Diss. ETH No. XXX

Machine Intelligence for Environmental Analysis and Control

A thesis submitted to attain the degree of

Doctor of Sciences of ETH Zurich
(Dr. sc. ETH Zurich)

presented by
Yun Cheng

born on 29.05.1988
citizen of
China
Beijing

accepted on the recommendation of
Prof. Dr. Lothar Thiele, examiner
Prof. Dr. Christoph Hüglin, co-examiner

2022

Institut für Technische Informatik und Kommunikationsnetze
Computer Engineering and Networks Laboratory

TIK-SCHRIFTENREIHE NR. XXX

Yun Cheng

Machine Intelligence for Environmental Analysis and Control

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

A dissertation submitted to
ETH Zurich
for the degree of Doctor of Sciences

DISS. ETH NO. XXX

Prof. Dr. Lothar Thiele, examiner
Prof. Dr. Christoph Hüglin, co-examiner

Examination date: June 24, 2022

To my family.
Für mini Familie.

Abstract

We are helping to tackle one of the most challenging tasks of our time: Understanding how air quality evolves and implementing intelligent strategies to make earth stay habitable for generations to come.

Current air pollution analysis and control algorithms that are based on data science concepts often fail to produce satisfactory results. Firstly, the data used as input to such algorithms are often neither accurate nor reliable, which especially impacts data-driven approaches. Secondly, long-term operation of a dense sensor deployment incurs enormous maintenance expenses and efforts. Lastly, current prediction methods are inaccurate, which restricts real-time control and multi-region applications.

In this thesis, we propose a closed-loop solution that leverages machine intelligence for environmental analysis and control, bridging the gap between environmental monitoring and immediate cyber-physical or administrative response. Specifically, we offer a holistic view of data analysis by improving the quality of input data, enabling the cost-effective dense deployment, and combining advanced data analysis with model development. This unified view on the whole pipeline from data acquisition to knowledge extraction and decision-making enables the deployment of short-term and long-term mitigation strategies. The main contributions of this thesis are:

- We propose a generalized many-to-many calibration scheme called SensorFormer based on the successful Transformer model which takes both past and future raw measurements into account. The procedure is able to *(ii)* significantly improve the calibration accuracy, *(ii)* boost the performance of compensating altered sensitivity, *(iii)* efficiently run on low-power microcontrollers with very limited computational and storage capabilities.

- We propose In-field Calibration Transfer (ICT), a calibration scheme that transfers the calibration parameters of source sensors (with access to references) to target sensors (without access to references). Experiments show that ICT is able to calibrate the target sensors as if they had direct access to the references.

- 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. This approach greatly improves the cost-effectiveness of dense sensor deployment.

- We propose a new attention based seq2seq model to track pollution propagation for accurate air quality prediction. We evaluate our model on datasets from Beijing area and compare the results to several state-of-the-art baselines. Experiments show that the proposed approach can successfully capture pollution transfer patterns between different sites in the area, which is crucial knowledge for making pollution control strategies.

- We propose iSpray, the first-of-its-kind data analytics engine for fine-grained $PM_{2.5}$ and PM_{10} control at key urban areas via cost-effective water spraying. iSpray combines domain knowledge with machine learning to profile and model how water spraying affects $PM_{2.5}$ and PM_{10} concentrations in time and space. It also utilizes predictions of pollution propagation paths to schedule a minimal number of sprayers to keep the pollution concentrations at key spots under control. In-field evaluations reveal the effectiveness of iSpray.

Zusammenfassung

Wir tragen dazu bei, eine der größten Herausforderungen unserer Zeit zu bewältigen: Zu verstehen, wie sich die Luftqualität entwickelt, und intelligente Strategien umzusetzen, damit die Erde auch für kommende Generationen bewohnbar bleibt.

Aktuelle Algorithmen zur Analyse und Kontrolle der Luftverschmutzung, die auf datenwissenschaftlichen Konzepten beruhen, liefern oft keine zufriedenstellenden Ergebnisse. Erstens sind die Daten, die als Input für solche Algorithmen verwendet werden, oft weder genau noch zuverlässig, was insbesondere datengesteuerte Ansätze beeinträchtigt. Zweitens verursacht der langfristige Betrieb eines dichten Sensoreinsatzes enorme Wartungskosten und -aufwände. Und schließlich sind die derzeitigen Vorhersagemethoden ungenau, was die Echtzeitsteuerung und Anwendungen mit mehreren Regionen einschränkt.

In dieser Arbeit schlagen wir eine geschlossene Lösung vor, die maschinelle Intelligenz für die Umweltanalyse und -kontrolle einsetzt und die Lücke zwischen Umweltüberwachung und unmittelbarer cyber-physikalischer oder administrativer Reaktion überbrückt. Wir bieten eine ganzheitliche Sichtweise der Datenanalyse, indem wir die Qualität der Eingabedaten verbessern, eine kosteneffiziente, dichte Bereitstellung ermöglichen und fortschrittliche Datenanalyse mit Modellentwicklung kombinieren. Diese einheitliche Sicht auf die gesamte Pipeline von der Datenerfassung bis zur Wissensextraktion und Entscheidungsfindung ermöglicht den Einsatz kurz- und langfristiger Abhilfestrategien. Die wichtigsten Beiträge dieser Arbeit sind:

- Wir schlagen ein verallgemeinertes Many-to-Many-Kalibrierungsschema namens SensorFormer vor, das auf dem erfolgreichen Transformer-Modell basiert und sowohl vergangene als auch zukünftige Rohmessungen berücksichtigt. Das Verfahren ist in der Lage *(ii)* die Kalibrierungsgenauigkeit deutlich zu verbessern, *(ii)* verbesserung der Leistung der kompensierenden geänderten Empfindlichkeit, *(iii)* effizient auf stromsparenden Mikrocontrollern mit sehr begrenzten Rechen- und Speicherkapazitäten laufen.

- Wir schlagen In-Field Calibration Transfer (ICT) vor, ein Kalibrierungsschema, das die Kalibrierungsparameter von Quellensensoren (mit Zugang zu Referenzen) auf Zielsensoren (ohne Zugang zu Referenzen) überträgt. Experimente zeigen, dass ICT in der Lage ist, die Zielsensoren so zu kalibrieren, als ob sie direkten Zugang zu den Referenzen hätten.

- Wir entwickeln MapTransfer, ein Verfahren zur Erstellung von Luftqualitätskarten, das die aktuellen Sensormessungen aus dem herunterskalierten spärlichen Einsatz mit geeigneten historischen Daten aus dem anfänglichen dichten Einsatz ergänzt. Dieser Ansatz verbessert die Kosteneffizienz des dichten Sensoreinsatzes erheblich.

- Wir schlagen ein neues, auf Aufmerksamkeit basierendes seq2seq-Modell vor, um die Ausbreitung der Verschmutzung zu verfolgen und die Luftqualität genau vorherzusagen. Wir evaluieren unser Modell anhand von Datensätzen aus dem Raum Peking und vergleichen die Ergebnisse mit verschiedenen State-of-the-Art-Baselines. Experimente zeigen, dass der vorgeschlagene Ansatz erfolgreich Verschmutzungsübertragungsmuster zwischen verschiedenen Standorten in der Region erfassen kann, was ein entscheidendes Wissen für die Entwicklung von Strategien zur Verschmutzungskontrolle ist.

- Wir schlagen iSpray vor, die erste Datenanalysemaschine ihrer Art für die feinkörnige Kontrolle von $PM_{2.5}$ und PM_{10} in wichtigen städtischen Gebieten durch kosteneffizientes Wassersprühen. iSpray kombiniert Fachwissen mit maschinellem Lernen, um ein Profil zu erstellen und zu modellieren, wie sich das Sprühen von Wasser auf die Konzentrationen von $PM_{2.5}$ und PM_{10} in Zeit und Raum auswirkt. Es nutzt auch Vorhersagen über die Ausbreitungswege der Verschmutzung, um eine minimale Anzahl von Sprühern zu planen, um die Verschmutzungskonzentrationen an Schlüsselstellen unter Kontrolle zu halten. Die Wirksamkeit von iSpray wurde in Feldversuchen unter Beweis gestellt.

Acknowledgements

Contents

1

Introduction

Urban air pollution threatens the health of the world's population. Yet 91% of all humans live in areas where air quality levels exceed WHO limits, which causes 8.8 million extra deaths a year worldwide. In response, many cities have deployed large-scale sensor networks to monitor urban air pollution, generate fine-grained air quality maps, and forecast heavily polluted areas for citizens to adjust their journey plans accordingly.

In addition to passive monitoring of urban air pollution, active control strategies are critical for improving urban air quality. Governments and agencies have implemented various policies and regulations to reduce emissions from factories, transportation, and households in order to improve overall air quality (e.g., at the city scale or annual average). However, these regulations currently rely on potentially inaccurate and non-real-time inputs and do not provide smart, balanced control suggestions. For example, current city-scale strategies may propose closing all nearby factories to meet requirements, rather than identifying and shutting down the polluters who dominate the current pollution in relevant regions. There is also a lack of fine-grained (e.g., specific districts, hourly averages) air pollution control measures. Administrators need explainable decision support to control and reduce air pollution for the benefit of society.

Deriving useful knowledge from low-cost air quality sensor data and proving intelligent control strategies in multiple scales (e.g., city-scales and district-scale) is challenging. Firstly, there is no systematic data processing pipeline to calibrate the low-cost sensors in the wild. These imprecise data make it difficult to extract useful knowledge from them. Secondly, maintaining the large-scale sensor network needs huge amounts of efforts and naïvely downscaling the deployment brings significant errors. Thirdly, current data-driven air quality prediction methods fail to capture the pollution transfer between cities, which are not

able to generate accurate prediction results and hurt the performance of pollution control. Lastly, there lack precise pollution reduction strategies with water spraying system.

In this dissertation, we propose components for a closed-loop solution that uses machine intelligence for environmental analysis and control. Those components apply data-driven approaches to provide accurate and reliable analysis results and intelligent control solutions. The presented solutions bridge the gap between air quality detection and pollution control. It is equipped with a data calibration and transfer module to improve the accuracy of measurement data, making them useful for the subsequent analysis pipeline. A map generation method with downscaled deployment is proposed to transfer the knowledge from historical dense data and boost the performance of current sparse data, thus leading to a cost-effective sensor deployment. To characterize the pollution transfer patterns between cities, we use the air flow trajectory data in a data-driven prediction framework. Lastly, we invented a first-of-its-kind precise control measures to protect critical urban spots from heavy air pollution.

Deploying and analyzing air quality sensing network requires a careful design of the system and needs to solve the following critical challenges.

Calibration in the Wild. The deployed low-cost sensor network needs regular calibration to assure the accuracy and reliability of the measurements. Providing powerful calibration models and their transferability to other sensor nodes is the key of this problem.

Cost-effective Deployment. Dense deployed low-cost sensor networks need huge efforts to maintain their running stage. How to reduce the cost while keeping an acceptable analysis accuracy remains to be solved.

Prediction and Control. Current air quality prediction models still lack precision, especially during those sudden change periods, which are, however, critical time slots for air pollution control. Additionally, they lack find-grained air pollution control relying on water spraying systems.

In the rest of this chapter, we will give a detailed review of the above three challenges and provide intuitions of how to solve them.

1.1 Calibrating Air Quality Networks in the Wild

We are considering air quality networks that typically consist of tens to hundreds of low-cost sensors installed either at static locations or on mobile vehicles to measure major air pollutant concentrations in real time [CLL+14b, JSBT+15, SHT15a]. If deployed in long-term and at large-scale, the network can provide fine-grained air quality information for quantitative studies and public services [YLM+15].

Unfortunately, the raw measurements reported by these low-cost air quality sensors can be seriously inaccurate compared to the measurements of expensive monitoring stations [CDS+17, JHW+16b]. Researchers report a significant accuracy drift in sensor measurements after only 1 month of deployment [MMH17a], making these measurements unreliable for quantitative studies. The reasons for such an accuracy degradation are linked to various limitations of low-cost air quality sensors, such as low selectivity and environment-dependent interference [SGV+15, SGV+17].

An effective approach to improve the data quality of low-cost air quality sensors is *calibration* [MZT18a]. By calibrating a low-cost sensor, its measurements are transformed in a way that the calibrated measurements agree with the measurements of a highly accurate reference e.g., the monitoring station. Although air quality sensors are often calibrated before deployment, the calibration parameters of the entire air quality network have to be frequently updated after deployment to maintain consistency among distributed sensors and ensure data quality of long-term deployments.

1.1.1 Calibration Method

A low-cost air quality sensor is calibrated via a *calibration model* to improve its data quality. A calibration model takes the raw measurements of a low-cost sensor and transforms them to calibrated measurements, leveraging prior knowledge e.g., data sheets or additional information e.g., measurements from auxiliary sensors. Various mathematical methods can be applied.

A *sensor array* is formed to compensate for low-selectivity and environmental dependencies of low-cost air quality sensors. A sensor array consists of co-located sensors that measure, in addition to the target air pollutants, a set of correlated pollutants and environmental parameters e.g., temperature. By concurrently measuring all the cross-sensitivities it is possible to compensate for all interfering pollutants and environmental factors. In fact, an increasing number of customized [TDMP16] and commercial [SGX14] air quality sensing nodes are integrated with multiple correlated sensors and report measurements of pollutants and environmental parameters simultaneously.

Popular mathematical methods for sensor array calibration include Multiple Least Squares (MLS) and Neural Networks (NN).

- **Multiple Least Squares (MLS).** A linear calibration model is sufficient for cross-sensitivity problems of certain gases. One of the most popular examples is the cross-sensitivity of NO_x electrochemical sensors on O_3 concentrations [MSHT16a], and vice-versa [PSL+17]. The effect of these cross-sensitivities follow a linear behavior. Hence a linear multiple least squares calibration can be successfully applied. It is shown that the measurement error

of the cross-sensitive NO_2 sensor can be reduced by over 80% by simply incorporating measurements of an additional O_3 sensor in the calibration [MSHT16a].

- **Neural Networks (NN).** It is shown that for a wide range of low-cost gas sensors, neural network based sensor array calibration outperforms linear models such as multiple least squares [SGV+15, SGV+17, VPMF09, VES+18]. For multiple O_3 and NO_2 sensors the coefficient of determination R^2 is improved from values below 0.3 to at least 0.85 and 0.55 respectively using neural networks instead of linear models [SGV+15, SGV+17].

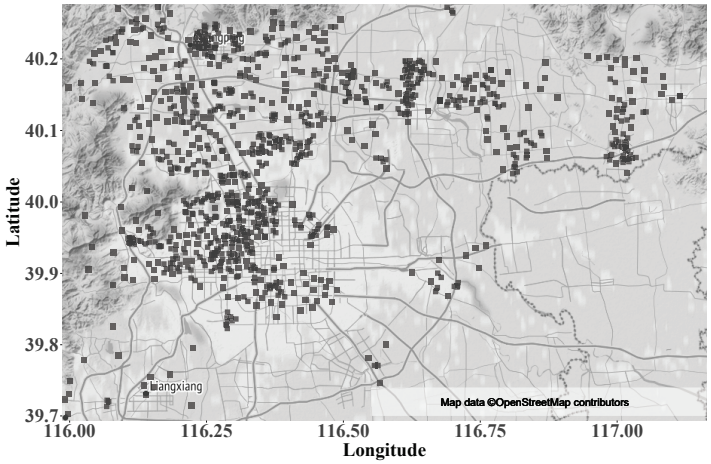
The common practice is to start with linear models before adopting complex non-linear models such as neural networks for sensor array calibration. Although neural networks dominate in non-linear sensor array calibration models, other machine learning methods [VES+18] also apply, e.g., Gaussian Process Regression [MLB+12], Support Vector Regression [SWL+11, VVMH12], Random Forest [ZPK+18] etc.

1.1.2 Calibration Scheme

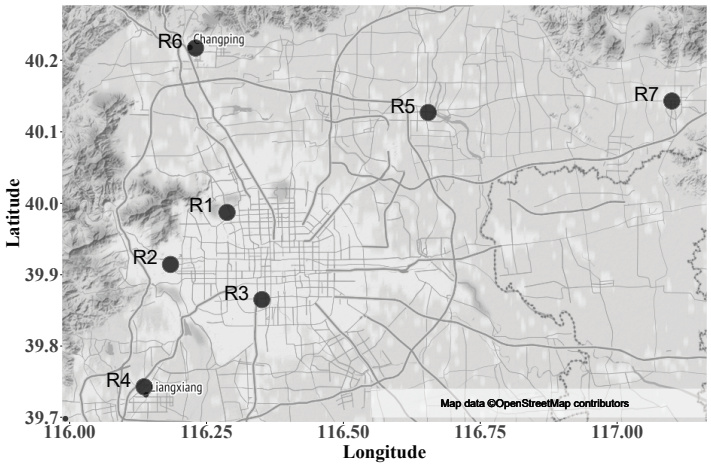
Many interesting environmental sensors require frequent calibration to maintain high quality sensor measurements [MZT18b]. Existing machine learning sensor calibration models can be classified with respect to the dimensionality of their inputs and outputs into *one-to-one* and *many-to-one* calibration methods.

One-to-one calibration models are parameterized mappings from context measurements at a single time point to a calibrated value at the same time point. Various machine learning methods are applied to calibrate low-cost air quality sensor values under the one-to-one scheme. Early linear regression models were used to improve field performance of low-cost gas sensors [SHT15b, MSHT16b, MZST17a]. A neural network was proposed to calibrate PM2.5 sensors in the field in [CLL+14c]. [LDC18a] proposed to combine a linear model with a random forest to further improve calibration accuracy. We refer an interested reader to two survey papers on sensor array calibration [MZT18b, CML+21] for an in-depth overview.

Newly, a many-to-one approach was applied to sensor calibration [YLG+20]. The proposed encoder-decoder architecture takes *historical measurements* into account for the next time step prediction. Two major issues prevent this approach from being used in real settings. (1) Reference data over the past time period must be provided as model input. These historical reference values help to remove drift in the next time step calibration and obtain accurate results. However, no historical reference measurements are available for the vast majority of deployed low-cost sensors. (2) The proposed deep learning model is



(a) Low-cost sensors



(b) Public Stations

Figure 1.1: An illustration of a static $PM_{2.5}$ sensor network deployed in Beijing, China. Among the $1,000$ low-cost $PM_{2.5}$ sensors, only 7 are installed next to highly accurate reference stations.

computationally expensive and hard to be deployed on edge sensors, resource-constrained and often battery-powered IoT devices.

1.1.3 Calibration Opportunities

Recall that many error sources of low-cost air quality sensors are environment-dependent. Thus periodic re-calibration is indispensable even if the sensor was calibrated via proper calibration models before its deployment. Both pre-deployment and post-deployment calibration usually shares the same calibration model. However, post-deployment calibration faces additional challenges because sensors often have irregular or no access to references after deployment.

Figure 1.1 shows the sensor locations of a $PM_{2.5}$ monitoring network

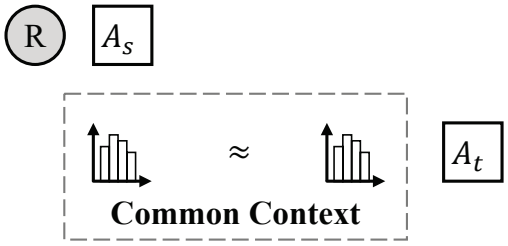


Figure 1.2: An illustration of calibration opportunities for air quality networks with common contexts for static sensors. R denotes a highly accurate reference. A_s and A_t represent sensors with and without access to a reference sensor after deployment.

in Beijing, China. Among the 1,000 low-cost $PM_{2.5}$ sensors deployed, only 7 are next to highly accurate monitoring stations and thus have access to references for calibration. Therefore, only these sensors can be periodically re-calibrated after deployment [CHZT19a].

Network calibration aims to calibrate an air quality network where not all sensors have access to highly accurate references. In principle, network calibration methods exploit various *calibration opportunities* in the air quality network to propagate the calibration from sensors which have access to references to those which do not.

Calibration is possible only if the sensors involved are assumed to be measuring the same physical phenomena. The calibration opportunity to propagate calibration in an air quality network from sensors with access to references to those without holds true if *common contexts* for *static* sensors exist (see Figure 1.2).

Common contexts refer to situations when the air pollutant *concentrations* at different locations or their *specific measures* are expected to be approximately the same. Identifying common contexts requires extra domain knowledge or empirical studies. For example, to calibrate low-cost O_3 sensors, Moltchanov et al. [MLE+15] assume that the O_3 concentration is uniform during nighttime (01:00-04:00 AM), when local emissions of precursors, e.g., NO_2 traffic emissions, are negligible. We [CHZT19a] observe that the distributions of $PM_{2.5}$ concentrations within one month at different locations in the same city exhibit similar patterns and can be utilized for calibration transfer.

1.2 Cost-effective In-field Deployment

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.

1.2.1 Low-cost Sensor Deployment and Research

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 [HM06, VFPGF03], there is a growing research interest to generate real-time, fine-grained air quality maps with a data-driven approach [CDL+19a, ZSL15, CLL+14b, HSW+14, JLF14, LLZ+18, ZLH13a].

One thread of data-driven methods emphasize fusion of heterogeneous urban data [CDL+19a, LLZ+18, WZY16, ZLH13a, ZYL+15a]. U-Air [ZLH13a] 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 [LLZ+18] feds images, weather data, and $PM_{2.5}$ data into two deep learning models for accurate $PM_{2.5}$ inference. PANDA [CDL+19a] 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. [WZY16] 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.

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. [WYP04] compare different spatial interpolation methods for air quality inference and report that Gaussian processes are fit for accurate air quality map generation. AirCloud [CLL+14b] applies Gaussian process to generate high-quality air quality maps with a large-scale static $PM_{2.5}$ sensor deployment. Jutzeler et al. [JLF14] 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 [HSW+14]. Cheng et al. [CLL+14a] 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.

1.2.2 Downscaled Sensing Opportunities

Advances in air quality sensor technologies have enabled urban-scale sensor deployments for fine-grained air pollution monitoring [CLL+14b, GDG+16, HSW+14, XCL+16]. With densely deployed sensors, real-

time, spatially-resolved air quality maps can be generated by spatial interpolation models like Gaussian processes [Ras04], even without training complex models and integrating heterogeneous data sources [CLL+14a, CLL+14b, JLF14, WYP04]. 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 [BMLT+06], is constantly monitored in many major cities in China via large-scale static [CLL+14b] or mobile sensor networks [GDG+16, XCL+16]. 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 [HSW+15, RTMH18].

Although many dense air quality sensor deployments have been reported from both the academia and industries [CLL+14b, GDG+16, HSW+14, XCL+16], 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 [MZT18b, SHT15a], and many may break after 3 months [CHZT19a]. 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 [KMM+15]. Particularly, a downscaled deployment may only contain a 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, the 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, whereas an MAE below 10 is considered accurate for applications such as urban $PM_{2.5}$ maps [BJZOV17a, CLL+14b, CHZT19a].

1.3 Air Quality Prediction

Accurately predicting air quality, especially its sudden changes, is highly valuable for citizens and governments to make personal and local decisions, design intelligent policies and control pollution at minimal cost. However, none of the existing methods achieves sufficient prediction accuracy for time intervals of sudden pollution change due to inability of existing models to take into account pollution propagation between different areas caused by air mass movement.

1.3.1 Literature Review

Recent works on air quality prediction mainly rely on deep learning models. FFA [ZYL+15b] was one of the first attempts to apply a data-driven method that considers the current meteorological data, weather forecasts and air quality data. The proposed hybrid model learns the relationship between spatial and temporal features. However, the shallow ensemble method failed to capture the complex interactions between influential factors. DeepAir [YZW+18] was proposed to learn air pollution patterns in a deep manner by simultaneously considering the individual and holistic influences.

To further improve the model capacity, GeoMAN [LKZ+18] used a three stage attention applied to local features, global features and temporal sequences for geo-sensory series prediction. This approach shows a potential to learn the dynamic spatio-temporal correlations and to interpret the model output. Lin et al. [LMG+18] tried to represent the spatial correlation in a graph with automatically selected important geographic feature types that largely affect $PM_{2.5}$ concentrations, and uses those important geographic features to compute the adjacency graph for the model. To conquer the challenge of lacking training samples, Chen et al. [CDL+19b] proposed a multi-task based approach to learn the representations of the relevant spatial and sequential data, as well as to build the correlation between air quality and these representations. Zhang et al. [ZLG+19] found that local fine-grained weather data is helpful for accurate air quality prediction. Their method fuses heterogeneous weather, air quality and Point-of-Interest (POI) data to learn interactions between different feature groups. Ensemble methods, such as the winning solution to air quality prediction for KDD Cup 2018 [LHH+19], are also used to further improve the accuracy of air quality prediction tasks. The winning Solution to Air Quality Prediction for KDD Cup 2018 [LHH+19] includes LightGBM, Gated-DNN and Seq2Seq models.

1.3.2 Challenges and Intuitions

Accurate air quality prediction, especially forecasting $PM_{2.5}$ concentrations, is an effective way of protecting public health by providing an early warning against harmful air pollutants [BWML18]. For example, air quality in Beijing can sometimes change from a good to an unhealthy level within a few hours due to pollution transfer from sources outside of the city, which is referred to as air quality sudden change. Being able to predict such sudden changes is vital to inform people and governments for decision-making, but very difficult to achieve due to sparsity of air quality monitoring observations and the underlying complex evolving environment [ZYL+15b].

As reported in the recent literature [ZLG+19, LKZ+18, ZYL+15b, YZW+18, LHH+19], sudden change predictions are challenging and cause

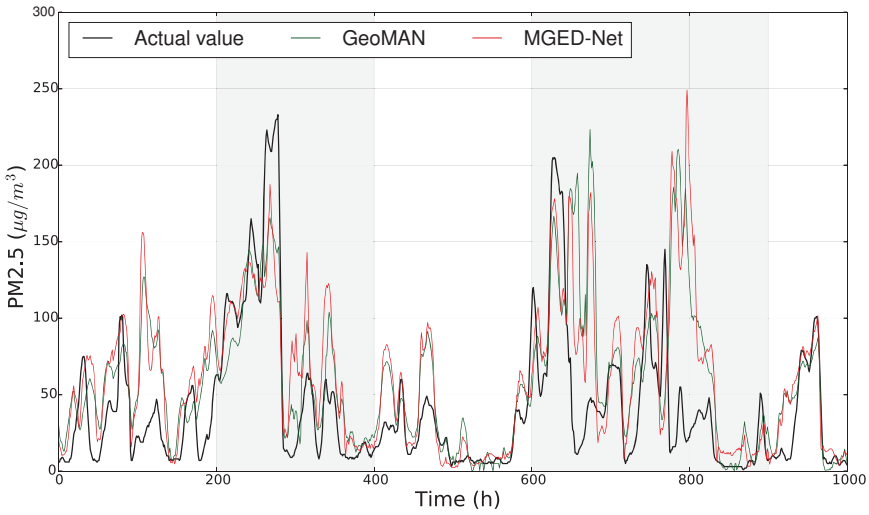


Figure 1.3: Prediction challenge: Shaded areas show time periods of sudden pollution change that appear difficult to handle by the state-of-art prediction methods leading to high errors.

high prediction errors. For example, for one of the 35 public air quality stations in Beijing, the mean value prediction for the next 19 to 24 hours is shown in Figure 1.3. We choose two state-of-art methods to perform the prediction task. Both methods take past air quality measurements and weather information in Beijing as input to compute a short-term prediction. We can observe that the overall mean absolute error (MAE) achieved by the state-of-the-art methods GeoMAN [LKZ+18] and MGED-Net [ZLG+19] is 22.4 and 20.2 respectively. However, both methods exhibit high prediction inaccuracies in the shaded zones of the plot which correspond to sudden change intervals.

Some existing works [ZZZ+17, DWC+19] already highlighted the importance of pollution transfer from surrounding areas, the phenomenon we refer to as pollution transfer. For example, pg-Causality [ZZZ+17] uses frequent pattern mining and Bayesian learning to identify spatio-temporal causal pathways for air pollutants of Beijing. In environmental science, HYSPLIT [SDR+15] is widely adopted to identify regional pollution sources [KVL+11] and propagation pathways [MC08].

As stated in research work [KVL+11], air flow trajectory analysis is one of the standard procedures for determining the spatial locations of possible emission sources affecting given receptors, and it is frequently used to enhance receptor modeling results. Furthermore, McGowan et al. [MC08] and Li et al. [LCCC17] identify regional pollution sources and propagation pathways. Based on the air flow trajectory data, Gao et al. [GTC+15] conduct research on the formation causes during two haze pollution events in urban Beijing, China. The results show that regional transport contributes to the elevated content of anthropogenic

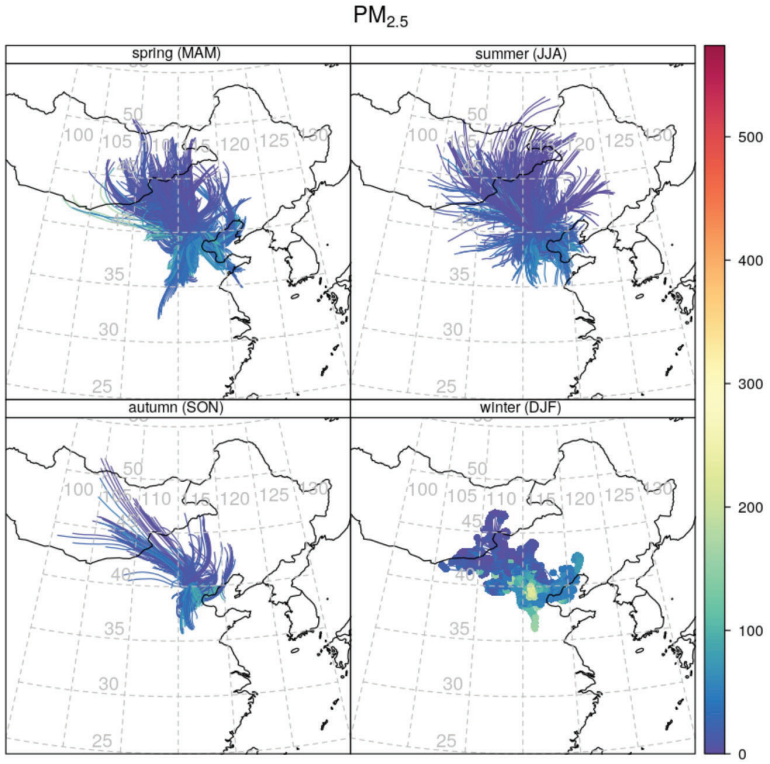


Figure 1.4: 24-hour HYSPLIT trajectories centred at Beijing, aggregated over a year and colored by the concentration of $PM_{2.5}$.

elements in $PM_{2.5}$. Wang et al. [WCC+10] also show that air pollution in urban cites is caused not only by local emission sources but also to a large extent by regional atmospheric pollution transport from surrounding areas, responsible for sudden pollution changes.

Back trajectories are extremely useful in air pollution and can provide important information on air mass origins. Figure 1.4 shows backward air flow trajectories aggregated over a year, centered in Beijing and colored by measured $PM_{2.5}$ values. We can observe that: (i) the air quality very probably worsens when the main air flow comes from the south of Beijing, and (ii) the air flow patterns and air quality evolving behavior differs significantly over a year, which motivates the need to take the datetime features into account to learn seasonal pollution variability.

Figure 1.5 plots the Potential Source Contribution Function (PSCF), which calculates the probability that a source is located at latitude i and longitude j [FMM12]. This result gives a clear indication that the principal (high) sources are dominated by source origins in the south of Beijing, predominantly in the south-west. The PSCF approach has been widely used in the analysis of air mass back trajectories. In our settings, we can use this approach to first analyze the air flow data in our target city and find all related surround cities.

One intuition to generate accurate sudden change predictions comes

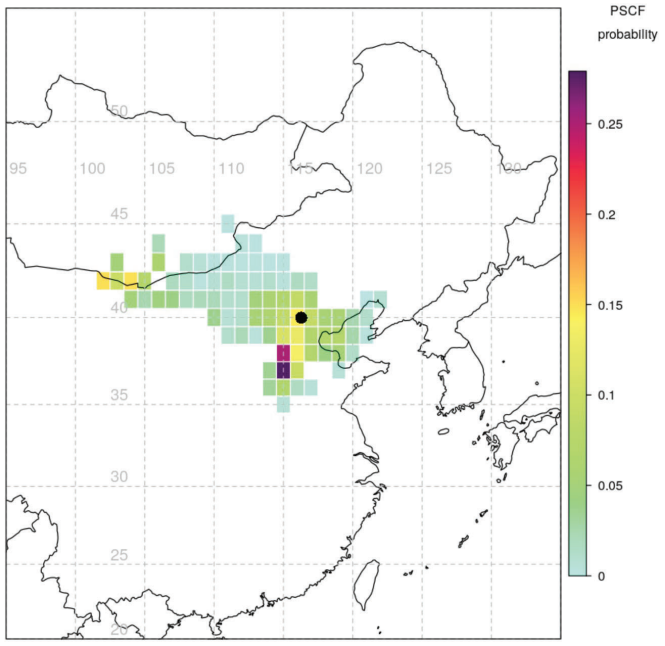


Figure 1.5: Potential Source Contribution Function (PSCF) for Beijing, which yields the probability that a pollution source is located at latitude i and longitude j [FMM12].

from the related works on pollution propagation analysis using HYSPLIT, that the air flow trajectories provide a useful representation to model pollution transfer between different areas.

1.4 Air Pollution Control

Despite regulations and policies to improve city-level air quality in the long run, there lack precise control measures to protect critical urban spots from heavy air pollution.

1.4.1 Ubiquitous Urban Air Pollution Sensing and Inference

The availability of portable sensors and urban data has enabled ubiquitous urban air pollution monitoring and inference services. Installed at hot spots [CLL+14b, CHZT20a, RSPB21], vehicles [HSW+15, JLF14, MLX+20, WXL+20] or carried by citizens [MZT18c, TDMP16], the low-cost gas and dust sensors provide real-time and fine-grained measurements to analyze urban airborne pollutant concentrations. With measurements collected from a large-scale deployment, the accurate air quality map can be generated via spatial interpolation such as Gaussian process [CLL+14b, CHZT20a, JLF14]. Access to air quality related urban data such as meteorological conditions, traffic flows, emission sources has enabled accurate air quality map generation with sparse sensor deployments by designing dedicated inference models

such as spatiotemporal co-training [ZLH13a], weather-aware auto-encoder [MLX+20], etc. Integrating sensor measurements with urban data also facilitates analytics beyond map generation. Examples include simultaneous air quality estimation and prediction [CDL+19a, CST21], pollution propagation pattern discovery [LCCC17], and sensor calibration function transfer [CHZT19a].

Prior urban air pollution map generation proposals [CLL+14b, CHZT20a, JLF14, MLX+20, ZLH13a] mainly model the *dispersion* of airborne pollutants from emission sources. Previous air quality analytics services only offer passive monitoring of pollution to raise awareness [CLL+14b, HSW+15, LCCC17, MLX+20, ZLH13a]. State-of-art methods do not characterize and model the *absorption* of pollutants due to water spraying.

1.4.2 Water Spraying Systems for Dust Control

Water spraying is widely adopted for dust control in factories and mines [Kis03, KSM14, TNHS+03] and its usage has recently been extended for particulate matter control in urban areas [dCMP+17, Yu14, LTZ14]. Del Corno et al. [dCMP+17] carry out an experiment of removing aerosols with the help of high-pressure water spray nozzle as they generate water droplets that are smaller in size compared to those from regular, low-pressure nozzles. The experiment was conducted in a transparent glass chamber of size $0.5m \times 1m \times 1.5m$, equipped with a high-pressure spray nozzle system. Yu et al. [Yu14] proposes a geoengineering scheme to reduce air pollution in the cities of China with water spray technology. The indoor experiment results show that the $PM_{2.5}$ concentration can be reduced significantly, the extent of which depends on the scavenging coefficients. However, the authors did not evaluate the spraying system in outdoor environment. Liu et al. [LTZ14] propose to use a sprinkling system along the roadside to mitigate $PM_{2.5}$ and PM_{10} concentrations. However, it is only a conceptual system without quantitative analysis and results.

From the above reviews of related works, we can find that: *(i)* The current works on water spraying systems for dust control still focus on single location (e.g., pollution sources) or indoor evaluations. How to characterize the pollution reduction in outdoor environment for multiple sprayer devices in still an unsolved research problem. *(ii)* Regarding the air pollution sensing research, current research are mainly about improving sensing data quality, generating air quality maps, doing spatial inference or temporal predictions, etc. However, how to improve the air quality with existing pollution control systems (e.g., spraying system) is still missing.

1.5 Thesis Outline and Contributions

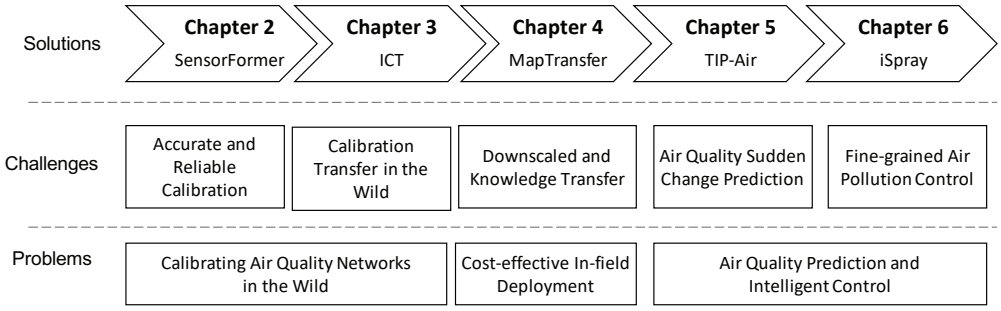


Figure 1.6: Overview of the problems, challenges and the proposed solutions discussed in each chapter of this thesis.

This thesis presents novel techniques and aims to push the boundaries towards accurate measurements and intelligent prediction and pollution control from low-cost air pollution sensors. To obtain a better understanding of the air quality environment we live in, we develop a new calibration model to calibrate the dense deployed low-cost sensors (Chapter 2) and transfer it safely to other locations (Chapter 3) to generate high-quality air quality maps. To reduce the long-term maintenance effort and cost for such large-scale dense deployment, we proposed new downscaled sensor deployment method (Chapter 4). Based on the more accurate and robust air quality dataset, we can make more accurate air quality predictions (Chapter 5), which can also explain the prediction results with the spatial-temporal causal flow diagram. Multi-stage air purification/control strategies could then be applied to improve air quality. For example, sprayers can be used to intelligently control air pollution in the local environment (Chapter 6) or to control pollution sources in the global environment, such as opening or closing factories, according to the spatial-temporal causal path graph (Chapter 5). The overview of this thesis and the specific topics are displayed in Figure 1.6. In summary, we aim to apply a viable data-driven framework and methods to predict and understand air pollution to control pollution with a smarter and more efficient approach.

In the following, we present the main contributions of each individual chapter.

Chapter 2: Efficient Sensor Array Calibration. In this chapter we tackle the accurate and reliable sensor calibration in the wild. As shown in Section 1.1.2, current approaches fail to include the *recent history* and *near future* sensor measurements. Also, state-of-art methods are still computation unfriendly, even when deployed to resource-constraint IOT devices. Motivated by these problems and challenges, we propose a generalized many-to-many calibration scheme called SensorFormer based on the successful Transformer model which takes both past

and future raw measurements into account. The procedure is able to *(ii)* significantly improve the calibration accuracy, *(ii)* boost the performance of compensating altered sensitivity, *(iii)* efficiently run on low-power microcontrollers with very limited computational and storage capabilities.

Chapter 3: In-field Calibration Transfer for Air Quality Sensor Deployments. To guarantee data accuracy and consistency, low-cost deployed sensors need periodic calibration after deployment. Since access to ground truth references is often limited in large-scale deployments, it is difficult to conduct city-wide post-deployment sensor calibration. In this chapter we propose In-field Calibration Transfer (ICT), a calibration scheme that transfers the calibration parameters of source sensors (with access to references) to target sensors (without access to references). Experiments show that ICT is able to calibrate the target sensors as if they had direct access to the references.

Chapter 4: Urban Air Quality Map Generation for Downscaled Sensor Deployments. This chapter focuses on cost-effective low-cost sensor deployment. 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.

Chapter 5: Tracking Pollution Transfer for Accurate Air Quality Prediction. Accurately predicting air quality, especially its sudden changes, is highly valuable for citizens and governments to make personal and local decisions, design intelligent policies and control pollution at minimal cost. However, none of the existing methods achieves sufficient prediction accuracy for time intervals of sudden pollution change due to inability of existing models to take into account pollution propagation between different areas caused by air mass movement. For the first time, we consider pollution transfer in the context of short-term air quality prediction and propose to use air flow trajectory data, widely used in

environmental sciences, to represent pollution transfer patterns between different locations. By learning trajectory representations, measurement location embedding vectors, and interrelationships between local weather at relevant locations, we propose a new attention based seq2seq model to track pollution propagation for accurate air quality prediction. We evaluate our model on datasets from Beijing area and compare the results to several state-of-the-art baselines. Experiments show that the proposed approach can successfully capture pollution transfer patterns between different sites in the area.

Chapter 6: Reducing Urban Air Pollution with Intelligent Water Spraying. In the last chapter we present a way to control the fine-grained air pollution. Despite regulations and policies to improve city-level air quality in the long run, there lack precise control measures to protect critical urban spots from heavy air pollution. In this chapter, we propose iSpray, the first-of-its-kind data analytics engine for fine-grained $PM_{2.5}$ and PM_{10} control at key urban areas via cost-effective water spraying. iSpray combines domain knowledge with machine learning to profile and model how water spraying affects $PM_{2.5}$ and PM_{10} concentrations in time and space. It also utilizes predictions of pollution propagation paths to schedule a minimal number of sprayers to keep the pollution concentrations at key spots under control. In-field evaluations show that compared with scheduling based on real-time pollution concentrations, iSpray reduces the total sprayer switch-on time significantly while assuring the good air quality levels.

2

Efficient Sensor Array Calibration

In this chapter, we introduce the basics of this thesis, namely how to improve the low-cost sensor quality with calibration methods. The calibrated sensor measurements will be further analyzed and used in subsequent tasks, such as air quality prediction and pollution sources detections.

Over the past several years, many low-cost air pollution sensors have been incorporated into platforms for measuring air quality. These densely deployed air quality sensors can provide valuable information about the underlying spatial and temporal pollution changing patterns. However, low-cost sensors usually suffer from inaccurate and unreliable measurements. In Section 1.1, we list various methods and schemes to calibrate the low-cost sensor readings, but the widely applied linear or non-linear methods only include the *current* measurements and are still computation unfriendly, even when deployed to resource-constraint IOT devices. These limitations pose a significant challenge to low-constraint pollution sensors in real-world applications.

In this chapter, we propose a novel sequence-to-sequence method to calibrate the low-cost sensor readings in the wild. The proposed algorithm is based on the Transformer model, which takes both *recent past* and *close future* raw measurements into account. We show that the proposed approach (1) outperforms other methods by improving calibration accuracy by 16.5 % to 20.4 % on public datasets and own field data, and (2) can efficiently run on low-power microcontrollers with very limited computational and storage capabilities. The latter is achieved by a novel optimization technique based on learnable input sub-sampling taking advantage of the properties of typical sensor data. We manage to reduce the model size by 20 % to 33 % and minimize the overall operation counts by 65 % while maintaining superior accuracy in comparison to state-of-the-art methods.

2.1 Introduction

Miniaturization of environmental sensors and their low power consumption have enabled numerous large-scale and high-resolution monitoring applications based on the Internet of Things (IoT) technology. Early warning scenarios, such as air quality assessment in homes and cities, monitoring of ammonia in agricultural fields, gas leakage detection, benefit from availability of low-cost networked sensors. The value of these systems, however, heavily depends on the quality of measured data and the ability of the system to extract useful information about the surrounding context. Since low-cost sensors often have reduced resolution, suffer from sensor noise, baseline and sensitivity drifts, environmental dependencies and other cross-sensitivities in the target environment, in-field sensor calibration methods gained popularity in recent years [CLL+14c, LDC18a, YLG+20], to improve the quality of measured data. For example, measuring gaseous pollutants in ambient air with low-cost technology requires periodic sensor calibration [MZT18b] or calibration transfer methods [CHZT19b].

Most state-of-the-art sensor calibration methods, ranging from linear regression [MSHT16c] to neural networks [CLL+14c], operate on an array of feature vectors measured at a single time step to predict a calibrated sensor value for the same time step (thus, one-to-one calibration). Recent methods also include feature vectors measured in the *recent past* to better capture the changing patterns of the measured phenomena and the temporal dynamics of the sensitive material of a low-cost sensor [MZT18b] (referred to as many-to-one calibration). These methods, such as AirNet [YLG+20] yield best state-of-the-art calibration performance. The schematic representation of one-to-one and many-to-one calibration schemes can be found in Figure 2.1-(a). More details can be found in Section 2.3

In this chapter, we take sensor calibration to the next level by calibrating low-cost measurements over multiple time steps in one inference pass through a many-to-many calibration model as shown in Figure 2.1-(a). The approach supports a gradual calibration refinement over time by taking future process development into account. This approach, however, faces several technical challenges.

Challenges. While calibration accuracy gains obtained by the inclusion of the *recent past* measurements into the calibration procedure seem obvious, the dependency of the correct calibration on the *close future* measurements is not. Low-cost sensors often suffer from slow response times [MZT18b], making future knowledge valuable to compensate for their delayed response. We include a motivating example in Section 2.3 to support the intuition. Being able to leverage future measurements, however, leads to a *delayed* availability of accurate calibrated outputs, which may not be tolerated in early-warning and disaster surveillance

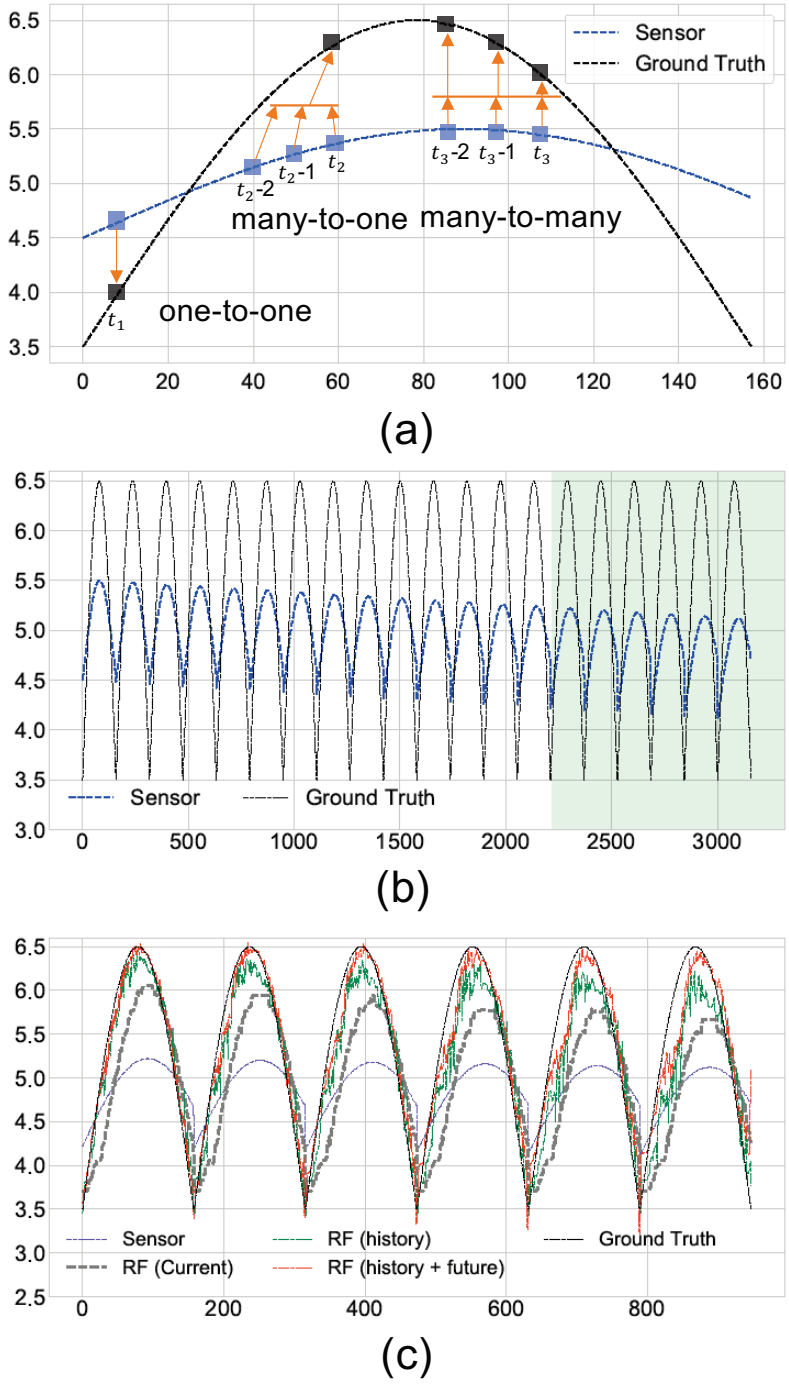


Figure 2.1: Motivation example: Why past and future data helps to achieve accurate calibration? (a) Schematic representation of different calibration schemes. (b) Synthetic data for testing sensor calibration methods; shaded green zone denotes the test set. (c) Results obtained by different calibration methods on synthetic dataset. RF: Random Forest. See detailed analysis in Section 2.3.2.

systems. We solve the problem by providing immediate calibration with its gradual refinement as further measurements become available, leading to a many-to-many calibration approach. The main challenge of implementing our method on resource-constrained devices is its high computation and storage overhead, which may hinder practical use of the proposed method despite its superior accuracy. We propose to tackle the challenge at multiple levels and show that the solution is implementable on a Cortex-M4 microcontroller with only 256 kB of RAM, while the optimized model still outperforms other methods.

Contributions and road-map. We are the first to propose many-to-many calibration, which shows significant benefits over existing one-to-one and many-to-one calibration methods with up to 20.4 % accuracy improvement, as shown on synthetic and real data in Section 2.3 and Section 2.6 respectively. Our many-to-many calibration method, called SensorFormer is based on the Transformer model discussed in Section 2.4, which has proven its efficiency in sequence-to-sequence prediction tasks. We leverage the fact that sensor data is often oversampled and propose a novel optimization procedure tailored to environmental sensor data to significantly reduce computation and storage overhead of the multi-head attention mechanism, which is the most resource-intensive block of Transformer models. Our optimized SensorFormer Lite model is detailed in Section 2.5. It reduces the model size by up to 33 %, requires 65 % fewer FLOPs to run and consumes only 38.8 mJ of energy on Arduino Nano 33 BLE Sense to compute a prediction. In Section 2.6 we show the benefits of SensorFormer and SensorFormer Lite on two datasets comprising PM2.5, PM10 [1] and ozone measurements gathered by low-cost IoT sensors. Section 2.2 summarizes related literature and Section 2.7 concludes this chapter. The source code is available at: *https://github.com/CalibrationMe/SensorFormer*

2.2 Related Work

As illustrated in Section 1.1, accurate *sensor calibration* is crucial for environmental sensors to capture relevant state information about the environment [MZT18b]. State-of-art models apply one-to-one or many-to-one calibration schemes to improve the sensor data quality. To the best of our knowledge, we are the first to propose a family of *many-to-many* sensor calibration methods and show their superior performance over existing works. This work touches upon two topics that have recently enjoyed significant progress in terms of algorithm development and their applications to solving real-world problems. (1) Sequence-to-sequence models gained popularity in processing time series data.

[1] Particulate matter with diameter less than 10 and 2.5 micrometer are denoted as PM10 and PM2.5, repectively.

In particular, Transformers emerged as a successful model architecture providing accuracy gains. Their high computational complexity is now in focus of the research community [RSVG21]. (2) *optimizing machine learning models for microcontrollers* gave rise to a tighter integration of hardware and software for specific tasks, and boosted the development of software frameworks and methods providing significant memory footprint reductions, energy savings and speed-ups [Goo]. This section puts our work in the context of these recent developments.

Transformers and Optimizations. Transformer [LWLQ21] is a prominent deep learning model that was originally proposed as a sequence-to-sequence model for machine translation. Due to its effectiveness over existing methods, it has been adopted in computer vision [CMS+20], audio processing [CWW+21] and even boosted progress in other disciplines, such as chemistry [SLG+19] and life sciences [RMS+21]. In the context of time series analysis, Transformer has been used for prediction tasks [ZZP+21], anomaly detection [CCZ+21] and domain adaptation [YLXH21].

To support efficient implementation of Transformer based models, in particular for its operation in resource-constrained environments, modeling advances and architectural innovations are proposed to tackle the computational complexity issue of the self-attention mechanism [TDBM20]. State-of-the-art efficient Transformer implementations approximate the quadratic cost attention matrix by applying some notion of sparsity to the otherwise dense attention mechanism. For example, limiting the field of view of attention results in local attention [PVU+18] or in a dilated window attention [BPC20]. Related literature further proposes to combine the distinct access patterns mentioned above. For example, Sparse Transformer [CGRS19] combines strides with local attention. In contrast to fixed attention patterns, researchers have recently proposed learnable sparse attention [KKL20, RSVG21], i.e., the attention access patterns are learned by a data-driven end-to-end approach to further reduce computational overhead. We refer an interested reader to a detailed survey on efficient transformers [TDBM20].

This work introduces Transformer optimizations based on information redundancy in measured sensor data to minimize the overall size of the self-attention matrix. The method is orthogonal to existing techniques.

Optimizing Machine Learning Models. A number of libraries, algorithms and tools have been developed to support ML on resource-constrained platforms. One example is TensorFlow Lite [Goo19], which includes a set of tools that enables on-device efficient machine learning. Various optimizations in TensorFlow Lite can be applied to models so that they can run within limited memory or computational power constraints of edge IoT devices. The effectiveness comes from the fact that deep networks are highly redundant and their model size can

be reduced without decreasing model accuracy. This motivated many network compression techniques and search for efficient subnetworks. For example, quantization and binarization rely on weights with discrete values, e.g., used by [WLe16] to quantize filters of a CNN. Decomposition and factorization explore low-rank basis of filters [GNGA20] to reduce model size and achieve inference speed-up.

This work proposes an optimization technique based on signal subsampling which is orthogonal to the existing optimization methods provided by TensorFlow Lite.

2.3 Problem Definition and Analysis

This section formulates the sensor calibration problem, highlights the gap in the existing literature addressed in this chapter, and presents analysis of the problem on synthetic data.

2.3.1 Sensor Calibration Schemes

Consider a low-cost sensor is co-located with a reference station. Measured data from both devices is collected over a period of time T. Low-cost sensor values are denoted as a time series $X \in \mathbb{R}^{|T| \times d}$, where $|T|$ is the length of the time series and d is the dimension of the input vectors measured at each time step. We further denote reference measurements $Y \in \mathbb{R}^{|T| \times 1}$ as a sequence of scalar values measured over T by an accurate and reliable reference sensor. We treat reference measurements as *ground truth*. Let $x_t \in X$ and $y_t \in Y$ represent a low-cost sensor measurement and a ground truth value at a time step t. The goal of *sensor calibration* is to learn a *calibration model* $C_\theta : \mathbb{R}^{p \times d} \to \mathbb{R}^{q \times 1}$ with input and output lengths p and q, and learnable parameters θ such that the distance between the calibrated values $\hat{Y} = C_\theta(X)$ and the ground truth measurements Y is minimized.

The most common calibration scheme is one-to-one [MZT18b] for which at each time step a function $C_\theta(x_t)$ maps a single raw measurement x_t to its calibrated value ($p = q = 1$). To obtain calibrated values $\{C_\theta(x_t)\}$ from raw measurements $\{x_t\}$ during a time period T one iterates over time steps $t \in T$.

We propose to include uncalibrated low-cost sensor measurements x_t over the past $\langle t - \tau_1, t \rangle$ or future $\langle t, t + \tau_2 \rangle$ time intervals as input to the calibration model C_θ. The intuition comes from the observation that the changing rates of multi-dimensional input features contained in the *recent past* and *close future* values is valuable to accurately calibrate the present measurement. many-to-one calibration schemes take the past low-cost sensor measurements as model input ($p > 1$) to produce a single calibrated output value ($q = 1$) at each time step t. In contrast, many-to-many methods calibrate multiple values within a window in one time

Table 2.1: Summary of calibration schemes. C_θ is a calibration function with learnable parameters θ.

	Objective function $\text{argmin}_\theta(\cdot)$
one-to-one	$\sum_{t=1}^{T} \mathcal{L}(y_t, C_\theta(x_t))$
many-to-one	$\sum_{t=1}^{T} \mathcal{L}(y_t, C_\theta(x_{\langle t-\tau_1, t\rangle}))$
many-to-many	$\sum_{t=1}^{T} \mathcal{L}(y_{\langle t-\tau_1, t+\tau_2\rangle}, C_\theta(x_{\langle t-\tau_1, t+\tau_2\rangle}))$

step ($p > 1, q > 1$). The objective functions of different calibration schemes are summarized in Table 2.1 and visualized in Figure 2.1-(a). many-to-many methods produce multiple consecutive calibrated values in each time step. The most accurate calibration result is achieved when future inputs $\langle t, t + \tau_2\rangle$ with respect to the target calibrated value at time t are known. This, however, introduces a *calibration delay* τ_2. If only past sensor measurements $\langle t - \tau_1, t\rangle$ are used to calibrate the current value, many-to-many schemes allow for a *real-time* or *instant* calibration at a price of a possibly reduced accuracy.

2.3.2 Calibration Analysis on Synthetically Generated Data

We use a synthetically generated dataset to validate the effectiveness of taking both past and future *raw* sensor measurements into account when learning a calibration model. The detailed data generation and experiment results can be found in the source code.

Setup. Synthetic data, comprising raw measurements $\{x(t)\}$ and reference values $\{y(t)\}$, is sampled equidistantly within the time interval $t \sim \mathcal{U}(0, 20\pi)$ as follows:

$$x(t) = 0.02 \cdot \sin(t \cdot (0.99 - 0.01\left\lfloor \frac{t}{\pi} \right\rfloor)) + 4.5 - \left\lfloor \frac{t}{\pi} \right\rfloor + \delta \tag{2.1}$$

$$y(t) = 3 \cdot \sin(t - \pi\left\lfloor \frac{t}{\pi} \right\rfloor) + 3.5 \tag{2.2}$$

where $\lfloor\cdot\rfloor$ is the floor operator. Reference data $y(t)$ is a periodic function with the period π.

The phase difference $(0.99 - 0.01\left\lfloor \frac{t}{\pi} \right\rfloor)$ over time t is used to simulate *sensitivity drift* of a low-cost sensor over time. The amplitude and bias difference between the two functions $x(t)$ and $y(t)$ capture other error sources such as environmental dependencies and cross-sensitivities to other gases. An extra noise term $\delta \sim \mathcal{N}(0, 1)$ is added to simulate the random noise of the low-cost sensor. Additionally, we introduce a response time delay by setting $x(t') = x(t + 2)$, i.e., low-cost sensor responses 2 time steps later to the signal. The generated dataset comprises 3138 samples, where the first 70% is used for training a calibration model

Table 2.2: Performance evaluation of different calibration schemes on a synthetically generated dataset. Taking both current, past and future values as input yields best accuracy.

	No calibration	Random Forest		
		[current value]	+[past]	+[past, future]
MAE	0.804	0.650	0.295	**0.195**
RMSE	0.841	0.570	0.159	**0.066**

and the remaining 30% for testing its quality. The synthetic dataset and a train-test split are shown in Figure 2.1-(b). The goal is to accurately calibrate $\{x(t)\}$ during the test period and evaluate accuracy gains due to the inclusion of the recent past and close future raw sensor measurements into the model input. We adopt root mean square error (RMSE) and mean absolute error (MAE) as accuracy measures.

Results. We chose Random Forest (RF) regression as a sample model in this study due to its simplicity and prior successful use in sensor calibration [LDC18a, MZT18b]. Its performance on synthetic data is shown in Figure 2.1-(c) and summarized in Table 2.2. We observe that MAE of uncalibrated sensor data is high (0.804). MAE can be reduced to 0.650 if the current time value is used as input to calibration, i.e., one-to-one approach. However, an artificially introduced sensitivity drift causes large errors on future test data. This large error is caused by the fact that one-to-one approach only takes current values into account and fails to capture the sequence alignment pattern, which is crucial for compensating sensor drift and improving calibration accuracy [YLG+20].

The calibration error gets dramatically reduced if both current value and two recent values are input to the RF model, i.e., many-to-one approach. MAE drops down to 0.295 providing over 55.0% reduction compared to using the current value only. The error can be further reduced to 0.195 (70.0% reduction compared to the same baseline as before) if both two past and two future raw measurements are input to the model. Results presented in Figure 2.1-(c) suggest that augmenting current measurements with both recent past and close future data better addresses the simulated sensor drift and delayed response issues and achieves accurate calibration.

Intuition. Synthetic data analysis lets us conclude: *(i)* Recent history greatly helps to learn an accurate calibration function, since consecutive past values contain information about temporal characteristics of a low-cost sensor and the measured process, e.g., the slope of the sensor drift and the speed of the process change. *(ii)* Including future data helps to further improve calibration accuracy, especially for the peak areas. This shows that the future signal dynamics can be used to improve calibration. We refer to this setting as to *delayed calibration*. To achieve *real-*

time calibration, the model has to produce multiple outputs for the same input while providing improved calibration as the time progresses and later measurements become available. The above observations motivate a calibration model structure with both multiple inputs and multiple outputs, i.e., a many-to-many calibration model, detailed in the next section.

2.4 Many-to-Many Sensor Calibration

This section first provides a preliminary background on the key modules of Transformer structure, then follows an overview of SensorFormer, our proposed many-to-many sensor calibration method.

2.4.1 Preliminary: Multi-Head Self Attention

Transformer utilizes a multi-headed self attention (MSA) mechanism to learn an alignment in which each element in the sequence learns to gather information from other tokens in the sequence. Using this mechanism, we can compensate sensor information, i.e., the hidden representations of the Transformer, from both the past and the future to calibriate the sensor values. Specifically, given a sequence input $\boldsymbol{X} \in \mathbb{R}^{|\tau| \times d}$, where $|\tau|$ denotes the length of the sequence and d is the feature dimensions. The operation for a single head is denoted as:

$$A_h = \text{Softmax}\left(\alpha Q_h K_h^\top\right) V_h \tag{2.3}$$

where $Q_h = \boldsymbol{W}_q X, K_h = \boldsymbol{W}_k X$ and $V_h = \boldsymbol{W}_v X$ are linear transformations applied on original input X with learnable weights $\boldsymbol{W}_q, \boldsymbol{W}_k$ *and* $\boldsymbol{W}_v$. α is a scaling factor that is typically set to $1/\sqrt{d}$ to alleviate gradient vanishing problem of the softmax function. In practice, multi-heads, $A_1, A_2, \ldots A_H$, are used to derive the underlying complex relations among sequence data, and the outputs of heads are concatenated together to produce the final attention weights. More details can be found in [TDBM20, LWLQ21].

2.4.2 SensorFormer

The architecture of the proposed SensorFormer model is shown in Figure 2.2. SensorFormer takes sequential sensor readings over a time window τ as input and produces calibrated values in each time step. Similarly to the study on synthetic data, we assume that input to the model is $X_\tau \in \mathbb{R}^{|\tau| \times d}$, where $|\tau|$ represents the sequence length and d denotes the dimension of the input vector in each step. The output of the model is $\hat{Y}_\tau \in \mathbb{R}^{|\tau| \times 1}$, where a scalar value is used as the ground truth in each time step.

Similarly to the usage of Transformers in natural language processing (NLP) [LWLQ21] or computer vision [DBK+20] domains, we propose to

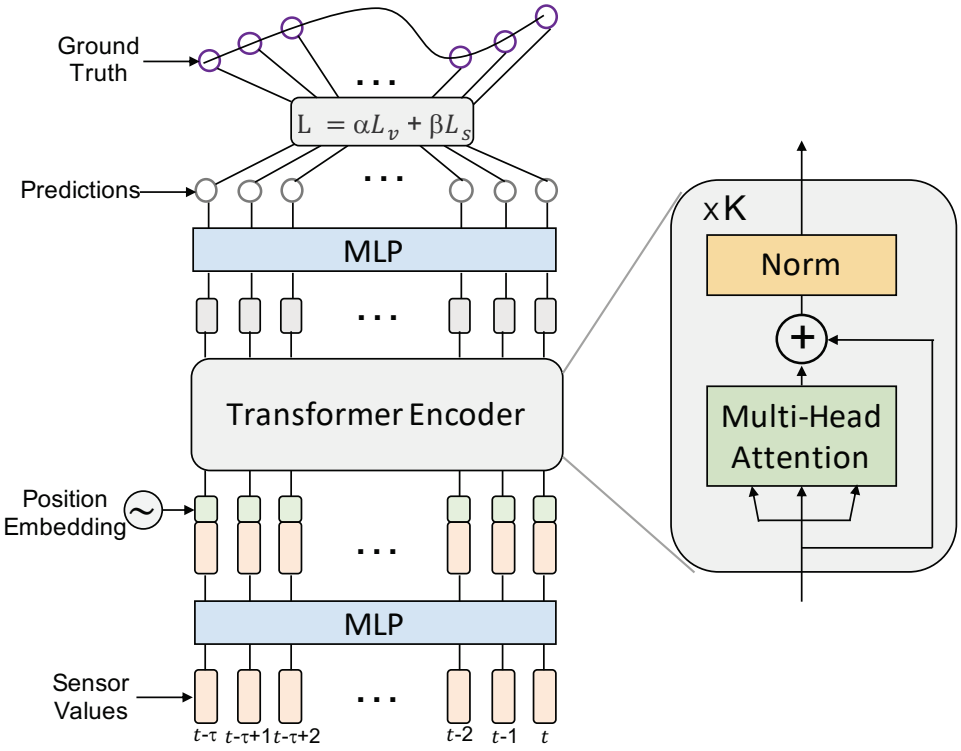
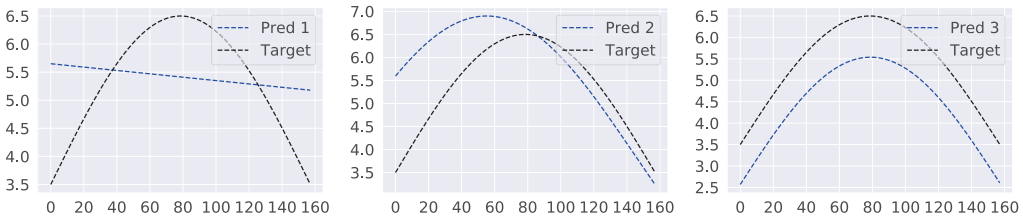


Figure 2.2: SensorFormer architecture. Our many-to-many calibration model based on the Transformer architecture.

use a learnable embedding module to map the original sensor features to a hidden representation. In our implementation, a multilayer perceptron (MLP) is chosen to embed the original input X_τ to its embedding $\mathbf{E} \in \mathbb{R}^{|\tau| \times s}$, where s is the dimension of the hidden embedding.

The Position Embedding module is used to retain positional information in the sequential input. We use the normalized absolute positional encoding $\mathbf{E_{pos}}$ in our model, since more complex methods show no benefit on our tested datasets.

The Transformer Encoder block computes a representation of all inputs. It consists of K blocks connected in a sequence. The computation proceeds as follows. First, a concatenation of $\mathbf{E}$ and $\mathbf{E_{pos}}$ is used as construct the input $\mathbf{z}_0$. Then, for a block $k \in 1..K$, the input first goes to the Multi-head Self Attention (MSA) module to learn an alignment of each element in the sequence with respect to other elements in the sequence [LWLQ21]. The residual connections follow the MSA block to learn deeper networks. Batch norm (BL) is applied at the end of each Transformer Encoder block to stabilize the training process. The overall



(a) MSE=0.888 (b) MSE=0.888 (c) MSE=0.888

Figure 2.3: Limitation of the MSE loss. The three predictions (a), (b) and (c) have similar MSE but quite different shape.

equations are as follows:

$$\begin{aligned} \mathbf{z}_0 &= [\mathbf{E}; \mathbf{E}_{pos}], & \mathbf{z}_0 &\in \mathbb{R}^{\tau \times s} \\ \mathbf{z}'_k &= \mathrm{MSA}\left(\mathbf{z}_{k-1}\right) + \mathbf{z}_{k-1}, & k &= 1..K \\ \mathbf{z}_k &= \mathrm{BN}\left(\mathbf{z}'_k\right), & k &= 1..K \end{aligned} \tag{2.4}$$

Following the last block of Transformer Encoder, the output $\mathbf{z}_K$ is used as input to the MLP layer to produce the final predictions $\hat{Y}_\tau$, that represent calibrated values computed by SensorFormer over all τ steps.

2.4.3 Loss Function

It is critical and challenging to define an effective loss function in multi-output regression problems. A widely used approach is based on the value difference, i.e., average error within each time window τ. Given a series of reference measurements Y_τ, the mean squared error (MSE) loss is defined as $\mathcal{L}_v = (1/|\tau|) \times \sum_{i=1}^{|\tau|} (Y_i - \hat{Y}_i)^2$.

However, relying on MSE may be inappropriate in our situation, as illustrated in Figure 2.3. Here, the target ground truth is shown in black line, and we present three predictions, shown in Figure 2.3-(a), (b) and (c), which share a similar MSE loss compared to the target, but quite different prediction shape. Prediction (a) fails to capture the overall shape of the target value. The predictions (b) and (c) reflect the real change of regime much better, as the changing shape is actually predicted, but with a slight advance in time (b) or with a slight error in magnitude (c).

A sudden change of the process of interest presents the most challenging scenario for accurate sensor calibration. Correctly capturing the shape of the change is thus an important task of a calibration algorithm. Motivated by the above example that time series with quite different shapes may share similar MSE, the loss between time series shape is now a hot research topic for multi-output regression problems. For example, [GT19] propose a *differentiable* loss function based on Dynamic Time Warping (DTW) to align different time series. We adopt the idea to compute the shape distortion between $\hat{Y}$ and Y and denote the new

shape loss as $\mathcal{L}_w$. This new shape loss between group truth and predicted time series is used to penalize those unsimilar shape predictions. With this novel loss constraint provided by $\mathcal{L}_w$, the many-to-many calibration model tends to generate prediction similar to Figure 2.3-(b), (c) instead of (a). A successful multi-output loss function should be the combination of both MSE loss $\mathcal{L}_v$ and the time series shape loss $\mathcal{L}_w$. Thus, the overall loss of SensorFormer is defined as follows.

$$\mathcal{L}_{\text{SF}} = \alpha\mathcal{L}_v + \beta\mathcal{L}_w \tag{2.5}$$

where α and β balance the importance between distance-based and shape-based loss functions.

2.5 SensorFormer Lite for Low-Power Microcontrollers

This section first gives an intuition what makes SensorFormer optimization possible. We then present the SensorFormer Lite architecture and discuss how its efficiency can be further improved when our optimization is coupled with standard optimization tools.

2.5.1 Learnable Sub-Sampling

The computation and memory bottleneck of the Transformer models mainly comes from the MSA module, since its computational complexity is $O(\tau^3)$, where τ represents the length of the input sequence [LWLQ21]. For example, the Transformer Encoder block of our method (see Section 2.4), whose main part is the MSA module, is responsible for over 91.96 % of total model's parameters and 96.30 % of multiply–accumulate (MAC) operations. The SensorFormer usability on resource-constrained devices thus depends on our ability to reduce the computational and storage complexity of the MSA module.

In this work, we propose to downsample the input sequence and to considerably decrease the input dimension τ. The key intuition comes from the observation that, in contrast to the problems in NLP and computer vision domains, time series measured by IoT sensors, as visualized on the bottom of Figure 2.4, are often oversampled to comply to the Nyquist sampling theorem [Vai01]. Sensor noise can be reduced by post-processing. Moreover, it is possible to represent changing patterns in time series using downsampled points, as shown on the upper of Figure 2.4. We will show in the evaluation section that those changing patterns rather than the raw time series values are more useful for sensor array calibration task. This motivates us to downsample a sub-sequence to represent the original time series data before using it in later blocks.

There are multiple ways to choose a sub-sampling strategy, such as random or uniform sub-sampling of the original input sequence, or by

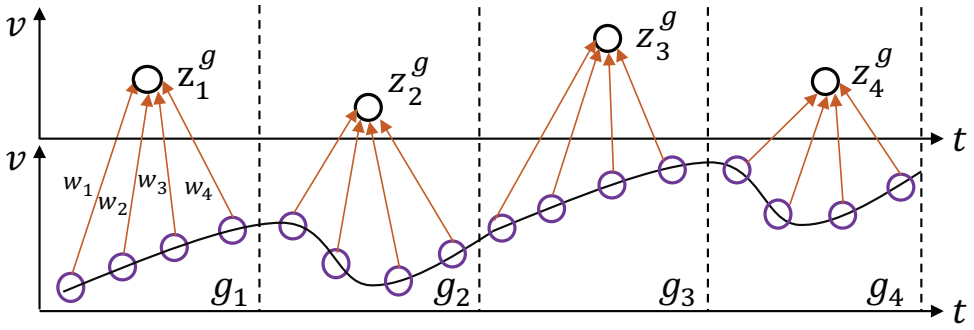


Figure 2.4: Group-wise input sampling. Input downsampling by group splitting and learning weighted embedding parameters.

using the mean of the adjacent values. These naïve and statistically inspired sub-sampling strategies yield poor calibration results, as argued in Section 2.6. We therefore propose a data-driven input sub-sampling method trained end-to-end simultaneously with other blocks of SensorFormer. We refer to the SensorFormer model with the proposed sub-sampling optimization as *SensorFormer Lite*. The design choices are detailed in the following subsection.

2.5.2 SensorFormer Lite

The architecture of the SensorFormer Lite model is shown in Figure 2.5. It differs from SensorFormer in the following three aspects: (1) A differentiable downsampling module, denoted as Group-wise Input Sampling (GIS), is added to decrease the length of the input τ to the Transformer Encoder block. (2) Following Transformer Encoder, a Group-wise Attention Sharing (GAS) module is added to share the attention weights among the members of the same group. It uses a simple but effective "copy-and-paste" approach and introduces no extra parameters. (3) Finally, the loss function used to train SensorFormer Lite includes a regularization of the learnable parameters in GIS. The details are explained below.

Given the input sequence data $X \in \mathbb{R}^{\tau \times d}$ and using the same pipeline as described in Section 2.4, we obtain hidden embeddings with positional information as $\mathbf{z}_0 \in \mathbb{R}^{\tau \times D}$. Instead of using $\mathbf{z}_0$ directly as input to K Transformer Encoder blocks, the input goes to the GIS block for downsampling performed in two steps by group splitting and subsequently computing a weighted embedding.

Group Splitting. We evenly divide each input sequence τ into n_g adjoint subgroups, each with $\lceil \tau / n_g \rceil$ members. n_g is a hyperparameter. For example in Figure 2.4, $\tau = 16$, $n_g = 4$, and the original data is divided into 4 subgroups g_1 through g_4.

Weighted Embedding. The downsampled measurement z_i^g is a linear

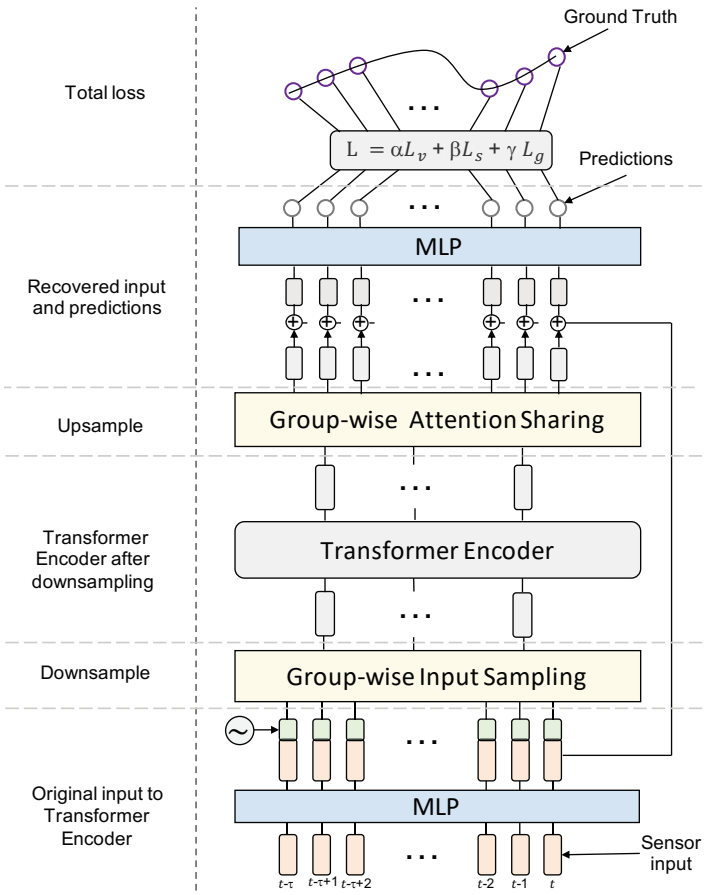


Figure 2.5: SensorFormer Lite architecture. The model learns an input sub-sampling strategy via Group-wise Input Sample (GIS) and Group-wise Attention Sharing (GAS) blocks.

combination of the samples in each group:

$$z_i^g = \sum_{j \in g_i} w_j z_j, \tag{2.6}$$

where $z_j \in \mathbb{R}^D$ denotes the j-th instance of $\mathbf{z}_0 \in \mathbb{R}^{\tau \times D}$, $j \in [1, \tau]$, w_j represents the weight to z_j, and z_i^g is a weighted embedding of all z_j comprising the group g_i.

We use the mean m_{g_i} over all instances in the group g_i to initialize the representation of the group before training. For an instance z_j in the group g_i, its distance to m_{g_i} is calculated as $d_j = \|z_j - m_{g_i}\|_2$, which denotes the relative importance of z_j in the group representation. Following [LMA20], we scale d_j with a learnable temperature coefficient t and compute the importance distribution in each group as follows

$$w_j = \frac{e^{-d_j^2/t^2}}{\sum_{s \in \mathcal{S}(g_i)} e^{-d_s^2/t^2}}. \tag{2.7}$$

Following the GIS module, the original $\mathbf{z}_0$ is mapped to the group-wise downsampled $\mathbf{z}_0^g \in \mathbb{R}^{n_g \times D}$, where $n_g < \tau$. For example, in Figure 2.4 the weighted embedding of each subgroup is used as its downsampled representation. Downsampled data still retain the changing pattern characteristics of the original time series data.

The downsampled $\mathbf{z}_0^g$ is input to the Transformer Encoder block. The latter yields the output $z_K \in \mathbb{R}^{n_g \times H}$ given by Eq. (2.4). GAS module shares the same hidden attention representation among all members of the group. The output z_K is then mapped to $z_K^g \in \mathbb{R}^{\tau \times H}$. A residual connection is added to retain the original information of each single input. The final prediction values are computed as follows:

$$\hat{Y} = \text{MLP}(z_K^g + z_0) \tag{2.8}$$

Loss Function. In the weighted embedding of our new GIS module, the weight w_j can be viewed as a probability distribution function over the points z_j. The temperature coefficient t controls the shape of this distribution. To regularize this new differentiable GIS module, we add a new loss term as $\mathcal{L}_g = t^2$. The loss function of SensorFormer Lite consists of three terms: MSE loss L_v, time series shape loss L_w and the new Group-wise Input Sampling regularization loss L_g. The overall loss is a balanced combination of all of them:

$$\mathcal{L}_{\text{SFL}} = \alpha \mathcal{L}_v + \beta \mathcal{L}_w + \gamma \mathcal{L}_g \tag{2.9}$$

where α, β and γ balance the importance between distance-based, shape-based and group regularization based loss functions.

2.5.3 Discussion

To the best of our knowledge, we are the first to leverage the properties of sensor data that allow downsampling the input sequence to reduce both computational and memory complexity of the critical MSA block. The proposed method is orthogonal to other Transformer optimization techniques and can be combined with the methods proposed in the literature. One research line on improving Transformer efficiency focuses on sparsifying attention matrix, e.g., by using approximation techniques. These can directly be added to the Transformer encoder in SensorFormer Lite. Another direction to compress deep learning models is to quantize their weights and reduce the model size for resource-constrained platforms. We empirically show that SensorFormer Lite can benefit from these compression techniques to further reduce the model's memory, compute and energy consumption overheads.

SensorFormer takes up relative large space and make it hard to be deployed on resource constraint IOT devices such as microcontrollers, especially when the calibration module is only one component of a

complex IOT system and limited space is allocated. In our scenario, we convert the trained model to TensorFlow Lite and quantizing its weights to reduce the model size, and only the sensing and calibration modules are tested.

2.6 Experimental Evaluation

This section discusses the performance results of the proposed models. We introduce two datasets and five baselines used to evaluate the accuracy achieved by SensorFormer and SensorFormer Lite. Then, we report the performance of the optimized SensorFormer Lite on Arduino Nano 33 BLE Sense featuring Contex-M4 microcontroller with 256 kB of RAM.

2.6.1 Datasets and Evaluation Metrics

We use two datasets collected in Beijing and Zurich to evaluate the proposed many-to-many calibration approach. At each deployed location, a low-cost air quality sensor is co-located with a governmental reference station providing ground truth data.

Beijing Dataset. The dataset comprises particulate matter measurements PM2.5 and PM10 (particles of diameter less than 2.5 and 10 microns, respectively) gathered at 7 locations in Beijing. Hourly measurements obtained with a low-cost Plantower sensor [pan] and ground truth data are collected between Mar 2018 and May 2019. The low-cost particle sensor is based on laser scattering technology. Those low-cost PM2.5 and PM10 sensors reports 7 low-level features, which are used to train a calibration model for accurate PM measurements.In each sample, we define a vector consisting of all 7 features as the input value, and a scale PM value from the reference station as the output we wish to align. In total, 60'450 samples are used in the experiments.

Zurich Dataset. A low-cost ozone (O3) sensor is deployed on the rooftop of a reference station. The sensor is MiCS-OZ-47 [et] based on the metal oxide semiconductor sensitive material, known to exhibit baseline and sensitivity drifts over time [MZT18b]. The low-cost sensor is located next to the air intakes of the highly accurate devices. In this way, low-cost sensors and reference devices can be measured at the same time. Low-cost ozone sensor samples are comprised of 3 features: ozone value, temperature, and humidity, and the scale value from the reference station is used as the ground truth. The data is collected hourly between Jan and Oct 2016 [MHS+19] and contains 5'180 samples.

We split all datasets into train, validation and test in chronological order at a ratio 6:2:2. Mean absolute error (MAE) is used to evaluate the accuracy of the proposed calibration models, while floating point operations per second (FLOPs) and the number of model parameters

(#param) are used to show resource-efficiency of SensorFormer Lite.

2.6.2 Benchmarks and Model details

We compare SensorFormer to the following state-of-the-art methods for sensor calibration described in the literature:

- **Naïve:** No calibration is performed. Raw values measured by a low-cost sensor are reported as calibrated values.
- **MLS [MSHT16c]:** Multiple linear regression is commonly used for sensor calibration, 7 and 3 input features are used as the independent variables for PM and ozone calibration scenario.
- **RF [MZT18b]:** Random Forest is widely used for sensor calibration due to its ability to learn non-linear functions.
- **MLS+RF [LDC18a]:** A method combines a linear model (MLS) and a non-linear model (RF).
- **AirNet [YLG+20]:** An RNN-based sequence model that leverages past data in the context of sensor array calibration.

In our experiments we fix the input sequence length $|\tau| = 12$ and the number of heads to 2. The number of hidden units in MLP blocks is set to 18. The number of transformer blocks K is 2 for PM2.5, and 1 for PM10 and O3 calibration. We use $\alpha = 0.15, \beta = 1, \gamma = 0.5$ as trade-off parameters of the loss functions. The model is written in Python and evaluated on Nvidia RTX 2080 Ti. Different sizes of the n_g groups are used to evaluate resource-efficiency of SensorFormer Lite, which is also evaluated on Arduino Nano 33 BLE Sense. More details can be found in the provided source code.

2.6.3 SensorFormer

Overall Performance. The results in Table 2.3 show that SensorFormer model outperforms other methods in all considered scenarios. State-of-the-art one-to-one schemes, i.e., fitting a linear (MLS [MSHT16c]), non-linear function (RF [MZT18b]) or a combination thereof (MLP+RF [LDC18a]), decrease calibration errors by learning an alignment between low-cost sensor features and ground truth labels. However as we illustrated in Section 2.3, by using only current time step features one-to-one methods are not capable of capturing time series changing patterns, and thus fail to achieve high calibration accuracy.

By contrast, many-to-one and many-to-many schemes solve the above challenge by using *many* sequential inputs to learn the underlying alignment function. The recently proposed AirNet [YLG+20] method

Table 2.3: Accuracy of different calibration methods. Performance results shown as MAE ($\mu g/m^3$). The proposed many-to-many models outperform other methods.

Model family	Model	PM2.5	PM10	O3
one-to-one	Naïve	31.25	37.68	6.63
	MLS [MSHT16c]	23.78	25.13	4.72
	RF [MZT18b]	21.34	24.61	4.61
	MLS+RF[LDC18a]	19.89	20.56	4.32
many-to-one	AirNet[YLG+20]	16.67	18.90	4.12
	SF-mo	15.11	17.18	3.82
many-to-many	SF-RT	13.86	15.05	3.44
	SF-W	**12.65**	**13.45**	**3.13**

yields significantly better results than one-to-one models. This shows that historical data contains valuable information for accurate sensor calibration. We denote the many-to-one version of our SensorFormer scheme, i.e., the version that uses past data to compute a single output (without $\mathcal{L}_w$ term in the loss function in Eq. (2.5)) as *SF-mo*. The results in Table 2.3 show that SF-mo, compared to AirNet, decreases MAE for PM2.5, PM10 and O3 by 9.36%, 9.10% and 7.28%, respectively. Compared to AirNet, SF-mo behaves better during peak periods as shown in Figure 2.6. The reason for this improvement is the power of the self-attention mechanism in SensorFormer over traditional RNN architectures used in AirNet.

When investigating the performance of the many-to-many method family, we first test SensorFormer for real-time calibration (denoted as *SF-RT*), i.e., only the first calibrated output in a sliding window is used even though multiple outputs are predicted. SF-RT thus includes $\mathcal{L}_w$ in its loss function in Eq. (2.5). The error achieved by SF-RT compared to AirNet is further decreased by 16.9 %, 20.4 % and 16.5 %, respectively. This highlights the effectiveness of SensorFormer when capturing patterns in time series data and learning the alignment between low-cost sensor values and the ground truth. If calibration latency can be tolerated, mean calibrated value over all time windows that include a given input is used. As a result, MAE can be further reduced by 24.1 %, 28.8 % and 24.0 % compared to AirNet for PM2.5, PM10 and O3, respectively. We refer to this approach as *SF-W*.

The calibration output for PM2.5 and O3 on test data is visualized in Figure 2.6. AirNet fails to capture the changing patterns, especially the peaks, and is not accurate enough for a practical use as discussed in [MZT18b]. SF-RT partly addresses the challenge while using only the first calibrated output, whereas SF-W achieves the best result by averaging

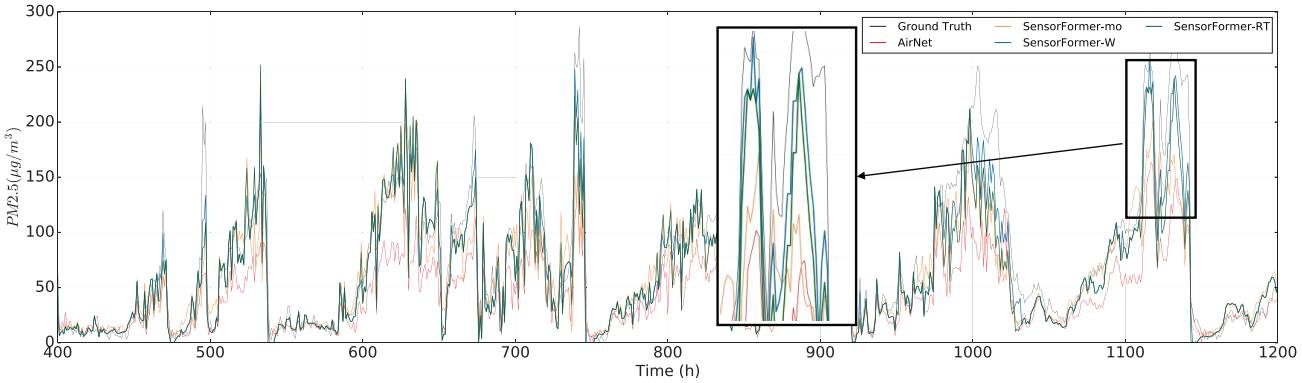
calibration outputs over multiple time windows for the same input. SF-W not only acquires the best calibration accuracy but captures the correct time series trend as the ground truth due to the constraint of our novel loss functions.

SensorFormer Analysis. We use an example to show the details of SensorFormer and argue why it outperforms state-of-art methods. Figure 2.7-(a) shows the calibration results for the input and output length $|\tau| = 12$. Shaded grey areas represent the variance of the calibration outputs over all sliding windows for a specific input. The solid line shows the mean of these calibration outputs, i.e., over 12 calibrated values. The grey area is narrow if air quality is stable or changes slowly, yet widens during the periods of sudden change of the measured process. Thus, large grey area can be interpreted as model's uncertainty in the correctness of the calibration result.

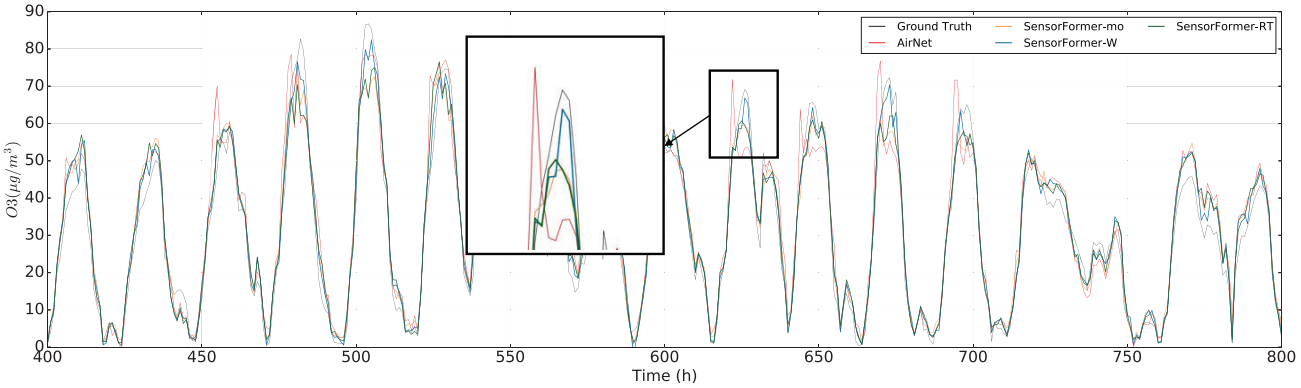
To better understand the model, we selected three time points A, B and C annotated in Figure 2.7-(a) and examine attention maps for the real-time calibration values for those points. Example A is located during a stable period, whereas B and C are chosen at peak points of signal variability. Figure 2.7-(b)-(d) show raw sensor input preceding the current time point. The 12 calibrated outputs from each calibration window forming the shaded area in Figure 2.7-(a) are shown in Figure 2.7-(e)-(g). The ground truth (black line) are located inside the variance zones. We plot attention maps learned by Transformer Encoder block 1 and head 1 to get an idea of the area the model is focusing on when processing the above sequential inputs. As shown in Figure 2.7-(h)-(j), the model pays attention to the changing period in all examples (see Figure 2.7-(b)-(d)). The model indeed learned to recognize that changing patterns are critical for sensor calibration. Figure 2.7-(h) assigns similar weights to all inputs, which indicates that the model relies on the mean from all inputs during a stable period. Another finding is that attention maps reveal group clusters, i.e., similar inputs share similar attention weights. This supports the intuition behind the proposed group-wise input downsampling and later upsampling optimization realized in SensorFormer Lite.

2.6.4 SensorFormer Lite

Overall Evaluation. The effect of subgroup sizes n_g on the number of floating point operations per second (FLOPs) and calibration error MAE is shown in Table 2.4. SensorFormer can be seen as an extreme case of SensorFormer Lite with $n_g = \tau$, i.e., 12 inputs in our case. We set n_g to 4 and 6 sequentially to evaluate the performance of the calibration models on different datasets. Compared to SensorFormer, SensorFormer Lite with $n_g = 6$ decreases the number of FLOPs by over 48 % at a cost of less than 3.2 % increased MAE by downsampling the original sequence by the

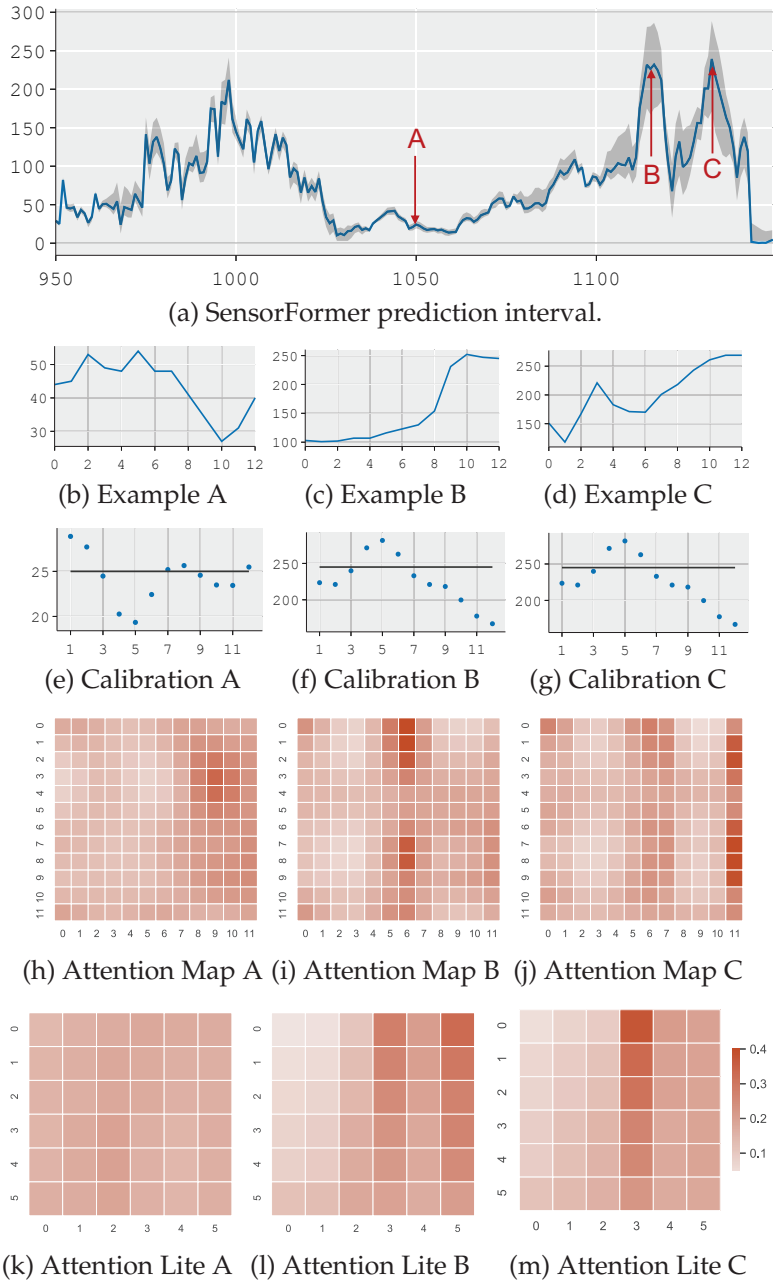


(a) PM2.5



(b) O3

Figure 2.6: Calibration results achieved by different methods on test data. AirNet fails to handle signal changes, especially the peaks; SF-mo and SF-RT partly addresses the challenge, whereas SF-W achieves the best performance.



(a) SensorFormer prediction interval.

(b) Example A (c) Example B (d) Example C

(e) Calibration A (f) Calibration B (g) Calibration C

(h) Attention Map A (i) Attention Map B (j) Attention Map C

(k) Attention Lite A (l) Attention Lite B (m) Attention Lite C

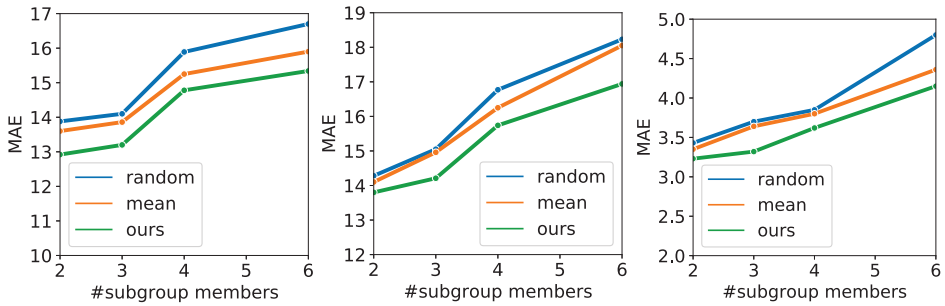
Figure 2.7: Example-based SensorFormer and SensorFormer Lite model analysis on PM2.5 dataset. Example A is located during a stable period; B and C are chosen at peak points shown in (a). (b)-(d) show model inputs for each example; (e)-(g) show the window-based calibration results, solid line is the ground truth; (h)-(j) depict attention maps of SensorFormer with a focus on variable parts of the input; (k)-(m) show attention maps of the optimized SensorFormer Lite model.

factor of 2. If the downsampling factor increases to 3 with $n_g = 4$, MAE increases by 6.1 %.

SensorFormer Lite Analysis. Using the same examples A, B and

Table 2.4: Performance evaluation of SensorFormer Lite. FLOPs (×1000), #param (×1000) and MAE ($\mu g/m^3$)

	Metric	SensorFormer $n_g = 12$	SensorFormer Lite $n_g = 6$	SensorFormer Lite $n_g = 4$
PM2.5	FLOPs	127.10	64.90 (-49%)	44.16 (-65%)
	#param	5.80	-	-
	MAE	12.65	12.92 (+2.1%)	13.20 (+4.3%)
PM10	FLOPs	64.75	33.64 (-48%)	23.28 (-64%)
	#param	5.80	-	-
	MAE	13.45	13.80 (+2.6%)	14.21 (+5.7%)
O3	FLOPs	13.42	33.56(-48%)	22.98(-65%)
	#param	1.42	-	-
	MAE	3.13	3.23(+3.2%)	3.32 (+6.1%)



(a) PM2.5 (b) PM10 (c) O3

Figure 2.8: Efficiency of weighted embedding. Learned weighted embedding outperforms mean and random baselines.

C shown in Figure 2.7-(a), we plot the SensorFormer Lite attention maps with $n_g = 6$ in Figure 2.7-(k)-(m). The downsampled attention maps by SensorFormer Lite share similar patterns with those learned by SensorFormer. Specifically, SensorFormer Lite attention map of example B captures the same changing patterns as SensorFormer, thus can be expected to yield similar calibration accuracy. SensorFormer Lite attention maps of examples A and C lack minor input signal variations, e.g., at the end of each sequence, due to a limited resolution of the attention mechanism. This explains minor accuracy drop reported in Table 2.4.

To illustrate effectiveness of the proposed weighted embedding method in SensorFormer Lite, we compare its performance to the following two baselines: *(i) randomly* selecting one sample to represent the group, and *(ii)* using *mean* value over all group members. Our method uses a weighted combination of group members as shown in Figure 2.4. The results in Figure 2.8 suggest that our method yields better results on all datasets. The effect is more pronounced for larger groups.

Table 2.5: Model generalization Comparison. $n_g = 4$ and performance results shown as MAE ($\mu g/m^3$).

	Naïve	AirNet	SensorFormer	SensorFormer Lite
PM2.5	42.1	23.5	15.0	15.8
PM10	49.8	30.1	17.8	18.4

2.6.5 Model Generalization Study

We design the following experiment to test the generalization ability of our proposed SensorFormer and SensorFormer Lite on unseen data in the distant future. A new PM2.5 and PM10 dataset was collected between Dec 2019 and Mar 2020 following the same setup described in Beijing Dataset. The data comes from a different spatial location and time period compared to the original Beijing Dataset. We evaluate the calibration performance of all related models trained with Beijing Dataset on the newly collected data.

The results in Table 2.5 show that Naïve MAE is higher (42.1 $\mu g/m^3$ and 49.8 $\mu g/m^3$ for PM2.5 and PM10) than previously reported due to a generally more significant and faster changing air pollution levels in winter 2020 in the new dataset than in the data from 2019. AirNet [YLG+20] reduces MAE by including history data. However, it fails to correctly capture the underlying changing patterns of low-cost sensors. The calibration error is too high for a real use case [CHZT19b].

SensorFormer acquires the best calibration performance by including both history and future readings to derive the changing patterns and learn the correct alignment function using self-attention mechanism. Our designed method tends to focus more on the critical time series changing patterns instead of only raw values, thus leads to a more stable calibration performance. SensorFormer Lite also acquires a reasonable accuracy while reducing computational overhead significantly.

2.6.6 Analysis of the Evaluation Results

The evaluation results on SensorFormer and SensorFormer Lite reveal that sensor data alignment task benefits from both historical and future data, and our proposed approach successfully capture their interactions. SensorFormer Lite acquires comparable accuracy due to ability to learn the input sequence changing patterns, which reflects that those changing patterns instead of the raw values play more important role in boosting the performance of sensor array alignment task. We can also validate this phenomenon from the attention maps in Figure 2.7, SensorFormer Lite successfully derives the sequence changing patterns as SensorFormer and generate accurate calibration results.

SensorFormer is preferred when the calibration accuracy is critical and energy is not a concern. Given the limited budget of usable energy,

Table 2.6: Further model optimization using standard methods. Achieved accuracy (MAE) after conversion to TensorFlow Lite and weight quantization. Percentage in brackets shows MAE increase relative to the model before optimization.

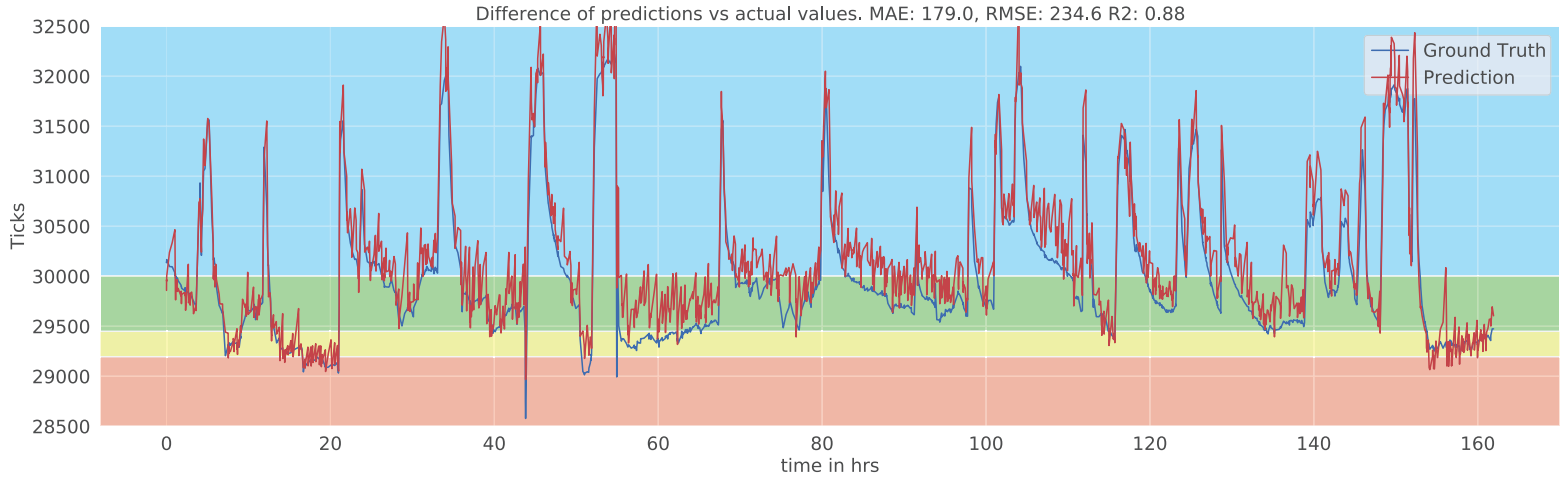
	Model	Model Size [kb] +TF Lite	+Quant.	MAE $[\mu g/m^3]$
PM2.5	SensorFormer	48	40	12.86 (+1.7 %)
	SensorFormer Lite	40	32	13.44 (+6.2 %)
PM10	SensorFormer	44	36	13.74 (+2.2 %)
	SensorFormer Lite	36	28	14.30 (+6.3 %)
O3	SensorFormer	28	24	3.19 (+1.9 %)
	SensorFormer Lite	20	16	3.34 (+6.7 %)

SensorFormer Lite is viewed as a more suitable choice as it balances both energy and acceptable accuracy.

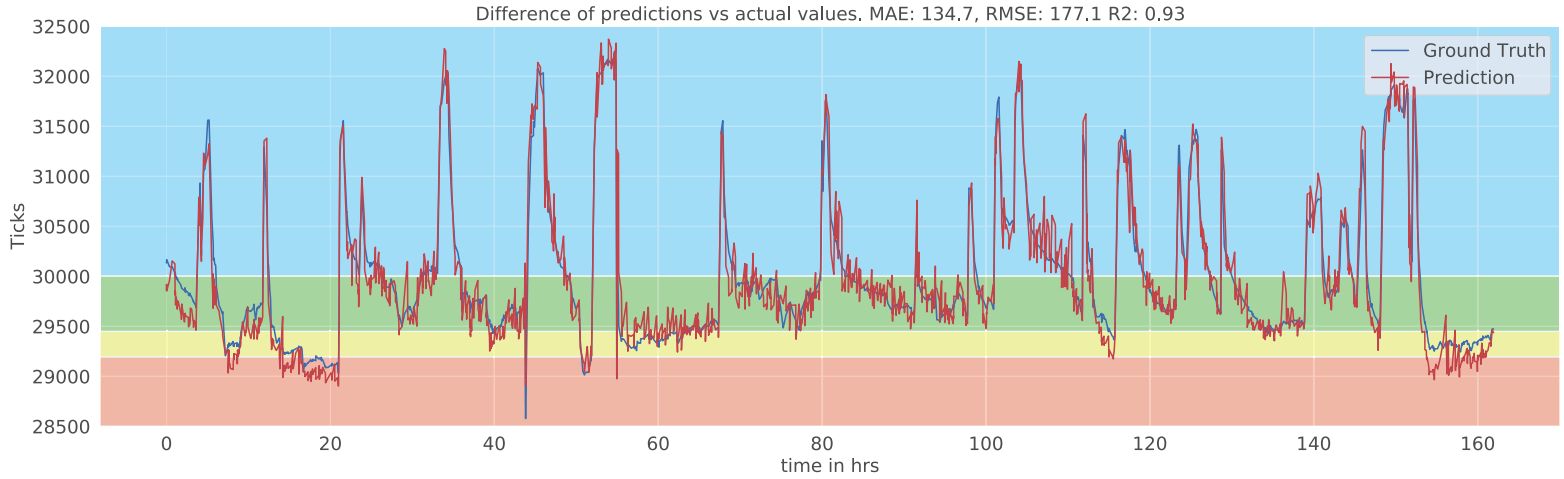
2.6.7 Further Model Optimization for IoT Devices

In this section, we evaluate the performance of SensorFormer and SensorFormer Lite ($n_g = 4$) in combination with other standard deep model optimization methods. By converting the trained model to TensorFlow Lite [Goo] and quantizing its weights we further optimize the model with insignificant performance decay. The results in Table 2.6 show that *(i)* optimized SensorFormer Lite has 20–33 % smaller model size than SensorFormer, and *(ii)* the performance of the optimized SensorFormer Lite drops by up to 6.7 % compared to results reported in Table 2.3. Nevertheless, the achieved accuracy is higher than achieved by the state-of-art methods.

SensorFormer Lite optimization reduces the number of FLOPs by up to 65 % and decreases the model size by up to 33 % for different datasets at a cost of up to 6.7 % decline in accuracy. Thus, the proposed SensorFormer Lite significantly reduces resource requirements on the target platform, and can now run on IoT devices. We deploy the proposed methods on Arduino Nano 33 BLE Sense featuring the Cortex-M4 microcontroller, and the power consumption for the calibration process was measured using the RocketLogger [SGL+16]. Taking PM2.5 as an example, each calibration operation consumes extra 110.4 mJ for SensorFormer and 38.8 mJ for SensorFormer Lite, while the sensing module itself consumes 302.5 mJ. Thus, each reported calibrated sensor reading requires 302.5 + 38.8 = 341.3 mJ energy with SensorFormer Lite, which means an error reduction of over 57.8 % compared to an uncalibrated reading with an overhead of only 38.8/(38.8 + 302.5) = 11.4 % of additionally consumed energy.



(a) GRU-based Method



(b) SensorFormer

Figure 2.9: Compensating Altered Sensitivity of Duty-Cycled MOX Gas Sensors with (a) GRU based method [GCGS21] and (b) our proposed SensorFormer.

2.6.8 Model Performance on Other Dataset and Task

To further validate the effectiveness of our proposed SensorFormer framework, one additional public MOX sensors dataset [2] is used. The task is to recover accurate continuous-sensor measurements from transient responses obtained from a duty cycled sensor and compensate for an altered multi-gas cross-sensitivity profile using machine learning methods. More details regarding the dataset and task can be found in [GCGS21].

The dimension of the input data is $\mathbb{R}^{12\times160}$, where 12 is the number of transient responses and 160 is the dimension of each transient response. The target is to recover the continuous sensor readings for each step. The state-of-art method proposed in [GCGS21] used a GRU-based encoder-decoder framework to generate the predictions and the result is shown in Figure 2.9-(a) with an overall MAE of 179.0. We can find that the prediction model behaves poorly during the changing periods. Compared to GRU-based models, our proposed SensorFormer decreases the MAE from 179.0 to 134.7, with a reduction of over 24.7%. SensorFormer also predict accurately during those changing periods, which makes it reliable for air quality analysis.

From the above evaluation results, we can safely conclude that SensorFormer based models are effective in capturing the relationships among sequence data from sensor readings and can generate accurate prediction results.

2.7 Summary

In this chapter, we show that including both recent past and close future raw sensor measurements in sensor calibration model improves model accuracy. The observation can be justified by slow response times of low-cost sensors, their cross-sensitivities and sensitivity drifts, that become apparent when future measurements are available. This motivates the design of a new family of many-to-many calibration methods, and its instance called SensorFormer proposed in this work. To reduce high resource consumption of the proposed method, due to redundant computations and a high overhead of the multi-head attention block, we propose a novel optimization technique based on signal sub-sampling, specifically tailored to often oversampled sensor data. The optimized SensorFormer Lite model is effective under resource constrains of a typical microcontroller, yet yields superior performance than state-of-the-art benchmarks. We believe the proposed algorithm can be an essential step in the design of the low-cost air quality sensor network.

Many-to-many calibration methods, such as those presented in this work, make the distinction between sensor calibration and short-term

[2] https://github.com/TUG-EIP/MOX-Compensation-SGP30

prediction vanish. One future direction is to investigate this intriguing property in other related tasks, such as warning systems with real-time low-cost sensor readings.

The calibration method presented in this chapter improves the sensor readings accuracy and makes the follow-up analysis tasks reliable, e.g., finding and controlling the pollution sources based on sensor readings will be more efficient in Chapter 6. However, in real deployment, most of the sensors have no access to references and makes it challenges to calibrate those low-cost sensor measurements. In the next chapter, we will detail the problems and present our solutions on how to transfer the calibration model from source sensors (with access to references) to target sensors (without access to references).

3

In-field Calibration Transfer for Air Quality Sensor Deployments

In recent years, hundreds of inexpensive air quality sensors have been deployed citywide to monitor urban air pollution. To guarantee data accuracy and consistency, these sensors need periodic calibration after deployment. As we discussed in Chapter 2, given the coexisting low-cost sensor measurements and the ground truth data, various on-site calibration models can be applied to improve the accuracy of those low-cost sensors. Specifically, our proposed SensorFormer method, which uses both *recent history* and *close future* data, acquires the best calibration accuracy and has the ability to be deployed to resource-constraint IOT devices. However, since access to ground truth references is often limited in large-scale deployments (see examples in Section 1.1.3), it is difficult to conduct city-wide post-deployment sensor calibration. In this chapter, we propose In-field Calibration Transfer (ICT), a calibration scheme that transfers the calibration parameters of source sensors (with access to references) to target sensors (without access to references). ICT is capable of transferring different types of calibration models, linear or non-linear ones (Section 1.1.1), to target sensors and acquire satisfactory results. The key observations are that *(i)* the distributions of ground truth in both source and target locations are similar, and *(ii)* the transformation is approximately linear. Therefore, ICT derives the transformation based on the similarity of distributions with a novel optimization formulation. The performance of ICT is further improved by exploiting spatial prediction of air quality levels and multi-source fusion. Experiments show that ICT is able to calibrate the target sensors as if they had direct access to the references.

3.1 Introduction

Motivation. Recently, many large-scale air pollution monitoring systems have been deployed, where tens to hundreds of low-cost air quality sensors are installed across the city to measure air pollution concentrations in real time [FRD17, SHT15a, CLL+14b, XCL+16].However, the raw measurements of these deployments often lack sufficient accuracy due to sensor noise, inter-device differences or environmental interference [JHW+16a, MZT18a].

An effective approach to improve the data quality of air quality sensors is calibration [XBP+12, SHT15a, LDC18b, MZT18a]. To calibrate a low-cost sensor, its measurements are transformed in a way that the calibrated measurements agree with the measurements of a highly accurate reference. While air quality sensors are usually calibrated before deployment (*pre-deployment calibration*), the calibration parameters still need to be frequently adjusted in the field after deployment (*post-deployment calibration*) [MZT18a]. It is reported that the calibration parameters may drift within one month after sensor deployment without re-calibration [MMH17b].

Challenges. Post-deployment calibration is challenging particularly for large-scale static air pollution monitoring deployments. This is because once deployed, these sensors tend to have irregular or even no access to references. Figure 3.1 shows a real sensor deployment for $PM_{2.5}$ monitoring in Beijing, China. Among the $1,000$ $PM_{2.5}$ sensors deployed, only 7 are installed next to highly accurate reference stations. Most existing post-deployment calibration schemes focus on *mobile* deployments, where virtual references are created when sensors meet in space and time, i.e., sensor rendezvous [XBP+12, SHT15a, XCL+16, MZST17b]. However, since the sensors do not physically meet in a static deployment, rendezvous-based calibration does not apply. A few pioneer proposals [TYIM05, MMH17b] leverage special situations when pollution concentrations are expected to be uniform in certain regions to calibrate sensors in a static deployment. This approach offers calibration opportunities of near-zero concentrations and is only useful for simple offset and gain calibration [MZT18a]. Yet the calibration model for $PM_{2.5}$ can be complex [CLL+14b, LDC18b] and needs to be derived with measurements covering a wide concentration range. It remains open how to calibrate a $PM_{2.5}$ sensor without access to a reference, a common problem faced in urban-scale static deployments.

Our Approach. To conduct *post-deployment calibration* for static sensor deployments, we take an approach inspired by *calibration transfer* [ZTK+11, YZ16] in *pre-deployment calibration*. Calibration transfer is a calibration paradigm for sensors without access to references (*target* sensors) leveraging those with access to references (*source* sensors). It

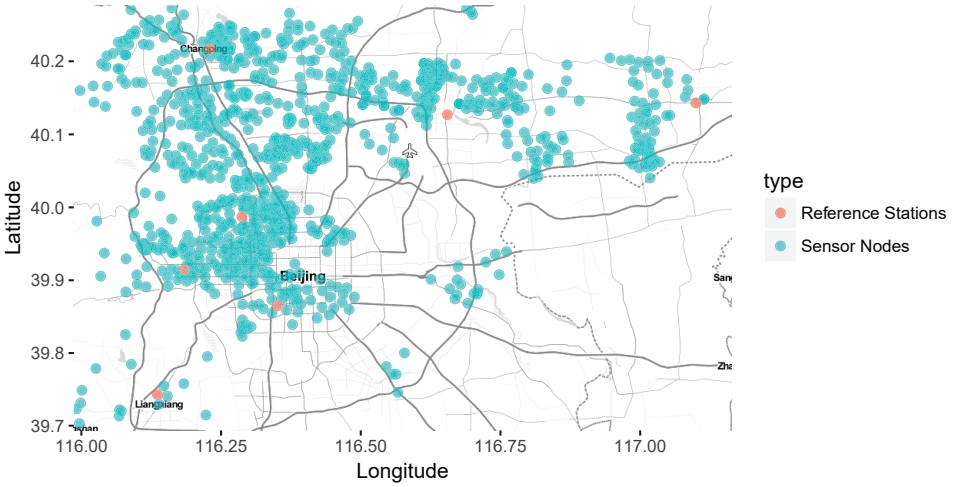


Figure 3.1: An illustration of sensors deployed in Beijing, China for $PM_{2.5}$ monitoring. Among the 1000 sensors deployed, only a few are installed close to the public environment monitoring stations, which are used as reference stations.

calibrates a target sensor by transferring the calibration parameters of a source sensor to a target sensor. The method has been adopted to reduce the *pre-deployment* calibration overhead in mass sensor production [ZTK+11, YZ16, YKZ18]. A pre-requisite of conventional calibration transfer is that measurements of the source and target sensors should be *synchronized*, i.e., the two sets of measurements from both sensors can be organized into pairs, in which both measurements are made upon *the same ground truth*. Synchronized measurements are guaranteed in pre-deployment calibration by putting both the source and the target sensors in the same testing environment. However, for post-deployment calibration, there is often limited, if any, prior knowledge on which pair of measurements from the source and the target sensors are made upon the same ground truth. That is, the measurements are largely *unsynchronized*. Hence conventional calibration transfer for pre-deployment calibration cannot be directly applied to post-deployment calibration.

In this chapter, we ask the question: *can we transfer the calibration parameters of source sensors to a target sensor, when no synchronized measurements are available?* We formulate the question as an *unsynchronized calibration transfer* problem, which aims to learn a transformation of the calibration parameters of the source sensors, and applies the transferred calibration on the target sensor to achieve high accuracy, even if the measurements of the source and the target sensors are unsynchronized. Note that unsynchronized measurements are not

comparable, and it can be erroneous to directly learn a transformation using unsynchronized measurements. Although it is difficult to solve the generic unsynchronized calibration transfer problem, we make a key observation that helps to solve the unsynchronized calibration transfer problem for urban air pollution monitoring deployments. Specifically, we observe that the $PM_{2.5}$ concentrations at two separate yet sufficiently close locations during the same period of time exhibit similar distributions. It implies that for a source sensor and a target sensor deployed at different locations, the ground truth concentrations of their measurements during the same period of time conform to similar distributions. Using this similarity between distributions of ground truth as a common reference, we develop a solution called *statistical calibration transfer* to this special unsynchronized calibration transfer problem.

On this basis, we propose In-field Calibration Transfer (ICT), an optimization based solution to the unsynchronized calibration transfer problem for static air quality sensor deployments. ICT has three technical novelties.

- We introduce statistical calibration transfer, which makes use of the similarity in distributions of the ground truth at different locations as common references rather than rely on synchronized measurements. Statistical calibration transfer learns the transformation from the estimated distribution of measurements using a novel optimization objective, which can be solved via Bayesian optimization.

- We reduce the search space in statistical calibration transfer by assuming a linear transformation between the calibration parameters between the source and target sensors. This assumption has been tested in labs [ZTK+11, YZ16, YKZ18] and we extend it into in-field scenarios.

- We further improve the accuracy of statistical calibration transfer by using an extra air pollution inference engine to generate $PM_{2.5}$ concentration level estimates for the target location. We augment the original optimization objective of statistical calibration transfer with an additional term. We empirically show that even coarse-grained $PM_{2.5}$ concentration levels suffice to improve the calibration accuracy.

Contributions and Roadmap. The main contributions of this work are summarized as follows. *(i)* To the best of our knowledge, ICT is the first solution to the unsynchronized calibration transfer problem for low-cost air quality sensors. It offers a practical solution to conduct post-deployment calibration for large-scale static urban air pollution monitoring deployments. *(ii)* We evaluate the performance of ICT on real deployment data and experimental results show that ICT is able to

provide approximately equally good calibration performance as if the target sensors have direct access to references, which could potentially increase the usability of large-scale air pollution monitoring sensor deployments.

In the rest of the chapter, we first review relevant literature (Section 3.2) and present the background and the problem (Section 3.3). Then we elaborate on the ICT (Section 3.4) and its evaluation (Section 3.5). Finally we conclude this work (Section 3.6).

3.2 Related Work

Our work is a post-deployment sensor calibration scheme for static air pollution monitoring deployments. It is inspired by applications of transfer learning in sensor calibration. We review the closely related literature below.

3.2.1 Post-deployment Calibration for Air Quality Sensors

Although low-cost air quality sensors are usually calibrated before installation, periodic post-deployment calibration is still necessary to ensure long-term data accuracy of urban air pollution monitoring systems. Unlike pre-deployment calibration, where every sensor has a reference e.g., in labs, a unique challenge in post-deployment calibration is the lack of references. Virtual references can be created if the sensors are *mobile* and meet in space and time, i.e., sensor rendezvous [XBP+12]. Sensors in a rendezvous are supposed to sense the same phenomenon and can be utilized as references for calibration [SHT15a, MZST17b]. However, rendezvous-based calibration only applies to mobile sensors and a sensor with no rendezvous cannot be calibrated [FRD17].

For static sensors, post-deployment calibration is viable by exploiting situations where all sensors measure the same pollution concentrations so that they can share the same reference for calibration. Tsujita et al. [TYIM05] the NO_2 concentrations are almost uniform within the city if the concentrations are low. Thus they propose to calibrate the offset of NO_2 sensors deployed in the city to four references once a NO_2 concentration below 10 ppb is reported. Mueller et al. [MMH17b] assume that O_3 and NO_2 concentrations are uniform during night at inner city locations and during the afternoon at outer city locations. Correspondingly, the sensors in the inner/outer city can be calibrated to a remote reference in the inner/outer city during night/afternoon.

Our work is also a post-deployment calibration scheme for static sensors, but differs from existing efforts in two-fold. *(i)* Previous studies on gas sensors [TYIM05, MMH17b] are built upon *linear* calibration models. As we will show in Section 3.3.2, linear models are insufficient for dust sensors e.g., $PM_{2.5}$ in our case. *(ii)* The calibration opportunities

in [TYIM05, MMH17b] only provide near-zero concentrations, which will yield large calibration errors if they are used in complex non-linear calibration models (see the NZ-ICT baseline in Section 3.5). Therefore these two prior studies are not directly applicable to in-field $PM_{2.5}$ calibration transfer. In contrast, our work applies to both simple and complex calibration models.

3.2.2 Transfer Learning in Sensor Calibration

Transfer learning is a machine learning paradigm aims to improve the learning of the target predictive function in the target domain using the knowledge in a source domain and a source learning task [PY10a]. It has broad applications in text mining [PTKY11], computer vision [OBLS14], urban computing [WZY16, GLZ+18], etc.

In the sensor and measurement research, the concept of transfer learning has been mainly applied in calibrating electronic noses (e-noses). E-noses are sensor arrays for hazardous odor detection. Due to their significant inter-device differences, per-instrument calibration is necessary, and transfer learning is utilized to reduce the calibration overhead in mass production [ZTK+11, FFGG+16, YZ16, YKZ18]. Assume a source e-nose and a target e-nose. The raw measurements of the target are first standardized to those of the source e-nose. Then the source e-nose is calibrated to a reference and finally the calibration parameters can be directly adopted on the target e-nose.

Our work is inspired by the concept of calibration transfer in e-noses. However, most e-nose calibration transfer studies are performed *in labs* for *pre-deployment calibration* while we focus on *in-field* calibration transfer for *post-deployment calibration*. The former assumes the source and the target sensors are measuring the same phenomenon in the same lab setting, i.e., synchronized. Yet the latter is more challenging because the source and the target sensors are installed at different locations and their measurements are largely unsynchronized.

3.3 Problem Definition and Analysis

In this section, we first introduce the basics of sensor calibration (Section 3.3.1) and then conduct a measurement study on a $PM_{2.5}$ monitoring deployment to motivate the need for calibration transfer (Section 3.3.2). Finally we formally define the unsynchronized calibration transfer problem (Section 3.3.3).

3.3.1 Primer on Air Pollution Sensor Calibration

Calibration is an efficient approach to improve the data quality of low-cost sensors. It finds a *calibration model* that maps the measurements of

a *low-cost sensor* to those of an accurate *reference sensor* [MZT18a]. Given a set of measurements $X = \{x_1, x_2, \ldots, x_N\}$ of a low-cost sensor and a set of measurements $Y = \{y_1, y_2, \ldots, y_N\}$ of a reference sensor, a calibration model C establishes a relationship between X and Y such that certain error metric between the calibrated measurements $C(X)$ and the reference measurements Y is minimized. There has been extensive research on how to derive calibration models suited for different air pollution sensors and error sources. We refer interested readers to [MZT18a] for a comprehensive review.

For air pollution sensors, it is crucial to conduct both *pre-deployment* and *post-deployment* calibration. Pre-deployment calibration identifies the proper calibration model, while periodic post-deployment calibration is important to maintain the data quality of long-term deployment. One major challenge in post-deployment calibration is the lack of reference sensors to re-calibrate the low-cost sensors. This is particularly the case for large-scale static air pollution sensor deployments, which our work focuses on.

3.3.2 Measurement Study

This subsection presents a measurement study on a real-world $PM_{2.5}$ sensor deployment to motivate the need for calibration transfer. Specifically, the measurement study aims to answer three questions: *(i)* Is a linear calibration model sufficient for $PM_{2.5}$ sensor calibration? *(ii)* Is it necessary to periodically re-calibrate $PM_{2.5}$ sensors? *(iii)* Is it feasible to directly apply calibration parameters of one sensor to sensors at other locations?

Sensor Deployment and Dataset. We collect measurements from a large-scale $PM_{2.5}$ monitoring system deployed in Beijing, China. It consists of $1,000$ low-cost sensors measuring $PM_{2.5}$, temperature and humidity (see Figure 3.1). In addition to $PM_{2.5}$ concentration, the *PM* sensor [Yun18] in each sensor box (see Figure 3.2-(b)) also reports 12 low-level features. Each sensor uploads its readings to a back end server every minute. Among the $1,000$ low-cost sensors, only 7 (denoted as $S1$ to $S7$ in Figure 3.2-(a)) are installed next to highly accurate air pollution monitoring stations as references (see Figure 3.2-(c)). The remaining sensors have no access to the reference stations. The low-cost $PM_{2.5}$ sensors are based on light scattering principles [Yun18], while the reference stations are based on beta-attenuation or tapered element oscillating microbalance method [oEE18].

For the measurement study, we collect readings ($PM_{2.5}$ concentration and the 12 low-level features) from the 7 sensors as well as the $PM_{2.5}$ readings from the corresponding 7 reference stations as ground truth [Bei18]. The dataset collected covers a time period of 10 months from October 1, 2017 to July 31, 2018.

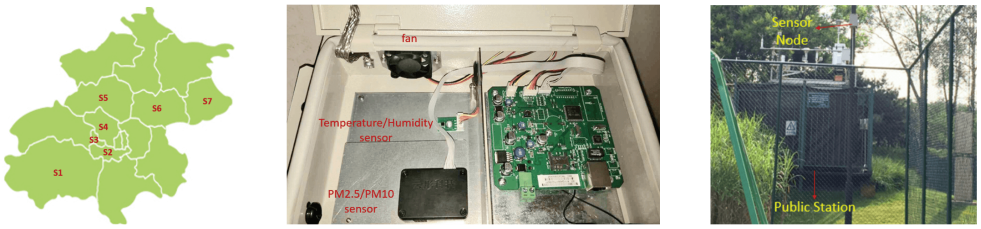


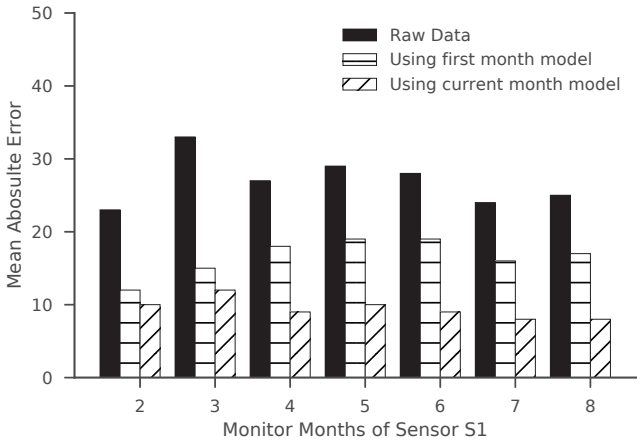
Figure 3.2: Illustration of sensor deployment. (a) Locations of sensors (S_1 to S_7) with access to public reference stations (R_1 to R_7). (b) Hardware of sensor. (c) Installation of a sensor next to a reference station.

Table 3.1: MAEs of different calibration models.

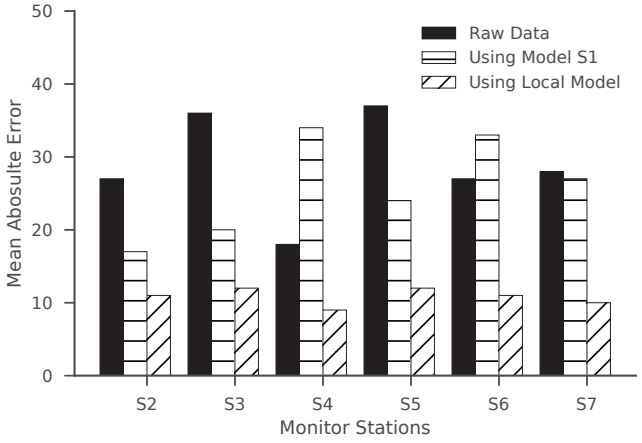
Calibration Model	Raw Data	Multiple Regression	Random Forest
MAE	30	22	9

Whether a Linear Calibration Model is Sufficient for $PM_{2.5}$ Calibration. While linear calibration models are prevalent in gas sensor calibration [TYIM05, MMH17b], non-linear models are often needed for dust sensor calibration such as $PM_{2.5}$ [LDC18b]. Table 3.1 shows the mean absolute errors (MAE) of applying the popular linear (multiple linear regression [TYIM05, MMH17b]) and non-linear (random forest [LDC18b]) models to calibrate the raw measurements of the low-cost $PM_{2.5}$ sensor with reference to their co-located highly accurate reference station. MAE is a widely used metric to evaluate the data accuracy of air pollution sensors [MZT18a]. For $PM_{2.5}$ concentration, a MAE below 10 is considered accurate for data mining applications [BJZOV17b]. The evaluation is conducted on the sensor node S6 and the MAEs are averaged over 10 months (October 1, 2017 to July 31, 2018). For each month, 70% of the data are used for training and 30% for testing to calculate the MAEs of different calibration methods. The results show that linear calibration models fail to yield satisfactory accuracy of calibration on $PM_{2.5}$ sensors. This suggests that previous calibration transfer studies [TYIM05, MMH17b], which are built upon linear calibration models, are not directly applicable. In the rest of this chapter, we take the random forest described in [LDC18b] as the calibration model for $PM_{2.5}$ sensors.

Whether Periodic Calibration is Necessary. Figure 3.3a shows the MAEs of the uncalibrated raw measurements of Sensor S1, and two different calibration approach applied on it. The first approach is to train the calibration model on the 70% training set in the first month, and then test it on the 30% testing set in the next 7 months. The second approach is to directly train the calibration model on the current month's training



(a)



(b)

Figure 3.3: MAEs of directly applying the calibration model learned from measurements of $S1$ collected in first month to calibrate (a) measurements of $S1$ collected from second month to the eighth month and (b) measurements of the other 6 sensors ($S2$ to $S7$) collected in the first month.

set and test on the testing set. As shown in the figure, the second approach provides lower MAEs. This result indicates that the optimal calibration parameters for the same sensor do vary over time. It may induce large errors by directly adopting a previously trained calibration model to calibrate even the same sensor after a long period of time.

Whether One Set of Calibration Parameters is applicable to Sensors at Different Locations. Figure 3.3b plots the MAEs by applying the calibration model of $S1$ to calibrate the $PM_{2.5}$ measurements of the other 6 sensors collected in the same month. The results show that the MAEs can be even larger than those of the raw measurements without calibration, which indicates that the optimal calibration parameters for sensors at different locations can differ significantly. Therefore, the calibration

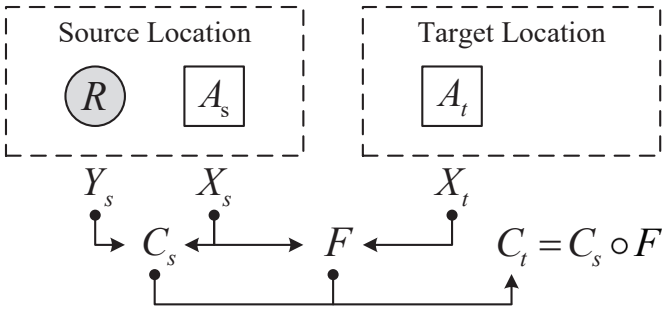


Figure 3.4: An illustration of calibration transfer problem. A source sensor A_s is co-located with a reference sensor R at the source location, while a target sensor A_t has no access to any reference at the target location. Their measurements are X_s, Y_s, and X_t, respectively. A calibration model C_s for A_s can be learned from X_s and Y_s. The calibration transfer problem tries to derive a function F using X_s and X_t such that the calibration model C_s can be transferred to A_t, where the calibration model for A_t can be calculated as $C_t = C_s \circ F$.

model learned for one sensor requires to be adapted (transferred) to be used on other sensors.

Summary. $PM_{2.5}$ sensor calibration needs non-linear models e.g., random forests. It is necessary to conduct periodic re-calibration for each deployed sensor, which can be expensive and labor-intensive. This is particularly the case when large numbers of sensors are static and have no access to the references. To reduce the overhead of post-deployment calibration, we explore to *transfer* the calibration results from source sensors (with access to references) to target sensors (without access to references).

3.3.3 Unsynchronized Calibration Transfer Problem

For ease of presentation, we explain our problem by using one source sensor and one target sensor (Figure 3.4). We discuss extensions to multi-source scenarios in Section 3.4.4. The **calibration transfer problem** is defined as follows.

Denote A_s as a source sensor, which is co-located with a highly accurate reference station R_s. We use $X_s = \{x_s^{(i)}\}_{i=1}^{N_s}$ to represent the measurements of A_s, where $x_s^{(i)} \in \mathbb{R}^d$ is the i^{th} measurement, i.e., a d-dimension feature vector, and N_s is the number of measurements of A_s. Similarly, $Y_s = \{y_s^{(i)}\}_{i=1}^{N_s}$ represents the measurements of R_s, where $y_s^{(i)} \in \mathbb{R}$ is the i^{th} measurement, i.e., the ground truth $PM_{2.5}$ concentration corresponding to the measurement $x_s^{(i)}$. Then a calibration model $C_s : \mathbb{R}^d \rightarrow \mathbb{R}$ can be learned for the source sensor A_s from X_s and Y_s, as discussed in Section 3.3.1 by minimizing $\|Y_s - C_s(X_s)\|_F^2$, where $\|\cdot\|_F$ is the Frobenius Norm. Finally, denote A_t as a target sensor, and $X_t = \{x_t^{(i)}\}_{i=1}^{N_t}$ as its measurements ($x_t^{(i)} \in \mathbb{R}^d$, and N_t is the number of measurements of A_t). Y_t is used to denote the

corresponding ground truth $PM_{2.5}$ concentration at the target location. The calibration transfer problem aims to find a transformation function $F : \mathbb{R}^d \to \mathbb{R}^d$, such that $\|Y_t - C_s(F(X_t))\|_F^2$ is minimized. In other words, the calibration model C_s for A_s is transferred to $C_t = C_s \circ F$ for A_t.

There are two types of calibration transfer problems, **synchronized calibration transfer problem** and **unsynchronized calibration transfer problem**. The former assumes that the measurement set X_s and X_t are synchronized, i.e., $N_s = N_t = N$ and for each $i = 1 \dots N$, and we have $y_s^{(i)} = y_t^{(i)}$. This type of calibration transfer can be solved by direct standardization [FFGG+16], which assumes the transformation function F to be linear. It has been applied to calibrate large numbers of instruments *in labs* when it is time-consuming to learn a (often complex) calibration model for each instrument [ZTK+11, FFGG+16, YZ16, YKZ18]. In our particular interest is the latter, i.e., **unsynchronized calibration transfer problem**, where Y_t is not known, and X_t *cannot be synchronized* to X_s. This is the common case for static air pollution sensor deployments. In this case, it remains open how to learn F from X_t and X_s, which is the focus of this work.

3.4 In-field Calibration Transfer

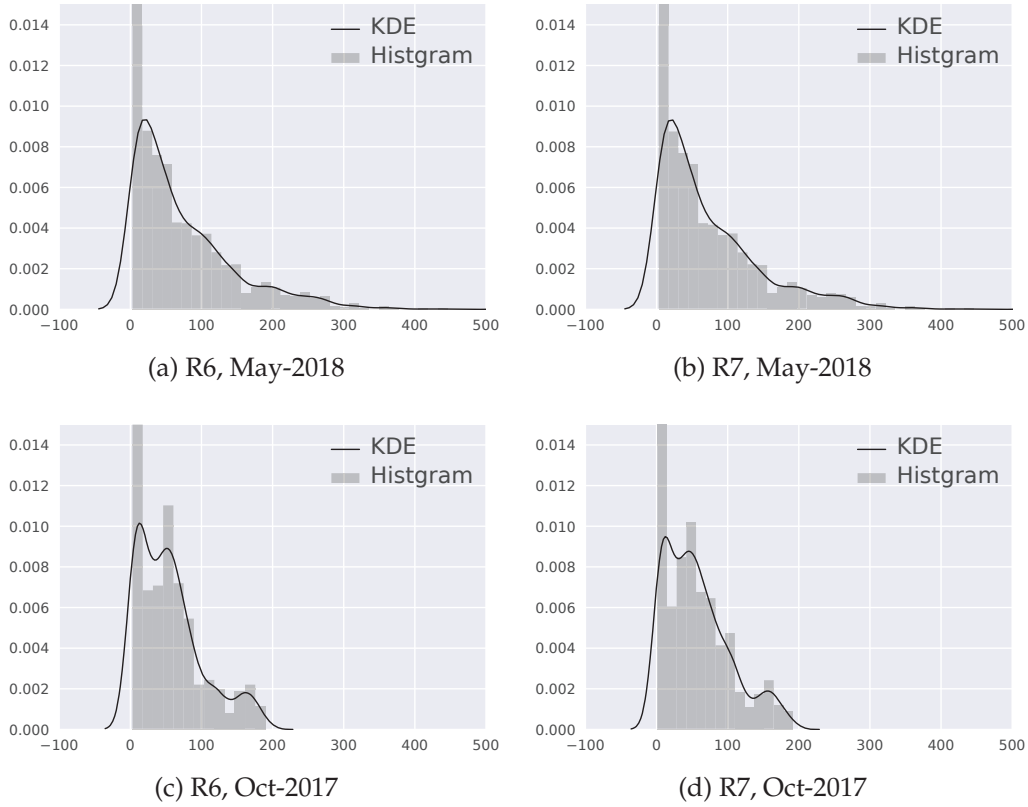
To solve the unsynchronized calibration transfer problem, we propose ICT (in-field calibration transfer). We elaborate each technique in ICT for single-source calibration transfer, including statistical calibration transfer (Section 3.4.1), exploiting linearity of the transformation (Section 3.4.2), and exploiting results from spatial predictions (Section 3.4.3). Then we extend ICT to the multi-source scenario (Section 3.4.4).

3.4.1 Statistical Calibration Transfer

Main Idea. One fundamental challenge in the unsynchronized calibration transfer problem is that Y_t is unknown, so there is no common reference to synchronize X_t and X_s. The key idea of our solution is based on the following assumption: for the same period of time, and when the distance between A_s and A_t are small enough, we have $p(Y_s) \approx p(Y_t)$, where $p(\cdot)$ denotes the probability distribution. Based on this assumption, it is possible to find the transformation function F by solving the following optimization problem:

$$\underset{F}{\operatorname{argmin}}\ d_{KL}[\hat{p}(Y_s), \hat{p}(C_s(F(X_t)))] \tag{3.1}$$

where $\hat{p}$ is a histogram density estimator, and $d_{KL}[\cdot, \cdot]$ is the Kullback-Leibler (KL) divergence. Instead of synchronizing individual measurements, we learn the transformation F by minimizing the difference



(a) R6, May-2018 (b) R7, May-2018

(c) R6, Oct-2017 (d) R7, Oct-2017

Figure 3.5: The ground truth distribution of reference sensor *R6* and *R7*, co-located near the low cost sensor *S6* and *S7*, respectively, as shown in Figure 3.2. The distribution of different reference sensors in the same month is similar, while the distribution of even the same sensor among different months varies a lot.

between the *estimated distribution of calibrated target measurement* and the *ground truth at source location*.

While conventional calibration transfer requires explicit pre-knowledge of Y_t, such that both measurement sets X_s and X_t can be synchronized accordingly, our statistical calibration transfer loosens this requirement: when the distributions of ground truth in both location, $p(Y_s)$ and $p(Y_t)$, are known to be similar, it is enough to transfer the calibration.

Empirical Validation of Key Assumption. The effectiveness of statistical calibration transfer relies on the key assumption that for the same period of time, and when the distance between A_s and A_t are small enough, we have $p(Y_s) \approx p(Y_t)$. This assumption is built upon our observation that *during the same period of time*, when the source location and target location *both locate near to each other* (e.g., in the same city), the distributions of the true $PM_{2.5}$ concentrations in both location are similar. Below we empirically demonstrate this observation.

While the $PM_{2.5}$ concentration usually varies in space and time,

its distributions over a certain period of time may be similar at different locations because of e.g., similar land-use and pollution sources. Figure 3.5 plots the ground truth $PM_{2.5}$ concentrations measured by reference stations $R6$ and $R7$, co-located next to the low-cost sensors $S6$ and $S7$ shown in Figure 3.2 over two different months. Figure 3.5a and Figure 3.5b shows the histogram and density distributions of reference sensor $R6$ and $R7$ during May, 2018, which are quite similar to each other. The same phenomenon can also be observed in Figure 3.5c and Figure 3.5d. However, on the other hand, the distribution in different months of the same reference sensors varies greatly (e.g., Figure 3.5a vs. Figure 3.5c).

To explore whether this observation is an artefact of the Beijing dataset, we collect $PM_{2.5}$ measurements from public stations in three other major cities in China, Tianjin, Shanghai and Shenzhen, which are 130, 1200 and 1900 kilometres away from Beijing, respectively. From each of these cities, 9 public stations are selected. KL divergence is calculated between the monthly $PM_{2.5}$ concentration distributions of two sensors, and averaged over the 12 months from October, 2017 to October, 2018. The results are shown in Figure 3.6.

In Figure 3.6-(a)–(d), we calculate the KL divergence among $PM_{2.5}$ measurement distributions of different public stations located within each of the four cities. The overall KL divergence of the other three cities is at the same low level as Beijing. As shown in Section 3.5, the similarity among the measurement distributions within Beijing suffices for our ICT to provide a solid calibration performance. Since the same level of similarity is also observed in other cities, we believe that ICT is generally applicable to intracity post-deployment calibration problems.

To further investigate the similarity of $PM_{2.5}$ measurement distributions from sensors in different locations, we compare sensors located in different cities with each other and summarise the results in Figure 3.6-(e)–(g). As shown in the figure, the $PM_{2.5}$ measurement distributions in Beijing & Tianjin have higher similarity than Beijing & Shanghai, which then have higher similarity than Beijing & Shenzhen. In general, we observed that the similarity of $PM_{2.5}$ measurement distributions between two cities is negatively correlated to the geological distance between them.

We make the following comments on the usage of the observation.

- We only consider mapping the measurements between the source and the target collected during the same period of time in statistical calibration transfer. We do not consider transferring measurements of different months because their distributions are likely to differ due to seasonal changes.

- We limit the spatial range between the source and the target within a city for statistical calibration transfer. It is also possible that the observation may not hold at certain locations within the city.

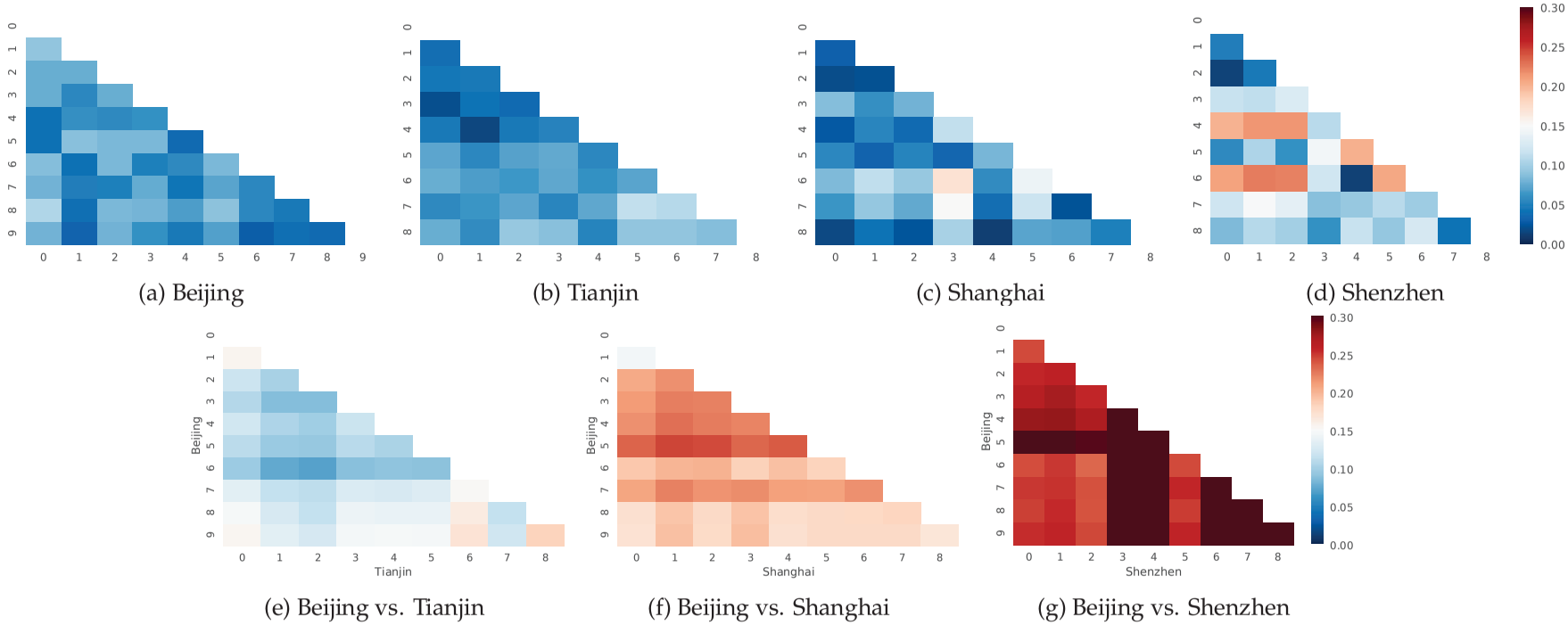


Figure 3.6: $PM_{2.5}$ distribution differences (measured by KL divergence) between stations in (a) Beijing; (b) Tianjin; (c) Shanghai; (d) Shenzhen; (e) Beijing and Tianjin; (f) Beijing and Shanghai and (g) Beijing and Shenzhen

Nevertheless, as we will introduce in Section 3.4.4, our method still works because it can regard these locations as negative transfer samples with the help of multi-source calibration transfer.

3.4.2 Exploiting Linearity of Transformation Function F

As discussed earlier, direct standardization [FFGG+16] learns F directly from X_s and X_t. Previous studies [ZTK+11, FFGG+16, YZ16, YKZ18] have assumed that the transformation function F to be linear and have shown that this assumption works well for gas sensors in labs. We extend this assumption to in-field scenario, i.e., we assume transformation function F is also a linear function in in-field situation.

The experiment results Section 3.5 shows that this assumption allows ICT to provide decent calibration accuracy, while notably reducing the searching space of F and increases the efficiency of the algorithm.

3.4.3 Exploiting Spatial Prediction Results

In practice, sometimes there are inferences of the ground truth at the target area provided by other methods, e.g., air quality map. The performance of ICT can be further improved by making use of these inferred target ground truth, denoted as $Y_t^{'}$. Specifically, a new term can be added to Eq. (3.1):

$$\underset{F}{\operatorname{argmin}}\, d_{KL}[\hat{p}(Y_s), \hat{p}(C_s(F(X_t)))] + \lambda \cdot d_c[Y_t^{'}, C_s(F(X_t^{'}))] \tag{3.2}$$

where $d_c[\cdot,\cdot]$ is a measure of distance between two sample sets and λ is a parameter used to adjust the influence of the inferred target ground truth.

In this work, we take $PM_{2.5}$ concentration levels inferred by an air quality map as $Y_t^{'}$ at the location of the target sensor. Instead of accurate $PM_{2.5}$ concentration, $Y_t^{'}$ consists of integer value ranging from 1 to 6, which represent different $PM_{2.5}$ concentration levels. Hence $d_c[\cdot,\cdot]$ in Eq. (3.3) is defined as the classification error rate. As we will show in the evaluation section, a small fraction of inferences with high confidence suffices to provide considerable improvement in calibration transfer.

In ICT, we apply Gaussian process regression [CLL+14b] for inferred ground truth generation. While other models for air quality inference also apply, we choose Gaussian process regression for its simplicity and efficiency. We take sensor readings from high quality public stations, GPS location information, eight categories of POI data (culture & education, parks, sports, hotels, shopping malls & supermarkets, entertainment, decoration & furniture markets, and vehicle services) and meteorological data (temperature, humidity, wind speed, wind direction) as the input of the spatial predictor and the output is $y_t^{'(i)}$ with an according variance $\sigma_t^{(i)}$, which indicates the confidence of the prediction. We only use the

prediction data $y_t^{'(i)}$ with $\sigma_t^{(i)}$ smaller than some threshold τ. We form a prediction set $Y_t^\tau = \{y_t^{'(i)} | \sigma_t^{(i)} < \tau\}$ and its corresponding measurement set from target sensor X_t^τ, and Eq. (3.2) then becomes

$$\underset{F}{\operatorname{argmin}}\, d_{KL}[\hat{p}(Y_s), \hat{p}(C_s(F(X_t)))] + \lambda \cdot d_c[Y_t^\tau, C_s(F(X_t^\tau))] \tag{3.3}$$

3.4.4 Extension to Multi-source Calibration Transfer

This subsection extends ICT to support calibration transfer from multiple source sensors. The main challenge is to select the most promising sources to avoid negative transfer [GGN+14].

To support multi-source selection and transfer, we first need to quantify the differences between the environment of the source sensors and that of the target sensor. We use a classifier induced divergence measure called $\mathcal{H}$ distance [KBDG04] which measures the divergence that only affects the classification accuracy. We use $\mathcal{D}_s$ to represent the source domains, which has reference sensors and calibration models, while $\mathcal{D}_t$ is used to represent the target domain. We want to transfer the models learned in $\mathcal{D}_s$ to the target domain $\mathcal{D}_t$. Then, we use $d_{\mathcal{H}}^y(\mathcal{D}_s, \mathcal{D}_t)$ to represent the distance between the source environment and the target environment. The smaller the distance, the more similar the two environments are. We define the similarity between a source and the target as below.

$$\Phi^y(s,t) = 1 - d_{\mathcal{H}}^y(\mathcal{D}_s, \mathcal{D}_t) \tag{3.4}$$

Then we select the most promising sources according to their relative similarity weight, which is defined as:

$$\Psi\left(s_j\right) = \frac{\Phi\left(\mathcal{D}_{s_j}, \mathcal{D}_t\right)}{\sum_{s=1}^{m} \Phi\left(\mathcal{D}_s, \mathcal{D}_t\right)} \tag{3.5}$$

where s_j is the j^{th} source domain among m source domains, and $\Psi\left(s_j\right)$ is the weight used for ensemble. We can use these similarity weights to calculate the ensemble calibration transfer result. The similarity between feature values of the source and target domains reflects the similarity of domains. If a pair of domains are more similar, we can rely more on the calibration transfer result between them and put more weight on it.

Algorithm 1 shows the entire process of ICT for multi-source in-field calibration transfer.

3.5 Experimental Evaluation

This section presents the evaluations of ICT. We introduce the experiment setups (Section 3.5.1), present the overall performance (Section 3.5.2), and then conduct micro-benchmark evaluations to understand the

Algorithm 1: In-field calibration transfer

Input: m source measurement and ground truth pairs (X_{s_j}, Y_{s_j}), target measurements X_t, certainty threshold τ, readings from other high quality public stations, GPS location information, POI data, and meteorological data

Output: Ensembled calibrated measurements of target sensor Y_t^e

Conduct spatial prediction and forms the data tuple (X_t^τ, Y_t^τ)
for *each source sensor* $j \in 1...m$ **do**
    Get calibration model C_{s_j} of source sensor A_{s_j} using the data pairs (X_{s_j}, Y_{s_j})
    Using the Bayesian optimization to solve the objective function and get transformation matrix F_j
    Calculate the similarity weight $\Psi\left(s_j\right)$
end
$Y_t^e = \sum_{j=1}^{m} \Psi\left(s_j\right)(C_{s_j}(F_j(X_t)))$
return Y_t^e

performance of each module in ICT (Section 3.5.3). Finally we conduct a case study on pollution source localization as an application of calibration transfer (Section 3.5.4).

3.5.1 Experiment Setup

Datasets. We mainly evaluate the performance of ICT using measurements from the 7 low-cost sensors, denoted as $S1 \ldots S7$, which are installed next to public stations (Figure 3.1). These public stations are also called *reference stations*, denoted as $R1 \ldots R7$. For each sensor, we collect the 12 low-level features as well as its $PM_{2.5}$ readings, which is called a *measurement*. The $PM_{2.5}$ concentration recorded by each reference station is collected as *ground truth*. Each measurement and ground truth at the same time are formed as a *tuple*. We collect one tuple per hour for 10 months (from October 1, 2017 to July 31, 2018), which covers various weather conditions and $PM_{2.5}$ concentration range. In total, there are more than 50,210 tuples. In our experiments, 70% of the collected data are used as training set, and 30% as test set.

Note that ICT also needs data to infer ground truth via spatial prediction (see Section 3.4.3). To built up the spatial prediction model, we collect $PM_{2.5}$ concentration data from all the 35 public stations in Beijing [Bei18] (including $R1 \ldots R7$), meteorological data [The18] (including temperature, humidity, wind speed, wind direction), GPS location information, as well as POI data [Inc21] (see Section 3.4.3 for details) for the same time period.

Metrics. Mean absolute error (MAE) after calibration is the main metric used to assess the performance of ICT. To make the results more semantically useful for end users, we also calculate the classification accuracy on the officially defined 6 discrete $PM_{2.5}$ levels [CLL+14b], which

is indicated as $L1 \ldots L6$.

Baselines. The following baselines are used.

- **Direct Transfer**: It directly use the calibration parameters of a source sensor to a target sensor without any transformation.
- **TCA**: It is a popular transfer learning method in computer vision. Some research [MZT18a] suggests it is also a potential solution to calibration transfer. TCA [PTKY11] calculates a common transfer Φ which applies on both X_s and X_t. The source calibration model C_s is learned by minimizing $\|Y_s - C_s(\Phi(X_s))\|_F^2$, rather than $\|Y_s - C_s(X_s)\|_F^2$. This source calibration is then applied on the transformed target measurement, i.e., $C_S(\Phi(X_t))$, which is evaluated with the ground truth Y_t. We use TCA with a polynomial kernel to reduce the feature space to 10 dimensions, i.e., $\Phi(x_s^{(i)}) \in \mathbb{R}^{10}$.
- **NZ-ICT**: It represents ICT using only near-zero measurements as references. Some studies on gas sensors [TYIM05, MMH17b] use near-zero measurements for calibration transfer. We consider $PM_{2.5}$ concentrations under 35, i.e., *level 1* as near zero data, and feed them into the standard ICT for calibration transfer.

When comparing the performance of the above baselines in multi-source calibration transfer scenario, we use the corresponding ensemble solution of these methods.

3.5.2 Overall Performance

Table 3.2, Table 3.3, Table 3.4 and Table 3.5 show the results of single-source calibration transfer using different methods for all of the seven sensor pairs. The MAEs greater than 20 are marked in red. As is shown, direct transfer has the worst MAEs. TCA and NZ-ICT yields smaller MAEs but there are still some large MAEs marked in red. In contrast, all the MAEs of ICT are below 18. Worth mentioning is that the emboldened diagonal elements of these three tables represent the calibration transfer of the 7 sensors with themselves. In other words, they are *directly calibrated* with their co-located reference stations $R1 \ldots R7$. As we can see from Table 3.5, the performance of ICT (non-diagonal elements) is already close to that of direct calibration (diagonal elements). This can be seen as a proof of the effectiveness of ICT.

Table 3.6 shows the cross-validation results using multi-source calibration transfer. Each sensor is set as target sensor and we transfer the calibration model from all other sensors. Again, *direct calibration* represents the performance of the calibration model learned directly using target sensor data and the the ground truths, which can be seen as the best possible performance for any calibration transfer method. The result

Table 3.2: Direct transfer result

source sensor \ target sensor	S1	S2	S3	S4	S5	S6	S7
S1	**10**	14	15	21	36	28	17
S2	13	**12**	12	12	31	22	15
S3	18	18	**9**	9	28	17	12
S4	25	21	13	**7**	25	14	14
S5	53	49	42	34	**11**	26	43
S6	39	36	24	15	22	**10**	24
S7	19	19	12	15	30	18	**9**

Table 3.3: TCA transfer result

source sensor \ target sensor	S1	S2	S3	S4	S5	S6	S7
S1	**10**	19	24	23	19	28	15
S2	19	**12**	18	13	22	27	18
S3	21	15	**9**	12	18	17	21
S4	30	19	10	**7**	12	18	19
S5	18	18	29	12	**11**	17	18
S6	21	32	15	25	22	**10**	31
S7	29	15	14	15	18	22	**9**

Table 3.4: NZ-ICT transfer result

source sensor \ target sensor	S1	S2	S3	S4	S5	S6	S7
S1	**10**	17	18	22	24	23	16
S2	17	**12**	19	15	26	31	20
S3	17	16	**9**	15	22	19	18
S4	36	22	13	**7**	13	21	17
S5	31	29	22	24	**11**	19	25
S6	31	22	17	21	17	**10**	26
S7	22	17	12	18	23	17	**9**

Table 3.5: ICT transfer result

source sensor \ target sensor	S1	S2	S3	S4	S5	S6	S7
S1	**10**	15	12	13	14	13	13
S2	12	**12**	12	9	14	10	13
S3	13	14	**9**	10	15	10	12
S4	13	13	11	**7**	15	9	11
S5	13	15	12	11	**11**	14	16
S6	13	13	11	7	14	**10**	11
S7	13	16	11	11	17	10	**9**

shows that ICT in the multi-source scenario achieves the best performance among all the methods, and it provides an almost equal performance as the direct calibration.

Table 3.6: Cross-validation results using multi-source calibration transfer

	S1	S2	S3	S4	S5	S6	S7
Ensemble Direct Transfer	23	20	19	16	24	28	17
Ensemble TCA	18	17	18	15	14	14	15
Ensemble NZ-ICT	16	18	17	14	18	21	14
Ensemble ICT	**11**	**12**	**10**	**8**	**11**	**8**	**11**
Direct Calibration	**10**	**12**	**9**	**7**	**11**	**10**	**9**

3.5.3 Micro-benchmarks

In this series of experiments, we set $S6$ as the target sensor and the other six sensors ($S1 \ldots S5, S7$) as the source sensors, and evaluate the impact of different techniques in ICT on the overall performance.

Table 3.7: Direct transfer matrix
MAE=36, Acc=0.48

Ground Truth	Predictions L1	L2	L3	L4	L5	L6	Recall
L1	**444**	265	16	2	3	0	0.61
L2	26	**228**	169	15	3	3	0.51
L3	1	4	**89**	120	75	8	0.30
L4	1	0	4	**22**	109	17	0.14
L5	0	0	6	5	**47**	119	0.27
L6	0	0	0	0	0	**62**	1.00
	0.94	0.46	0.31	0.13	0.20	0.30	
	Precision						

Table 3.8: TCA transfer matrix
MAE=17, Acc=0.68

Ground Truth	Predictions L1	L2	L3	L4	L5	L6	Recall
L1	**516**	211	1	2	1	0	0.71
L2	25	**391**	15	1	2	0	0.90
L3	0	120	**137**	33	6	1	0.46
L4	1	9	42	**81**	20	0	0.53
L5	0	8	8	29	**119**	13	0.67
L6	0	0	0	0	39	**23**	0.37
	0.95	0.53	0.67	0.55	0.64	0.62	
	Precision						

Table 3.9: NZ-ICT transfer matrix
MAE=24, Acc=0.58

Ground Truth	Predictions L1	L2	L3	L4	L5	L6	Recall
L1	**685**	43	3	0	0	0	0.94
L2	147	**275**	11	0	1	0	0.63
L3	2	208	**83**	2	2	0	0.28
L4	1	13	130	**9**	0	0	0.06
L5	0	10	57	86	**24**	0	0.14
L6	0	0	0	12	48	**2**	0.03
	0.82	0.50	0.29	0.08	0.32	1.0	
	Precision						

Table 3.10: ICT transfer matrix
MAE=13, Acc=0.78

Ground Truth	Predictions L1	L2	L3	L4	L5	L6	Recall
L1	**577**	146	5	2	1	0	0.79
L2	27	**351**	53	1	2	0	0.81
L3	1	34	**239**	17	5	1	0.80
L4	1	2	38	**99**	12	1	0.65
L5	0	1	14	25	**135**	2	0.76
L6	0	0	0	0	26	**36**	0.58
	0.95	0.66	0.68	0.69	0.75	0.9	
	Precision						

Performance of Single-source Calibration Transfer. Here we show the calibration transfer results from $S7$ to $S6$. Table 3.7 shows the performance of direct transfer. The MAE is 36 and the classification accuracy of the 6 $PM_{2.5}$ levels ($L1 \ldots L6$) is only 0.48, which are almost unusable. Table 3.8 shows the results of TCA. The MAE decreases to 17 and the overall accuracy improves to 0.68. Table 3.9 shows the results of NZ-ICT. Using only near-zero measurements as references, NZ-ICT provides only MAE of 24 and accuracy of 0.58, which is even not as good as TCA. Finally, Table 3.10 shows the results of our ICT, with λ in Eq. (3.3) set to 0.3. The MAE and classification accuracy is improved to 13 and 0.78, respectively. Moreover, the recall of each level is generally better than the other three methods.

Performance of Multi-source Calibration Transfer. Table 3.11 shows the resulting ensemble weights. The weights of $S5$ and $S7$ are relatively large because they are close to $S6$ and may have a similar environment. Table 3.12 shows the results using ensemble direct transfer. The MAE decreases from 36 in the single-source scenario to 28 in the multi-source scenario, and the accuracy improves from 0.48 to 0.55. Ensemble TCA also achieves a better result than the single-source scenario, with $MAE = 14$ and $Acc = 0.76$, as shown in Table 3.13. Ensemble NZ-ICT acquires only a slightly better transfer result when compared with ensemble direct transfer, with $MAE = 21$ and $ACC = 0.621$, as shown in Table 3.14. Using the full ensemble ICT approach, however, the final MAE is improved

Table 3.11: Ensemble weights for different source sensors.

Source Sensor	S1	S2	S3	S4	S5	S7
Ensemble Weight	0.07	0.16	0.15	0.13	0.22	0.27

Table 3.12: Ensemble direct transfer
MAE=28, Acc=0.55

Ground Truth	**Predictions**						
	L1	**L2**	**L3**	**L4**	**L5**	**L6**	
L1	**470**	242	15	1	3	0	0.64
L2	18	**264**	136	12	2	2	0.61
L3	1	7	**120**	119	46	4	0.40
L4	1	0	5	**34**	109	4	0.22
L5	0	0	7	7	**73**	90	0.41
L6	0	0	0	0	0	**62**	1.00
	0.96	0.51	0.42	0.20	0.31	0.38	
	Precision						

Recall

Table 3.13: ensemble TCA transfer
MAE=14, Acc=0.76

Ground Truth	**Predictions**						
	L1	**L2**	**L3**	**L4**	**L5**	**L6**	
L1	**568**	153	7	2	1	0	0.78
L2	24	**337**	68	3	2	0	0.78
L3	1	25	**231**	31	6	3	0.78
L4	1	0	27	**93**	30	2	0.61
L5	0	2	13	17	**133**	12	0.75
L6	0	0	0	0	10	**52**	0.84
	0.96	0.65	0.67	0.64	0.73	0.75	
	Precision						

Recall

Table 3.14: ensemble NZ-ICT transfer
MAE=21, Acc=0.61

Ground Truth	**Predictions**						
	L1	**L2**	**L3**	**L4**	**L5**	**L6**	
L1	**678**	49	4	0	0	0	0.93
L2	131	**289**	14	0	0	0	0.67
L3	2	183	**106**	4	2	0	0.35
L4	1	7	124	**20**	1	0	0.13
L5	0	9	41	88	**39**	0	0.22
L6	0	0	0	3	56	**3**	0.05
	0.83	0.54	0.37	0.17	0.39	1.0	
	Precision						

Recall

Table 3.15: ensemble ICT transfer
MAE=8, Acc=0.86

Ground Truth	**Predictions**						
	L1	**L2**	**L3**	**L4**	**L5**	**L6**	
L1	**666**	65	0	0	0	0	0.91
L2	33	**372**	29	0	0	0	0.86
L3	1	33	**243**	20	0	0	0.82
L4	0	2	18	**115**	18	0	0.75
L5	0	1	9	17	**142**	8	0.80
L6	0	0	0	2	5	**55**	0.89
	0.95	0.79	0.81	0.75	0.86	0.87	
	Precision						

Recall

to 8 and accuracy to 0.86, as shown in Table 3.15, which is significantly better than the other methods. The results show that in general, multi-source calibration transfer outperforms the corresponding single-source calibration transfer, and our ensemble ICT also outperforms the other methods.

Validation of Linear Transformation Function *F*. To validate the linearity of the transformation function, we deploy an extra low-cost sensor $S6'$ near the reference sensor $R6$. Figure 3.7 illustrates the transformation function F derived from direct standardization [FFGG+16]. The diagonal elements represent the linear relationship between the same features of the two sensors, while the non-diagonal elements can be seen as the linear relationship between different features, i.e., cross-features or cross-sensitivity [LCL+12]. As is shown, the diagonal elements are more dominant than the non-diagonal ones, indicating that F can be approximated as a linear function.

Impact of Spatial Prediction. In Eq. (3.3), we add the second term $d_c[Y_t^\tau, C_s(F(X_t^\tau))]$ to the optimization objective to improve the performance

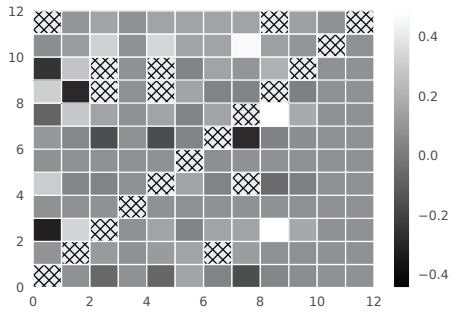


Figure 3.7: Transformation matrix derived from direct standardization. Blocks marked with nets are weights greater than 0.5. The diagonal elements are more dominant than the non-diagonal ones.

of ICT via spatial prediction. Figure 3.8a shows the MAEs of ICT with and without the help of spatial prediction. For comparison, we also plot the MAEs using direct transfer. ICT without spatial prediction already decreases the MAEs from 28 to 11 compared with direct transfer. ICT with the help of spatial prediction can further reduce the MAEs by 12%.

We are also interested in how many spatial prediction results are necessary to improve the performance of ICT. We select different fractions of predicted $PM_{2.5}$ concentration level $Y_t^{'}$ by changing the threshold τ and forming Y_t^{τ}. Note that the smaller τ is, the higher average accuracy of Y_t^{τ} is. By setting $\tau = 100$, we select around 25% of all predicted $PM_{2.5}$ concentration level in the location where $S6$ is installed, which yields an overall accuracy of 0.96. Then we select different sizes of Y_t^{τ} and evaluate the MAEs of ICT.

Figure 3.8b shows the MAEs to transfer calibration parameters from different sources to $S6$ using ICT with different sizes of Y_t^{τ}, ranging from 5% to 25% (the percentage represents $||Y_t^{\tau}||/||Y_t^{'}||$). When the size of Y_t^{τ} increases from 5% to 15%, the MAE decreases. However, the decreasing of MAE stops after that. This suggests 15% of the most accurate predictions from $Y_t^{'}$ is sufficient.

Visualization of Transferred Measurements. To better understand the cause behind the varying results of four approaches, we use principal component analysis (PCA) to illustrate the difference between the transformed measurements of source sensor $S7$ and target sensor $S6$, i.e., X_s and $F(X_t)$. In order to enable a visualization of the results, we choose the two largest components and show them in a 2 dimensional visualization in Figure 3.9. Since we have already empirically proven that the distribution of ground truth in both source and target locations are similar in Figure 3.5, i.e., $p(Y_s) \approx p(Y_t)$, and the same calibration model is applied on both the transformed measurements of source and target sensor, we can reasonably assume that the overlapping area of the largest two components is positively correlated to the calibration transfer

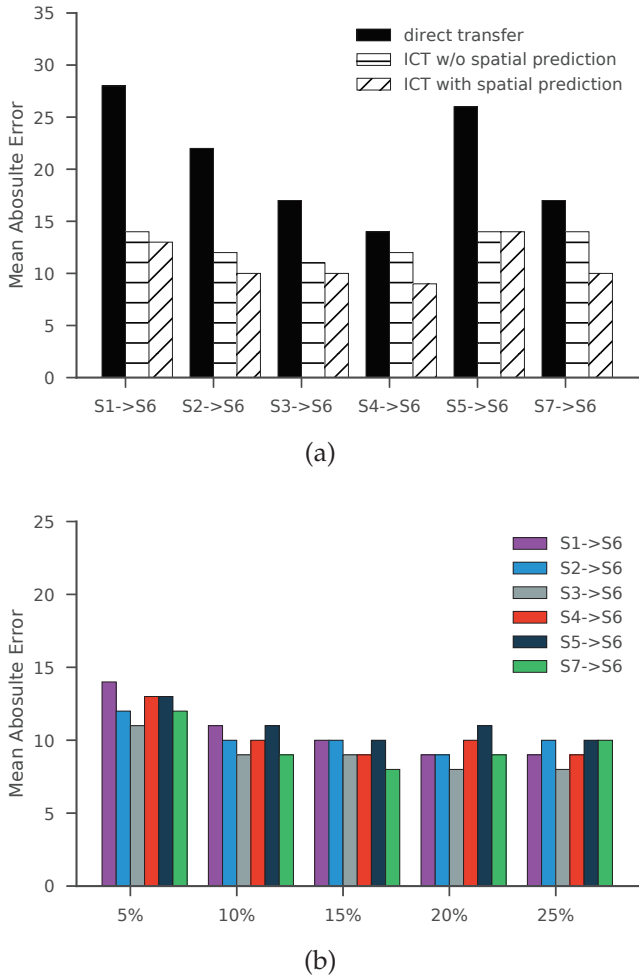


Figure 3.8: Impact of spatial prediction. (a) Performance of ICT with and without spatial prediction (d_c). (b) Performance using different size of Y_t^τ. The percentage represents $\|Y_t^\tau\|/\|Y_t'\|$.

accuracy.

In direct transfer, since no transformation is applied, i.e., $F(X_t) = X_t$, the PCA results directly represent the original measurements X_s and X_t, as shown in Figure 3.9a. There are obvious shifts and misalignments shown in the figure, which could explain the reason why the performance of direct transfer is limited. TCA tries to correct the shifts between the measurement features by applying dimension reduction, i.e., removing less important feature dimensions. As shown in Figure 3.9b, the two largest PCA components of $\Phi(X_s)$ and $\Phi(X_t)$ have a larger overlapping area, which could explain the reason of accuracy improvement. Instead, ICT does not reduce dimensions and use the original measurement features to find the linear transformation. Figure 3.9d shows that the overlapping area between the components of X_s and $F(X_t)$ is much larger compared to direct transfer and TCA method. Notice that if only the near

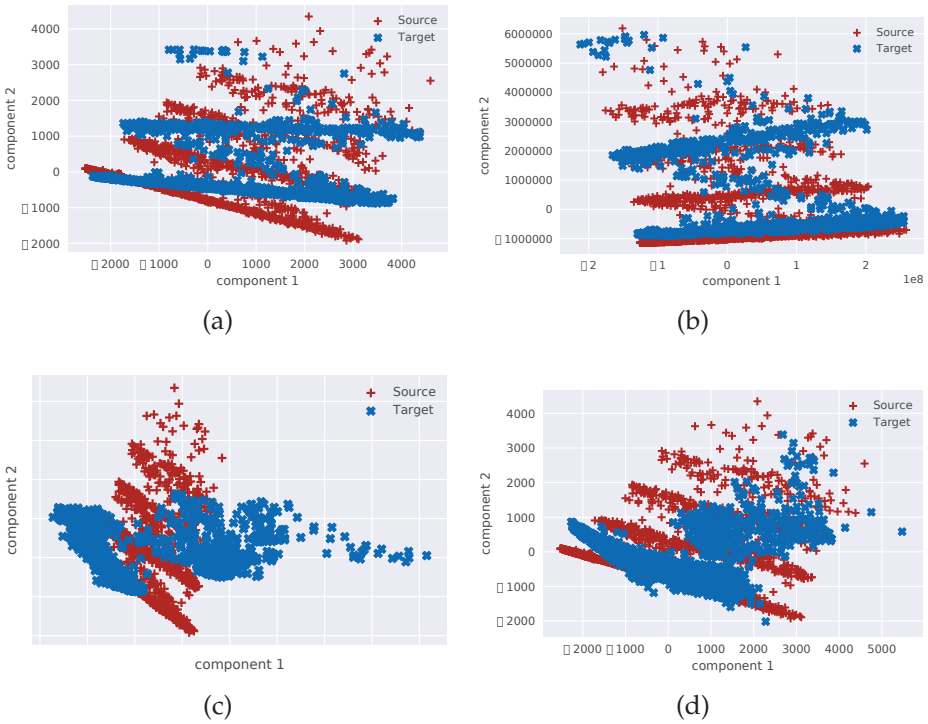


Figure 3.9: PCA visualization between the transformed measurements of source sensor *S7* and target sensor *S6*. (a) Blue dots represent the largest two PCA components of X_t from *S6*, while red dots represent X_s from *S7*; (b) TCA transfer method; (c)NZ-ICT transfer method; (d) ICT transfer method.

zero data is used (NZ-ICT), the overlapping area is smaller and has a visible shift compared with the full ICT, as shown in Figure 3.9c.

3.5.4 Case Study: Pollution Source Location Inference

Due to lack of ground truth of $PM_{2.5}$ concentrations from co-located reference stations, it is difficult to evaluate the performance of ICT on a sensor deployed at an arbitrary location in Beijing. Alternatively, this subsection aims to *indirectly* assess the performance of different algorithms via a case study, in which calibration parameters are transferred from a single source sensor to tens of target sensors in arbitrary locations within a certain range. Specifically, we apply different calibration transfer methods on the raw sensor readings, and compare their performance to infer the locations of pollution sources from the calibrated sensor readings. The rationale is that pollution concentrations should be high at locations close to the pollution sources. Therefore an intuitive way to locate pollution sources is to firstly generate a heat map of air pollution concentration from the sensor measurements, then find locations/areas where the concentration peaks. The accuracy of pollution source localization is correlated to the accuracy of the calibrated sensor measurements, thus

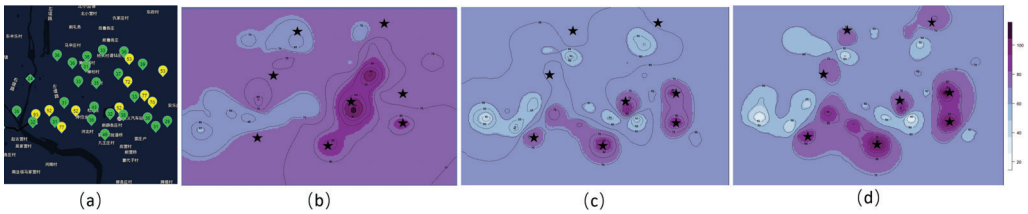


Figure 3.10: (a) Locations of 30 low-cost sensors within a 5km × 5km area around S6 and heat maps generated with measurements of the 30 sensors calibrated by using (b) direct transfer, (c) TCA and (d) ICT. The ground truth pollution source locations are marked by stars.

an indirect assessment of the effectiveness of different sensor calibration algorithms.

We conduct a case study in a 5km × 5km area around S6, where 30 low-cost sensors (including S6) are deployed. Figure 3.10-(a) shows the locations of the 30 low-cost sensors. The squared spot is the location of the high-cost reference station co-located with S6. We focus on this area because we have access to the ground truth locations of the pollution sources of $PM_{2.5}$ within this area, which is generally inaccessible for other areas in Beijing. We use the pollution source locations as the ground truth for pollution source location inference. We perform calibration transfer using three methods: *(i)* direct transfer, *(ii)* TCA and *(iii)* ICT. Then we average the calibrated sensor measurements over one month and use the Gaussian process regression model in [CLL+14b] to generate the heat map of the area. The locations of the ground truth pollution sources within this area are marked by stars. Ideally, the peaks (high concentration locations) in the heat map should match with the pollution source locations.

Figure 3.10-(b), Figure 3.10-(c) and Figure 3.10-(d) show the heat maps generated by applying direct transfer, TCA and ICT for sensor calibration. By comparing the highly polluted locations in the heat maps to the ground truth pollution source locations, we observe that the heat map generated by sensor measurements calibrated by direct transfer is able to correctly locates three out of the eight pollution sources, while five are located when applying TCA. With ICT, however, all eight pollution sources are correctly located. These results indicate that ICT outperforms direct transfer and TCA in calibration accuracy on these 30 sensors.

3.6 Summary

The low-cost air quality sensor network introduced the opportunity for large scale deployments with high spatial coverage. However, how to maintain the sensor accuracy and reliability in the wild remain an unsolved challenge. In Chapter 2, we explore calibrating sensor measurements with a novel many-to-many scheme. Using the proposed

SensorFormer method, low-cost sensor readings can be calibrated. However, as illustrated in Section 3.3, transferring the calibration model to locations without access to ground truth will lead to huge errors.

In this chapter, we propose In-field Calibration Transfer (ICT), a calibration scheme that transfers the calibration parameters of sensors with access to references (source sensors) to those without access to references (target sensors). It is challenging to derive such a transformation between the source and target sensors installed at different locations because their measurements are unsynchronized. On observing that *(i)* the distributions of ground truth in both source and target locations are similar and *(ii)* the transformation is approximately linear, ICT learns the transformation based on the similarity of distributions with a novel optimization formulation. The performance of ICT is further improved by using spatial prediction of air quality level as an aid for calibration transfer task, and using ensemble techniques to enable multi-source calibration transfer. Experiments show that ICT is able to provide approximately equal calibration performance as if the target sensors have direct access to references. We believe ICT notably increases the usability of large-scale air pollution monitoring deployments. Those calibrated sensor readings can be safely used for the analysis and predictions tasks introduced in the following chapters.

The novel calibration method SensorFormer in Chapter 2 and the calibration transfer method ICT introduced in this chapter are orthogonal and can be used in parallel to improve the air quality sensor network accuracy. i.e., SensorFormer can be used to learn the accurate calibration model at locations with access to references, then ICT can transfer this model to arbitrary locations. To this end, we can safely argue that the sensor network is accurate sufficient to generate reliable air quality maps or be used for analysis tasks.

One future extension of this work is to explore the possibility of conduct ICT on larger scale scenarios e.g., inter-city calibration transfer. We can also extend ICT for moving sensors by augmenting the measurements of the moving sensor collected from multiple locations to perform multi-source multi-target calibration transfer.

4

Urban Air Quality Map Generation for Downscaled Deployments

It has been proven that dense deployments of commodity air quality sensors can provide spatially-resolved urban air quality data in real time. However, long-term operation of a dense sensor deployment requires enormous maintenance efforts and costs. To guarantee the accuracy and reliability of the sensing measurements, a novel on-site calibration method called SensorFormer is introduced in Chapter 2, which acquires the best performance. Furthermore, a calibration transfer model called ICT is presented in Chapter 3 to assure that the on-site calibration model can be safely used in target sensors (without access to reference). The sensor measurements are sufficiently accurate for the following analysis tasks. Nevertheless, sensor deployments still pose a big challenge because of their sheer number. In this chapter, we propose MapTransfer, a method for accurate air quality map generation for downscaled deployments. 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.

4.1 Introduction

The deployment of dense air quality sensors has been reported both by academia and industries [CLL+14b, GDG+16, HSW+14, XCL+16], only a small number remain active after a certain time. The tedious maintenance and high costs involved in dense deployments are two major reasons for their short lifespans. One practical approach to reduce the cost is to downscale the deployment [KMM+15]. However, as we illustrated in Section 1.2, naïvely downscaling the sensor deployment (e.g., to 1/3 or 1/4) will dramatically increase the air quality map generation error.

To improve the accuracy of air quality maps generated with only real-time measurements from a sparse downscaled deployment, one generic solution framework is *map generation transfer*, which augments the current measurements with appropriate historical data collected from the initial dense deployment for map generation. 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 very similar to that at some time point in the history.* Therefore it may improve the accuracy of the air quality map generated with the sparse deployment by *properly transferring knowledge* of air quality in this region from the historical data obtained in the dense deployment. In environmental science, a popular approach to transfer knowledge between Gaussian process modeled environmental phenomena is Multi-Output Gaussian Process (MOGP) [LCO18], which learns the air quality distribution on both the real-time measurements from downscaled deployment and historical measurements from initial dense deployment. MOGP-based knowledge transfer schemes have proven effective in many applications such as weather estimation [ORR+08], soil heavy metal prediction [ZY17], groundwater depth estimation [AS08].

However, designing an MOGP-based map generation transfer scheme suited for air quality is challenging. A prerequisite for MOGP-based methods to work on air quality map generation transfer is that only historical data from the dense deployment strongly correlated to the current data in the downscaled deployment are selected, so as to enable positive knowledge transfer for the underlying phenomenon ($PM_{2.5}$ concentrations in our case). Due to the complexity and dynamics of urban $PM_{2.5}$ concentrations, it is non-trivial to select the best measurements for transfer. On the one hand, previous studies [ORR+08, G+97, ZY17] adopt simple unsupervised selection criteria such as Root Mean Square Error (RMSE) or Correlation Coefficient (CORR). As we show in real-world air quality measurements, these criteria often lead to negative transfer, which refers to the phenomenon that the model hurts the performance in the target domain[PY10b], in the generated air quality map. On the other hand, we observe that the correlation between measurements in

the initial and the downscaled deployments is not homogeneous over the entire monitoring region. Such spatial locality in correlation indicates that transferring knowledge at the scale of the entire region as prior research may also impair the performance of map generation transfer.

To address the challenges above, we propose MapTransfer, a new MOGP-based scheme for accurate air quality map generation transfer. To avoid negative transfer due to improperly selected measurements from the dense deployment, MapTransfer adopts learning-based instance selection (LIS). It extracts a rich set of features from both $PM_{2.5}$ measurements as auxiliary meta data sources e.g., meteorological information, and exploits an artificial neural network to select the best instances for map generation transfer. To exploit spatial locality during map generation transfer, MapTransfer utilizes sub-region selection (SRS) to split the whole region into sub-regions and search for the best transfer option for each sub-region. Evaluations on measurements collected from a real-world sensor deployment show that MapTransfer is able to reduce the overall MAE by 45.9%, compared with the air quality maps generated with data merely from the downscaled deployment.

The contributions of this chapter are summarized as follows.

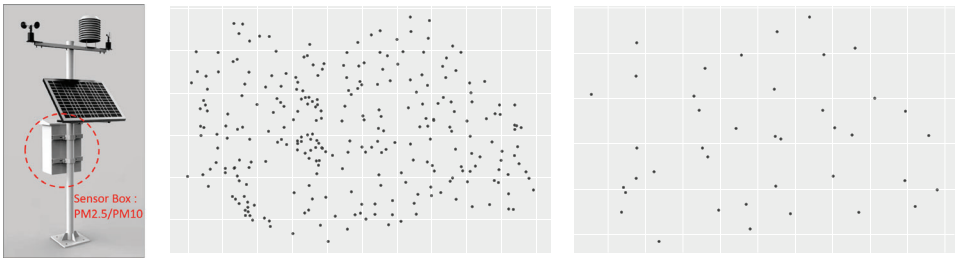
- We propose MapTransfer, the first practical urban air quality map generation scheme for downscaled sensor deployments by transferring knowledge and augmenting historical data from the initial dense deployment.

- 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 [HSW+15, RTMH18], as well as data mining applications [BJZOV17a, LCCC17].

In the rest of this chapter, we formally define our problem and introduce our datasets in Section 4.2. Then we present an overview of our MapTransfer method in Section 4.3 and explain its core modules in Section 4.4, Section 4.5 and Section 4.6. The evaluations are shown in Section 4.7 and we discuss the limitations of our method in Section 4.8. We finally conclude in Section 4.9.

4.2 Preliminaries

In this section, we formally define the problem of air quality map generation from downscaled sensor deployments, and then introduce



(a) Sensor (b) Dense Deployment (c) Sparse Deployment

Figure 4.1: 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.

the datasets collected from a large-scale $PM_{2.5}$ sensor deployment that will be used throughout this chapter.

4.2.1 Problem Definition

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

Definition 4.1 (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 4.2 (dense deployment)**.** *A dense deployment is a sensor network used during the initial phase. All sensors are activated in a dense deployment.*

Definition 4.3 (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 4.4 (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.*

4.2.2 Datasets

We collect measurements from a large-scale $PM_{2.5}$ monitoring deployment consisting of 260 low-cost sensors (see Figure 4.1a) 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 Figure 4.1b and Figure 4.1c with a resolution of 1 $km \times 1$ km, which is denoted as *grid* in this chapter. 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*.

4.3 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 Figure 4.2.

- **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 Section 4.4. 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 Section 4.5.

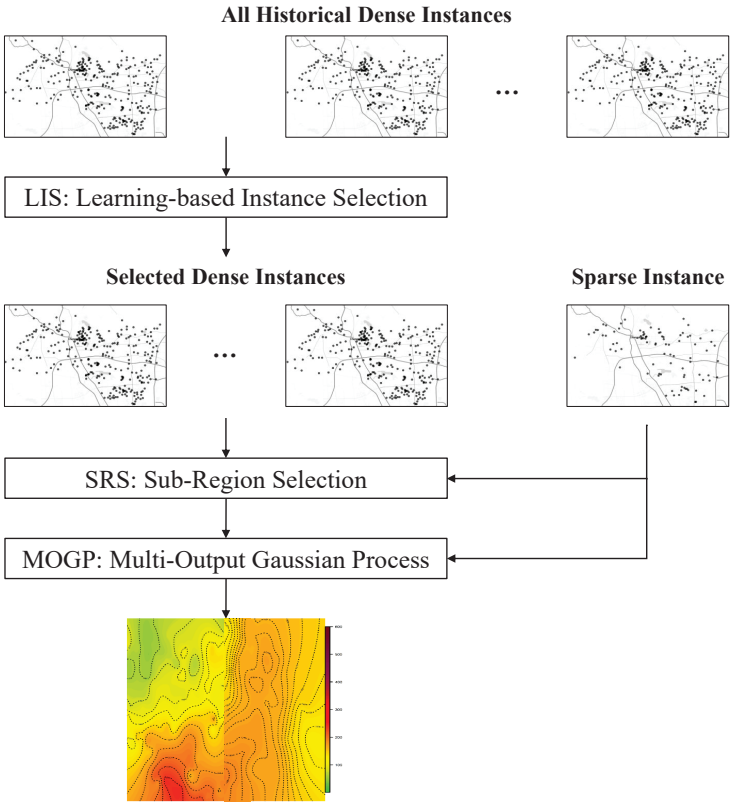


Figure 4.2: Workflow of MapTransfer.

- **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 Section 4.5.

- **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 Section 4.6.

In real-world situation, the constantly changing environment provides challenges for MapTransfer to be 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 chapter.

4.4 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.

4.4.1 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) [Ras04] proves effective to model and learn this mapping when measurements from a dense deployment are available [CLL+14b, CLL+14a]. They assume that the function f is distributed as a GP with mean function m and covariance function k [Ras04], which can be written as:

$$f \sim \mathcal{GP}(m, k) \tag{4.1}$$

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 ($1\ km \times 1\ km$). Finally we have an air quality map generated by the measurement 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 Section 4.2.2 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 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 [BJZOV17a]. The example shows that air quality maps generated with merely sparse instances have limited accuracy and augmenting historical dense instances is necessary.

4.4.2 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 [LCO18]. 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:

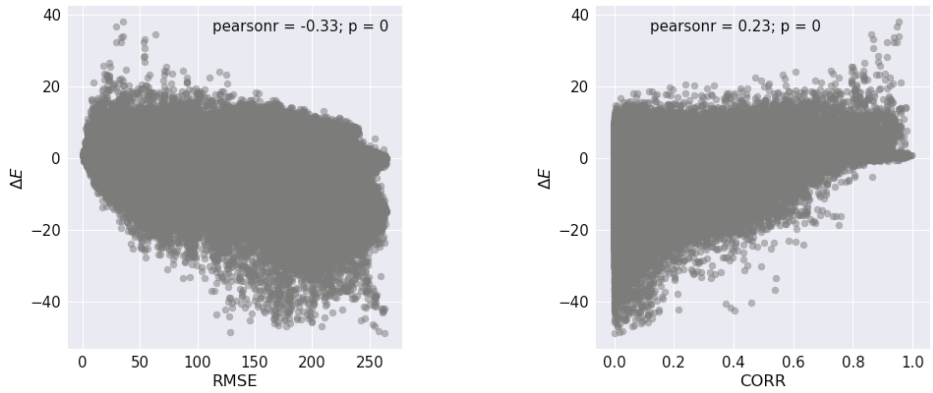
$$\begin{bmatrix} f_1 \\ f_2 \end{bmatrix} \sim \mathcal{GP}\left(\begin{bmatrix} m_1 \\ m_2 \end{bmatrix}, \begin{bmatrix} k_{1,1} & k_{1,2} \\ k_{2,1} & k_{2,2} \end{bmatrix}\right) \tag{4.2}$$

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.

4.5 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.



(a) (b)

Figure 4.3: 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.

4.5.1 Dense Instance Selection via Unsupervised Criteria

RMSE and correlation coefficient (CORR) are two commonly used unsupervised criteria for instance selection in environmental science [Ros07, ZY17, ORR+08, AS08]. 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.

Figure 4.3 plots the relationship between transfer gain ΔE and the value of used instance selection criterion, i.e., RMSE (Figure 4.3a) or CORR (Figure 4.3b). 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.

4.5.2 Dense Instance Selection via Supervised Learning

Figure 4.4 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

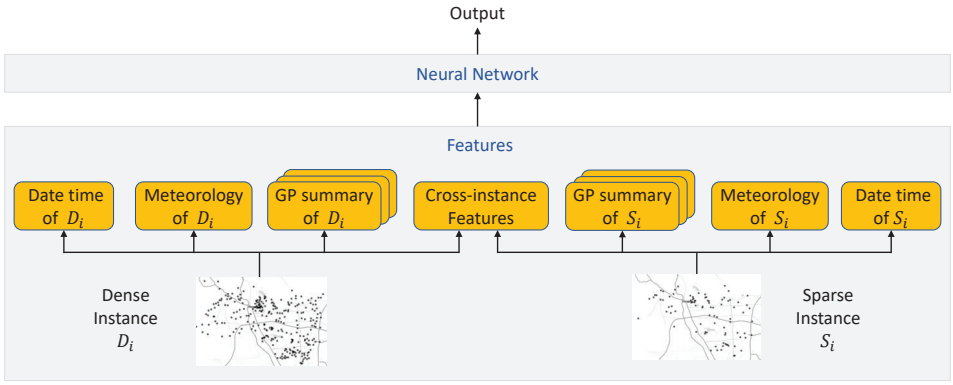


Figure 4.4: An illustration of learning-based dense instance selection (LIS).

Table 4.1: 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

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.

4.5.2.1 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 4.1 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 [ZCWY14]. 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 Figure 4.3a).

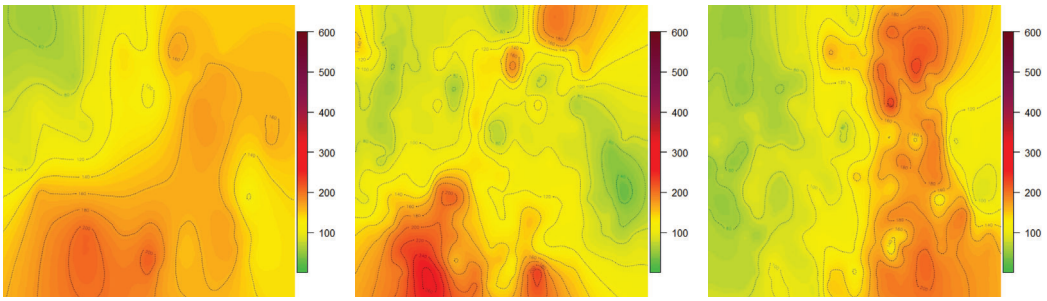
- *GP summary feature vectors F_{GP}.* Since our end goal is to improve the accuracy of the generated air quality maps, it is reasonable to utilize the parameters of the maps i.e., parameters of the Gaussian processes as features for instance selection. We choose all the optimized GP parameters as F_{GP} which includes *nug_psill*, *nug_range*, *nug_kapple*, *sph_psill*, *sph_range*, *sph_kapple* [Ros07, Mat63]. 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 [Mur12] 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 [Mur12] 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*.

4.5.2.2 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^{train}_{MOGP} is the MAE of predicting



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

Figure 4.5: 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.

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.

4.6 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.

4.6.1 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. Figure 4.5a 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 Figure 4.5b and Figure 4.5c, 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 Figure 4.5b looks very similar to the same left half of Figure 4.5a. Also the right half of Figure 4.5c looks similar to the right half of Figure 4.5a. 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

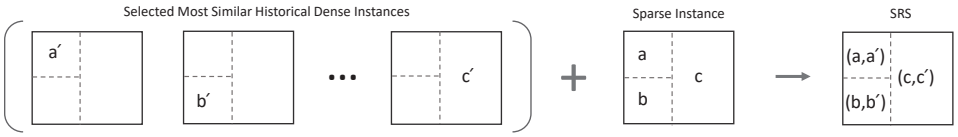


Figure 4.6: Proposed Sub-Region Selection method. Suppose the sparse instance is split into three zones (a, b, c) according to a **splitting point**, SRS method tries to find the best match subset in dense instances with the same splitting grid. Then, subset MOGP could be done in each subset zone and produce the overall result.

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.

4.6.2 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.

4.6.2.1 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.

4.6.2.2 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

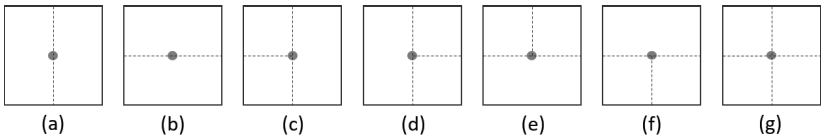


Figure 4.7: Potential different splitting methods to split the whole region to 2, 3 or 4 sub-regions.

monitoring region into multiple adjacent rectangular sub zones, as shown in Figure 4.7. 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 [M+14] to find the best splitting point and best splitting method, which yield the lowest overall CV-MOGP error.

4.6.2.3 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 Figure 4.7. 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:

$$\min_{(l_x,l_y),m} \sum_{i=1}^{N_s} \min_{j} E^{CV}_{i,j} \tag{4.3}$$

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 Figure 4.2.

4.7 Evaluation

This section presents the evaluation of MapTransfer. We first introduce the experiment setup (Section 4.7.1) and then present the overall performance (Section 4.7.2). Finally we show the effectiveness of each module (Section 4.7.3 and Section 4.7.4).

4.7.1 Experiment Setup

Datasets and Metrics. We evaluate the performance of different map generation transfer schemes using the same setting and datasets in Section 4.2.2. 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 (MAE), because MAE is used in various air quality related research, including sensor calibration, spatial interpolation [CLL+14a], temporal prediction [LBC19] and data mining [LCCC17].

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 4.1) 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 [GBC16]. All the codes are implemented in python and the experiments were conducted with a Linux machine with 32 cores.

4.7.2 Overall Performance

Table 4.2 shows the overall performance of different map generation transfer methods. We are also interested in the performance of these

Table 4.2: Overall performance and performance in case of heavy pollution ($PM_{2.5}$ concentration > 150 ug/m^3) of different map generation transfer methods.

	Overall Performance		Performance on Heavy Pollution	
Method	**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**

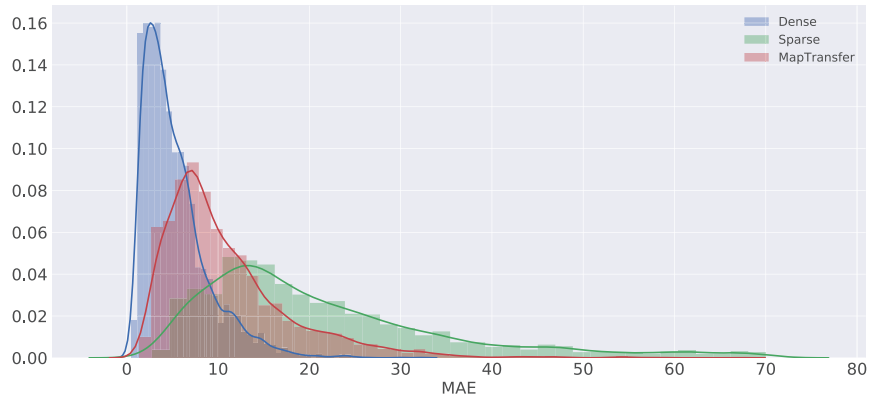
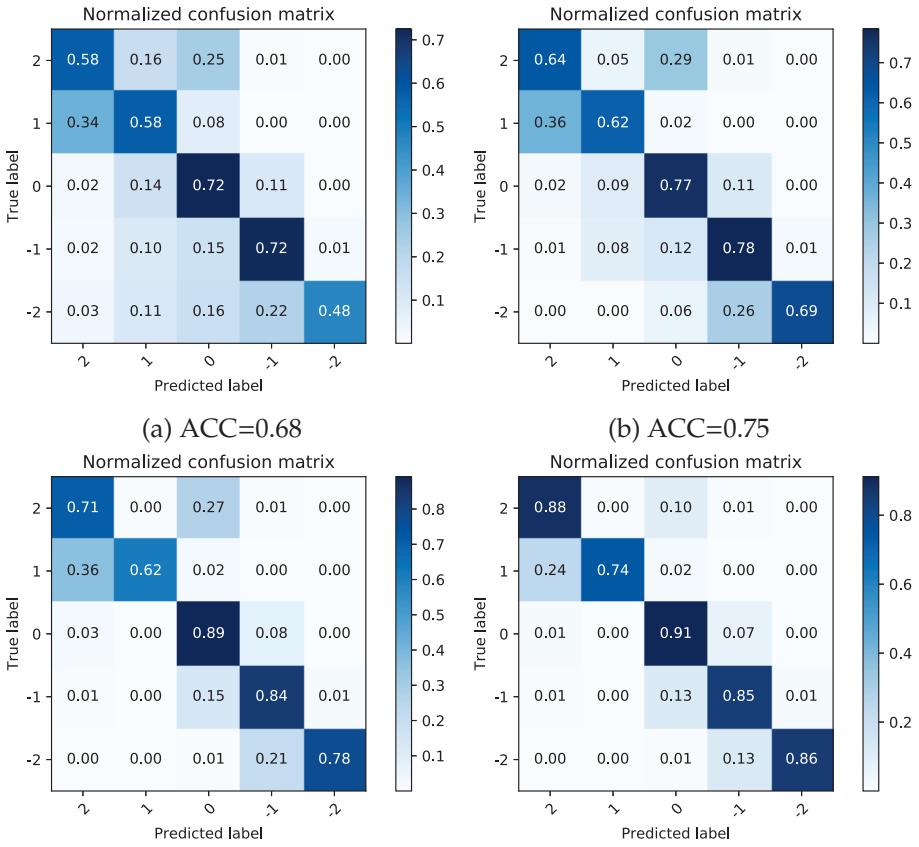


Figure 4.8: Histograms of MAEs of air quality maps generated using different transfer methods.

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.

Figure 4.8 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 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.



(a) ACC=0.68 (b) ACC=0.75

(c) ACC=0.83 (d) ACC=0.88

Figure 4.9: 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.

4.7.3 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 .

Figure 4.9 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 Section 4.5. 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.

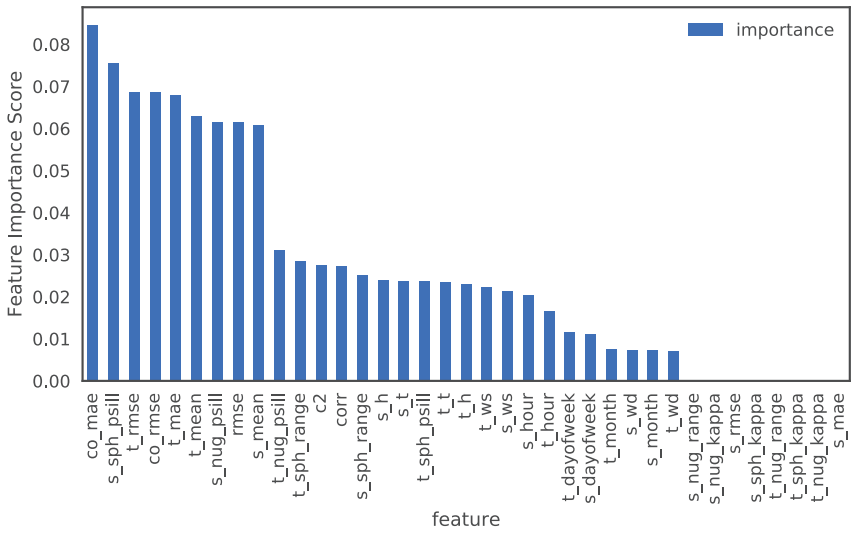
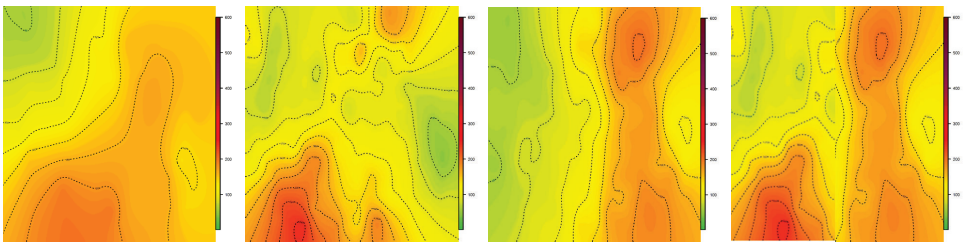


Figure 4.10: Feature importance for predicting the transfer gain ΔE.



(a) GP(Sparse) (b) GP(Dense-L) (c) GP(Dense-R) (d) GP(Dense-S)

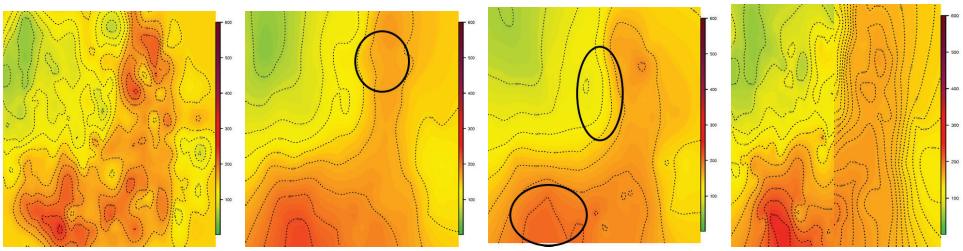
Figure 4.11: Air quality maps generated by GP with (a) a sparse instance (denoted as Sparse); (b) one best dense instance selected by LIS (denoted as Dense-L); (c) another best dense instance selected by LIS (denoted as Dense-R); and (d) a dense instance stitched by SRS (denoted as Dense-S).

Figure 4.10 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 4.1) 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.

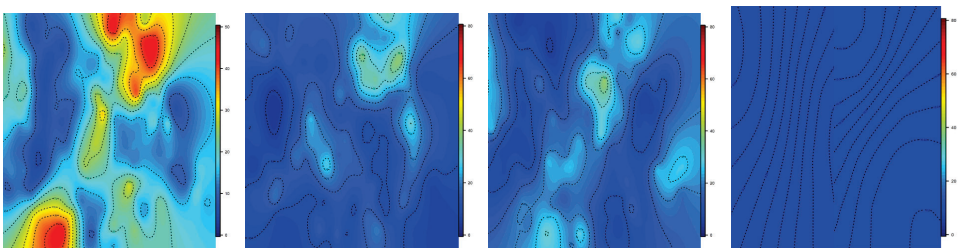
4.7.4 Effectiveness of Sub-Region Selection

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



(a) Ground Truth (b) MOGP(S,D-L) (c) MOGP(S,D-R) (d) MOGP(S,D-S)

Figure 4.12: (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.

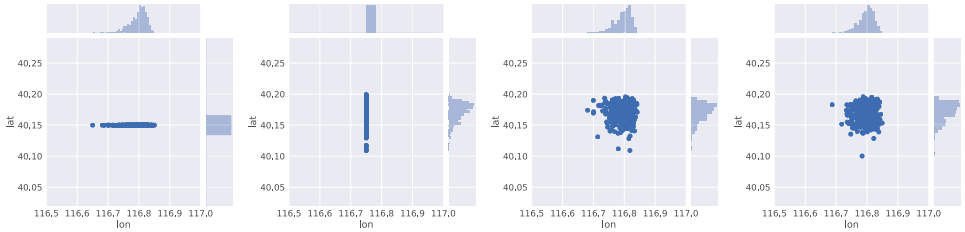


(a) GP(S) (b) MOGP(S,D-L) (c) MOGP(S,D-R) (d) MOGP(S,D-S)

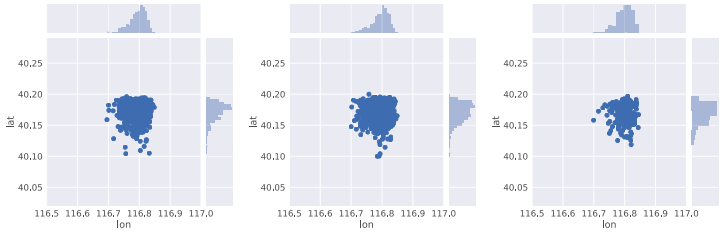
Figure 4.13: Errors of air quality maps generated by (a) GP with Sparse; (b) MOGP with Sparse and Dense-L; (c) MOGP with Sparse and Dense-R; (d) MOGP with Sparse and Dense-S.

the sub-regions selected by SRS.

Figure 4.11a shows an air quality map generated by GP using a sparse instance collected at 4:00 a.m. on February 23rd, 2019. Figure 4.11b and Figure 4.11c plot the maps generated by GP using the two best dense instances selected by LIS (denoted as Dense-L and Dense-R, respectively). Figure 4.11d 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. Figure 4.12b, Figure 4.12c and Figure 4.12d 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 Figure 4.12a, 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



(a) 21.8% (b) 25.5% (c) 8.1% (d) 9.0%



(e) 14.4% (f) 17.1% (g) 4.1%

Figure 4.14: 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.

Figure 4.13, 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-error areas in Figure 4.13b and Figure 4.13c. 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 Figure 4.13d). 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 Figure 4.14. 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 methods described in Section 4.6, the methods which splits the region to 2 sub-regions (Figure 4.14a and Figure 4.14b) account for 47.3% of all the sparse instances. The methods which split the region to 3 sub-regions (see Figure 4.14c, Figure 4.14d, Figure 4.14e, Figure 4.14f) 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 (Figure 4.14g) only has 4.1% shares among the splitting results.

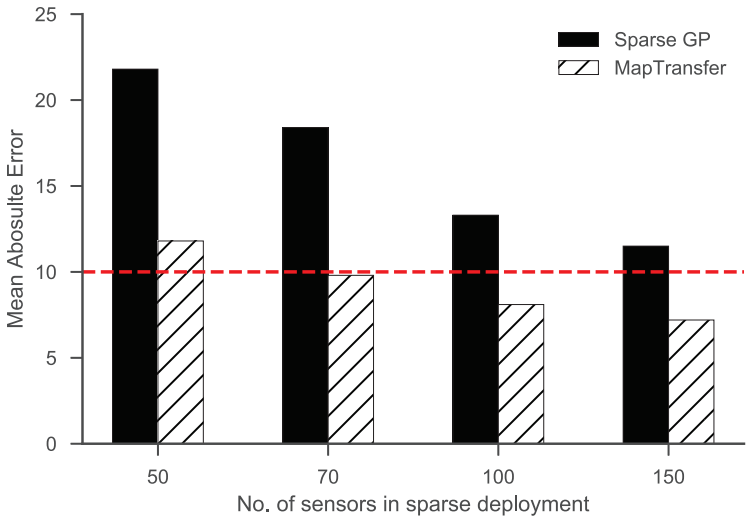


Figure 4.15: Impact of number of sensors in sparse deployment on air quality map generation accuracy.

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.

4.7.5 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 $MAE < 10$.

Figure 4.15 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.

4.8 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 [CDL+19a, LLZ+18, ZLH13a]. 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 [CLL+14a]. 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 [CDL+19a, LLZ+18, ZLH13a] 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 [HLZ15, YZB+18, NAB20]. 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 [ZTXF19] can be applied before inputting the data into our method.

4.9 Summary

In this chapter, 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.

An accurate and cost-effective sensor network is now available by applying the calibration method in Chapter 2, Chapter 3 and the sensor downscale approach in this chapter. Those features ensure the usability and scalability of the sensor network, which is critical for long term deployments. In the next chapters of this thesis, we aim to answer the following research questions: what can be obtained with the accurate and reliable air quality sensor network? Specifically, an accurate prediction method will be presented in Chapter 5 and a data-driven immediate cyber-physical response system will be introduced in Chapter 6.

5

Tracking Pollution Transfer for Accurate Air Quality Prediction

So far, we have presented a novel calibration method called SensorFormer (Chapter 2) to derive the underlying time series alignment patterns and generate state-of-art calibration performance. To guarantee the calibration results in all sensor locations, especially those without access to reference, ICT (Chapter 3) is proposed to enable the calibration model transfer from on-site location to target locations. The sensor measurement after the above pipelines can be viewed as accurate and reliable to be used for analysis. Furthermore, MapTransfer (Chapter 4) is introduced to improve the map generation accuracy for downscaled deployment, which significantly reduces the deployment cost while keeping reasonable accuracy. A follow-up question then arises: *Could we analyze the data and use the derived knowledge to reduce the pollution?* We positively answer this question and present the pollution suppression research results in city-level (in this chapter) and street level (Chapter 6). In this chapter, we design a data-driven method to characterize the pollution transfer between cities, which can be used to boost the performance of air quality prediction. Specifically, accurately predicting air quality, especially its sudden changes, is highly valuable for citizens and governments to make personal and local decisions, design intelligent policies and control pollution at minimal cost. However, none of the existing data-driven methods achieves sufficient prediction accuracy for time intervals of sudden pollution change due to inability of existing models to take into account pollution propagation between different areas caused by air mass movement. For the first time, we consider pollution transfer in the context of short-term air quality prediction and propose to use air flow trajectory data, widely used in environmental sciences, to represent pollution transfer patterns between different locations. By learning trajectory representations, measurement location embedding vectors, and

interrelationships between local weather at relevant locations, we propose a new attention based seq2seq model to track pollution propagation for accurate air quality prediction. We evaluate our model on datasets from Beijing area and compare the results to several state-of-the-art baselines. Experiments show that the proposed approach can successfully capture pollution transfer patterns between different sites in the area. Our model outperforms all the baselines and decreases prediction errors by 9.6% to 22.4%. The method allows interpreting prediction results visually and analytically, and provides tools for making pollution reduction strategies.

5.1 Introduction

Air quality is of vital importance to human health. Medical studies have shown that $PM_{2.5}$ (particulate matter of diameter less than 2.5 micron) can be easily absorbed by the lungs, and prolonged exposure to its high concentrations may lead to respiratory impairments, blood diseases and neuro developmental disorders, such as autism, attention deficit disorders and cognitive delays [CSSM+19]. Air pollution is also found to have a negative effect on the cognitive functions in elderly adults [RSS+09].

In the context of current pandemic, recent studies show that long-term exposure to $PM_{2.5}$ and NO2 increases our susceptibility to SARS-CoV-2 [HLL+20, LPP+20] and contributes to higher fatality rates [Oge20, WNS+20]. There is also a worrying evidence that the virus can be found in outdoor particulate matter [SPDG+20]. Therefore, accurate air quality prediction, especially forecasting $PM_{2.5}$ concentrations, is an effective way of protecting public health by providing an early warning against harmful air pollutants [BWML18]. For example, air quality in Beijing can sometimes change from a good to an unhealthy level within a few hours due to pollution transfer from sources outside of the city, which is referred to as air quality sudden change. Being able to predict such sudden changes is vital to inform people and governments for decision making, but very difficult to achieve due to sparsity of air quality monitoring observations and the underlying complex evolving environment [ZYL+15b].

Some existing works [ZZZ+17, DWC+19] already highlighted the importance of pollution transfer from surrounding areas, the phenomenon we refer to as pollution transfer. For example, pg-Causality [ZZZ+17] uses frequent pattern mining and Bayesian learning to identify spatio-temporal causal pathways for air pollutants of Beijing. The results show that the surrounding cities play important roles in propagating pollution, as shown in Figure 5.1-(a). One naïve approach to encode this knowledge is to consider the air quality readings in the surrounding cities and use them in the prediction task. However, our experimental results described in Section 5.5 show that naïvelly concatenating air quality readings from the surrounding stations is not helpful and may even worsen the final

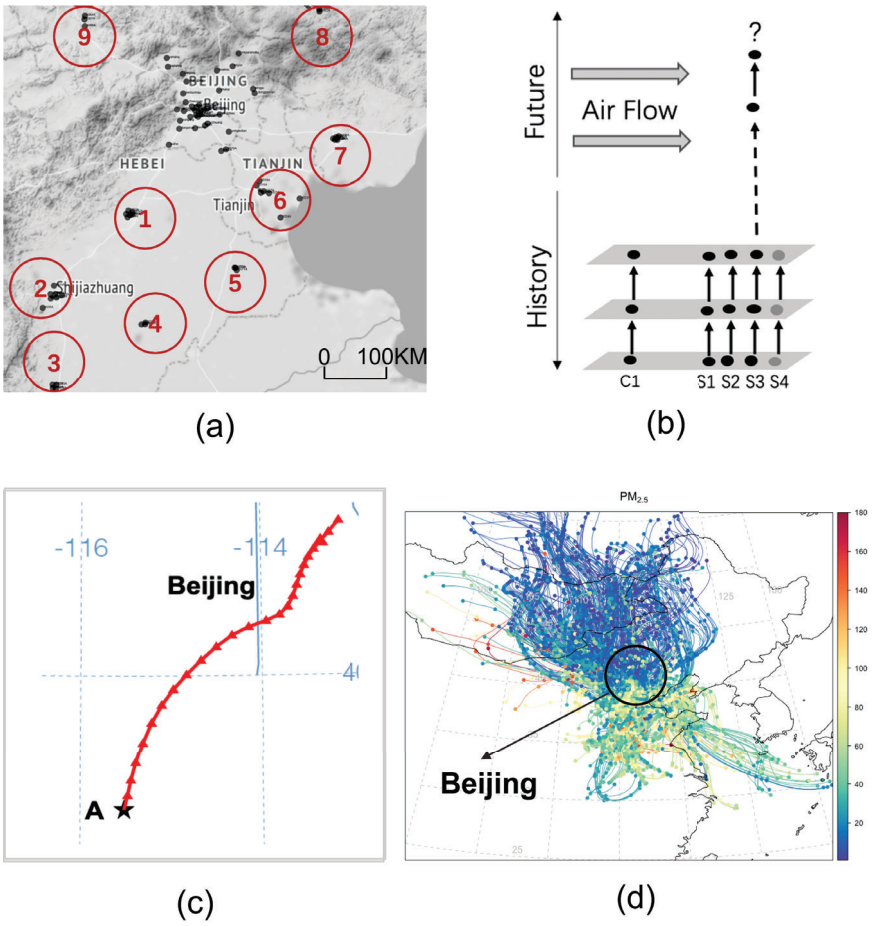


Figure 5.1: Pollution transfer and the role of air flow trajectories. (a) 9 main cities around Beijing involved into air pollution transfer. (b) Schematic view of a possible pollution transfer from a remote city $C1$ to a group of locations $S1-4$ in the same city S. (c) An air flow trajectory. (d) Aggregated air flow trajectories over a year colored by measured $PM_{2.5}$ values.

prediction.

In environmental science, HYSPLIT [SDR+15] is widely adopted to identify regional pollution sources [KVL+11] and propagation pathways [MC08]. Based on meteorological data, HYSPLIT attempts to trace back the trajectories of many air parcels starting from a given area for each timestamp. Figure 5.1-(c) shows one such forward trajectory starting in city A. Every dot in the shown sequence represents the GPS coordinate location at hourly resolution. In this figure we can observe that the air flow propagates the pollutants from city A to Beijing. Figure 5.1-(d) shows all trajectories aggregated over one year, colored by their $PM_{2.5}$ concentration values. We can see that the main air flow trajectories coming from the south of Beijing bring polluted air to the city, while northern trajectories seem to contribute cleaner air.

Based on the previous observations and findings, we know that *(i)* air quality readings from more surrounding cities may be helpful to improve the accuracy of air quality prediction; and *(ii)* air flow trajectories could be closely related to the pollution transfer between cities. However, using all this data and encoding this knowledge into a model is not straightforward for the following reasons. Firstly, the relationship between air quality among surrounding cities is complex and constantly changing due to the dynamic environment. Secondly, it is unclear how to encode the 2-D air flow trajectory data and how to use it efficiently in a predictive model.

In this chapter, we propose a novel approach, called TIP-Air, to track pollution transfer based on air quality readings and air flow trajectory data. As shown in Figure 5.1-(b), our model captures the underlying complex relationship encoded in air quality readings between the current prediction target ($S3$) and all other locations ($S1 - S4$ in the same city; $C1$ in a remote city) by extracting knowledge from air flow trajectory data. Furthermore, the model is able to interpret the prediction results and to provide a reasonable evidence of why sudden changes happen, which is important for governments to design intelligent air pollution reduction strategies. To the best of our knowledge, this is the first deep learning model that captures pollution transfer between different areas and predicts sudden pollution changes with reasonable accuracy.

Overall, the contributions of TIP-Air are:

- We propose a seq2seq model to learn air flow trajectory representations to model pollution transfer between different areas. The representation is robust and easy to apply to other research tasks in the context of air quality research.
- For the purpose of learning spatial weights, we develop a new attention mechanism based on the air flow trajectories and representations of locations of air quality measurement stations.
- We conduct experiments on real world datasets from the greater Beijing area. The results offer evidence that the proposed method outperforms existing air quality predictive models in terms of both accuracy and interpretation, especially for the intervals of sudden pollution changes.

5.2 Related Work

In this section, we briefly review related literature on attention mechanism in deep learning and pollution propagation analysis using HYSPLIT.

Attention Mechanism. Recently, attention mechanisms have become popular due to their success in general sequence-to-sequence problems. Bahdanau et al. [BCB14] first introduced a general attention model that

did not assume a monotonic alignment. Qin et al. [QSC+17] proposed a dual-stage attention-based recurrent neural network (DA-RNN) to select the relevant driving series at each time interval. They introduced an input attention mechanism to adaptively extract relevant driving series (so called input features) at each time step by referring to the previous encoder hidden state. Following a similar input attention mechanism, Liang et al. [LKZ+18] predicted the air quality index by putting attention weights on the input data. Apart from the input attention mechanism, temporal attention is also used widely to align the temporal patterns between encoder and decoder states. Shin et al. [SSL19] used a set of filters to extract time-invariant temporal patterns and capture temporal patterns across multiple time steps. Muralidhar et al. [MMR19] proposed a novel hierarchical attention mechanism for long-term time series state forecasting.

Pollution Propagation Analysis Using HYSPLIT. For the model-based analysis of air pollution propagation, HYSPLIT [SDR+15], developed by NOAA's Air Resources Laboratory, is one of the most widely used models for atmospheric trajectory and dispersion calculations. As stated in research work [KVL+11], air flow trajectory analysis is one of the standard procedures for determining the spatial locations of possible emission sources affecting given receptors, and it is frequently used to enhance receptor modeling results. Furthermore, McGowan et al. [MC08] and Li et al. [LCCC17] identify regional pollution sources and propagation pathways. Based on the air flow trajectory data, Gao et al. [GTC+15] conduct research on the formation causes during two haze pollution events in urban Beijing, China. The results show that regional transport contributes the elevated content of anthropogenic elements in $PM_{2.5}$. Wang et al. [WCC+10] also show that air pollution in urban cites is caused not only by local emission sources but also to a large extent by regional atmospheric pollution transport from surrounding areas, responsible for sudden pollution changes.

Our intuition comes from the related works on pollution propagation analysis using HYSPLIT, that the air flow trajectories provide a useful representation to model pollution transfer between different areas. In contrast to the current approaches described in the air quality prediction section (Section 1.3), we consider local air pollution emissions and remote pollution propagation from surround cities simultaneously, and propose a novel attention mechanism to learn the weights for all air quality and weather stations dynamically.

5.3 Problem Definition and Analysis

Definition 5.1. *(Air flow trajectory) An air flow trajectory is a sequence of sample points from the underlying route of the moving air flow [SDR+15].*

At each timestamp i, staring from a city k, we collect air flow trajectory data $\mathbf{J}_k^i \in \mathbb{R}^{\tau \times 2}$ for the next τ hours, where $\mathbf{J}_k^i$ denotes the air flow GPS coordinates. Trajectories starting from multiple surrounding cities N_c are aggregated into an overall air flow pattern $\mathbf{J}^i = \sum_{k=1}^{N_c} \mathbf{J}_k^i$ for our target station at timestamp i.

Suppose we have N_g air quality stations located in the target area as well as surrounding areas or cities. $\mathbf{X}^i \in \mathbb{R}^{N_g \times m}$ and $\mathbf{W}^i \in \mathbb{R}^{N_g \times n}$ represent air quality readings and weather readings from all stations at time i, respectively, where m and n are the number of observed features. Given a time window of length T, air quality features are specified as $\mathbf{X} = (\mathbf{X}^1, \mathbf{X}^2, \ldots, \mathbf{X}^T)$, and forecast weather features are specified as $\mathbf{W} = \left(\mathbf{W}^{T+1}, \mathbf{W}^{T+2}, \ldots, \mathbf{W}^{T+\tau}\right)$, where τ is the length of the forecasting time window. Similarly, the forecast air flow trajectory data staring from city k in the next τ hours is denotes as $\mathbf{J}_k = \left(\mathbf{J}_k^{T+1}, \mathbf{J}_k^{T+2}, \ldots, \mathbf{J}_k^{T+\tau}\right) \in \mathbb{R}^{\tau \times 2 \times \tau}$. Then, $\mathbf{J} = \left(\mathbf{J}_1, \mathbf{J}_2, \ldots, \mathbf{J}_{N_c}\right)$ represents all forecast trajectory data from N_c cities for the next τ hours. Time features [ZLG+19], such as *day-of-week* and *hour-of-day* etc., are specified as $\mathbf{D} \in \mathbb{R}^p$, where p is the number of time features extracted from timestamps of data points.

Problem Statement. Given historical data over the past T hours and the weather forecast data for the next τ hours, we predict the air quality at the target location in the next τ hours as $\hat{y} = (\hat{y}^{T+1}, \hat{y}^{T+2}, \ldots \hat{y}^{T+\tau}) \in \mathbb{R}^\tau$. The purpose of the model is thus to predict:

$$\hat{y} = \mathcal{F}(\boldsymbol{X}, \boldsymbol{W}, \boldsymbol{D}, \boldsymbol{J}) \tag{5.1}$$

where $\mathcal{F}$ is the prediction function to be learned.

5.4 Proposed Model

This section presents our novel TIP-Air air pollution modeling approach which incorporates air flow trajectory data coupled with the spatiotemporal attention mechanisms.

5.4.1 Overall Framework

Figure 5.2 shows the overall framework of our proposed TIP-Air approach. We use representation learning to embed air flow trajectory data and GPS coordinates of the measurement stations into vectors as will be described in the **Representation Learning** subsection. Then, we propose a new spatial attention mechanism to learn the weights for the readings of each air quality station. The trajectory and station location vectors are used as input to the spatial attention module to learn the attention weights for sensor readings from each station. For example, if the air flow trajectory arrives from the south of the target in the next hours,

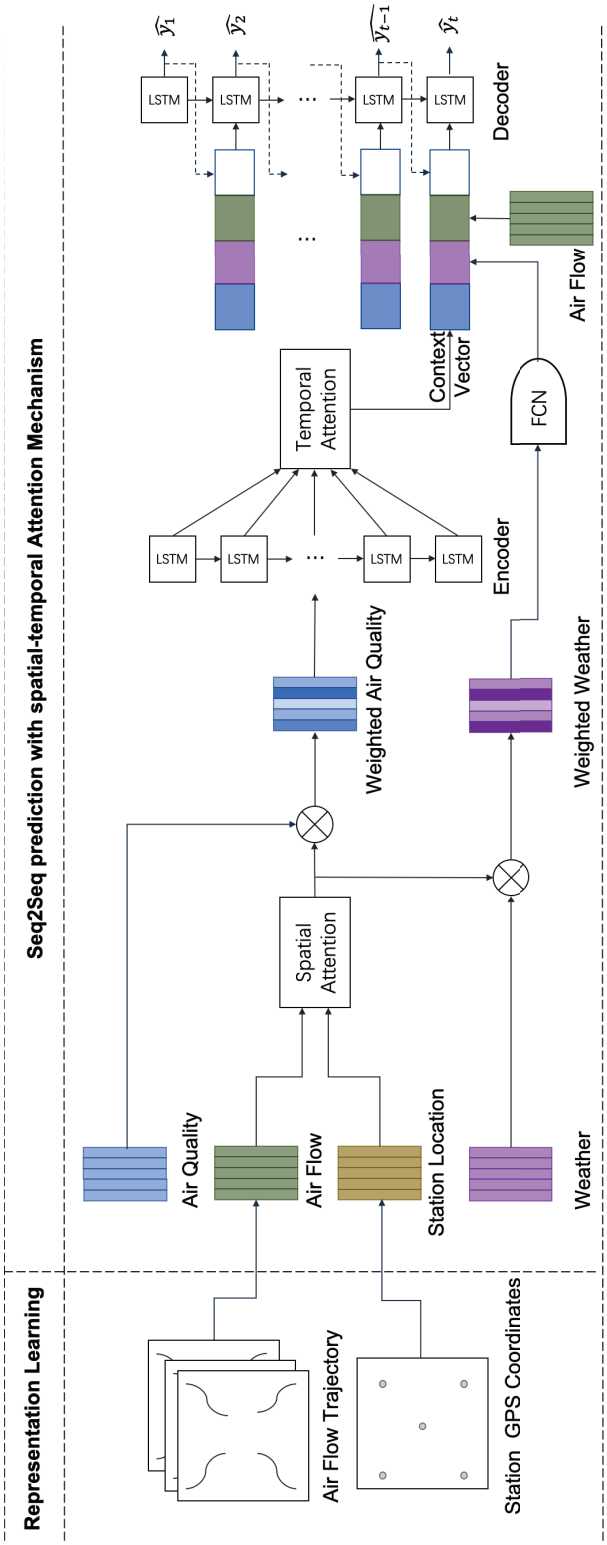


Figure 5.2: The proposed TIP-Air framework makes use of the air quality trajectory data combined with spatiotemporal attention mechanisms to accurately predict short-term air pollution development in a comp7lex region with multiple sources.

the model should pay more attention (weight) to the station nodes along the path, and vice versa. See details in the **Spatial Attention** section.

By multiplying the spatial attention weights with the raw air quality data and weather data, we get new weighted air quality and weather data. They will be used as the new input data for our seq2seq based prediction network. In the encoder, each block denotes the air quality readings from all stations at a given timestamp, which in general includes all the historical air quality data from all stations (N_g) during the historical time period T. Then, for each block of the decoder, the input contains three parts: *(1)* The context vector, which is a weighted combination of all the hidden states of the encoder. To solve the difficulty of capturing long-term dependency, we apply a temporal attention mechanism to learn the weights, see details in the **Temporal Attention** section. *(2)* The weather vector and the air flow trajectory vector. The weather information, which represents the local pollution evolving patterns at the target as well as at remote locations, is useful for accurate air quality predictions. This data is included as input to each decoder block. The air flow trajectory vector, which denotes the global pollution transfer pattern, is also included to capture the pollution transfer. *(3)* The prediction value of the previous time step. We are using a seq2seq framework to predict the next τ hours, and therefore the prediction values of previous step are beneficial for the current prediction.

5.4.2 Representation Learning

Trajectory Representation Learning. Learning representations for specific tasks has been a longstanding open problem in machine learning. Recently, inspired by the success of word2vec [MCCD13], the idea of learning general representations has been extended to paragraphs [LM14], networks [PARS14], trajectories [LZC+18], etc. To capture the sequential order information emerging in sequence processing tasks, encoder-decoder based Recurrent Neural Network models (RNNs) have been developed, such as sequence to sequence learning [CVMG+14], and skip-through vectors [KZS+15]. For our first task of learning trajectory representations, we use a sequence encoder-decoder model to encode each trajectory starting from city k at time i, namely $\mathbf{J}_k^i \in \mathbb{R}^{\tau \times 2}$, into one vector $\mathbf{V}_k^i \in \mathbb{R}^{q \times 1}$, where q is the dimension of the vector.

The whole module consists of a fully connected neural network and an encoder-decoder [CVMG+14] RNN, as shown in Figure 5.3. A fully connected network is applied to each single trajectory point to embed the point into a space with the same dimension as the desired trajectory representation. The embedded vectors are then input to the RNN encoder in a chronological order. We choose GRU to be the recurrent unit as in [CGCB14] as this architecture is empirically shown to be more efficient and requires less parameters than LSTM. The logic of adopting an RNN

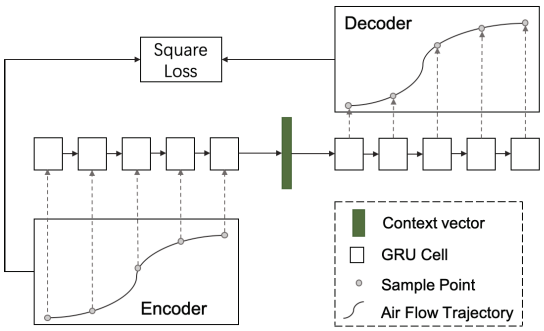


Figure 5.3: Representation learning for air flow trajectories using a sequence encoder-decoder model.

decoder here is similar to the way people come up with natural features of trajectories. If the decoder can restore the trajectory from the encoded feature representation, then we can assume that there is little information lost in the encoding, and the representation is thus suitable to facilitate $PM_{2.5}$ prediction.

The loss function is the square loss between the input and the restored trajectories. Formally, we denote two sequences $x = \langle x_t \rangle_{t=1}^{|x|}$ and $y = \langle y_t \rangle_{t=1}^{|y|}$ as the encoder ($\mathbf{J}_k^i$) and decoder ($\hat{\mathbf{J}}_k^i$) trajectory data, respectively. Each x_t and y_t denotes the trajectory point representation, and $|x|$ and $|y|$ represent the length of the trajectories. In our task, we model

$$\mathbb{P}\left(y_1, \ldots, y_{|y|} | x_1, \ldots, x_{|x|}\right) = \mathbb{P}\left(y_1 | x\right) \prod_{t=2}^{|y|} \mathbb{P}\left(y_t | y_{1:t-1}, x\right). \tag{5.2}$$

The encoder reads in and encodes the sequence x into a fixed-dimensional vector v. Since v encodes sequential information in x, we have

$$\mathbb{P}\left(y_t | y_{1:t-1}, x\right) = \mathbb{P}\left(y_t | y_{1:t-1}, v\right). \tag{5.3}$$

The decoder computes the probability $\mathbb{P}\left(y_t | y_{1:t-1}, v\right)$ at every position t by squashing $y_{1:t-1}$ and v into a hidden state. Then the loss is calculated by $\mathcal{L} = MSE(y, x)$. Since the context vector v acts as the initial hidden state of the decoder, it can be used as a representation vector of the trajectory path, which is denoted as $\mathbf{V}_k^i$. Then, $\mathbf{V}_J^i = \sum_{k=1}^{N_c} \mathbf{V}_k^i$ represents all the trajectory representations at time i.

Node Representation Learning. A straightforward approach to represent a station location is by using the centroid coordinates of the station (GPS coordinates) directly. The centroid coordinates of the stations naturally encode the spatial proximity for the stations but restrict the representations to a two-dimensional space. This makes it difficult for the loss function to further optimize the representations in their parameter space. Another widely used technique in representation learning is the

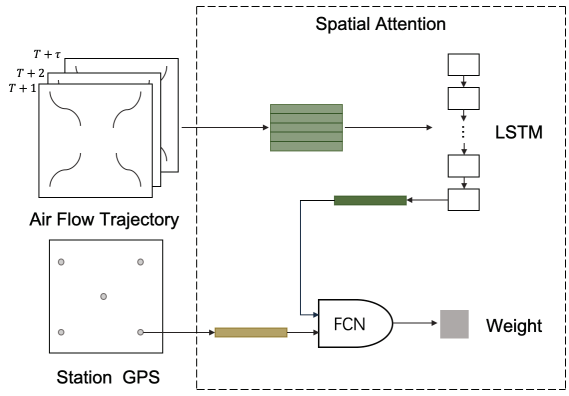


Figure 5.4: Spatial attention mechanism based on the sequence-to-sequence architecture with a dense layer (FCN) with air flow trajectory data and station coordinate information as input.

one-hot encoding, especially in the NLP domain [LZC+18]. One-hot representation could represent more meaningful underlying relationship between different words by learning a high dimensional vector for each word. However, the one-hot representation loses the spatial distance relation of the stations as all stations are then treated independently.

We borrow the idea of high dimension representations for each station location from one-hot encoding and map the i-th station location information ($\mathbf{L}^i$) to a high dimension representation ($\mathbf{V}_L^i$) via a fully connected network:

$$\mathbf{V}_L^i = FCN(\mathbf{L}^i). \tag{5.4}$$

This FCN is connected to the whole TIP-Air framework and parameters are learned by back propagation. The learned high dimension representations are supposed to be easier to discover and represent spatial relations between stations.

5.4.3 Spatial Attention

In the previous section, we encoded the air flow trajectory and station location data into vectors via representation learning. Following the previous example in the **Introduction** section, the air flow trajectory data imply the air pollution propagation patterns in the future. In other words, it can be used as an indicator of sudden changes. Therefore, instead of using all air quality measurements to make a prediction, we design a spatial attention module to learn the weights for each sensor reading.

Given all the trajectory data $\mathbf{J} = \left(\mathbf{J}_1, \mathbf{J}_2, \ldots, \mathbf{J}_{N_c}\right)$ from N_c cities in the next τ hours, we encode the trajectory data into vectors according to the method described in **Trajectory Representation Learning**, which is referred as $\mathbf{V}_J = \left(\mathbf{V}_J^{T+1}, \mathbf{V}_J^{T+2}, \ldots, \mathbf{V}_J^{T+\tau}\right) \in \mathbb{R}^{|\mathbf{V}_J^{T+i}| \times \tau}$, where $|\mathbf{V}_J^{T+i}|$ denotes the length of the encoded trajectory vector. Let $\mathbf{L} = \left(\mathbf{L}^1, \mathbf{L}^2, \ldots, \mathbf{L}^{N_g}\right)$

represent the GPS coordinates of the station. Then, the corresponding high dimension representation is denoted as $\mathbf{V}_L = \left(\mathbf{V}_L^1, \mathbf{V}_L^2, \ldots, \mathbf{V}_L^{N_g}\right)$, where $|\mathbf{V}_L^i|$ denotes the length of the encoded station location vector.

To incorporate the knowledge of air flow patterns over the next τ hours, a sequence to sequence architecture with a dense layer [HBB19] is proposed to learn the embedded hidden vector for $\mathbf{V}_J$, as shown in Figure 5.4. $\mathbf{V}_J^{T+i}$ is the i-th input to the block of the LSTM architecture. The output of the last block denotes a vector with the hidden information from the previous blocks defined as:

$$\mathbb{P}\left(\mathbf{V}_J^{'}|\mathbf{V}_J^{T+1}, \mathbf{V}_J^{T+2}, \ldots, \mathbf{V}_J^{T+\tau}\right) \tag{5.5}$$

After having obtained the embedded trajectory vector $\mathbf{V}_J^{'}$ in each prediction iteration, it is combined with the station location vector $\mathbf{V}_L^k$. The spatial attention α^k is then applied as follows:

$$e^k = \mathbf{V}_e \tanh\left(W_e \mathbf{V}_J^{'} + U_e \mathbf{V}_L^k + B_e\right) \tag{5.6}$$

$$\alpha^k = \frac{\exp\left(e^k\right)}{\sum_i^{N_g} \exp\left(e^i\right)}, \tag{5.7}$$

where W_e, V_e, U_e and B_e are parameters to be learned. The spatial attention weights $E = \left(e^1, e^2, \ldots, e^{N_g}\right)$ are treated with a softmax function to ensure their sum equals to 1. The output vector of the spatial attention layer is represented as $\left(\alpha^1 \mathbf{X}_1, \alpha^2 \mathbf{X}_2, \ldots, \alpha^{N_g} \mathbf{X}_{N_g}\right) \in \mathbb{R}^{N_g \times T}$, where $\mathbf{X}_i$ is the i-th station readings during the time period T. The aforementioned output vector is processed by a LSTM layer to get the encoder output $Z_t = \left(z_t^1, z_t^2, \ldots, z_t^s\right)^T \in \mathbb{R}^{s \times T}$, where s is the dimension of the hidden state. The encoder output serves as the input to the temporal attention layer.

5.4.4 Temporal Attention

Since the performance of the proposed encoder-decoder architecture will degrade rapidly as the encoder length increases [CVMG+14], we apply a temporal attention mechanism to adaptively select the relevant hidden states of the encoder and to produce the context vector, which is then used as part of the input to the decoder. Specifically, the attention weight of each hidden state of the encoder at time t is calculated based on the previous decoder hidden state $\mathbf{d}_{t-1} \in \mathbb{R}^p$ and the cell state of the LSTM unit $\mathbf{s}'_{t-1} \in \mathbb{R}^p$ with

$$l_t^i = \mathbf{v}_d^\top \tanh\left(\mathbf{W}_d \left[\mathbf{d}_{t-1}; \mathbf{s}'_{t-1}\right] + \mathbf{U}_d \mathbf{h}_i\right), \quad 1 \leq i \leq \tau \tag{5.8}$$

and

$$\beta_t^i = \frac{\exp\left(l_t^i\right)}{\sum_{j=1}^T \exp\left(l_t^j\right)}, \tag{5.9}$$

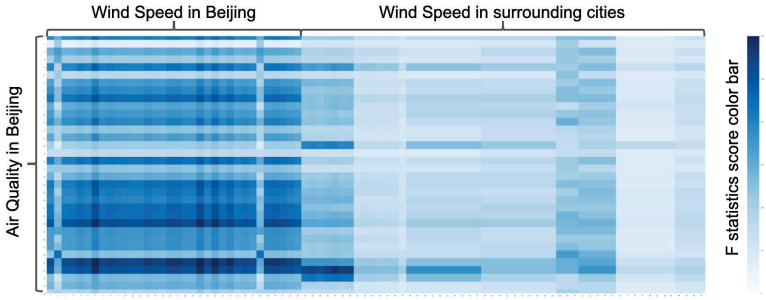


Figure 5.5: Granger causality test [Gra69] confusion matrices between air quality measurements in Beijing and the wind speed data in both Beijing and surrounding cities.

where $\left[\mathbf{d}_{t-1}; \mathbf{s}'_{t-1}\right] \in \mathbb{R}^{2p}$ is a concatenation of the previous hidden state and cell state of the LSTM unit. h_i denotes the i-th encoder hidden state, and v_d, W_d, U_d are parameters to be learned. Then, the attention mechanism computes the context vector c_t as a weighted sum of all the encoder hidden states $\{\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_T\}$ as

$$\mathbf{c}_t = \sum_{i=1}^{T} \beta_t^i \mathbf{h}_i. \tag{5.10}$$

5.4.5 Weather Fusion Module

The Granger causality test [Gra69] is used to determine if one time series is useful to forecast another variable by investigating causality between two variables in a time series. Figure 5.5 plots the Granger causality test confusion matrices between the air quality readings and the corresponding wind speeds for every air quality station. The result shows that *(i)* the local wind speed for every air quality station is closely related to the future air quality; and *(ii)* the wind speed at remote locations helps to predict the air quality in Beijing.

Instead of using the local weather data, we propose to include all weather data including the data from remote locations based on the following intuition. We argue that the air quality at remote locations affects the air quality evolving patterns at the target location, i.e., the pollution may propagate from remote cities to the target city. Under this setting, weather conditions at remote locations provide important information about the future air quality changes. For example, strong wind at remote locations locally decrease the local pollution levels and thus less pollution propagates to the target location. Based on this intuition, we use a combination of the weighted data $\tilde{\mathbf{W}}$, especially the wind speed data, to represent the overall influence of weather conditions on the air quality evolution in the future as shown in Figure 5.2.

5.4.6 Encoder Decoder and Model Training

In the encoder, after getting the spatial attention weights for each station $\alpha^k \left(1 \leq k \leq N_g\right)$, we multiply the attention values with raw station readings to get the newly weighted air quality data $\tilde{\mathbf{X}}$. We feed them as the new input to the encoder and update the hidden state at time t by using $\mathbf{h}_t = f_e\left(\mathbf{h}_{t-1}, \tilde{\mathbf{X}}^t\right)$, where f_e denotes an LSTM unit.

In the decoder, once we get the weighted sum context vector $\mathbf{c}_{t'}$ at a future time t', we combine it with external factors, such as weighted weather vector $\tilde{\mathbf{W}}^{t'}$ and air flow trajectory vector $\mathbf{V}_J^{t'}$, and the last output of decoder $\hat{y}_{t'-1}$ in order to update the decoder hidden state with $\mathbf{d}_{t'} = f_d\left(\mathbf{d}_{t'-1}, \left[\hat{y}^i_{t'-1}; \tilde{\mathbf{W}}^{t'}; \mathbf{V}_J^{t'}; \mathbf{c}_{t'}\right]\right)$, where f_d is an LSTM unit used in the decoder. Then, we concatenate the context vector $\mathbf{c}_t$ with the hidden state $d_{t'}$, which becomes the new hidden state from which we make final predictions as follows:

$$\hat{y}_{t'} = \mathbf{v}_y^\top \left(\mathbf{W}_m \left[\mathbf{c}_{t'}; \mathbf{d}_{t'}\right] + \mathbf{b}_m\right) + b_y, \tag{5.11}$$

where $\mathbf{v}_y^\top$, $\mathbf{W}_m$, $\mathbf{b}_m$ and b_y are parameters to be learned. The mean squared error (MSE) loss is minimized by the Adam optimizer [KB14].

5.5 Experimental Evaluation

In this section, we first introduce the datasets used, list relevant baselines we compare to, and provide implementation details. Then, we show that the proposed approach outperforms the state-of-art by a fair margin. We then empirically verify the influence of main architectural decisions on the prediction quality through ablation studies. Finally, a case study is given to show the effectiveness of the sudden change prediction and the interpretability of the results.

5.5.1 Experimental setup

Datasets. We collect *air quality data*[1], including PM2.5, PM10, O3, NO2, CO and SO2, from 35 stations in Beijing and 55 stations in the surrounding cities between Jan 1, 2016 and Jan 31, 2018. The system collects *meteorological data*[2] from related cities/districts every hour. Beijing has district-level granularity for the data, while surrounding cities have a city-level report. Each record consists of weather (sunny, cloudy, overcast, foggy, snow, small rain, moderate rain, and heavy rain), humidity, temperature, pressure, wind speed, and wind direction. To represent the air flow propagation between cities, we include the *air flow trajectory data* produced by National Oceanic and Atmospheric Administration

[1] https://quotsoft.net/air/

[2] https://rda.ucar.edu/datasets/ds084.1/

Table 5.1: Performance comparisons of different models.

	1-6h	7-12h	13-18h	19-24h
Naïve	14.87	26.00	32.21	35.45
LSTM	14.17	25.88	32.67	37.04
Seq2Seq	14.13	23.99	30.14	33.61
GeoMAN	14.03	19.42	22.95	24.23
MGED-Net	13.44	18.05	20.95	21.91
TIP-Air	**12.15**	**15.34**	**16.25**	**17.21**

(NOAA). In our study, we crawl HYSPLIT's forward trajectories[3] starting from surrounding 9 cities of Beijing. Each trajectory represents the predicted air flow propagation path over 24 hours.

Baselines. We compare TIP-Air to the following baselines:

- **Naïve approach:** Uses the current hour as the predicted value for all future hours.
- **LSTM:** Uses historical 24 hour readings to predict the future.
- **Seq2Seq:** The architecture features stacked LSTMs in both encoder and decoder and uses historical data over the past 24 hours for future predictions.
- **GeoMAN [LKZ+18]:** A feature fusion encoder-decoder architecture with multi-level attention to learn feature importance.
- **MGED-Net [ZLG+19]:** A deep model to fuse heterogeneous fine-grained weather data for air quality prediction.

Evaluation Metrics and Model Details. We use MAE to evaluate our algorithms on predicting the values of all 35 stations in Beijing for the next 24 hours. The sequence length for both encoder and decoder is set to 24. Grid search is used to decide on the optimal hyperparameter combination. We set the learning rate to 0.001, batch size to 64, and apply early stopping for model training. We use Adam to update parameters and Mean Squared Error (MSE) as loss function. All experiments are performed on a machine with two NVIDIA GTX 2080 Ti GPUs.

5.5.2 Experiment Results

Overall Performance Comparison. Table 5.1 presents achieved prediction performance of all methods introduced above. Among all

[3]https://ready.arl.noaa.gov/HYSPLIT_traj.php

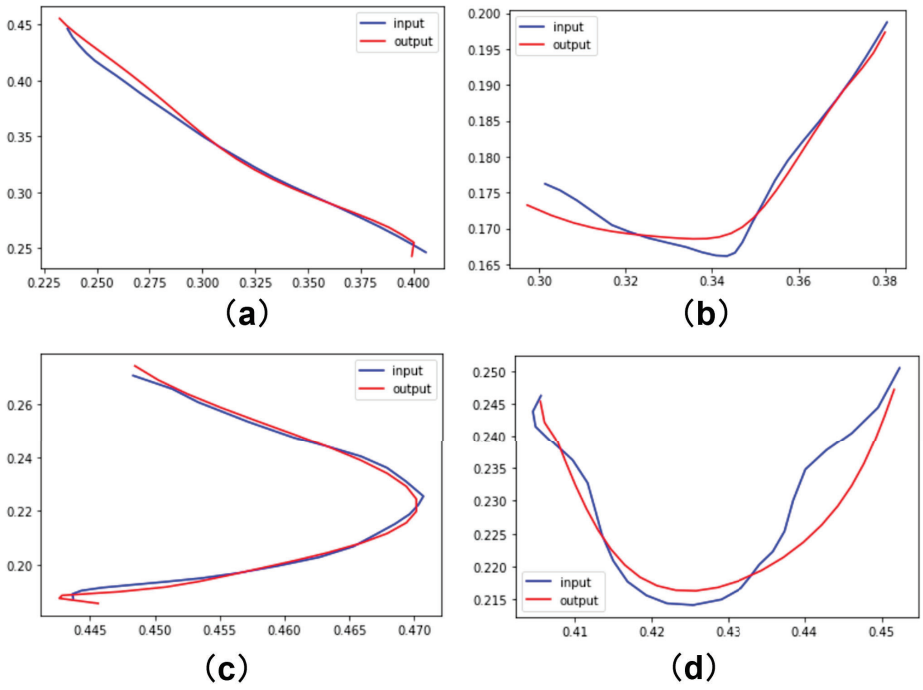


Figure 5.6: The original air flow trajectory (in blue) and the corresponding decoder output from the embedding representations (in red). Figures (a)-(d) show several examples of different complexity.

models, the proposed TIP-Air approach yields the best performance for 1 to 24 hours predictions. We can also observe that GeoMAN achieves a significantly better performance for longer time horizons than LSTM and Seq2Seq methods. The major drawback of GeoMAN is that feature interactions are not well modeled by its feature fusion architecture. MGED-Net solves this problem by including a group interactions module to fuse the data from multi-domains and further improve the prediction accuracy. However, both GeoMAN and MGED-Net fail to capture pollution transfer from remote locations. Therefore, TIP-Air outperforms all baselines and achieves between 9.6% and 22.4 % improvement in MAE.

Evaluation on Feature Representations. For the trajectory representation learning, the encoder and decoder length are both 24 and the dimension of the context vector $\mathbf{V}_k^i$ is 10×1. The dimension of node representation $\mathbf{V}_L^i$ is 256×1. Figure 5.6 shows the effectiveness of the proposed trajectory representation learning method. The embedding representation vector $\mathbf{V}_k^i$ could successfully recover air flow trajectories of various shape, which means the vector captures the fundamental features well, removes potential noise and can be used to represent the air flow data.

We compare two different options to represent air flow trajectory data and station GPS coordinates: (i) raw values of the data, or (ii) embedded into a vector space via representation learning. They are applied to both

Table 5.2: Evaluation of feature representation. rawTraj denotes raw trajectory data; rawGPS denotes raw GPS coordinates.

	1-6h	7-12h	13-18h	19-24h
TIP-Air-rawTraj	13.21	16.45	18.34	19.22
TIP-Air-rawGPS	12.85	16.12	17.23	18.11
TIP-Air	**12.15**	**15.34**	**16.25**	**17.21**

Table 5.3: Evaluation of spatial attention. **A** denotes addition of air quality readings from other cities; **J** denotes trajectory data; **na** means no spatial attention on air quality data; **nw** means no spatial attention on weather data; **ns** means no spatial attention.

	1-6h	7-12h	13-18h	19-24h
GeoMAN+**A**	14.09	19.58	24.36	26.16
GeoMAN+**A** + **J**	13.89	19.01	21.03	22.62
MGED-Net+**A**	13.64	18.91	22.56	23.85
MGED-Net+**A** + **J**	13.01	17.78	19.88	20.15
TIP-Air-na	13.22	18.78	20.65	21.01
TIP-Air-nw	12.88	15.92	16.99	18.20
TIP-Air-ns	13.76	18.92	20.78	21.45
TIP-Air	**12.15**	**15.34**	**16.25**	**17.21**

encoder and decoder. As shown in Table 5.2, using air flow trajectory data leads to a significant improvement of the prediction quality, even if raw trajectory data is used directly. The result of using raw air flow trajectory data is already better than what can be achieved by the state-of-art methods. The same phenomenon applies to the GPS data. In this case, the prediction performance is not as good as for our full TIP-Air model, since two-dimensional GPS coordinates are not powerful enough to encode the station location information. This hinders the positive effect of the spatial attention.

Evaluation of Spatial Attention. To evaluate the effect of the proposed spatial attention mechanism on both air quality data and weather data, we test all combinations and list the result in Table 5.3. We observe that: (i) The combination of the spatial attention mechanism for both air quality data and weather data shows great improvement against each individual contribution, which shows the importance of using a joint spatial attention mechanism on both air quality data and weather data. (ii) The fact that TIP-Air outperforms GeoMAN and MGED-Net, which both include the air quality data in remote cities and air flow trajectory data, verifies the advantages of our spatial attention against the

Table 5.4: Evaluation of the impact of weather factors. **nl** denotes no local weather data; **ng** means no global weather data; **ne** denotes no weather data.

	1-6h	7-12h	13-18h	19-24h
TIP-Air-nl	12.18	15.88	16.99	18.02
TIP-Air-ng	12.89	16.03	17.23	18.82
TIP-Air-ne	13.01	16.72	18.01	19.34
TIP-Air	**12.15**	**15.34**	**16.25**	**17.21**

modules used in the previous methods. Here, they intend to learn the similarity between measurements from different stations from historical data and assign higher weights when station readings are more similar to the target one. In such settings, including extra air quality readings in surrounding cities has a negative influence on the results, as shown in Table 5.3. However, as we illustrated in the initial example, the air flow trajectory data in the future indicate the pollution propagation patterns, especially pollution transfer between cities. Therefore, our proposed spatial attention mechanism, which is derived from air flow trajectory data and station location data, is more appropriate to capture pollution propagation patterns and assign higher weights to the measurements from relevant station.

Evaluation of External Weather Factors. Current works take only local weather into account to predict air quality, i.e., the weather close to the target location. However, both air quality and local weather at remote locations matter in the context of pollution transfer. From the results shown in Table 5.4, we make the following observations: *(1)* Without using weather data, the prediction performance is worse compared to other baselines. *(2)* Including only the local weather data at the target location is not enough to capture the air quality development, especially when the air pollution transfers from remote cites. Similarly, the weather data in remote cites alone can not reflect the pollution dispersion in the target environment. *(3)* TIP-Air with the inclusion of weather data at local and remote locations works best and shows that the proposed weather attention and fusion mechanism is necessary to capture the aforementioned relationships.

Case Study. We use a case study to evaluate the interpretation ability of TIP-Air. Figure 5.7-(a) shows the air quality readings of Beijing and City 1 (C1), which is located in the south of Beijing and thought as one of the major causes of pollution for Beijing. At two prediction timestamps T_a and T_b, we make predictions for the next 19-24 hours indicated as shaded zones in the plot. Current approaches include solely historical data of Beijing and predict the air quality to be at a normal level in the future.

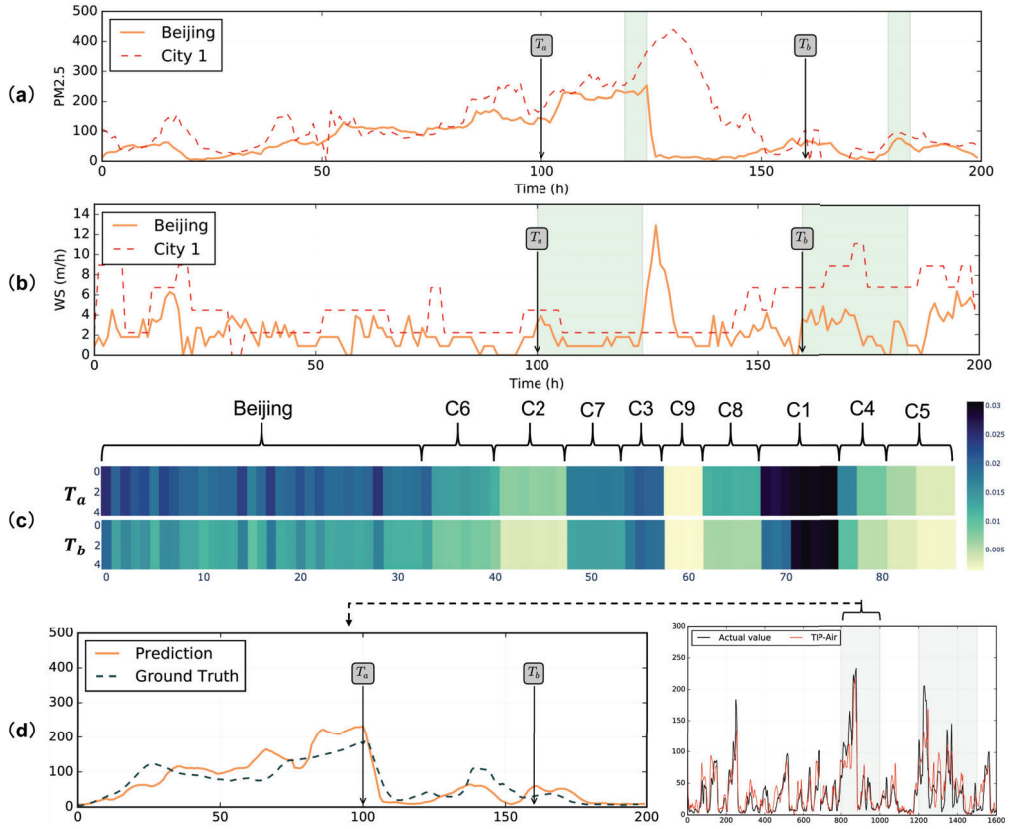


Figure 5.7: A case study to evaluate the TIP-Air interpretation ability: (a) Air quality readings. (b) Wind speed readings. (c) Spatial attention weights for prediction time T_a and T_b. (d) Prediction for the mean value of next 19-24 hours.

However, the spatial attention weights in TIP-Air show that the pollution propagate from City 1 (C1) to Beijing in both time periods, as shown in Figure 5.7-(c). The air quality data in C1 should also be included to make accurate predictions.

The prediction behaviors at timestamps T_a and T_b also differ a lot in terms of how the air quality in remote cities changes in the future. As shown in Figure 5.7-(c), at both timestamps, our spatial attention results indicate that more attention should be focused on the south of Beijing, e.g., C1. At timestamp T_a, pollution propagates from C1 to Beijing but the weather situation in C1, e.g., the low wind speed as shown in Figure 5.7-(b), is appropriate for pollution accumulation. While for prediction at T_b, the wind speed in C1 is quite strong in the future, which facilitates pollution dispersion and the air quality in C1 will probably decrease to low levels. In this scenario, even if air flow propagates pollution from C1 to Beijing, the effect is largely weakened and the air quality in Beijing will stay at a normal level instead of a hazardous. Compared with the state-of-art prediction results, our proposed method can capture the underlying pollution transfer between cities while taking local dynamics into account

and results in more accurate predictions, especially for sudden changes (shaded zones in the plot), as shown in Figure 5.7-(d).

5.5.3 Discussion

The use of the TIP-Air framework in other areas. We use Beijing as an example to illustrate and evaluate the proposed framework, however, TIP-Air is generalizable and can be easily adapted to predict air quality, especially sudden changes, in other areas. First, given a set of measurement stations in the area and a target, we can detect influential pollution sources by analyzing the backward air flow trajectories as shown in Section 5.3, then following the method proposed in Section 5.4, the prediction model could consider air quality and weather information in both target and surrounding locations to provide more accurate and interpretable predictions.

The use of the TIP-Air framework in other air quality contexts. The effectiveness of the proposed framework shows that pollution transfer is an important phenomenon in the air quality research domain, necessary to be considered for accurate prediction of complex pollution patterns. This motivates further research in the following air quality research contexts:

Air quality spatial interpolation. Current approaches [ZLH13b, CLL+14c] apply data-driven and deep learning methods to learn the relationship between sparsely deployed stations and infer the air quality values for unknown locations. Accompanied by the proven effective air flow trajectory data, new spatial interpolation method can be proposed to derive the pollution patterns and generate more accurate fine-grained air quality maps in large-scale area (e.g., 16,410.5 km^2 of Beijing) with a small number of stations (e.g., 35 in Beijing). Furthermore, those learned patterns could be more robust to air quality map transfer and downscaled sensor deployment tasks [CHZT20b].

Air quality missing data imputation. Current approaches [LZCY19] for missing air quality imputation fill missing values with values computed from local air quality data, local weather or land use data. New models which include air flow data and attention mechanisms can be designed to achieve better accuracy with interpretability.

The robust trajectory representation learning and attention-based prediction could be easily adapted to the above mentioned research problems.

5.6 Summary

In this chapter, we proposed TIP-Air, an attention based seq2seq model to predict the air quality, especially for those sudden changes, by tracking intercity pollution propagation. In the first time, we propose

to use the air flow trajectory data to represent the pollution propagation phenomenon between cities and embed the 2D trajectory data into vectors via representation learning, which could be used as pollution propagation representation vectors and applied in various applications and models. To derive the underlying complex interactions of the air quality and weather data between local and remote cities stations, we invented a novel spatial attention mechanism based on the air flow and sensor location data, which encodes the pollution propagation patterns during the future hours and then maps them to the weighed air quality and weather data used for the final seq2seq framework. In the seq2seq framework, a temporal attention mechanism is applies to capture the relationship between long term encoder steps and decoder step. Also, a weather fusion module is combined to include all the weather conditions in related locations and helps to predict those sudden changes. The experiments on real data set shows that our proposed method outperforms all the state-of-art baselines in a large margin, especially for those sudden change predictions. It also reveals a potential to interpret the prediction results visually and analytically, which is helpful for the government to make intelligent policies.

Predicting the city-scale air quality changes accurately is beneficial for administrators to make intelligent decisions. For example, where and when to close the pollution sources in order to reduce the pollution transfer to target cities. In the next chapter, we will introduce how to reduce the local air pollution with a novel water spraying system.

6

Reducing Urban Air Pollution with Intelligent Water Spraying

Various regulations and policies are made to improve city-level air quality in the long run, e.g., accurate predictions can help the government to make intelligent control policies (Chapter 5). Those pollution control strategies include policies such as closing the factories, traffic control etc.. However, there lack precise control measures to protect critical urban spots from heavy air pollution. In this chapter, we propose iSpray, the first-of-its-kind data analytics engine for fine-grained $PM_{2.5}$ and PM_{10} control at key urban areas via cost-effective water spraying. iSpray is a data-driven approach and uses the low-cost sensor measurements as its input to the analytics engine. The calibration model introduced in Chapter 2 and transfer method presented in Chapter 3 guarantee the accuracy of the low-cost sensing data, and the map generation method with downscaled deployment, i.e., MapTransfer in Chapter 4 enables a cost-effective sensing system. Those methods server as the building block of an accurate, reliable and affordable air quality sensing system, and will be an essential step on the success of our iSpray system. To effectively reduce the pollution, iSpray combines domain knowledge with machine learning to profile and model how water spraying affects $PM_{2.5}$ and PM_{10} concentrations in time and space. It also utilizes predictions of pollution propagation paths to schedule a minimal number of sprayers to keep the pollution concentrations at key spots under control. In-field evaluations show that compared with scheduling based on real-time pollution concentrations, iSpray reduces the total sprayer switch-on time by 32%, equivalent to $1,782\ m^3$ water and $18,262\ kWh$ electricity in our deployment, while decreasing the days of poor air quality at key spots by up to 16%.

6.1 Introduction

In addition to *passive monitoring* of urban air pollution, *active control* strategies are also crucial. Governments and authorities have launched various policies and regulations to reduce emissions from factories, transport, and household to improve the *overall* (e.g., city-level, annual average) air quality [OWS+20, CWZ+17]. However, there lacks measures for *fine-grained* (e.g., specific districts, hourly average) air pollution control. Such measures are complementary to the city-level policies and regulations and aim to offer precise protection to critical points of interest (POIs) such as residential areas, schools, hospitals, etc. within the city.

In this chapter, we explore water spraying for precise $PM_{2.5}$ and PM_{10} control at key urban POIs. Water spraying proves effective for dust control at construction and mining sites [Kis03, KSM14, TNHS+03] and has recently been applied for PM reduction in urban areas [Yu14]. The principle is to atomize water into micro droplets to fall in combination with ambient dusts [Pro13]. The fog produced by commodity sprayers can spread 10 to 100 meters and our field studies show water spraying reduces $PM_{2.5}$ and PM_{10} concentrations by 20% to 30% (up to 13 $\mu g/m^3$ for $PM_{2.5}$ and 19 $\mu g/m^3$ for PM_{10}), in various weather conditions (see Section 6.4.1), which is considered significant improvements in air pollution control [GS18]. Note that a reduction of $10\mu g/m^3$ in $PM_{2.5}$ and PM_{10} concentrations is valuable for the health of residents, especially on human respiratory system [XXSL16]. Clinical research indicated that the average life span was extended by 0.35 years for every $10\mu g/m^3$ decrease of $PM_{2.5}$ [CPID+13], whereas the mortality of cardiopulmonary diseases and lung cancer increased by 6% and 8%, respectively, for every $10\mu g/m^3$ increase of $PM_{2.5}$ [TKPI+11]. It is also shown that for each increase of PM_{10} by $10\mu g/m^3$, the overall morbidity increased by 0.38% [KC89], and the mortality related to respiratory diseases increased by 0.58% [AKD+06]. Therefore, water spraying holds potential for effective urban $PM_{2.5}$ and PM_{10} control at fine spatiotemporal granularity, and provides valuable benefits for human health, especially when the pollution reduction is over 10 $\mu g/m^3$.

Designing an urban water spraying system, however, faces multiple technical challenges. *(i)* There lacks quantitative models on how water spraying reduces $PM_{2.5}$ and PM_{10} concentrations in the urban outdoor space. Existing models are primarily derived for indoor environments with controlled ventilation [Yu14, dCMP+17]. They are unfit for profiling pollution reduction outdoors due to the complex aerodynamics and meteorological factors in the open urban space. It is difficult to decide which sprayers to switch on without a quantitative pollution reduction model. *(ii)* The water spraying system should be cost-effective, i.e., a minimal number of sprayers are switched on to keep the $PM_{2.5}$ and PM_{10} concentrations at the key POIs within the desired range. For example,

a single sprayer in our deployment consumes 0.6 m^3 water and 5 kWh electricity per hour, which adds up to 792 m^3 water and 6600 kWh electricity a day if all the sprayers are operating non-stop. We empirically show that a strategically selected sprayer subset would suffice to ensure the air quality level at given POIs.

To this end, we propose iSpray, a data analytics engine for fine-grained air pollution control at key urban POIs via cost-effective water spraying. We exploit both domain knowledge and data-driven approaches to characterize and model the spraying-induced pollution reduction in time and space. The hybrid approach enables accurate pollution reduction modeling even with limited spraying data for training. We further propose a sprayer scheduling scheme based on the predictions of pollution propagation paths. By prioritizing water spraying along the pollution propagation paths, we avoid unnecessary spraying that only marginally suppresses the pollution at the targeting POIs. The main contributions of this chapter are summarized as follows.

- To the best of our knowledge, we are the first to characterize the effect of commodity water sprayers on $PM_{2.5}$ and PM_{10} reduction in outdoor urban areas. Field studies show that the spraying-induced pollutant reduction at the sprayer's location is non-linearly weather-dependent, which can be modeled via a neural network, and the model generalizes across sprayer locations.

- We design an explainable model to integrate water spraying into urban air quality map generation. We exploit domain knowledge to isolate the impact of spraying on the pollutant's spatial distribution for easy sprayer scheduling and accurate map generation with limited spraying data. Evaluations show that our approach outperforms pure data-driven map generation by 7.9 to 9.3 in mean absolute error (MAE).

- We propose a propagation-aware sprayer scheduling algorithm for cost-effective air pollution control at key urban spots. Compared with the baseline strategy that switches on sprayers according to the current pollutant concentration, our scheduling scheme reduces the total sprayer switch-on time by 32%, or equivalently $1,782$ m^3 water and $18,262$ kWh electricity for our deployment, while decreasing the days of poor $PM_{2.5}$ and PM_{10} air quality at key POIs by 13% and 16%.

In the rest of this chapter, we provide an overview of iSpray in Section 6.2, explain the deployment and data collection in Section 6.3, and elaborate on each module in Section 6.4, Section 6.5 and Section 6.6. We present the overall evaluations of iSpray in Section 6.7 and conclude in Section 6.8.

6.2 iSpray Overview

iSpray is a data analytics engine for urban air pollution control with commodity sprayer hardware. It offers *(i)* pollution reduction modeling at single sprayer locations, *(ii)* pollution map generation, and *(iii)* cost-effective sprayer scheduling. Figure 6.1 illustrates the functional modules in iSpray. Table 6.1 summarizes the major notations that will be used throughout this chapter.

The *pollution reduction modeling module* (see Section 6.4) characterizes and quantifies the impact of water spraying on $PM_{2.5}$ and PM_{10} concentrations at the locations where the sprayers are installed. It is the foundation to integrate the impact of water spraying into air quality map generation (i.e., spatial distribution of pollutant concentration). Existing pollution reduction models for water spraying either halt at simulations [Yu14] or are designed for indoor scenarios with controlled ventilation [Kis03, KSM14]. They are unfit for modeling pollution reduction outdoors because they fail to account for the complex aerodynamics and meteorological factors in the open urban space. iSpray takes a data-driven approach to model how water spraying reduces outdoor air pollution under various environmental conditions. Through in-field studies, iSpray learns a neural network that quantifies the reduction in $PM_{2.5}$ or PM_{10} concentration at single sprayer locations given specific spraying time, meteorological conditions, and other environmental factors.

The *pollution map generation module* (see Section 6.5) models how water spraying affects the *spatial* distribution of $PM_{2.5}$ and PM_{10} concentrations. Due to limited spraying data for effective training, we model the spatial pollution reduction with both domain knowledge and data-driven approaches. Instead of feeding all data into a machine learning model as previous studies [HSW+15, CLL+14a, CDL+19a], we exploit a Gaussian plume model [Zan90, Ary04] to simulate pollution reduction in space by regarding the sprayer as a sink that absorbs pollution. We also propose a parameter learning strategy to estimate the inaccessible parameters in the Gaussian plume model from historical data. Evaluations show our hybrid modeling method outperforms pure data-driven schemes in modeling spraying-induced pollution reduction maps (see Section 6.5.4).

The *sprayer scheduling module* (see Section 6.6) aims to keep the air pollution at crucial POIs under predefined thresholds by switching on a minimal set of sprayers. Our measurements show that the spraying-induced pollution reduction is non-uniform across space (e.g., due to wind direction) and is non-linear to multiple environmental factors (e.g., weather). Therefore, the amount of pollution reduction at a given POI varies if a different sprayer is switched on. iSpray proposes a propagation-path-based heuristic to rank the importance of sprayers to the pollutant reduction at each POI, so as to turn on a minimal number of sprayers without exceeding the targeting pollution threshold at crucial POIs.

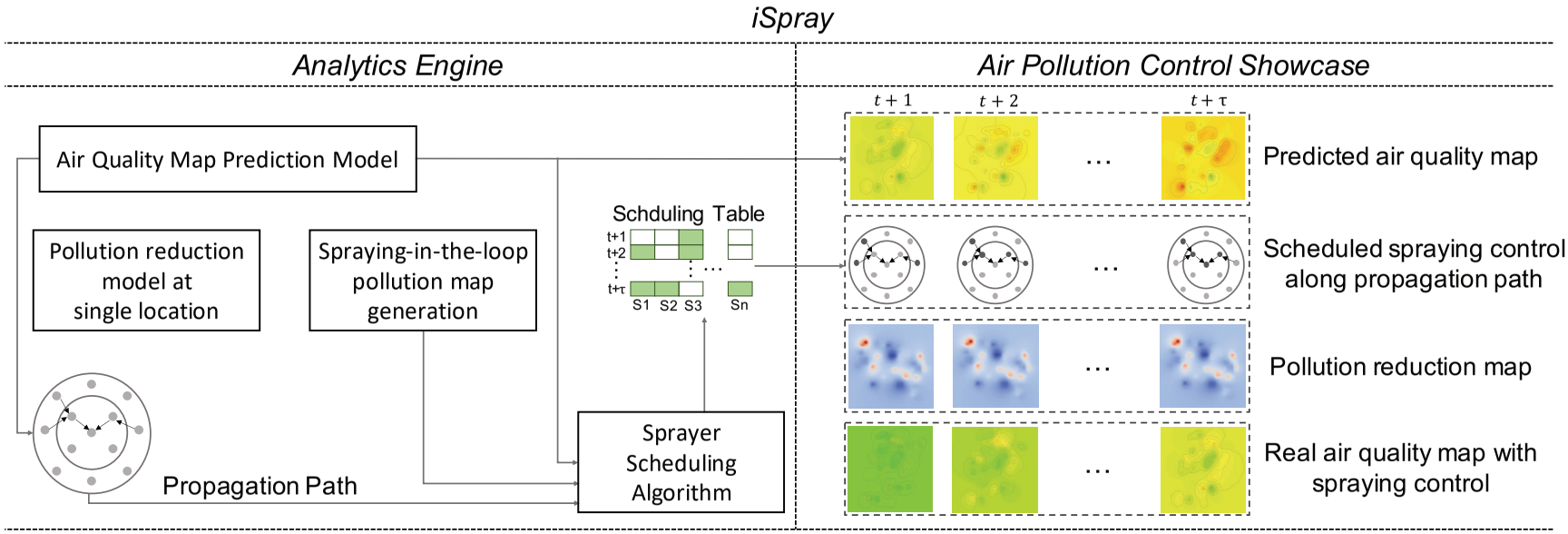


Figure 6.1: iSpray overview: functional modules of iSpray (left); workflow of iSpray (right).

Table 6.1: Summary of major notations.

Notation	Explanation
$C(g)$	ground truth pollution concentration in grid g
$\hat{C}(g)$	estimated pollution concentration in grid g
$C_d(g)$	pollution concentration in grid g due to dispersion
$R(g)$	overall pollution reduction in grid g due to water spraying
$d_{cross}(.)$	crosswind distance between two grids
$d_{down}(.)$	downwind distance between two grids
g	grid in 2-dimensional space
i	index for POI
k	index for sprayer
L_i	location of POI
$\boldsymbol{m}$	mean function of Gaussian process
$\boldsymbol{v}$	covariance function of Gaussian process
K	total number of sprayers in the region of interest
o	operating status of a sprayer, which can be *on* or *off*
$r(g\|s_k)$	pollutant reduction in grid g with sprayer s_k switched on
$s_k =< g_k, o_k >$	sprayer s_k in grid g_k with operating status o_k
t	discrete time index
Δt	time duration, set to 1 to 6 hours
$\phi(.)$	learned function for Gaussian plume dispersion parameter σ
Q_{air}	pollution emission rate
$Q_{s_k}(\Delta t)$	accumulative pollution reduction in grid of sprayer s_k
Q_{s_k}	abbreviation for $Q_{s_k}(\Delta t)$ when Δt is set to 1 hour
σ	Gaussian plume dispersion parameter
$\bar{w}$	average horizontal wind speed

6.3 Hardware Deployment and Data Collection

iSpray is designed as a software solution that works with commodity sprayer hardware. We used an existing deployment infrastructure, which includes both air quality sensors and sprayers, in this chapter to design and evaluate our algorithm. The full access to this deployment enables us to select the experiment areas, design the evaluation methods and test our proposed algorithms, which are our main contributions. In this section, we will present the sprayer hardware deployment and data collection in this study.

6.3.1 Sprayer Hardware and Deployment

Water spraying is widely used for dust control in the construction and mining industries [Kis03, KSM14, TNHS+03] and has also been applied for ambient particulate matter reduction in urban areas [Yu14]. The principle of water spraying for dust suppression is to atomize water into droplets of size comparable to fine particulate matters e.g., $1\mu m$ to $8\mu m$. These droplets can stay suspended in the air for a long time and will then fall in

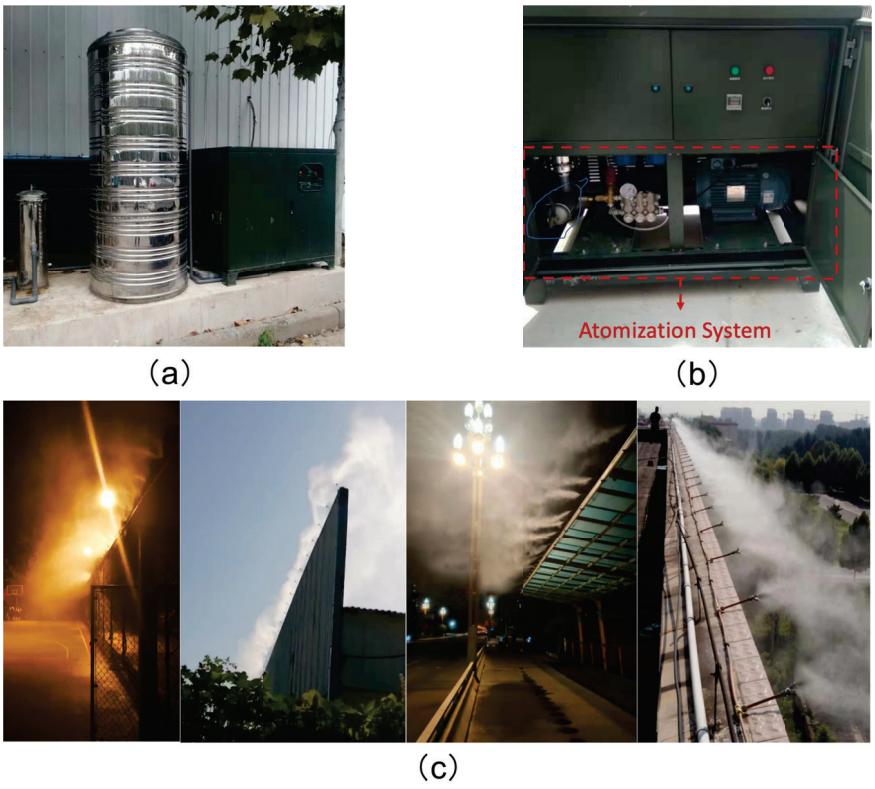


Figure 6.2: Ground hardware components of a commodity water sprayer: (a) exterior of the atomization system and the water tank; (b) internal design of the atomization system. (c) Example sprayer deployment at critical urban POIs: School, Hospital, Road and Factory. Note that only the multi-nozzle sprinkler of the sprayers are shown.

combination with the ambient dusts and particulate matters [Pro13].

A commodity sprayer exploits an electric motor to press water through high-pressure resistant pipes and atomizing nozzles to produce micro droplets. A single nozzle can produce fog lengths of 3 to 5 meters, which can spread 10 to 30 meters in windless conditions and 100 meters in windy conditions. A typical sprayer consists of an atomization system, a water tank, a multi-nozzle sprinkler and other control modules. The atomization system and water tank are normally installed on the ground (see Figure 6.2-a and Figure 6.2-b) while the sprinkler is usually installed high above the ground e.g., at the edge of rooftops, for better dust suppression performance (see Figure 6.2-c). iSpray is designed as part of the control module to intelligently switch on and off the sprayer.

Since we aim at cost-effective air pollution control at critical urban POIs, we use sprayers at various pollution-sensitive POIs such as schools and hospitals. We also use sprayers at locations of representative $PM_{2.5}$ and PM_{10} sources such as factories and roadsides to profile the impact of water spraying on air pollution reduction. 55 sprayers were installed at diverse urban POIs covering an area of 18 km $\times$ 24 km in a metropolis

in China. A portable air quality sensing box is also installed in the close vicinity of each sprayer to collect real-time $PM_{2.5}$ and PM_{10} concentrations as well as weather measurements including air temperature, relative humidity, air pressure, wind speed and wind direction. All the data are transmitted via NB-IoT to a central server.

We partition our sprayer deployment into three groups: *Research Area*, *Target Area* and *Control Area* (see Figure 6.3-c). The principles of area selection are as follows.

- The *Research Area* covers sprayers with co-located air quality sensing boxes. That is, each site within the *Research Area* consists of a sprayer and an air quality sensing box as shown in Figure 6.3-b. There are 55 such pairs of sprayers and air quality sensing boxes in the *Research Area*. This area is used for modeling and testing single-location pollution reduction (Section 6.4) as well as pollution reduction maps (Section 6.5).

- The *Target Area* is a sub-area of the *Research Area* where we would like to control the $PM_{2.5}$ and PM_{10} levels. It covers critical POIs as those shown in Figure 6.2-(c). We randomly pick two Target Areas that contain diverse POIs. *Target Area 1* contains 7 pairs of sprayers and sensing boxes and *Target Area 2* contains 8. We mainly use *Target Area 1* to test the sprayer scheduling performance of iSpray in the evaluations (Section 6.7) and the slightly smaller *Target Area 2* to assess the generalization of iSpray (Section 6.7.4).

- The *Control Areas* are used as control groups against the *Target Areas* to evaluate the effectiveness of water spraying. Each site in the *Control Areas* only has an air quality sensing box without a sprayer. We select *Control Areas* with the following criteria. First, the $PM_{2.5}$ and PM_{10} distributions of the *Control Area* should be similar to those in the *Target Areas* when the sprayers are closed. The similarity is measured by the Kullback-Leibler (KL) divergence as in [CHZT19a]. Second, the *Control Areas* are located at different orientations relative to the *Target Areas*. In total, three *Control Areas* are chosen, with 5,4 and 5 air quality boxes, respectively. We use the average $PM_{2.5}$ and PM_{10} concentrations of the three *Control Areas* as the control group for the *Target Areas*.

6.3.2 Data Collection

We collected measurements from the 55 sprayers and their co-located air quality sensing boxes in *Research Area*, including data from 7 air quality sensing boxes from *Target Area 1*, and 8 from *Target Area 2*. Meanwhile, data from the 14 (5 + 4 + 5) air quality sensing boxes data in the three

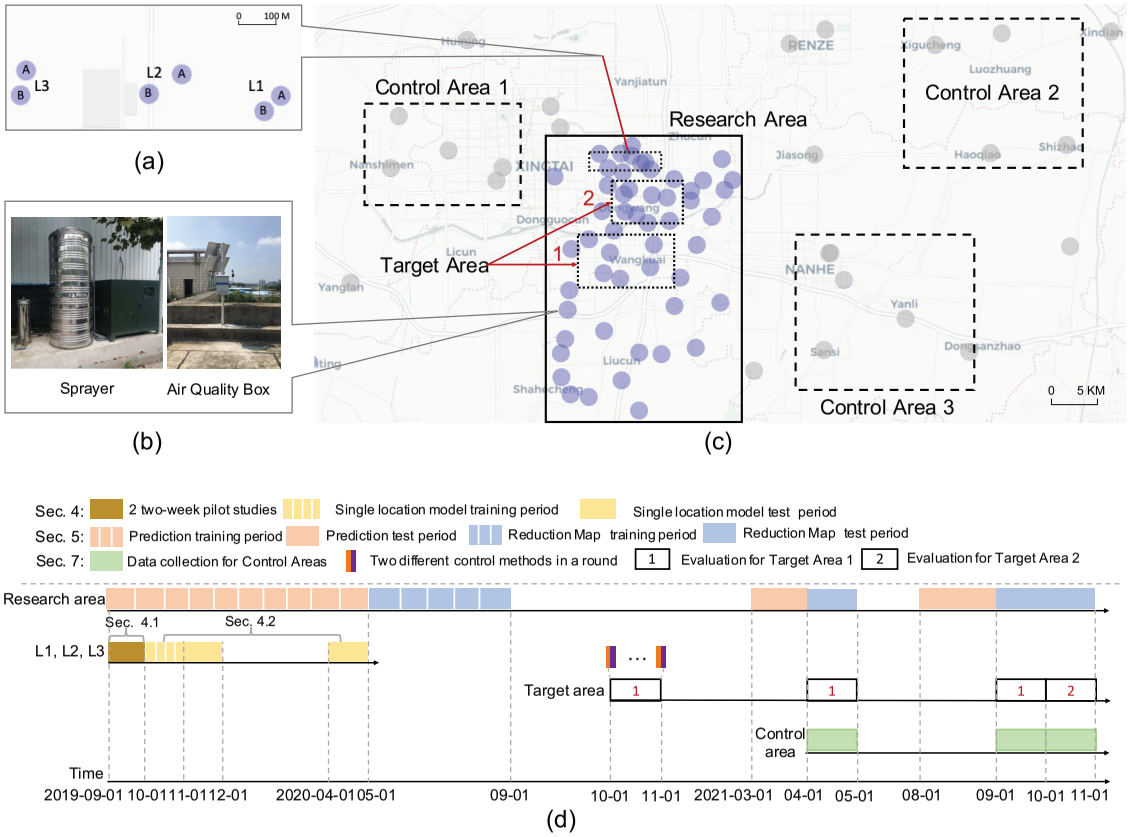


Figure 6.3: Summary of data collection campaigns. (a) Sprayers deployed at three locations. L_1 is factory, L_2 is roadside and L_3 is residential area. At each location, there are two closely deployed sprayers (also with air quality sensing boxes) A and B. We use these sprayers to analyze and model spraying-induced pollution reduction at single locations. (c) Overall deployment. Each site within the Research Area consists of a sprayer and a co-located air quality sensing box as shown in (b). Each site in the Control Area only has an air quality sensing box. The Target Areas are sub-areas of the Research Area for pollution control. (d) Time split of data campaigns.

Control Areas are also collected. The dataset contains the following data collected spanning from September 1st, 2019 to November 1st, 2021.

- **Air quality and local weather data:** We sample real-time air quality and weather-related readings from the air quality sensing boxes at every minute. The air quality readings include $PM_{2.5}$, and PM_{10}. The local weather information includes Air Temperature (AT), Relative Humidity (RH), Air Pressure (AP), Wind Speed (WS), and Wind Direction (WD) measured at the location of the sensing box. Prior research [ZLH13a, CLL+14b, CDL+19a, MLX+20] showed that these factors affect the $PM_{2.5}$ and PM_{10} concentration. The values of all the weather variables are normalized to the range of $[0, 1]$.

- **Sprayer data:** We record sprayer data including sprayer operating status, which is either *on* or *off*, as well as the usage of water and electricity. The sampling rate is also every minute.

- **Forecast weather data:** In addition to the local weather data sampled at each air quality sensing box, we also collect public weather *forecast* data [1] for the entire region of interest. These weather records contain $1km \times 1km$ grid-level air temperature, relative humidity, air pressure, wind speed, and wind direction for every hour. These data will be used in the *air quality map prediction* module (see Section 6.5.1).

Figure 6.3-(d) summarizes our data collection campaigns.

- **Data collection for single-location pollution reduction**. We use data correspond to the three locations L_1, L_2, and L_3 in Figure 6.3-(a) for characterizing and modeling air pollution at single locations (see Section 6.4). The selection of these three locations is deferred to Section 6.4.1. Its data collection period is from September 1st, 2019 to April 30th, 2020. Specifically, two two-week pilot studies, from September 1st, 2019 to September 15th, 2019, and from September 16th, 2019 to September 30th, 2019, respectively, are adopted to analyze the spraying-induced air pollution reduction at single locations (see Section 6.4.1). Afterwards, we use the data collected from October 2019, as well as from November 2019 (Autumn dataset) and April 2020 (Spring dataset), to train and test our single-location air pollution reduction model .

- **Data collection for pollution reduction map generation**. We use data collected from the *Research Area* for training and testing air pollution map generation (Section 6.5). Specifically, we use the data from September 1st, 2019 to April 30th, 2020 for training the pollution map prediction without spraying (Section 6.5.1) and data in March and August 2021 for testing (Section 6.5.4). Similarly, we

[1] `https://www.ecmwf.int/en/forecasts/datasets` visited 2021-11-01

use the data from May 1st, 2020 to August 31st, 2020 for training the pollution reduction map (Section 6.5.2), and data in April, September, and October 2021 for testing (Section 6.5.4).

- **Data collection for sprayer scheduling**. Note that the scheduling algorithm of iSpray does not involve training other than the above models for pollution reduction (see Section 6.6). Therefore, we only need datasets for testing. Specifically, we use the data collected *(i)* in October 2020 from *Target Area 1* to compare different scheduling strategies (see Section 6.7.2); *(ii)* in April 2021 and September 2021 from *Target Area 1* to test the performance of iSpray scheduling (see Section 6.7.3); and *(iii)* in October 2021 from *Target Area 2* to test the generalization of iSpray scheduling (see Section 6.7.4). Meanwhile, we collect the data from the *Control Areas* for the corresponding months as the control group, i.e., without any water spraying.

Note that both $PM_{2.5}$ and PM_{10} are particulate matters and the only difference lies in size of the particle. Also $PM_{2.5}$ is more critical to the human health [XXSL16]. Therefore, in the rest of this chapter, we will mainly use $PM_{2.5}$ to illustrate our technical details, but provide the evaluations for PM_{10} mainly in Section 6.7.

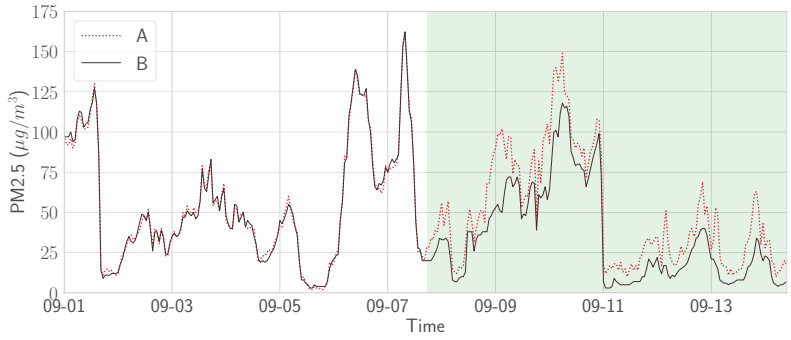
6.4 Characterizing Spraying on Single-Spot Air Pollution Reduction

In this section, we conduct preliminary studies to answer the following two questions: *(i)* Does water spraying reduce air pollution at single POIs? *(ii)* Can we model the amount of air pollution reduction at a single POI as a function of sprayer time and other environmental factors? We answer these questions with data collected at the three locations in Figure 6.3-a.

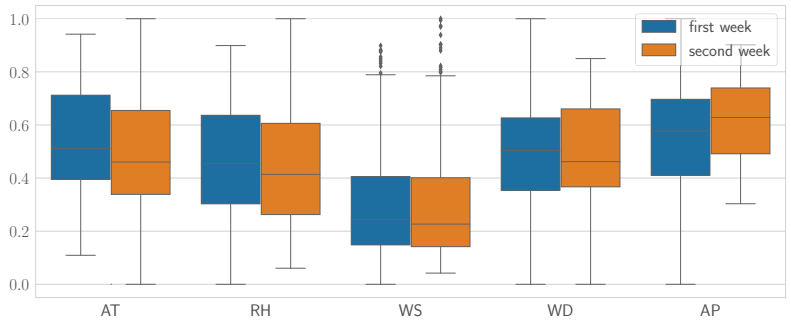
6.4.1 Water Spraying Suppresses Air Pollution at Single Locations

We first investigate whether water spraying notably decreases air pollution concentrations in the outdoor open air via two field studies.

Pollution Reduction over Time. We randomly choose six sprayers (with co-located air quality sensing boxes) from the *Research Area* in Figure 6.3-(c) for a two-week field study (from September 1st, 2019 to September 15th, 2019). Specifically, sprayers from three locations are selected, where there are two closely deployed (< 100 meters) sprayers at these three locations (see Figure 6.3-(a)). The sprayers labeled as A at each location are used as the control group. That is, they are kept switch off during the entire two weeks. The sprayers labeled as B at each location are switched off in the first week and switched on in the second week. We use the $PM_{2.5}$ and PM_{10} concentrations as well as the local weather data (i.e., air



(a)



(b)

Figure 6.4: Impact of water spraying on hourly averaged $PM_{2.5}$ concentration measured at two closely deployed sprayers (A and B) at L_1. (a) $PM_{2.5}$ concentrations measured at A and B over time. Both A and B were switched off in the first week and B was switched on in the second week (portions with green background). (b) Distributions of local weather data in the first and the second week.

temperature AT, relative humidity RH, air pressure AP, wind speed WS, wind direction WD) for this study.

Figure 6.4-(a) plots the $PM_{2.5}$ concentrations measured at sprayer A and B at location L_1 (factory) in these two weeks. We average the minute-resolution $PM_{2.5}$ values into hourly resolution to highlight the general trend over two weeks. In the first week, where both sprayers were switched off, the mean absolute difference in the $PM_{2.5}$ readings of sprayer A and B is within $0.5\mu g/m^3$. In contrast, this difference in $PM_{2.5}$ concentration increases to $13.0\mu g/m^3$ for the second week, where sprayer A remained off while sprayer B was switched on (portions with green background in Figure 6.4-(a)). Similar results are observed for sprayer A and B at location L_2 and L_3. Specifically, for location L_2, the mean absolute difference between sprayer A and B is $0.6\mu g/m^3$ in the first week, and $10.7\mu g/m^3$ in the second week. For location L_3, the mean absolute difference between sprayer A and B is $0.8\mu g/m^3$ in the first week, and $9.5\mu g/m^3$ in the second week. The significant change in the $PM_{2.5}$ measurements at the two closely deployed sprayers indicates that water spraying notably affects the air pollution.

Table 6.2: Results of t-tests for weather conditions.

(a) p-values for weather conditions between the first and second week.

	AT	AP	RH	WD	WS
L_1	0.32	0.44	0.66	0.69	0.79
L_2	0.33	0.42	0.65	0.71	0.77
L_3	0.30	0.42	0.66	0.72	0.76

(b) p-values for weather conditions between sprayer A and B.

	AT	AP	RH	WD	WS
L_1	0.90	0.90	0.89	0.92	0.93
L_2	0.89	0.91	0.90	0.92	0.93
L_3	0.92	0.90	0.89	0.93	0.94

The difference in $PM_{2.5}$ concentration might be caused by notable changes in environmental conditions in the first and the second week. For example, factors such as *wind* are known to affect the spatiotemporal distribution of $PM_{2.5}$ concentrations [ZLH13a, CDL+19a]. Figure 6.4-(b) plots the distributions of the weather data (i.e., air temperature AT, relative humidity RH, wind speed WS, wind direction WD and air pressure AP in the first and the second week. It is observed that the weather conditions are similar for the first and the second week. It implies that the change in $PM_{2.5}$ concentrations at the two sprayers is mainly due to change in sprayer status i.e., B was switched on in the second week. As a more quantitative measure, we use t-test to assess the difference in weather conditions across both weeks for L_1, L_2 and L_3. For each weather variable measured at each location, measurements in the first week and the second week are used as the two independent inputs for the t-test. The hypothesis is that two independent samples have identical average (expected) values and a p-value larger than 0.05 is explained as a positive signal to support the hypothesis. Table 6.2a shows the p-values for all meteorological variables, which range from 0.30 to 0.79. Therefore, the measurements of weather conditions from these two weeks can be considered drawn from the same distribution, i.e., similar to each other.

Table 6.3 shows the difference in pollution concentrations measured at sprayer A and B at the three locations in the second week. As is shown, water spraying decreases $PM_{2.5}$ and PM_{10} concentrations by over 20% at representative urban POIs, which is considered remarkable improvements in air pollution control [GS18].

Pollution Reduction at Finer Time Granularity. In this field study, the

Table 6.3: Difference in hourly averaged $PM_{2.5}$/PM_{10} concentration of sprayer B compared with sprayer A in the second week. $(A \rightarrow B)$/*reduction* means the mean value changes from A to B, and the reduction percentage.

	L_1 (Factory)	L_2 (Roadside)	L_3 (Residential)
$PM_{2.5}$	$(49.5 \rightarrow 36.5)/-26.2\%$	$(45.0 \rightarrow 34.3)/-23.6\%$	$(41.2 \rightarrow 31.8)/-22.9\%$
PM_{10}	$(65.8 \rightarrow 46.7)/-29.1\%$	$(62.1 \rightarrow 46.3)/-25.4\%$	$(55.0 \rightarrow 42.3)/-23.1\%$

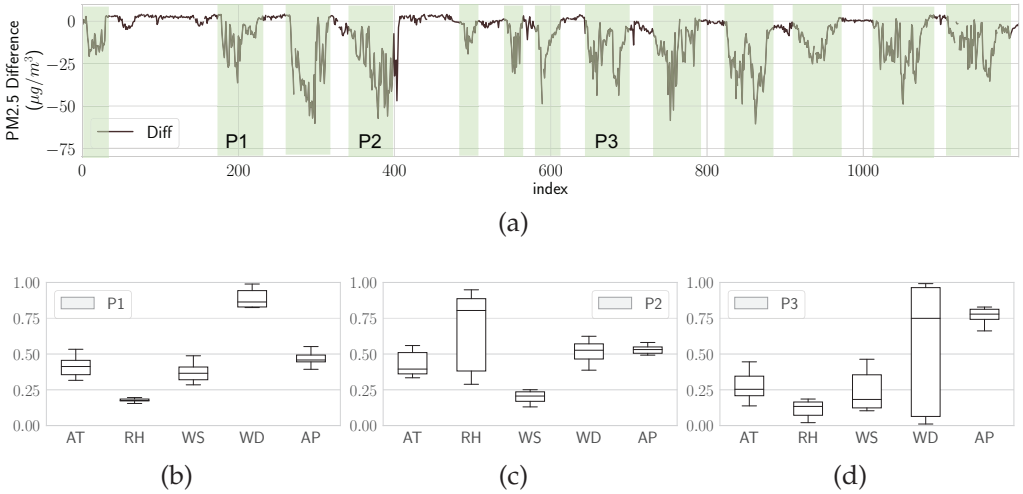


Figure 6.5: Illustration of weather-dependent pollution reduction: (a) difference of $PM_{2.5}$ between B and A at location $L1$, where $P1$ to $P3$ are three random periods of the same duration when B is switched on; (b)-(d): local weather data during $P1$-$P3$.

setups follow those in the above section, except that instead of keeping the sprayers B at locations L_1 to L_3 switched on continuously in the second week, we regularly switched these sprayers on and off for a random duration from 15 minutes to 24 hours. The study was conducted from September 16th, 2019 to September 30th, 2019. The local weather conditions can be considered as similar between two nearby locations (i.e., A and B at each location) during these two weeks. As a quantitative measure, we conduct a t-test for all weather variables between A and B. The p-values are between 0.89 to 0.94, which are larger than 0.05 (see Table 6.2b), indicating the weather data at A and B are similar. Thus, the difference of pollution concentrations between sprayer A and B at these locations is primarily due to spraying.

Figure 6.5-(a) plots the difference of the $PM_{2.5}$ concentrations (averaged for every 15 minutes) between B and A for these 14 days. The zones colored in green are periods with sprayer B switched on. We make the following observations. *(i)* The $PM_{2.5}$ difference in the uncolored zones is almost zero, meaning the $PM_{2.5}$ concentrations at A and B are almost the same. This is expected because A and B experience similar

weather conditions and there is no air pollution reduction by water spraying during these periods. *(ii)* The $PM_{2.5}$ difference in the green zones ranges from $5.5\mu g/m^3$ to $62.0\mu g/m^3$. The air pollution reduction owes to water spraying. However, the amount of reduction varies over time. To understand the reasons for such variations, we investigate the air pollution reduction from three random periods of the same duration (10 hours), P_1 to P_3 in Figure 6.5-(a). Our hypothesis is that the spraying-induced pollution reduction is weather-dependent. Figure 6.5-(b) to Figure 6.5-(d) show the local normalized weather data during the three periods P_1 to P_3. The average $PM_{2.5}$ reduction in these three periods are $8.8\mu g/m^3$, $22.9\mu g/m^3$, and $15.4\mu g/m^3$, which notably differ. The local weather data during these three periods also vary. For example, the wind direction of $P1$, $P2$ and $P3$ differs from each other (with mean values of 0.81, 0.49 and 0.62). This indicates pollution propagates to location L_1 from different locations during $P1$, $P2$ and $P3$, which might partially explain the difference in spraying-induced pollution reduction. In fact, the heavy precipitation and strong wind in $P2$ facilitates pollution dispersion and increases the pollution reduction rate. The analysis implies that the varied $PM_{2.5}$ reduction in the same time duration at the same location attributes to the difference in *local weather conditions*, as will be shown next.

6.4.2 Modeling Air Pollution Reduction at Single Locations

From the field studies in Section 6.4.1, water spraying reduces air pollution at single locations but the reduction varies and is likely *weather-dependent*. In this subsection, we aim to quantify the *accumulative pollution reduction* over time as a function of *weather conditions*. We prefer modeling *accumulative* to *instant* pollution reduction since the accumulative reduction model facilitates decisions on whether to switch off a sprayer after a given period. That is, given a time slot Δt at sprayer s_k and all the needed features, the air pollution reduction model will predict the accumulative pollution reduction $Q_{s_k}(\Delta t)$.

Neural Network Based Pollution Reduction Model. To model the accumulative pollution reduction as a function of weather conditions, we explore both linear (multi-variant linear regression) and non-linear (neural network) models. Specifically, we feed all the forecast weather data as input features. Additionally, we also include *(i)* pollution levels features, such as $PM_{2.5}$ levels; *(ii)* POI features such as *road, park, factory*; *(iii)* time unit features and *(iv)* date features such as *hour of day*. Previous studies [CHZT19a, HSW+15] show that these features also benefit air pollution related modeling. Table 6.4 summarizes all the input features.

Comparisons of Single-Location Pollution Reduction Models. We empirically explore whether the non-linear or the linear model is suited for single-location pollution reduction.

Table 6.4: List of input features for single-location pollution reduction.

Category	Features
Weather	air temperature, air pressure, wind speed, wind direction etc.
Pollution Level	We defined 6 discrete PM2.5 levels [CLL+14b]
POI	We selected 10 common POIs from [Inc21]
Time Unit	Time unit after the opening of spraying system
Date	Hour-of-Day, Day-of-Week, Month-of-year, isHoliday

Table 6.5: Accuracy of single-location pollution reduction models ($\mu g/m^3$).

#hours	Model	Autumn ($RM \rightarrow$)			Spring ($RM \rightarrow$)		
		L_1	L_2	L_3	L_1	L_2	L_3
2	linear (linear regression)	20.2	25.6	19.5	25.9	24.3	29.0
	non-linear (neural network)	3.2	3.4	3.9	4.6	4.1	3.5
4	linear (linear regression)	28.3	32.1	26.7	34.9	33.1	38.0
	non-linear (neural network	5.9	7.1	4.3	6.1	6.5	4.8
6	linear (linear regression)	33.2	35.6	38.1	40.3	48.8	45.6
	non-linear (neural network	6.3	7.8	5.2	6.1	8.0	9.1

We collect data from L_1, L_2, and L_3 during October 2019 for training, and November 2019 and April 2020 for testing. During these periods, the sprayers were set to be switched on when the $PM_{2.5}$ concentration exceeded $35\mu g/m^3$, the excellent air quality level defined in Section 6.7.1; and switched off when the $PM_{2.5}$ concentration dropped below the threshold. We define the data from November 2019 as the Autumn dataset and the one from April 2020 as the Spring dataset. The architecture and hyperparameters of the neural network are automatically optimized using grid-search in Sweeps[2], the final structure used for MLP model is 24(*input layer*) $\times$ 35($first$ *hidden layer*) $\times$ 10(*second hidden layer*) $\times$ 1(*output layer*) with a dropout rate of 0.2.

Table 6.5 shows the accuracy of single-location pollution reduction models (RM) on predicting next 2 to 6 hours reduction values using the Autumn and Spring test sets. RM is trained using the data from L_1 on Autumn dataset and used to test the performance from L_1 to L_3. Neural network works best for all test sets with MAE errors ranging from 3.2 to 9.1, much less than the results from linear model (MAEs from 19.5 to 48.8). Also, we can find that neural network generalize well in a different season (Spring) and locations (L_2 and L_3). Those results reveal the necessity of using neural network in modeling the single-location pollution reduction.

As a case study, we also test the above models on data collected from

[2]https://docs.wandb.ai/guides/sweeps

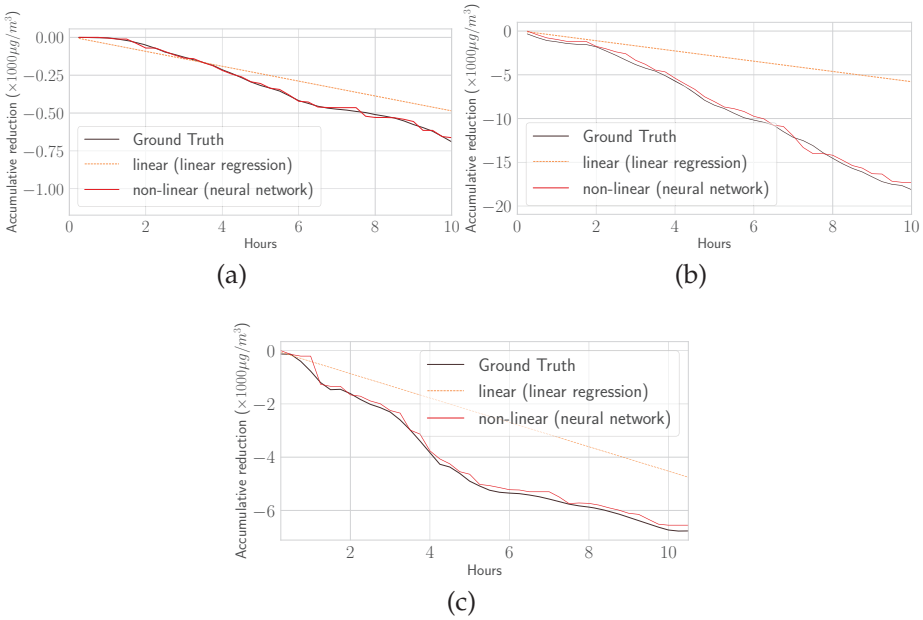


Figure 6.6: Testing single-location pollution reduction models on (a) $P1$ (b) $P2$ and (c) $P3$ in Figure 6.5-(a).

$P1$, $P2$, and $P3$ in Figure 6.5-(a). As shown in Figure 6.6, the linear model fails to capture the complex relationship between the input features and the accumulative $PM_{2.5}$ reduction (MAE of 59.2), whereas the estimations of the neural network are highly accurate (MAE of 5.5).

6.5 Spatial Modeling of Water Spraying on Air Pollution

In addition to the pollution reduction at single locations, we also need to model how water spraying affects the *spatial* distribution of pollutant concentration so as to schedule the sprayers for effective pollution control at key urban POIs. Specifically, suppose a set of sprayers are switched on at time t and operate for Δt, we aim to generate an *air pollution reduction map* to depict spraying-induced pollution reduction *in space* at time $t + \Delta t$. In this section, we first present a scheme for air quality map prediction without water spraying (Section 6.5.1), based on which we propose an accurate air pollution reduction map generation (Section 6.5.2) and its parameter learning method (Section 6.5.3). Finally, we present the evaluations for air pollution reduction map generation (Section 6.5.4).

6.5.1 Air Quality Map Prediction without Water Spraying

Although a new air quality map prediction model (without water spraying) is not our focus, highly accurate predictions are important because they will be used for parameter learning of the pollution reduction model (see Section 6.5.3) and pollution propagation path

generation. In response, we adapt a state-of-the-art air quality prediction model [LBC20] which is built upon convolutional long-short-term-memory (convLSTM) modules [SCW+15]. Specifically, we add two modifications to improve the prediction accuracy. *(i)* We feed the model with air quality readings from a dense deployment rather than a sparse one to improve the sensor data quality. *(ii)* We design a new weather encoder module to better incorporate the weather influence on air quality changes. Figure 6.7 shows our air quality prediction model called *Air-convLSTM*. Assume that air quality map data and weather data are both on grid-level with shape of $(M \times N)$, the historical length and prediction steps are equal as τ, our model consists of the following submodules:

- Air Encoder: it takes the historical air quality map data as input with shape of $(\tau \times M \times N)$ and produces the hidden encoding state H_{air} with shape of $(M \times N \times |H_{air}|)$, where $|H_{air}|$ denotes its hidden dimension.

- Weather Encoder: it inputs the gird-level weather data with shape of $(\tau \times M \times N \times N_{wea})$, where N_{wea} is the weather data dimensions. We use the hidden state of each convLSTM cell as the output of this encoder with the shape of $(\tau \times M \times N \times |H_{wea}|)$, where $|H_{wea}|$ is the hidden dimension of the weather encoder.

- Air Decoder: it takes the air encoder results as input and produces an output with the shape of $(\tau \times M \times N \times |H_{air\_dec}|)$, where $|H_{air\_dec}|$ is its hidden state dimension.

- Weather Fusion: For each decoder step $i, i \in (1 \dots \tau)$, concatenate the hidden state of air decoder and weather encoder and prepare the input to a fully-connected network (*FCN*). The input dimension is $(M \times N \times |H_{air+wea}|)$, where $|H_{air+wea}| = |H_{air\_dec}| + |H_{wea}|$. The FCN will incorporate the weather influence on air quality changes and produce the adapted values with shape of $(M \times N)$ at each decoding step. The overall dimension of prediction map is $(\tau \times M \times N)$.

6.5.2 Building Air Pollution Reduction Map with Domain Knowledge

Following the conventions in the air pollution map generation literature [CLL+14a, HSW+15], we discretize the entire 2-dimensional region of interest into grids $\{g\}$. Consider K sprayers deployed in the entire region where a set of sprayers are switched on at time t and will be operating for duration Δt, our aim is to estimate the reduction $R(g)$ in pollutant concentration for every grid at time $t + \Delta t$. We consider a grid size of $1km \times 1km$ and a time resolution of 1 hour because *(i)* $1km \times 1km$ is widely used in related research [ZLH13a, ZYL+15a, CDL+19a]; *(ii)* 1 hour is a common time resolution to evaluate the air quality. We use Q_{s_k} to

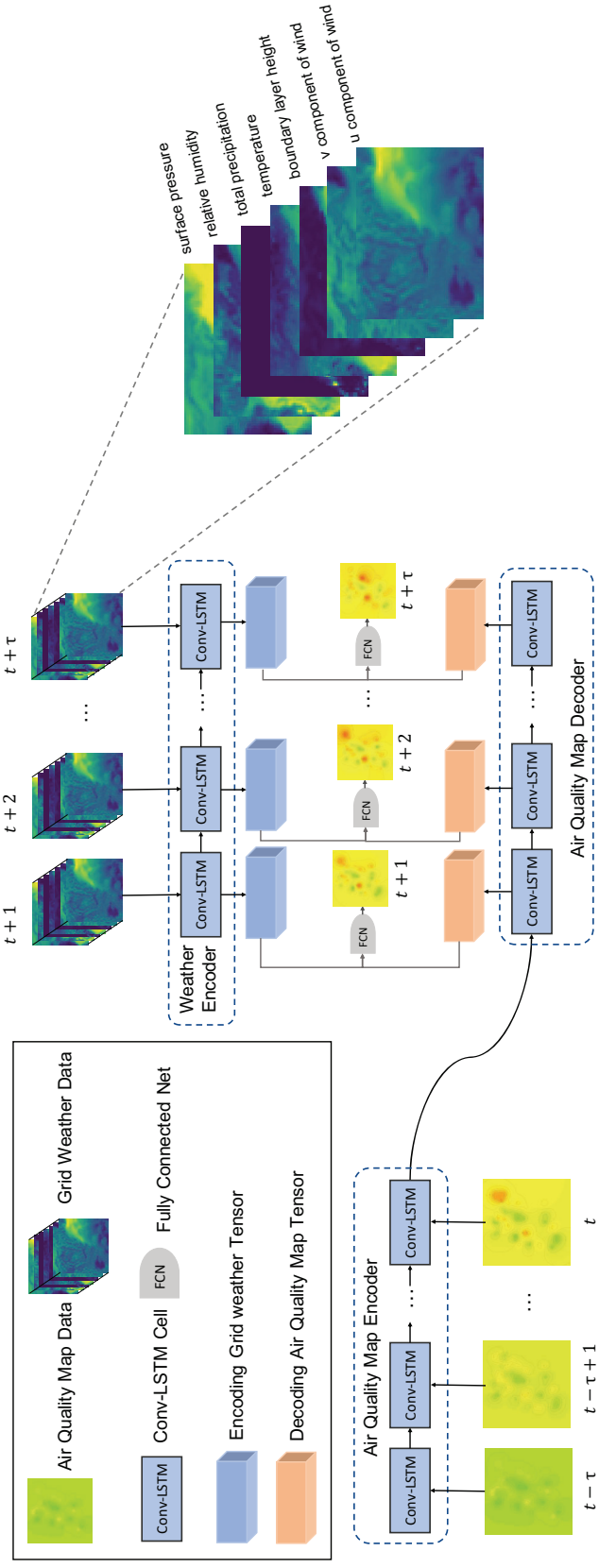


Figure 6.7: Air quality map prediction model (*Air-convLSTM*) used in iSpray.

represent the accumulative pollution reduction at a single location over Δt hours afterwards.

One may integrate the sprayer data with emission source and weather data to directly learn an air quality map prediction model. We *separately* consider pollution absorption and dispersion due to *limited water spraying data* for effective training. The limited water spraying data also motivate us to model air pollution reduction maps with *domain knowledge*. We empirically compare the accuracy of our approach with jointly learning of both pollution absorption and dispersion in Section 6.5.4.

6.5.2.1 Spatial Pollution Reduction of a Single Sprayer

We first model the pollution reduction $r(g|s_k)$ in grid g due to sprayer s_k. The model is inspired by the classical Gaussian plume model to assess the impacts of emission sources on urban air pollution [Zan90, Ary04]. Specifically, the Gaussian plume model describes the pollution dispersion $c(g|e)$ in grid g (in 2-dimension) due to an emission source e as a Gaussian distribution in vertical directions.

$$c(g|e) = \frac{Q_{air}}{2\pi\sigma\bar{w}} \exp\left(-\frac{1}{2}\left(\frac{d_{cross}(g,e)}{\sigma}\right)^2\right) \tag{6.1}$$

where Q_{air} is the pollution emission rate, $d_{cross}(g,e)$ is the crosswind distance [3] between g and the grid of e. $\bar{w}$ is the average horizontal wind speed, and σ is the Gaussian plume dispersion parameter, which is a function of the downwind distance $d_{down}(g,e)$ between g and the grid of e (see footnotes for definition).

Since pollution *absorption*, i.e., pollution reduction due to water spraying in our case, can be considered as the inverse process of *dispersion*, we hypothesize the pollution reduction $r(g|s_k)$ in grid g due to sprayer s_k behaves similar as in Eq. (6.1). Due to the difficulty to obtain parameters such as δ, we modify the original Gaussian plume model to characterize pollution reduction $r(g|s_k)$ in grid g due to sprayer s_k as follows:

$$r(g|s_k) = \frac{Q_{s_k}}{2\pi\phi(d_{down}(g,g_k), f_{eta})\bar{w}_{s_k}} \exp\left(-\frac{1}{2}\left(\frac{d_{cross}(g,g_k)}{\phi(d_{down}(g,g_k), f_{eta})}\right)^2\right) \tag{6.2}$$

where Q_{s_k} is the pollution reduction over time period Δt in the grid where sprayer s_k is installed, which is modeled as Section 6.4.2, $d_{down}(g,g_k)$ and $d_{cross}(g,g_k)$ are the downwind and crosswind distances between g and the grid g_k of sprayer s_k, respectively. $\bar{w}_{s_k}$ is the average horizontal wind speed. $\phi(.)$ is a learnable function to determine δ. The input to $\phi(.)$ is downwind

[3] Let's make a 2-D coordinate axis with the wind direction as x and the orthogonal one as y, which centers at e. For grid g, the distance between the vertical mapping of g to x axis and e is called the downwind distance, while the distance between vertical mapping of g to y axis and e is called the crosswind distance.

distance $d_{cross}(g, g_k)$ and extra features f_{eta} as described in Table 6.4. We use a multi-layer perceptron (MLP) to implement the function of $\phi(.)$. The detailed parameter learning procedure is deferred to Section 6.5.3.

6.5.2.2 Spatial Pollution Reduction of Multiple Sprayers

Consider K sprayers deployed in the entire region of interest. Then the total pollution reduction $R(g)$ in grid g over time period Δt is given by:

$$R(g) = \sum_{k=1}^{K} I(s_k) c(g|s_k) \tag{6.3}$$

where $I(s_k)$ is an indicator function of the operating status o_k of sprayer s_k, i.e.,

$$I(s_k) = \begin{cases} 1 & \text{if } o_k \text{ is } on \\ 0 & \text{if } o_k \text{ is } off \end{cases} \tag{6.4}$$

6.5.3 Parameter Learning for Air Pollution Reduction Maps

Although our air pollution reduction map modeling is built upon domain knowledge, some parameters are still difficult to access, which are captured by the MLP parameters in $\phi(.)$ of Eq. (6.2). This subsection explains how to learn these MLP parameters (see Figure 6.8). Our idea is to first generate the air quality map due to pollution dispersion $C_d(g)$ for $t + \Delta t$ via the air quality prediction model in Section 6.5.1. Then we calculate the air quality reduction map $R(g)$ at $t + \Delta t$ following Eq. (6.3). The final air quality map at $t + \Delta t$ is calculated as:

$$\hat{C}(g) = C_d(g) - R(g) \tag{6.5}$$

This map can be compared with the ground truth air quality map $C(g)$ by interpolating the air quality sensor measurements at $t + \Delta t$. The difference between these two maps enables us to update the parameters in $\phi(.)$.

As we will show in Section 6.5.4, the air quality map prediction model without spraying influence is accurate for small Δt. It means the main air quality estimation error comes from $R(g)$, i.e., the MLP parameters we would like to learn. Given the sensor measurements at $t + \Delta t$, we can generate the ground truth air quality map $C(g)$ by Gaussian processes [Ras04]:

$$C(g) \sim \mathcal{GP}(\boldsymbol{m}, \boldsymbol{v}) \tag{6.6}$$

where $\boldsymbol{m}$ and $\boldsymbol{v}$ are the mean and covariance function, respectively. Gaussian process based interpolation proves highly accurate with a dense pollutant sensor deployment [CLL+14b, CHZT20a], which is the case in our scenario. We define the loss function as the difference between ground truth $C(g)$ and predicted one $\hat{C}(g)$. Using this procedure and loss function, we can successfully learn the parameters in $\phi(.)$ by optimizing and decreasing the loss.

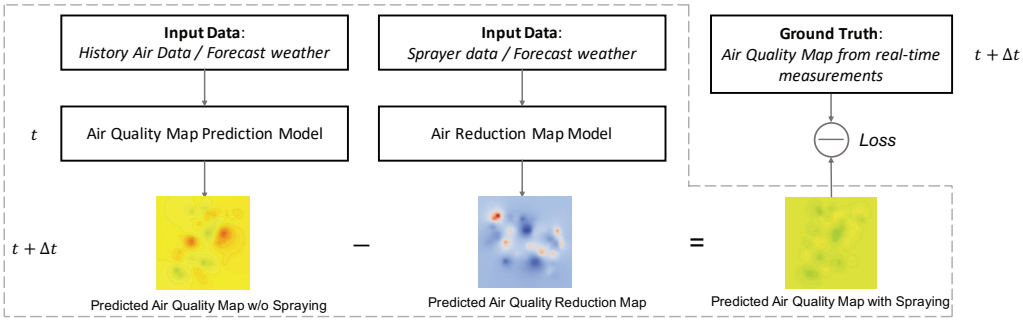


Figure 6.8: Parameter learning in air pollution reduction map $R(g)$. The air quality map without water spraying $C_d(g)$ is generated via the air quality prediction model in Section 6.5.1. The ground truth air quality map $C(g)$ is generated by interpolating the air quality sensor measurements.

6.5.4 Evaluations on Air Pollution Reduction Map Generation

As mentioned, accurate air pollution reduction map generation is crucial for effective sprayer scheduling. Next, we assess the accuracy of air pollution map generation ignoring and considering water spraying in sequel.

Effectiveness of Air Quality Map Prediction without Spraying. We first evaluate the air quality map prediction without considering water spraying.

We compare our *Air-convLSTM* (see Section 6.5.1) with three baselines:

- *Naïve:* use the current timestamp value as the predictions for future hours.
- *ConvLSTM:* use historical air quality map as input and ConvLSTM [SCW+15] to predict future maps [LBC20],
- *w-ConvLSTM:* concatenate weather maps with air quality maps, and use ConvLSTM [SCW+15] for prediction.

We use the air quality sensor box data of the *Research Area* from September 1st, 2019 to April 30th, 2020 for training of each algorithm. Then we use the data in March, 2021 as the *Spring* test dataset and those from August, 2021 as the *Summer* test dataset. We assess the air quality map prediction accuracy for the next 6 hours since the prediction for the next 6 hours suffice for our scheduling algorithm (see Section 6.6).

Table 6.6 shows the MAEs for air quality map prediction. *Air-ConvLSTM* acquires the best overall prediction accuracy in both test sets. Compared with *w-ConvLSTM*, our method decreases the prediction error of $PM_{2.5}$ and PM_{10} by 25.0% and 46.4% in Spring test period, and 40.0% and 46.2% in Summer test period. More importantly, *Air-ConvLSTM* also successfully predicts the air quality changing patterns.

Table 6.6: Accuracy of air quality map prediction (without water spraying) measured by MAE.

Model	Spring (Mar. 2021)		Summer (Aug. 2021)	
	$PM_{2.5}$ ($\mu g/m^3$)	PM_{10} ($\mu g/m^3$)	$PM_{2.5}$ ($\mu g/m^3$)	PM_{10} ($\mu g/m^3$)
Naive	4.5	5.6	3.8	4.9
ConvLSTM	3.4	4.7	3.2	4.1
w-ConvLSTM	1.6	2.8	1.5	2.6
Air-ConvLSTM	**1.2**	**1.5**	**0.9**	**1.4**

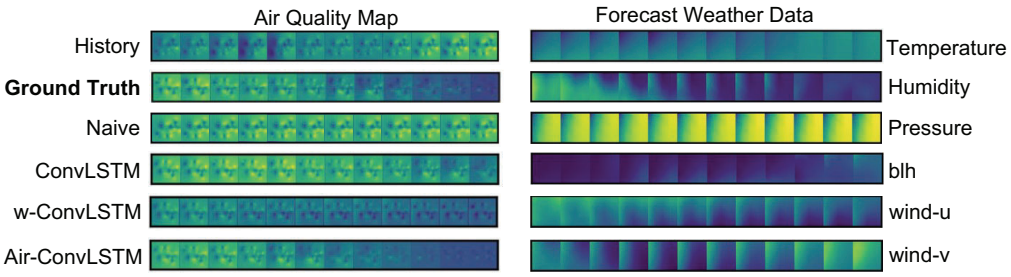


Figure 6.9: Air quality map prediction case with weather data.

i.e., forecasting the air quality map sudden changing time slots and pollution evolving patterns. This is the key prerequisite for the success of the air pollution propagation path algorithm in Section 6.6.2. Figure 6.9 shows an example of predicting the pollution sudden change patterns using all methods. We can find that *ConvLSTM fails to predict accurately without using the weather data*. *w-ConvLSTM* partially solves the problem and improves the performance by including weather features. However, simply concatenating weather data fails to learn the air quality changing patterns, and leads to constant good air quality predictions in all future hours as shown in Figure 6.9. Instead, *Air-ConvLSTM* concatenates the weather encoder information with the air quality map predictions, thus predicting the changing patterns from bad to good with precise time slots. This greatly helps for the pollution propagation path detection and thus the overall success of iSpray.

We also evaluate the impact of prediction steps (in hours) and training data length on the accuracy of *Air-ConvLSTM*. As shown in Figure 6.10-(a), increasing the prediction steps from 1 to 8, the MAE of *Air-ConvLSTM* increases from 0.6 to 1.8. We choose to predict the next 6 hours in iSpray because the prediction MAE 1.4 is relatively low and 6-hour predictions are also suitable for our scheduling algorithm (see Section 6.6). When increasing the training data length from 2 to 8 months, the prediction MAE of *Air-ConvLSTM* decreases from 2.3 to 1.4 (see Figure 6.10-(b)). This is expected because more historical data improves prediction accuracy.

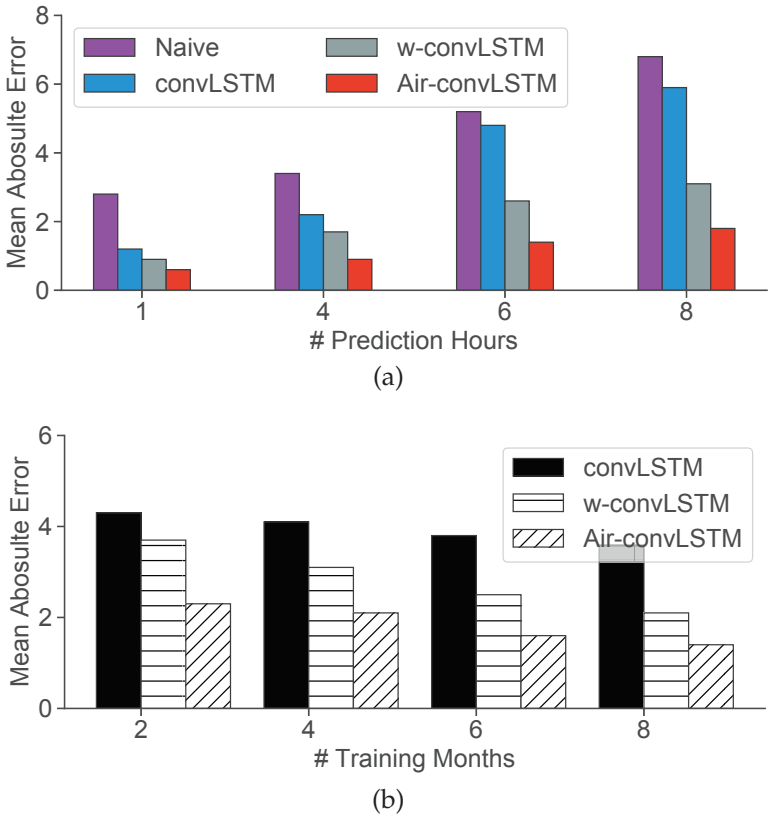


Figure 6.10: Parameter study of air quality map prediction. (a) Impact of prediction steps; (b) Impact of training data amount.

Effectiveness of Spraying-induced Air Pollution Reduction Map. Next we verify the effectiveness of our air pollution reduction map for iSpray. The main aim is to show the necessity to separate pollution reduction map generation as in Section 6.5.2.

We compare our method for pollution reduction map generation with the following three baselines:

- *Nävе:* use the current air quality map as the prediction of next step with spraying influence.
- *Land Use Regression:* use Land use regression model [HSW+15] and spraying information as input and predict the air quality map at next step.
- *Prediction-based:* concatenate spraying data to the input of FCN module in *Air-ConvLSTM* for prediction.

We use data from the *Research Area* during May 1st, 2020 to August 31st, 2020 for training, and data collected in April 2021 (Spring dataset), and September 1st, 2021 to October 31st, 2021 (Autumn dataset) for testing. The sprayer scheduling strategies for the training and testing periods are as follows. During the training period, we follow the same sprayer

Table 6.7: Air pollution reduction map accuracy comparison for $PM_{2.5}$ ($\mu g/m^3$) measured by MAE.

Dataset	Naïve	Land use regression	Prediction-based	Reduction Map
Spring	16.4	14.3	9.8	**1.9**
Autumn	19.2	15.6	11.9	**2.6**

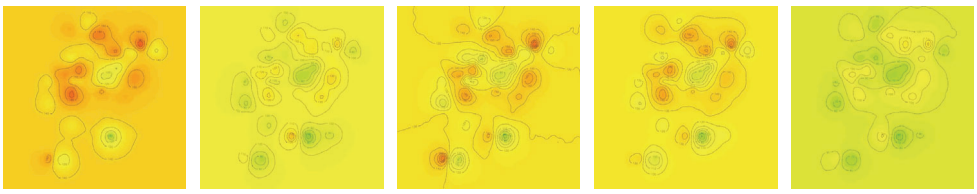


Figure 6.11: Air pollution reduction map accuracy comparison case study: (a) air quality map at t, which is also the output of Naïve; (b) ground truth air quality map at $t + 1$; (c) Land use regression baseline; (d) Prediction-based baseline; (e) iSpray.

scheduling scheme as described in Section 6.4.2, i.e., opening the sprayer once the local air quality in above the good air quality threshold. During the testing periods, the sprayer devices are operated by following the schedule timetable produced by iSpray Section 6.6.2.

Given the current air quality sensing box measurements and the sprayer status information for the next hour, the task is to predict the air quality map in the next hour. We use MAE to quantify the prediction accuracy of each algorithm.

Table 6.7 shows the overall results. Land use regression model and Prediction-based model fail to generate accurate air quality map with the spraying influence. By decomposing the problem into air quality map prediction without spraying influence and spraying-based air quality modeling, iSpray successfully learns the unknown parameters in the model and make accurate air quality map with a small amount of data. iSpray achieves MAEs of 1.9 and 2.6 for Spring and Autumn datasets, a significant improvement over all the baselines.

Figure 6.11 shows the example maps generated by different methods. iSpray benefits from the air pollution reduction model and generates the most accurate map with spraying influence, which is an essential component for our scheduling algorithm.

6.6 Cost-Effective Sprayer Scheduling

The air pollution reduction map (Section 6.5) enables us to quantify the impact of switching on each sprayer on the spatial distribution of pollutant concentrations. We now present our cost-effective sprayer

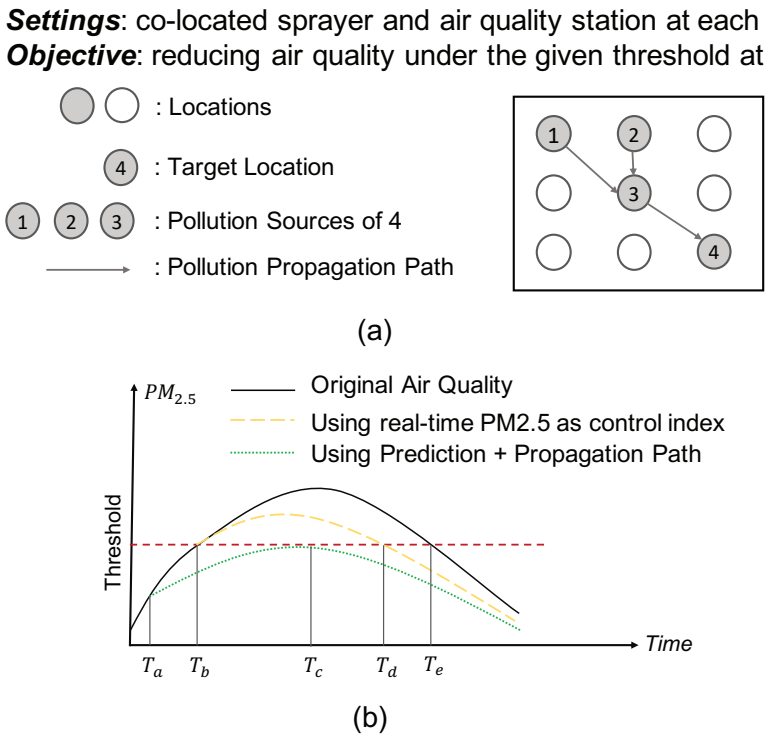


Figure 6.12: Example of cost-effective scheduling intuition: (a) an example setting; (b) scheduling based on real-time $PM_{2.5}$ values fails, while switching on the sprayer (and those along the pollution propagation path) in advance may keep the $PM_{2.5}$ concentration within the threshold with less total switch-on time.

scheduling scheme to the control the air pollution at key POIs with minimal number of operating sprayers.

6.6.1 Feasibility of Cost-Effective Sprayer Scheduling

Given certain POIs in the region of interest, we aim to make a spraying schedule for the next τ hours (i.e., whether each sprayer should be switched on or off in each hour) such that *(i)* the pollution concentrations in the grids where the POIs reside are within a given threshold and *(ii)* the total switch-on hours are minimized. Minimizing the total switch-on hours is necessary because a single sprayer consumes 120 kWh electricity and 14.4 m^3 water if operating non-stop in a day. We explain the intuitions for cost-effective scheduling via an example below.

Figure 6.12-(a) shows a simplified setting of our problem, where the space is partitioned into 9 grids and there is a sprayer and a co-located air quality sensing box in each grid. Our goal is to decide the scheduling timetable for all the 9 sprayers such that the $PM_{2.5}$ concentration in the target grid i.e., grid 4 in Figure 6.12-(a) is under a given threshold, which is shown by the red dotted line in Figure 6.12-(b). One scheduling strategy is

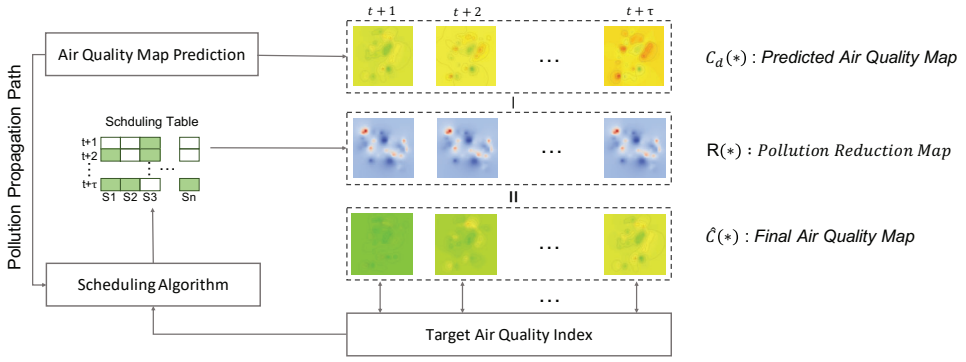


Figure 6.13: Overview of our cost-effective sprayer scheduling scheme.

to decide whether to switch on a sprayer according to the real-time $PM_{2.5}$ measurements at the co-located air quality sensing box. Suppose the real-time $PM_{2.5}$ concentration in grid 4 exceeds the threshold at time T_b. This strategy will then switch on the sprayer in grid 4 at time T_b till T_d, when the real-time $PM_{2.5}$ concentration falls below the threshold, as shown by the yellow dotted line in Figure 6.12-(b). Since the pollution reduction is not instant, this method will fail to keep the $PM_{2.5}$ concentration within the threshold during T_b to T_d. Our solution is to switch on the sprayer in grid 4 *in advance* as well as the a set of sprayers *along the pollution propagation path* towards grid 4. That is, we switch on the sprayer in grid 4 at time T_a, when the $PM_{2.5}$ concentration is still within the threshold. This way, the peak $PM_{2.5}$ concentration at T_c will be under the threshold, as shown by the green dotted line in Figure 6.12-(b). Note that the switch-on time of the sprayer in grid 4 can be short if certain sprayers along the propagation path (i.e., grid 1, 2 and 3, where the arrow denotes the direction of pollution propagation) have been switched on before the pollution propagates to grid 4. The example implies the following:

- Switching on sprayers based on real-time pollution concentration fails to keep the pollution at target POIs under control due to delays in spraying-induced pollution reduction. Therefore, it is important to predict the future pollution concentration and switch on the sprayers in advance.

- Switching on sprayers along the pollution propagation path holds promise to keep the pollution at target POIs under control and reduce the total switch-on time by suppressing the pollution near the source.

6.6.2 Scheduling Method

Inspired by the motivation example in Section 6.6.1, Figure 6.13 illustrates our cost-effective sprayer scheduling scheme. It first predicts the air

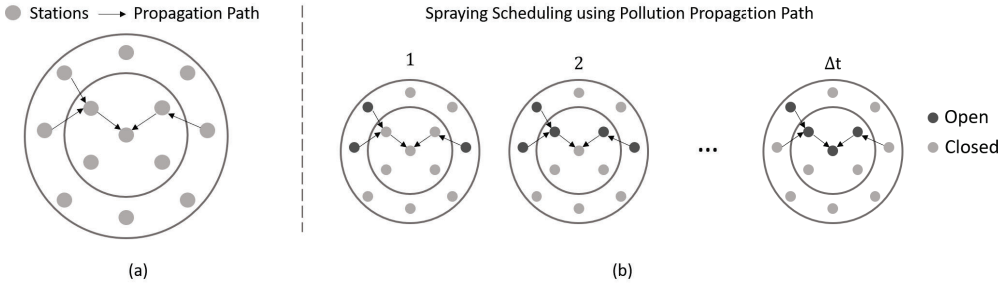


Figure 6.14: (a) Air pollution propagation path to one target location. (b) one example of spraying scheduling along the air pollution propagation path.

quality maps for the next τ hours without water spraying (Section 6.5.1) and then generate the pollution propagation path towards the target grids. Then we take the predictions, propagation path and the given threshold to make a scheduling timetable to guarantee the air quality in the target grids with small total sprayer switch-on time. We give the implementation details below.

Deriving Pollution Propagation Path. We identify pollution propagation paths by adapting the method in [LCCC17]. The key observation is that if the uptrend interval (pollution propagation) of grid a is ahead of b, then a is considered a causal parent node of b. Their method builds causal graphs and finds the *top-k* patterns from all generated graphs using historical data. These patterns are the *statistically* frequent pollution propagation behaviors. For our case, however, we should identify the predicted pollution propagation patterns *at the current timestamp*. Given the spatiotemporal air quality data, we aim to estimate the pollution propagation paths. Algorithm 2 illustrates the entire process.

- Firstly, we characterize the causal parent nodes of each grid as [LCCC17]. However, we add two more constraints: *(i)* the maximum values of parent nodes should be larger than the ones of child node; and *(ii)* the distance between them should be smaller than some threshold. These two constraints facilitate finding the pollution propagation path for a single timestamp.

- Secondly, we introduce the *pollution influence circles* to iteratively find the causal parent nodes of target grid from inner circles to outside ones (see the circles centered at the target grid in Figure 6.14-(a)).

- Finally, we apply the above two steps for each target grid, and then we can derive the propagation path, as the arrows shown in Figure 6.14-(a).

Algorithm 2: Deriving propagation paths

Input: Predicted air quality readings at all locations for next Δt hours, pollution influence circles list PC, target locations L, distance threshold d
Output: Air pollution propagation paths $Path$ to target locations L

for *each target location* L_i *in* L **do**
  $Path_i = [L_i]$; `// initialize the propagation path for location` L_i
  Conduct pollution influence circles (C_j, $j \in (1 \dots l)$, where smaller index represents small diameter) centered in target location L_i using the given values in PC (black circles in Figure 6.14);
  for $j \in (1 \dots l)$ **do**
    Find all stations S_c located in C_j apart from stations in $Path_i$;
    for *for each station* x *in* S_c *and each* y *in* $Path_i$ **do**
      if *(the distance between* $(x, y) \leq d$*) and (max value of* $x \geq$ *max value of* y*) and the uptrend interval of* x *is ahead of* y **then**
        add station x to $Path_i$
      end
    end
  end
end
Concatenate all propagation path $Path_i$ and get the overall propagation path $Path$
return $Path$

Putting it Together. Algorithm 3 illustrates our sprayer scheduling algorithm, which generates a timetable for all the sprayers in the next τ hours. The algorithm works as follows.

- For each target location, we first compute the highest prediction pollution concentration in next τ hours. If the peak concentration exceeds the threshold, the propagation path estimation module is called to identify the propagation path to this target location, e.g., the arrows in Figure 6.14-(a).
- Given the pollution propagation path in previous step, we greedily decide the sprayer status. We start with the first time slot and set the sprayers along the propagation path as open if the concentrations in the grids of these sprayers are above the threshold. After each time slot, we update the future air quality predictions to incorporate the influence of spraying. We continue the process till the predicted pollution concentrations are below the threshold or all sprayers are switched on. One example scheduling timetable is shown in Figure 6.14-(b).
- We apply the same pipeline to all target locations until their predicted peak concentrations are below the threshold or all sprayers are used.

Visualization of iSpray Scheduling. We use one real water spraying control case to illustrate the effectiveness of the scheduling algorithm in

Algorithm 3: Scheduling algorithm

Input: Predicted air quality map in next τ hours; target locations L; threshold value V_{thres}; air quality map prediction model and air pollution reduction map model

Output: Schedule timetable T_{skd} for all sprayer systems in next τ hours

1 For all target locations L, find the highest predicted air quality readings V_{L_i} at location L_i in next τ hours, assume the timestamp is Δt
2 **if** $V_{L_i} > V_{thres}$ **then**
3 Conduct the pollution propagation path for target location L_i using Algorithm 2
4 **for** $t' \in (1 \ldots \Delta t)$ **do**
5 Set sprayer to open if the local predicted $PM_{2.5}$ values are greater than V_{thres}
6 Conduct air pollution reduction map for t' using the method in Section 6.5
7 Update air quality map predictions after the current timestamp
 `// update predictions after each scheduling step to incorporate the influence of spraying`
8 **end**
9 Get the scheduling timetable T^i_{skd} for location L_i
10 **end**
11 Go to step **1** until convergence or all sprayer systems have been scheduled.
12 Concatenate all scheduling timetable T^i_{skd} and get the overall scheduling timetable T_{skd}
13 **return** T_{skd}

iSpray, i.e., Algorithm 3. Assume the current timestamp is t and we try to decide the scheduling timetable for next τ hours (in our case, $\tau = 6$) to suppress the air pollution in the target area. Following the scheduling algorithm in Algorithm 3, iSpray works as follows to produce the scheduling timetable for all spraying systems.

- iSpray first predicts the air quality maps for the next 6 hours as shown in first row of Figure 6.15. The highest prediction values for target area is in $t + 4$, so $\Delta t = 4$.
- Using the propagation path found using Algorithm 2, iSpray schedules the sprayers from the sources to target area step by step, and generates the new air quality map with spraying influence. iSpray only schedules those sprayers along the propagation path instead of all where the predicted concentrations are above the threshold.
- After deciding the sprayers to switch on in the next 4 hours, iSpray generates new air quality maps with spraying influence (third row in Figure 6.15). The air quality readings in the target area are now below the threshold, so iSpray terminates.

We can see that iSpray only schedules the necessary sprayers along the propagation path which affect the *Target Area*. Therefore our sprayer

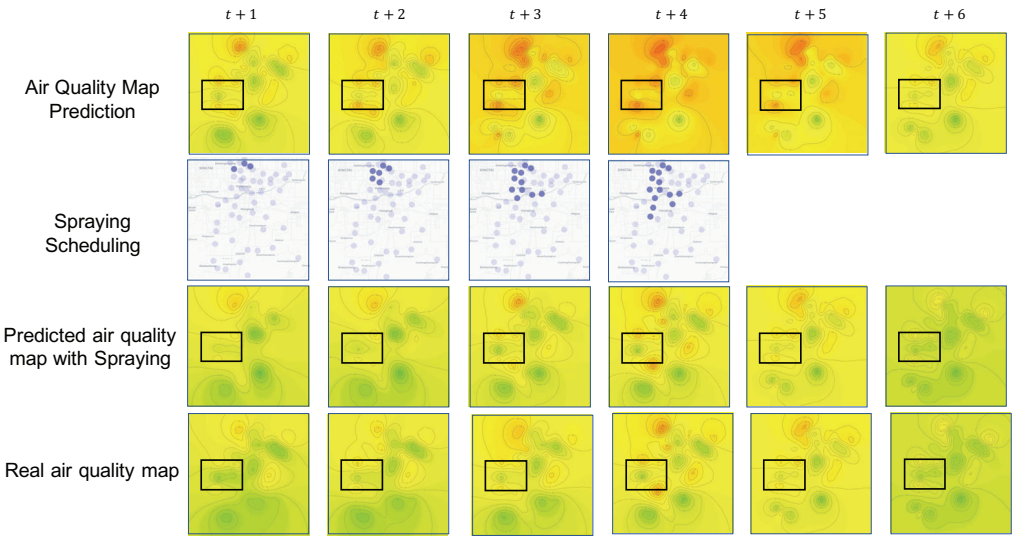


Figure 6.15: One real water spraying control use case using iSpray. Black box represents the *Target Area*.

scheduling is cost-effective. The difference between the predicted air quality map and the ground truth one (generated by Gaussian process interpolation, see Section 6.5.3) is small (see Section 6.5.4 and Section 6.5.4 for quantitative results), which also validating the effectiveness of iSpray.

6.7 Evaluation of iSpray Scheduling

This section evaluates the scheduling of iSpray and discuss its limitations and extensions.

6.7.1 Overall Experiment Setups

Since it is impossible to test sprayer scheduling schemes simultaneously at the same location, we test sprayer scheduling in the *Target Areas* and use the *Control Areas* as the control group without spraying to derive quantitative performance metrics. Note that the scheduling algorithm involves training expect those for air quality prediction and pollution reduction map generation. In the following evaluations, these models are trained using the datasets in Section 6.5. We only explain the detailed setups for the testing datasets in each experiment below.

We use water usage (m^3) and electricity usage (kWh) to compare the cost-effectiveness of different sprayer scheduling methods. We use the mean of real-time and 24-hour average value of $PM_{2.5}$ and PM_{10} ($\mu g/m^3$), as well as the excellent quality rate of $PM_{2.5}$ and PM_{10} ($PM_{2.5} \leq 35\mu g/m^3$ according to China Strandard [4], $PM_{10} \leq 40\mu g/m^3$ as adopted in the chapter) to assess the air quality, which are also used as the threshold in

[4]China National Standard: https://healthandsafetyinshanghai.com/china-air-quality/

our model. One sprayer consumes 5 kW electricity 0.6 m^3 water per hour. We consider a scheduling resolution of an hour, as in Section 6.6.1. For all the evaluations below, we use the average performance of the three *Control Areas* shown in Figure 6.3-(c) to mitigate the impact of relative orientations to the *Target Areas*.

6.7.2 Performance of Different Scheduling Algorithms

We mainly compare iSpray with the baseline method that controls sprayers based on the real-time pollution concentrations measured at the co-located sensing box, which is denoted as *Real-Time-Values* afterwards.

Setups. We test these two sprayer scheduling schemes in October 2020 in *Target Area 1*. For fair comparison, we choose the first 30 days and split them into 15 pairs. In each pair of two days, we randomly choose iSpray or the baseline for scheduling in the first day and the other for the second day. Therefore we have 15 test rounds in total, as shown in Figure 6.3-(d). We report the average performance of these 15 test rounds.

Results. Table 6.8 summarizes the performance of iSpray and the *Real-time-values* baseline. If all the sprayers are operating non-stop for 15 days, the water and electricity usage are $11,880\,m^3$ $(0.6 * 24 * 55 * 15)$ and $99,000\,kWh$ $(5*24*55*15)$, respectively. Both scheduling schemes notably reduce the usage of water and electricity, where our iSpray requires only $3,326\,m^3$ water and $24,309\,kWh$ electricity, which reduces the water and electricity usage by 34.8% and 42.3% compared with the *Real-time-values* baseline. Meanwhile, the mean values of real-time $PM_{2.5}$ and PM_{10} decrease by $6\mu g/m^3$ and $10\mu g/m^3$, which accounts for 15.0% and 21.7%. The mean values of 24-hour average $PM_{2.5}$ and PM_{10} also decrease by $5\mu g/m^3$ and $8\mu g/m^3$, which accounts for 13.2% and 18.2%. The excellent quality rate of $PM_{2.5}$ and PM_{10} increase by 13% and 16%.

6.7.3 Performance with and without iSpray Scheduling

This experiment quantifies the air pollution reduction due to iSpray. Since it is difficult to directly measure the air quality with and without water spraying at the same location and time, we adopt the distribution similarity concept [CHZT19a] for indirect comparison. Specifically, it is observed that the air quality distributions of different regions within a city are similar in the same time period [CHZT19a]. This allows us to assess the impact of water spraying for the same time period by comparing with the *Control Areas*.

Setups. We select March to April 2021 for testing in Spring, and August to September 2021 for testing in Autumn. Specifically, we switch off all the sprayers in *Target Area 1* in March 2021 and August 2021, and schedule the sprayers by iSpray in *Target Area 1* in April 2021 and September 2021.

Table 6.8: Performance comparison between different scheduling methods.

Method	Water (m^3)	Electricity (kWh)	Real-time $PM_{2.5}$/ PM_{10} ($\mu g/m^3$)	24-hour average $PM_{2.5}$/ PM_{10} ($\mu g/m^3$)	Excellent Quality Rate $PM_{2.5}$/ PM_{10}
Real-time-values (15 days)	5,108	42,571	40 / 46	38/44	72% / 75%
iSpray (15 days)	3,326	24,309	34 / 36	33/36	85% / 91%
iSpray over Real-time-values	-34.8%	-42.3%	-15.0% / -21.7%	-13.2% / -18.2%	+13% / +16%

Table 6.9: Comparison between the $PM_{2.5}$/PM_{10} distributions of *Control Areas* and *Target Area 1*.

Time	Areas	iSpray	min	1st quartile	median	3rd quartile	max
Mar. 2021	*Control*	-	4/20	39/86	60/135	100/193	194/371
	Target 1	OFF	7/18	41/87	63/133	103/187	186/366
Apr. 2021	*Control*	-	4/18	22/52	33/76	47/111	267/455
	Target 1	ON	4/13	20/33	27/46	33/65	146/379
Aug. 2021	*Control*	-	3/4	12/26	19/40	33/59	138/201
	Target 1	OFF	4/5	14/22	21/36	32/56	137/215
Sep. 2021	*Control*	-	4/5	17/41	33/76	61/129	188/326
	Target 1	ON	4/4	20/31	25/58	39/81	119/265

We use the air quality data during the same months from the three *Control Areas* as the control group (by averaging across the three *Control Areas*.

Results. We first show that the distribution similarity proposed in [CHZT19a] holds for our deployment. Specifically, we plot the air quality distributions of *Target Area 1* and the *Control Areas* in March 2021 and August 2021, when all sprayers were switched off. As shown in Figure 6.16-(a),(e) and Figure 6.16-(c),(g), the two distributions of the target and the control groups are similar. Therefore, the differences in air quality distributions during the same time period are mainly due to water spraying. This is shown in Figure 6.16-(b),(f) and Figure 6.16-(d),(h), where the only difference is that the sprayers in *Target Area 1* were switched on by iSpray. We can observe notable air pollution reduction.

More quantitatively, we use a 5-number-summary (*min,1st quartile, median, 3rd quartile and max*) to compare the distribution difference between the target area and the control area in April 2021 and September 2021. Table 6.9 summarizes the differences. According to Table 6.9, in April 2021, the $PM_{2.5}$ median and 3rd quartile in *Target Area 1* are 27 and 33, which are reduced by 21.2% and 29.8% compared with those in the *Control Areas*. The PM_{10} median and 3rd quartile of *Target Area 1* are 46 and 65, a reduction of 39.5% and 41.2% compared with those in the *Control Areas*. Similarly, in September 2021, the $PM_{2.5}$ median and 3rd quartile in *Target Area 1* are 25 and 39, which are reduced by 24.2% and 36.1% compared with those in the *Control Areas*. The PM_{10} median and 3rd quartile of *Target Area 1* are 58 and 81, yielding a reduction of 23.7% and 37.2% compared with those in the *Control Areas*.

To clearly illustrate the pollution reduction on a daily basis, we plot the daily $PM_{2.5}$ box-plots of *Target Area 1* and the *Control Areas* for April and September, 2021 in Figure 6.17-(a),(b). We have the following observations. *(i)* The $PM_{2.5}$ distributions of *Target Area 1* and the *Control Areas* are similar if the $PM_{2.5}$ concentrations are low. This is because

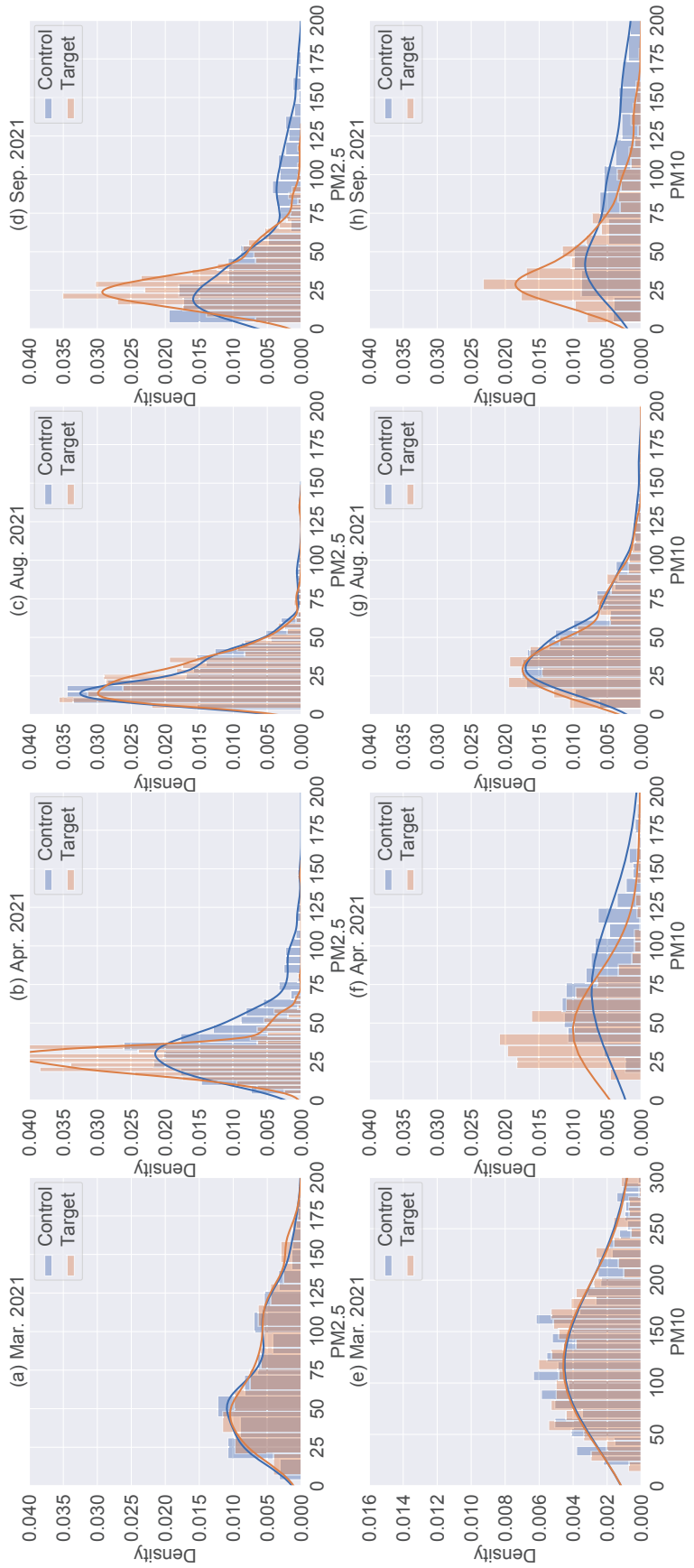
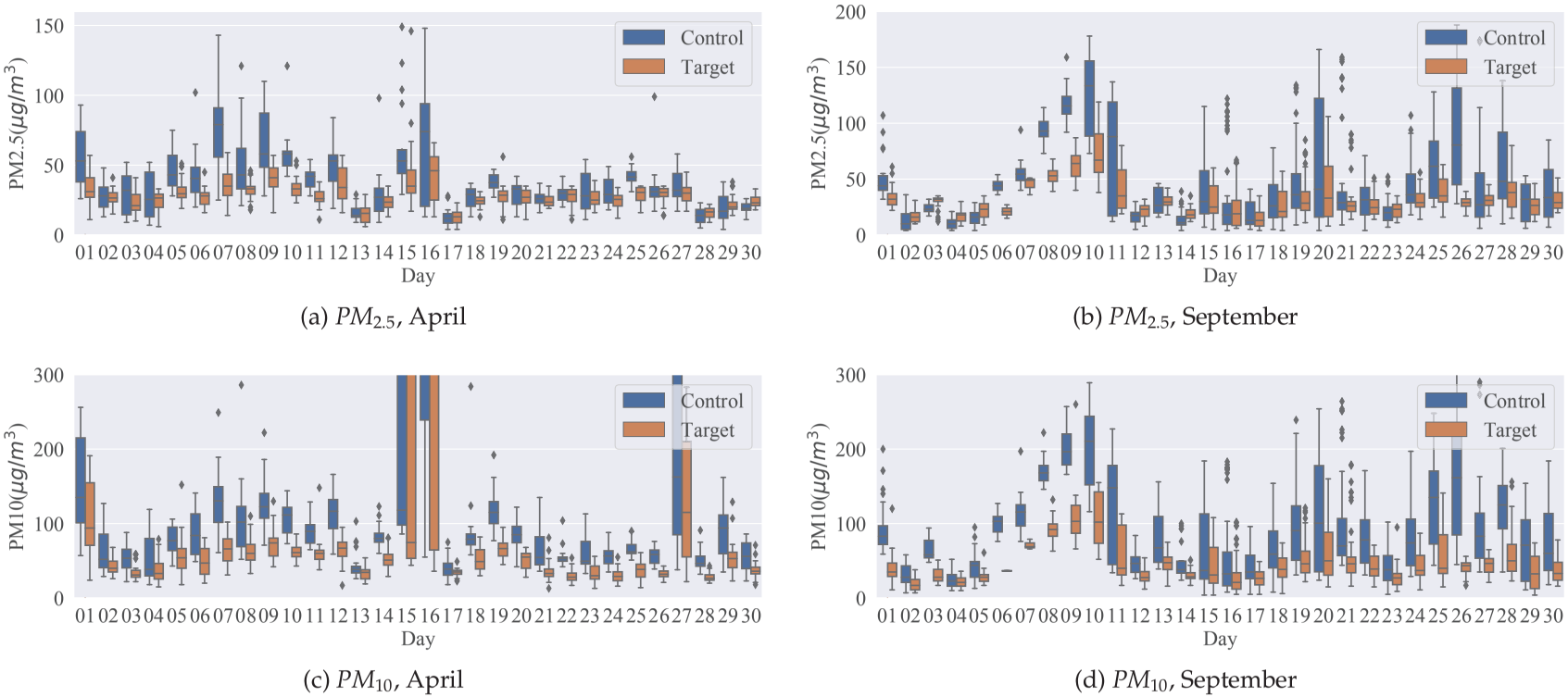


Figure 6.16: Visualization of air quality ($PM_{2.5}$/PM_{10}) distributions of the *Control Areas* and *Target Area 1*.



(a) $PM_{2.5}$, April

(b) $PM_{2.5}$, September

(c) PM_{10}, April

(d) PM_{10}, September

Figure 6.17: Daily pollution control results in Target Area 1 compared with Control Areas in 2021.

Table 6.10: Total days above excellent air quality level in Target Area 1 compared with Control Areas in 2021.

Pollution Type	April		September	
	Control Area	Target Area	Control Area	Target Area
PM2.5 ($> 35\mu g/m^3$)	14	3	19	9
PM10 ($> 40\mu g/m^3$)	28	17	29	17

iSpray will switch off the sprayers when the pollution level is low. *(ii)* The median and max values are significantly reduced during high $PM_{2.5}$ period, indicating that iSpray switches on the sprayers to suppress pollution during these times. The observations also hold for PM_{10}, as shown in Figure 6.17-(c),(d).

To further analyze the performance of iSpray in reducing the air pollution, especially its effectiveness in dropping the pollution from a polluted level to an excellent one, the total number of days above the excellent air quality level is calculated by comparing the 24-hour average value. The results in Table 6.10 show that the total number of pollutant days in April are 14 and 28 for $PM_{2.5}$ and PM_{10} in Control Area, and they are reduced to 3 and 17 days by applying iSpray in Target Area 1, which accounts for a reduction of 79% and 39%, respectively. Similarly, a reduction of 53% and 41% can also be found for $PM_{2.5}$ and PM_{10} in the September dataset.

6.7.4 Performance in Different Target Areas

This experiment demonstrates the generality of iSpray.

Setups. We select a different target area as shown in Figure 6.3-(c), denoted as *Target Area 2* to test our iSpray scheduling algorithm. The test took place in October 2021. As with Section 6.7.3, we use the air quality data of the *Control Areas* from the same period as the control group.

Results. Figure 6.18-(a) plots the pollution concentration distribution in *Target Area 2* and the *Control Areas*. We observe notable reduction of high pollution concentrations in the *Target Area 2*. Quantitatively, the $PM_{2.5}$ median and 3rd quartile in *Target Area 2* are 30 and 45, which are reduced by 23.1% and 35.7% compared with those in the *Control Areas*. The PM_{10} median and 3rd quartile of *Target Area 2* are 51 and 75, resulting in a reduction of 32.6% and 37.5% compared with those in the *Control Areas*. Figure 6.18-(b) further illustrates the daily air pollution distributions. We observe the same patterns for *Target Area 1*. The days above the excellent air quality level for Control Area are 16 and 26 for $PM_{2.5}$ and PM_{10}, and iSpray reduces them to 8 and 16 days for Target Area 2, yielding a

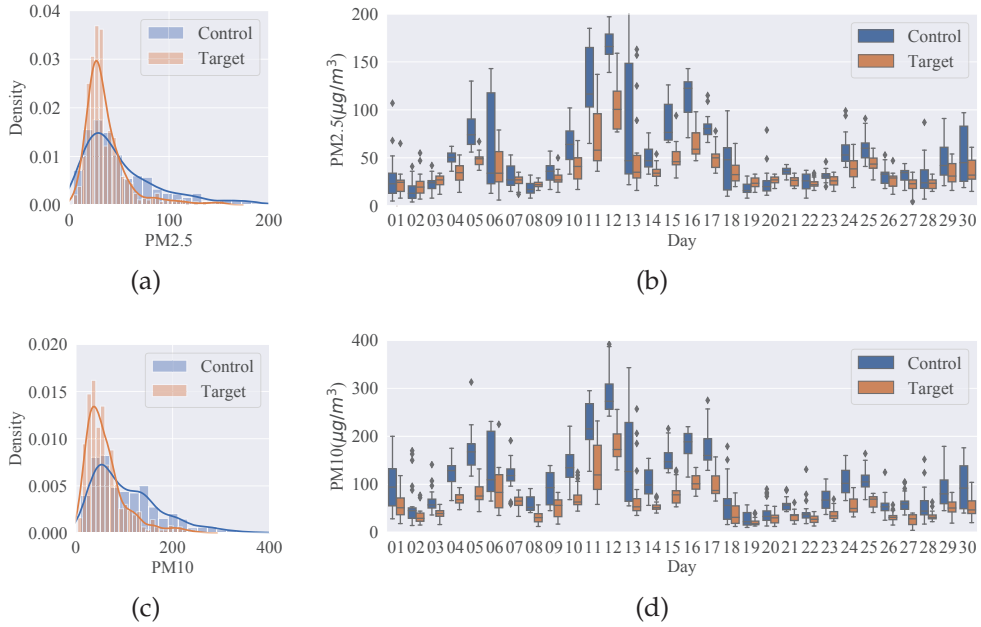


Figure 6.18: (a) $PM_{2.5}$ and (c) PM_{10} distributions of the *Control Areas* and *Target Area 2*; Daily (b) $PM_{2.5}$ and (d) PM_{10} control results comparison between *Target Area 2* and the *Control Areas*.

reduction of 50% and 31%, respectively.

In summary, the extent of pollution reduction by iSpray is similar in *Target Area 2* as in *Target Area 1*, validating the generality of our method.

6.7.5 Discussions

We briefly discuss the hyperparameter selection in iSpray and the potential extensions to mobile deployments.

Hyperparameters in iSpray. We set the threshold values (V_{thres} in Algorithm 3) of $PM_{2.5}$ and PM_{10} to $35\mu g/m^3$ and $40\mu g/m^3$, respectively, which are also the excellent air quality threshold in China, where our system is deployed. In practice, reducing the threshold values V_{thres} to near zero tends to keep all the sprayer switched on in Algorithm 3, leading to non-stop water spraying. Increasing the V_{thres} to higher values will decrease the water sprayer usage time. For our evaluation, we aim to control the pollution level according to the local standards. The spray schedule timetable of the proposed approach depends on the selected threshold, and iSpray aims to clean the local air quality to a level below the threshold. When it is impossible to achieve the local standard, all available sprayers will be open. In this case, the mobile sprayer solutions or hybrid one may help to further suppress the pollution level, as described below.

Extensions to Mobile Sprayer Deployments. In addition to the static deployments like iSpray, there is also extensive research interest in

exploiting mobile sensor deployments for air pollution monitoring [DYW+21, GDG+16, HSW+15, JLF14, MLX+20, WXL+20]. We can also extend our scheduling algorithms to mobile settings e.g., with water sprayers mounted on trucks as follows. *(i)* Derive the pollution propagation paths in the next few hours as Algorithm 2. *(ii)* Revise the schedule algorithm in Algorithm 3 with two more considerations: limited water storage and the travel time of the mobile sprayer to specific locations. One solution is to decide locations where sprayers are needed in each time slot using Algorithm 3, and then adapt existing route planning algorithms for spatial crowdsourcing [TTL+21, TZZ+18, ZTC19] to dispatch mobile sprayers to these locations at the targeting time slots while satisfying the water storage constraints.

A further extension is a hybrid mobile and static water spraying system where mobile sprayers act as backups when pollution control with static water spraying fails (line 9 in Algorithm 3). In this case, mobile sprayers can be scheduled to further suppress the pollution level.

6.8 Summary

In this chapter, we propose iSpray, a data analytics engine for $PM_{2.5}$ and PM_{10} control at critical POIs by cost-effective water spraying. Its design systematically combines domain knowledge from environmental sciences and machine learning techniques. iSpray offers learnable pollution reduction modeling at single locations, accurate air pollution reduction map generation, and propagation-path-aware sprayer scheduling. Evaluations with in-field sprayer deployments show that iSpray reduces the total sprayer switch-on time by 32%, while decreasing the days of high $PM_{2.5}$ and PM_{10} concentrations at key POIs by 12% and 16%. We envision our work as one of the first endeavors for precise urban air pollution control with ubiquitous data and commodity hardware. In practice, the pollution reduction strategies may have to refer to the results from both the intercity pollution transfer analysis (Chapter 5) and the water spraying system introduced in this chapter. Firstly, the pollution transfer result reveals the interactions between cities and provides the insight of whether to close the pollution sources in remote cities. Secondly, the local water spraying system can be applied to further reduce the pollution under health levels, especially for those key POI locations. A cost-effective and efficient pollution reduction strategy is supposed to be a combination of the above two pipelines, which could be a future research direction.

7

Conclusion and Outlook

Air quality research still remains a hot topic in recent years. Main directions include air quality monitoring, data analysis, air quality predictions and generating pollution reduction strategies, etc. Dense deployed low-cost sensor network normally consists of tens to hundreds of low-cost sensors, which measures fine-grained spatial and temporal signals at a high frequency. These large scale sensor networks are able to generate high spatial and temporal resolution air pollution data, which can be further used for data analysis such as finding the pollution sources, air quality predictions or making strategies to reduce the pollution.

However, common air quality monitoring and analysis works suffer from multiple limitations and challenges. Firstly, the accuracy and reliability of low-cost sensors drop significantly during the deployment, which hurts the follow-up data analysis tasks. Furthermore, it takes huge efforts to maintain such large-scaled sensor networks. Lastly, current intercity pollution transfer is not well characterized, and the prediction accuracy is not sufficient to generate efficient pollution reduction strategies. Also, the fine-grained pollution control with water spraying is not studied.

The aim of this thesis is to develop data-driven methods to improve the performance of air quality analysis and pollution control. We developed different calibration methods to increase the low-cost sensor accuracy. We focus on providing cost-effective downscaled deployment while maintaining a satisfactory map generation accuracy. By characterizing the pollution transfer patterns between cities and deriving pollution propagation paths among dense deployed sensors, we are able to generate accurate predictions for administrative or immediate cyber-physical response.

In the remainder of this chapter we present the main contribution of this thesis and layout possible future research directions.

7.1 Contributions

Efficient Sensor Array Calibration (Chapter 2). Low-cost sensors need frequent post-calibration to maintain the accuracy. However, current calibration approaches only include the current measurements and still suffer from the poor data quality. To improve the accuracy, a generalized many-to-many calibration scheme is presented as SensorFormer. The calibration scheme accounts for both past and future raw measurements. Evaluation shows that our approach significantly boost the performance, which can also be used in resource-constraint IOT devices.

In-field Calibration Transfer (Chapter 3). Since access to ground truth references is often limited in large-scale deployments, it is difficult to conduct city-wide post-deployment sensor calibration. We design ICT, a novel calibration scheme which transfers the calibration parameters to target sensors without access to reference data. To be the best of our knowledge, we are the first to conduct the in-field calibration transfer for large-scaled sensor deployment. Experiments show that ICT is able to calibrate the target sensors as if they had direct access to the references.

Downscaled Map generation (Chapter 4). The deployment of sensors poses many challenges due to their sheer number. One widely applied approach is to downscale the deployment size. However, dramatic accuracy degradation occurs in air quality maps generated using the downscaled sparse deployment. To overcome this problem, MapTransfer is proposed to complement downscaled sparse sensor measurements with historical information about the initial dense deployment. A learning-based pipeline is introduced to find the best-fit snapshot from history, which transfers the underlying knowledge to the current sparse deployment.

Tracking Pollution Transfer for Prediction (Chapter 5). Accurate air quality prediction is an essential step towards efficient pollution reduction strategies. Nevertheless, none of the existing data-driven methods achieves sufficient prediction accuracy for time intervals of sudden pollution change due to the inability of existing data-driven models to take into account pollution propagation between different areas caused by air mass movement. We consider pollution transfer for the first time when predicting air quality over the short term, and propose the use of air flow trajectory data in our data-driven framework. The results can be used for making pollution control strategies.

Pollution Reduction with Water Spraying (Chapter 6). Finally, the intelligent water spraying system to reduce the urban pollution is investigated. We proposed the first-of-its-kind analysis engine to profile and model how water spraying affects $PM_{2.5}$ and PM_{10} concentrations in time and space, and evaluate a cost-effective system in real-world

deployment. This work brings the gap between air quality monitoring and pollution control, and can be views as one of the first attempts for air pollution control in urban areas using ubiquitous data and commodity hardware.

7.2 Possible Future Directions

Context-aware Network Calibration. As presented in Chapter 3, network calibration for static air quality networks need to identify common contexts where it is safe to assume multiple sensors at different locations are measuring the same or similar phenomenon. The availability of big data and urban computing offers additional contextual information to identify the potential common contexts. For example, by classifying sensor locations according to their land-use context, e.g., nearby traffic, elevation or population density, a number of confident and new calibration opportunities can be increased.

Few-data Calibration. Machine learning methods, such as SensorFormer presented in Chapter 2, are becoming popular mathematical methods for air quality sensor calibration. Along with their capability to model complex non-linear relationships for sensor calibration, they require large amounts of measurements for training. This can be a burden for sensors that have limited reference samples. Some recent study [MZT18c] has exploited techniques such as semi-supervised learning to reduce the amounts of training data for sensor array calibration. However, it remains open how to reduce the measurements needed for network calibration and achieve consistent calibration accuracy for all sensors in the network.

Quantification of Uncertainty. Due to limited access to reference data in an air quality sensor network, not only is network calibration a challenge but also the evaluation of the calibration performance. Metrics such as accuracy bounds for sensor measurements [HST13] and discrete reputation scores [GBS08] can be applied in a network-wide trust model to provide a notion of quality of service of the air quality network. With an uncertainty metric, one can apply filtering methods to assure high data quality in calibration methods such as SensorFormer (Chapter 2) or calibration transfer approaches like ICT (Chapter 3). However, a unified uncertainty metric and its unified usage in air quality network calibration are still largely open.

On-demand Sensing. As we illustrated in Chapter 4, it is promising to improve the map generation accuracy for downscaled deployment by transferring the knowledge from historical data. However, the downscaled size is fixed in MapTransfer. One future work is to analyze the possibility of on-demand sensing network. i.e., the number of active sensing nodes is dynamic during the deployment and is determined

by satisfying some optimization targets. One challenge will be how to transfer the knowledge among those dynamically changing sensing networks.

Conteract air quality prediction and pollution control. Air pollution is a complex environment problem and involves various underlying evolving patters. Those pose big challenges to understand how the pollution changes and how to effectively reduce the pollution. In this thesis, we separately analyze the pollution changing behaviors between cities (TIP-Air in Chapter 5) and among urban grids (iSpray in Chapter 6). However, the best solution will be a combination of multiple analysis results, including both TIP-Air and iSpray. The research question then becomes how to fuse the derived knowledge from multimodality results and generate strategies to reduce the pollution.

Bibliography

[AKD+06] A. Analitis, K. Katsouyanni, K. Dimakopoulou, E. Samoli, A. K. Nikoloulopoulos, Y. Petasakis, G. Touloumi, J. Schwartz, H. R. Anderson, K. Cambra, et al. Short-term effects of ambient particles on cardiovascular and respiratory mortality. *Epidemiology*, 17(2):230–233, 2006.

[Ary04] N. K. Arystanbekova. Application of gaussian plume models for air pollution simulation at instantaneous emissions. *Mathematics and Computers in Simulation*, 67(4-5):451–458, 2004.

[AS08] S. H. Ahmadi and A. Sedghamiz. Application and evaluation of kriging and cokriging methods on groundwater depth mapping. *Environmental Monitoring and Assessment*, 138(1-3):357–368, 2008.

[BCB14] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[Bei18] Beijing Municipal Environmental Monitoring Center. Beijing municipal environmental monitoring center official website. `http://www.bjmemc.com.cn/`, 2018.

[BJZOV17a] 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*, 17(1):907, 2017.

[BJZOV17b] 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*, 17(1):907, 2017.

[BMLT+06] 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*, 21(6):449–458, 2006.

[BPC20] I. Beltagy, M. E. Peters, and A. Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.

[BWML18] L. Bai, J. Wang, X. Ma, and H. Lu. Air pollution forecasts: An overview. *International journal of environmental research and public health*, 15(4):780, 2018.

[CCZ+21] Z. Chen, D. Chen, X. Zhang, Z. Yuan, and X. Cheng. Learning graph structures with transformer for multivariate time series anomaly detection in iot. *IEEE Internet of Things Journal*, 2021.

[CDL+19a] 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*, 3(1):2, 2019.

[CDL+19b] 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*, 3(1):1–17, 2019.

[CDS+17] N. Castell, F. R. Dauge, P. Schneider, M. Vogt, U. Lerner, B. Fishbain, D. Broday, and A. Bartonova. Can commercial low-cost sensor platforms contribute to air quality monitoring and exposure estimates? *Environment International*, 99(Supplement C):293 – 302, 2017.

[CGCB14] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[CGRS19] R. Child, S. Gray, A. Radford, and I. Sutskever. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*, 2019.

[CHZT19a] 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*, 3(1):6:1–6:19, 2019.

[CHZT19b] 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*, 3(1):1–19, 2019.

[CHZT20a] Y. Cheng, X. He, Z. Zhou, and L. Thiele. Maptransfer: Urban air quality map generation for downscaled sensor deployments. In *Proceedings of the IEEE/ACM International Conference on Internet-of-Things Design and Implementation*, pages 14–26, Piscataway, NJ, USA, 2020. IEEE Press.

[CHZT20b] Y. Cheng, X. He, Z. Zhou, and L. Thiele. Maptransfer: Urban air quality map generation for downscaled sensor deployments. In *2020 IEEE/ACM Fifth International Conference on Internet-of-Things Design and Implementation (IoTDI)*, pages 14–26. IEEE, 2020.

[CLL+14a] 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, pages 126–134. Springer, 2014.

[CLL+14b] 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 the ACM Conference on Embedded Networked Sensor Systems*, pages 251–265. ACM, 2014.

[CLL+14c] 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 the 12th ACM Conference on Embedded Network Sensor Systems*, pages 251–265, 2014.

[CML+21] F. Concas, J. Mineraud, E. Lagerspetz, S. Varjonen, X. Liu, K. Puolamäki, P. Nurmi, and S. Tarkoma. Low-cost outdoor air quality monitoring and sensor calibration: A survey and critical analysis. *ACM Transactions on Sensor Networks (TOSN)*, 17(2):1–44, 2021.

[CMS+20] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pages 213–229. Springer, 2020.

[CPID+13] A. W. Correia, C. A. Pope III, D. W. Dockery, Y. Wang, M. Ezzati, and F. Dominici. The effect of air pollution control on life expectancy in the united states: an analysis of 545 us counties for the period 2000 to 2007. *Epidemiology (Cambridge, Mass.)*, 24(1):23, 2013.

[CSSM+19] D. A. Cory-Slechta, M. Sobolewski, E. Marvin, K. Conrad, A. Merrill, T. Anderson, B. P. Jackson, and G. Oberdorster. The impact of inhaled ambient ultrafine particulate matter on developing brain: Potential importance of elemental contaminants. *Toxicologic Pathology*, 47(8):976–992, 2019.

[CST21] Y. Cheng, O. Saukh, and L. Thiele. Tip-air: Tracking pollution transfer for accurate air quality prediction. In *Adjunct Proceedings of the 2021 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2021 ACM International Symposium on Wearable Computers*, pages 589–599, New York, NY, USA, 2021. ACM.

[CVMG+14] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[CWW+21] X. Chen, Y. Wu, Z. Wang, S. Liu, and J. Li. Developing real-time streaming transformer transducer for speech recognition on large-scale dataset. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5904–5908. IEEE, 2021.

[CWZ+17] S. Cai, Y. Wang, B. Zhao, S. Wang, X. Chang, and J. Hao. The impact of the "air pollution prevention and control action plan" on pm2. 5 concentrations in jing-jin-ji region during 2012–2020. *Science of the Total Environment*, 580:197–209, 2017.

[DBK+20] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[dCMP+17] A. del Corno, S. Morandi, F. Parozzi, L. Araneo, and F. Casella. Experiments on aerosol removal by high-pressure water spray. *Nuclear Engineering and Design*, 311:28–34, 2017.

[DWC+19] Z. Deng, D. Weng, J. Chen, R. Liu, Z. Wang, J. Bao, Y. Zheng, and Y. Wu. Airvis: Visual analytics of air pollution propagation. *IEEE transactions on visualization and computer graphics*, 26(1):800–810, 2019.

[DYW+21] R. Ding, Z. Yang, Y. Wei, H. Jin, and X. Wang. Multi-agent reinforcement learning for urban crowd sensing with for-hire vehicles. In *Proceedings of the IEEE Annual International Conference on Computer Communications*, pages 1–10, Piscataway, NJ, USA, 2021. IEEE Press.

[et] e2v technologies. Mics-oz-47 sensor. `https://gitlab.ethz.ch/tec/public/opensense/-/wikis/files/micsoz47.pdf`. Accessed: 2021-09-30.

[FFGG+16] J. Fonollosa, L. Fernández, A. Gutiérrez-Gálvez, R. Huerta, and S. Marco. Calibration transfer and drift counteraction in chemical sensor arrays using direct standardization. *Sensors and Actuators B: Chemical*, 236:1044–1053, 2016.

[FMM12] Z. L. Fleming, P. S. Monks, and A. J. Manning. Untangling the influence of air-mass history in interpreting observed atmospheric composition. *Atmospheric Research*, 104:1–39, 2012.

[FRD17] K. Fu, W. Ren, and W. Dong. Multihop calibration for mobile sensing: k-hop calibratability and reference sensor deployment. In *Proceedings of International Conference on Computer Communications*. IEEE, 2017.

[G+97] P. Goovaerts et al. *Geostatistics for natural resources evaluation*. Oxford University Press on Demand, 1997.

[GBC16] I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*. MIT press, 2016.

[GBS08] S. Ganeriwal, L. K. Balzano, and M. B. Srivastava. Reputation-based framework for high integrity sensor networks. *ACM Transactions on Sensor Networks*, 4(3):15:1–15:37, June 2008.

[GCGS21] M.-P. Gherman, Y. Cheng, A. Gomez, and O. Saukh. Compensating altered sensitivity of duty-cycled mox gas sensors with machine learning. In *2021 18th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, pages 1–9. IEEE, 2021.

[GDG+16] 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*, pages 1–9. IEEE, 2016.

[GGN+14] L. Ge, J. Gao, H. Ngo, K. Li, and A. Zhang. On handling negative transfer and imbalanced distributions in multiple source transfer learning. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 7(4):254–271, 2014.

[GLZ+18] B. Guo, J. Li, V. W. Zheng, Z. Wang, and Z. Yu. Citytransfer: Transferring inter-and intra-city knowledge for chain store site recommendation based on multi-source urban data. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 1(4):135:1–135:22, 2018.

[GNGA20] A. Golubeva, B. Neyshabur, and G. Gur-Ari. Are wider nets better given the same number of parameters?, 2020.

[Goo] Google. Tensorflow lite. `https://www.tensorflow.org/lite`. Accessed: 2021-09-30.

[Goo19] Google. TensorFlow Lite for Microcontrollers. `https://www.tensorflow.org/lite/microcontrollers` (last accessed: 2021-02-21), 2019.

[Gra69] C. W. Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica: journal of the Econometric Society*, pages 424–438, 1969.

[GS18] M. Greenstone and P. Schwarz. Is china winning its war on pollution?, 2018.

[GT19] V. L. Guen and N. Thome. Shape and time distortion loss for training deep time series forecasting models. *arXiv preprint arXiv:1909.09020*, 2019.

[GTC+15] J. Gao, H. Tian, K. Cheng, L. Lu, M. Zheng, S. Wang, J. Hao, K. Wang, S. Hua, C. Zhu, et al. The variation of chemical characteristics of pm2. 5 and pm10 and formation causes during two haze pollution events in urban beijing, china. *Atmospheric Environment*, 107:1–8, 2015.

[HBB19] H. Hewamalage, C. Bergmeir, and K. Bandara. Recurrent neural networks for time series forecasting: Current status and future directions. *arXiv preprint arXiv:1909.00590*, 2019.

[HLL+20] Y. Han, J. C. Lam, V. O. Li, P. Guo, Q. Zhang, A. Wang, J. Crowcroft, S. Wang, J. Fu, Z. Gilani, et al. The effects of outdoor air pollution concentrations and lockdowns on covid-19 infections in wuhan and other provincial capitals in china. *Preprints*, 2020.

[HLZ15] 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*, pages 437–446, 2015.

[HM06] 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*, 40(30):5902–5928, 2006.

[HST13] D. Hasenfratz, O. Saukh, and L. Thiele. Model-driven accuracy bounds for noisy sensor readings. In *Proceedings of International Conference on Distributed Computing in Sensor Systems*, pages 165–174. IEEE, 2013.

[HSW+14] 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*, pages 69–77. IEEE, 2014.

[HSW+15] 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*, 16:268–285, 2015.

[Inc21] B. Inc. Baidu poi. `http://lbsyun.baidu.com/index.php?title=androidsdk/guide/search/poi`, 2021.

[JHW+16a] W. Jiao, G. Hagler, R. Williams, R. Sharpe, R. Brown, D. Garver, R. Judge, M. Caudill, J. Rickard, M. Davis, et al. Community air sensor network (cairsense) project: Evaluation of low-cost sensor performance in a suburban environment in the southeastern united states. *Atmospheric Measurement Techniques*, 9(11):5281–5292, 2016.

[JHW+16b] W. Jiao, G. Hagler, R. Williams, R. Sharpe, R. Brown, D. Garver, R. Judge, M. Caudill, J. Rickard, M. Davis, L. Weinstock, S. Zimmer-Dauphinee, and K. Buckley. Community air sensor network (cairsense) project: evaluation of low-cost sensor performance in a suburban environment in the southeastern united states. *Atmospheric Measurement Techniques*, 9(11):5281–5292, 2016.

[JLF14] 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.

[JSBT+15] M. Jovašević-Stojanović, A. Bartonova, D. Topalović, I. Lazović, B. Pokrić, and Z. Ristovski. On the use of small and cheaper sensors

and devices for indicative citizen-based monitoring of respirable particulate matter. *Environmental Pollution*, 206:696 – 704, 2015.

[KB14] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[KBDG04] D. Kifer, S. Ben-David, and J. Gehrke. Detecting change in data streams. In *VLDB*, volume 4, pages 180–191. Toronto, Canada, 2004.

[KC89] H. Kan and B. Chen. Analysis of exposure-response relationships of air particulate matter and adverse health outcomes in china. *Journal of Environment and Health*, 1989.

[Kis03] F. N. Kissell. *Handbook for Dust Control in Mining*. NIOSH, Cincinnati, OH, USA, 2003.

[KKL20] N. Kitaev, Ł. Kaiser, and A. Levskaya. Reformer: The efficient transformer. *arXiv preprint arXiv:2001.04451*, 2020.

[KMM+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*, 75:199–205, 2015.

[KSM14] J. C. Kurnia, A. P. Sasmito, and A. S. Mujumdar. Dust dispersion and management in underground mining faces. *International Journal of Mining Science and Technology*, 24(1):39–44, 2014.

[KVL+11] D. Koracin, R. Vellore, D. H. Lowenthal, J. G. Watson, J. Koracin, T. McCord, D. W. DuBois, L.-W. A. Chen, N. Kumar, E. M. Knipping, et al. Regional source identification using lagrangian stochastic particle dispersion and hysplit backward-trajectory models. *Journal of the Air & Waste Management Association*, 61(6):660–672, 2011.

[KZS+15] R. Kiros, Y. Zhu, R. R. Salakhutdinov, R. Zemel, R. Urtasun, A. Torralba, and S. Fidler. Skip-thought vectors. In *Advances in neural information processing systems*, pages 3294–3302, 2015.

[LBC19] 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.

[LBC20] V.-D. Le, T.-C. Bui, and S.-K. Cha. Spatiotemporal deep learning model for citywide air pollution interpolation and prediction. In *Proceedings of the IEEE International Conference on Big Data and Smart Computing*, pages 55–62, Piscataway, NJ, USA, 2020. IEEE Press.

[LCCC17] 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*, pages 1863–1872, 2017.

[LCL+12] X. Liu, S. Cheng, H. Liu, S. Hu, D. Zhang, and H. Ning. A survey on gas sensing technology. *Sensors*, 12(7):9635–9665, 2012.

[LCO18] H. Liu, J. Cai, and Y.-S. Ong. Remarks on multi-output gaussian process regression. *Knowledge-Based Systems*, 144:102–121, 2018.

[LDC18a] Y. Lin, W. Dong, and Y. Chen. Calibrating low-cost sensors by a two-phase learning approach for urban air quality measurement. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 2(1):1–18, 2018.

[LDC18b] Y. Lin, W. Dong, and Y. Chen. Calibrating low-cost sensors by a two-phase learning approach for urban air quality measurement. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 2(1):18:1–18:18, 2018.

[LHH+19] Z. Luo, J. Huang, K. Hu, X. Li, and P. Zhang. Accuair: Winning solution to air quality prediction for kdd cup 2018. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1842–1850, 2019.

[LKZ+18] Y. Liang, S. Ke, J. Zhang, X. Yi, and Y. Zheng. Geoman: Multi-level attention networks for geo-sensory time series prediction. In *IJCAI*, pages 3428–3434, 2018.

[LLZ+18] 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*, 2(1):20, 2018.

[LM14] Q. Le and T. Mikolov. Distributed representations of sentences and documents. In *International conference on machine learning*, pages 1188–1196, 2014.

[LMA20] I. Lang, A. Manor, and S. Avidan. Samplenet: Differentiable point cloud sampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7578–7588, 2020.

[LMG+18] Y. Lin, N. Mago, Y. Gao, Y. Li, Y.-Y. Chiang, C. Shahabi, and J. L. Ambite. Exploiting spatiotemporal patterns for accurate air quality forecasting using deep learning. In *Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 359–368, 2018.

[LPP+20] J. Lelieveld, A. Pozzer, U. Pöschl, M. Fnais, A. Haines, and T. Münzel. Loss of life expectancy from air pollution compared to other risk factors: a worldwide perspective. *Cardiovascular Research*, 2020.

[LTZ14] S. Liu, K. Triantis, and L. Zhang. The design of an urban roadside automatic sprinkling system: Mitigation of pm2. 5–10 in ambient air in megacities. *Chinese Journal of Engineering*, 2014:1–12, 2014.

[LWLQ21] T. Lin, Y. Wang, X. Liu, and X. Qiu. A survey of transformers. *arXiv preprint arXiv:2106.04554*, 2021.

[LZC+18] X. Li, K. Zhao, G. Cong, C. S. Jensen, and W. Wei. Deep representation learning for trajectory similarity computation. In *2018 IEEE 34th International Conference on Data Engineering (ICDE)*, pages 617–628. IEEE, 2018.

[LZCY19] Y. Luo, Y. Zhang, X. Cai, and X. Yuan. E2gan: End-to-end generative adversarial network for multivariate time series imputation. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 3094–3100. AAAI Press, 2019.

[M+14] K. M. Mullen et al. Continuous global optimization in r. *Journal of Statistical Software*, 60(6):1–45, 2014.

[Mat63] G. Matheron. Principles of geostatistics. *Economic Geology*, 58(8):1246–1266, 1963.

[MC08] H. McGowan and A. Clark. Identification of dust transport pathways from lake eyre, australia using hysplit. *Atmospheric Environment*, 42(29):6915–6925, 2008.

[MCCD13] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[MHS+19] B. Maag, D. Hasenfratz, O. Saukh, Z. Zhou, C. Walser, J. Beutel, and L. Thiele. Ozone and carbon monoxide dataset collected by the opensense zurich mobile sensor network, 2019.

[MLB+12] J. G. Monroy, A. Lilienthal, J. L. Blanco, J. González-Jimenez, and M. Trincavelli. Calibration of mox gas sensors in open sampling systems based on gaussian processes. In *SENSORS, 2012 IEEE*, pages 1–4. IEEE, 2012.

[MLE+15] S. Moltchanov, I. Levy, Y. Etzion, U. Lerner, D. M. Broday, and B. Fishbain. On the feasibility of measuring urban air pollution by wireless distributed sensor networks. *Science of The Total Environment*, 502:537 – 547, 2015.

[MLX+20] R. Ma, N. Liu, X. Xu, Y. Wang, H. Y. Noh, P. Zhang, and L. Zhang. Fine-grained air pollution inference with mobile sensing systems: A weather-related deep autoencoder model. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 4(2):52:1–52:21, 2020.

[MMH17a] M. Mueller, J. Meyer, and C. Hueglin. Design of an ozone and nitrogen dioxide sensor unit and its long-term operation within a sensor network in the city of zurich. *Atmospheric Measurement Techniques*, 10(10):3783–3799, 2017.

[MMH17b] M. Mueller, J. Meyer, and C. Hueglin. Design of an ozone and nitrogen dioxide sensor unit and its long-term operation within a sensor network in the city of zurich. *Atmospheric Measurement Techniques*, 10(10):3783–3799, 2017.

[MMR19] N. Muralidhar, S. Muthiah, and N. Ramakrishnan. Dyat nets: dynamic attention networks for state forecasting in cyber-physical systems. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 3180–3186. AAAI Press, 2019.

[MSHT16a] B. Maag, O. Saukh, D. Hasenfratz, and L. Thiele. Pre-deployment testing, augmentation and calibration of cross-sensitive sensors. In *Proceedings of International Conference on Embedded Wireless Systems and Networks*, pages 169–180. ACM, 2016.

[MSHT16b] B. Maag, O. Saukh, D. Hasenfratz, and L. Thiele. Pre-deployment testing, augmentation and calibration of cross-sensitive sensors. In *Conference on Embedded Wireless Systems and Networks (EWSN)*, pages 169–180, 2016.

[MSHT16c] B. Maag, O. Saukh, D. Hasenfratz, and L. Thiele. Pre-deployment testing, augmentation and calibration of cross-sensitive sensors. In *EWSN*, pages 169–180, 2016.

[Mur12] K. P. Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[MZST17a] B. Maag, Z. Zhou, O. Saukh, and L. Thiele. SCAN: Multi-hop calibration for mobile sensor arrays. *ACM Journal on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT and UBICOMP)*, 1(2):19:1–19:21, 2017.

[MZST17b] B. Maag, Z. Zhou, O. Saukh, and L. Thiele. Scan: Multi-hop calibration for mobile sensor arrays. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 1(2):19:1–19:21, 2017.

[MZT18a] B. Maag, Z. Zhou, and L. Thiele. A survey on sensor calibration in air pollution monitoring deployments. *IEEE Internet of Things Journal*, 5(6):4857–4870, 2018.

[MZT18b] B. Maag, Z. Zhou, and L. Thiele. A survey on sensor calibration in air pollution monitoring deployments. *IEEE Internet of Things Journal*, 5(6):4857–4870, 2018.

[MZT18c] B. Maag, Z. Zhou, and L. Thiele. W-air: Enabling personal air pollution monitoring on wearables. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 2(1):24:1–24:25, 2018.

[NAB20] 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*, pages 326–327, 2020.

[OBLS14] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, pages 1717–1724. IEEE, 2014.

[oEE18] M. of Ecology and Environment. Ambient air quality standards. `https://tinyurl.com/ybqnswnc`, 2018.

[Oge20] Y. Ogen. Assessing nitrogen dioxide (no2) levels as a contributing factor to the coronavirus (covid-19) fatality rate. *Science of The Total Environment*, page 138605, 2020.

[ORR+08] 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*, pages 109–120. IEEE, 2008.

[OWS+20] Y. Ou, J. J. West, S. J. Smith, C. G. Nolte, and D. H. Loughlin. Air pollution control strategies directly limiting national health damages in the us. *Nature Communications*, 11(1):1–11, 2020.

[pan] panttwower. panttwower sensor. `http://www.plantower.com/`. Accessed: 2021-09-30.

[PARS14] B. Perozzi, R. Al-Rfou, and S. Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710, 2014.

[Pro13] D. Prostański. Use of air-and-water spraying systems for improving dust control in mines. *Journal of Sustainable Mining*, 12(2):29–34, 2013.

[PSL+17] X. Pang, M. D. Shaw, A. C. Lewis, L. J. Carpenter, and T. Batchellier. Electrochemical ozone sensors: A miniaturised alternative for ozone measurements in laboratory experiments and air-quality monitoring. *Sensors and Actuators B: Chemical*, 240(Supplement C):829 – 837, 2017.

[PTKY11] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang. Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks*, 22(2):199–210, 2011.

[PVU+18] N. Parmar, A. Vaswani, J. Uszkoreit, L. Kaiser, N. Shazeer, A. Ku, and D. Tran. Image transformer. In *International Conference on Machine Learning*, pages 4055–4064. PMLR, 2018.

[PY10a] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010.

[PY10b] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010.

[QSC+17] Y. Qin, D. Song, H. Chen, W. Cheng, G. Jiang, and G. Cottrell. A dual-stage attention-based recurrent neural network for time series prediction. *arXiv preprint arXiv:1704.02971*, 2017.

[Ras04] C. E. Rasmussen. Gaussian processes in machine learning. *Advanced Lectures on Machine Learning: ML Summer Schools*, 3176:63, 2004.

[RMS+21] A. Rives, J. Meier, T. Sercu, S. Goyal, Z. Lin, J. Liu, D. Guo, M. Ott, C. L. Zitnick, J. Ma, et al. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences*, 118(15), 2021.

[Ros07] D. Rossiter. Co-kriging with the gstat package of the r environment for statistical computing. *Web: http://www. itc. nl/rossiter/teach/R/R ck. pdf*, 2007.

[RSPB21] F. Rollo, B. Sudharsan, L. Po, and J. G. Breslin. Air quality sensor network data acquisition, cleaning, visualization, and analytics: A real-world iot use case. In *Adjunct Proceedings of the 2021 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2021 ACM International Symposium on Wearable Computers*, pages 67–68, New York, NY, USA, 2021. ACM.

[RSS+09] U. Ranft, T. Schikowski, D. Sugiri, J. Krutmann, and U. Krämer. Long-term exposure to traffic-related particulate matter impairs cognitive function in the elderly. *Environmental research*, 109(8):1004–1011, 2009.

[RSVG21] A. Roy, M. Saffar, A. Vaswani, and D. Grangier. Efficient content-based sparse attention with routing transformers. *Transactions of the Association for Computational Linguistics*, 9:53–68, 2021.

[RTMH18] F. Ramos, S. Trilles, A. Muñoz, and J. Huerta. Promoting pollution-free routes in smart cities using air quality sensor networks. *Sensors*, 18(8):2507, 2018.

[SCW+15] X. SHI, Z. Chen, H. Wang, D.-Y. Yeung, W.-k. Wong, and W.-c. WOO. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, pages 802–810, Red Hook, NY, USA, 2015. Curran Associates, Inc.

[SDR+15] A. Stein, R. R. Draxler, G. D. Rolph, B. J. Stunder, M. Cohen, and F. Ngan. Noaa's hysplit atmospheric transport and dispersion modeling system. *Bulletin of the American Meteorological Society*, 96(12):2059–2077, 2015.

[SGL+16] L. Sigrist, A. Gomez, R. Lim, S. Lippuner, M. Leubin, and L. Thiele. Rocketlogger: Mobile power logger for prototyping iot devices: Demo abstract. In *Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems CD-ROM*, pages 288–289, 2016.

[SGV+15] L. Spinelle, M. Gerboles, M. G. Villani, M. Aleixandre, and F. Bonavitacola. Field calibration of a cluster of low-cost available sensors for air quality monitoring. part a: Ozone and nitrogen dioxide. *Sensors and Actuators B: Chemical*, 215(Supplement C):249 – 257, 2015.

[SGV+17] L. Spinelle, M. Gerboles, M. G. Villani, M. Aleixandre, and F. Bonavitacola. Field calibration of a cluster of low-cost commercially available sensors for air quality monitoring. part b: No, co and co2. *Sensors and Actuators B: Chemical*, 238(Supplement C):706 – 715, 2017.

[SGX14] SGX Sensortech. MiCS-OZ-47 ozone sensor (datasheet). `http://goo.gl/C49tcw`, 2014.

[SHT15a] 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*, pages 274–285. ACM, 2015.

[SHT15b] O. Saukh, D. Hasenfratz, and L. Thiele. Reducing multi-hop calibration errors in mobile sensor networks. In *ACM/IEEE Conference on Information Processing in Sensor Networks (IPSN)*, pages 274–285, 2015.

[SLG+19] P. Schwaller, T. Laino, T. Gaudin, P. Bolgar, C. A. Hunter, C. Bekas, and A. A. Lee. Molecular transformer: a model for uncertainty-calibrated chemical reaction prediction. *ACS central science*, 5(9):1572–1583, 2019.

[SPDG+20] L. Setti, F. Passarini, G. De Gennaro, P. Barbieri, M. G. Perrone, M. Borelli, J. Palmisani, A. Di Gilio, V. Torboli, F. Fontana, et al. Sars-cov-2rna found on particulate matter of bergamo in northern italy: First evidence. *Environmental Research*, page 109754, 2020.

[SSL19] S.-Y. Shih, F.-K. Sun, and H.-y. Lee. Temporal pattern attention for multivariate time series forecasting. *Machine Learning*, 108(8-9):1421–1441, 2019.

[SWL+11] K. Song, Q. Wang, Q. Liu, H. Zhang, and Y. Cheng. A wireless electronic nose system using a fe2o3 gas sensing array and least squares support vector regression. *Sensors*, 11(1):485–505, 2011.

[TDBM20] Y. Tay, M. Dehghani, D. Bahri, and D. Metzler. Efficient transformers: A survey. *arXiv preprint arXiv:2009.06732*, 2020.

[TDMP16] R. Tian, C. Dierk, C. Myers, and E. Paulos. MyPart: Personal, portable, accurate, airborne particle counting. In *Proceedings of CHI Conference on Human Factors in Computing Systems*, pages 1338–1348. ACM, 2016.

[The18] The Public Weather Service Center of CMA. Beijing weather. `http://bj.weather.com.cn/`, 2018.

[TKPI+11] M. C. Turner, D. Krewski, C. A. Pope III, Y. Chen, S. M. Gapstur, and M. J. Thun. Long-term ambient fine particulate matter air pollution and lung cancer in a large cohort of never-smokers. *American journal of respiratory and critical care medicine*, 184(12):1374–1381, 2011.

[TNHS+03] E. Tjoe Nij, S. Hilhorst, T. Spee, J. Spierings, F. Steffens, M. Lumens, and D. Heederik. Dust control measures in the construction industry. *Annals of Occupational Hygiene*, 47(3):211–218, 2003.

[TTL+21] Q. Tao, Y. Tong, S. Li, Y. Zeng, Z. Zhou, and K. Xu. A differentially private task planning framework for spatial crowdsourcing. In *Proceedings of the IEEE International Conference on Mobile Data Management*, pages 9–18, Piscataway, NJ, USA, 2021. IEEE Press.

[TYIM05] W. Tsujita, A. Yoshino, H. Ishida, and T. Moriizumi. Gas sensor network for air-pollution monitoring. *Sensors and Actuators B: Chemical*, 110(2):304 – 311, 2005.

[TZZ+18] Y. Tong, Y. Zeng, Z. Zhou, L. Chen, J. Ye, and K. Xu. A unified approach to route planning for shared mobility. *Proceedings of the VLDB Endowment*, 11(11):1633–1646, 2018.

[Vai01] P. Vaidyanathan. Generalizations of the sampling theorem: Seven decades after nyquist. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 48(9):1094–1109, 2001.

[VES+18] S. D. Vito, E. Esposito, M. Salvato, O. Popoola, F. Formisano, R. Jones, and G. D. Francia. Calibrating chemical multisensory devices for real world applications: An in-depth comparison of quantitative machine learning approaches. *Sensors and Actuators B: Chemical*, 255:1191 – 1210, 2018.

[VFPGF03] S. Vardoulakis, B. E. Fisher, K. Pericleous, and N. Gonzalez-Flesca. Modelling air quality in street canyons: a review. *Atmospheric Environment*, 37(2):155–182, 2003.

[VPMF09] S. D. Vito, M. Piga, L. Martinotto, and G. D. Francia. Co, no2 and nox urban pollution monitoring with on-field calibrated electronic nose by automatic bayesian regularization. *Sensors and Actuators B: Chemical*, 143(1):182 – 191, 2009.

[VVMH12] S. Vembu, A. Vergara, M. K. Muezzinoglu, and R. Huerta. On time series features and kernels for machine olfaction. *Sensors and Actuators B: Chemical*, 174:535–546, 2012.

[WCC+10] F. Wang, D. Chen, S. Cheng, J. Li, M. Li, and Z. Ren. Identification of regional atmospheric pm10 transport pathways using hysplit, mm5-cmaq and synoptic pressure pattern analysis. *Environmental Modelling & Software*, 25(8):927–934, 2010.

[WLe16] J. Wu, C. Leng, and et. al. Quantized convolutional neural networks for mobile devices. In *CVPR*, pages 4820–4828, 2016.

[WNS+20] X. Wu, R. C. Nethery, B. M. Sabath, D. Braun, and F. Dominici. Exposure to air pollution and covid-19 mortality in the united states. *medRxiv*, 2020.

[WXL+20] D. Wu, T. Xiao, X. Liao, J. Luo, C. Wu, S. Zhang, Y. Li, and Y. Guo. When sharing economy meets iot: Towards fine-grained urban air quality monitoring through mobile crowdsensing on bike-share system. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 4(2):61:1–61:26, 2020.

[WYP04] 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*, 14(5):404, 2004.

[WZY16] Y. Wei, Y. Zheng, and Q. Yang. Transfer knowledge between cities. In *Proceedings of SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1905–1914. ACM, 2016.

[XBP+12] Y. Xiang, L. Bai, R. Piedrahita, R. P. Dick, Q. Lv, M. Hannigan, and L. Shang. Collaborative calibration and sensor placement for mobile sensor networks. In *Proceedings of International Conference on Information Processing in Sensor Networks*, pages 73–84. ACM, 2012.

[XCL+16] 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*, pages 376–377. ACM, 2016.

[XXSL16] Y.-F. Xing, Y.-H. Xu, M.-H. Shi, and Y.-X. Lian. The impact of pm2. 5 on the human respiratory system. *Journal of thoracic disease*, 8(1):E69, 2016.

[YKZ18] K. Yan, L. Kou, and D. Zhang. Learning domain-invariant subspace using domain features and independence maximization. *IEEE Transactions on Cybernetics*, 48(1):288–299, 2018.

[YLG+20] H. Yu, Q. Li, Y.-a. Geng, Y. Zhang, and Z. Wei. Airnet: A calibration model for low-cost air monitoring sensors using dual sequence encoder networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 1129–1136, 2020.

[YLM+15] W. Y. Yi, K. M. Lo, T. Mak, K. S. Leung, Y. Leung, and M. L. Meng. A survey of wireless sensor network based air pollution monitoring systems. *Sensors*, 15(12):31392–31427, 2015.

[YLXH21] J. Yang, J. Liu, N. Xu, and J. Huang. Tvt: Transferable vision transformer for unsupervised domain adaptation. *arXiv preprint arXiv:2108.05988*, 2021.

[Yu14] S. Yu. Water spray geoengineering to clean air pollution for mitigating haze in china's cities. *Environmental Chemistry Letters*, 12(1):109–116, 2014.

[Yun18] YunTong. Yuntong. `http://http://www.yuntongkeji.com/uploadfile/2017128174038670.pdf`, 2018.

[YZ16] K. Yan and D. Zhang. Calibration transfer and drift compensation of e-noses via coupled task learning. *Sensors and Actuators B: Chemical*, 225:288 – 297, 2016.

[YZB+18] 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)*, pages 1–6. IEEE, 2018.

[YZW+18] X. Yi, J. Zhang, Z. Wang, T. Li, and Y. Zheng. Deep distributed fusion network for air quality prediction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 965–973, 2018.

[Zan90] P. Zannetti. Gaussian models. In *Air Pollution Modeling*, pages 141–183. Springer, Berlin, Germany, 1990.

[ZCWY14] Y. Zheng, L. Capra, O. Wolfson, and H. Yang. Urban computing: concepts, methodologies, and applications. *ACM Transactions on Intelligent Systems and Technology*, 5(3):38, 2014.

[ZLG+19] Y. Zhang, Q. Lv, D. Gao, S. Shen, R. Dick, M. Hannigan, and Q. Liu. Multi-group encoder-decoder networks to fuse heterogeneous data for next-day air quality prediction. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 4341–4347. AAAI Press, 2019.

[ZLH13a] 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*, pages 1436–1444. ACM, 2013.

[ZLH13b] Y. Zheng, F. Liu, and H.-P. Hsieh. U-air: When urban air quality inference meets big data. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1436–1444, 2013.

[ZPK+18] N. Zimmerman, A. A. Presto, S. P. N. Kumar, J. Gu, A. Hauryliuk, E. S. Robinson, A. L. Robinson, and R. Subramanian. A machine learning calibration model using random forests to improve sensor performance for lower-cost air quality monitoring. *Atmospheric Measurement Techniques*, 11(1):291–313, 2018.

[ZSL15] 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)*, pages 612–617. IEEE, 2015.

[ZTC19] Y. Zeng, Y. Tong, and L. Chen. Last-mile delivery made practical: An efficient route planning framework with theoretical guarantees. *Proceedings of the VLDB Endowment*, 13(3):320–333, 2019.

[ZTK+11] L. Zhang, F. Tian, C. Kadri, B. Xiao, H. Li, L. Pan, and H. Zhou. Online sensor calibration transfer among electronic nose instruments for monitoring volatile organic chemicals in indoor air quality. *Sensors and Actuators B: Chemical*, 160(1):899 – 909, 2011.

[ZTXF19] 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*, 6(4):6618–6628, 2019.

[ZY17] B. Zhang and Y. Yang. Spatiotemporal modeling and prediction of soil heavy metals based on spatiotemporal cokriging. *Scientific Reports*, 7(1):16750, 2017.

[ZYL+15a] 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*, pages 2267–2276. ACM, 2015.

[ZYL+15b] 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 the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 2267–2276, 2015.

[ZZP+21] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of AAAI*, 2021.

[ZZZ+17] J. Y. Zhu, C. Zhang, H. Zhang, S. Zhi, V. O. Li, J. Han, and Y. Zheng. pg-causality: Identifying spatiotemporal causal pathways for air pollutants with urban big data. *IEEE Transactions on Big Data*, 4(4):571–585, 2017.

List of Publications

The following list includes publications that form the basis of this thesis. The corresponding chapters are indicated in parentheses.

Y. Cheng, O. Saukh, L. Thiele. **SensorFormer: Efficient Many-to-Many Sensor Calibration with Learnable Input Sub-Sampling.** *In IEEE Internet of Things Journal (IoTJ), 2022, Vol 5., No. 6.* IEEE, 2022. (Chapter 2)

M. P. Gherman, Y. Cheng, A. Gomez, O. Saukh. **Compensating Altered Sensitivity of Duty-Cycled MOX Gas Sensors with Machine Learning.** *In 2021 18th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON) (pp. 1-9).* ***Best paper.*** (Chapter 2)

Y. Cheng, X. He, Z. Zhou, L. Thiele. **ICT: In-field calibration transfer for air quality sensor deployments.** *In Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT), 3(1), 1-19.* (Chapter 3)

Y. Cheng, X. He, Z. Zhou, L. Thiele. **MapTransfer: Urban air quality map generation for downscaled sensor deployments.** *In 2020 IEEE/ACM Fifth International Conference on Internet-of-Things Design and Implementation (IoTDI) (pp. 14-26).* (Chapter 4)

Y. Cheng, O. Saukh, L. Thiele. **TIP-Air: Tracking Pollution Transfer for Accurate Air Quality Prediction.** *In Adjunct Proceedings of the 2021 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2021 ACM International Symposium on Wearable Computers (pp. 589-599).****Best paper Runner-up.*** (Chapter 5)

Y. Cheng, Z. Zhou, L. Thiele. **iSpray: Reducing Urban Air Pollution with Intelligent Water Spraying.** *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, 6(1), pp.1-29.* (Chapter 6)

The following list includes publications that were written during the PhD studies, yet are not part of this thesis.

E. Johanna, Y. Cheng, F. Papst, and O. Saukh. **Transferable Models to Understand the Impact of Lockdown Measures on Local Air Quality.** *In 2021 17th International Conference on Distributed Computing in Sensor Systems (DCOSS).* ***Best paper.***

L. Chang, H. Zhang, Z. Cheng, J. Shen, J. Zhao, Y. Wang, S. Wang, and Y. Cheng. **Emulation of an atmospheric gas-phase chemistry solver through deep learning: Case study of Chinese Mainland.** *Atmospheric Pollution Research.* 2021.

X. Li, G. Cong, and Y. Cheng. **Spatial transition learning on road networks with deep probabilistic models.** *n 2020 IEEE 36th International Conference on Data Engineering (ICDE).* 2020.

Z. Qu, Z. Zhou, Y. Cheng, and L. Thiele. **Adaptive loss-aware quantization for multi-bit networks.** *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).* 2020.

X. Li, G. Cong, and Y. Cheng. **Learning travel time distributions with deep generative model.** *n The World Wide Web Conference (WWW).* 2019.