Data-Optimized Spatial Field Predictions for Robotic Adaptive Sampling: A Gaussian Process Approach

Zachary Nathan
Dartmouth College, zachary.nathan.22@dartmouth.edu

Follow this and additional works at: https://digitalcommons.dartmouth.edu/cs_senior_theses

Part of the Applied Statistics Commons, Artificial Intelligence and Robotics Commons, and the Data Science Commons

Recommended Citation
https://digitalcommons.dartmouth.edu/cs_senior_theses/3
Abstract

We introduce a framework that combines Gaussian Process models, robotic sensor measurements, and sampling data to predict spatial fields. In this context, a spatial field refers to the distribution of a variable throughout a specific area, such as temperature or pH variations over the surface of a lake. Whereas existing methods tend to analyze only the particular field(s) of interest, our approach optimizes predictions through the effective use of all available data. We validated our framework on several datasets, showing that errors can decline by up to two-thirds through the inclusion of additional colocated measurements. In support of adaptive sampling, this framework offers recommendations for sampling locations based on different objectives. Robotic adaptive sampling is essential in various applications such as environmental monitoring, underwater exploration, and resource management. This work, by leveraging all available data and increasing the number of fields used for prediction, contributes to robotic adaptive sampling research and supports the development of more effective robotic systems.
# Contents

1 Introduction 3

2 Related work 4

3 Method 5

3.1 Pre-processing .......................... 5

3.2 Gaussian Process models .................. 6

3.2.1 Sensor field prediction .................. 6

3.2.2 Sampling field prediction .................. 9

3.3 Sampling recommendation .................. 9

4 Results 11

4.1 Real-world data .................................. 11

4.1.1 Demonstration .................................. 11

4.1.2 Field cross-validation .................. 13

4.2 Generated data .............................. 18

4.2.1 Approximate GP .............................. 18

4.2.2 Exact GP .............................. 21

5 Discussion 26

5.1 Future directions .............................. 28

6 Conclusion 29

References 29
List of Figures

1  Method: visual overview ....................................................... 5
2  Context for real-world data sources ....................................... 12
3  Real-world dataset demonstration (part 1) ............................... 15
4  Real-world dataset demonstration (part 2) ............................... 16
5  Cross-validation experimental results ..................................... 17
6  Experimental parameter visualizations .................................... 20
7  Approximate GP experimental results ..................................... 22
8  Exact GP experimental results .............................................. 25

List of Tables

1  Demonstration parameters by dataset. ..................................... 13
2  Independent variables for approximate GP experiments. .............. 19
3  Independent variables for exact GP experiments. ....................... 24

List of Algorithms

1  Bounded adaptive inducing point selection ............................... 8
2  Sampling location recommendation ....................................... 10
1 Introduction

This project introduces a novel framework for predicting spatial fields, leveraging Gaussian Process (GP) models in combination with robotic sensor measurements and sampling data. We consider sensor measurements as abundant and continuous observations along a path or throughout a space, whereas sampling data refer to sparse observations of distinct fields which are typically latent, or otherwise more difficult to measure than the former. Our proposed approach expands upon the existing frameworks by increasing the number of spatial fields utilized for prediction, allowing the model to effectively leverage all available data. This method is particularly designed to support robotic adaptive sampling, using the predicted fields to recommend locations for additional samples. These recommendations can be made according to various data-driven objectives, such as targeting the maxima or minima within fields, or the areas having the greatest or least uncertainty.

Robotic adaptive sampling is essential for various applications such as environmental monitoring, underwater exploration, and resource management [1]. It enables robots to efficiently gather data in environments where direct measurements are challenging, time-consuming, or expensive [2]. By choosing sampling locations in an adaptive manner, robots can optimize the collected data to maximize information gain while minimizing resource consumption, ultimately improving the overall understanding of the environment. While existing methods for adaptive sampling have made remarkable contributions, there remains room for improvement in aspects such as fully exploiting spatial correlations across the totality of data. Motivated by this challenge, we develop a framework to address this limitation, seeking enhancements to spatial field predictions for adaptive sampling.

Gaussian Process models serve as a robust solution for these challenges, providing a flexible, non-parametric method for regression and classification tasks. GPs offer several advantages, such as the ability to model complex relationships, provide uncertainty estimates, and incorporate prior knowledge through kernel functions [3]. In the context of robotics, GPs have been applied to various related problems including localization [4], motion planning [5], and environmental modeling [6]. Kriging, a geostatistical interpolation technique, is closely related to GPs and has been utilized in many applications including meteorology to predict spatial fields [7].

In this paper, we present a comprehensive framework that combines the strengths of GP models and robotic adaptive sampling to address some of the challenges faced by existing methods. By integrating GPs with robotic sensor measurements and sampling data, the framework efficiently predicts spatial fields, provides uncertainty estimates, and recommends optimal sampling locations based on various objectives. Through this approach, we aim to contribute to the ongoing research in robotic adaptive sampling, supporting the development of more efficient and effective robotic systems for environmental monitoring and exploration.
2 Related work

We begin with a selection of studies that introduce the underlying concepts of our project – robotic learning and adaptive sampling – while explaining their real-world significance. Taylor et al. present a survey of active learning in robotics, emphasizing the role of embodiment, movement, and physical interaction as crucial factors in the learning process [8]. Bai et al. review information-driven path planning in robotics, highlighting its significance in environmental modeling and exploration of unknown environments [9]. Adaptive sampling strategies have been extensively studied for environmental monitoring applications; Fossum et al. provide a comprehensive overview of adaptive sampling in marine robotics [10], while Caccia et al. discuss the use of autonomous surface vehicles (ASVs) for harbor water monitoring [11]. Both works highlight the significance of adaptive sampling in real-world applications.

Gaussian Process models have been widely employed for efficient mapping and exploration of spatial fields in robotics. Hüllermeier and Waegeman distinguish aleatoric and epistemic uncertainty in machine learning, offering useful context for understanding uncertainties in spatial field predictions from GP models [12]. We draw inspiration from several works wherein teams of robots, aided by GP models and other predictive algorithms, sought to optimize data collection across spatial fields. Salam and Hsieh propose a distributed strategy for robot teams to adaptively sample and track dynamic processes, creating a reduced-order model from sparse sensor measurements to estimate unknown field values [13]. Luo and Sycara introduce an approach that uses a mixture of GP models to optimize multi-robot exploration within Voronoi cells [14]. Marchant and Ramos devise a layered Bayesian optimization approach with two GPs for continuous path optimization of initially unknown phenomena, addressing the trade-offs between exploration and exploitation [15]. Manjanna et al. propose a heterogeneous multi-robot system that leverages GP modeling and frontier-based exploration for efficient mapping, serving as another valuable example of predictive algorithms that this project aims to build upon [16].

Malencia et al. present ASLaP-HR, an adaptive sampling method that maximizes information gain with respect to a latent spatial field [17]. Using sensor measurements of two observable spatial fields, they use a GP model of the inter-field correlations to predict a third, latent field. This work is the main inspiration for our proposed method, as it demonstrates the predictive power of learned correlations among colocated fields. Building upon their foundation, we strive for a predictive model that incorporates as much relevant data as possible, analyzing the extent to which additional spatial fields can further optimize GP predictions.
3 Method

Figure 1 gives a visual overview of our method, showing the flow of information from start to finish. We publish our python implementation of the method and experiments on Github, at https://github.com/ZackNathan/thesis_code.

3.1 Pre-processing

Input data for this method, including sensor measurements and samples along with their positions, are pre-processed to enable more efficient and accurate learning. First, any latitude or longitude coordinates are converted into Universal Transverse Mercator (UTM) and centered at zero, yielding $x$ and $y$ positions measured in meters. Next, if necessary, sensor measurements within a specified radius of the first and last points are discarded for the sake of cleaning; for example, our test datasets included outlier measurements from before the ASVs were launched, as well as after terminating their missions. Finally, all feature values are converted into standard scores ($z = \frac{x-\mu}{\sigma}$).
to guarantee zero-means and consistent scales before training the GP models. Standardization is reversed following prediction, such that results are returned in the same scales as inputs.

### 3.2 Gaussian Process models

The proposed method for predictive sampling uses a two-stage Gaussian Process system. We feed the sensor measurements into (i) an approximate (variational) GP that predicts the underlying fields, followed by (ii) an exact GP for the sampling field, that extrapolates from sparse sample measurements by leveraging the former predictions. This two-stage system is necessary because sensor measurements (thousands) are typically much more numerous than samples (dozens), requiring a variational model to manage the time complexity. Moreover, the approximate GP yields predictions in the specific format utilized by the exact GP to leverage their correlations. We detail these individual models below, with fig. 1 illustrating their functions within the proposed method as a whole. Both GPs are implemented using GPyTorch with Gaussian likelihood, Adam optimization [18], no priors, and a scaled Matérn covariance (kernel) where \( \nu = 3/2 \) [3].

\[
C_{3/2}(r) = \left( 1 + \frac{\sqrt{3}r}{\rho} \right) \exp \left( -\frac{\sqrt{3}r}{\rho} \right)
\]

Equation 1: Matérn covariance function with \( \nu = 3/2 \).

We chose the Matérn covariance function for its versatility as well as its relevance to spatial fields. Alternatively, the Gaussian (radial basis or squared exponential) covariance function is widely applicable, however its infinite differentiability may yield implausible results for spatial fields [19]. The Matérn kernel can be conceptualized as a generalization of the Gaussian kernel, but with finite differentiability (provided that \( \nu < \infty \)) enabling it to better capture physical processes [20]. We experimented with various kernels during the early stages of development, with Matérn (at \( \nu = 3/2 \)) yielding the best results by a significant margin. To enable output scaling in GPyTorch, we decorate the MaternKernel object with a ScaleKernel.

#### 3.2.1 Sensor field prediction

First, we use an approximate GP model with variational evidence lower bound (ELBO) loss to predict sensor fields from robotic sensor measurements [21]. Since GP regression has a time complexity of \( O(N^3) \) for a training sample of size \( N \), a sparse approximation is necessary for large datasets (our test datasets had thousands of measurements for each field). To that end, we employ a variational framework to parameterize the model using only a subset of the \( N \) data points, known as inducing
To achieve results comparable to an exact GP model, it’s necessary to strategically select \( M \) inducing points that capture the most important features of the data.

We approach this problem with a version of adaptive inducing point selection [23]. By itself, this algorithm builds the set of inducing points \( Z \) by iterating over the observations \( X \), selecting points \( x \) where \( \max(k(x, Z)) \) is less than the acceptance threshold \( \rho \). That is, an observation \( x \) becomes an inducing point when the kernel function \( k \) returns values less than \( \rho \), for \( x \) in combination with all previously selected inducing points in \( Z \). This algorithm does not specify the number of inducing points \( M \), which depends on various factors including the structure of the dataset, the kernel, and \( \rho \). Therefore, we adjust the threshold \( \rho \) while repeating the algorithm until \( M_{\text{min}} \leq M \leq M_{\text{max}} \). This modified algorithm, called bounded adaptive inducing point selection, is fully described in alg. 1. Thus, we optimally select inducing points while reducing the training time complexity to \( O(M^2N) \), where \( M \ll N \) [3]. By default, we constrain the number of inducing points between \( M_{\text{min}} = 250 \) and \( M_{\text{max}} = 300 \).

Once the approximate GP model has been trained on a field’s sensor measurements, we obtain the multivariate normal distribution containing the posterior mean and covariance of the field. We then generate predictions over the grid \( \mathcal{G} \) (defined in eq. (2)), saving the predicted mean and variance for each coordinate pair in the discretized space. These inferences are highly optimized, estimating the posterior covariance using the Lanczos algorithm, a computational method for quickly approximating the eigenvalues and eigenvectors of large matrices [24]. The results of these predictions are the sensor-measured field, with any gaps in \( \mathcal{G} \) filled in according to the GP model’s best fit, along with a parallel field of the corresponding variance at each point. Using these predicted fields, we extrapolate from sampling locations and measurements at the next step. Examples of inputs and outputs for the Approximate GP model can be seen in fig. 3, using real sensor measurements from several datasets.

\[
\mathcal{G} = X \times Y
\]

\[
X = \{x_i \mid x_i = x_{\text{min}} + i \frac{x_{\text{max}} - x_{\text{min}}}{\Delta - 1}, i = 0, 1, \ldots, \Delta - 1\} \tag{2}
\]

\[
Y = \{y_j \mid y_j = y_{\text{min}} + j \frac{y_{\text{max}} - y_{\text{min}}}{\Delta - 1}, j = 0, 1, \ldots, \Delta - 1\}
\]

Equation 2: Grid \( \mathcal{G} \) for discretized field predictions. The values \( x_{\text{max}}, x_{\text{min}}, y_{\text{max}}, y_{\text{min}} \) refer to spatial coordinates within the input data set. The grid resolution \( \Delta \) defaults to 400 for both axes.
Algorithm 1: Bounded adaptive inducing point selection

Data: observations $X$, kernel function $k$, bounds for the number of inducing points $[M_{\text{min}}, M_{\text{max}}]$

Result: inducing points $Z$

\begin{align*}
\rho &\leftarrow 0.5; \quad \text{// initial acceptance threshold} \\
\delta\rho &\leftarrow 0.1; \quad \text{// threshold adjustment amount} \\
s &\leftarrow \text{null}; \quad \text{// state boolean: true if } \|Z\| > M_{\text{max}}, \text{ false if } \|Z\| < M_{\text{min}} \\
Z &\leftarrow X; \quad \text{// if } M_{\text{min}} \leq \|X\| \leq M_{\text{max}}, \text{ we simply return } X
\end{align*}

while $\|Z\| < M_{\text{min}}$ or $\|Z\| > M_{\text{max}}$ do

\begin{align*}
Z &\leftarrow \emptyset; \\
\text{for } x \in X \text{ do} \\
&\quad \text{if } Z = \emptyset \text{ then} \\
&\quad \quad Z \leftarrow \{x\}; \quad \text{// the first } x \text{ is always selected} \\
&\quad \text{else} \\
&\quad \quad d \leftarrow \max(k(x, Z)); \quad \text{// this block is the original algorithm [23]} \\
&\quad \quad \text{if } d < \rho \text{ then} \\
&\quad \quad \quad Z \leftarrow Z \cup x; \quad \text{// subsequent } x \text{ are selected conditionally} \\
&\quad \quad \text{end} \\
&\quad \text{end} \\
\text{end} \\
\text{if } \|Z\| < M_{\text{min}} \text{ then} \\
&\quad \text{if } s = \text{true} \text{ then} \\
&\quad \quad \delta\rho \leftarrow \delta\rho/2; \quad \text{// halve the adjustment amount if previously } \|Z\| > M_{\text{max}} \\
&\quad \text{end} \\
&\quad s \leftarrow \text{false}; \\
&\quad \rho \leftarrow \rho + \delta\rho; \quad \text{// raise the threshold and try again} \\
\text{else if } \|Z\| > M_{\text{max}} \text{ then} \\
&\quad \text{if } s = \text{false} \text{ then} \\
&\quad \quad \delta\rho \leftarrow \delta\rho/2; \quad \text{// halve the adjustment amount if previously } \|Z\| < M_{\text{min}} \\
&\quad \text{end} \\
&\quad s \leftarrow \text{true}; \\
&\quad \rho \leftarrow \rho - \delta\rho; \quad \text{// lower the threshold and try again} \\
\text{end}
\end{align*}

end
3.2.2 Sampling field prediction

Using the results from the previous step – the predicted sensor fields – as input features, we move onto predicting the sampling field with an exact GP model. We train this model on sparse sampling data, with loss measured according to the exact marginal log likelihood (MLL). As opposed to the approximate GP which selects $M \ll N$ inducing points from $N$ observations for estimation, the exact GP retains the entire set of $N$ data points. Thus, an exact GP is more practical here for prediction from samples; as defined for this project, samples are necessarily much fewer in number than sensor measurements. We see that predictions improve to some extent when additional samples are given, although we observe diminishing returns for $N > 30$ samples, as shown in fig. 8b.

The observation (input) matrix for this model consists of: (i) the sampling measurement values, (ii) the points in the grid $G$ closest to these sampling locations, and (iii) the corresponding sensor field values for these locations, as predicted by the approximate GPs. In this way, we exploit all available data in order to extrapolate as accurately as possible from a limited number of samples. To predict the sampling field in areas without samples, the GP model leverages the learned covariance between them and all other spatial fields in the grid $G$; namely, the already-predicted sensor fields.

After training, we obtain the posterior multivariate normal distribution, saving the predicted mean and variance of the sampling field over the grid $G$ in a manner similar to the approximate GPs. The result is the sampled field with any gaps filled in according to the exact GP model’s best fit, with a parallel field of the corresponding variance at each point. Using these predicted fields, we recommend optimal sampling locations for various objectives at the next step.

3.3 Sampling recommendation

After predicting the sampling field using the exact GP with colocated sensor fields as features, we proceed to recommending locations within the field as targets for further sampling. These locations are selected greedily based on the given acquisition function $f$, and guaranteed to be dispersed over the field through the procedure described in alg. 2. The degree to which these locations are dispersed can be fine-tuned by adjusting the scaling factor $C$, which determines the effective radius of the update function shown in eq. (3).

Acquisition functions are featured in Bayesian optimization sampling algorithms, estimating the acquisition value of prospective samples in support of robotic path planning. Typically, they seek to maximize the expected utility or minimize the expected risk [25]; common acquisition functions include the probability of improvement, the expected improvement, and the upper confidence bound [15]. Here, the acquisition function $f$ is defined over $G$ such that the points $x$ having the highest $f(x)$
correspond to the most desirable sampling locations. We implement four base acquisition functions: maxima, minima, max-variance, and min-variance. These functions will yield the sampling field’s predicted maxima, minima, most uncertain, and least uncertain locations, respectively.

We highlight the flexibility of our recommendation framework, in that acquisition functions can be combined and transformed across various fields in pursuit of wide-ranging sampling objectives. Acquisition functions may be combined in a similar manner to GP covariance functions: adding acquisition functions behaves like OR and multiplying like AND. To illustrate the potential for combinations, we implement the combined acquisition function as maxima×min-variance, which yields only the maxima candidates predicted with high confidence. Moreover, it’s possible to incorporate other fields into these functions as well; for instance, maxima may be combined with minima to yield recommendations with high predicted temperature and low predicted pH. In principle, any arbitrary function over $G$ can act as an acquisition function (or a part of one) for these purposes.

**Algorithm 2:** Sampling location recommendation

**Data:** grid $G$, acquisition function $f$, number of points $N$, scaling factor $C$

**Result:** recommended points $R$

$R \leftarrow \{\}$;

$i \leftarrow 0$;

while $i < N$ do

$p \leftarrow \text{argmax}_G f(x)$; // select the point with the greatest $f(x)$

$R \leftarrow R \cup \{p\}$; // add the point to the recommended set

$f(x) \leftarrow f(x) \ast g(x, p)$; // update the acquisition function per eq. (3)

$i \leftarrow i + 1$;

end

$$g(x, x') = 1 - \exp \left(-\frac{d(x, x')}{C \ast d_G/N}\right)$$

$$d(x, x') = \sqrt{\sum_{i=1}^{n} (x_i - x'_i)^2}, \quad d_G = d \left(\begin{bmatrix} x_{\text{min}} \\ y_{\text{min}} \end{bmatrix}, \begin{bmatrix} x_{\text{max}} \\ y_{\text{max}} \end{bmatrix}\right)$$

Equation 3: The acquisition update function $g$ is a penalty applied in alg. 2 to ensure spatial dispersal of the recommended points. Points $x$ nearest to each recommended point $x'$ are penalized by a scaling factor that shrinks exponentially with respect to their Euclidean distance $d$. $d_G$ denotes the diagonal size of the grid $G$. 
4 Results

4.1 Real-world data

We present results for three real-world datasets referenced throughout this article: (i) Catabot ASV sensor measurements from Lake Sunapee, New Hampshire [26]. (ii) ASV sensor measurements from Lac Hertel, Quebec [16]. Both ASV sensor measurement datasets were collected via multiparameter sonde water quality monitoring systems affixed to the vehicles. Some measured fields, such as salinity and chlorophyll concentration, were discarded due to the prevalence of discontinuities, or other irregularities rendering the data unfit for Gaussian Process analysis. (iii) Atmospheric column density readings from the satellite Copernicus Sentinel-5P, integrated via Google Earth Engine [27]. We queried an arbitrary one-square-degree area in southern Ontario over a one-month period, taking the mean values (in areas having sufficient data) from seven bands within the COPERNICUS/S5P/NRTI collection. Although it’s possible to combine data from different satellite collections, our method is designed to leverage connections between fields; therefore, we sought measurements of interrelated fields when constructing this dataset. We give context for these datasets in fig. 2, showing the Catabot ASV along with satellite imagery of each location.

4.1.1 Demonstration

Before proceeding to controlled experiments, we first demonstrate the capabilities of our method by walking through the process from start to finish. We begin by plotting the pre-processed sensor measurements, choosing one example field from each dataset. These are among the measurements serving as input features for the approximate GP models. Other sensor fields have been pre-processed in parallel, but are omitted here for the sake of brevity. Beside each set of measurements in fig. 3, we plot the corresponding fields as predicted by the approximate GP models. The measurements on the left (figs. 3a, 3c and 3e) are directly comparable to the predicted fields on the right (figs. 3b, 3d and 3f); similarities are visibly apparent in the areas covered by measurements. For these demonstrations, we used the default range of $M_{\text{min}} = 250$ to $M_{\text{max}} = 300$ inducing points, with 200 iterations of optimization.

After using sensor measurements to predict the fields, we select another example field from each dataset to stand in as the sampling field. We haven’t collected physical samples from the dataset locations; therefore, it’s necessary to derive simulated samples from the available data. In figs. 4a, 4c and 4e, we overlay the simulated sampling locations over each of the designated sampling fields. Samples are simulated in lattice patterns over $G$, with Gaussian noise added in proportion to the standard deviations of each underlying field. Specifically, we add random noise $N \sim N(0, (\sigma_n^2 \sigma_f^2))$.
(a) The Catabot, a research-oriented ASV designed by the Dartmouth Reality and Robotics Lab for reliable in-water sensing [26].

(b) Satellite imagery of the Catabot test site at Herrick Cove in Lake Sunapee, New Hampshire, USA.

(c) Satellite imagery of the data collection area in Lac Hertel, Quebec, Canada [16].

(d) Satellite imagery of southwestern Ontario, Canada, showing the query area for the Sentinel-5P dataset from Google Earth Engine [27].

Figure 2: Context for the real-world data sources. We show the Catabot ASV along with Google Earth satellite imagery of the locations for each dataset.

to the samples, where $\sigma_n^s$ is the sample noise ratio experiment parameter, and $\sigma_f$ is the standard deviation of the relevant field in $\mathcal{G}$. The addition of noise is motivated by the uncertainties inherent in real-world measurements; simulating samples without noise may constitute information leakage and skew the results. The noise ratio parameter $\sigma_n^s$ (and later $\sigma_n^m$) is a convenient mechanism for adding proportional levels of noise to observations from fields of different magnitudes. For this experiment, we simulated samples with a noise ratio of $\sigma_n^s = 0.1$. 

Next, we move onto the exact GP models for predicting the sampling fields. Here, the other \( n_f - 1 \) fields in each dataset serve as the observation features, enabling these models to exploit any correlations or covariance between fields. The results are plotted in figs. 4b, 4d and 4f. As in fig. 3, these predictions are directly comparable to the sampling fields shown to their left. Once again, the observations are visibly similar to their respective results. For the final step of this process, we run alg. 2 on the predicted sampling fields, recommending locations for further sampling according to the specified acquisition functions. In figs. 4b, 4d and 4f, we overlay the recommended locations over each of the predicted sampling fields. We list the acquisition functions along with other relevant parameters in table 1.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Lake Sunapee</th>
<th>Lac Hertel</th>
<th>Sentinel-5P</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of fields ( n_f )</td>
<td>4</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>Number of samples ( n_s )</td>
<td>20</td>
<td>30</td>
<td>40</td>
</tr>
<tr>
<td>Number of recommendations ( n_r )</td>
<td>6</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>Acquisition function ( f )</td>
<td>min-variance</td>
<td>maxima</td>
<td>minima</td>
</tr>
</tbody>
</table>

4.1.2 Field cross-validation

We move onto our quantitative experiments, using the same real-world datasets to test our main hypothesis: that field predictions can be optimized by leveraging all available data. To that end, we design a cross-validation experiment that repeatedly predicts each field at different levels of data visibility. Using the exact GP, we predict each field using every possible set of features; that is, all combinations of the \( n_f \) fields in the same dataset. Therefore, testing a dataset of \( n_f \) fields involves \( 2^{n_f} \) total predictions with the exact GP. The full experiment involves the following steps:

1. Predict each field from sensor measurements using the approximate GP. We used the default range of \( M_{\text{min}} = 250 \) to \( M_{\text{max}} = 300 \) inducing points, with 200 iterations of optimization.

2. Simulate samples from each of the predicted fields. We took \( n_s = 30 \) samples per field at a noise ratio of \( \sigma_n^2 = 0.3 \).

3. Each field is predicted again, using the samples with the exact GP. These predictions are repeated for every possible feature set, that is, every combination of the dataset’s other fields.
4. Report the relative mean absolute error for each prediction; that is, the mean absolute error between the exact GP results and approximate GP predictions (the source of the samples), divided by the mean absolute deviation of the latter. This enables comparisons across fields having different numerical scales. We present the formula in eq. (4).

\[
\text{RMAE}(s, p) = \frac{\sum_{i \in \mathcal{G}} |s_i - p_i|}{\sum_{i \in \mathcal{G}} |s_i - \mu_s|}
\]  

(4)

Equation 4: Relative mean absolute error (RMAE) between the source field \(s\) and predicted field \(p\). We divide the mean absolute error between \(s\) and \(p\) by the mean absolute deviation of \(s\). The result is a metric of relative error, comparable across fields having different numerical scales. Fields are discretized over the grid \(\mathcal{G}\) (see eq. (2)), with individual data points denoted as \(s_i\) or \(p_i\).

In fig. 5, we report the relationship between the number of feature fields and the relative error across predictions. The results for Lake Sunapee and Lac Hertel support our hypothesis, with all fields showing decreasing trends in error as the number of feature fields grows. These trends are notably steeper for some fields than others, such as Lac Hertel’s dissolved oxygen field which sees a two-thirds reduction in error from the addition of 5 feature fields in fig. 5b. In contrast, others including Lac Hertel’s turbidity field exhibit only slight declines in error across the same range. The results for Sentinel-5P are mixed, with different fields showing decreasing, flat, or increasing trends in error in fig. 5c. Whereas the \(\text{H}_2\text{O}\) density field’s error falls by approximately one-third from adding 6 feature fields, absorbing aerosol sees a comparable rise in error, while \(\text{NO}_2\) density sees no change whatsoever. We include a more detailed discussion of these results in section 5.
(a) Temperature sensor measurements from Lake Sunapee.

(b) Approximate GP prediction of temperature using 250-300 inducing points.

(c) Dissolved oxygen saturation sensor measurements from Lac Hertel.

(d) Approximate GP prediction of dissolved oxygen saturation using 250-300 inducing points.

(e) NO2 column number density measurements from the satellite Copernicus Sentinel-5P.

(f) Approximate GP prediction of NO2 column number density using 450-500 inducing points.

Figure 3: Demonstration of field predictions from sensor measurements, using the Approximate GP on three real-world data sources. In each row, sensor measurements on the left can be compared to the resulting fields.
(a) Dissolved oxygen saturation field in Lake Sunapee, showing 20 simulated sample locations.

(b) Exact GP prediction of dissolved oxygen from samples, with 6 recommendations for min-variance.

(c) Turbidity field in Lac Hertel, showing 30 simulated sample locations.

(d) Exact GP prediction of turbidity from samples, with 12 recommendations for maxima.

(e) HCHO field from Copernicus Sentinel-5P, showing 40 simulated sample locations.

(f) Exact GP prediction of HCHO column density from samples, with 12 recommended minima.

Figure 4: Demonstration of field predictions from samples, using the Exact GP on three real-world data sources. In each row, the samples sources can be compared to the resulting fields on the right. Note that some markers are partially hidden along edges and in corners of the plots.
Figure 5: Cross-validation test results on the relationship between error and the number of feature fields using real data. We note the general downward trend in error as the number of feature fields grows, with some variation from one field to the next.
4.2 Generated data

As opposed to real-world datasets, the remaining experiments involve randomly generated spatial fields from the GSTools library. We observe a lack of suitable datasets online, finding only Lac Hertel [16] as a publicly-available dataset of robotic sensor measurements over colocated spatial fields. We supplement this with our own lab’s dataset from Lake Sunapee [26], and by constructing a third dataset of satellite measurements using Google Earth Engine [27]. Seeking additional test cases beyond these three datasets, we turn to random generation. GSTools generates random spatial fields for a given covariance model; for the sake of consistency, we use the same Matérn covariance function as our GP models, with $\nu = 3/2$ as shown in eq. (1).

For experimental purposes, generated fields offer several advantages over physical datasets. Principally, we’re empowered to manipulate the field generation parameters in order to quantify our method’s performance over a range of field characteristics, in a controlled manner. Namely, we vary the number of test fields $n_f$, the length scale of the covariance model $\lambda_f$, and the similarity factor between fields $\rho_f$. Another consequence of generating data is having complete knowledge of the “ground truth” fields behind the sensor measurements. Therefore, we can directly compare the GP prediction results against their underlying fields. In these experiments, we measure the models’ accuracy by reporting the relative mean absolute error between the predictions and their respective source fields, as defined in eq. (4). Such comparisons are not possible using the real-world datasets, as ground truths are unknown. We perform two experiments with generated data, testing our method’s approximate and exact GP models in turn.

4.2.1 Approximate GP

We begin our experiments on generated data by testing the approximate GP, which extrapolates spatial fields from dense sensor measurements. We isolate the approximate GP first because the exact GP depends on its results, and the two models are affected by different experimental parameters. With this experiment, we seek to determine the optimal parameters for approximate GP prediction, learning how its accuracy is affected by each of the independent variables. Specifically, we vary the field length scale $\lambda_f$, number of turns $n_t$, and inducing point bounds $[M_{\min}, M_{\max}]$. We list the experimental parameters with more detail in table 2, and provide visualizations of $\lambda_f$ and $n_t$ in fig. 6. The results of this experiment will shed light on the trade-offs between performance and cost, in terms of computational resources and data collection capacity.

Unlike the real-world datasets which are comprised of sensor measurements, the randomized fields are generated without gaps over $\mathcal{G}$. Therefore, before extrapolating from sensor measurements, we must first simulate the collection of this data over each field. Measurements are simulated
Table 2: Independent variables for approximate GP experiments.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Values</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n_f$</td>
<td>5</td>
<td>Number of test fields, randomly generated for each $\lambda$ using GSTools. Test cases are repeated for each of these fields.</td>
</tr>
<tr>
<td>$\lambda_f$</td>
<td>{0.3, 0.5, 0.75, 1.0, 1.3, 1.7}</td>
<td>Length scale of the covariance model for randomly generated fields.</td>
</tr>
<tr>
<td>$n_t$</td>
<td>{3, 4, 5, 7, 9, 12, 15}</td>
<td>Number of turns, for simulated lawnmower measurements. Represents the density of information over the field, and determines the number of measurements.</td>
</tr>
<tr>
<td>$\sigma^m_n$</td>
<td>0.1</td>
<td>Sensor noise ratio; $\sigma$ of Gaussian noise added to the simulated sensor measurements, in proportion to $\sigma$ of the underlying field.</td>
</tr>
<tr>
<td>$[M_{\text{min}}, M_{\text{max}}]$</td>
<td>{[11, 20], [21, 40], [41, 70], [71, 110], [111, 160], [161, 220], [221, 290], [291, 370]}</td>
<td>Inducing point bounds for the approximate GP models. Each test case has $M$ inducing points where $M_{\text{min}} \leq M \leq M_{\text{max}}$, determined according to bounded adaptive inducing point selection (see alg. 1).</td>
</tr>
</tbody>
</table>

In lawnmower patterns over $\mathcal{G}$, moving across the space between $x_{\text{min}}$ and $x_{\text{max}}$, then turning around while changing the $y$ position, repeating up to the number of turns $n_t$. Intuitively, the pattern resembles the letter C when $n_t = 1$ and S when $n_t = 2$. To illustrate this effect, we show examples of lawnmower simulations for different $n_t$ in figs. 6c and 6d. The number of turns $n_t$ directly determines the number of measurements $n_m$, serving as a proxy for the amount of data collected as well as the density of information over $\mathcal{G}$. When simulating sensor measurements, we add Gaussian noise to the data in proportion to the standard deviations of each underlying field. This process is identical to that of section 4.1.1; we add random noise $N \sim N(0,(\sigma^m_n \sigma_f)^2)$ to the sensor measurements, where $\sigma^m_n$ is the sensor noise ratio experiment parameter, and $\sigma_f$ is the standard deviation of the relevant field in $\mathcal{G}$. As before, the addition of noise is motivated by the uncertainties inherent in real-world data collection.

The approximate GP experiment involves the following steps:

1. Generate $n_f$ random fields using GSTools, for each value of the field length scale $\lambda_f$. 

19
2. For each combination of the number of turns $n_t$ and the sensor noise ratio $\sigma^m_n$, simulate sensor measurements of all fields in a lawnmower pattern over $\mathcal{G}$.

3. For each set of simulated measurements and each pair of inducing point bounds $[M_{\text{min}}, M_{\text{max}}]$, select inducing points using alg. 1 to predict each field using the approximate GP. As a time-saving measure, we used 100 iterations of optimization rather than the default of 200. We note that reducing the number of iterations had negligible effect on the results.

4. Report the relative mean absolute error for each prediction, as defined in eq. (4).
We note that the number of test cases grows exponentially as more independent variables are introduced, since all combinations are tested. Therefore, it’s necessary to be prudent with the number of these variables as well as their ranges. Our experiment involved 1680 predictions of the approximate GP, with a total runtime of 37m:21s (without multiprocessing, including field generation and other overheads) on a MacBook Pro M1 Max with 64GB of RAM. Unlike for cuda devices, GPyTorch does not support GPU acceleration on the M1’s Metal Performance Shaders.

The results of this experiment are shown in fig. 7, in terms of the relationships between relative error, field length scale $\lambda_f$, number of measurements $n_m$, and number of inducing points $M$. Unsurprisingly, predictions tend to become more accurate with increasing numbers of measurements and inducing points. Figure 7b shows that both of these parameters must be sufficiently large in order to yield high-quality predictions; increasing just one confers no benefit as long as the other remains too small, as seen for $n_m = 400$ and $M \leq 40$. We also note that the error-inducing-points curve in fig. 7a resembles a power law relationship, especially at greater length scales $\lambda_f$. We recall fig. 6, which illustrates that smaller $\lambda_f$ generates fields having more spatial variation, and hence, more room for error. The consequences are clearly demonstrated in fig. 7a, where errors decrease neatly in order from the smallest to largest $\lambda_f$. See section 5 for further discussion of these results.

### 4.2.2 Exact GP

In this experiment, we test our main hypothesis – that field predictions can be optimized by leveraging all available data – focusing on the exact GP model’s performance under different conditions. The method is similar to the cross-validation procedure in section 4.1.2, in that fields are repeatedly predicted at different levels of data visibility. Seeking a larger sample size, we randomly generate fields in order to learn how the model’s accuracy is affected by the independent variables. Namely, we vary the field similarity factor $\rho_f$, number of samples $n_s$, and sample noise ratio $\sigma_n^s$ (see table 3 for more detail). These results will reveal under which conditions our hypothesis holds, as well as the types of fields and samples that are best-suited to our method.

As in the previous experiment, we take advantage of our control over the field generation parameters. Unlike the approximate GP which predicts each field in isolation, the exact GP looks at multiple fields concurrently in order to learn from their correlations. Therefore, it stands to reason that fields which are more or less closely related will respond differently (in terms of error) to changes in the number of feature fields. To that end, we introduce the field similarity factor $\rho_f$, which controls the level of similarity among an ensemble of fields on a scale from 0 to 1. For each group of fields, we first generate a common basis field $f_b$; then, the $n_f$ test fields undergo the linear transformation shown in eq. (5). Thus, at $\rho_f = 0.0$, the fields are generated independently. With increasing $\rho_f$, the fields become more similar until $\rho_f = 1.0$, where they’re all duplicates of $f_b$. 21
(a) Plotting the relative error against \( M \) produces distinct curves for each \( \lambda_f \), with some variation by \( n_m \). Each point represents the median error over \( n_f = 5 \) fields, where all other parameters are equal.

(b) The effect of \( M \) on error strengthens as \( n_m \) grows; as does the effect of \( n_m \) as \( M \) grows. Here, test cases are limited to \( \lambda_f = 1.0 \), and grouped into four bins based on \( M \).

Figure 7: Results of the approximate GP experiments, illustrating the relationship between prediction error and the field length scale \( \lambda_f \), number of measurements \( n_m \), and number of inducing points \( M \).

\[ f_i^* = f_b \rho_f + f_i(1 - \rho_f) \]

Equation 5: We control the level of similarity among generated fields using the factor \( \rho_f \in [0,1] \). Here, the field \( f_i \) transforms into \( f_i^* \) through a linear combination with the common basis field \( f_b \).

Moving onto the next parameter, we recall from section 4.1.2 that testing \( n_f \) fields for every combination of features would result in \( 2^{n_f} \) total predictions with the exact GP. We use \( n_f \) = 10 here, and we repeat these experiments in a grid search over every combination of independent variables, as in section 4.2.1. To slow the exponential growth of test cases, we introduce the combinations limit \( C_{\text{max}} \). Instead of repeating predictions for all \( \binom{n_f-1}{k} \) combinations of \( k \) features, we randomly select \( C_{\text{max}} \) combinations if \( C_{\text{max}} < \binom{n_f-1}{k} \), for each \( k \) from 1 to \( n_f - 1 \). This step decreases the number of exact GP predictions in our experiment from 27648 to 7020, limiting repetition while maintaining the breadth of independent variables.

Once again, given our knowledge of the “ground truth” generated fields, we measure model accuracy using the relative mean absolute error defined in eq. (4) between the final predictions and their
respective source fields. We note the subtle differences in error calculations between our three quantitative experiments: in section 4.1.2, using real-world measurements without knowledge of the ground truth, we compare the approximate GP predictions to the exact GP predictions. In section 4.2.1, we compare the ground truth to the approximate GP predictions; here, we compare the ground truth to the exact GP predictions. The full experiment involves the following steps:

1. Generate $n_f$ random fields using GSTools, for each value of the field similarity factor $\rho_f$ and length scale $\lambda_f$. Having already tested length scales in section 4.2.1, we use a single value here to avoid unnecessary repetition.

2. Simulate sensor measurements of all fields in a lawnmower pattern over $\mathcal{G}$, using the single values provided for the number of turns $n_t$ and sensor noise ratio $\sigma^m_{n}$. 

3. Predict the fields using the approximate GP on the simulated measurements, with the single pair of inducing point bounds $[M_{\text{min}}, M_{\text{max}}]$ provided and 50 iterations of optimization.

4. For each combination of the number of samples $n_s$ and the sample noise ratio $\sigma^s_{n}$, simulate samples from the ground truth generated fields.

5. For each set of simulated samples, and each $k$ from 1 to $n_f - 1$, predict each field using the exact GP (100 iterations) with $k$ other fields as features. This is performed for all $\binom{n_f - 1}{k}$ combinations of $k$ features, or for $C_{\text{max}}$ randomly selected combinations if $C_{\text{max}} < \binom{n_f - 1}{k}$. To match the process for real-world data and avoid information leakage, we use the approximate GP predictions as features here, as opposed to the ground truth.

6. Report the relative mean absolute error for each prediction, as defined in eq. (4).

This experiment involved a total of 7050 predictions, of which 7020 were from the exact GP and the remaining 30 from the approximate GP. The total runtime was 56m:41s (without multiprocessing, including field generation and other overheads) using the same 64GB MacBook Pro M1 Max as the previous experiment.

We present the results in fig. 8, focusing on how the relationship between error and feature count is affected by varying the sample noise $\sigma^s_{n}$, number of samples $n_s$, and field similarity $\rho_f$. For most combinations of these variables, we observe results in support of our hypothesis. Figure 8a shows that in order for additional feature fields to reduce error, some uncertainty in the sample measurements is necessary, with $\sigma^s_{n} = 0.6$ seeing significant reductions in error, while $\sigma^s_{n} = 0.3$ exhibits a smaller decline and $\sigma^s_{n} = 0.0$ shows a slight increase. In fig. 8b, we see that $n_s = 15$ is not a sufficient number of samples, and note that the results are quite similar between $n_s = 30$ and $n_s = 45$, indicating diminishing returns. As for fig. 8c, we see comparable levels of error across
the different field similarity values, except that the decline occurs more quickly for \( \rho_f = 0.6 \), where the common basis field \( f_b \) can be learned from fewer features. Section 5 includes a more detailed discussion of these results.

Table 3: Independent variables for exact GP experiments.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Values</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( n_f )</td>
<td>10</td>
<td>Number of test fields, randomly generated for each ( \lambda ) using GSTools. For each test field, we repeat the experiments using ( k )-sized combinations of the other test fields as features, from ( k = 0 ) to ( k = n_f - 1 ).</td>
</tr>
<tr>
<td>( \lambda_f )</td>
<td>1.0</td>
<td>Length scale of the covariance model for randomly generated fields.</td>
</tr>
<tr>
<td>( \rho_f )</td>
<td>{0.0, 0.3, 0.6}</td>
<td>Similarity factor for randomly generated fields. At 0.0, fields are fully independent; this scales linearly up to 1.0, where all fields are equivalent.</td>
</tr>
<tr>
<td>( n_t )</td>
<td>5</td>
<td>Number of turns, for simulated lawnmower measurements over each field. Represents the spatial density of information (for the approximate GP), and determines the number of measurements.</td>
</tr>
<tr>
<td>( \sigma_m )</td>
<td>0.1</td>
<td>Sensor noise ratio; ( \sigma ) of Gaussian noise added to the simulated sensor measurements, in proportion to ( \sigma ) of the underlying field.</td>
</tr>
<tr>
<td>([M_{\min}, M_{\max}])</td>
<td>([250, 300])</td>
<td>Inducing point bounds for the approximate GP models. Each test case has ( M ) inducing points where ( M_{\min} \leq M \leq M_{\max} ).</td>
</tr>
<tr>
<td>( n_s )</td>
<td>{15, 30, 45}</td>
<td>Number of samples, for simulated sampling in lattice patterns over each field. Represents the spatial density of information for the exact GP.</td>
</tr>
<tr>
<td>( \sigma_n )</td>
<td>{0.0, 0.3, 0.6}</td>
<td>Sample noise ratio; ( \sigma ) of Gaussian noise added to the simulated samples, in proportion to ( \sigma ) of the underlying field.</td>
</tr>
<tr>
<td>( C_{\max} )</td>
<td>3</td>
<td>Combinations limit for experimental repetition. Each test case involves predicting one field while ( k ) others act as features. This is repeated for all ( \binom{n_f - 1}{k} ) combinations, or ( C_{\max} ) randomly selected, whichever is fewer.</td>
</tr>
</tbody>
</table>
Figure 8: Test results on the relationship between error and the number of feature fields under varying conditions. In each chart, one test parameter varies while the other two remain constant. We note the general downward trend in error as the number of feature fields grows.

(a) Varying the sample noise ratio $\sigma_n^s$, with $n_s = 30$ samples and $\rho_f = 0.3$ field similarity.

(b) Varying the number of samples $n_s$, with $\rho_f = 0.3$ field similarity and $\sigma_n^s = 0.3$ sample noise.

(c) Varying the field similarity factor $\rho_f$, with $n_s = 30$ samples and $\sigma_n^s = 0.3$ sample noise.
5 Discussion

Our results in section 4.1.1 demonstrate successful predictions of spatial fields using real-world data. In particular, we emphasize the visual similarity between figs. 3a and 3b (Lake Sunapee, approximate GP); figs. 3c and 3d (Lac Hertel, approximate GP); and figs. 4c and 4d (Lac Hertel, exact GP) as evidence of our method’s capabilities. As for the Sentinel-5P dataset, our models appear to learn the general structures of the fields, while struggling with fine-grained variations as seen in figs. 3e and 3f. Considering the sampling location recommendations, we observe in figs. 4d and 4f that our method recommends plausible candidates for maxima and minima within the predicted fields. These locations are spread out across $\mathcal{G}$, as intended to mitigate inaccuracies. However, the locations recommended based on variance in fig. 4b are less consistent. In seeking those locations having minimal posterior variance, we had expected the results to align with densely-measured areas in fig. 3a, which is not the case. With that in mind, we suggest a more rigorous approach to uncertainties in section 5.1.

We recall the main hypothesis that motivates our method: that spatial field predictions can be improved by feeding additional data into the model. By incorporating colocated fields as input features for the exact GP, we provide the model with crucial spatial information to enable extrapolation from minimal samples. The results of the cross-validation experiment in section 4.1.2 and fig. 5 affirm our hypothesis in principle, while revealing which types of datasets are best-suited to this technique. The ASV-produced datasets from Lake Sunapee and Lac Hertel exhibit downward trends in error across all fields, as additional features are brought into the model. For the Lake Sunapee dataset in fig. 5a, these downward trends are significant in all fields except for pH, which the model struggles to predict at any number of features. The Lac Hertel dataset in fig. 5b shows similar downward trends at varying levels among fields; we highlight the results for conductivity and dissolved oxygen, wherein the error is reduced by more than half by the addition of five feature fields.

Moving onto the Sentinel-5P dataset in fig. 5c, we observe mixed results, with different fields showing decreasing, constant, or increasing trends in error as features are added. In particular, the absorbing aerosol field predictions are seemingly confounded by the inclusion of more fields; we note that it’s the only field not of column density, and therefore suggest choosing closer-related fields for future analysis. Overall, considering the results in section 4.1.1 as well, it’s apparent that this dataset is not a good fit for our method. Relative to the ASV data, these satellite measurements are much greater in number, with fewer spatial gaps as shown in fig. 3e. Therefore, we suggest that our method be applied to datasets with empty areas in need of extrapolation.
Whereas the real-world data experiments in section 4.1 demonstrate the viability of our method, the wide-ranging randomized tests in section 4.2 offer insights into its performance under all kinds of different conditions. Starting with the approximate GP experiment, we note that the error results from a complex interplay of the independent variables, emphasizing their interactive nature. In fig. 7a, we observe general declines in error as the number of inducing points $M$ grows; however, the shape of this trend differs among field length scales $\lambda_f$ and numbers of measurements $n_m$. The length scale relationship follows logically from the fact that, all else being equal, fields with shorter $\lambda_f$ have greater entropy, with room for more short-scale variations as seen in figs. 6a and 6b. Hence, greater $M$ and $n_m$ are required in order to achieve comparable levels of error on fields with shorter $\lambda_f$. At longer length scales and greater $n_m$, we observe decays in error that closely resemble a power law relationship. This suggests that each additional inducing point results in a proportional decrease in error; thus, under the right conditions, the error is proportional to $M^{-k}$ for some positive $k$. Sollich et al. note a similar power law relationship in GP learning curves [28], while Jin et al. derive the power-law decay rate for error in GP regression based on the prior and target [29].

Recalling again our main hypothesis – that spatial field predictions can be optimized by learning from colocated fields – we proceed to the randomized exact GP experimental results. Seeking a larger sample size, we adapt the cross-validation procedure from section 4.1.2 to run on groups of fields generated with different similarity levels $\rho_f$. The results in fig. 8c suggest that field similarity determines the rate at which error decreases as new fields are added to the feature set. At the higher $\rho_f = 0.6$, we observe a sharp decline in error from 0 to 1 colocated fields, followed by unchanging error from 2 through 7. This is contrasted with the middle $\rho_f = 0.3$, which exhibits a steady decline in error from 0 to 7 features. We posit that the GP model requires fewer features to learn the common basis field $f_b$, as defined in eq. (5), from fields having greater similarity $\rho_f$. At the lowest $\rho_f = 0.0$, we observe only a slight decrease in error, if any; noting that clusters of outliers cause some boxes to appear artificially low.

As for the number of samples $n_s$, we observe in fig. 8b a similar relationship to error as the number of inducing points $M$ with regard to the approximate GP experiment. Here, results for the fewest samples show no trend in error, suggesting that $n_s \leq 15$ samples are insufficient for our method of
learning spatial fields. Nonetheless, persistent downward trends are present in the results having greater $n_s$. We note that errors from the higher $n_s = 45$ are only slightly, but consistently, lower than those of $n_s = 30$; this suggests that additional samples enhance the model’s learning capabilities, with diminishing returns. We note that unlike in fig. 8c, in which the three colors denote results from three different sets of random fields (for three different $\rho_f$), the three categories in figs. 8a and 8b are measured against the same group of $n_f = 10$ fields (at $\rho_f = 0.3$).

Lastly, we see in fig. 8a how these trends are affected by the sample noise ratio $\sigma^s_n$, which acts as a proxy for the uncertainty of real-world measurements. At the higher $\sigma^s_n = 0.6$, we observe a considerable decline in error from 0 through 4 features, with little to no changes thereafter. The same trend exists, to a lesser degree, for the middle $\sigma^s_n = 0.3$. However, in the absence of noise where $\sigma^s_n = 0.0$, we observe the lowest errors from predictions having 0 feature fields. Thus, for datasets in which the measurements comprise perfect observations of the ground truth, additional colocated fields serve only to confuse the exact GP model. Looking at the results for 7 feature fields in particular, we posit that sample noise/uncertainty imposes a lower bound on the relative error, with larger uncertainties obscuring the ground truth to a greater extent.

Taking all of our experimental results into account, we suppose that our hypothesis holds true under certain conditions. Namely, spatial field predictions are improved by adding data, provided that: (i) some uncertainty exists in the measurements, (ii) sufficient samples have been collected, (iii) the sampling and feature fields are non-trivially related, and (iv) the approximate GP has sufficient data and inducing points to predict effectively. The cross-validation test results in fig. 5 suggest that two of our three real-world datasets (Lake Sunapee and Lac Hertel) meet these conditions for success. We hope that our results serve to illustrate the value of supplementary information in the context of adaptive sampling. Although colocated observations may seem extraneous, it pays dividends to integrate this data into predictive spatial models.

### 5.1 Future directions

We propose several extensions to our method, seeking to improve its capabilities and introduce new use cases. A natural extension would be the capacity for a third spatial dimension (depth or height) of data and prediction. Similarly, our method could be adapted to model dynamic environments, through the analysis of time-series data. This would be especially practical for environmental monitoring, which often involves repeated observations of the same areas. Roberts et al. discuss Bayesian modeling and Gaussian processes for time-series data analysis, emphasizing the importance of domain knowledge in model design [30]. Much of our current implementation already supports higher-dimensional data; however, whether spatial or temporal, it remains an open question how these extensions would affect its performance.
The remaining extensions are concerned with the GP models in particular. Whereas our method employs the Matérn covariance function as a one-size-fits-all solution, we can’t rule out that different kernels may better fit some datasets. To that end, Duvenaud et al. propose a method for searching over the space of kernel structures [31]. We also see room for improvement in our method’s handling of uncertainty. Variational inference may offer a more rigorous framework, given that sensor measurements and samples are inherently uncertain [32]. Moreover, we identify Deep GPs as a potential formalization of our proposed two-stage GP method [33]. Finally, online variational conditioning enables new data to be added without fully re-training the models [34]. This framework for online decision-making could transform our method into a comprehensive adaptive sampling system, recommending new locations in real-time according to streams of data.

6 Conclusion

In this work, we presented a novel framework integrating Gaussian Process (GP) models, robotic sensor measurements, and sampling data for predicting spatial fields, with a focus on supporting robotic adaptive sampling. We thoroughly tested this method using a variety of experiments on real-world datasets as well as randomly generated fields. Our central motivating hypothesis, on the potential for colocated fields to enhance spatial prediction, was shown to be conditionally true. Taken together, the results suggest that our approach is not a one-size-fits-all solution, but rather a first step in exploring how we can optimize predictions by effectively using all available data. While our method seemed to perform better under certain conditions – specifically when applied to ASV datasets from Lake Sunapee and Lac Hertel – it struggled with others, revealing the limitations of our current approach. Finally, we proposed several extensions to our method, in the interest of further optimization and development into a comprehensive adaptive sampling system.

References


