MapTransfer: Urban Air Quality Map Generation for Downscaled Sensor Deployments

Yun Cheng
ETH Zurich
Zurich, Switzerland
chengyu@ethz.ch

Xiaoxi He
ETH Zurich
Zurich, Switzerland
hex@ethz.ch

Zimu Zhou
Singapore Management University
Singapore, Singapore
zimuzhou@smu.edu.sg

Lothar Thiele
ETH Zurich
Zurich, Switzerland
thiele@ethz.ch

***Abstract*—Dense deployments of commodity air quality sensors have proven effective to provide spatially-resolved information on urban air pollution in real-time. However, long-term operation of a dense sensor deployment incurs enormous maintenance expenses and efforts. A cost-effective alternative is to first collect measurements with an initial dense deployment and then rely on a small subset of sensors for air quality map generation. To avoid dramatic accuracy degradation in air quality maps generated using the downscaled sparse deployment, we design MapTransfer, an air quality map generation scheme which augments the current sensor measurements from the downscaled sparse deployment with appropriate historical data from the initial dense deployment. Due to the spatiotemporal complexity of air pollution, it is challenging to select the best historical data and fuse them with measurements from the downscaled deployment to accurate map generation. To overcome this challenge, MapTransfer adopts a learning-based data selection scheme and integrates the best historical data with the current measurements via a multi-output Gaussian process model at sub-region levels. Evaluations on a large-scale $PM_{2.5}$ sensor deployment show that MapTransfer reduces the overall mean absolute error of air quality maps by 45.9%, compared with using data from the downscaled deployment alone.**

I. Introduction

Advances in air quality sensor technologies have enabled urban-scale sensor deployments for fine-grained air pollution monitoring [1]–[4]. With densely deployed sensors, real-time, spatially-resolved air quality maps can be generated by spatial interpolation models like Gaussian processes [5], even without training complex models and integrating heterogeneous data sources [1], [6]–[8]. The availability of such urban air quality maps not only raises public awareness of air pollution but also empowers authorities to craft and evaluate policies. For instance, the concentration of particulate matter (PM) with diameters less than 2.5 micron ($PM_{2.5}$), an air pollutant that may cause respiratory diseases [9], is constantly monitored in many major cities in China via large-scale static [1] or mobile sensor networks [2], [4]. The hourly updated $PM_{2.5}$ city maps generated via these sensor measurements facilitate citizens to adjust travel plans and authorities to make policy and control emissions [10], [11].

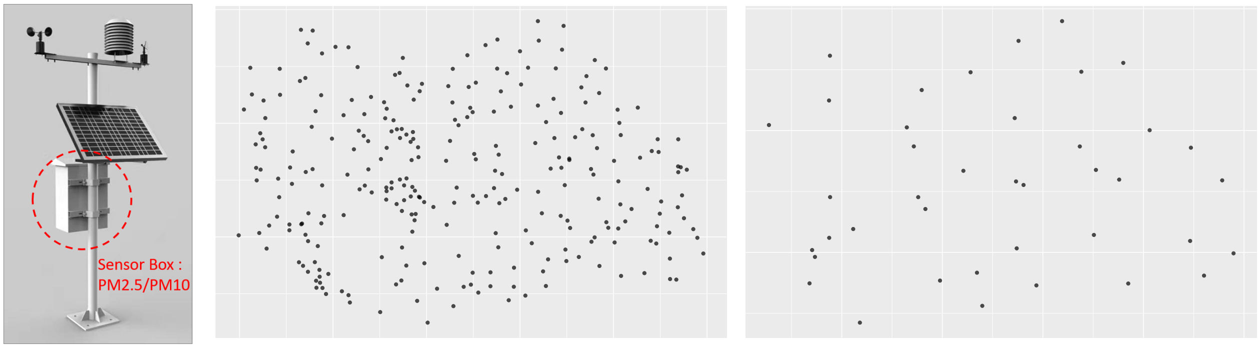
Although many dense air quality sensor deployments have been reported from both the academia and industries [1]–[4], much fewer remain operating after certain period of time. A major reason for the short life-time of dense deployments is the tedious efforts and high costs for sensor maintenance. For instance, low-cost air quality sensors have to be periodically re-calibrated [12], [13], and many may break after 3 months [14]. In practice, many companies have to *downscale* their deployments (*i.e.*, only keep a sensor subset of the initial deployment) for long-term air quality monitoring due to budget concerns [15]. Particularly, a downscaled deployment may only contain small portion (*e.g.*, 1/3 or 1/4) of sensors in the original dense deployment. Due to the dynamics and complexity of urban air pollution, fine-grained air quality map generated with such a sparse sensor deployment is likely to suffer significant accuracy drop. According to our experiments with an urban $PM_{2.5}$ monitoring deployment, the average mean absolute error (MAE) of air quality maps generated using measurements from a dense deployment of 200 sensors would dramatically increase from 5.1 to 21.8 if measurement of merely 50 sensors are used (see Sec. IV-A), whereas an MAE below 10 is considered accurate for applications such as urban $PM_{2.5}$ maps [1], [14], [16].

To improve the accuracy of air quality maps generated with measurements from a sparse downscaled deployment, we design MapTransfer, a novel air quality map generation scheme which augments the current measurements with historical data collected from the initial dense deployment. The underlying rationale is intuitive: *the downscaled deployment monitors the same region as the initial dense deployment, and hence it is probable that the current air quality distribution over the whole region is the same or similar to that at some time point in history.* Hence it may improve the accuracy of the air quality map generated with the sparse deployment by *properly transferring and augmenting knowledge* of air quality in this region from the historical data in the dense deployment.

However, it faces multiple challenges to transform the idea above into a working system.

1) *How to integrate current measurements from the downscaled deployment and historical data from the initial dense deployment for map generation? Air quality maps generated from the dense deployment typically rely on Gaussian processes [5]–[7]. To seamlessly integrate knowledge learned from the dense deployment, we adopt*

Zimu Zhou is corresponding author



(a) Sensor

(b) Dense Deployment

(c) Sparse Deployment

Fig. 1. Deployment of air quality sensors in a 50 $km \times 30$ km region: (a) illustration of an air quality sensor; (b) dense initial deployment with 200 sensors; (c) sparse downscaled deployment with 50 sensors.

Multi-Output Gaussian Process (MOGP) [17], an extension of Gaussian processes, for map generation.

2) *How to select the best historical data to improve the accuracy of air quality map generation?* Due to the complexity and dynamics of urban air pollutant concentrations, it is non-trivial to select the best historical measurements. As we show in real-world $PM_{2.5}$ measurements, conventional criteria such as such as Root Mean Square Error (RMSE) or Correlation Coefficient (CORR) often lead to negative transfer [18] and even an accuracy drop in the generated air quality map (see Sec. V-A). MapTransfer exploits a learning-based instance selection (LIS). It extracts a rich set of features from both $PM_{2.5}$ measurements and auxiliary meta data sources *e.g.*, meteorological information, and applies a neural network to select the best instances for air quality map generation.

3) *At what scale should data augmentation be performed?* We observe that the correlation between measurements in the initial and the downscaled deployments is not homogeneous over the entire monitoring region (see Sec. VI-A). Such spatial locality in correlation indicates that knowledge transfer and data augmentation at the scale of the entire region may impair the performance of map generation. To exploit spatial locality during data augmentation, MapTransfer utilizes sub-region selection (SRS) to split the whole region into sub-regions and search for the best instance transfer for each sub-region.

The contributions of this paper are summarized as follows.

- We propose MapTransfer, the first practical urban air quality map generation scheme for downscaled sensor deployments by transferring knowledge and augmenting historical data from the initial dense deployment.
- We comprehensively evaluate the performance of MapTransfer with measurements collected from a large-scale $PM_{2.5}$ monitoring system consisting of 260 sensors over one and a half years. Experimental results show that MapTransfer is able to reduce the overall MAE of $PM_{2.5}$ maps generated with a downscaled deployment by 45.9% (from 21.8 to 11.8), achieving an accuracy suited to raise public awareness and take measures for emission control [10], [11], as well as data mining applications [16], [19].

In the rest of this paper, we formally define our problem and introduce our datasets in Sec. II. Then we present an overview of our MapTransfer method in Sec. III and explain its core modules in Sec. IV, Sec. V and Sec. VI. The evaluations are shown in Sec. VII and we discuss the limitations of our method in Sec. VIII. We review related work in Sec. IX, and finally conclude in Sec. X.

II. Preliminaries

In this section, we formally define the problem of air quality map generation from downscaled sensor deployments, and then introduce the datasets collected from a large-scale $PM_{2.5}$ sensor deployment that will be used throughout this paper.

A. Problem Definition

We start by defining some basic concepts that will be used throughout this paper.

Definition 1 (deployment): A deployment refers to a sensor network activated during a certain period of time.

In our *first-dense-then-sparse* scenario, a dense sensor network is used for air quality monitoring only during the initial phase. It is then downscaled to a sparse sensor network by activating only a subset of the original sensors.

Definition 2 (dense deployment): A dense deployment is a sensor network used during the initial phase. All sensors are activated in a dense deployment.

Definition 3 (sparse deployment): The sparse deployment is the downscaled sensor network. Only a subset of the sensors in the initial phase are activated in the sparse deployment.

Definition 4 (Air Quality Map Generation for Downscaled Sensor Deployments Problem): Given a dense and a sparse deployment covering the same urban region, the problem is to effectively utilize the historical sensor measurements from the dense deployment to increase the accuracy of the air quality map generated from the sparse deployment.

B. Datasets

We collect measurements from a large-scale $PM_{2.5}$ monitoring deployment consisting of 260 low-cost sensors (see Fig. 1a) in Beijing, China. The sensors upload their readings to a server every minute. We collect $PM_{2.5}$ readings from the 260 sensors over a period of 18 months from January 1st, 2018 to July 1st, 2019. These 260 sensors are randomly divided into two groups with 200 and 60 sensors. The data collected from the 200 sensors are used to generate hourly air quality maps for the 50 $km \times 30\ km$ rectangular area in Fig. 1b and Fig. 1c, with a resolution of 1 $km \times 1\ km$. The other 60 sensors are used for testing the accuracy of the generated maps. Since we aim to generate hourly air quality maps, we down-sample the per minute raw sensor measurements to per hour by averaging all the measurements within each hour.

We simulate the scenario from an initial dense deployment to a downscaled sparse sensor deployment as follows. The 200 sensors during the whole year of 2018 are regarded as the dense deployment. Then during the first half year of 2019, it is downscaled to the sparse deployment with 50 randomly picked sensors from the original 200 sensors. Consequently, the $PM_{2.5}$ measurements collected from these 200 sensors form two datasets:

- *dense dataset:* It contains the hourly $PM_{2.5}$ measurements from all 200 sensors (dense deployment) from January 1st, 2018 to January 1st, 2019.
- *sparse dataset:* It contains the hourly $PM_{2.5}$ measurements from the 50 sensors (sparse deployment) from January 2nd, 2019 to July 1st, 2019.

For ease of presentation, we call one instance (*i.e.*, the hourly averaged $PM_{2.5}$ measurements from a sensor deployment) in the dense (sparse) dataset as a *dense (sparse) instance*.

III. MapTransfer Overview

MapTransfer adopts multi-output Gaussian process (MOGP) to integrate data from both the dense and the sparse deployments for map generation. Furthermore, to boost the accuracy of air quality map generation, we add two novel modules before MOGP: learning-based dense instance selection (LIS) and sub-region selection (SRS), which are illustrated in Fig. 2.

- **Multi-Output Gaussian Process (MOGP)**: MOGP serves as a unified map generation model which takes multiple instances to generate air quality maps. Specifically, a current sparse instance and an appropriate historical dense instance are used as the input of MOGP, whereas the output is the improved air quality map of the sparse instance. Details are explained in Sec. IV. In the workflow of MapTransfer, MOGP is also used to generate the training dataset for the Learning-based Instance Selection module, which is described in Sec. V.
- **Learning-based Instance Selection (LIS)**: The aim of LIS is to avoid negative transfer in dense instance selection. LIS extracts a rich set of features from both the dense and the sparse datasets as well as auxiliary meta data sources such as meteorological data, then it selects the best dense instances using an artificial neural network (ANN). Given a current sparse instance, LIS selects the top-n best dense instances, which will be used together with the current sparse instance in the following Sub-Region Section module. Details are explained in Sec. V.
- **Sub-Region Selection (SRS)**: The aim of SRS is to further improve the accuracy of air quality map generation by exploiting spatial locality. SRS explores different region splitting scheme to divide the whole region into sub-regions, and searches among the top-n dense instances selected by LIS for the one that yields the most accurate air quality map in each sub-region for the current sparse instance. Then these sub-regions of different dense instances are stitched into a fictive dense instance, which is fed into MOGP with the current sparse instance for map generation. Details are explained in Sec. VI.

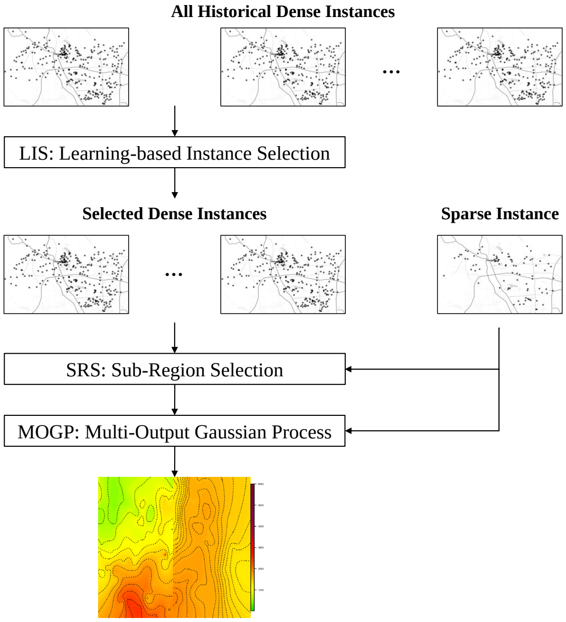
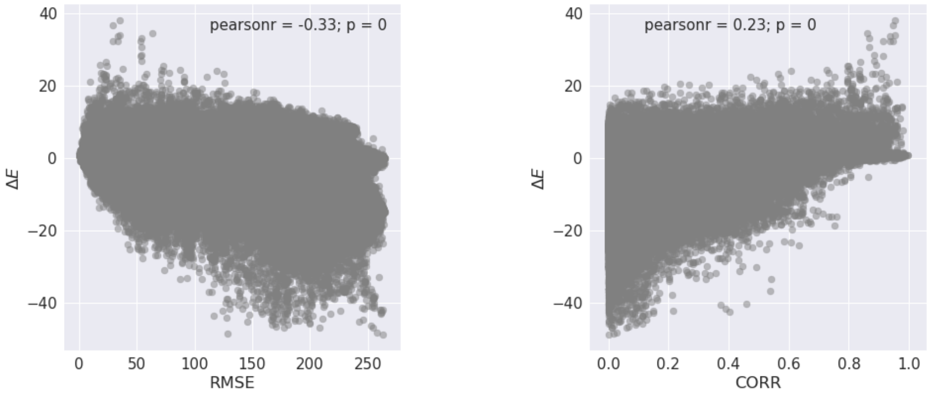

All Historical Dense Instances
...
LIS: Learning-based Instance Selection
Selected Dense Instances
...
Sparse Instance
SRS: Sub-Region Selection
MOGP: Multi-Output Gaussian Process

Fig. 2. Workflow of MapTransfer.

In real-world situation, the constantly changing environment arises challenges for MapTransfer to keep being effective over long-term sparse deployments. Significant changes in the urban environment, like new high-rise buildings causing changes in meteorological dynamics, will severely reduce the transferable knowledge in the local region. Therefore, when the accuracy of the air quality map generated by MapTransfer severely decreases in some local regions, new sensors need to be deployed to explore and learn the changes of the environment. Furthermore, when the changed environment is monitored by the added sensors after some period of time, these sensors can be downscaled again in order to reduce maintenance costs. The data collected during this temporary local dense deployment can be used by MapTransfer to improve the air quality map in the future. The detailed procedures of this mechanism is however out of the scope of this paper.



(a) (b)

Fig. 3. Accuracy improvement in air quality map versus similarity between the sparse and the best dense instance, where the similarity is measured by (a) RMSE and (b) correlation coefficient.

IV. Multi-Output Gaussian Process Model

In this section, we first review how to generate an air quality map from a dense (sparse) instance and then explain how to integrate a dense and a sparse instance for map generation.

A. Map Generation via Gaussian Process

To generate an air quality map, we need a mapping $\mathbf{x} \mapsto f(\mathbf{x})$, where $\mathbf{x} \in \mathbb{R}^2$ is a 2-dimensional geographical coordinate, and $f(\mathbf{x}) \in \mathbb{R}$ is the real-valued air quality index. Gaussian processes (GP) [5] proves effective to model and learn this mapping when measurements from a dense deployment are available [1], [6]. They assume that the function f is distributed as a GP with mean function m and covariance function k [5], which can be written as:

$$f \sim \mathcal{GP}(m, k) \tag{1}$$

Given a measurement instance, the parameters in the mean function m and covariance function k are learned on this instance, then the learned GP distribution is used to calculate the real-valued air quality indices at each grid. Finally we have an air quality map generated by the instance via GP.

The accuracy of the air quality map generated by GP heavily relies on the density of the sensor deployment. As an example, we compare the accuracy of the air quality maps generated by the dense instances and the sparse instances collected from the two deployments in Sec. II-B using GP. Specifically, we use measurements from the two deployments collected during the same period of time (January 1st, 2019 to July 1st, 2019), and assess the accuracy of the generated air quality maps. The map accuracy is assessed by MAEs calculated at the locations of the 60 test sensors. The MAEs averaged over the first half year of 2019 of maps generated using dense and sparse instances are 5.1 and 21.8, respectively. For $PM_{2.5}$ concentration, an MAE below 10 is considered accurate for data mining applications [16]. The example shows that air quality maps generated with merely sparse instances have limited accuracy and augmenting historical dense instances is necessary.

B. Map Generation via Multi-Output Gaussian Process

To augment the current sparse instances with historical dense instances for air quality map generation, we adopt Multi-Output Gaussian Process [17]. It is an extension of GP which jointly considers multiple correlated distributions. Suppose we have one sparse instance and one dense instance. $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$ are the air quality indices over the monitored region at the hour of the sparse and dense instance. MOGP assumes that the distributions of f_1 and f_2 are correlated, and they conform to a multi-output Gaussian process:

$$\begin{bmatrix} f_1 \\ f_2 \end{bmatrix} \sim \mathcal{GP}\left(\begin{bmatrix} m_1 \\ m_2 \end{bmatrix}, \begin{bmatrix} k_{1,1} & k_{1,2} \\ k_{2,1} & k_{2,2} \end{bmatrix}\right) \tag{2}$$

where the multi-output mean functions $\begin{bmatrix} m_1 \\ m_2 \end{bmatrix}$ and multi-output covariance functions $\begin{bmatrix} k_{1,1} & k_{1,2} \\ k_{2,1} & k_{2,2} \end{bmatrix}$ are learned on both the sparse and dense instance. Here we still use $f_1(\mathbf{x})$ to generate the air quality map at the hour of the sparse instance. However, instead of being solely decided by the sparse instance, this air quality map generated via MOGP also takes the dense instance into account.

A crucial assumption for MOGP yielding high-accuracy air quality maps is that the selected dense instance is strongly correlated to the sparse instance in question, which is necessary to enable positive knowledge transfer for the underlying phenomenon ($PM_{2.5}$ in our case). If the sparse and dense instances are similar, the accuracy of the dense map will also benefit from the measurements of sparse instance. Due to the complexity and dynamics of urban $PM_{2.5}$ concentrations, it is challenging to select the best dense instance and properly apply MOGP for accurate $PM_{2.5}$ map generation.

V. Learning-based Dense Instance Selection

This section first shows that traditional unsupervised criteria for dense instance selection leads to negative transfer, and then explains our learning-based dense instance selection in detail.

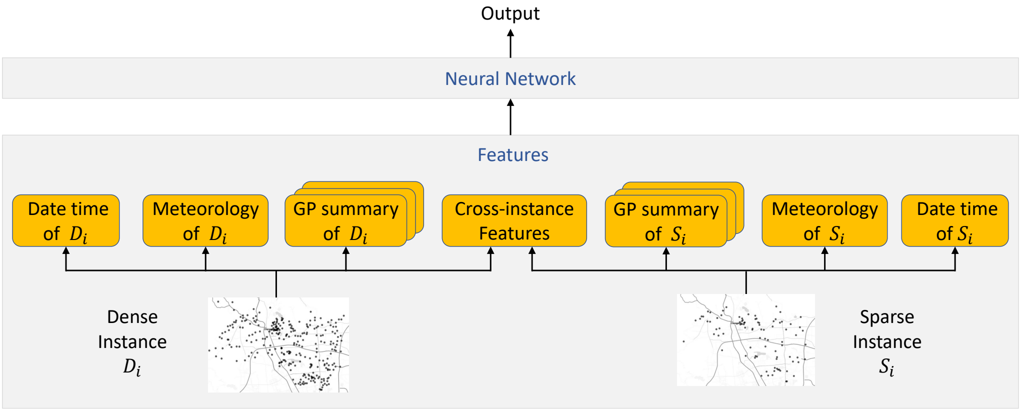


Fig. 4. An illustration of learning-based dense instance selection (LIS).

TABLE I
ALL FEATURES USED IN LIS.

Categories	Features	No.
F_T	*hour of day*, *day of week*, *month* and *isHoliday*	4×2
F_M	*temperature*, *humidity*, *pressure*, *wind speed* and *wind power*	$5 \times 2 \times 9$
F_{GP}	GP Features : *nug_psill*, *nug_range*, *nug_kapple*, *sph_psill*, *sph_range*, *sph_kapple*; Statistical Features : *mean/minimum/maximum* values of all the observations; Cross validation features : *mae, rmse*	$11 \times 2 \times 3$
F_C	*RMSE* and *correlation coefficient*; *co_rmse, co_mae*	4

A. Dense Instance Selection via Unsupervised Criteria

RMSE and correlation coefficient (CORR) are two commonly used unsupervised criteria for instance selection in environmental science [20]–[23]. As a measurement study, we first randomly pick one sparse instance to generate an air quality map using GP and measure its MAE (denoted as E_{GP}) by comparing with the 60 test sensors. Then we select one dense instance with the two selection criteria and generate another air quality map via MOGP, where its MAE is denoted by E_{MOGP}. Hence we can quantify the *transfer gain* by $\Delta E = E_{GP} - E_{MOGP}$, where a positive value means an improvement in air quality map accuracy, and a negative value means a degradation due to negative transfer.

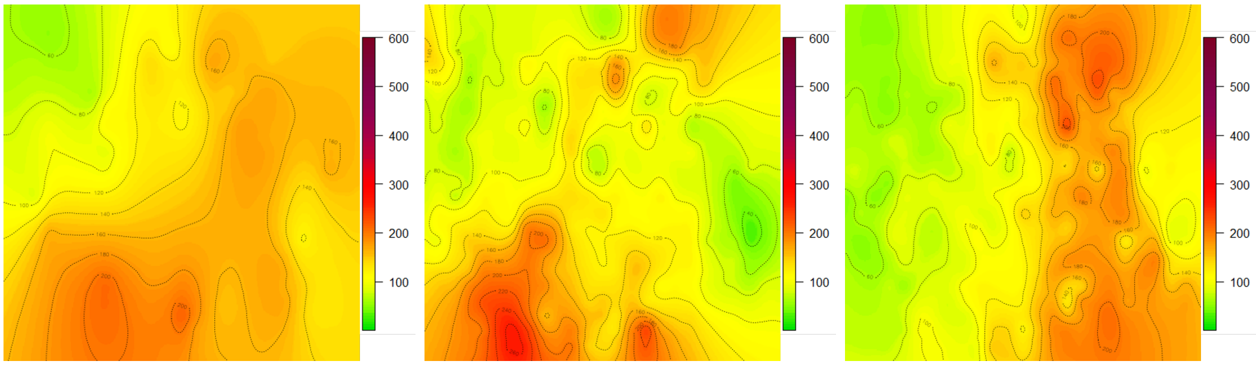
Fig. 3 plots the relationship between transfer gain ΔE and the value of used instance selection criterion, *i.e.*, RMSE (Fig. 3a) or CORR (Fig. 3b). As shown in the figures, there is no strong relationship between the transfer gain and the two criteria. In many cases, the value of ΔE becomes negative, indicating dense instances selected by RMSE or CORR may often decrease the air quality map accuracy.

B. Dense Instance Selection via Supervised Learning

Fig. 4 shows the structure of our learning-based instance selection (LIS) scheme. The core of LIS is a neural network which captures the potential non-linear relationship between the accuracy improvement and the instance similarity. The neural network compares the current sparse instance with each historical dense instance and predicts the *transfer gain* of the dense instance, *i.e.*, how much the accuracy of the generated air quality map will improve by using both the dense and the sparse instances via MOGP, over using only the sparse instance via GP. The neural network also accounts for other meta data such as time and meteorological information when assessing the transfer gain. Given a current sparse instance, LIS will go over all the historical dense instances, selects the top-n dense instances with the highest predicted transfer gain, and then passes these dense instances to the SRS module.

1) Input Features: We pick features from both the dense and sparse instances as well as the corresponding meta data. Specifically, the following categories of features are considered. (see Table I for a complete list of features used in LIS).

- *Date time feature vector F_T.* Intuitively, month and day of week are correlated to the periodical changes in air quality [24]. So we use *hour of day*, *day of week*, *month* and *isHoliday* from both the dense and sparse instances as our date time feature vectors F_T.
- *Meteorological feature vector F_M.* Air quality is influenced by many meteorological factors. We use five meteorological features for instance selection: *temperature*, *humidity*, *pressure*, *wind speed* and *wind power*. These features are collected from 9 meteorological stations in our deployment (see Fig. 3a).
- *GP summary feature vectors F_{GP}.* Since our end goal is to improve the accuracy of the generated air quality maps, it is reasonable to utilize the parameters of the maps *i.e.*, parameters of the Gaussian pro-



(a) Sparse (b) Dense-R (c) Dense-L

Fig. 5. Air quality maps generated by (a) a sparse instance on February 23rd, 2019; (b) one dense instance in the selected from history based on RMSE; and (c) another dense instance selected from history based on RMSE.

cesses as features for instance selection. We choose all the optimized GP parameters as F_{GP} which includes *nug_psill*, *nug_range*, *nug_kapple*, *sph_psill*, *sph_range*, *sph_kapple* [20], [25]. In addition to the above GP parameters features, we also add the statistical values such as *mean/minimum/maximum* values of all the observations, and prediction power index such as the Leave-p-out cross-validation error [26] from both the dense and sparse instances, which is denoted as *s_mae,s_rmse, t_mae, t_rmse*. What's more, to account for the temporal dynamics of air quality maps, we also include the GP summary features one hour before and after the current instance.

- *Cross-instance feature vectors* F_C. Apart from the commonly used unsupervised criterion of *RMSE* and *correlation coefficient* as the interaction feature vectors between dense and sparse instances, we also add another Leave-p-out cross-validation error measurement [26] as an index of how the dense instance helps. This index uses p observations in sparse instance as the validation set, MOGP uses the observations in dense instance and the remaining observations in sparse instance as the training set and test the errors on validation set. This is repeated for all observation in sparse instance in which sparse observations can be separated this way, and then the error is averaged for all trials, to give overall effectiveness. We denote this error as *co_rmse,co_mae*.

2) Training: We only rely on the historical dense data to train the neural network. Specifically, we use the data collected by the sensors in the sparse deployment in the year of 2018 as *training sparse data*. The data from the remaining 150 sensors in the dense deployment in the year of 2018 are used as *target data*. The number of dense instances and the number of sparse instances used for training are both 8650.

To generate the ground truth labels to train the neural network, consider one instance from the training sparse data (*training sparse instance*), one instance at the same hour from the target data (*target instance*), and one arbitrary dense instance. E_{MOGP}^{train} is the MAE of predicting the target instance using both the training sparse instance and the dense instance via MOGP, and E_{GP}^{train} is that using only the training sparse instance via GP. Then the ground truth $\Delta E = E_{GP}^{train} - E_{MOGP}^{train}$ is used to train the neural network.

VI. Sub-Region Selection

Recall that SRS aims to improve the accuracy of generated air quality maps by exploit spatial locality in correlation among instances. In this section, we first demonstrate the spatial locality via measurements, and then explain the two core issues in sub-region selection.

A. Spatial Locality of Correlation between Instances

Here we show that directly using the entire monitoring region of dense instances may lead to sub-optimal performance. Fig. 5a shows an air quality map generated by one sparse instance on February 23rd, 2019, with $E_{GP} = 23.5$. We then select two historical dense instances with lowest RMSE, denoted by Dense-L and Dense-R, and use MOGP to generate two air quality maps, as shown in Fig. 5b and Fig. 5c, respectively. When using these two dense instances for MOGP-based map generation transfer, the transfer gains are 6.9 and 7.6, indicating an improvement to a certain degree. However, it is easy to observe that the left half of Fig. 5b looks very similar to the same left half of Fig. 5a. Also the right half of Fig. 5c looks similar to the right half of Fig. 5a. This indicates that the dense instance Dense-L correlated to the sparse instance more in the left half sub-region, and the Dense-R more in the right half sub-region. If we stitch the left half sub-region of the Dense-L and right half sub-region of the Dense-R together to form a stitched instance, and then use this to augment the current sparse instance, the resulting transfer gain increases to $\Delta E = 13.4$, which almost doubles the transfer gain than using the entire monitoring region of Dense-L or Dense-R individually. Hence due to the spatial locality of correlations between sparse instances and historical dense instances, it is preferable to transfer information from sub-regions of different dense instances, instead of the whole monitoring region of one single dense instance.

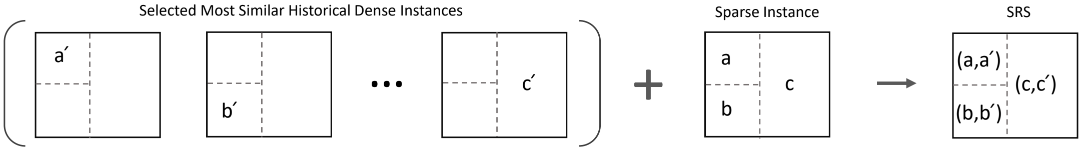


Fig. 6. Proposed Sub-Region Selection method. Suppose the sparse instance is split into three zones (a, b, c) according to a **splitting point**, SRS method tries to find the best match subset in dense instances with the same splitting grid. Then, subset MOGP could be done in each subset zone and produce the overall result.

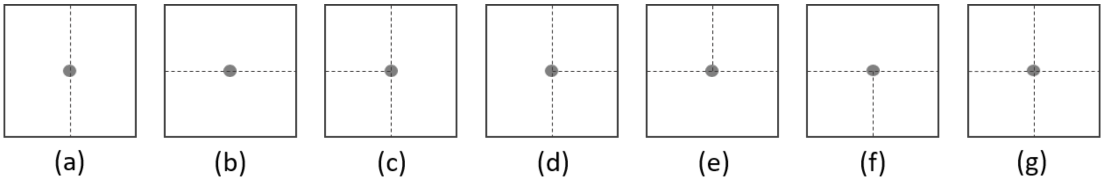


Fig. 7. Potential different splitting methods to split the whole region to 2, 3 or 4 sub-regions.

B. Sub-Region Selection as Two-Step Optimization

The output of SRS is one stitched instance, which is made up of several sub-regions from different dense instances. SRS needs to address the following two issues: *(i)* how to split the sub-regions, and *(ii)* which dense instance contributes the most in each sub-region. This can be seen as a two-step optimization problem whose objective is to maximize the transfer gain of the final stitched instance.

1) Finding the most suitable dense instance for each sub-region: To solve this two-step optimization problem, we start with the second step. Consider one sparse instance A and one dense instance B. Assume that the whole monitoring region is already divided into several sub-regions. In one of the sub-regions, the measurements of instance A form a *sub-instance* a, and that of instance B form b. We calculate the cross validation error of MOGP with a and b, *i.e.*, for each measurement in sub-instance a, we use the rest of the measurements and also sub-instance b to estimate this measurement via MOGP and calculate the estimation error. Then the estimation error is averaged over all measurements in a, and we get the cross validation MOGP (CV-MOGP) error metric. For each sub-region, we select the dense instance with the lowest CV-MOGP error and stitch them together to output the stitched instance.

2) Dividing monitoring region into sub-regions: Given the most suitable dense instance for each sub-region, we can now search for the best splitting scheme to divide the whole monitoring region into sub-regions. We introduce a splitting point to divide the monitoring region into multiple adjacent rectangular sub zones, as shown in Fig. 7. Given a splitting point location and a splitting method, we could use the CV-MOGP error metric to find the most suitable dense instances for each sub-region and compute the overall CV-MOGP error, *i.e.*, the summation of CV-MOGP error of each sub-region. Hence we can use Dual Annealing method [27] to find the best splitting point and best splitting method, which yield the lowest overall CV-MOGP error.

3) Two-step Optimization: We denote the location of splitting point in 2D space with (l_x, l_y), and splitting method with m, which is an categorical variable from (a) to (g) as shown in Fig. 7. The LIS module outputs N_d dense instances, and the number of split sub-regions is denoted by N_s. For the current sparse instance, we denote the CV-MOGP error in the i-th sub-region with the j-th dense instance as $E^{CV}_{i,j}$, where $i = 1, \cdots, N_s$ and $j = 1, \cdots, N_d$. The SRS module addresses the two-step optimization:

$$\min_{(l_x,l_y),m} \sum_{i=1}^{N_s} \min_{j} E^{CV}_{i,j} \tag{3}$$

Finally, SRS combines the sub-region of each selected dense instance into one stitched instance, which is then combined with the current sparse instance to generate an air quality map via MOGP, as shown in Fig. 2.

VII. Evaluation

This section presents the evaluation of MapTransfer. We first introduce the experiment setup (Sec. VII-A) and then present the overall performance (Sec. VII-B). Finally we show the effectiveness of each module (Sec. VII-C and Sec. VII-D).

A. Experiment Setup

Datasets and Metrics. We evaluate the performance of different map generation transfer schemes using the same setting and datasets in Sec. II-B. That is, for a given sparse instance from the sparse dataset, each map generation transfer scheme selects dense instances from the dense dataset and generates an air quality map. We then assess the accuracy of the map using measurements from the 60 testing sensors. Although we conduct our evaluations using data collected in Beijing, China, the principles of our method is not specific to Beijing and applies to other regions as well. We mainly evaluate the accuracy of the air quality map by the mean absolute error

TABLE II

OVERALL PERFORMANCE AND PERFORMANCE IN CASE OF HEAVY POLLUTION ($PM_{2.5}$ CONCENTRATION $> 150\ ug/m^3$) OF DIFFERENT MAP GENERATION TRANSFER METHODS.

Method	Overall Performance		Performance on Heavy Pollution	
	MAE on Test Dataset	Reduction in MAE (%) over Sparse GP	MAE on Test Dataset	Reduction in MAE (%) over Sparse GP
Sparse GP	21.8	-	36.4	-
RMSE + MOGP	18.2	16.5	32.2	11.5
RMSE + SRS + MOGP	16.9	22.5	30.3	16.8
COEF + MOGP	17.1	21.6	26.9	26.1
COEF + SRS + MOGP	16.2	25.7	25.8	29.1
LIS + MOGP	13.6	37.6	20.8	42.9
MapTransfer	**11.8**	**45.9**	**17.9**	**50.8**

(MAE), because MAE is used in various air quality related research, including sensor calibration, spatial interpolation [6], temporal prediction [28] and data mining [19].

Baselines. We compare the performance of our MapTransfer (*LIS* + *SRS* + *MOGP*) with the following baselines.

- *Sparse GP*: It directly generates an air quality map with a sparse instance without any dense instance.
- *RMSE* + *MOGP*: It adopts RMSE for dense instance selection and MOGP for map generation transfer.
- *CORR* + *MOGP*: It applies the correlation coefficient (CORR) for dense instance selection and MOGP for map generation transfer.
- *RMSE* + *SRS* + *MOGP*: It uses RMSE for dense instance selection and SRS to stitch dense instances, before applying MOGP for map generation transfer.
- *CORR* + *SRS* + *MOGP*: It uses CORR for dense instance selection and SRS to stitch dense instances, before applying MOGP for map generation transfer.
- *LIS* + *MOGP*: It uses LIS for dense instance selection and then MOGP for map generation transfer without using SRS to stitch dense instances.

Other Experimental Settings. We implement the *LIS* module using a fully-connected neural network with the architecture of (168,84,22,1), where 168 is the dimension of input features (see Table I) of the neural network and 1 is the dimension of the output *i.e.*, transfer gain $\Delta E = E_{GP} - E_{MOGP}$. The hyper-parameter of the two hidden layer dimensions, (84,21) are selected via grid search [29]. All the codes are implemented in python and the experiments were conducted with a Linux machine with 32 cores.

B. Overall Performance

Table II shows the overall performance of different map generation transfer methods. We are also interested in the performance of these methods in case of heavy pollution ($PM_{2.5}$ concentration $> 150\ ug/m^3$) because accurate air quality maps during heavily polluted days are particularly important for authorities to take proper actions. The overall MAE is 21.8 if only a sparse instance is used to generate the air quality map. Without the assist of any dense instance, the MAE increases to 36.4 in case of high $PM_{2.5}$ concentration. Even the basic map generation transfer methods help reduce the MAEs, *i.e.*, a reduction of 16.5% in overall MAE with *RMSE* + *MOGP* and 21.6% with *CORR* + *MOGP*. Our *LIS* method notably outperforms the two conventional criteria (*RMSE* and *CORR*), achieving a reduction of 37.6% in overall MAE and 42.9% in case of heavy pollution, compared with the air quality map generated with only a sparse instance. Our *SRS* scheme reduces the MAEs with all the three dense instance selection methods. Combining *LIS* and *SRS*, our MapTransfer yields the best performance: a reduction of 45.9% in overall MAE and 50.8% in high pollution cases, compared with *Sparse GP*. The overall MAE is reduced to 11.8.

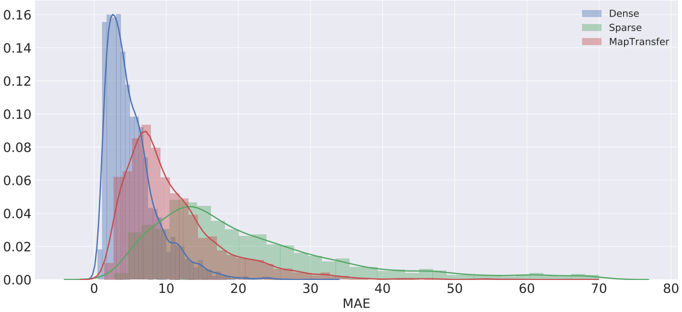
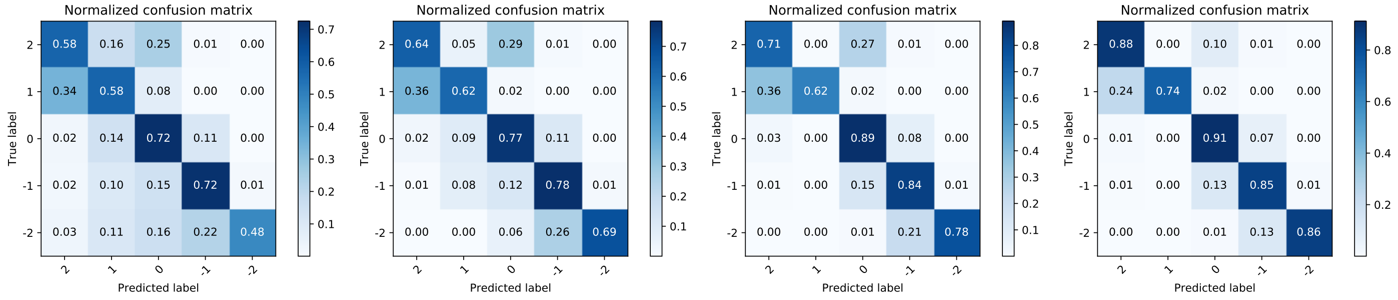


Fig. 8. Histograms of MAEs of air quality maps generated using different transfer methods.

Fig. 8 shows distributions of MAEs using our method. We also plot the MAE distribution using *Sparse GP* and *Dense GP*, where the latter refers to generating air quality maps with a dense instance. The accuracy of the maps generated by *Sparse GP* and *Dense GP* serves as the upper and lower bounds of map generation transfer. As is shown, MapTransfer not only reduces the average MAEs, but also significantly decreases the variance of MAEs.

Summary of Results. MapTransfer is the most effective among all map generation transfer schemes. Compared with air quality maps generated using sparse instances only, it reduces the overall MAE of air quality maps from 21.8 to 11.8, a reduction of 45.9%. The improvement is more significant in



(a) ACC=0.68

(b) ACC=0.75

(c) ACC=0.83

(d) ACC=0.88

Fig. 9. Confusion matrices of prediction accuracy of ΔE by using only $RMSE + CORR$; and sequentially adding (b) all cross-instance features, (c) GP summary features, and (d) meteorological features.

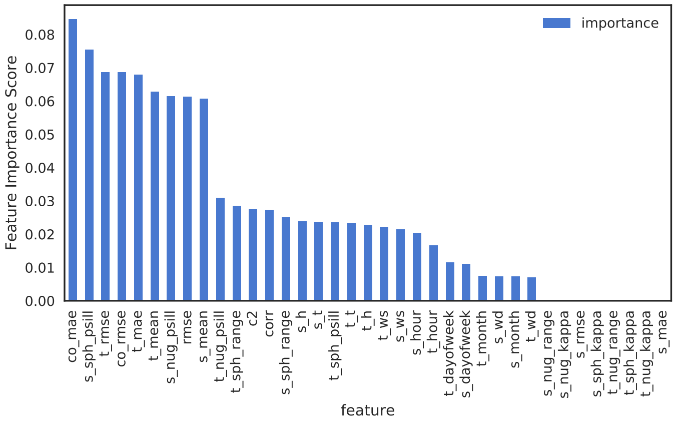


Fig. 10. Feature importance for predicting the transfer gain ΔE.

case of heavy pollution, where the reduction in MAE reaches 50.8%. Meanwhile, MapTransfer also dramatically reduces the variations of errors in the generated air quality maps.

C. Effectiveness of Learning-based Dense Instance Selection

This series of experiments investigates the contributions of different features in LIS on the performance of map generation transfer. For ease of illustration, the predicted transfer gain is quantized into integer labels from -2 to 2, which correspond to ΔE in the following ranges: below -10, -10 to -5, -5 to 5, 5 to 10, and above 10 .

Fig. 9 shows the normalized confusion matrices of the prediction accuracy of ΔE (quantized into an integer from -2 to 2) using different feature vectors described in Sec. V. If only $RMSE + CORR$ are used, the prediction accuracy is only 0.68 with a large variance. The prediction accuracy increases to 0.75 and 0.83 after adding all the cross-instance features and the GP summary features, respectively. When the meteorological features are also added, the final prediction accuracy reaches 0.88. Compared with using only the conventional $RMSE+CORR$, the prediction accuracy improves by about 20% when all features are used.

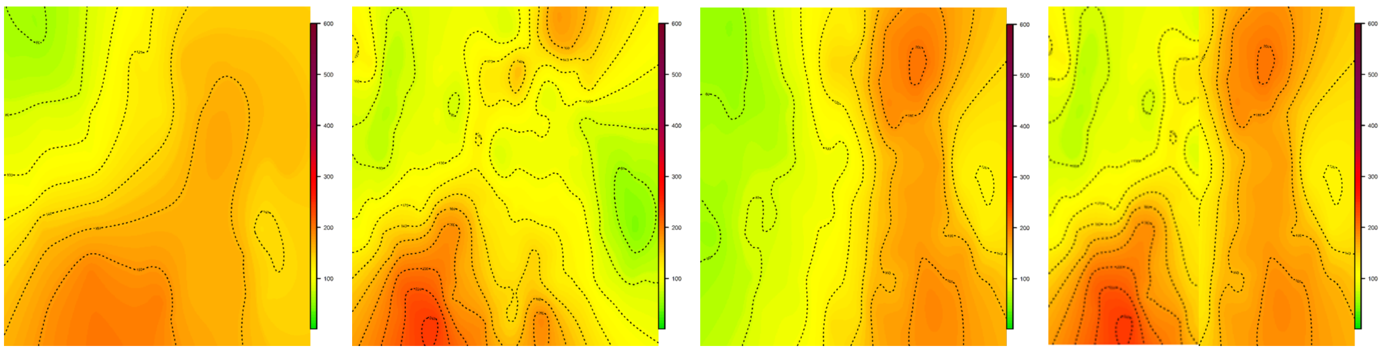
Fig. 10 shows the importance of each feature used in LIS. As is shown, cross-instance features such as *co_mae*, *co_rmse* are significant. GP summary features such as *sph_psill*, *nug_psill* and meteorological features also help improve the prediction accuracy of the transfer gain.

Summary of Results. Using a rich feature set for dense instance selection (see Table I) improves the prediction accuracy of transfer gain ΔE by about 20% than using merely RMSE and CORR. Cross-instance features and Gaussian process summary features are essential for dense instance selection.

D. Effectiveness of Sub-Region Selection

In this subsection, we first take a closer look at the performance of SRS on map generation transfer for a single sparse instance, and then analyze the sub-regions selected by SRS.

Fig. 11a shows an air quality map generated by GP using a sparse instance collected at 4:00 a.m. on February 23rd, 2019. Fig. 11b and Fig. 11c plot the maps generated by GP using the two best dense instances selected by LIS (denoted as Dense-L and Dense-R, respectively). Fig. 11d illustrates the map generated by GP using the best dense instance stitched by SRS (denoted by Dense-S). We plot the maps generated by GP rather than the raw instances for ease of visualization. As is shown, even the best historical dense instances selected by LIS do not resemble the sparse instance in the entire region. Conversely, the dense instance output by SRS, which properly stitches certain sub-regions of the two best dense instances, looks notably more similar to the sparse instance, and potentially results in an air quality map with a higher accuracy. Fig. 12b, Fig. 12c and Fig. 12d show the air quality maps generated by MOGP using the sparse instance and the two best dense instances (Dense-L and Dense-R) as well as the stitched dense instance (Dense-S). Compared with the ground truth in Fig. 12a, the map generated by augmenting the sparse instance with the stitched instance is the most similar to the ground truth. The results are more obvious when we plot the errors of the generated maps in Fig. 13, using only the sparse instance, or augmented by either dense instance or the stitched one. The average MAE of using the sparse instance is 23.5, whereas the average MAE reduces to 14.3 or 15.2 if Dense-L or Dense-R is combined with the sparse instance. The errors are still unsatisfactory and there are notable high-



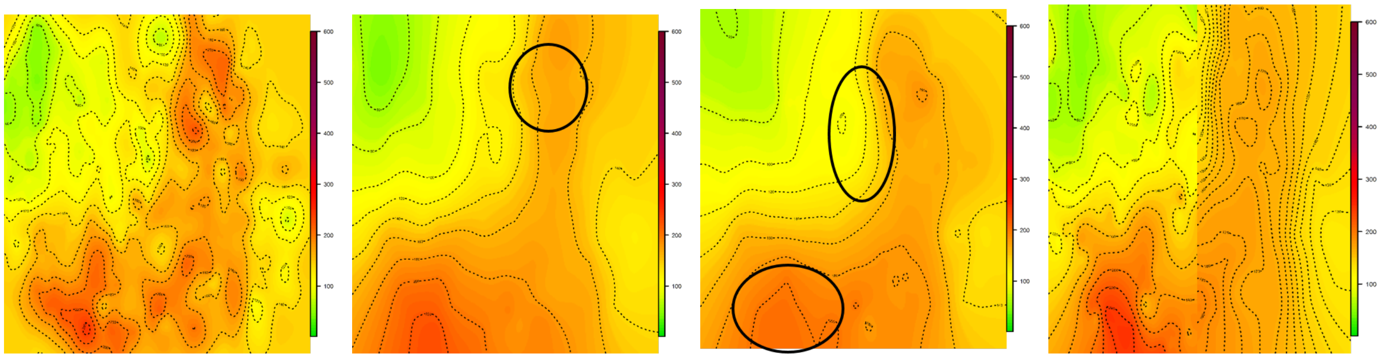
(a) GP(Sparse)

(b) GP(Dense-L)

(c) GP(Dense-R)

(d) GP(Dense-S)

Fig. 11. Air quality maps generated by GP with (a) a sparse instance (denoted as Sparse); (b) one best dense instance selected by LIS (denoted as Dense-L); (c) another best dense instance selected by LIS (denoted as Dense-R); and (d) a dense instance stitched by SRS (denoted as Dense-S).



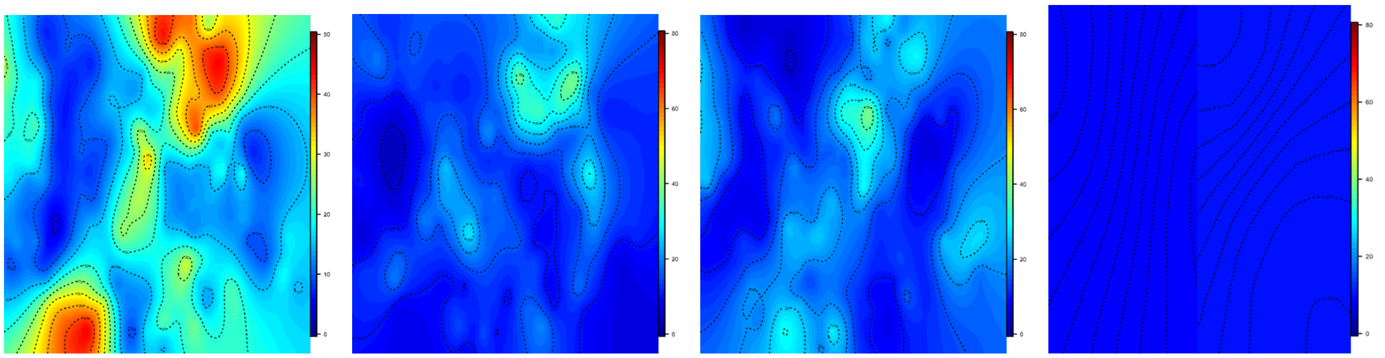
(a) Ground Truth

(b) MOGP(Sparse,Dense-L)

(c) MOGP(Sparse,Dense-R)

(d) MOGP(Sparse,Dense-S)

Fig. 12. (a) Group truth; air quality maps generated by MOGP with the sparse instance *i.e.*, Sparse, and (b) Dense-L, (c) Dense-R, (d) Dense-S. The black circles are areas with large errors.



(a) GP(Sparse)

(b) MOGP(Sparse,Dense-L)

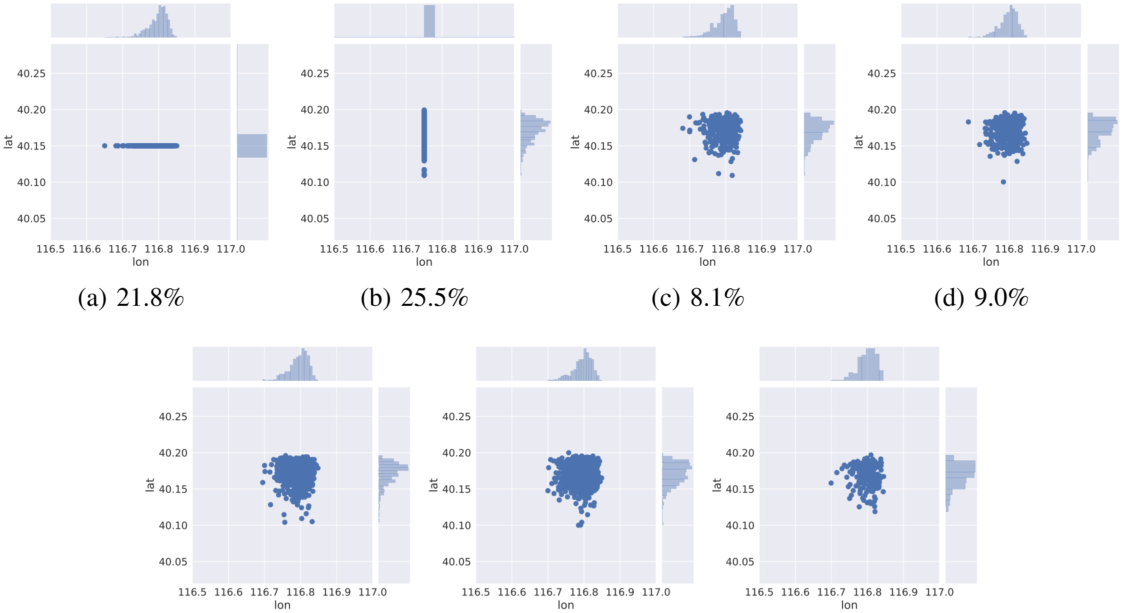
(c) MOGP(Sparse,Dense-R)

(d) MOGP(Sparse,Dense-S)

Fig. 13. Errors of air quality maps generated by (a) GP with Sparse; (b) MOGP with Sparse and Dense-L; (c) MOGP with Sparse and Dense-R; (d) MOGP with Sparse and Dense-S.

error areas in Fig. 13b and Fig. 13c. In contrast, when the stitched dense instance is used along with the sparse instance, the average MAE drops to 10.1, which is acceptable in many data mining applications. More importantly, we observe more evenly distributed errors across the entire region of interests (see Fig. 13d). The results show that SRS is able to eliminate high-error areas in air quality maps and thus improves the overall accuracy of air quality maps.

To understand how the regions are split when applying SRS on the sparse instances, we plot the distributions of splitting points and the number of split sub-regions in Fig. 14. We have the following observations. *(i)* Most splitting points locate around the center of whole monitoring region, which avoids sub-regions with very few sensors. *(ii)* For all the 7 splitting



(a) 21.8%

(b) 25.5%

(c) 8.1%

(d) 9.0%

(e) 14.4%

(f) 17.1%

(g) 4.1%

Fig. 14. Splitting points distributions and ratios. The points in (a) and (b) split the regions into 2 sub-regions; The points in (c), (d), (e) and (f) split the region into 3 sub-regions; The points in (g) splits the regions into 4 sub-regions.

methods described in Sec. VI, the methods which splits the region to 2 sub-regions (Fig. 14a and Fig. 14b) account for 47.3% of all the sparse instances. The methods which split the region to 3 sub-regions (see Fig. 14c, Fig. 14d, Fig. 14e, Fig. 14f) take up 8.1%, 9.0%, 14.4% and 17.1% of all the cases. Finally the method that splits the region to 4 sub-regions (Fig. 14g) only has 4.1% shares among the splitting results.

Summary of Results. Directly transferring a dense instance of the entire region improve the accuracy of the generated air quality maps (an average MAE of about 14.7), yet leads to high-error sub-regions. SRS wisely stitches dense instances, which potentially eliminates high-error sub-regions and thus yields air quality maps of higher accuracy (an average MAE of 10.1). For a rectangular region of $50\ km \times 30\ km$, splitting it into 2 to 3 sub-regions suffices to achieve high accuracy.

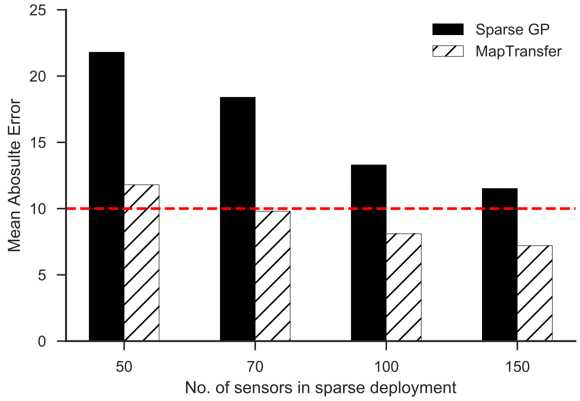


Fig. 15. Impact of number of sensors in sparse deployment on air quality map generation accuracy.

E. Impact of Numbers of Sensors in Sparse Deployment

In this subsection, we evaluate how the number of sensors in the sparse deployment affects the accuracy of the air quality maps and identifies the number of sensors needed to obtain an $MAE < 10$.

Fig. 15 shows the MAE of the air quality maps generated with sparse instances of different numbers of sensors. When the number of sensor in the sparse deployment increases from 50 to 150, the MAE of *sparse GP*, which directly generates an air quality map with a sparse instance without any dense instance, decreases from 21.8 to 11.7.

The MAE of MapTransfer also decreases constantly with the increasing number of sensors in the sparse deployment. When using 70 sensors, the MAE of air quality map generation drops to below 10 (9.8). When increasing the number of sensors in the sparse deployment to 100 and 150, the MAE of MapTransfer is further reduced to 8.1 and 7.2, respectively.

VIII. Discussions

Gaussian Process Regression for Map Generation. Our air quality map generation method is based on the Gaussian Process Regression for the following two reasons. *(i)* We aim to take advantage of the historical data from a dense deployment to improve the air quality map accuracy generated from the sparse deployment. We do not assume access to rich heterogeneous urban data, as required in many other air quality map generation schemes [30]–[32]. Our work is a best-effort exploration on the accuracy maintainable after a dense deployment is downscaled to a sparse one. *(ii)* Gaussian Process regression proves effective in case of a dense sensor deployment [6]. Fusion of additional urban data is complementary to our work and may further improve the accuracy of air quality map generation. However, due to limited access to urban data in the region where our sensors are deployed, it is difficult to implement urban data based air quality inference methods [30]–[32] for direct performance comparison.

Dealing with Changes of Environmental Characteristics. Our solution explicitly assumes that the environmental characteristics relevant to air quality are relatively stable in the long-term. This assumption may break in case of abnormal climate changes. Hence it is important to detect the changes of environmental characteristics in the monitored area. One solution is to exploit the uncertainty of the air quality estimates. Specifically, given the sparse instances, we can use spatial interpolation methods to predict the air quality index with the corresponding uncertainty at a given location, as in [33]–[35]. If the uncertainty at certain locations changes frequently, it indicates that the environmental characteristics have changed and new sensors should be deployed.

Data Duration and Imputation. We choose one year as the length of training dataset based on the following observations: *(i)* Our target is to select the best dense instance from historical data, so one year is a reasonable choice which covers seasonal variations of urban air quality; *(ii)* We only have access to 18 months of data, so we split the data to 2:1 as the training data (one year training data) and test data (half year test data). Note that there can be missing sensor measurements in long-term sensor deployments. In case of missing data, existing missing data completion methods for air quality data such as [36] can be applied before inputting the data into our method.

IX. Related Work

The availability of low-cost sensors and big urban data has revolutionized the landscape of urban air quality monitoring. In addition to conventional model-driven methods [37], [38], there is a growing research interest to generate real-time, fine-grained air quality maps with a data-driven approach [1], [3], [7], [30]–[32], [39].

One thread of data-driven methods emphasize fusion of heterogeneous urban data [30]–[32], [40], [41]. U-Air [32] infers fine-grained air quality information throughout Beijing, China based on the air quality data reported by 35 monitoring stations and a variety of urban data such as meteorology, traffic flow, human mobility, road networks, and point of interests (POIs). Third-Eye [31] feds images, weather data, and $PM_{2.5}$ data into two deep learning models for accurate $PM_{2.5}$ inference. PANDA [30] utilizes a deep multi-task learning based model for air quality prediction by using the 6 monitoring stations in Hangzhou, China and urban features including meteorology, traffic, factory air pollutant emission, road network and POIs. Wei *et al.* [40] propose a multi-modal transfer learning method to transfer knowledge on urban air quality from one city with sufficient multi-modal data and labels, to cities lack of such data and labels. Our work is most relevant to [40] yet differs in two problem settings. *(i)* We aim at knowledge transfer and data augmentation from an initial dense deployment to a downscaled sparse deployment, whereas [40] targets at inter-city knowledge transfer. *(ii)* Our solution is designed for spatial interpolation based map generation methods *e.g.*, Gaussian processes (see below) while [40] is suited for data fusion based air quality inference schemes.

The other category of popular data-driven methods relies more on measurements collected from a large-scale monitoring system with low-cost air quality sensors. The idea is to interpolate air quality reading from measurements collect by sensors nearby. Wong *et al.* [8] compare different spatial interpolation methods for air quality inference and report that Gaussian processes are fit for accurate air quality map generation. AirCloud [1] applies Gaussian process to generate high-quality air quality maps with a large-scale static $PM_{2.5}$ sensor deployment. Jutzeler *et al.* [7] design a region-based Gaussian process model for ultra-fine particle concentration inference with a mobile low-cost deployment, and show that the model yields higher accuracy than land-use regression [3]. Cheng *et al.* [6] compare different spatial interpolation methods given a dense air quality monitoring deployment and find that Gaussian process outperforms the others in terms of the accuracy of the generated air quality maps. Since an initial dense sensor deployment is available in our problem, we mainly adopt Gaussian-based spatial interpolation methods for accurate air quality map generation.

X. Conclusion

In this paper, we propose MapTransfer, an air quality map generation method for downscaled sensor deployments. Key novelties of MapTransfer include a multi-output Gaussian process model to integrate both the sparse and the dense instances, a learning-based dense instance selection module that avoids negative transfer, and a sub-region selection scheme that exploits spatial locality among instances to improve accuracy of air quality map generation. Experiments on real-world air quality sensor deployments show that compared with air quality maps generated with a sparse instance only, MapTransfer reduces the overall MAEs by 45.9%, achieving an air quality map accuracy sufficient for many data mining applications. We envision our work as a practical solution for long-term cost-effective urban air quality monitoring with a downscaled sensor deployment.

Acknowledgements

We are grateful for our shepherd Professor Shahriar Nirjon and the anonymous reviewers for their valuable comments and suggestions. This work was funded in part by the Swiss National Science Foundation (SNSF) under the FLAG-ERA CONVERGENCE project.

References

[1] Y. Cheng, X. Li, Z. Li, S. Jiang, Y. Li, J. Jia, and X. Jiang, "Aircloud: a cloud-based air-quality monitoring system for everyone," in *Proceedings of Conference on Embedded Network Sensor Systems*. ACM, 2014, pp. 251–265.

[2] Y. Gao, W. Dong, K. Guo, X. Liu, Y. Chen, X. Liu, J. Bu, and C. Chen, "Mosaic: A low-cost mobile sensing system for urban air quality monitoring," in *Proceedings of Annual IEEE International Conference on Computer Communications*. IEEE, 2016, pp. 1–9.

[3] D. Hasenfratz, O. Saukh, C. Walser, C. Hueglin, M. Fierz, and L. Thiele, "Pushing the spatio-temporal resolution limit of urban air pollution maps," in *Proceedings of International Conference on Pervasive Computing and Communications*. IEEE, 2014, pp. 69–77.

[4] X. Xu, X. Chen, X. Liu, H. Y. Noh, P. Zhang, and L. Zhang, "Gotcha ii: Deployment of a vehicle-based environmental sensing system," in *Proceedings of Conference on Embedded Network Sensor Systems*. ACM, 2016, pp. 376–377.

[5] C. E. Rasmussen, "Gaussian processes in machine learning," *Advanced Lectures on Machine Learning: ML Summer Schools*, vol. 3176, p. 63, 2004.

[6] Y. Cheng, X. Li, Z. Li, S. Jiang, and X. Jiang, "Fine-grained air quality monitoring based on gaussian process regression," in *International Conference on Neural Information Processing*. Springer, 2014, pp. 126–134.

[7] A. Jutzeler, J. J. Li, and B. Faltings, "A region-based model for estimating urban air pollution," in *Proceedings of AAAI Conference on Artificial Intelligence*. AAAI, 2014.

[8] D. W. Wong, L. Yuan, and S. A. Perlin, "Comparison of spatial interpolation methods for the estimation of air quality data," *Journal of Exposure Science and Environmental Epidemiology*, vol. 14, no. 5, p. 404, 2004.

[9] E. Boldo, S. Medina, A. Le Tertre, F. Hurley, H.-G. Mücke, F. Ballester, I. Aguilera *et al.*, "Apheis: Health impact assessment of long-term exposure to pm 2.5 in 23 european cities," *European Journal of Epidemiology*, vol. 21, no. 6, pp. 449–458, 2006.

[10] D. Hasenfratz, O. Saukh, C. Walser, C. Hueglin, M. Fierz, T. Arn, J. Beutel, and L. Thiele, "Deriving high-resolution urban air pollution maps using mobile sensor nodes," *Pervasive and Mobile Computing*, vol. 16, pp. 268–285, 2015.

[11] F. Ramos, S. Trilles, A. Muñoz, and J. Huerta, "Promoting pollution-free routes in smart cities using air quality sensor networks," *Sensors*, vol. 18, no. 8, p. 2507, 2018.

[12] B. Maag, Z. Zhou, and L. Thiele, "A survey on sensor calibration in air pollution monitoring deployments," *IEEE Internet of Things Journal*, vol. 5, no. 6, pp. 4857–4870, 2018.

[13] O. Saukh, D. Hasenfratz, and L. Thiele, "Reducing multi-hop calibration errors in large-scale mobile sensor networks," in *Proceedings of International Conference on Information Processing in Sensor Networks*. ACM, 2015, pp. 274–285.

[14] Y. Cheng, X. He, Z. Zhou, and L. Thiele, "Ict: In-field calibration transfer for air quality sensor deployments," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 3, no. 1, p. 6, 2019.

[15] P. Kumar, L. Morawska, C. Martani, G. Biskos, M. Neophytou, S. Di Sabatino, M. Bell, L. Norford, and R. Britter, "The rise of low-cost sensing for managing air pollution in cities," *Environment international*, vol. 75, pp. 199–205, 2015.

[16] C. Bellinger, M. S. M. Jabbar, O. Zaïane, and A. Osornio-Vargas, "A systematic review of data mining and machine learning for air pollution epidemiology," *BMC Public Health*, vol. 17, no. 1, p. 907, 2017.

[17] H. Liu, J. Cai, and Y.-S. Ong, "Remarks on multi-output gaussian process regression," *Knowledge-Based Systems*, vol. 144, pp. 102–121, 2018.

[18] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[19] X. Li, Y. Cheng, G. Cong, and L. Chen, "Discovering pollution sources and propagation patterns in urban area," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2017, pp. 1863–1872.

[20] D. Rossiter, "Co-kriging with the gstat package of the r environment for statistical computing," Tech. Rep., 2007.

[21] B. Zhang and Y. Yang, "Spatiotemporal modeling and prediction of soil heavy metals based on spatiotemporal cokriging," *Scientific Reports*, vol. 7, no. 1, p. 16750, 2017.

[22] M. A. Osborne, S. J. Roberts, A. Rogers, S. D. Ramchurn, and N. R. Jennings, "Towards real-time information processing of sensor network data using computationally efficient multi-output gaussian processes," in *Proceedings of International Conference on Information Processing in Sensor Networks*. IEEE, 2008, pp. 109–120.

[23] S. H. Ahmadi and A. Sedghamiz, "Application and evaluation of kriging and cokriging methods on groundwater depth mapping," *Environmental Monitoring and Assessment*, vol. 138, no. 1-3, pp. 357–368, 2008.

[24] Y. Zheng, L. Capra, O. Wolfson, and H. Yang, "Urban computing: concepts, methodologies, and applications," *ACM Transactions on Intelligent Systems and Technology*, vol. 5, no. 3, p. 38, 2014.

[25] G. Matheron, "Principles of geostatistics," *Economic Geology*, vol. 58, no. 8, pp. 1246–1266, 1963.

[26] K. P. Murphy, *Machine learning: a probabilistic perspective*, Cambridge, MA, 2012.

[27] K. M. Mullen *et al.*, "Continuous global optimization in r," *Journal of Statistical Software*, vol. 60, no. 6, pp. 1–45, 2014.

[28] V.-D. Le, T.-C. Bui, and S. K. Cha, "Spatiotemporal deep learning model for citywide air pollution interpolation and prediction," *arXiv preprint arXiv:1911.12919*, 2019.

[29] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[30] L. Chen, Y. Ding, D. Lyu, X. Liu, and H. Long, "Deep multi-task learning based urban air quality index modelling," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 3, no. 1, p. 2, 2019.

[31] L. Liu, W. Liu, Y. Zheng, H. Ma, and C. Zhang, "Third-eye: a mobilephone-enabled crowdsensing system for air quality monitoring," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 2, no. 1, p. 20, 2018.

[32] Y. Zheng, F. Liu, and H.-P. Hsieh, "U-air: When urban air quality inference meets big data," in *Proceedings of SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2013, pp. 1436–1444.

[33] H.-P. Hsieh, S.-D. Lin, and Y. Zheng, "Inferring air quality for station location recommendation based on urban big data," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2015, pp. 437–446.

[34] Y. Yang, Z. Zheng, K. Bian, L. Song, and Z. Han, "Sensor deployment recommendation for 3d fine-grained air quality monitoring using semi-supervised learning," in *2018 IEEE International Conference on Communications (ICC)*. IEEE, 2018, pp. 1–6.

[35] S. D. Narayanan, A. Agnihotri, and N. Batra, "Active learning for air quality station location recommendation," in *Proceedings of the 7th ACM IKDD CoDS and 25th COMAD*, 2020, pp. 326–327.

[36] Y.-F. Zhang, P. J. Thorburn, W. Xiang, and P. Fitch, "Ssim—a deep learning approach for recovering missing time series sensor data," *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 6618–6628, 2019.

[37] N. S. Holmes and L. Morawska, "A review of dispersion modelling and its application to the dispersion of particles: an overview of different dispersion models available," *Atmospheric Environment*, vol. 40, no. 30, pp. 5902–5928, 2006.

[38] S. Vardoulakis, B. E. Fisher, K. Pericleous, and N. Gonzalez-Flesca, "Modelling air quality in street canyons: a review," *Atmospheric Environment*, vol. 37, no. 2, pp. 155–182, 2003.

[39] J. Y. Zhu, C. Sun, and V. O. Li, "Granger-causality-based air quality estimation with spatio-temporal (st) heterogeneous big data," in *2015 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2015, pp. 612–617.

[40] Y. Wei, Y. Zheng, and Q. Yang, "Transfer knowledge between cities," in *Proceedings of SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2016, pp. 1905–1914.

[41] Y. Zheng, X. Yi, M. Li, R. Li, Z. Shan, E. Chang, and T. Li, "Forecasting fine-grained air quality based on big data," in *Proceedings of SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2015, pp. 2267–2276.