Spatial optimization of an existing, low-cost sensor network for air pollution in London

by

Alex Herrera

B.S. Computer Science and Engineering
Massachusetts Institute of Technology, 2021

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2022

Author .....
Department of Electrical Engineering and Computer Science
May 6, 2022

Certified by .....
David Hsu
Associate Professor
Thesis Supervisor

Accepted by .....
Katrina LaCurts
Chair, Master of Engineering Thesis Committee

Spatial optimization of an existing, low-cost sensor network for air pollution in London

by

Alex Herrera

Submitted to the Department of Electrical Engineering and Computer Science
on May 6, 2022, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Electrical Engineering and Computer Science

Abstract

Air pollution sensors are rapidly decreasing in cost and can provide measurements with higher spatial and temporal resolution. In this paper we combine two different Gaussian process methods to optimize spatially an existing low-cost sensor network for air pollution in London. We demonstrate the practical utility of these combined algorithms using a cross-validation approach, applied to air pollution data obtained from 75 sensors within the London Air Quality Network (LAQN) in 2011. The analysis steps were as follows. First, based on a training subset of the original data, we trained a spatio-temporal variational Gaussian process model to quantify the uncertainty within the area of London for a year at daily intervals. A second Gaussian process algorithm was then used to optimize sensor placements by maximizing mutual information to recommend relocating a subset of the existing sensors. Evaluating a second training subset of the original data, as if sensors were relocated to the new recommended locations, we find (on average) that the second model reflecting our new recommended locations increases the mutual information across the area of London by 27.3% while maintaining the same performance on prediction root mean square error within a third subset of the original data (the validation set). We then apply this procedure to a model trained on all 75 sensors to generate an optimized redistribution of the LAQN air pollution sensors. We conclude with ideas for further extensions of this work.

Thesis Supervisor: David Hsu
Title: Associate Professor

Acknowledgments

First and foremost, I would like to thank my parents, Alejandro and Lourdes Herrera, my abuelita Isabel, and my sister, Alexandria, for their unconditional lifelong support. Their love and encouragement motivate me to work hard and inspire me to dream big every day.

I would like to thank my supervisor, David Hsu, for his invaluable expertise and guidance. The conversations we had about the societal impact of our work re-energized me whenever my spirits were low due to issues with our code. I would also like to thank my research partner, Ana Fiallo, for her excellent teamwork skills and strong dedication to the project. This project would not have been possible without her help.

Lastly, I would like to thank my friends who made my journey at MIT unforgettable. The memories I made while studying on weekdays and having fun on the weekends alongside José Guajardo, Francisco Zepeda, Nicole Goridkov, Victor Reyes, Daniel Shkreli, Sol Rodríguez, Iris Abrahantes Morales, and Sarah Vu are some of my most cherished memories.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Air pollution is a major public health and climate concern across the globe as an important source of morbidity, mortality, and climate change [1, 2, 3]. In 1952, the Great Smog of London, lethal smog which covered the city for five days, resulted in thousands of deaths. This tragedy prompted the British government to pass its first law focused on significantly reducing air pollution, the Clean Air Act, four years later [4]. Since then, air pollution has been cautiously regulated and monitored by numerous policies and scientific tools and methods.

Today, the Environmental Research Group (ERG) of Imperial College London oversees the London Air Quality Network (LAQN) [5]. The LAQN is a community air pollution sensor network, where each borough funds monitoring in its own area [6]. Measurement data from each of the sensors are sent to ERG to develop maps and models to predict the pollution levels throughout the city of London. These maps and models are used to track air pollution trends over time, assess how different government policies affect air pollution, and help with research into the health effects of air pollution [5].

Establishing monitoring sites in air pollution sensor networks such as the LAQN requires a large time and monetary investment due to the sensors' complexity and high cost [5, 7]. As a result, such networks are typically static and sparse. This has led to the recent rise in the deployment of low-cost air pollution sensors as they are easier to move due to their smaller size and significantly cheaper than their conventional

counterparts. Though these low-cost sensors are less accurate and therefore unreliable when used alone, a study conducted by the Environmental Protection Agency (EPA) has shown that these sensors could be valuable supplements to existing conventional sensor networks by using the conventional sensors as a benchmark [8]. We expect low-cost sensor deployment into conventional sensor networks to continue increasing as more research goes into their improvement [9].

Prior work has shown that optimizing sensor placements in a network could allow the network to capture more information using the same number of sensors [10]. This thesis builds on this prior work by finding the near-optimal air pollution sensor placements of the LAQN, maximizing the amount of information captured by the network. Since redistributing the sensors is a big investment, we verify our optimization procedure by simulating the redistribution of the sensors to new locations and evaluating its effects on the network's quality. The results of our optimization procedure can be used to identify where to place new sensors so that they best contribute to the existing network. These suggestions lower the barrier of entry into the network and allow community members to make more meaningful contributions using low-cost sensors. Ultimately, our work aims to improve the maps and models the ERG develops by optimizing the LAQN to capture the most amount of information it can.

Chapter 2

Materials and Methods

2.1 Overview

Figure 2-1 shows a high-level flowchart of our LAQN optimization methodology. We first collect, filter, and transform data from the LAQN sensors. We then train a machine learning model on the data and select the most informative sensor locations using an optimization algorithm which relies on outputs from the model. The chosen sensor locations represent the sensor placements which maximize the information captured by the LAQN across the area of London.

2.2 Dataset

For our study, we use average daily NO_2 measurement data collected from January 1st, 2011 to December 31st, 2011 by the LAQN sensors. We use both the openair

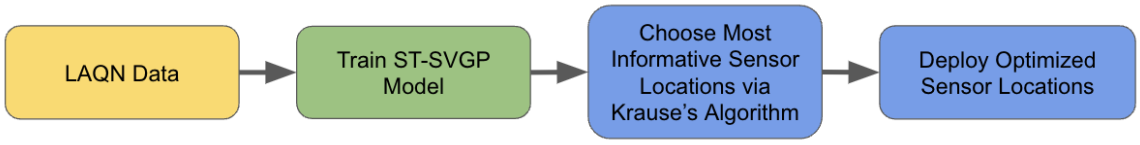

Figure 2-1: Overview of our LAQN optimization methodology. The yellow block represents our data collection phase, the green block represents our model training phase, and the blue blocks represent our optimization phase.

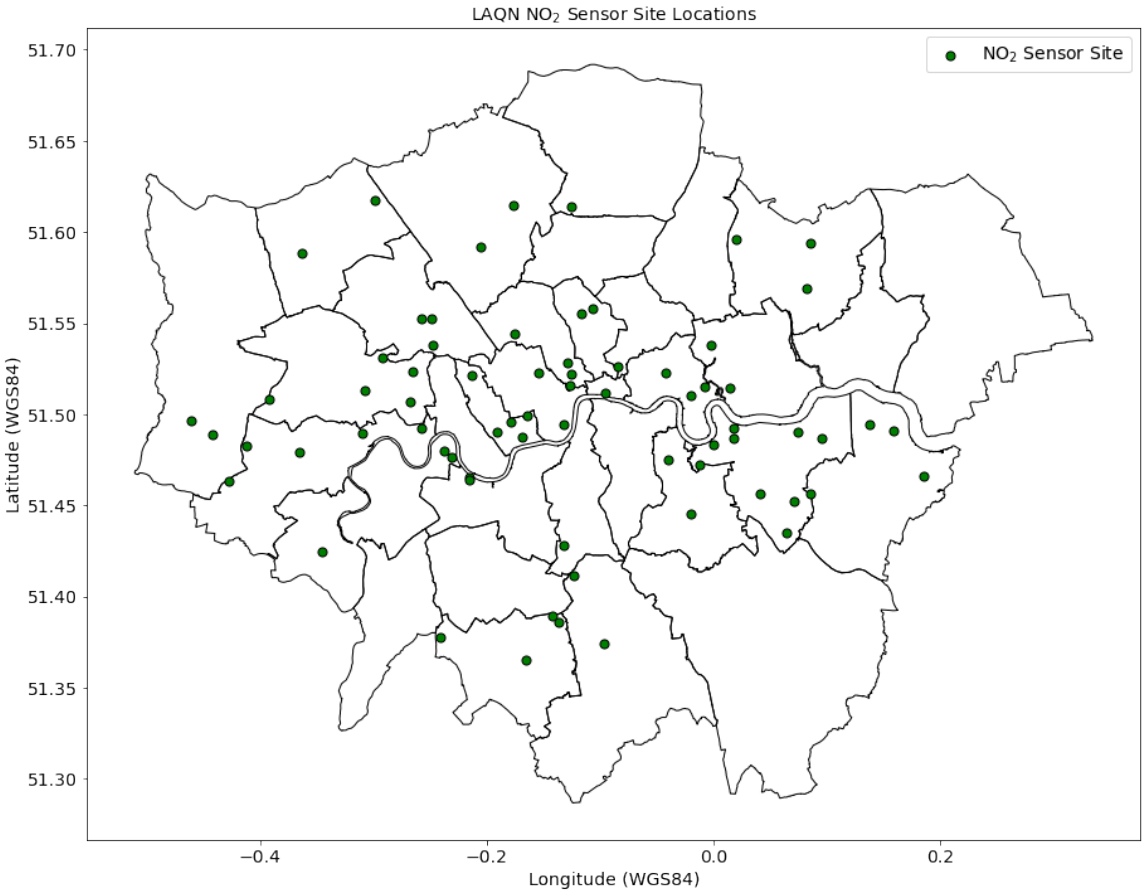

Figure 2-2: Locations of the 75 NO_2 sensor sites in the LAQN which collected data for more than 292/365 (80%) of the days in 2011

library [11] and the LAQN API [12] to download the data. Each of our data points is characterized by a timestep (integer from 0 to 364), latitude (WGS84), longitude (WGS84), and NO2 concentration measurement ($\mu g/m^3$). We filter out data from sites in the network which did not collect data for at least 292/365 (80%) of the timesteps. We normalize the latitude and longitude coordinates by removing the mean and scaling to unit variance. For data management and manipulation we use pandas [13] and scikit-learn [14].

Figure 2-2 shows the locations of the 75 NO_2 sensors in the LAQN which collected sufficient data. Notice how most of the sites are spread horizontally across central London. Additionally, notice how some of the sensors are very close to each other. Intuition tells us that as the sensor density increases in an area, the less informa-

tive each individual sensor is. Therefore, by moving a sensor from a high density area to a lower density area, we believe that the network can benefit more from the measurements collected by that sensor and thus increase the amount of information the network captures. The following sections describe the tools we need to quantitatively confirm this intuition and optimally redistribute these sensors to maximize the amount of information captured between them across the entire area of London.

2.3 Gaussian Processes

In order to begin optimizing the LAQN, we first need an appropriate way to model sensor networks. In our case, we would like a model that allows us to predict the NO_2 concentration at any given (timestep, latitude, longitude) tuple. Additionally, we would like our model to tell us the variance associated with our prediction and the covariances across a set of multiple different predictions. Such a model would allow us to calculate both prediction accuracy and uncertainty across the entire area of London. This would enable us to study and evaluate how different sets of sensor locations affect the information captured by the sensor network.

Given the mean vector μ_V and covariance matrix Σ_{VV} of a set of sensor locations V, multivariate Gaussians allow us to predict the measurement value at one location given the measurements from the other locations [15]. Importantly, multivariate Gaussians can only predict measurement values at locations where we already have sensors.

Gaussian processes (GPs) generalize multivariate Gaussians so that we can predict measurement values at any location of interest [15]. Analogous to multivariate Gaussians, a GP is specified by a mean function $M(\cdot)$ and kernel (or covariance) function $K(\cdot, \cdot)$. The mean function allows us to predict the measurement values at any timestep and location, while the kernel function tells us the variance of our prediction. The kernel function can also be used to generate a covariance matrix between the different time and location pairs we're making predictions at. The covariance matrix is the basis for our optimization methods.

Modern GP training methods utilize supervised machine learning to train the model and learn the mean function and kernel. A typical GP trains on input data (X, Y) where X is an $n \times d$ input matrix, Y is an $n \times 1$ output vector, n is the number of training data points, d is the number of features the input data has, and the rows in X and Y correspond to one another. In our case, n varies by experiment, $d = 3$ since our input features are timestep, latitude, and longitude, and the one output feature is NO_2 concentration.

Unfortunately, off-the-shelf GP model training libraries scale poorly with the number of input data points and struggle to capture spatial-temporal trends within the data. Since we are working with a large number of spatial-temporal data, we need a more tractable model.

2.4 Spatio-Temporal Variational Gaussian Processes

Spatio-Temporal Variational GPs (ST-SVGP) provide the same functionality as GPs. Their improvements come in the way they train. ST-SVGPs take advantage of the fact that the sensor locations in our network are the same at every timestep in order to drastically decrease training time while improving model performance on RMSE and negative log predictive density (NLPD) when compared to SVGPs and GPs [16]. The model uses spatio-temporal filtering to reduce the computational scaling in the number of timesteps from cubic to linear.

We use ST-SVGPs to model the sensor networks in our experiments. We use the same hyperparameters used in the air-quality experiment published in the original ST-SVGP paper [17]. Now that we have a model, we need a proper way to assess its quality.

2.5 Mutual Information

Krause et al found that optimizing GPs for maximum mutual information (MI) resulted in the best performing models, achieving the lowest root mean squared error

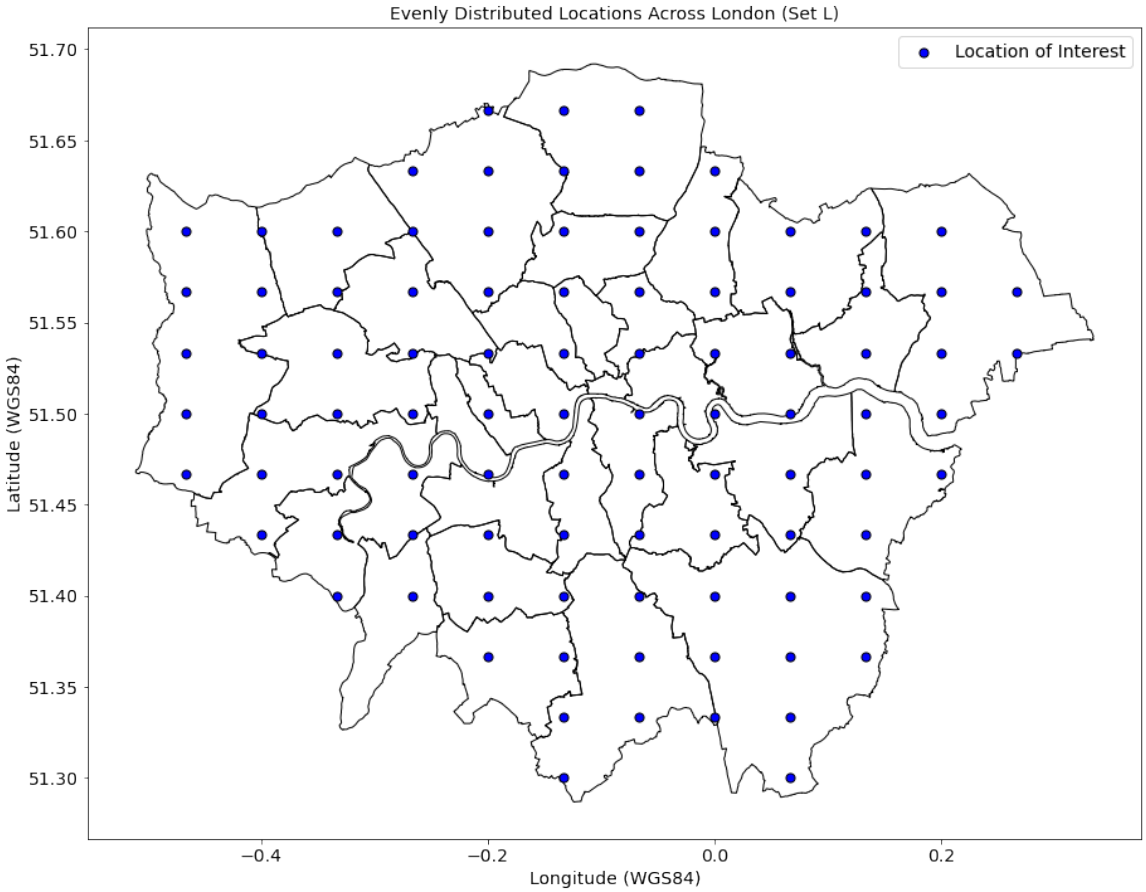

Figure 2-3: A set of evenly distributed locations across the area of London. We use this set to calculate the mutual information of our ST-SVGP models.

(RMSE) on test data, when compared to other optimization criteria, such as entropy [15]. The mutual information criterion seeks to find sensor placements that are most informative about locations where we want to monitor but have no sensors at. It directly measures the effect of sensor placements on the posterior uncertainty of the GP.

We define mutual information as the function:

$$MI(GP, X, Y) = \frac{1}{2} * \log(det(\Sigma_{XX}) * det(\Sigma_{YY})/det(\Sigma_{(X \cup Y)(X \cup Y)}))$$

where GP is an ST-SVGP model, X is a set of sensor locations, Y is a set of locations which we want to monitor but do not have sensors at, and Σ_{ZZ} is the covariance matrix (averaged across the time dimension) for the locations in set Z generated by GP [18].

This calculates how informative the sensors in set X are about the locations in set Y based on the correlations learned by GP. In this thesis, when we refer to the mutual information of some ST-SVGP model M, we are referring to $MI(M, training\_sites\_M, L)$ where $training\_sites\_M$ is the set of sensor locations which M trained on and L is the set of evenly distributed locations across the area of London (figure 2-3) unless explicitly stated otherwise.

In this way, the mutual information of an ST-SVGP model tells us how informative the model is across the entire area of London. We use the mutual information metric to evaluate the quality of our ST-SVGP models.

2.6 Krause's Algorithm

In order to maximize the mutual information of our ST-SVGP models, we use a greedy 1-1/e (63%) approximation algorithm developed by Krause et al (Krause's algorithm). Figure 2-4 shows pseudo-code for the algorithm.

Krause et al define the algorithm as the function:

$$KA(V = S \cup U, \Sigma_{VV}, k) \rightarrow A$$

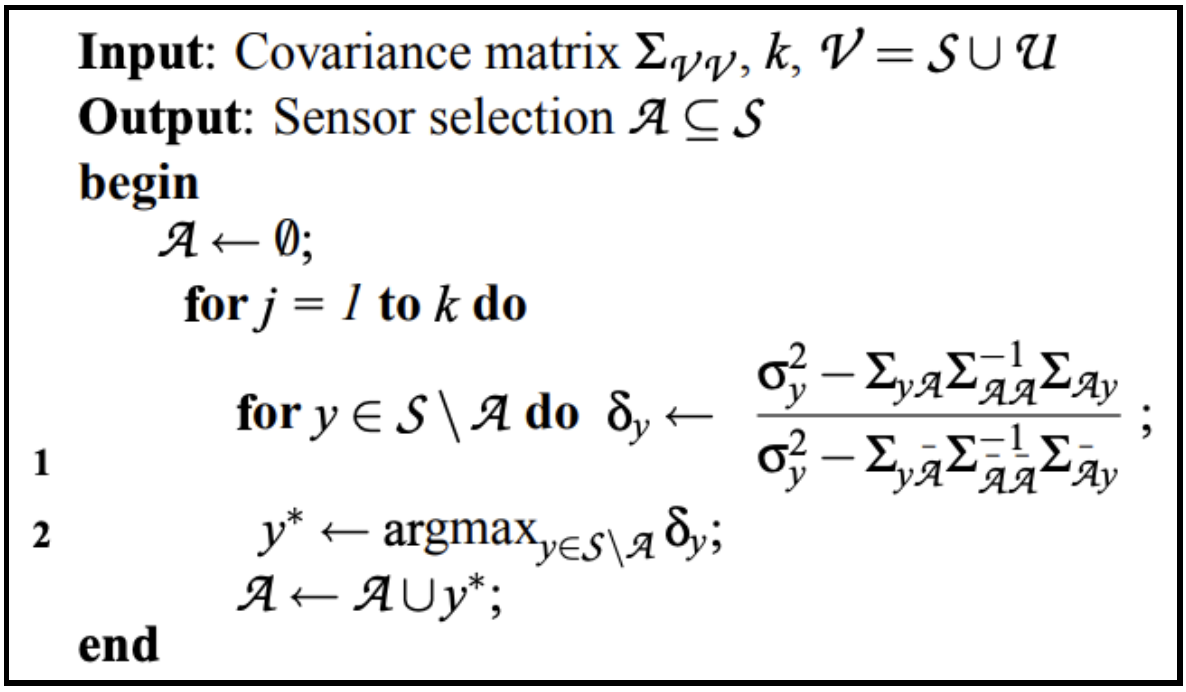

Figure 2-4: Pseudo-code for the $(1 - 1/e)$ greedy approximation algorithm which we use to optimize our ST-SVGP models for maximum mutual information. Figure reproduced from [15].

where S is the set of locations which we can place a sensor at (but not necessarily currently have a sensor at), U is the set of locations which we want to monitor but cannot place a sensor at, Σ_{VV} is the covariance matrix of the locations in set V, and A (the output) is the set of the k most informative sites in S. We use $KA(GP, S, U, k) = KA(V = S \cup U, \Sigma_{VV}, k)$ since Σ_{VV} is generated by GP. When generating Σ_{VV}, we average the covariances across the time dimension.

The algorithm works by greedily selecting the sensor location y which, when added to set A, maximizes $MI(GP, A, V \setminus A)$ at each of the k iterations. We utilize Krause's algorithm to select the most informative sensor locations with the goal that a new ST-SVGP model trained on set A will have a higher mutual information score than the original model.

2.7 Code

All of the Python code used for data collection, model training, optimization, plotting, and running experiments can be found at: `github.com/bikeclub51/london-aq`.

Chapter 3

Experiments and Results

We first show that Krause's algorithm can be used to increase the mutual information of an ST-SVGP model by redistributing some of the sensors it trained on to new locations. We then apply this procedure to an ST-SVGP model trained on the entire LAQN to generate an optimized set of LAQN sensor locations.

3.1 Experiment 1 Design

1. Randomly split up the 75 sites into three groups: training (50), candidate (20), and validation (5)

2. Train an ST-SVGP model (M_1) on the training sites' data

3. Run Krause's algorithm on our ST-SVGP model

- $KA(GP = M_1, S = training\_sites \cup candidate\_sites, U = L, k = 50) \rightarrow A$

4. Train a new ST-SVGP model (M_2) on the 50 most informative sites selected by Krause's algorithm (A)

5. Evaluate and compare the original and optimized models

- $MI(M_1, training\_sites\_M_1, L)$ vs. $MI(M_2, training\_sites\_M_2, L)$

- $RMSE(M_1, validation\_sites)$ vs. $RMSE(M_2, validation\_sites)$

- $NLPD(M_1, validation\_sites)$ vs. $NLPD(M_2, validation\_sites)$

The goal of this experiment is to assess Krause's algorithm's ability to increase the mutual information of a sensor network by selecting novel sensor placements. Krause's algorithm chooses the most informative site locations from the set of sites which the first model trained on and a set of candidate sites which the model hasn't seen before. A second model is then trained on this new set of sites and we compare the two models using mutual information and validation site prediction accuracy.

In the real world, we would not immediately know whether a redistribution of the sensor locations in the LAQN results in a better network. This is because will not have data at the new sensor locations to evaluate the effects of the redistribution. In order to determine if the new distribution of the sensors is better, we will need to move the sensors and wait until they collect enough data for us to train a new model and compare it with the previous one. This requires a significant amount of time and money. Furthermore, if the redistribution turns out to be worse, then we will have wasted valuable resources and energy. In this experiment, we only consider site placements at locations which we have data for so that we can immediately assess the quality of the redistribution, rapidly simulating a real-world sensor redistribution scenario. This simulation is crucial as we must be confident that our procedure works before applying it in the real world. We conduct this experiment 20 times.

3.2 Experiment 1 Results and Discussion

Figures 3-1 to 3-8 visualize the results of one of the 20 experiments. Figure 3-1 shows a contour plot of the uncertainty of the original ST-SVGP model generated in step 2. The training sites are plotted as green circles, the candidate sites as orange squares, and the validation sites as purple triangles (step 1). We observe low uncertainty in the central area of London where the training site density is higher. Similarly, we observe high uncertainty in the top right and bottom right regions of London where there are no training sites.

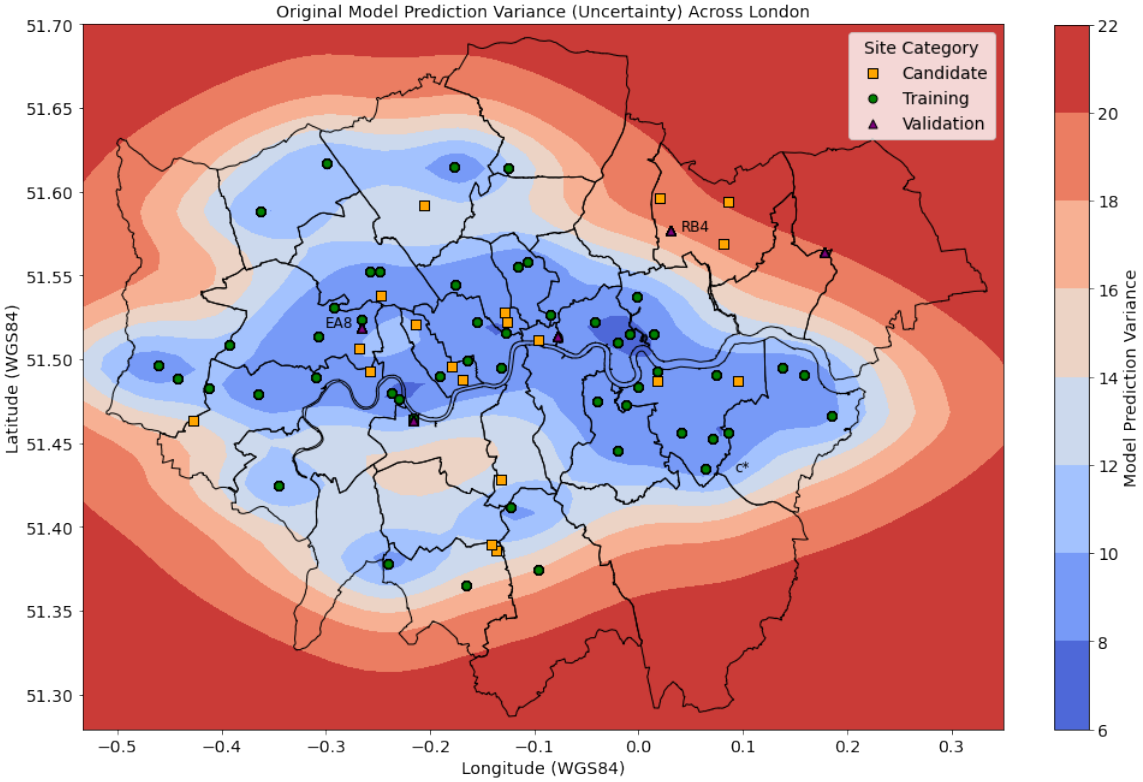

Figure 3-1: Contour plot of the uncertainty of the original model (generated in experiment 1) across the area of London

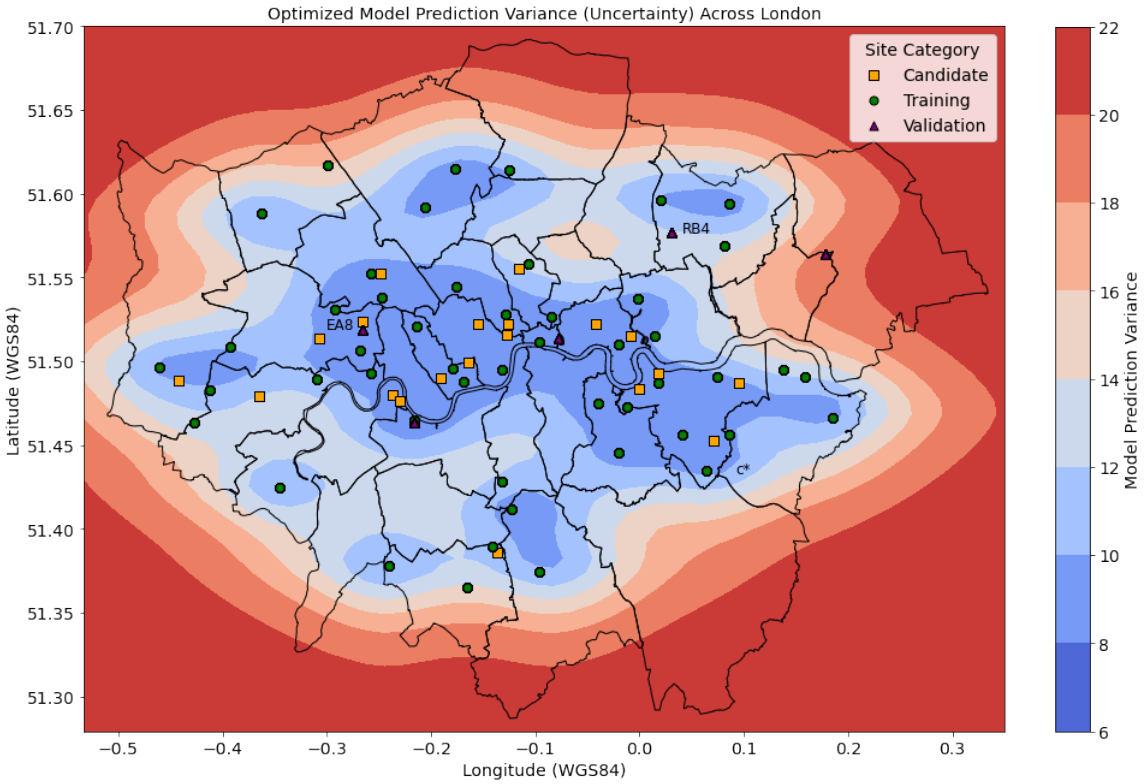

Figure 3-2: Contour plot of the uncertainty of the optimized model (generated in experiment 1) across the area of London

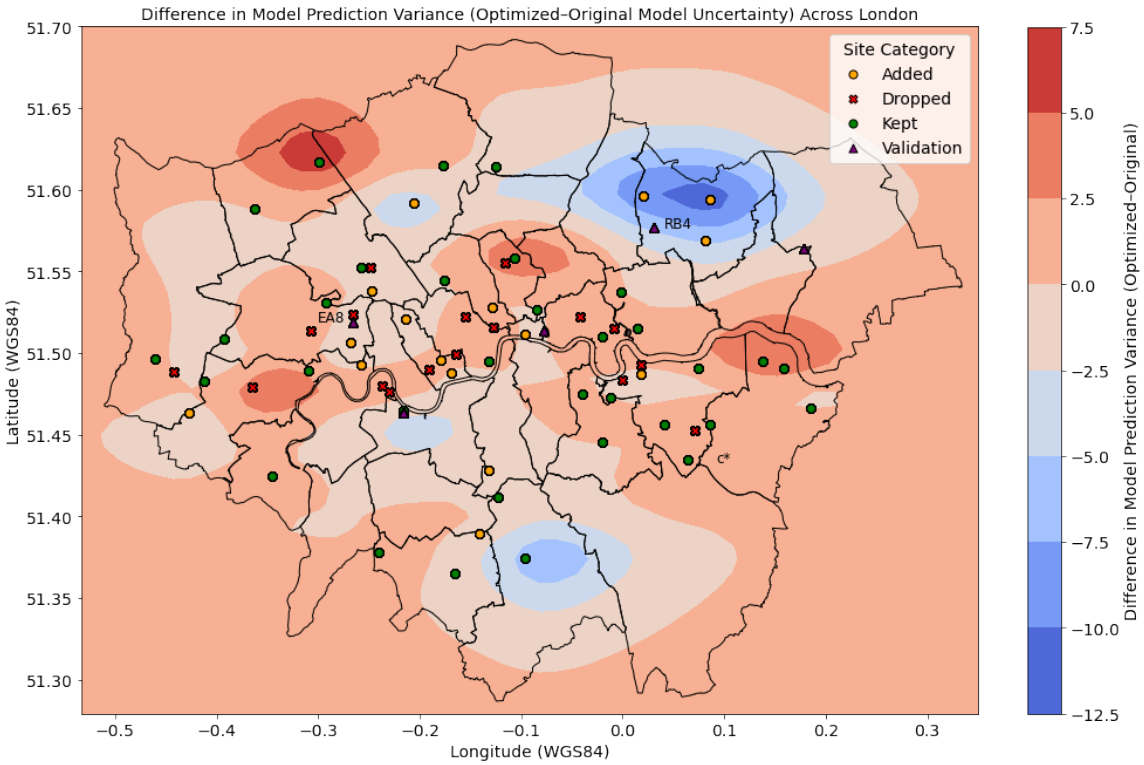

Figure 3-3: Contour plot of the difference in the uncertainty of the optimized and original model (generated in experiment 1) across the area of London

Figure 3-2 shows a contour plot of the uncertainty of the optimized ST-SVGP model generated in step 4. This model was trained on the 50 sites (among the 70 training and candidate sites) which maximize the MI contained by the original model across the area of London. The original training sites which were kept are plotted as green circles, the dropped training sites as red Xs, the newly added training sites as orange circles, and the (same) validation sites as purple triangles.

Figure 3-3 shows a contour plot of the difference in uncertainty between the optimized and original models (negative values show improvement). Here, we highlight how Krause's algorithm is able to identify redundant sensors and relocate them to address areas with high uncertainty. First, notice the cluster of four sites (3 kept and 1 dropped) in the middle-right region of London (labeled $c*$). Notice how Krause's algorithm dropped the site near the center of the other three sites. This is because the original model believed the four sites are highly correlated, so the site near the center of the cluster was found to not be contributing as much to the mutual information

of the model as the other three. By dropping that site, the sensor can be moved to a new location where there is higher uncertainty and contribute more to the MI of the new model. This trend is observable among other close clusters (especially neighboring sites) in this figure. Second, we observe how Krause's algorithm chose to place sensors in the top right region where there were previously no training sites, drastically improving the model uncertainty there. Furthermore, despite there being no available sites in the bottom right region, the new model is able to use learned correlations between the new set of training sites to improve the uncertainty in that region.

Importantly, since the number of sensors is kept fixed, we cannot expect to improve prediction variance in one area without it declining in another. In this case, we trade off lower uncertainty on the outskirts of London for higher uncertainty in the top left region and center of London. We utilize Krause's algorithm to make this trade off in such a way that the mutual information of the model improves.

Figure 3-4 compares the mutual information of the original and optimized models. We calculated an MI score of 2.79 and 3.61 for the original and optimized models, respectively. This is an increase of 29.3%. Furthermore, we observe that the optimized model captures the same amount of mutual information as the original model using 23 fewer sensors.

Figures 3-5 and 3-6 show the original and optimized models' predictions at validation site RB4 (labeled on figure 3-3). We observe an improvement of -19.8 (-57.2%) in the RMSE between the optimized and original models. We can also see that the optimized model captures the temporal trends much better than the original model. This is due to Krause's algorithm's decision to place sensors at the candidate sites near RB4 (where there were previously no sensors) in step 3.

Figures 3-7 and 3-8 show the original and optimized models' predictions at validation site EA8 (labeled on figure 3-3). Despite Krause's algorithm's decision to remove the training site which was located really close to this site, we observe an improvement of -1.50 (-10.2%) in the RMSE between the optimized and original models. This highlights the ST-SVGP model's ability to use learned correlations to make accurate

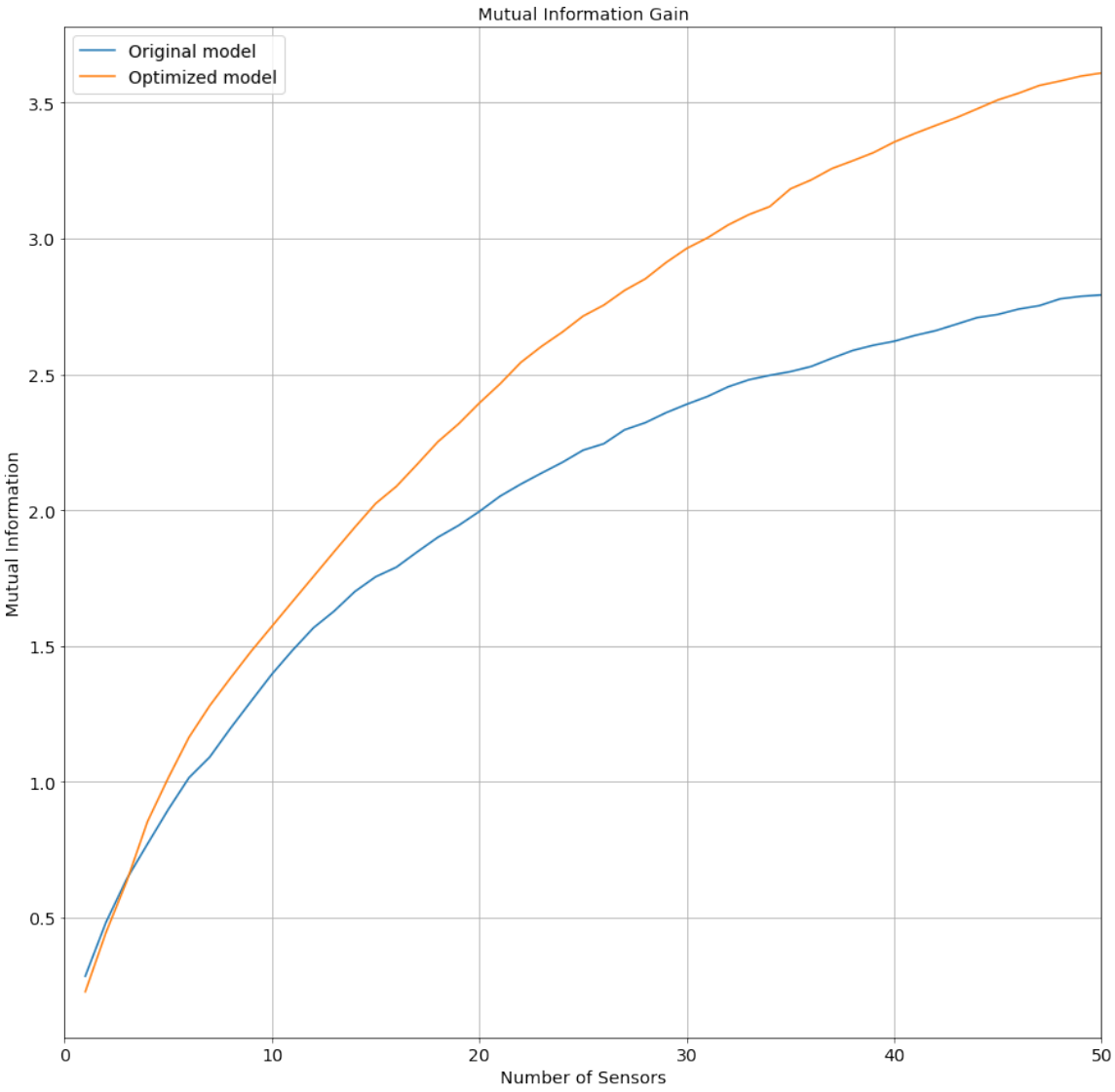

Figure 3-4: Plot of the total mutual information per sensor in the network for the original and optimized models

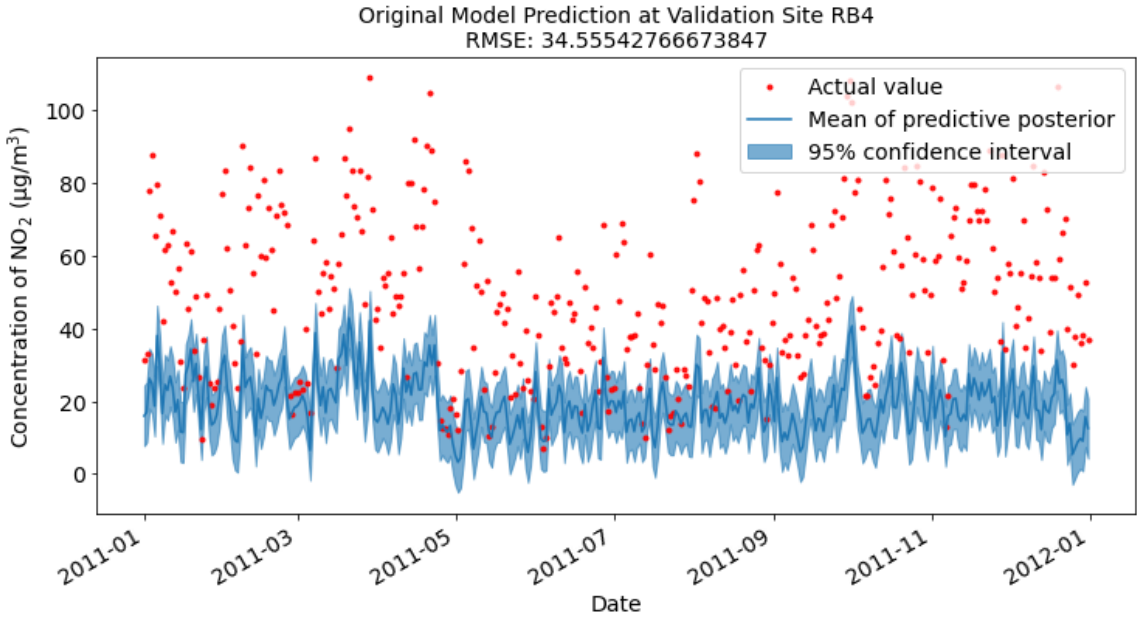

Figure 3-5: Time series plot of predicted vs. actual NO_2 concentration at validation site RB4 for the original model in experiment 1

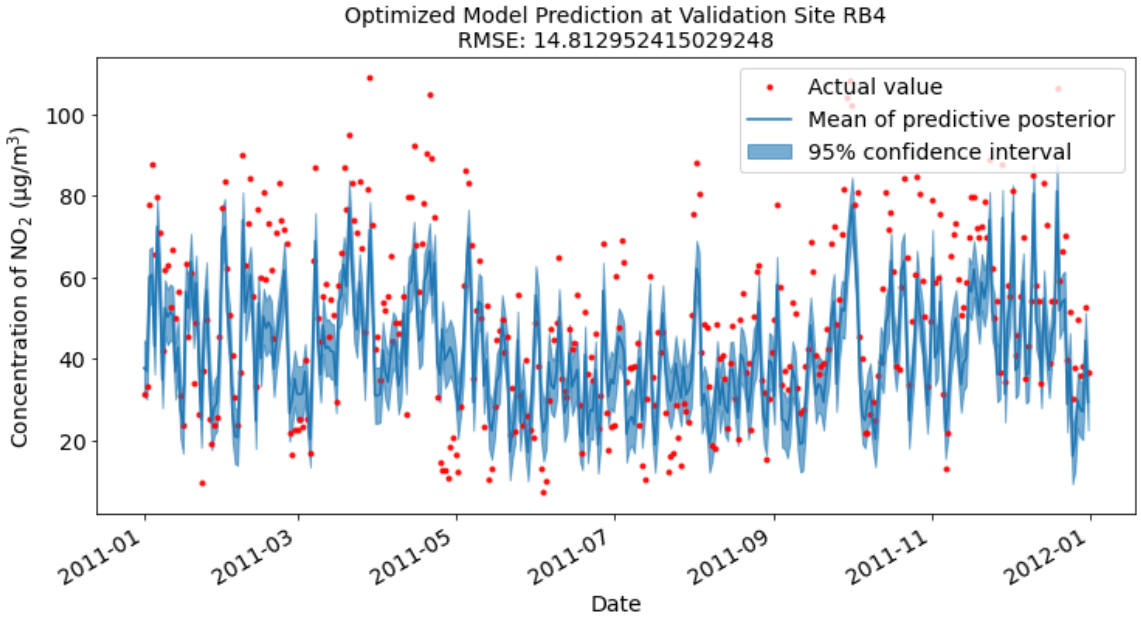

Figure 3-6: Time series plot of predicted vs. actual NO_2 concentration at validation site RB4 for the optimized model in experiment 1

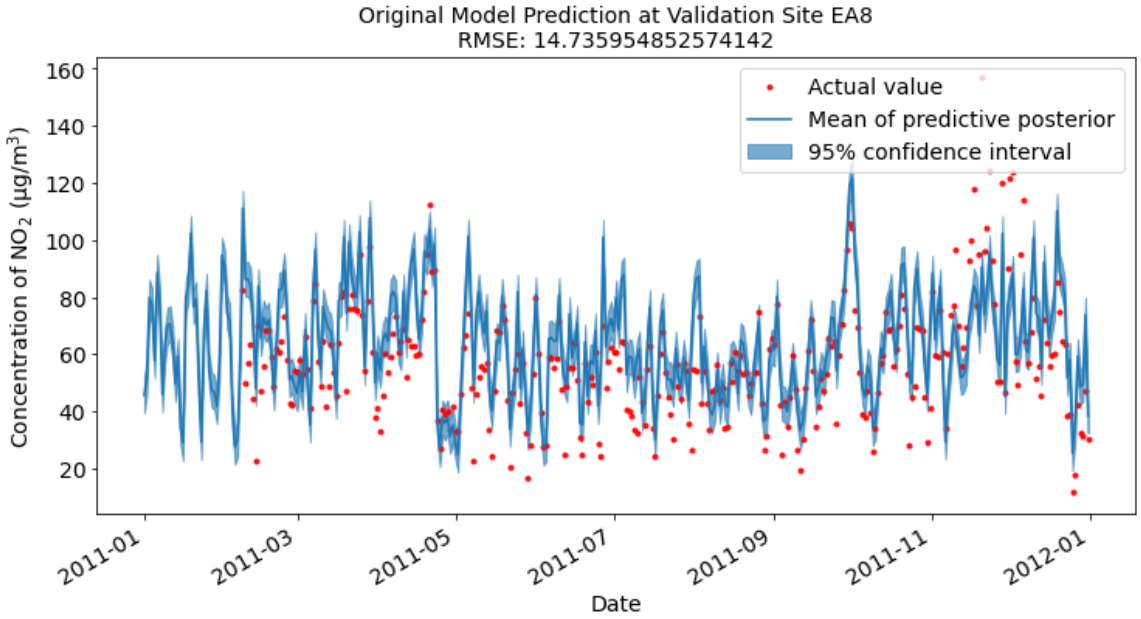

Figure 3-7: Time series plot of predicted vs. actual NO_2 concentration at validation site EA8 for the original model in experiment 1

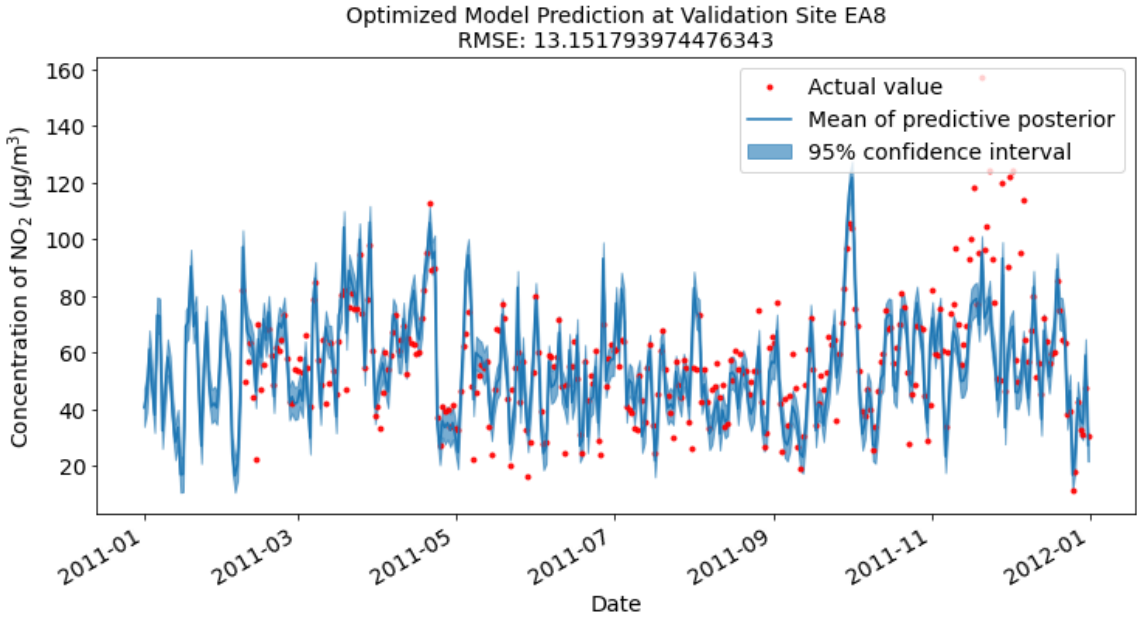

Figure 3-8: Time series plot of predicted vs. actual NO_2 concentration at validation site EA8 for the optimized model in experiment 1

predictions.

We conducted this experiment 20 times and observed an average increase of 0.779 (+27.3%) in the mutual information contained across the area of London. Additionally, we observed an average decrease of 7.89 (+2.23%) in the NLPD and 1.78 (-0.875%) in the RMSE of the validation sites. On average, the mutual information drastically improved while maintaining virtually the same performance on RMSE and slightly improving on NLPD. [1]

This experiment shows that Krause's algorithm is consistently able to improve a sensor network by redistributing the sensor sites to new locations by leveraging the spatial and temporal generalizability of ST-SVGP models.

3.3 Experiment 2 Design

1. Randomly split the data from all 75 sites into training (80%) and test (20%) datasets
2. Train an ST-SVGP model (M) on the training data
3. Run Krause's algorithm on our ST-SVGP model

- $KA(GP = M, S = LAQN\_sites \cup L, U = \emptyset, k = 75)$

The goal of this experiment is optimize the sensor locations of the LAQN by allowing Krause's algorithm to select sensor placements at locations all across London (not just locations where we have data).

3.4 Experiment 2 Results and Discussion

The model we trained in this experiment scored an NLPD of 44.6 and an RMSE of 28.7 on the test data. Figure 3-9 shows a contour plot of the uncertainty in the

[1] For our particular study, NLPD and RMSE are weaker evaluation metrics than mutual information since the data that we calculate NLPD and RMSE on is limited by the locations of the existing sites.

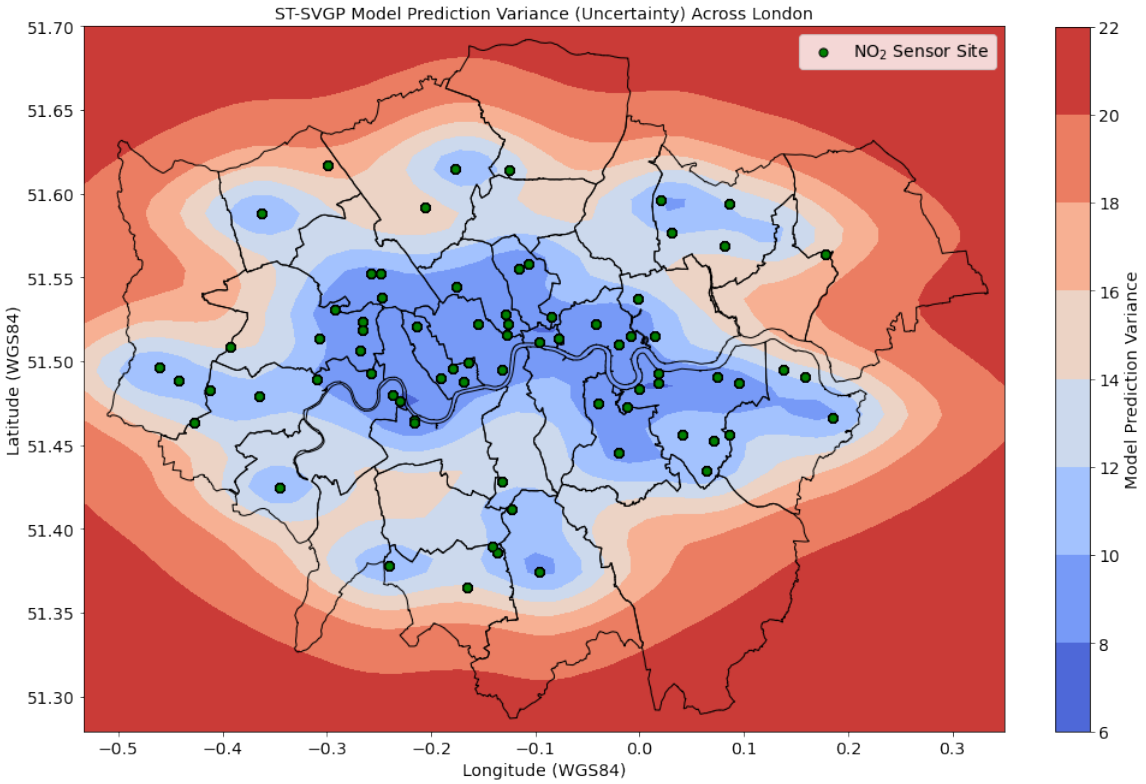

Figure 3-9: Contour plot of the uncertainty of the ST-SVGP model trained on all 75 LAQN sensors across the area of London (experiment 2)

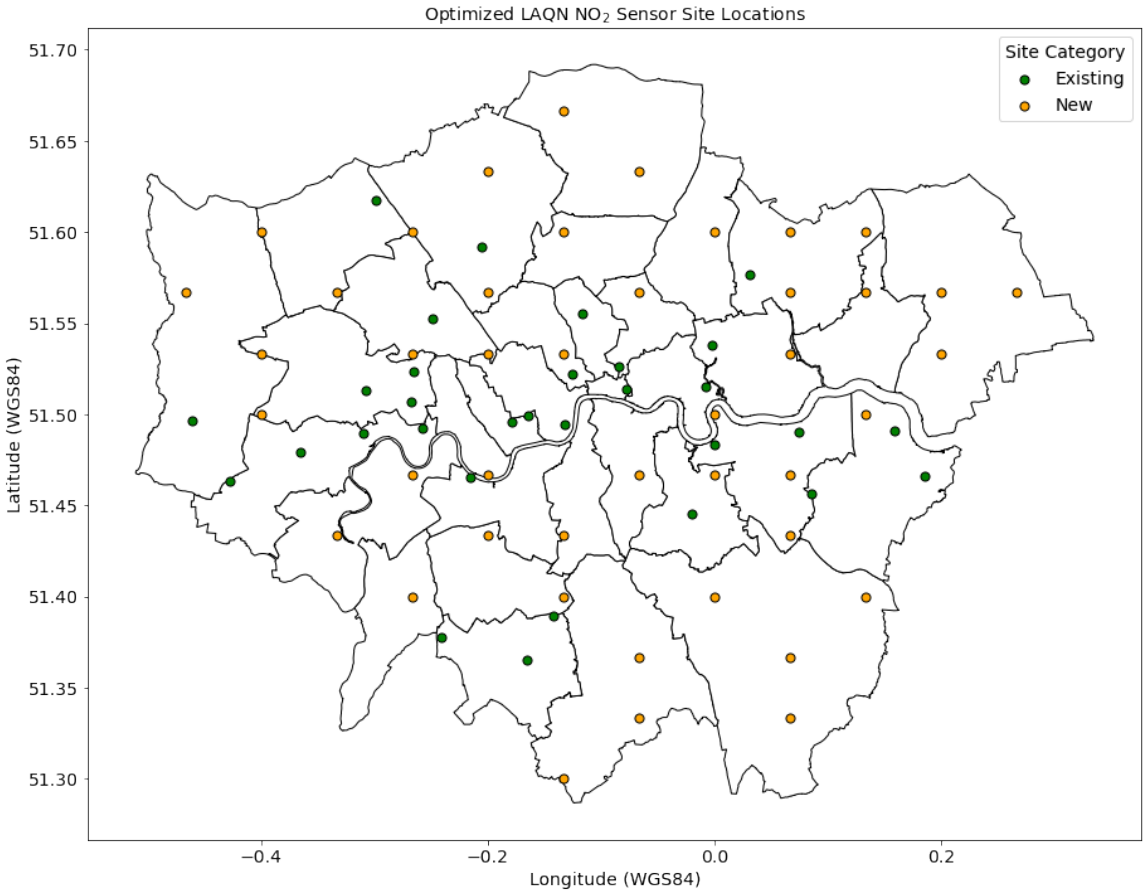

Figure 3-10: Optimized redistribution of the LAQN NO_2 sensors

model across the area of London. We observe a similar trend as before, with lower uncertainty near the center horizontal region of London and higher uncertainty near the outskirts of London. Figure 3-10 visualizes the optimized LAQN site locations generated by Krause's algorithm. The existing sites are plotted in green and the new sites in orange. In general, we observe these suggested locations are much more evenly distributed across the area of London, with the new sites being in areas where uncertainty is high. We provide a table of the selected sensors and their coordinates in the appendix (table A.1).

We cannot calculate the mutual information of the optimized model trained on these new site locations since that would require having measurement data for those locations. We can, however, predict the mutual information captured by the new site locations on the future optimized model by using our current model. Figure 3-11 compares the current mutual information of our ST-SVGP model and the predicted mutual information. We calculated an MI score of 4.53 and 5.49 for the current and predicted models, respectively. This is an increase of 21.2%. Even without this prediction, the results of experiment 1 suggest that these new site locations will yield an ST-SVGP model which contains higher mutual information across the area of London, resulting in a more informative network.

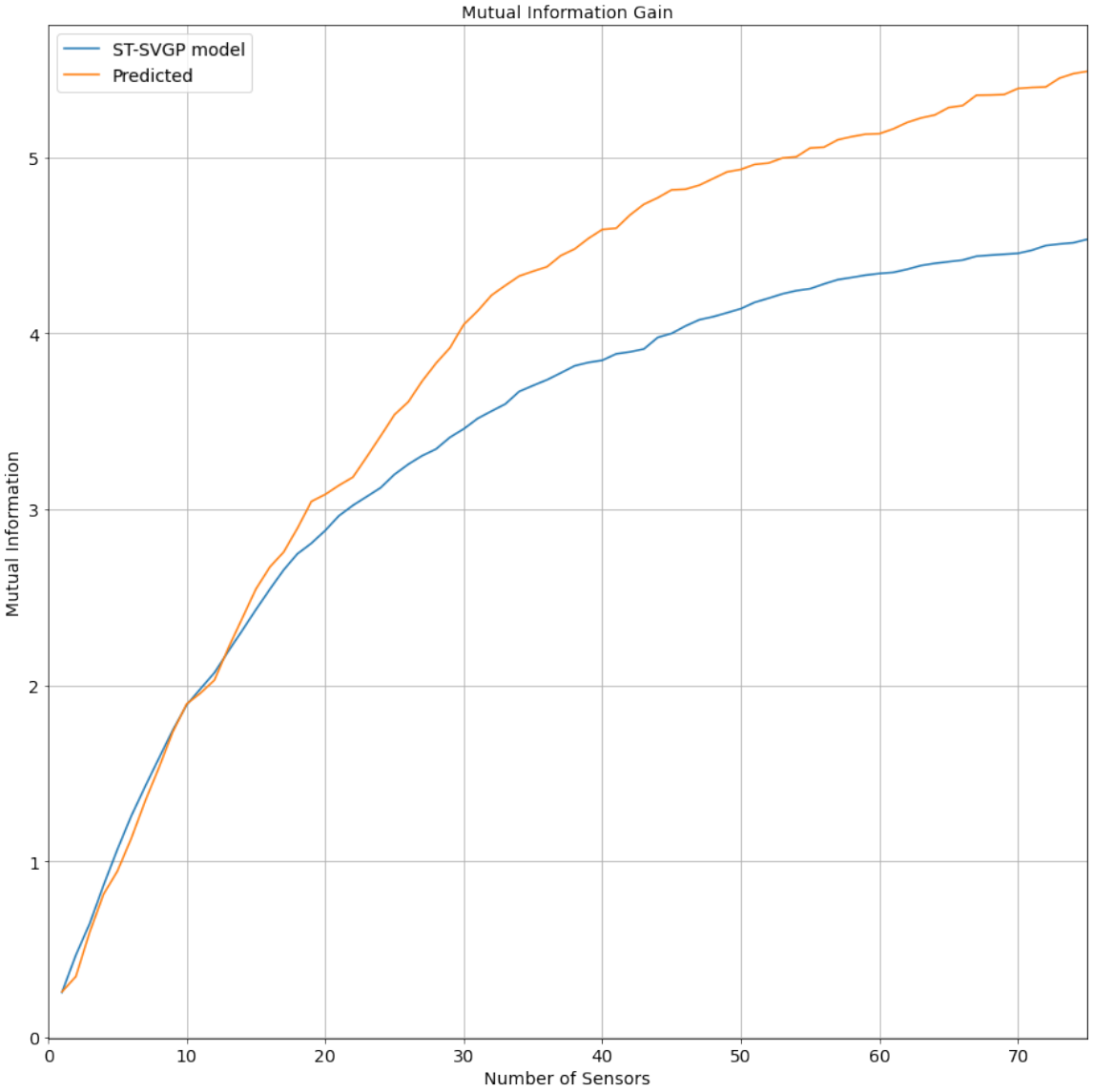

Figure 3-11: Plot of the total mutual information per sensor in the network for the ST-SVGP model in experiment 2 and its predicted mutual information on the optimized sensor placements

Chapter 4

Future Work

We identify a few directions for potential improvement in our approach.

1. **Multi-pollutant objective.** Most of the sensor sites in our study monitor more than one pollutant. Thus, redistributing the sensors to optimize for NO_2 prediction accuracy and certainty could cause a decline in the information captured for other pollutants. An ST-SVGP model trained to output predictions for multiple pollutants would allow us to optimize the sensor placements of the LAQN while taking into account all monitored pollutants. Multi-output GPs (and GPs in general) are a very active area of research.

2. **Feature engineering.** In our study, we used timestep, latitude, and longitude for the features of our input data. Adding more insightful futures to the input data could help an ST-SVGP train and improve its performance. For example, we could add features representing month, season, day/night, weekday/weekend, or altitude. We can also increase the granularity of our data by using average hourly measurements.

3. **Environmental justice.** We can combine census data with our ST-SVGP model to inform and address environmental justice issues. Figure 4-1 shows a bivariate heatmap of the uncertainty of the model generated in experiment 2 and economic inactivity by London ward. This map allows us to identify equity issues in access to reliable air pollution monitoring. White wards show

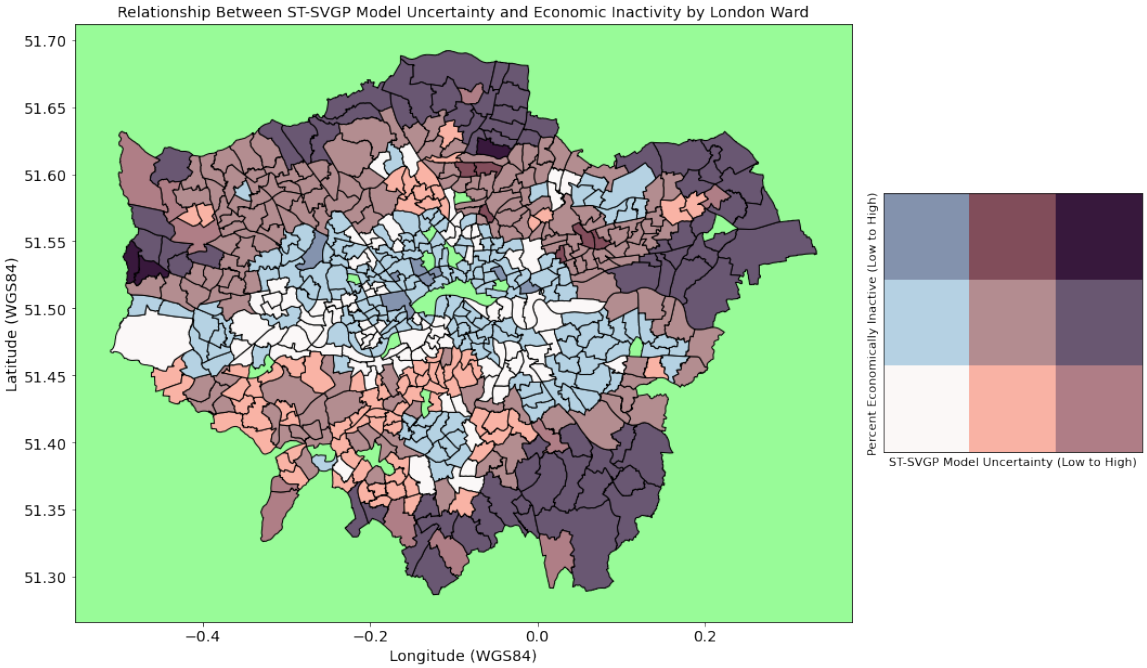

Figure 4-1: A bivariate heatmap of the uncertainty of the ST-SVGP model generated in experiment 2 and economic inactivity by London ward

where economic inactivity and model uncertainty are both low, blue wards show where economic inactivity is high and model uncertainty is low, red wards show where economic inactivity is low and model uncertainty is high, and purple wards show where economic inactivity and model uncertainty are both high. A modified Krause's algorithm and/or mutual information definition could weigh disadvantaged areas such that they are prioritized when selecting sensor placements.

Chapter 5

Conclusion

In conclusion, we demonstrated a viable framework for optimizing existing sensor networks. We combined ST-SVGPs and Krause's algorithm to optimize spatially an existing low-cost sensor network for air pollution in London. We demonstrated the practical utility of these combined methods using a cross-validation approach, applied to air pollution data obtained from 75 sensors within the London Air Quality Network (LAQN) in 2011. We found (on average) that our methods increase the mutual information captured by a simulated sub-network across the area of London by 27.3% while maintaining the same performance on prediction root mean square error within a third subset of the original data (the validation set). We then applied this procedure to a model trained on all 75 sensors to generate an optimized redistribution of the LAQN air pollution sensors with a predicted 21.2% improvement.

Appendix A

Optimized LAQN NO_2 Sensor Coordinates

Selection Order	Site Code	Latitude	Longitude	Selection Order	Site Code	Latitude	Longitude
1	L1	51.366667	0.066667	39	L28	51.366667	-0.066667
2	L2	51.566667	0.2	40	HS6	51.479129	-0.36476
3	L3	51.666667	-0.133333	41	L29	51.466667	0
4	L4	51.333333	-0.066667	42	L30	51.533333	0.066667
5	L5	51.5	0.133333	43	L31	51.433333	-0.2
6	L6	51.533333	-0.4	44	L32	51.466667	-0.266667
7	L7	51.633333	-0.066667	45	BX2	51.49061	0.158914
8	HR1	51.617327	-0.298775	46	L33	51.533333	-0.2
9	ST3	51.365318	-0.165936	47	IS2	51.555378	-0.116146
10	BN2	51.591901	-0.205992	48	EA1	51.513323	-0.307529
11	L8	51.566667	0.066667	49	WM0	51.494681	-0.131938
12	L9	51.433333	-0.133333	50	L34	51.466667	-0.2
13	L10	51.533333	0.2	51	HK6	51.526454	-0.08491
14	L11	51.4	0	52	L35	51.533333	-0.266667
15	L12	51.566667	-0.466667	53	GN0	51.490532	0.074003
16	L13	51.433333	-0.333333	54	TH4	51.515046	-0.008418
17	L14	51.6	0.133333	55	L36	51.3	-0.133333
18	L15	51.4	-0.266667	56	L37	51.6	0.066667
19	L16	51.4	0.133333	57	LW1	51.445468	-0.020266
20	GB6	51.4563	0.085606	58	KC2	51.495504	-0.178809
21	L17	51.5	-0.4	59	L38	51.566667	0.133333
22	ST5	51.389287	-0.141662	60	BL0	51.522287	-0.125848
23	L18	51.6	-0.4	61	L39	51.4	-0.133333
24	L19	51.633333	-0.2	62	RB4	51.57661	0.030858
25	HS7	51.463402	-0.427525	63	HS4	51.492507	-0.257252
26	L20	51.6	0	64	CT3	51.513847	-0.077766
27	BX1	51.465983	0.184877	65	L40	51.566667	-0.2
28	L21	51.566667	-0.333333	66	L41	51.466667	0.066667
29	L22	51.566667	-0.066667	67	L42	51.6	-0.266667
30	L23	51.333333	0.066667	68	GR5	51.483449	-0.000146
31	L24	51.6	-0.133333	69	WA9	51.465033	-0.215825
32	NM2	51.537598	-0.002138	70	ST6	51.377923	-0.240414
33	EA2	51.506628	-0.267934	71	L43	51.533333	-0.133333
34	L25	51.466667	-0.066667	72	EI1	51.523608	-0.265503
35	L26	51.5	0	73	L44	51.566667	0.266667
36	BT5	51.552656	-0.248774	74	HS5	51.489398	-0.310081
37	HI0	51.496309	-0.460826	75	KC3	51.49914	-0.164338
38	L27	51.433333	0.066667				

Table A.1: Table of the optimized sensor placements

Bibliography

[1] "Air pollution," *Health Topics*. [Online]. Available: https://www.who.int/health-topics/air-pollution

[2] B. Brunekreef and S. T. Holgate, "Air pollution and health," *The Lancet*, vol. 360, no. 9341, pp. 1233–1242, 2002. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0140673602112748

[3] M. Kampa and E. Castanas, "Human health effects of air pollution," *Environmental Pollution*, vol. 151, no. 2, pp. 362–367, 2008, proceedings of the 4th International Workshop on Biomonitoring of Atmospheric Pollution (With Emphasis on Trace Elements). [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0269749107002849

[4] J. Martinez, "Great smog of london," *Encyclopædia Britannica*. [Online]. Available: https://www.britannica.com/event/Great-Smog-of-London

[5] "About londonair," *London Air Quality Network*. [Online]. Available: https://www.londonair.org.uk/LondonAir/General/about.aspx

[6] "What is the laqn?" *London Air Quality Network*. [Online]. Available: https://www.londonair.org.uk/LondonAir/Guide/WhatIsLAQN.aspx

[7] P. Kumar, L. Morawska, C. Martani, G. Biskos, M. Neophytou, S. Di Sabatino, M. Bell, L. Norford, and R. Britter, "The rise of low-cost sensing for managing air pollution in cities," *Environment International*, vol. 75, pp. 199–205, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0160412014003547

[8] W. Jiao, G. Hagler, R. Williams, R. Sharpe, R. Brown, D. Garver, R. Judge, M. Caudill, J. Rickard, M. Davis, L. Weinstock, S. Zimmer-Dauphinee, and K. Buckley, "Community air sensor network (cairsense) project: evaluation of low-cost sensor performance in a suburban environment in the southeastern united states," *Atmospheric Measurement Techniques*, vol. 9, no. 11, pp. 5281–5292, 2016. [Online]. Available: https://amt.copernicus.org/articles/9/5281/2016/

[9] M. R. Giordano, C. Malings, S. N. Pandis, A. A. Presto, V. McNeill, D. M. Westervelt, M. Beekmann, and R. Subramanian, "From low-cost sensors to high-quality data: A summary of challenges and best

practices for effectively calibrating low-cost particulate matter mass sensors," *Journal of Aerosol Science*, vol. 158, p. 105833, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0021850221005644

[10] A. J. Boghozian, "An exercise in selecting low-cost air quality sensor placements within an urban environment," *Massachusetts Institute of Technology*. [Online]. Available: https://hdl.handle.net/1721.1/130793

[11] D. C. Carslaw and K. Ropkins, "openair — an r package for air quality data analysis," *Environmental Modelling Software*, vol. 27–28, no. 0, pp. 52–61, 2012.

[12] "Data feeds for london air quality," *London Air Quality Network*. [Online]. Available: https://www.londonair.org.uk/Londonair/API/

[13] T. pandas development team, "pandas-dev/pandas: Pandas," Feb. 2020. [Online]. Available: https://doi.org/10.5281/zenodo.3509134

[14] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[15] A. Krause, A. Singh, and C. Guestrin, "Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies," *Journal of Machine Learning Research*, vol. 9, pp. 235–284, 02 2008. [Online]. Available: https://jmlr.org/papers/volume9/krause08a/krause08a.pdf

[16] O. Hamelijnck, W. J. Wilkinson, N. A. Loppi, A. Solin, and T. Damoulas, "Spatio-temporal variational gaussian processes," 2021. [Online]. Available: https://arxiv.org/abs/2111.01732

[17] O. Hamelijnck, W. Wilkinson, N. Loppi, A. Solin, and T. Damoulas, "Spatio-temporal variational Gaussian processes," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. [Online]. Available: https://github.com/AaltoML/spatio-temporal-GPs

[18] R. Kleeman, "Information theory and predictability lecture 7: Gaussian case," *Courant Institute of Mathematical Sciences*, 2017. [Online]. Available: https://www.math.nyu.edu/faculty/kleeman/