Modelling calibration uncertainty in networks of environmental sensors

Citation for published version (APA):
Smith, M. T., Ross, M., Ssematimba, J., Alvarado, P. A., Alvarez, M., Bainomugisha, E., & Wilkinson, R. (2022). *Modelling calibration uncertainty in networks of environmental sensors*.

Citing this paper
Please note that where the full-text provided on Manchester Research Explorer is the Author Accepted Manuscript or Proof version this may differ from the final Published version. If citing, it is advised that you check and use the publisher's definitive version.

OPEN ACCESS

Modelling calibration uncertainty in networks of environmental sensors

Michael Thomas Smith[1] | **Magnus Ross**[2] | **Joel Ssematimba**[3] | **Pablo A. Alvarado**[1] | **Mauricio Álvarez**[2] | **Engineer Bainomugisha**[3] | **Richard Wilkinson**[4]

[1]Department of Computer Science, University of Sheffield, UK

[2]Department of Computer Science, University of Manchester, UK

[3]Department of Computer Science, Makerere University, Kampala, Uganda

[4]School of Mathematical Sciences, University of Nottingham, UK

Correspondence
Michael Thomas Smith, Department of Computer Science, University of Sheffield, Sheffield, S1 4DP, UK
Email: m.t.smith@sheffield.ac.uk

Funding information
google.org.
EPSRC GCRF grant, EP/T00343X/1.

Networks of low-cost sensors are becoming ubiquitous, but often suffer from poor accuracies and drift. Regular colocation with reference sensors allows recalibration but is complicated and expensive. Alternatively the calibration can be transferred using low-cost, mobile sensors. However inferring the calibration (with uncertainty) becomes difficult. We propose a variational approach to model the calibration across the network. We demonstrate the approach on synthetic and real air pollution data, and find it can perform better than the state of the art (multi-hop calibration). We extend it to categorical data produced by citizen-scientist labelling. In Summary: The method achieves uncertainty-quantified calibration, which has been one of the barriers to low-cost sensor deployment and citizen-science research.

KEYWORDS

air pollution, Bayesian modelling, calibration, Gaussian processes, low-cost sensors, variational inference

1 | INTRODUCTION

Large networks of low-cost sensors are becoming increasingly common across a range of domains, including weather (van de Giesen et al., 2014), snow-cover (Pohl et al., 2014), wildlife (Dyo et al., 2010) and air pollution (Khedo et al., 2010; Liu et al., 2020). The aim is to collect far more data at low-cost. Unfortunately, across all these domains is the common issue of how to ensure low-cost sensors remain calibrated, while still achieving substantial cost-savings.

Citizen science data collection initiatives often have similar issues, necessitating validation by experts, replication and statistical modelling (see Kosmala et al., 2016, for a review). In particular, the problem of maintaining calibration over time, in low cost sensor networks, typically requires regular colocation recalibration (Rai et al., 2017).

Considerable work has already been invested in developing approaches for the reliable use of low-cost sensors for air pollution monitoring (see Section 1.2). Rai et al. (2017) emphasise that the key issue with the use of low-cost sensors is data quality: For the network to have some use for policy makers the sensors need frequent calibration in 'final deployment conditions' such that sensor calibration drift should be accounted for.

This work is motivated by the need to manage the calibration of a network of low-cost PM2.5, air pollution sensors deployed in Kampala, Uganda. Kampala is a low-income, East African city, associated with a tropical, high-humidity, dusty environment; leading to additional issues with sensor reliability and drift. Calibration in this network is achieved through regular but brief visits to the static (low-cost and reference) sensors, by mobile low-cost sensors mounted on motorbike taxis. The motorbike taxi visits are often opportunistic and usually last less than ninety minutes. This leads to a complex network of in-situ colocation events. Our experience is that drift (due to dust, etc) experienced in Kampala is much greater than similar sensors running in the UK, suggesting that this should be included in any model. We found that most of the research on pollution monitoring were for cities in temperate climates. Very few environmental monitoring networks are on the continent of Africa (Mao et al., 2019). Hopefully the approach described will help guide future deployments across the continent.

1.1 | The task

We have a set of particulate (PM2.5) sensors. Some are mobile and some are static. Some of the static sensors are 'reference' instruments which we assume measure the true pollution. The rest are low-cost sensors. We assume that the true pollution is a parametric function, $\phi(\text{raw measurement}, \text{parameters})$, of the raw observation from the low-cost sensor. The task is to estimate this calibration function (or rather its vector of parameters) for each sensor, for any given time, to allow us to use the low-cost sensors to estimate the true air pollution in locations where there is no reference instrument.

The full problem consists of both estimating the calibration functions and modelling the pollution over space and time across the city. Modelling these two aspects jointly may have advantages, but we focus in this paper on simply solving the calibration problem, where each sensor-pair colocation occurs at an arbitrary, undefined spatial location.

For a given sensor, s at time, t, we wish to find the value of the calibration function's parameter vector, $f_s(t)$. We will estimate this using a list of colocations. Each colocation records: (a) The time of the colocation (b) The id of the two sensors (c) The raw (uncalibrated) measurements of the two sensors. To illustrate with an example, the calibration function could be a simple linear regression with two parameters (an offset and a gradient), such that: true pollution = raw measurement × gradient + offset. The gradient and offset make up a vector of two parameters, which we wish to estimate. These we believe can change over time, and differ between sensors. We want a method that: (1) Can infer the calibration function's parameters even if the associated sensor has not been directly visited by a reference sensor. (2) Can combine calibration information arriving through multiple pathways of calibration. (3) Uses both static and mobile sensor data. (4) Has a reasonable model of how the measurement is related to the true pollution. (5) Provides principled estimates of the uncertainty in the calibration, taking into account time.

1.2 | Background

A review by Delaine et al. (2019) divides the calibration literature in several ways, considering reference instrumentation, sensor mobility, types of calibration (time dependency and complexity) and 'grouping strategies'. Calibration can occur prior to deployment or can occur in-situ by finding those times in which pollution is believed to be similar across large distances, for example when few local sources of pollution exist at night (e.g. Miskell et al., 2018). We are interested in those approaches which use physical post-deployment colocation.

Consider first Arfire et al. (2015). They have a model-based approach which assumes a low-cost chemical sensor is calibrated by co-location with a reference sensor regularly. This doesn't then handle the more complex network that we are analysing (in which colocations between low-cost sensors also occurs). They consider temporal drift by either a linear or a hyperbolic term with respect to time. There is no indication that they compute or assess the uncertainty in the model's predictions.

Of more interest to our application are studies looking at networks of sensors such that the low-cost sensors are not necessarily directly calibrated by the reference instrument but potentially by intermediate sensors (e.g. mobile sensors). Tsujita et al. (2004) considers one such model. However they make a series of assumptions (e.g. how the sensor calibration drift is a fixed linear number). The calibration is also done in a non-probabilistic approach; with an estimate for the pollution at a location simply the average of all sensors, even though some will, presumably, be more reliable than others (e.g. if they have just visited the reference station). The model also doesn't report its confidence. Setting aside these issues, the paper does suggest that a network approach to calibration might be of benefit. Xiang et al. (2012) use the uncertainty in a given sensor's calibration when combining measurements. The problem this paper considers largely matches ours. One major difference is the error model. In our experience the calibration we are concerned with involves scaling of the true pollution, not just an offset. One potentially could alter their paper by using the log of sensor measurements, which will then make their sum a product. They also require that the calibration computation and data remains local to the device, while we are computing this using all the data from all sensors. Markert et al. (2018) assign weights to various estimates but, as with almost all the papers, the correlations between the calibration estimates are not considered. Their treatment is partly Bayesian (for example in how different uncertainties are combined), but has a single value for drift (subtracting the calibrated reading from the raw reading). Also by only looking at previous colocations, it fails to propagate all the calibration information available. We suggest therefore that there are likely to be considerable improvements in calibration accuracy and uncertainty quantification by using a process model of calibration over time, which will be able to leverage prior knowledge around temporal structure in calibration functions. A fully Bayesian treatment will also be able to handle more complex measurement functions. These criticisms aside, Markert et al. (2018) provide a useful foundation and demonstration of the benefits of a network of sensors. Another related work (Hasenfratz et al., 2012) looks at a similar problem, this time using a polynomial fit for the sensor data (rather than just adding an offset). They handle multiple sensors by computing weighted sums, with weights dependent on the time difference (between query and calibration) although not in a particularly probabilistic manner. The paper computes a single ML estimate for the calibration.

Most other papers (e.g. Maag et al., 2017; Kizel et al., 2018; Bychkovskiy et al., 2003; Fonollosa et al., 2016) do not consider drift over time. Many, (e.g. Maag et al., 2017; Bychkovskiy et al., 2003) do not handle uncertainty in their calibration estimates. Finally note that most of the papers mentioned in this section assume linear/scaling-only calibration. In summary, there is currently no framework for performing calibration across an arbitrary network of sensor-rendezvous events which probabilistically handles the quantification of uncertainty in the calibration with temporal drift. The method described in this paper correctly estimates the uncertainty in calibration across a complex network consisting of mobile, static, low-cost and reference instruments.
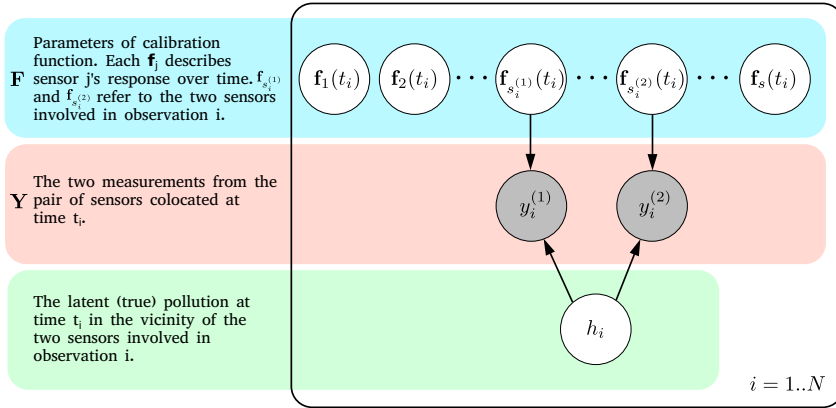
FIGURE 1 Calibration pair model. *A priori* the vectors of GP random variables in $\mathbf{F}$ are independent, but become correlated in the posterior due to colocated observations connecting them together in pairs.

1.2.1 | Related work in probabilistic modelling

We did originally explore a joint model, in which both the calibration parameters <u>and the pollution</u> were modelled as Gaussian processes. This had potentially more power as non-colocated (but still correlated) measurements could still inform the estimates of the calibration, but it was found to be somewhat intractable, and hence we focus in this paper on the calibration pair model (in which we treat the colocations independently of their spatial locations) and leave the joint model for future work.

It is however worth considering the necessary tools for inference in the joint model, as it connects with other literature. The joint model involved a calibration function parameterised by a Gaussian process (modelling, e.g. the calibration offset) operating on another Gaussian process (the true pollution over time and space). This is reminiscent, if the calibration function was simply scaling the pollution, of Wilson et al. (2012), in which a weighted sum of GPs is computed, with the weights consisting of GPs themselves. They experimented with both MCMC (using elliptical slice sampling) and variational EM to perform the inference on the latent variables. We experimented with using MCMC for inference in both models (see supplementary) but found mixing was challenging as the network grew larger. They noted that even with their relatively simple model, Gibbs sampling would 'mix poorly because of the tight correlations between the weights and the nodes'. We found that even with whitening, HMC and other tricks, the strong correlations in the posterior made it very difficult to perform inference. Alvarado and Stowell (2017) more recently computed a similarly structured product of Gaussian processes, and used Gauss-Hermite quadrature. In our experiments we used simple Monte Carlo to compute the expectations, following the approaches in Hensman et al. (2013) and Salimbeni and Deisenroth (2017).

1.3 | Calibration pair model

1.3.1 | Model definition

Each of the $i = 1..N$ observations, $\mathbf{Y} \in \mathcal{R}^{N \times 2}$, consist of two measurements, $y_i^{(1)}, y_i^{(2)}$, by two sensors with indices $s_i^{(1)}, s_i^{(2)}$, colocated at time t_i. One could, in principle, have more than two sensors colocated, but for simplicity in representation and likelihood function we have used pair-wise colocation. The model assumes that a de-

terministic parametric function, ϕ, describes how a measurement is related to the true pollution: true pollution = ϕ(raw measurement, parameters) + unstructured noise.

For each sensor, j, the calibration function's parameter vector consists of C parameters that are each functions of time: $\boldsymbol{f}_j(t)$. Our model assigns an independent Gaussian process prior (over time) to each of these $c = 1..C$ parameters for each of the $j = 1..S$ sensors. So, let $\boldsymbol{F}(t)$ be an $S \times C$ matrix of <u>independent</u> Gaussian processes $[f_j]_c(t) \sim GP(0, k_{j,c})$: We are assuming, *a priori*, that the calibration parameters of the sensors are independent between sensors and the parameters are also independent within sensors. In our implementation we have different kernels for sensors of different types and for different parameters, allowing us to model, for example, that one sensor type's offset drifts at a slower rate than its scaling, etc. The aim of the method is to infer the values of $\boldsymbol{F}(t)$.

We emphasise that, in this model we are discarding location information and just using information about which sensor pairs are colocated, and when.

1.3.2 | The likelihood

Here, by likelihood, we mean the probability of the observations, $\boldsymbol{Y}$, given the model, its hyper-parameters and any latent variables, $\boldsymbol{F}$. To compute $\log p(\boldsymbol{Y}|\boldsymbol{F})$ we first assume that this can be factorised across observations: $p(\boldsymbol{Y}|\boldsymbol{F}) = \prod_{i=1}^{N} p(\boldsymbol{y}_i|\boldsymbol{F})$. Where $p(\boldsymbol{y}_i|\boldsymbol{F})$ is the probability of one pair of colocated observations, given our model and latent calibration parameters. We pick from $\boldsymbol{F}$ the two rows associated with the two sensors $s_i^{(1)}$ and $s_i^{(2)}$ that took the two measurements. So the calibrated predicted pollution from the two observations are, $\phi(y_i^{(1)}, \boldsymbol{f}_{s_i^{(1)}}(t_i))$ and $\phi(y_i^{(2)}, \boldsymbol{f}_{s_i^{(2)}}(t_i))$.

Let h_i be the (latent) true pollution associated with the colocated observation pair $\boldsymbol{y}_i$. We assume that these calibration-corrected predictions of the pollution are normally distributed around the true (latent) pollution, i.e.:

$$p\left(\begin{bmatrix} \phi(y_i^{(1)}, \boldsymbol{f}_{s_i^{(1)}}(t_i)) \\ \phi(y_i^{(2)}, \boldsymbol{f}_{s_i^{(2)}}(t_i)) \end{bmatrix} \middle| h_i\right) \sim N\left(\begin{bmatrix} h_i \\ h_i \end{bmatrix}, \begin{bmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{bmatrix}\right). \tag{1}$$

If we place a wide, fairly uninformative, normal prior on h_i (with mean zero, and variance γ^2), we can integrate out h_i in closed form. Allowing us to write the probability of a given (calibrated) observation pair as a 2d joint multivariate Gaussian,

$$p\begin{pmatrix} \phi(y_i^{(1)}, \boldsymbol{f}_{s_i^{(1)}}(t_i)) \\ \phi(y_i^{(2)}, \boldsymbol{f}_{s_i^{(2)}}(t_i)) \end{pmatrix} = \int p\left(\begin{bmatrix} \phi(y_i^{(1)}, \boldsymbol{f}_{s_i^{(1)}}(t_i)) \\ \phi(y_i^{(2)}, \boldsymbol{f}_{s_i^{(2)}}(t_i)) \end{bmatrix} \middle| h_i\right) \times N\left(h_i \middle| 0, \gamma^2\right) \; dh_i \tag{2}$$

$$\sim N\left(0, \begin{bmatrix} \sigma^2 + \gamma^2 & \gamma^2 \\ \gamma^2 & \sigma^2 + \gamma^2 \end{bmatrix}\right). \tag{3}$$

For inference using Hensman et al. (2013) we need an expression for the conditional probability of the observations $\boldsymbol{y}$ given the latent calibration $\boldsymbol{f}$. So we need to perform a change of variable, to covert from the distribution of $p(\phi(\boldsymbol{y}, \boldsymbol{f}))$ to $p(\boldsymbol{y}|\boldsymbol{f})$. To do this we multiply the probability of a vector of calibrated observations, ϕ, by the deter-

minant of the vector's Jacobian with respect to y (which is diagonal in this case), evaluated at the two observations.

$$p\begin{pmatrix} y_i^{(1)} \\ y_i^{(2)} \end{pmatrix}\bigg| F\end{pmatrix} = p\begin{pmatrix} \phi(y_i^{(1)}, f_{s_i^{(1)}}(t_i)) \\ \phi(y_i^{(2)}, f_{s_i^{(2)}}(t_i)) \end{pmatrix} \times \frac{\partial \phi(z_1, f_{s^{(1)}}(t_i))}{\partial z_1}\bigg|_{z_1 = y_i^{(1)}} \times \frac{\partial \phi(z_2, f_{s^{(2)}}(t_i))}{\partial z_2}\bigg|_{z_2 = y_i^{(2)}} \tag{4}$$

For implementation, we use the log probabilities, and so the product becomes the sum of the log probability of ϕ and the log of the two partial derivatives.

Reference Instruments We define a binary vector r, which describes which sensors are reference sensors. In those cases, we use the identity function instead of the calibration function, i.e. replacing $\phi(y_i^{(a)}, f_{s_i^{(a)}}(t_i))$ with $y_i^{(a)}$.

1.3.3 | Variational inference for the calibration pair model

We wish to compute the calibration parameters from the observations, $p(F|Y)$. A simple application of Bayes rule, $p(F|Y) = p(Y|F)p(F)/p(Y)$, is not soluble or tractable in closed form, as N may be large and the likelihood function is non-Gaussian. We quickly step through the variational approach, largely based on (Hensman et al., 2013). We approximate the posterior, $p(F|Y)$ with a variational distribution over F: $q(F)$. We want to make this distribution similar to the true posterior. To this end we aim to minimise the KL divergence between the two:

$$\mathcal{D}_{KL}\Big[q(F)||p(F|Y)\Big] = -\int q(F)\log\frac{p(F|Y)}{q(F)}dF \tag{5}$$

$$= -\int q(F)\log\frac{p(F,Y)}{p(Y)q(F)}dF \tag{6}$$

$$= -\int q(F)\log\frac{p(F,Y)}{q(F)}dF + \int q(F)\log p(Y)dF \tag{7}$$

$$= \underbrace{-\int q(F)\log\frac{p(F,Y)}{q(F)}dF}_{\text{The ELBO}} + \log p(Y) \tag{8}$$

The last term is constant wrt the variational distribution so we can minimise the KL divergence simply by maximimising the integral (the evidence lower bound, ELBO). This is still not tractable without the approximation provided by inducing points. We introduce an additional vector u of values that we assume can describe F, evaluated at inducing point locations, Z. Each row of Z has a time, and a sensor id/parameter id, to index the associated GP in F. We found that spacing the inducing points evenly over the sensor domains, rather than being optimised works well for the datasets we experimented with. We can compute the posterior over both F and u (we assume that Y is conditionally independent of u given F), $p(F, u|Y) \propto p(Y|F)p(F|u)p(u)$. We don't compute this directly, but instead use our approximation, $q(F, u|Y) = p(F|u)q(u)$. Here we have assumed that u is a sufficient statistic to determine F so that $p(F|u, Y) = p(F|u)$. We substitute this approximation into the ELBO and augment with the inducing point values,

$$\mathcal{L} = \int\int q(F, u)\log\frac{p(F, u, Y)}{q(F, u)}dF du \tag{9}$$

$$= \int\int q(F, u)\log\frac{p(Y|F)p(F|u)p(u)}{p(F|u)q(u)}dF du \tag{10}$$

$$= \int\int q(F, u)\log p(Y|F)dF du + \int\int q(F, u)\log\frac{p(u)}{q(u)}\,du\,dF \tag{11}$$

The second term's integral over F integrates out, leaving:

$$\mathcal{L} = \int \int q(F, u) \log p(Y|F) dF du + \int q(u) \log \frac{p(u)}{q(u)} du \qquad (12)$$

$$= \mathbb{E}_{q(F,u)} \Big[\log p(Y|F) \Big] - \mathcal{D}_{KL} \Big[q(u) || p(u) \Big] \qquad (13)$$

We now maximise $\mathcal{L}$ wrt the parameters of the variational distribution, described by the mean, m and triangular matrix R (covariance = $RR^\top$), using stochastic gradient descent. Specifically we approximate the expectation by sampling from $q(F)$ and then computing the mean of the log likelihoods for all the samples. We use TensorFlow's automatic differentiation to maximise $\mathcal{L}$ wrt the variational parameters, m and R using this stochastic approximation to the true gradient (stochastic gradient descent). This formulation also allows us to use minibatches, if N becomes intractably large. See Algorithm 1, in the Supplementary Material for the overview of the computation. Finally we can use the estimates of the variational parameters to predict $q(F)$ for a test point at time t_* for all sensors.

1.3.4 | Optimising hyperparameters

To avoid excessive model complexity we used just four hyperparameters: the lengthscale of static sensors; the length-scale of mobile sensors; likelihood Gaussian noise; and the scale for all kernels. One could treat these as random variables and integrate over them as with other variables. However, a common approach when using Gaussian processes is to find a point estimate either by maximising the marginal likelihood, or in our case, by optimising an error metric in held out data. We used Bayesian optimisation (using the GPyOpt library, GPyOpt authors, 2016) to find such a point estimate, which we used in a model tested on new data. Although we are not now reflecting the uncertainty in the hyperparameters, this point estimate means that the model when deployed in a data processing pipeline, is quicker and more robust. More importantly our lengthscale can reflect our prior belief that a sensor might degrade quickly even if it hasn't yet done so yet in the training data. If such a failure doesn't yet exist in the dataset, the model is liable to select very long lengthscales. By manually selecting lengthscales we can incorporate our domain knowledge in sensor quality and performance.

1.3.5 | Sampling

For the synthetic data example below, we found the algorithm typically optimised the calibration between low-cost sensors first, then would very slowly move these distribution means to match the reference sensors. This behaviour is inevitable due to the highly correlated posterior and the imbalance in samples (with most being between non-reference pairs of instruments). Many elements of the approximating distribution mean must all move together in precisely the right direction, to increase the fit to the reference sensor data. As a side note, this might suggest parametrisations of the approximating distribution's mean vector should be investigated that might allow this manipulation using fewer variables. This problem is compounded by the relatively infrequent reference sensor observations: Many mini-batches contain few or no samples from the reference instrument co-locations. The gradient from the reference sensors only directly applies to those sensors that were co-located with it, while the co-locations of these sensors with the other non-reference collection of sensors will exert a gradient in the opposing direction.

We found for some experiments that importance sampling (Csiba and Richtárik, 2018) largely solved this issue and led to fast, reliable optimisation. We simply oversampled co-locations involving the reference sensors and corrected for the biased sampling by adjusting the noise variance appropriately.

We did not find this a problem in the experiment with real data from Kampala. This might be due to the shallow nature of the network, making it easier for the reference co-locations to influence the variational distribution across the whole network. Part of our preprocessing of the real data involved averaging samples of co-located observations (blocks of 10 observations, typically over 15 minutes, were averaged), such averaging is standard practice in the field (e.g. Stedman et al., 2006). This is necessary due to the considerable noise associated with individual observations (each averaging over only 80 seconds). This may have led to some balancing between the reference-sensors (which make measurements averaging over longer periods) and other sensors, which could explain why importance sampling was unnecessary for the real dataset.

1.3.6 | Other variables

OPC PM2.5 sensors are sensitive to environmental factors. Specifically, a high relative humidity can lead to large biases (e.g. Crilley et al., 2018, in particular figure 6). In this framework such variables could be included by simply adding additional columns to Y. These can then be passed to the calibration function, ϕ, and involved in the parametric expression that computes the 'true' pollution.

1.4 | Calibration over categorical data

To demonstrate the flexibility of the approach, we apply the method to another challenge from our research group: How to combine (low accuracy) citizen-science species-labelling of bee videos to predict the true species (with uncertainty quantification). Some of the videos were also labelled by an experienced and trained researcher, which we could consider ground-truth. Can we use the same calibration approach as above, treating each citizen-scientist as a low-cost sensor and the experienced researcher as a reference sensor? Previously (for sensor j) we were modelling the parameters, $f_j(t_i)$ of a calibration function, $\phi(y, f_j(t_i))$, between measured and true pollution - for example the gradient and offset. In the bee-labelling case, the measured observations are the labels guessed by the citizen scientists. The parameters we model are the probabilities in a conditional probability confusion matrix. These are generated using a softmax function applied to a matrix of latent variables which are given Gaussian process priors, allowing each citizen scientist's labelling to vary over time. Maybe people improve over time, with practice, or forget how to distinguish species over the winter (when fewer bees are foraging).

Previously the two observations $y_i^{(1)}$ and $y_i^{(2)}$ were taken at the same time t_i. In this categorical example we may need two times (one for each person) - as they may have performed the labelling in different orders and at different times, $t_i^{(1)}$ and $t_i^{(2)}$.

To switch to a categorical dataset, the only part of the algorithm that needs to change is our likelihood function defined previously in (4). With number of species = A, we can construct a conditional confusion matrix, P, using $|f_j| = A^2$ parameters. Each element, $P_{y,\psi} = p(y|\psi, f_j)$, of P describes the probability of the observed result <u>given</u> the real bee was of (latent) species, ψ. We build this matrix from the vector of parameters, f_j, by simply reshaping the vector into a square $A \times A$ matrix, C, and use the softmax function to normalise each column, so that $\sum_y P_{y,\psi} = 1$:

$$P_{y,\psi} = e^{C_{y,\psi}(t)} / \sum_z e^{C_{z,\psi}(t)}. \tag{14}$$

The likelihood function takes two observations of the same bee, $y_i^{(1)}$ and $y_i^{(2)}$ and two parameter vectors $f_{s_i^{(1)}}(t_i^{(1)})$ and $f_{s_i^{(2)}}(t_i^{(2)})$, where $s_i^{(1)}$ and $s_i^{(2)}$ are the identities of the sensors (citizen scientists). We compute P for the

two sensors/citizen scientists involved and take the relevant column from each, to get $p(y_i^{(1)}|\psi, \boldsymbol{f}_{s_i^{(1)}}(t_i^{(1)}))$ and $p(y_i^{(2)}|\psi, \boldsymbol{f}_{s_i^{(2)}}(t_i^{(2)}))$ We multiply these together, assuming that there is conditional independence between the citizen scientists, to compute the conditional joint probability,

$$p(y_i^{(1)}, y_i^{(2)}|\psi, \boldsymbol{f}_{s_i^{(1)}}(t_i^{(1)}), \boldsymbol{f}_{s_i^{(2)}}(t_i^{(2)})) = p(y_i^{(1)}|\psi, \boldsymbol{f}_{s_i^{(1)}}(t_i^{(1)})) \, p(y_i^{(2)}|\psi, \boldsymbol{f}_{s_i^{(2)}}(t_i^{(2)})). \qquad (15)$$

Using either the training data, or wider ecological literature, we define a categorical prior distribution over the true species $p(\psi)$. We multiply the conditional joint probability vector with this prior and sum over the resulting matrix, marginalising out the latent species:

$$p\left(y_i^{(1)}, y_i^{(2)}|\boldsymbol{f}_{s_i^{(1)}}(t_i^{(1)}), \boldsymbol{f}_{s_i^{(2)}}(t_i^{(2)})\right) = \sum_{\psi} p\left((y_i^{(1)}|\psi, \boldsymbol{f}_{s_i^{(1)}}(t_i^{(1)})\right) \, p\left(y_i^{(2)}|\psi, \boldsymbol{f}_{s_i^{(2)}}(t_i^{(2)})\right) \, p\left(\psi\right). \qquad (16)$$

As before, we sample values of $\boldsymbol{f}$ from the variational approximation, $q(\boldsymbol{f})$, and compute a Monte Carlo approximation to the likelihood. An autodiff framework, such as tensorflow, again allows us to optimise the variational parameters. In the implementation the only component that changed from the air pollution example was the method for computing the likelihood.

1.5 | Multi-hop instant calibration

For comparison with our calibration pair method, for the air pollution examples we developed and implemented an algorithm loosely based on the Multi-hop Instant Calibration method from Hasenfratz et al. (2012) and the rendezvous connection graph from Saukh et al. (2015), the former paper is a little unclear about combining weights, while the latter one handles sensor drift by only using observations within each time window. In Saukh et al. (2015) a shortest path algorithm is applied (personal communication with authors) to convert the network to an acyclic graph.

We augment the original graph in Saukh et al. (2015) by adding edges between the same sensor, in neighbouring time windows. The intuition being that the scaling information provided by a long series of hops within the window might be less accurate than a direct reference calibration 'carried over' from a different time window. The edges connecting time windows can have a different weight than those connecting sensors in the same time period. The choice of weight ratio (i.e. time-edge-weight/colocation-edge-weight) will decide whether the model leans towards using longer chains of colocations in the same time window (high ratio) vs relying on colocations that occurred a long time ago (low ratio). Figure 2 illustrates. We consider the calibration to be a simple scaling factor in the experiments in this paper and so we restrict the calibration function in both the multi-hop and in our more complex variational inference calibration tool to also scaling-only when performing comparisons.

The implementation of the graph building and prediction are detailed in Algorithms 3 and 4 respectively, in the Supplementary Material.
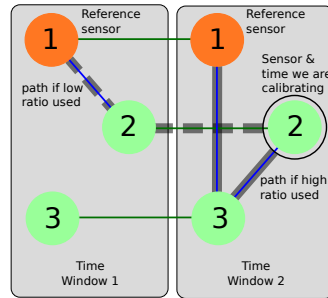
FIGURE 2 Multi-hop: The ratio of edge weights for neighbouring times (green) over weights for colocated sensors (blue) affects the choice of shortest path to a reference instrument. A low ratio will lead to shorter chains of colocation but relying on colocations from different time windows (dashed grey path), a high ratio will lead to a preference for colocations from nearby times, but with longer chains of colocations pairs (solid grey path).

2 | RESULTS

2.1 | Synthetic example

2.1.1 | The synthetic data

Before considering real data we explored the method's response to a representative, but synthetic, dataset. We simulated ten static and four mobile sensors over a 180 day period. Four of the static sensors are assumed to provide noise-free, unbiased, reference measurements, the rest have a time-varying bias and added Gaussian white-noise with variance of 100 $(\mu g/m^3)^2$. In the Kampala data discussed later, each mobile sensor is typically localised to one part of the city, with only occasional visits further afield. We simulate this by making the probability of a visit by a mobile sensor to a static sensor, proportional to the inverse distance between a 'home' and the static sensor. Simple sinusoidal functions were used to generate both the true pollution and the scaling of the mobile and static sensors. The period of the sinusoids was different for each sensor, but the static sensors had a longer period on average (median = 3070 hours vs 1007 hours). This variation was to (a) emphasise the heterogeneous nature of the sensors and (b) illustrate how the mobile sensors are probably less stable. Figure 4 shows the scalings actually used.

2.1.2 | Variational pair model

We applied the method described in Section 1.3 to the synthetic data. As mentioned in Section 1.3.4, we used Bayesian optimisation (30 iterations, optimising the NMSE) to select the model hyperparameters for the variational calibration pair method (kernel variance, likelihood variance and two lengthscales). We optimised using the synthetic data with added noise scale of $10 \mu g m^{-3}$ for all the results. The synthetic data was resampled each iteration of the Bayesian optimisation (and for testing): the sensor locations, colocation events and noise were all resampled; but the frequencies of the synthetic calibration sinusoids remained the same.

Even though each sensor's scaling function has a different period, we use just two lengthscales to model the static and mobile sensors. On the first dataset generated, the Bayesian optimisation chose 2201 hours and 581 hours for the lengthscales of the static and mobile sensors, respectively. This likely reflects both the underlying generating function, but also the strength of evidence for shorter lengthscales. The longer lengthscales assigned to the static sensors is longer than the period of the median sinusoid, but it potentially allows those static sensors with a longer lengthscale to
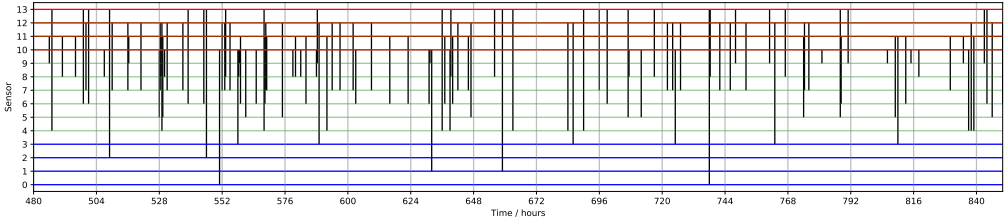
FIGURE 3 Synthetic sensor example: Colocation visits over a representative 350 hours of the total 4320 hours: Visits to the reference (blue) and low-cost (green) static sensors by the mobile ones (red), indicate by vertical black lines.

| Noise Scale / $\mu g\ m^{-3}$		NMSE		NLPD
	No Method	Multi-hop	Cal. Pair Model	Cal. Pair Model
2	0.77 (0.02)	0.27 (0.01)	0.18 (0.01)	42.6 (2.8)
5	0.78 (0.02)	0.44 (0.03)	0.21 (0.01)	49.5 (2.3)
10	0.97 (0.02)	0.87 (0.07)	0.35 (0.01)	58.2 (2.3)
20	1.46 (0.02)	3.18 (0.69)	0.84 (0.02)	87.7 (1.5)

TABLE 1 NMSE using raw data, the multi-hop calibration and the variational calibration pair model, for synthetic data with four levels of added noise. Bracketed values are the standard error of the mean based on predictions generated from 10 synthetic datasets (for each, the multi-hop was optimised on the test data, while the calibration pair model was optimised once on training data with noise scale = 10).

still be modelled well, by allowing more data to be used for any given prediction. It also permits calibration information to be passed over longer periods of time. The likelihood variance and kernel scale (variance) were 6.02 $(\mu g/m^3)^2$ and 7.64 $(\mu g/m^3)^2$ respectively. These reflect the variance of the scaling factor, so it is hard to compare directly to the noise added to the data. The choice of likelihood variance seems to overestimate the true value a little, and it may be that the longer lengthscales chosen for the static sensors have their inaccuracy 'explained away' by this added noise term.

2.1.3 | Multi-hop calibration and comparison

We also applied the multi-hop calibration method to the synthetic data. We first performed a grid search to optimise the window size and the ratio of edge weights between those edges representing time and those representing a colocation (again, performed on separately generated training data). We unfortunately found this model performed very poorly on this data. The best configurations usually had very long window sizes, leading to little or no ability to model the variation over time. We found that this configuration marginally improved on simply using the raw measurements. See Table 1.

The estimates from the variational method provided considerable improvements over the multi-hop approach. And we can also compute the negative log predictive as we predict a distribution. Figure 4 shows these predictions for the 6 non-reference static sensors and the four mobile, for both the multi-hop method and the calibration pair model. One can see some errors in the calibration pair model. In particular, the high frequency drift of the low-cost
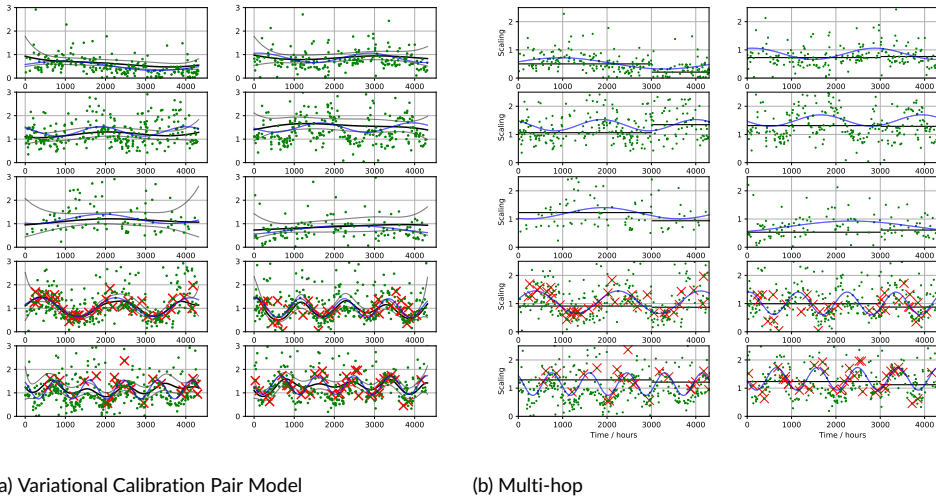
(a) Variational Calibration Pair Model   (b) Multi-hop

FIGURE 4   Predictions for the scaling in the synthetic sensor example for both the Variational Pair Model and the Multi-hop method. Noise scale = 10 $\mu g/m^3$. Upper six plots in each, static sensors. Lower four plots, mobile sensors. Black line, posterior mean. Grey lines, 95% of the posterior density. Blue line, true scaling of each sensor. Green dots, ratio of colocation observation pairs involving this sensor. Red crosses, ratio of colocations observations between this sensor and a reference sensor.

mobile sensor has not been properly detected by the model in all the low-cost sensors, leading to this drift appearing in the predictions for the other sensors.

The synthetic data was engineered specifically to provide low-quality, noisy, infrequent and challenging data, in an attempt to reflect the problems that are typical in real data. Figure 3 illustrates how few visits were made to the reference instruments, over a representative sample of the synthetic data. To explore this further, we added different amounts of noise to the data, and reran the analysis, although we kept the variational calibration method's hyperparameters as those trained with a noise scale of 10 $\mu g\ m^{-3}$, to see how well it adapts to misspecified noise. Table 1 reports the NMSE for synthetic data with four added noise-scales for the two methods. The multi-hop method is allowed to optimise its parameters for each noise scale and, as might be expected, its window size reduces as the noise is reduced (for noise scales of 2, 5, 10 and 20 $\mu g m^{-3}$ the selected window sizes were 292, 1184, 3010 and 4800 hours, respectively). We can also see that the accuracy of the multi-hop method approaches the calibration pair model's, as the noise is reduced. This is probably because the calibration pair model makes use of multiple pathways through multiple colocations and over time to support a prediction, whilst the multi-hop method is constrained to use a single path through the network for a given prediction, making it more vulnerable to the effect of noise. The use of multiple pathways in the pair model gives it considerable robustness against noisy data.

2.2 | Kampala air pollution data

Recently a network of low-cost particulate air pollution sensors has been deployed across Kampala. These consist of pairs of PMS5003 (Plantower) optical particle counters (OPCs) mounted at fixed, static locations and on motorbike taxis, known as boda bodas (Figure 6b). Before discussing the network we will briefly inspect the measurements of
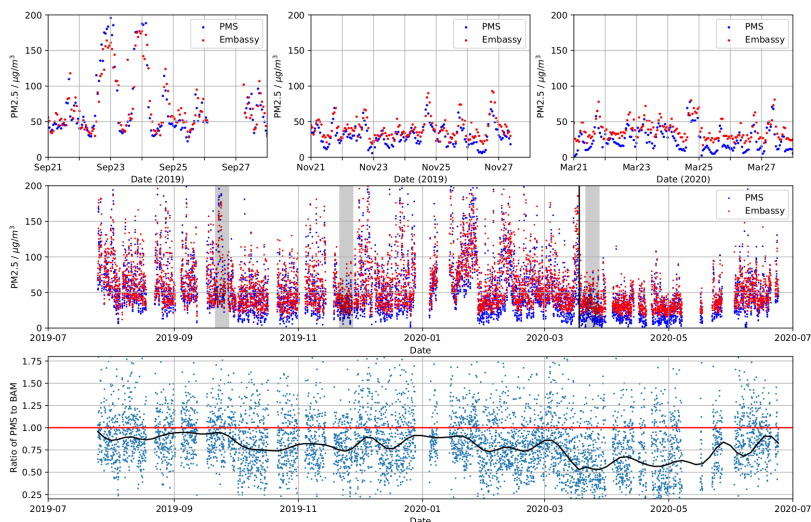
FIGURE 5 Upper plots, enlargements of three weeks (highlighted in grey in the middle plot), showing correlation between the low cost sensor and the reference instrument. Middle plot, the full period of measurements. Discontinuity on 18th March, 2020, due to the covid-19 lock down (vertical black line). Lower plot, the ratio of the two sensors. A GP with RBF kernel has been fitted to the log ratio and the exponent of its posterior mean plotted (pre- and post- lock down modelled with independent GPs). Times are in UTC.

a static OPC mounted 10m from the US Embassy's air pollution monitor (a regularly calibrated BAM sensor, which provides hourly ground truth measurements of PM2.5 pollution). This brief comparison will give some insight into the calibration problem: What the calibration function might consist of, and how it might vary over time.

Figure 5 illustrates the strong correlation between the two sensors, but also highlights some differences. The earlier plot from September seems to show very similar measurements, with a ratio near one, on average. But later measurements seem to have a greater discrepancy with the low cost OPC sensor significantly underestimating the true pollution. The ratio itself is not enough to explain this change. Figure 6a illustrates how an offset can explain some of the difference. In all three the gradient of the linear fit appears to remain between 0.85 and 0.95, however the offset maybe drifts from positive to negative, explaining the increasing disparity in the previous graphs.

2.3 | Calibration over the network

51 sensors were in the network in the second-half of 2020, consisting of 2 reference sensors (BAM 1022 Particulate Monitor, Met One), 4 mobile OPCs and 45 static OPCs (PMS5003, plantower). Between the 15th July, 2020 and 3rd February, 2021 (203 days) there were 6,118,977 measurements of PM2.5 recorded. We use this dataset for the analysis. We built a dataset of 433,935 'encounters', consisting of every pair of measurements that occurred within 40m and 30 minutes of each other. We averaged ten minute periods and removed observations outside the $10 - 300\mu g/m^3$ range, as these we found to sometimes be associated with faults. This left 40,432 records (approximately 200/day). The majority of these are associated with permanently co-located pairs of static sensors. In Figure 7, (a) shows the physical locations of the sensors and (b) the number of colocations between them. The two reference sensors '-24516' (on Makrerere University Campus) and '-24517' (in Nakawa), are approximately 5km apart.
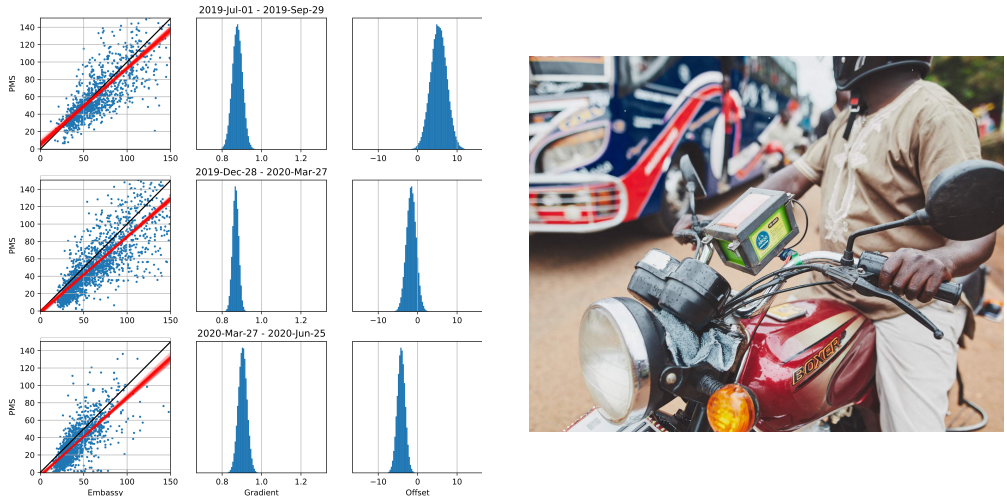
(a) Static OPC measurements vs the embassy data (b) A mobile OPC

FIGURE 6 (left) OPC measurements vs the embassy data. Each row corresponds to 90 days of measurements. The left scatter plot shows the raw connection between the two sensors. A black line indicates equality while red lines are sampled from the linear fit's posterior probability distribution. Middle and right plots are the gradient and offset marginal posterior distributions respectively. The gradient distribution remains roughly constant, while the offset appears to 'drift' downwards over time. (right) One of the mobile low-cost sensors mounted on the front of a motorbike taxis ('boda boda'), photographed in Kampala by AirQo (`www.airqo.net`).
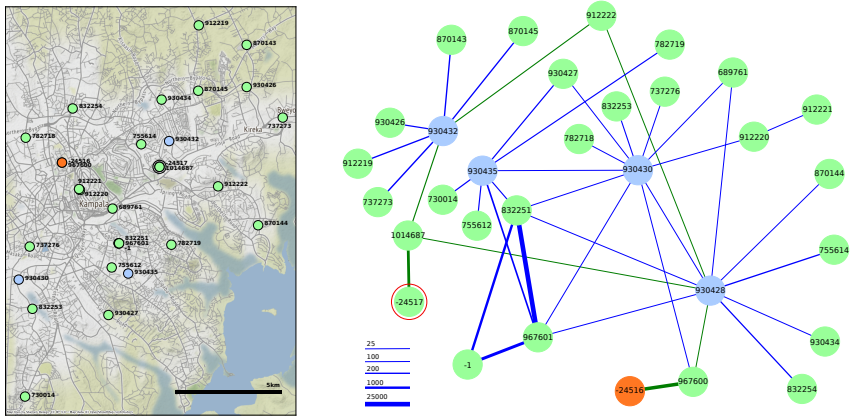
(a) Kampala with sensor locations (b) Colocation frequency graph

FIGURE 7 The AirQo air pollution monitoring network (sensors that have been colocated). The four blue nodes are the mobile sensors. The orange node is the reference sensor at Makerere University, the circled node is the reference sensor whose calibration we are testing. (a) Locations of sensors, the latest locations of mobile sensors are plotted, with one not in the city limits. (b) The network of colocations between sensors (from July 2020 to February 2021). The line thicknesses indicates the number of colocations in that time. The green lines indicate two possible paths from the reference to the test sensor.

Metric	No calibration	Multi-hop calibration	Calibration Pair Model	
MAE	12.71	8.67	5.545 (0.47)	$\mu g\ m^{-3}$
NMSE	0.321	0.180	0.078 (0.011)	

TABLE 2 Kampala experimental results, with synthetic drift added. Using no calibration, the multi-hop calibration and the pair model (bracketed values are standard error from 20 runs. Not required for multi-hop as it's deterministic).

To test the models we only defined -24516 as being a reference sensor and let the models estimate the calibration required of -24517. We gave the models all the colocation data (none was held out) as we are interested in the prediction of the calibration scaling correction for the test sensor. There are co-location observations at the test sensor over a period of 69 days from the 203 days total.

We would expect the test sensor to have a calibration scaling of exactly 1.0 as it's a high quality, carefully maintained and calibrated BAM sensor. Our Gaussian process model has a prior mean of the log-ratio of zero, which is exactly a gain of 1.0, while the multi-hop model doesn't have a prior. To avoid our prior giving an unfair advantage and to simulate the effect of a drifting calibration, as is seen in the low-cost OPCs (Section 2.2), we artificially scaled the measurements of the reference sensor to slowly increase over time (starting at 1.0× on the first day of observations at the test sensor, and increasing to 1.83× by the 69th day). In summary: The algorithms being tested must reconstruct this drift using the network of colocations from the known reference sensor. For our calibration pair model we select hyperparameter values that we believe are appropriate for this dataset (lengthscale for static and mobile sensors = 100 days, RBF kernel variance = 9 $(\log(\mu g/m^3))^2$, Bias kernel variance = 3 $(\log(\mu g/m^3))^2$, likelihood (ratio) noise variance = 0.2 $(\log(\mu g/m^3))^2$). This approach is necessary, as discussed previously, to allow us to include our prior knowledge around lengthscales associated with sensor drift and failure. We again compare to the multihop method described in Section 1.5. For comparison with our method, we optimised the parameters of the competing multi-hop method (the window size and weight ratio between time and colocation) on the dataset we also test with. Effectively giving an upper bound on its capability.

Results After running the grid search optimisation, the optimum multi-hop configuration had a window size of just 24 hours and an edge weight ratio of 0.21 (it roughly equally weights a colocation with a reference sensor from 5 days earlier and via an intermediate sensor during the same 24 hours). We computed error metrics between the test values of the reference sensor and the calibration predictions. To review the aim: The two calibration methods used the other reference instrument and the network of colocations to estimate the calibration of the test instrument (with the added synthetic drift). Table 2 outlines the results for these approaches, while Figure 8 shows the distribution of errors from the different approaches. The 'raw measurement' errors are caused by our artificial scaling (without this the raw measurements would have no error). We find the multi-hop method does help improve the accuracy of the predictions to an extent, but the variational method performs considerably better. Besides accuracy, the variational pair-model method provides uncertainty estimates for its predictions, and the associated 95% confidence intervals include the true scaling function. Variational inference is associated with underestimates of the variance of the posterior (Blei et al., 2017) so this result is reassuring. The co-location data is very noisy. To illustrate this, the green markers in Figure 9 are the ratios between the test sensor and other sensor(s) that visit. Note that we are not assuming that these are just noisy samples from the true latent scaling - as these OPC sensors will themselves have some scaling associated with their estimates. But plotting them gives the reader a sense as to the scale of the noise in the dataset.

Real Calibration As a final brief experiment we tried calibrating the low cost OPC sensor (1014687) that is colocated with the test reference sensor, then using the calibration combined with the OPC observations to predict the
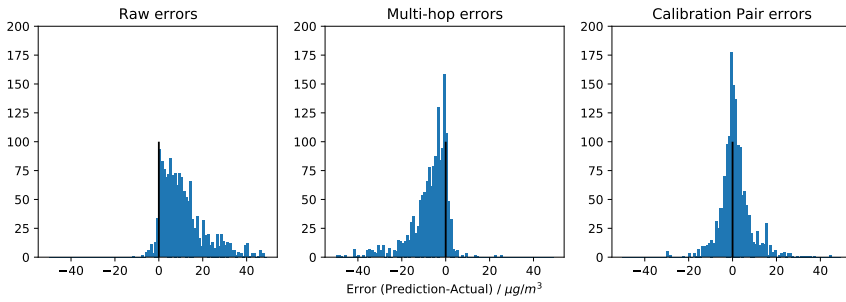
FIGURE 8   The distribution of errors in pollution predictions at the test sensor. As we give the three models the measured pollution, in principle the error could be zero if the model correctly estimates the calibration. The left graph is the distribution of predictions if we just use the measurements as the predictions. The added, synthetic, drift causes this to over-estimate. The central graph is using the multi-hop approach, this appears to underestimate. The right graph is for the calibration pair model. Runs of the variational inference optimisation end with slightly varying results here. This is a typical example (with a MAE about the average).
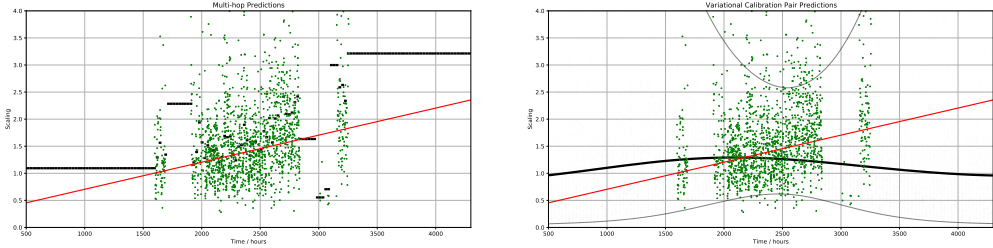
(a) Multi-hop scale predictions

(b) Variational pair model predictions

FIGURE 9   The predictions of the scaling at the test sensor. Red line, our synthetically generated scaling drift we are trying to predict. Black line, the prediction. Green dots, ratios between the test sensor and other sensor(s) that visit. The variational prediction posterior mean varied quite a lot per run of the algorithm, probably due to high uncertainty, leaving the choice of mean difficult to determine.

values measured by the test sensor. The sensor was new and we only have 69 days of data associated. We found the actual scaling was the OPC was underestimating the reference sensor by only about 6% which equates to an error of about 3 μg/m^3. The raw data had a MAE of 15.08 μg/m^3 while the multi-hop and variational pair model approaches had a MAE of 16.07 and 14.87 respectively. Neither approach really had much effect on the MAE. The earlier demonstration with the more significant artificial scaling was however modelled successfully, giving us confidence in the approach.

2.4 | Categorical data

2.4.1 | Synthetic example

We demonstrate the approach initially with a synthetic dataset of 300 bees, approximately evenly split between three species (36%, 30%, 34%). We simulated three non-expert (NE) citizen scientists, and an expert (providing effectively the ground-truth reference). The NE capabilities vary over time, with NE A perfectly labelling the first half, then

Method	Accuracy (%)	NLPD
Calibration (EQ kernel)	80%	53.5
Calibration (Bias kernel)	77%	78.1
Collaborative Filtering	53%	128.1
Most Guessed	74%	81.9
Most Guessed ('trust' weighted)	76%	81.7
Most Guessed (prior weighted)	75%	92.1
Most Common ('chance')	38%	139.5

TABLE 3 For synthetic 'bee' labelling data, we have data from the labelling of three non-experts and a ground truth expert.

degrading to chance linearly over the remaining half. NE B starting at chance, improving linearly to perfect over the first half and providing perfect predictions for the last half. NE C distinguishes between classes {1,2} and 3 initially but, linearly, their classification can distinguish between {1,3} and 2 at half way, and then almost chance by the end.

For training, 173 rows had access to the ground truth, leaving the algorithm to predict the remaining 127. The three NEs (A,B and C) made 178, 190, and 200 observations respectively.

To demonstrate we ran the algorithm using either an EQ kernel (scale=25, lengthscale=25% of the length of the list of images) or a Bias kernel (scale=25). The scale and lengthscales were set *a priori*. The result is a Gaussian process for each of the 27 latent function priors (nine in each confusion matrix for each of the three citizen scientists). For comparison we used a collaborative filtering matrix factorisation approach, in which a binary matrix was first computed for each species (ones are where the species exists in the original dataset). We then used probabilistic matrix factorisation with bias offsets (Hug, 2020) to predict the missing values in the binary matrices for the reference column, then assign the species to the matrix with the greatest value. We also computed more straightforward baselines: 'most guessed' simply does voting, while 'trust weighted' assigns a weight to each citizen scientist. The weights are optimised on the training data. We manually optimised the NLPD for these methods to give them the best chance of beating the calibration pair approach (e.g. we found adding 0.2 to the trust weighted sum of guesses for each class, minimised the NLPD for the synthetic data). We also tried weighting the votes by the prior (number of times each bee appears). We finally just report the most common.

Table 3 shows both calibration pair methods achieve higher accuracy and better uncertainty quantification than the others, but the EQ kernel version substantially improves on this uncertainty estimation.

2.4.2 | Bumblebee video classification

We collected a dataset of 103 slow motion bee videos for a separate project (Ollett, 2021). These were labelled by an experienced and trained researcher 'ground truth' and by six non-experts (NEs). These included five humans and one deep convolutional neural network (trained on photos from a public dataset). See Ollett (2021) for details. *Bombus terrestris* and *Bombus lucorum* were combined into a single class as workers of these species are often visually indistinguishable. Those seven species which appear fewer than three times in the data were removed.[1] This included

---

[1]the list of removed bees: *Bombus bohemicus*, *Bombus jonellus*, *Bombus sylvestris*, *Bombus muscorum*, *Bombus campestris*, *Bombus monticola* and the non-bumblebee *Apis mellifera*.
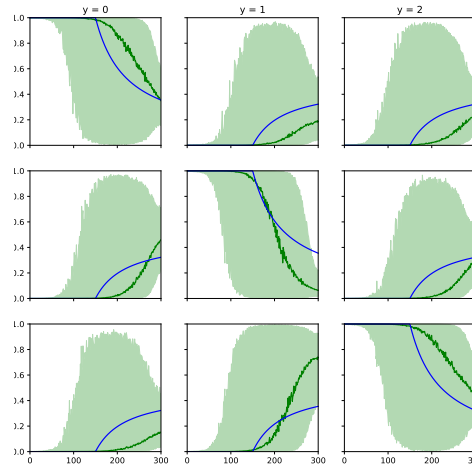
FIGURE 10 For the synthetic categorical dataset: The posteriors for the nine Gaussian processes that describe NE A's predictive ability: Blue line shows the synthetic generating function's 'true' conditional probability, in green is the model's estimate with its 95% CI. The probability refers to the likelihood of observing class y, given the true bee is of class ψ, with each row for a given <u>true</u> bee ($\psi = 0, 1, 2$ respectively), while each column is the reported observed species. The x-axis in each plot indexes the 300 observations.

FIGURE 11 Parts of two frames from a video of a *Bombus hortorum* foraging. In (a) the yellow bands on <u>both</u> its thorax and abdomen are visible. In (b) one can also see the long face and tongue that distinguish it from *Bombus jonellus*. It was sighted in northern England, making it unlikely to be *Bombus ruderatus*. This video was labelled by two non-expert humans who identified it as (1) *Bombus terrestris/lucorum*, (2) *Bombus hortorum* and by the convolutional neural network as *Bombus sylvestris*.

five species which only appeared in the NE labels. The result was 93 bees labelled by at least one NE (221 labels) and by ground truth. The NE accuracies and coverage varied considerably.[2] The order the labelling was conducted was not known, so we decided to just use the Bias kernel. We ran a 5-fold cross validation experiment. In spite of only using the Bias kernel we found the model still performed well. Table 4 shows it achieves higher accuracy and with better uncertainty quantification than the other methods compared.

3 | DISCUSSION

In this paper we've proposed a method to calibrate a network of sensors using observations consisting of pairs of potentially bias or transformed observations in which a few observers are assumed to provide ground truth. We

---

[2] The five human NE accuracies (correct/total): 5/5, 56/60, 12/12, 21/25, 16/20 and the CNN NE: 62/89.

Method	Accuracy (%)	NLPD
Calibration (Bias kernel)	84%	59.9
Collaborative Filtering	58%	inf
Most Guessed	80%	69.4
Most Guessed ('trust' weighted)	82%	91.7
Most Guessed (prior weighted)	79%	98.4
Most Common ('chance')	54%	130.5

TABLE 4 Results for the real bee video labelling data. For context the CNN NE (the only NE to label all the videos) achieved 70% accuracy.

applied this to our main dataset - of static and mobile air pollution sensors in Kampala, adding a synthetic drift term to illustrate the potential issues in longer term data. We found the calibration pair method models the drift appropriately, making the predictions far more accurate. It also, importantly, provides uncertainty estimates on the calibration. For the real data without the synthetic drift, we found it had only a 1.5% improvement in the MAE. Potentially over longer periods, we might expect the approach to become more relevant (as demonstrated by its utility when the drift was added). We also demonstrate how, replacing the likelihood function we can apply the method to a categorical dataset, in which the measurements are labels provided by citizen-scientists. We found the method was effective, and had good uncertainty quantification compared to other approaches.

Comparing the results of the multi-hop method and the variational calibration pair method, it seems that, for larger networks, the latter does better on both the synthetic and real data. It was somewhat surprising that it had much of an advantage in the Kampala case (especially with the synthetic drift added). In simulations we typically found the performance was considerably better than the multi-hop method when there were many sources of data to integrate. In the synthetic dataset for example, there were four reference sensors. So, with only one reference sensor, we anticipated the two approaches to perform similarly. However not only is the number of reference sensors relevant here, but also the number of paths between the reference sensor and the test sensor. Figure 7b indicates two routes exist from the reference sensor to the test sensor in the real dataset. Finally it isn't simply the path from the reference sensor to the test sensor that supports the variational result. The network graph doesn't show the full complexity of data availability over time, but importantly we envisage that the other sensors visited provide support. By also being visited, they can help constrain the calibration, indirectly.

3.1 | Calibration in context

Many studies exist in which low-cost air pollution sensors are co-located with reference sensors, and calibration functions are then derived to, in principle, allow the low-cost sensor to then be used without the reference sensor (Lewis et al., 2016; Liu et al., 2017; Zimmerman et al., 2018; Barcelo-Ordinas et al., 2018; Crilley et al., 2018; Badura et al., 2019; Wang et al., 2019; Datta et al., 2020; Lee et al., 2020; Ferrer-Cid et al., 2020). This can also extend to calibrating remote observation data (for example Shaddick et al. (2018)). It has been noted many times that on-site field calibration is better than lab-based (Rai et al., 2017), and that sensor drift due to ageing or contamination means that regular recalibration is necessary (Crilley et al., 2018; Marathe et al., 2021; Castell et al., 2017, Section 3.2.3; and the results in this paper, Section 2.2). Rai et al. (2017) wrote,

...it is necessary to perform a field calibration for each sensor individually. Moreover, the calibration pa-
rameters might change over time depending on the meteorological conditions and the location, i.e., once
the nodes are deployed it will be difficult to determine if they are under-or over-estimating the pollutant
concentrations.

Although it is widely agreed regular field recalibration is essential, the complexity of achieving this makes it potentially the main barrier to large-scale, low-cost sensor deployment (Rai et al., 2017). Several attempts have been made to look at how to do field-recalibration using mobile reference sensors (Hasenfratz et al., 2012; Arfire et al., 2015). But using low-cost mobile sensors to act as 'intermediaries' (Tsujita et al., 2004; Hasenfratz et al., 2012) is far more practical. The low-cost sensors (e.g. the plantower pms5003) are typically smaller and often will be more resilient to being moved, and can be replaced more cheaply if damaged. In the AirQo Project in Kampala, moving the reference instruments regularly would have been very challenging. An alternative would be to swap the static low-cost sensors between reference colocation sites and their static locations. However such access required site permissions and expertise, while the mobile low-cost sensors could visit the static stations simply by the motor-bike taxi drivers parking next to the sensor. This simplicity provided by permanently mobile low-cost sensors is discussed in Zhao et al. (2021).

It is arguably only worth deploying, calibrating and maintaining the low-cost sensors if they can provide useful information to policy makers. Castell et al. (2017) suggest that they are limited to non-policy/science applications. They however looked at data from a city with relatively low-pollution. The average PM2.5 in Kampala was roughly 30-60 $\mu g/m^3$. The ten minute-averaged sample error in our experiment for the low cost sensor (with a calibration computed across the network) was about $15\mu g/m^3$. Once averaged to daily or annual estimates this would almost certainly be reduced below the 50% threshold in the EU Data Quality Objectives (DQO) (for daily PM2.5 estimates). (Janssen et al., 2017). This is independent of the calibration approach and is probably possible in Kampala simply due to the high background pollution, reflecting the findings of Castell et al. (2017), who found their relative accuracy improved at those times and locations with most pollution. This suggests achieving the EU's DQOs may be easier in high-pollution environments. It would be interesting to consider whether, in high-pollution locations, low-cost sensors could provide estimates of sufficient accuracy that they can be used for policy-making decisions. In-field calibration would still be necessary. By using a probabilistic approach to calibration and spatiotemporal modelling, one should also be able to state where (and when) in the domain the model reaches sufficient confidence.

A related field is the detection of sensor failure in a network of sensors. Miskell et al. (2016) used proxies (reference stations in an area with a similar land use) to detect potential sensor failure. Our method might be useful in the detection of drift, or similar signs of hardware failure. For example combining our method for calibration with a spatiotemporal model with leave one-sensor-out cross-validation. Poor prediction accuracy may be due to poor modelling assumptions or inference, e.g. we haven't included some aspect of the local environment; sensor failure, e.g. the inlet is clogged; or training data errors, e.g. if neighbouring sensors are malfunctioning. For more nuance it might be better to follow other approaches to fault detection. Peng et al. (2017), for example, distinguish between types of fault.

Combining sensor estimates using a Bayesian approach is not novel. Talampas and Low (2012) take into account each sensor's uncertainty to combine their measurements using a Bayesian approach, and find it performs much better than simply averaging. In a different field Perälä et al. (2020) use Gaussian processes for another type of calibration problem: combining expert estimates, potentially with biases using a hierarchical GP. This could be applicable, in particular if different types of sensor can be grouped.

3.2 | The problem of spatially structured bias

Sousan et al. (2016) showed how different sources of particulate pollution lead to different responses by optical particle counters. Castell et al. (2017) and Crilley et al. (2018) also discuss this issue. Sandradewi et al. (2008) propose that one deliberately use the change in the optical properties of the pollution to infer its source. We have assumed that the calibration function remains the same over space, and similar over short periods of time. If the former assumption doesn't hold we will need to colocate the reference instrument at each low-cost sensor location. If the latter assumption doesn't hold, then we can't use low-cost sensors at all. Chu et al. (2020) suggest one could model the former, spatial-non stationarity. Presumably the temporal non-stationarity could be addressed in a similar way. They were investigating PM2.5 in a high humidity climate, so potentially applicable to our work in Kampala. Low cost OPC sensors are affected by high humidity, so the variation in humidity over Taiwan lead to changing biases. We could use these relevant features (such as humidity) in the calibration function. Unfortunately the type of pollution (road dust, diesel exhaust or charcoal burning) isn't so easily measured, so some sort of spatio-temporal component might need to be introduced to the calibration function.

In the worst case we could imagine that one part of the city has pollution that is underestimated by all the network's low-cost sensors (e.g. a source such as burning tyres), thus all calibration that occurs between low-cost sensors in that area will fail to detect this bias. The only solution is to colocate a reference instruments in those locations. Carefully distributing the small number of reference instruments to sample a broad range of land-use types in the city is therefore prudent.

3.3 | Choice of model and choice of inference

We didn't explore in this paper the potential capabilities of the joint model (in which the pollution is modelled explicitly in time and space), instead we focused on the 'pair model' (in which the pollution remains a hidden, latent random variable) as we found it (in unreported work) more reliable than the joint model which had high correlations in the posterior mediated via both the prior and each co-location (via the likelihood function). Conversely, the pair model scaled and behaved in a far more robust manner, with reliable optimisation (when using VI).

We have (in unreported work) experimented with using MCMC to approximate the variational distribution (Hensman et al., 2015) but found it failed to achieve timely mixing (probably due to the correlations in the posterior) when the network was more than two or three edges deep, so we concentrated on the VI approach.

The variational calibration pair model was quite successful in modelling the posterior variance. One reason might be that the approximating distribution included a full covariance matrix, a necessity due to the strong correlations between sensor parameters in the posterior. By avoiding the mean field approximation we might also have avoided some of the extreme underestimation of the variance (see Figure 1 in Blei et al., 2017, for an explanation). We did (unreported) experiment with using the mean-field approximation in the joint model and found the variance often 'collapsed', giving highly confident predictions, supporting this explanation.

One issue that might impact inference in larger networks is how the number of inducing points would roughly scale with time(T) × sensors(S) × parameters(C), as we assume the number of inducing points is proportional to time. This would be even worse for the categorial example in which the number of parameters scales by the number of species-squared (A^2). This would pose both a computational challenge, with $O((CST)^3)$ time complexity, but also leads to a huge number of parameters to express the variational distribution's covariance matrix, $\propto (CST)^2$. For the categorical model, we could alter our model to add an assumption of independence between species, making the inverse more tractable and reducing the number of parameters from $(A^2ST)^2/2$ to $A(AST)^2/2$.

Using fixed hyperparameters for our model was driven partly by domain knowledge that past stability in calibration does not guarantee future stability, but we also found them somewhat difficult to estimate using black box variational inference, an experience shared by others, e.g. Nguyen and Bonilla (2014) report that they 'have found this to be problematic, ineffectual, and time-consuming.'

3.4 | Hetroscedastic likelihood

In the likelihood used, the variance of the normal distribution was a fixed value. But some sensors (and locations) may have differing amounts of noise, and just as the calibration is expected to vary, so too will this likelihood variance. In the supplementary we explain how to include hetroscedastic noise (see also, for example Lázaro-Gredilla and Titsias, 2011). We also propose that for the variational approximation of noise, one could use just a Dirac distribution (rather than a multivariate Gaussian) and so just specify the mean. The motivation being that estimating the uncertainty in the noise is unrealistic. Our implementation allows either a fixed likelihood variance, a point estimate 'Dirac' approximation, or a full multivariate Gaussian distribution, but in the examples in the paper we used the fixed version, as it seemed to perform sufficiently for our datasets.

3.5 | Summary and future work

In summary, the calibration of a network through co-located observations of pairs of (potentially drifting) sensors can be performed by modelling each sensor's calibration function parameters using one or more Gaussian processes, and providing a likelihood function computed for two observations. This provides a principled and robust Bayesian approach to handling this type of data. This will enable those designing sensor deployments to potentially use pairwise calibration as part of the design solution, in which a reference sensor does not need to visit every low-cost sensor directly.

To put this paper in context, Cui et al. (2021) describe a four stage calibration process for an air pollution monitoring system: (1 and 2) lab based, (3) field based reference sensor colocation (4) field based with occasional visits of mobile reference sensor. The calibration pair method would support the last of these four stages, allowing complex network structures, involving low-cost mobile sensors.

Future work includes optimising the choice of co-locations to minimise error: specifically choosing the best order to visit the static sensors in Kampala. There might be some sensors that are more useful to calibrate (due to their importance in the spatiotemporal model or location in the network). Cost is also a constraint: In our case, mobile-unit visits to the sensors had different (financial) costs, depending on their distance from the driver's base (known locally as their 'stage'). In particular the reference sensors were quite distant.

The categorial bee-labelled problem has its own sampling-optimisation exploitation/exploration questions: deciding which videos to show which citizen-scientist/expert. We might wish to show a video to reduce the uncertainty in their calibration or to show them bees that the model believes they can help classify or maybe help improve their capabilities through training.

We do not consider uncertainty in the multi-hop model. We anticipate in future work looking at performing probabilistic inference over a simplified graph. We think this could be a good compromise (averaging over longer periods and making a relatively simplified graph network), potentially leading to faster inference than the variational Gaussian process approach, while still leading to good estimates of calibration and uncertainty, albeit in a more constrained modelling space.

The main limitation to future work in developing approaches towards networks of calibration the area of air

pollution is that of data. Many low-cost network sensors exist (e.g. Parmar et al., 2017; Rai et al., 2017; Nyarku et al., 2018; Feinberg et al., 2019; Zhao et al., 2021) but few perform regular co-location calibrations. Possibly this is because there are no well developed, easily used, computational tools to perform the in-field calibration discussed in this paper. Hopefully methods, such as the one outlined in this paper, will help solve this chicken-and-egg problem.

The example networks used in this paper (both synthetic and real) were not very deep. We could envisage that, with larger datasets (for example those collected on `www.zooniverse.org`) this network could become more complex, both potentially benefiting more from this method, but also providing a more challenging inference problem as the number of GPs grows. One exciting possibility is to apply the method to the `beewalk.org.uk` project, a large, citizen-science initiative in which volunteers walk transects every month and count the number of each bumblebee species. If beewalkers were to occasionally walk other walkers' routes, could the data quality be improved through pair-wise calibration?

3.6 | Conclusion

The problem of calibrating low-cost, large-scale, air pollution sensor networks is quickly becoming acute: A number of large scale deployments of low-cost air pollution networks have happened recently, including one running on 260 cars in Beijing (Zhao et al., 2021). The difficulty in field-calibration and bounding the accuracy of estimates from these deployments suggests that methods that can provide automatic low-cost recalibration across the network are becoming increasingly necessary.

The calibration pair method described enables sensor deployments with complex networks of colocations. It allows the uncertainty in the predictions from these networks to be quantified, providing guaranteed confidence bounds and robust insights that mean the results from such networks can start being used by policy makers.

conflict of interest

The authors declare that they have no conflict of interest, and that there is no financial interest to report.

code and data

The variational and multi-hop calibration methods are available as a python module which can be downloaded and installed from `https://github.com/lionfish0/calibration`, which includes a demonstration jupyter notebook, and the notebooks used to make the figures etc in this paper.

Kampala sensors: This data is owned and controlled by AirQo and is not available for public release. In particular, the raw boda boda data can reveal locations of individual journeys and the home locations of the taxi drivers.

Bee videos and labels: we have uploaded these videos and labels to the University of Sheffield's data repository (ORDA): `https://doi.org/10.15131/shef.data.19704538`.

references

Alvarado, P. A. and Stowell, D. (2017) Efficient learning of harmonic priors for pitch detection in polyphonic music. *arXiv preprint arXiv:1705.07104*.

Arfire, A., Marjovi, A. and Martinoli, A. (2015) Model-based rendezvous calibration of mobile sensor networks for monitoring air quality. In *2015 IEEE SENSORS*, 1–4. IEEE.

Badura, M., Batog, P., Drzeniecka-Osiadacz, A. and Modzel, P. (2019) Regression methods in the calibration of low-cost sensors for ambient particulate matter measurements. *SN Applied Sciences*, **1**, 1–11.

Barcelo-Ordinas, J. M., Garcia-Vidal, J., Doudou, M., Rodrigo-Muñoz, S. and Cerezo-Llavero, A. (2018) Calibrating low-cost air quality sensors using multiple arrays of sensors. In *2018 IEEE Wireless Communications and Networking Conference (WCNC)*, 1–6. IEEE.

Blei, D. M., Kucukelbir, A. and McAuliffe, J. D. (2017) Variational inference: A review for statisticians. *Journal of the American statistical Association*, **112**, 859–877.

Bychkovskiy, V., Megerian, S., Estrin, D. and Potkonjak, M. (2003) A collaborative approach to in-place sensor calibration. In *Information processing in sensor networks*, 301–316. Springer.

Castell, N., Dauge, F. R., Schneider, P., Vogt, M., Lerner, U., Fishbain, B., Broday, D. and Bartonova, A. (2017) Can commercial low-cost sensor platforms contribute to air quality monitoring and exposure estimates? *Environment international*, **99**, 293–302.

Chu, H.-J., Ali, M. Z. and He, Y.-C. (2020) Spatial calibration and PM2.5 mapping of low-cost air quality sensors. *Scientific reports*, **10**, 1–11.

Crilley, L. R., Shaw, M., Pound, R., Kramer, L. J., Price, R., Young, S., Lewis, A. C. and Pope, F. D. (2018) Evaluation of a low-cost optical particle counter (alphasense OPC-N2) for ambient air monitoring. *Atmospheric Measurement Techniques*, **11**, 709–720.

Csiba, D. and Richtárik, P. (2018) Importance sampling for minibatches. *The Journal of Machine Learning Research*, **19**, 962–982.

Cui, H., Zhang, L., Li, W., Yuan, Z., Wu, M., Wang, C., Ma, J. and Li, Y. (2021) A new calibration system for low-cost sensor network in air pollution monitoring. *Atmospheric Pollution Research*, **12**, 101049.

Datta, A., Saha, A., Zamora, M. L., Buehler, C., Hao, L., Xiong, F., Gentner, D. R. and Koehler, K. (2020) Statistical field calibration of a low-cost PM2.5 monitoring network in baltimore. *Atmospheric Environment*, **242**, 117761.

Delaine, F., Lebental, B. and Rivano, H. (2019) In situ calibration algorithms for environmental sensor networks: A review. *IEEE Sensors Journal*, **19**, 5968–5978.

Dyo, V., Ellwood, S. A., Macdonald, D. W., Markham, A., Mascolo, C., Pásztor, B., Scellato, S., Trigoni, N., Wohlers, R. and Yousef, K. (2010) Evolution and sustainability of a wildlife monitoring sensor network. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, 127–140.

Feinberg, S. N., Williams, R., Hagler, G., Low, J., Smith, L., Brown, R., Garver, D., Davis, M., Morton, M., Schaefer, J. et al. (2019) Examining spatiotemporal variability of urban particulate matter and application of high-time resolution data from a network of low-cost air pollution sensors. *Atmospheric environment*, **213**, 579–584.

Ferrer-Cid, P., Barcelo-Ordinas, J. M., Garcia-Vidal, J., Ripoll, A. and Viana, M. (2020) Multisensor data fusion calibration in IoT air pollution platforms. *IEEE Internet of Things Journal*, **7**, 3124–3132.

Fonollosa, J., Fernandez, L., Gutiérrez-Gálvez, A., Huerta, R. and Marco, S. (2016) Calibration transfer and drift counteraction in chemical sensor arrays using direct standardization. *Sensors and Actuators B: Chemical*, **236**, 1044–1053.

van de Giesen, N., Hut, R. and Selker, J. (2014) The trans-African hydro-meteorological observatory (TAHMO). *Wiley Interdisciplinary Reviews: Water*, **1**, 341–348.

GPyOpt authors (2016) GPyOpt: A Bayesian optimization framework in python. `http://github.com/SheffieldML/GPyOpt`.

Hasenfratz, D., Saukh, O. and Thiele, L. (2012) On-the-fly calibration of low-cost gas sensors. In *European Conference on Wireless Sensor Networks*, 228–244. Springer.

Hensman, J., Fusi, N. and Lawrence, N. D. (2013) Gaussian processes for big data. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence*, 282–290.

Hensman, J., Matthews, A. G., Filippone, M. and Ghahramani, Z. (2015) MCMC for variationally sparse Gaussian processes. In *Advances in Neural Information Processing Systems*, 1648–1656.

Hug, N. (2020) Surprise: A python library for recommender systems. *Journal of Open Source Software*, **5**, 2174. URL: `https://doi.org/10.21105/joss.02174`.

Janssen, S., Guerreiro, C., Viaene, P., Georgieva, E. and Thunis, P. (2017) Guidance document on modelling quality objectives and benchmarking. `https://fairmode.jrc.ec.europa.eu/document/fairmode/WG1/Guidance_MQO_Bench_vs2.1.pdf`.

Khedo, K. K., Perseedoss, R., Mungur, A. et al. (2010) A wireless sensor network air pollution monitoring system. *International Journal of Wireless & Mobile Networks*, **2**.

Kizel, F., Etzion, Y., Shafran-Nathan, R., Levy, I., Fishbain, B., Bartonova, A. and Broday, D. M. (2018) Node-to-node field calibration of wireless distributed air pollution sensor network. *Environmental Pollution*, **233**, 900–909.

Kosmala, M., Wiggins, A., Swanson, A. and Simmons, B. (2016) Assessing data quality in citizen science. *Frontiers in Ecology and the Environment*, **14**, 551–560.

Lázaro-Gredilla, M. and Titsias, M. K. (2011) Variational heteroscedastic Gaussian process regression. In *ICML*.

Lee, H., Kang, J., Kim, S., Im, Y., Yoo, S. and Lee, D. (2020) Long-term evaluation and calibration of low-cost particulate matter (PM) sensor. *Sensors*, **20**, 3617.

Lewis, A. C., Lee, J. D., Edwards, P. M., Shaw, M. D., Evans, M. J., Moller, S. J., Smith, K. R., Buckley, J. W., Ellis, M., Gillot, S. R. et al. (2016) Evaluating the performance of low cost chemical sensors for air pollution research. *Faraday discussions*, **189**, 85–103.

Liu, D., Zhang, Q., Jiang, J. and Chen, D.-R. (2017) Performance calibration of low-cost and portable particular matter (PM) sensors. *Journal of Aerosol Science*, **112**, 1–10.

Liu, X., Jayaratne, R., Thai, P., Kuhn, T., Zing, I., Christensen, B., Lamont, R., Dunbabin, M., Zhu, S., Gao, J. et al. (2020) Low-cost sensors as an alternative for long-term air quality monitoring. *Environmental research*, **185**, 109438.

Maag, B., Zhou, Z., Saukh, O. and Thiele, L. (2017) Scan: Multi-hop calibration for mobile sensor arrays. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, **1**, 1–21.

Mao, F., Khamis, K., Krause, S., Clark, J. and Hannah, D. M. (2019) Low-cost environmental sensor networks: recent advances and future directions. *Frontiers in Earth Science*, **7**, 221.

Marathe, S., Nambi, A., Swaminathan, M. and Sutaria, R. (2021) Currentsense: A novel approach for fault and drift detection in environmental IoT sensors. In *Proceedings of the International Conference on Internet-of-Things Design and Implementation*, 93–105.

Markert, J.-F., Budde, M., Schindler, G., Klug, M. and Beigl, M. (2018) Privacy-preserving collaborative blind macro-calibration of environmental sensors in participatory sensing. *EAI Endorsed Trans. Internet Things*, **18**, 1–8.

Miskell, G., Salmond, J., Alavi-Shoshtari, M., Bart, M., Ainslie, B., Grange, S., McKendry, I. G., Henshaw, G. S. and Williams, D. E. (2016) Data verification tools for minimizing management costs of dense air-quality monitoring networks. *Environmental science & technology*, **50**, 835–846.

Miskell, G., Salmond, J. A. and Williams, D. E. (2018) Solution to the problem of calibration of low-cost air quality measurement sensors in networks. *ACS sensors*, **3**, 832–843.

Nguyen, T. V. and Bonilla, E. V. (2014) Automated variational inference for Gaussian process models. *Advances in Neural Information Processing Systems*, **27**, 1404–1412.

Nyarku, M., Mazaheri, M., Jayaratne, R., Dunbabin, M., Rahman, M. M., Uhde, E. and Morawska, L. (2018) Mobile phones as monitors of personal exposure to air pollution: Is this the future? *PLoS One*, **13**, e0193150.

Ollett, J. (2021) Bumblebee classification with convolutional neural networks. *Undergraduate Dissertation. University of Sheffield*.

Parmar, G., Lakhani, S. and Chattopadhyay, M. K. (2017) An IoT based low cost air pollution monitoring system. In *2017 International Conference on Recent Innovations in Signal processing and Embedded Systems (RISE)*, 524–528. IEEE.

Peng, Y., Qiao, W., Qu, L. and Wang, J. (2017) Sensor fault detection and isolation for a wireless sensor network-based remote wind turbine condition monitoring system. *IEEE Transactions on Industry Applications*, **54**, 1072–1079.

Perälä, T., Vanhatalo, J. and Chrysafi, A. (2020) Calibrating expert assessments using hierarchical Gaussian process models. *Bayesian analysis*, **15**, 1251–1280.

Pohl, S., Garvelmann, J., Wawerla, J. and Weiler, M. (2014) Potential of a low-cost sensor network to understand the spatial and temporal dynamics of a mountain snow cover. *Water Resources Research*, **50**, 2533–2550.

Rai, A. C., Kumar, P., Pilla, F., Skouloudis, A. N., Di Sabatino, S., Ratti, C., Yasar, A. and Rickerby, D. (2017) End-user perspective of low-cost sensors for outdoor air pollution monitoring. *Science of The Total Environment*, **607**, 691–705.

Salimbeni, H. and Deisenroth, M. P. (2017) Doubly stochastic variational inference for deep Gaussian processes. In *NIPS*.

Sandradewi, J., Prévôt, A. S., Szidat, S., Perron, N., Alfarra, M. R., Lanz, V. A., Weingartner, E. and Baltensperger, U. (2008) Using aerosol light absorption measurements for the quantitative determination of wood burning and traffic emission contributions to particulate matter. *Environmental science & technology*, **42**, 3316–3323.

Saukh, O., Hasenfratz, D. and Thiele, L. (2015) Reducing multi-hop calibration errors in large-scale mobile sensor networks. In *Proceedings of the 14th International Conference on Information Processing in Sensor Networks*, 274–285.

Shaddick, G., Thomas, M. L., Green, A., Brauer, M., van Donkelaar, A., Burnett, R., Chang, H. H., Cohen, A., Dingenen, R. V., Dora, C. et al. (2018) Data integration model for air quality: a hierarchical approach to the global estimation of exposures to ambient air pollution. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, **67**, 231–253.

Sousan, S., Koehler, K., Hallett, L. and Peters, T. M. (2016) Evaluation of the alphasense optical particle counter (OPC-N2) and the grimm portable aerosol spectrometer (pas-1.108). *Aerosol Science and Technology*, **50**, 1352–1365.

Stedman, J., Abbott, J., Willis, P. and Bower, J. (2006) Review of background air quality data and methods to combine these with process contributions. *Environment Agency for England and Wales: Bristol*.

Talampas, M. C. R. and Low, K.-S. (2012) Maximum likelihood estimation of ground truth for air quality monitoring using vehicular sensor networks. In *TENCON 2012 IEEE Region 10 Conference*, 1–6. IEEE.

Tsujita, W., Ishida, H. and Moriizumi, T. (2004) Dynamic gas sensor network for air pollution monitoring and its auto-calibration. In *SENSORS, 2004 IEEE*, 56–59. IEEE.

Wang, Y., Du, Y., Wang, J. and Li, T. (2019) Calibration of a low-cost PM2.5 monitor using a random forest model. *Environment international*, **133**, 105161.

Wilson, A. G., Knowles, D. A. and Ghahramani, Z. (2012) Gaussian process regression networks. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, 1139–1146.

Xiang, Y., Bai, L. S., Pledrahita, R., Dick, R. P., Lv, Q., Hannigan, M. and Shang, L. (2012) Collaborative calibration and sensor placement for mobile sensor networks. In *2012 ACM/IEEE 11th International Conference on Information Processing in Sensor Networks (IPSN)*, 73–83. IEEE.

Zhao, B., Yu, L., Wang, C., Shuai, C., Zhu, J., Qu, S., Taiebat, M. and Xu, M. (2021) Urban air pollution mapping using fleet vehicles as mobile monitors and machine learning. *Environmental Science & Technology*, **55**, 5579–5588.

Zimmerman, N., Presto, A. A., Kumar, S. P., Gu, J., Hauryliuk, A., Robinson, E. S., Robinson, A. L. and Subramanian, R. (2018) A machine learning calibration model using random forests to improve sensor performance for lower-cost air quality monitoring. *Atmospheric Measurement Techniques*, **11**, 291–313.

Supplementary Material

Algorithm 1 Variational Inference for calibration.

Note: For implementation we structure an input matrix X to hold the time, sensor and component as three columns. This matrix is $2C$ times the length of Y. Split into C submatrices, each pair of rows in each submatrix is associated with one row of Y. f now becomes a vector with each item associated with one row of X. The parameters are selected using slice notation.

Inputs

Observation time, sensor and parameter, $X = \{x_i\}_{i=1}^{2CN}$;

Observation pair values, $Y = \{[y_i^{(1)}, y_i^{(2)}]\}_{i=1}^{N}$;

Inducing point locations (time, sensor and parameter), $Z = \{z_i\}_{i=1}^{M}$;

Calibration function, $\phi(y, c)$;

Number of parameters used by function, C;

Kernel, $k(\cdot, \cdot)$;

Likelihood variance σ^2;

Reference sensors, $r = \{0, 1\}^S$;

Number of samples in MC approximation, P.

Outputs

Approximating Gaussian distribution parameters: mean m and factor R (where distribution covariance is $RR^\top$).

1: **procedure** VI($X, Y, Z, \phi(\cdot), C, k(\cdot, \cdot), \sigma^2, r$)
2:     **while** m or R not converged **do**
3:   ▷ $q(f)$ is the approximate posterior over all latent variables defined in X.
4:         $q(f) \sim N(K_{xz}K_{zz}^{-1}m, K_{xx} - K_{xz}K_{zz}^{-1}K_{zx} + K_{xz}K_{zz}^{-1}RR^\top K_{zz}^{-1}K_{zx})$
5:         **for** $j = 1..P$ **do**   ▷ Sample P times
6:             **for** $i = 1..N$ **do**
7:   ▷ Sample the latent variables relevant to the two sensors.
8:                 $s_i^{(1)}, s_i^{(2)} = \text{sample}[q(f_{2i-1::2N}), q(f_{2i::2N})]$ ▷ $s_i^{(1)}$ and $s_i^{(2)}$ are each of length C.
9:                 $L_{ij} = \log p\left(\begin{bmatrix} y_i^{(1)} \\ y_i^{(2)} \end{bmatrix} \middle| \begin{bmatrix} s_i^{(1)} \\ s_i^{(2)} \end{bmatrix}\right)$ ▷ Compute likelihood of sample, using (4)
10:            **end for**
11:        **end for**
12:        $\mathcal{L} \leftarrow \frac{1}{P}\sum_{j=1}^{P}\sum_{i=1}^{N}(L_{ij}) - D_{KL}\left(N(m, RR^\top), N(0, K_{ZZ})\right)$ ▷ Compute ELBO
13:        compute $\frac{d\mathcal{L}}{m}$ and $\frac{d\mathcal{L}}{R}$ ▷ using automatic differentiation.
14:        update m and R using stochastic gradient descent (using Adam).
15:    **end while**
16: **end procedure**

Algorithm 2 Prediction for Algorithm 1

    Inputs

Test time, sensor and component, $\boldsymbol{X}_* = \{\boldsymbol{x}_{*i}\}_{i=1}^{N_*}$; $\boldsymbol{y}_* = \{y_{*i}\}_{i=1}^{N_*}$ raw observations (unlike normal regression we need to give uncalibrated observations at test time)

Inducing point locations (time, sensor and component), $\boldsymbol{Z} = \{\boldsymbol{z}_i\}_{i=1}^{M}$;

Calibration function, $\phi(y, \boldsymbol{c})$;

Number of components used by function, C;

Kernel, $k(\cdot, \cdot)$;

Approximating Gaussian distribution parameters: $\boldsymbol{m}$ and $\boldsymbol{R}$.

Number of samples for each test point, P.

    Outputs

An $N_* \times P$ matrix of P samples for each of the N_* test points, $\boldsymbol{S}$

1: **procedure** Predict($\boldsymbol{X}_*, \boldsymbol{Y}_*, \boldsymbol{Z}, \phi(\cdot), C, k(\cdot, \cdot)$)
2:                                     ▷ $q(\boldsymbol{f})$ is the approximate posterior over all latent variables defined in $\boldsymbol{X}_*$.
3:     $q(\boldsymbol{f}) \sim N(K_{x_*z}K_{zz}^{-1}\boldsymbol{m}, K_{x_*x_*} - K_{xz}K_{zz}^{-1}K_{zx_*} + K_{x_*z}K_{zz}^{-1}RR^{\top}K_{zz}^{-1}K_{zx_*})$
4:     **for** $i = 1..N_*$ **do**
5:                                      ▷ Sample the latent variables relevant to each test point.
6:         $\boldsymbol{s}_i = \phi(y_i, \text{sample}[q(\boldsymbol{f}_{i::N_*})])$ ▷ Sample K times.
7:     **end for**
8: **end procedure**

Algorithm 3 Multi-hop 'Graph' algorithm

   Inputs

Observation time, sensor id pair, $\boldsymbol{X} = \{\boldsymbol{x}_i\}_{i=1}^N$, so $\boldsymbol{X}$ is $(N \times 3)$;

Observation pair values, $\boldsymbol{Y} = \{[y_{i1}, y_{i2}]\}_{i=1}^N$, so $\boldsymbol{Y}$ is $(N \times 2)$;

Window size, δ;

Edge 'distances', connect by colocation events and over time, $d_{\text{colocation}}, d_{\text{time}}$;

Reference sensors, $\boldsymbol{r} = \{0, 1\}^{|A|}$.

   Outputs

Dictionary of scaling factors for (sensor,window) tuples $\boldsymbol{F}$.

1: **procedure** BuildGraph($\boldsymbol{X}, \boldsymbol{Y}, \delta, \boldsymbol{r}$)

2:     **for** w in all time windows W of width δ **do**

3:        **for** s_i, s_j in all pairs of sensors from X, where $s_i \neq s_j$ **do**

4:           $Y' \leftarrow Y_{X_{:,2}=s_i \wedge X_{:,3}=s_j \wedge X_{:,1} \in w}$ ▷ Selects observation pairs that are in the time window and between sensors s_i and s_j

5:           **if** $|Y'| \geq 5$ **then**   ▷ We don't add edges where there are fewer than five observations

6:              G.addEdge($(s_i, w) \rightarrow (s_j, w)$, value=mean($\log(Y'_{:,1}) - log(Y'_{:,2})$, distance=$d_{colocation}$))

7:           **end if**

8:        **end for**

9:     **end for**

10:    **for** w in all time windows $W - 1$ of width δ **do**

11:       **for** s_i in all sensors from X **do**

12:          G.addEdge($(s_i, w) \rightarrow (s_i, w + 1)$, value=0, distance=$d_{time}$)

13:          G.addEdge($(s_i, w + 1) \rightarrow (s_i, w)$, value=0, distance=$d_{time}$)

14:       **end for**

15:    **end for**

16:    **for** w in all time windows $W - 1$ of width δ **do**

17:       **for** s_i in all sensors from X **do**

18:          $P \leftarrow$ Shortest path (using Dijkstra) from (s_i, w) to any reference sensor, specified in r.

19:          **if** Path P exists **then**

20:            $F[s_i, w] = \sum_{p \in P} p_{value}$ ▷ Sum log ratios over path

21:          **end if**

22:       **end for**

23:    **end for**

24: **end procedure**

Algorithm 4 Multi-hop 'Graph' prediction algorithm

Inputs

Observation time, sensor id, $\boldsymbol{X}_* = \{\boldsymbol{x}_i\}_{i=1}^N$, so $\boldsymbol{X}_*$ is ($N_* \times 2$);

Observed raw values, $\boldsymbol{y}_{\text{raw}}$, so $\boldsymbol{y}_{\text{raw}}$ is an ($N_* \times 1$) vector;

Window size, δ;

Dictionary of scaling factors for (sensor,window) tuples $\boldsymbol{F}$, from Algorithm 3.

Outputs

Predicted calibrated values, $\boldsymbol{y}_*$, so $\boldsymbol{y}_*$ is an ($N_* \times 1$) vector.

1: **procedure** Predict($\boldsymbol{X}_*, \boldsymbol{y}_{\text{raw}}, \delta, \boldsymbol{r}, G$)
2:     **for** $\boldsymbol{x}_*$ in $\boldsymbol{X}_*$; y_* in $\boldsymbol{y}_*$; and y_{raw} in $\boldsymbol{y}_{\text{raw}}$ **do**
3:         $w \leftarrow$ window computed for $[\boldsymbol{x}_*]_1$ using window size δ
4:         **if** ($[\boldsymbol{x}_*]_2, w) \notin F$ **then**
5:             Raise exception: No path to a reference sensor.
6:         **end if**
7:         $y_* \leftarrow e^{F[[\boldsymbol{x}_*]_2, w]} \times y_{\text{raw}}$
8:     **end for**
9: **end procedure**