UNIVERSITY OF TURKU

Automated greenhouse gas plume detection from satellite data using an unsupervised clustering algorithm

Elias Ervelä

MSc thesis
December 2023

Reviewers:
PhD Laia Amorós
MSc Antti Mikkonen
Prof. Ion Petre

DEPARTMENT OF MATHEMATICS AND STATISTICS

UNIVERSITY OF TURKU
Department of Mathematics and Statistics

ERVELÄ, ELIAS:: Automated greenhouse gas plume detection from satellite data using an unsupervised clustering algorithm
MSc Thesis, 55 pages, 1 appendix pages
Mathematics
December 2023

A crucial part of tackling the problem of climate change is the monitoring of human-caused greenhouse gas emissions. To reach a global scale, greenhouse gas measuring satellites appear to be the best solution. The massive amounts of data produced by the satellites has increased the need for automated, efficient tools to extract knowledge from the data. Emissions from point sources, such as power plants, can produce distinct plumes that are visible from satellite data. Automated plume detection is key to identify and monitor the largest sources of human-caused greenhouse gas emissions.

This thesis presents a comprehensive literature review of existing plume detection methods. Moreover, a new unsupervised plume detection method, called SCEA (**S**patial **C**lustering of **E**levated-valued **A**reas), is introduced. Inspired by the DBSCAN algorithm [1], SCEA is a clustering algorithm that finds distinct high-valued areas in non-gridded data points.

The performance of the SCEA algorithm is evaluated with the simulated satellite data set of SMARTCARB [2] in its ability to find point sources with co-located plumes in different noise scenarios. The SCEA algorithm reached a precision of 0.974, 0.884, and 0.661 in noise-free, low-noise, and high-noise scenarios, respectively. For point sources with annual emissions of 1000 tonnes, the SCEA reached a recall of 0.758, 0.660, and 0.548 for noise-free, low-noise, and high-noise scenarios, respectively.

Keywords: Plume detection, satellites, unsupervised learning, greenhouse gas monitoring, climate change.

Contents

1 Introduction

Global climate change is one of the greatest challenges facing humanity today. The global surface temperature has already exceeded 1.1°C in 2023 from pre-industrial levels and the main cause is the increase in atmospheric greenhouse gas concentrations of carbon dioxide (CO_2) and methane (CH_4) caused by human activity, [3]. As of December 2023, 195 member states of the UNFCCC (United Nations Framework Convention on Climate Change) have signed the Paris Agreement, which aims to limit the increase in global average temperature to 1.5°C or well below 2°C compared to pre-industrial levels,[4]. The current consequences of climate change have measurably affected the atmosphere, oceans, cryosphere, and biosphere, which has already led to an increase in extreme weather and climate conditions, causing adverse impacts and related losses to nature and people. The effects include sea level rise, biodiversity loss, heatwaves, and food and water insecurity, to name a few. Furthermore, the effects are disproportionately affecting more vulnerable communities that have historically contributed the least to climate change. With current trajectories, it is estimated that the 1.5°C mark will be surpassed, which will have more severe consequences for the global climate, [3]. That is why a lot of research is currently being done to monitor the status of climate change and climate efforts to achieve the ambitious goal set in the Paris Agreement.

A component of the Paris agreement is the UNFCCC (United Nations Climate Change Conference) global stocktake, which occurs every five years. Its purpose is to monitor the implementations and assess the progress on achieving the agreed goal of the Paris Agreement. With UNFCCC, we can identify opportunities to further increase climate action. The global stocktake has a three-phase structure. The first phase is to collect all the necessary information on the status of climate change and climate action. The second phase is the technical assessment of the information and the identification of challenges and opportunities. The third phase is the communication and support of political action from the lessons learnt from the previous two phases. The first stocktake was held in 2023, and the COP28 UN Climate Change Conference was just concluded at the time of writing. The second stocktake is to take place in 2028, [5].

One key part of the Paris Agreement and the global stocktake is the monitoring, reporting, and verification of anthropogenic emissions, which means emissions originating from human activity. With proper monitoring, reporting, and verification systems in place, we will be able to help track the progress of climate action and guide nations on the effectiveness of policies. This will require global high-accuracy monitoring systems, which has been a hot topic of research, [6]. Most emission estimates come from compiled bottom-up inventories, which mostly rely on self-reported numbers and assumptions on fuel consumption and efficiencies, and thus are often imperfect and lacking [7], highlighting the need for good top-down emission monitoring systems to complement existing ones. Greenhouse gases are being measured with in-situ and ground-based systems, for example, the Total Carbon Column Observing Network (TCCON), which includes 23 operational high-precision instruments worldwide [8], but to reach a global scale, satellites appear to be the only solution. The rapid development of satellite technology in recent years in terms of accuracy and coverage of satellites has allowed it to play an increasingly important role in the monitoring of greenhouse gas emissions, with the help of ground-based measurements for verification and calibration, [9].

Greenhouse gas measuring satellites measure atmospheric concentrations of gases by analysing the electromagnetic radiation from sunlight after reflecting off the surface of the Earth. The molecules absorb some of this radiation in certain wavelengths, and that footprint can be measured from the light reaching the satellite's instruments. The satellites are equipped with spectrometers that measure these specific wavelength bands from the light. To obtain the amount of certain gases, say CO_2, in the column below the satellite, the amount of the dry air column signal is divided by the amount of CO_2 signal to obtain the column-averaged value of carbon dioxide XCO_2. Factors such as surface pressure and atmospheric optical path length are taken into account within the process. To indicate the measurement technique, the gases measured in this way get an "X" in front of the gas symbol, for example, the column-averaged value of CH_4 is marked with the symbol XCH_4. Usually these are reported in units such as parts per million (ppm), molecules per square centimetre (molec/cm^2) or moles per square metre (mol/m^2). Satellite measurements usually also include a "quality value", which indicates the confidence of the measurement being accurate, since some meteorological conditions make the measurements much more difficult, for example, clouds and aerosols interfering with the light.

The first space-based measurements of greenhouse gases date back to 1996 with the launch of the ADEOS 1 and later the ADEOS 2 satellites, which both unfortunately ended after less than a year due to malfunctions, [10]. In 2002, the European Space Agency (ESA) launched the ENVISAT satellite [11], and since then we have been receiving spaceborne measurements of greenhouse gases, with 14 satellite missions flown since then. Different satellites today are capable of detecting greenhouse gases from daily global coverage down to individual facilities with finest resolutions reaching 50 m scale, [12]. Many even more capable satellite missions are currently in preparation, and if all goes according to plan, at least 22 missions will be flown by 2030, [13]. The growing fleet of satellites means a growing number of data each day. For example, the Sentinel-5p satellite alone gathers at least 244 GB of data each day, [14]. Since the paradigm is shifting from data-limited to model-limited, there has become a growing need for efficient automated methods and tools to process all these data.

A component of the European Union's space programme is the Copernicus Program, which is focused on Earth observation. One of its goals is to take part in the monitoring and verification of greenhouse gases, [15]. A large part of it are the Sentinel missions, which includes a family of Earth-observing satellite missions developed by ESA for land, ocean, and atmospheric monitoring, [16]. Five of the Sentinel missions have already been operational, and many more are to come. In addition to the seven original missions, the Sentinel lineup is getting six more with the Copernicus Sentinel Expansion missions to further broaden the capabilities of the Copernicus Program in the coming decade, [17].

One of the most anticipated missions by the greenhouse gas remote sensing community is the CO2M mission of the Sentinel expansion missions, which will include at least two sun-synchronous Earth observing satellites, CO2M-A and CO2M-B, which are planned to be launched in 2026, [18]. A third satellite is also being planned to further expand the amount of spatial and temporal coverage. The mission aims to monitor anthropogenic greenhouse gas emissions and provide an independent source of information to assess the effectiveness of decarbonisation efforts, [19]. The instruments on board the CO2M satellites will be able to measure the amount of CO_2, NO_2 and CH_4 with high precision and a relatively high resolution of 2 km $\times$ 2 km with a swath of more than 250 km, with which

one should have the requirements to quantify point sources and plumes of anthropogenic greenhouse gas emissions, [20, 21]. The satellites will have nearly global coverage with an orbital time of 99.5 min and repeat cycles of 11 days, [22]. Due to the nature of sun-synchronous satellites, the revisit time for a location is more frequent the farther you are from the equator. For example, with three satellites, the CO2M constellation will be able to measure the same location four to six times during its 11-day cycle around the latitudes of Berlin, however, valid data are not always able to be collected due to changing meteorological conditions, [2].

Currently, a lot of research is being done on tools and methods for analysing the data from CO2M in advance of the launch. A big part of inspiration for this thesis' work is from the upcoming CO2M mission. As an early look at the data produced by the CO2M satellites, the SMARTCARB (Satellite Measurements of Auxiliary Reactive Trace gases for fossil fuel CARBon dioxide emission estimation) project was initiated to develop synthetic satellite measurements that simulate CO2M satellites, [2]. They simulated a year worth of satellite data around central Europe using atmospheric simulations with different emission inventories and many more factors as a source for greenhouse gas fluxes. The SMARTCARB data set is used as a ground for a case study in Section 4, where in Section 4.2 the SMARTCARB data set is introduced in detail.

Despite playing a significant part in climate change, there remain major uncertainties regarding our knowledge of human-caused CO_2 emissions due to the difficult nature of the molecule. CO_2 is a long-lived and well-mixed gas in the atmosphere with a lifetime on the order of centuries, [23]. Moreover, the global annual natural fluxes of CO_2, including, for example, respiration, fire, photosynthesis, and ocean-atmosphere gas exchange, far exceed the anthropogenic fluxes, which are the main cause of the imbalance in the carbon cycle, [24]. Thus, the local increases in CO_2 caused by individual point sources of anthropogenic emissions are usually small compared to the background concentration and its natural variability, and often not much greater than the noise in the satellite instruments. This makes remote monitoring of CO_2 very difficult.

Fortunately, nitrogen dioxide (NO_2) is a much easier molecule to monitor and is generally co-emitted with CO_2 in very noticeable amounts when burning fossil fuels at high temperatures, [25]. NO_2 has an atmospheric lifetime on the order of hours, so the signal of NO_2 is usually very clear compared to the background levels near an emission source. NO_2 retrievals can tolerate larger errors, making them less sensitive to clouds, giving more usable data. Moreover, the spatial resolution and the swath of NO_2 satellite measurements far exceed the capabilities of satellite CO_2 measurements, [26]. For example, NO_2 retrievals of the TROPOMI instrument aboard the Sentinel-5p reaches a swath of approximately 2600 km, while the CO_2 measuring satellite OCO-2 has a swath of 10 km, [14, 27]. These features make NO_2 highly useful as a proxy for CO_2 emissions, which helps in the detection and quantification of CO_2 point source emissions, [28, 29, 30].

One of the greatest challenges for satellite-based greenhouse gas monitoring systems is to differentiate natural sources and background variation from emissions originating from an anthropogenic source. A large portion of anthropogenic emissions come from sources with very small distinct areas, which makes them possible to detect from satellite data. Certain terms regarding this are now defined. A discrete source of emissions in a specific stationary location, for example a power plant, is called a *point source*, and a larger

Figure 1: An example of what a plume coming from a point source, like a power plant, might look like. Image generated with Bing image generator powered by DALL-E 3 using promt "A single plume coming from a single power plant in a city. A simple photograph taken from ground level".

continuous area of emissions, for example a city, is called an *area source*. Sometimes in the literature, multiple nearby point sources or a dense area source is called an emission clump. The cloud of gases that emerges from an emission source is called a *plume*, an example of which can be seen in Figure 1. Studying these different types of emission sources and their plumes in satellite data is crucial in the study of anthropogenic emissions. The precision and resolution of greenhouse gas measuring satellites today have enabled a large amount of research on the quantification of emissions from city to facility scale plumes in the last couple of years, [31, 32, 33, 34, 35, 30, 36, 37]. This pile of research is only going to grow with more capable satellite missions coming in the near future.

To further enable the monitoring of greenhouse gases, the identification of point sources and their plumes from satellite data has become a big part of the research. With good plume detection systems in place, we might be able to find new unreported sources of emissions to further enlarge our emissions inventories and get a better grasp of anthropogenic greenhouse gas emissions. Research on automatic plume detection has grown rapidly very recently, [28, 38, 39, 40, 41, 42]. Plume detection research is bound to grow in the future with many upcoming satellite missions. This thesis dwells deep on this subject with a literature review of the existing methods in Section 2, and a new plume detection method introduced in Section 3. Many of the current plume detection methods lean on machine learning technology, including the one presented in this work, so the machine learning aspects are emphasised in this thesis.

Machine learning and satellite data is a duo that has a huge potential. The enormous amounts of data coming from satellites emphasise the need for efficient automatic

processing techniques, and data hungry machine learning methods might the superior solution, since they can learn highly complex relations given enough data. The use of machine learning for satellite data has grown exponentially [43], and it seems that there is still a lot of untapped potential. The growing amount of available data, the increase in availability of computational resources, and the improvement of accessibility of application programming interfaces and libraries for machine learning, such as Keras [44], Pytorch [45], Tensorflow [46] and Scikit-Learn [47], have enabled the fusion of remote sensing and machine learning to grow at a rapid pace with no signs of stopping. The use of machine learning in the field of greenhouse gases and satellites has led to a surge of research in the very recent history. Machine learning has been used for applications that include, but are not limited to, improving measurement quality and coverage [48, 49, 50, 51, 52], estimation of point source emissions [53, 54], atmospheric modelling [55], plume segmentation [50], and plume detection [38, 39, 40].

One challenge with satellite data is obtaining well-representing, high-quality, labelled data for supervised training. It is often very difficult for even a human expert to say what is going on under the hood in satellite retrievals. For example, in plume detection, it is very difficult to confidently say that there is a plume in a scene or that there is no plume. For certain applications, ground-based sensors are used as sources of ground truth, as they are often more precise, but they cannot be utilised in all applications. Therefore, some existing machine learning methods rely on simulated data for training, [28, 40], since all details are known. Fortunately, there has been a good deal of research on atmospheric greenhouse gas simulations over the years, [2, 56, 57, 58, 59].

The use of non-supervised learning techniques, such as unsupervised, semi-supervised, self-supervised, weakly supervised learning or transfer learning has not yet been very popular among the greenhouse gas satellite community, but has been showing growing interest in other applications of remote sensing, [43]. Unsupervised learning means finding patterns in data without any training data, semi-supervised learning means combining labelled and unlabelled data in the training process, self-supervised learning means creating the labels automatically from the data itself, weakly supervised learning means having noisy or incomplete labels for the training, and transfer learning means using a pre-trained model from a closely related task. However, the use of only image-level labels for plume segmentation as a middle step in [39] can be interpreted as a form of weakly supervised learning, the use of simulated data in [28, 40] can be interpreted as a form of transfer learning, and the use of statistical methods in [28] can be interpreted as a form of unsupervised learning. The upside of using these different forms of non-supervised learning is that they are effective in situations where there are very little or no labelled data.

A large part of this thesis is the new plume detection method introduced in Section 3 called SCEA. Unlike the other existing plume detection methods, SCEA is a method that relies on unsupervised techniques for plume detection without any training data. The method can be applied to various data sets where the task is to find areas of elevated values in points with a spatial location and a value associated to it, but it was designed with plume detection in mind. The performance of the method is studied in Section 4 with its ability to find plumes and the point sources under them. SCEA reaches a precision of 0.974 in a noise-free scenario and a precision of 0.883 and 0.661 in low- and high-noise scenarios, respectively.

The structure of the thesis is as follows. First, in Section 2, an extensive literature

review on plume detection methods is provided where the notable papers are uncovered. In Section 3, the new plume detection method SCEA is introduced in detail with examples of it in action. In Section 4, the performance of the SCEA plume detection method is tested in a case study with simulated satellite data from SMARTCARB. Lastly, in Section 5, conclusions, closing remarks, and potential future work are discussed. Figures in this thesis are plotted in colour-vision deficiency friendly and perceptually-uniform colour maps from [60] when possible. This thesis was done in collaboration with the Finnish Meteorological Institute, where an internship project on automatic plume detection in the summer of 2022 later evolved into a master's thesis. Since the internship, Laia Amorós and Antti Mikkonen have been supervising the project on behalf of the Finnish Meteorological Institute.

2 Plume detection with satellite data

With the growing number of good quality satellite data on greenhouse gases, there has been a very recent surge of research on plume detection from satellite data. Majority of the literature has been published during 2022 and 2023, so the field has progressed significantly while writing this thesis. The problem of plume detection involves automatically finding sources of emissions by the plumes they create in the satellite data. The related problem of plume segmentation or plume masking has also grown on the side, [61, 62, 63, 37], which is sometimes also referred to as plume detection, but these methods are not considered in this literature review.

In this section, many of the existing plume detection methods are presented. Here, the methods are divided into three categories: statistical methods [28], deep learning methods [38, 39, 40], and image processing methods [41, 42], which are presented in sections 2.1, 2.2, and 2.3, respectively. The papers are reviewed one by one, separated by unnumbered sub-subsections, with the paper title as the sub-subsection title. All of the deep learning methods utilise forms of convolutional neural networks which have been the most popular choice of architecture in computer vision for a while, since they can effectively learn the spatial dependencies within the data.

The methods tackle a range of different pollutants: CO_2 [28], NO_2 [38, 28], CH_4 [39, 40, 42] and CO [41], which all have their own different set of challenges. The instruments that measure these gases have varying levels of resolutions, coverages, precisions, and availability. Moreover, the gases themselves come from different kinds of sources and have differing levels of background concentrations and lifetime in the atmosphere.

In this section, the following scoring metrics are mentioned.

$$\begin{aligned} \text{precision} &= \frac{\text{TP}}{\text{TP} + \text{FN}}, \quad \text{recall} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \\ \text{F1} &= \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} = \frac{\text{TP}}{\text{TP} + 0.5 \cdot (\text{FP} + \text{FN})}, \end{aligned} \tag{1}$$

where TP, FP and FN, are the number of True Positives, False Positives and False Negatives, respectively. Recall and precision are usually reported in pairs since they often have an inverse relationship when tuning model sensitivity, and in isolation, recall and precision do not tell much. The F1 score is a combined metric which can be interpreted as a harmonic mean of precision and recall. These are very commonly used as performance metrics in classification problems. In the context of plume detection methods, a true positive can refer to either a pixel-level prediction, such as whether a pixel was correctly classified as a plume pixel, as in [28], or a plume-level prediction, such as whether a plume was correctly detected in an image, as in [38, 39, 40] and in the case study in Section 4. However, the way to define a true positive pixel or a true positive plume varies in the literature.

2.1 Statistical methods

Detectability of CO_2 emission plumes of cities and power plants with the Copernicus Anthropogenic CO_2 Monitoring (CO2M) mission (G. Kuhlmann et al., 2019)

Kuhlmann et al. [28] created a simple plume detection method to test the detectability of plumes in the SMARTCARB simulations [2], which attempts to imitate data from the upcoming CO2M mission, [18]. A more thorough description of the data set is provided in Section 4.2. A method that uses statistical tests was developed to find significantly elevated levels of CO_2 and NO_2 compared to instrument noise and variability in background levels.

They used the Z-test for their method. It is a statistical test where the test statistic is approximately normally distributed and its variance is known. The Z score is

$$Z = \frac{\bar{x} - \mu}{\sigma},$$

where μ is the mean value under the null hypothesis and $\bar{x}$ is the sample mean of our test statistic. Their difference is divided by the standard deviation σ to scale the variance of Z to one. This is compared to the quantile function of normal distributions $z(p)$. The interpretation of $z(p)$ is that there is a probability p to get a value of $z(p)$ or smaller, when randomly sampling the standard normal distribution.

The average value of the neighbourhood X_{obs} is taken for every pixel and the estimated background value X_{bg} is subtracted to obtain the amount of signal. This is then divided by the estimated standard deviation to obtain the signal-to-noise ratio (SNR)

$$\text{SNR} = \frac{\overbrace{X_{\text{obs}}}^{\text{mean value of the neighbourhood}} - \overbrace{X_{\text{bg}}}^{\text{background value}}}{\sqrt{\underbrace{\frac{s_{\text{ssp}}^2}{n_p}}_{\text{variance of the instrument}} + \underbrace{s_{\text{bg}}^2}_{\text{variance of the background}}}},$$

which is interpreted as the Z score.

The noise component consists of the variance of the instrument and the variance of the background. The variance of the instrument here is the single sounding precision s_{ssp}^2, which is assumed to be normally distributed. It is divided by the number of pixels in the neighbourhood to get the variance of the neighbourhood's mean value. The variance of the background s_{bg}^2 is considered to be systematic across all pixels, thus it is not divided by the number of pixels in the neighbourhood. Since the two variance terms are independent, they can be summed. To get the standard deviation of the noise, a square root is taken.

The null hypothesis is that the mean value of the neighbourhood of a pixel X_{obs} is the same as the mean background value X_{bg} and the alternative hypothesis is that the neighbourhood X_{obs} is larger, so a one-tailed Z-test is applied. If p-values above a set threshold are obtained, the pixels are marked as significant. That is, a pixel is significant if

$$\text{SNR} > z(p).$$

After the significant pixels are recognised, a group of adjacent pixels are treated as one single plume. In the paper, only the plumes of a few known large sources are attempted

to detect, thus all the plumes not within a pre-defined radius of a source in a pre-defined list are removed. For cities, this radius is larger. For the plume detection performance evaluation they set the p to 99% and the neighbourhood size to 37 pixels in a circle-like area around the pixel. The mean and variance of background are estimated from the model tracers excluding the emissions of the source for a 200×200 km^2 around cities and a 100×100 km^2 around each power plant.

The method was tested for plumes of specifically chosen sources in different noise scenarios. The pixels had a label to indicate whether they were part of the true plume or not, which was determined on the simulations without noise or background concentration. The performance of the method was tested by its ability to identify these true plume pixels. A plume was considered successfully detected if there were at least 100 pixels detected and pixels found by the method had a precision (Eq. 1), of above 80 %. With only CO_2 measurements, the results were pretty bad. The method was able to find a plume from 12 % to 32 % of times in different noise scenarios. With NO_2 it performed expectedly much better in finding plumes with rates from 68 % to 70 %.

One must be careful when interpreting the precision (Eq. 1) alone, as it does not account for false positives and true negatives at all. One could detect only a single, highest-confidence pixel and get a precision of 1.0. The problem is highlighted in the fact that they got much smaller precision for NO_2 plumes in all noise scenarios, which have objectively much clearer plume signals, compared to CO_2. This was handled by only considering plumes with at least 100 pixels detected, which balances the downfalls of precision.

They only evaluated the method for large specific known plumes and did not test how well it finds plumes without prior knowledge. The method just threw away the significant pixels if they were not near any known sources. Their estimation of the background was not realistic for real satellite measurements, because the background value was obtained from the simulations without the plumes, which is not possible in a real-world setting, where the background value must be estimated. One may wonder if the use of Student's t-test rather than Z-test would be better in their method, since Student's t-test takes into account the uncertainty of variance, which is the case here. The neighbourhood averaging lessens the effect noise has on the measurements, but the signal of plumes that are smaller than the neighbourhood can get lost and most of the point sources are small enough to create only sub-pixel sized plumes. Moreover, the outline of the plume might be harder to detect. Using a Gaussian filter over the image could sound more natural, as it reduces the impact of the neighbourhood the further one moves from the centre, and thus could help to counteract some of the drawbacks of neighbourhood averaging.

To summarise, their method is not built for general, real-world plume detection, but in their defence, they did not try to build one. The point of their work was to answer the question of is plume detection possible at all in the upcoming CO2M mission and how is noise affected. However, this question is highly dependent on the method used, especially the effect of the amount of noise on potential plume detection. The method presented could still be adequate for continuous monitoring of known large point sources, which, however, dismisses smaller and non-reported sources.

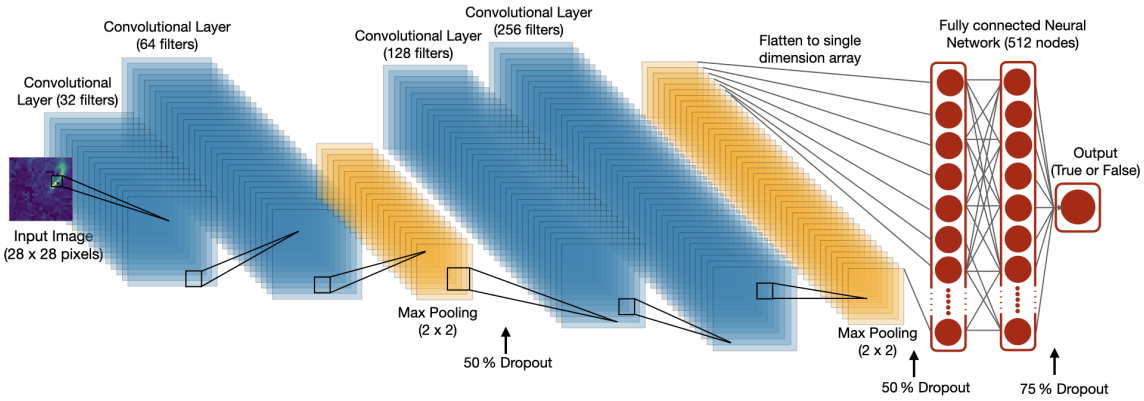

Figure 2: Convolutional neural network architecture used by Finch et al. in [38] for their plume detection. Figure source: [38].

2.2 Deep Learning methods

Automated detection of atmospheric NO_2 plumes from satellite data: a tool to help infer anthropogenic combustion emissions (D. Finch et al., 2022)

One of the first uses of neural networks in plume detection was surprisingly recently by Finch et al. [38] in 2022. They used a convolutional neural network (CNN) to identify plumes from NO_2 measurements. The TROPOMI data of the Sentinel-5P satellite was used, which has a resolution of 7 km by 3.5 km and a swath of approximately 2600 km [14], [64]. It was fed with 28 by 28 pixel images, approximately corresponding to 266 km $\times$ 133 km, where NO_2 values had been standardised to zero mean and variance of one.

The data was gathered by hand labelling images provided by the satellites. A crowd-labelling approach was first attempted, but that ultimately ended up falling short of expectations due to the large number of inconsistent results the human labellers produced. This emphasises the fact that it is hard to objectively define what is a plume and what is not. The authors ended up labelling most of the images themselves. They ended up with 3043 images that were labelled as having a plume and 3043 that did not have one. Then the CNN was trained using these data, the architecture of which is shown in Figure 2. The output of the network is a value between 0 and 1 depicting the confidence of the model that there is a plume somewhere in the image. A threshold of 0.75 was set to classify the image as containing a plume.

The method was tested on all Sentinel-5P NO_2 measurements from 2018 to 2020. The method assigned 310 020 images that contain at least one plume. It was determined that approximately 20 % of the plumes were close to an active fire, as identified by the VIIRS thermal anomaly data [65], so it was assumed that these plumes were associated with biomass burning. The remaining 247 980 images were assumed to be plumes originating from anthropogenic combustion. The anthropogenic sources were further classified to nearby major urban areas, power stations, and gas and oil flaring regions using supplement maps and data sets on population, power plants, and flaring. The plumes found were compared to the ODIAC [66] data set, which is a global open-source Data Inventory for Anthropogenic CO_2 emissions. It was reported that the plumes sightings correspond to 56 % of the emission sources in ODIAC, accumulating to 92 % of the emissions present in the database. The 44 % of the sources that were missed consisted mainly of smaller

emissions, which is expected. The pixel size of the measurements is approximately 24 km^2, so it is understandable that smaller sources might not give out that large of a signal. From the detections, about 10 % are not connected to any known emissions in ODIAC. This might be explained by incomplete data in ODIAC, or biomass burning not detected by VIIRS, or just false positives.

In the paper, there was no mention about the use of data augmentation in the data gathering process. Data augmentation is the process of creating more training data by slightly altering the images by, for example, adding noise, shifting, rotating, or some combination of these. It can help reduce overfitting and increase data size. It seems like data augmentation could have been a highly useful tool to utilise in this situation, especially since they had data labelling problems and neural networks tend to be very data hungry.

It might be reasonable to interpret the reported numbers a bit critically. It was reported in their abstract that the CNN can find plumes with an accuracy of over 90 %, but that number is the accuracy on their test set, which I understand to be mostly clear and unambiguous high-signal plumes, which is not the case in the real world for most plumes. The reported 92 % of the emissions that were found does not sound that remarkable considering that it is only about the largest 25 % of sources and there were two years worth of data where one could be found. It was not reported how often a source was found out of all overpasses at the location. Furthermore, it was not reported how a plume was connected to a source from ODIAC. Some sort of radius must have been used, but there is no mention of that, which is strange because the results would be heavily affected by that. Their code and data are not publicly available, so doing independent experiments to verify their claims is not possible.

Automated detection and monitoring of methane super-emitters using satellite data (B. Schuit et al., 2023)

In [39] a two-step machine learning approach was used for the detection of methane plumes from satellite data by Schuit et al. First, a convolutional neural network was used to identify plume-like structures from a 32 x 32 pixel images. For the second step, the plume candidates were further classified as real, empty, or artefact with a support vector machine using many additional variables like wind data, ground features, atmosphere features and other retrieval variables from pixels of the plume candidate.

The convolutional neural network only outputs a number between 0 and 1 depicting the confidence of a plume structure existing anywhere in the picture without quantifying where exactly it is. To find the specific pixels of the plume, a Grad-CAM class activation map [67] was used to identify which pixels gave the strongest signal in the prediction of the model. With the knowledge of the specific pixels in the potential plume, focus can be directed towards that area for the next step. Additional features are extracted from these pixels and fed to a support vector machine where the potential plume is further classified as a real or not.

The class activation map is a method that creates a map of the image, where each pixel is given a value of how important it was to the output of the network. Convolutional neural networks were shown to possess an emerging ability to localise objects in images without ever explicitly training them with object location data, only image-level labels, [68]. In

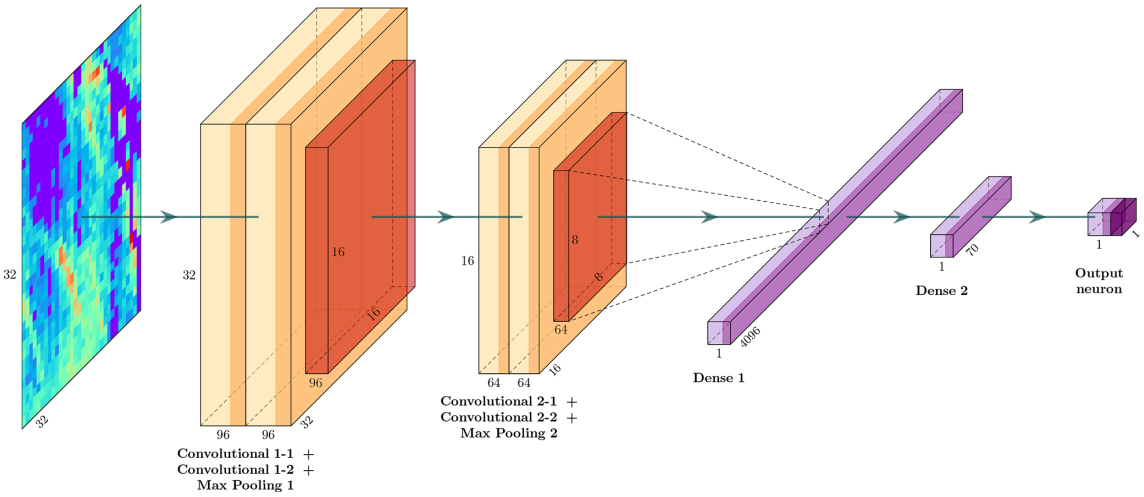

Figure 3: Neural network architecture used by, [39]. Figure source: [39]

the original paper [69], a class activation map was successfully built for certain types of convolutional neural networks, but in [67] the authors introduced Grad-CAM, which was able to create class activation maps for a wide variety of CNN models using target label gradients. It is especially useful for model interpretability and object localisation in images. This method was shown to be a fine fit to segment the plume pixels in the image in the plume detection method by, [39]. Together with the methane measurement values and the class activation map, a combined map was created by multiplying the value of XCH_4 and the class activation map value for each pixel. To obtain plume masks for different levels of confidence, different thresholds were used.

The data were gathered by manual inspection from more than 60 persistently emitting locations identified using long-term wind-rotated averages, [70]. Gathering data using only certain types of point sources poses a danger of having a too narrow representation of all different possible plumes, so the more different locations are used, the better. A set of negative examples were gathered from different times of orbital cycles with different surface and meteorological conditions. To enlarge the data set, the samples were augmented by 90° rotations and flips.

Of all the data, 20% of it was selected as a test set, for which the recall and F1 score were reported to reach 0.956 and 0.958, respectively. These numbers probably do not translate to the real-world, as they used the same 60 locations for training and testing, which might leak bias in the test set scores.

Their method pipeline was tested for a year worth of TROPOMI methane data from 2021. The method was able to identify 4869 unique plumes, and of those, according to the manual inspection of the authors, 2974 were determined as confident plumes and 745 as potential plumes. If all potential and confident plumes are accept as real plumes, the pipeline achieved up to 76 % precision. Recall was not reported, since there was no way to measure how many true plumes were missed.

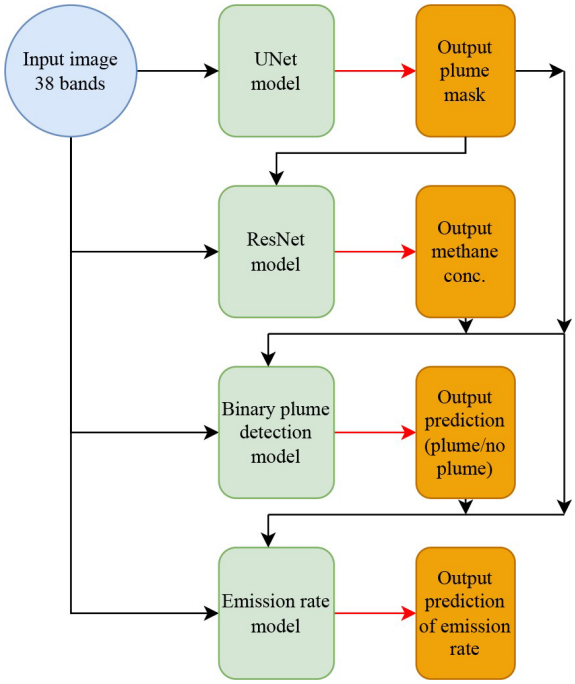

Figure 4: Model architecture used in, [40]. It features four different neural networks in sequence for different purposes. Green boxes indicate neural networks, and orange boxes indicate the outputs of the neural networks. Figure source: [40].

Using a deep neural network to detect methane point sources and quantify emissions from PRISMA hyperspectral satellite images (P. Joyce et al., 2023)

In [40], a four-step method with four different neural networks was introduced by P. Joyce et al. The high-level architecture of the method can be seen in Figure 4. It was trained on images with simulated plumes attached to real-world satellite data. This architecture was designed to detect and quantify methane point source emissions. The method features an U-Net model used to segment a plume out of an image, a ResNet model performing regression to output the methane concentrations in each plume pixel, and two other convolutional neural networks, which featured only an encoder branch.

The U-Net architecture [71] is a type of encoder-decoder-style convolutional neural network designed for image segmenting. The encoder side has blocks of repeated convolutional layers and a max pooling layer, which all make the size of the input smaller, but in turn, the depth of the feature channels are increased, allowing for more complex representations. In turn, the decoder side has blocks that up-samples the input with 2x2 convolution and narrows the feature depth with more convolutional layers. At the end of the decoder, there is a 1x1 convolution, which outputs the segmented image.

A ResNet architecture [72] was used for methane retrievals, as it is lightweight and powerful in computer vision regression problems. It uses residual blocks, which feature a skip connection that adds the input to the output of some layers, which helps avoid the problem of vanishing gradients, which can occur with very deep networks. The idea of a residual network is very versatile and can be implemented in many neural network architectures.

Simulations of methane plumes were generated by using the Weather Research and

Forecasting (WRF) model system with a modified version of a large eddy simulation (LES) case to simulate methane plumes. Different scenarios with different winds speeds from 1 m/s to 9 m/s were applied. From the different scenarios, 1800 scenes were extracted in total. These plumes were added to backgrounds from real-world satellite measurements of the PRISMA satellite [73] with 36 different background images covering a variety of spatial and temporal scales throughout the approximately 3 years of PRISMA data available. A total number of 9700 256×256 images were generated with different combinations of backgrounds and plume simulations, with random augmentations of 90° rotations, horizontal and vertical flipping added during the training. The images were then standardised to a mean value of zero and a variance of one, as it is understood to help the neural network with training. To create ground truths for every image, a binary label was determined for every pixel with a methane concentration threshold that was lowered as training progressed from 8×10^{18} molecules per cm^2 down to 4×10^{18} molecules per cm^2.

Model training was divided into four steps for the four different models. First, a U-Net architecture was trained with the task of segmenting a plume from the image. Second, a ResNet model was trained to predict the amount of methane from the plume mask produced by the first neural network. Third, a binary prediction model trained to predict whether a plume is present in the image or not. If the model thinks there is a plume in the image, the fourth model is used, which was trained to predict the emission rate from the point source. Each model was trained in this order without updating the weights of the other models.

For the test set of size 2000 simulated images, the model achieved a F1, precision, and recall (Eq. 1) of 0.95, 0.96 and 0.92, respectively. It must be noted that the accuracies on the test set might not reflect on the real-world capabilities, as it is only simulated data used in training and testing. The model was then tested with 40 PRISMA scenes obtained from an oil field, which is known for frequent point source emission of methane. A total of 21 plumes were identified from the scenes. The accuracy of the model was not determined with real-world satellite data, probably due to the challenges of defining the true labels in real-world data.

2.3 Image processing methods

Plume detection and emission estimate for biomass burning plumes from TROPOMI carbon monoxide observations using APE v1.1 (M. Goudar et al., 2023)

In [41], Goudar et al. introduced an automated plume detection and emission estimation algorithm (APE), where multiple image processing algorithms are combined to do plume detection around potential plume locations. A region-based segmentation algorithm with multiple specific steps are used, where similar pixels are clustered together to form a homogeneous region. This approach was applied to CO data from the TROPOMI instrument. The found plume pixels are then used to estimate emissions from that source.

First, active fires are identified with the Visible Infrared Imaging Radiometer Suite (VIIRS) instrument aboard Suomi NPP satellite, which is flying three minutes ahead of the TROPOMI wielding Sentinel-5P. For each single-point fire source identified, the CO data is selected around that point to obtain a 41×41 pixel image, an example of which can be seen in Figure 5a. Then a plume detection algorithm is applied to outline the plume. As the first step of the plume detection algorithm, a smoothed image $\mathbf{I}_s$ is obtained using

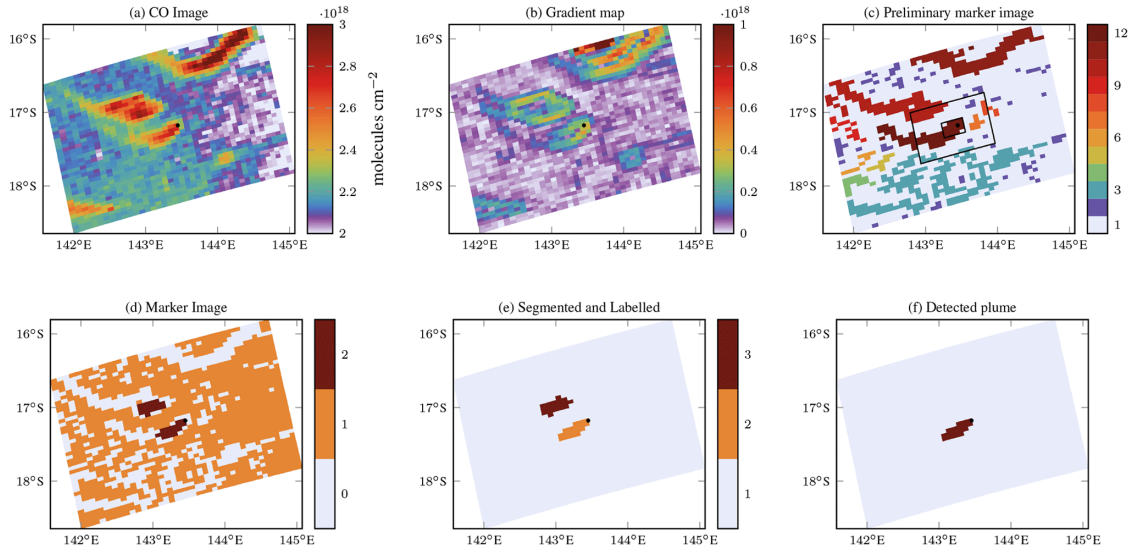

Figure 5: Illustration of the plume detection steps in the APE v1.1 plume detection algorithm. Figure source: [41].

a Gaussian filter over the image. To emphasise the edges of the plumes, the approximate gradient map $\mathbf{I}_{\text{grad}}$ is calculated by applying convolution to $\mathbf{I}_s$ with a kernel called Sobel operator [74], which is defined as

$$\mathbf{I}_{\text{grad}} = \sqrt{\mathbf{G}_x + \mathbf{G}_y}, \quad \text{where} \quad \mathbf{G}_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} * \mathbf{I}_s, \quad \mathbf{G}_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * \mathbf{I}_s,$$

where $*$ is the convolution operation. The resulting gradient map can be seen in Figure 5b.

Then a marker image $\mathbf{I}_{\text{mark}}$ is introduced, which is constructed with the following steps.

1. Initialise pixels with zeros in $\mathbf{I}_{\text{mark}}$.

2. The pixels without a CO enhancement, that is all pixels under median of $\mathbf{I}_s$ or under the mean of the 15×15 pixels centred at $\mathbf{I}_s$, get a label "1" in $\mathbf{I}_{\text{mark}}$.

3. All the connected pixels in $\mathbf{I}_{\text{mark}}$ with the same marker are identified using a "label" algorithm, [75]. Each connected region is identified by a unique integer. The result is the preliminary marker image seen in Figure 5c.

4. Next step is to zoom in on a 5×5 area around the fire source and all connected areas in the preliminary marker image are considered as potential plumes. The potential plumes are then expanded by going to 15×15 pixels around the fire source using pixels with the same label.

5. From these pixels the CO mean is then calculated and used as a threshold value for all pixels within the 15×15 area. Pixels above this threshold get a label "2", which can be seen from 5d.

With the marker image $\mathbf{I}_{\text{mark}}$ and the gradient map $\mathbf{I}_{\text{grad}}$, a marker based watershed algorithm [76] is deployed to determine if the marked pixels are multiple different plumes or just one. Metaphorically, the watershed algorithm finds tops from bottoms by flooding water to the topographical interpretation of the image. The marked version of this algorithm does a distance transformation to the marker image, where each pixel is assigned a value equal to the distance to the nearest marker. Then to this transformed image, the original watershed algorithm is applied. The result is the final plume, as seen in Figure 5f, which is then used for emission estimation.

Out of 1327 good quality CO data granules, 882 plume signatures were identified. Out of these detections only 378 cases were considered good enough for emission estimation as many plumes were flagged as too short or having multiple fire sources present in the detected plumes. It is not possible to calculate accuracy of the method here, since it is not guaranteed that each 1327 of the data granules had a detectable plume. The method was used only for specific types of sources that are identifiable by the VIIRS instrument, but the method could work on other plumes as well, if the potential point source is provided in other ways.

Global assessment of oil and gas methane ultra-emitters (T. Lauvaux et al., 2022)

In [42], Lauvaux et al. made efforts to globally detect and quantify methane ultra-emitters, the massive sporadic releases of methane, which most likely occur during the maintenance of oil and gas production infrastructure, or during equipment failures. These significantly large emission releases are not accounted for in current greenhouse gas inventory estimates. Methane retrievals from the TROPOMI instruments on board the Sentinel-5p were used between 2019 and 2020.

The plume detection algorithm was not designed for detection of smaller point sources or to be fully automatic without a human supervision. Rather, it was used as a tool to guide the authors to look for the right places, but for the sake of completeness and comparison, the method is introduced.

For every orbit of Sentinel-5p, 13 to 14 images are produced. For every image, their plume detection algorithm is applied, which goes as follows.

1. The image is denoised using a Gaussian filter, [77].

2. Local standard deviation and background values are computed. For each pixel, the standard deviation σ of the surrounding 11×11 pixel patch is calculated and considered as the standard deviation of the centre pixel. The background value at this pixel is computed as

$$\begin{cases} m, & \text{when } \frac{\bar{x}-m}{\sigma} > 0.3 \\ m \cdot l - \bar{x}(l-1), & \text{otherwise} \end{cases},$$

   where m is the median, $\bar{x}$ is the mean and σ is the standard deviation of the patch. They do not disclose the parameter value l used, but they mention that it is typically around 2.5. The background value is computed this way to be robust to the influence of the plume. If the pixel distribution is strongly skewed, the difference between the mean and median would have a too significant impact on the estimate, thus the condition $\frac{\bar{x}-m}{\sigma} > 0.3$ is added.

3. Elevated values are segmented with an anomaly map

$$\text{AnomalyMap} = \text{Image} - \text{Background} - k * \sigma,$$

   where contiguous groups of positive pixels are selected as plume candidates.

4. A sharpening kernel and a watershed algorithm [76] are applied to separate distinct plumes that appear contiguous on the anomaly map.

5. A plume is discarded if the average value of XCH_4 in pixels of the plume is below a set threshold, which they had set most of the time as 25. A plume is also discarded if there are too few pixels with a quality value greater than 75%, which they had most of the time set as 5 pixels.

After automated plume detections, a human manually verified and labelled the results.

Their plume detection algorithm was applied to the TROPOMI retrievals and the emissions were quantified for each plume using a Lagrangian particle model called HYSPLIT, [78]. Over 1800 significant enough plumes were detected over the 2 years of study period, most of them co-located with oil and gas production. The accuracy of the plume detection method was not studied as it was not the main point of the paper.

3 SCEA - Unsupervised spatial clustering algorithm for finding areas of elevated values

In this section, a new unsupervised spatial clustering algorithm designed for plume detection, called SCEA (**S**patial **C**lustering of **E**levated-valued **A**reas), is presented. In Section 3.1, the premise and the idea of the algorithm are presented. In Section 3.2, the algorithm is explained in detail, and in Section 3.3, examples of SCEA applied to different test data sets are presented.

3.1 The premise

The SCEA algorithm, which is developed for plume detection, finds high-value areas in spatial measurement data by assigning points to clusters. The assumption is that every point has a location in space and a measurement value associated with it, and the task is to find areas with elevated values. For example, in the case of NO_2 satellite measurements, every point has coordinates and a measurement value of column-averaged nitrogen dioxide XNO_2 in that location. Traditional clustering methods used in unsupervised machine learning are not suitable for these kind of problems, since they only take into account the position of points and the distances between them to determine clusters and they cannot differentiate the spatial dimension from the measurement value dimension. When trying to find areas of high value, an area may have a large number of data points but low measurement values associated to them, and thus those areas should not be clustered. The SCEA algorithm works with various types of data, but here it is applied in the case of plume detection with satellite data as an example throughout the text.

The SCEA algorithm is inspired by the DBSCAN (**D**ensity **B**ased **S**patial **C**lustering of **A**pplications with **N**oise) clustering method, [1]. The idea of DBSCAN is that contiguous regions of high point density form clusters that are separated from other clusters by regions of low point density. In practise, this is accomplished by clustering points together if they are close enough to each other. In the DBSCAN algorithm, we start from a random point, and every point that is within a certain predetermined radius of the starting point gets added to the same cluster. The same procedure is repeated for these newly added points and is continued as long as new points are found. When no new points are near enough to be added, the clustering stops, and that cluster is finalised. Then we go to the next random point not already clustered and do the same procedure. If a single point is further than the set radius from any other points, it is labelled as noise. The algorithm stops when every point is assigned a cluster or labelled as noise. Although DBSCAN, which was introduced in 1996, is approaching three decades of age, it and its close variants are still widely used at the time of writing, [79]. The strength of DBSCAN is that it can easily handle clusters of arbitrary shape even in high dimensions and it is not bothered by outliers.

For plume detection, DBSCAN does not work without major modifications. The fact that in the original DBSCAN method the radius is a parameter that stays constant during the run-time raises a problem of generality, since for every application a proper radius must be deliberately found. Moreover, it does not properly take into account the difference in spatial location of a point and its associated measurement value. In contrast to the original DBSCAN algorithm, the radii do not stay constant in the SCEA algorithm. The radius is affected by the value of the point, but this feature alone is not enough, as can be seen in

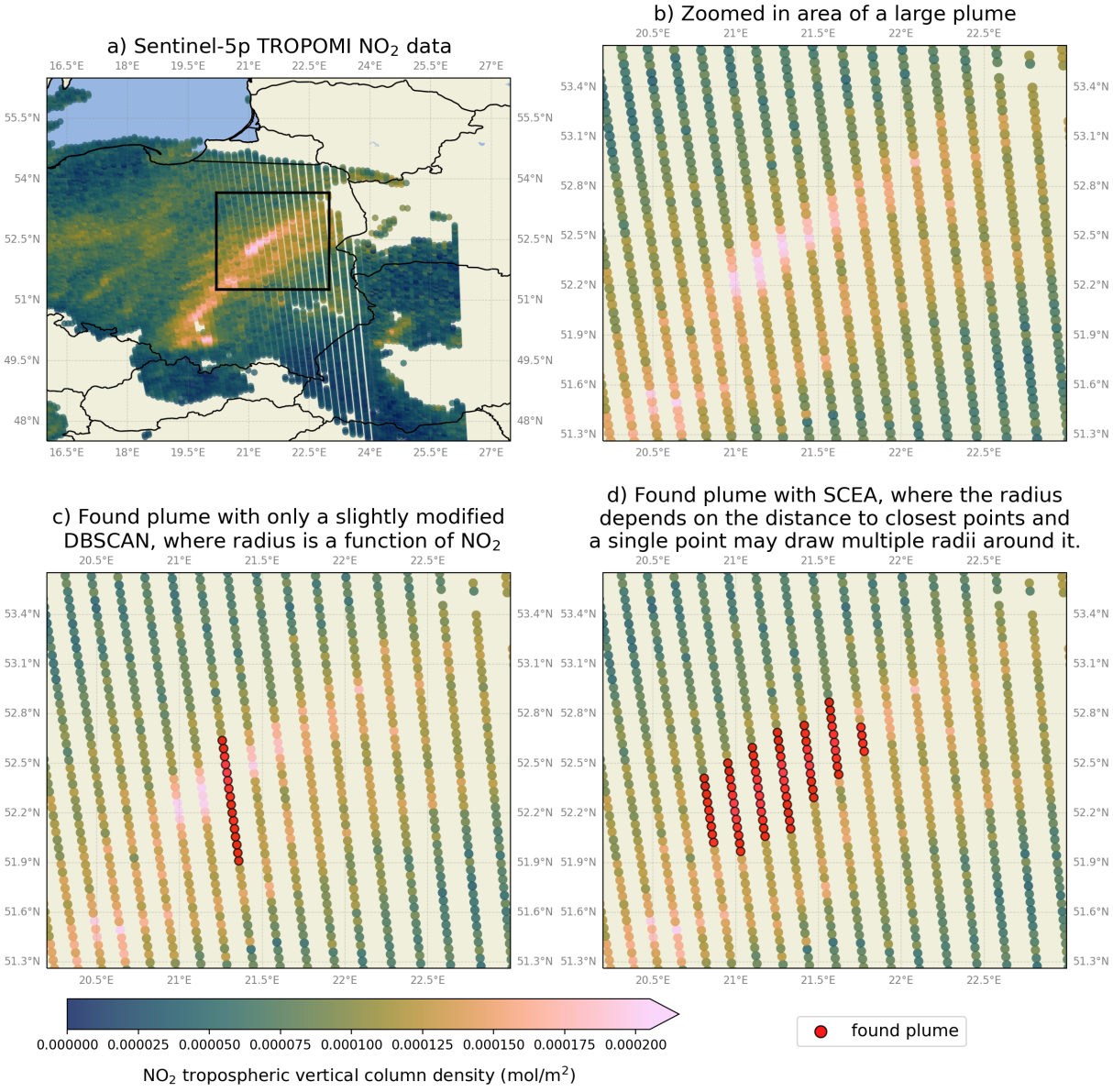

Figure 6: Comparisons of two different DBSCAN [1] based algorithms on plume detection. In c), there are the clustering results of a slightly modified version of DBSCAN, where the length of a radius around a point is a function of the point's value, where greater values correspond to greater radii. In d), the SCEA algorithm is used, where the radius is the distance to the closest point scaled by a factor determined from the value of the point. Moreover, a single point may draw multiple different radii over multiple iterations. Note that in this figure, the axis of the satellites forward movement is much more denser compared to the axis of the satellite's swath, which can be a challenge for plume detection methods. This demonstrates the need to define the radius in a more robust way, combined with the concept of radiating point. Data from Sentinel-5p TROPOMI NO_2 measurements on date 2019-12-05.

Figure 6, where this simple solution is attempted. There the plume of interest is located at the outskirts of the satellite's swath, where the points are very sparsely located in the axis of the swath, due to the large viewing angle of the satellite. In Figure 6c, a slightly modified DBSCAN algorithm is tested, where points with higher measurement values get longer radii drawn around them. The varying density of points on different axes presents great challenges to the slightly modified DBSCAN algorithm.

In SCEA, the radius is also affected by the distance to nearby points, and a single point may have multiple different radii drawn around it in different iterations of the algorithm. The radius is defined as the distance to the closest point not in the cluster, which is multiplied by a coefficient determined by a function of the point's value called the *radius function*. This way of determining the radius for each point in SCEA is very robust to the scale of the data, making it suitable for use in a variety of different data sets. It can handle dynamically varying levels of density when it comes to the location of points, as can be seen in Figure 6d.

Moreover, due to the unique way of determining the radii, SCEA is robust to cases where there are missing values in the data set, which is a common problem with satellite data. A plume in satellite data may suddenly be cut off by a wall of missing data, but the method might still be able to identify points belonging to the same plume from other sides of the gap. This trait is further demonstrated in Section 3.3.

As can be seen in the Figure 6, the data from satellites are rarely collected in a regular grid on the surface of the Earth. The SCEA algorithm can take the raw data points collected from the satellite and does not require the data to be interpreted in a even regular grid, contrary to many other existing plume detection methods. For example, in [41], which is discussed in Section 2.3, they had Sentinel-5p TROPOMI CO measurements with pixel sizes 7 km across and 5.5 km along, but as far as I know, the watershed algorithm treats these two dimensions equal. The method from [38], discussed in Section 2.2, treats the image dimensions equally, although every Sentinel-5p TROPOMI NO_2 pixel is 7 km $\times$ 3.5 km. The method from [39], discussed also in Section 2.2, used 32 $\times$ 32 pixel images of CH_4 from Sentinel-5p TROPOMI for their neural network plume detection. These images can be up to 176 km $\times$ 448 km for larger viewing angles, but the method still treats the dimensions as equal length. However, the two plume detection methods in [38] and [39] are neural network-based plume detection methods, so the neural network might be able to implicitly learn the difference between the two axes.

3.2 SCEA algorithm

Here, the SCEA algorithm is introduced. It is divided into two parts: the way the method creates a single cluster from a starting point is described in pseudo-code in Algorithm 1, and the way that algorithm is used to find multiple clusters is described in pseudo-code in Algorithm 2. First, the terms, variables, and parameters are explained, followed by a more thorough verbal description of the algorithm's flow.

Before explaining all input parameters to the algorithm, some key terms are defined. The process of adding new points to a cluster in SCEA comes from drawing radii around points already in the cluster, and if a new point falls within the radius, it gets added to the cluster. We call a point that draw these radii around them a *radiating point*. In each iteration of an algorithm the radius is drawn for each radiating point. The newly added

Algorithm 1: SCEA - Find One Cluster

Input:

$P \leftarrow \{\mathbf{p}_0, \ldots, \mathbf{p}_d\}$: set of points,
$V \leftarrow \{v_0, \ldots, v_d \mid v_i \in \mathbb{R}\}$: values of points
`startingPoint` $\in P$: first point to append to the cluster

Parameters:

`maxPointsInStartingRadius` $\in \mathbb{N}$, a point stops "radiating" if there are more than this number of points in the starting radius.
`radiusFunc()`
`distance()`
`localAreaRadius` > 0

Output:

C: set of points, a cluster

1 $P \leftarrow \{\mathbf{p}_i \in P \mid$ `distance(`*startingPoint*, $\boldsymbol{p}_i$`) < localAreaRadius`$\}$ `// Take only the local area around the starting point.`
2 $V \leftarrow \{\mathbf{v}_i \in V \mid$ `distance(`*startingPoint*, $\boldsymbol{p}_i$`) < localAreaRadius`$\}$
3 Standardise points in V.
4 $C \leftarrow$ {`startingPoint`} ; `// Initialise the cluster with a starting point`
5 $C_R \leftarrow$ {`startingPoint`} ; `// Set of "radiating points"`
6 **while** C_R *is not empty* **do**
7 **foreach** $\mathbf{p}_i \in C_R$ **do**
8 `startingRadius` $\leftarrow$ `min(distance(`$\mathbf{p}_i, \mathbf{p}_j$`))` $\forall \mathbf{p}_j \notin C$;
9 **if** number of points around $\mathbf{p}_i$ in radius of `startingRadius` > `maxPointsInStartingRadius` **then**
10 remove $\mathbf{p}_i$ from C_R
11 skip to the next iteration at line 7
12 **end**
13 `pointsToAdd` $\leftarrow$ `distance(`$\mathbf{p}_i, \mathbf{p}_j$`) < startingRadius` $\cdot$ `radiusFunc(`v_i`)`, $\forall j \neq i$;
14 **if** *pointsToAdd is empty* **then**
15 remove $\mathbf{p}_i$ from C_R
16 skip to the next iteration at line 7
17 **end**
18 C.`append(`*pointsToAdd*`)`;
19 C_R.`append(`*pointsToAdd*`)`;
20 **end**
21 **end**
22 **return** C

Algorithm 2: SCEA - Find multiple clusters with maximum value threshold method

Input:

$P \leftarrow \{\mathbf{p}_1, \ldots, \mathbf{p}_d\}$: coordinate of points

$V \leftarrow \{\mathbf{v}_1, \ldots, \mathbf{v}_d\}$: value of points

Parameters:

t: threshold for stopping the search of clusters

Output:

C: set of clusters

```
1 i ← 1
2 while max(V) > t do
3     StartingPoint ← (p_argmax(V), max(V)) // Starting point is the one
          with maximum value
4     C_i ← findOneCluster(P, V, StartingPoint) // Algorithm 1
5     C.append(C_i)
6     remove elements of C_i from P and V
7     i ← i + 1
8 end
9 return C
```

point to a cluster will always start as a radiating point and will stop radiating when it either fails to find new points or there is a sufficient amount of clustered points nearby. When calculating the radius for each radiating point, the *starting radius* is the first step in determining the radius. The starting radius is the distance to the closest point not in the cluster. After that, the length of the starting radius is multiplied by the output of the radius function, which takes the value of the point as input.

Data the algorithm 1 takes as an input are as follows:

1. the set of spatial location of points $P = \{\mathbf{p}_1, \ldots \mathbf{p}_n\}$, where $\mathbf{p}_i$ are elements of a metric space,
2. the set of values of points $V = \{v_1, \ldots, v_n\}$, where $v_i \in \mathbb{R}$,
3. `startingPoint` $\in P$.

The points in P are the spatial location of the data points, which are elements of some metric space. A metric space X is a set to which a distance metric is defined, [80]. A function d is a metric if, for all $x, y, z \in X$, it satisfies the following three conditions:

(i) $d(x, z) \leq d(x, y) + d(y, z)$ (triangle inequality)

(ii) $d(x, y) = d(y, x)$ (symmetry)

(iii) $d(x, y) = 0 \iff x = y$.

For example, $\mathbb{R}^n$ is a metric space with the natural Euclidean distance.

The set of values of the points V is, for example, the amount of NO_2 from satellite measurements. The `startingPoint` is a point from P where the clustering starts, for

example, the highest-valued point in the data set, local maxima or some known location of a point source.

To fine-tune the inner workings of the algorithm 1, the following parameters and functions can be adjusted:

1. `maxPointsInStartingRadius` $\in \mathbb{N}$,
2. `localAreaRadius` $\in \mathbb{R} > 0$,
3. `radiusFunc(`v`)` $\mapsto \mathbb{R}$, where $v \in \mathbb{R}$, is the measurement value of the point.
4. `distance(`$\mathbf{p}_1$, $\mathbf{p}_2$`)` $\mapsto \mathbb{R} \geq 0$, where $\mathbf{p}_i \in P$. Calculate the distance between two points according to some metric.

The parameter `maxPointsInStartingRadius` is the maximum number of points that can be in the starting radius of a radiating point. Effectively, the larger this value, the more likely it is to handle irregular grids and jump large gaps between points. From empirical testing, a value between 5 and 7 usually works.

The `localAreaRadius` is a parameter that determines the size of the local area that is considered when finding the cluster. All points outside this radius are removed as the first step of the algorithm 1. Higher values of `localAreaRadius` make the calculation of the pairwise distance more expensive, and the standardisation of V is done on a more broad scale. The distance function for this operation does not necessarily have to be the same as the distance function used in the latter parts of the algorithm. In the code implementation, L_∞-norm was used around the starting point to reduce computation time, which is defined as

$$||\mathbf{x}||_\infty = \mathtt{max}(|x_1|, \ldots, |x_n|),$$

where $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{R}^n$. This means that it considers only points within the set radius in each dimension, which is much faster than calculating Euclidean distances. In two dimensions, this looks like a square around the starting point.

The `radiusFunc()` returns a coefficient that is multiplied by the length of the starting radius. For plume detection, a general radius function was developed:

$$\mathtt{radiusFunc}(v) = \mathtt{min}(1 + v - r_\sigma,\ 2), \tag{2}$$

where r_σ is a value for how many standard deviations away from the mean value the measurements are and v is the normalised measurement input value. Since the values of x have a mean of 0 and a variation of 1, the shifted value $v - r_\sigma$ is positive for the values of v where it is above the standard deviation threshold set in r_σ. Thus, $1 + v - r_\sigma$ is greater than 1 for these values and therefore increases the starting radius. Moreover, in Equation (2), the value 2 is a cut-off value, which limits the output to that level at maximum. In the tests, in Sections 3.3 and 4, the value was set to 2.

The function `distance()` calculates the distance between two elements of a metric space. The distance can make use of any metric, but the natural Euclidean distance was used with the WGS 84 ellipsoid coordinates as axes in tests in Section 4. It is considerably faster than using more accurate metrics for points on Earth, and the error is small when the data is not near the poles. For a more accurate distance metric, one could use the

Harvesine distance for points on Earth, i.e., the distance of two points along the surface of a sphere, which is defined as

$$d_h(\mathbf{p}_1, \mathbf{p}_2) = 2r \arcsin\left(\sqrt{\sin^2\left(\frac{\phi_2 - \phi_1}{2}\right) + \cos(\phi_1) \cdot \cos(\phi_2) \cdot \sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)}\right),$$

where $\mathbf{p}_1 = (\phi_1, \lambda_1)$ and $\mathbf{p}_2 = (\phi_2, \lambda_2)$ are points with longitude ϕ and latitude λ in radians and r is the radius of the sphere. Although not exactly spherical, when talking about Earth, the globally average radius is most commonly considered to be 6 371 km, as it is very close to the three reference values provided by the International Union of Geodesy and Geophysics, [81].

Next, the flow of the algorithm is described. The method starts with algorithm 2, where the data set P, V is inputted and a threshold t is set as a parameter. The algorithm finds clusters one at a time starting from the maximum valued point, which is saved to the variable `startingPoint`. Generally, the algorithm 2 is for providing the algorithm 1 with starting points. Alternative approaches could be considered, for example, trying to find local maxima in the data set with some method, and starting the clustering with those.

The algorithm 1 is called with P, V and `startingPoint` as input, with parameters `maxPointsInStartingRadius`, `localAreaRadius`, `radiusFunc()` and `distance()`. First, the algorithm 1 considers only points in a certain distance from the starting point as candidates for the cluster. This helps with the standardisation, since points far away will not affect the local trends around the starting point. Moreover, it helps with the computation time since there are fewer pairwise distances to be computed, which can be a heavy operation with large data sets. The size of the area around the starting point can be tuned with the `localAreaRadius` parameter.

The `startingPoint` is then added to the cluster C and to the set of radiating points C_R as seen in Figure 7b. Every point that gets added to the cluster starts as a radiating point, which means that it can add nearby points to the cluster. A point can stay radiating for multiple iterations.

In each iteration, the starting radius is first determined for every radiating point, which is the distance to the closest point not in the cluster, as can be seen in Figure 7c. If the starting radius contains too many points that are already in the cluster, then the point stops radiating (Figure 7f). Then the length of the starting radius is multiplied by the output of the `radiusFunc` (Figure 7d), which takes the value of the point v_i as an input. All points that fall within the new radius get added to the cluster and are added to the set `pointsToAdd`, which is later appended to the set of radiating points C_R (Figure 7e). If the radius function outputs a number that is less than 1, then no new points get added, since there are only points on the edge of the starting radius due to the radius being the distance to the closest one not in the cluster. If no new points are added, the point stops radiating (Figure 7h). If the radius function value is greater than or equal to 1, at least one point gets added to the cluster. The algorithm then proceeds to check all radiating points in that iteration the same way. If there are no radiating points left for the next iteration, the method is done finding the cluster C and returns it.

Once the cluster is found, we return to the algorithm 2 and all the points that were added to the cluster C_i are removed from the data set. The algorithm then finds the next point with the maximum value from the remaining data set. We repeat this until the

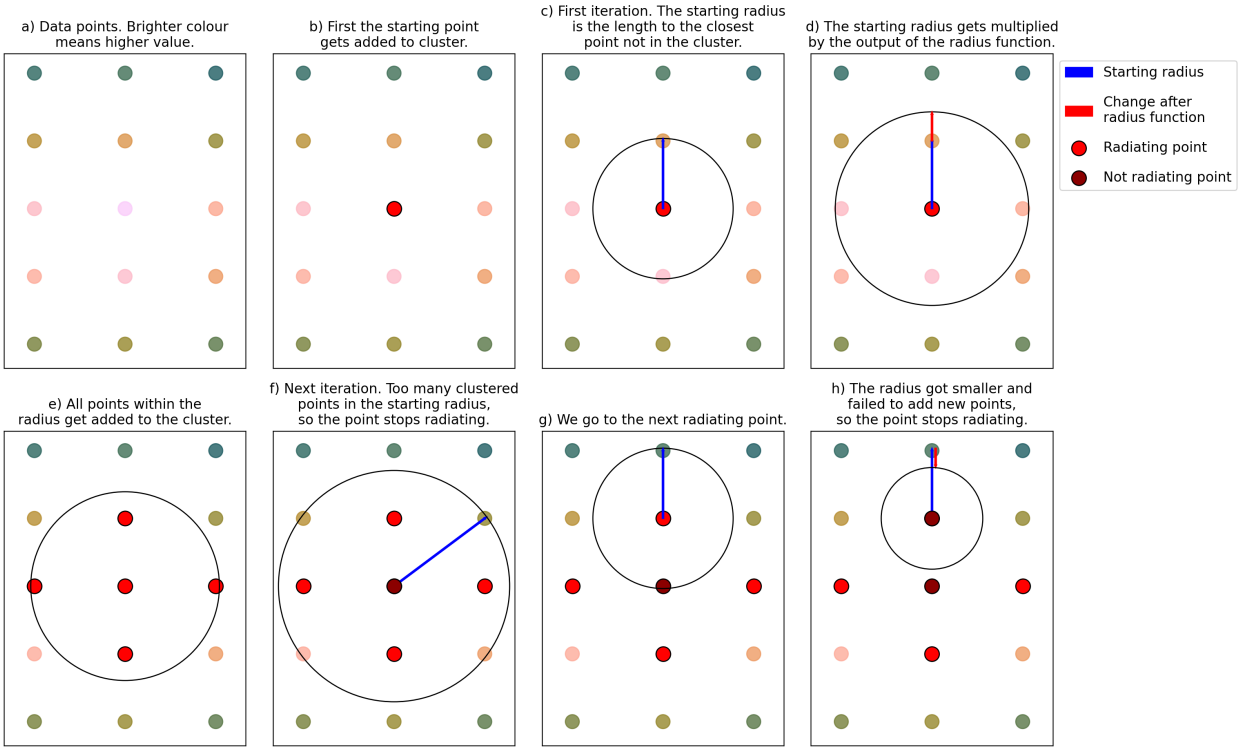

Figure 7: Example of the workflow of the SCEA algorithm.

maximum value of points is below the threshold t. After that, the algorithm stops and returns the set of clusters $\mathbf{C}$.

Although there seem to be a lot of parameters to tune, there is not much tuning to be done in practice. The distance metric is trivial to choose in many applications, since there is a limited number of options, and the `localAreaRadius` parameter does not have big effects on the output from empirical testing. The `maxPointsInStartingRadius` was usually kept at 5, and there seemed to be no reason to go any higher than 8. The threshold t might need some tweaking depending on the application. In the implementation, this parameter was set as some number of standard deviations above the median, as it works quite generally for many different data. That is,

$$t = \text{med} + s|\sigma|,$$

where med is the median of the data set, σ is the standard deviation of the data set and s is a coefficient that can be tuned. The radius function introduced in Equation (2), was used in the implementation, as it can work with various data sets with just a little tuning of r_σ. The implementation of algorithms 1 and 2 in Python can be found under the GitHub link in appendix A.

3.3 Example results of SCEA on toy data and real world satellite data

To demonstrate the performance of SCEA, we provide examples of the SCEA algorithm applied to toy data sets and to real-world satellite data in an attempt to detect plumes. Various toy data sets were generated to test SCEA. The toy data sets features high-valued

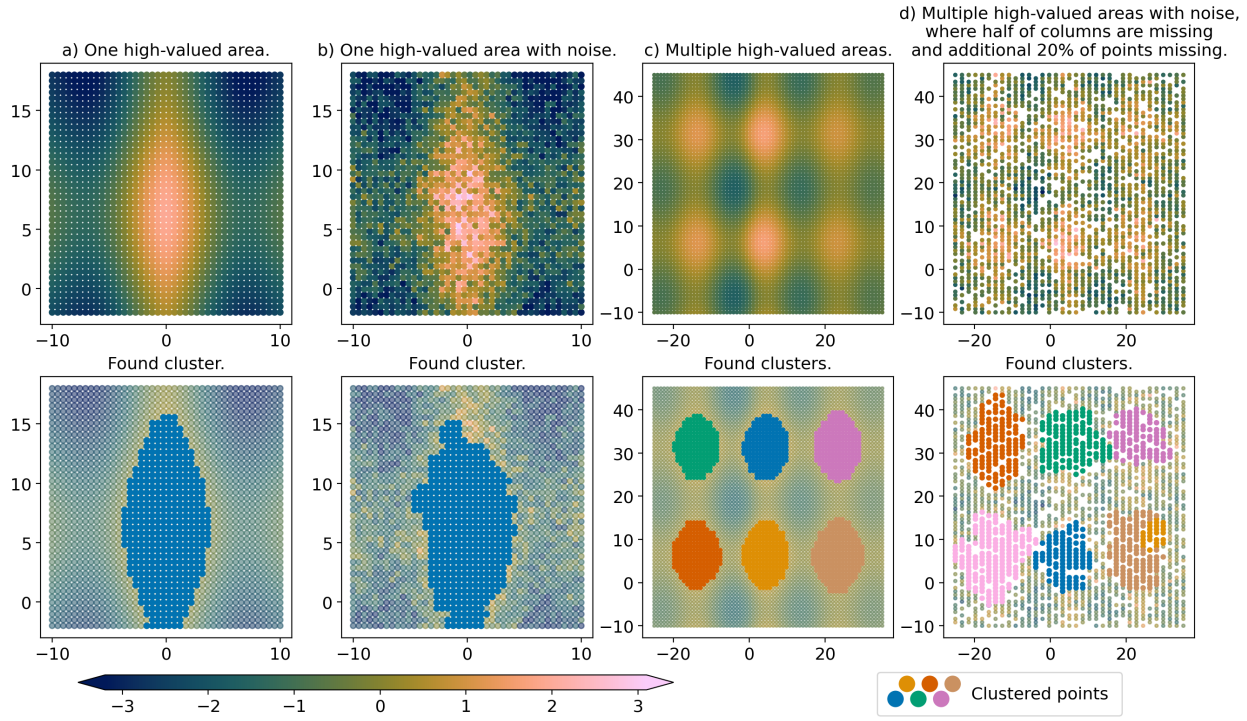

Figure 8: Demonstrations of SCEA with toy data, where in the bottom row of subfigures are the clusters found from the corresponding data above. Points belonging to the same clusters are denoted with the same colour. In a) there is a single high-value area in the figure, created with the function (4). The points are equally spaced, with 42×42 points on each axis. In b), there are the same data as in a), but with added Gaussian noise to make clustering harder, as seen in function (5). In c), there are six high-value areas present in the data with 70×70 equally spaced points on each axis, created with the function (6). In d) there are the same data as in c), but with added Gaussian noise, as can be seen in function (7), with half of the point columns missing, and in addition to that, 20% of the remaining points are missing.

areas of varying shapes and sizes with different challenging scenarios of missing data and uneven spacing. In many of the toy data sets, Gaussian noise is added, which means adding a random term to each point sampled from the normal distributions probability density function

$$N(x \mid \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \tag{3}$$

where μ is the mean value and σ is the standard deviation. A sampled value of the normal distribution (Eq. 3) will be referenced with $\varepsilon \sim N(\mu, \sigma)$.

In Figure 8, multiple different examples of SCEA are presented, where it is challenged with various toy data sets. The toy data sets in each subfigure of Figure 8 have been created

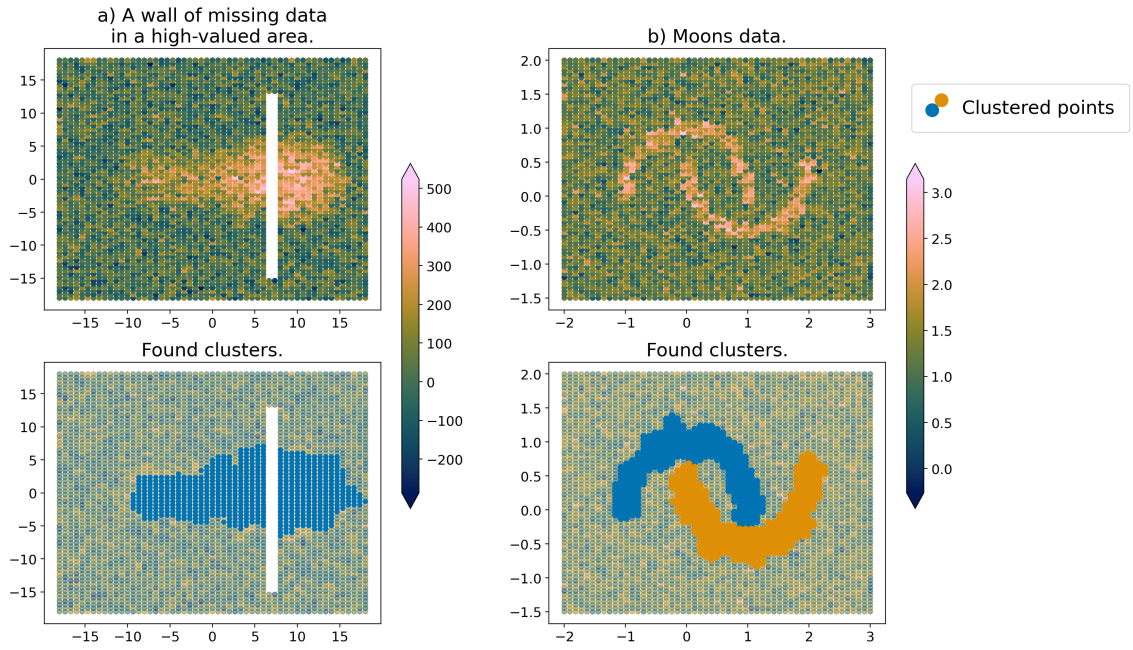

Figure 9: More demonstrations of SCEA on toy data sets. In a) there is a high-valued area separated by a gap of missing data. Moreover, the data is slightly more sparse on the direction of the x-axis, which makes the gap even larger. In b) there are two high-valued areas in the shape of crescent of moon, which demonstrates the algorithms ability to find areas of arbitrary shape. Note that in Figures a) and b), the values are in a completely different scale.

with the following functions indexed with the corresponding subfigure.

$$f_{8a}(x, y) = \cos(x/2) + \sin(y/4) - 0.2|x| \tag{4}$$

$$f_{8b}(x, y) = \cos(x/2) + \sin(y/4) - 0.2|x| + \varepsilon, \quad \text{where } \varepsilon \sim N(0, 0.6) \tag{5}$$

$$f_{8c}(x, y) = \frac{\sin(x/3) + \sin(y/4))}{0.05|x| + 1} \tag{6}$$

$$f_{8d}(x, y) = \frac{\sin(x/3) + \sin(y/4)}{0.05|x| + 1} + \varepsilon, \quad \text{where } \varepsilon \sim N(0, 0.6). \tag{7}$$

In functions 4 and 5, there are trigonometric functions with an additional term $-0.2|x|$, which grows more negative the further we get from $x = 0$, to emphasise only one high-valued area. In functions 6 and 7, the trigonometric functions are divided by an additional term $0.05|x| + 1$ to reduce the height of the peaks as we move away from $x = 0$, resulting in peaks of different sizes. Additional Gaussian noise is added to functions 5 and 7, to make the task harder. Figure 8 presents an even more challenging situation, with half the sampling density on the x-axis compared to the y-axis and additional 20% of points randomly removed. This shows that the SCEA algorithm is robust to varying levels of sampling densities and missing data.

In Figure 9 additional difficult scenarios are presented to SCEA. Figure 9a is created

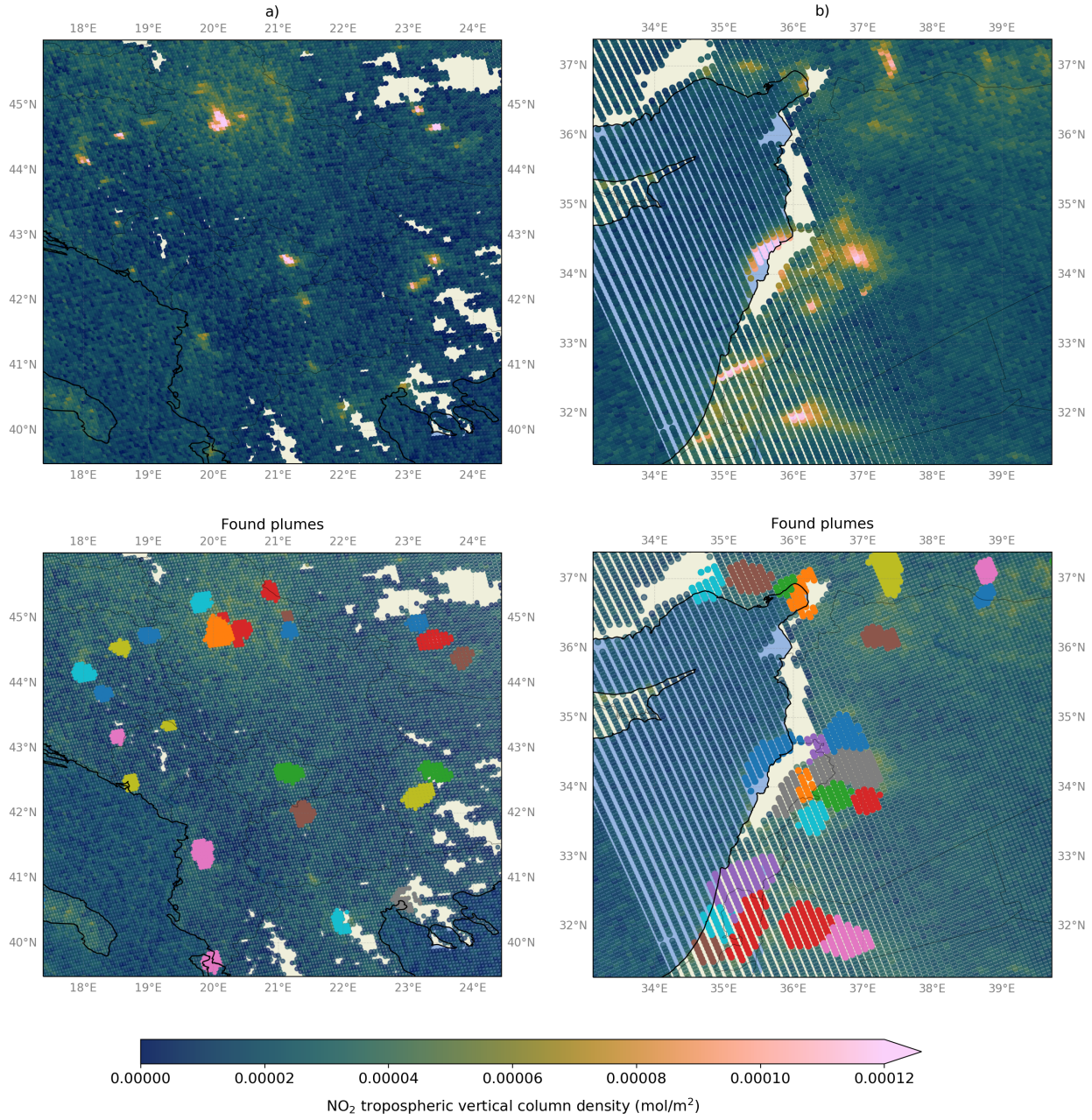

Figure 10: SCEA used as a plume detection method on two different scenes, with bottom row of subfigures showing the results. Data from different scenes of Sentinel-5p TROPOMI NO_2 instrument on date 2019-08-06. Quality filter of 75 % was used as recommended by the Sentinel-5p data manual, [14]. Same SCEA parameters used for both a) and b) scenes. Note that different clusters may appear having same colour due to the limited range of colours in the palette.

with the following function

$$f_{9a}(x,y) = \begin{cases} g(x,y), & \text{when } g(x,y) > 0 \\ 0, & \text{when } g(x,y) \leq 0 \end{cases} + \varepsilon \sim N(0,80), \tag{8}$$
$$\text{where } g(x,y) = -0.01x^4 + 1.3x^2 - 6y^2 + 12y + 180.$$

In Figure 9a, a single large high-value area is present, but a wall of data is removed in the middle of the area. Furthermore, the data are sampled 29% sparser in the direction of x-axis, making the gap even more difficult. SCEA is still able to jump the gap and identify the areas as the same cluster.

Figure 9b is created with the help of the Scikit-learn Python library's method called make_moons(), which samples random points in the shape of crescents of two moons opposite to each other, [47]. A mesh grid is created on top of the sampled points with values 1. If a point in the mesh grid is within a radius of 0.1 from a sampled moon point, it gets a value of 2. After that, Gaussian noise is added with parameters $N(0, 0.4)$. This figure demonstrates well the ability of SCEA to find areas of arbitary shape.

Examples of SCEA applied to real-world Sentinel-5p TROPOMI NO_2 satellite data are presented in Figure 10, where two different scenes are analysed. The figure demonstrates the ability of SCEA to be used as a plume detection method for real-world satellite data. However, it is hard to say how many plumes were missed, because it is challenging even for a human to pin-point all the spots with plumes.

4 Evaluation of SCEA as a plume detection method with the SMARTCARB data set

The SCEA algorithm, introduced in Section 3, was developed with plume detection problems in mind. In this section, the SCEA algorithm is tested as a plume detection method. Evaluation of the performance of plume detection methods is not easy because there are large amounts of uncertainties when dealing with satellite data. In Section 4.1 the difficulties are discussed in detail. In Section 4.2 the SMARTCARB data set is introduced along with the emission inventory used there, the TNO-MACC III data set. In Section 4.3, specifications of the study are laid out, and in Section 4.4 the results of the experiment are analysed.

4.1 The problem of measuring the performance of plume detection methods

To compare different plume detection methods, a way to measure their performance is needed. This is not a trivial task, since using traditional classification metrics like accuracy, precision, or recall (Eq. 1) is not possible without knowing the true labels. Due to the nature of satellite measurements, there is no easy and objective way to label a detection as a real plume or as background noise. This means that it is troublesome determining metrics based on true positive (a detection is an actual plume), false positive (a detection is not a plume), true negative (no detection where there is no plume), and false negative (no detection where there actually is a plume).

Labelling detections as true positives could be done with co-location of known point sources of emissions using, for example, official emission registers like the European Pollutant Release and Transfer Register (E-PRTR) [82] or the Open Source Data Inventory for Anthropogenic CO_2 (ODIAC) [66], but there are several problems with these inventories. These tend to have missing, outdated, or inaccurate data due to the vast number of reporting practices, inaccurate measurements, technical errors, or intentional underreporting of the data, [7, 83]. When there is money involved in emitting, as is the case with carbon emission trading schemes, there is a financial incentive to underestimate or hide emissions. Even if they are perfectly reported, emission inventories have a lot of possible downfalls when it comes to using their point sources as true labels for plume detections. It is impossible to cover all possible types of emission globally with accurate measurements. There might be accidental gas leaks, moving point sources like ships, or natural phenomena that cannot be covered by any emission inventory.

Moreover, the reported emissions are usually annual, and it might not reflect at all what a point source is emitting at a certain snapshot in time. Point sources usually do not emit all the time at the same rate since their emissions might fluctuate based on the time of the day, week, or year. For example, peaking power plants only operate in peak energy consumption times, which can be very narrow time frames, and may be operating annually for only a couple of hours and can still emit massive amounts of greenhouse gases, [84]. The plumes created by these point sources might be transported far away from the source after operating hours, thus making it very difficult to connect plumes to the actual source if the satellite measurement happens afterwards. Most of the greenhouse gas measuring Earth observation satellites are sun synchronous, meaning that they always

measure at the same local solar time. This makes them have a very narrow coverage in terms of time of day, thus creating a bias in the measurements. There could exist a real risk of strategically emitting less during the times of satellite overpasses if satellites continue to play a bigger role in the monitoring and verification of greenhouse gases. The position and orbit information of satellites are public information, making an overpass of a satellite easy to predict. Thus, one could emit less during these times since there is financial incentive to do so. This adds to the pitfall of using emission inventories as a label for truth.

Having only a measure for true positives or false negatives (e.g. was the method able to find this point source's plume or not) and not for false positives can be misleading and might not be an objective way to measure the performance of plume detection methods. In [38], discussed in Section 2.2, they measured their plume detection method based on whether they could map the detected plumes to an existing inventory of emitters and concluded that they found 56 % of the point sources in the ODIAC inventory, which can be interpreted as the recall of the method. If the method sensitivity is tuneable, which was the case in their plume detection method, one can tune the sensitivity arbitrarily large and find almost all of the positives, but do not get punished by the large number of false positives that emerge as a side effect. Taking this to the extreme, one can label every point as a plume and get 100 % accuracy on all emission inventories. They concluded that detections that are not near any known source might very well be genuine detections, which sounds highly plausible, but it still highlights the problem of assigning an objective performance metric to a plume detection method.

As discussed in Section 2.2, in [39] it was reported that their plume detection method achieved a precision of 76 %, which is the proportion of true positives out of all samples marked as positive by their method. This was determined by a manual inspection, which is not objective at all and is not possible to verify. Note that the precision does not take into account false negatives, that is, plumes missed by the method. Again, taking the abuse of measurement metric to the extreme, one could find only one plume and achieve a precision of 100 %.

Determining a negative result (no plume is found) between true or false is also extremely challenging, since most point sources give little to no signal due to the small amounts of emissions produced, and these might not present in any emission inventories. There are numerous factors that affect the atmospheric concentration of greenhouse gases, such as changes in background concentration and biospheric fluxes, which have their own annual, synoptic, and diurnal cycles, [85, 86, 87]. When it comes to XCO_2 satellite measurements, these can represent more than half of the signal, and the combined standard deviation around the annual mean can be up to 30 ppm on the surface, [88]. Furthermore, high winds and other meteorological conditions can make small emission points impossible to detect and might cause artefacts in the data. On top of all this, the precision and the quality of measurements of the satellite instruments cause their own inevitable noise in the data, making small emissions undetectable.

A potential way to investigate if a detection is true or false is to combine other available information like satellite images, building data, meteorological data or geographical data to further investigate, but that brings its own challenges into the equation. A topological structure or a change in surface type might cause an artefact in the satellite measurements, and these can resemble the shape of a plume. In [39] they first had a neural network

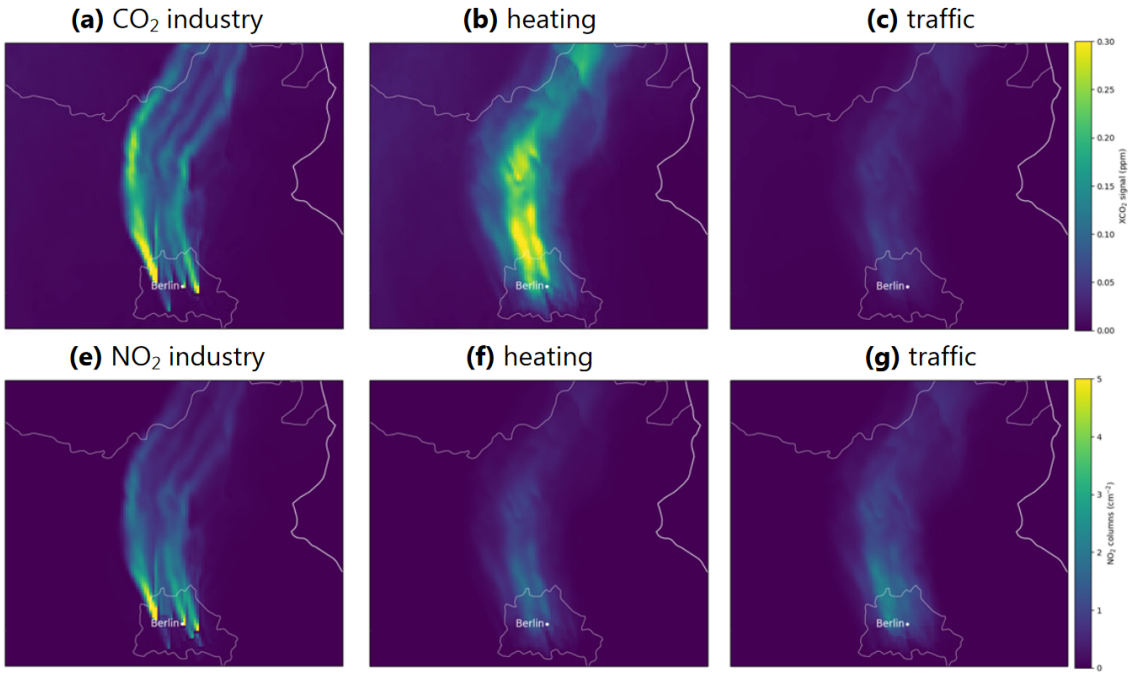

Figure 11: Estimated emissions from different sectors for the city of Berlin. This demonstrates that an area source like a city consists of multiple smaller emissions sources, which together add to a large plume. This is one of the reasons a plume might be hard to connect to a single unambiguous source. Figure source: [2].

detect plume structures and then a second step with a support vector machine to further investigate the detection with additional information like sounding parameters, weather conditions, and topological maps. Further ways to tackle this would be to try to find a signal in multiple satellite overpasses or take temporal averages of a location, given that the source is static in a location. If a plume is found in a significant number of overpasses, there might be a real signal there, although not all point sources emit for more than one overpass of a satellite. One of the biggest assets a plume detection model can offer is the discovery of new unknown emission sources, so it can be very harmful to falsely punish a detection as a false positive when it is actually a newly found emission source.

Area sources can pose even more problems when it comes to labelling plumes. Area sources may not have a well-defined area, and the emissions can fluctuate a lot locally and can be continuous in nature when it comes to the location and the amount of emissions, which is illustrated in Figure 11. These usually consist of numerous tiny point sources of various kinds that together add to a clear signal that can be seen as a plume. Large cities are the most noticeable area sources that typically consist of traffic emissions, industrial areas with multiple flue gas stacks, residential combustion and even human and livestock respiration, which can reach up to 30% of the fossil fuel emissions in some cities, [89]. Determining where an area source starts and where it ends is highly subjective. Inside cities, there are usually large point source emitters, but these mostly blend in with the area source, making it difficult to pinpoint it there.

Biomass burning is also a type of area source that needs to be taken into account, which includes human-initiated burning as well as natural wildfires. They can produce a

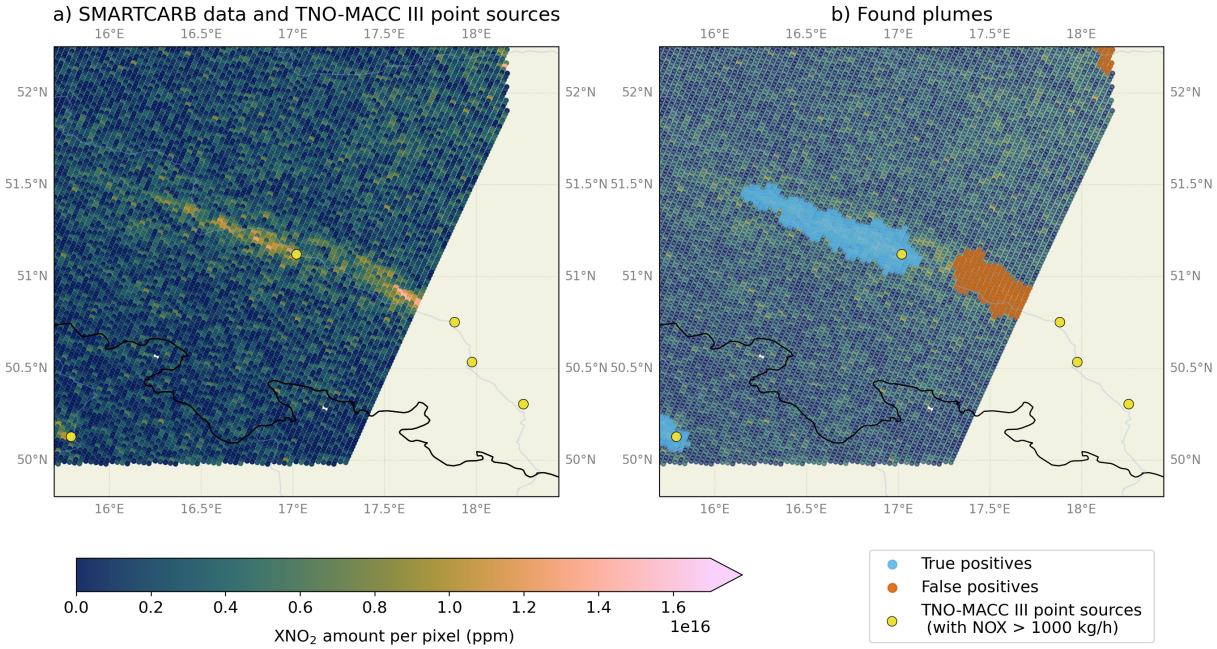

Figure 12: SMARTCARB data on date 2015-07-03. If we use location of known point sources as an indicator of a plume being true positive or false positive, we can run in to the situation where the point source is outside the bounds of valid pixels, but its plume is still visible in bounds. Looking at the figure, it seem very likely, that the other detected plume is a true detection, but labelled as a false positive.

very distinct plume in satellite measurements. If the goal is to find anthropogenic point sources with plume detection methods, it might be useful to disregard biomass burning. Fortunately, there are quite sophisticated remote sensing tools to detect fires such as NASA's MODIS data product, [65]. In [38] the authors ran their NO_2 plume detection model across the globe and ignored every plume that was within 15 km of a biomass burning site identified by the Visible Infrared Imaging Radiometer Suite (VIIRS) as a thermal anomaly to differentiate biomass burning from fossil fuel emissions.

The challenges do not end there. Even if we had perfect knowledge of the point source or area source locations, there are still a lot of problematic scenarios ahead. If a point source is just outside of a satellite sounding area, it could still show a clear plume signal due to the wind transporting the plume inside the bounds like in Figure 4.1. This plume is very difficult to assign to a specific point source or label as a true positive. One would need to rely on accurate information on wind speed and wind direction and sophisticated inversion techniques to know the exact location of the emission sources. Moreover, the atmosphere is highly chaotic in nature and gusts of wind may temporarily disconnect the tail of a plume, making one plume see like two plumes. It is also not unusual to get a disconnected plume due to bad quality data or missing pixels, like in Figure 13a, which is a very common sight in satellite data when using appropriate quality filters. Often multiple plumes overlap, making it very difficult to know if multiple detected plumes near each other should be considered as one or if one large plume should be considered as multiple smaller ones, like in Figure 13b. Even with perfect knowledge, it might still be ambiguous.

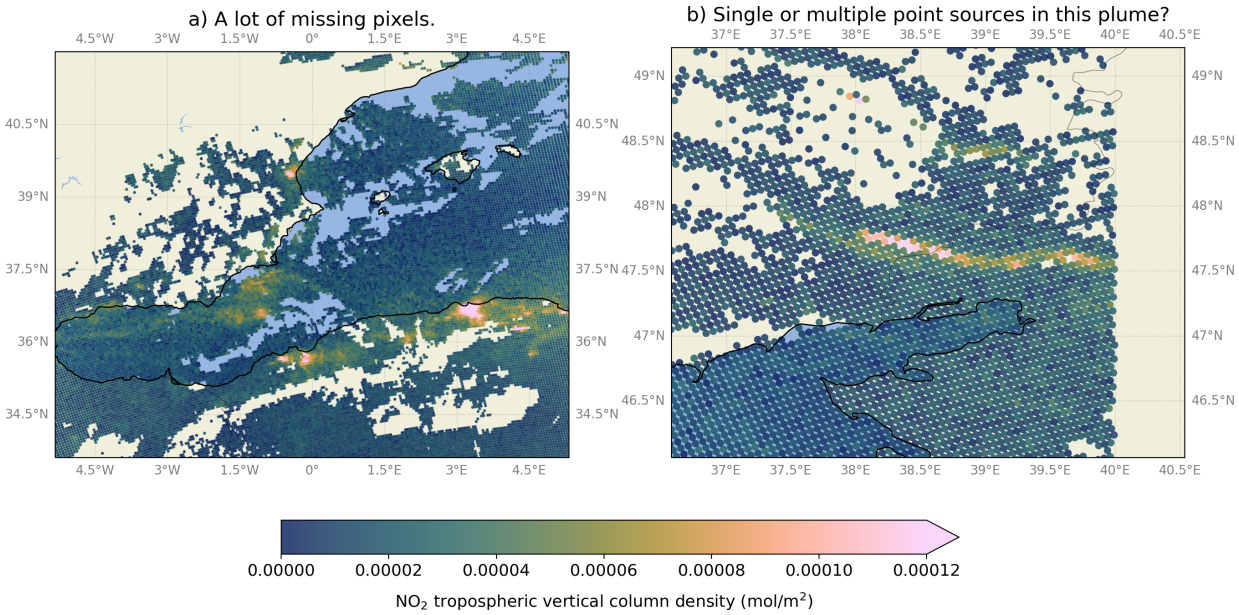

Figure 13: a) A common sight in satellite data. A lot of missing pixels when setting the quality filter to 75 % as recommended by the Sentinel-5p data manual, [14]. Many potential plumes are broken up by missing pixels. b) A very long tailed plume. Does this consist of multiple different point sources or just a single one? Looking just by the satellite data it is really hard to tell. Data from Sentinel-5p TROPOMI NO_2 instrument on date 2019-08-07.

The ability to find a plume and the source associated with it is not the only way to measure a plume detection method. Another way is to examine the pixels that are labelled in the plume. An asset a plume detection method might offer is the outline of a plume. However, there are also plume segmentation methods designed to do only that, [61, 62, 63]. Most plume-based emission estimates require outlining the boundaries of a plume and determining exactly which pixels are considered in the plume, [50, 30]. In [28], as discussed in Section 2.1, the authors of SMARTCARB tested their plume detection method with the synthetically simulated satellite data of SMARTCARB. In the work, the "correct" plume pixels were determined with the simulations without background variability or noise, and a threshold was placed to that. A correctly identified plume was defined as having at least 100 pixels detected and of all detected pixels, 80% are true pixels. This way of measuring a plume detection method is not feasible with real-world satellite data because it is highly ambiguous to say where a plume starts and where it ends and finding an objective measure for it is impossible.

Like in [28], one way to measure the performance of plume detection methods is to use generated toy data, or simulations as a test ground, since in theory every aspect of the simulations should be known, and we could more objectively assign every plume as a true plume or a false plume. Moreover adding noise to simulations could help us answer how robust the methods are to noise, since real-world satellite data can be very noisy at times. That is why this study is primarily focused on the simulated SMARTCARB data set.

4.2 Introduction to SMARTCARB and TNO-MACC III data sets

As discussed in the previous section, measuring the performance of a plume detection method is not an easy task with real world data. That is why we turn our attention to simulated data where we have much more knowledge of the mechanism underneath. Here the SMARTCARB data set is introduced, in which plume detection is attempted with the SCEA algorithm. To measure the performance, it is calculated how often a plume of certain point sources is correctly identified by SCEA.

The SMARTCARB (**S**atellite **M**easurements of **A**uxiliary **R**eactive **T**race gases for fossil fuel **CARB**on dioxide emission estimation) [2, 90] is a synthetic data set that simulates the data of the instruments on the upcoming Copernicus CO2M satellites. This data set offers an opportunity to work on methods for the provided data in advance of launch. It was a project initiated and funded by ESA in 2015 and led by Empa, the Swiss Federal Laboratories for Materials Science and Technology, with the main objectives of assessing the potential of plume detection in different noise scenarios with CO_2 instruments combined with concurrent measurements of NO_2 and CO, and analysing the potential of quantifying emissions based on these plumes. The possibility of plume detection was explored in [28], which was discussed in Section 2.1. The effect on revisit times of different-sized satellite constellations in different latitudes was also examined. The SMARTCARB project was concluded in January 2019. For the purpose of this case study of SCEA with plume detection, we are going to dive deeper into the data that are behind the SMARTCARB simulations.

The data set features simulated satellite-like gridded measurement data on CO_2, CO, and NO_2 for the whole year 2015 with different levels of noise over central Europe. An example of an overpass can be seen in Figure 14. The different available noise levels σ_{VEG50} for CO_2 are 1.0 ppm, 0.7 ppm and 0.5 ppm, for high, medium, and low levels, respectively. The noise σ_{VEG50} is defined as a so-called VEG50 scenario, which takes into account the features of the ground. This makes the noise more spatially dependent, making the noise more realistic to real-world satellite data. The different available noise levels σ_{ref} for NO_2 are $1 \cdot 10^{15}$ molecules cm^2 and $2 \cdot 10^{15}$ molecules cm^2 for high and low scenarios, respectively. The error in NO_2 is defined a bit differently compared to CO_2, but systematic errors such as ground features are also taken into account.

Multiple different data sources were used under the hood for the simulations. For most of the anthropogenic emissions, the TNO-MACC III inventory [91, 92] was used, but the largest point sources were treated separately and removed from the TNO-MACC III inventory. Also, the emissions of the city of Berlin were removed and the "Senatsverwaltung für Stadtentwicklung und Umwelt" inventory was used instead because of the higher spatial resolution. The emission inventories provide only total annual data, so to obtain hourly emissions, time-based scaling factors were used for the diurnal, weekly, and annual cycles, as described in detail in, [88]. For the biospheric fluxes of CO_2, such as photosynthesis and respiration, the VPRM (Vegetation Photosynthesis and Respiration Model) was used to produce hourly biospheric fluxes. For the simulations, the COSMO-GHG (Consortium for Small-Scale Modelling - Greenhouse Gases) numerical weather prediction model was used, which is specialised for simulations of the passive transport of greenhouse gases.

To assess the performance of plume detection, ground-truth labels of emission sources are needed. To identify them in the SMARTCARB simulations, focus needs to be directed on the TNO-MACC III data set, since the all input data are not available from

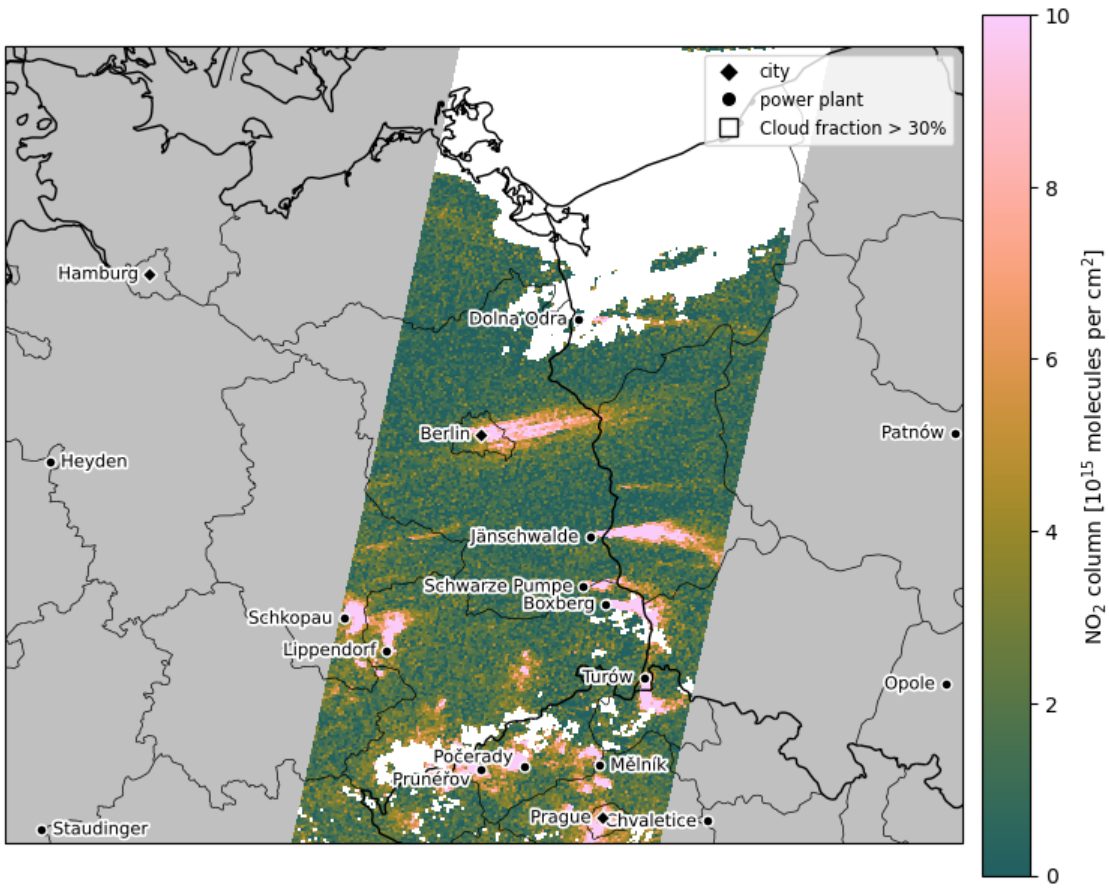

Figure 14: Example of the SMARTCARB simulations with NO_2 retrievals in a low-noise scenario. Figure data provided by the ICOS (Integrated Carbon Observation System) Carbon Portal Collaboration Server.

the SMARTCARB separately. TNO-MACC is a project developed by the Dutch Organisation for Applied Scientific Research TNO. The now legacy TNO-MACC project is replaced today by the CAMS (Copernicus Atmosphere Monitoring Service) project under the European Union's Earth observation programme Copernicus.

TNO-MACC III is a gridded inventory of estimated greenhouse gas emissions for the year 2011 with a relatively high resolution of 7 km × 7 km, [91, 92]. The SMARTCARB team used this data set as a source for most point sources since it is the most recent data set to 2015. In the inventory, they combined multiple bottom-up source estimations from different sectors. They mostly used the EMEP (European Monitoring and Evaluation Programme) database [93] as a source to which European countries report their emissions to air. It includes emissions of air pollutants by country, year, pollutant, and sector. For missing or bad data, they complemented it with the GAINS model from IIASA (International Institute for Applied Systems Analysis), or with TNOs own bottom-up emission inventory developed for this project. They also used their own national shipping emission data set developed from sea shipping emissions database MARIN and European emission data base for the year 2005 called PAREST.

After gathering all sector specific emission data, the TNO-MACC team had to distribute them spatially on a map. For the location of point sources they used the EU's register for air pollutants E-PRTR (European Pollutant Release and Transfer Register), [82]. Furthermore, for other point sources not covered by the E-PRTR, they used TNO's own database [94], where they studied electricity and large heat production plants, oil refineries, coke ovens, fossil fuel production plants, iron and steel production plants, production of non-ferrous metals, cement production, chemical industry, airports and harbours. For area sources such as road transport, residential combustion, and animal husbandry, they used the same TNO database [94] where they used proxies such as population density and road maps.

4.3 Pre-processing and methods

TNO-MACC III data for the year 2011 was obtained by contacting the authors of the now legacy data set used in SMARTCARB simulations, which includes the point sources and the amount of annual emissions per pollutant. Access to all SMARTCARB data was granted through ICOS (Integrated Carbon Observation System) services.

The SMARTCARB data set includes observations for both CO_2 and NO_2, but it was decided that the more plume detection friendly NO_2 retrievals from the data set were to be used. For the NO_2 soundings, three noise scenarios are offered in the simulations, high-noise, low-noise, and noise-free.

The first round of results included some weird anomalies. After some investigation, it turns out that within $\sim$ 50 km of the model boundary there is a relaxation zone where boundary conditions are merged into the model domain, making the simulation broken in those parts. It is recommended not to use the data near boundaries by the public data manual, [90]. Therefore, data at least 50 km close to the boundary were removed by narrowing it to only detections from longitudes of 8.3071919°E to 18.17086367°E and latitudes of 49.97822361°N to 54.6941934°N.

Since we are looking at the NO_2 retrievals of the SMARTCARB data set, the values of NO_x, which includes both NO and NO_2, were used for each points source from the TNO-MACC III. For the TNO-MACC III data set, all point sources within 2.828 km to each

other were combined as one, using the DBSCAN algorithm, which was also discussed in Section 3.1. In satellite measurements, nearby point sources together may seem as emitting only one plume, so that is why nearby points were clustered together, giving the "point" sources a better measure of their emissions. The number 2.828 was chosen for the radius, since the SMARTCARB measurements have a resolution of 2 km × 2 km and the diameter of a single pixel from corner to corner is $\sqrt{2^2 + 2^2} \approx 2.828$. For the combined clusters of point sources, the coordinate values were given the average values of longitude and latitude in the cluster and the NO_x values were summed together.

The SCEA algorithm was run through every overpass of the SMARTCARB data set, and for each overpass, the following things were checked. First, point sources further than 5 km from any valid data points were removed. For every point source remaining, it was checked whether a plume was detected within 5 km of the point source. If so, then that point source was interpreted as successfully found, and was marked as true positive. If there were no plumes near enough, the point source was interpreted missed, and was marked as false negative. Similar measures were taken from the perspective of plumes found by SCEA. For every plume detected in each overpass, it was checked whether a plume had any point sources within 5 km. If so, then that plume was interpreted as successfully found, and it was marked as a true positive. If not, that plume was marked as a false positive. As discussed in Section 4.1, satellite data contain large amounts of missing data, even the realistic SMARTCARB data set, so it is impossible to say if a signal of a plume came outside area of valid pixels or not. That is why every plume that was detected and had any missing values near it was labelled as having missing values near it.

For this case study, the SCEA algorithm was run with parameter values of $s = 3.3$, $r_\sigma = 1.3$, `maxPointsInStartingRadius` = 5, and `localAreaRadius` = 3, equipped with an Euclidean metric. The most important parameters for tuning the algorithm are s and r_σ. The s controls the sensitivity of how many plumes are detected, and the r_σ controls the sensitivity of what points are considered in a single plume. See Section 3.2 for a more thorough explanation of the parameters. The parameters were decided with a visual inspection of empirical testing. The same parameters were used in all noise scenarios.

4.4 Results

The results of how often each point source was found in different NO_2 noise scenarios offered by the SMARTCARB data set are shown in Figure 15. On the y-axis, there is the ratio of the number of times found divided by the number of times missed, which is the same as recall (Eq. 1). On the x-axis, there is the size of each point source in terms of annual emissions. To make the plot more readable, the x-axis is in logarithmic scale because the disproportionate frequency of point sources by emissions seems to sparse out exponentially when going larger in terms of emission amounts. The moving average is not taken in terms of NO_x, rather it is taken in terms of the indexes in the ordered list of point sources. It was taken this way because the moving window with a constant size in terms of NO_x did not make sense with the exponential nature of the point source sizes. See Figure 16 for the results with different ranges of NO_x emissions on a linearly scaled x-axis. The cumulative recall tells the expected recall of the plume detection algorithm when trying to find point sources at least a certain size. Some numbers from that plot are highlighted in Table 1.

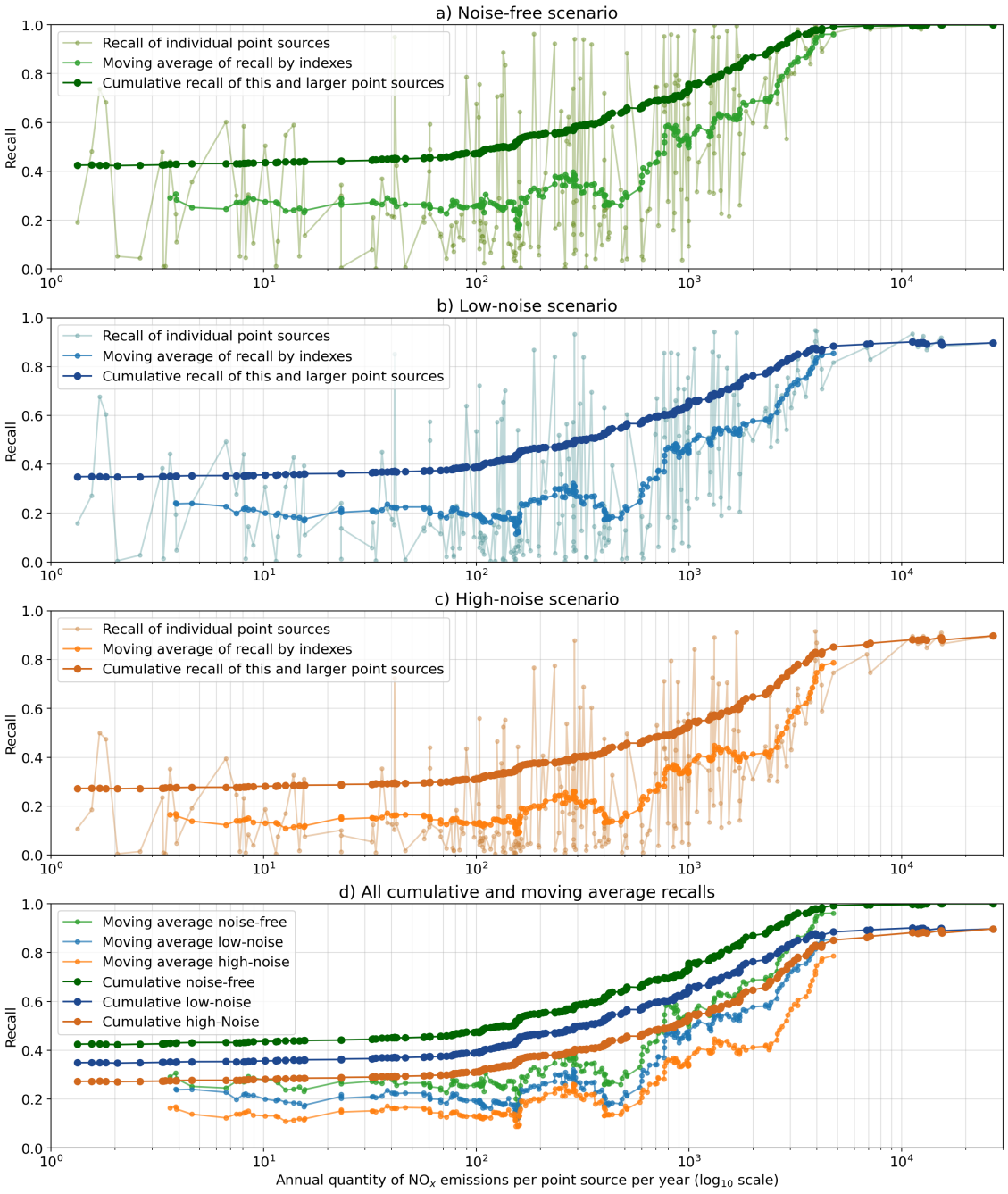

Figure 15: Percentage of how many times a point source was found by SCEA compared to the number of times it was present, or recall, in three different noise scenarios. In the x-axis there is the amount of emissions the point source emits annually in logarithmic scale. The moving average was taken with a centered rolling window of 20 indexes when sorting the facilities by emission size. The cumulative recall was calculated for each point by taking the ratio of all the times larger or equal sized facilities were found and missed summed together.

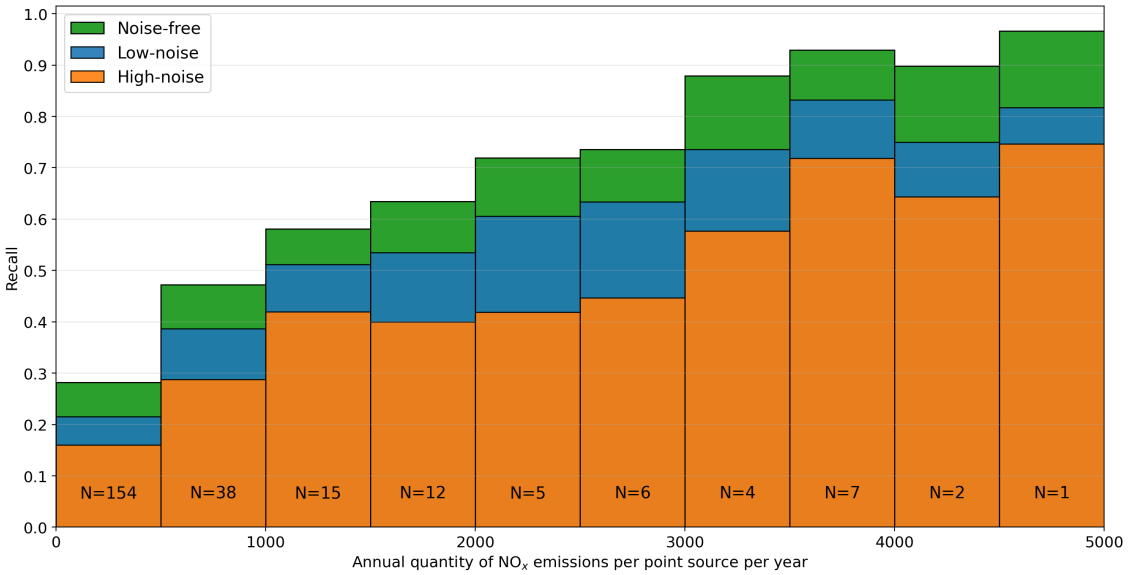

Figure 16: Box plot of overall recall in different point source size ranges with different noise levels found by SCEA. In the bottom of each column, there is the amount of point sources in each size range, denoted with N.

	Noise free	Low noise	High noise
Expected recall with point source size at and above of			
1000 tonnes	0.758	0.660	0.548
500 tonnes	0.648	0.556	0.449
Expected size of the point source at			
0.7 recall	821 tonnes	1513 tonnes	2659 tonnes
0.5 recall	131 tonnes	307 tonnes	859 tonnes

Table 1: Some highlighted numbers from the results, which can also be seen in Figure 15. The first part of the table shows the expected performance when trying to find point sources of a certain size and above. The second part of the table shows the expected size of point sources when targeting a certain performance.

Looking at the point sources and how often they were found by SCEA in Figure 15, there seems to be a lot of variance within each point source size range. This might happen because a smaller point source might be flagged as detected often if there are larger point sources nearby. Moreover, the point sources might stand within an area source, making the plume signal much bigger, even though the emissions are not coming from that point source exactly. In addition, there are temporal coefficients applied to different types of point sources in the SMARTCARB data set, which might make some types of point sources more detectable at certain times of day, week, or year. When plume detection is applied to real-world satellite data, the variance that emerged in this study suggests that it might not be feasible to draw definitive conclusions when looking at individual overpasses.

Although the variance is high when looking at individual point sources, the overall trends seem to be clear when looking at the moving averages and cumulative recall in Figure 15 and the box plot of Figure 16. The accuracy grows fairly consistently when looking for larger point sources in terms of emissions. Unfortunately, for detecting point sources, most of them are in the lower ends of the emission spectrum, making most of the point sources difficult to detect.

The effect of noise seems also to be very consistent across the different noise scenarios. Compared to the noise-free scenario, low noise drops the overall recall down 0.076 and high noise drops the recall 0.153. The noise-free scenario was also cloud-free unlike the low-noise and high-noise scenarios, which had the effects of clouds built in. This means that in the noise-free scenario, there are more occurrences of the points sources. The combined number of times those point sources were present in the noise-free scenario was 173 681, in contrast to the low- and high-noise scenarios where there were only 51 302 and 51 910 occurrences, respectively. However, the same shape is present for all the plots in different noise scenarios in Figure 15, meaning that the lower number of occurrences still caught the underlying trends.

Looking at the results from the perspective of the plumes proved to be challenging as was expected based on the discussions in Section 4.1. The results are laid out in Table 2. The SCEA algorithm found in total 15262, 12573 and 23798 plumes from the high-noise, low-noise, and noise-free scenarios, respectively. Note that the noise-free scenario was also cloud-free, thus having much more data available for finding plumes. A plume was considered as a true positive if there was any point source within 5 km from the plume points. The results are divided into two different sections, one in which plumes with missing pixels around are dismissed and the other in which they are not.

Of all the detected plumes, a precision of 0.942, 0.638 and 0.472 for noise-free, low-noise, and high-noise scenarios, respectively. These numbers might not reflect the real results since, especially with cloudy retrievals, there are a lot of missing pixels, which can make point sources emit signal from areas where there are no data points at all, thus they cannot be considered detected with these methods, like in Figure 4.1. That is why it is more sensible to calculate the metrics on plumes without any missing pixels around them, which are shown bottom in Table 2. The effect of missing pixels is highlighted by the fact that in the noise-free scenario, where there were no missing pixels within the data, a much smaller difference is seen in the precision of the filtered plumes and all the plumes. Moreover, the drop from noise-free to low-noise is massive in the all plumes scenario, compared to the much less extreme drop with the filtered plumes.

It can be hypothesised that more genuine true detections are removed due to the missing

	Noise free	Low noise	High noise
All plumes detected			
True positives	22413	8016	7199
False positives	1385	4557	8063
Precision	0.942	0.638	0.472
Only plumes without any missing values around			
True positives	11594	1332	1529
False positives	308	176	783
Precision	**0.974**	**0.883**	**0.661**

Table 2: Number of times a plume, found by SCEA, was co-located with a point source in different noise scenarios. The upper table features the results if all the plumes are considered. The bottom table is the more reasonable version where only plumes which do not have any missing values around it are considered. Note that the noise-free scenario is also cloud free, making it have much more available data.

pixels compared to the false positives, since often it seems like false positive plumes are much smaller, thus having a lower chance of having missing pixels around them. If we remove all the false positive plumes with missing values around them, we get precisions of 0.986, 0.979 and 0.902 for noise-free, low-noise, and high-noise scenarios, respectively, but these numbers are inherently biased toward the upside, due to the unfair filtering of plumes, thus not reported in the Table 2. However, they are interesting to compare.

The effect of noise seems to affect the SCEA algorithm noticeably. The performance drops a considerable amount when turning up the noise. Since real-world satellite data can be very noisy at times, the results suggest that the SCEA algorithm might not be able to capture small point sources accurately enough with real-world satellite data. More work needs to be done to make the SCEA algorithm more robust to noise. When comparing the numbers of true positives, the low-noise and high-noise scenarios have a similar magnitude, but when comparing the false positives, the high-noise scenario seems to find much more unique plumes, from the noise assumably. For all the noise scenarios, the same sensitivity parameters were used, but the results suggest that it might be reasonable to turn down the sensitivity with more noisy data.

5 Conclusions and discussion

In this thesis, a detailed overview of existing plume detection methods and the theory behind them was conducted. Moreover, a new unsupervised plume detection method, SCEA, was introduced along with a case study in which the SCEA algorithm was applied to the SMARTCARB data set. In the study, it was examined if plumes detected by SCEA were co-located with known point sources in the SMARTCARB data set. In different noise scenarios, the method reached a precision of 0.974, 0.883, and 0.661 in noise-free, low-noise, and high-noise scenarios, respectively. The recall for point sources with annual emissions of more than 1000 tonnes was 0.758, 0.660, and 0.548 for noise-free, low-noise, and high-noise scenarios, respectively. For point sources with annual emissions of more than 500 tonnes, the recall reached 0.648, 0.556, and 0.449 for different noise scenarios, respectively.

These results indicate that when it comes to finding point sources in satellite retrievals, noise affects the SCEA algorithm considerably. The smaller point sources were really difficult to get consistently covered. The comparison between precision and recall in the results suggests that more experiments should be done with the algorithm parameters. The precision was noticeably higher in all noise scenarios, indicating that there could be room to increase the sensitivity of the method. A potential future study could be on the sensitivity of the parameters of SCEA and how they affect the results. The lack of computational resources and the timeline for the thesis did not allow for a more thorough investigation of the effect of algorithm parameters. These results also suggest that experiments with CO_2 retrievals might not yield any appealing results due to the much noisier and more chaotic nature of atmospheric CO_2.

The challenges in evaluating plume detection methods were a major difficulty in this study. Therefore, the evaluation was performed with simulated data. However, there may still be much room for improvement. All details about the simulations were not known in this study, so with a better understanding of all the factors within the simulation that affect the plumes, a more objective study could be conducted. With perfect knowledge, one could determine the amount of emissions from certain point sources and area sources for every single pixel, or additionally, the amount of background signal and noise. The ability to outline the plumes could also be measured. There is potential for future collaboration with the SMARTCARB authors.

The way in which a detected plume and a potential source were co-located in the study could be refined. The SCEA algorithm only outputs the points in an area where there are higher values, but it does not try to find the point source. In the study, a point source was co-located with the plume if it was anywhere within this area. Future work could involve determining a potential location for the point source, for example, assigning it to the maximum value within the plume. In this way, the plume and the point source could be connected more realistically. However, challenges will occur in situations where multiple point sources combine into a larger plume.

In addition to the study with simulated data, one potential future study could also be conducted with real-world satellite data, comparing the results to, for example, the global point source inventory ODIAC [66], as in [38], where a plume detection method was studied using the help of the ODIAC inventory. However, one must be careful when using such point source inventories for the evaluation of plume detection methods, so as to avoid

the potential pitfalls highlighted in Section 4.1 and in Section 2.2, where a review of [38] was provided.

There is also a lot of potential to improve the SCEA algorithm for plume detection. Currently, the SCEA method picks staring points for the clusters in a "greedy" way choosing the highest value not yet clustered. A more sophisticated way to find the starting points could be to determine local maxima from the two-dimensional data. A positive consequence from determining starting points that way would be that one point could be part of multiple clusters, since there is no need to remove the already clustered points.

The SCEA algorithm currently includes points with lower values, perhaps too easily in the cluster at the edges of high-valued areas. Clustering stops when the radiating points do not add more points to the cluster, but the points that fail to add any new points are still included in the cluster. That is why the clusters formed by SCEA might be too bloated in some cases. This problem could be tackled by excluding the radiating points from the cluster, which do not add any new point. The radii that get drawn around the points could also be influenced by nearby points' values, with higher values having a greater change of getting added.

If we want to optimise the algorithm for plume detection, one idea could be to take wind data into account in the process of the algorithm. Since plumes move with the wind, it could be useful to have the cluster expand more easily in the direction of the wind. It could be also beneficial to do some sort of prepossessing to the data, such as attempts to remove background or trying to de-noise the data with, for example, a Gaussian filter. Alternatively, if we want to optimise for point source detection, the temporal axis could be taken into account. If the SCEA algorithm finds persistent plumes at a certain location, there is likely a point source there.

Other potential future work could include adding a second step to the plume detection method. First, the SCEA algorithms finds clusters in an unsupervised way, and then as a second step, classify the plumes as true or false with, for example, a supervised machine learning method. With additional features such as retrieval variables, wind data, ground topography or satellite images, the second method could be able to improve the results and retrieve additional knowledge that an unsupervised method could not. This idea is partly inspired from the work in [39], discussed in Section 2.2, where a two-step plume detection method was introduced, with first a neural network finding plume-like structures and as a second step, a support vector machine classifying the plume as real, empty, or artefact. As another potential second step, the plumes could be further classified as a point source plume or an area source plume. Moreover, another method could identify the exact location of the point source with a second method given the potential plume from the SCEA algorithm.

The SCEA algorithm could also be used as a tool to find training data for a supervised plume detection method such as a convolutional neural network. With SCEA one could find a more diverse set of plume images than, for example, only sampling some set of known locations of point sources. Moreover, SCEA could find highly valuable scenes where there seems to be a plume, but with closer inspection there is not. When training a supervised model, the need for versatile well-representing data is huge so these edge cases are immensely valuable data to make supervised machine learning techniques robust. A form of weakly supervised pre-training for a neural network could also be done where the plumes found by SCEA are labelled as true plumes without human inspection, so the

neural network has a good starting point with the model weights, and is later trained with more confident labels.

It would be interesting to apply the SCEA algorithm to other problems than satellite retrievals. The SCEA could excel in applications, where there is need to find high-valued ares in massive amounts of data with irregular sampling. For example, finding contiguous high population density areas with only postal code level data. This kind of problem could be useful, for example, in finding suitable places for stores and services.

Research on plume detection from satellite data is still very young. A big chunk of the work on this problem has only come out very recently. With the growing amount of satellite missions, and consequently, the growing amount of satellite data, more research on plume detection is bound to come. The community is especially anticipating the CO2M satellite mission, set to launch in 2026, which is designed to enable basis for point source detection. There is still a lot of time to develop and refine the necessary tools for robust plume detection methods for the future greenhouse gas satellite missions.

References

[1] Martin Ester et al. 'A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise'. In: *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*. KDD'96. Portland, Oregon: AAAI Press, 1996, pp. 226–231.

[2] G. Kuhlmann et al. *SMARTCARB – Use of Satellite Measurements of Auxiliary Reactive Trace Gases for Fossil Fuel Carbon Dioxide Emission Estimation, Final report of ESA study contract n°4000119599/16/NL/FF/mg, project led by Empa (Switzerland)*. 2018. URL: https://www.empa.ch/documents/56101/617885/FR_Smartcarb_final_Jan2019.pdf.

[3] IPCC. *Climate Change 2023: Synthesis Report. Contribution of Working Groups I, II and III to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change*. [Core Writing Team, Lee, H. and Romero, J.] Geneva, Switzerland: IPCC, 2023, pp. 1–34. DOI: 10.59327/IPCC/AR6-9789291691647.001.

[4] United Nations Framework Convention on Climate Change. *Paris Agreement*. UNTC XXVII 7.d. 12th Dec. 2015. URL: https://treaties.un.org/pages/ViewDetails.aspx?src=TREATY&mtdsg_no=XXVII-7-d&chapter=27&clang=_en (visited on 06/12/2023).

[5] UNFCCC Secretariat. *Technical dialogue of the first global stocktake. Synthesis report by the co-facilitators on the technical dialogue*. Tech. rep. UN Climate Change Conference - United Arab Emirates Nov/Dec 2023. UNFCCC, Sept. 2023.

[6] G. Janssens-Maenhout et al. 'Toward an Operational Anthropogenic CO2 Emissions Monitoring and Verification Support Capacity'. In: *Bulletin of the American Meteorological Society* 101 (8 Sept. 2020), E1439–E1451. ISSN: 0003-0007. DOI: 10.1175/BAMS-D-19-0017.1. URL: https://journals.ametsoc.org/view/journals/bams/101/8/bamsD190017.xml.

[7] "valentin". *E-PRTR: 4 reasons why we love and hate the E-PRTR database*. CaptureMap. 21st Sept. 2022. URL: https://www.capturemap.no/four-reasons-why-we-love-and-hate-the-e-prtr-database/ (visited on 17/08/2023), archived at https://web.archive.org/web/20230817095750/https://www.capturemap.no/four-reasons-why-we-love-and-hate-the-e-prtr-database/.

[8] Debra Wunch et al. *Documentation for the 2014 TCCON Data Release*. Oct. 2015. DOI: 10.14291/TCCON.GGG2014.DOCUMENTATION.R0/1221662.

[9] Debra Wunch et al. 'Comparisons of the Orbiting Carbon Observatory-2 (OCO-2) XCO2 measurements with TCCON'. In: *Atmospheric Measurement Techniques* 10 (6 June 2017), pp. 2209–2238. ISSN: 18678548. DOI: 10.5194/AMT-10-2209-2017.

[10] *ADEOS (Advanced Earth Observing Satellite) / Midori*. Satellite Missions Catalogue, eoPortal. European Space Agency. URL: https://www.eoportal.org/satellite-missions/adeos#eop-quick-facts-section (visited on 07/12/2023), archived at https://web.archive.org/web/20231207090843/https://www.eoportal.org/satellite-missions/adeos#eop-quick-facts-section.

[11] *EnviSat (Environmental Satellite)*. Satellite Missions Catalogue, eoPortal. European Space Agency. URL: https://www.eoportal.org/satellite-missions/envisat#spacecraft (visited on 07/12/2023), archived at https://web.archive.org/web/20231207104504/https://www.eoportal.org/satellite-missions/envisat#spacecraft.

[12] Daniel J Jacob et al. 'Quantifying methane emissions from the global scale down to point sources using satellite observations of atmospheric methane'. In: *Atmos. Chem. Phys* 22 (2022), pp. 9617–9646. DOI: 10.5194/acp-22-9617-2022. URL: https://doi.org/10.5194/acp-22-9617-2022.

[13] Sean Crowell. 'The Next Great Challenge and Opportunity: The Global Constellation of Greenhouse Gas Observing Satellites'. In: *IWGGMS-19 Conference, Poster Session* (July 2023). URL: https://cdn.gather.town/storage.googleapis.com/gather-town.appspot.com/uploads/83XKkBDs1wCjtP6p//RJsvwT5OtwhH9OiAXahWaq, archived at https://web.archive.org/web/20231207110305/https://cdn.gather.town/storage.googleapis.com/gather-town.appspot.com/uploads/83XKkBDs1wCjtP6p/RJsvwT5OtwhH9OiAXahWaq.

[14] Henk Eskes et al. *Sentinel-5 precursor/TROPOMI Level 2 Product User Manual Nitrogendioxide*. Version 4.1.0. 11th July 2022. URL: https://sentinel.esa.int/documents/247904/2474726/Sentinel-5P-Level-2-Product-User-Manual-Nitrogen-Dioxide.pdf.

[15] *About Copernicus*. Copernicus. URL: https://www.copernicus.eu/en/about-copernicus (visited on 19/06/2023), archived at http://web.archive.org/web/20230619135927/https://www.copernicus.eu/en/about-copernicus.

[16] *ESA - The Sentinel missions*. European Space Agency. URL: https://www.esa.int/Applications/Observing_the_Earth/Copernicus/The_Sentinel_missions (visited on 19/06/2023), archived at http://web.archive.org/web/20230619140542/https://www.esa.int/Applications/Observing_the_Earth/Copernicus/The_Sentinel_missions.

[17] *ESA - Copernicus Sentinel Expansion missions*. European Space Agency. URL: https://www.esa.int/Applications/Observing_the_Earth/Copernicus/Copernicus_Sentinel_Expansion_missions (visited on 19/06/2023), archived at https://web.archive.org/web/20230619140754/https://www.esa.int/Applications/Observing_the_Earth/Copernicus/Copernicus_Sentinel_Expansion_missions.

[18] *ESA - Full steam ahead for carbon dioxide monitoring mission*. European Space Agency. 23rd May 2022. URL: https://www.esa.int/Applications/Observing_the_Earth/Copernicus/Full_steam_ahead_for_carbon_dioxide_monitoring_mission (visited on 19/06/2023), archived at https://web.archive.org/web/20230619140919/https://www.esa.int/Applications/Observing_the_Earth/Copernicus/Full_steam_ahead_for_carbon_dioxide_monitoring_mission.

[19] Bernd Sierk et al. 'The Copernicus CO2M mission for monitoring anthropogenic carbon dioxide emissions from space'. In: *INTERNATIONAL CONFERENCE ON SPACE OPTICS — ICSO 2021* 11852 (11 June 2021), pp. 1563–1580. ISSN: 1996756X. DOI: 10.1117/12.2599613. URL: https://www.spiedigitallibrary.org/conference-proceedings-of-spie/11852/118523M/The-Copernicus-CO2M-mission-for-monitoring-anthropogenic-carbon-dioxide-emissions/10.1117/12.2599613.full.

[20] Tim Hill and Ray Nassar. 'Pixel Size and Revisit Rate Requirements for Monitoring Power Plant CO_2 Emissions from Space'. In: *Remote Sensing 2019, Vol. 11, Page 1608* 11 (13 July 2019), p. 1608. ISSN: 2072-4292. DOI: 10.3390/RS11131608. URL: https://www.mdpi.com/2072-4292/11/13/1608/htm%20https://www.mdpi.com/2072-4292/11/13/1608.

[21] Yasjka Meijer et al. *Copernicus CO_2 Monitoring Mission Requirements Document*. Version 3.0. Earth and Mission Science Division, European Space Agency, 1st Oct. 2020. URL: https://esamultimedia.esa.int/docs/EarthObservation/CO2M_MRD_v3.0_20201001_Issued.pdf.

[22] *CEOS EO Handbook – Mission Summary - Sentinel CO2M-C*. CEOS. URL: https://database.eohandbook.com/database/missionsummary.aspx?missionID=1038 (visited on 19/06/2023), archived at https://web.archive.org/web/20230619141017/https://database.eohandbook.com/database/missionsummary.aspx?missionID=1038.

[23] David Archer et al. 'Atmospheric lifetime of fossil fuel carbon dioxide'. In: *Annual Review of Earth and Planetary Sciences* 37 (May 2009), pp. 117–134. ISSN: 00846597. DOI: 10.1146/ANNUREV.EARTH.031208.100206.

[24] J.G. Canadell et al. 'Global Carbon and Other Biogeochemical Cycles and Feedbacks'. In: *Climate Change 2021: The Physical Science Basis. Contribution of Working Group I to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change*. Ed. by Masson-Delmotte et al. Cambridge University Press, 2021. Chap. 5, pp. 673–816. DOI: 10.1017/9781009157896.007.

[25] Tamaki Fujinawa et al. 'First Concurrent Observations of NO2 and CO2 From Power Plant Plumes by Airborne Remote Sensing'. In: *Geophysical Research Letters* 48 (14 July 2021), e2021GL092685. ISSN: 1944-8007. DOI: 10.1029/2021GL092685. URL: https://onlinelibrary.wiley.com/doi/full/10.1029/2021GL092685%20https://onlinelibrary.wiley.com/doi/abs/10.1029/2021GL092685%20https://agupubs.onlinelibrary.wiley.com/doi/10.1029/2021GL092685.

[26] Fei Liu et al. 'A methodology to constrain carbon dioxide emissions from coal-fired power plants using satellite observations of co-emitted nitrogen dioxide'. In: *Atmospheric Chemistry and Physics* 20 (1 Jan. 2020), pp. 99–116. ISSN: 16807324. DOI: 10.5194/ACP-20-99-2020.

[27] D. Crisp. 'Measuring atmospheric carbon dioxide from space with the Orbiting Carbon Observatory-2 (OCO-2)'. In: ed. by James J. Butler, Xiaoxiong (Jack) Xiong and Xingfa Gu. Sept. 2015, p. 960702. DOI: 10.1117/12.2187291.

[28] Gerrit Kuhlmann et al. 'Detectability of CO_2 emission plumes of cities and power plants with the Copernicus Anthropogenic CO_2 Monitoring (CO2M) mission'. In: *Atmospheric Measurement Techniques Discussions* (June 2019), pp. 1–35. DOI: 10.5194/amt-2019-180.

[29] Maximilian Reuter et al. 'Towards monitoring localized CO2 emissions from space: Co-located regional CO2 and NO2 enhancements observed by the OCO-2 and S5P satellites'. In: *Atmospheric Chemistry and Physics* 19 (14 July 2019), pp. 9371–9383. ISSN: 16807324. DOI: 10.5194/ACP-19-9371-2019.

[30] Janne Hakkarainen et al. 'Analyzing nitrogen oxides to carbon dioxide emission ratios from space: A case study of Matimba Power Station in South Africa'. In: *Atmospheric Environment: X* 10 (Apr. 2021), p. 100110. ISSN: 2590-1621. DOI: 10.1016/J.AEAOA.2021.100110.

[31] Dien Wu et al. 'Space-based quantification of per capita CO2 emissions from cities'. In: *Environmental Research Letters* 15 (3 Feb. 2020), p. 035004. ISSN: 1748-9326. DOI: 10.1088/1748-9326/AB68EB. URL: https://iopscience.iop.org/article/10.1088/1748-9326/ab68eb%20https://iopscience.iop.org/article/10.1088/1748-9326/ab68eb/meta.

[32] Emily G. Yang et al. 'Using Space-Based Observations and Lagrangian Modeling to Evaluate Urban Carbon Dioxide Emissions in the Middle East'. In: *Journal of Geophysical Research: Atmospheres* 125 (7 Apr. 2020), e2019JD031922. ISSN: 2169-8996. DOI: 10.1029/2019JD031922. URL: https://onlinelibrary.wiley.com/doi/full/10.1029/2019JD031922%20https://onlinelibrary.wiley.com/doi/abs/10.1029/2019JD031922%20https://agupubs.onlinelibrary.wiley.com/doi/10.1029/2019JD031922.

[33] Xinxin Ye et al. 'Constraining Fossil Fuel CO2 Emissions From Urban Area Using OCO-2 Observations of Total Column CO2'. In: *Journal of Geophysical Research: Atmospheres* 125 (8 Apr. 2020), e2019JD030528. ISSN: 2169-8996. DOI: 10.1029/2019JD030528. URL: https://onlinelibrary.wiley.com/doi/full/10.1029/2019JD030528%20https://onlinelibrary.wiley.com/doi/abs/10.1029/2019JD030528%20https://agupubs.onlinelibrary.wiley.com/doi/10.1029/2019JD030528.

[34] Bo Zheng et al. 'Observing carbon dioxide emissions over China's cities and industrial areas with the Orbiting Carbon Observatory-2'. In: *Atmospheric Chemistry and Physics* 20 (14 July 2020), pp. 8501–8510. ISSN: 16807324. DOI: 10.5194/ACP-20-8501-2020.

[35] Ray Nassar et al. 'Advances in quantifying power plant CO2 emissions with OCO-2'. In: *Remote Sensing of Environment* 264 (Oct. 2021), p. 112579. ISSN: 0034-4257. DOI: 10.1016/J.RSE.2021.112579.

[36] Ray Nassar et al. 'Tracking CO2 emission reductions from space: A case study at Europe's largest fossil fuel power plant'. In: *Frontiers in Remote Sensing* 3 (Oct. 2022), p. 1028240. ISSN: 2673-6187. DOI: 10.3389/FRSEN.2022.1028240.

[37] Janne Hakkarainen et al. 'D4.4 Benchmarking of plume detection and quantification methods'. Version 1.0. In: (Feb. 2023). CoCO2 project. URL: https://coco2-project.eu/sites/default/files/2023-04/CoCO2-D4-4-V1_2023-02-17.pdf.

[38] Douglas P. Finch, Paul I. Palmer and Tianran Zhang. 'Automated detection of atmospheric NO_2 plumes from satellite data: A tool to help infer anthropogenic combustion emissions'. In: *Atmospheric Measurement Techniques* 15 (Feb. 2022), pp. 721–733. ISSN: 18678548. DOI: 10.5194/AMT-15-721-2022.

[39] B. J. Schuit et al. 'Automated detection and monitoring of methane super-emitters using satellite data'. In: *Atmospheric Chemistry and Physics Discussions* 2023 (2023), pp. 1–47. DOI: 10.5194/acp-2022-862. URL: https://acp.copernicus.org/preprints/acp-2022-862/.

[40] Peter Joyce et al. 'Using a deep neural network to detect methane point sources and quantify emissions from PRISMA hyperspectral satellite images'. In: *Atmos. Meas. Tech* 16 (2023). DOI: 10.5194/amt-16-2627-2023. URL: https://doi.org/10.5194/amt-16-2627-2023.

[41] Manu Goudar et al. 'Plume detection and emission estimate for biomass burning plumes from TROPOMI carbon monoxide observations using APE v1.1'. In: *Geosci. Model Dev* 16 (2023), pp. 4835–4852. DOI: 10.5194/gmd-16-4835-2023. URL: https://doi.org/10.5194/gmd-16-4835-2023.

[42] T. Lauvaux et al. 'Global assessment of oil and gas methane ultra-emitters'. In: *Science* 375 (6580 Feb. 2022). ISSN: 10959203. DOI: 10.1126/SCIENCE.ABJ4351/SUPPL_FILE/SCIENCE.ABJ4351_SM.PDF. URL: https://www.science.org/doi/10.1126/science.abj4351.

[43] Benyamin Hosseiny et al. 'Beyond Supervised Learning in Remote Sensing: A Systematic Review of Deep Learning Approaches'. eng. In: *IEEE journal of selected topics in applied earth observations and remote sensing* 17 (2024), pp. 1035–1052. ISSN: 1939-1404. DOI: 10.1109/JSTARS.2023.3316733.

[44] Francois Chollet et al. *Keras*. 2015. URL: https://github.com/fchollet/keras.

[45] Adam Paszke et al. 'PyTorch: An Imperative Style, High-Performance Deep Learning Library'. In: *Advances in Neural Information Processing Systems 32*. Ed. by H. Wallach et al. Curran Associates, Inc., 2019, pp. 8024–8035. URL: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[46] Martín Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. 2015. URL: https://www.tensorflow.org/.

[47] F. Pedregosa et al. 'Scikit-learn: Machine Learning in Python'. In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[48] Jie Li et al. 'High-spatiotemporal resolution mapping of spatiotemporally continuous atmospheric CO 2 concentrations over the global continent'. In: *International Journal of Applied Earth Observations and Geoinformation* 108 (2022). DOI: 10.1016/j.jag.2022.102743. URL: http://creativecommons.org/licenses/by/4.0/.

[49] Spurthy Maria Pais, Shrutilipi Bhattacharjee and Anand Kumar Madasamy. 'Prediction of High-Resolution Atmospheric CO2 Concentration from OCO-2 using Machine Learning'. In: *ACM International Conference Proceeding Series* (Jan. 2023), pp. 243–247. DOI: 10.1145/3570991.3571062. URL: https://dl.acm.org/doi/10.1145/3570991.3571062.

[50] Joffrey Dumont Le Brazidec et al. 'Segmentation of XCO_2 images with deep learning: application to synthetic plumes from cities and power plants'. In: *Geoscientific Model Development* 16 (13 July 2023), pp. 3997–4016. ISSN: 1991-9603. DOI: 10.5194/GMD-16-3997-2023. URL: https://gmd.copernicus.org/articles/16/3997/2023/.

[51] Xinyue Ai et al. 'Quantification of Central and Eastern China's atmospheric CH 4 enhancement changes and its contributions based on machine learning approach'. In: *Journal of Environmental Sciences* 138 (2024), pp. 236–248. DOI: 10.1016/j.jes.2023.03.010. URL: www.sciencedirect.com.

[52] 'Machine learning-based global air quality index development using remote sensing and ground-based stations'. In: *Environmental Advances* 15 (2024), p. 100456. DOI: 10.1016/j.envadv.2023.100456. URL: https://doi.org/10.1016/j.envadv.2023.100456.

[53] Joffrey Dumont Le Brazidec et al. 'Deep learning applied to CO_2 power plant emissions quantification using simulated satellite images'. In: (). DOI: 10.5194/gmd-2023-142. URL: https://doi.org/10.5194/gmd-2023-142.

[54] 'MethaNet – An AI-driven approach to quantifying methane point-source emission from high-resolution 2-D plume imagery'. In: *Remote Sensing of Environment* 269 (Feb. 2022), p. 112809. ISSN: 0034-4257. DOI: 10.1016/J.RSE.2021.112809.

[55] Lixiang Liu, Ming Xu and Renqiang Li. 'Modeling temporal patterns of methane effluxes using multiple regression and random forest in Poyang Lake, China'. In: *Wetlands Ecology and Management* 26 (1 Feb. 2018), pp. 103–117. ISSN: 15729834. DOI: https://doi.org/10.1007/s11273-017-9558-7. URL: https://link.springer.com/article/10.1007/s11273-017-9558-7.

[56] Tao Zheng, Ray Nassar and Martin Baxter. 'Estimating power plant CO2 emission using OCO-2 XCO2 and high resolution WRF-Chem simulations'. In: *Environmental Research Letters* 14 (8 July 2019), p. 085001. ISSN: 1748-9326. DOI: 10.1088/1748-9326/AB25AE. URL: https://iopscience.iop.org/article/10.1088/1748-9326/ab25ae%20https://iopscience.iop.org/article/10.1088/1748-9326/ab25ae/meta.

[57] Dominik Brunner et al. 'Accounting for the vertical distribution of emissions in atmospheric CO2 simulations'. In: *Atmospheric Chemistry and Physics* 19 (7 Apr. 2019), pp. 4541–4559. ISSN: 16807324. DOI: 10.5194/ACP-19-4541-2019.

[58] Yilong Wang et al. 'PMIF v1.0: Assessing the potential of satellite observations to constrain CO2 emissions from large cities and point sources over the globe using synthetic data'. In: *Geoscientific Model Development* 13 (11 Nov. 2020), pp. 5813–5831. ISSN: 19919603. DOI: 10.5194/GMD-13-5813-2020.

[59] Johan Strandgren et al. 'Towards spaceborne monitoring of localized CO2 emissions: An instrument concept and first performance assessment'. In: *Atmospheric Measurement Techniques* 13 (6 June 2020), pp. 2887–2904. ISSN: 18678548. DOI: 10.5194/AMT-13-2887-2020.

[60] Fabio Crameri. *Scientific colour maps*. Version 8.0.0. Zenodo, June 2023. DOI: 10.5281/zenodo.8035877.

[61] Zhan Zhang et al. 'Detecting and quantifying methane emissions from oil and gas production: algorithm development with ground-truth calibration based on Sentinel-2 satellite imagery'. In: *Atmospheric Measurement Techniques* 15 (23 Dec. 2022), pp. 7155–7169. ISSN: 18678548. DOI: 10.5194/AMT-15-7155-2022.

[62] Joffrey Dumont Le Brazidec et al. 'Segmentation of XCO2 images with deep learning: application to synthetic plumes from cities and power plants'. In: *Geoscientific Model Development* 16 (13 July 2023), pp. 3997–4016. ISSN: 19919603. DOI: 10.5194/GMD-16-3997-2023.

[63] Elena Sánchez-García et al. 'Mapping methane plumes at very high spatial resolution with the WorldView-3 satellite'. In: *Atmospheric Measurement Techniques* 15 (6 Mar. 2022), pp. 1657–1674. ISSN: 18678548. DOI: 10.5194/AMT-15-1657-2022.

[64] European Space Agency. *Copernicus Sentinel-5P (processed by ESA), Sentinel-5 precursor/TROPOMI Level 2 Product User Manual Nitrogendioxide*. Version 02. 2021. DOI: https://doi.org/10.5270/S5P-9bnp8q8.

[65] Giglio. L. et al. *MODIS/Terra+Aqua Burned Area Monthly L3 Global 500m SIN Grid V061*. Version 061. NASA EOSDIS Land Processes Distributed Active Archive Center, 2021. DOI: 10.5067/MODIS/MCD64A1.061.

[66] Tomohiro Oda and Shamil Maksyutov. *ODIAC Fossil Fuel CO_2 Emissions Dataset*. Center for Global Environmental Research, National Institute for Environmental Studies, 19th Aug. 2015. DOI: 10.17595/20170411.001.

[67] Ramprasaath R. Selvaraju et al. 'Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization'. In: *International Journal of Computer Vision* 128 (2 Oct. 2016), pp. 336–359. DOI: 10.1007/s11263-019-01228-7. URL: http://arxiv.org/abs/1610.02391%20http://dx.doi.org/10.1007/s11263-019-01228-7.

[68] Bolei Zhou et al. 'Object Detectors Emerge in Deep Scene CNNs'. In: *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings* (Dec. 2014). URL: https://arxiv.org/abs/1412.6856v2.

[69] Bolei Zhou et al. 'Learning Deep Features for Discriminative Localization'. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 2016-December (Dec. 2015), pp. 2921–2929. ISSN: 10636919. DOI: 10.1109/CVPR.2016.319. URL: https://arxiv.org/abs/1512.04150v1.

[70] Joannes D. Maasakkers et al. 'Using satellites to uncover large methane emissions from landfills'. In: *Science Advances* 8 (32 Aug. 2022), p. 9683. ISSN: 23752548. DOI: 10.1126/SCIADV.ABN9683/SUPPL_FILE/SCIADV.ABN9683_SM.PDF. URL: https://www.science.org/doi/10.1126/sciadv.abn9683.

[71] Olaf Ronneberger, Philipp Fischer and Thomas Brox. 'U-net: Convolutional networks for biomedical image segmentation'. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 9351 (2015), pp. 234–241. ISSN: 16113349. DOI: 10.1007/978-3-319-24574-4_28/COVER. URL: https://link.springer.com/chapter/10.1007/978-3-319-24574-4_28.

[72] Kaiming He et al. 'Deep residual learning for image recognition'. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 2016-December (Dec. 2016), pp. 770–778. ISSN: 10636919. DOI: 10.1109/CVPR.2016.90.

[73] Claudio Galeazzi et al. 'The prisma program'. In: *International Geoscience and Remote Sensing Symposium (IGARSS)* 4 (1 2008). DOI: 10.1109/IGARSS.2008.4779667.

[74] Irwin Sobel and Gary Feldman. *An Isotropic 3x3 Image Gradient Operator*. Aug. 2015. DOI: 10.13140/RG.2.1.1912.4965. URL: https://www.researchgate.net/publication/281104656_An_Isotropic_3x3_Image_Gradient_Operator.

[75] Christophe Fiorio and Jens Gustedt. 'Two linear time Union-Find strategies for image processing'. In: *Theoretical Computer Science* 154 (2 Feb. 1996), pp. 165–181. ISSN: 0304-3975. DOI: 10.1016/0304-3975(94)00262-2.

[76] Richard Beare. 'A locally constrained watershed transform'. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28 (7 July 2006), pp. 1063–1074. ISSN: 01628828. DOI: 10.1109/TPAMI.2006.132.

[77] A. Buades, B. Coll and J. M. Morel. 'A Review of Image Denoising Algorithms, with a New One'. In: *https://doi.org/10.1137/040616024* 4 (2 July 2006), pp. 490–530. ISSN: 15403459. DOI: 10.1137/040616024. URL: https://epubs.siam.org/doi/10.1137/040616024.

[78] A. F. Stein et al. 'NOAA's HYSPLIT Atmospheric Transport and Dispersion Modeling System'. In: *Bulletin of the American Meteorological Society* 96 (12 Dec. 2015), pp. 2059–2077. ISSN: 0003-0007. DOI: 10.1175/BAMS-D-14-00110.1. URL: https://journals.ametsoc.org/view/journals/bams/96/12/bams-d-14-00110.1.xml.

[79] Erich Schubert et al. 'DBSCAN Revisited, Revisited: Why and How You Should (Still) Use DBSCAN'. In: *ACM Transactions on Database Systems* 42 (3 July 2017). DOI: 10.1145/3068335. URL: https://doi.org/10.1145/3068335.

[80] Jussi Väisälä. *Topologia 1*. Limes ry, 2007.

[81] H. Moritz. 'Geodetic reference system 1980'. In: *Bulletin Géodésique* 54 (3 Sept. 1980), pp. 395–405. ISSN: 00074632. DOI: 10.1007/BF02521480/METRICS. URL: https://link.springer.com/article/10.1007/BF02521480.

[82] *European Pollutant Release and Transfer Register (E-PRTR) data set*. Version 18. European Environment Agency. Feb. 2020. URL: https://www.eea.europa.eu/data-and-maps/data/member-states-reporting-art-7-under-the-european-pollutant-release-and-transfer-register-e-prtr-regulation-23/european-pollutant-release-and-transfer-register-e-prtr-data-base.

[83] Chris Mooney et al. *Climate pledges built on flawed emissions data, Post investigation finds*. The Washington Post. 7th Nov. 2021. URL: https://www.washingtonpost.com/climate-environment/interactive/2021/greenhouse-gas-emissions-pledges-data/ (visited on 17/08/2023), archived at https://web.archive.org/web/20230817103315/https://www.washingtonpost.com/climate-environment/interactive/2021/greenhouse-gas-emissions-pledges-data/.

[84] Clean Energy Group and Strategen. *The Peaker Problem. An Overview of Peaker Power Plant Facts and Impacts in Boston, Philadelphia, and Detroit*. July 2022.

[85] Sydney A. Jensen et al. 'Seasonal variability of CO2, CH4, and N2O content and fluxes in small agricultural reservoirs of the northern Great Plains'. In: *Frontiers in Environmental Science* 10 (Oct. 2022), p. 895531. ISSN: 2296665X. DOI: 10.3389/FENVS.2022.895531/BIBTEX.

[86] Abirlal Metya et al. 'Diurnal and seasonal variability of CO2 and CH4 concentration in a semi-urban environment of western India'. In: *Scientific Reports 2021 11:1* 11 (1 Feb. 2021), pp. 1–13. ISSN: 2045-2322. DOI: 10.1038/s41598-021-82321-1. URL: https://www.nature.com/articles/s41598-021-82321-1.

[87] Irène Xueref-Remy et al. 'Diurnal, synoptic and seasonal variability of atmospheric CO2 in the Paris megacity area'. In: *Atmospheric Chemistry and Physics* 18 (5 Mar. 2018), pp. 3335–3362. ISSN: 16807324. DOI: 10.5194/ACP-18-3335-2018.

[88] Yu Liu, Nicolas Gruber and Dominik Brunner. 'Spatiotemporal patterns of the fossil-fuel CO_2 signal in central Europe: results from a high-resolution atmospheric transport model'. In: *Atmos. Chem. Phys* 17 (2017), pp. 14145–14169. DOI: 10.5194/acp-17-14145-2017. URL: https://doi.org/10.5194/acp-17-14145-2017.

[89] Qixiang Cai et al. 'The impact of human and livestock respiration on CO2 emissions from 14 global cities'. In: *Carbon Balance and Management* 17 (1 Dec. 2022), pp. 1–12. ISSN: 17500680. DOI: 10.1186/S13021-022-00217-7/FIGURES/6. URL: https://cbmjournal.biomedcentral.com/articles/10.1186/s13021-022-00217-7%20http://creativecommons.org/publicdomain/zero/1.0/.

[90] Gerrit Kuhlmann et al. *Synthetic XCO2, CO and NO2 observations for the CO2M and Sentinel-5 satellites*. Zenodo, Oct. 2020. DOI: 10.5281/zenodo.4048228. URL: https://doi.org/10.5281/zenodo.4048228.

[91] *MACC-III Monitoring Atmospheric Composition and Climate 3. Final report*. Feb. 2016. URL: https://atmosphere.copernicus.eu/sites/default/files/repository/MACCIII_FinalReport.pdf.

[92] J. J.P. Kuenen et al. 'TNO-MACC-II emission inventory; A multi-year (2003-2009) consistent high-resolution European emission inventory for air quality modelling'. In: *Atmospheric Chemistry and Physics* 14 (20 Oct. 2014), pp. 10963–10976. ISSN: 16807324. DOI: 10.5194/ACP-14-10963-2014.

[93] *EMEP/CEIP 2023 Present state of emission data*. EMEP, 2023. URL: https://www.ceip.at/webdab-emission-database/reported-emissiondata (visited on 05/12/2023).

[94] Hugo Denier Van Der Gon et al. 'A high resolution European emission data base for the year 2005 Teilbericht zum F&E-Vorhaben "Strategien zur Verminderung der Feinstaubbelastung-PAREST"'. In: (2013). URL: http://fuer-mensch-und-umwelt.de/.

Appendix A Code and data availability

The source code for the SCEA algorithm, the case study with SMARTCARB data, and most of the plots included in this thesis can be found in a GitHub repository here:
https://github.com/ervelae/SCEA_masters_thesis

SMARTCARB data used in this thesis was accessed through the ICOS JupyterHub Collaboration Server:
https://jupyter.icos-cp.eu/hub/login?next=%2Fhub%2F

Sentinel-5p XNO_2 data (processor version 1.3.2) used in this thesis was accessed through the NASA Goddard Earth Sciences (GES) Data and Information Services Center (DISC):
https://disc.gsfc.nasa.gov/datasets/S5P_L2__NO2____HiR_2/summary