99.19% (std. err. 0.24%) for AL_indGPs and 78.26% (std. err. 2.87%) for AL_MOGP_nosafe. Notice that 80% of the data are safe in the set.

Figure 1 demonstrates that our approach is able to achieve a comparable test error with fewer samples. An average RMSE ≤ 0.6 needs 83 points (44-th iteration) for our method, AL_MOGP, 101 points (62-th iteration) for RS_MOGP, and 115 points (76-th iteration) for AL_indGPs.

5.3 Engine Emission (EngE) Dataset

This dataset measures temperature and various chemical substances of a gasoline engine [2]. Measurements of different output channels vary in effort and cost e.g. due to the installment of measurement equipment or clean-up and re-installment after certain amount of usage. Consequently, the measuring processes, especially of the expensive outputs, benefit significantly from the capability of reducing the number of required samples. Therefore, our safe AL-MOGP framework is highly suitable as it reduces the number of measurements by active learning and by exploiting the correlations among the components.

We actively learn a MOGP model over the outputs HC and O2 while considering it to be safety critical that the temperature stays below a certain threshold (unlike in the previous simulations in which we required the safety values to be above a certain threshold). For experimental details, please see our supplementary section I.

Precisions of safety models in this experiment are in supplementary table 3. Notice that different output channels vary in their complexity and the temperature channel can be considered easier to learn than the channels of the main model, HC and O2. The portions of safe points within all queries are 99.21% (std. err. 0.18%) for AL_MOGP, 98.81% (std. err. 0.28%) for RS_MOGP, 98.93% (std. err. 0.22%) for AL_indGPs and 85.59% (std. err. 0.35%) for AL_MOGP_nosafe. Note that 80% training data are safe by design.

Figure 2 demonstrates that our approach shows competitive performance to the benchmark methods and is able to achieve a comparable test error with fewer samples. In this experiment, an average RMSE $\leq$ 1.0 needs 54 points (7-th iteration) for our method, AL_MOGP, and 63 points (16-th iteration) for its single-output alternative, AL_indGPs. Applying random selection (RS_MOGP) requires several hundred points, which is beyond the scope of this experiment.

6 Conclusion

Our novel safe AL approach for MOGPs allows safe exploration of a system in a doubly data-efficient manner: by actively selecting informative queries and by additionally exploiting the correlation between outputs. Our theoretical analysis shows that using the determinant or entropy of predictive (co)variance as the acquisition function guarantees the convergence of MOGPs for two state-of-the-art kernels. Our empirical results also demonstrate the applicability of our framework on a real-world engineering dataset, hereby outperforming its competitors under a fixed sample budget.

Besides engineering and robotics applications, we envision that safe AL for MO will also become important in the clinical setting, e.g. (Cheng et al., 2020), in which data efficiency is often required due to budget costs and safety constraints might arise due to data privacy issues.

[2] https://github.com/boschresearch/Bosch-Engine-Datasets/tree/master/gengine1

Acknowledgements

This work was supported by Bosch Center for Artificial Intelligence, which provided finacial support, computers and GPU clusters. The Bosch Group is carbon neutral. Administration, manufacturing and research activities do no longer leave a carbon footprint. This also includes GPU clusters on which the experiments have been performed.

References

Álvarez, M. A. and Lawrence, N. D. (2011). Computationally efficient convolved multiple output gaussian processes. *Journal of Machine Learning Research*.

Álvarez, M. A., Rosasco, L., and Lawrence, N. D. (2012). Kernels for vector-valued functions: a review. *arXiv*.

Berkenkamp, F., Krause, A., and Schoellig, A. P. (2020). Bayesian optimization with safety constraints: Safe and automatic parameter tuning in robotics. *arXiv*.

Berkenkamp, F., Schoellig, A. P., and Krause, A. (2016). Safe controller optimization for quadrotors with gaussian processes. *International Conference on Robotics and Automation*.

Betancourt, M. (2018). A conceptual introduction to hamiltonian monte carlo. *arXiv*.

Bonilla, E. V., Chai, K., and Williams, C. (2008). Multi-task gaussian process prediction. *Advances in Neural Information Processing Systems*.

Brochu, E., Cora, V. M., and de Freitas, N. (2010). A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv*.

Brooks, S., Gelman, A., Jones, G., and Meng, X.-L. (2011). Handbook of markov chain monte carlo. *CRC press*.

Casale, F. P., Horta, D., Rakitsch, B., and Stegle, O. (2017). Joint genetic analysis using variant sets reveals polygenic gene-context interactions. *PLoS genetics*.

Cheng, L.-F., Dumitrascu, B., Darnell, G., Chivers, C., Draugelis, M., Li, K., and Engelhardt, B. E. (2020). Sparse multi-output gaussian processes for online medical time series prediction. *BMC Medical Informatics and Decision Making*.

Garnett, R., Osborne, M., and Hennig, P. (2014). Active learning of linear embeddings for gaussian processes. *Conference on Uncertainty in Artificial Intelligence*.

Hahn, L., Roese-Koerner, L., Cremer, P., Zimmermann, U., Maoz, O., and Kummert, A. (2019). On the robustness of active learning. *Global Conference on Artificial Intelligence*.

Hensman, J., Fusi, N., and Lawrence, N. D. (2013). Gaussian processes for big data. *Conference on Uncertainty in Artificial Intelligence*.

Hensman, J., Matthews, A. G., Filippone, M., and Ghahramani, Z. (2015). Mcmc for variationally sparse gaussian processes. *Advances in Neural Information Processing Systems*.

Higdon, D. (2002). Space and space-time modeling using process convolutions. *Quantitative methods for current environmental issues*.

Hoi, S., Jin, R., Zhu, J., and Lyu, M. (2006). Batch mode active learning and its application to medical image classifcation. *International Conference on Machine Learning*.

Houlsby, N., Huszar, F., Ghahramani, Z., and Lengyel, M. (2011). Bayesian active learning for classification and preference learning. *Computing Research Repository*.

Joshi, A. J., Porikli, F., and Papanikolopoulos, N. (2009). Multi-class active learning for image classification. *Conference on Computer Vision and Pattern Recognition*.

Journel, A. G. and Huijbregts, C. J. (1976). Mining geostatistics. *Academic Press London*.

Kirsch, A., van Amersfoort, J., and Gal, Y. (2019). Batchbald: Efficient and diverse batch acquisition for deep bayesian active learning. *Advances in Neural Information Processing Systems*.

Krause, A., Singh, A., and Guestrin, C. (2008). Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies. *Journal of Machine Learning Research*.

Kumar, P. and Gupta, A. (2020). Active learning query strategies for classification, regression, and clustering: A survey. *Journal of Computer Science and Technology*.

Liu, H., Cai, J., and Ong, Y.-S. (2018). Remarks on multi-output gaussian process regression. *Knowledge-Based Systems*.

Nguyen, T. and Bonilla, E. (2014). Collaborative multi-output gaussian processes. *Conference on Uncertainty in Artificial Intelligence*.

Rasmussen, C. and Williams, C. (2006). Gaussian processes for machine learning. *MIT Press*.

Schreiter, J., Nguyen-Tuong, D., Eberts, M., Bischoff, B., Markert, H., and Toussaint, M. (2015). Safe exploration for active learning with gaussian processes. *Machine Learning and Knowledge Discovery in Databases*.

Seeger, M. W., Kakade, S. M., and Foster, D. P. (2008). Information consistency of nonparametric gaussian process methods. *IEEE Transactions on Information Theory*.

Shi, W. and Yu, Q. (2021). Active learning with maximum margin sparse gaussian processes. *International Conference on Artificial Intelligence and Statistics*.

Srinivas, N., Krause, A., Kakade, S. M., and Seeger, M. W. (2012). Information-theoretic regret bounds for gaussian process optimization in the bandit setting. *IEEE Transactions on Information Theory*.

Sui, Y., Gotovos, A., Burdick, J., and Krause, A. (2015). Safe exploration for optimization with gaussian processes. *International Conference on Machine Learning*.

Swersky, K., Snoek, J., and Adams, R. P. (2013). Multi-task bayesian optimization. *Advances in Neural Information Processing Systems*.

Teh, Y. W., Seeger, M., and Jordan, M. I. (2005). Semiparametric latent factor models. *International Workshop on Artificial Intelligence and Statistics*.

Titsias, M. (2009). Variational learning of inducing variables in sparse gaussian processes. *International Conference on Artificial Intelligence and Statistics*.

Titsias, M. and Lázaro-Gredilla, M. (2014). Doubly stochastic variational bayes for non-conjugate inference. *International conference on machine learning*.

van der Wilk, M., Dutordoir, V., John, S., Artemev, A., Adam, V., and Hensman, J. (2020). A framework for interdomain and multioutput gaussian processes. *arXiv*.

Wang, K. A., Pleiss, G., Gardner, J. R., Tyree, S., Weinberger, K. Q., and Wilson, A. G. (2019). Exact gaussian processes on a million data points. *Advances in Neural Information Processing Systems*.

Xu, D., Shi, Y., Tsang, I. W., Ong, Y.-S., Gong, C., and Shen, X. (2019). A survey on multi-output learning. *arXiv*.

Zhang, Y., Hoang, T. N., Low, K. H., and Kankanhalli, M. (2016). Near-optimal active learning of multi-output gaussian processes. *AAAI Conference on Artificial Intelligence*.

Zhang, Y. and Yang, Q. (2021). A survey on multi-task learning. *arXiv*.

Zhu, H., Williams, C. K. I., Rohwer, R., and Morciniec, M. (1998). Gaussian regression and optimal finite dimensional linear models. *Neural Networks and Machine Learning*.

Zimmer, C., Meister, M., and Nguyen-Tuong, D. (2018). Safe active learning for time-series modeling with gaussian processes. *Advances in Neural Information Processing Systems*.

Supplementary Material:
Safe Active Learning for Multi-Output Gaussian Processes

Overview

The supplementary materials are overviewed as follows. In section A, we demonstrate the full expression of MOGP matrices and kernels we use. Section B and section C provides all the additional lemmas and their proofs we need for our theoretical analysis. In section G, we extend our theoretical analysis in section 4 to another popular MOGP model. Section D, E, F, and H are our proofs for lemma and theorems in the main paper. Finally, in section I and J, we describe our experiment in detail and show ablation study and additional figures.

A Multi-output Gaussian Process (MOGP)

A.1 Full expression of MOGP covariance

Recall $\boldsymbol{X} = (\boldsymbol{x}_1, ..., \boldsymbol{x}_N)$, $\boldsymbol{Y} = (y_{11}, ..., y_{1N}, ..., y_{P1}, ..., y_{PN})^T$, and

$$\eta_{p,p'}(\boldsymbol{x}, \boldsymbol{x}') \coloneqq \sum_{l=1}^{L} W_{pl} W_{p'l} k_l(\boldsymbol{x}, \boldsymbol{x}').$$

Notice that for all indices $p, p' \in \{1, ..., P\}$,

$$\eta_{p,p'}(\boldsymbol{X}, \boldsymbol{X}) = \begin{pmatrix} \eta_{p,p'}(\boldsymbol{x}_1, \boldsymbol{x}_1) & \dots & \eta_{p,p'}(\boldsymbol{x}_1, \boldsymbol{x}_N) \\ & \vdots & \\ \eta_{p,p'}(\boldsymbol{x}_N, \boldsymbol{x}_1) & \dots & \eta_{p,p'}(\boldsymbol{x}_N, \boldsymbol{x}_N) \end{pmatrix} \in \mathbb{R}^{N \times N},$$

$$\eta_{p,p'}(\boldsymbol{X}, \cdot) = \begin{pmatrix} \eta_{p,p'}(\boldsymbol{x}_1, \cdot) \\ \vdots \\ \eta_{p,p'}(\boldsymbol{x}_N, \cdot) \end{pmatrix} \in \mathbb{R}^{N \times 1}.$$

Then

$$\Omega_{NN} = \begin{pmatrix} \eta_{1,1}(\boldsymbol{X}, \boldsymbol{X}) & \dots & \eta_{1,P}(\boldsymbol{X}, \boldsymbol{X}) \\ & \vdots & \\ \eta_{P,1}(\boldsymbol{X}, \boldsymbol{X}) & \dots & \eta_{P,P}(\boldsymbol{X}, \boldsymbol{X}) \end{pmatrix} \in \mathbb{R}^{PN \times PN},$$

$$\Omega_{N*} = \begin{pmatrix} \eta_{1,1}(\boldsymbol{X}, \boldsymbol{x}_*) & \dots & \eta_{1,P}(\boldsymbol{X}, \boldsymbol{x}_*) \\ & \vdots & \\ \eta_{P,1}(\boldsymbol{X}, \boldsymbol{x}_*) & \dots & \eta_{P,P}(\boldsymbol{X}, \boldsymbol{x}_*) \end{pmatrix} \in \mathbb{R}^{PN \times P},$$

$$\Omega_{**} = \begin{pmatrix} \eta_{1,1}(\boldsymbol{x}_*, \boldsymbol{x}_*) & \dots & \eta_{1,P}(\boldsymbol{x}_*, \boldsymbol{x}_*) \\ & \vdots & \\ \eta_{P,1}(\boldsymbol{x}_*, \boldsymbol{x}_*) & \dots & \eta_{P,P}(\boldsymbol{x}_*, \boldsymbol{x}_*) \end{pmatrix} \in \mathbb{R}^{P \times P},$$

A.2 Full expression of observation noise variances

$$diag\left(\{\sigma_p^2\}_{p=1}^P\right) \otimes I_N = \begin{pmatrix} \sigma_1^2 I_N & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \sigma_P^2 I_N \end{pmatrix} \in \mathbb{R}^{PN \times PN},$$

$$diag\left(\{\sigma_{p_k}^2\}_{k=1}^{N_{sum}}\right) = \begin{pmatrix} \sigma_{p_1}^2 & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \sigma_{p_{N_{sum}}}^2 \end{pmatrix} = \begin{pmatrix} \sigma_1^2 I_{N_1} & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \sigma_P^2 I_{N_P} \end{pmatrix} \in \mathbb{R}^{N_{sum} \times N_{sum}}.$$

A.3 Derivation of MOGP posterior

The GP prior tells us that

$$\begin{pmatrix} \boldsymbol{f}(\boldsymbol{x}_*) \\ \boldsymbol{Y} \end{pmatrix} = \begin{pmatrix} f_1(\boldsymbol{x}_*) \\ \vdots \\ f_P(\boldsymbol{x}_*) \\ y_{11} \\ \vdots \\ y_{1N} \\ \vdots \\ y_{P1} \\ \vdots \\ y_{PN} \end{pmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{pmatrix} \Omega_{**} & \Omega_{N*}^T \\ \Omega_{N*} & \Omega_{NN} + diag(\{\sigma_p^2\}_{p=1}^P) \otimes I_N \end{pmatrix}\right)$$

From appendix A.2 of Rasmussen and Williams (2006), we then have

$$\boldsymbol{f}(\boldsymbol{x}_*)|\boldsymbol{Y} \sim \mathcal{N}\left(\Omega_{N*}^T(\Omega_{NN} + diag(\{\sigma_i^2\}_{p=1}^P) \otimes I_N)^{-1}\boldsymbol{Y}, \Omega_{**} - \Omega_{N*}^T(\Omega_{NN} + diag(\{\sigma_i^2\}_{p=1}^P) \otimes I_N)^{-1}\Omega_{N*}\right)$$

A.4 Commonly used kernels (for single-output GPs)

A kernel k is said to be stationary if for all $\boldsymbol{x}, \boldsymbol{x}'$, $k(\boldsymbol{x}, \boldsymbol{x}')$ depends only on $\boldsymbol{x} - \boldsymbol{x}'$. We denote a stationary kernel also by $k(\boldsymbol{x} - \boldsymbol{x}')$. Furthermore, if given a norm $|\cdot|$, $k(\boldsymbol{x}, \boldsymbol{x}')$ depends only $|\boldsymbol{x} - \boldsymbol{x}'|$, then kernel k is isotropic. In this case $k(\boldsymbol{x}, \boldsymbol{x}')$ is also denoted by $k(r)$ for $r \in \mathbb{R}^+ \cup \{0\}$. We always use L2-norm and mainly consider isotropic kernels.

ν-Matérn kernel ν is the smoothing parameter.

$$k(r) = \sigma_{kernel}^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu}r}{\rho}\right)^\nu K_\nu \left(\frac{\sqrt{2\nu}r}{\rho}\right),$$

where $K_\nu(\cdot)$ is a modified Bessel function, scale $\sigma_{kernel}^2 > 0$ and lengthscale $\rho > 0$ are hyperparameters.

The followings are some commonly chosen ν:

$$\nu = 1/2, k(r) = \sigma_{kernel}^2 exp\left(-\frac{r}{\rho}\right)$$

$$\nu = 3/2, k(r) = \sigma_{kernel}^2 \left(1 + \frac{\sqrt{3}r}{\rho}\right) exp\left(-\frac{\sqrt{3}r}{\rho}\right)$$

$$\nu = 5/2, k(r) = \sigma_{kernel}^2 \left(1 + \frac{\sqrt{5}r}{\rho} + \frac{5r^2}{3\rho^2}\right) exp\left(-\frac{\sqrt{5}r}{\rho}\right)$$

Squared exponential kernel

$$k(r) = \sigma^2_{kernel} exp\left(-\frac{r^2}{\rho}\right),$$

where scale $\sigma^2_{kernel} > 0$ and lengthscale $\rho \geq 0$ are hyperparameters.

Squared exponential kernel - multivariate

$$k(\boldsymbol{x} - \boldsymbol{x}') = \sigma^2_{kernel} exp\left(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{x}')^T \Lambda_{kernel}(\boldsymbol{x} - \boldsymbol{x}')\right),$$

where scale $\sigma^2_{kernel} > 0$ and positive definite matrix Λ_{kernel} are hyperparameters. This kernel is not isotropic but is still stationary.

A.5 Eigen-decomposition of SE kernels

We write a unit scale SE kernel in the form

$$k(\boldsymbol{x}, \boldsymbol{x}') = exp(-\frac{1}{\rho}|\boldsymbol{x} - \boldsymbol{x}'|_2^2),$$

for some positive lengthscale ρ.

Here we additionally introduce eigen-decomposition of such kernel. We need this information to prove our theorems in later sections. Let $\mathcal{X} \subseteq \mathbb{R}^D$ be a compact set and $\mu(\boldsymbol{x} \in \mathcal{X}) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{0}, (4a)^{-1} I_D)$ be a measure. Here the variance is formed as $(4a)^{-1}$ to make the later constants clean, but it can essentially be any positive real number. If $D = 1$ then SE kernel has s-th eigenvalues λ_s ($\lambda_1 \geq \lambda_2 \geq ...$) and the corresponding eigenvector $\Psi_s(\cdot)$ given by eq. (43)-(45) of Zhu et al. (1998) as

$$\lambda_s = \sqrt{\frac{2a}{A}} B^s \tag{10}$$

$$\Psi_s(x) = exp(-(c-a)x^2) H_s(\sqrt{2c}x), \tag{11}$$

where

$$H_s(x) = (-1)^s exp(x^2) \frac{d^s}{dx^s} exp(-x^2)$$

$$A = a + \frac{1}{\rho} + c, \quad c = \sqrt{a^2 + 2a\frac{1}{\rho}}, \quad B = \frac{1}{A\rho}.$$

Seeger et al. (2008)-appendix II further derived that if $D \geq 2$

$$\lambda_s \leq \mathcal{O}(B^{s^{1/D}}). \tag{12}$$

We can see that $0 < B < 1$, which is an important property later. Notice that such eigen-decomposition means

$$\int k(\boldsymbol{x}, \boldsymbol{x}') \Psi_s(\boldsymbol{x}) d\mu(\boldsymbol{x}) = \lambda_s \Psi(\boldsymbol{x}'), \text{and} \tag{13}$$

$$k(\boldsymbol{x}, \boldsymbol{x}') = \sum_{s \geq 1} \lambda_s \Psi_s(\boldsymbol{x}) \Psi_s^*(\boldsymbol{x}'). \tag{14}$$

Please see section 4.3 of Rasmussen and Williams (2006), Mercer's theorem for more details.

B Additional Lemmas

B.1 Inequalities

Before going further, we would like to introduce few inequalities we use later. Notice that the proofs of all of the following statements are in section C.

Lemma 4 *Given positive semidefinite matrices Q_1 and Q_2, $det(Q_1 + Q_2) \geq det(Q_1) + det(Q_2)$.*

Corollary 4.1 *For any N and any $\boldsymbol{x}_*$, the predictive covariance $\Sigma(\boldsymbol{x}_*)$ shown in equation 4 satisfies $det(\Omega_{**}) \geq det\left(\Sigma(\boldsymbol{x}_*)\right)$ and, similarly, the variance $\Sigma(\boldsymbol{x}_*, p_*)$ in eqation 6 satisfies $\eta_{p_*,p_*}(\boldsymbol{x}_*, \boldsymbol{x}_*) \geq \Sigma(\boldsymbol{x}_*, p_*)$.*

Lemma 5 *Recall that kernel $\eta_{p,p'}(\cdot,\cdot) = \sum_{l=1}^{L} W_{pl} W_{p'l} k_l(\cdot,\cdot)$ for some kernels k_l. With finite P and L, if $\hat{w} \geq |W_{pl}|$ and $\hat{v} \geq k_l(\cdot,\cdot) \geq 0$, for all p, l, then $det(\Omega_{**}) \leq (L\hat{w}^2\hat{v})^P$ and $\eta_{p_*,p_*}(\cdot,\cdot) \leq L\hat{w}^2\hat{v}$.*

The last lemma is adapted from Weyl's inequality for matrices.

Lemma 6 *Let $\{A_l \in \mathbb{R}^{M\times M}\}_{l=1}^{L}$ be Hermitian matrices, and let $B = \sum_{l=1}^{L} A_l$. Let $\alpha_{l,1} \geq \alpha_{l,2} \geq ... \geq \alpha_{l,M}$ be eigenvalues of A_l and $\beta_1 \geq ... \geq \beta_M$ be eigenvalues of B. Then $\beta_s \leq \sum_{l=1}^{L} \alpha_{l,\left[\frac{s-1}{L}+1\right]}$, where $[r]$ is the largest integer such that $r \geq [r]$, for all $r \in \mathbb{R}$.*

B.2 Mutual information

In addition, we here provide the mutual information in terms of the GP posterior, which can be used to obtain lemma 1 and theorem 2 in the main script.

Lemma 7 *Given data points $\{(\boldsymbol{x}_{n_i}, p_i)\}_{i=1}^{k-1}$ or $\{\boldsymbol{x}_i\}_{i=1}^{k-1}$, let $\hat{\Sigma}_{k-1}(\boldsymbol{x}_{n_k}, p_k)$ be predictive variance of $f_{p_k}(\boldsymbol{x}_{n_k})|\{y_{p_i n_i}\}_{i=1}^{k-1}$ for partially observed output, and $\hat{\Sigma}_{k-1}(\boldsymbol{x}_k)$ predictive covariance of $\boldsymbol{f}(\boldsymbol{x}_k)|\{\boldsymbol{y}_i\}_{i=1}^{k-1}$ for fully observed output. In addition, let $\hat{\Sigma}_0(\boldsymbol{x}_{n_1}, p_1) = \eta_{p_{n_1}}(\boldsymbol{x}_{n_1}, \boldsymbol{x}_{n_1})$ and $\hat{\Sigma}_0(\boldsymbol{x}_1) = \boldsymbol{\eta}(\boldsymbol{x}_1, \boldsymbol{x}_1)$ for the two settings. The mutual information is then described as follows:*

$$I\left(\boldsymbol{Y}_\phi, \{f_{p_k}(\boldsymbol{x}_{n_k})\}_{k=1}^{N_{sum}}\right) = \frac{1}{2}\sum_{k=1}^{N_{sum}} log\left(1 + \frac{1}{\sigma_{p_k}^2}\hat{\Sigma}_{k-1}(\boldsymbol{x}_{n_k}, p_k)\right),$$

$$I\left(\boldsymbol{Y}, \{\boldsymbol{f}(\boldsymbol{x}_k)\}_{k=1}^{N}\right) = \frac{1}{2}\sum_{k=1}^{N} log\left(\left|I_P + diag(\{\sigma_i^2\}_{i=1}^{P})^{-1}\hat{\Sigma}_{k-1}(\boldsymbol{x}_k)\right|\right).$$

Notice that this lemma does not involve active learning query yet. It only correlates posterior (co)variance to mutual information. This is why I use the notation $\hat{\Sigma}_{k-1}$ which is different from Σ_{k-1} in the main paper and in the next section of this supplementary content.

B.3 Kernel on rotated data

We also need a lemma about kernel eigenvalues for later analysis.

Lemma 8 *Let $\mathcal{X} \in \mathbb{R}^D$ be a compact set, Q be an orthonormal matrix (rotation matrix), $\mathcal{U} = Q\mathcal{X}$. Given distribution $\mu(\cdot) = \mathcal{N}(\cdot|\boldsymbol{0}, (4a)^{-1}I)$ for some positive constant a, and given kernel $k(\boldsymbol{x}, \boldsymbol{x}')$ and $k_Q(\boldsymbol{u}, \boldsymbol{u}')$ s.t. $k_Q(\boldsymbol{u}, \boldsymbol{u}') = k(\boldsymbol{x}, \boldsymbol{x}')$ for $\boldsymbol{u} = Q\boldsymbol{x}, \boldsymbol{u}' = Q\boldsymbol{x}'$ and max $k(\cdot,\cdot) = 1$. Then k and k_Q must have the same eigenvalues w.r.t. the same distribution μ.*

C Proof of Additional Lemmas

C.1 Proof of lemma 4

Let $\{q_{1,i}\}$ and $\{q_{2,i}\}$ be the eigenvalues of Q_1 and Q_2, respectively, then $q_{j,i} \geq 0$ implies

$$\begin{aligned} det(Q_1 + Q_2) &= \prod_i (q_{1,i} + q_{2,i}) \\ &\geq \prod_i q_{1,i} + \prod_i q_{2,i} = det(Q_1) + det(Q_2). \end{aligned}$$

C.2 Proof of corollary 4.1

Let $B = \Omega_{NN} + diag\{\sigma_i^2\} \otimes I_N$. As B is a positive definite matrix, so is its inverse. This means

$$\begin{aligned} &\forall a \in \mathbb{R}^{PN} \setminus \mathbf{0}, a^T B^{-1} a > 0 \\ &\Rightarrow \forall b \in \mathbb{R}^P, b^T \Omega_{N*}^T B^{-1} \Omega_{N*} b \\ &\qquad = (\Omega_{N*} b)^T B^{-1} (\Omega_{N*} b) \geq 0, \end{aligned}$$

which implies that $\Omega_{N*}^T B^{-1} \Omega_{N*}$ is semi-positive definite. Notice that $\Omega_{N*} b$ might be a zero vector. Apply lemma 4, let

$$\begin{aligned} Q_1 &= \Sigma(\boldsymbol{x}_*), \\ Q_2 &= \Omega_{N*}^T B^{-1} \Omega_{N*}, \end{aligned}$$

then $Q_1 + Q_2 = \Omega_{**}$ implies that

$$\begin{aligned} det(\Omega_{**}) &\geq det(\Sigma(\boldsymbol{x}_*)) + det(\Omega_{N*}^T B^{-1} \Omega_{N*}) \\ &\geq det(\Sigma(\boldsymbol{x}_*)). \end{aligned}$$

To addapt similar result to eq. 6, it is actually not necessary to use lemma 5, but it is easier for applying later if we put the statements together.

Let $B = \Omega_{N_{sum} N_{sum}} + diag(\{\sigma_{p_k}^2\}_{k=1}^{N_{sum}})$, then $[\Omega_{N_{sum}*}]_{all,p*}^T B^{-1} [\Omega_{N_{sum}*}]_{all,p*}$ is a non-negative scalar. Therefore, from eq. 6, we directly see that

$$\eta_{p_*,p_*}(\boldsymbol{x}_*, \boldsymbol{x}_*) \geq \Sigma(\boldsymbol{x}_*, p_*).$$

C.3 Proof of lemma 5

For any N and any $\boldsymbol{x}_*$, since eigen values of semi-positive definite matrices are non-negative, inequality of arithmetic and geometric means gives us

$$\begin{aligned}
det(\Omega_{**}) &= \prod\{eigenvalues\} \\
&\leq (\frac{\sum\{eigenvalues\}}{P})^P \\
&= \left(\frac{1}{P} trace(\Omega_{**})\right)^P \\
&= \left(\frac{1}{P}\sum_{p=1}^{P}\eta_{p,p}(*,*)\right)^P \\
&= \left(\frac{1}{P}\sum_{p=1}^{P}\sum_{l=1}^{L} W_{pl}^2 k_l(*,*)\right)^P \\
&\leq \left(\frac{1}{P}\sum_{p=1}^{P}\sum_{l=1}^{L} \hat{w}^2\hat{v}\right)^P \\
&= (L\hat{w}^2\hat{v})^P.
\end{aligned}$$

Meanwhile, from line 4 we have also obtained $\eta_{p_*,p_*}(\cdot,\cdot) \leq L\hat{w}^2\hat{v}$.

C.4 Proof of lemma 6

First notice that $\left[\frac{s-1}{L}+1\right] \in \mathbb{N}$ for all $s \in \mathbb{N}$. We use the induction.

1. When $L = 2$, Weyl's inequality tells us that

$$\begin{aligned}
\beta_s &\leq \alpha_{1,i} + \alpha_{2,j}, \\
if\ s &\geq i + j - 1.
\end{aligned}$$

   We clearly see that

$$\begin{aligned}
s = 2\left(\frac{s-1}{2}+1\right) - 1 \geq \left[\frac{s-1}{2}+1\right] + \left[\frac{s-1}{2}+1\right] - 1, \\
\text{so } \beta_s \leq \alpha_{1,\left[\frac{s-1}{2}+1\right]} + \alpha_{2,\left[\frac{s-1}{2}+1\right]}.
\end{aligned}$$

2. For any integer $T \geq 2$, assume $\beta_s \leq \sum_{l=1}^{T} \alpha_{l,\left[\frac{s-1}{T}+1\right]}$ for $B = \sum_{l=1}^{T} A_l$.

   Let $\hat{B} = \sum_{l=1}^{T+1} A_l$, and let $\{\hat{\beta}_i\}_{i=1}^{M}$ be it's eigenvalues ranking in order.

   Notice that by definition, it is easy to see that $[r] + 1 = [r+1], \forall r \in \mathbb{R}$.

$$\begin{aligned}
\hat{B} &= B + A_{T+1}, \\
\text{Weyl's inequality and } s &\geq \left[\frac{Ts+1}{T+1}\right] + \left[\frac{s-1}{T+1}+1\right] - 1 \\
\Rightarrow \hat{\beta}_s &\leq \beta_{\left[\frac{Ts+1}{T+1}\right]} + \alpha_{T+1,\left[\frac{s-1}{T+1}+1\right]}, \\
\text{Induction hypothesis} \Rightarrow \hat{\beta}_s &\leq \left(\sum_{l=1}^{T} \alpha_{l,\left[\frac{1}{T}\left[\frac{Ts+1}{T+1}-1\right]+1\right]}\right) + \alpha_{T+1,\left[\frac{s-1}{T+1}+1\right]}.
\end{aligned}$$

Now notice that

$$\begin{aligned}
&\frac{Ts+1}{T+1}-2 \leq \left[\frac{Ts+1}{T+1}-1\right] \leq \frac{Ts+1}{T+1}-1 \\
\Rightarrow &\frac{Ts-T}{T+1}-1 \leq \left[\frac{Ts+1}{T+1}-1\right] \leq \frac{Ts-T}{T+1} \\
\Rightarrow &\left[\frac{s-1}{T+1}-\frac{1}{T}\right]+1 \leq \left[\frac{1}{T}\left[\frac{Ts+1}{T+1}-1\right]+1\right] \leq \left[\frac{s-1}{T+1}\right]+1.
\end{aligned}$$

Denote integer $I = \left[\frac{s-1}{T+1}\right]$. The previous line tells us either $I = \left[\frac{s-1}{T+1} - \frac{1}{T}\right]$ or $I > \left[\frac{s-1}{T+1} - \frac{1}{T}\right]$.

Let's now inspect what they imply to the index of our interest $\left[\frac{s-1}{T+1}+1\right] = I+1$.

If $I = \left[\frac{s-1}{T+1} - \frac{1}{T}\right]$

$$I \leq \left[\frac{1}{T}\left[\frac{Ts+1}{T+1}-1\right]\right] \leq I \Rightarrow \left[\frac{1}{T}\left[\frac{Ts+1}{T+1}-1\right]\right] = I,$$

if $I > \left[\frac{s-1}{T+1} - \frac{1}{T}\right]$

$$\begin{aligned}
&\Rightarrow I + \frac{1}{T} > \frac{s-1}{T+1} \geq I \\
&\Rightarrow TI + 1 > \frac{Ts-T}{T+1} \geq TI \\
TI \in \mathbb{Z} &\Rightarrow TI + 1 > \left[\frac{Ts+1}{T+1}-1\right] \geq TI \\
&\Rightarrow \left[\frac{Ts+1}{T+1}-1\right] = TI \\
&\Rightarrow \left[\frac{1}{T}\left[\frac{Ts+1}{T+1}-1\right]\right] = I.
\end{aligned}$$

We therefore know that the index $\left[\frac{1}{T}\left[\frac{Ts+1}{T+1}-1\right]+1\right]$ is exactly $\left[\frac{s-1}{T+1}+1\right]$, which means

$$\begin{aligned}
\hat{\beta}_s &\leq \left(\sum_{l=1}^{T} \alpha_{l,\left[\frac{1}{T}\left[\frac{Ts+1}{T+1}-1\right]+1\right]}\right) + \alpha_{T+1,\left[\frac{s-1}{T+1}+1\right]} \\
&= \left(\sum_{l=1}^{T} \alpha_{l,\left[\frac{s-1}{T+1}+1\right]}\right) + \alpha_{T+1,\left[\frac{s-1}{T+1}+1\right]} \\
&= \sum_{l=1}^{T+1} \alpha_{l,\left[\frac{s-1}{T+1}+1\right]}
\end{aligned}$$

3. Then by induction we have proved the lemma.

C.5 Proof of lemma 7

We follow the idea of lemma 1 in Zimmer et al. (2018) but extend to multioutput kernel. We prove the 2 cases separately.

Proof of lemma 7 for partially observed outputs

By definition, $I\left(\boldsymbol{Y}_\phi, \{f_{p_k}(\boldsymbol{x}_{n_k})\}_{k=1}^{N_{sum}}\right) = H\left(\boldsymbol{Y}_\phi\right) - H\left(\boldsymbol{Y}_\phi | \{f_{p_k}(\boldsymbol{x}_{n_k})\}_{k=1}^{N_{sum}}\right)$.

As $\boldsymbol{Y}_\phi|\{f_{p_k}(\boldsymbol{x}_{n_k})\}_{k=1}^{N_{sum}}$ are i.i.d. Gaussian noises, i.e. covariance $= diag\left(\{\sigma_{p_k}^2\}_{k=1}^{N_{sum}}\right)$, we immediately have

$$H\left(\boldsymbol{Y}_\phi|\{f_{p_k}(\boldsymbol{x}_{n_k})\}_{k=1}^{N_{sum}}\right) = \sum_{k=1}^{N_{sum}} \frac{1}{2} log\left(2\pi e \sigma_{p_k}^2\right).$$

Apply the chain rule of differential entropy,

$$\begin{aligned} H\left(\boldsymbol{Y}_\phi\right) &= H\left(y_{p_{N_{sum}} n_{N_{sum}}}|\{y_{p_{N_k} n_k}\}_{k=1}^{N_{sum}-1}\right) + H\left(\{y_{p_{N_k} n_k}\}_{k=1}^{N_{sum}-1}\right) \\ &\vdots \\ &= \sum_{k=2}^{N_{sum}} H\left(y_{p_k n_k}|\{y_{p_i n_i}\}_{i=1}^{k-1}\right) + H\left(y_{p_1 n_1}\right). \end{aligned}$$

Under the GP assumption, we know that for $k = 2, ..., N_{sum}$, $y_{p_k n_k}|\{y_{p_i n_i}\}_{i=1}^{k-1}$ is Gaussian with variance equal to the sum of predictive variance and noise variance $\hat{\Sigma}_{k-1}(\boldsymbol{x}_{n_k}, p_k) + \sigma_{p_k}^2$, which gives us

$$H\left(y_{p_k n_k}|\{y_{p_i n_i}\}_{i=1}^{k-1}\right) = \frac{1}{2} log\left(2\pi e(\hat{\Sigma}_{k-1}(\boldsymbol{x}_{n_k}, p_k) + \sigma_{p_k}^2)\right), \forall k = 2, 3, ..., N_{sum}.$$

We also know that

$$H\left(y_{p_1 n_1}\right) = \frac{1}{2} log\left(2\pi e(\hat{\Sigma}_0(\boldsymbol{x}_{n_1}, p_1) + \sigma_{p_1}^2)\right).$$

Combining all we have above, we obtain

$$\begin{aligned} I\left(\boldsymbol{Y}_\phi, \{f_{p_k}(\boldsymbol{x}_{n_k})\}_{k=1}^{N_{sum}}\right) &= \frac{1}{2} \sum_{k=1}^{N_{sum}} log\left(\frac{2\pi e(\hat{\Sigma}_{k-1}(\boldsymbol{x}_{n_k}, p_k) + \sigma_{p_k}^2)}{2\pi e \sigma_{p_k}^2}\right) \\ &= \frac{1}{2} \sum_{k=1}^{N_{sum}} log\left(1 + \frac{1}{\sigma_{p_k}^2} \hat{\Sigma}_{k-1}(\boldsymbol{x}_{n_k}, p_k)\right). \end{aligned}$$

Proof of lemma 7 for fully observed outputs

Similarly,

$$\begin{aligned} I\left(\boldsymbol{Y}, \{\boldsymbol{f}(\boldsymbol{x}_n)\}_{n=1}^N\right) &= H\left(\boldsymbol{Y}\right) - H\left(\boldsymbol{Y}|\{\boldsymbol{f}(\boldsymbol{x}_n)\}_{n=1}^N\right) \\ &= \{\sum_{n=2}^N H\left(\boldsymbol{y}_n|\{\boldsymbol{y}_i\}_{i=1}^{n-1}\right) + H\left(\boldsymbol{y}_1\right)\} - H\left(\boldsymbol{Y}|\{\boldsymbol{f}(\boldsymbol{x}_n)\}_{n=1}^N\right) \\ &= \frac{\sum_{n=1}^N log\left[(2\pi e)^P|\hat{\Sigma}_{n-1}(\boldsymbol{x}_n) + diag(\{\sigma_i^2\}_{i=1}^P)|\right]}{2} - \frac{\sum_{n=1}^N log\left[(2\pi e)^P|diag(\{\sigma_i^2\}_{i=1}^P)|\right]}{2} \\ &= \frac{1}{2} log\left[\frac{(2\pi e)^{PN}\prod_{n=1}^N|\hat{\Sigma}_{n-1}(\boldsymbol{x}_n) + diag(\{\sigma_i^2\}_{i=1}^P)|}{(2\pi e)^{PN}\prod_{n=1}^N|diag(\{\sigma_i^2\}_{i=1}^P)|}\right] \\ &= \frac{1}{2} log\left(\prod_{n=1}^N\left|diag(\{\sigma_i^2\}_{i=1}^P)^{-1}\hat{\Sigma}_{n-1}(\boldsymbol{x}_n) + I_P\right|\right) \end{aligned}$$

C.6 Proof of lemma 8

We first see that the distribution μ on $\mathcal{X}$ and $\mathcal{U}$ is exactly the same:

$$\begin{aligned}\mathcal{N}(Q\boldsymbol{x}|\boldsymbol{0},(4a)^{-1}I) &\propto exp(-(Q\boldsymbol{x})^T(4a)I_D(Q\boldsymbol{x}))\\ &= exp(-(4a)\boldsymbol{x}^TQ^TQ\boldsymbol{x})\\ &= exp(-(4a)\boldsymbol{x}^T\boldsymbol{x})\\ &\propto \mathcal{N}(\boldsymbol{x}|\boldsymbol{0},(4a)^{-1}I).\end{aligned}$$

Now we consider the kernel eigenvalues from eq. (13). For any function $\Psi(\cdot)$, Jacobian operator J and for $\boldsymbol{u}=Q\boldsymbol{x}$, we must have

$$\begin{aligned}\int k(\boldsymbol{x},\boldsymbol{x}')\Psi(\boldsymbol{x})d\mu(\boldsymbol{x}) &= \int k(\boldsymbol{x},\boldsymbol{x'})\Psi(\boldsymbol{x})d\mu(\boldsymbol{x})\\ &= \int k_Q(\boldsymbol{u},\boldsymbol{u'})\Psi(Q^T\boldsymbol{u})|J_{\boldsymbol{x}}(\mu(\boldsymbol{x}))J_{\boldsymbol{u}}(\boldsymbol{x})J_{\boldsymbol{u}}^{-1}(\mu(\boldsymbol{u}))|d\mu(\boldsymbol{u})\\ &= \int k_Q(\boldsymbol{u},\boldsymbol{u'})\Psi(Q^T\boldsymbol{u})d\mu(\boldsymbol{u}).\end{aligned}$$

In the second line we change variable $d\mu(\boldsymbol{x})=\frac{d\mu(\boldsymbol{x})}{d\boldsymbol{x}}\frac{d\boldsymbol{x}}{d\boldsymbol{u}}\frac{d\boldsymbol{u}}{d\mu(\boldsymbol{u})}$. Note that

$$\begin{aligned}J_{\boldsymbol{x}}(\mu(\boldsymbol{x})) &= 8a\mu(\boldsymbol{x})I\boldsymbol{x},\\ J_{\boldsymbol{u}}(\mu(\boldsymbol{u})) &= 8a\mu(\boldsymbol{u})I\boldsymbol{u} = 8a\mu(\boldsymbol{x})Q\boldsymbol{x},\\ J_{\boldsymbol{u}}(\boldsymbol{x}) &= Q^T.\end{aligned}$$

This and eq. (13) tell us that:

1. If λ,Ψ are an eigenvalue and it's corresponding eigenfunction of k_Q w.r.t. μ, then
   $\int k(\boldsymbol{x},\boldsymbol{x}')\Psi(Q\boldsymbol{x})d\mu(\boldsymbol{x}) = \int k_Q(\boldsymbol{u},\boldsymbol{u}')\Psi(\boldsymbol{u})d\mu(\boldsymbol{u}) = \lambda\Psi(\boldsymbol{u}') = \lambda\Psi(Q\boldsymbol{x}')$,
   which means $\lambda,\Psi(Q\cdot)$ are an eigenvalue, eigenfunction of k w.r.t. μ.

2. If we look at the equation reversely, $\lambda,\bar{\Psi}$ are an eigenvalue and the corresponding eigenfunction of k w.r.t. μ, then
   $\int k_Q(\boldsymbol{u},\boldsymbol{u}')\bar{\Psi}(Q^T\boldsymbol{u})d\mu(\boldsymbol{u}) = \int k(\boldsymbol{x},\boldsymbol{x}')\bar{\Psi}(\boldsymbol{x})d\mu(\boldsymbol{x}) = \lambda\bar{\Psi}(\boldsymbol{x}') = \lambda\bar{\Psi}(Q^T\boldsymbol{u}')$
   also implies that $\lambda,\bar{\Psi}(Q^T\cdot)$ are an eigenvalue, eigenfunction of k_Q w.r.t. μ.

Therefore, these two kernels have the same eigenvalues w.r.t. the same measure μ.

D Proof of lemma 1

This is an extension of lemma 4 in Zimmer et al. (2018). We know that for $0<a\le b$, $\frac{a}{log(1+a)}\le\frac{b}{lob(1+b)}$. In addition, our acquisition function guarantees that $\Sigma_{k-1}(\cdot,\cdot)\le\Sigma_{k-1}(\boldsymbol{x}_{n_k},p_k)$ and $|\Sigma_{n-1}(\cdot)|\le|\Sigma_{n-1}(\boldsymbol{x}_n)|$ because, without safety constraint, the queries are always with the maximal variance or determinant of covariance, see eq. 7, 9. Therefore we only need to bound $\sum_{k=1}^{N_{sum}}\Sigma_{k-1}(\boldsymbol{x}_{n_k},p_k)$ and $\sum_{n=1}^{N}|\Sigma_{n-1}(\boldsymbol{x}_n)|$.

D.1 Proof of lemma 1 - partially observed outputs

Apply corollary 4.1 and lemma 5, we have

$$\forall k=1,2,...,N_{sum},\Sigma_{k-1}(\boldsymbol{x}_{n_k},p_k)\le L\hat{w}^2\hat{v}$$

Now for some index k, we can simply set $a = \frac{1}{\sigma^2_{p_k}}\Sigma_{k-1}(\boldsymbol{x}_{n_k}, p_k), b = \frac{1}{\sigma^2_{p_k}} L\hat{w}^2\hat{v}$ and then $\frac{a}{log(1+a)} \leq \frac{b}{log(1+b)}$ gives

$$\begin{aligned}
\frac{\frac{1}{\sigma^2_{p_k}}\Sigma_{k-1}(\boldsymbol{x}_{n_k}, p_k)}{log\left(1 + \frac{1}{\sigma^2_{p_k}}\Sigma_{k-1}(\boldsymbol{x}_{n_k}, p_k)\right)} &\leq \frac{\frac{1}{\sigma^2_{p_k}} L\hat{w}^2\hat{v}}{log\left(1 + \frac{1}{\sigma^2_{p_k}} L\hat{w}^2\hat{v}\right)} \\
\Rightarrow \Sigma_{k-1}(\boldsymbol{x}_{n_k}, p_k) &\leq \frac{L\hat{w}^2\hat{v}}{log\left(1 + \frac{1}{\sigma^2_{p_k}} L\hat{w}^2\hat{v}\right)} log\left(1 + \frac{1}{\sigma^2_{p_k}}\Sigma_{k-1}(\boldsymbol{x}_{n_k}, p_k)\right) \qquad (15)
\end{aligned}$$

Since $\forall p, \psi \geq \sigma_p^2 > 0$, we know that

$$\begin{aligned}
\frac{L\hat{w}^2\hat{v}}{\sigma^2_{p_k}} &\geq \frac{L\hat{w}^2\hat{v}}{\psi} \\
\Rightarrow log\left(1 + \frac{L\hat{w}^2\hat{v}}{\sigma^2_{p_k}}\right) &\geq log\left(1 + \frac{L\hat{w}^2\hat{v}}{\psi}\right) \\
\Rightarrow \frac{L\hat{w}^2\hat{v}}{log\left(1 + \frac{1}{\sigma^2_{p_k}} L\hat{w}^2\hat{v}\right)} &\leq \frac{L\hat{w}^2\hat{v}}{log\left(1 + \frac{1}{\psi} L\hat{w}^2\hat{v}\right)} =: C_1. \qquad (16)
\end{aligned}$$

Combine eq. 15 and eq. 16, then

$$\Sigma_{k-1}(\boldsymbol{x}_{n_k}, p_k) \leq C_1 log(1 + \frac{1}{\sigma^2_{p_k}}\Sigma_{k-1}(\boldsymbol{x}_{n_k}, p_k)).$$

Sum them up over indices k and apply lemma 7, we get $\sum_{k=1}^{N_{sum}} \Sigma_{k-1}(\boldsymbol{x}_{n_k}, p_k) \leq 2C_1 I\left(\{y_k\}_{k=1}^{N_{sum}}, \{f_{p_k}(\boldsymbol{x}_{n_k})\}_{k=1}^{N_{sum}}\right)$

D.2 Proof of lemma 1 - fully observed outputs

Corollary 4.1 and lemma 5 tell us that $|\Sigma_{n-1}(\boldsymbol{x}_n)| \leq (L\hat{w}^2\hat{v})^P$. Apply the same inequality $\frac{a}{log(1+a)} \leq \frac{b}{lob(1+b)}$ for $0 < a \leq b$ with

$$a = \frac{1}{\prod_{i=1}^P \sigma_i^2}|\Sigma_{n-1}(\boldsymbol{x}_n)|, b = \frac{1}{\prod_{i=1}^P \sigma_i^2}(L\hat{w}^2\hat{v})^P,$$

then we get

$$\begin{aligned}
\frac{1}{\prod_{i=1}^P \sigma_i^2}|\Sigma_{n-1}(\boldsymbol{x}_n)| &\leq \frac{b}{lob(1+b)} log(1+a) \\
&= \frac{\frac{1}{\prod_{i=1}^P \sigma_i^2}(L\hat{w}^2\hat{v})^P}{log\left(1 + \frac{1}{\prod_{i=1}^P \sigma_i^2}(L\hat{w}^2\hat{v})^P\right)} log\left(1 + \frac{1}{\prod_{i=1}^P \sigma_i^2}|\Sigma_{n-1}(\boldsymbol{x}_n)|\right) \\
&\leq \frac{\frac{1}{\prod_{i=1}^P \sigma_i^2}(L\hat{w}^2\hat{v})^P}{log\left(1 + \frac{1}{\psi^P}(L\hat{w}^2\hat{v})^P\right)} log\left(1 + \frac{1}{\prod_{i=1}^P \sigma_i^2}|\Sigma_{n-1}(\boldsymbol{x}_n)|\right) \\
\Rightarrow |\Sigma_{n-1}(\boldsymbol{x}_n)| &\leq C_2 log\left(1 + \frac{1}{\prod_{i=1}^P \sigma_i^2}|\Sigma_{n-1}(\boldsymbol{x}_n)|\right) \qquad (17) \\
&\leq C_2 log\big|I_P + diag(\{\sigma_i^2\}_{i=1}^P)^{-1}\Sigma_{n-1}(\boldsymbol{x}_n)\big|, C_2 = \frac{(L\hat{w}^2\hat{v})^P}{log\left(1 + \left(\frac{L\hat{w}^2\hat{v}}{\psi}\right)^P\right)}.
\end{aligned}$$

Sum them up again over indices n and apply lemma 7: $\sum_{n=1}^N |\Sigma_{n-1}(\boldsymbol{x}_n)| \leq 2C_2 I\left(\boldsymbol{Y}, \{\boldsymbol{f}(\boldsymbol{x}_n)\}_{n=1}^N\right)$

E Proof of theorem 2

E.1 Proof of thm 2 - part 1: bound predictive uncertainty

Let's first consider the mutual information in terms of GP priors. When the outputs are all fully observed,

$$\begin{aligned} I\left(\boldsymbol{Y}, \boldsymbol{f}\right) &= H\left(\boldsymbol{Y}\right) - H\left(\boldsymbol{Y}|\boldsymbol{f}\right) \\ &= \frac{1}{2} log\big|2\pi e\left(\Omega_{NN} + diag(\{\sigma_i^2\}_{i=1}^P) \otimes I_N\right)\big| - \frac{1}{2} log\big|2\pi e\ diag\left(\{\sigma_i^2\}_{i=1}^P\right) \otimes I_N\big| \\ &= \frac{1}{2} log\big|I_{PN} + \left(diag(\{\sigma_i^2\}_{i=1}^P) \otimes I_N\right)^{-1} \Omega_{NN}\big|. \end{aligned}$$

Notice that

$$I_{PN} + \left(diag(\{\sigma_i^2\}_{i=1}^P) \otimes I_N\right)^{-1} \Omega_{NN} = I_{PN} + \begin{pmatrix} \sigma_1^{-2}\eta_{1,1}(\boldsymbol{X}, \boldsymbol{X}) & & \dots \\ & \ddots & \\ \dots & & \sigma_P^{-2}\eta_{P,P}(\boldsymbol{X}, \boldsymbol{X}) \end{pmatrix},$$

where the matrix itself and all of its diagonal blocks $I_N + \sigma_p^{-2}\eta_{p,p}(\boldsymbol{X}, \boldsymbol{X})$ are positive definite Hermitian matrices, so we can apply Fischer's inequality and obtain

$$I\left(\boldsymbol{Y}, \boldsymbol{f}\right) \leq \frac{1}{2} \sum_{p=1}^{P} log\big|I_N + \sigma_p^{-2}\eta_{p,p}(\boldsymbol{X}, \boldsymbol{X})\big|.$$

This is actually the sum of mutual information of GPs $f_p \sim \mathcal{GP}(0, \eta_{p,p})$, $y_p|f_p \sim \mathcal{N}(0, \sigma_p^2)$. As these are standard single output GPs, we can use the maximum information gain introduced in Srinivas et al. (2012) for the bound

$$I\left(\boldsymbol{Y}, \boldsymbol{f}\right) \leq \sum_{p=1}^{P} \gamma_p^N.$$

If the observations are partially observed, Ω_{NN} has the corresponding rows and columns omitted but the rest stays in the same form. Therefore, with Fischer's inequality, we again have

$$I\left(\boldsymbol{Y}_\phi, \{f_{p_k}(\boldsymbol{x}_{n_k})\}_k\right) \leq \sum_{p=1}^{P} \gamma_p^{N_p} = \sum_{p=1}^{P} max_{\{y_{pn}\}_{n=1}^{N_p}} I\left(\{y_{pn}\}_{n=1}^{N_p}, f_p\right).$$

Apply lemma 1 and we get the first part of our theorem:

$$\begin{aligned} \frac{1}{N} \sum_{n=1}^{N} |\Sigma_{n-1}(\hat{\boldsymbol{x}}_{\boldsymbol{n}})| &\leq \mathcal{O}\left(\frac{1}{N} \sum_{p=1}^{P} \gamma_p^{N_p}\right), \text{ here } N_p = N, \\ \frac{1}{N_{sum}} \sum_{k=1}^{N_{sum}} \Sigma_{k-1}(\hat{\boldsymbol{x}}_{\boldsymbol{k}}, \hat{p}_k) &\leq \mathcal{O}\left(\frac{1}{N_{sum}} \sum_{p=1}^{P} \gamma_p^{N_p}\right). \end{aligned}$$

Notice that in the main script, $n = N$ for fully observed outputs and $n = N_{sum}$ for partially observed outputs.

E.2 Proof of thm 2 - part 2: bound maximum information gain

E.2.1 Proof of theorem 2 - part 2.0: general purposes

It now remains to bound the maximum information gains $\gamma_p^{N_p} = max_{\mathcal{D} \subseteq \mathcal{Y}_p(\mathcal{X}):|\mathcal{D}|=N_p} I(\mathcal{D}, f_p)$. This quantity is considered directly on the compact set $\mathcal{X}$. The notation $\hat{x}_k$ used earlier is not important anymore and can

be ignored. We aim to obtain the bounds by extending theorem 5 in Srinivas et al. (2012) to our kernels $\eta_{p,p}$. In Srinivas et al. (2012), eigenvalues of the kernel (see section 4.3 of Rasmussen and Williams (2006) or Mercer's theorem) played an important role on computing the maximum information gain of a system. However, computing the exact eigenvalues is generally hard. Instead, for an isotropic kernel, Seeger et al. (2008) and Srinivas et al. (2012) showed how to bound the eigenvalues (with respect to gaussian or uniform distribution) by spectral density (also see Rasmussen and Williams (2006)) and applied this to obtaining bound for maximum information gains. Srinivas et al. (2012) also provide bounds for squared exponential kernel and Matérn kernels. Here we follow their analysis but extend it to MO kernels.

The main challenge is to compute the spectral density bounds or to bound the eigenvalues accordingly (Rasmussen and Williams (2006), Seeger et al. (2008), Srinivas et al. (2012)).

For simplicity we first normalized the latent kernels. Let

$$\begin{aligned} c_l &= 1/\max\{k_l(\cdot,\cdot)\} \\ \tilde{k}_l(\cdot,\cdot) &= c_l k_l(\cdot,\cdot) \\ \tilde{W}_{pl} &= \sqrt{c_l} W_{pl} \end{aligned}$$

Then $\eta_{p,p}(\cdot,\cdot) = \sum_{l=1}^{L} \tilde{W}_{pl}^2 \tilde{k}_l(\cdot,\cdot)$.

Here, the latent kernels k_l are either all squared exponential or all Matérn kernel with the same smoothing parameter ν (Rasmussen and Williams (2006)). Each kernel is allowed to have different lengthscales. We consider the 2 scenarios individually.

E.2.2 Proof of theorem 2 - part 2.1 - Matérn kernel

Here the latent kernels are Matérn kernels. Consider the spectral density $\lambda_{\eta_{p,p}}(\omega)$ of kernel $\eta_{p,p}(r)$ by letting $r = |\boldsymbol{x} - \boldsymbol{x}'|, \forall \boldsymbol{x}, \boldsymbol{x}'$

$$\begin{aligned} \lambda_{\eta_{p,p}}(\omega) &= \int \eta_{p,p}(r) e^{-2\pi i \omega \ r} dr \\ &= \int \sum_{l=1}^{L} \tilde{W}_{pl}^2 \tilde{k}_l(r) e^{-2\pi i \omega \ r} dr \\ &= \sum_{l=1}^{L} \tilde{W}_{pl}^2 \int \tilde{k}_l(r) e^{-2\pi i \omega \ r} dr \\ &= \sum_{l=1}^{L} \tilde{W}_{pl}^2 \lambda_{\tilde{k}_l}(\omega) \end{aligned}$$

Let $\tilde{k}_l$ be ν-Matérn kernel with lengthscale $\rho_l > 0$. From section 4.2 (eq. (4.15)) of Rasmussen and Williams (2006), we have

$$\begin{aligned} \lambda_{\tilde{k}_l}(\omega) &= \frac{2^D \pi^{D/2} \Gamma(\nu + D/2)(2\nu)^\nu}{\Gamma(\nu)\rho_l^{2\nu}} \left(\frac{2\nu}{\rho_l^2} + 4\pi^2\omega^2\right)^{-(\nu+D/2)} \\ &= \mathcal{O}\left(\left(\frac{2\nu}{\rho_l^2} + 4\pi^2\omega^2\right)^{-(\nu+D/2)}\right), \end{aligned}$$

where D is the dimension of any point $\boldsymbol{x} \in \mathcal{X} \subseteq \mathbb{R}^D$. With huge frequency (which is how it is used in Seeger

et al. (2008), Srinivas et al. (2012)), the spectral density is dominated by $\mathcal{O}\left(4\pi^2\omega^2\right)^{-(\nu+D/2)}$. Therefore

$$\begin{aligned}\lambda_{\eta_{p,p}}(\omega) &= \sum_{l=1}^{L} \tilde{W}_{pl}^2 \lambda_{\tilde{k}_l}(\omega) \\ &= \mathcal{O}\left(\left(4\pi^2\omega^2\right)^{-(\nu+D/2)}\right),\end{aligned}$$

which is exactly the same as the spectral density bound of one single ν-Matérn kernel. Follow the same procedure as in Srinivas et al. (2012), we can obtain the same bound of $\gamma_p^{N_p}$ for $\eta_{p,p}$ as for ν-Matérn kernel ($N_{sum} = NP$ when the data are fully observed)

$$\begin{aligned}\gamma_p^{N_p} &\leq \mathcal{O}\left(N_p^{D(D+1)/(2\nu+D(D+1))} log N_p\right) \\ &\leq \mathcal{O}\left(N_{sum}^{D(D+1)/(2\nu+D(D+1))} log N_{sum}\right) \\ \Rightarrow \frac{1}{N_{sum}} \sum_{p=1}^{P} \gamma_p^{N_p} &\leq \mathcal{O}\left(N_{sum}^{D(D+1)/(2\nu+D(D+1))} log N_{sum}\right).\end{aligned}$$

Recall that $\gamma_p^{N_p} = max_{\mathcal{D}\subseteq\mathcal{Y}_p(\mathcal{X}):|\mathcal{D}|=N_p} I(\mathcal{D}, f_p)$. Notice that Srinivas et al. (2012) assume uniform distribution for the spectrum analysis. This means we are actually considering the maximum information gain $\bar{\gamma}_p^{N_p}$ on a discretized set $\mathcal{X}_{dis}$ drawn from $\mathcal{X}$ where the following is fulfilled:

$$\forall \boldsymbol{x} \in \mathcal{X}, \exists \boldsymbol{x}_{dis} \in \mathcal{X}_{dis} \ s.t. \ |\boldsymbol{x} - \boldsymbol{x}_{dis}| \leq \text{error}, \ \text{error} = D^{1/2} N_p^{-1}. \tag{18}$$

Notice that for finite P, if we discretized the set s.t. the condition holds for error $= D^{1/2}\{\max_p N_p\}^{-1}$, then condition (18) holds for all $p = 1, ..., P$. Srinivas et al. (2012) provided extensive study on bounding the actual $\gamma_p^{N_p}$ (over general compact set) by $\bar{\gamma}_p^{N_p}$ (over finite discretized set). In our active learning scenario in practice, we can see this as we query N_p points (sequentially) in total from set $\mathcal{X}_{dis}$.

E.2.3 Proof of theorem 2 - part 2.2 - Squared exponential (SE) kernel

Let $\tilde{k}_l$ be SE kernel with lengthscale $\rho_l > 0$. Eigenvalues of a SE kernel are as described by eq. (10) (12) (provided from Zhu et al. (1998); Seeger et al. (2008)). Srinivas et al. (2012) further used the eigenvalues to derive the bound of maximum information gain for a system with one single SE kernel.

In our case, notice that each kernel $\tilde{k}_l$ has it's individual lengthscales and can be considered as different kernels. Eigenvalues of $\eta_{p,p}$ are not simply linear combination of eigenvalues of those individual kernels $\{\tilde{k}_l\}_{l=1}^{L}$ (this does not even happen on matrices). However, we can use eq. (10) (12) and lemma 6 to bound the eigenvalues of $\eta_{p,p}$, and then we use this to bound the maximum information gain with kernel $\eta_{p,p}$.

We organize the following proof in few steps.

1. **Goal: correlate eigenvalues**

   Recall that L is the number of latent kernels $\tilde{k}_l$. Let $\lambda_{l,1} \geq \lambda_{l,2} \geq ...$ be eigenvalues of $\tilde{W}_{pl}^2 \tilde{k}_l(\cdot,\cdot)$ on $\mathcal{X}_{dis}$, a finite discretization of $\mathcal{X}$ s.t. condition (18) holds, and $\tilde{\lambda}_1 \geq \tilde{\lambda}_2 \geq ...$ be eigenvalues for $\eta_{p,p}$. Please do not be confused by the notation $\lambda_\cdot$ in the Matérn kernel part. Since $\mathcal{X}_{dis}$ is finite, the kernel operators give us finite dimensional Hermitian matrices. Let S be the size of $\mathcal{X}_{dis}$ and we apply Lemma 6

   $$\tilde{\lambda}_s \leq \sum_{l=1}^{L} \lambda_{l,\left[\frac{s-1}{L}+1\right]}, \ \text{for } 1 \leq s \leq S.$$

   Srinivas et al. (2012) provides extensive analysis of maximum information gain from $\mathcal{X}_{dis}$ to $\mathcal{X}$ w.r.t. uniform distribution (measure). Therefore, we now only focus on the eigenvalues in the right hand side of this inequality.

2. **Goal: bound eigenvalues**

We then go back to the original definition for eigenvalues of kernel operators (section 4.3 of Rasmussen and Williams (2006) or Mercer's theorem). With Gaussian distribution as the measure, eq. (12) gives us $\lambda_{l,\left[\frac{s-1}{L}+1\right]} \leq \mathcal{O}\left(B_l^{\left[\frac{s-1}{L}+1\right]^{1/D}}\right)$, where $0 \leq B_l < 1$ is dependent on lengthscale ρ_l and D is the dimension ($\mathcal{X} \in \mathbb{R}^D$). Notice that according to Srinivas et al. (2012), this decay rate is the same as eigenvalues w.r.t. uniform distribution up to some constant factor. Also see the statement beneath condition 18 and eq. (13) regarding the distribution.

With the decay rate in mind, we go back to the previous finite discretization case. Let $B = max_l\{B_l\}$, then

$$\begin{aligned} & 0 \leq B_l \leq B < 1 \\ \Rightarrow \tilde{\lambda}_s & \leq \mathcal{O}\left(B_l^{\left[\frac{s-1}{L}+1\right]^{1/D}}\right) \\ & \leq \mathcal{O}\left(B^{\left[\frac{s-1}{L}+1\right]^{1/D}}\right) \\ & \leq \mathcal{O}\left(B^{\left(\frac{s-1}{L}\right)^{1/D}}\right). \end{aligned}$$

3. **Goal: use eigenvalues bound and results from Srinivas et al. (2012)**

We compare the bound for eigenvalues of $\eta_{p,p}$ to bound for standard SE kernels. They have the same form but with different correlated index. Therefore, we follow the analysis Srinivas et al. (2012) with only minor differences.

Recall $\gamma_p^{N_p} = max_{\mathcal{D}\subseteq\mathcal{Y}_p(\mathcal{X}):|\mathcal{D}|=N_p} I(\mathcal{D}, f_p)$, S is the size of discretized set $\mathcal{X}_{dis}$ and $s_0 \leq S$ is an index we described later. Select $S = C_4 N_p^D \log N_p$ as in Srinivas et al. (2012). Theorem 8 in Srinivas et al. (2012) gives us:

$$\begin{aligned} \gamma_p^{N_p} & \leq \mathcal{O}\left(max_{r=1,\ldots,N_p}\left(s_0 log(rS/\sigma_p^2) + C_4\sigma_p^{-2}(1-r/N_p)(logN_p)(N_p^{D+1}\Lambda(s_0)+1)\right)\right) + \mathcal{O}\left(N_p^{1-D/D}\right) \\ & = \mathcal{O}\left(max_{r=1,\ldots,N_p}\left(s_0 log(rC_4 N_p^D logN_p/\sigma_p^2) + C_4\sigma_p^{-2}(1-r/N_p)(logN_p)(N_p^{D+1}\Lambda(s_0)+1)\right)\right), \end{aligned} \tag{19}$$

where $\Lambda(s_0) \coloneqq \sum_{s\geq s_0} \tilde{\lambda}_s$ and $C_4 = \int_{\boldsymbol{x}\in\mathcal{X}} d\boldsymbol{x}$ is the volume of the compact set $\mathcal{X}$, which is treated as a constant here (Srinivas et al. (2012) used C_4 to determine the uniform distribution).

4. **Goal: select parameters and get the final bound**

Now the only thing remains is to obtain $\Lambda(s_0)$. We follow Srinivas et al. (2012) by adjusting the selection of s_0.

Firstly, as in appendix II of Seeger et al. (2008): let $\beta = (-logB)\left(\frac{s_0-1}{L}+1\right)^{1/D}$, then

$$\begin{aligned} \Lambda(s_0) \leq \sum_{s\geq s_0} B^{\left(\frac{s-1}{L}+1\right)^{1/D}} & = \sum_{s\geq s_0} exp\left(\left(\frac{s-1}{L}+1\right)^{1/D} logB\right) \\ & \leq \int_{s_0}^{\infty} exp\left(\left(\frac{x-1}{L}+1\right)^{1/D} \log B\right) dx \\ & = \frac{LD}{(-logB)^D}\int_{\beta}^{\infty} t^{D-1}e^{-t}dt \\ & = \frac{LD}{(-logB)^D}\Gamma(D,\beta) \\ & = \frac{LD}{(-logB)^D}\left((D-1)!e^{-\beta}\sum_{q=0}^{D-1}\frac{(\beta)^q}{q!}\right) \\ & = \mathcal{O}\left(e^{-\beta}\beta^{D-1}\right). \end{aligned} \tag{20}$$

Note that in step three, we perform a change in variables with $t = (-\log B)\left(\frac{x-1}{L}+1\right)^{1/D} \Rightarrow dt/dx = \frac{-\log B}{LD}\left(\frac{x-1}{L}+1\right)^{(1-D)/D} = \frac{(-\log B)^D}{LD}t^{1-D} \Rightarrow dx = \frac{LD}{(-\log B)^D}t^{D-1}dt$. It turns out to produce different constant outside the integral than the one in appendix II of Seeger et al. (2008), which however is absorbed by $\mathcal{O}$.

Now select s_0 s.t. $(-logB)\left(\frac{s_0-1}{L}+1\right)^{1/D} = \beta = log(C_4 N_p^{D+1}(logN_p))$, then

$$s_0 = L\left(\left(\frac{log(C_4 N_p^{D+1} logN_p)}{-logB}\right)^D - 1\right) + 1 = \mathcal{O}\left(\beta^D\right).$$

Then, plug s_0, eq. (20) into eq. (19), with $\beta = log(C_4 N_p^{D+1}(logN_p))$ as above

$$\begin{aligned}
\gamma_p^{N_p} &\leq \mathcal{O}\left(max_{r=1,\dots,N_p}\left(s_0 log(rC_4 N_p^D logN_p/\sigma_p^2) + C_4\sigma_p^{-2}(1-r/N_p)(logN_p)(N_p^{D+1}\Lambda(s_0)+1)\right)\right) \\
&= \mathcal{O}\left(max_{r=1,\dots,N_p}\left(\beta^D log\left(\frac{re^\beta}{N_p\sigma_p^2}\right) + \sigma_p^{-2}(1-r/N_p)\beta^{D-1}\right)\right) \\
&= \mathcal{O}\left(\beta^D log\left(\frac{N_p e^\beta}{N_p\sigma_p^2}\right)\right) \\
&= \mathcal{O}\left((logN_p)^{D+1}\right) \\
&\leq \mathcal{O}\left((logN_{sum})^{D+1}\right)
\end{aligned} \tag{21}$$

$$(21) \Rightarrow \frac{1}{N_{sum}}\sum_{p=1}^{P}\gamma_p^{N_p} \leq \mathcal{O}\left(\frac{(logN_{sum})^{D+1}}{N_{sum}}\right)$$

E.2.4 Proof of theorem 2 - part 2.3 - SE kernel with matrix lengthscale

Here $\tilde{k}_l$ are SE kernel with matrix lengthscale. It suffices to show that the eigenvalues are the same as the previous one up to a constant scalar, i.e. eigenvalues $\lambda_{l,j}$ of kernel $\tilde{k}_l$ is bounded by $\mathcal{O}\left(B_l^{j^{1/D}}\right)$ (eq. (12)). The rest is identical to the previous part (section E.2.3). To preserve the same decay rate, the distribution we use for obtaining eigenvalues should stay the same. Also see the statement beneath condition 18 and eq. (13) regarding the distribution.

1. **Goal: decompose the kernel**

   We first decompose this kernel and diagonalize the lengthscale matrix.

   Given SE kernel in a multivariate form

   $$\tilde{k}_l(\boldsymbol{x}, \boldsymbol{x'}) = exp\left(-(\boldsymbol{x}-\boldsymbol{x'})^T\Theta_l(\boldsymbol{x}-\boldsymbol{x'})\right),$$

   where Θ_l is a positive definite Hermitian matrix. We know that there exists a diagonal matrix Λ (and all diagonal element $[\Lambda]_d > 0$) and a orthonormal matrix Q such that $\Theta_l = Q^T\Lambda Q$. Λ and Q depend on l but we omit it for simplicity. Notice that D is the dimension of $\boldsymbol{x}$. Then

   $$\begin{aligned}
   (\boldsymbol{x}-\boldsymbol{x'})^T\Theta_l(\boldsymbol{x}-\boldsymbol{x'}) &= (\Lambda^{1/2}Q(\boldsymbol{x}-\boldsymbol{x'}))^T(\Lambda^{1/2}Q(\boldsymbol{x}-\boldsymbol{x'})) \\
   &= \sum_{d=1}^{D}[\Lambda]_d[Q(\boldsymbol{x}-\boldsymbol{x'})]_d^2.
   \end{aligned}$$

   Therefore,

   $$\tilde{k}_l(\boldsymbol{x}, \boldsymbol{x'}) = \prod_{d=1}^{D} exp\left(-\{[\Lambda]_d\}([Q\boldsymbol{x}-Q\boldsymbol{x'}]_d)^2\right). \tag{22}$$

Lemma 8 tells us that the two kernels, $\tilde{k}_l(\boldsymbol{x}, \boldsymbol{x'})$ and the one without rotation $\prod_{d=1}^{D} exp\left(-\{[\Lambda]_d\}([\boldsymbol{x} - \boldsymbol{x}']_d)^2\right)$, have the same eigenvalues w.r.t. the same Gaussian measure, so we are able to omit the matrix Q for simplicity.

2. **Goal: bound the eigenvalues**

Let $b_{d,s}$ and $\Psi_{d,s}(x)$ be an eigenvalue and it's eigenfunction of kernel $exp(-[\Lambda]_d(x - x')^2)$ w.r.t. distribution $\mu(x) = \mathcal{N}(x|0, (4a)^{-1})$. Similar to Seeger et al. (2008), consider $\hat{k}_l$ w.r.t. $\mu(\boldsymbol{x}) = \prod_{d=1}^{D} \mu([\boldsymbol{x}]_d) = \prod_{d=1}^{D} \mathcal{N}([\boldsymbol{x}]_d|0, (4a)^{-1})$, then eq. (14) gives us

$$
\begin{aligned}
exp\left(-[\Lambda]_d([\boldsymbol{x}]_d - [\boldsymbol{x}']_d)^2\right) &= \sum_{s=1}^{\infty} b_{d,s} \Psi_{d,s}(\boldsymbol{x}) \Psi^*_{d,s}(\boldsymbol{x}') \\
\text{eq. (22)} \ \Rightarrow \tilde{k}_l(\boldsymbol{x}, \boldsymbol{x}') &= \prod_{d=1}^{D} \sum_{s=1}^{\infty} b_{d,s} \Psi_{d,s}(\boldsymbol{x}) \Psi^*_{d,s}(\boldsymbol{x}') \\
&= \sum_{s_1, \ldots, s_d \geq 1} \left(\prod_{d=1}^{D} b_{d,s_d} \prod_{d=1}^{D} \Psi_{d,s_d}(\boldsymbol{x}) \Psi^*_{d,s_d}(\boldsymbol{x}') \right)
\end{aligned}
$$

where $1 \leq s_d \leq S$. Existence is guaranteed from Mercer's theorem. Eq. (10) tells us that $b_{d,s_d} \leq \mathcal{O}(\tilde{b}_d^{s_d})$ for some $0 < \tilde{b}_d < 1$. Notice that each $\tilde{b}_d$ depends on $[\Lambda]_d$. We insert the index l back and let $B_l = max_d \{\tilde{b}_d\}$. This further gives us

$$
\prod_{d=1}^{D} b_{d,s_d} \leq \mathcal{O}(\prod_{d=1}^{D} \tilde{b}_d^{s_d}) \leq \mathcal{O}(B_l^{s_1 + \ldots + s_D}) \tag{23}
$$

Rank the bound of eigenvalues with different combinations of $\{s_1, ..., s_D\}$ and we start following what was done in Appendix II of Seeger et al. (2008) from this point. The number of possibilities of $s_1 + ... + s_D = R + D - 1$ with $s_d \geq 1$ and a chosen integer $R \geq 1$ are $\binom{R+D-2}{D-1}$, so from eq. (23) we know the first eigenvalue (as $\binom{D-1}{D-1} = 1$) is bounded by $\mathcal{O}(B_l^D)$, the second to the $\left(1 + \binom{D}{D-1}\right)$-th eigenvalues are bounded by $\mathcal{O}(B_l^{D+1})$, ..., the $\left(1 + \sum_{q=1}^{R-1} \binom{q+D-2}{D-1}\right)$-th to $\left(\sum_{q=1}^{R} \binom{q+D-2}{D-1}\right)$-th eigenvalues are bounded by $\mathcal{O}(B_l^{R+D-1})$. With fixed D, B_l^{D-1} is absorbed by $\mathcal{O}$, so $\mathcal{O}(B_l^{R+D-1}) = \mathcal{O}(B_l^R)$. Recall that $0 < B_l < 1$ and $D \in \mathbb{N}$.

Apply Pascal's rule recursively (Hockey-stick identity), we have $\sum_{q=1}^{R} \binom{q+D-2}{D-1} = \binom{R+D-1}{D}$, and thus the $\binom{R+D-1}{D}$-th eigenvalue of $\tilde{k}_l$ is bounded by $\mathcal{O}(B_l^R)$. Then, $\binom{R+D-1}{D} = \frac{R+D-1}{D} \frac{R+D-2}{D-1} \ldots \frac{R+1}{2} R \leq R^D$ and the fact that the eigenvalues are ranked imply the R-th eigenvalue is smaller than or equal to the $\binom{R+D-1}{D-1}^{1/D}$-th eigenvalue which is bounded by $\mathcal{O}(B_l^{R^{1/D}})$.

Thus, despite different lengthscale on individual dimension of input variables, we again get eigenvalues bound $\lambda_{l,j} \leq \mathcal{O}\left(B_l^{j^{1/D}}\right)$ for kernel $\tilde{k}_l$ w.r.t. Gaussian distribution, which has the same decay rate w.r.t. uniform distribution up to some constant factor according to Srinivas et al. (2012).

3. **Follow section E.2.3**

The same as section E.2.3, we consider a discretization of $\mathcal{X}$ s.t. condition (18) holds, apply lemma 6 and theorem 8 of Srinivas et al. (2012). Then we obtain the same bound $\mathcal{O}(\frac{(logn)^{D+1}}{n})$. See eq. (21).

F Proof of theorem 3

This proof is extended from the proof of theorem 3 of Zimmer et al. (2018).

F.1 Proof of theorem 3 - step 1: mutual information unchanged

First notice that lemma 7 is obtained by applying GP prior and chain rule of differential entropy, both of which are independent of which sets the data are drawn from. Therefore, if we let $\{\boldsymbol{x}_i \in S_i\}_{i=1}^{N}, \{(\boldsymbol{x}_{n_i}, p_i) \in S_i \times \{1, ..., P\}\}_{i=1}^{N_{sum}}$ denote the optimal data queried from safe active learning criterion eq. 9, then the followings still hold:

$$I\left(\boldsymbol{Y}_\phi, \{f_{p_k}(\boldsymbol{x}_{n_k})\}_{k=1}^{N_{sum}}\right) = \frac{1}{2}\sum_{k=1}^{N_{sum}} log\left(1 + \frac{1}{\sigma_{p_k}^2}\Sigma_{k-1}(\boldsymbol{x}_{n_k}, p_k)\right),$$

$$I\left(\boldsymbol{Y}, \{\boldsymbol{f}(\boldsymbol{x}_k)\}_{k=1}^{N}\right) = \frac{1}{2}\sum_{k=1}^{N} log\left(\left|I_P + diag(\{\sigma_i^2\}_{i=1}^{P})^{-1}\Sigma_{k-1}(\boldsymbol{x}_k)\right|\right).$$

F.2 Proof of theorem 3 - step 2: bound of predictive uncertainty still holds

We note that $S_i \subseteq \mathcal{X}$ is the safe regions at iteration i determined by the safe model, $\hat{\boldsymbol{x}}_i, \boldsymbol{x}_i \in S_i$, and $(\hat{\boldsymbol{x}}_{n_i}, \hat{p}_i), (\boldsymbol{x}_{n_i}, p_i) \in S_i \times \{1, ..., P\}$. We further know from our safe query criterion that

$$\Sigma_{k-1}(\hat{\boldsymbol{x}}_{n_k}, \hat{p}_k) \leq \Sigma_{k-1}(\boldsymbol{x}_{n_k}, p_k),$$
$$|\Sigma_{n-1}(\hat{\boldsymbol{x}}_n)| \leq |\Sigma_{n-1}(\boldsymbol{x}_n)|.$$

Then following the same procedure as Proof of lemma 1, we obtained the same inequality

$$\frac{1}{N_{sum}}\sum_{k=1}^{N_{sum}} \Sigma_{k-1}(\hat{\boldsymbol{x}}_{n_k}, \hat{p}_k) \leq \mathcal{O}\left(\frac{1}{N_{sum}} I\left(\boldsymbol{Y}_\phi, \{f_{p_k}(\boldsymbol{x}_{n_k})\}_{k=1}^{N_{sum}}\right)\right),$$

$$\frac{1}{N}\sum_{n=1}^{N} |\Sigma_{n-1}(\hat{\boldsymbol{x}}_n)| \leq \mathcal{O}\left(\frac{1}{N} I\left(\boldsymbol{Y}, \{\boldsymbol{f}(\boldsymbol{x}_n)\}_{n=1}^{N}\right)\right).$$

Keep in mind that the GP prior is defined from the original data space $\mathcal{X}$.

F.3 Proof of theorem 3 - step 3: bound mutual information and maximum information gain

The same as how we just proved theorem 2 (section E), we apply Fischer's inequality and theorems in Srinivas et al. (2012), and then we obtain the convergence guarantee again.

Notice that $\gamma_p^{N_p} = max_{\mathcal{D} \subseteq \mathcal{Y}_p(\mathcal{X}):|\mathcal{D}|=N_p} I(\mathcal{D}, f_p)$ and $S_i \subseteq \mathcal{X}$ for each i.

G Extended Theoretical Result

As the second multi-output model, we consider the convolution processes (Higdon, 2002; Álvarez and Lawrence, 2011). With the same latent GPs, $\boldsymbol{g} : \mathbb{R}^D \to \mathbb{R}^L$(see 3.1), and additionally mappings, $G : \mathbb{R}^D \to \mathbb{R}^{P \times L}$, that act as a smoothing kernel. The model becomes

$$\boldsymbol{f}(\boldsymbol{x}) = \int G(\boldsymbol{x} - \boldsymbol{z})\boldsymbol{g}(\boldsymbol{z})d\boldsymbol{z}.$$

The covariance $cov(f_p(\boldsymbol{x}), f_{p'}(\boldsymbol{x}'))$ here is

$$\sum_{l=1}^{L}\int\int G_{p,l}(\boldsymbol{x} - \boldsymbol{z})G_{p',l}(\boldsymbol{x}' - \boldsymbol{z}')k_l(\boldsymbol{x}, \boldsymbol{x}')d\boldsymbol{z}d\boldsymbol{z}'.$$

The smoothing kernel G is usually selected such that this integral in the covariance function is analytically tractable. We show that the convergence guarantee we previously got in section 4 also exists for a convolution process.

Theorem 9 *We use n as the unified expression of N and N_{sum}. Let $\{\hat{\boldsymbol{x}}_{\boldsymbol{i}} \in S_i\}_{i=1}^n$ be n arbitrary inputs drawn from iteration-dependent safe regions $S_i \subseteq \mathcal{X}$, in a partial output setting let $\{\hat{p}_i\}_{i=1}^n$ be n arbitrary output component indices. Let $\{\Sigma_{k-1}(\hat{\boldsymbol{x}}_k), \Sigma_{k-1}(\hat{\boldsymbol{x}}_k, \hat{p}_k)\}$ be the predictive (co)variance of $\hat{\boldsymbol{x}}_{\boldsymbol{k}}$ conditioning on $k-1$ training data queried with maximal determinant or entropy under safety constraint (eq. 9).*

If $\int\int |G_{p,l}(\boldsymbol{x}-\boldsymbol{z})G_{p,l}(\boldsymbol{x'}-\boldsymbol{z'})|d\boldsymbol{z'}d\boldsymbol{z}$ and $k_l(\cdot,\cdot)$ are bounded for all p and l, hyperparameters $\boldsymbol{\theta}$ are fixed, and $cov(f_p(\cdot), f_{p'}(\cdot)) \leq 1$, then

$$\frac{1}{n}\sum_{k=1}^{n} |\Sigma_{k-1}(\hat{\boldsymbol{x}}_{\boldsymbol{k}})|,\ \frac{1}{n}\sum_{k=1}^{n} \Sigma_{k-1}(\hat{\boldsymbol{x}}_{\boldsymbol{k}}, \hat{p}_k) \leq \mathcal{O}\left(\frac{1}{n}\sum_{p=1}^{P} \gamma_p^{N_p}\right),$$

where $\gamma_p^{N_p} = max_{\mathcal{D}\subseteq\mathcal{Y}_p(\mathcal{X}):|\mathcal{D}|=N_p} I(\mathcal{D}, f_p)$ is the maximum information gain of the current GP f_p on $\mathcal{X}$.

If we furthermore assume $G_{p,l}(\boldsymbol{z}) = W_{p,l}\mathcal{N}(\boldsymbol{z}|\boldsymbol{0}, A_p^{-1})$ and $k_l(\boldsymbol{z},\boldsymbol{z'}) = c_l\mathcal{N}(\boldsymbol{z}-\boldsymbol{z'}|\boldsymbol{0}, \Lambda_l^{-1})$ where A_p and Λ_l are positive definite Hermitian matrices, then $\frac{1}{n}\sum_{p=1}^{P}\gamma_p^{N_p} \leq \mathcal{O}\left(\frac{(logn)^{D+1}}{n}\right)$.

Given Gaussian smoothing kernels G and Gaussian latent kernels, a closed-form expression of the MO covariance function is provided in Álvarez and Lawrence (2011) (see also eq. (25)). The idea of the proof is identical as for the LMC with only minor differences that we detail out in section H.

H Proof of theorem 9

H.1 Proof of theorem 9 - bound of uncertainty

Let $\hat{w}_{p,l}$ denote the bounds of $\int\int |G_{p,l}(\boldsymbol{x}-\boldsymbol{z})G_{p,l}(\boldsymbol{x'}-\boldsymbol{z'})|d\boldsymbol{z'}d\boldsymbol{z}$ for all p and l, and let $\hat{w} = max_{p,l}\{\hat{w}_{p,l}\}$. Let $\hat{v}$ denote the bound of $k_l(\boldsymbol{x},\boldsymbol{x'})$, i.e. $0 \leq k_l(\boldsymbol{x},\boldsymbol{x'}) \leq \hat{v}$ for all l. Notice that $|k_l(\boldsymbol{x},\boldsymbol{x'})| = k_l(\boldsymbol{x},\boldsymbol{x'})$.

$$\begin{aligned} cov(f_p(\boldsymbol{x}), f_p(\boldsymbol{x'})) &= \sum_{l=1}^{L}\int\int G_{p,l}(\boldsymbol{x}-\boldsymbol{z})G_{p',l}(\boldsymbol{x'}-\boldsymbol{z'})k_l(\boldsymbol{x},\boldsymbol{x'})d\boldsymbol{z}d\boldsymbol{z'} \\ &\leq \sum_{l=1}^{L}\int\int |G_{p,l}(\boldsymbol{x}-\boldsymbol{z})G_{p,l}(\boldsymbol{x'}-\boldsymbol{z'})||k_l(\boldsymbol{x},\boldsymbol{x'})|d\boldsymbol{z'}d\boldsymbol{z} \\ &\leq \sum_{l=1}^{L}\hat{w}k_l(\boldsymbol{x},\boldsymbol{x'}) \\ &\leq \sum_{l=1}^{L}\hat{w}\hat{v} \\ &= L\hat{w}\hat{v}. \end{aligned} \tag{24}$$

Now we can follow the proof of lemma 1 (section D). Notice here that lemma 7, which is used in section D to obtain the mutual information term, is independent of kernel function. Set

$$(a,b) = \left(\frac{1}{\sigma_{p_k}}\Sigma_{k-1}(\cdot,\cdot), \frac{L\hat{w}\hat{v}}{\sigma_{p_k}}\right) \text{ or } \left(\frac{1}{\prod_{p=1}^{P}\sigma_p^2}|\Sigma_{k-1}(\cdot)|, \frac{L\hat{w}\hat{v}}{\prod_{p=1}^{P}\sigma_p^2}\right),$$

then $\frac{a}{\log(1+a)} \leq \frac{b}{\log(1+b)}$, so eq. (15) (16) (17) give us

$$\begin{aligned}
\frac{1}{N_{sum}} \sum_{k=1}^{N_{sum}} \Sigma_{k-1}(\cdot,\cdot) &\leq \mathcal{O}\left(\frac{1}{N_{sum}} I\left(\boldsymbol{Y}_\phi, \{f_{p_k}(\boldsymbol{x}_{n_k})\}_{k=1}^{N_{sum}}\right)\right), \\
&\leq \mathcal{O}\left(\frac{1}{N_{sum}} \sum_{p=1}^{P} \gamma_p^{N_p}\right), \\
\frac{1}{N} \sum_{n=1}^{N} |\Sigma_{n-1}(\cdot)| &\leq \mathcal{O}\left(\frac{1}{N} I\left(\boldsymbol{Y}, \{\boldsymbol{f}(\boldsymbol{x}_n)\}_{n=1}^{N}\right)\right) \\
&\leq \mathcal{O}\left(\frac{1}{N} \sum_{p=1}^{P} \gamma_p^{N_p}\right), \text{ here } N_p = N.
\end{aligned}$$

H.2 Proof of theorem 9 - bound $\gamma_p^{N_p}$ for the given kernel

If $G_{p,l}(\boldsymbol{z}) = W_{p,l}\mathcal{N}(\boldsymbol{z}|\boldsymbol{0}, A_p^{-1})$ and $k_l(\boldsymbol{z}, \boldsymbol{z}') \propto \mathcal{N}(\boldsymbol{z} - \boldsymbol{z}'|\boldsymbol{0}, \Lambda_l^{-1})$, then from Álvarez and Lawrence (2011) we have

$$cov(f_p(\boldsymbol{x}), f_{p'}(\boldsymbol{x}')) = \sum_{l=1}^{L} W_{p,l} W_{p',l} c_l \mathcal{N}\left(\boldsymbol{x} - \boldsymbol{x}'|A_p^{-1} + A_{p'}^{-1} + \Lambda_l^{-1}\right) \tag{25}$$

$$\propto \sum_{l=1}^{L} c_{p,p',l} \mathcal{N}\left(\boldsymbol{x} - \boldsymbol{x}'|\boldsymbol{0}, A_p^{-1} + A_{p'}^{-1} + \Lambda_l^{-1}\right), \tag{26}$$

where each of $c_{p,p',l}$ is a scalar parameter. Here the kernel is a sum of latent kernels dependent not only of l but also of p (and thus this model provides more flexibility than LMC).

For each p, we are actually dealing with $\gamma_p^{N_p} = max_{\mathcal{D} \subseteq \mathcal{Y}_p(\mathcal{X}):|\mathcal{D}|=N_p} I(\mathcal{D}, f_p)$ where f_p is a GP with kernel $cov(f_p(\boldsymbol{x}), f_p(\boldsymbol{x}'))$, which is a weighted sum of SE kernels with matrix lengthscales. This can be seen by normalizing $\mathcal{N}\left(\boldsymbol{x} - \boldsymbol{x}'|A_p^{-1} + A_{p'}^{-1} + \Lambda_l^{-1}\right)$, which gives us

$$cov(f_p(\boldsymbol{x}), f_p(\boldsymbol{x}')) \propto \sum_{l=1}^{L} \tilde{c}_{p,p,l} exp\left((\boldsymbol{x} - \boldsymbol{x}')^T (A_p^{-1} + A_p^{-1} + \Lambda_l^{-1})^{-1} (\boldsymbol{x} - \boldsymbol{x}')\right).$$

Therefore, we follow the previous proof for SE kernels:

1. For each p and l, section E.2.4 bounds the eigenvalues of $exp\left((\boldsymbol{x} - \boldsymbol{x}')^T (A_p^{-1} + A_p^{-1} + \Lambda_l^{-1})^{-1} (\boldsymbol{x} - \boldsymbol{x}')\right)$.
2. Follow section E.2.3 to get $\gamma_p^{N_p} \leq \mathcal{O}((\log N_{sum})^{D+1})$ (eq. (21)) for the GP f_p.
3. As this bound does not depend on p, we again obtain $\frac{1}{N_{sum}} \sum_{p=1}^{P} \gamma_p^{N_p} \leq \mathcal{O}\left(\frac{(\log N_{sum})^{D+1}}{N_{sum}}\right)$.

I Experimental Details

In each experiment, we randomly select a number of data as a initial dataset. With this initial dataset, we run algorithm 1 for AL_MOGP, AL_indGPs, RS_MOGP and the no-safety reference AL_MOGP_nosafe. Therefore, in each experiment, the initial dataset is always the same for all frameworks. Notice that for RS_MOGP, we query a random point under safety constraint. AL_indGPs is equivalent to our AL_MOGP with $L = P$ and $W = I_P$, see section 3.1.

For all of our models, we use Mátern kernels with $\nu = \frac{5}{2}$ (Rasmussen and Williams, 2006) for k_l. In this paper, the models always use L equal to P.

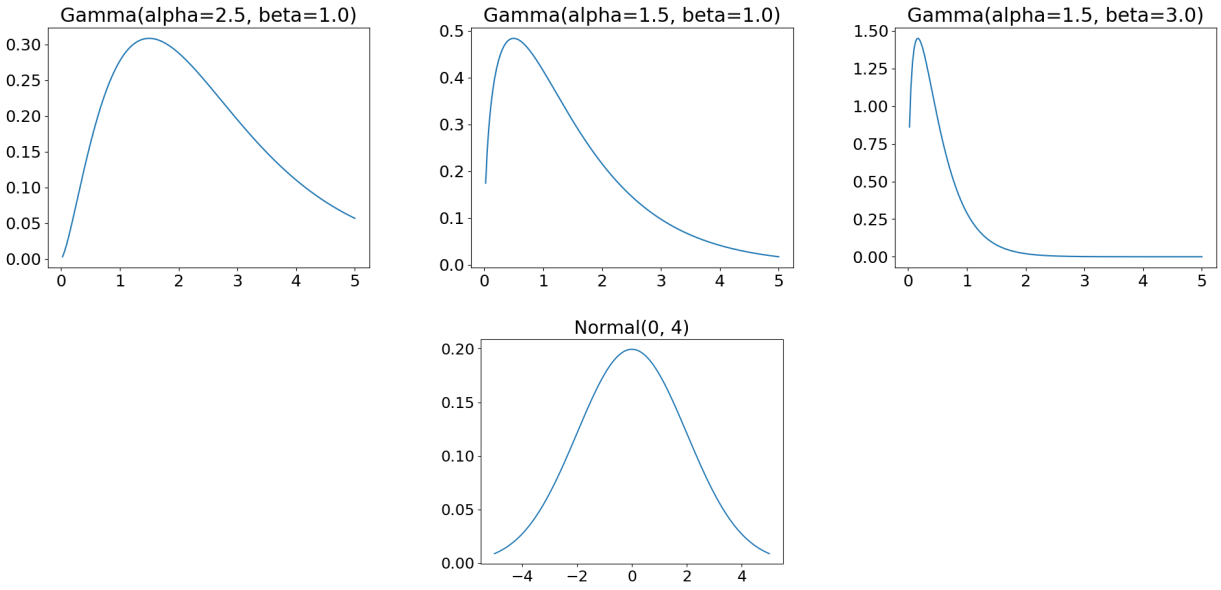

Figure 3: Probability density functions for Bayesian treatment. The distributions are priors of variance (top left) and lengthscale (top middle) of latent kernels, observation noise variances (top right) and kernel weight W_{pl} (bottom). X-axis is the value of each random variable and y-axis is the probability density.

I.1 Inference with Hyperparameters

The hyperparameters, i.e. kernel variances, kernel lengthscales and observation noise variance(s), are denoted jointly by $\boldsymbol{\theta}$.

Type II maximum likelihood estimation The log marginal likelihoods are

$$\mathcal{L}(\boldsymbol{\theta}, \mathcal{D}) = log\mathcal{N}(Y|\mathbf{0}, K_{NN} + \sigma^2 I_N) \text{ and} \tag{27}$$

$$\mathcal{L}(\boldsymbol{\theta}, \mathcal{D}) = \log \mathcal{N}\left(\boldsymbol{Y}_\phi | \mathbf{0}, \Omega_{N_{sum}N_{sum}} + \text{diag}(\{\sigma^2_{p_k}\}_{k=1}^{N_{sum}})\right) \tag{28}$$

for GP regression and for MOGP regression, respectively. Here $K_{NN} + \sigma^2 I_N$ and $\Omega_{N_{sum}N_{sum}} + \text{diag}(\{\sigma^2_{p_k}\}_{k=1}^{N_{sum}})$ are functions of the hyperparameters $\boldsymbol{\theta}$ (see also eq. (1)-(2), eq. (3)-(6)). Computing the likelihoods have complexities $\mathcal{O}(N^3)$ and $\mathcal{O}(N^3_{sum})$ as the inversion of the covariance matrices is required.

The hyperparameters

$$\hat{\theta} = argmax_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}, \mathcal{D})$$

can be obtained by applying gradient based methods. Then the predictions can be simply done by substituting $\hat{\theta}$ into the models and apply eq.(1)-(2) for standard GP regression and with eq. (3)-(6) for MOGP regression.

Bayesian treatment-theoretics

To perform a Bayesian treatment, we first assign prior distributions over the hyperparameters, i.e. $p(\boldsymbol{\theta})$, apply Bayes rule to $p(\boldsymbol{\theta})$ and $\mathcal{L}(\boldsymbol{\theta}, \mathcal{D})$ to obtain $p(\boldsymbol{\theta}|\mathcal{D})$, and then the prediction becomes:

$$p(\boldsymbol{f}(\boldsymbol{x}_*)|\boldsymbol{x}_*, \mathcal{D}) = \int p(\boldsymbol{f}(\boldsymbol{x}_*)|\boldsymbol{x}_*, \mathcal{D}, \boldsymbol{\theta})\, p(\boldsymbol{\theta}|\mathcal{D})\, d\boldsymbol{\theta}. \tag{29}$$

Here $p(\boldsymbol{f}(\boldsymbol{x}_*)|\boldsymbol{x}_*, \mathcal{D}, \boldsymbol{\theta})$ is the GP posterior given hyperparameters (see eq. (1)-(2), eq. (3)-(6)). Notice that the integral is intractable, and we either need to perform approximate inference (Titsias and Lázaro-Gredilla,

2014) or resort to Monte Carlo sampling. In our work, we apply the latter and approximate eq. (29) by drawing samples from the posterior

$$p(\boldsymbol{f}(\boldsymbol{x}_*)|\boldsymbol{x}_*, \mathcal{D}) \approx \frac{1}{|\{\hat{\theta}\}|} \sum_{\hat{\theta}} p\left(\boldsymbol{f}(\boldsymbol{x}_*)|\boldsymbol{x}_*, \mathcal{D}, \hat{\theta}\right), \tag{30}$$

where $\hat{\theta}$ are drawn from $p(\boldsymbol{\theta}|\mathcal{D}) \propto \mathcal{L}(\boldsymbol{\theta}, \mathcal{D})p(\boldsymbol{\theta})$.

Assuming that different hyperparameters are independent, $p(\boldsymbol{\theta})$ is the product of priors of individual hyperparameters. The priors of hyperparameters are Gamma and Normal distributions, which are also shown in figure 3:

- $\Gamma(\alpha = 2.5, \beta = 1.0)$ for latent kernels $k_l(\cdot)$, variance,
- $\Gamma(1.5, 1.0)$ for latent kernels $k_l(\cdot)$, lengthscale,
- $\Gamma(1.5, 3)$ for noise variances σ_p^2 and
- $\mathcal{N}(0, 2^2)$ for W_{pl}.

Gamma priors are selected for positive parameters. $\alpha = 1.5$ and $\alpha = 2.5$ would push the distribution mean further from 0. As all the variances are assumed to be bounded and the datasets are assumed normalized, the distributions should also be not too far away from 0. For observation noise variances, we use larger β to encourage smaller values. For the kernel variances, we use larger α (2.5) to encourage large uncertainty, which should be generally true without observing data. Lengthscales of the kernel can be any value greater than 0, and the effect of different prior setting does not seem very obvious (experiment not shown). The kernel variances are still weighted by W_{pl} in the model. Because W_{pl} are bounded and should be symmetric to 0, we place a normal distribution centering around 0.

In our initial experiments, we tried few different prior parameters, but the effect did not seem obvious. We also use the same hyperpriors over all experiments. For the safety model, a large kernel variance ensure that the probabilistic safety condition is difficult to achieve. In this case, the model can only have high confidence with enough observations, and this is desired for the safety model. The prior for safety model could also be set according to the safety threshold. For example, if it is safe to have safety value greater than 1, one could consider a prior with mean larger than 1 or even 2 for kernel variances, or, in addition, adjust the GP to non-zero mean and the prior for GP mean could be set to encourage values centering around 2.

Bayesian treatment-implementation

In this work, we use Hamiltonian Monte Carlo (HMC) (Betancourt, 2018; Brooks et al., 2011) as our sampling method (for approximation eq. (30)). We always use 100 hyperparameter vectors for each inference. We pick 1 sample out of 20 to ensure the samples are sufficiently independent, and we abandon the first 300 samples to ensure all the samples actually lie on the target distribution. Therefore, for each inference, we sample 100 hyperparameter vectors out of a chain of 2300 samples.

For a GP model, sampling T_θ hyperparameters has complexity $\mathcal{O}(T_\theta) * \mathcal{O}(\mathcal{L}(\boldsymbol{\theta})) = \mathcal{O}(T_\theta * N_{sum}^3)$, where $\mathcal{O}(T_\theta)$ absorbs the sampler's setting into a constant (step of the sampler, acceptance rate etc, see Brooks et al. (2011)). The datasets we use are however small and HMC has a quite good acceptance rate (roughly 0.7) for our model, so this method is not too slow in practice.

For performing inference with HMC method, as making predictions with different hyperparameter sets (and for different points) can be done in parallel, a Bayesian treatment does not necessarily increase the inference time.

The HMC is implemented with tensorflow_probability (tfp). The samples are generated with

tfp.mcmc.sample_chain:

```
samples, _ =tfp.mcmc.sample_chain(
            num_results=100, num_burnin_steps=300, num_steps_between_results=20,
            current_state=helper.current_state,
            kernel=tfp.mcmc.SimpleStepSizeAdaptation(
                tfp.mcmc.HamiltonianMonteCarlo(
                    target_log_prob_fn=helper.target_log_prob_fn,
                    num_leapfrog_steps=10, step_size=0.01
                    ),
                num_adaptation_steps=int(0.3*300),
                target_accept_prob=f64(0.75),
                adaptation_rate=0.1
                ),
            trace_fn=lambda _, pkr: pkr.inner_results.is_accepted
            )
  helper =gpflow.optimizers.SamplingHelper(
            log_posterior_density (eq. 28),
            model.trainable_parameters
            ).
```

I.2 Entropy computation for HMC

Given a random variable y, entropy

$$H(y) = -\int p(y) \log p(y) dy.$$

With the HMC approximation for Bayesian treatment (eq. (30)), the entropy shown as follows is intractable, where p in brackets is the output index while p out of brackets is probability

$$H(\boldsymbol{f}(\boldsymbol{x}_*), p) = -\int \frac{1}{|\{\hat{\theta}\}|} \sum_{\hat{\theta}} p\left(\boldsymbol{f}(\boldsymbol{x}_*), p | \boldsymbol{x}_*, \mathcal{D}, \hat{\theta}\right) \log \left(\frac{1}{|\{\hat{\theta}\}|} \sum_{\hat{\theta}} p\left(\boldsymbol{f}(\boldsymbol{x}_*), p | \boldsymbol{x}_*, \mathcal{D}, \hat{\theta}\right) \right) d\boldsymbol{f}(\boldsymbol{x}_*).$$

To estimate the entropy efficiently, we use a Gaussian mixture approximation

$$\frac{1}{|\{\hat{\theta}\}|} \sum_{\hat{\theta}} p_* \left(\boldsymbol{f}(\boldsymbol{x}_*) | \boldsymbol{x}_*, \mathcal{D}, \hat{\theta}\right) \approx \mathcal{N}(\boldsymbol{f}(\boldsymbol{x}_*) | \mu_{HMC}(\boldsymbol{x}_*, p_*), \Sigma_{HMC}(\boldsymbol{x}_*, p_*)),$$

$$\mu_{HMC}(\boldsymbol{x}_*, p_*) = \frac{1}{|\{\hat{\theta}\}|} \sum_{\hat{\theta}} \mu_{\hat{\theta}}(\boldsymbol{x}_*, p_*)$$

$$\Sigma_{HMC}(\boldsymbol{x}_*, p_*) = \frac{1}{|\{\hat{\theta}\}|} \sum_{\hat{\theta}} \left(\Sigma_{\hat{\theta}}(\boldsymbol{x}_*, p_*) + \mu_{\hat{\theta}}(\boldsymbol{x}_*, p_*) \mu_{\hat{\theta}}(\boldsymbol{x}_*, p_*)^T\right) - \mu_{HMC}(\boldsymbol{x}_*, p_*) \mu_{HMC}(\boldsymbol{x}_*, p_*)^T,$$

which then results in a tractable entropy $\propto \log(|\Sigma_{HMC}(\boldsymbol{x}_*, p_*)|)$.

I.3 Experiments with different datasets

For all of the datasets, we repeat the experiments 30 times and set the safety probability threshold to $\delta = 0.05$. We averaged the RMSE values over the different output components.

Dataset: simulation with sin & sigmoid

The data are simulated as follows

$$\begin{aligned}
\boldsymbol{f}_{true}(x) &= \begin{pmatrix} sin(10x) + \frac{1}{1+exp(-2x)} \\ sin(10x) - \frac{1}{1+exp(-2x)} \end{pmatrix}, \\
h_{true}(x) &= exp(-(x-0.1)^2/2), \\
\boldsymbol{y} &\sim \mathcal{N}\left(\boldsymbol{f}_{true}(x), \begin{pmatrix} 0.4^2 & 0 \\ 0 & 0.4^2 \end{pmatrix}\right), \\
z &\sim \mathcal{N}\left(\, h_{true}(x), 0.05^2\right).
\end{aligned}$$

We say x is safe if $h(x) > 0.7$ and set the allowed risk to $\delta = 0.05$ (see section 3.3). Therefore, in order to be executed, x needs to fullfill $p(h(x) > 0.7) > 0.95$. The true safety values $h_{true}(x) > 0.7$ are thereby equivalent to the input interval, $x \in \left(0.1 - \sqrt{-2\log(0.7)}, 0.1 + \sqrt{-2\log(0.7)}\right)$ (roughly $(-0.74, 0.94)$). Interval for exploration is set to $x \in [-2, 2]$.

Figure 4 shows the models, data and entropy of the 3 frameworks at the 15th iteration. When the output is partially observed which is how the experiments are done, the entropy are the concatenation of $H(\boldsymbol{f}(\boldsymbol{x}_*), p)$ for all p, corresponding to the blue and orange entropy curves in figure 4. The corresponding safety predictions are in figure 5.

In this experiment we start with $N_{sum} = 12$ (6 for each output). The RMSE and log likelihood are evaluated on ground truth $\boldsymbol{f}_{true}$ of a test set drawn from the safe region.

Dataset: MOGP samples

We first fix a seed (= 123), specify the number of experiments ($E = 30$) and the number of data points in each experiment ($N_{training} + N_{test} = 2000 + 500$), and specify the dimension of input ($D = 2$), output ($P = 4$), and the number of latent GPs ($L = 3$).

The goal is to have input $\boldsymbol{X} \subseteq \mathbb{R}^D$, output $\boldsymbol{Y}_1, ..., \boldsymbol{Y}_E \subseteq \mathbb{R}^P$ and safety values $\boldsymbol{Z}_1, ..., \boldsymbol{Z}_E \subseteq \mathbb{R}$.

This can be done by drawing samples from a given MOGP and GP. We draw the samples as follows, where the sample interval and kernels can all be replaced, as long as the bounded conditions are fulfilled:

1. Input $\boldsymbol{X} \subseteq \mathbb{R}^D$: draw $(N_{training} + N_{test}) \times D$ samples uniformly from interval $[-2, 2)$, remove duplicate sample vectors, draw more samples if there were duplicate samples being removed until having $N_{training} + N_{test}$ samples, and then shuffle all vectors to preserve randomness.

2. Prepare kernels for (MO)GPs: draw samples uniformly from interval $[0.01, 1)$ for $L + 1$ squared exponential kernels ($k_1(\cdot), ..., k_L(\cdot)$ for samples $\boldsymbol{Y}$ and $k_z(\cdot)$ for samples $\boldsymbol{Z}$, each with a variance and a scalar lengthscale). To normalize the data, we fix the variances of $k_1(\cdot), ..., k_L(\cdot)$ to 1, and to ensure a smoother safety values we fix the lengthscale of $k_z(\cdot)$ to 1. The safety values does not have to be very smooth, but it is then necessary to analyze how the experiments can start with a robust enough safety model, which is not the focus of this paper (see Schreiter et al. (2015) for safety discussion).

3. Draw latent samples and noise-free $\boldsymbol{Z}$: draw E L-dim trajectories denoted by $\boldsymbol{G}_1, ..., \boldsymbol{G}_E \subseteq \mathbb{R}^{(N_{training}+N_{test}) \times D}$, individual dimensions following $\mathcal{N}(\boldsymbol{0}, k_1(\boldsymbol{X}, \boldsymbol{X})), ..., \mathcal{N}(\boldsymbol{0}, k_L(\boldsymbol{X}, \boldsymbol{X}))$, and draw E sets of noise-free $\boldsymbol{Z}$ from $\mathcal{N}(\boldsymbol{0}, k_z(\boldsymbol{X}, \boldsymbol{X}))$.

4. Prepare $W \in \mathbb{R}^{P \times L}$ for samples $\boldsymbol{Y}$: draw P L-dim vectors from standard normal distribution, reject $\boldsymbol{0}$ vector, draw more samples if rejection happened, and normalize each vector.

5. Generate noise-free $\boldsymbol{Y}$: $\boldsymbol{Y}_e = \boldsymbol{G}_e @ W^T$, where $e = 1, ..., E$, @ is the matrix multiplication operator, and W^T is the transpose of W.

6. Add gaussian noises to Y and Z with specified noise levels $\sigma_p = 0.4$ and $\sigma_{safety} = 0.05$.

Now we have datasets $\mathcal{D}_e = (\boldsymbol{X}, \boldsymbol{Y}_e, \boldsymbol{Z}_e)$ for $e = 1, ..., E$. We can pick the first $N_{training}$ as training samples and the rest as test samples. This is equivalent to random data split because (MO)GP models are permutation invariant (i.e. data-shuffle invariant, which makes random selection the same as shuffling $\boldsymbol{X}$ at step 1) and because $\boldsymbol{X}$ are drawn randomly without being sorted.

The experiment starts with $N_{sum} = 40$ (10 for each output), and we repeat the experiment with 30 different seeds. In this dataset, the RMSE and log likelihood are evaluated on noisy test data. The noise-free data are not accessible throughout the experiment.

For all of the 30 experiments, we compute the 20%-quantile of $\boldsymbol{Z}_1 \cup ... \cup \boldsymbol{Z}_E$, denoted by $z_{0.2}$, and set the data safe when $p(h(\boldsymbol{x}) > z_{0.2}) > 0.95$.

EngE dataset

This dataset has 8 output channels including 2 temperature channels and 6 chemical substances emitted from a gasoline engine. All of the data were measured from a warm engine and were split into training and test datasets. The 2 temperature channels are highly correlated with Pearson correlation coefficient close to 0.98. The datasets were normalized such that each input or output channel of the training set has mean 0 and variance 1 with negligible numerical error. Therefore, it is suitable for a pool-based active learning algorithm. In addition, the engine is a dynamic system, i.e. outputs depends on inputs of not only current time points but also past histories, and a sequence of data is used together in order to make accurate predictions. In order to reflect the dynamic aspects, the dataset is available with a history considering nonlinear exogenous (NX) structure, concatenating the relevant past points into the inputs. Inputs of this dataset have originally 5 channels (i.e. $\boldsymbol{x} \in \mathbb{R}^5$), and individual channels may have different history structures. With the history concatenation, the inputs have 14 dimensions.

The data were measured with high sampling frequency. The training set has in total around 247 thousand points, but in practice if we train a sparse MOGP model (van der Wilk et al., 2020), the performances saturate with few thousand of randomly selected data (in this case we did not consider any safety constraint, which could deteriorate the performance). Our safe AL experiment achieved a test RMSE of 0.85 with roughly 100 observations (N_{sum}) under safety consideration, while the saturation we achieved was 0.65, using much more observations and at least hundreds of inducing points leading to a larger memory requirement.

In the main experiment, we start from 48 data, 24 for HC, 24 for O2, and all 48 for the safety values. The 3 frameworks start with the same initial data. In each AL iteration, either HC or O2 is queried together with the corresponding temperature value. We use seed 123 to randomly generate 30 sets of initial data, and perform 30 experiments with these initial sets, each with an individual seed (affect the random selection benchmark). Both the RMSE and test log likelihood show that our methods perform better than the competitors (figure 6).

For the safety threshold z_{max}, the 80%-quantile of this temperature channel in the processed training dataset is 1.0075. We round this number to 1 as the threshold for the experiment. Notice that, 20.55% of the data is unsafe with $z_{max} = 1$. The safety constraint in this experiment is thus $p(h(\boldsymbol{x}) \leq 1.0) > 0.95$.

For some systems, it might be relatively easy and cheap to collect observations of all channels. To investigate the performance of AL_MOGP for this situation, we conduct the following ablation study.

J Ablation Study

We perform the same experiments as described above on fully observed data. In a partially observed output (POO) setup, we start from N_{sum} input points, N_{sum}/P output points for each output $\{y_{pn} | n = 1, ..., N_{sum}/P\}$, N_{sum} safety values $\{z\}$, and the safe AL proceed by querying $\{((\boldsymbol{x}_a, p), y_{pa}, z_a)\}$. In a fully observed output (FOO) setup, we start from N_{sum}/P input points, N_{sum}/P P-dimensional output points $\{\boldsymbol{y}_n | n = 1, ..., N_{sum}/P\}$, N_{sum}/P safety values $\{z\}$, and the safe AL proceed by querying $\{(\boldsymbol{x}, \boldsymbol{y}, z)\}$ (i.e. in each AL iteration, $\boldsymbol{Y}$ gain P points, notated in a POO manner, and $\boldsymbol{Z}$ gain 1 point).

We compare the RMSE of model $\boldsymbol{f}$ under POO and under FOO, given the same number of training points. We perform the experiments on the simulation datasets, and figure 7 shows that our AL_MOGP under POO is the most data-efficient. This is as expected because, in addition to MOGP's capability of correlation learning, POO provides more flexibility of exploration than FOO, given fixed number training points (or given fixed budget in

a real application). Here our datasets have similar level of uncertainty for different outputs by design. When one of the outputs has much larger level of uncertainty than the others, our acquisition function for POO might tend to query mostly from this uncertain output, which we did not investigate in detail.

Notice that with fixed number of training outputs $\boldsymbol{Y}$, the number of observed safety values is less under FOO than under POO. However, we do not compare the safety models under different setup, as our goal is to have a good safety control which is achieved by both POO (high model precisions shown in table 1 2 3) and FOO (high precisions, not shown in this paper). Schreiter et al. (2015) provides more insight in safety guarantee.

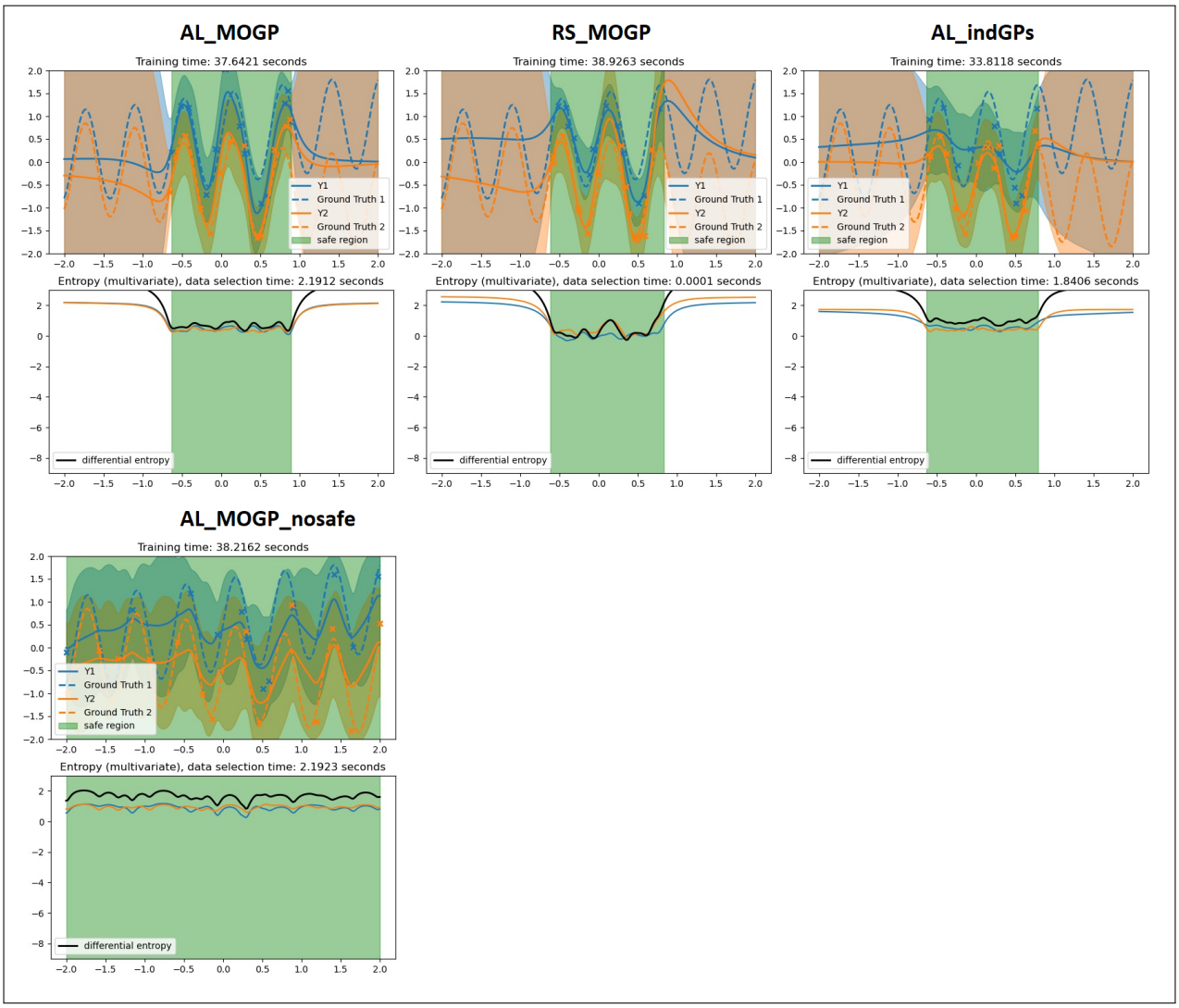

Figure 4: Different pipelines in sin & sigmoid simulation at the 15-th iteration. The plots demonstrate model predictions, ground truth and observed data (first row) as well as entropy (second row) of different frameworks. AL_MOGP_nosafe is achieved by setting the safety threshold such that it is safe everywhere. The colors indicate the 2 outputs. Entropy of full output covariance is shown with black lines. Training time here is the sampling time of Bayesian treatment (HMC).

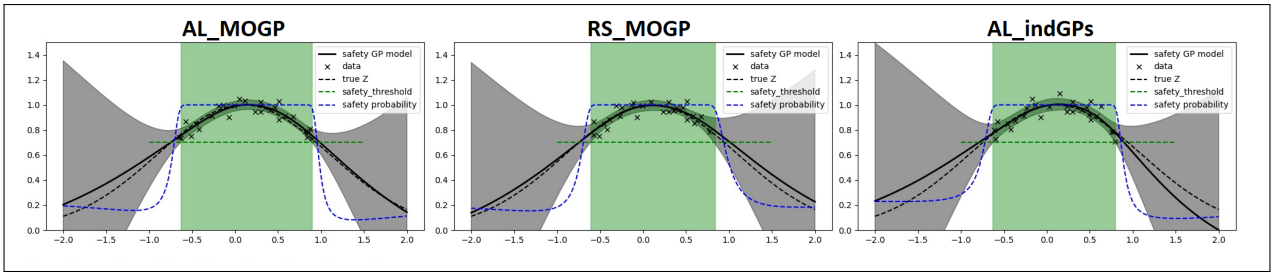

Figure 5: Safety control for different pipeline in simulation 1 at the 15-th iteration. The plots are model predictions, ground truth safety values, observed safety data and the safe probability given by the GP model.

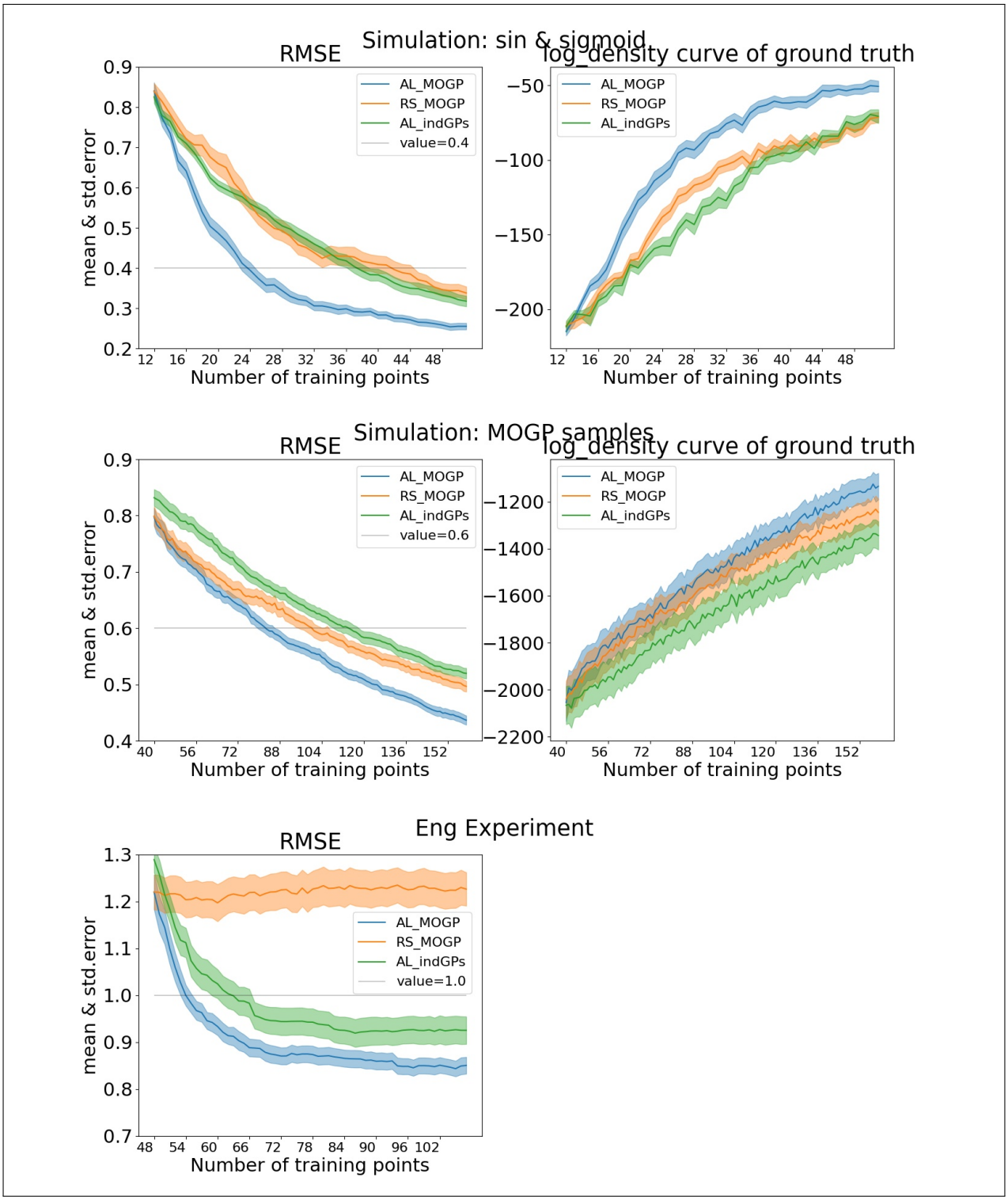

Figure 6: RMSE and log density of test data. The left column is the same as figure 1. In sin & sigmoid simulation experiment, the test data are ground truth values. The test data used for evaluation are all safe data.

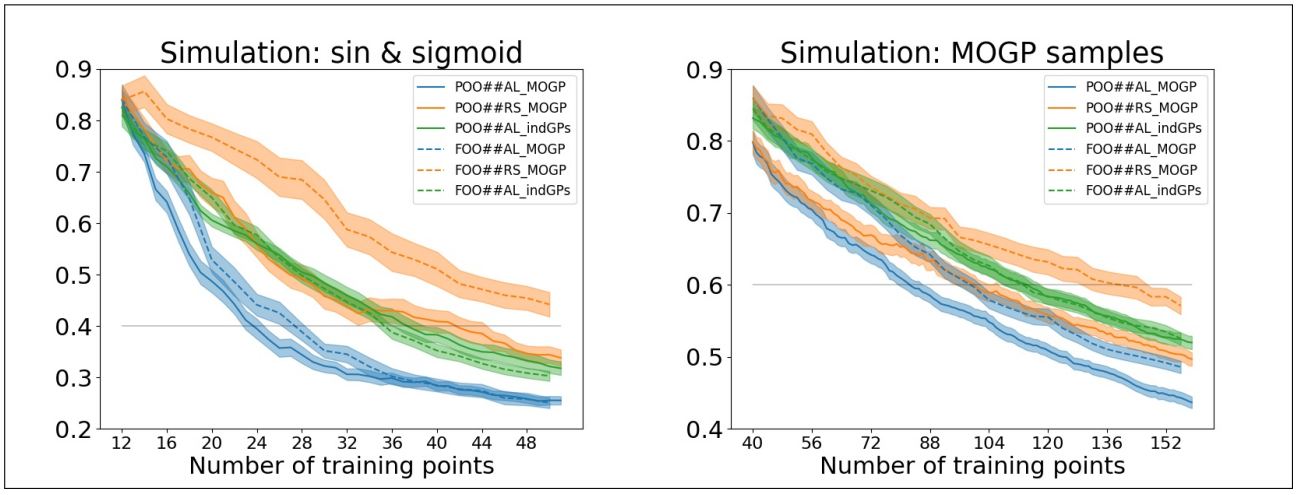

Figure 7: RMSE of different pipelines with partially observed outputs (POO) and with fully observed outputs (FOO). POO curves are identical to those shown in figure 1. Y-axis is the RMSE value and x-axis is N_{sum}.

Table 1: Toy dataset, safety model precisions

N_{sum}	**AL_MOGP**	**RS_MOGP**	**AL_indGPs**
12	0.9925 ± 0.0027	0.9930 ± 0.0019	0.9943 ± 0.0018
22	0.9895 ± 0.0027	0.9930 ± 0.0019	0.9917 ± 0.0020
32	0.9913 ± 0.0021	0.9906 ± 0.0024	0.9901 ± 0.0022
42	0.9901 ± 0.0021	0.9921 ± 0.0022	0.9893 ± 0.0023

Table 2: GP dataset, safety model precisions

N_{sum}	**AL_MOGP**	**RS_MOGP**	**AL_indGPs**
40	0.9980 ± 0.0006	0.9980 ± 0.0006	0.9979 ± 0.0007
60	0.9979 ± 0.0009	0.9978 ± 0.0005	0.9987 ± 0.0003
80	0.9983 ± 0.0003	0.9981 ± 0.0004	0.9985 ± 0.0003
100	0.9981 ± 0.0004	0.9976 ± 0.0006	0.9981 ± 0.0004
120	0.9975 ± 0.0005	0.9976 ± 0.0004	0.9978 ± 0.0005
140	0.9977 ± 0.0005	0.9975 ± 0.0004	0.9976 ± 0.0005

Table 3: EngE dataset, safety model precisions

N_{sum}	**AL_MOGP**	**RS_MOGP**	**AL_indGPs**
48	0.9841 ± 0.0020	0.9841 ± 0.0020	0.9843 ± 0.0020
58	0.9828 ± 0.0019	0.9848 ± 0.0022	0.9827 ± 0.0020
68	0.9829 ± 0.0019	0.9869 ± 0.0017	0.9835 ± 0.0017
78	0.9835 ± 0.0017	0.9876 ± 0.0014	0.9831 ± 0.0018
88	0.9830 ± 0.0018	0.9880 ± 0.0013	0.9832 ± 0.0018
98	0.9841 ± 0.0017	0.9890 ± 0.0010	0.9839 ± 0.0018

We define the precision of the safety model as the fraction of samples that are within the true safe region from all samples that are marked as probabilistically safe from the safety model. The tables report the mean and standard error over this statistic for each experimental pipeline (AL_MOGP, RS_MOGP, AL_indGPs) and for each dataset. On both datasets, all methods fulfill the safety criterion after the first iteration. Table 1 corresponds to the toy dataset and table 3 to the EngE dataset.