Probabilistic Inversion Modelling of Atmospheric Gaseous Emissions

Thomas H. P. Newman,

B.Sc.(Hons.), M.Sci, M.Res

Lancaster University

Submitted for the degree of Doctor of Philosophy at Lancaster University.

September 2025

STOR-i

excellence with impact

Abstract

Greenhouse gas (GHG) emissions are a primary driver of contemporary climate change, contributing to rising global temperatures, increasing frequency of extreme weather events, and widespread ecological disruption. Effective mitigation depends not only on reducing emissions but also on accurately detecting, locating and quantifying them. Reliable source characterisation underpins national climate strategies, industrial compliance, and international agreements aimed at stabilising the global climate system.

This thesis develops probabilistic inversion frameworks that integrate atmospheric transport models, Bayesian inference, and machine learning to improve the estimation of gas emission source characteristics from ground-based measurements.

First, we address limitations of the widely used Gaussian plume model, where dispersion parameters are often fixed via atmospheric stability classes, introducing bias when meteorological classifications are inaccurate. We propose a gradient-based Markov chain Monte Carlo inversion scheme that jointly infers dispersion parameters alongside source location, emission rate, background concentration, and sensor error. Application to both controlled-release data and simulations demonstrates improved accuracy and uncertainty quantification compared to traditional methods.

Second, we tackle the challenge of real-time inversion in obstructed, unsteady-state flow fields, where computational fluid dynamics (CFD) solvers are too expensive for sequential inference. We design deep-learning surrogate models trained on high-fidelity CFD outputs and embed them within particle filters, enabling near-instantaneous Bayesian estimation of time-varying source parameters. Validation on the Chilbolton methane release dataset and complex synthetic environments shows comparable accuracy to full CFD inversion at orders-of-magnitude lower computational cost.

Finally, we synthesise these contributions, explore integration with emerging satellite-based inversion systems, and outline pathways for scaling these methods to regional and global monitoring networks.

Together, these contributions provide physically grounded, computationally efficient tools for GHG monitoring. By advancing both parameter estimation accuracy and operational feasibility, this work supports scalable, uncertainty-aware frameworks for emissions quantification, informing policy, compliance, and mitigation strategies.

Acknowledgements

First and foremost, to Phil, Chris, and Matthew – thank you for the unwavering support you have given me. I have learned so much from each of you. Although most of our interactions took place remotely[1], I will treasure the memories of these past three years and the many fun conversations we shared.

Phil, thank you for your time and encouragements. You were always available (often well beyond reasonable hours) and never hesitated to dedicate time to carefully review my work. Chris, thank you for your kindness and reliability. It meant a lot, especially during crunch times. Matthew, thank you for your resourcefulness, enthusiasm for my research, and especially for navigating all the delightful internal company policies on my behalf. The lessons I've learned from all three of you will stay with me throughout my career, and for that I am deeply grateful.

Many thanks to the STOR-i students for making the office far more than just a workplace, but a real community. I thoroughly enjoyed being part of STOR-i bake-off, the legendary 100-bus crawl and STOR-i football. I would also like to thank the staff for handling all of the admin work (Wendy, Nicky, Kim, Kate, and Keilah), for keeping the computers running smoothly (Dan, Oli, James, and Bashar), and for creating and maintaining such an exceptional programme (Kevin, Jon, Idris, Anna, and Rachel).

[1] Phil in Wales; Chris in England; Matthew in the Netherlands; Thomas in: France, Germany, the Netherlands, England, Switzerland, Scotland, Austria, Spain, and more.

To my family — Maman, Dad, and Lewis — thank you for always being there for me. Throughout this thesis, the thought of eventually coming home to spend time together kept me going. I cherish those heartwarming family gatherings, filled with delicious food and Mediterranean views, admiring Lewis's artwork before briefly (to everyone's great relief) updating you on my work, mentioning Markov chains and making you long for more rosé and louder cicadas.

Lewis, t'as ta propre dédicace. On n'a jamais été aussi proches, et ça me rend tellement heureux. Te voir vivre cette vie d'artiste, avec tous les petits boulots et les galères que ça entraîne, a vraiment ajouté une touche d'aventure et de légèreté à mon quotidien si cadré et prévisible.

Next, I would like to thank my friends – Ilya, Lisa, Ben, Danielle, Connor, Carla, Matthew, TJ, Jack, Chris – et mes amis Théo, Lucas, Walid, Édouard, Pidon, Tho, Jade, Maxime, Arnoult, et Jean. Although I've prioritised my PhD these past three years, during which my mind was constantly preoccupied with maths (yes, that explains those deep thinking frowns), you have all remained a constant in my life, giving me so much more than numbers to care about. Each of you provided me with a sense of renouveau, a breath of fresh air, that kept burnout at bay every time we were together.

And to the Perschthaler family (Anina, Michael, Lili, and Thomas) and Kinga[2]. Ich kann euch gar nicht genug dafür danken, was ihr alles für mich getan habt. Ihr habt mir hier in München eure Türen geöffnet und mich in eure Familie aufgenommen. So konnte ich den größten Teil dieser Dissertation, umgeben von Lachen, Freude und sprachlichem Austausch, verfassen. Dank euch werde ich meine Promotionsjahre nicht nur wegen der Forschung in Erinnerung behalten, sondern auch wegen all der schönen Unternehmungen und Erlebnisse, die sie begleitet haben.

[2] Hellaw!

Finally, to Johanna – thank you for the unconditional love and support you've continuously given me. You were always there to catch me when I pushed myself too far to meet deadlines. Your patience with my frustrations, your humour, and your unshakable optimism make me one incredibly lucky man. One who can only hope to return the same as your own PhD adventure is just about to begin.

Declaration

I declare that the work in this thesis has been done by myself and has not been submitted elsewhere for the award of any other degree.

Thomas H. P. Newman

Contents

List of Figures

List of Tables

List of Abbreviations

ABL	Atmospheric Boundary Layer
AFM	Area Flux Mapper
ASC	Atmospheric Stability Class
BIC	Bayesian Information Criterion
CFD	Computational Fluid Dynamic
CFL	Courant-Friedrichs-Lewy
CNN	Convolutional Neural Network
CPU	Central Processing Unit
FVM	Finite Volume Method
GB	Gigabyte
GHG	Greenhouse Gas
GNN	Graph Neural Network
GPU	Graphics Processing Unit
IID	Independent and Identically Distributed
IME	Integrated Mass Enhancement
IPCC	Intergovernmental Panel on Climate Change
KF	Kalman Filter
LDS	Laser Dispersion Spectrometer
LPDM	Lagrangian Particle Dispersion Model
LSTM	Long Short-Term Memory

MALA	Metropolis-Adjusted Langevin Algorithm
MAPE	Mean Absolute Percentage Error
MCMC	Markov Chain Monte Carlo
MH	Metropolis-Hastings
MLP	Multilayer Perceptron
M-MALA	Manifold Metropolis-Adjusted Langevin Algorithm
OGMP	Oil and Gas Methane Partnership
OU	Ornstein-Uhlenbeck
PDE	Partial Differential Equation
PINN	Physics-Informed Neural Network
PPM	Parts Per Million
PSI	Point Source Imager
RMSE	Root Mean Square Error
SeLU	Scaled Exponential Linear Unit
SIR	Sequential Importance Resampling
SMC	Sequential Monte Carlo

List of Symbols

Note: To enhance clarity and consistency, each variable, parameter, and notation used throughout this thesis is assigned a unique symbol listed here. In some cases, common symbols from the literature were modified to avoid overlap. For example, while the standard notation for convolutional neural network dimensions is $H \times W \times C$, here we use $n_h \times n_w \times n_c$ instead.

(x, y, z)	Spatial coordinates.
$(\dot{x}, \dot{y}, \dot{z})$	Sensor location.
$(\tilde{x}, \tilde{y}, \tilde{z})$	Gas source location.
$\boldsymbol{A}$	Gaussian plume coupling matrix.
A	Cell face's area.
a	Activation value.
α	Proposal acceptance probability function.
b	Activation function bias.
a_H, b_H	Horizontal dispersion coefficients.
a_V, b_V	Vertical dispersion coefficients.
β	Atmospheric background gas concentration.
C	Gas concentration function.
CH_4	Methane.
CO_2	Carbon dioxide.

d	Sensor measurement.
η	Learning rate.
δ_H	Horizontal offset distance.
δ_R	Downwind distance.
δ_V	Vertical offset.
ϵ_{evo}	Latent state evolution error.
ϵ_{obs}	Sensor measurement error.
ϵ_{ss}	MCMC step size.
F	Advective flux.
$\boldsymbol{f}$	Forcing term.
f_c	Advection-diffusion solver.
f_{obs}	Observation model.
f_{state}	State evolution model.
f_v	Navier-Stokes solver.
G	Diffusive flux.
Γ	Convolutional filter.
γ	Wind direction.
K	Gas diffusion coefficient function.
κ	Dependency window start time.
l	Current MCMC iteration.
$\boldsymbol{\mathcal{L}}$	Loss function.
$\boldsymbol{\lambda}_H$	Horizontal wind direction rolling standard deviation.
$\boldsymbol{\lambda}_V$	Vertical wind direction rolling standard deviation.
μ_H	Source aperture half-width.
μ_V	Source aperture half-height.
$\mathbf{n}$	Unit normal vector.
n_{c}	Number of channels.

n_f	Number of convolutional filters.
n_h	Number of pixel per column.
n_nrn	Number of neuron.
n_obs	Number of sensor measurement.
n_par	Number of parameters.
n_prt	Number of particles.
n_refl	Maximum number of gas reflections.
n_sim	Number of CFD simulations.
n_sns	Number of sensors.
n_src	Number of gas source.
n_T	Number of time points.
n_w	Number of pixel per row.
ν	Kinetic viscosity.
Ω	Spatial domain.
ω	Obstacle.
P	Height of atmospheric boundary layer.
$\boldsymbol{p}$	Air pressure field.
$p\left(\cdot\right)$	Probability distribution.
ϕ	Ornstein-Uhlenbeck process value.
$\boldsymbol{\psi}$	Parameters estimated using M-MALA within Gibbs.
$\boldsymbol{Q}$	Preconditioning matrix.
$q\left(\cdot\right)$	Proposal distribution.
ρ	Gas density.
S	Gas emission function.
s	Gas source emission rate.
$\boldsymbol{\Sigma}$	Covariance matrix.
σ_af	Activation function.

σ_H	Horizontal wind sigma.
σ^2_{ou}	Ornstein-Uhlenbeck variance.
σ^2_{sns}	Sensor measurement error variance.
σ_V	Vertical wind sigma.
t	Time.
$\boldsymbol{\Theta}$	Neural network weights and biases.
Θ	Ornstein-Uhlenbeck time correlation.
θ	Latent state or parameter.
u	Wind speed.
V	Cell's volume.
$\boldsymbol{v}$	Flow field.
$\boldsymbol{W}$	Weight matrix.
$\mathbf{W}$	State evolution covariance matrix.
w	Connection's or particle's weight.
$\boldsymbol{\Xi}$	Gaussian plume coupling matrix parameters.
$\boldsymbol{x}_{\text{nn}}$	Neural network input.
$\boldsymbol{y}_{\text{nn}}$	Neural network output.
ζ	Pre-activation value.

Chapter 1

Introduction

This thesis investigates methods for estimating gas emission source location and emission rate by combining atmospheric gas measurements with Bayesian inversion methods for uncertainty quantification. The work addresses several important challenges in this field, focusing on specific problems where methodological advances can deliver significant improvements. Notably, we address atmospheric gas transport model misspecification and develop a framework for estimating time-varying parameters in complex atmospheric environments. While the scope does not attempt to exhaustively resolve all issues related to emission estimation – a task that would far exceed the capacity of a single volume – this thesis concentrates on some key problems for which we provide detailed and practical solutions.

In this chapter, we first motivate the importance of accurately estimating gas emission sources (Section 1.1), then discuss the principal scientific and technical challenges inherent to this task (Section 1.2). Finally, we position the contributions of this thesis within the broader research landscape, outlining how they advance current capabilities in emission source detection, localisation, and quantification (Section 1.3).

1.1 Motivation to atmospheric gas source estimation

Throughout Earth's history, shifts in atmospheric composition have been key drivers of global climate transitions, which in turn have shaped biodiversity and influenced planetary habitability (Royer et al., 2004; Lyons et al., 2014; Lee and Romero, 2023).

The scientific understanding of the greenhouse effect dates back to 1822, when Joseph Fourier first proposed that solar radiation could be trapped by Earth's atmosphere (Fourier, 1822). This concept was further developed three decades later by Eunice Newton Foote, who in her work "Circumstances affecting the heat of the Sun's rays" (Foote, 1856), experimentally demonstrated that carbon dioxide (CO_2) retains heat more effectively than ambient air. Foote hence suggested that increasing atmospheric CO_2 concentrations could lead to a warming climate – a conclusion that has proven prescient.

More than a century later, from 1996 to 2005, the European Project for Ice Coring in Antarctica (EPICA) extracted deep ice cores from Dome C in East Antarctica. These cores, containing air bubbles trapped within ice layers, provided direct samples of atmospheric gas concentrations spanning the last 800,000 years (Augustin et al., 2004) (See Figure 1.1.1). Analysis of these cores revealed a strong correlation between atmospheric CO_2 levels and global climate states: during glacial periods, CO_2 levels dropped to around 180 parts per million (ppm), while interglacial periods saw concentrations rise to approximately 280 ppm (Lüthi et al., 2008). These data provided some of the earliest and clearest empirical confirmation of Foote's hypothesis, reinforcing the central role of CO_2 in regulating Earth's climate.

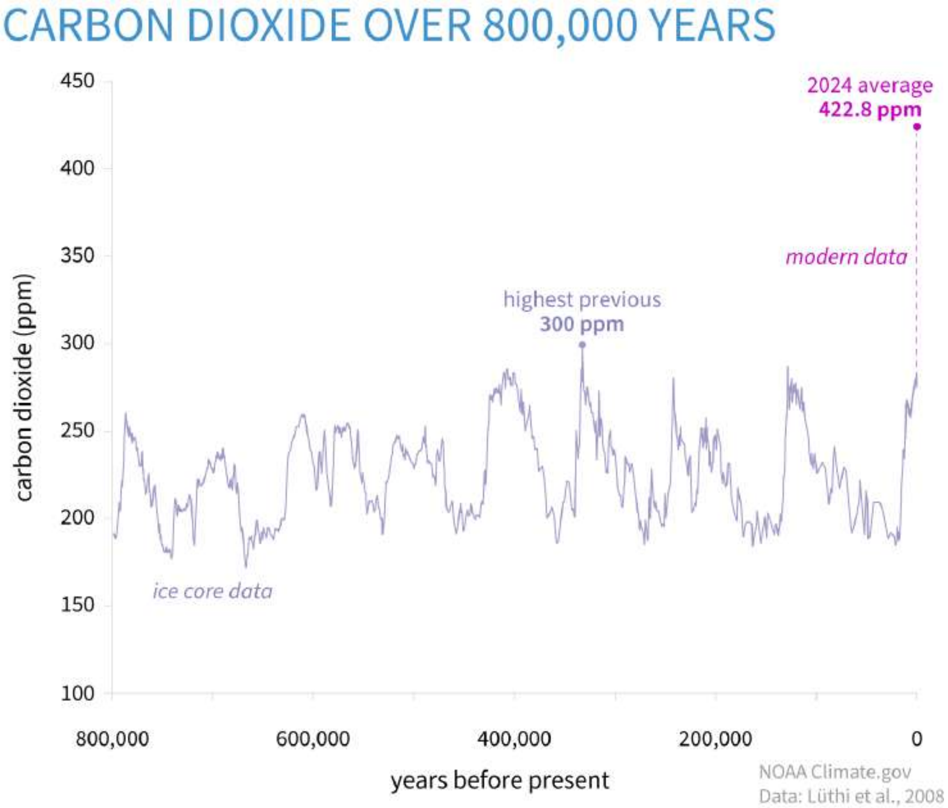


Figure 1.1.1: Atmospheric CO_2 concentrations over the past 800,000 years, measured in parts per million (ppm), are shown from ice-core records (light purple line) alongside the 2024 value (bright magenta dot). The alternating peaks and troughs in the record correspond to glacial periods (low CO_2) and warmer interglacial periods (higher CO_2). During this entire timespan, CO_2 never exceeded 300 ppm (light purple dot, between 300,000 and 400,000 years ago). By contrast, the increase observed over just the past 60 years has occurred about 100 times faster than any natural rise in the record. On the geological time scale, the jump from the end of the last ice age to today's level of 422.8 ppm (dashed magenta line) appears almost instantaneous. Credit: NOAA Climate.gov based on data from Lüthi et al. (2008), via NOAA NCEI Palaeoclimatology Program.

In the past half-century, both atmospheric greenhouse gas (GHG) concentrations and global surface temperatures have risen sharply (see Figure 1.1.2). According to the latest Intergovernmental Panel on Climate Change (IPCC) report (Lee and Romero, 2023), global temperatures have increased more rapidly since the 1970s than during any comparable 50-year period over the last two millennia. The IPCC further concludes (with high confidence) that current atmospheric CO_2 levels are the highest in at least 2 million years, and methane concentrations have reached their highest levels in over

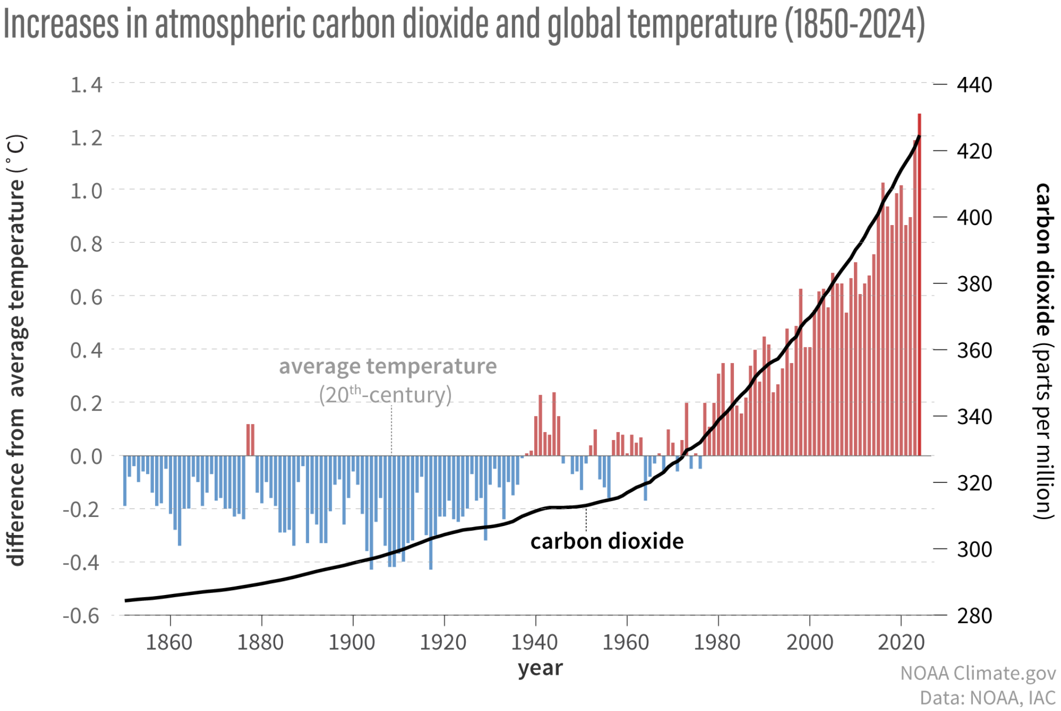


Figure 1.1.2: Bar chart: Annual global temperatures from 1850–2024 shown as differences from the twentieth-century average. Red bars indicate warmer-than-average years, blue bars indicate cooler-than-average years. Line graph: Atmospheric carbon dioxide concentrations from 1850 to 1958 based on Institute for Atmospheric and Climate Science (ETH Zurich) data, and from 1959 to 2023 based on NOAA Global Monitoring Laboratory measurements. Credit: NOAA Global Monitoring Lab. NOAA Climate.gov graph, adapted from original by Dr. Howard Diamond (NOAA ARL).

800,000 years (with very high confidence). These unprecedented levels have coincided with an increasing frequency of environmental disruptions.

Among the most visible biological indicators of climate stress is coral bleaching, particularly on the Great Barrier Reef. Mass bleaching events, driven by elevated sea surface temperatures, frequently lead to long-term degradation of coral ecosystems (Hughes et al., 2017). At the same time, the incidence of extreme weather events has surged: intense precipitation, prolonged droughts, severe heatwaves, cyclones, hurricanes, and devastating wildfires are becoming more common and more destructive (Coumou and Rahmstorf, 2012).

The scientific consensus now is that GHGs are a primary driver of climate change, and that this change is already causing widespread ecological damage. We are approaching irreversible losses and can say (with very high confidence) that mass mortality events are currently being observed on land and in the oceans (Lee and Romero, 2023); Section 2.1.2 paragraph 3.

While Earth's climate has naturally fluctuated over the geological time scale, the rapid and unprecedented increase in GHGs since the Industrial Revolution is fundamentally different in scale and origin. Today, more than 60% of methane emissions (Saunois et al., 2025) and roughly one-third of atmospheric CO_2 (rising from 278 ppm in 1750 to 417 ppm in 2022 (Friedlingstein et al., 2023)) are attributed to human activity. These trends pose a global threat, underscoring the urgent need for collective mitigation and adaptation efforts.

In response to the escalating climate emergency, several major international frameworks have been established to curb GHG emissions. The 2015 Paris Agreement, adopted by 196 parties under the United Nations Framework Convention on Climate Change, sets the long-term goal of limiting global warming to below 2 °C, with efforts to restrict it to 1.5 °C. It requires each country to submit and periodically update Nationally Determined Contributions outlining their mitigation and adaptation targets, and to report transparently on progress through an enhanced global stocktake process. Complementing this broad framework, the Oil and Gas Methane Partnership (OMGP2.0, 2024), launched by the United Nations Environment Programme, provides industry-aligned protocols for measurement-based methane reporting and encourages adoption of best practices for leak detection and repair. Finally, most recently, in a landmark ruling in July 2025, the International Court of Justice affirmed that states have a legal obligation to safeguard the environment from GHG pollution.

Given the scale and urgency of the climate crisis, it is essential to identify effective mitigation strategies. Central to these efforts is the ability to detect, quantify, and monitor GHG emissions with high accuracy and spatial resolution. Before emissions can be reduced, they must first be reliably measured and attributed to their respective sources. This is particularly important because GHG emissions originate from a diverse array of anthropogenic activities, including fossil fuel extraction, agriculture, waste management, transport, and industrial processes. Without a detailed understanding of where emissions occur and how much is being emitted, targeted policy interventions, technological improvements, and compliance mechanisms cannot be effectively designed or enforced. As such, robust emission detection and quantification frameworks are foundational to both national climate strategies and international agreements aimed at reducing global GHG levels.

1.2 Challenges in atmospheric gas source estimation

Estimating atmospheric gas emission sources involves a series of interrelated scientific and practical challenges. This section outlines the most significant of these – which are addressed at various points throughout this thesis – and common solutions for these.

A central consideration in atmospheric gas inversion is the acquisition of measurements that contain sufficient information to infer source location and emission rate. Because the exact source location is initially unknown, its emission rate cannot be measured directly. Instead, concentrations are recorded downwind (or in the relative vicinity) of the source, and the source parameters are inferred from these observations. A fundamental limitation arises here: it is impossible to measure the full three-dimensional, time-varying concentration field of the gas, as an exhaustive sensor coverage is financially and physically unrealistic. This limitation manifests differently depending on the spatial scale of the inversion, which may range from global and regional analyses to local, facility-scale studies. At global scales, inversions typically rely on satellite-based observations that provide broad spatial coverage but relatively coarse spatial and temporal resolution. Regional inversions may combine satellite data with aircraft campaigns or tall-tower measurements, while local inversions often depend on dense networks of ground-based sensors that offer high temporal resolution but limited spatial extent. As a result, available observations inevitably trade spatial coverage against temporal resolution, ranging from datasets with broad spatial coverage to those with fine temporal sampling. The suitability of a given measurement configuration depends on the intended application, spatial scale of interest, and available resources, and these constraints must be accounted for when interpreting inversion results.

Beyond the measurements themselves, gas inversion requires a means of relating atmospheric concentrations to the underlying source parameters. This relationship is described by a "forward model", which is most naturally expressed through the advection–diffusion equation (Stockie, 2011). This partial differential equation depends on the three-dimensional, time-dependent wind field, which cannot be directly measured in its entirety. In practice, wind observations are often limited to single-point measurements from anemometers. Such measurements provide only a crude representation of the full wind field, particularly over large spatial domains, and they do not fully capture turbulent transport processes, which play a key role in atmospheric dispersion. Consequently, forward modelling in gas inversion necessarily involves balancing physical realism against the limitations imposed by available meteorological data.

Even with a physically consistent forward model, inferring the true source parameters remains a significant challenge. The inversion problem is inherently probabilistic: given a forward model, the most plausible source parameters are those that most consistently reproduce the observed concentrations. The parameter space, which may include continuous variables such as source location and emission rate, must be explored efficiently and without bias. Naive approaches such as grid searches are poorly suited to this task, as they scale unfavourably with the number of parameters, introduce discretisation bias, and provide limited information about uncertainty. Effective gas inversion therefore requires parameter estimation strategies that can explore complex parameter spaces while quantifying uncertainty in the inferred solutions.

These challenges – measurement limitations, forward model selection, and efficient parameter estimation – recur throughout the thesis and form the basis for the methodological developments presented.

1.3 Contributions and thesis outline

The focus of this thesis is to estimate gas emission sources' location and emission rate using Bayesian inversion methods to capture estimation uncertainty. The thesis provides a literature review on atmospheric gas transport modelling (Chapter 2) and Bayesian parameter estimation (Chapter 3). Contribution to the field is made through new research (Chapter 4, 5, and 6), cross-pollinating ideas presented in the literature review chapters. Finally, a forward-looking perspective is presented, outlining a potential direction for future research and discussing the broader outlook for this field (Chapter 6). We now provide a brief outline of each chapter:

Chapter 2: Atmospheric Transport of Gas Emissions

This chapter provides a review of existing approaches to model atmospheric gas emission transport. We discuss the fundamental equations on which the field is built and present how these complex partial differential equations are being solved. These include: approximate analytical solutions, numerical solutions, and deep learning surrogate solutions.

Chapter 3: Bayesian Parameter Estimation

This chapter motivates and presents the Bayesian methods used later in this thesis

(Chapter 4 and 5) to estimate emission source location and emission rate. We provide an explanation as to why uncertainty quantification is important and why we chose Bayesian statistics to capture it. We focus on Monte Carlo methods, which allow us to perform Bayesian inversion when the posterior distribution is intractable. Furthermore, we explain how temporally varying parameter values are estimated by employing state-space models.

Chapter 4: Probabilistic Inversion Modelling of Gas Emissions: A Gradient-Based MCMC Estimation of Gaussian Plume Parameters

This chapter is a journal contribution, published in The Annals of Applied Statistics. Co-authors Christopher Nemeth, Matthew Jones and Philip Jonathan. Paper's DOI: 10.1214/25-AOAS2101.

This chapter proposes to improve emission source parameter estimation by addressing model misspecification in the Gaussian plume model. The Gaussian plume model is the most popular approximate analytical solution to atmospheric gas transport. It is parameterised by what are commonly known as “wind sigmas” which traditionally are fixed based on Pasquill's atmospheric stability class. This classification relies on local meteorological conditions, which are often inaccurately measured and fixes wind sigmas according to tables of values drawn from historical experiments. Our methodology removes bias from atmospheric stability class-based Gaussian plumes by simultaneously estimating wind sigmas with emission source location, rate, background gas concentration and sensor measurement error variance. The inversion is performed on a simulation study and a real controlled release dataset using M-MALA-within-Gibbs.

Chapter 5: Deep Learning Surrogates for Real-Time Gas Emission Inversion

This chapter is a journal contribution and has been submitted for publication with co-authors Christopher Nemeth, Matthew Jones and Philip Jonathan. This paper is avail-

able as an arXiv preprint, arXiv:2506.14597.

This chapter presents a methodology to estimate time-varying emission source parameters in obstructed unsteady-state flow fields. Tracking time-varying parameters requires a fast and iterative parameter estimation method. However, obstructed and unsteady-state flow fields cannot be measured directly and must be solved using a numerical solver; this is computationally expensive. We propose a framework for developing neural network-based approximations of numerical solver solutions for predefined monitoring sites. This enables extremely fast forward model predictions, allowing online parameter estimation using particle filters. Our approach is applied to a simulation study and a real dataset, where we achieve a lower computational cost than the Gaussian plume model and equivalent accuracy to the numerical solver.

Chapter 6: Conclusions and Outlook

This chapter marks the conclusion of the thesis. We begin by synthesising its core contributions, emphasising their significance to the field of probabilistic inversion modelling of gas emissions. We then outline a pathway for future research, shaped by the insights gained throughout this work. Finally, we reflect on the broader importance of this field, offering perspectives on its relevance and potential impact in the years ahead.

Chapter 2

Atmospheric Transport of Gas Emissions

Atmospheric transport of gas emissions refers to the physical processes governing how gases move and disperse within the Earth's atmosphere. While the basic concept is easy to grasp, the underlying dynamics are driven by a complex interplay of factors, including wind patterns, temperature, humidity, solar radiation, local topography, physical boundaries, gas composition, and source location and emission rate. These interactions can lead to highly variable and sometimes unpredictable dispersion behaviours.

Accurately modelling atmospheric gas transport is essential for many real-world scenarios, such as setting safety measures for accidental or intentional hazardous gas releases, predicting pollutant dispersion for air quality management, and estimating the emission source location and rate – the latter being the focus of this thesis. At its core, atmospheric gas transport modelling aims to capture the relationship between spatio-temporal gas concentrations and the factors that influence them. Over the years, researchers have developed a variety of modelling approaches that differ in their assumptions, computational demands, and level of physical realism.

This chapter introduces key methods for modelling atmospheric gas transport, spanning from computationally inexpensive, low-fidelity models to more resource-intensive, high-fidelity simulations. Section 2.1 introduces the advection–diffusion and Navier–Stokes equations, which provide the fundamental mathematical framework for describing atmospheric gas movement. Section 2.2 discusses approximate analytical solutions, which offer fast but simplified representations. Section 2.3 examines numerical solutions, which can capture complex flow dynamics with high accuracy at the cost of significant computational effort. Finally, Section 2.4 looks at emerging deep learning surrogate models, which hold promise for delivering efficient and accurate predictions by combining data-driven approaches with physical principles.

2.1 Advection-diffusion and Navier-Stokes partial differential equations

In the 1700s, the contributions of Bernoulli, Euler, D'Alembert, and Lagrange were crucial milestones in the development of fluid mechanics. Bernoulli linked flow and energy conservation in his 1738 *Hydrodynamica* (Bernoulli, 1738); Euler formulated the first general equations for inviscid fluid motion (Euler, 1755); D'Alembert highlighted the limitations of inviscid theory (D'Alembert, 1768); and Lagrange formalised particle-based descriptions and variational methods (Lagrange, 1788). These foundational ideas culminated in Claude-Louis Navier's 1822 work on viscous fluids (Navier, 1823), later refined by George Gabriel Stokes in 1845 (Stokes, 1845), resulting in the Navier–Stokes equations – the cornerstone of modern fluid dynamics.

Meanwhile, in 1822, Fourier published *Théorie analytique de la chaleur* (Fourier, 1822), introducing the famous heat equation, building on the earlier work of Euler, Bernoulli, and Lagrange. This laid the groundwork for Adolf Fick's 1855 formulation of Fick's law of diffusion (Fick, 1855), describing molecular diffusion in fluids. By the late 19th and early 20th centuries, Fick's diffusion framework had been combined with Eulerian advection to yield the advection-diffusion equation:

$$\frac{\partial C}{\partial t} + \nabla \cdot (C\boldsymbol{v}) = \nabla \cdot (K \nabla C) + S. \tag{2.1.1}$$

Equation 2.1.1 describes the transport of a scalar quantity in a three-dimensional space (x, y, z), where the mass concentration or temperature is given by $C(x, y, z, t)$; $S(x, y, z, t)$ represents source emissions; $K(x, y, z)$ denotes the diffusion coefficients (accounting for eddy and molecular diffusion); and $\boldsymbol{v}(x, y, z, t)$ is the flow field. The diffusion coefficients represent an effective diffusivity that accounts for unresolved turbulent mixing in addition to molecular diffusion, which is negligible at the spatial and temporal scales considered here. This eddy diffusivity formulation arises from Reynolds averaging, whereby the effects of subgrid-scale turbulent fluctuations are parameterised rather than explicitly resolved. The term “subgrid-scale” reflects that these motions occur below the resolution of the methods used to solve or measure spatio-temporal gas transport. Its use assumes a separation of scales between the resolved mean flow and smaller-scale turbulence, allowing turbulent transport to be approximated as a diffusive flux proportional to local concentration gradients. Atmospheric turbulence is difficult to model because it spans a wide range of interacting spatial and temporal scales that cannot be fully resolved in practical simulations. Key results from turbulence theory, such as the existence of an energy cascade (Kolmogorov, 1991) from large energy-containing eddies to smaller dissipative scales, provide the physical basis for representing unresolved turbulence through averaged or parameterised effects. However, these results also highlight the limitations of such approaches when scale separation is weak or when turbulence is strongly anisotropic or intermittent.

In this thesis, the scalar quantity corresponds to a specific gas, and the flow field is the wind. Additionally, we focus primarily on small-scale gas transport – typically within hundreds of metres – due to the size of the sites where emission sources are being identified. As such, we assume constant gas density, therefore neglecting buoyancy effects. Furthermore, because low-level atmospheric wind speeds generally have low Mach numbers, we use the incompressible Navier–Stokes equations:

$$\nabla \cdot \boldsymbol{v} = 0, \quad \frac{\partial \boldsymbol{v}}{\partial t} + (\boldsymbol{v} \cdot \nabla)\boldsymbol{v} = -\frac{1}{\rho}\nabla \boldsymbol{p} + \nu \nabla^2 \boldsymbol{v} + \boldsymbol{f}, \tag{2.1.2}$$

where, ρ represents a constant gas density, ν the constant kinetic viscosity, $\boldsymbol{p}(x, y, z, t)$ the pressure field and $\boldsymbol{f}$ corresponds to external forcing terms (e.g. gravity, electromagnetism). For larger sites, a simple Boussinesq approximation can be added to capture buoyancy effects due to density variations, without solving the fully compressible equations, which would be computationally expensive.

Together, the Navier–Stokes equations – which describe how the velocity field evolves in a viscous fluid – and the advection–diffusion equation – which uses that velocity field to predict how gases are transported and mixed – form the bedrock of modern atmospheric gas transport modelling.

2.2 Approximate analytical solutions

In practical applications, one typically chooses a single forward model, selecting the most suitable atmospheric gas emission transport model depends on the specific requirements and resources of the modeller. When computational speed is prioritised over physical realism, approximate analytical solutions are best suited. In general, the Navier-Stokes and advection-diffusion equations cannot be solved exactly analytically. However, under certain conditions, some closed-form solutions do exist. The most popular closed-form solution is the Gaussian plume model (Arystanbekova, 2004; Ramadan et al., 2008; Lushi and Stockie, 2010; Lotrecchiano et al., 2020). Parametrised by a relatively low number of variables and relying on strong assumptions, this model is extremely efficient to compute (Stockie, 2011).

Gaussian Plume Model. The Gaussian plume model provides long-term averaged gas concentrations in three dimensions, denoted (x, y, z), from a single point source located at $(\tilde{x}, \tilde{y}, \tilde{z})$, under the following assumptions:

1. The source emits at a constant rate, i.e. $S(\tilde{x}, \tilde{y}, \tilde{z}, t) = s$,
2. The wind velocity is constant and spatially uniform, i.e. $\boldsymbol{v}(x, y, z, t) = u$,
3. The solution is steady-state, i.e. $\frac{\partial C(x,y,z,t)}{\partial t} = 0$,
4. The ground and the atmospheric boundary layer perfectly reflect the gas,
5. There are no obstacles or boundaries in the (x, y) plane,
6. The ground is levelled, taken as the plane, i.e. $z = 0$.

The spatially uniform and temporally constant wind velocity assumption defines the flow field $\boldsymbol{v}$, therefore not requiring a Navier-Stokes solver. From the Stockie derivations (Stockie, 2011), which are based on the advection-diffusion equation (2.1.1), the distribution of the plume perpendicular to the wind direction has a two-dimensional Gaussian form (see Figure 2.2.1). Meaning, the gas concentration measured at location (x, y, z) can be written as:

$$C(x, y, z; \tilde{x}, \tilde{y}, \tilde{z}) = \frac{10^6}{\rho} \frac{s}{2\pi u |\boldsymbol{\Sigma}|^{\frac{1}{2}}} \exp\left\{ -\frac{1}{2} \boldsymbol{\delta}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\delta} \right\}, \tag{2.2.1}$$

where $\rho \in \mathbb{R}^+$ is the gas density [kg/m^3], the term $10^6/\rho$ ensures the gas concentration is in parts per million (a common unit in atmospheric gas measurements), u is the wind

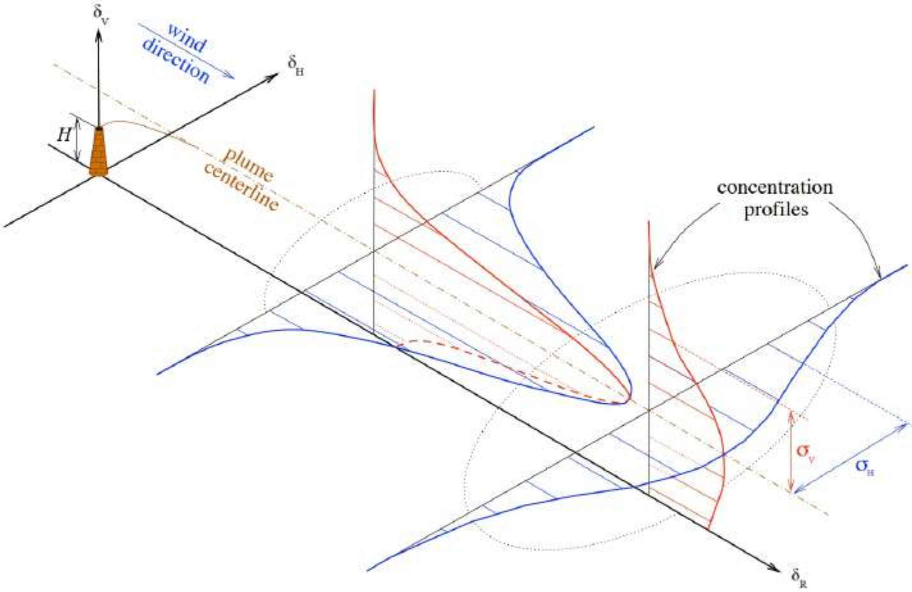


Figure 2.2.1: Representation of the Gaussian plume model. Credit: Stockie (2011).

speed [m/s], and s is the source emission rate [kg/s] for a source at location $(\tilde{x}, \tilde{y}, \tilde{z})$ [m]. Additionally, $\boldsymbol{\delta} = (\delta_H, \delta_V)$, for:

$$\begin{pmatrix} \delta_R \\ \delta_H \\ \delta_V \end{pmatrix} = \begin{pmatrix} \cos\gamma & \sin\gamma & 0 \\ -\sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{pmatrix} \times \left(\begin{bmatrix} x \\ y \\ z \end{bmatrix} - \begin{bmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{z} \end{bmatrix} \right),$$

where γ is the wind direction [rad], and:

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_H^2(\delta_R) & 0 \\ 0 & \sigma_V^2(\delta_R) \end{bmatrix},$$

with σ_H and σ_V being parametrised as a function of downwind distance δ_R and representing the standard deviation of the horizontal and vertical Gaussian distributions (see Figure 2.2.1); these are commonly referred to as “wind sigmas”.

A large literature exists on choosing wind sigmas and originated with Pasquill's atmospheric stability classes (ASC) (Pasquill, 1961). Pasquill's approach first determines the local ASC using meteorological data, then uses a dispersion scheme to fix wind sigmas according to the ASC. Nowadays, ASC-based dispersion schemes remain popular in practice (Kahl and Chapman, 2018), with Briggs (Briggs, 1973), Smith (Pasquill and Smith, 1983), Pasquill-Gifford, and Chinese National Standard being common choices (Mao et al., 2020). These power-law dispersion schemes based on downwind distances fix the wind sigma parameters by selecting dispersion parameters from ASC-based tables. However, atmospheric conditions are extremely complex, and by fixing the dispersion parameters we risk misspecifying them (Finn et al., 2016). In Chapter 4, we address this model misspecification by presenting a method to estimate the wind sigmas by estimating the dispersion parameters without relying on the ASCs.

Equations 2.2.1 can be rewritten to explicitly highlight the wind sigma parameters:

$$C(x, y, z; \tilde{x}, \tilde{y}, \tilde{z}) = \frac{10^6}{\rho} \frac{s}{2\pi u \sigma_H \sigma_V} \exp\left\{ -\frac{1}{2} \left(\frac{\delta_H^2}{\sigma_H^2(\delta_R)} + \frac{\delta_V^2}{\sigma_V^2(\delta_R)} \right) \right\}. \tag{2.2.2}$$

So far, we have only included horizontal and vertical offsets. However, we can extend Equation 2.2.2 to account for gas reflections against surrounding boundaries, similar work can has been done by Micallef and Micallef (2024). Assuming there are no horizontal boundaries, the horizontal term remains:

$$\exp\left\{ -\frac{\delta_H^2}{2\sigma_H^2(\delta_R)} \right\}.$$

However, the ground corresponding to $z = 0$ and the atmospheric boundary layer (ABL) with height $z = P$ (defined as the lowest part of Earth's atmosphere), form vertical boundaries against which the gas is assumed to reflect perfectly. Denoting the source height by $\tilde{z}$, the vertical reflections are captured by:

$$\begin{aligned}\sum_{j=1}^{n_{\text{refl}}} \Bigg[&\exp\left\{ -\frac{1}{2}\frac{(2\lfloor (j+1)/2 \rfloor P + (-1)^j(\delta_V + \tilde{z}) - \tilde{z})^2}{\sigma_V^2(\delta_R)} \right\} \\ &+ \exp\left\{ -\frac{1}{2}\frac{(2\lfloor j/2 \rfloor P + (-1)^{j-1}(\delta_V + \tilde{z}) + \tilde{z})^2}{\sigma_V^2(\delta_R)} \right\} \Bigg]. \end{aligned} \tag{2.2.3}$$

The first exponential term in Equation 2.2.3 corresponds to the reflection against the ABL, and the second exponential term to the reflection against the ground. The number of reflections against the ABL and against the ground is denoted by n_{refl}. Adding the vertical reflections from Equation 2.2.3 to the vertical and horizontal offsets of Equation 2.2.2, we obtain the following Gaussian plume equation:

$$\begin{aligned} C(x,y,z;\tilde{x},\tilde{y},\tilde{z}) = \frac{10^6}{\rho}\frac{s}{2\pi u \sigma_H \sigma_V} \exp\left\{ -\frac{\delta_H^2}{2\sigma_H^2(\delta_R)} \right\} \times \Bigg(&\exp\left\{ -\frac{\delta_V^2}{2\sigma_V^2(\delta_R)} \right\} \\ + \sum_{j=1}^{n_{\text{refl}}} \Bigg[&\exp\left\{ -\frac{1}{2}\frac{(2\lfloor (j+1)/2 \rfloor P + (-1)^j(\delta_V + \tilde{z}) - \tilde{z})^2}{\sigma_V^2(\delta_R)} \right\} \\ &+ \exp\left\{ -\frac{1}{2}\frac{(2\lfloor j/2 \rfloor P + (-1)^{j-1}(\delta_V + \tilde{z}) + \tilde{z})^2}{\sigma_V^2(\delta_R)} \right\} \Bigg] \Bigg). \end{aligned} \tag{2.2.4}$$

There exists an extension of the Gaussian plume model that captures some temporal variability in the wind field by relaxing the steady-state assumption. This extension, known as the Gaussian puff model, represents spatio-temporal gas transport as a sequence of discrete puffs that are released from the source and advected and dispersed independently according to time-varying meteorological conditions. By allowing wind speed and direction to vary between successive puffs, Gaussian puff models can account for non-stationary transport while retaining much of the analytical structure and computational efficiency of Gaussian plume formulations (Jia et al., 2025). A widely used

Variable	Definition	Units
$C(x, y, z) \in \mathbb{R}^+$	Plume gas concentration at (x, y, z)	ppm
$(x, y, z) \in \mathbb{R}^3$	Three-dimensional coordinates	m
$(\tilde{x}, \tilde{y}, \tilde{z}) \in \mathbb{R}^3$	Source location	m
$\rho \in \mathbb{R}^+$	Modelled gas density	kg/m^3
$s \in \mathbb{R}^+$	Source emission rate	kg/s
$u \in \mathbb{R}^+$	Wind speed	m/s
$P \in \mathbb{R}^+$	Height of the ABL	m
$j \in \mathbb{Z}^{1+}$	Reflection number	
$n_{\text{refl}} \in \mathbb{Z}^{1+}$	Maximum number of reflections	
$\sigma_H,\, \sigma_V \in \mathbb{R}^+$	Wind sigmas	m

Table 2.2.1: Gaussian plume equation parameters' definition.

implementation of this approach is the CALPUFF modelling system (Scire et al., 2000), which applies Gaussian puff theory and incorporates spatially and temporally varying meteorological fields, terrain effects, and boundary-layer parameterisations. While Gaussian puff models offer increased realism compared to steady-state plume models at a relatively modest computational cost (Ludwig et al., 1977), their detailed formulation and implementation are not considered further here, as they are not required for the objectives of this thesis.

2.3 Numerical solutions

For most real-world applications in atmospheric gas transport modelling, predicting only long-term averaged gas behaviour is insufficient. The simplifying assumptions behind approximate analytical solutions, such as the Gaussian plume model, often prove too restrictive. Atmospheric wind fields are inherently unsteady, terrain and topography are rarely uniform, and complex physical boundaries are common. To realistically capture how gases interact with such complex and dynamic features, it is necessary to move beyond approximate analytical solutions and directly solve the governing partial differential equations (PDEs). Namely, the Navier-Stokes equations (2.1.2) for wind motion and the advection-diffusion equation (2.1.1) for gas transport.

Because these equations generally do not admit exact analytical solutions except under very idealised conditions, they must be solved numerically. This is known as computational fluid dynamics (CFD) and involves approximating the behaviour of continuous PDEs through discretisation. Numerical approaches can be broadly divided into Eulerian and Lagrangian frameworks, which differ in how the gas transport is represented and discretised. These processes transform the continuous problem into a system of algebraic equations that can be handled computationally. The fidelity of these solutions depends heavily on the resolution of the discretisation. Higher resolution typically yields greater accuracy but also demands significantly more computational resources.

Lagrangian and Eulerian schemes differ primarily in how they represent and track gas motion. In Lagrangian schemes, individual gas parcels or particles are followed as they move through space, carrying properties such as velocity, density, and pressure; advection is handled naturally through particle motion, and the spatial resolution is defined by the particle distribution rather than a fixed mesh. Eulerian schemes, on the other hand, use a fixed spatial grid and calculate gas properties at each grid point, with advection treated numerically across the grid and boundary conditions applied directly to the cells. In essence, Lagrangian methods move with the flow, while Eulerian methods observe the flow from fixed positions in space.

While Lagrangian-based models, such as Lagrangian Particle Dispersion Models (for example: HYSPLIT (Draxler and Rolph, 2010), NAME (Jones et al., 2007), STILT (Lin et al., 2003), and FLEXPART (Bakels et al., 2024)), are widely used for simulating atmospheric gas transport and offer advantages in capturing advection and dispersion of individual parcels, the focus of this work is on Eulerian numerical solvers. Eulerian methods solve the governing equations on a fixed spatial grid, providing a straightforward framework for directly computing the evolution of gas concentration fields across the whole domain. By concentrating on Eulerian solvers, we leverage their suitability for efficiently handling complex flow fields, boundary conditions, and the coupled interaction between advection and diffusion in atmospheric gas transport.

Over the past several decades, various Eulerian numerical methods have been developed and refined to solve these equations efficiently and accurately. Common discretisation approaches include the finite difference method (Ferziger and Perić, 2002), the finite element method (Donea and Huerta, 2003), and the spectral method (Canuto et al., 2006). However, for most practical CFD applications – especially those involving complex geometries and the need to conserve physical quantities – the finite volume method (FVM) has emerged as the dominant approach. This method inherently respects conservation laws, ensuring that mass, momentum, and energy are preserved across each cell, and can flexibly accommodate structured or unstructured meshes (Moukalled et al., 2016; Eymard et al., 2000). We will now demonstrate how FVMs work through a simple example where we solve the three-dimensional advection-diffusion equation.

Finite Volume Method. The finite volume formulation begins by integrating Equation 2.1.1 over a cell's finite volume V, yielding:

$$\int_V \frac{\partial C}{\partial t}\,\mathrm{d}V + \int_{\partial V} C\boldsymbol{v}\cdot\mathbf{n}\,\mathrm{d}A = \int_{\partial V} K\nabla C\cdot\mathbf{n}\,\mathrm{d}A + \int_V S\,\mathrm{d}V.$$

Here, $\boldsymbol{v} = (v_x, v_y, v_z)^T$ is the flow field, $\mathbf{n}$ denotes the unit normal vector on the cell surface ∂V and A is the area of a face of the cell. This formulation highlights the physical interpretation of fluxes entering and exiting each cell. The divergence terms in Equation 2.1.1 have been converted to surface integrals using Gauss's divergence theorem (Moukalled et al., 2016). According to this fundamental theorem, the total volume of all sources, represented by the volume integral of the divergence, is precisely equivalent to the net flow across the boundary of the volume (Byron and Fuller, 1992).

For this example, we assume a structured Cartesian grid with uniform spacing Δx, Δy, and Δz along each axis. Let $C^t_{i,j,k}$ represent the average value of C in cell (i, j, k) at time step t. Here, the time derivative is discretised using the explicit forward Euler method, which gives:

$$C^{t+1}_{i,j,k} = C^t_{i,j,k} + \frac{\Delta t}{V}\left(-F^t_{i,j,k} + G^t_{i,j,k} + S^t_{i,j,k}\right),$$

where $V = \Delta x \Delta y \Delta z$ is the volume of each cell, $F^t \in \mathbb{R}$ denotes the total advective flux out of the cell, and $G^t \in \mathbb{R}$ represents the net diffusive flux at time t. The time step Δt must be chosen with care to ensure numerical stability. For the forward Euler method, a commonly used stability criterion is the Courant-Friedrichs-Lewy (CFL) condition (Courant et al., 1928), which for all cells (i, j, k) in this problem reads:

$$\Delta t \leq \min_{i,j,k}\left[\frac{1}{\frac{|v_{x_{i,j,k}}|}{\Delta x} + \frac{|v_{y_{i,j,k}}|}{\Delta y} + \frac{|v_{z_{i,j,k}}|}{\Delta z} + 2\left(\frac{K_x}{\Delta x^2} + \frac{K_y}{\Delta y^2} + \frac{K_z}{\Delta z^2}\right)}\right],$$

ensuring that information does not travel farther than one grid cell in a single numerical solver time step.

The advection term models transport due to bulk motion, and its discretisation must respect the directionality of the flow to avoid unphysical oscillations. For the flux along the x-axis, the advective term is:

$$F^t_{x_{i,j,k}} = \Delta y \Delta z\left[(C^t v^t_x)_{i+1/2,j,k} - (C^t v^t_x)_{i-1/2,j,k}\right].$$

However, since values at face centres (such as $C^t_{i+1/2,j,k}$) are not directly available, we approximate them using the upwind scheme. This method selects the upstream value of

C^t based on the sign of the velocity component $v^t_{x_{i,j,k}}$, which helps prevent non-physical backward propagation of information. Specifically:

$$(C^t v^t_x)_{i+1/2,j,k} = \begin{cases} v^t_{x_{i,j,k}}\, C^t_{i,j,k}, & \text{if } v^t_{x_{i,j,k}} \geq 0, \\ v^t_{x_{i,j,k}}\, C^t_{i+1,j,k}, & \text{if } v^t_{x_{i,j,k}} < 0, \end{cases}$$

$$(C^t v^t_x)_{i-1/2,j,k} = \begin{cases} v^t_{x_{i,j,k}}\, C^t_{i-1,j,k}, & \text{if } v^t_{x_{i,j,k}} \geq 0, \\ v^t_{x_{i,j,k}}\, C^t_{i,j,k}, & \text{if } v^t_{x_{i,j,k}} < 0. \end{cases}$$

This expression can be rewritten in the following compact and branch-free form:

$$F^t_{x_{i,j,k}} = \Delta y \Delta z \left[(C^t_{i,j,k} - C^t_{i-1,j,k}) \frac{|v^t_{x_{i,j,k}}| + v^t_{x_{i,j,k}}}{2} - (C^t_{i+1,j,k} - C^t_{i,j,k}) \frac{|v^t_{x_{i,j,k}}| - v^t_{x_{i,j,k}}}{2} \right].$$

The total advective flux through all faces of the cell is computed by summing the contributions along the x, y, and z axes, as shown by the following equation:

$$F^t_{i,j,k} = F^t_{x_{i,j,k}} + F^t_{y_{i,j,k}} + F^t_{z_{i,j,k}}.$$

Each term involves the appropriate area of the face and the corresponding velocity component.

In contrast to advection, the diffusion term represents tracer spreading due to unresolved mixing processes; in atmospheric applications this term is dominated by turbulent eddy diffusion rather than molecular motion. As such, it is well-approximated by the central difference scheme (Moukalled et al., 2016). The diffusive flux along the x-direction is given by:

$$G^t_{x_{i,j,k}} = K_x \Delta y \Delta z \left(\frac{C^t_{i+1,j,k} - 2C^t_{i,j,k} + C^t_{i-1,j,k}}{\Delta x^2} \right),$$

with analogous terms for the y- and z-directions. The total diffusion term is the sum:

$$G^t_{i,j,k} = G^t_{x_{i,j,k}} + G^t_{y_{i,j,k}} + G^t_{z_{i,j,k}}.$$

To fully specify the numerical solution, boundary conditions must be imposed at the edges of the domain. We typically apply Dirichlet boundary conditions, in which the scalar field C is prescribed on the boundaries. These values are directly inserted into the flux calculations whenever a neighbouring cell lies outside the computational domain. For certain boundaries, such as walls or the ground, Neumann (no-flux) conditions may be more appropriate, assuming no mass crosses the boundary.

It is important to note that the implementation described here represents just one possible realisation of the FVM. In practice, many variations exist depending on the nature of the grid (structured or unstructured), the treatment of fluxes (e.g., higher-order schemes like WENO or QUICK), and the temporal discretisation (e.g., implicit or semi-implicit schemes such as the Crank-Nicolson method (Crank and Nicolson, 1947)). The method outlined in this section focuses on a structured Cartesian grid, explicit time integration, and first-order upwind and central difference approximations, which together provide a transparent and pedagogically sound demonstration of FVM principles. A more exhaustive survey of advanced FVM techniques, including adaptive mesh refinement, flux limiters, and hybrid discretisations, is beyond the scope of this thesis.

A range of commercial and open-source CFD solvers implement FVMs and are widely used in both research and industry. Notable examples include ANSYS Fluent (Matsson, 2022), a popular commercial solver known for its robustness and extensive turbulence modelling capabilities, and OpenFOAM (Weller et al., 1998), a widely adopted open-source CFD toolbox that provides great flexibility for customising solvers and physical models. Other examples include STAR-CCM+ (Zou et al., 2018), Phiflow (Holl and Thuerey, 2024), and SimScale (Murad, 2021), each offering specialised capabilities depending on the application domain. Phiflow is used in Chapter 5 to simulate atmospheric gas emission transports under obstructed unsteady-state flow fields.

Today, numerical solvers form the backbone of high-fidelity atmospheric and environmental fluid flow modelling. Sophisticated CFD frameworks allow researchers and practitioners to study gas dispersion over complex terrains, around obstacles, and under changing meteorological conditions with remarkable detail. However, these methods can be computationally intensive, particularly for large domains or when high temporal and spatial resolution is required. As a result, the field continues to push for improvements in numerical algorithms, parallel computing, and hybrid approaches that balance accuracy and efficiency. This has motivated recent and growing interest in alternative approaches, such as data-driven and deep learning-based surrogate models, which seek to approximate these complex dynamics more efficiently.

2.4 Deep learning surrogate models

Despite the significant advances made through numerical solvers, the high computational cost associated with solving the Navier–Stokes and advection–diffusion equations at high resolutions and over large domains remains a major bottleneck. This limitation has motivated a surge of interest in developing surrogate models (Kutz, 2017). These data-driven approaches can approximate the solutions of the governing equations orders of magnitude faster than traditional CFD methods.

Deep learning surrogate models leverage the universal function approximation capabilities of neural networks to learn the complex mappings between input conditions (such as boundary conditions, source terms, or flow parameters) and the resulting fluid flow or scalar transport fields. Once trained, these models can generate accurate predictions with significantly reduced computational effort, enabling applications such as real-time forecasting and many-query scenarios like design optimisation and inverse problems.

A range of neural network architectures has been explored for this purpose. Convolutional Neural Networks (CNNs) have been employed to capture spatial correlations in structured flow fields (Thuerey et al., 2020), while more recent approaches have used Graph Neural Networks (GNNs) to handle unstructured meshes and irregular domains (Pfaff et al., 2020). Physics-Informed Neural Networks (PINNs) represent another major development. They embed the governing PDEs directly into the loss function of the network, allowing the model to respect physical constraints even when data is limited (Raissi et al., 2019).

Recent work has demonstrated the effectiveness of these models across a range of fluid dynamics applications. For example, Guo et al. (2016) used a CNN to predict steady flows around obstacles, showing excellent agreement with traditional solvers at a fraction of the computational cost. Similarly, Pathak et al. (2018) introduces a framework for the prediction of large spatio-temporally chaotic systems. More recently, Fourier Neural Operators (Li et al., 2021) have emerged as a powerful approach for learning solution operators for PDEs, generalising across varying input geometries and initial conditions.

We now more formally introduce two neural network architectures, which are used later in this thesis as CFD surrogate models, namely, the *Multilayer Perceptron* (used in Chapter 5) and the *Convolutional Neural Network* (used in Chapter 6).

Multilayer Perceptron. An Multilayer Perceptron (MLP) (Rosenblatt, 1958) is a type of artificial neural network widely used in machine learning for tasks such as classification and regression. Their flexibility and simple implementation make them widely applicable across domains. An MLP generates outputs $\boldsymbol{y}_{\text{nn}}$ and is trained on a dataset of input–output pairs $(\boldsymbol{x}_{\text{nn}}, \boldsymbol{d})$. During training, the MLP learns to approximate $\boldsymbol{d}$ by minimising a predefined distance metric between $\boldsymbol{y}_{\text{nn}}$ and $\boldsymbol{d}$ for given $\boldsymbol{x}_{\text{nn}}$. It

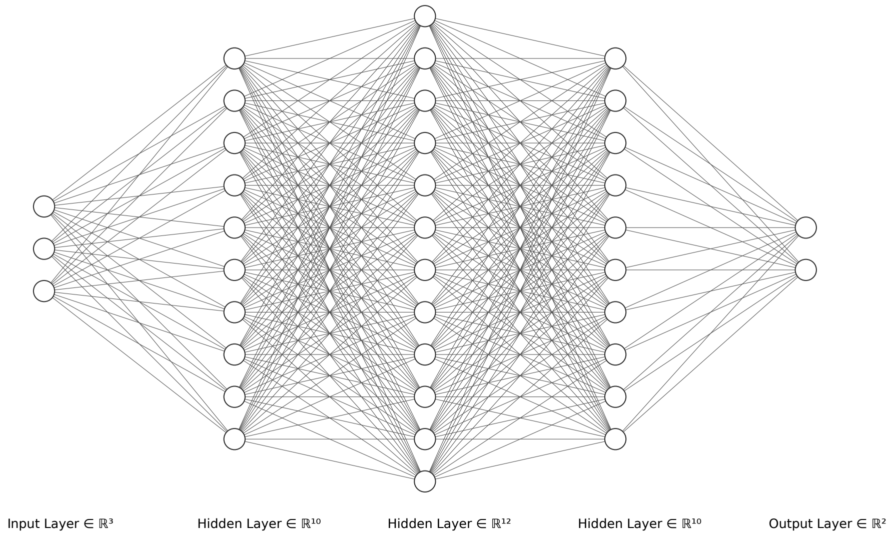


Figure 2.4.1: Visualisation of an MLP's architecture with 3 input neurons, 3 hidden layers with 10, 12, and 10 neurons respectively and 2 output neurons. Each circle represents a neuron and each line a connection, depicting the fully connected architecture.

contains an input layer, one or more hidden layers, and an output layer. Each layer consists of multiple neurons, and the layers are fully connected through weights (See Figure 2.4.1).

Each neuron in a layer computes a weighted sum of its inputs, adds a bias term, and passes the result through an activation function. For neuron i in layer j, the preactivation value is defined by:

$$\zeta_{i,j} = \sum_{k=1}^{n_{\mathrm{nrn}}} w_{i,k}\zeta_{k,j-1} + b_i,$$

where weights $w_{i,k}$, represent the strength of the connection between the i-th neuron in the j-th layer and the k-th neuron in the $(j-1)$-th layer, n_{nrn} is the number of neuron in the layer $j-1$, and $b_i \in \mathbb{R}$ is the bias. The activation (output) of the neuron is given by:

$$a_{i,j} = \sigma_{\mathrm{af}}(\zeta_{i,j}),$$

where σ_{af} is a non-linear activation function. Activation functions introduce nonlinearity, enabling the MLP to model complex functions. Common activation functions include:

- Sigmoid: $\sigma_{\text{af}}(\zeta) = \dfrac{1}{1 + e^{-\zeta}}$, output in $(0, 1)$,
- Hyperbolic tangent (Tanh): $\sigma_{\text{af}}(\zeta) = \dfrac{e^{\zeta} - e^{-\zeta}}{e^{\zeta} + e^{-\zeta}}$, output in $(-1, 1)$,
- ReLU (Rectified Linear Unit): $\sigma_{\text{af}}(\zeta) = \max(0, \zeta)$.

The forward pass computes layer-by-layer outputs from the input to the output. For a hidden layer j, with input $\boldsymbol{a}^{(j-1)}$, the pre-activation and activation vectors are:

$$\boldsymbol{\zeta}^{(j)} = \boldsymbol{W}^{(j)}\boldsymbol{a}^{(j-1)} + \boldsymbol{b}^{(j)}, \quad \boldsymbol{a}^{(j)} = \sigma_{\text{af}}\left(\boldsymbol{\zeta}^{(j)}\right),$$

where $\boldsymbol{W}^{(j)}$ is the weight matrix for layer j, $\boldsymbol{b}^{(j)}$ is the bias vector, and σ_{af} is applied element-wise. MLPs are trained using backpropagation, which adjusts weights and biases to minimise a loss function. The process involves:

1. Forward pass: Compute network output $\boldsymbol{y}_{\text{nn}}$ from input $\boldsymbol{x}_{\text{nn}}$,
2. Loss computation: Evaluate loss $\mathcal{L}(\boldsymbol{y}_{\text{nn}}, \boldsymbol{d})$ using vector of data $\boldsymbol{d}$, (e.g., mean squared error or cross-entropy),
3. Backward pass: Compute gradients $\frac{\partial \mathcal{L}}{\partial \boldsymbol{W}^{(j)}}, \frac{\partial \mathcal{L}}{\partial \boldsymbol{b}^{(j)}}, \forall\ j$
4. Gradient descent update:

$$\boldsymbol{W}^{(j)} \leftarrow \boldsymbol{W}^{(j)} - \eta \cdot \frac{\partial \mathcal{L}}{\partial \boldsymbol{W}^{(j)}}, \quad \boldsymbol{b}^{(j)} \leftarrow \boldsymbol{b}^{(j)} - \eta \cdot \frac{\partial \mathcal{L}}{\partial \boldsymbol{b}^{(j)}},$$

where η is the learning rate.

While MLPs are powerful for learning complex mappings, they have significant limitations when applied to high-dimensional inputs such as images. Each neuron in an MLP is fully connected to the previous layer, leading to a large number of parameters and making the model prone to overfitting, vanishing or exploding gradients, and computational inefficiency. Moreover, MLPs do not exploit the spatial structure of data, such as the local correlations and translation invariance present in images. To address these challenges, Convolutional Neural Networks were developed.

Convolutional Neural Network. A Convolutional Neural Network (CNN) (LeCun et al., 1989) is a class of neural networks specifically designed to process grid-like data such as images (e.g., two-dimensional fluid fields). Unlike MLPs, which connect each neuron to all inputs, CNNs learn features in a layered, hierarchical way and capture spatially local patterns, enabling more efficient learning from high-dimensional data. CNNs are particularly effective for tasks involving image classification, object detection, or next image prediction.

Similar to MLPs, CNNs are trained using input-output pairs, where the model learns to map inputs (e.g., images of gas fields) to labels, values, or another image by minimising a loss function using backpropagation and gradient descent. However, CNNs possess a different architecture (see Figure 2.4.2).

A typical CNN consist of the following components:

1. Feature extraction: The input data (e.g., an image) is passed through a series of convolutional layers, each followed by non-linear activation functions.

2. Intermediate downsampling and/or upscaling : Depending on the task, intermediate layers may reduce and/or increase the spatial resolution.

3. Prediction head (task-specific): The extracted features are passed to a task-dependent output module:

    - For classification, this typically involves flattening or global average pooling

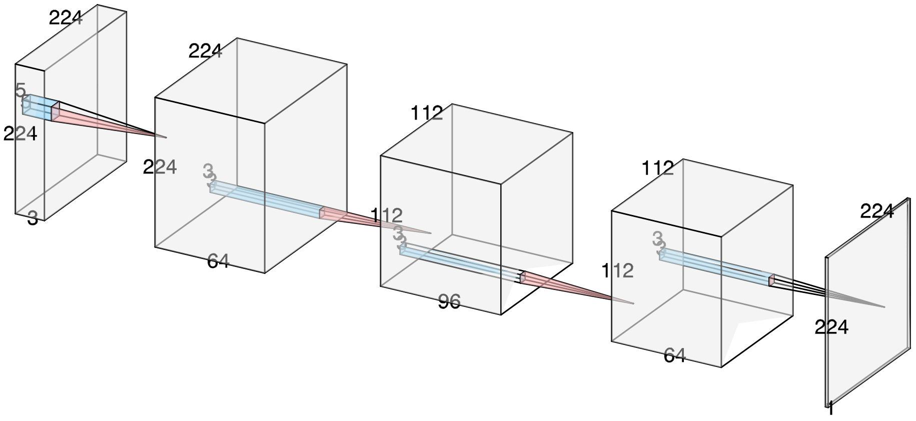


Figure 2.4.2: CNN example visualisation. Let $n_{\mathrm{h}} \times n_{\mathrm{w}} \times n_{\mathrm{c}}$ denote a tensor with height n_{h}, width n_{w}, and n_{c} channels. The first block represents the input image with dimensions $224 \times 224 \times 3$, corresponding to an RGB image. The second block is a convolutional layer with 64 output channels, a kernel size of 5×5, and output dimensions $224 \times 224 \times 64$, indicating that spatial resolution is preserved. The third block applies max pooling with a 3×3 kernel, reducing the spatial dimensions by half and resulting in an output of shape $112 \times 112 \times 96$. The fourth block is a convolutional layer with a 3×3 kernel, which maintains the same spatial resolution but reduces the number of channels to 64, giving an output of $112 \times 112 \times 64$. Finally, the fifth block represents the output layer, which applies a up-convolution with a 3×3 kernel and produces a single-channel output of dimensions $224 \times 224 \times 1$, matching the original input resolution.

followed by one or more fully connected layers.

- For segmentation or image-to-image tasks, the prediction head often consists of additional convolutional (and possibly up-convolutional) layers that preserve or restore spatial resolution, producing a dense output such as a pixel-wise label map or a generated image.

We will now provide a fuller description of these steps. Let's assume an image is passed as the input to a CNN. Each colour component is stored in its own matrix, commonly referred to as a channel. For example, a grayscale image has 1 channel, while an RGB image has 3 channels (red, green, blue). Thus, the input to a CNN has shape $n_\mathrm{h} \times n_\mathrm{w} \times n_\mathrm{c}$, where n_h is the number of pixels in each column, n_w is the number of pixels in each row, and n_c is the number of channels. The input data is processed through a convolutional layer composed of multiple filters (also referred to as kernels). Each filter, denoted $\mathbf{\Gamma}$, is a tensor of dimensions $k \times k \times n_\mathrm{c}$, where k denotes the spatial extent (typically 3 or 5 pixels), and n_c is the number of input channels. The total number of filters, denoted as n_f, determines the number of output channels in the resulting feature representation. Each filter slides over the input spatially, performing an element-wise multiplication followed by a summation at each location to produce a single scalar output. This operation generates a two-dimensional activation map, commonly referred to as a feature map, with spatial dimensions $n'_\mathrm{h} \times n'_\mathrm{w}$. When preserving the spatial dimension of the input is important, padding is typically applied. Padding involves adding extra rows and columns (usually filled with zeros) around the input prior to convolution, thereby ensuring that the output dimensions $n'_\mathrm{h} \times n'_\mathrm{w}$ remain consistent with design requirements. The collection of n_f feature maps produced by the filters is aggregated into a single output tensor of shape $n'_\mathrm{h} \times n'_\mathrm{w} \times n_\mathrm{f}$, which serves as the input to the subsequent layer in the network.

More formally, for a given input $\mathbf{X} \in \mathbb{R}^{n_\mathrm{h} \times n_\mathrm{w} \times n_\mathrm{c}}$ and a set of n_f filters $\mathbf{\Gamma}^{(1)}, \ldots, \mathbf{\Gamma}^{(n_\mathrm{f})}$, each of shape $k \times k \times n_\mathrm{c}$, a convolutional layer produces an output tensor $\mathbf{Z} \in \mathbb{R}^{n'_\mathrm{h} \times n'_\mathrm{w} \times n_\mathrm{f}}$. Each filter $\mathbf{\Gamma}^{(\tau)}$, for $\tau = 1, 2, \ldots, n_\mathrm{f}$, is convolved over the spatial dimensions of the input, summing across its channels:

$$Z_{i,j,\tau} = \sum_{e=1}^{k} \sum_{f=1}^{k} \sum_{g=1}^{n_\mathrm{c}} \Gamma^{(\tau)}_{e,f,g} \cdot X_{i+e-1, j+f-1, g} \ ,$$

where $Z_{i,j,\tau}$ is the activation at spatial position (i, j) in the τ-th feature map.

To improve computational efficiency, capture more abstract features and expand the receptive field, CNNs use pooling, stride and/or dilation. Pooling layers reduce the dimensions of feature maps by summarising local regions, which decreases computation. For instance, 2×2 max pooling selects the maximum value from each non-overlapping 2×2 pixel region, effectively downscaling the number of height and width pixels (including padding pixels) by a factor of 2. Strided convolution achieves a similar downsampling effect by moving the filter in steps larger than one. For example, a convolution with stride 2 skips every other location, effectively halving the output size. Unlike pooling, stride is a property of the convolution itself, part of the learned filtering, allowing feature extraction and downsampling in a single operation without creating an additional layer. Dilation increases the receptive field of filters without reducing resolution or increasing parameters by introducing gaps between filter elements. For example, a 3×3 filter with dilation rate 2 covers a 5×5 region of the input, enabling the network to capture broader context while preserving spatial resolution (see Figure 2.4.3). Together, these mechanisms make CNNs more efficient and expressive, allowing them to scale to large inputs and learn complex hierarchical patterns.

After several convolutional and pooling layers, CNNs typically produce feature maps that are passed through a task-specific prediction head. This may include flattening and fully connected layers, global pooling, or up-convolutional layers, depending on the architecture and objective. Importantly, the output is not always a single class label. In

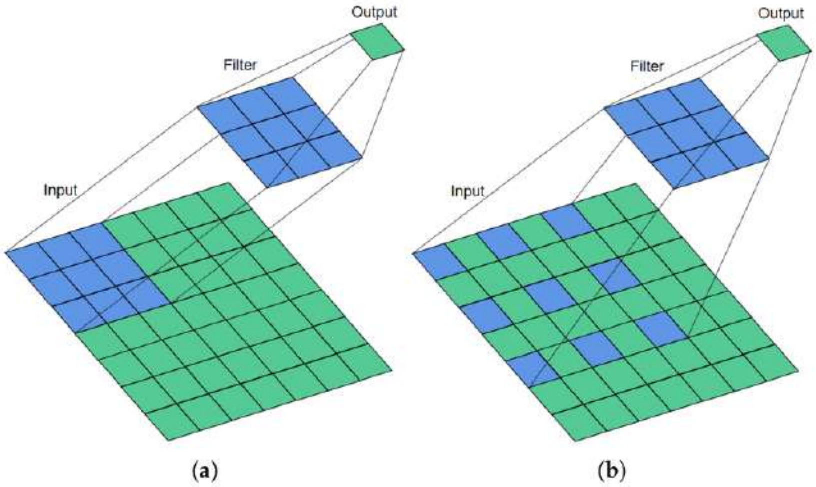


Figure 2.4.3: Effect of dilation rate on convolutional filters. (a) A 3×3 filter with dilation rate 1 (no dilation) covers a 3×3 region of the input. (b) The same filter with dilation rate 2 spans a 5×5 region, allowing the network to capture a wider context without reducing spatial resolution. Credit: He et al. (2021).

many applications, such as image segmentation, super-resolution, or image translation, the network produces another image or a dense spatial map as output, rather than a single value or vector.

CNNs offer major improvements over MLPs for structured data like images. They significantly reduce the number of parameters through local connectivity and weight sharing, improve generalisation by capturing spatial features, and scale better to high-dimensional inputs. While we have described the basic structure of convolutional neural networks, it is worth noting that many more advanced architectures exist. Models such as U-Nets (Ronneberger et al., 2015), ResNets (He et al., 2016), and EfficientNets (Tan and Le, 2019) incorporate sophisticated components like skip connections, residual blocks, and multi-scale feature extraction to enhance performance on complex tasks. However, a detailed discussion of these architectures is beyond the scope of this thesis and will not be addressed here.

While deep learning surrogate models like MLPs and CNNs offer tremendous promise, they also come with challenges. High-quality training data can be expensive to generate, and ensuring generalisation outside the training distribution remains an active area of research (Mishra et al., 2025). Moreover, integrating physical constraints robustly into deep architectures to avoid unphysical predictions continues to be an important focus for the community (Liu et al., 2025).

Nevertheless, the rapid progress in this area suggests that data-driven surrogate modelling will increasingly complement – and in some contexts, augment – conventional numerical solvers. As computational resources grow and machine learning techniques become more sophisticated, hybrid workflows combining high-fidelity CFD simulations with fast and flexible surrogate models are likely to play a pivotal role in addressing complex, real-world atmospheric and environmental modelling problems.

Chapter 3

Bayesian Parameter Estimation

The primary objective of inverse problems is to estimate an unobserved parameter or set of parameters, denoted by $\boldsymbol{\theta}$, based on indirect measurements $\boldsymbol{d}$. Here, unobserved means that $\boldsymbol{\theta}$ cannot be measured directly. In a probabilistic framework, this is typically posed as determining the values of $\boldsymbol{\theta}$ that maximise the probability of observing the data $\boldsymbol{d}$. In other words, we ask: "*Given all possible values of $\boldsymbol{\theta}$, which ones best explain the observations?*". This question is traditionally addressed using maximum likelihood estimation.

However, maximum likelihood estimation is a frequentist method, and its uncertainty quantification is typically based on asymptotic results such as the central limit theorem and the Fisher information. While these approximations are valid in large-sample settings, they can be unreliable when data are sparse, leading to potentially misleading inference and less reliable decision-making.

An alternative framework is Bayesian inference, which characterises uncertainty through the posterior distribution of the parameters $\boldsymbol{\theta}$. Instead of relying on point estimates and asymptotic approximations, Bayesian methods quantify the plausibility of different parameter values directly in terms of probability density functions (PDFs), providing a coherent way to account for uncertainty even in small-sample scenarios.

Throughout this thesis, we employ a range of algorithms and modelling frameworks for Bayesian parameter estimation. In this chapter[1], Section 3.1 motivates Monte Carlo methods. Section 3.2 presents *Markov chain Monte Carlo* methods. Section 3.3 then describes *state-space models* – which provide a flexible framework for representing dynamic systems in the Bayesian context – and introduces *particle filters*.

3.1 Motivating Monte Carlo methods

In a Bayesian approach, we model our uncertainty about continuous variables using the posterior density $p(\boldsymbol{\theta} \mid \boldsymbol{d})$, which represents the probability distribution of $\boldsymbol{\theta}$ given the observed data $\boldsymbol{d}$. Furthermore, the Bayesian framework shines through its ability to incorporate prior knowledge about the parameters using a prior distribution $p(\boldsymbol{\theta})$. This enables the integration of expert opinion or prior empirical evidence about the values of $\boldsymbol{\theta}$ into the estimation process, providing a more comprehensive basis for inference. Bayes' theorem formalises this relationship using the following equation:

$$p(\boldsymbol{\theta} \mid \boldsymbol{d}) = \frac{p(\boldsymbol{d} \mid \boldsymbol{\theta})\, p(\boldsymbol{\theta})}{p(\boldsymbol{d})}.$$

Here, $p(\boldsymbol{d} \mid \boldsymbol{\theta})$ is the likelihood function – the same function that maximum likelihood estimation seeks to maximise – while $p(\boldsymbol{\theta})$ is the prior density. The denominator:

$$p(\boldsymbol{d}) = \int p(\boldsymbol{d} \mid \boldsymbol{\theta})\, p(\boldsymbol{\theta})\, \mathrm{d}\boldsymbol{\theta},$$

known as the marginal likelihood or evidence, serves to normalise the posterior.

In practice, however, the marginal likelihood is often intractable. High-dimensional pa-

[1] Throughout this chapter, the words "parameter" and "state" will be used interchangeably.

rameter spaces or complex, nonlinear models can render the integral computationally infeasible. This prohibits direct computation of the posterior density using a closed-form solution. One common workaround is the Laplace approximation, which approximates the posterior as a Gaussian distribution centred at its mode. Yet, this assumption of normality is not always valid – particularly in complex applications such as gas emission inversions, where multiple interacting sources can result in multi-modal or highly skewed posterior distributions. To address these challenges, more flexible computational methods are required.

A key tool in this context is the family of Monte Carlo methods, which rely on sampling to approximate quantities that are difficult or impossible to compute analytically, such as the posterior distribution. Classical Monte Carlo techniques, such as rejection sampling or importance sampling, generate independent samples from a simpler proposal distribution and then correct for the difference between the proposal and the posterior distribution. Sampling using these methods is extremely computationally efficient and yields independent and identically distributed (iid) samples. However, their efficiency in approximating distributions declines rapidly in high-dimensional or complex posterior settings where the number of iid samples required to approximate the posterior density grows exponentially.

To address this, Markov chain Monte Carlo (MCMC) methods construct a Markov chain that has the targeted posterior distribution as its stationary distribution. Unlike independent Monte Carlo methods, MCMC samples are inherently dependent. Each sample is generated conditionally based on the previous sample, which allows the algorithm to explore complex, high-dimensional distributions more effectively by making local moves. However, informed dependent proposals often rely on computationally expensive local information such as gradients or the Hessian, making MCMC sampling less computationally efficient. Furthermore, the dependence introduces autocorrelation, so one must run the chain long enough to ensure good mixing and disregard the burn-in period to approximate independent draws.

While MCMC methods are well-suited for exploring complex posterior distributions, they operate in an offline manner and are ill-equipped for problems where data arrive sequentially. In many real-world applications – such as online monitoring, tracking, or environmental forecasting – new observations are continuously collected, and inference must be updated in real-time. MCMC, requiring a full dataset and long chains for convergence, is often impractical in such settings.

To address this, sequential Monte Carlo (SMC) methods offer a natural extension of Monte Carlo techniques to sequential data. Rather than drawing a long Markov chain, SMC maintains a population of weighted samples (particles) that evolve over time, adapting to new data as it becomes available. These methods approximate the posterior distribution recursively, making them particularly useful for online inference in dynamic systems.

A widely used SMC variant is the particle filter, which is designed for state-space models, where hidden variables evolve over time and are observed indirectly through noisy measurements. Particle filters use a combination of prediction, weighting, and resampling to estimate the posterior over latent states, making them highly effective for real-time applications.

The following sections explore these ideas in more depth. We begin with a detailed description of MCMC methods and their role in high-dimensional posterior inference. We then introduce state-space models and show how particle filters implement sequential Bayesian inference in these dynamic settings.

3.2 Markov chain Monte Carlo

When the marginal likelihood function is intractable, MCMC methods are often considered the gold standard for approximating the posterior density. In this section, we introduce two fundamental algorithms in the MCMC literature, namely, *Gibbs sampling* (Section 3.2.1) and the *Metropolis-Hastings* algorithm (Section 3.2.2).

3.2.1 Gibbs sampling

Gibbs sampling (Geman and Geman, 1984) is a fundamental technique in the family of MCMC methods. It is particularly advantageous when the joint posterior distribution $p(\boldsymbol{\theta} \mid \boldsymbol{d})$ is complex and difficult to sample from directly, but the conditional posterior distributions are available in closed-form.

Suppose the parameter vector is $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_{n_{\text{par}}})$. In Gibbs sampling, we generate a sequence of samples by iteratively drawing each parameter θ_i from its conditional posterior distribution given the current values of all other parameters and the data, i.e.:

$$\theta_i^{(l+1)} \sim p(\theta_i \mid \boldsymbol{\theta}_{-i}^{(l+1)}, \boldsymbol{d}), \quad i = 1, \ldots, n_{\text{par}},$$

where n_{par} is the number of estimated parameters, l is the iteration number, and $\boldsymbol{\theta}_{-i}$ denotes all parameters except θ_i.

When the prior and likelihood are conjugate, these conditional posterior distributions often have known analytical forms, making the sampling straightforward. By repeatedly updating each component parameter in this way, the Markov chain has the target posterior $p(\boldsymbol{\theta} \mid \boldsymbol{d})$.

3.2.2 Metropolis-Hastings

The Metropolis–Hastings (MH) algorithm (Metropolis et al., 1953; Hastings, 1970) is a flexible and widely used MCMC method for sampling from a complex posterior distribution $p(\boldsymbol{\theta} \mid \boldsymbol{d})$ when direct sampling is infeasible. Unlike Gibbs sampling, which requires closed-form posterior conditional distributions, MH only requires that the posterior density can be evaluated up to a normalising constant.

Given the current state $\boldsymbol{\theta}^{(l)}$, a candidate value $\boldsymbol{\theta}^*$ is proposed from a proposal distribution $q(\boldsymbol{\theta}^* \mid \boldsymbol{\theta}^{(l)})$. This proposal is accepted with probability:

$$\alpha(\boldsymbol{\theta}^{(l)}, \boldsymbol{\theta}^*) = \min\left\{1,\ \frac{p(\boldsymbol{\theta}^* \mid \boldsymbol{d})\, q(\boldsymbol{\theta}^{(l)} \mid \boldsymbol{\theta}^*)}{p(\boldsymbol{\theta}^{(l)} \mid \boldsymbol{d})\, q(\boldsymbol{\theta}^* \mid \boldsymbol{\theta}^{(l)})}\right\}.$$

If accepted, the chain moves to $\boldsymbol{\theta}^{(l+1)} = \boldsymbol{\theta}^*$; otherwise, it remains at $\boldsymbol{\theta}^{(l+1)} = \boldsymbol{\theta}^{(l)}$. This mechanism guarantees that the Markov chain has the desired target posterior distribution as its stationary distribution.

A common choice for the proposal distribution is a random walk proposal, such as:

$$q(\boldsymbol{\theta}^* \mid \boldsymbol{\theta}^{(l)}) = \mathrm{N}(\boldsymbol{\theta}^*; \boldsymbol{\theta}^{(l)}, \boldsymbol{\Sigma}),$$

where $\boldsymbol{\Sigma}$ controls the step size and direction of the proposed moves. For symmetric proposals (i.e., $q(\boldsymbol{\theta}^* \mid \boldsymbol{\theta}^{(l)}) = q(\boldsymbol{\theta}^{(l)} \mid \boldsymbol{\theta}^*)$), the acceptance probability simplifies to:

$$\alpha(\boldsymbol{\theta}^{(l)}, \boldsymbol{\theta}^*) = \min\left\{1,\ \frac{p(\boldsymbol{\theta}^* \mid \boldsymbol{d})}{p(\boldsymbol{\theta}^{(l)} \mid \boldsymbol{d})}\right\}.$$

A powerful refinement of the basic MH algorithm is the use of proposals informed by the local geometry of the target distribution. In the Metropolis-adjusted Langevin algorithm (MALA) (Roberts and Tweedie, 1996), the proposal distribution incorporates the gradient of the log-posterior to guide the chain toward regions of higher probability. Given the current state $\boldsymbol{\theta}^{(l)}$, a candidate state $\boldsymbol{\theta}^*$ is proposed using a discretised Langevin diffusion:

$$q(\boldsymbol{\theta}^* \mid \boldsymbol{\theta}^{(l)}) = \mathrm{N}\Big(\boldsymbol{\theta}^* \,;\, \boldsymbol{\theta}^{(l)} + 0.5\epsilon_{\mathrm{ss}}\boldsymbol{Q}^{-1}(\boldsymbol{\theta}^{(l)})\nabla_{\boldsymbol{\theta}^{(l)}} \log p(\boldsymbol{\theta}^{(l)} \mid \boldsymbol{d}),\ \epsilon_{\mathrm{ss}}\boldsymbol{Q}^{-1}(\boldsymbol{\theta}^{(l)})\Big),$$

where ϵ_{ss} is the step size and $\boldsymbol{Q}^{-1}$ is a preconditioning matrix, often chosen to be the identity matrix $\mathbb{I}$.

In MALA, the step size ϵ_{ss} plays a crucial role in balancing exploration and acceptance rates. If ϵ_{ss} is too small, the chain will move very slowly through the state space, resulting in highly correlated samples and poor mixing. Conversely, if ϵ_{ss} is too large, the discretised Langevin proposal may no longer approximate the continuous diffusion well, leading to a low acceptance rate and potential bias. Optimal scaling results (Roberts and Rosenthal, 1998) suggest that for high-dimensional targets, an acceptance rate around 57.4% is ideal for MALA.

To further improve sampling efficiency, one can adapt the proposal covariance $\boldsymbol{Q}^{-1}$ to match the local curvature of the posterior. A common strategy is to precondition the proposal using an approximation to the inverse Hessian matrix of the log-posterior:

$$\boldsymbol{Q}^{-1}(\boldsymbol{\theta}^{(l)}) = \left[\nabla^2 \log(p(\boldsymbol{\theta}^{(l)} \mid \boldsymbol{d}))\right]^{-1}.$$

When $\boldsymbol{Q}(\boldsymbol{\theta}^{(l)})$ is positive definite, its inverse can serve as a local approximation to the posterior covariance. This motivates more sophisticated samplers, such as the Riemannian manifold Langevin, also known as manifold Metropolis-adjusted Langevin algorithm (M-MALA) – which we use in Chapter 4 – and Hamiltonian Monte Carlo methods (Girolami and Calderhead, 2011), which use the local geometry to adjust both the direction and scale of proposals.

Proper preconditioning allows the proposal to adapt to anisotropic or highly correlated posteriors, enabling larger and more efficient moves in directions of low curvature while preventing overly large steps in directions of high curvature. This is especially important in complex Bayesian inverse problems, where the posterior may exhibit strong correlations.

3.3 State-space models

So far, we have considered scenarios in which the parameters to be estimated are assumed to remain constant for all collected data. However, in many real-world situations, this assumption does not hold: the underlying parameter values may evolve over time as new data are observed. This is especially common in applications involving temporal measurements. For example, when estimating the position of a moving object, tracking heart rate using photoplethysmography (as in wearable optical sensors), or in our case an emission rate, the parameters of interest naturally change as time progresses. These changes are often governed by underlying physical or physiological dynamics, which, if appropriately modelled, can substantially improve our estimates of how the parameter evolves. In the case of heart rate monitoring, for instance, the true heart rate may not be measured directly at every instant but can be modelled as a function of physical effort, allowing us to make more accurate inferences.

State-space models provide a framework for modelling such dynamical systems (Jazwinski, 1970; Kitagawa, 1996; Douc and Cappé, 2005; West and Harrison, 2006). A state-space model is defined by two key components: the *state evolution equation* and the *measurement equation.* The state evolution equation:

$$\boldsymbol{\theta}_t = f_{\text{state}}(\boldsymbol{\theta}_{t-1}, \boldsymbol{\theta}_{t-2}, \ldots) + \boldsymbol{\epsilon}_{\text{evo}_t}, \tag{3.3.1}$$

describes how the latent (unobserved) state evolves over time t, where $\boldsymbol{\epsilon}_{\text{evo}_t}$ is a multivariate Gaussian process noise term, while the measurement equation:

$$\boldsymbol{d}_t = f_{\text{obs}}(\boldsymbol{\theta}_t) + \boldsymbol{\epsilon}_{\text{obs}_t}, \tag{3.3.2}$$

relates this latent state to the actual observations with some measurement error $\boldsymbol{\epsilon}_{\text{obs}_t}$. Here, f_{state} represents the process model that maps past state values to the current state, while f_{obs} specifies how the current state generates the observed data.

Bayesian methods for dynamic parameter estimation have been extensively developed within the state-space framework. Among these, the Kalman filter (KF) (Kalman, 1960) is one of the most widely used algorithms. The KF operates iteratively, alternating between predicting the state based on the state evolution equation and updating this prediction in light of new observations. Its computational efficiency arises from the fact that it performs Bayesian updates in closed-form, using only matrix operations. However, this efficiency comes at the cost of critical assumptions. The observation function f_{obs} must be linear and the process noise must be Gaussian.

In many practical applications – such as atmospheric gas transport, which is governed by the Navier-Stokes and advection-diffusion equations – this linearity assumption breaks down. The resulting relationships between the observed data and the underlying states can be highly nonlinear – this is the case when estimating an emission source location. While extensions such as the extended Kalman filter (Jazwinski, 1970) and unscented Kalman filter (Julier et al., 2002) relax the linearity assumption locally, they can still fail when the system is strongly nonlinear or multimodal.

To address these challenges, more flexible algorithms have been developed for Bayesian parameter estimation in nonlinear dynamical systems. Among these, SMC methods – and in particular, the *particle filter* – provide a powerful alternative. Particle filters (Kitagawa, 1996; Del Moral, 1997) approximate the posterior distribution by propagating a set of weighted particles through time, capturing complex, nonlinear, and non-Gaussian dynamics that would be intractable using classical filters.

The remainder of this section introduces the particle filter algorithm, which is employed in Chapter 5 of this thesis to estimate gas emission source parameters.

3.3.1 Particle filter

Particle filters (Kitagawa, 1996; Del Moral, 1997) are a class of SMC algorithms specifically designed to estimate time-varying parameters, such as source emission rates. They rely on a set of n_{prt} samples – commonly called particles and denoted as $\{\boldsymbol{\theta}^{(i)}\}_{i=1}^{n_{\text{prt}}}$ – providing a discretised approximation of the posterior $p(\boldsymbol{\theta} \mid \boldsymbol{d}_{1:t})$ at time t for data collected from time 1 to t. Each particle i possesses a weight $w_t^{(i)}$ indicating its relative plausibility given the data. Particle filters track the parameters' change through an iterative process which updates these particles. The process consists of the following steps:

1. Initialisation: At $t = 0$, draw n_{prt} particles $\{\boldsymbol{\theta}_0^{(i)}\}$ from the prior $p(\boldsymbol{\theta}_0)$, and set all weights $w_0^{(i)} = 1/n_{\text{prt}}$.

2. Prediction (Propagate): For each particle at time $t-1$, sample a new particle according to the state equation (3.3.1).

3. Update (Weight): Upon receiving the observation $\boldsymbol{d}_t$, compute a likelihood weight for each particle, based on how well that particle's state explains the measurement according to the measurement equation (3.3.2).

In practice, Steps 2 and 3 may be run multiple times while waiting for new observations to be received.

Several methods exist to propagate the particles and update their weight. Sequential importance sampling iteratively propagates a set of particles defined at initialisation and updates their weights. However, this often leads to progressively increasing variance in the particles and fewer and fewer highly weighted particles. This is referred to as particle degeneracy. To overcome it, the sequential importance resampling algorithm was developed. This avoids degeneracy by resampling particles based on their weights. This essentially kills off particles with unlikely parameter values while reproducing likely ones. As the sequential importance resampling is used in Chapter 5 to estimate gas emission location and rate, we will now describe it in more detail.

Sequential Importance Resampling. Sequential importance resampling (SIR) is built upon the sequential importance sampling framework by adding a resampling step to mitigate the problem of degeneracy, where most particles carry negligible weights over time (Doucet et al., 2000; Arulampalam et al., 2002).

At each time step, the algorithm performs the following steps:

1. Initialisation:

$$\begin{aligned}\boldsymbol{\theta}_0^{(i)} &\sim p(\boldsymbol{\theta}_0), \quad i = 1, \dots, n_{\text{prt}} \\ w_0^{(i)} &= \frac{1}{n_{\text{prt}}}\end{aligned}$$

2. Prediction (Sampling):

$$\boldsymbol{\theta}_t^{(i)} \sim p(\boldsymbol{\theta}_t \mid \boldsymbol{\theta}_{t-1}^{(i)})$$

3. Weight Update:

$$\tilde{w}_t^{(i)} = w_{t-1}^{(i)} \cdot p(\boldsymbol{d}_t \mid \boldsymbol{\theta}_t^{(i)})$$

4. Weight Normalisation:

$$w_t^{(i)} = \frac{\tilde{w}_t^{(i)}}{\sum_{i=1}^{n_{\text{prt}}} \tilde{w}_t^{(i)}}$$

5. Resampling: Resample n_{prt} particles from $\{\boldsymbol{\theta}_t^{(i)}\}$ with probabilities $\{w_t^{(i)}\}$ to obtain an equally-weighted new set. Reset weights:

$$w_t^{(i)} = \frac{1}{n_{\text{prt}}}$$

While resampling focuses computational resources on more promising particles, it also introduces its own challenges, such as particle impoverishment (loss of diversity). Particles will identify regions of high likelihood and lose their collective exploratory capabilities. If the parameter value changes too abruptly for the state evolution equation to predict it – such as an emission suddenly stopping – SIR particles will not be able to keep track and successfully estimate the parameter. There exist variants such as the auxiliary particle filter (Pitt and Shephard, 1999), resample-move methods (Berzuini and Gilks, 2001), Rao-Blackwellised (Doucet et al., 2000), and two-layer filters (Garcia-Fernandez et al., 2013), which attempt to mitigate these issues and improve performance in more challenging scenarios.

Beyond the Monte Carlo, MCMC, and particle filtering approaches discussed above, a variety of recent methods have been developed to address analytically intractable posteriors. These include variational inference techniques, which approximate the posterior through optimisation rather than sampling (Blei et al., 2017), and approximate Bayesian computation methods for likelihood-free inference (Sisson et al., 2018). Advances in sequential Monte Carlo samplers (Chopin and Papaspiliopoulos, 2020) and pseudo-marginal methods (Andrieu and Roberts, 2009) have also broadened the range of models that can be efficiently handled. More recently, normalising flows (Papamakarios et al., 2021) and other deep generative models have emerged as powerful tools for flexible posterior approximation. While these methods offer promising alternatives or complements to the techniques presented here, their detailed treatment lies beyond the scope of this thesis.

Chapter 4

Probabilistic Inversion Modelling of Gas Emissions: A Gradient-Based MCMC Estimation of Gaussian Plume Parameters

4.1 Introduction

Many inversion methods have been proposed to estimate source emission rates and locations; these can generally be grouped into optimisation (Qiu et al., 2018; Albani et al., 2020; Wang et al., 2020) and Markov chain Monte Carlo (MCMC) (Hirst et al., 2013, 2020; Ma et al., 2021; IJzermans et al., 2024) algorithms. Forward models describing how gas disperses in the atmosphere can be used to attempt to explain measured gas concentrations. As described in Chapter 3, inversion methods estimate the parameters that, given the forward model, would best describe the data collected. The most commonly used forward model for gas dispersion is the Gaussian plume model, motivated by the solution of an underlying system of partial differential equations (PDEs) (Stockie, 2011). The accuracy of the inversion is therefore closely linked to the accuracy of the forward model. The Gaussian plume model is very sensitive to the standard deviation (σ_H and σ_V) of its Gaussian concentration distributions; we will now refer to these as "wind sigma" parameters. In the literature, wind sigmas are often chosen based on the Pasquill atmospheric stability class (ASC) (Pasquill, 1961; Cui et al., 2019; Mao et al., 2020). However, estimating the exact local ASC is often difficult in practice, and misspecifying it can substantially bias the inversion estimation.

Conventionally, the practitioner estimates the ASC prior to inversion analysis for source characterisation. To estimate the ASC, the practitioner must first gather field evidence for quantities such as net solar radiation index, wind speed, etc. With this information, the current conditions can, in principle, be allocated to a given ASC. However, in practice, it is not straightforward to gather this evidence unambiguously. As a result, there is often uncertainty about the appropriate ASC for a given measurement. Secondly, once an ASC is assumed, the practitioner reads the values of four dispersion parameters, which specify "wind sigmas" for that ASC from historical tables of values (Briggs, 1973). These tables were established many decades ago from field measurements. Therefore, there is uncertainty associated with the dispersion parameter values adopted for a given ASC. As a result, the conventional practice of specifying the ASC and hence specifying fixed values of the four dispersion parameters is a source of bias in the inversion analysis of source characterisation. A more principled approach to source characterisation accommodates uncertainty in our knowledge of wind field dispersion parameters, and estimates them and their uncertainties as part of the inference.

Given that there is imprecise knowledge of the four dispersion parameters, one approach to inversion for source characterisation would be to examine a large number of potential plume models, each associated with fixed combinations of dispersion parameters, and then to use some information criterion (such as DIC, BIC, WAIC) to select the most appropriate model to retain. For example, the practitioner might perform the source inversion assuming in turn each of the possible ASCs, and then model select between them. This approach forces the practitioner to adopt a single best input value for the four dispersion parameters, which is not desirable, since it ignores uncertainty in the estimation of the dispersion parameters. Moreover, it may be the case that the set of combinations of dispersion parameters considered does not include the specific combination of dispersion parameters most relevant for the current local atmospheric conditions. The approach adopted in the current work is to estimate the dispersion parameters together with source parameters in a single Bayesian inference. This is preferable, since the uncertainty in dispersion parameters is captured naturally alongside that of source location and emission rate. Moreover, our proposed approach is computationally considerably more efficient than a grid search over dispersion parameter combinations corresponding to ASCs, followed by model selection. Note that Mao et al. (2021) attempts to optimise wind sigmas using a genetic algorithm over an ASC-based Briggs scheme parameterisation.

Another key challenge when accurately estimating sources' location and emission rate relies on adequately accounting for factors influencing the sensors' measurements. Background gas concentration and measurement error substantially influence the recorded concentration and can introduce bias in our estimation if not correctly accounted for.

Objectives. In this chapter, we propose an MCMC inversion method jointly estimating source emission rates, locations, background concentrations, measurement error variance, and wind sigmas. We demonstrate that estimating the wind sigmas is beneficial for the practising environmental modeller, in terms of improved inferences, and demonstrate how incorrect gas characterisation can be when using ASC. Our methodology is based on the principles of probabilistic inversion, which allows us to incorporate uncertainties and prior knowledge effectively.

Chapter's Layout. Section 4.2 focuses on the forward model, our framework for simulating atmospheric gas dispersions, unsteady-state wind fields, and ground sensor measurements. We employ the Gaussian plume model for gas dispersion (Stockie, 2011), the Ornstein-Uhlenbeck (OU) process for unsteady-state wind fields (Uhlenbeck and Ornstein, 1930), and simulate point sensor measurements to mimic real-world data acquisition scenarios, incorporating background concentration. Section 4.3 introduces the core of our inference methodology. We present our parameter estimation method using the manifold Metropolis-adjusted Langevin algorithm (M-MALA) in combination with Gibbs sampling. Section 4.4 presents the results from our simulation case study, where we test the robustness of our inversion method for single-source cases under varying atmospheric, data collection, and sensor layout conditions. We also show the importance of estimating wind sigmas when estimating sources' location and emission rate. In Section 4.5, we implement our inversion method on data from a real-world field experimental campaign (see Figure 4.1.1) reported by Hirst et al. (2020). This dataset contains wind field and methane measurements for controlled release trials. Finally, Section 4.6 summarises the chapter and suggests potential lines of future work.

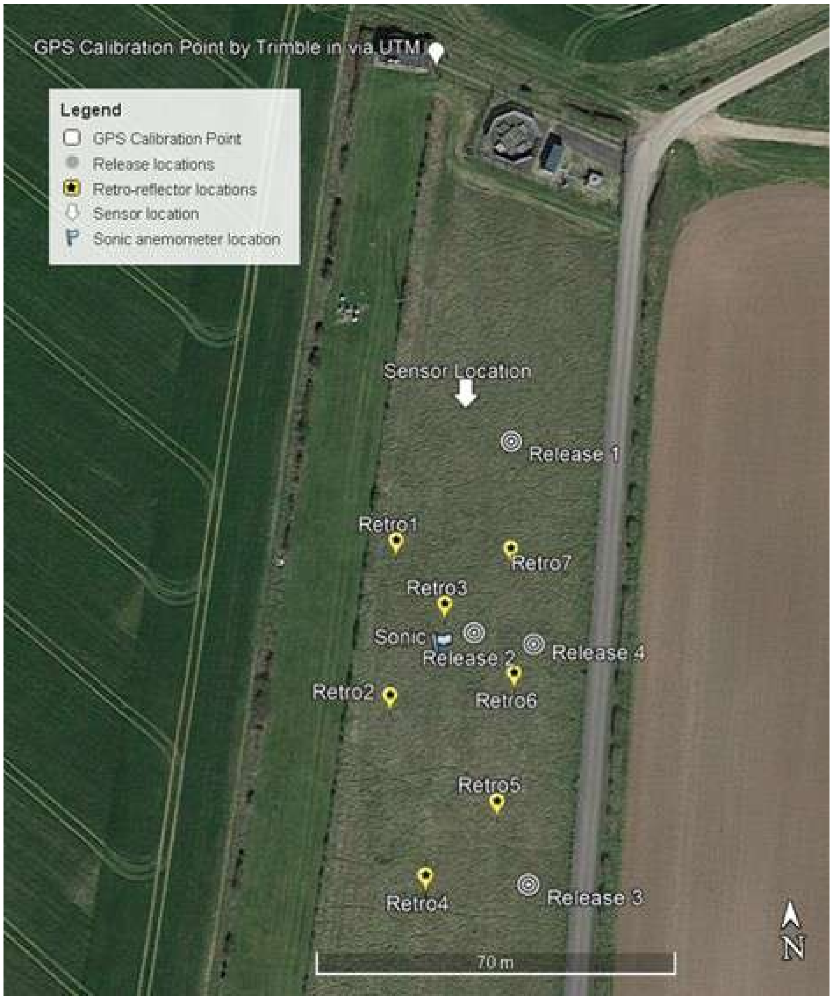


Figure 4.1.1: Methane release set up at the Chilbolton observatory, UK. A laser dispersion spectrometer measures path-averaged CH_4 concentrations between the sensor and each retros. “Sonic” indicates the emplacement of the three-dimensional ultrasonic anemometer measuring the wind speed and direction. Credit: Hirst et al. (2020) Supporting Information S1.

4.2 Atmospheric gas concentration and sensor measurements

In this section, we present the modelling framework of the simulation, incorporating the Gaussian plume model for gas dispersion (Section 4.2.1 & 4.2.2); see Figure 2.2.1 for visual representation, OU process for wind fields (Section 4.2.3), and sensor measurements when accounting for background gas concentration (Section 4.2.4). By combining these three elements, we gain a holistic perspective on air quality dynamics, enabling a deeper understanding of pollutant transport.

The formulation of the forward model sets the stage for the subsequent exploration of inversion modelling, where we aim to estimate sources' location and emission rate by leveraging the simulated sensor observations and gas dispersion patterns. We seek to estimate point sources mixed with a spatially varying background concentration.

4.2.1 Modelling gas dispersion using the Gaussian plume model

A variety of gas dispersion models have been developed, each differing in accuracy and complexity, with three primary categories being prominent. Gaussian plume models, exemplified by ISC3 (Atkinson et al., 1997), AERMOD (Cimorelli et al., 2005), and ADMS 6 (Carruthers et al., 1994), operate on the assumption of a Gaussian distribution and are widely utilised. Gaussian puff models, such as CALPUFF (Scire et al., 2000), conceptualise the plume as composed of discrete puffs, while high-fidelity computational fluid dynamics (CFD) models, like Fluidyn-Panache (Libre et al., 2011), employ rigorous numerical techniques.

In practical application, selecting the most suitable model depends on the specific requirements and resources of the modeller. Gaussian plume and puff models are often preferred due to their practicality, especially when comprehensive spatio-temporal wind field data required by CFD models are not readily available to set the initial condition and boundary conditions. Typically, wind data is collected at single points in space, limiting the applicability of CFD models.

The Gaussian plume model is noteworthy for its computational efficiency and straightforward implementation. It is a closed-form analytical expression that allows simulation of the continuous emission from a single source under the assumption of unidirectional wind flow in an unbounded space. Gaussian plume models have found widespread application in various industries, often serving as a tool for monitoring and regulating emissions from industrial projects. An example of their use can be seen in the work of Lushi and Stockie (2010), who employed a Gaussian plume model to estimate the emission rates of a large lead-zinc smelting operation in Trail, British Columbia. Similarly, Ramadan et al. (2008) utilised this model to calculate the concentration of sulfur dioxide resulting from existing power stations in Kuwait. These applications demonstrate the practical utility of the Gaussian plume model in assessing and managing the dispersion of pollutants, aiding in environmental impact assessments, urban planning, and emergency responses, among other critical areas.

In this chapter, the Gaussian plume model is used to model the dispersion of methane. We use the Gaussian plume equation described in Chapter 2 Section 2.2. For the simulation case study (Section 4.4) and the inversion on the real-world dataset (Section 4.5), we set $n_{\text{refl}} = 3$, based on previous applications' experience.

4.2.2 Parametrisation of the wind sigmas

The Gaussian plume model (2.2.4) contains atmospheric parameters that influence the shape of the plume, such as the horizontal and vertical wind sigmas, σ_H, and σ_V. These represent the standard deviation of the horizontal and vertical Gaussian distributions for gas concentration, which shape the Gaussian plume model. A large literature exists on choosing wind sigmas and originated with Pasquill's ASCs (Pasquill, 1961). Pasquill's approach first determines the local ASC using meteorological data, then uses a dispersion scheme to fix wind sigmas according to the ASC. Nowadays, ASC-based dispersion schemes remain popular in practice (Kahl and Chapman, 2018), with Briggs (Briggs, 1973), Smith (Pasquill and Smith, 1983), Pasquill-Gifford, and Chinese National Standard being common choices (Mao et al., 2020). These power-law dispersion schemes based on downwind distances fix the wind sigma parameters by selecting dispersion parameters from ASC-based tables. However, atmospheric conditions are extremely complex, and by fixing the dispersion parameters we risk misspecifying them (Finn et al., 2016). In this chapter, we present a method to estimate the wind sigmas by estimating the dispersion parameters without relying on the ASCs. We generalise the Draxler power-law parametrisation from Hirst et al. (2013) by adding dispersion parameters $a_H \in \mathbb{R}^+$, $a_V \in \mathbb{R}^+$, $b_H \in (0,1]$, and $b_V \in (0,1]$. For time $t = 1, 2, \cdots, n_T$ and fixed location (x, y, z) :

$$\begin{aligned} \sigma_{H_t} &= a_H \left(\delta_R \tan(\lambda_{H_t})\right)^{b_H} + \mu_H, \\ \sigma_{V_t} &= a_V \left(\delta_R \tan(\lambda_{V_t})\right)^{b_V} + \mu_V, \end{aligned} \tag{4.2.1}$$

where $\boldsymbol{\lambda_H} \in \mathbb{R}_+^{n_{\text{win}}}$ and $\boldsymbol{\lambda_V} \in \mathbb{R}_+^{n_{\text{win}}}$ are the 1 minute rolling window standard deviation of the horizontal and vertical wind direction time series for n_{win} windows, $\delta_R \in \mathbb{R}^+$ is the downwind distance of location (x, y, z) from the source located at $(\tilde{x}, \tilde{y}, \tilde{z})$, $\mu_H \in \mathbb{R}^+$ and $\mu_V \in \mathbb{R}^+$ are the horizontal and vertical source's aperture half-width. When the measurement location is upwind from the source, we set the Gaussian plume concentration contribution to zero. This can be seen as an extension of Cartwright et al. (2019) where where we allow the power of the wind-sigmas to be estimated. In Section 4.4, we show the impact of misspecified wind sigmas on source parameter estimation and how estimating dispersion parameters reduces this bias.

4.2.3 Simulating unsteady-state wind field using Ornstein - Uhlenbeck process

The OU process is a stochastic process often used to model the behaviour of physical systems that tend to revert toward a mean or equilibrium state (Uhlenbeck and Ornstein, 1930). When simulating wind speeds and wind directions, the OU process can be useful for generating realistic, time-varying, turbulence-free wind measurements. In the simulation study, we model the wind speed and direction as two separate stochastic processes. Their temporal evolution is modelled using an OU process with mean set to the desired average wind speed and direction. By incorporating the OU process into wind simulation models, it is possible to generate wind measurements that exhibit realistic temporal and spatial variability, which is useful for many applications such as wind energy production, air pollution dispersion modelling, and neuronal activity (Arenas-López and Badaoui, 2020; Boughton et al., 1987; Ricciardi and Sacerdote, 1979).

The OU process can be numerically simulated using the Euler-Maruyama method (Maruyama, 1955). The Euler-Maruyama scheme discretises the OU process into a series of time steps, and the stochastic differential equation governing the process is approximated using a finite-difference equation. We can therefore simulate an OU process numerically with standard deviation $\sigma_{\text{ou}} \in \mathbb{R}^+$ and correlation time $\Theta \in \mathbb{R}^+$ using:

$$\phi(t + \mathrm{d}t) = \phi(t) - \Theta \, \mathrm{d}t \, \phi(t) + \nu_t \sigma_{\text{ou}} \sqrt{2 \, \mathrm{d}t \, \Theta},$$

where ν_t is a random number sampled independently at every time-step $\mathrm{d}t \in \mathbb{R}^+$ from a standard normal distribution (Kloeden et al., 2002).

4.2.4 Point and beam sensor measurements

Our inversion model utilises measurements of atmospheric gas concentration. Different types of gas sensor platforms are available, such as satellites, aeroplanes, drones, line-of-sight/beam sensors, and point detectors (Fox et al., 2019), with each sensor type having its advantages. Point sensors can provide very high-accuracy measurements but have poor spatial coverage, whereas satellites can cover vast areas at the cost of measurement precision. This chapter focuses on ground sensors (point and beam sensors) since these are the most common techniques for continuous fence line monitoring of assets.

Ground sensors measure gas concentrations over time at fixed locations. Assuming measurement error $\boldsymbol{\epsilon}_{\text{obs}}$ and Gaussian plume model concentrations $\boldsymbol{A}\boldsymbol{s}$, the data collected can be represented by the following equation:

$$\boldsymbol{d} = \boldsymbol{A}\boldsymbol{s} + \boldsymbol{\beta} + \boldsymbol{\epsilon}_{\text{obs}}. \tag{4.2.2}$$

The n_{obs} data points collected are denoted by a $n_{\text{obs}} \times 1$ vector $\boldsymbol{d}$, while $\boldsymbol{A}$ is a coupling matrix with dimensions $n_{\text{obs}} \times n_{\text{src}}$, where n_{src} represent the maximum number of sources we allow our model to detect. The elements of $\boldsymbol{A}$ are the Gaussian plume model concentrations at the sensor locations for a unit source emission at the source location. Here, we use a Gaussian plume model. However, more accurate spatial discretisation models of the gas dispersion equations are potential alternatives, such as a finite volume method discretisation of the advection-diffusion (Moukalled et al., 2016; Calhoun and LeVeque, 2000), as discussed in Chapter 2 Section 2.3. The vector $\boldsymbol{s}$ has dimensions $n_{\text{src}} \times 1$ and contains the emission rate for each source. The spatially varying and temporally stationary – across n_T time points – background gas concentration is represented by the $n_{\text{obs}} \times 1$ vector $\boldsymbol{\beta} = \boldsymbol{\beta'} \otimes \mathbf{1}_{n_T}$, from a Gaussian field with $\boldsymbol{\beta'} \sim \text{N}(\boldsymbol{\mu}_{\boldsymbol{\beta'}}, \boldsymbol{\Sigma}_{\boldsymbol{\beta'}})$, for $\boldsymbol{\mu}_{\boldsymbol{\beta'}} \in \mathbb{R}^+$ and covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{\beta'}}$. Lastly, $\boldsymbol{\epsilon}_{\text{obs}}$ denotes the measurement error vector, where $\epsilon_{\text{obs}_k} \overset{\text{iid}}{\sim} \text{N}(0, \sigma^2_{\text{sns}})$ for $k = 1, 2, \cdots, n_{\text{obs}}$ and $\sigma^2_{\text{sns}} \in \mathbb{R}^+$.

4.3 Probabilistic inversion of gas emission problems

Building upon Section 4.2, we now explore the inversion model implemented to estimate the source locations and emission rates. By leveraging the simulated sensor observations and the knowledge of gas dispersion patterns, the inversion model offers a valuable tool for identifying and quantifying the precise source location and emission rate.

Using $i = 1, 2, \cdots, n_{\text{src}}$, $j = 1, 2, \cdots, n_{\text{sns}}$, and $t = 1, 2, \cdots, n_T$ to represent sources, sensors, and observation time points, respectively, for every pair (j, t) we have recorded measurements $\boldsymbol{d}_j = (d_{j,1}, d_{j,2}, \cdots, d_{j,n_T})^T$. Each sensor takes a measurement at time t giving a $n_{\text{obs}} \times 1$ vector of observations $\boldsymbol{d} = (\boldsymbol{d}_1^T, \boldsymbol{d}_2^T, \cdots, \boldsymbol{d}_{n_{\text{sns}}}^T)^T$. Each sensor's concentration measurements are a combination of gas emitted from the sources, background gas concentration, and measurement error variance. The sources' contributions for unit emission rates are denoted by the $n_{\text{obs}} \times n_{\text{src}}$ matrix $\boldsymbol{A}(\boldsymbol{\Xi})$, modelled using the Gaussian plume equation (2.2.4), where $\boldsymbol{\Xi} = \{\tilde{x}_i, \tilde{y}_i, \tilde{z}_i, P, n_{\text{refl}}, \mu_H, \mu_V, a_H, a_V, b_H, b_V\}$. For example, let $k = 1, \cdots, n_{\text{obs}}$, then $\boldsymbol{A}_{k,i}$ is obtained by computing Equation 2.2.4 for a specified source location $(\tilde{x}_i, \tilde{y}_i, \tilde{z}_i)$, ABL height P, number of reflections n_{refl}, horizontal and vertical source aperture μ_H, μ_V, and dispersion parameters a_H, a_V, b_H, b_V from Equation 4.2.1. Each source has an emission rate denoted s_i used to rescale the coupling matrix $\boldsymbol{A}$. Each sensor's measurements contain a different spatially varying background gas concentration $\boldsymbol{\beta}_j = (\beta_{j,1}, \beta_{j,2}, \cdots, \beta_{j,n_T})^T$ where $\beta_{j,1} = \beta_{j,2} = \cdots = \beta_{j,n_T} = \beta'_j$ giving $\boldsymbol{\beta} = (\boldsymbol{\beta}_1^T, \boldsymbol{\beta}_2^T, \cdots, \boldsymbol{\beta}_{n_{\text{sns}}}^T)^T = \boldsymbol{\beta'} \otimes \mathbf{1}_{n_T}$, for $\boldsymbol{\beta'} = (\beta'_1, \beta'_2, \cdots, \beta'_{n_{\text{sns}}})^T$.

We are interested in estimating emission rates $\boldsymbol{s} = (s_1, s_2, \cdots, s_{n_{\text{src}}})^T$ and corresponding source locations $(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}, \tilde{\boldsymbol{z}}) = ((\tilde{x}_1, \tilde{x}_2, \cdots, \tilde{x}_{n_{\text{src}}})^T, (\tilde{y}_1, \tilde{y}_2, \cdots, \tilde{y}_{n_{\text{src}}})^T, (\tilde{z}_1, \tilde{z}_2, \cdots, \tilde{z}_{n_{\text{src}}})^T)$. These are estimated simultaneously with $\boldsymbol{\beta'}$, σ^2_{sns}, and a_H, a_V, b_H, b_V to reduce bias. For simplicity, we fix the sources' height near the ground $\tilde{z}_i \approx 0$.

Inversion modelling is a powerful technique in various scientific disciplines, particularly geophysics and statistics, and aims to infer unknown parameters or variables from observed data. It involves the mathematical formulation of a forward model that simulates the observed data given a set of input parameters. Inversion modelling reverses this process by estimating the most likely parameter values that produced the observed data. MCMC methods are frequently employed in inversion modelling for their ability to explore complex, high-dimensional parameter spaces that are otherwise intractable. MCMC is particularly useful when dealing with nonlinear and non-Gaussian problems, providing robust and probabilistic estimates of model parameters while accounting for uncertainties in both data and model assumptions.

Let $\boldsymbol{\psi} = \{\boldsymbol{\theta}, \sigma^2_{\text{sns}}, \boldsymbol{\beta'}\} = \{\{\boldsymbol{s}, \tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}, a_H, b_H, a_V, b_V\}, \sigma^2_{\text{sns}}, \boldsymbol{\beta'}\}$. We can write the full posterior distribution of our inversion problem as follows:

$$p(\boldsymbol{\psi} \mid \boldsymbol{d}) \propto p(\boldsymbol{d} \mid \boldsymbol{\psi}) p(\boldsymbol{\psi}).$$

The common set of parameters for prior distributions used during the simulation case study (Section 4.4) and the inversion on the Chilbolton dataset (Section 4.5) are listed in Appendix A.1.3.

4.3.1 Gibbs sampling

Gibbs sampling (Geman and Geman, 1984) is a fundamental technique in MCMC methods, particularly advantageous in scenarios where the joint distribution is difficult to sample directly but where conditional distributions are known or can be easily calculated. When the prior and likelihood functions belong to a conjugate pair, the posterior distribution has a known analytical form. This allows for posterior samples drawn by sequentially sampling from the conditional posterior distributions. See Chapter 3 Section 3.2.1 for more information regarding Gibbs sampling. The parameters σ^2_{sns} and $\boldsymbol{\beta'}$

are estimated using Gibbs sampling with the following priors:

$$\sigma^2_{\text{sns}} \sim \text{Inv-Gamma}(a, b),$$
$$\boldsymbol{\beta'} \sim \text{N}(\boldsymbol{\mu}_{\boldsymbol{\beta'}}, \boldsymbol{\Sigma}_{\boldsymbol{\beta'}}),$$

where $a \in \mathbb{R}^+$, $b \in \mathbb{R}^+$ set using sensor's manufacturer's guidelines, and $\boldsymbol{\mu}_{\boldsymbol{\beta'}}$ set using historical average background gas concentrations and $\boldsymbol{\Sigma}_{\boldsymbol{\beta'}}$ is a diagonal matrix. The mathematical derivations of the following conjugate posteriors are provided in Appendix A.1.1:

$$\sigma^2_{\text{sns}} \mid \boldsymbol{\psi} \setminus \{\sigma^2_{\text{sns}}\} \sim \text{Inv-Gamma}\left(\frac{n_{\text{obs}}}{2} + a\ ,\ b + \frac{\sum_k^{n_{\text{obs}}} (\boldsymbol{d}_k - \boldsymbol{\beta}_k - (\boldsymbol{A}\boldsymbol{s})_k)^2}{2}\right),$$

$$\boldsymbol{\beta'} \mid \boldsymbol{\psi}\setminus\{\boldsymbol{\beta'}\} \sim \text{N}\left(\left(\frac{1}{\sigma^2_{\text{sns}}}\mathbb{I} + \boldsymbol{\Sigma}_{\boldsymbol{\beta'}}{}^{-1}\right)^{-1}\left(\frac{1}{\sigma^2_{\text{sns}}}(\boldsymbol{d'} - (\boldsymbol{A}\boldsymbol{s})') + \boldsymbol{\Sigma}_{\boldsymbol{\beta'}}{}^{-1}\boldsymbol{\mu}_{\boldsymbol{\beta'}}\right), \left(\frac{1}{\sigma^2_{\text{sns}}}\mathbb{I} + \boldsymbol{\Sigma}_{\boldsymbol{\beta'}}{}^{-1}\right)^{-1}\right).$$

Note: $\boldsymbol{d'}$ and $(\boldsymbol{A}\boldsymbol{s})'$ correspond to subsets of vectors $\boldsymbol{d}$ and $\boldsymbol{A}\boldsymbol{s}$ respectively where every n_T value was kept.

4.3.2 Manifold Metropolis-adjusted Langevin algorithm sampling

Gibbs sampling is only possible when analytical forms of the conditional posterior distribution are available. The emission rates, locations, and dispersion parameters have a nonlinear relationship, making the derivation of the conditional posteriors extremely challenging. In such cases, gradient-based MCMC methods like the Metropolis-adjusted Langevin algorithm (MALA) offer a valuable alternative to sample from the posterior (Grenander and Miller, 1994). Gradient-based MCMC leverages gradient information from the target distribution to guide proposals, leading to faster convergence and better exploration of the probability landscape.

Considering the structure of the Gaussian plume model, we expect variables to be correlated. Traditional MALA schemes rely on local gradient information, resulting in inefficient sampling in this scenario. Here, M-MALA presents compelling advancements over MALA by accounting for these interdependencies using a Riemann metric tensor to adapt to the local curvature of our target distribution (Girolami and Calderhead, 2011; Xifara et al., 2014). This ensures a more efficient and accurate sampling procedure, which has been shown to work for similar problems (Karimi et al., 2023). See Chapter 3 Section 3.2.2 for more information regarding M-MALA. Under M-MALA, sampling of $\boldsymbol{\theta}$ is performed by a Metropolis-Hastings (MH) step with the following proposal distribution:

$$\boldsymbol{\theta}^* \sim \mathrm{N}_{n_{\mathrm{par}}}\left(\boldsymbol{\theta}^{(l-1)} + 0.5\epsilon_{\mathrm{ss}}^{(l-1)}\boldsymbol{Q}^{-1}(\boldsymbol{\theta}^{(l-1)})\nabla_{\boldsymbol{\theta}^{(l-1)}}\log(p(\boldsymbol{\theta}^{(l-1)} \mid \boldsymbol{d})), \epsilon_{\mathrm{ss}}^{(l-1)}\boldsymbol{Q}^{-1}(\boldsymbol{\theta}^{(l-1)})\right),$$

where l is the current MCMC iteration, n_{par} is the number of parameters, $\boldsymbol{\theta}^*$ are the proposed parameter values, $\boldsymbol{\theta}^{(l-1)}$ are the parameter values in the Markov Chain at iteration $l-1$, $\epsilon_{\mathrm{ss}}^{(l-1)}$ is the step size at iteration $l-1$, and $\boldsymbol{Q}(\boldsymbol{\theta}^{(l-1)}) = \nabla^2\log(p(\boldsymbol{\theta}^{(l-1)} \mid \boldsymbol{d}))$ is the Hessian matrix.

However, it is essential to note that M-MALA is often computationally expensive due to the calculation of the Hessian scaling as $\mathcal{O}(n^3)$. To address this challenge and enhance computational speed, our code is implemented in JAX (Bradbury et al., 2018), a library for automatic differentiation and high-performance computing, enabling efficient sampling, gradient and Hessian computation for large-scale Bayesian inference tasks.

4.3.3 Positively constrained manifold-MALA-within-Gibbs

Combining Gibbs sampling with MH algorithms yields a hybrid approach known as MH-within-Gibbs (Chib and Greenberg, 1995); here we use M-MALA-within-Gibbs. This methodology leverages the strengths of both techniques to efficiently sample from complex posterior distributions, particularly in scenarios with correlated parameters and nonlinear relationships. The pseudocode for our implementation of M-MALA-within-Gibbs is presented in Appendix A.1.2, and the full code is available at the GitHub repository provided at the end of this chapter. We employ log transformations to enforce positivity constraints on emission rates and dispersion parameters, ensuring physically realistic parameter values throughout the sampling process. However, we do not enforce $b_H, b_V \leq 1$ in the MCMC scheme; exceptional values observed serve as an indicator of model misspecification.

4.4 Simulation study

A simulation study was conducted to assess the performance of our inversion methodology and identify its limitations. The experiments presented in this section help to understand how varying factors impact parameter estimation, demonstrating the robustness of our approach and highlighting the necessary conditions for it to perform optimally. These are fundamental steps towards applying our method to real-world data (see Section 4.5), where some factors cannot be controlled and the true parameter values are often unknown. In this section, we demonstrate our ability to simultaneously estimate the source emission rate, location, background gas concentration, measurement error variance, and dispersion parameters in single-source cases. We then highlight the importance of estimating dispersion parameters by comparing source estimations when dispersion parameters are assumed to be known or estimated.

In order to simulate the data for all experiments, we follow the steps in Section 4.2 and generate realisations of point sensor temporal observations. Parameter estimation was performed using 20,000 M-MALA-within-Gibbs iterations with initialisation values set by a coarse grid search on the emission rate and location, followed by a Latin hypercube on all parameters. The code is available at the GitHub repository listed below and was run using Python version 3.10.12 on 4 cores Intel® Xeon® Gold 6248R and 16 GB RAM. The algorithm uses fixed seed pseudorandom numbers for all MCMC samples to ensure reproducibility of results and comparability between simulations.

4.4.1 Single source estimation

Simple yet realistic single-source scenarios are useful for examining the inversion capabilities of our model. Our simulations showcase the model's ability to estimate parameters and assess its robustness to parameter variations using the following experimental design. The simulated parameter variations considered are the following: (a) WDC: wind direction coverage in anticlockwise degrees [°] with 0° pointing east, (b) DPV: dispersion parameter values, (c) SER: source emission rate [kg/s], (d) DTS: distance between the source and sensors [m], (e) OPS: number of observations per sensor, and (f) SL: sensor layout. These variations assess the robustness of our inversion methodology under different atmospheric conditions (a, b), source characteristics (c, d), and data collection conditions (e, f). For each of the six factors (a)-(f), we define low (L), medium (M), and high (H) levels as detailed in Figure 4.4.2 and Figure 4.4.3. We then perform a "main effects" analysis, changing each factor in turn from L to M and then to H, holding all other factors at level M. The level M conditions correspond to an emission source positioned at coordinates (50m, 50m, 5m) within a 110m × 110m square, emitting at a rate of 0.00039 kg/s (corresponding approximately to the Chilbolton release rates), with all plume dispersion parameters set to 1.0. A grid of 36 evenly spaced sensors positioned downwind of the plume (see Figure 4.4.1) collects 100 measurements per sensor at a frequency of 1 Hz and with a measurement error standard deviation $\sigma = 1\text{e-}3$ ppm. In practice, sensor layouts will be informed by the local prevailing wind conditions and the physical characteristics of the site. We believe the sensor setup adopted here is useful to explore the role of key design parameters on the quality of inference. An OU process simulates wind speeds with a mean of 6 m/s, and the wind direction varies every second – high frequency comparable to the controlled release trial data – encompassing a 140° range as depicted in the left plot of Figure 4.4.1. Results of the analysis are shown in Figure 4.4.2 and Figure 4.4.3 in terms of box-whisker plots summarising the marginal posterior distributions of parameters from the MCMC. In all subplots of Figure 4.4.2 and Figure 4.4.3, the middle box-whisker plot corresponds to level M for all factors. All MCMC chains used for the box-whisker plots in Figure 4.4.2 and Figure 4.4.3 are available in Appendix A.2.1–A.2.5.

Varying Atmospheric Conditions. In practice, the wind direction coverage is often positively correlated to the observation period. The longer we collect data, the higher the chances of observing a wide range of wind directions. However, a region's prevailing wind can result in narrow wind direction coverage, especially when the observation period is small, e.g. 100 seconds in this simulation. The first column in Figure 4.4.3 demonstrates the difficulty of estimating dispersion parameters when the wind direction coverage is too small, shown by large uncertainty when the wind direction covers only 60°. However, a full 360° coverage does not lead to optimal inference, due to sensors spending the majority of time outside the plume. The second column contains varying dispersion parameters and shows the model's robustness to different atmospheric conditions. In-depth studies of the impact of varying wind direction coverage are included in Appendix A.2.1. These reveal the following atmospheric conditions for our inversion method to perform optimally:

1. At least one sensor must be in the plume for the majority of the observation

period. For a given source location, this is determined by the wind directions and point sensor placements.

2. The horizontal range of wind directions must exceed the horizontal plume width. This ensures that no point sensor is always in the plume, which makes identification of the dispersion parameters difficult.

Varying Source Characteristics. Source location and emission rate are crucial when monitoring for gas emissions. It is therefore interesting to understand how these affect the inversion capability of our model. From the third column in Figure 4.4.2 and Figure 4.4.3, it is clear that an increase in the emission rate reduces our estimation uncertainty. This is likely due to a more pronounced distinction between the source contribution and the atmospheric background concentration. Similarly, the fourth column indicates a positive correlation between estimation uncertainty and the distance between the source and the sensors.

Varying Data Collection Conditions. The fifth column demonstrates the reduction in bias and estimation uncertainty as the sample size increases. The sixth column shows that the sensor layout is a fundamental factor influencing our estimation accuracy. Losing the vertical coverage in the sensor layout has significantly impacted our ability to estimate the vertical dispersion parameters. Due to the structure of the Gaussian plume model, there are positive and negative correlations between the emission rate and the dispersion parameters. To explain the observed gas measurements, a trade-off exists between the dispersion parameters and the emission rate. The former can narrow/widen the shape of the plume while the latter is decreased/increased to explain the measured concentrations. This identifiability issue is shown to be significantly influenced by the sensor layout; we explore it in more detail in Appendix A.2.5. In the next section, we illustrate the bias in source estimation when dispersion parameters are

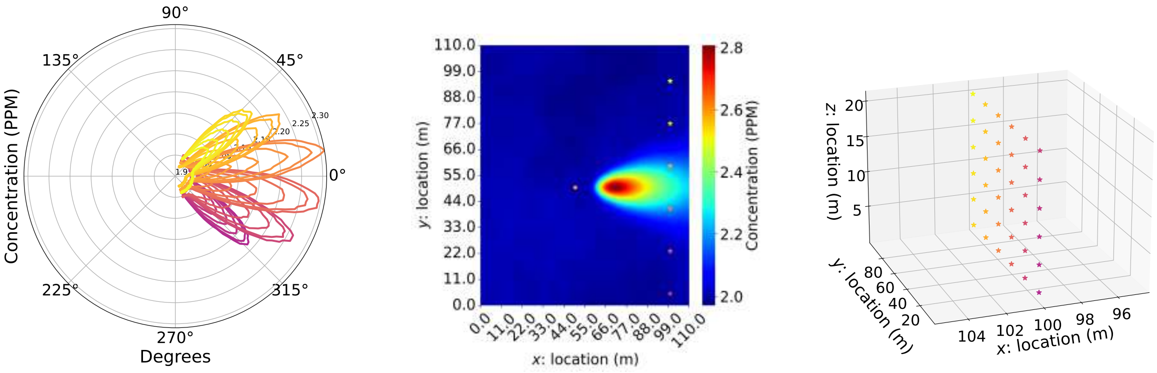


Figure 4.4.1: Simulation's level M conditions and sensor layout. The left plot represents the CH_4 concentration measurements recorded by the sensors as a function of wind direction. Each line corresponds to a different sensor. The middle plot illustrates the Gaussian plume, background CH_4 concentration, and the location of sensors at ground level for wind direction 0°. The right plot represents a grid of 36 evenly spaced point sensors.

misspecified, highlighting the importance of correctly estimating them.

Overall, the model and inversion methodology presented in Sections 4.2 and 4.3 have demonstrated the ability to estimate all parameters simultaneously and have shown general robustness to the changing atmospheric, source, and data collection conditions applied. However, both wind direction coverage and sensor layout indicate potential limitations of our approach in practice. Dispersion parameters become difficult to estimate when wind direction coverage is small or in the absence of a vertical sensor layout.

4.4.2 Estimating dispersion parameters

We now focus on a significant limitation of many gas inversion methods when applying them to real data: the misspecification of dispersion parameters. Similar work was done by Cartwright et al. (2019), who tuned the wind sigmas by estimating a horizontal and a vertical scaling parameter. The parametrisation of the wind sigmas used in their work also uses four dispersion parameters. However, these are taken from ASC-based tables, and the ASC was determined using a Monin–Obukh length and an effective

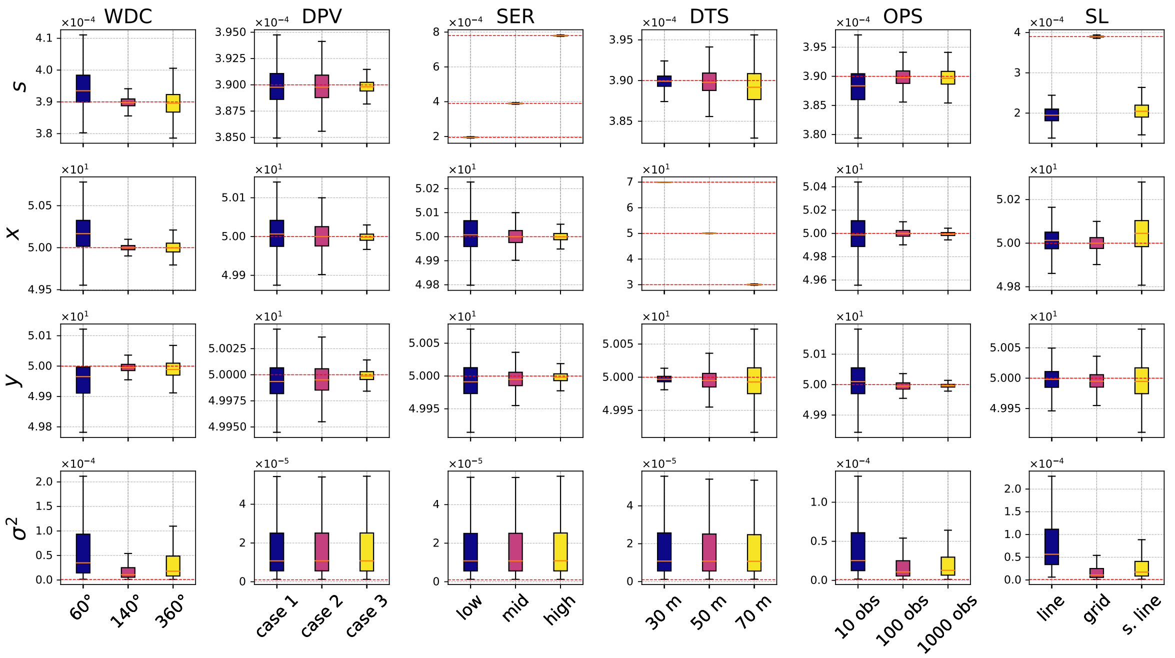


Figure 4.4.2: Parameter estimation performance main effects simulation analysis. Each column compares simulations where a single parameter was varied. The column heading indicates the parameter being varied. WDC: wind direction coverage [degrees°], DPV: dispersion parameter values, SER: source emission rate [kg/s], DTS: distance between the source and sensors [m], OPS: number of observations per sensor, and SL: sensor layout. The rows correspond to the different parameters estimated using M-MALA-within-Gibbs. **case 1**: $a_H = 1.4, b_H = 0.9, a_V = 1.2, b_V = 0.95$. **case 2**: $a_H = 1.0, b_H = 1.0, a_V = 1.0, b_V = 1.0$. **case 3**: $a_H = 0.9, b_H = 0.8, a_V = 0.7, b_V = 0.85$. **low**: 0.000195 kg/s. **mid**: 0.00039 kg/s. **high**: 0.00078 kg/s. **line**: 36 × 1 line of sensors. **grid**: 6 × 6 grid of sensors. **s.line**: 6 × 1 sparse line of sensors. The red dashed lines represent the true values of the estimated parameters.

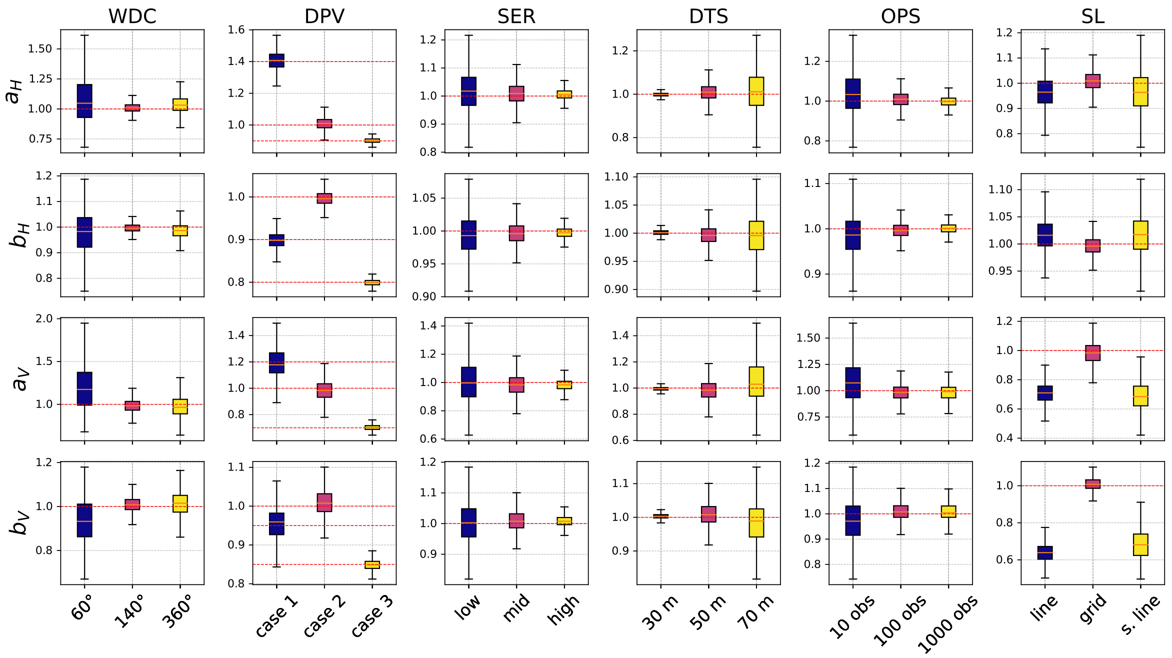


Figure 4.4.3: Parameter estimation performance main effects simulation analysis. Each column compares simulations where a single parameter was varied. The column heading indicates the parameter being varied. For the meanings of acronyms in the titles, see caption to Figure 4.4.2. The rows correspond to the different parameters estimated using M-MALA-within-Gibbs. **case 1**: $a_H = 1.4, b_H = 0.9, a_V = 1.2, b_V = 0.95$. **case 2**: $a_H = 1.0, b_H = 1.0, a_V = 1.0, b_V = 1.0$. **case 3**: $a_H = 0.9, b_H = 0.8, a_V = 0.7, b_V = 0.85$. **low**: 0.000195 kg/s. **mid**: 0.00039 kg/s. **high**: 0.00078 kg/s. **line**: 36×1 line of sensors. **grid**: 6×6 grid of sensors. **s.line**: 6×1 sparse line of sensors. The red dashed lines represent the true values of the estimated parameters.

roughness length. This approach fixes the dispersion parameters and assumes that rescaling the pairs $\{a_H, b_H\}$ and $\{a_V, b_V\}$ can sufficiently improve the Gaussian plume model. We propose additional flexibility by allowing all four dispersion parameters to be directly, individually, and jointly estimated. Therefore, removing bias introduced by the ASC, Monin-Obukh length, and effective roughness length. To the best of our knowledge, there is currently no method other than the one we propose in this chapter that simultaneously estimates source location, emission rate, background concentration, measurement error variance, and dispersion parameters. In practice, it is common for dispersion parameters to be chosen based on the local ASC. However, as shown in Figure 4.4.4, dispersion parameter misspecification can introduce substantial bias in the source estimation. We estimated the source location, emission rate, background concentration, and measurement error variance for various dispersion parameter misspecifications to quantify this bias. We misspecified each dispersion parameter at a time, a_H, a_V, b_H, b_V, while fixing the remaining three to their true value. This enables us to identify the main effect biases of each dispersion parameter. Moreover, this experimental design reflects an optimistic reality where three of the four dispersion parameters are correctly specified. In practice, we expect all four ASC-based dispersion parameters to be misspecified because the tables these come from discretise the dispersion parameters when these are in fact continuous. Additionally, the meteorological data required to correctly identify the local ASC is not always available. Parameter estimation based on misspecified dispersion parameters is compared to "est.": where all four dispersion parameters are estimated simultaneously, and to "truth": the unrealistic scenario where all four dispersion parameters have been fixed to the truth. The true values of the dispersion parameters in this simulation are $a_H = 1.0$, $a_V = 1.0$, $b_H = 0.8$, $b_V = 0.8$, and all other conditions are set to level M.

From Figure 4.4.4, it can be observed that when a_H is too small or too large, the emission rate is under- or overestimated; the same is true for a_V. The estimated source distance to the grid of sensors is overestimated when the horizontal dispersion parameters are too small or the vertical dispersion parameters are too large. Similarly, the distance is underestimated when horizontal dispersion parameters are too large and vertical dispersion parameters are too small. However, estimation of the source location coordinate y is robust to misspecification due to the sensors and source layout (see Figure 4.4.1). There is no bias in its estimation, but a reduction in uncertainty when dispersion parameters are correctly specified.

Overall, the misspecification of dispersion parameters shows a strong bias in estimating source location and emission rate. Meanwhile, estimating the dispersion parameters significantly reduces this bias, as shown by Figure 4.4.4 where the “est” and “truth” box-whisker plots are almost identical.

4.5 Chilbolton real data

Site-level gas emission monitoring is an emerging field that is gaining significance as governments and private companies increasingly prioritise emissions reporting. Given the novelty of this research area, high-quality controlled release data remains scarce, and open-source datasets are even more limited. The Chilbolton dataset is one such rare resource, which we will utilise to apply our inversion methodology.

The Chilbolton dataset contains controlled methane releases made on the flat terrains of the Chilbolton Observatory, Hampshire, UK. As these are controlled releases, the true source locations and emission rates are known. This enables comparability of our results against other methods applied on the same dataset (Hirst et al., 2020; Voss et al., 2024). The data for these controlled releases were collected using a multiple open-path laser dispersion spectrometer (LDS) and a single three-dimensional ultrasonic 20Hz anemometer; our inversion uses the anemometer data rather than modelling wind using an OU process. The LDS measures path-averaged methane concentrations

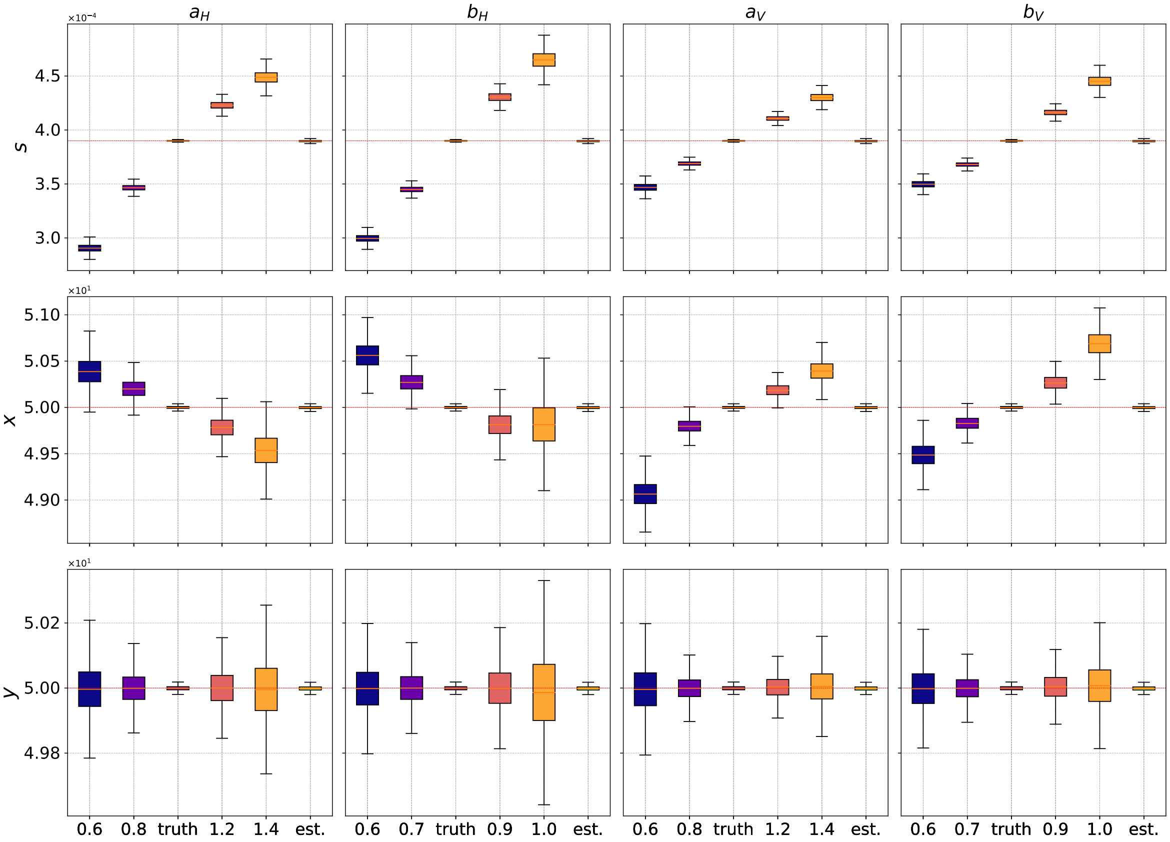


Figure 4.4.4: Quantification of dispersion parameter misspecification impact on source estimations. The true values of the estimated parameters are represented by the red dashed lines. The columns correspond to the dispersion parameters being misspecified, with the x-axis indicating the value chosen when misspecifying. These are compared to source estimations when the true dispersion parameter values are chosen, “truth”, and when all dispersion parameters are being estimated “est.”.

along beams between the LDS and seven fixed reflectors (see Figure 4.5.1), with continuous sequential scanning of all beams taking 3 seconds to cover all reflectors. To ensure compatibility between the coupling matrix's spatially discetised point concentration structure and the beam's spatially continuous path-averaged measurements, coupling matrix concentrations were averaged over 40cm spaced point locations were created along the beams. The Chilbolton experiments contain three controlled releases, two with single sources (Source 1 and Source 2), and one with two sources (Source 3 and Source 4), which we tackle using an extended two-source methodology. Each release event includes multiple sub-releases to increase wind direction coverage in the dataset. Sources were created by 2m×2m aluminium frames laid on the ground, evenly perforated with 1cm spaced holes. The Gaussian plume model is a representation of the long-term time-averaged concentration under steady-state wind conditions. Thus, over short time scales, it can be a poor representation of the observed data. Averaging the data over longer periods can improve the correspondence between the model and the (averaged) data. Consequently, measurements from each beam, taken every 3 seconds, were aggregated within 1-minute intervals before estimating the parameters. See Appendix A.3.1 for details regarding data processing.

4.5.1 Deficiencies in pre-specifying atmospheric stability classes for inversion

Using the inversion procedure presented in Section 4.3, we considered each of the four parametric descriptions of the variation of wind sigma with downwind distance and compared their estimation accuracy. In turn, these: 1) assume fixed ASC-based dispersion parameters using the Briggs scheme, 2) assume fixed ASC-based dispersion parameters using the Smith scheme, 3) estimate dispersion parameters using an ASC-free Smith scheme, and 4) estimate dispersion parameters using ASC-free Equation 4.2.1.

From now on, we refer to 3) and 4) as "estimated Smith" and "estimated Draxler", respectively.

As with all real-world datasets, the Chilbolton dataset lacks the necessary meteorological data to reliably determine the local ASC. The only available information consists of the recorded wind speeds and a photograph of the Chilbolton site taken during the releases. Based on Table A.3.3 in Appendix A.3.4, the observed wind speeds and the overcast sky possibly suggest an ASC B or C. Clearly, this remains an approximation and is insufficient to identify a single local ASC confidently, as required for ASC-based wind sigma parameterisations.

To address the absence of definitive meteorological evidence, we conducted an exploratory data analysis, which involved comparing Smith- and Briggs-based model predictions of spatial gas concentration measurements to the real data. The details of this analysis are provided in Appendix A.3.4. The results revealed significant differences between ASC-based model predictions for different classes, yet no single model consistently provided the best approximation of the observed data. For a given Chilbolton experiment (for a specific source location, monitored over multiple days), the most accurate predictions were equally distributed among the Smith B, Smith C, Briggs A, and Briggs B schemes. This lack of a clear preference is problematic, as it prevents the reliable selection of a specific ASC. In practice, this uncertainty undermines trust in ASC-based inference.

The risk of introducing bias through incorrect ASC specification presents a major challenge in gas emission estimation, highlighting the advantage of estimating dispersion parameters directly from data as part of the inversion. Further details on the Smith and Briggs schemes, as well as ASC determination, can be found in Appendix A.3.2–A.3.3.

4.5.2 Results

For each model, we ran 10,000 and 5,000 iterations of the M-MALA-within-Gibbs for Sources 1 and Source 2, respectively, with a 4,000 and 1,000 burn-in, which took on average 27 and 30 minutes for ASC-based models (estimating 11 parameters, 3 with M-MALA) and 48 and 54 minutes for estimated Smith and Draxler approaches (estimating 15 parameters, 7 with M-MALA). MCMC convergence was evaluated by investigating traceplots of estimated parameters; these are available in Appendices A.3.5 and A.3.6. Based on equivalent diagnostics, we found that the MCMC of the multiple source scenario (Source 3 and Source 4) failed to converge after 50,000 iterations, taking 1631 minutes. While brief investigations suggest this is likely due to insufficient wind direction coverage in the real data, for now, this stands as a limitation of the method. Figure 4.5.2 compares the models' estimation of Source 1 and Source 2 emission rate and location. As expected from the exploratory data analysis, we observe a significant difference between the estimation of ASC-based models. For both Sources 1 and Source 2, estimation accuracy decreases as ASC moves away from ASC A. However, both estimated Smith's and estimated Draxler's estimations are close to the truth, comparable to ASC B-based models, and outperform ASC C, D, E, and F-based models. Figure 4.5.3 provides a visual representation of the estimated source locations relative to the true source positions and beams. Figures 4.5.2 and 4.5.3 collectively illustrate that our ASC-free estimation method outperforms ASC-based models with incorrect ASC specification. Therefore, alleviating the risk of model misspecification associated with using ASC-based methods. While the "estimated Smith" and "estimated Draxler" inversions each took approximately 50 minutes to run, executing all ASC-based Briggs and Smith schemes required 168 and 84 minutes, respectively.

We now compare estimated models using the Bayesian information criterion (BIC) (Schwarz, 1978), and assess model goodness of fit using methane concentration root

		Briggs						Smith			Estimated	
		A	B	C	D	E	F	B	C	D	Smith	Draxler
Source 1	BIC	2048	2189	2450	2667	3181	3502	**2014**	2021	2167	2025	2058
	RMSE	0.63	0.68	0.76	0.87	1.12	1.29	0.62	0.66	0.90	**0.61**	0.62
Source 2	BIC	4386	4537	5212	5707	6323	6543	4541	4343	4412	4223	**3539**
	RMSE	0.58	0.60	0.69	0.76	0.86	0.90	0.60	0.61	0.67	0.55	**0.48**

Table 4.5.1: Comparing models' inversion performances for Source 1 and Source 2 using BIC and methane concentrations RMSE. The best performing model for each performance measure is given in bold. "Estimated Smith" and "Estimated Draxler" estimate the dispersion parameters instead of using ASC-based tables.

mean square error (RMSE). Table 4.5.1 gives values of BIC and RMSE for each model considered. In terms of RMSE, Source 1 was best estimated using estimated Smith inference, but Smith B inference was preferred when penalising for model complexity using BIC. For Source 2, estimated Draxler inference produced the lowest RMSE and BIC, markedly better than all competitors. Overall, ASC-free models consistently yield the lowest RMSE inversions and, when accounting for model complexity, achieve the lowest BIC for Source 2 and a close second for Source 1.

We conclude from the Chilbolton field study that estimating dispersion parameters as part of the inversion inference avoids the problematic selection of a single ASC, whilst providing accurate and computationally efficient source characterisation.

4.6 Discussions

We consider a Gaussian plume forward model for atmospheric gas dispersion, parameterised inter alia by dispersion parameters known colloquially as "wind sigmas", related to atmospheric stability classes (ASCs). We propose an MCMC inversion scheme to estimate the gas source emission rate, location, gas background concentration, measurement error variance, as well as dispersion parameters. When the ASC is known a priori, our results show that ASC-based inference (assuming known ASC) and ASC-free

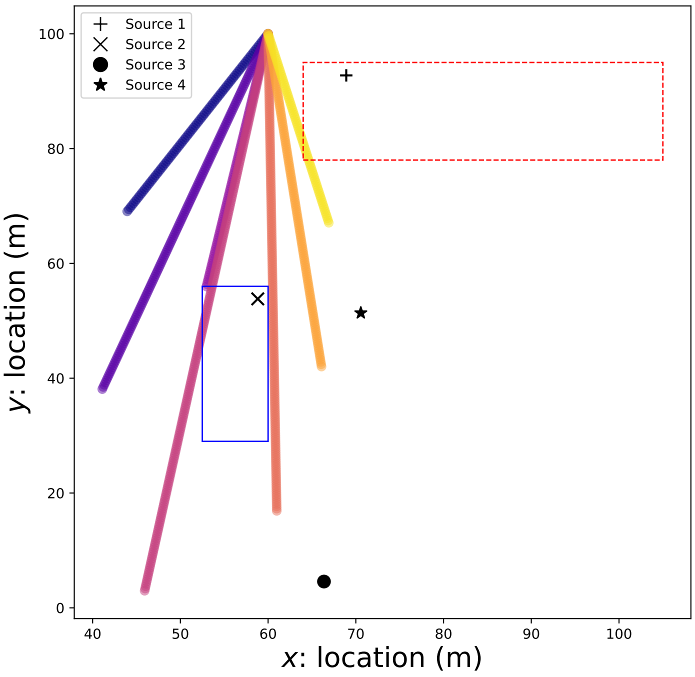


Figure 4.5.1: Sensor, beam, and source positions for the Chilbolton experiment. Each colored line corresponds to a different sensor-reflector path. The straight blue line and dashed red line boxes correspond respectively to the plotting area of Source 1 and Source 2 in Figure 4.5.3.

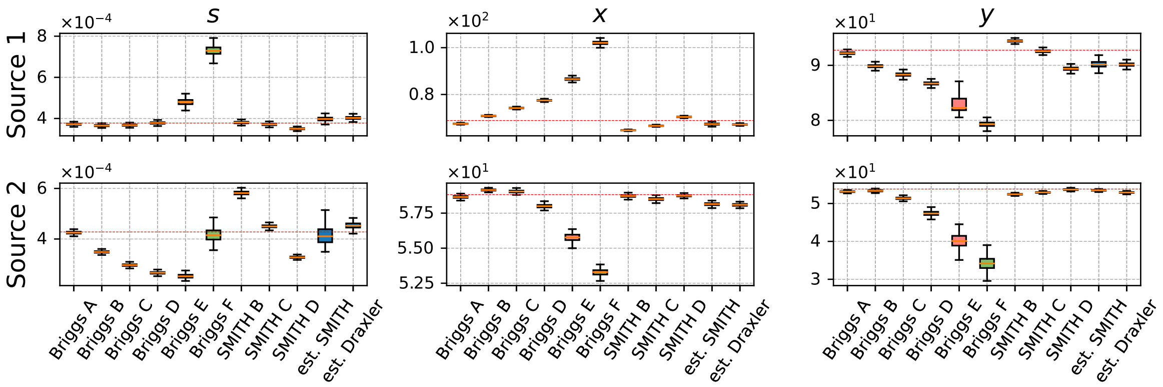


Figure 4.5.2: Source 1 and Source 2 emission rate and location estimations for all ASC-based models tested, estimated Smith, and estimated Draxler. The red dashed line represents the true sources' location and rate.

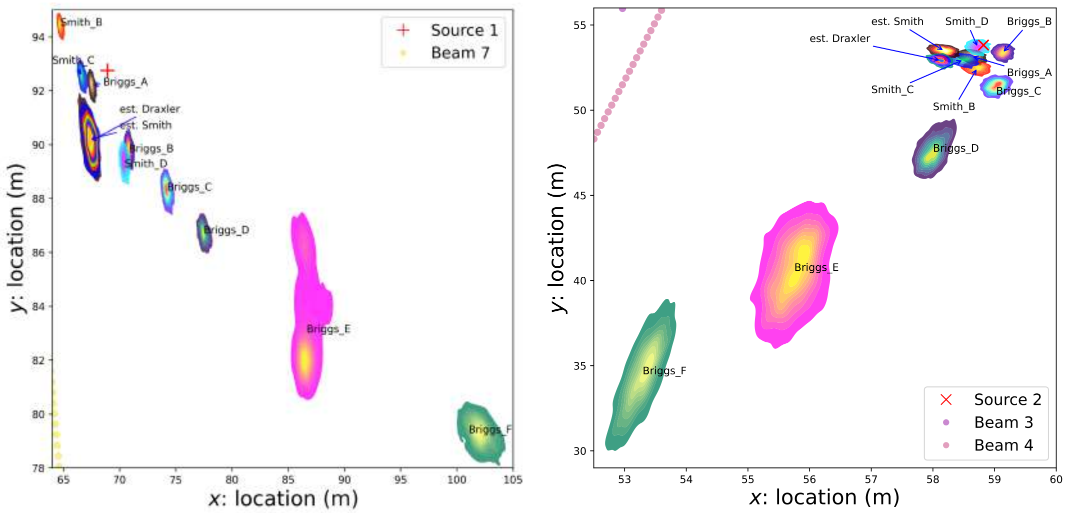


Figure 4.5.3: Source 1 (left) and Source 2 (right) location estimation density contours from all ASC-based models tested, estimated Smith, and estimated Draxler. Note: different scales for the x and y axes.

inference (estimating dispersion parameters) are of similar quality. However, determining the ASC accurately in practical applications is challenging. As a result, ASC-based models are prone to bias and high computational costs. In contrast, estimating dispersion parameters as part of inversion inference reduces bias and offers more reliable source characterisation.

Whenever possible, plume dispersion parameters should therefore be estimated rather than fixed a priori, whilst also exploiting any prior information about possible relevant ASCs, obtained from meteorological and other information from the measurement campaign. The simulation study in Section 4.4 shows that this is only feasible when data are recorded using an appropriate sensor layout and sufficient wind direction coverage. A large literature exists demonstrating the importance of appropriate sensor layouts in inverse problems (Liu et al., 2022; Liu and Li, 2022; Dia et al., 2024). These conditions ensure optimal inversion estimation and should therefore serve as guidelines when installing ground sensors and collecting data for monitoring purposes. Additionally, the simulation study demonstrated the robustness of our inversion methodology to a wide range of atmospheric, source, and data collection conditions. In Section 4.5, we proved the effectiveness of our method in practice by applying it to real data.

A number of relatively straightforward extensions of the model are possible. The observation Equation 4.2.2 can be adapted so that the background gas concentration, the sensor measurement errors, and gas dispersion are represented in a more physically realistic way. For example, background gas concentration $\boldsymbol{\beta}$ could be modelled using a spatio-temporal Gaussian process or Gaussian Markov random field, potentially accounting for wind field-induced dependence. Under this assumption, it is important to jointly estimate $\boldsymbol{\beta}$ and the measurement error variance σ^2_{sns}. Additionally, the assumption that sensor measurement errors are independently and identically normally distributed may be relaxed to include serially correlated errors and modelling sporadic error spikes. In the current work, the source is assumed to be located near the ground $\tilde{z} \approx 0$, appropriate for many applications. This assumption can, of course, be relaxed to include estimation of source elevation. Source horizontal and vertical aperture half-widths $\{\mu_H, \mu_V\}$ can be estimated similarly. We could also consider non-spherical sources.

Furthermore, the coupling matrix $\boldsymbol{A}$ could be computed using a more physically realistic forward model. Kennedy and O'Hagan (2001) consider emulation of a plume model for atmospheric dispersion of radionuclides, in their application of Bayesian calibration of computer models. In the application, the forward model, assumed to be a Gaussian plume, is characterised by a Gaussian process, required to predict the outputs of the computer model, namely remote concentrations of radionuclides for given inputs, namely source strength, wind speed, direction, and plume "wind sigmas" for an assumed known source location. This approach could be extended to account for unknown source locations. Kennedy and O'Hagan (2001) also consider a discrepancy model to correct the output of their Gaussian process emulated to observations, thereby accommodating a degree of flexibility within the statistical model to a misspecified Gaussian plume. We also note literature on non-Bayesian calibration methods, such as L_2-based frequentist approaches. Here, calibration parameters are estimated by minimising the squared difference between model outputs and observations. This is often achieved by estimating discrepancy nonparametrically and then optimising parameters via least squares (Wong et al., 2017; Tuo, 2019; Plumlee, 2019). These methods provide computationally efficient point estimates with strong asymptotic guarantees, without requiring priors, but rely on bootstrap or asymptotic approximations for uncertainty quantification. We could go a step further and account for obstacles in the flow field. Traditional numerical methods discussed in Chapter 2 Section 2.3 have been extensively studied for this purpose.

A grid-based version of our method was originally considered, with the centre of the grid cell serving as a potential source location. In practice, the number of sources is expected to be small compared to the number of cells; our methodology, therefore, incorporated a spike and slab prior on the emission rates to constrain the number of cells corresponding to sources. However, this method was abandoned due to its computational cost. Assuming cell-centred sources introduces bias in the parameter estimation, which can be reduced by increasing the grid's resolution. Unfortunately, using a fine grid creates a high-dimensional inversion problem with strong correlations between parameters. Nonetheless, we believe it would be interesting to use the gridded approach, with a computationally cheaper non-Hessian-based MCMC method, to identify the number of sources and their emission rates. These estimates could then be used as starting solutions for our inversion method. See, for example, Van de Kerkhof et al. (2024) and Hirst et al. (2013).

Finally, for practical online source monitoring applications, it is critical to accommodate temporal variations in source, background, and dispersion parameter characteristics, due to effects of e.g. weather conditions, human activities, and seasonality. To account for these temporal variations, we believe extending our work to state-space models whilst enforcing source sparsity would be an exciting area of research. Voss et al.

(2024) has presented promising results for such approaches on the Chilbolton dataset.

4.7 Data and code availability

4.7.1 Data availability statement

The raw Chilbolton data that support the findings of this chapter are openly available at the following: https://edata.stfc.ac.uk/items/5c88d121-0e19-4840-a26b-499dba49419a.

4.7.2 Code availability statement

To further promote accessibility and reproducibility, we released a Python package on PyPI (https://pypi.org/project/sourceinversion/) and made the documented code openly available on GitHub (https://github.com/NewmanTHP/Probabilistic-Inversion-Modeling-of-Gas-Emissions).

Chapter 5

Deep Learning Surrogates for Real-Time Gas Emission Inversion

5.1 Introduction

The inversion methodologies discussed so far assume that the source emission rate and location are spatio-temporally constant within the monitored domain. In many practical settings, however, multiple emission sources may be present – for example at landfill sites (Scheutz et al., 2011) – and individual sources may vary in time. In such cases, inversion techniques capable of resolving fully spatio-temporal emission patterns are required.

In this chapter, we focus on ground-level sensor networks delivering high-frequency, spatially sparse measurements. A Bayesian inversion framework, traditionally relying on MCMC (Brooks et al., 2011; Robert et al., 1999; Fearnhead et al., 2024), quantifies uncertainty robustly but suffers from poor scalability and slow convergence in multimodal, time-varying settings. Real-time inference necessitates an alternative that balances accuracy and efficiency.

Objectives. We embed a CFD surrogate within a Bayesian state-space model and perform sequential inference using an SIR particle filter (Gordon et al., 1993); see Chapter 3 Section 3.3 for more information regarding SIR. Our surrogate employs an MLP trained to emulate high-fidelity CFD solvers, providing near-instantaneous predictions of sensor concentrations for any candidate source configuration; see Chapter 2 Section 2.4 for more information regarding MLP. This approach retains the physical realism of numerical solvers while reducing per-evaluation cost to milliseconds. In this chapter, we apply our inversion methodology to the Chilbolton controlled-release dataset (Hirst et al., 2018, 2020; Voss et al., 2024; Newman et al., 2025), demonstrating that our surrogate-based SIR filter achieves comparable accuracy to full CFD and Gaussian plume models at a fraction of the computational cost. We further validate robustness under simulated obstructed flow fields and temporally varying emission rates, highlighting the scalability of our framework.

Chapter's Layout. Section 5.2 presents the SIR-based inversion algorithm. Section 5.3 details the MLP surrogate construction. Section 5.4 evaluates performance on Chilbolton data, and Section 5.5 extends the approach to complex, obstructed scenarios.

5.2 Spatio-temporal gas source inversion using particle filters

We cast the gas source inversion problem in a Bayesian state-space framework, which is commonly used in a range of areas, including econometrics (Aoki and Havenner, 1991; Hamilton, 1994), target tracking (Arulampalam et al., 2002; Nemeth et al., 2012; Stone et al., 2013) and epidemiology (Prashad, 2025; Hooker et al., 2011). Let the latent state at time t be $\boldsymbol{\theta}_t$, which encodes the unknown source parameters – notably the possibly time-varying source coordinates $(\tilde{x}, \tilde{y}, \tilde{z})_t$ and emission rate s_t. We specify a prior distribution $p(\boldsymbol{\theta}_0)$ over the initial state (for example, a uniform prior over the site for the source location and a broad prior for the emission rate) to capture our initial uncertainty. The state-space model is then defined by two components: a *measurement model*, which relates the state to the observed gas concentrations, and a *state evolution model* describing how the latent state changes over time.

Measurement Model (Observation Equation). At any time t, we receive sensor measurements $\boldsymbol{d}_t$ (e.g. gas concentration readings at fixed sensor locations). We model these observations as noisy functions of the current source parameters. In particular, we assume the observation equation:

$$\hat{\boldsymbol{d}}_t = C(\dot{\boldsymbol{x}}, \dot{\boldsymbol{y}}, \dot{\boldsymbol{z}} \mid (\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}, \tilde{\boldsymbol{z}})_{\kappa:t})_t \times \boldsymbol{s}_{\kappa:t} + \boldsymbol{\beta}_t + \boldsymbol{\epsilon}_{\text{obs}_t}, \tag{5.2.1}$$

where $(\dot{x}, \dot{y}, \dot{z})$ are the known coordinates of the sensor location(s). $C(\dot{x}, \dot{y}, \dot{z} \mid (\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}, \tilde{\boldsymbol{z}})_{\kappa:t})_t$ is the gas concentration function (derived from a CFD model or its surrogate) that predicts the concentration at time t at the sensors given a history of source location $(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}, \tilde{\boldsymbol{z}})_{\kappa:t}$ and $\boldsymbol{s}_{\kappa:t}$ denotes the history of the source's emission rate. $t - \kappa$ up to t, represent the fact that in a local/regional settings gas concentration at time t can depend on emission rate and location in the recent past (within a window of length κ). $\boldsymbol{\beta}_t$ is the ambient background gas concentration at the sensor (e.g. baseline methane levels) at time t. The final term $\boldsymbol{\epsilon}_{\text{obs}_t}$ represents measurement noise (sensor error), which we typically model as independent zero-mean Gaussian noise. If multiple sensors are deployed, $\boldsymbol{d}_t$ is a vector of all sensor readings at time t, and we assume the components of $\boldsymbol{\epsilon}_{\text{obs}_t}$ are independent (i.e., each sensor has independent noise). For simplicity, we also assume the source's influence on the flow field is negligible (i.e. the wind field is not altered by the emission), so that the dispersion of gas is linearly related to the emission rate. This means C can be computed for a unit emission and then scaled by $\boldsymbol{s}_{\kappa:t}$, as reflected in (5.2.1).

State Evolution Model (Dynamic Equation). We allow the source parameters to evolve in time according to a latent dynamics model. In general, the source location might be static (or moving slowly), and the emission rate s_t could potentially vary over time. We capture any uncertainty or evolution in these parameters with a stochastic dynamic equation. A simple choice (which we adopt here) is a random-walk model:

$$\boldsymbol{\theta}_t = \boldsymbol{\theta}_{t-1} + \boldsymbol{\epsilon}_{\text{evo}_t}, \tag{5.2.2}$$

where $\boldsymbol{\epsilon}_{\text{evo}_t} \sim \mathrm{N}(\mathbf{0}, \mathbf{W})$ is a multivariate Gaussian process noise term with covariance

$\mathbf{W}$. This model assumes that from one time step to the next, the source location and emission rate do not change dramatically, only undergoing small random perturbations. In practice, if the true source is stationary (constant location and emission), the random walk (with a small covariance $\mathbf{W}$) serves to maintain diversity in our particle filter (preventing all particles from collapsing to a single point). If the source emission rate genuinely varies over time, $\mathbf{W}$ can be tuned to account for those changes. We assume the state process $\boldsymbol{\theta}_t$ is Markovian (i.e. given $\boldsymbol{\theta}_{t-1}$, the next state $\boldsymbol{\theta}_t$ is independent of earlier times) and that process noise $\boldsymbol{\epsilon}_{\mathrm{evo}_t}$ is independent across time steps. We also assume observations are conditionally independent given the corresponding state (with the caveat that $\boldsymbol{d}_t$ may depend on a short history of emissions rates $\boldsymbol{s}_{\kappa:t}$ and locations $(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}, \tilde{\boldsymbol{z}})_{\kappa:t}$, which can be incorporated by extending the state to include recent emissions up to κ). Under these assumptions, the model is a state-space model amenable to Bayesian filtering techniques to recover the latent process $\boldsymbol{\theta}_{1:t}$.

Bayesian Filtering. Our goal is to infer the posterior distribution of the source parameters given the sequence of measurements up to time t, denoted $p(\boldsymbol{\theta}_t \mid \boldsymbol{d}_{1:t})$. Using Bayes' rule and the state-space model assumptions, the posterior can be updated sequentially: starting from a prior $p(\boldsymbol{\theta}_0)$, we incorporate new observations as they arrive. As seen in Chapter 3 Section 3.3, the update from time $t-1$ to t is given by:

- **Prediction:** $p(\boldsymbol{\theta}_t \mid \boldsymbol{d}_{1:t-1}) = \int p(\boldsymbol{\theta}_t \mid \boldsymbol{\theta}_{t-1}) p(\boldsymbol{\theta}_{t-1} \mid \boldsymbol{d}_{1:t-1}) \mathrm{d}\boldsymbol{\theta}_{t-1}$,
- **Bayes Update:** $p(\boldsymbol{\theta}_t \mid \boldsymbol{d}_{1:t}) \propto p(\boldsymbol{d}_t \mid \boldsymbol{\theta}_t) p(\boldsymbol{\theta}_t \mid \boldsymbol{d}_{1:t-1})$.

Here, $p(\boldsymbol{d}_t \mid \boldsymbol{\theta}_t)$ is the likelihood of the new observation given state $\boldsymbol{\theta}_t$, which from (5.2.1) (assuming Gaussian sensor noise) can be written as, for example, $p(\boldsymbol{d}_t \mid \boldsymbol{\theta}_t) = \mathrm{N}\big(\boldsymbol{d}_t \big| C(\dot{x}, \dot{y}, \dot{z} \mid (\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}, \tilde{\boldsymbol{z}})_{\kappa:t})_t \times \boldsymbol{s}_{\kappa:t} + \boldsymbol{\beta}_t, \sigma^2_{\mathrm{sns}}\big)$ for some noise variance σ^2_{sns}. In general, these integrals and proportionality are intractable to solve in closed form due to the nonlinearity of C and the high dimensionality of the state. We therefore resort to a

Monte Carlo approximation, specifically, the SIR particle filter (Gordon et al., 1993), to perform the Bayesian update numerically.

In a SIR particle filter, we maintain a set of n_{prt} random samples (particles) $\{\boldsymbol{\theta}_t^{(i)}\}_{i=1}^{n_{\text{prt}}}$ that provides a discrete approximation of the posterior $p(\boldsymbol{\theta}_t \mid \boldsymbol{d}_{1:t})$. Each particle $\boldsymbol{\theta}_t^{(i)} = \{\tilde{x}_t^{(i)}, \tilde{y}_t^{(i)}, \tilde{z}_t^{(i)}, s_t^{(i)}\}$ is a possible source location and emission rate. We also associate a weight $w_t^{(i)}$ with each particle, indicating its relative plausibility given the data. The particle filter sequentially propagates and updates these weighted samples as new data arrive. Over time, the particle ensemble $\{\boldsymbol{\theta}_t^{(i)}, w_t^{(i)}\}_{i=1}^{n_{\text{prt}}}$ evolves to track the posterior distribution $p(\boldsymbol{\theta}_t \mid \boldsymbol{d}_{1:t})$. In effect, the particle filter provides a numerical approximation of the Bayesian solution for the gas source inversion problem. This approach is well-suited to our setting as it can handle nonlinear and non-Gaussian relationships (unlike, e.g., a Kalman filter (Kalman, 1960)) and it naturally accommodates the sequential arrival of data in a dynamic environment. By using a sufficiently large number of particles, the SIR filter can approximate the true posterior to any desired accuracy, enabling robust spatio-temporal estimation of the source location and emission rate even under complex, unsteady flow conditions.

5.3 Multilayer perceptron surrogate modelling of atmospheric gas measurements in unsteady-state flow fields

The particle filter outlined in Section 5.2 requires repeated evaluations of the likelihood $p(\boldsymbol{d}_{1:t} \mid \boldsymbol{\theta}_t)$ for each particle $\{\boldsymbol{\theta}_t^{(i)}\}_{i=1}^{n_{\text{prt}}}$, which in the case of the gas concentration model, requires computing the concentration function $C(\dot{x}, \dot{y}, \dot{z} \mid (\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}, \tilde{\boldsymbol{z}})_{\kappa:t}^{(i)})_t$ at each potential source location. High-fidelity CFD models exist to compute C, but they are often too slow to run for each particle i and time step t in real-time. We can overcome this bottleneck by utilising a deep learning surrogate model for the CFD simulation. Specifically, we train a MLP to approximate the mapping from the source parameters to the sensor measurements, effectively serving as a fast emulator of the physical gas dispersion model.

5.3.1 High-fidelity gas dispersion simulation training data

To generate training data for the surrogate, we first require a ground-truth model of how gas disperses in an unsteady flow field. We use a CFD solver (Holl and Thuerey, 2024) that captures the physics of air flow and gas transport in our monitored site Ω. In particular, we solve the time-dependent Navier-Stokes equations (2.1.2) (which govern fluid flow) to obtain the wind velocity field, and then solve the advection-diffusion equation (2.1.1) (which governs the transport and diffusion of the gas) to obtain gas concentrations. These equations are discretised and integrated over time to simulate the evolution of wind and gas in the domain. Denoting by f_v the Navier-Stokes solver and f_c the advection-diffusion solver, we can formalise the process as follows: given a history of wind boundary conditions (e.g. measured wind speed and direction over time) $\boldsymbol{u}_{\kappa:t}$ and corresponding pressure field $\boldsymbol{p}_{\kappa:t}$, and given any fixed obstacles or terrain features ω over the spatial domain Ω, the CFD model produces a *flow field* $f_v(\boldsymbol{u}_{\kappa:t}, \boldsymbol{p}_{\kappa:t}, \omega, \Omega)$ describing the wind velocities in Ω over time. Using the flow field, the gas transport solver f_c computes the resulting gas concentration field for a source at location history $(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}, \tilde{\boldsymbol{z}})_{\kappa:t}$. We then evaluate this concentration field at the sensor coordinates $(\dot{x}, \dot{y}, \dot{z})$. Let C_{ns} denote the concentration output of the full Navier-Stokes-based numerical solver. We can express the solver's prediction as:

$$C_{\text{ns}}(\dot{x}, \dot{y}, \dot{z} \mid (\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}, \tilde{\boldsymbol{z}})_{\kappa:t}) = f_c((\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}, \tilde{\boldsymbol{z}})_{\kappa:t}, f_v(\boldsymbol{u}_{\kappa:t}, \boldsymbol{p}_{\kappa:t}, \omega, \Omega))\Big|_{\dot{x},\dot{y},\dot{z}}, \tag{5.3.1}$$

which represents the gas concentration at time t at the sensor location due to a source at $(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}, \tilde{\boldsymbol{z}})_{\kappa:t}$ under the given unsteady wind conditions. In other words, we use the CFD solver to simulate the propagation of gas from a candidate source through the evolving wind field, and we record what concentration would be measured at the sensor. Equation 5.3.1 is essentially the model behind the concentration function C in (5.2.1). This high-fidelity simulation accounts for complex effects, such as turbulent eddies and time-varying wind direction, providing accurate ground truth concentrations for given source parameters. Additional information regarding the numerical solver-based atmospheric gas transport simulation is provided in Appendix B.1.

However, running such a CFD simulation for every candidate source is computationally expensive. For example, solving the Navier-Stokes and advection-diffusion equations even once (for a given source configuration) might take seconds to hours, which is prohibitively expensive when deployed within a particle filter that could require thousands of evaluations. Therefore, we will use (5.3.1) offline to generate training datasets, and then train a fast MLP surrogate to mimic its output.

To construct the training data, we sample a large number n_{sim} of hypothetical source scenarios and simulate each with the CFD model. In our study, we assume source locations to be fix over the simulation period and drew locations uniformly from the area of interest Ω (each location $(\tilde{x}, \tilde{y}, \tilde{z})$ corresponds to a different training sample). For each source location, we assume a fixed emission rate (e.g. a unit emission for simplicity) and run the CFD solver to obtain $C_{\text{ns}}(\dot{x}, \dot{y}, \dot{z} \mid \tilde{x}, \tilde{y}, \tilde{z})$ via (5.3.1). All simulations use the same physical environment and flow conditions representative of the scenario we care about. In particular, we leverage data from the Chilbolton experiment (see Section 5.4): a time series of wind measurements (from an anemometer) provides the unsteady wind boundary conditions $\boldsymbol{u}_{\kappa:t}$ for the solver, and the site is relatively flat and unobstructed (no ω features), which allows us to simplify the simulation. Because the vertical variation in this experiment was minimal, we performed the CFD simulations in two dimensions (assuming all sources and sensors lie in the same horizontal plane). This two-dimensional approximation greatly reduces computational cost while introducing only small errors for a flat site. We use the recorded time-varying wind profile uniformly across the domain (spatially uniform but temporally varying wind) when solving the Navier–Stokes equations, given the small size of the site. In summary, our dataset consists of many pairs $\{(\tilde{x}^{(k)}, \tilde{y}^{(k)}), \boldsymbol{d}_t^{(k)}\}$ for $k = 1, \cdots, n_{\text{sim}}$, where $(\tilde{x}^{(k)}, \tilde{y}^{(k)})$ is a sampled source location and $\boldsymbol{d}_t^{(k)} = C_{\text{ns}}(\dot{x}, \dot{y} \mid \tilde{x}^{(k)}, \tilde{y}^{(k)})_t$ is the corresponding sensor reading at time t produced by the CFD simulation (for a given wind sequence and unit emission). These synthetic data samples form the ground truth that our MLP will learn to emulate.

5.3.2 MLP surrogate model: architecture and training

We design an MLP to serve as a surrogate for the CFD-based concentration function. The MLP is a fully-connected feed-forward neural network that takes the source location as input and outputs the predicted gas concentration at the sensor. In our case, the input vector to the MLP is $(\tilde{x}, \tilde{y})$ (the two-dimensional coordinates of a potential source). The output is the predicted sensor measurement $\boldsymbol{d}_t$ at time t (or a vector of concentrations if multiple sensors are present; one output per sensor). Because the relationship from source location to sensor concentration can be quite complex (highly nonlinear due to the physics of dispersion), we choose a sufficiently expressive network architecture. For the real data surrogate modelling in Section 5.4, we employ an MLP with four hidden layers and 100 neurons per hidden layer. In the case study involving obstacles in the flow field in Section 5.5, the mapping from source location to sensor measurements is more complex, a deeper architecture is used, consisting of five hidden layers with 500 neurons per hidden layer. We employ SeLU activations (Klambauer et al., 2017) at the hidden layers (SeLU: Scaled Exponential Linear Unit, designed to self-normalise the neural network while avoiding exploding/vanishing gradients and dying neurons), and a linear activation at the output layer (since we are performing a regression to predict a continuous concentration value). We initialise the network weights using standard Xavier initialisation (Glorot and Bengio, 2010) and train them to minimise the error between the MLP's predictions and the true concentrations from the CFD simulations.

Training. We use a supervised learning approach to train the MLP on the dataset of simulated source scenarios. We define a loss function $\mathcal{L}$ as the Mean Squared Error (MSE) between the MLP's prediction C_{MLP} and the ground-truth solver output C_{ns} over all training samples. Formally, if the training set is $\{(\tilde{x}^{(k)}, \tilde{y}^{(k)}, \boldsymbol{d}_t^{(k)})\}_{k=1}^{n_{\text{sim}}}$ (with $\boldsymbol{d}_t^{(k)} = C_{\text{ns}}(\dot{x}, \dot{y} \mid \tilde{x}^{(k)}, \tilde{y}^{(k)})_t$, as above), the loss is:

$$\mathcal{L}(\boldsymbol{\Theta}) = \frac{1}{n_{\text{sim}}} \sum_{k=1}^{n_{\text{sim}}} \left\| C_{\text{MLP}}\left(\tilde{x}^{(k)}, \tilde{y}^{(k)}; \boldsymbol{\Theta}\right) - \boldsymbol{d}_t^{(k)} \right\|^2,$$

where $\boldsymbol{\Theta}$ denotes the learnable weights of the network. We minimise this loss using stochastic gradient descent. The training is run for enough epochs until the error plateaus without overfitting, which in our experiments was on the order of a few tens of thousands of epochs. After training, we obtain an approximate functional mapping C_{MLP} that is a fast proxy for the CFD solver's output C_{ns}, meaning the MLP's prediction of sensor measurements for a given source is nearly the same as the high-fidelity CFD prediction (5.3.1), but can be computed instantaneously. Once this surrogate is trained, it can be plugged into the particle filter. Whenever we need to evaluate the likelihood of a particle (i.e. compute the expected sensor reading for a potential source), we use the MLP instead of running a CFD simulation.

Practical Considerations. The flow conditions are time-varying, so the mapping from source location to sensor reading could drift over long periods as winds change. To handle the temporal non-stationarity, we adopt a sliding time-window approach in training the surrogate. Instead of training a single MLP on the entire duration of data (which might force it to average over different wind regimes), we train separate MLP models for consecutive time-windows of the data. For example, we can segment the simulation (and real data) into shorter intervals (each spanning a few minutes), and train one MLP on data from each interval. By keeping the time-window short – on the order of the gas transport time across the site – we ensure each MLP sees a relatively homogeneous wind condition, allowing it to more accurately learn the input-output mapping for that period. In effect, the surrogate model is updated periodically to account for changes in the flow. In our case, the window length can be chosen based on the maximum travel time for gas to reach the farthest sensor (which depends on wind speed and domain size). When deploying the inversion in real-time, the particle filter can then switch to the appropriate MLP corresponding to the current time-window of data. This sequential MLP training strategy enables the surrogate to capture transient behaviours and temporal evolution of the gas plume that a single global model might miss.

Post-Training Assessment. We evaluate the MLP surrogate on held-out test cases (source locations not seen during training) to ensure it generalises well. We use metrics such as the mean absolute percentage error (MAPE) between C_{MLP} and C_{ns} on these test simulations, and we also compare the surrogate's outputs against real sensor measurements when available (e.g. from the Chilbolton release trial). The MLP consistently achieves low prediction error, indicating that it captures the physical relationship between source and sensor effectively. Furthermore, the surrogate is extremely fast: evaluating the MLP for a given input takes on the order of milliseconds or less, which is orders of magnitude faster than running a full CFD simulation for the same scenario. In fact, our learned model is even faster than the simplified Gaussian plume equations (2.2.4) (which are themselves a closed-form approximation) while retaining the accuracy of the CFD approach. This balance of physical fidelity and computational efficiency is what enables our overall inversion framework to operate in near-real-time. In the next section, we demonstrate that using the MLP surrogate within the particle filter yields accurate and rapid source inversion results, effectively combining rigorous Bayesian estimation with a fast learned physics model.

5.4 Real data: Chilbolton gas emission inversion

We now assess the performance of our inversion framework on real data from the Chilbolton Observatory, where controlled methane releases were conducted under varying atmospheric conditions. The known ground truth for source locations and emission rates provides a basis for quantitative validation.

The dataset includes path-averaged methane concentration measurements collected using a laser dispersion spectrometer, which scanned seven retroreflectors every 3 seconds. Wind measurements were obtained via a three-dimensional ultrasonic anemometer, and the sources consisted of perforated 2m × 2m ground frames; see Figure 4.1.1 for site layout. The flat topography and high-frequency measurements make this dataset well-suited for evaluating spatio-temporal inversion methods.

5.4.1 Sensor measurements prediction

We first evaluate the predictive performance of the MLP surrogate model against two baselines: a atmospheric stability class-based Gaussian plume model and the high-fidelity numerical solver used for training data generation. Predictions from our three models were made minute-by-minute and evaluated against the last 20-second averaged sensor measurements. Predicting the last 20 seconds is consistent with the time-averaging Gaussian plume model assumption and ensures the numerical solver's simulated gas reaches the sensors from anywhere on the Chilbolton site. For each minute interval, the Gaussian plume model used the averaged wind inputs, while the numerical solver simulated transport using the full minute of wind data and averaged the last 20 seconds. Additional information regarding how the numerical solver simulation duration and the time-window length were chosen is included in Appendix B.2.

The MLP was trained to predict the numerical solver's 20-second averaged sensor measurements using 484 simulations with distinct source locations, all under the same wind boundary conditions obtained by our anemometer. The true source location was withheld to assess interpolation performance. All models were then given true source locations and evaluated on two test cases: 10 minutes of data from Source 1 (release 2) and 15 minutes from Source 2 (release 5); these reflect periods of ideal wind conditions, where sensors were exposed to the gas plumes.

Table 5.4.1 reports the MAPE for each model. The numerical solver achieved the lowest MAPE but required approximately 21 minutes of computational time for all predictions. The Gaussian plume model was much faster (1.2 seconds) but substantially less accurate. The MLP surrogate achieved accuracy close to the numerical solver, outperforming the plume model, while requiring only milliseconds per prediction; faster than even the plume model. It took 6 minutes to generate the data used to train each MLP and 2 minutes for model training using 90 CPUs and 250 GB of memory, though this can be reduced with GPU acceleration. Each MLP comprises 4 hidden layers with 100 neurons per layer. These results confirm that the MLP surrogate delivers both high accuracy and real-time prediction capability in unsteady flow conditions.

5.4.2 Gas source inversion

We next evaluate the inversion framework by estimating source location using the MLP surrogate within the SIR particle filter. As a baseline, we compare to an inversion using

	Gaussian Plume	Numerical Solver	MLP
Source 1	17.25%	13.28%	13.38%
Source 2	10.63%	9.19%	9.57%
Time (s)	1.20	1220.53	0.25

Table 5.4.1: Gaussian plume model, numerical solver and surrogate model predictions' MAPE for 10 minutes of Source 1 release 2 and 15 minutes of Source 2 release 5 given true source locations. Computational times highlight the extreme efficiency of the surrogate.

		MLP	Gaussian Plume
Source 1	Distance (m)	5.82	11.09
	Time (min)	83.3	173.6

Table 5.4.2: Gaussian plume model and surrogate model-based SIR particles' mean estimation of Source 1's location. The mean is computed by averaging the distance from all particles to the true source location at the SIR's last iteration. Computational times highlight the efficiency of the surrogate – the MLP computational time includes training data generation, training, and SIR inversion.

an atmospheric stability class-free Gaussian plume model (identical to those described in Chapter 4), reducing model misspecification introduced by traditional Gaussian plume models. To ensure informative updates, we used 4-minute sliding time-windows of data over the 10 minutes of Source 1 measurements; therefore, using four MLPs and one Gaussian plume model per window.

Figure **??** shows the posterior distribution of estimated source locations for Source 1 using both the MLP and plume-based filters. The MLP-based inversion yields a tighter and more accurate posterior. Table 5.4.2 quantifies this, reporting the mean distance across all particles to the source location for the last SIR iteration. The MLP-based inversion reduced mean localisation error by nearly half compared to the plume model (5.82m vs. 11.09m) and required only half of the computation time (83.3 min vs. 173.6 min), including surrogate training and particle filtering. Together, these results demonstrate that our surrogate-based framework achieves high inversion accuracy with substantially reduced computational cost, enabling real-time spatio-temporal inference with quantified uncertainty in real-world scenarios. Appendix B.3 contains additional information regarding the Chilbolton source inversion.

5.5 Case-study: source inversion in obstructed unsteady-state flow fields

We now demonstrate the scalability and robustness of our proposed inversion methodology in synthetically generated, more complex monitoring environments. Specifically, those featuring obstacles, time-varying emissions, and fixed source locations, for which no real datasets are currently available. We simulate three distinct 10-minute methane emission events with temporally fluctuating emission rates, each within a spatial domain populated by obstructions. Detailed simulation parameters and setup are provided in the Appendix B.4.

To emulate real-world operational constraints, we implement a sequential inversion protocol in which sensor data are processed minute-by-minute, and a 3-minute sliding time-window of the data is used for the likelihood evaluation. A new MLP surrogate is trained each minute on data from the most recent flow conditions, and the SIR particle filter is subsequently updated by sliding the data window, refining the posterior over source parameters. Specifically, 499 CFD-based training simulations are used to train each MLP, with training performed in parallel across 90 CPU cores, consuming approximately 250 GB of memory. The inversion itself – including 100 iterations of particle filtering between each minute with 1,000 particles – is executed on a modest workstation with only 4 CPU cores and 15 GB of memory. Each MLP comprises four hidden layers with 500 neurons per layer; architectural and training details are further elaborated in the Appendix B.4.

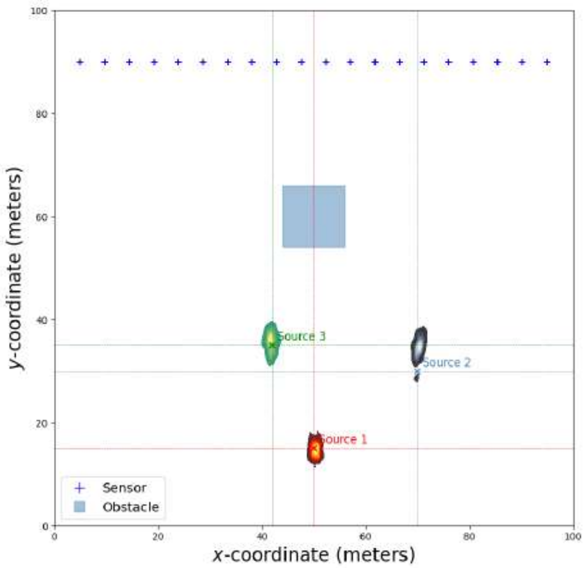


Figure 5.5.1: Case study results for surrogate-based SIR filter source location estimation in unsteady and obstructed flow fields. Data is collected using a line of 20 point-sensors. The three location posterior densities show location estimation at the end of the 10 minutes of gas release.

Figure 5.5.1 visualises the posterior over source locations after the full 10-minute observation window, clearly demonstrating the model's ability to accurately infer fixed source positions even when occluded by structural obstacles. Figure 5.5.2 further highlights the framework's capability to track dynamically varying emission rates: all three sources exhibit time-varying emission profiles, and our particle-based posterior adapts accordingly, accurately reconstructing the temporal evolution of each source's emission intensity. However, the delayed adjustment in estimating Source 2's emission rate following a sharp drop at minute 5 reveals a limitation of the SIR filter: abrupt changes in emission behaviour may require more responsive approaches, such as an interacting multiple model filter (Blom and Bar-Shalom, 2002).

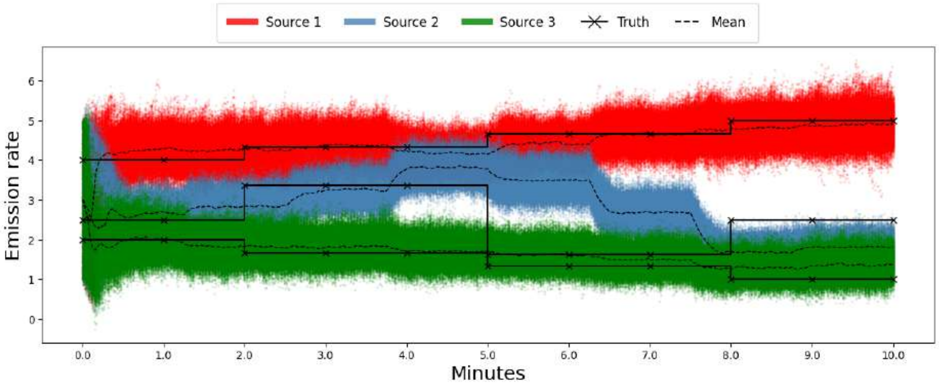


Figure 5.5.2: Case study results for temporally varying emission rate estimation and fix source location. The surrogate-based SIR successfully tracks increasing, decreasing and fluctuating emission rates. The delayed adjustment after Source 2's emission rate shape drop at minute 5 highlights a limitation – such scenarios may require a more responsive two-layered SIR filter.

5.6 Conclusion

We have introduced a real-time Bayesian inversion framework for estimating gas emission source location and emission rate in unsteady atmospheric flow fields. By embedding an MLP surrogate for high-fidelity CFD within an SIR particle filter, we enable accurate and computationally efficient inference in complex environments. This combination allows for rapid, data-driven updates as new observations arrive, making the method suitable for time-critical spatio-temporal environmental monitoring tasks.

Our experiments demonstrate that the MLP surrogate achieves accuracy comparable to full numerical solvers while requiring orders of magnitude less computation. On the Chilbolton dataset, our method outperforms the Gaussian plume model. We further show that the framework generalises to more challenging, obstructed flow scenarios, accurately recovering hidden sources and tracking temporally varying emission rates.

Limitations and Future Directions. While our two-dimensional surrogate assumption is appropriate for flat sites with near-sensor level sources, vertical source-sensor offsets could degrade accuracy in more complex terrains. A lightweight extension using vertical Gaussian plume corrections is proposed in Appendix B.5. Furthermore, the inference involves the specification of several tuning parameters – e.g. the number of SIR particles and iterations; and length of the time-window of the data – appropriate choices for which are likely to be application-dependent. This is also true in how the surrogate model's training data was simulated. Some obstacle shapes may obstruct and delay the gas flow, requiring CFD runs considerably longer than 1 minute. Additionally, retraining a separate MLP per time-window introduces overhead, which could be alleviated via online or transfer learning approaches (Weiss et al., 2016). Finally, robustness to real-world uncertainties – e.g., sensor dropout, wind misestimation, and multi-source interference – remains an open avenue for future work. This study underscores the promise of combining physics-informed machine learning with Bayesian inference for scalable, accurate, and fast environmental monitoring in dynamic conditions.

5.7 Data and code availability

5.7.1 Data availability statement

The raw Chilbolton data that support the findings of this chapter are openly available at the following: https://edata.stfc.ac.uk/items/5c88d121-0e19-4840-a26b-499dba49419a.

5.7.2 Code availability statement

The documented code is openly available on GitHub (https://github.com/NewmanTHP/Deep-Learning-Surrogates-for-Real-Time-Gas-Emission-Inversion).

Chapter 6

Conclusions and Outlook

Having established the importance of probabilistic inversion modelling for gas emissions (Chapter 1), reviewed the fundamental tools for estimating source parameters (Chapters 2 and 3), and introduced new methodological frameworks that advance the field (Chapters 4 and 5), the core of this thesis is now complete. This final chapter serves both as a conclusion and a bridge forward: first, by recapping the key contributions of this work (Section 6.1); then, by outlining a promising avenue for the extension of this thesis (Section 6.2); and finally, by reflecting on the broader implications and future trajectory of atmospheric probabilistic inversion modelling (Section 6.3).

6.1 Summary of contributions

Detecting, locating, and quantifying emission sources is a complex challenge that draws on physics (Chapter 2), statistics (Chapter 3), and computer science (the software infrastructure required to implement methods from Chapters 2 and 3). Source characterisation begins with atmospheric gas measurements, which must be linked to source properties through atmospheric transport models. This thesis has concentrated on ground-based measurements rather than observations collected from satellites.

In Chapter 2, we reviewed the range of transport models available for this task, highlighting their respective advantages. Traditionally, approximate analytical solutions (Section 2.2) and numerical solvers (Section 2.3) defined the modelling spectrum. Recent advances in computational power and machine learning, however, have enabled the rise of surrogate models (Section 2.4), which combine the physical fidelity of numerical solvers with the computational efficiency of analytical formulations. This development, in turn, has made it possible to implement better-suited Bayesian inversion methods for estimating source parameters – capabilities that were previously computationally infeasible.

Chapter 3 then addressed the statistical machinery required for such inference. We examined Monte Carlo methods (Section 3.1), Markov chain Monte Carlo algorithms (Section 3.2), and state-space models (Section 3.3), providing a spectrum of parameter estimation techniques. The choice of method depends on the type of data available, the parameters of interest, and computational constraints, but together they establish a toolkit for probabilistic inversion.

Building on this foundation, the thesis makes two original contributions to the field:

First contribution (Chapter 4). We developed a methodology to estimate the Gaussian plume model's dispersion parameters. By inferring dispersion parameters from data rather than fixing them via atmospheric stability classes, we demonstrated achieving improved accuracy and reliability. Given the widespread use of the Gaussian plume model in practice, this contribution has the potential for broad adoption. To further promote accessibility and reproducibility, we released a Python package on PyPI (`https://pypi.org/project/sourceinversion/`) and made the documented code openly available on GitHub (`https://github.com/NewmanTHP/Probabilistic-Inversion-Modeling-of-Gas-Emissions`).

Second contribution (Chapter 5). We introduced a framework for estimating source parameters in real time under complex fluid flow conditions. Specifically, we combined state-space modelling with deep learning surrogate models of Navier–Stokes–based advection–diffusion equations to capture temporally varying source location and emission rate in obstructed, unsteady environments. To our knowledge, this is the first attempt to perform real-time probabilistic inversion of time-varying emission parameters in such physically realistic settings. This work therefore represents a major step toward meeting the monitoring and reporting standards set by initiatives such as OGMP 2.0. Beyond its immediate application, it also provides a proof-of-concept for the full pipeline: from computational fluid dynamics surrogate model development to deployment within a probabilistic inference framework. The documented code is openly available on GitHub (https://github.com/NewmanTHP/Deep-Learning-Surrogates-for-Real-Time-Gas-Emission-Inversion).

Together, we believe these contributions mark significant progress in probabilistic inversion modelling of gas emissions. While much work remains, this thesis establishes both methodological advances and practical tools that we hope will serve as a foundation for future research in this rapidly evolving field.

6.2 Towards improved satellite inversion modelling

We consider the extension of this thesis towards satellite-based source characterisation to be both a natural and an essential next step to achieving global GHG emission monitoring. Although ground-based sensors still provide superior measurement accuracy (defined as the closeness of the observation to the true gas concentration) compared to satellites (Lorente et al., 2022; Dowd et al., 2024), the performance gap is steadily narrowing, and satellite observations bring several unique advantages. First, most satellite datasets are openly available through platforms such as the Copernicus Data Space

Ecosystem, requiring only free registration, which makes them highly suitable for research, transparency, and collaborative efforts. Second, satellite observations enable global monitoring[1], thereby reducing dependence on private companies to deploy local monitoring infrastructure and providing a means to independently verify reported emissions. Third, satellites offer vastly greater spatial coverage than ground-based networks, with instruments such as TROPOMI (Veefkind et al., 2012) and GOSAT (Yokota et al., 2009) already providing global methane observations. Finally, while temporal resolution has historically been a limitation (e.g. TROPOMI's once-daily revisit), this constraint is diminishing as satellite constellations expand. Sub-daily monitoring of individual sites is now feasible, either through high-frequency revisits of a single satellite (Sánchez-García et al., 2021) or through synergistic use of multiple satellites (De Jong et al., 2025).

Given these developments, this section focuses on satellite-based inversion with particular emphasis on methane emission sources. We propose a research direction that extends the core ideas of this thesis: the development of a surrogate model for atmospheric methane transport tailored to satellite inversion. Such a framework would combine the efficiency of CFD surrogates with probabilistic parameter estimation, scaling and complementing the ideas developed in this thesis for local in-situ applications to global monitoring.

The remainder of this section is structured as follows: we first review the current literature on satellite-based methane source parameter estimation (Section 6.2.1). We then explain our decision to focus on satellites with large spatial coverage, highlighting the limitations of current inversion methods for these instruments (Section 6.2.2). Finally, we examine the state of surrogate modelling for atmospheric gas transport, discuss why existing approaches are insufficient for our purposes, and present potential strategies to address these challenges (Section 6.2.3).

[1] Subject to meteorological (e.g. cloud cover) and surface-related (e.g. high albedo) conditions.

6.2.1 Literature review on satellite inversion

Satellite instruments designed to observe atmospheric methane concentrations are generally divided into two categories: area flux mappers (AFMs) and point source imagers (PSIs) (see Figure 6.2.1). The primary distinction lies in their spatial resolution and coverage. PSIs, such as GHGsat (Jervis et al., 2021) and WorldView-3 (Sánchez-García et al., 2021), achieve very high spatial resolution, typically with pixels smaller than 60 m^2. In contrast, AFMs, such as TROPOMI and GOSAT, operate at much coarser resolution, with pixel sizes often exceeding 3 km^2, but provide much broader coverage. This reflects a fundamental trade-off: while both AFMs and PSIs are capable of daily revisits, AFMs offer near-global coverage, whereas PSIs concentrate on specific regions or targeted facilities. Figure 6.2.2 illustrates this contrast by comparing TROPOMI and GHGsat observations of a methane plume over Casablanca, Morocco. The literature on satellite-based methane source characterisation reflects these differences, with distinct approaches developed for detection, localisation, and quantification depending on the satellite type. We now briefly review these methods for AFMs and PSIs before outlining potential directions for future contributions.

Detection. Detection is predominantly carried out with AFMs, as they continuously scan wide areas without targeting specific areas, unlike PSIs. The objective is to identify the presence of methane plumes and, by extension, the existence of emission sources. However, because AFMs generally have high concentration detection thresholds, only relatively large emitters can be reliably detected (Jacob et al., 2022). A variety of detection approaches have been explored: for instance, Maasakkers et al. (2022) used visual inspection of global maps created from one year of aggregated methane observations, while more recent work by Schuit et al. (2023) introduced a two-stage machine learning framework. Their method applies a CNN to assign a plume probability to 32×32 pixel image patches issued from a single observation, followed by a support vector ma-

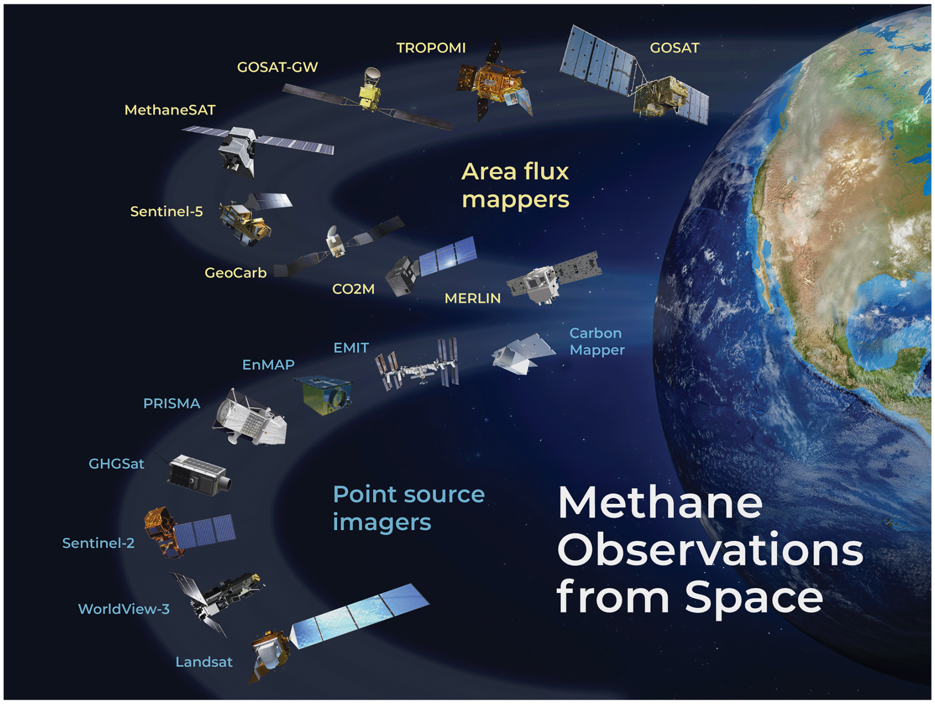


Figure 6.2.1: Satellite instruments observing methane in the shortwave infrared (SWIR). Area flux mappers are designed to quantify total emissions at regional to global scales, while point source imagers target individual facilities by observing atmospheric plumes. Credit: Jacob et al. (2022).

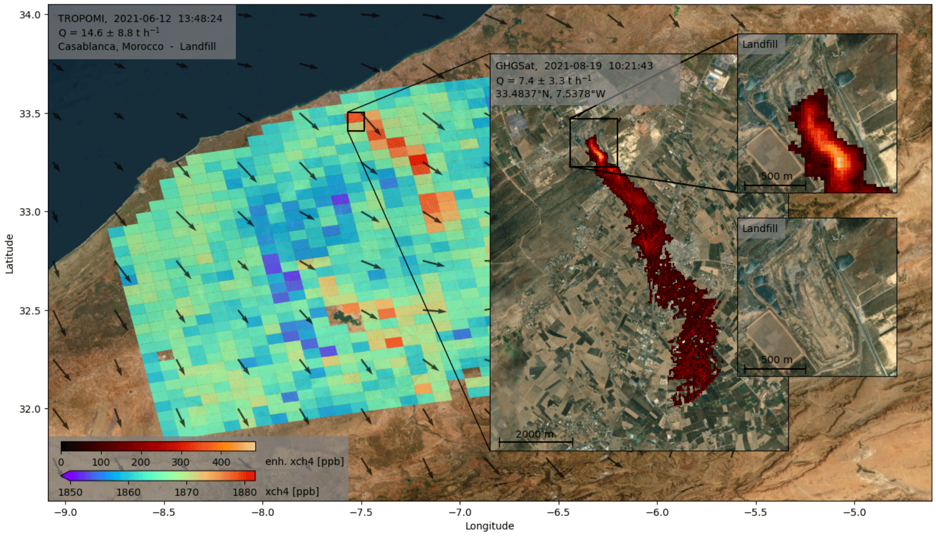


Figure 6.2.2: Methane plumes detected near Casablanca, Morocco, on two different days, shown with TROPOMI and GHGSat data. Credit: Schuit et al. (2023).

chine classifier to distinguish real plumes from artefacts. This progression highlights the methodological spectrum, ranging from visual diagnostics to fully automated detection pipelines.

Localisation. For PSIs, source localisation is often performed visually, aided by the instruments' fine spatial resolution and relatively narrow observation swaths, which allow potential emission points to be identified directly. For AFMs, localisation is considerably more challenging: individual sources typically cannot be identified visually unless a single candidate is found upwind (e.g. oil and gas extraction sites, coal mines, etc.) (Schuit et al., 2023). When no obvious candidate is found, source locations are typically indirectly estimated by estimating emission rates over a grid (Maasakkers et al., 2022). In this framework, each grid cell is treated as a potential emitter, with estimated emission rates constrained to be positive if a source is present and zero otherwise.

Quantification. Estimating emission rates is arguably the most challenging task, as true source strengths are rarely known and cannot be verified visually. When the full plume is observed at high resolution, such as with PSIs, the integrated mass enhancement (IME) method is widely used. This approach has been shown to outperform plume-based forward-model inversion, which struggles with small-scale turbulence and wind variability (Varon et al., 2018). In contrast, the IME method becomes less reliable when applied to AFM data since the plume's shape and concentration are often not sufficiently determined. For such cases, grid-based inversion using atmospheric transport models (e.g. GEOS-Chem, WRF-Chem, and NAME) is more commonly employed (Maasakkers et al., 2019, 2022; Western et al., 2021). A major limitation of these forward-model approaches is their computational cost, which cannot keep up with the large volume of data collected. For example, TROPOMI records approximately seven million pixel measurements per day (Butz et al., 2012); likely to be an underestimation since the pixel size upgrade of August 2019. To address this, it is standard practice to assume that the emission rate does not influence atmospheric transport, effectively treating it as a simple scaling factor. Moreover, since simulations for each grid cell differ only by source location (i.e. assume identical wind fields and boundary conditions), the computation of transport can be amortised into a Jacobian matrix (Brasseur and Jacob, 2017); this acts as the coupling matrix $\boldsymbol{A}$ in Chapter 4. This matrix maps any cell-centred source to its resulting concentration field at a given time, thereby greatly reducing computational demands. With the additional assumption that both emission rates and observational errors follow Gaussian distributions, Bayesian inversion can then be carried out using analytical solutions (Brasseur and Jacob, 2017). Recently, machine learning has also been introduced as a direct quantification tool: Roberts et al. (2024) demonstrated the use of CNNs to infer emission rates from TROPOMI observations, while related work has applied similar methods to airborne and drone data, training models on large-eddy simulations (Jongaramrungruang et al., 2022).

Based on this brief review and considering the scope of this thesis, we conclude that the most significant advances can be achieved by improving localisation and quantification methods for AFMs. This priority is justified by their extensive spatial coverage and by the common practice of PSIs leveraging AFM results to guide more targeted follow-up investigations.

6.2.2 Limitations in current area flux mapper-based inversions

Current approaches for estimating emission rates and source locations from AFM data can broadly be grouped into two categories: grid-based inversion methods (Maasakkers et al., 2019, 2022; Western et al., 2021; Nesser et al., 2024) and machine learning-based direct estimation (Jongaramrungruang et al., 2022; Schuit et al., 2023; Roberts et al., 2024).

Grid-based inversions rely on numerical forward models, such as chemical transport models or simplified atmospheric dispersion models, to infer emission patterns. These methods offer a principled, physics-based framework for parameter estimation, often using Jacobian-based approaches to relate emissions to observed concentrations. However, the high computational cost of running such models at relevant spatial and temporal resolutions imposes strict constraints on model complexity. In practice, this often necessitates strong simplifying assumptions. For example, fixing the set of potential source locations to a predefined grid and selecting a subset of the wind field during which the emission rate is assumed constant. These assumptions reduce flexibility and may introduce structural biases, particularly in regions where emission sources are intermittent or poorly aligned with the chosen grid resolution.

The machine learning approaches referenced above, by contrast, bypass the forward-modelling step and directly estimate emissions from observational data. Their primary strengths lie in computational efficiency and scalability: once trained, they can rapidly produce emission estimates over large domains. However, most current implementations are purely data-driven and thus limited in two important respects. First, they generally lack mechanisms for rigorous uncertainty quantification, producing point estimates without probabilistic confidence intervals. Second, they provide limited means of incorporating physical constraints or prior knowledge, such as atmospheric transport physics, emission plausibility bounds, or known relationships with meteorological drivers, in a principled way. This can lead to solutions that fit the data statistically but lack physical interpretability or robustness, particularly in regions with sparse or noisy satellite coverage.

These challenges are not unique to AFM-based inversion. In the broader atmospheric inversion literature, several approaches have sought to alleviate the computational bottleneck of inversion while preserving uncertainty quantification by introducing surrogates for backward Lagrangian particle dispersion models (LPDMs). For instance, Cartwright et al. (2023) developed a convolutional variational autoencoder-Gaussian process hybrid to emulate LPDM footprints[2] using reference runs at selected sites for interpolation. Similarly, Fillola et al. (2023) trained a gradient-boosted regression tree surrogate parameterised solely by meteorological inputs, eliminating the need for post-training LPDM simulations. Comparable approaches include graph neural networks and hybrid Gaussian plume models (Kocijan et al., 2023; Fillola et al., 2025; He et al., 2025; Dadheech et al., 2025).

While successful for sparse sensor networks, these LPDM-based surrogates do not scale well to satellite data, where effectively every pixel acts as a sensor. Generating sufficient particle-based training data to cover the diversity of AFM retrievals would likely be computationally prohibitive. Moreover, incorporating atmospheric chemistry into LPDM frameworks remains challenging, yet chemical transformations can be non-negligible when modelling large-scale plumes.

By contrast, Eulerian models (described in Chapter 2 Section 2.3) solve atmospheric transport equations on structured three-dimensional grids, making them inherently better suited to satellite inversion. They naturally produce vertically resolved concentration fields consistent with AFM retrievals and can incorporate atmospheric chemistry in a physically consistent manner.

We therefore propose the development of a spatio-temporal surrogate model of atmospheric methane transport. Such a surrogate would emulate the outputs of numerical forward models while retaining the computational efficiency characteristic of machine learning. When embedded within a Bayesian inversion framework, this approach would combine the speed of machine learning surrogates with the rigorous uncertainty quantification of probabilistic inference. The surrogate could serve as a continuous analogue to the Jacobian matrix: emission sources would no longer be confined to discrete grid

[2] Footprints are sensitivity maps showing how emissions at a location affect a sensor's measurements.

cells but could instead be placed anywhere within the inversion domain. Moreover, by conditioning directly on meteorological inputs, the surrogate would rapidly accommodate inversions under varying weather conditions, eliminating the need to recompute Jacobians or rerun expensive transport models for each new scenario. This combination of flexibility, physical consistency, and computational efficiency would address several of the most persistent challenges currently limiting AFM-based inversion methodologies.

6.2.3 Efficient probabilistic inversion of area flux mapper data

We envision a surrogate model parameterised by meteorological and geological inputs that directly outputs spatio-temporal methane concentration fields, enabling direct comparison with AFM retrievals. Incorporating time-dependence is essential, as satellite observations may capture plumes at any stage of their evolution, and the surrogate must be capable of representing this dynamical behaviour. The computational efficiency gained through such a model would make a range of inversion methods feasible that are currently impractical due to the prohibitive cost of traditional forward modelling.

The broader CFD and climate modelling communities provide encouraging evidence that deep learning surrogates can accurately emulate Eulerian systems at scale. For example, Kim et al. (2019) developed two- and three-dimensional CNN-based surrogates for high-resolution continuous fluid simulations, demonstrating accurate temporal interpolation capabilities. Rasp and Thuerey (2021) employed residual neural networks (ResNets) to emulate global weather fields from Eulerian models, comparing fixed-lead-time predictions with a continuous lead-time framework. Their models, trained on climate simulations and reanalysis data, produced five-day forecasts of geopotential, temperature, and precipitation with accuracy comparable to low-resolution physical models. More recently, Benson et al. (2025) investigated UNets, graph neural networks, spherical Fourier neural operators, and vision transformers as surrogates for atmospheric CO_2 transport. All four architectures generated stable, physically consistent global CO_2 fields over monthly timescales, underscoring their potential for rapid carbon-cycle monitoring and inversion. Collectively, these advances demonstrate that deep neural networks can now capture the spatio-temporal dynamics of complex Eulerian systems with a fidelity that was previously unattainable.

These developments highlight a clear opportunity: the construction of a realistic, Eulerian-based (e.g. GEOS-Chem) high-resolution spatio-temporal surrogate model of atmospheric methane transport, specifically designed to produce outputs directly comparable to AFM column-integrated retrievals (e.g. from TROPOMI) after coarse-gridding. Running simulations at high resolution would resolve fine-scale transport and chemical processes that influence the larger-scale signals observed by satellites. Coarse-gridding this output to the satellite resolution would ensure that inversion comparisons reflect both unresolved sub-grid variability and the actual observational resolution, improving both consistency and accuracy.

One promising architecture for such a surrogate model is the convolutional long short-term memory network (ConvLSTM), which simultaneously models spatial and temporal dependencies in sequential image-like data. Unlike fully connected LSTM layers, ConvLSTM layers apply convolution operations in both the input-to-state and state-to-state transitions, enabling them to capture local spatial correlations while preserving temporal dynamics. ConvLSTM networks have already proven successful in forecasting precipitation (Shi et al., 2015; Moishin et al., 2021), tropical cyclones (Tong et al., 2022), aurora evolution (Jiang et al., 2023), and marine water levels (Yang et al., 2024). A similar framework could be adapted for atmospheric methane transport, offering a balance of physical interpretability, spatio-temporal fidelity, and computational efficiency.

Training such a surrogate would require a large set of high-resolution Eulerian simulations, encompassing a wide variety of meteorological conditions, emission magnitudes, and spatial configurations to adequately sample the relevant state space. This is nontrivial; the parameter space of atmospheric transport is inherently high-dimensional, and neural networks trained on insufficiently diverse conditions are prone to distribution shift and hallucinated dynamics when deployed in previously unseen scenarios (Dueben and Bauer, 2018; Quiñonero-Candela et al., 2022). Ensuring generalisation across regions and timescales would therefore necessitate carefully designed sampling strategies, domain adaptation methods, or hybrid training schemes that incorporate physical constraints (Karniadakis et al., 2021).

A practical strategy might be to begin with regional prototypes – domains where both high-quality AFM and independent validation data are available – and progressively expand to continental and ultimately global scales. However, even this staged approach introduces complexity. The coupling between atmospheric dynamics, sub-grid turbulence, and chemical loss processes varies across climates and emission regimes. Moreover, integrating atmospheric chemistry during training is essential for capturing methane oxidation, hydroxyl variability, and interactions with other reactive species over longer transport timescales. However, this would significantly increase both data and computational requirements, as it effectively multiplies the state variables. Without such integration, the surrogate risks systematically misrepresenting methane lifetime and plume dissipation, particularly in regions with strong chemical reactions.

Collectively, these challenges underscore that while recent advances in deep learning surrogates provide a compelling proof of concept, translating those advances into a methane-specific inversion-ready model would require a carefully planned research effort. Success would depend not only on architecture selection but also on training data design, physics-informed regularisation, and robust validation frameworks.

Equally critical is the design of the inversion framework itself. Embedding the surrogate within a Bayesian inference context would enable the simultaneous estimation of emissions and their associated uncertainties, while retaining full flexibility in specifying prior information (e.g., inventories, sectoral activity data, or expert-informed constraints). The dramatic computational efficiency of a neural-network-based surrogate could reduce posterior sampling runtimes by orders of magnitude compared with traditional chemical transport models, where a single high-resolution inversion often requires multiple days of computation even on large-scale clusters. For instance, Benson et al. (2025) achieved a 75% reduction in runtime using a UNet-based surrogate for atmospheric CO_2 forecasting, while Chandra et al. (2020) demonstrated that employing an MLP surrogate for their landscape evolution forward model cut parallel tempering MCMC sampling costs by more than 50%.

The research pathway described here is motivated by both the current state of the field and the findings of this thesis. It is intended to provide a foundation of ideas for future investigations, offering a promising route toward fast, flexible, and probabilistic inversion of AFM data at a global scale – a step toward routinely producing robust, policy-relevant methane emission estimates with the speed and resolution that modern climate mitigation efforts demand.

6.3 Broader implications and future perspectives

The ability to transform raw atmospheric observations into meaningful, actionable knowledge is at the heart of modern climate science. Probabilistic inversion modelling provides a powerful framework where physical models, prior knowledge and expert judgment can coalesce to deliver uncertainty-aware diagnostics of the atmosphere. Enabled by advances in computational power, machine learning, and deep neural networks, these Bayesian approaches now harness unprecedented flexibility and efficiency, opening the door to a new generation of tools capable of tracing greenhouse gas emissions with great speed, resolution and insight drawn from vast datasets.

Such tools will not only quantify emissions – they will contextualise them. In the near future, a methane leak in a remote region could be detected, localised, and quantified in near-real time, not as an isolated data point but as part of a dynamic, probabilistic understanding of gas emissions across the globe. This shift holds profound implications for climate policy, industry accountability, and international cooperation. Instead of relying on coarse inventories and delayed reporting cycles, nations and organisations could negotiate based on shared, continuously updated evidence of who emits what, where, and when.

Beyond gases, the same principles apply to a broader spectrum of atmospheric constituents (e.g. volcanic ash, airborne pathogens, pollen). The architecture of probabilistic inversion models is not gas emission-specific; it is a general strategy which can be used to learn from the atmosphere. Combining satellite data, ground-based networks, airborne campaigns, and even unconventional sensors (from ships, drones, or low-cost citizen-science devices) within a unified probabilistic framework would transform the atmosphere from a challenging measurement domain into a well-mapped, continuously monitored system.

The implications are not confined to Earth. As humanity extends its reach into the Solar System and beyond, the same inversion methodologies that decode our planet's methane emissions could help reveal the dynamics of extraterrestrial atmospheres. Whether probing the methane haze of Titan, the tenuous exospheres of icy moons, or a future terraformed world, probabilistic inversion techniques could act as the lens through which we translate sparse, remote measurements into an evolving understanding of other planetary environments. In this sense, the tools developed for Earth's climate challenge may ultimately become instruments of planetary exploration, offering insight into the formation, evolution, and habitability of worlds far beyond our own.

This evolution also carries intellectual and societal opportunities. For researchers, it pushes the boundary of what is computationally and conceptually tractable. It invites interdisciplinary collaboration between atmospheric scientists, computer scientists, statisticians, planetary scientists, and policymakers. For society, it promises transparency. A shared, evidence-based understanding of the invisible forces shaping our climate. In a future where trust in atmospheric data underpins not only international agreements but also planetary missions, the credibility and accessibility of probabilistic, uncertainty-aware modelling may prove as important as the measurements themselves.

Looking further ahead, inversion models may evolve beyond diagnostic applications to encompass predictive capabilities. By integrating with economic, agricultural, energy, and planetary system models, such frameworks could simulate the impacts of both human activities and natural variability in real time. These integrated systems could function as early-warning networks as well as decision-support platforms, providing actionable insights for strategies ranging from local infrastructure planning to global mission design.

What began as a methodological challenge – how to infer emissions from sparse, noisy, and indirect measurements – thus becomes an enabling technology for a more transparent, responsive, and ultimately effective climate response, while also laying the groundwork for decoding the atmospheres of distant worlds. Probabilistic inversion modelling of atmospheric gas emissions stands at this intersection of science, technology, and policy, not as an end in itself, but as a foundation for a future in which our collective atmospheric impact is visible, traceable, and, crucially, actionable.

"There is no real ending. It's just
the place where you stop the story."

— Frank Herbert

Appendices to Chapter 4 and 5

Appendix A

Supplementary Material for Chapter 4

This appendix contains all the supplementary materials supporting Chapter 4.

A.1 Supplementary material: mathematical theories and derivations

Context: In the Chapter 4, we performed parameter estimation using M-MALA-within-Gibbs. This section provides additional details regarding the MCMC algorithm.

Content: Section A.1.1 contains the derivations of conditional posteriors used for Gibbs sampling. Section A.1.2 contains the pseudocode of the M-MALA-within-Gibbs. Section A.1.3 contains the prior distributions.

A.1.1 MCMC posterior derivations

Conditional posterior $p(\sigma^2_{\text{sns}}|\boldsymbol{d},\boldsymbol{s},\boldsymbol{\beta})$

The conditional posterior distribution for the variance of the measurement error, σ^2_{sns}, can be written as:

$$\begin{aligned} p(\sigma^2_{\text{sns}}|\boldsymbol{d},\boldsymbol{s},\boldsymbol{\beta}) &= \frac{p(\boldsymbol{d}|\boldsymbol{s},\boldsymbol{\beta},\sigma^2_{\text{sns}})p(\boldsymbol{s})p(\boldsymbol{\beta})p(\sigma^2_{\text{sns}})}{\displaystyle\int p(\boldsymbol{d}|\boldsymbol{s},\boldsymbol{\beta},\sigma^2_{\text{sns}})p(\boldsymbol{s})p(\boldsymbol{\beta})p(\sigma^2_{\text{sns}})\mathrm{d}\sigma^2_{\text{sns}}}, \\ &= \frac{p(\boldsymbol{d}|\boldsymbol{s},\boldsymbol{\beta},\sigma^2_{\text{sns}})p(\sigma^2_{\text{sns}})}{\displaystyle\int p(\boldsymbol{d}|\boldsymbol{s},\boldsymbol{\beta},\sigma^2_{\text{sns}})p(\sigma^2_{\text{sns}})\mathrm{d}\sigma^2_{\text{sns}}}. \end{aligned}$$

To make it clearer lets denote $R = \displaystyle\int p(\boldsymbol{d}|\boldsymbol{s},\boldsymbol{\beta},\sigma^2_{\text{sns}})p(\sigma^2_{\text{sns}})\mathrm{d}\sigma^2_{\text{sns}}$, i.e. the normalising constant.

$$\begin{aligned} p(\sigma^2_{\text{sns}}|\boldsymbol{d},\boldsymbol{s},\boldsymbol{\beta}) &= \frac{1}{R} \times p(\boldsymbol{d}|\boldsymbol{s},\boldsymbol{\beta},\sigma^2_{\text{sns}})p(\sigma^2_{\text{sns}}), \\ &= \frac{1}{R}\Big(2\pi\sigma^2_{\text{sns}}\Big)^{-\frac{n_{obs}}{2}} \exp\Bigg\{-\frac{1}{2\sigma^2_{\text{sns}}}\sum^{n_{obs}}(\boldsymbol{d}-\boldsymbol{\beta}-A\boldsymbol{s})^2\Bigg\}\frac{b^a}{\Gamma(a)}\Bigg(\frac{1}{\sigma^2_{\text{sns}}}\Bigg)^{a+1} \\ &\quad \times \exp\Bigg\{-\frac{b}{\sigma^2_{\text{sns}}}\Bigg\}. \end{aligned}$$

We can now absorb the terms that do not depend on σ^2_{sns} into R. We obtain:

$$\begin{aligned} p(\sigma^2_{\text{sns}}|\boldsymbol{d},\boldsymbol{s},\boldsymbol{\beta}) &= \frac{1}{R}\Big(\sigma^2_{\text{sns}}\Big)^{-\frac{n_{obs}}{2}-a-1} \exp\Bigg\{-\frac{1}{2\sigma^2_{\text{sns}}}\sum^{n_{obs}}(\boldsymbol{d}-\boldsymbol{\beta}-A\boldsymbol{s})^2 - \frac{b}{\sigma^2_{\text{sns}}}\Bigg\}, \\ &= \frac{1}{R}\Big(\sigma^2_{\text{sns}}\Big)^{-(\frac{n_{obs}}{2}+a)-1} \exp\Bigg\{-\frac{1}{\sigma^2_{\text{sns}}}\Bigg(\frac{\sum^{n_{obs}}(\boldsymbol{d}-\boldsymbol{\beta}-A\boldsymbol{s})^2}{2}+b\Bigg)\Bigg\}. \end{aligned}$$

This corresponds to the Inverse-Gamma distribution:

$$\sigma^2_{\text{sns}}|\boldsymbol{d},\boldsymbol{s} \sim \text{Inv-Gamma}\left(\frac{n_{obs}}{2}+a\ ,\ b+\frac{\sum^{n_{obs}}(\boldsymbol{d}-\boldsymbol{\beta}-A\boldsymbol{s})^2}{2}\right).$$

Conditional posterior $p(\boldsymbol{\beta'}|\boldsymbol{d},\boldsymbol{s},\sigma^2_{\textbf{sns}})$

The conditional posterior distribution for the background concentration $\boldsymbol{\beta'}$ where $\boldsymbol{\beta} = \boldsymbol{\beta'} \otimes \mathbf{1}_{n_T}$ can be written as:

$$\begin{aligned} p(\boldsymbol{\beta'}|\boldsymbol{d},\boldsymbol{s},\sigma^2) &= \frac{p(\boldsymbol{d}|\boldsymbol{s},\boldsymbol{\beta'},\sigma^2_{\text{sns}})p(\boldsymbol{s})p(\boldsymbol{\beta'})p(\sigma^2_{\text{sns}})}{\displaystyle\int p(\boldsymbol{d}|\boldsymbol{s},\boldsymbol{\beta'},\sigma^2_{\text{sns}})p(\boldsymbol{s})p(\boldsymbol{\beta'})p(\sigma^2_{\text{sns}})\mathrm{d}\boldsymbol{\beta'}}, \\ &= \frac{p(\boldsymbol{d}|\boldsymbol{s},\boldsymbol{\beta'},\sigma^2_{\text{sns}})p(\boldsymbol{\beta'})}{\displaystyle\int p(\boldsymbol{d}|\boldsymbol{s},\boldsymbol{\beta'},\sigma^2_{\text{sns}})p(\boldsymbol{\beta'})\mathrm{d}\boldsymbol{\beta'}}. \end{aligned}$$

To make it clearer lets denote $R = \displaystyle\int p(\boldsymbol{d}|\boldsymbol{s},\boldsymbol{\beta'},\sigma^2_{\text{sns}})p(\boldsymbol{\beta'})\mathrm{d}\boldsymbol{\beta'}$, i.e. the normalising constant.

$$\begin{aligned} p(\boldsymbol{\beta'}|\boldsymbol{d},\boldsymbol{s},\sigma^2_{\text{sns}}) &= \frac{1}{R} \times p(\boldsymbol{d'}|\boldsymbol{s},\boldsymbol{\beta'},\sigma^2_{\text{sns}})p(\boldsymbol{\beta'}), \\ &= \frac{1}{R}\left(2\pi\sigma^2\right)^{-\frac{n_{\text{sns}}}{2}} \exp\Big\{-\frac{1}{2\sigma^2_{\text{sns}}}\left(\boldsymbol{d'}-\boldsymbol{\beta'}-(\boldsymbol{A}\boldsymbol{s})'\right)^2\Big\} \\ &\qquad \times (2\pi)^{-\frac{n_{\text{sns}}}{2}} |\Sigma_{\boldsymbol{\beta'}}|^{-\frac{1}{2}} \exp\Big\{-\frac{1}{2}\left(\boldsymbol{\beta'}-\boldsymbol{\mu}_{\boldsymbol{\beta'}}\right)^T \Sigma^{-1}_{\boldsymbol{\beta'}}\left(\boldsymbol{\beta'}-\boldsymbol{\mu}_{\boldsymbol{\beta'}}\right)\Big\}, \\ &= \frac{1}{R}\exp\Big\{-\frac{1}{2}\Big[\boldsymbol{\beta'}^T(\frac{1}{\sigma^2_{\text{sns}}}\mathbb{I}+\Sigma^{-1}_{\boldsymbol{\beta'}})\boldsymbol{\beta'} - 2\boldsymbol{\beta'}^T\left(\frac{1}{\sigma^2_{\text{sns}}}(\boldsymbol{d'}-(\boldsymbol{A}\boldsymbol{s})')+\Sigma^{-1}_{\boldsymbol{\beta'}}\boldsymbol{\mu}_{\boldsymbol{\beta'}}\right)\Big]\Big\}. \end{aligned}$$

This is the kernel of a multivariate Gaussian distribution, therefore:

$$\boldsymbol{\beta'} \mid \psi \backslash \{\boldsymbol{\beta'}\} \sim \mathrm{N}\left(\left(\frac{1}{\sigma^2_{\text{sns}}}\mathbb{I} + {\boldsymbol{\Sigma}_{\boldsymbol{\beta'}}}^{-1}\right)^{-1} \left(\frac{1}{\sigma^2_{\text{sns}}}(\boldsymbol{d'} - (\boldsymbol{As})') + {\boldsymbol{\Sigma}_{\boldsymbol{\beta'}}}^{-1}\boldsymbol{\mu}_{\boldsymbol{\beta'}}\right), \left(\frac{1}{\sigma^2_{\text{sns}}}\mathbb{I} + {\boldsymbol{\Sigma}_{\boldsymbol{\beta'}}}^{-1}\right)^{-1} \right).$$

Note: $\boldsymbol{d'}$ and $(\boldsymbol{As})'$ correspond to subsets of vectors $\boldsymbol{d}$ and $\boldsymbol{As}$ respectively where every n_T value was kept.

A.1.2 M-MALA-within-Gibbs pseudocode

Note: The pseudocode allows either to keep the step-size ϵ_{ss} fixed throughout the MCMC or to tune it during burn-in to achieve an optimal acceptance rate of 70% (Girolami and Calderhead, 2011) then fixing it at the end of the burn-in period.

Additionally, using Table A.1.1 in Appendix A.1.3 we define $\boldsymbol{\Sigma_s} = \sigma_s^2\mathbb{I}$, $\boldsymbol{\Sigma_{xy}} = [\sigma_x^2, \sigma_y^2]\mathbb{I}$, $\boldsymbol{\Sigma_a} = \sigma_a^2\mathbb{I}$, and $\boldsymbol{\Sigma_b} = \sigma_b^2\mathbb{I}$.

$$\begin{aligned} p(\boldsymbol{\theta} \mid \boldsymbol{d}) = \mathrm{N}\left(\boldsymbol{d} \mid \boldsymbol{As} + \boldsymbol{\beta}, \sigma^2_{\text{sns}}\mathbb{I}\right) \times \Bigg\{ &\mathrm{N}\left(\boldsymbol{s} \mid \boldsymbol{\mu}_s, \boldsymbol{\Sigma}_s\right) \times \mathrm{N}\left(\begin{bmatrix} \tilde{\boldsymbol{x}} \\ \\ \tilde{\boldsymbol{y}} \end{bmatrix} \mid \boldsymbol{\mu}_{xy}, \boldsymbol{\Sigma}_{xy} \right) \\ &\times \mathrm{N}\left(\begin{bmatrix} a_H \\ \\ a_V \end{bmatrix} \mid \boldsymbol{\mu}_a, \boldsymbol{\Sigma}_a \right) \times \mathrm{N}\left(\begin{bmatrix} b_H \\ \\ b_V \end{bmatrix} \mid \boldsymbol{\mu}_b, \boldsymbol{\Sigma}_b \right) \Bigg\}, \end{aligned}$$

$$q(\boldsymbol{\theta}^* \mid \boldsymbol{\theta}^{(l-1)}) = \mathrm{N}\left(\boldsymbol{\theta}^* \mid \boldsymbol{\theta}^{(l-1)} + 0.5\epsilon_{\text{ss}}^{(l-1)}\boldsymbol{Q}^{-1}(\boldsymbol{\theta}^{(l-1)})\nabla\log(p(\boldsymbol{\theta}^{(l-1)} \mid \boldsymbol{d})), \epsilon_{\text{ss}}^{(l-1)}\boldsymbol{Q}^{-1}(\boldsymbol{\theta}^{(l-1)})\right).$$

```
Algorithm 1: M-MALA-within-Gibbs
Input: Number of iterations: $L$; initialise variables: $\boldsymbol{\theta}^{(0)} = [s, \tilde{x}, \tilde{y}, a_H, a_V, b_H, b_V]$;
       $\sigma^2_{\text{sns}}$; $\boldsymbol{\beta'}$; data: $\boldsymbol{d}$; log-likelihood: $\log(p(\boldsymbol{\theta}|\boldsymbol{d}))$; step size: $\epsilon^{(0)}_{\text{ss}}$.
Output: Samples for: $\boldsymbol{\theta}$; $\sigma^2_{\text{sns}}$; $\boldsymbol{\beta'}$; log-likelihoods; acceptance rates.
for $l = 1, 2, \ldots, L$ do
    // M-MALA step
    Evaluate $\log(p(\boldsymbol{\theta}^{(l-1)}|\boldsymbol{d}))$, $\nabla \log(p(\boldsymbol{\theta}^{(l-1)}|\boldsymbol{d}))$, $\nabla^2 \log(p(\boldsymbol{\theta}^{(l-1)}|\boldsymbol{d}))$, and $\boldsymbol{A}^{(l-1)}$
    Propose: $\boldsymbol{\theta}^* \sim \mathrm{N}_n\Big(\boldsymbol{\theta}^{(l-1)} + 0.5\, \epsilon^{(l-1)}_{\text{ss}} \boldsymbol{Q}^{-1} \nabla \log(p(\boldsymbol{\theta}^{(l-1)}|\boldsymbol{d})), \epsilon^{(l-1)}_{\text{ss}} \boldsymbol{Q}^{-1}\Big)$
    Evaluate $\log(p(\boldsymbol{\theta}^*|\boldsymbol{d}))$, $\nabla \log(p(\boldsymbol{\theta}^*|\boldsymbol{d}))$, $\nabla^2 \log(p(\boldsymbol{\theta}^*|\boldsymbol{d}))$, and $\boldsymbol{A}^*$
    Compute acceptance probability:
        $$\text{acc-prob} = \log(p(\boldsymbol{\theta}^*|\boldsymbol{d})) - \log(p(\boldsymbol{\theta}^{(l-1)}|\boldsymbol{d})) + q(\boldsymbol{\theta}^{(l-1)}|\boldsymbol{\theta}^*) - q(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{(l-1)})$$
    Draw $u \sim \text{Uniform}(0, 1)$
    if $\log(u) < \textit{acc-prob}$ then
        $\boldsymbol{\theta}^{(l)} = \boldsymbol{\theta}^*$
        $\boldsymbol{A}^{(l)} = \boldsymbol{A}^*$
        sum-accept $+= 1$
    else
        $\boldsymbol{\theta}^{(l)} = \boldsymbol{\theta}^{(l-1)}$
        $\boldsymbol{A}^{(l)} = \boldsymbol{A}^{(l-1)}$
    end
    acceptance-rate$^{(l)} = 0.01\times$ acceptance-rate$^{(l-1)} + 0.99 \times \frac{\text{sum-accept}}{l}$
    Update step size: $\epsilon^{(l)}_{\text{ss}}$ using acceptance-rate$^{(l)}$
    // Gibbs step
    Sample:
        $$\sigma^{2^{(l)}}_{\text{sns}} \sim \text{Inv-Gamma}\left(\frac{n_{obs}}{2} + a,\ b + \frac{\sum_{i=1}^{n_{obs}} (\boldsymbol{d} - \boldsymbol{\beta}^{(l-1)} - \boldsymbol{A}^{(l)}\boldsymbol{s}^{(l)})^2}{2}\right)$$
        $$\boldsymbol{\beta'}^{(l)} \sim \mathrm{N}\left(\Big(\frac{1}{\sigma^{2^{(l)}}_{\text{sns}}}\mathbb{I} + \Sigma^{-1}_{\boldsymbol{\beta'}}\Big)^{-1}\Big(\frac{1}{\sigma^{2^{(l)}}_{\text{sns}}}(\boldsymbol{d'} - (\boldsymbol{A}^{(l)}\boldsymbol{s}^{(l)})') + \Sigma^{-1}_{\boldsymbol{\beta'}}\boldsymbol{\mu}_{\boldsymbol{\beta'}}\Big), \Big(\frac{1}{\sigma^{2^{(l)}}_{\text{sns}}}\mathbb{I} + \Sigma^{-1}_{\boldsymbol{\beta'}}\Big)^{-1}\right)$$
    return $\boldsymbol{\theta}^{(l)}, \sigma^{2^{(l)}}_{\text{sns}}, \boldsymbol{\beta'}^{(l)}$, acceptance-rate$^{(l)}$, $\log(p(\boldsymbol{\theta}^{(l)}|\boldsymbol{d}))$.
end
```

A.1.3 Prior specification

In this section, we present the common set of priors used during our simulation case study and the inversion on the Chilbolton dataset. Parameters estimated using M-MALA-within-Gibbs $\{\log(\boldsymbol{s}), \tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}, \log(a_H), \log(a_V), \log(b_H), \log(b_V)\}$ follow a normal distribution (Girolami and Calderhead, 2011). The prior for $\log(\boldsymbol{s})$ reflects emission ranges observed in previous facilities leaks, priors for $\tilde{\boldsymbol{x}}$ and $\tilde{\boldsymbol{y}}$ represent the center of the site with enough variance to fully cover it, and priors for $\log(a_H), \log(a_V), \log(b_H), \log(b_V)$ cover physically realistic values. We chose an inverse gamma prior for the sensor measurement error variance as it is a conjugate prior and ensures that $\sigma^2 \geq 0$. Finally, we expect gas background concentrations to be normally distributed, therefore, a normal prior is a sensible choice allowing efficient Gibbs sampling.

Distribution	**Prior**	**Value**	**Unit**
$\log(\boldsymbol{s}) \sim \mathrm{N}_{n_{\mathrm{src}}}(\boldsymbol{\mu_s}, \sigma_s^2 \mathbb{I}_{n_{\mathrm{src}}})$	μ_s	-7.5	kg/s
	σ_s^2	1.5	
$\tilde{\boldsymbol{x}} \sim \mathrm{N}_{n_{\mathrm{src}}}(\boldsymbol{\mu_x}, \sigma_x^2 \mathbb{I}_{n_{\mathrm{src}}})$	μ_x	50.0	m
	σ_x^2	25.0	
$\tilde{\boldsymbol{y}} \sim \mathrm{N}_{n_{\mathrm{src}}}(\boldsymbol{\mu_y}, \sigma_y^2 \mathbb{I}_{n_{\mathrm{src}}})$	μ_y	50.0	m
	σ_y^2	25.0	
$\log(a_H), \log(a_V) \sim \mathrm{N}(\mu_a, \sigma_a^2)$	μ_a	log(0.6)	m
	σ_a^2	0.5^2	
$\log(b_H), \log(b_V) \sim \mathrm{N}(\mu_b, \sigma_b^2)$	μ_b	log(0.6)	m
	σ_b^2	0.2^2	
$\sigma_{\mathrm{sns}}^2 \sim \text{Inv-Gamma}(a, b)$	a	1e-11	PPM
	b	1e-8	
$\boldsymbol{\beta}' \sim \mathrm{N}(\boldsymbol{\mu_{\beta'}}, \boldsymbol{\Sigma_{\beta'}})$	$\boldsymbol{\mu_{\beta'}}$	$\vec{2.0}$	PPM
	$\boldsymbol{\Sigma_{\beta'}}$	$0.1^2\mathbb{I}$	

Table A.1.1: Prior specification.

A.2 Supplementary material for Chapter 4 Section 4.4: simulation study MCMC results

Context: In the Chapter 4 Section 4.4, we performed parameter estimation on simulated scenarios using M-MALA-within-Gibbs. This section provides plots of the MCMC chains.

Content: Section A.2.1 displays the chains for different wind coverages. Section A.2.2 displays the chains for different dispersion parameter values. Section A.2.3 displays the chains for different emission rates. Section A.2.4 displays the chains for different sensor-source distances. Finally, Section A.2.5 displays the chains for different sensor layouts.

A.2.1 Chains for different wind direction coverage

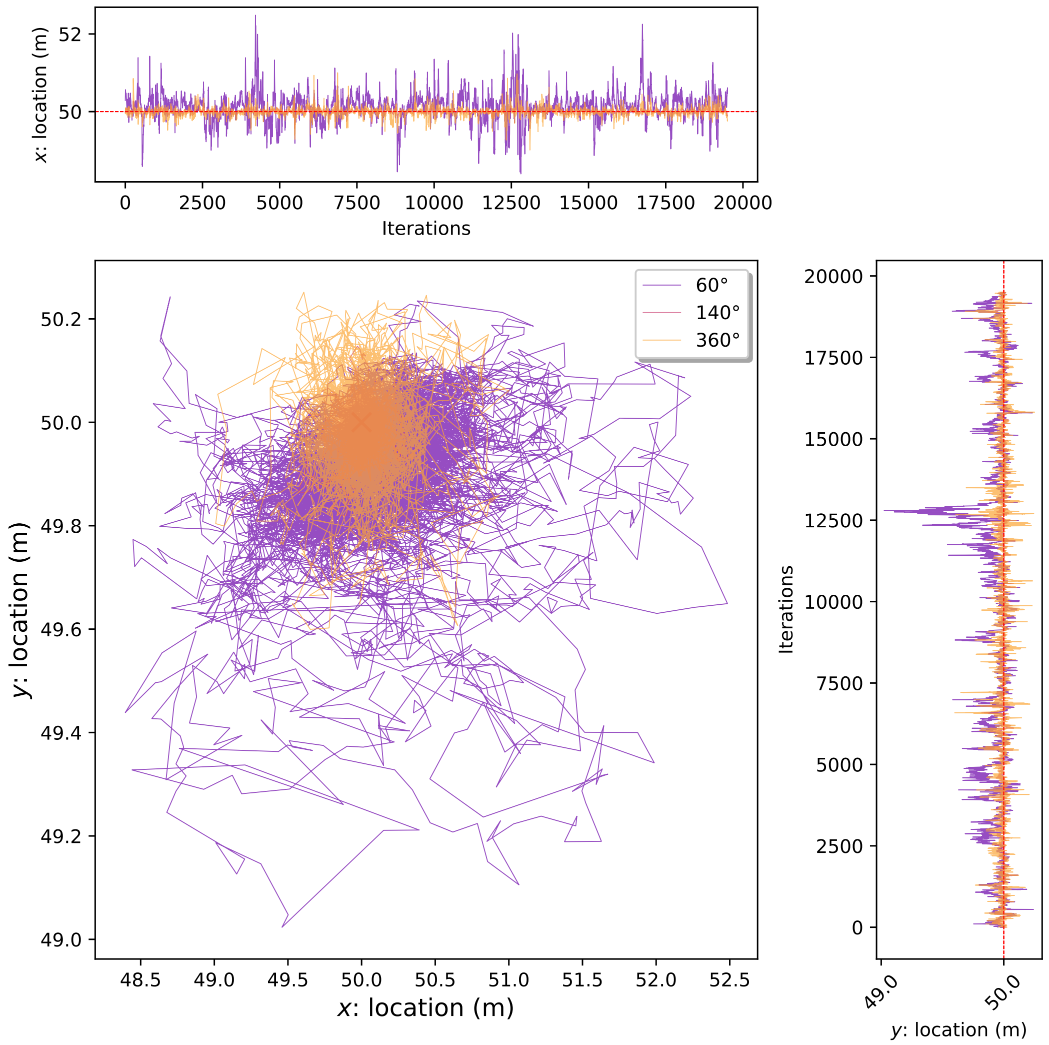


Figure A.2.1: Source location estimation chains for varying wind direction coverage.

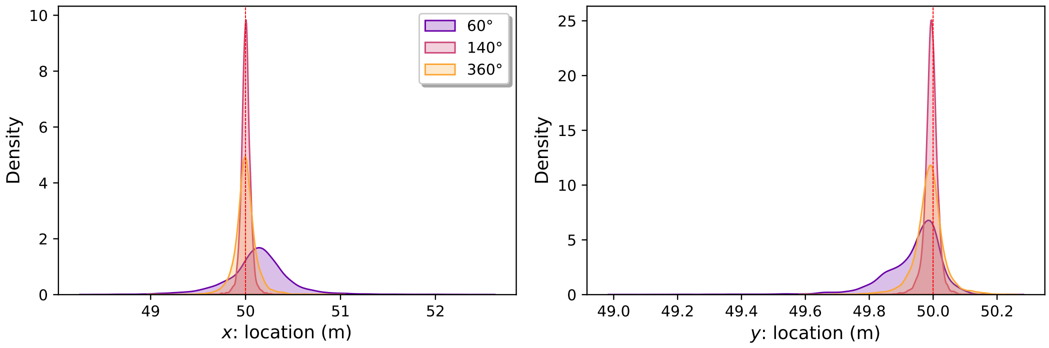


Figure A.2.2: Source location estimation densities for varying wind direction coverage.

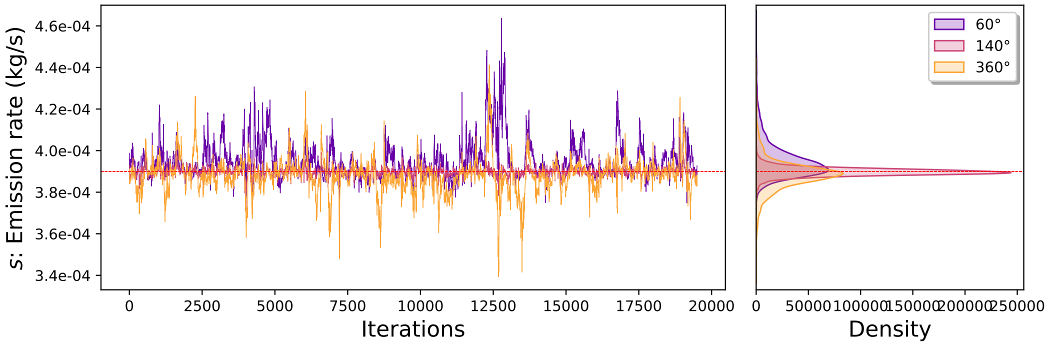


Figure A.2.3: Source emission rate location estimation chains for varying wind direction coverage.

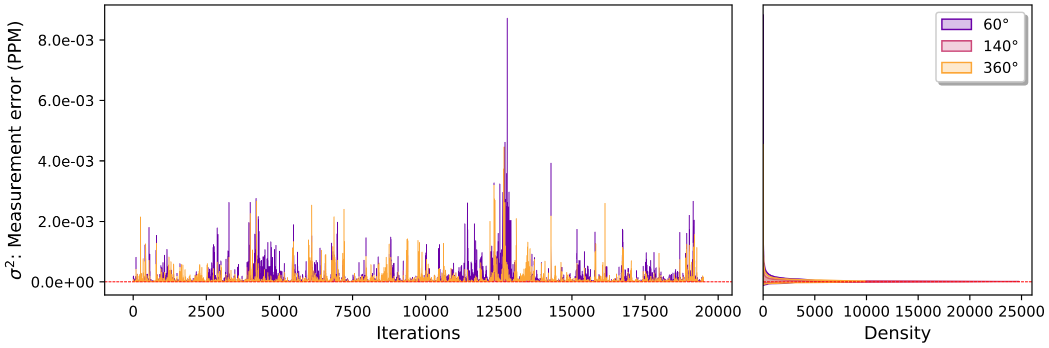


Figure A.2.4: Sensor measurement error variance estimation chains for varying wind direction coverage.

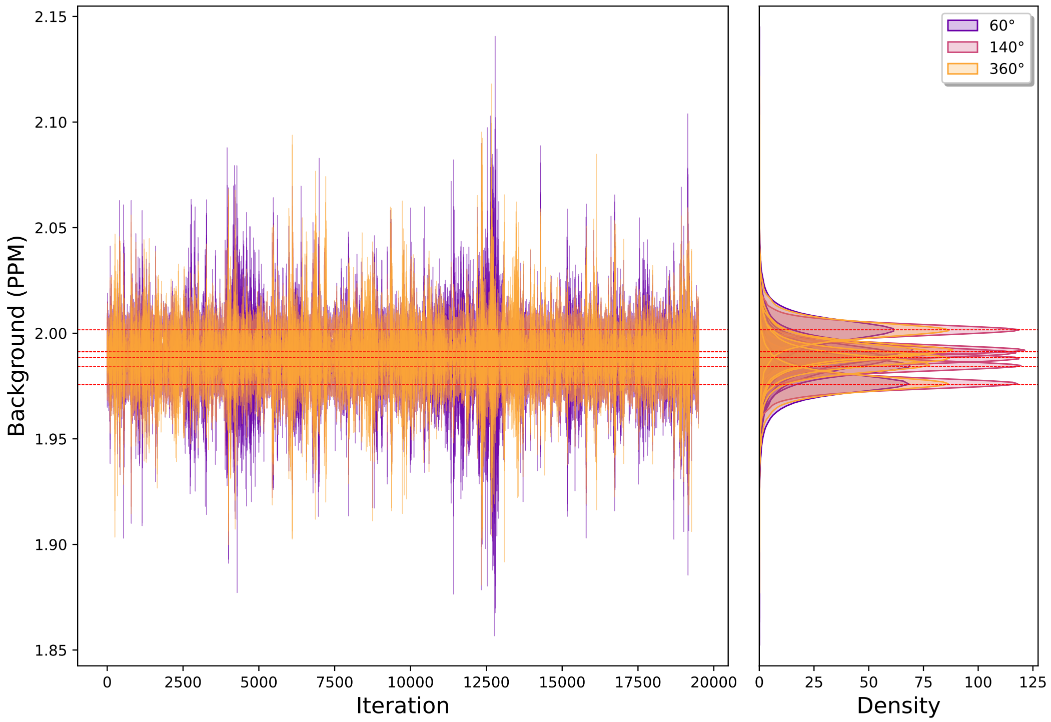


Figure A.2.5: Background concentration estimation chains for varying wind direction coverage.

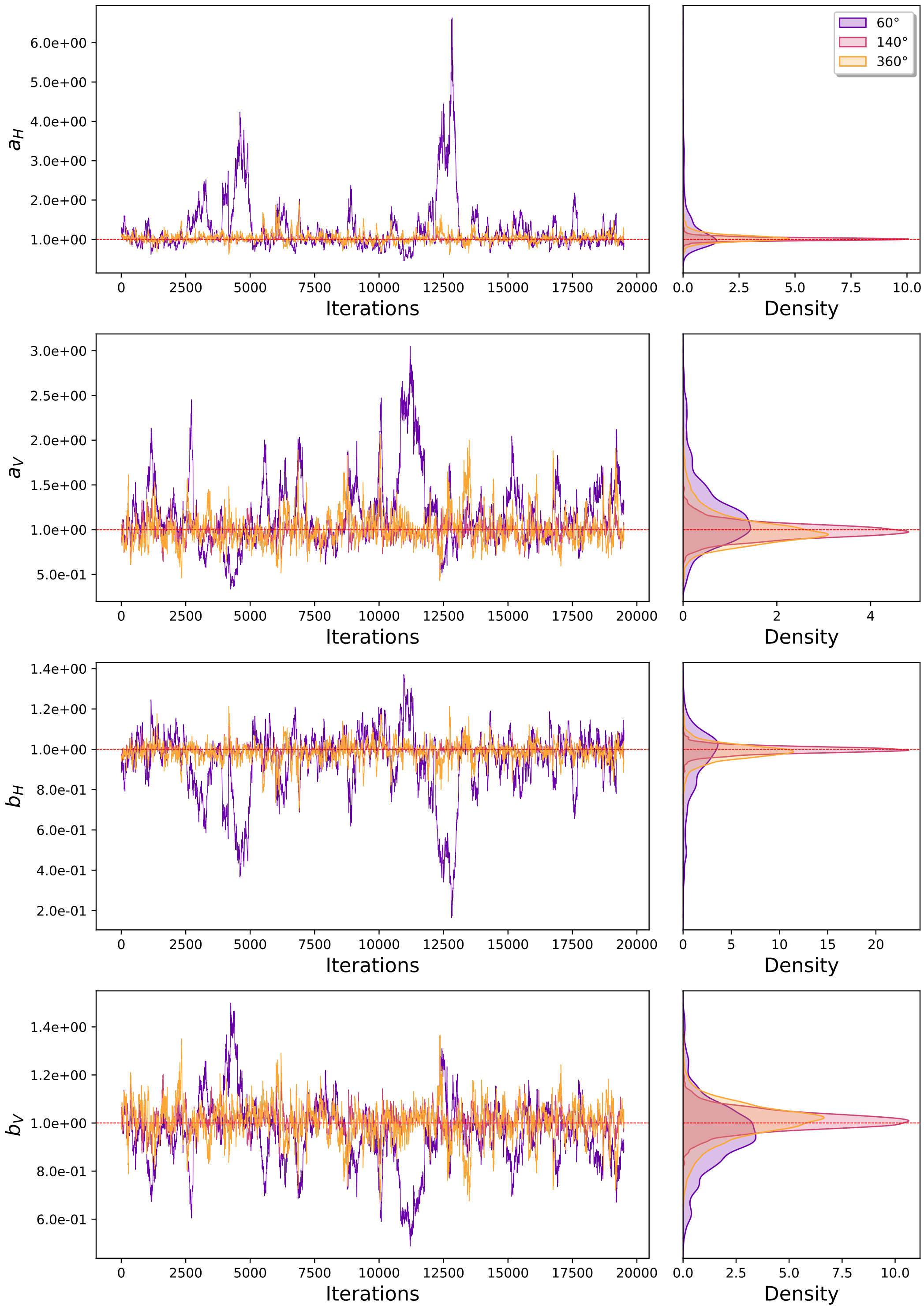


Figure A.2.6: Diffusion parameter estimation chains for varying wind direction coverage.

In-depth simulation: Here we estimated the parameters for a range of different wind direction coverage. The sharp increase in estimation accuracy and reduction in estimation uncertainty corresponds to the change from a wind direction coverage that is smaller than the plume's width to a larger one. This is illustrated in Figure A.2.9.

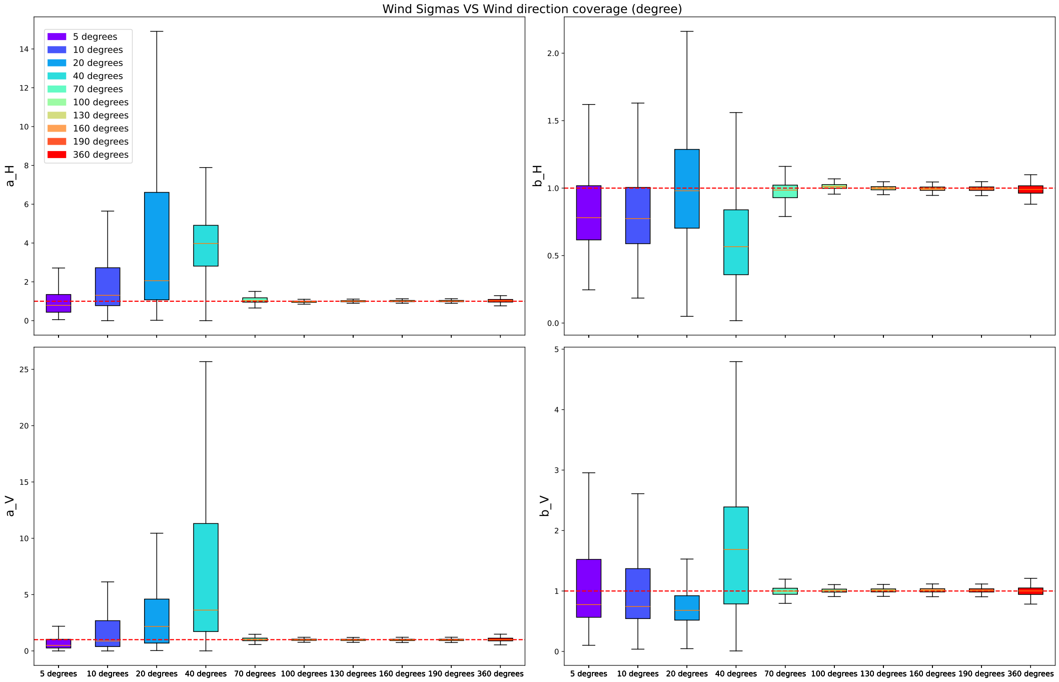


Figure A.2.7: Dispersion parameters estimation for varying sensor layout.

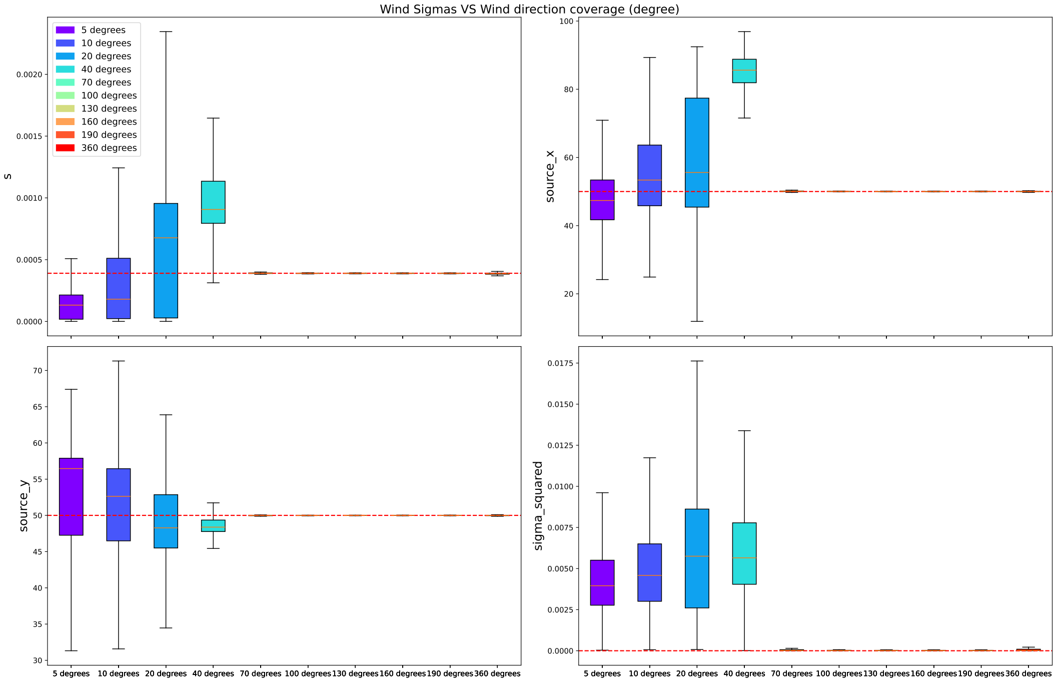


Figure A.2.8: Source emission rate, location, and sensor measurement error variance estimation for varying wind sensor layout.

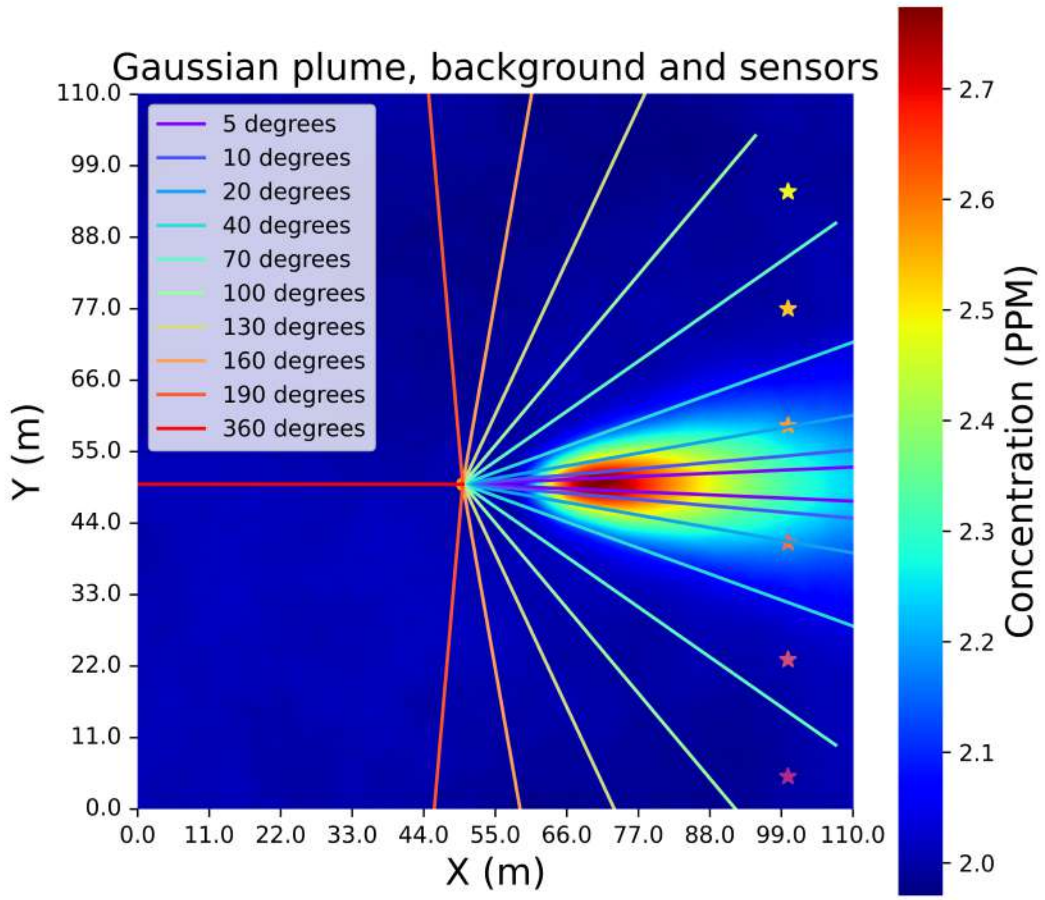


Figure A.2.9: Visual representation of varying wind direction coverage with source location and sensor layout.

A.2.2 Chains for different dispersion parameters' value

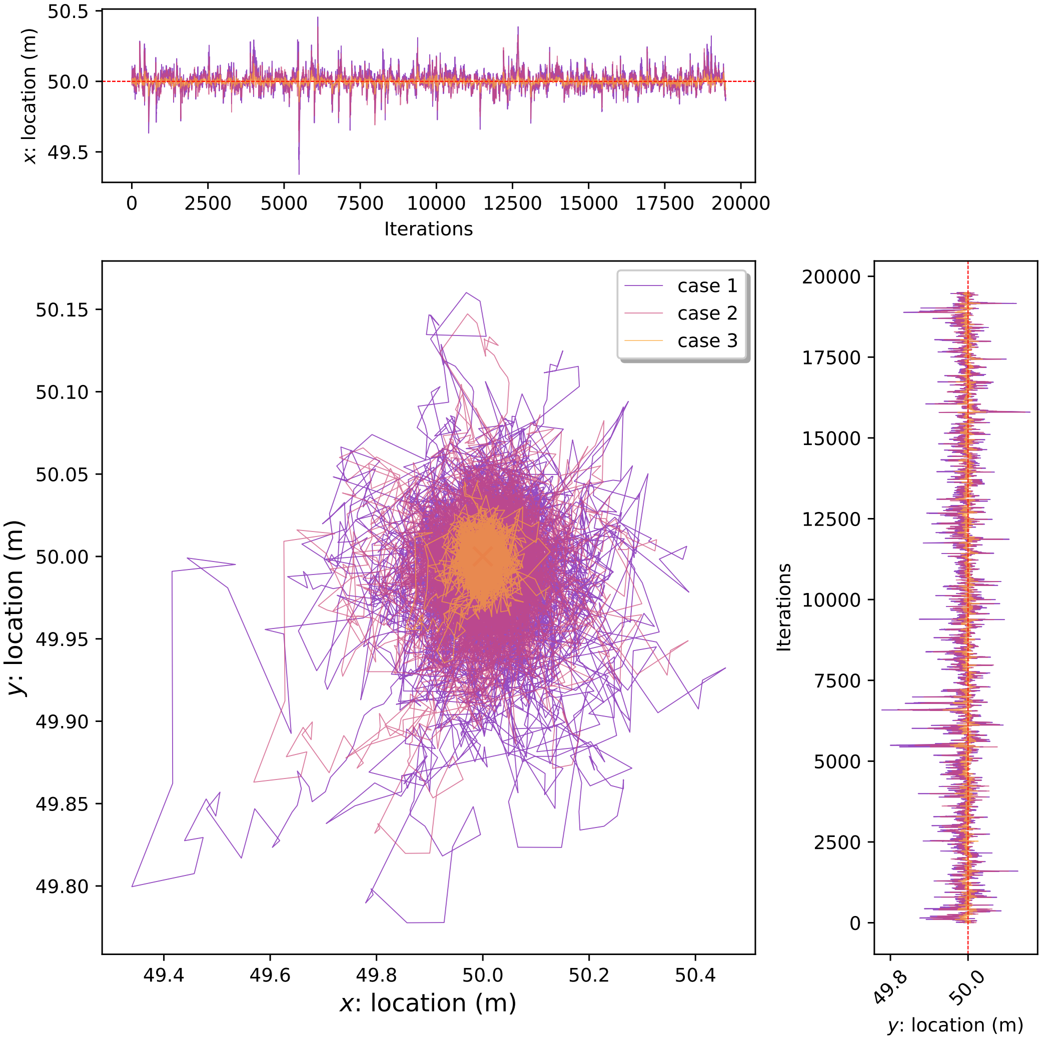


Figure A.2.10: Source location estimation chains for varying dispersion parameter values.

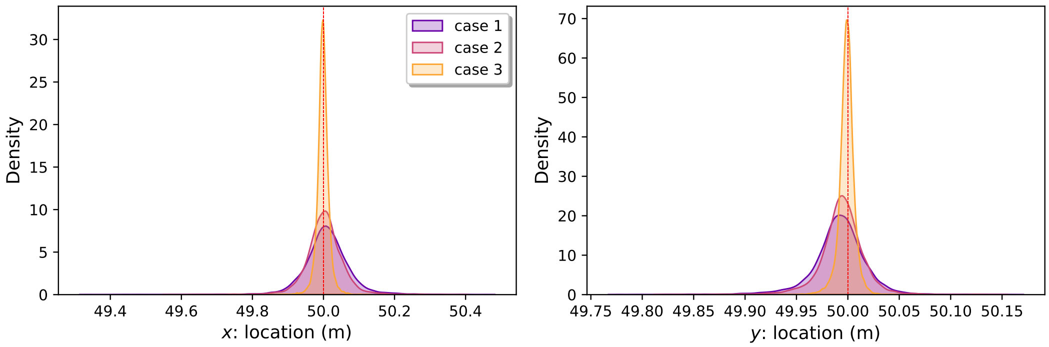


Figure A.2.11: Source location estimation densities for varying dispersion parameter values.

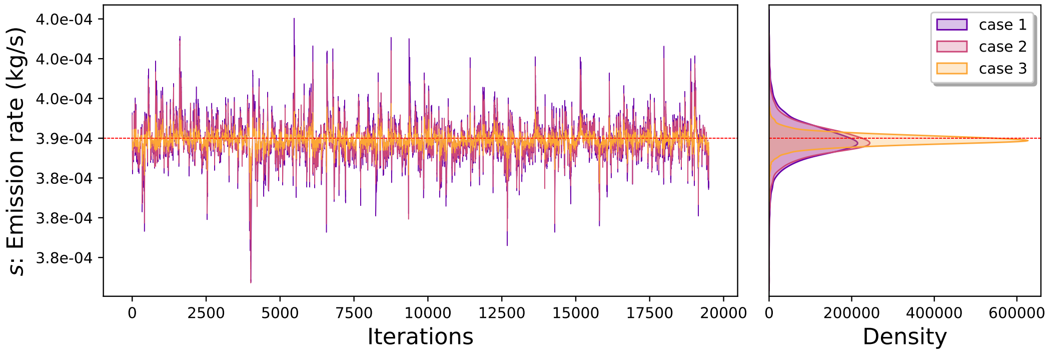


Figure A.2.12: Source emission rate location estimation chains for varying dispersion parameter values.

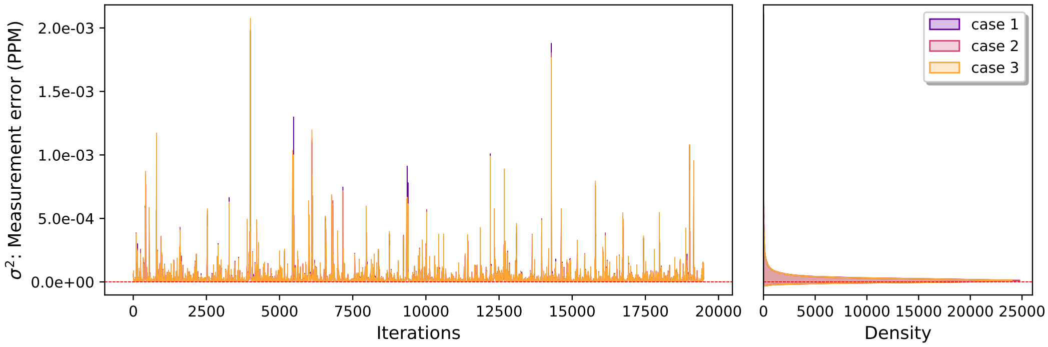


Figure A.2.13: Sensor measurement error variance estimation chains for varying dispersion parameter values.

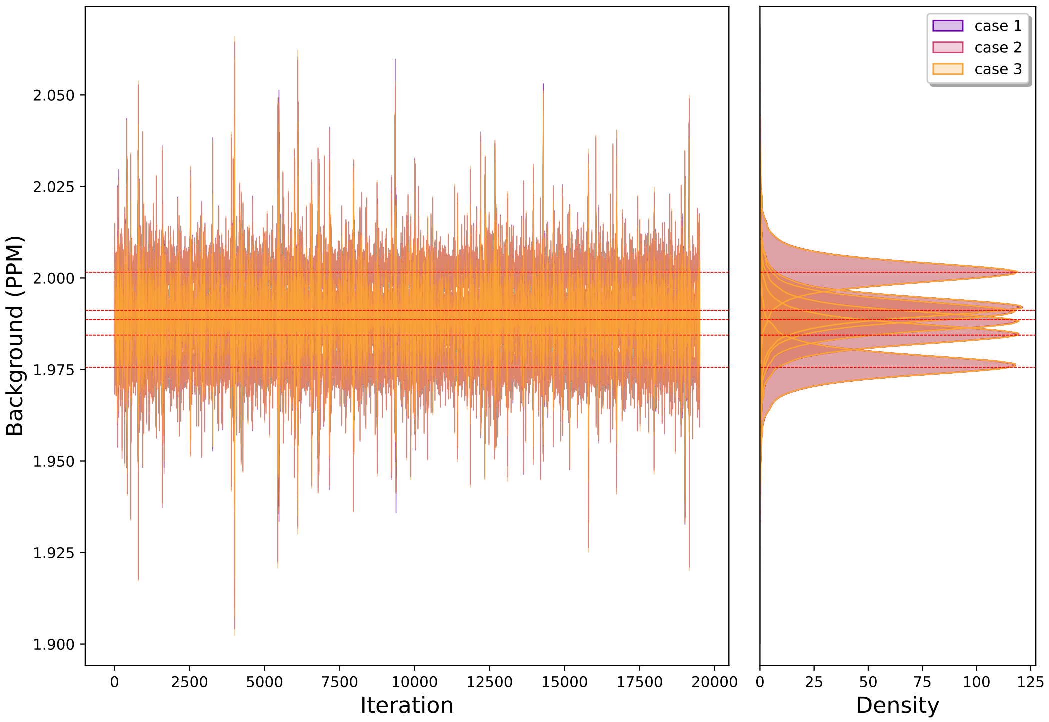


Figure A.2.14: Background concentration estimation chains for varying dispersion parameter values.

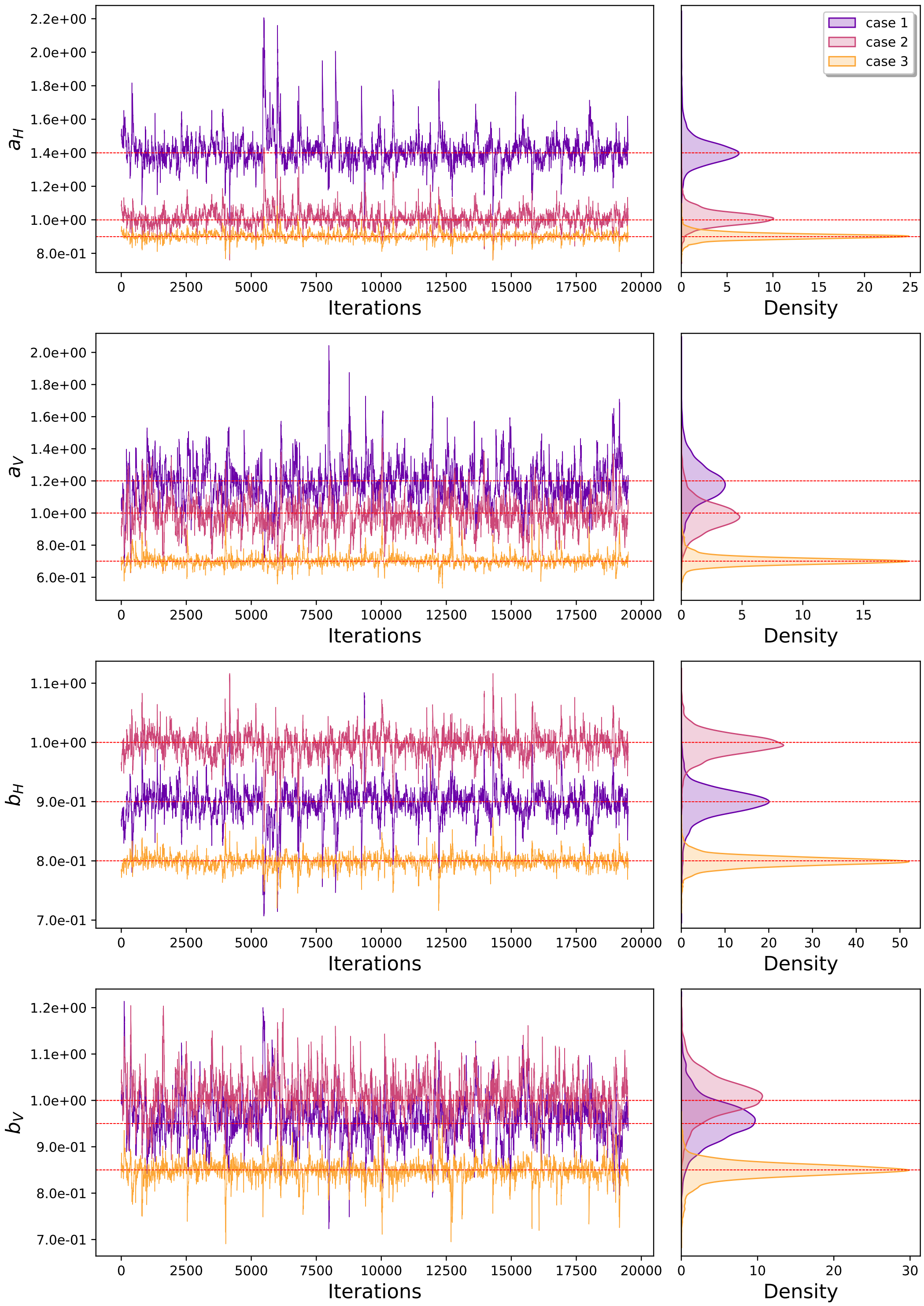


Figure A.2.15: Diffusion parameter estimation chains for varying dispersion parameter values.

A.2.3 Chains for different source emission rate

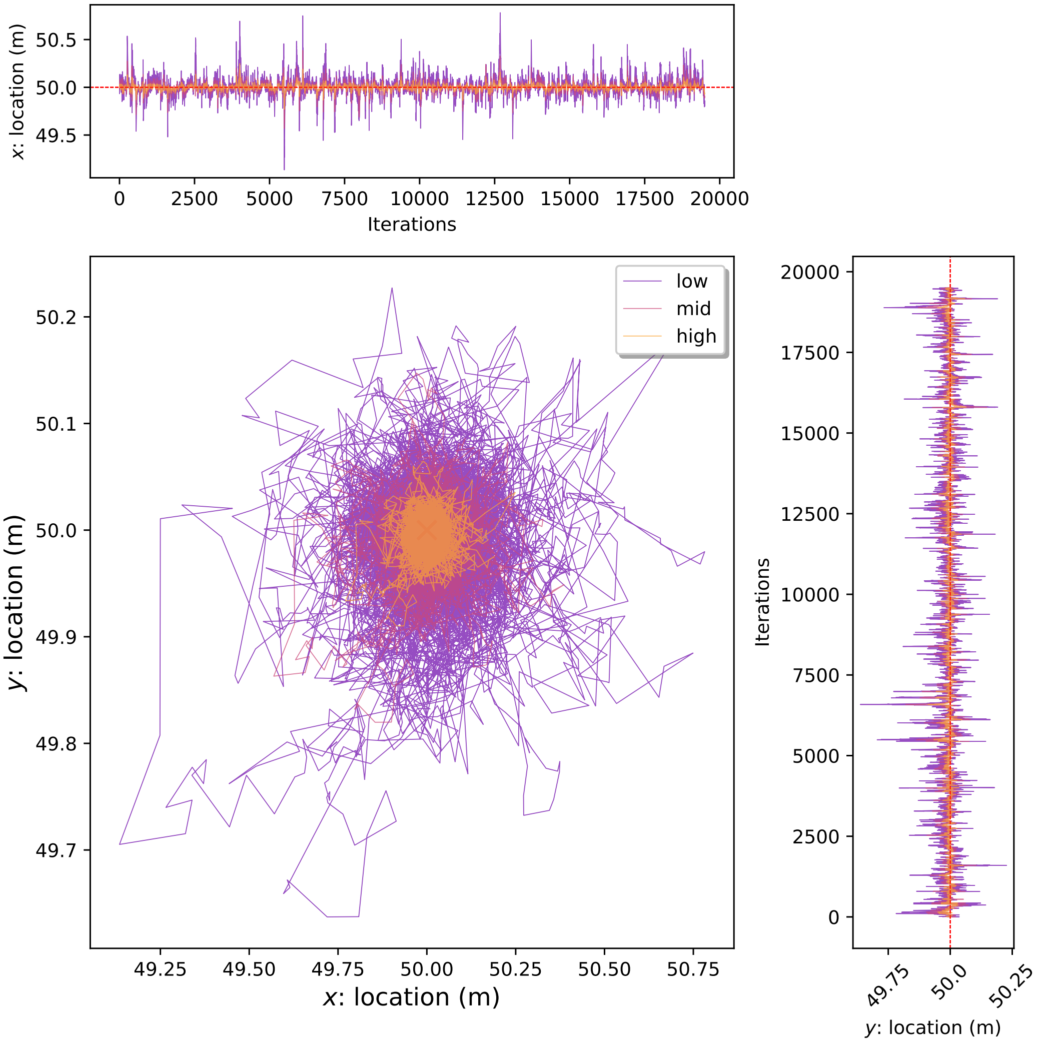


Figure A.2.16: Source location estimation chains for varying source emission rate.

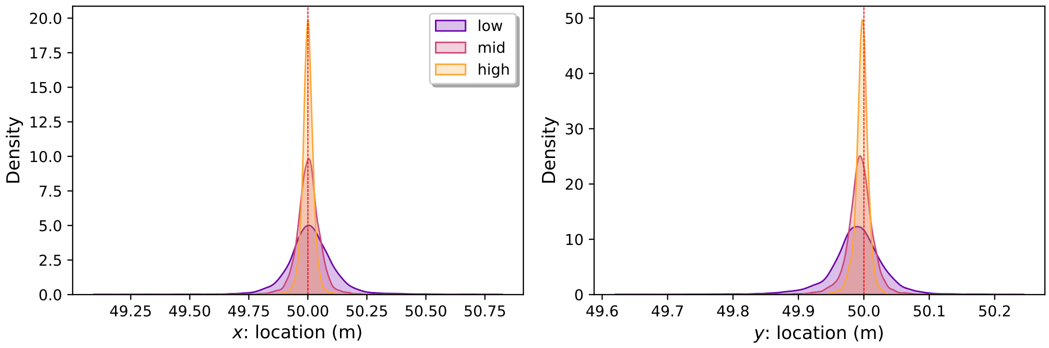


Figure A.2.17: Source location estimation densities for varying source emission rate.

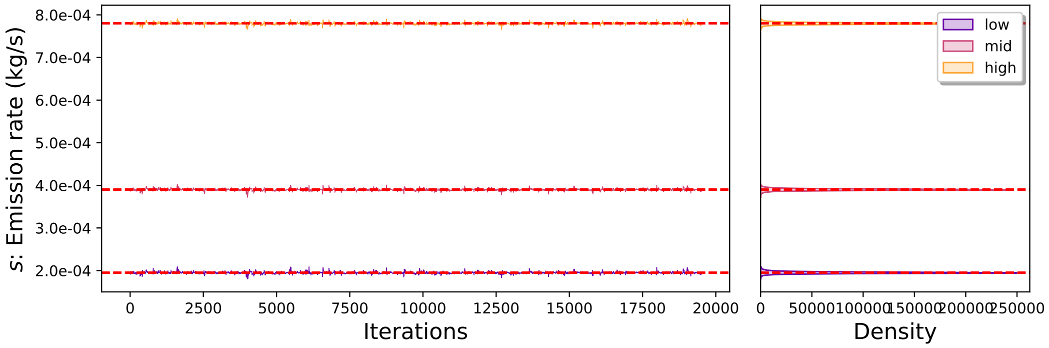


Figure A.2.18: Source emission rate location estimation chains for varying source emission rate.

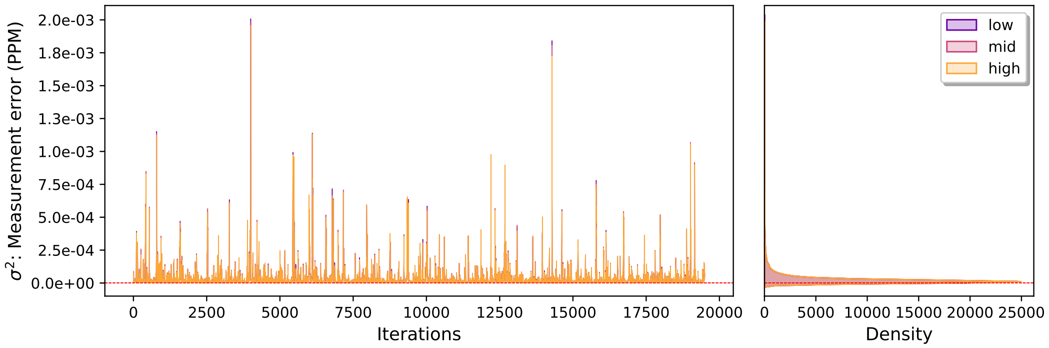


Figure A.2.19: Sensor measurement error variance estimation chains for varying source emission rate.

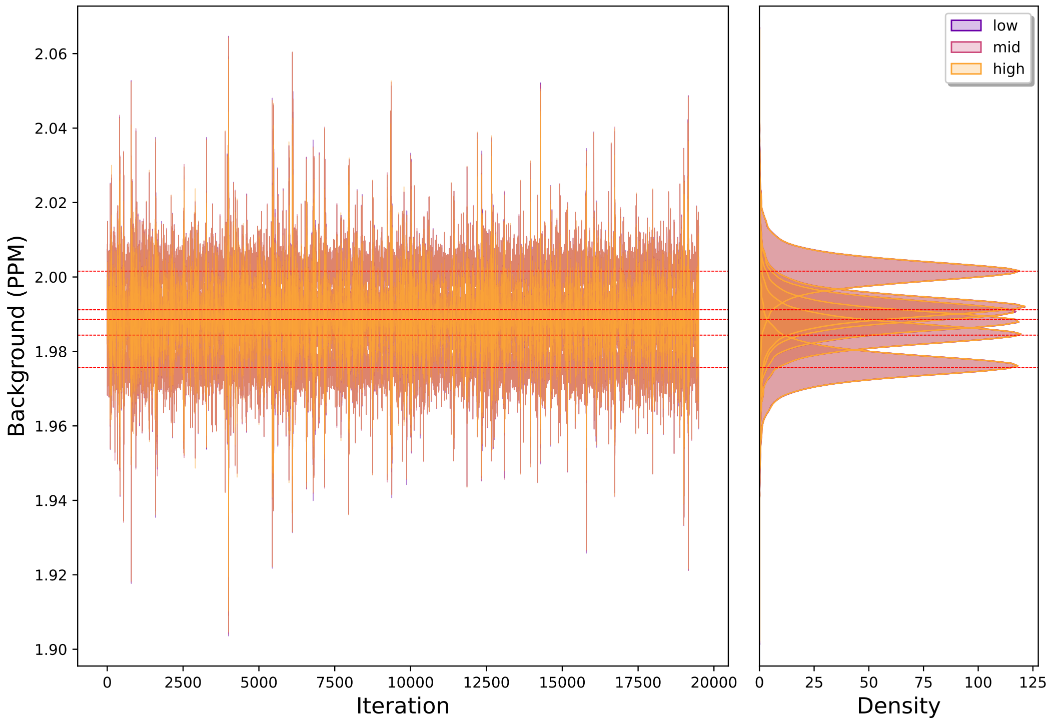


Figure A.2.20: Background concentration estimation chains for varying source emission rate.

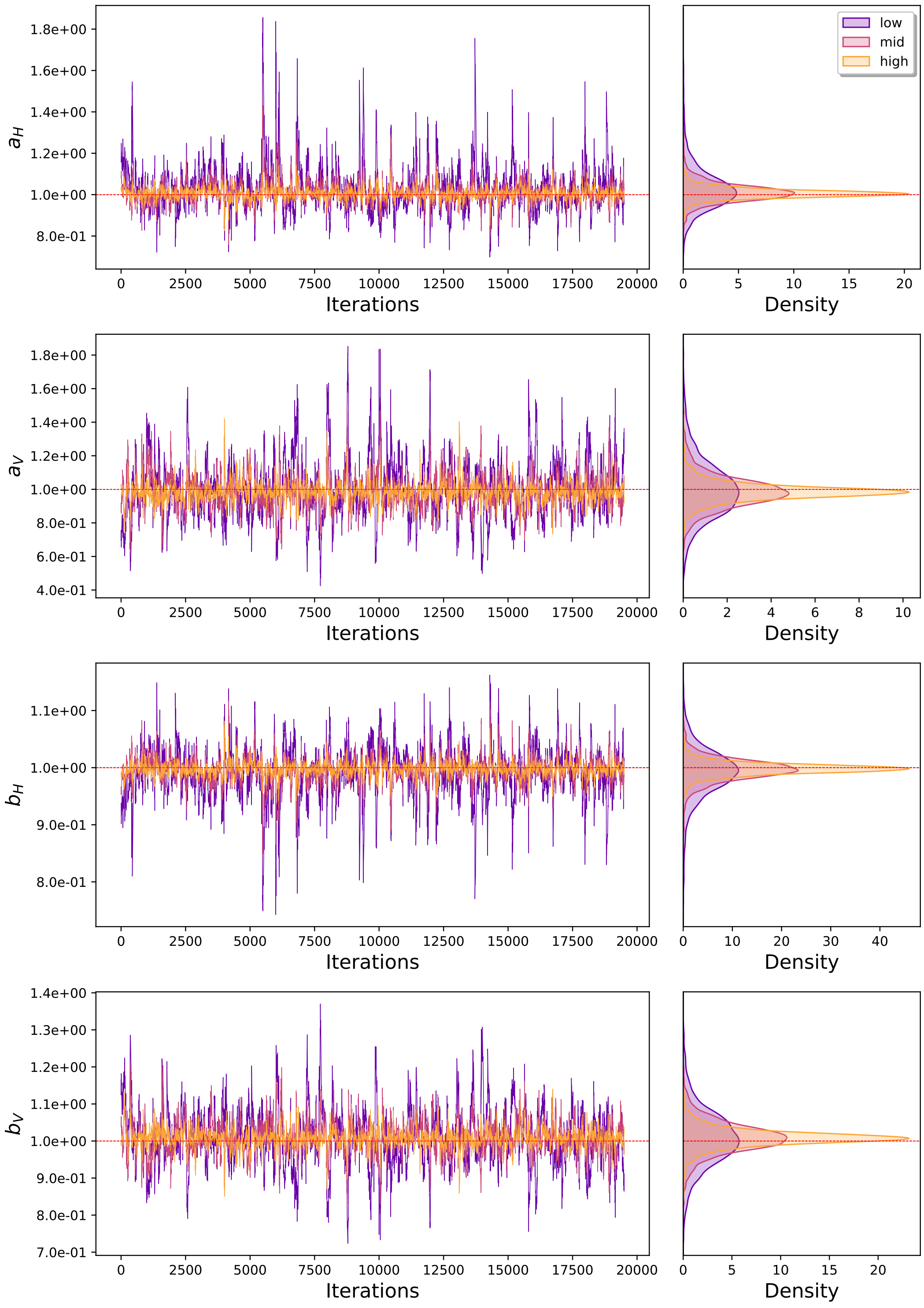


Figure A.2.21: Diffusion parameter estimation chains for varying source emission rate.

A.2.4 Chains for different source to sensor distance

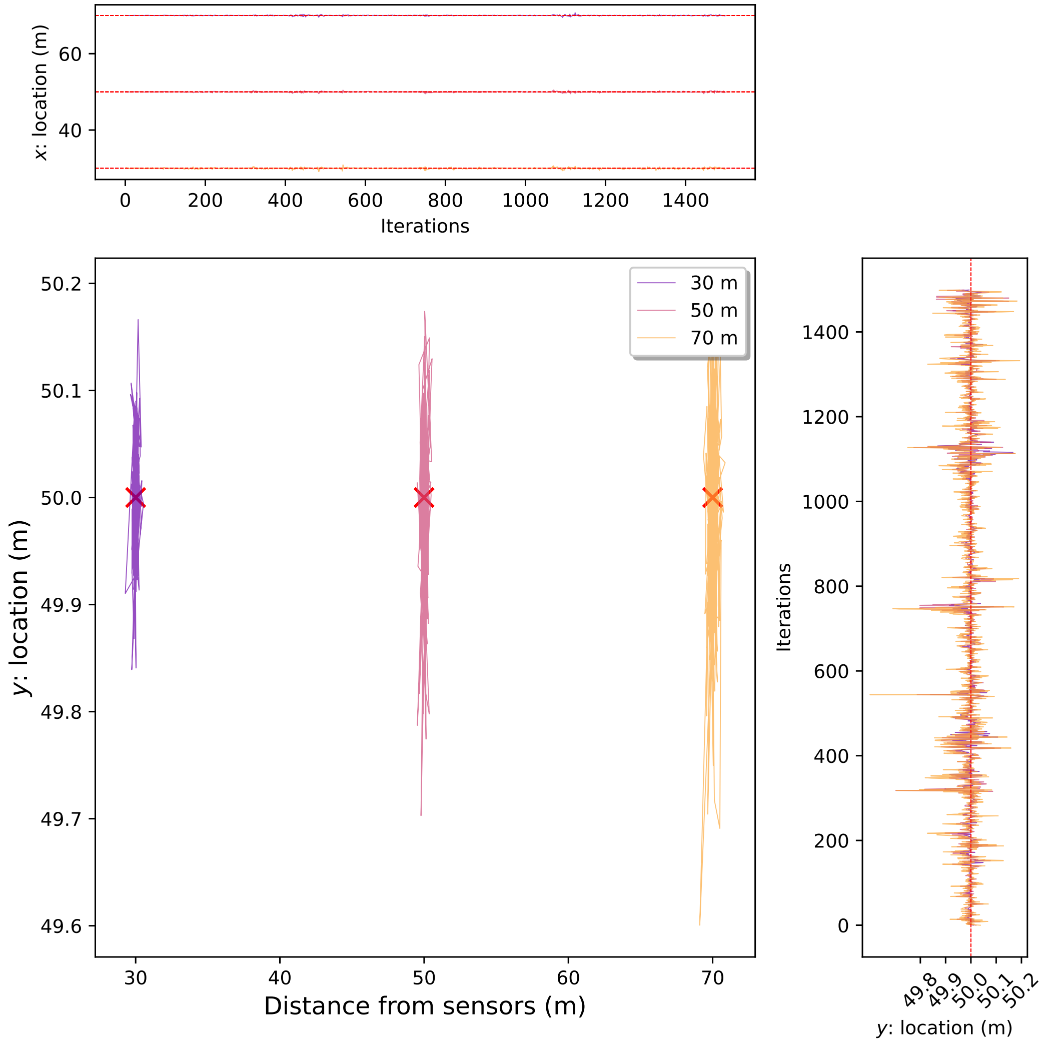


Figure A.2.22: Source location estimation chains for varying distance between source and sensors.

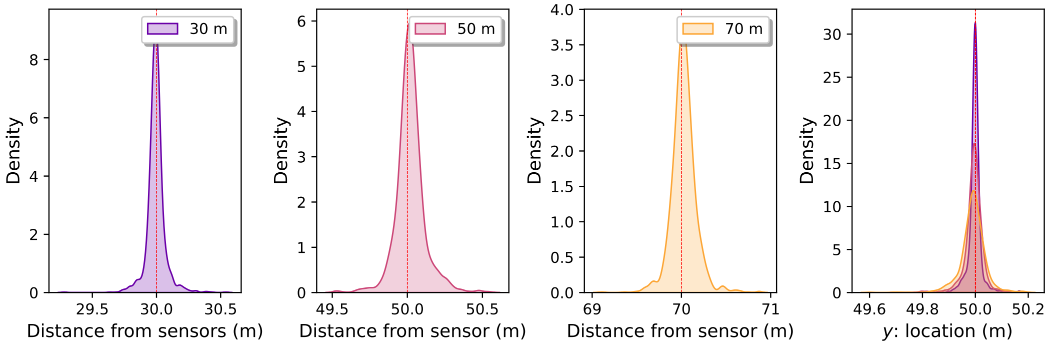


Figure A.2.23: Source location estimation densities for varying distance between source and sensors.

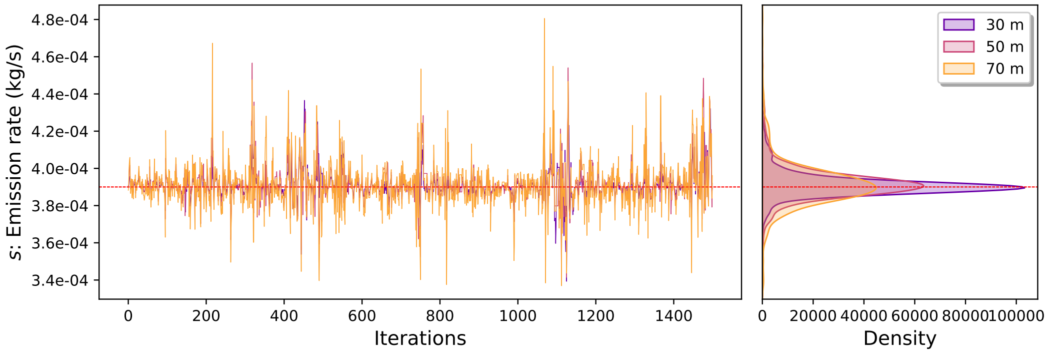


Figure A.2.24: Source emission rate location estimation chains for varying distance between source and sensors.

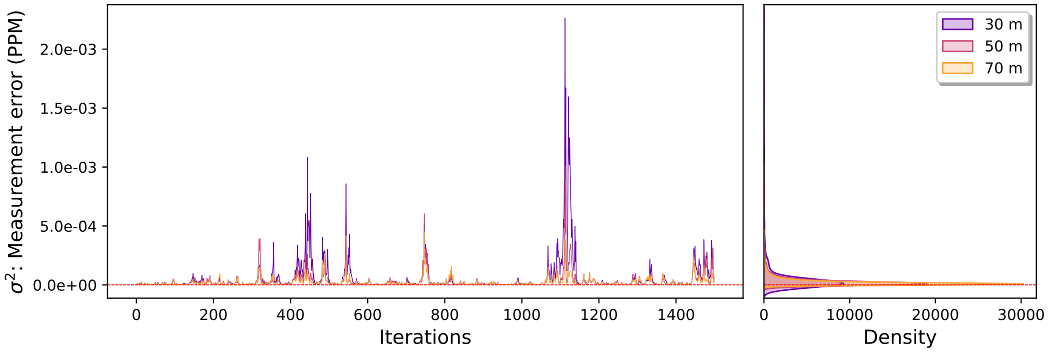


Figure A.2.25: Sensor measurement error variance estimation chains for varying distance between source and sensors.

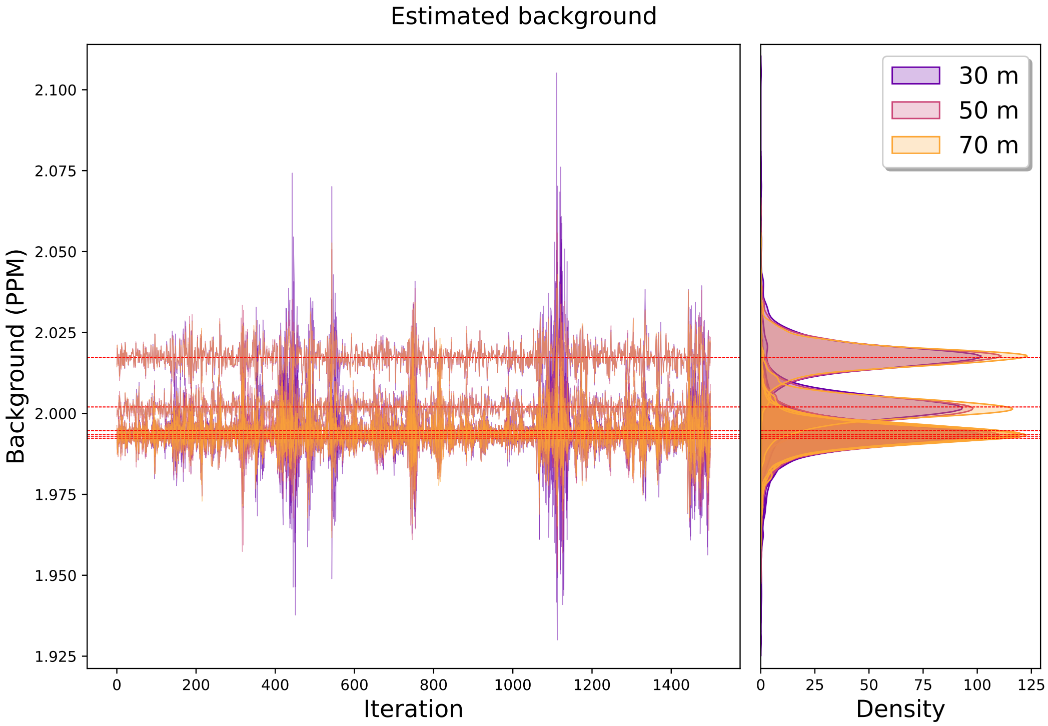


Figure A.2.26: Background concentration estimation chains for varying distance between source and sensors.

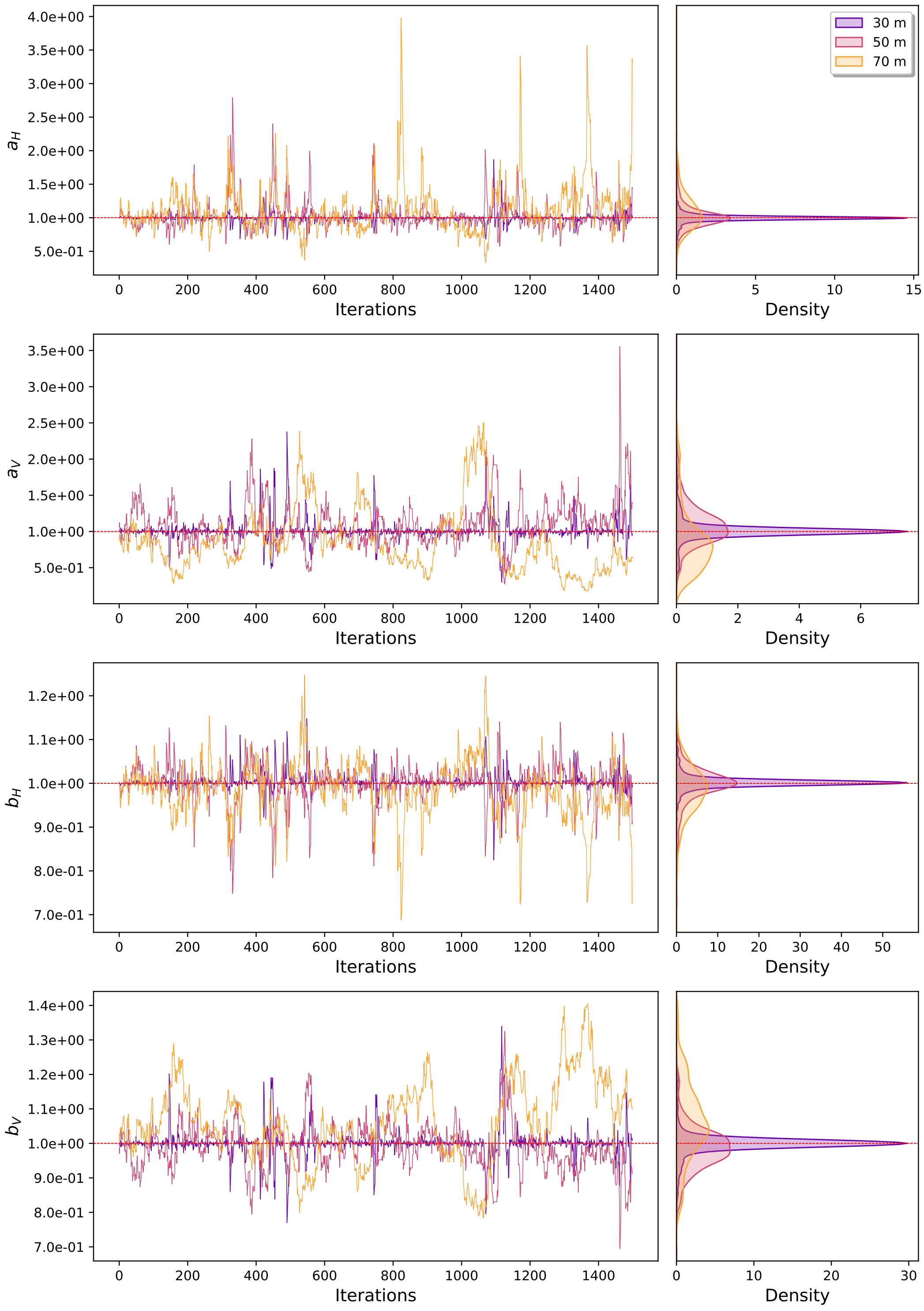


Figure A.2.27: Diffusion parameter estimation chains for varying distance between source and sensors.

A.2.5 Chains for different sensor layout

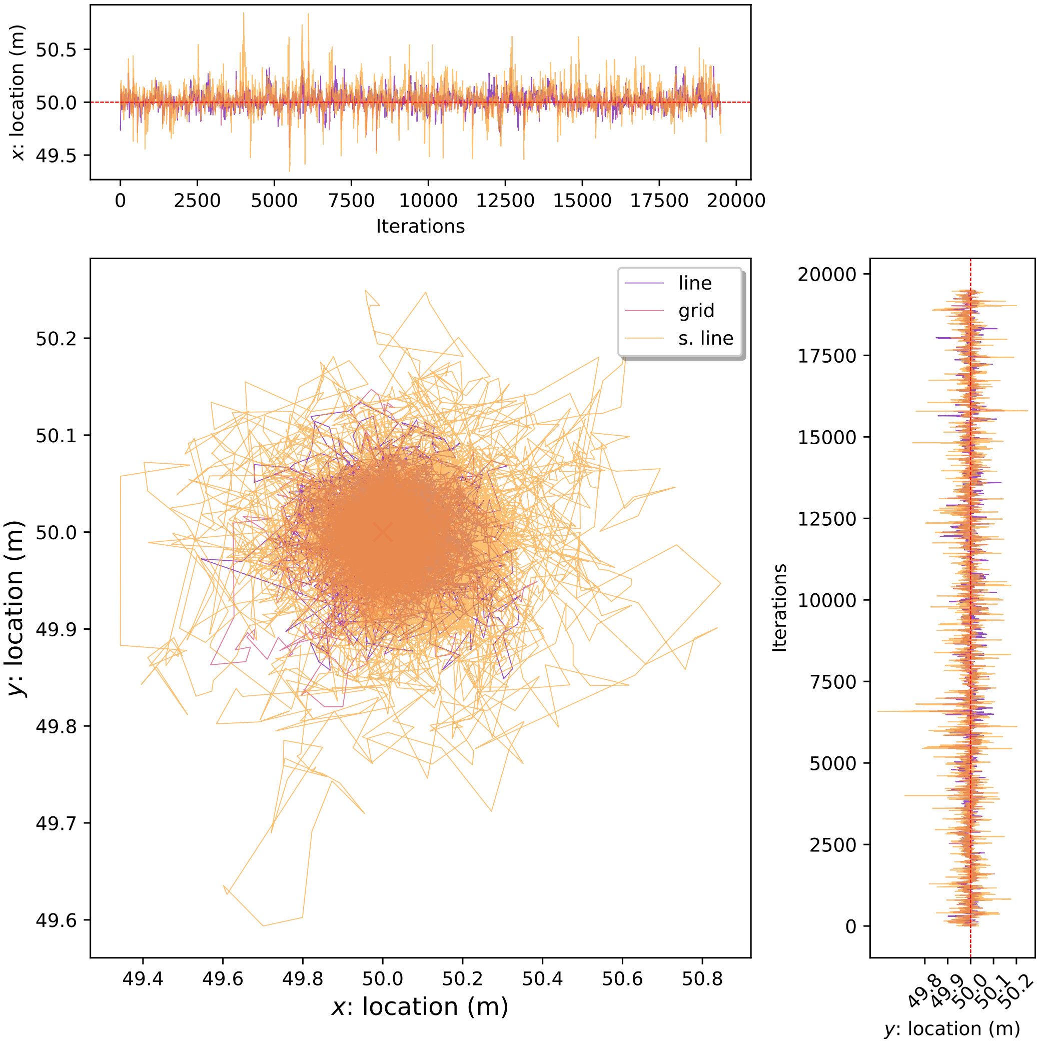


Figure A.2.28: Source location estimation chains for varying the sensor layout. **line**: a line of 36x1 sensors, **grid**: a 6x6 grid of sensors, and **s.line**: a line of 6x1 sensors.

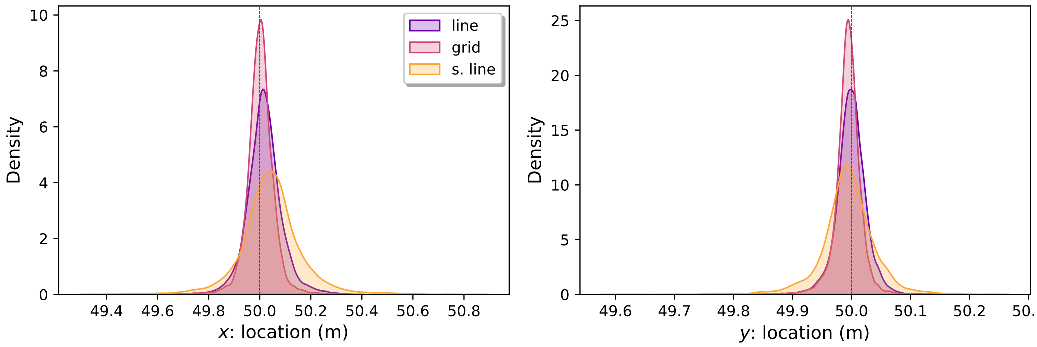


Figure A.2.29: Source location estimation densities for varying the sensor layout. **line**: a line of 36x1 sensors, **grid**: a 6x6 grid of sensors, and **s.line**: a line of 6x1 sensors.

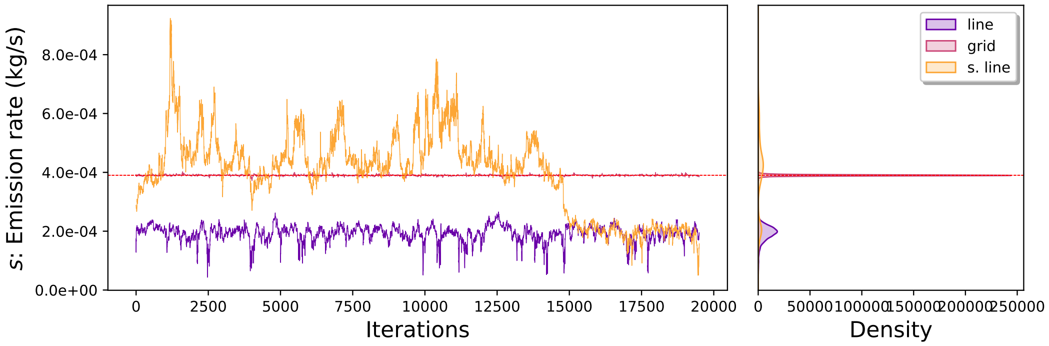


Figure A.2.30: Source emission rate location estimation chains for varying the sensor layout. **line**: a line of 36x1 sensors, **grid**: a 6x6 grid of sensors, and **s.line**: a line of 6x1 sensors.

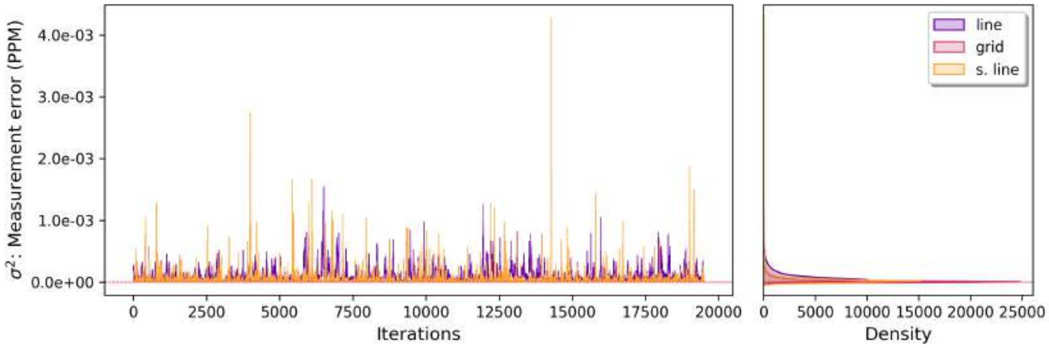


Figure A.2.31: Sensor measurement error variance estimation chains for varying the sensor layout. **line**: a line of 36x1 sensors, **grid**: a 6x6 grid of sensors, and **s.line**: a line of 6x1 sensors.

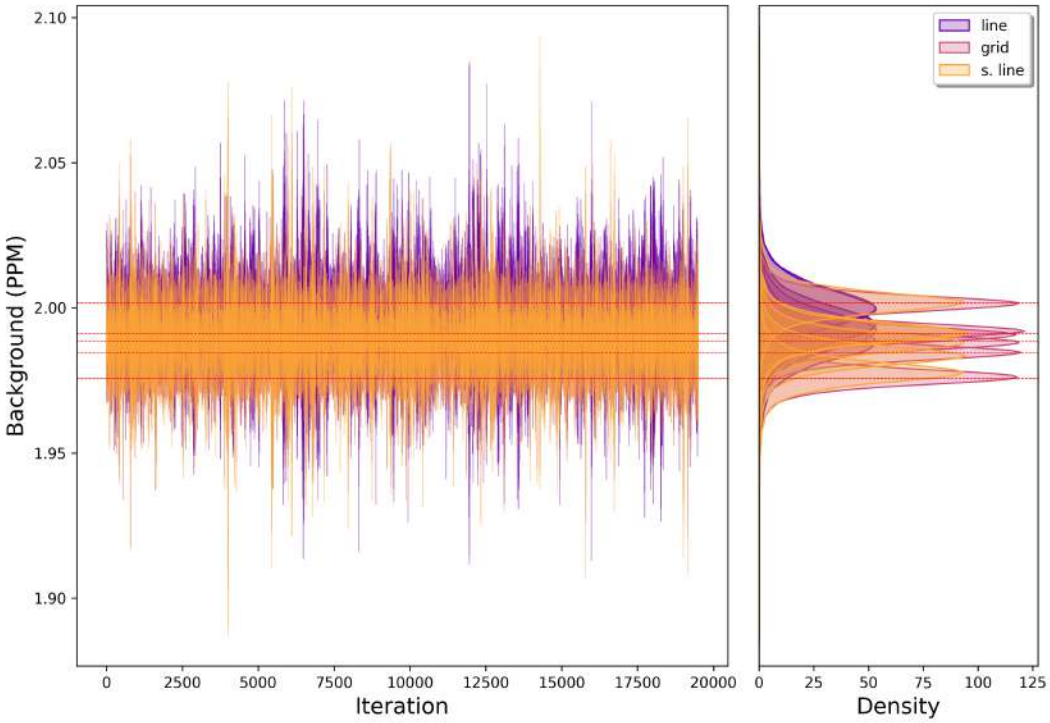


Figure A.2.32: Background concentration estimation chains for varying the sensor layout. **line**: a line of 36x1 sensors, **grid**: a 6x6 grid of sensors, and **s.line**: a line of 6x1 sensors.

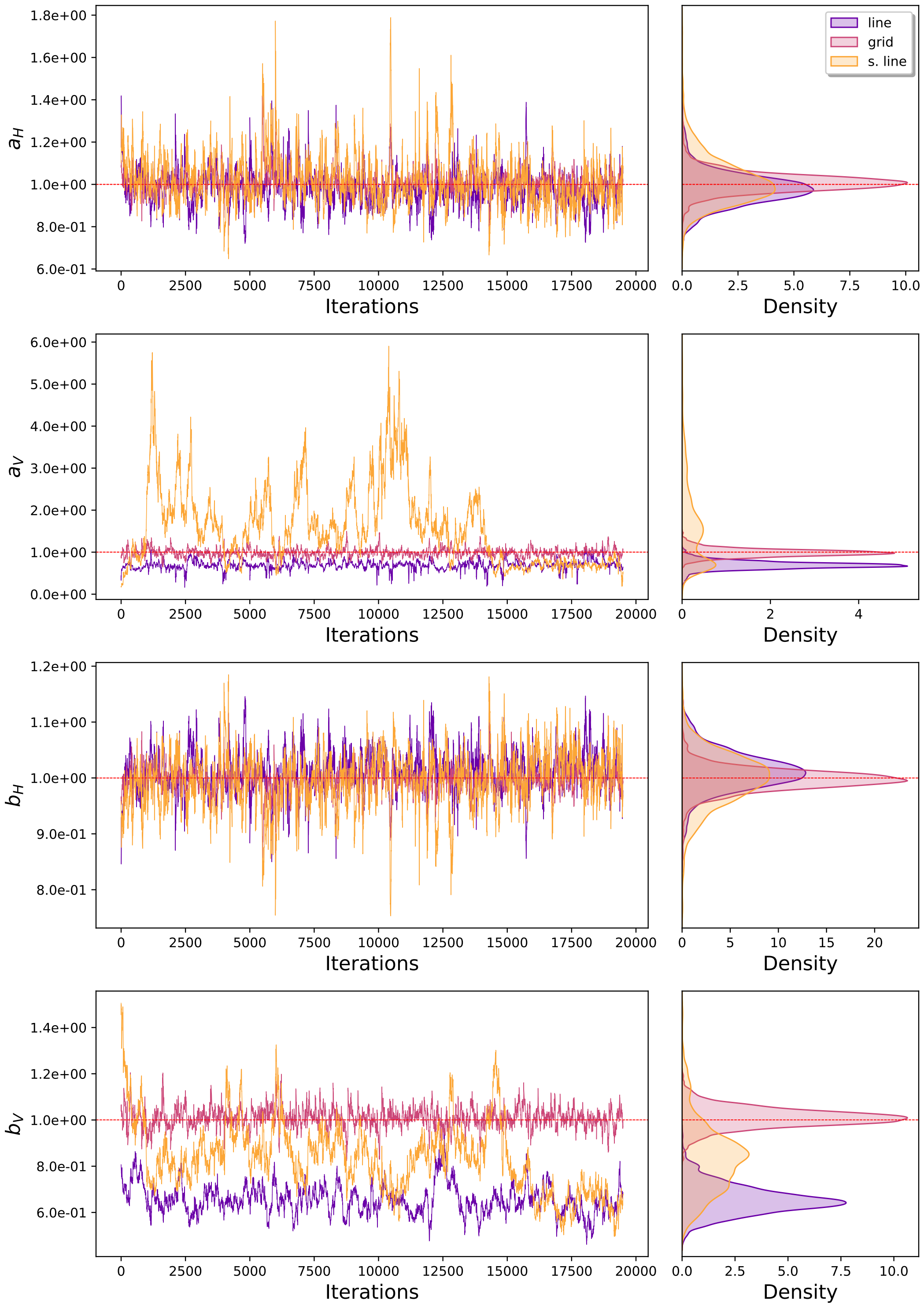


Figure A.2.33: Diffusion parameter estimation chains for varying the sensor layout. **line**: a line of 36x1 sensors, **grid**: a 6x6 grid of sensors, and **s.line**: a line of 6x1 sensors.

In-depth simulation: Here we estimated the parameters for a range of sensor layouts and different numbers of sensors: 4, 9, 16, and 36. We have grid formations: $2 \times 2, 3 \times 3, 4 \times 4$ and 6×6 and line formations: $4 \times 1, 9 \times 1, 16 \times 1$ and 36×1. The loss of vertical coverage in the line formations (equally spaced sensors at fixed height impacts the ability to accurately estimate the vertical dispersion parameters correctly and impacts the emission rate estimation. To explain the data, this increases the measurement error variance. However, grid formations show greater robustness even with a low number of sensors.

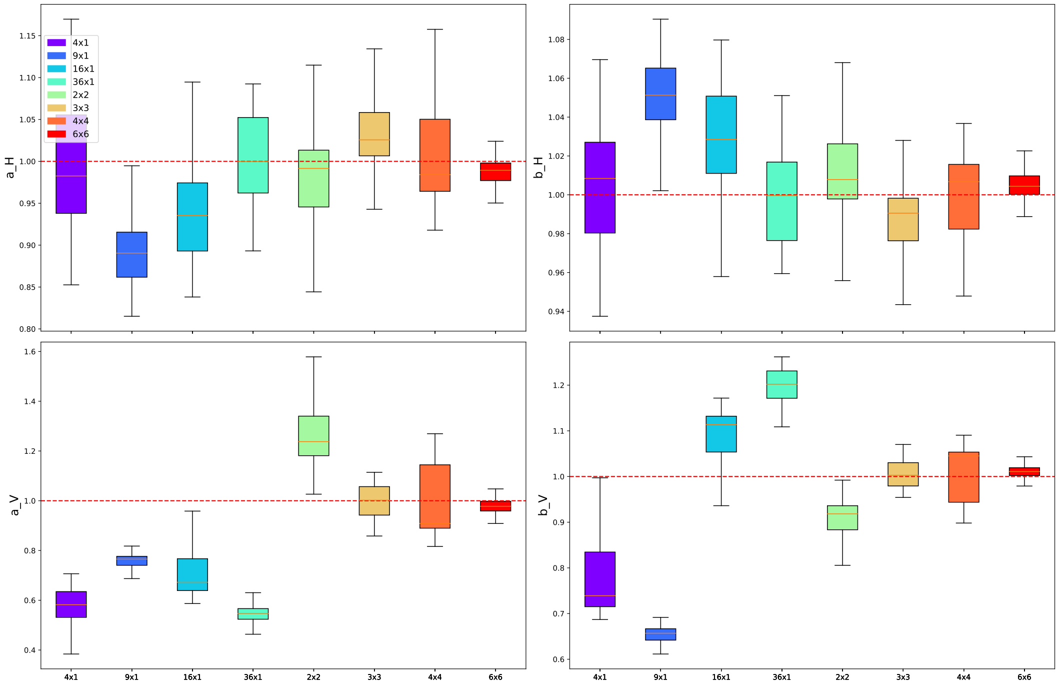


Figure A.2.34: Dispersion parameters estimation for varying the sensor layout. We are comparing line layouts and grid layouts for different number of sensors.

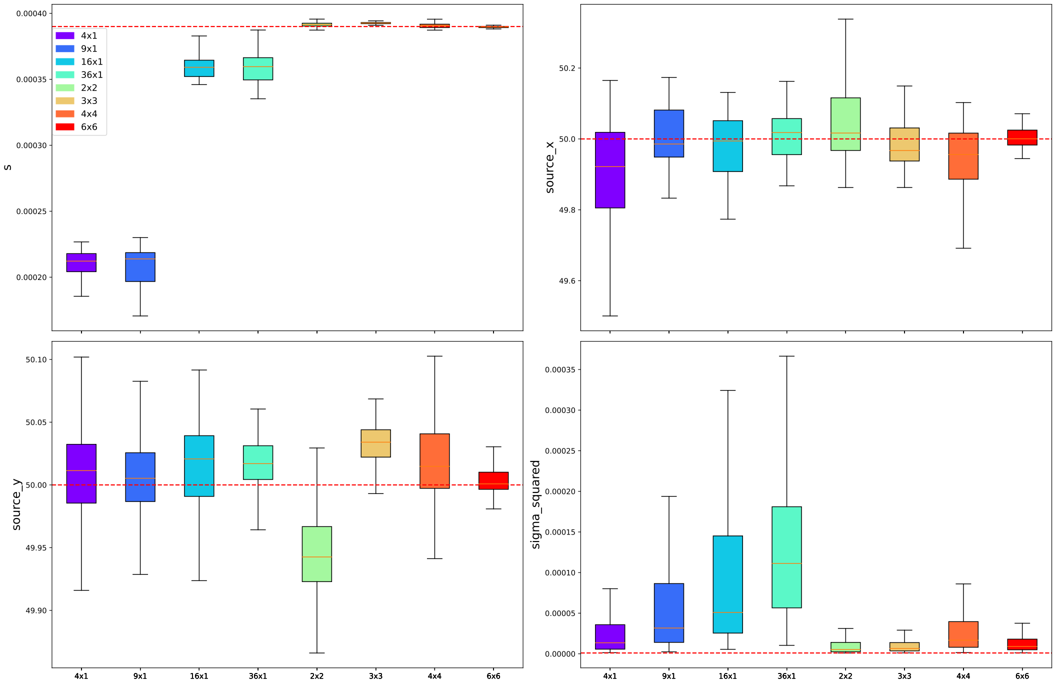


Figure A.2.35: Source emission rate, location, and sensor measurement error variance estimation for varying the sensor layout. We are comparing line layouts and grid layouts for different number of sensors.

A.3 Supplementary material for Chapter 4 Section 4.5: Chilbolton

Context: In the Chapter 4 Section 4.5, we performed parameter estimation on real-data using M-MALA-within-Gibbs. This section provides additional information regarding how this was performed.

Content: Section A.3.1 provides additional information regarding the data processing. Sections A.3.2 and A.3.3 introduce the SMITH and Briggs schemes and corresponding atmospheric stability class-based dispersion parameters tables. Section A.3.4 presents Pasquill's atmospheric stability class and presents our attempt at estimating the class during the Chilbolton releases. Finally, Sections A.3.5 and A.3.6 contain the MCMC chains for Source 1 and Source 2 parameter estimation.

A.3.1 Data processing

The GitHub repository available at `https://github.com/NewmanTHP/Probabilistic-Inversion-Modeling-of-Gas-Emissions/tree/master/Code/Chilbolton_Case_Study%20(Section%205%20%2B%20SM.B%202)/Data%20Processing%20(SM.B%202.1)` contains Jupyter notebooks detailing how the raw data was processed.

A.3.2 SMITH scheme

The SMITH scheme used to fix the wind sigma parameters in the Gaussian plume model has the following parametric form:

$$\sigma_H = a_H \delta_R^{b_H},$$

$$\sigma_V = a_V \delta_R^{b_V},$$

where $\delta_R \in \mathbb{R}^+$ is the downwind distance. The dispersion parameters a_H, b_H, a_V, b_V are chosen based on the local atmospheric stability class (ASC) at the time the data was collected. Once the ASC is determined, the dispersion parameters can be chosen using Table A.3.1 from Hanna et al. (1982).

ASC	a_H	b_H	a_V	b_V
B	0.4	0.91	0.41	0.91
C	0.36	0.86	0.33	0.86
D	0.32	0.78	0.22	0.78

Table A.3.1: Briggs ASC-based dispersion parameter table

A.3.3 Briggs scheme

The Briggs scheme used to fix the wind sigma parameters in the Gaussian plume model has the following parametric form:

$$\sigma_H = a\delta_R(1 + 0.0001\delta_R)^{-0.5},$$

$$\sigma_V = \begin{cases} b\delta_R & \text{if ASC is A or B} \\ b\delta_R(1 + c\delta_R)^d & \text{if ASC is C, D, E or, F.} \end{cases}$$

where $\delta_R \in \mathbb{R}^+$ is the downwind distance. The dispersion parameters a, b, c, d are chosen based on the local atmospheric stability class (ASC) at the time the data was collected. Once the ASC is determined, the dispersion parameters can be chosen using Table A.3.2 from Hanna et al. (1982).

ASC	a	b	c	d
A	0.22	0.20	0	0
B	0.16	0.12	0	0
C	0.11	0.08	0.0002	-0.5
D	0.08	0.06	0.0015	-0.5
E	0.06	0.03	0.0003	-1
F	0.04	0.016	0.0003	-1

Table A.3.2: Briggs ASC-based dispersion parameter table for open country conditions

A.3.4 Atmospheric stability class determination

Pasquill's ASC system is determined by wind speed and net solar radiation index. The following classification table, Table A.3.3, is drawn.

Wind Speed (m/s)	**Daytime incoming solar radiation**		
	Strong	**Moderate**	**Slight**
<2	A	A-B	B
2-3	A-B	B	C
3-5	B	B-C	C
5-6	C	C-D	D
>6	C	D	D

Table A.3.3: Pasquill ASC table

In Table A.3.3, strong $> 700\text{Wm}^2$, moderate 350-700Wm^2, and slight $< 350\text{Wm}^2$ (Seinfeld and Pandis, 2016). During the Chilbolton experiments, wind speeds were between 1 and 5 m/s. There is no other measured information about the weather conditions at the time of the release. Furthermore, the UK Met Office data for the closest meteorological observation stations do not cover the period of interest. The only available information regarding the radiation index is a picture of the release site taken during the experiments. A light grey sky is visible, from which we assumed the radiation index was moderate. Therefore, using table A.3.3, we would expect the Chilbolton experiments to have taken place under ASC B.

To overcome the lack of evidence needed to confidently determine the ASC, we performed an exploratory data analysis to identify the ASC. Instead of selecting a single ASC using the poor meteorological data available, we compared Smith and Briggs-based model predictions of spatial gas concentration measurements to the real data. This was done by computing the average CH_4 measurements observed by each beam for a carefully chosen subset of data points, which we call a *slice*. This was computed using a discretised integration along each beam path. The subset is chosen as to contain observations where the wind direction always intersects some of the beams. This ensures that for some beams, the average CH_4 measurement is calculated using only observations where the plume is crossing their path. Then, for each Smith and Briggs ASC wind sigma parametrisation, we predicted the corresponding averaged CH_4 beam measurements, which is possible because we know the true source location and emission rate. Finally, we plot the CH_4 measurements against the downwind distance for the intersection points between the beams and the averaged wind direction line starting from the source location (as illustrated in Figure A.3.1, A.3.3, A.3.5, A.3.7, A.3.9, and A.3.11). Figure A.3.2, A.3.4, A.3.6, A.3.8, A.3.10, and A.3.12, shows the results for both Source 1 and Source 2, the local ASC could not be determined as the equally most accurate predictions came from the Smith B, Smith C, Briggs A, and Briggs B schemes. This means that we cannot select a unique ASC for our inversion. Predictions made from est. Smith and est. Draxler's maximum a posteriori estimations of source emission rate and dispersion parameters were also added for comparaison. These are represented by the dotted lines in Figure A.3.2, A.3.4, A.3.6, A.3.8, A.3.10, and A.3.12.

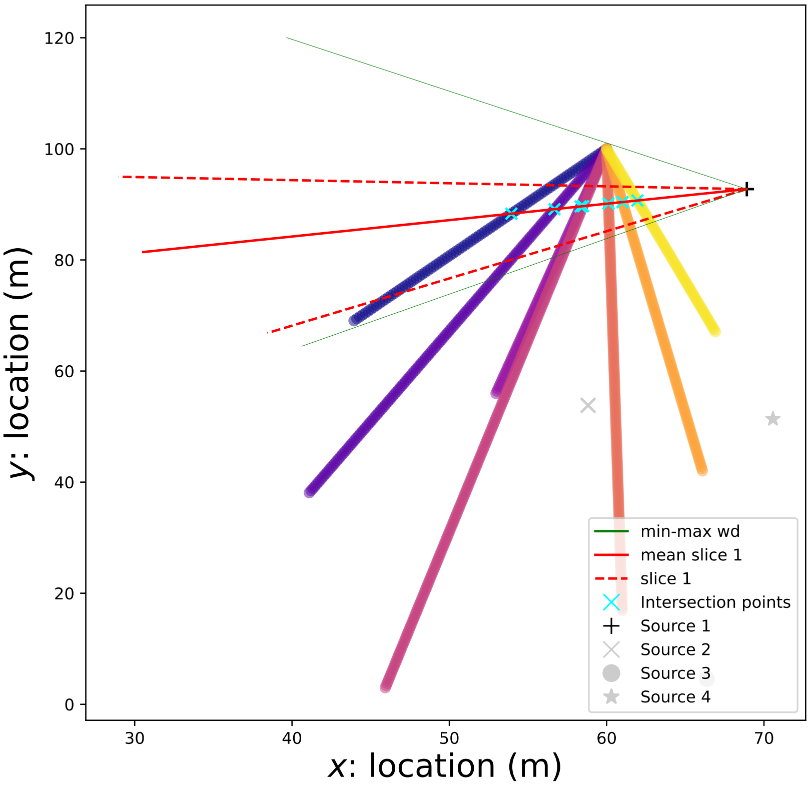


Figure A.3.1: The dashed red lines represent the range of wind directions in slice 1 of Source 1 data and the straight red line is the average wind direction.

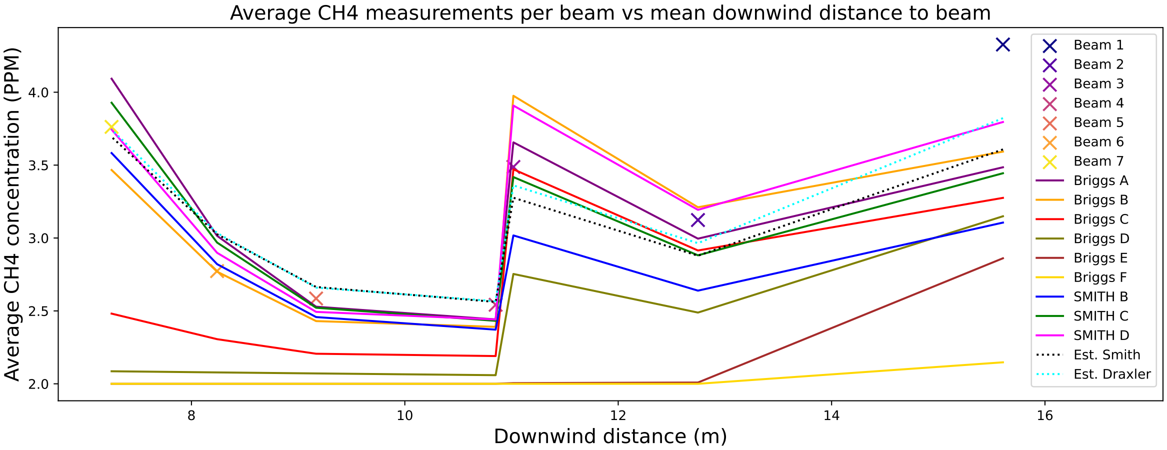


Figure A.3.2: This plot shows the CH_4 measurements as a function of downwind distance and corresponding model predictions for slice 1 of Source 1 data.

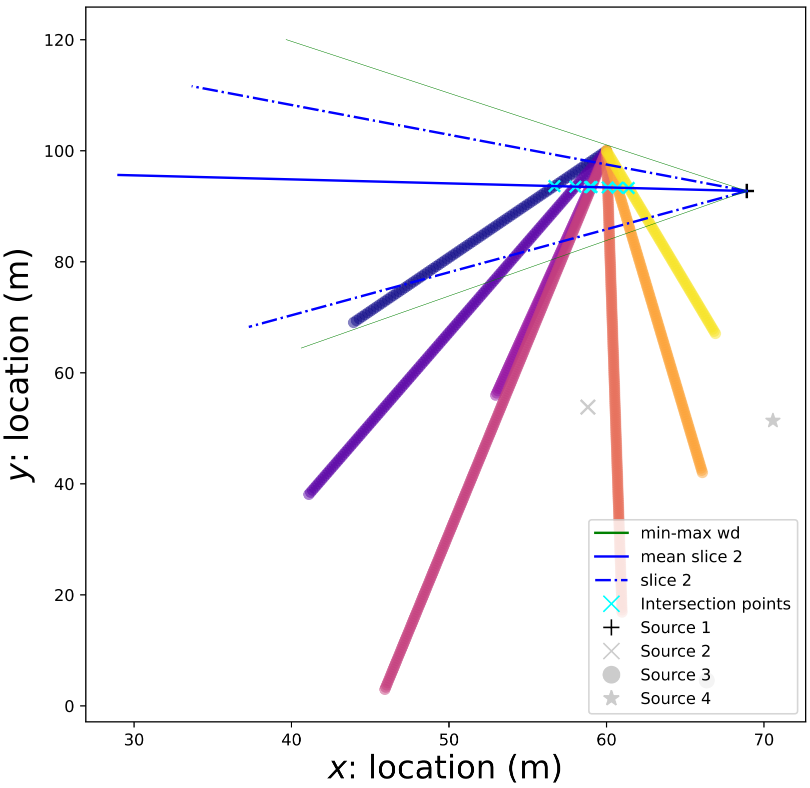


Figure A.3.3: The dashed red lines represent the range of wind directions in slice 2 of Source 1 data and the straight red line is the average wind direction.

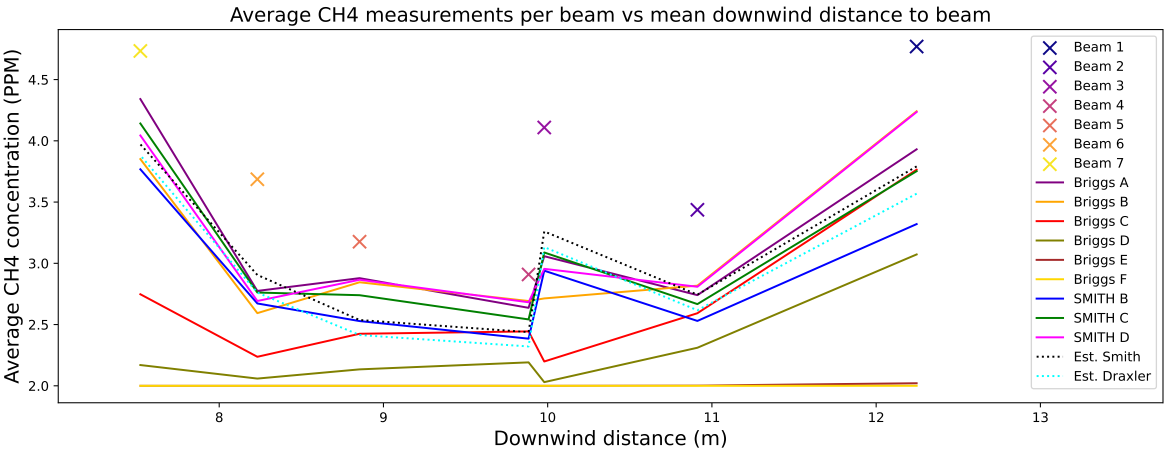


Figure A.3.4: This plot shows the CH_4 measurements as a function of downwind distance and corresponding model predictions for slice 2 of Source 1 data.

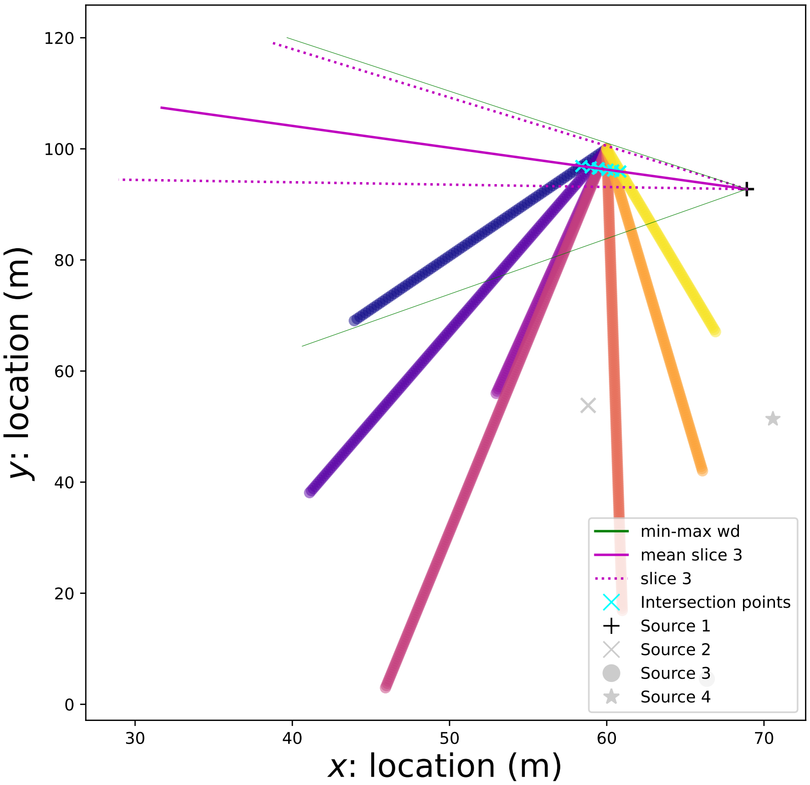


Figure A.3.5: The dashed red lines represent the range of wind directions in slice 3 of Source 1 data and the straight red line is the average wind direction.

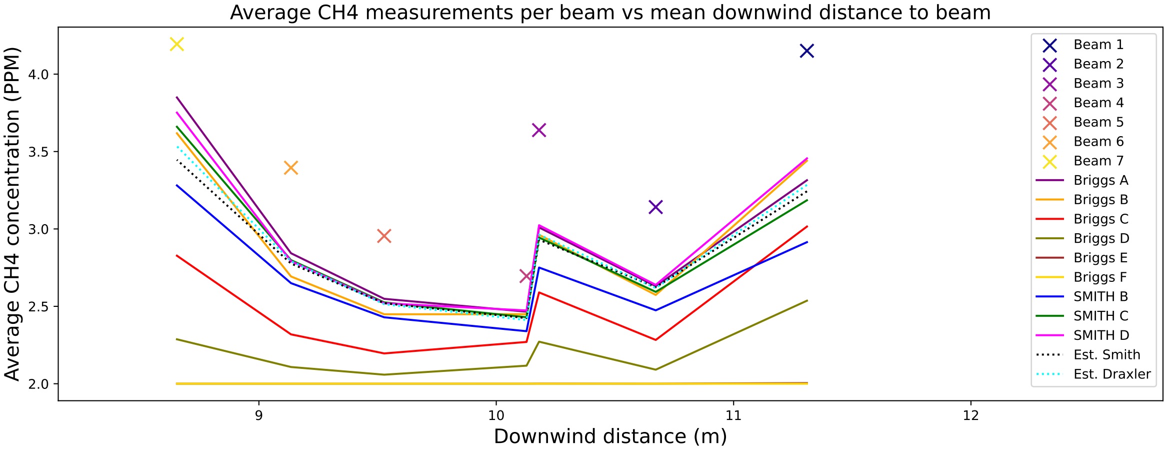


Figure A.3.6: This plot shows the CH_4 measurements as a function of downwind distance and corresponding model predictions for slice 3 of Source 1 data.

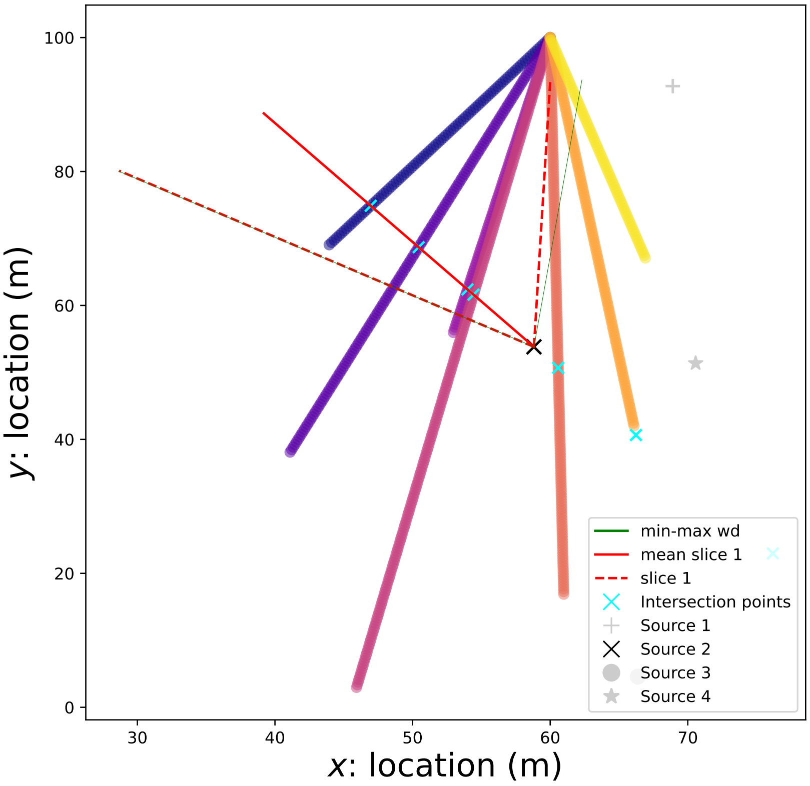


Figure A.3.7: The dashed red lines represent the range of wind directions in slice 1 of Source 2 data and the straight red line is the average wind direction.

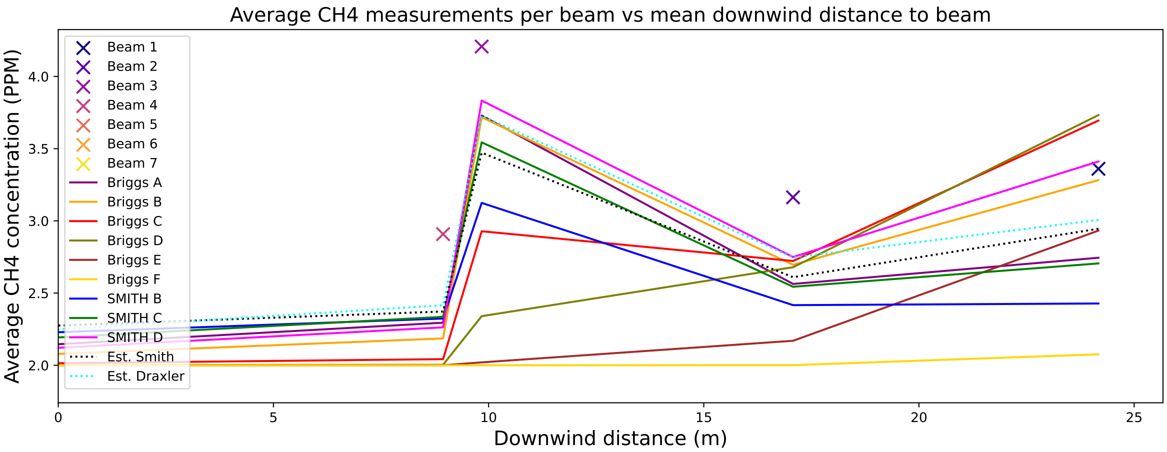


Figure A.3.8: This plot shows the CH_4 measurements as a function of downwind distance and corresponding model predictions for slice 1 of Source 2.

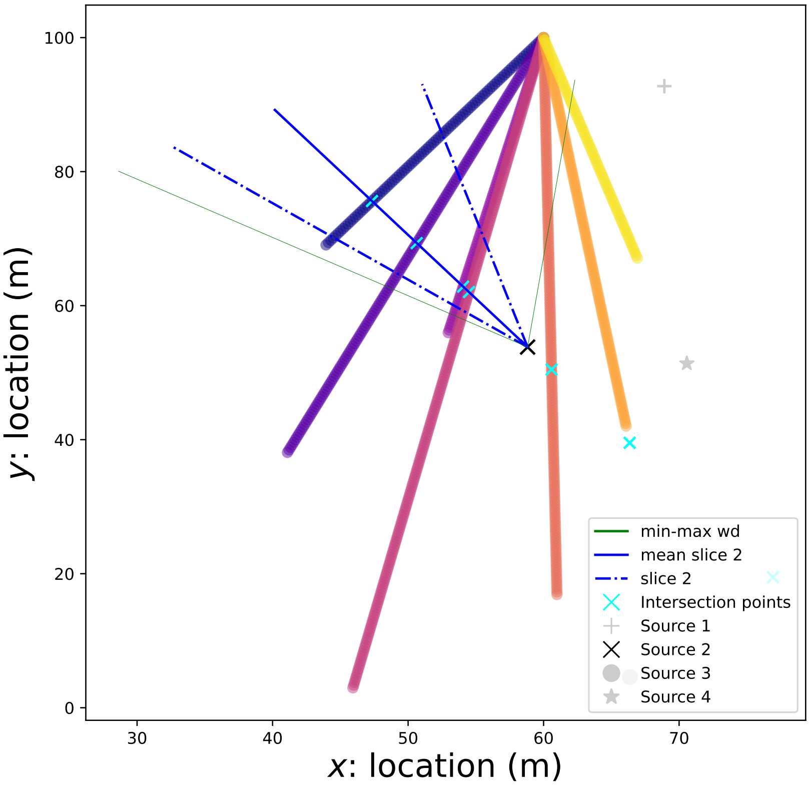


Figure A.3.9: The dashed red lines represent the range of wind directions in slice 2 of Source 2 and the straight red line is the average wind direction.

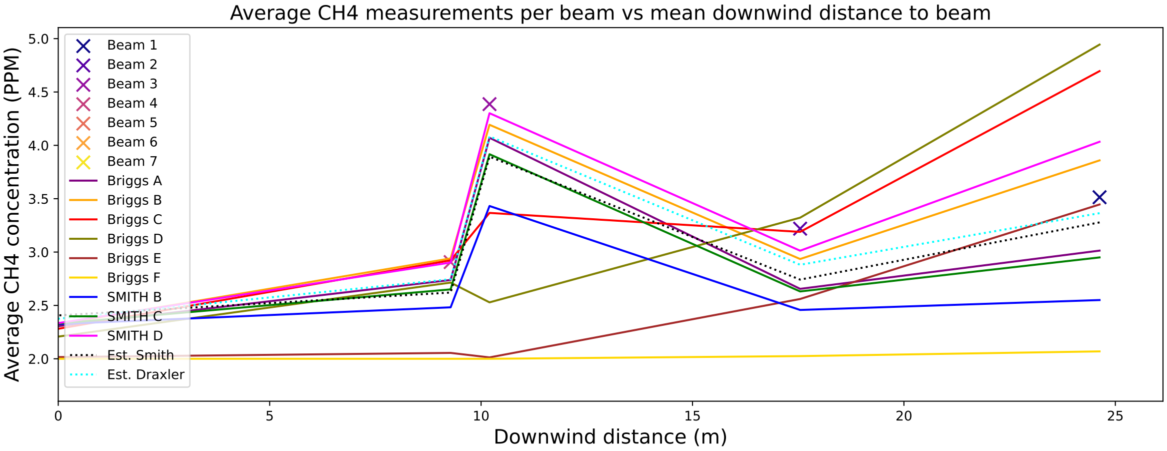


Figure A.3.10: This plot shows the CH_4 measurements as a function of downwind distance and corresponding model predictions slice 2 of Source 2.

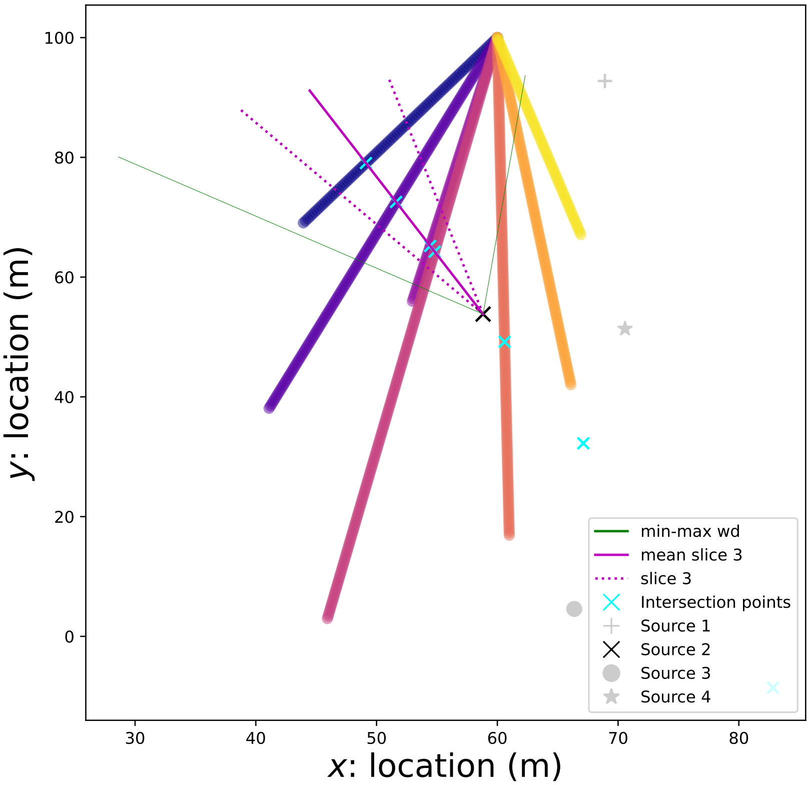


Figure A.3.11: The dashed red lines represent the range of wind directions in slice 3 of Source 2 and the straight red line is the average wind direction.

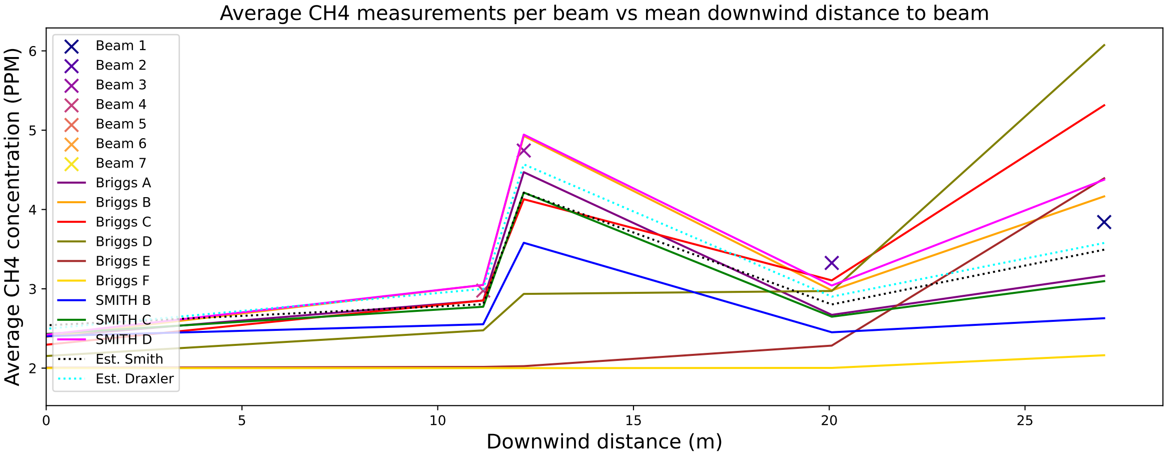


Figure A.3.12: This plot shows the CH_4 measurements as a function of downwind distance and corresponding model predictions slice 3 of Source 2.

A.3.5 Source 1 MCMC chains

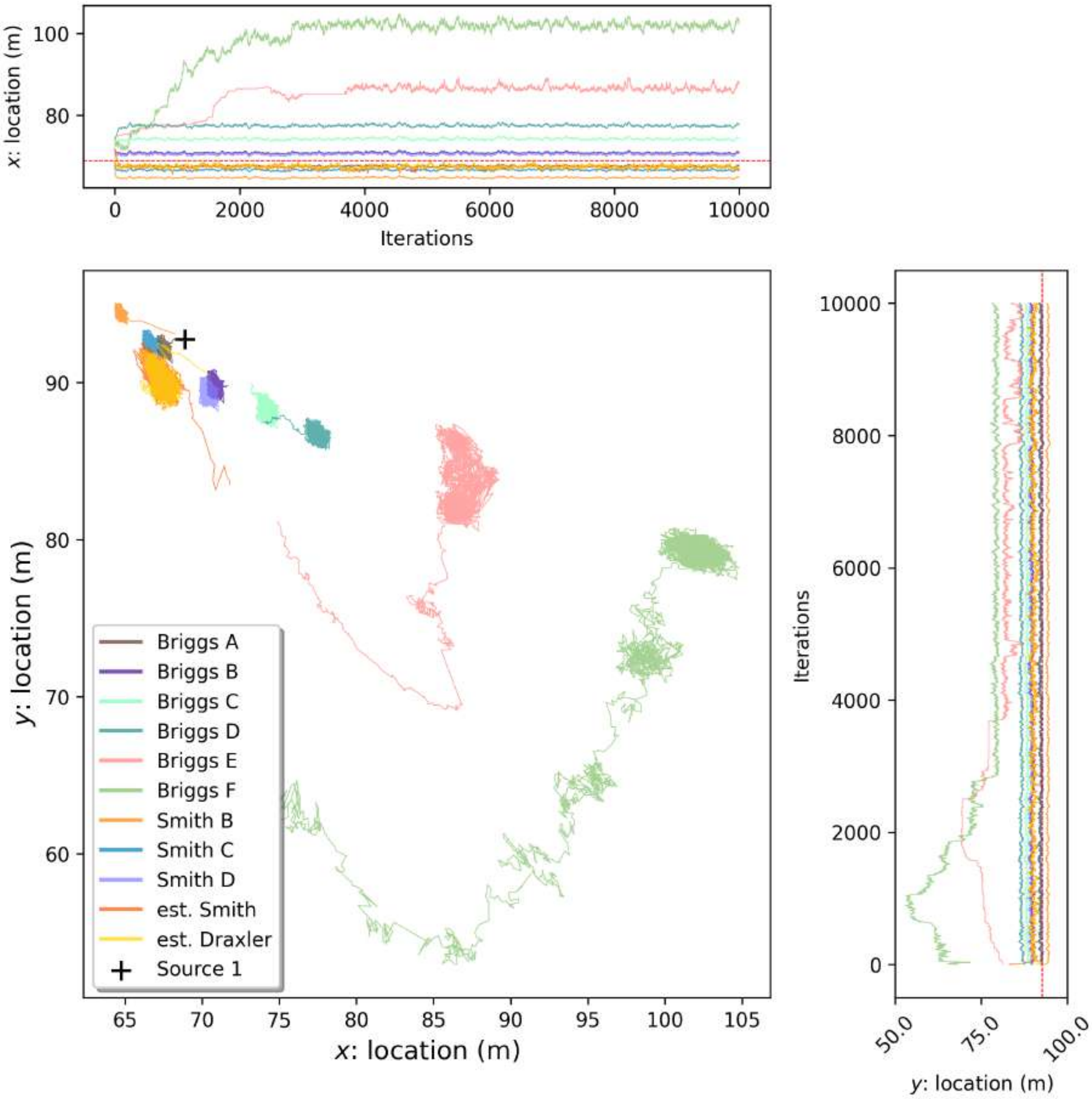


Figure A.3.13: Source 1 location chains.

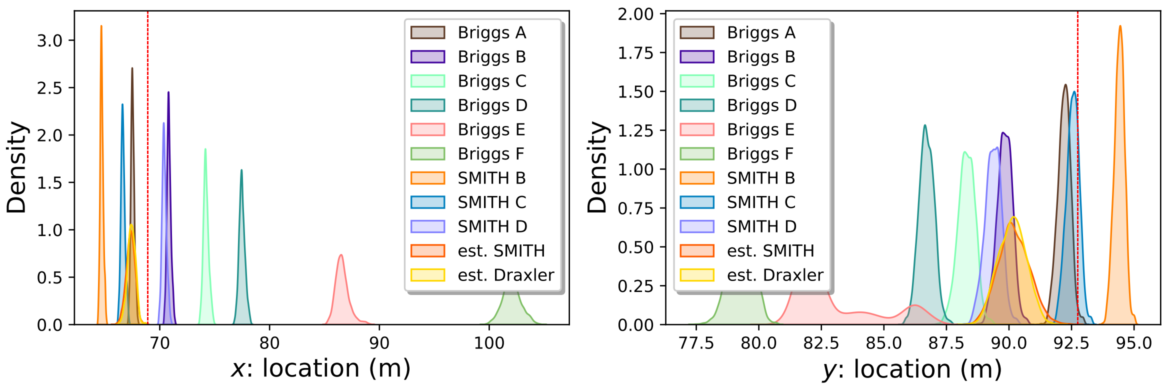


Figure A.3.14: Source 1 location densities.

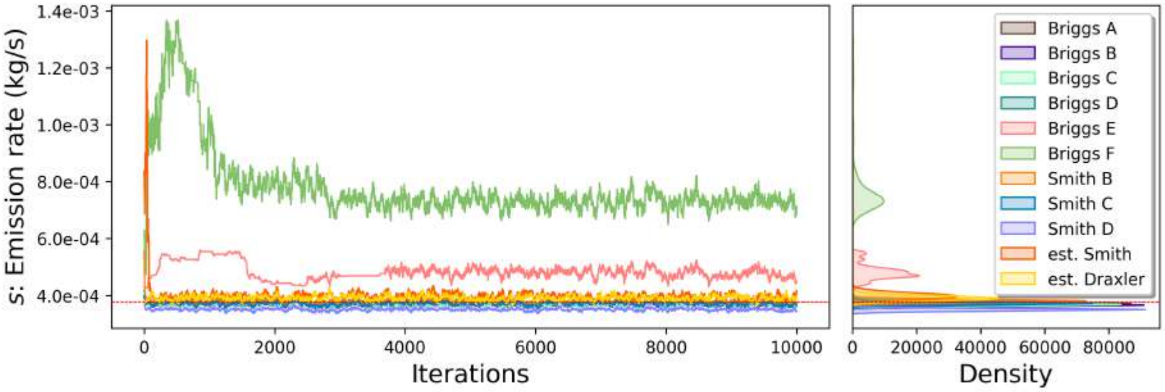


Figure A.3.15: Source 1 emission rate.

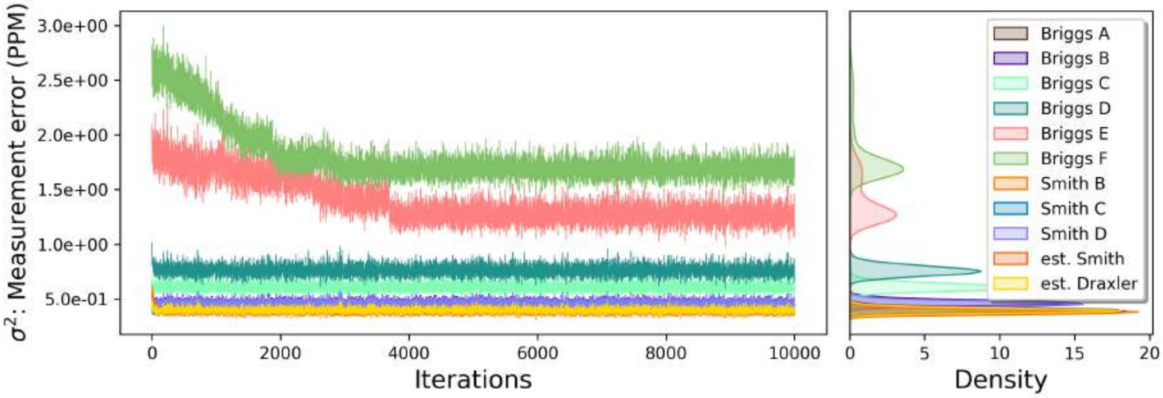


Figure A.3.16: Source 1 measurement error variance.

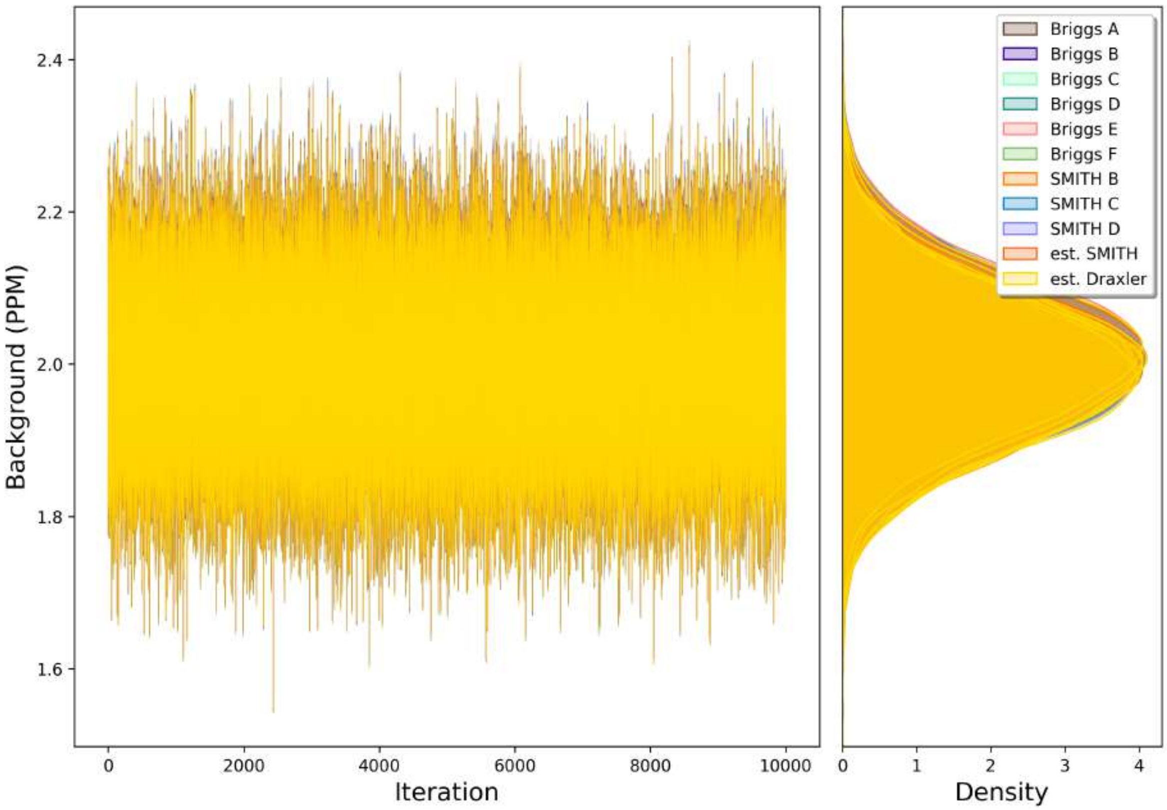


Figure A.3.17: Source 1 background.

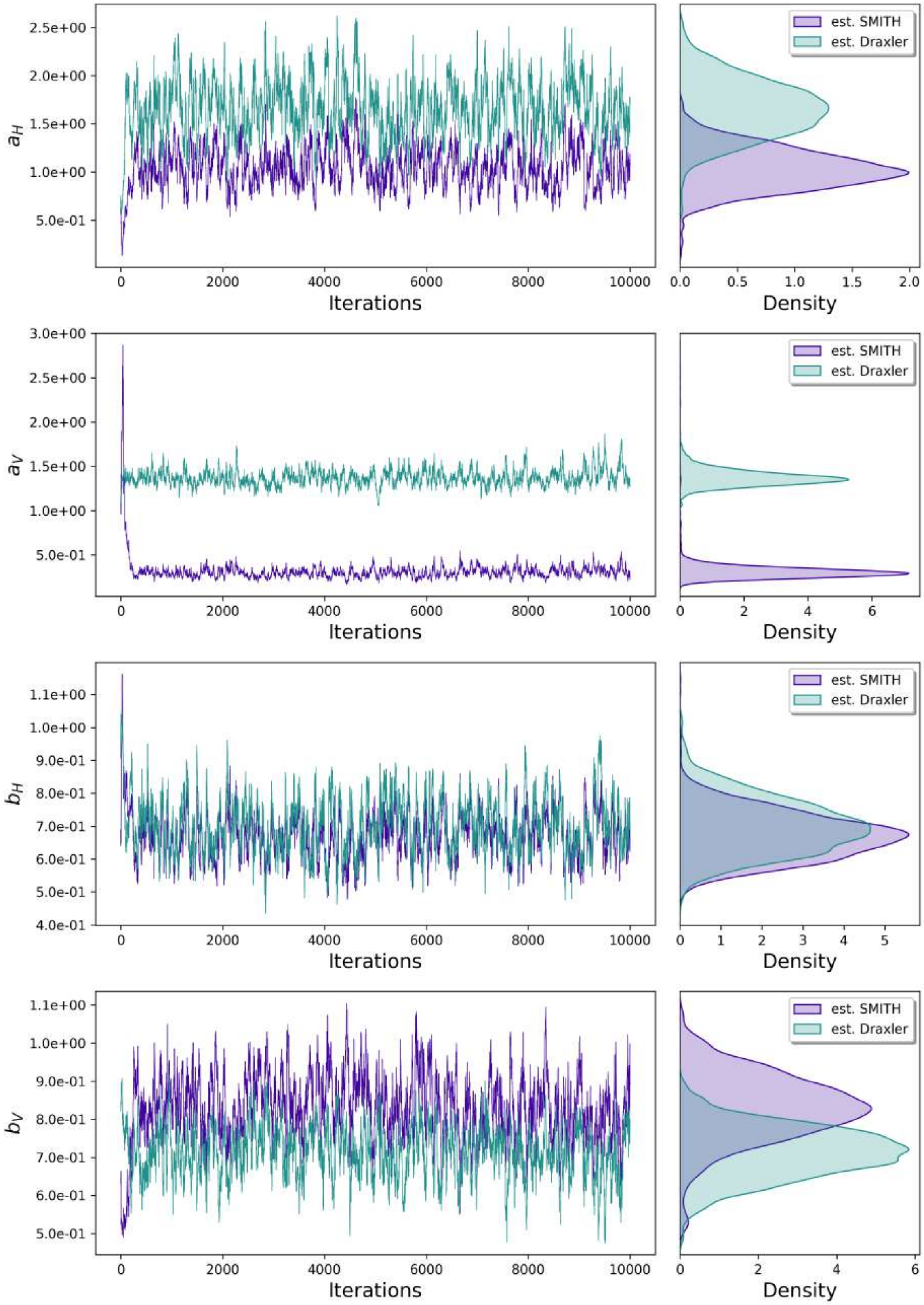


Figure A.3.18: Source 1 wind sigmas.

A.3.6 Source 2 MCMC chains

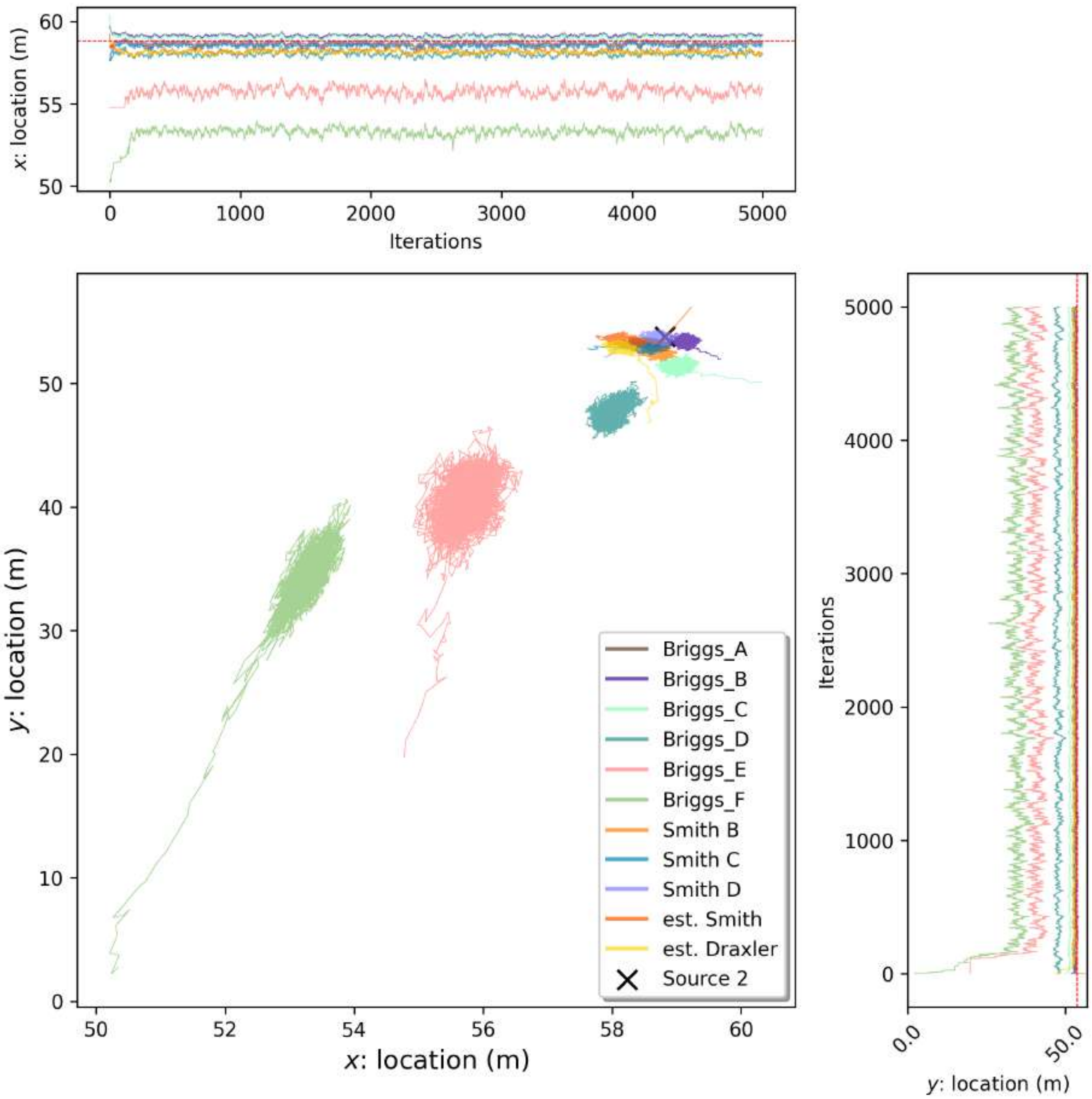


Figure A.3.19: Source 2 location chains.

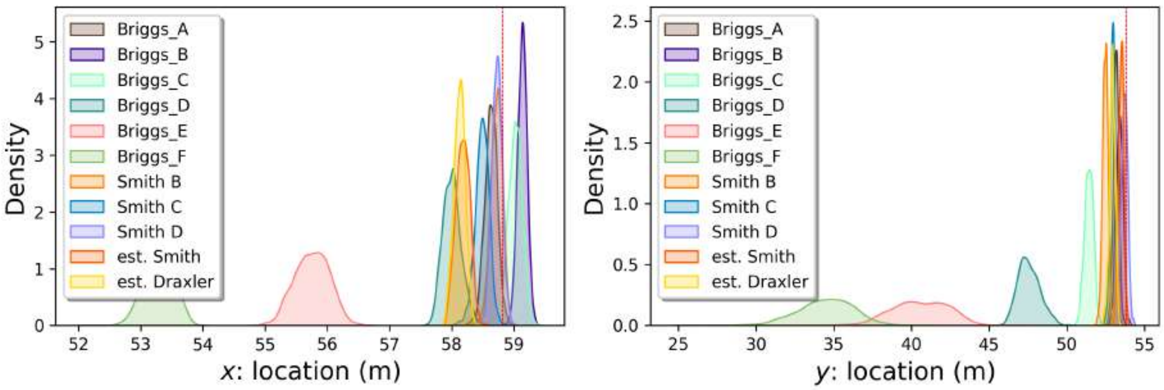


Figure A.3.20: Source 2 location densities.

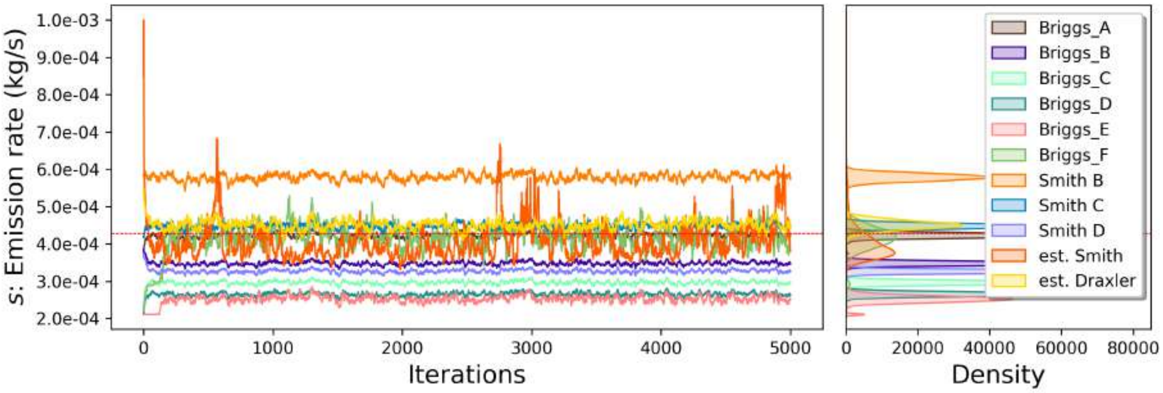


Figure A.3.21: Source 2 emission rate.

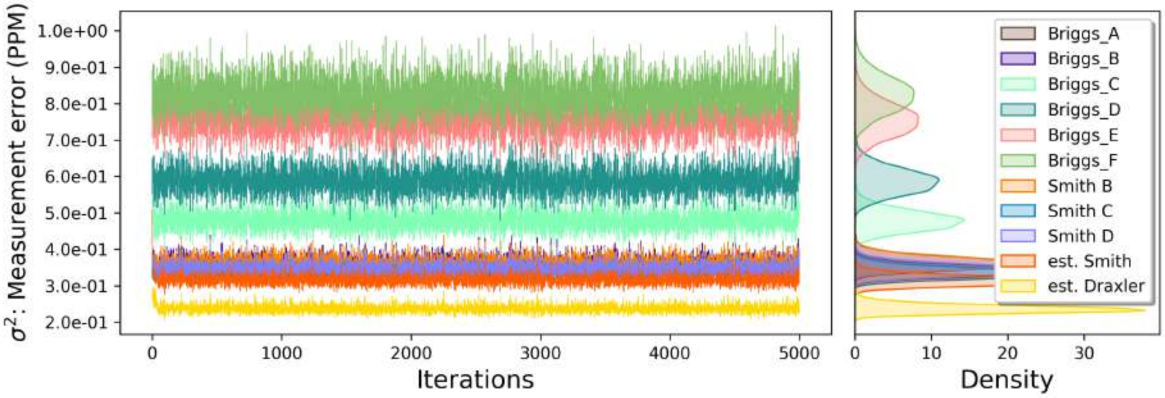


Figure A.3.22: Source 2 measurement error variance.

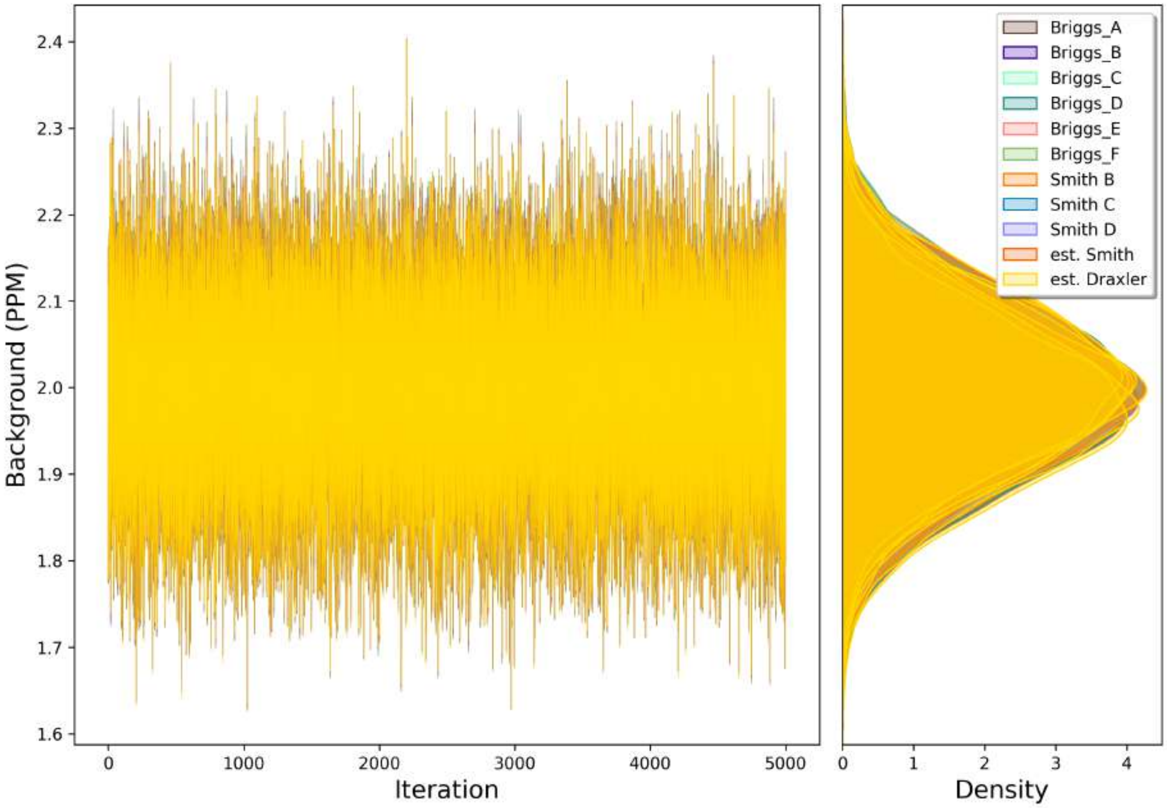


Figure A.3.23: Source 2 background.

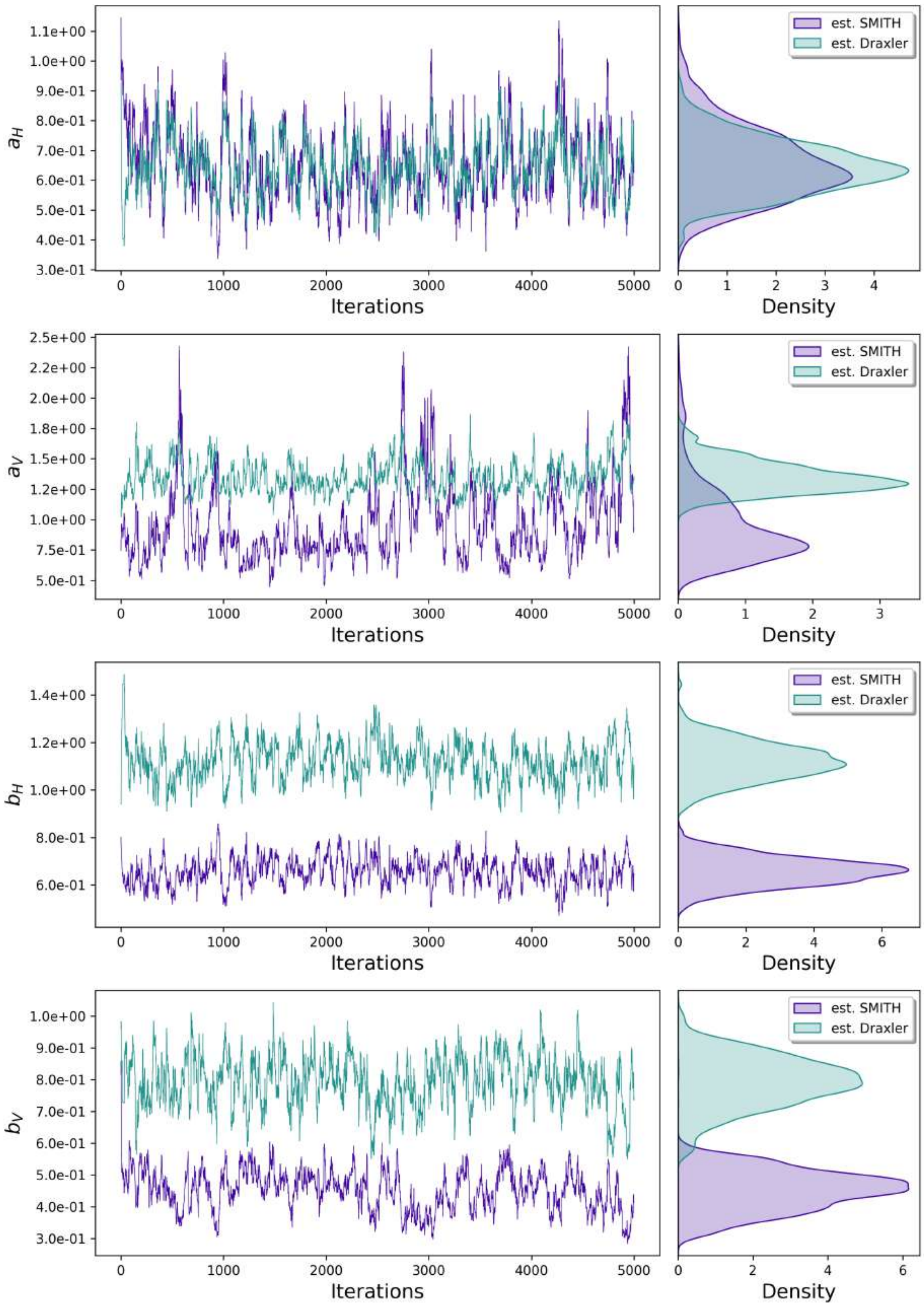


Figure A.3.24: Source 2 wind sigmas.

Appendix B

Supplementary Material for Chapter 5

This appendix contains all the supplementary materials supporting Chapter 5.

B.1 Supplementary material: numerical solver

Context: In the Chapter 5, we conduct surrogate-based inversion modelling of gas emissions. This supplementary material explains how atmospheric gas transport simulations were generated using `Phiflow` (Holl and Thuerey, 2024) to train our surrogate model.

Content: Section B.1.1 provides an introduction to the `Phiflow` Python package and its application context. Section B.1.2 outlines the meshes and boundary conditions employed in the numerical solver. Section B.1.3 examines the numerical stability of `Phiflow`.

B.1.1 About `Phiflow`

Physically realistic atmospheric gas transports are modelled using partial differential equations and used to train our deep neural network surrogate model. We use `Phiflow` (Holl and Thuerey, 2024), an open-source computational fluid dynamics simulation software available on PyPI, to simulate atmospheric gas transport. It integrates seamlessly with JAX, leveraging its just-in-time compilation for performance optimisation. `Phiflow` includes two advection solvers - `semi_lagrangian` and `mac_cormack` - as well as an incompressible solver, `make_incompressible`. In this work, diffusion is neglected since gas transport is predominantly influenced by wind conditions.

The `semi_lagrangian(v, v)` function is used for first-order approximation of flow field v advection with itself, based on the method described in Stam (1999). The `make_incompressible(v)` function computes pressure corrections to ensure incompressibility, adhering to the boundary conditions of the domain MacCormack (2002). Finally, `mac_cormack(c, v)` applies the MacCormack scheme to approximate the advected gas field c, following the approaches outlined in Chorin (1968); Temam (1968).

B.1.2 Numerical solver meshes and boundary conditions

The numerical solvers use two different mesh structures, both with Dirichlet boundary conditions, one for each of the flow and gas fields. Boundary extrapolation is applied to the gas mesh, a technique that pads the field and enhances computational efficiency compared to conventional boundary conditions. Temporally varying wind speeds are set as the boundary conditions when solving for the flow field. In practical scenarios, the complete wind field is typically unobservable; instead, a single high-frequency anemometer placed within the domain is often used to measure wind speeds. The wind field is then modelled as spatially uniform but temporally varying. For our simulations, two-dimensional wind speed data are modelled using Ornstein-Uhlenbeck processes.

This mean-reverting stochastic process is characterised by a mean value, a reversion rate, and a standard deviation. For our parameterisation, the mean wind speed is set to 3.3 m/s, reflecting realistic conditions observed in the Chilbolton dataset. The reversion rate is set to 0.6, representing a low lag-1 autocorrelation, and the standard deviation is 0.8, also derived from the Chilbolton data.

B.1.3 Testing `Phiflow`

The temporal discretisation step size Δt significantly affects the accuracy of the numerical solver's approximation to the advection-diffusion and Navier-Stokes equations in `Phiflow` simulations. While reducing the step size improves accuracy, it increases computational cost. To balance accuracy and efficiency, we select a step size that ensures numerical stability without being excessively small using the Courant-Friedrichs-Lewy (CFL) condition (Courant et al., 1928). For example, in our simulation case study, we use wind fields with a mean speed of 3.3 m/s, and the lowest mesh resolution to spatial coverage ratio is given by a high-resolution mesh (200 x 200) on a 100m $\times$ 100m domain, the CFL condition sets the following upper limit for Δt:

$$\Delta t \leq \frac{100/200}{3.3}.$$

Based on this, we choose $\Delta t = 0.1$ s. To demonstrate the impact of step size, we compared two gas transport simulations where only the step size varies: one with $\Delta t = 1.0$ s and the other with $\Delta t = 0.1$ s. To account for the numerical solver temporal step size, the wind speeds were linearly interpolated; as illustrated in Figure B.1.1.

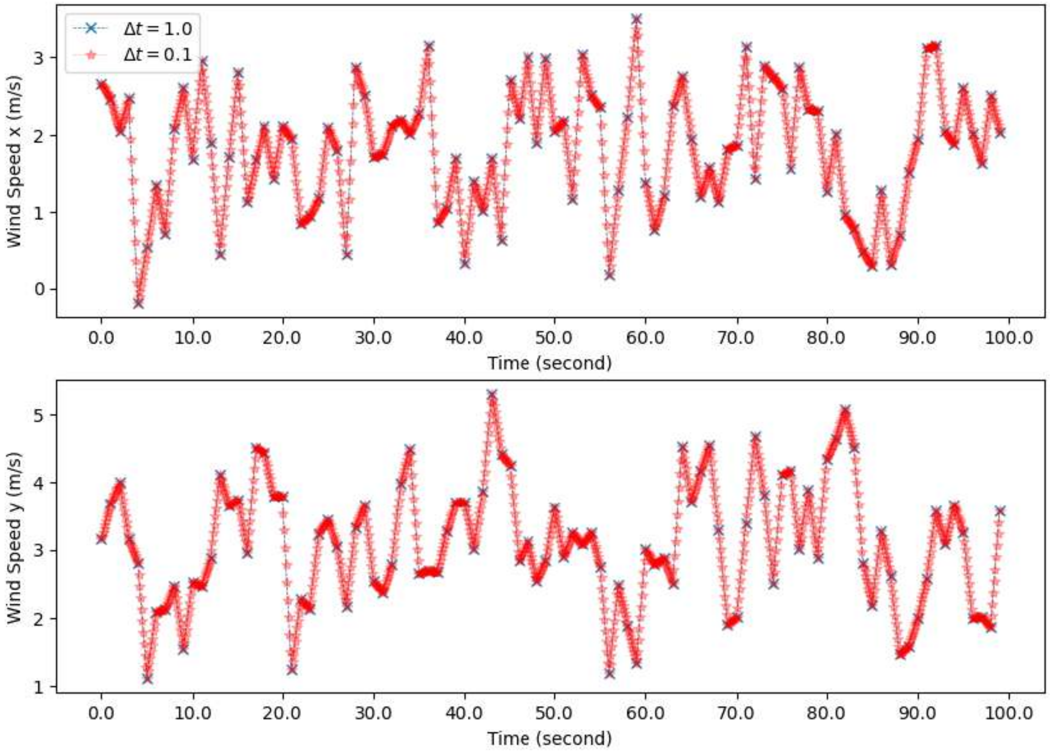


Figure B.1.1: Example of an OU process wind fields for different temporal step sizes. The line with blue marker “x” corresponds to the wind field with $\Delta t = 1.0$ s, and the line with red marker “*” corresponds to its linear interpolation for a step size $\Delta t = 0.1$ s.

B.2 Supplementary material: temporal window selection and clearing periods for online inversion

Context: In Chapter 5, we develop a real-time inversion framework for estimating gas source parameters using deep neural network surrogates trained on CFD simulations. This supplementary material explains how the temporal duration of these simulations is selected to ensure independence from unknown initial conditions, which is critical for accurate and timely parameter estimation in online settings.

Content: Section B.2.1 introduces the need for defining a clearing period based on atmospheric transport dynamics and site geometry. Section B.2.2 discusses how the clearing period was determined for the Chilbolton dataset based on wind speed and domain size. Section B.2.3 briefly addresses challenges and open questions when extending this analysis to obstructed environments.

B.2.1 Clearing period

To enable online parameter estimation, it is desirable to begin inference using the shortest possible window of recent data. The appropriate temporal window length depends both on the practitioner's desired granularity for tracking parameter changes (e.g., per minute, hourly, or daily) and on the interaction between site layout and atmospheric conditions.

Practitioners must first determine the temporal resolution at which they wish to monitor emission parameters. For example, estimating minute-by-minute changes requires that data be processed at most every minute. Simultaneously, the underlying physical environment constrains how soon accurate predictions can be made from simulations: specifically, the numerical solver must simulate gas transport long enough for predictions to become independent of unknown initial gas and flow field conditions. This is critical, as true initial conditions are not directly measurable in practice. We refer to the duration required to reach this independence as the "clearing period".

The clearing period varies with wind speed and site complexity. For instance, faster wind speeds reduce the time required for the gas to cross the longest path across the site, while obstacles may trap gas and necessitate longer simulations.

B.2.2 Chilbolton experiment

In the Chilbolton dataset, the site spans 40m × 100m, and the average wind speed is 3.3m/s with no significant obstacles. The maximum gas travel distance is roughly 108 meters, yielding a clearing period of approximately 33 seconds. To account for wind variability, we conservatively extend this to 40 seconds. To reduce model misspecification and stabilise the inversion likelihood surface, we use time-averaged predictions – an assumption required for the Gaussian plume model (Stockie, 2011). Accordingly, for each minute of real sensor measurement, we simulate 1 minute of gas transport using the numerical solver, discard the first 40 seconds (clearing period), and average the predicted sensor concentrations over the final 20 seconds.

B.2.3 Obstructed environments

While identifying an appropriate clearing period is straightforward in open, unobstructed sites, it becomes substantially more complex in the presence of obstacles that may disrupt flow patterns or delay mixing. Developing reliable methods to estimate clearing periods in such obstructed environments is left as future research.

B.3 Supplementary material for Chapter 5 Section 4: Chilbolton

Context: In Chapter 5, we conduct surrogate-based inversion modelling of gas emissions. This supplementary material provides additional details regarding the Chilbolton dataset and its sources' parameter estimation.

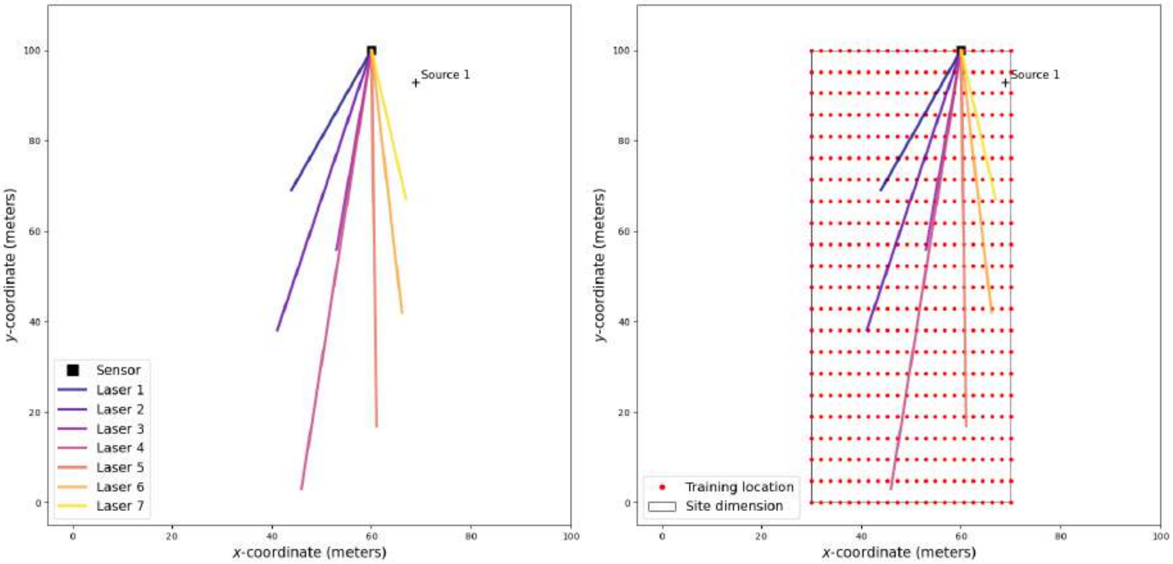
Content: Section B.3.1 provides information regarding the data processing performed before the parameter estimation. Section B.3.2 gives additional details about the source inversion methodology covered in the paper.

B.3.1 Data

The Chilbolton dataset contains controlled methane releases made on the flat terrains of the Chilbolton Observatory, Hampshire, UK. As these are controlled releases, the true source locations and emission rates are known. The site layout is presented in Figure 4.1.1. The raw data required preprocessing before implementing our source inversion methodology. The processing code can be found in the Jupiter notebooks in the folder: `Code / Chilbolton / Data / Processing` .

B.3.2 Source inversion

MLP Training. We trained the MLPs on the Chilbolton site using simulated gas transport from different source locations; this is shown in Figure 4.1.1 alongside a visualisation of the site layout. We used MLPs with four hidden layers, and each hidden layer has 100 neurons. We employ SeLU activations (Klambauer et al., 2017) at the hidden layers, and a linear activation at the output layer (since we are performing a regression to predict a continuous concentration value). We initialise the network



(a) Chilbolton setup.

(b) MLP training setup.

Figure B.3.1: These visualise the Chilbolton setup: sensor, Source 1 and lasers' locations, and the MLP training setup: training data source location, and site dimension.

weights using standard Xavier initialisation (Glorot and Bengio, 2010) and train them to minimise the error between the MLP's predictions and the true concentrations from the CFD simulations using stochastic gradient descent. Corresponding code can be found in the folder: `Code / Chilbolton / MLP_training`.

Measurement Predictions. True sensor measurements were predicted using: a Gaussian plume model, a two-dimensional numerical solver, and its surrogate model. Predictions were made using 10 minutes (from 2017-05-09 11:27:00 to 2017-05-09 11:37:00) and 15 minutes (from 2017-05-10 10:30:00 to 2017-05-10 10:45:00) of Source 1 and Source 2 measurements, respectively. These time intervals were chosen as they correspond to times when the wind blew the gas towards the sensors; this ensures appropriate data to assess our models' performance. Model comparison can be found in the following Jupyter notebooks: `Code / Chilbolton / Predictions / Source_1_predictions.ipynb`, and the predictions were made in the following Python scripts: `Code / Chilbolton / MLP_training / Source_1 /`

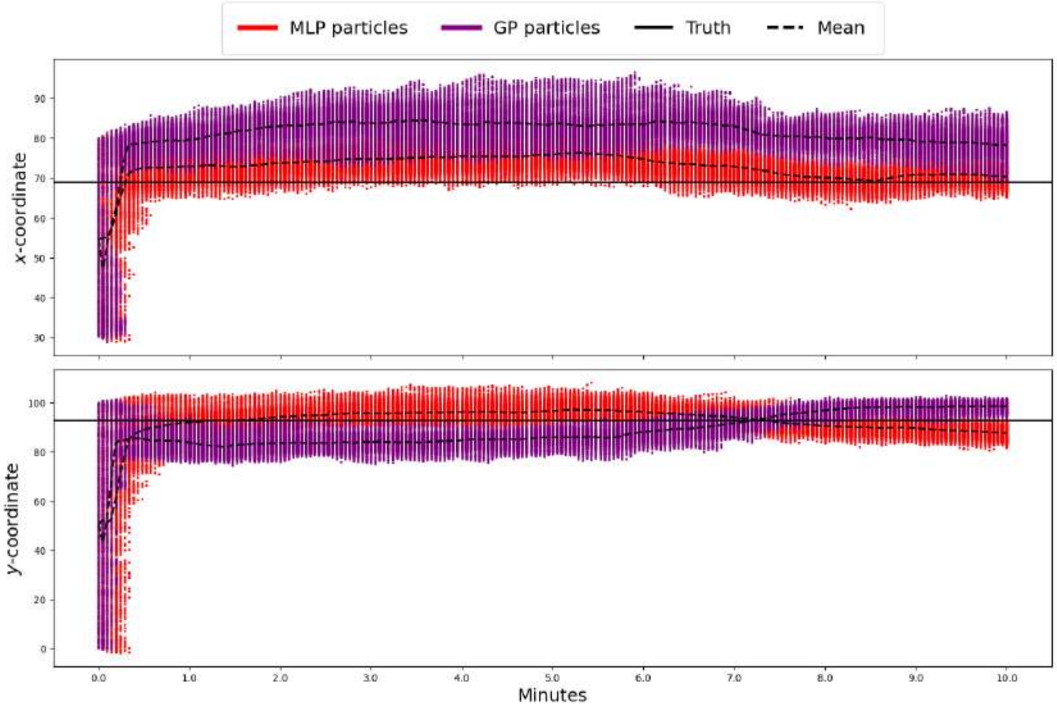


Figure B.3.2: Gaussian plume model and surrogate model-based SIR particles as a function of iterations and minutes. The surrogate-based particle values for the source location are closer to the true source location, providing more accurate parameter estimation.

`Source_1_release_2_mlp_training.py`.

Parameter Estimation. Source 1's location and emission rate were estimated using both the Gaussian plume-based and surrogate-based SIR particle filters and 4-minute sliding windows of data. Figure B.3.2 displays the source location estimation as a function of iterations and time, and Figure B.3.3 shows the estimation relative to the Chilbolton setup. The Python scripts for the particle filter inversion and Jupyter notebooks containing the visualisation can be found in: `Code / Chilbolton / Particle_filter / Source_1` .

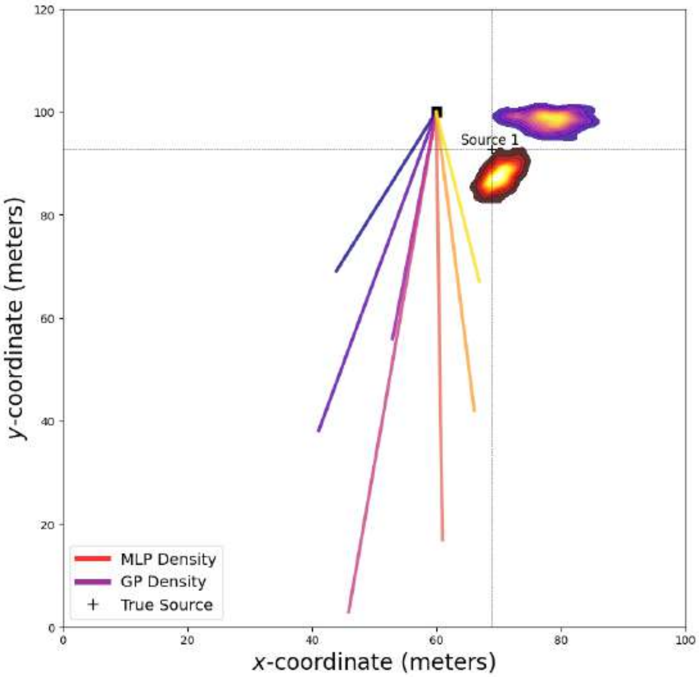


Figure B.3.3: Gaussian plume model and surrogate model-based SIR particles' posterior density of Source 1's location using 700 iterations and 1,000 particles. The posterior density closer to the true source location represents the surrogate-based inversion, providing more accurate parameter estimation.

B.4 Supplementary material for Chapter 5 Section 5: case study

Context: In Chapter 5, we conduct surrogate-based inversion modelling of gas emissions. This supplementary material provides additional details regarding the case study dataset and its sources' parameter estimation.

Content: Section B.4.1 provides information regarding the simulated emission sources. Section B.4.2 gives additional details about the source inversion methodology covered in the paper.

B.4.1 Data

To our knowledge, there are no publicly available datasets containing atmospheric gas concentration measurements from known emission sources in obstructed, unsteady-state flow fields. To address this gap, we simulated three 10-minute emission events within a 100m x 100m domain containing a single structural obstacle, as shown in Figure B.4.1. Source releases were simulated one at a time.

The wind boundary conditions used in the numerical solver were spatially uniform but temporally varying, modelled as an OU process. The process parameters – mean wind speed, standard deviation, and mean-reversion rate – were derived from Chilbolton anemometer data to ensure realism. Figure B.4.2 illustrates the flow and gas concentration fields at a representative time point, demonstrating the solver's ability to capture obstacle-induced turbulence. To capture obstruction-induced turbulence, we have embedded a second high-resolution mesh within the flow field mesh and around the obstacle (see Figure B.4.2).

Sensor measurements were generated by sampling the simulated gas concentration at sensor locations indicated in Figure B.4.1. These values were time-averaged following

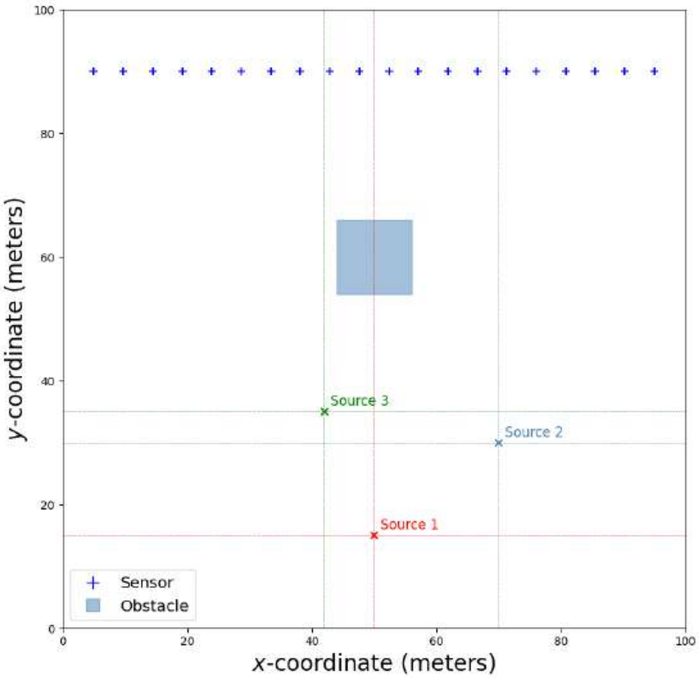
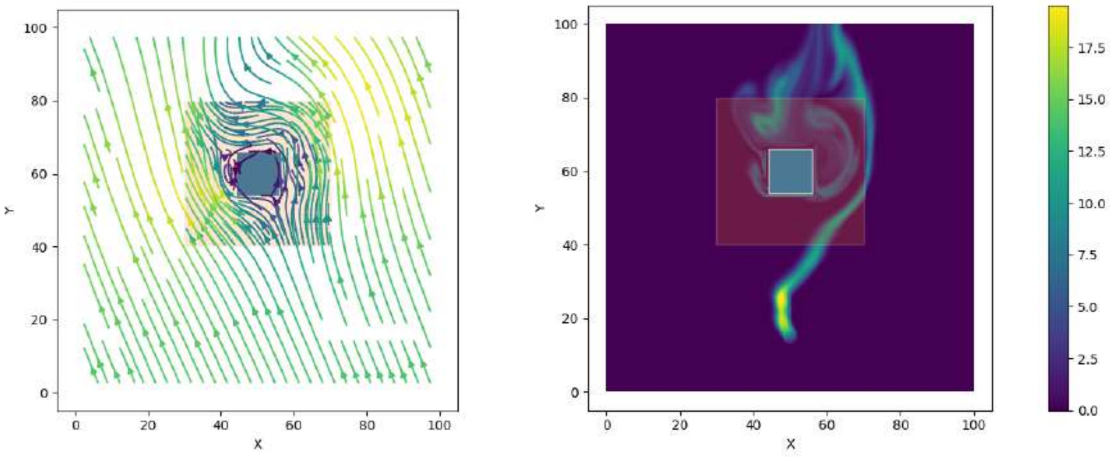


Figure B.4.1: Case study domain containing one obstacle, 20 sensors, and three sources. Source releases were simulated one at a time.

the procedure described in Appendix B.2. The resulting time-averaged measurements are shown in Figures B.4.3 and B.4.4.

B.4.2 Source inversion

MLP Training. We trained the MLPs on the case study site using simulated gas transport from different source locations; this is shown in Figure B.4.5. We used MLPs with five hidden layers, and each hidden layer has 500 neurons. We employ SeLU activations and a linear activation at the output layer. We initialise the network weights using standard Xavier initialisation and train them to minimise the error between the MLP's predictions and the true concentrations from the CFD simulations using stochastic gradient descent. Corresponding code can be found in the folder: `Code / Case_study / MLP_training` .



(a) Flow field.

(b) Gas field.

Figure B.4.2: Flow and gas fields at a fixed point in time during Source 1 simulation. The smallest blue square at the centre of the site represents the obstacle, the square around it represents the high-resolution (200 x 200) flow mesh embedded within the site's flow mesh.

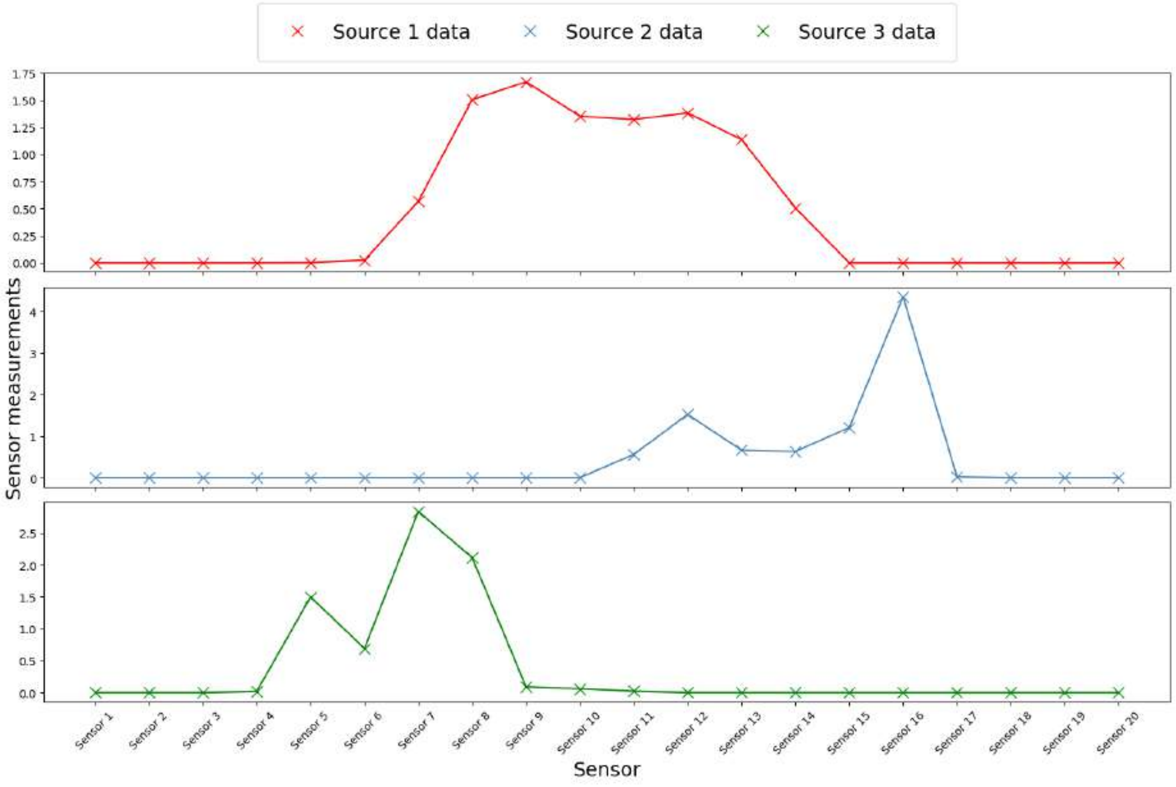


Figure B.4.3: Sensor measurements for all three case study simulated sources. The plot shows 20-second averaged sensor measurements (measurements after the clearing period) from the first minute of each source simulation. The sensors correspond to those in Figure B.4.1.

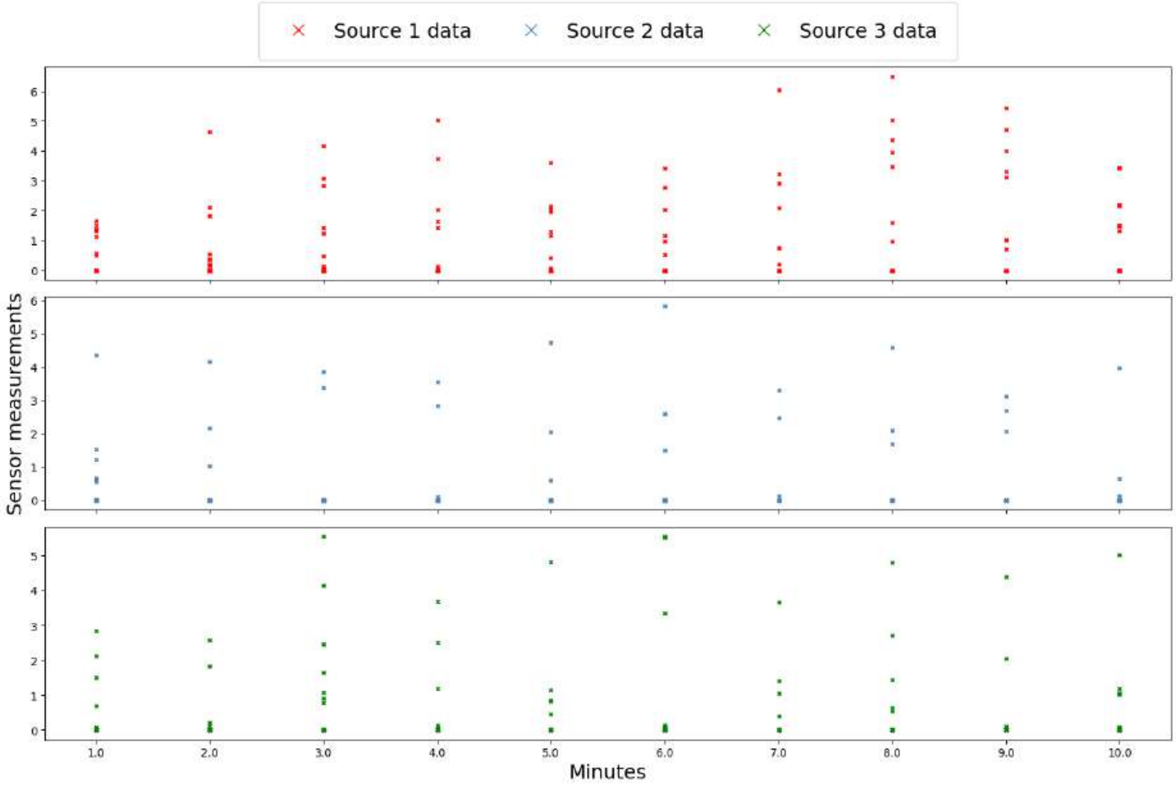


Figure B.4.4: Sensor measurements for all three case study simulated sources. The plot shows 20-second averaged sensor measurements (measurements after the clearing period) from the 10 minutes of each source simulation. Each minute shows the corresponding 20 sensor measurements. The sensors correspond to those in Figure B.4.1.

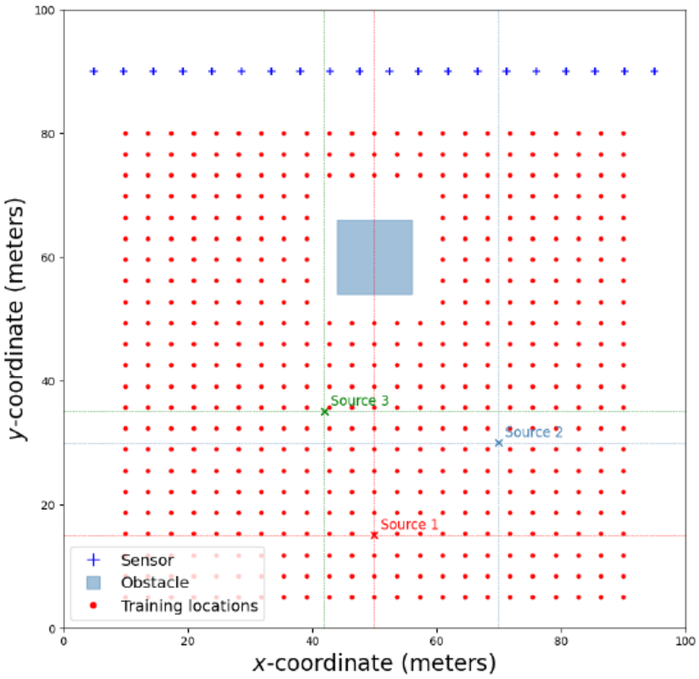


Figure B.4.5: Case study site containing one obstacle, 20 sensors, three sources, and the MLPs' training source locations.

Parameter Estimation. Sources' location and emission rate were estimated using the MLP-based SIR particle filter and 3-minute sliding windows of data. The source location and emission rate temporal estimations are displayed in Figure B.4.6 and Figure B.4.7, respectively. The Python scripts for the particle filter inversion and Jupyter notebooks containing the visualisation can be found in: `Code / Case_study / Particle_filter`.

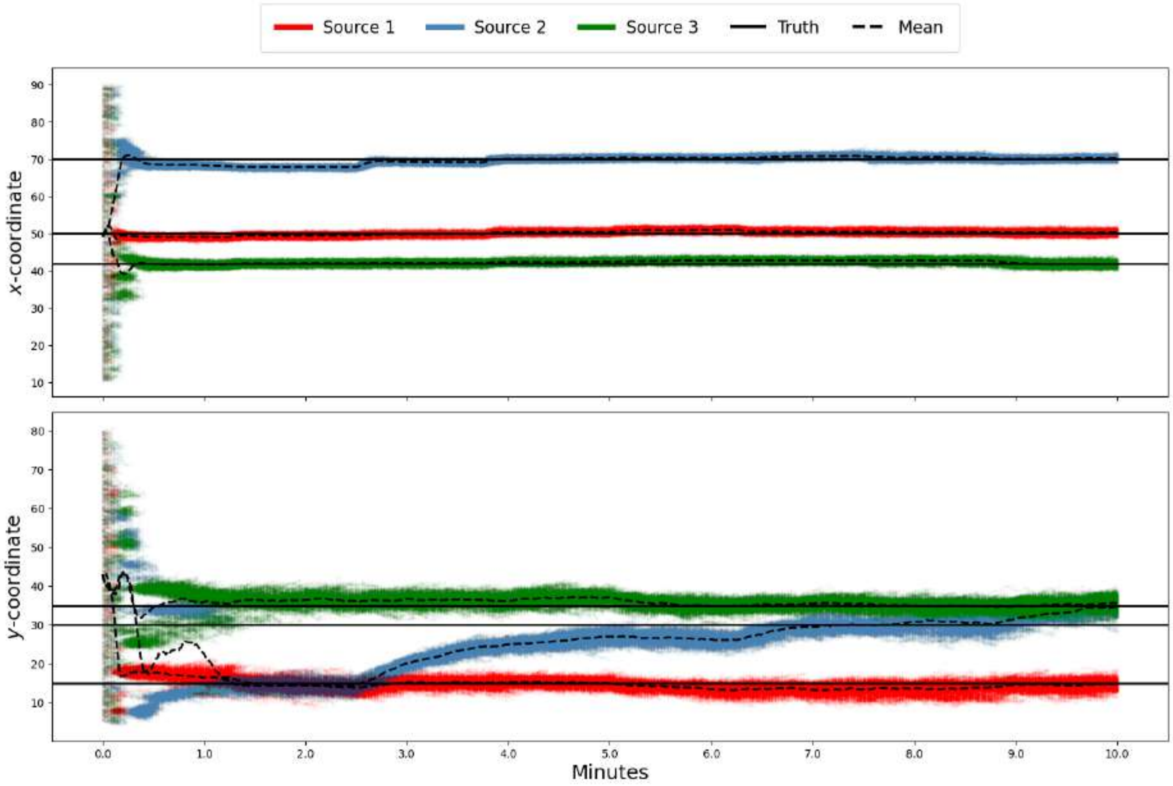


Figure B.4.6: Case study results for source location estimation. The surrogate-based SIR successfully tracks fixed source locations in obstructed flow fields. The delayed adjustment for Source 2 is likely due to the strong fluctuations in its emission rate.

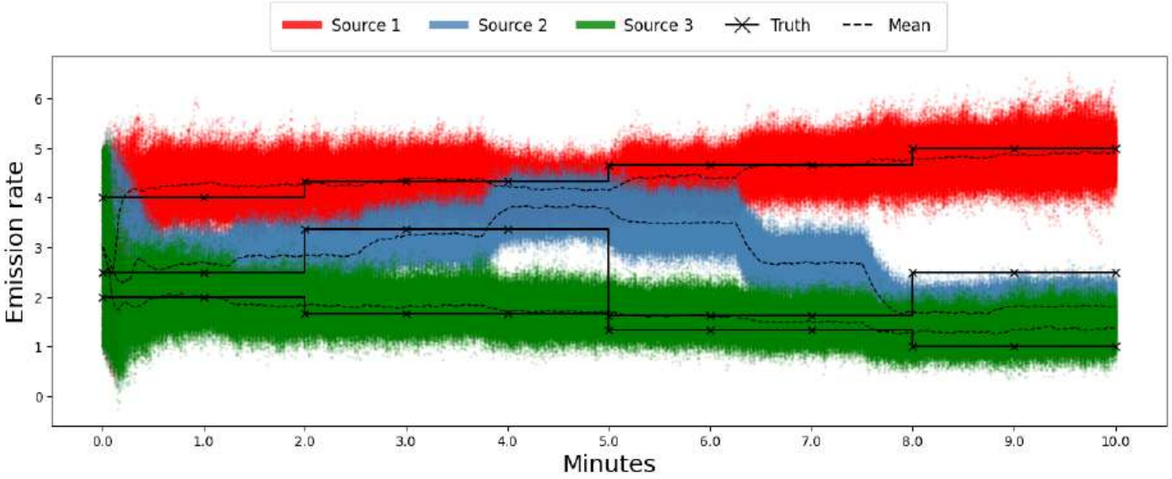


Figure B.4.7: Case study results for temporally varying emission rate estimation. The surrogate-based SIR successfully tracks increasing, decreasing and fluctuating emission rates. The delayed adjustment after Source 2's emission rate shape drop at minute 5 highlights a limitation – such scenarios may require a more responsive two-layered SIR filter.

B.5 Supplementary material: addressing model misspecification using vertical Gaussian plume correction

Context: In Chapter 5, the surrogate models used in our inversion framework are trained on two-dimensional gas transport simulations, which assume that sources, sensors, and obstacles lie on the same horizontal plane. While this approximation is reasonable for flat environments like the Chilbolton site, it introduces model misspecification in scenarios with significant vertical offsets between sources and sensors. This section outlines a computationally efficient method to address this issue using a vertical Gaussian plume correction.

Content: Section B.5.1 introduces the motivation for applying a vertical Gaussian plume correction to mitigate model misspecification in two-dimensional simulations. Section B.5.2 details how the correction is applied to numerical solver outputs and evaluates its impact on prediction accuracy. Section B.5.3 outlines how the correction can be incorporated into the MLP surrogate model for future integration into the inversion framework.

B.5.1 Motivation and method overview

Our current approach assumes a two-dimensional simulation domain, ignoring vertical transport. Extending the numerical solver to three dimensions would eliminate this assumption but would drastically increase computational cost. For instance, transitioning from a two-dimensional mesh of size 200×200 (i.e., 40,000 cells) to a three-dimensional mesh of size $200 \times 200 \times 200$ (i.e., 8 million cells) increases simulation cost. Depending on the solver's linear or quadratic scaling, this transforms a 3-minute simulation into one taking any time between 10 hours and 83 days.

We propose applying a vertical Gaussian plume correction to the two-dimensional numerical solver solution. This works by taking our current two-dimensional simulated gas transport denoted $C_{\text{ns}}(\dot{x}, \dot{y} \mid \tilde{x}, \tilde{y})$ and vertically expanding it using the following vertical Gaussian plume correction:

$$
\begin{aligned}
C_{\text{vgp}}(\sigma_V \mid \tilde{x}, \tilde{y}, \tilde{z}) = \frac{1}{\sqrt{2\pi}\sigma_V} \times \Bigg(& \exp\left\{-\frac{\delta_V^2}{2\sigma_V^2}\right\} \\
& + \sum_{j=1}^{n_{\text{refl}}} \Bigg[\exp\left\{ -\frac{1}{2}\frac{(2\lfloor (j+1)/2 \rfloor P + (-1)^j(\delta_V + \tilde{z}) - \tilde{z})^2}{\sigma_V^2} \right\} \\
& + \exp\left\{ -\frac{1}{2}\frac{(2\lfloor j/2 \rfloor P + (-1)^{j-1}(\delta_V + \tilde{z}) + \tilde{z})^2}{\sigma_V^2} \right\} \Bigg] \Bigg).
\end{aligned}
$$

The vertical Gaussian plume correction comes directly from the Gaussian plume equation (2.2.4). This leads to the following corrected numerical solution:

$$
C_{\text{ns + vgp}}(\dot{x}, \dot{y}, \dot{z} \mid \tilde{x}, \tilde{y}, \tilde{z}, \sigma_V) = C_{\text{ns}}(\dot{x}, \dot{y} \mid \tilde{x}, \tilde{y}) \times C_{\text{vgp}}(\sigma_V \mid \tilde{x}, \tilde{y}, \tilde{z}). \tag{B.5.1}
$$

For any prediction using this correction, the following assumptions apply:

1. Steady-state vertical wind: A reasonable assumption over short temporal windows (e.g., 1–2 minutes).
2. Vertically infinite obstacles: The gas is assumed not to flow over or under obstacles. This confines applicability to scenarios where vertical bypass is negligible.

Equation B.5.1 is parametrised by the vertical wind sigma σ_V – representing the standard deviation of the vertical Gaussian distribution forming part of the Gaussian plume model – which itself is typically parametrised using power-law distributions (Pasquill, 1961; Briggs, 1973; Hirst et al., 2013). We generalise the power-law parametrisation from Hirst et al. (2013) by adding dispersion parameters $a_V \in \mathbb{R}^+$, and $b_V \in (0, 1]$ as follows:

$$\sigma_V = a_V \left(\delta_R \tan(\gamma_V)\right)^{b_V} + h, \tag{B.5.2}$$

where $\boldsymbol{\gamma_V} \in \mathbb{R}^+$ is the 1 minute standard deviation of the vertical wind direction, $\delta_R \in \mathbb{R}^+$ is the downwind distance of location (x, y, z) from the source located at $(\tilde{x}, \tilde{y}, \tilde{z})$, and $h \in \mathbb{R}^+$ the source aperture's half-height. When the measurement location is upwind from the source, we set the Gaussian plume concentration contribution to zero.

B.5.2 Numerical solver + vertical Gaussian plume correction

Equation B.5.1 is parametrised by the vertical wind sigma σ_V, which itself is parametrised by the following dispersion parameters: a_V and b_V. These parameters directly control the shape of the vertical Gaussian plume correction and must therefore be estimated to best explain the sensor measurements.

To assess the benefit of the correction, we predicted Source 1 sensor measurements using the two-dimensional numerical solver and its vertical Gaussian plume corrected version. Figure B.5.1 shows the Gaussian correction applied to a single sensor measurement prediction; in practice, we truncate any values for negative vertical heights (i.e. below ground level). Figure B.5.2 shows the resulting time-averaged predictions, computed using the method in Appendix B.2.

Corrected predictions are closer to the true sensor measurements, leading to a 4% reduction in mean absolute percentage error (12.55% vs 13.02%). These results are reproducible via the Jupyter notebook included in the folder: `Code / Further_work / vertical_Gaussian_plume_correction` .

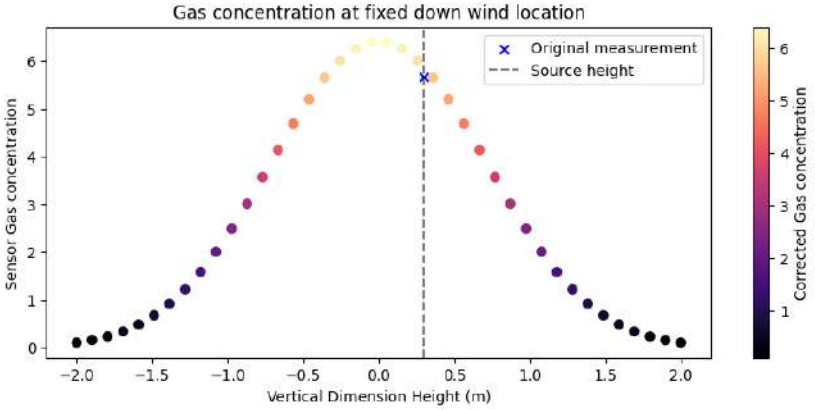


Figure B.5.1: Vertical Gaussian plume corrected sensor observation.

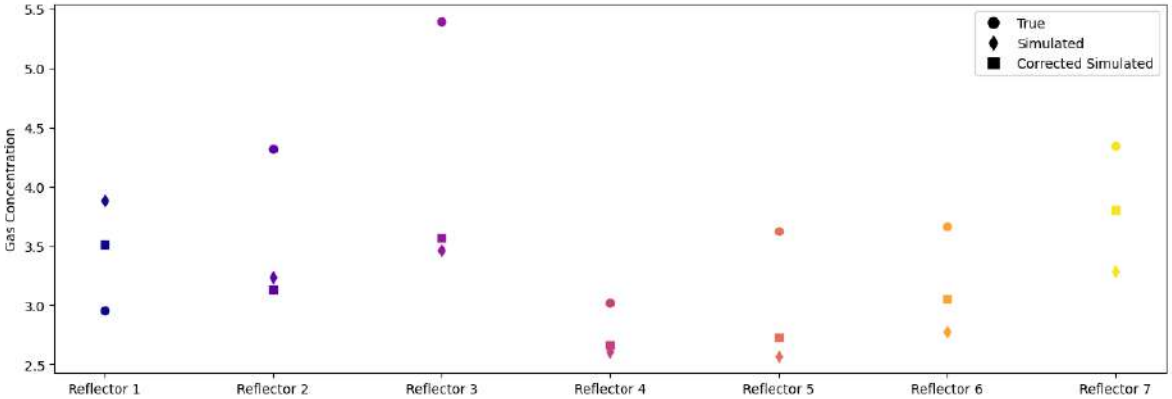


Figure B.5.2: Chilbolton Source 1 time-averaged sensor measurement prediction using a two-dimensional numerical solver and its vertically corrected version.

B.5.3 Integrating correction into MLP surrogates

This vertical correction can be incorporated directly into our surrogate-based inversion pipeline. We extend the input to the MLP surrogate to include the vertical dispersion parameters a_V and b_V, resulting in an input vector $(\tilde{x}, \tilde{y}, a_V, b_V)$, while keeping the output as the predicted sensor measurement.

The corrected training data is generated by applying the vertical correction to our original two-dimensional numerical solver simulations for various plausible combinations of a_V and b_V. This approach adds minimal overhead and enables the surrogate to learn mappings consistent with vertically corrected physics.

While this modified surrogate was not used in the main inversion experiments, it provides a natural extension to our framework. The vertical dispersion parameters can be estimated jointly with other source parameters via the SIR particle filter, allowing fully corrected inference with minimal added computational cost.

Bibliography

Albani, R. A., Albani, V. V., and Neto, A. J. S. (2020). Source characterization of airborne pollutant emissions by hybrid metaheuristic/gradient-based optimization techniques. *Environmental Pollution*, 267:115618.

Andrieu, C. and Roberts, G. O. (2009). The pseudo-marginal approach for efficient Monte Carlo computations. *The Annals of Statistics*, pages 697–725.

Aoki, M. and Havenner, A. (1991). State space modeling of multiple time series. *Econometric Reviews*, 10(1):1–59.

Arenas-López, J. P. and Badaoui, M. (2020). The Ornstein-Uhlenbeck process for estimating wind power under a memoryless transformation. *Energy*, 213:118842.

Arulampalam, M. S., Maskell, S., Gordon, N., and Clapp, T. (2002). A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188.

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

Atkinson, D. G., Bailey, D. T., Irwin, J. S., and Touma, J. S. (1997). Improvements to the EPA industrial source complex dispersion model. *Journal of Applied Meteorology*, 36(8):1088 – 1095.

Augustin, L., Barbante, C., Barnes, P. R., Marc Barnola, J., Bigler, M., Castellano, E., Cattani, O., Chappellaz, J., Dahl-Jensen, D., Delmonte, B., et al. (2004). Eight glacial cycles from an Antarctic ice core. *Nature*, 429(6992):623–628.

Bakels, L., Tatsii, D., Tipka, A., Thompson, R., Dütsch, M., Blaschek, M., Seibert, P., Baier, K., Bucci, S., Cassiani, M., et al. (2024). Flexpart version 11: Improved accuracy, efficiency, and flexibility. *Geoscientific Model Development*, 17(21):7595–7627.

Benson, V., Bastos, A., Reimers, C., Winkler, A. J., Yang, F., and Reichstein, M. (2025). Atmospheric transport modeling of CO_2 with neural networks. *Journal of Advances in Modeling Earth Systems*, 17(2):e2024MS004655.

Bernoulli, D. (1738). *Hydrodynamica, sive, de viribus et motibus fluidorum commentarii*. Sumptibus Johannis Reinholdi Dulseckeri, typis Joh. Henr. Deckeri.

Berzuini, C. and Gilks, W. (2001). Resample-move filtering with cross-model jumps. In *Sequential Monte Carlo Methods in Practice*, pages 117–138. Springer.

Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. (2017). Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877.

Blom, H. A. and Bar-Shalom, Y. (2002). The interacting multiple model algorithm for systems with Markovian switching coefficients. *IEEE Transactions on Automatic Control*, 33(8):780–783.

Boughton, B., Delaurentis, J., and Dunn, W. (1987). A stochastic model of particle dispersion in the atmosphere. *Boundary-Layer Meteorology*, 40:147–163.

Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., and Zhang, Q. (2018). JAX: Composable transformations of Python+NumPy programs. http://github.com/google/jax. Accessed: 2025-09-04.

Brasseur, G. P. and Jacob, D. J. (2017). *Modeling of Atmospheric Chemistry.* Cambridge University Press.

Briggs, G. A. (1973). Diffusion estimation for small emissions. *Atmospheric Turbulence and Diffusion Laboratory*, 965:83–145.

Brooks, S., Gelman, A., Jones, G., and Meng, X.-L. (2011). *Handbook of Markov Chain Monte Carlo.* CRC press.

Butz, A., Galli, A., Hasekamp, O., Landgraf, J., Tol, P., and Aben, I. (2012). TROPOMI aboard Sentinel-5 Precursor: Prospective performance of CH_4 retrievals for aerosol and cirrus loaded atmospheres. *Remote Sensing of Environment*, 120:267–276.

Byron, F. W. and Fuller, R. W. (1992). *Mathematics of Classical and Quantum Physics.* Dover Publications.

Calhoun, D. and LeVeque, R. J. (2000). A Cartesian grid finite-volume method for the advection-diffusion equation in irregular geometries. *Journal of Computational Physics*, 157(1):143–180.

Canuto, C., Hussaini, M. Y., Quarteroni, A., and Zang, T. A. (2006). *Spectral Methods: Fundamentals in Single Domains.* Springer.

Carruthers, D., Holroyd, R., Hunt, J., Weng, W., Robins, A., Apsley, D., Thompson, D., and Smith, F. (1994). UK-ADMS: A new approach to modelling dispersion in the Earth's atmospheric boundary layer. *Journal of Wind Engineering and Industrial Aerodynamics*, 52:139–153.

Cartwright, L., Zammit-Mangion, A., Bhatia, S., Schroder, I., Phillips, F., Coates, T., Negandhi, K., Naylor, T., Kennedy, M., Zegelin, S., Wokker, S., Deutscher, N., and

Feitz, A. (2019). Bayesian atmospheric tomography for detection and quantification of methane emissions: Application to data from the 2015 Ginninderra release experiment. *Atmospheric Measurement Techniques*, 12(9):4659–4676.

Cartwright, L., Zammit-Mangion, A., and Deutscher, N. M. (2023). Emulation of greenhouse-gas sensitivities using variational autoencoders. *Environmetrics*, 34(2).

Chandra, R., Azam, D., Kapoor, A., and Müller, R. D. (2020). Surrogate-assisted Bayesian inversion for landscape and basin evolution models. *Geoscientific Model Development*, 13(7):2959–2979.

Chib, S. and Greenberg, E. (1995). Understanding the Metropolis-Hastings algorithm. *The American Statistician*, 49(4):327–335.

Chopin, N. and Papaspiliopoulos, O. (2020). *An Introduction to Sequential Monte Carlo*. Springer.

Chorin, A. J. (1968). Numerical solution of the Navier-Stokes equations. *Mathematics of Computation*, 22(104):745–762.

Cimorelli, A. J., Perry, S. G., Venkatram, A., Weil, J. C., Paine, R. J., Wilson, R. B., Lee, R. F., Peters, W. D., and Brode, R. W. (2005). AERMOD: A dispersion model for industrial source applications. Part I: General model formulation and boundary layer characterization. *Journal of Applied Meteorology and Climatology*, 44(5):682–693.

Coumou, D. and Rahmstorf, S. (2012). A decade of weather extremes. *Nature Climate Change*, 2(7):491–496.

Courant, R., Friedrichs, K., and Lewy, H. (1928). Über die Partiellen Differenzengleichungen der Mathematischen Physik. *Mathematische Annalen*, 100(1):32–74.

Crank, J. and Nicolson, P. (1947). A practical method for numerical evaluation of solutions of partial differential equations of the heat-conduction type. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 43, pages 50–67. Cambridge University Press.

Cui, J., Lang, J., Chen, T., Cheng, S., Shen, Z., and Mao, S. (2019). Investigating the impacts of atmospheric diffusion conditions on source parameter identification based on an optimized inverse modelling method. *Atmospheric Environment*, 205:19–29.

Dadheech, N., He, T.-L., and Turner, A. J. (2025). High-resolution greenhouse gas flux inversions using a machine learning surrogate model for atmospheric transport. *Atmospheric Chemistry and Physics*, 25(10):5159–5174.

D'Alembert, J. l. R. (1768). Paradoxe proposé aux géomètres sur la résistance des fluides. *Opuscules Mathématiques*.

De Jong, T. A., Maasakkers, J. D., Irakulis-Loitxate, I., Randles, C. A., Tol, P., and Aben, I. (2025). Daily global methane super-emitter detection and source identification with sub-daily tracking. *Geophysical Research Letters*, 52(8):e2024GL111824.

Del Moral, P. (1997). Nonlinear filtering: Interacting particle resolution. *Comptes Rendus de l'Académie des Sciences-Series I-Mathematics*, 325(6):653–658.

Dia, B. M., Fehler, M., Kaka, S. I., Scarinci, A., bin Waheed, U., and Gu, C. (2024). Greedy selection of optimal location of sensors for uncertainty reduction in seismic moment tensor inversion. *Journal of Computational Physics*, 519:113431.

Donea, J. and Huerta, A. (2003). *Finite Element Methods for Flow Problems*. John Wiley & Sons.

Douc, R. and Cappé, O. (2005). Comparison of resampling schemes for particle filtering. In *ISPA 2005. Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis, 2005.*, pages 64–69. IEEE.

Doucet, A., de Freitas, N., and Gordon, N. (2000). Sequential Monte Carlo methods in practice. *Statistics and computing*, 10(3):197–208.

Dowd, E., Manning, A. J., Orth-Lashley, B., Girard, M., France, J., Fisher, R. E., Lowry, D., Lanoisellé, M., Pitt, J. R., Stanley, K. M., et al. (2024). First validation of high-resolution satellite-derived methane emissions from an active gas leak in the UK. *Atmospheric Measurement Techniques*, 17(5):1599–1615.

Draxler, R. and Rolph, G. (2010). Hysplit (hybrid single-particle lagrangian integrated trajectory) model access via noaa arl ready website (http://ready. arl. noaa. gov/hysplit. php). noaa air resources laboratory. *Silver Spring, MD*, 25(1).

Dueben, P. D. and Bauer, P. (2018). Challenges and design choices for global weather and climate models based on machine learning. *Geoscientific Model Development*, 11(10):3999–4009.

Euler, L. (1755). Remarques sur les mémoires précédens de M. Bernoulli. *Mémoires de l'Académie des Sciences de Berlin*, pages 196–222.

Eymard, R., Gallouët, T., and Herbin, R. (2000). Finite volume methods. *Handbook of Numerical Analysis*, 7:713–1018.

Fearnhead, P., Nemeth, C., Oates, C. J., and Sherlock, C. (2024). Scalable Monte Carlo for Bayesian Learning. *arXiv preprint arXiv:2407.12751*.

Ferziger, J. H. and Perić, M. (2002). *Computational Methods for Fluid Dynamics*, volume 586. Springer.

Fick, A. (1855). Ueber diffusion. *Annalen der Physik*, 170(1):59–86.

Fillola, E., Santos-Rodriguez, R., Manning, A., O'Doherty, S., and Rigby, M. (2023). A machine learning emulator for Lagrangian particle dispersion model footprints: A case study using NAME. *Geoscientific Model Development*, 16(7):1997–2009.

Fillola, E., Santos-Rodriguez, R., Tunnicliffe, R., Clark, J., Keshtmand, N., Ganesan, A., and Rigby, M. (2025). Enabling fast greenhouse gas emissions inference from satellites with gates: A graph-neural-network atmospheric transport emulation system. *EGUsphere*, 2025:1–27.

Finn, D., Clawson, K., Eckman, R., Liu, H., Russell, E., Gao, Z., and Brooks, S. (2016). Project Sagebrush: Revisiting the value of the horizontal plume spread parameter σ_y. *Journal of Applied Meteorology and Climatology*, 55(6):1305–1322.

Foote, E. (1856). Circumstances affecting the heat of the Sun's rays. *American Association for the Advancement of Science*, 23.

Fourier, J. B. J. (1822). *Théorie Analytique de la Chaleur*. Firmin Didot.

Fox, T. A., Barchyn, T. E., Risk, D., Ravikumar, A. P., and Hugenholtz, C. H. (2019). A review of close-range and screening technologies for mitigating fugitive methane emissions in upstream oil and gas. *Environmental Research Letters*, 14(5):053002.

Friedlingstein, P., O'Sullivan, M., Jones, M. W., Andrew, R. M., Bakker, D. C., Hauck, J., Landschützer, P., Le Quéré, C., Luijkx, I. T., Peters, G. P., et al. (2023). Global carbon budget 2023. *Earth System Science Data*, 15(12):5301–5369.

Garcia-Fernandez, A. F., Grajal, J., and Morelande, M. R. (2013). Two-layer particle filter for multiple target detection and tracking. *IEEE Transactions on Aerospace and Electronic Systems*, 49(3):1569–1588.

Geman, S. and Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern analysis and Machine Intelligence*, PAMI-6(6):721–741.

Girolami, M. and Calderhead, B. (2011). Riemann manifold Langevin and Hamiltonian Monte Carlo methods. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 73(2):123–214.

Glorot, X. and Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 249–256. JMLR Workshop and Conference Proceedings.

Gordon, N. J., Salmond, D. J., and Smith, A. F. (1993). Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings F (Radar and Signal Processing)*, 140(2):107–113.

Grenander, U. and Miller, M. I. (1994). Representations of knowledge in complex systems. *Journal of the Royal Statistical Society: Series B (Methodological)*, 56(4):549–581.

Guo, X., Li, W., and Iorio, F. (2016). Convolutional neural networks for steady flow approximation. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 481–490.

Hamilton, J. D. (1994). State-space models. *Handbook of Econometrics*, 4:3039–3080.

Hanna, S. R., Briggs, G. A., and Hosker Jr, R. P. (1982). Handbook on atmospheric diffusion. Technical report, National Oceanic and Atmospheric Administration, Oak Ridge, TN (USA). Atmospheric Turbulence and Diffusion Lab.

Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*.

He, J., Wu, P., Tong, Y., Zhang, X., Lei, M., and Gao, J. (2021). Bearing fault diagnosis via improved one-dimensional multi-scale dilated CNN. *Sensors*, 21(21):7319.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 770–778.

He, T.-L., Dadheech, N., Thompson, T. M., and Turner, A. J. (2025). Footnet v1. 0: Development of a machine learning emulator of atmospheric transport. *Geoscientific Model Development*, 18(5):1661–1671.

Hirst, B., Jonathan, P., del Cueto, F. G., Randell, D., and Kosut, O. (2013). Locating and quantifying gas emission sources using remotely obtained concentration data. *Atmospheric Environment*, 74:141–158.

Hirst, B., Randell, D., Jones, M., Chu, J., Kannath, A., Macleod, N., Dean, M., and Weidmann, D. (2020). Methane emissions: Remote mapping and source quantification using an open-path laser dispersion spectrometer. *Geophysical Research Letters*, 47(10).

Hirst, B., Randell, D., Jones, M., Weidman, D., and Dean, M. (2018). Mapping CO_2 and CH_4 emissions: Field-trial evaluation of LightSource for remotely estimating the locations and mass emission rates of sources. In *Fifth CO_2 Geological Storage Workshop*.

Holl, P. and Thuerey, N. (2024). Φ_{Flow} (PhiFlow): Differentiable simulations for PyTorch, TensorFlow and Jax. In *International Conference on Machine Learning*. PMLR.

Hooker, G., Ellner, S. P., Roditi, L. D. V., and Earn, D. J. (2011). Parameterizing state–space models for infectious disease dynamics by generalized profiling: Measles in ontario. *Journal of The Royal Society Interface*, 8(60):961–974.

Hughes, T. P., Kerry, J. T., Álvarez-Noriega, M., Álvarez-Romero, J. G., Anderson, K. D., Baird, A. H., Babcock, R. C., Beger, M., Bellwood, D. R., Berkelmans, R., et al. (2017). Global warming and recurrent mass bleaching of corals. *Nature*, 543(7645):373–377.

IJzermans, R., Jones, M., Weidmann, D., van de Kerkhof, B., and Randell, D. (2024). Long-term continuous monitoring of methane emissions at an oil and gas facility using a multi-open-path laser dispersion spectrometer. *Scientific Reports*, 14(1):623.

Jacob, D. J., Varon, D. J., Cusworth, D. H., Dennison, P. E., Frankenberg, C., Gautam, R., Guanter, L., Kelley, J., McKeever, J., Ott, L. E., et al. (2022). Quantifying methane emissions from the global scale down to point sources using satellite observations of atmospheric methane. *Atmospheric Chemistry and Physics Discussions*, 2022:1–44.

Jazwinski, A. H. (1970). *Stochastic Processes and Filtering Theory.* Academic Press.

Jervis, D., McKeever, J., Durak, B. O., Sloan, J. J., Gains, D., Varon, D. J., Ramier, A., Strupler, M., and Tarrant, E. (2021). The GHGSat-D imaging spectrometer. *Atmospheric Measurement Techniques*, 14(3):2127–2140.

Jia, M., Fish, R., Daniels, W. S., Sprinkle, B., and Hammerling, D. (2025). A fast and lightweight implementation of the Gaussian puff model for near-field atmospheric transport of trace gasses. *Scientific Reports*, 15(1):18710.

Jiang, J., Zou, Z., and Lu, Y. (2023). A ConvLSTM-based prediction model of aurora evolution during the substorm expansion phase. *Earth and Space Science*, 10(4):e2022EA002721.

Jones, A., Thomson, D., Hort, M., and Devenish, B. (2007). The uk met office's next-generation atmospheric dispersion model, name iii. In *Air pollution modeling and its application XVII*, pages 580–589. Springer.

Jongaramrungruang, S., Thorpe, A. K., Matheou, G., and Frankenberg, C. (2022). Methanet – an AI-driven approach to quantifying methane point-source emission from high-resolution 2-D plume imagery. *Remote Sensing of Environment*, 269:112809.

Julier, S., Uhlmann, J., and Durrant-Whyte, H. F. (2002). A new method for the nonlinear transformation of means and covariances in filters and estimators. *IEEE Transactions on Automatic Control*, 45(3):477–482.

Kahl, J. D. and Chapman, H. L. (2018). Atmospheric stability characterization using the Pasquill method: A critical evaluation. *Atmospheric Environment*, 187:196–209.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*.

Karimi, M., Massoudi, M., Dayal, K., and Pozzi, M. (2023). High-dimensional nonlinear Bayesian inference of poroelastic fields from pressure data. *Mathematics and Mechanics of Solids*, 28(9):2108–2131.

Karniadakis, G. E., Kevrekidis, I. G., Lu, L., Perdikaris, P., Wang, S., and Yang, L. (2021). Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440.

Kennedy, M. C. and O'Hagan, A. (2001). Bayesian calibration of computer models. *Journal of the Royal Statistical Society: Series B (Methodological)*, 63(3):425–464.

Kim, B., Azevedo, V. C., Thuerey, N., Kim, T., Gross, M., and Solenthaler, B. (2019). Deep fluids: A generative network for parameterized fluid simulations. In *Computer Graphics Forum*, volume 38, pages 59–70. Wiley Online Library.

Kitagawa, G. (1996). Monte Carlo filter and smoother for non-Gaussian nonlinear state space models. *Journal of Computational and Graphical Statistics*, 5(1):1–25.

Klambauer, G., Unterthiner, T., Mayr, A., and Hochreiter, S. (2017). Self-normalizing neural networks. *Advances in Neural Information Processing Systems*, 30.

Kloeden, P. E., Platen, E., and Schurz, H. (2002). *Numerical Solution of SDE Through Computer Experiments*. Springer Science & Business Media.

Kocijan, J., Hvala, N., Perne, M., Mlakar, P., Grašič, B., and Božnar, M. Z. (2023). Surrogate modelling for the forecast of seveso-type atmospheric pollutant dispersion. *Stochastic Environmental Research and Risk Assessment*, 37(1):275–290.

Kolmogorov, A. N. (1991). The local structure of turbulence in incompressible viscous fluid for very large Reynolds numbers. *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences*, 434(1890):9–13.

Kutz, J. N. (2017). Deep learning in fluid dynamics. *Journal of Fluid Mechanics*, 814:1–4.

Lagrange, J. L. (1788). *Mechanique Analitique*. La Veuve Desaint.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551.

Lee, H. and Romero, J. (2023). Climate Change 2023 Synthesis Report IPCC, 2023.

Li, Z., Kovachki, N., Azizzadenesheli, K., Liu, B., Bhattacharya, K., Stuart, A., and Anandkumar, A. (2021). Fourier neural operator for parametric partial differential equations. *International Conference on Learning Representations*.

Libre, J.-M., Guérin, S., Konaté, B., Castellari, A., Mames, C., Le Guellec, M., Mailliard, T., Tripathi, A., Souprayen, C., Connan, O., Leroy, C., Laguionie, P., Defenouillère, P., Letellier, B., Hébert, D., and Maro, D. (2011). Validation campaign of a CFD tool on a petrochemical site with wind fluctuations integration. In *HARMO 2011 - Proceedings of the 14th International Conference on Harmonisation within Atmospheric Dispersion Modelling for Regulatory Purposes*, page 37 – 41.

Lin, J., Gerbig, C., Wofsy, S., Andrews, A., Daube, B., Davis, K., and Grainger, C. (2003). A near-field tool for simulating the upstream influence of atmospheric observations: The stochastic time-inverted lagrangian transport (stilt) model. *Journal of Geophysical Research: Atmospheres*, 108(D16).

Liu, Q., Chu, M., and Thuerey, N. (2025). Config: Towards conflict-free training of physics informed neural networks. *International Conference on Learning Representations.*

Liu, X., Yeo, K., Klein, L., Hwang, Y., Phan, D., and Liu, X. (2022). Optimal sensor placement for atmospheric inverse modelling. In *2022 IEEE International Conference on Big Data (Big Data)*, pages 4848–4853. IEEE.

Liu, Z. and Li, X. (2022). Sensor layout strategy for source term estimation of external pollution sources in urban neighbourhoods. *Building and Environment*, 220:109276.

Lorente, A., Borsdorff, T., Martinez-Velarte, M. C., Butz, A., Hasekamp, O. P., Wu, L., and Landgraf, J. (2022). Evaluation of the methane full-physics retrieval applied to TROPOMI ocean sun glint measurements. *Atmospheric Measurement Techniques*, 15(22):6585–6603.

Lotrecchiano, N., Sofia, D., Giuliano, A., Barletta, D., Poletto, M., et al. (2020). Pollution dispersion from a fire using a Gaussian plume model. *International Journal of Safety and Security Engineering*, 10:431–439.

Ludwig, F., Gasiorek, L., and Ruff, R. (1977). Simplification of a Gaussian puff model for real-time minicomputer use. *Atmospheric Environment (1967)*, 11(5):431–436.

Lushi, E. and Stockie, J. M. (2010). An inverse Gaussian plume approach for estimating atmospheric pollutant emissions from multiple point sources. *Atmospheric Environment*, 44(8):1097–1107.

Lüthi, D., Le Floch, M., Bereiter, B., Blunier, T., Barnola, J.-M., Siegenthaler, U., Raynaud, D., Jouzel, J., Fischer, H., Kawamura, K., et al. (2008). High-resolution carbon dioxide concentration record 650,000–800,000 years before present. *Nature*, 453(7193):379–382.

Lyons, T. W., Reinhard, C. T., and Planavsky, N. J. (2014). The rise of oxygen in Earth's early ocean and atmosphere. *Nature*, 506(7488):307–315.

Ma, D., Gao, J., Zhang, Z., and Zhao, H. (2021). Identifying atmospheric pollutant sources using a machine learning dispersion model and Markov chain Monte Carlo methods. *Stochastic Environmental Research and Risk Assessment*, 35:271–286.

Maasakkers, J. D., Jacob, D. J., Sulprizio, M. P., Scarpelli, T. R., Nesser, H., Sheng, J.-X., Zhang, Y., Hersher, M., Bloom, A. A., Bowman, K. W., et al. (2019). Global distribution of methane emissions, emission trends, and oh concentrations and trends inferred from an inversion of GOSAT satellite data for 2010–2015. *Atmospheric Chemistry and Physics*, 19(11):7859–7881.

Maasakkers, J. D., Varon, D. J., Elfarsdóttir, A., McKeever, J., Jervis, D., Mahapatra, G., Pandey, S., Lorente, A., Borsdorff, T., Foorthuis, L. R., et al. (2022). Using satellites to uncover large methane emissions from landfills. *Science Advances*, 8(31):eabn9683.

MacCormack, R. W. (2002). The effect of viscosity in hypervelocity impact cratering. In *Frontiers of Computational Fluid Dynamics 2002*, pages 27–43. World Scientific Publishing Co Pte Ltd.

Mao, S., Lang, J., Chen, T., and Cheng, S. (2021). Improving source inversion performance of airborne pollutant emissions by modifying atmospheric dispersion scheme through sensitivity analysis combined with optimization model. *Environmental Pollution*, 284:117186.

Mao, S., Lang, J., Chen, T., Cheng, S., Wang, C., Zhang, J., and Hu, F. (2020).

Impacts of typical atmospheric dispersion schemes on source inversion. *Atmospheric Environment*, 232:117572.

Maruyama, G. (1955). Continuous Markov processes and stochastic equations. *Rendiconti del Circolo Matematico di Palermo*, 4:48–90.

Matsson, J. E. (2022). *An Introduction to ANSYS Fluent 2022*. SDC Publications.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092.

Micallef, A. and Micallef, C. (2024). The Gaussian Plume Model Equation for Atmospheric Dispersion Corrected for Multiple Reflections at Parallel Boundaries: A Mathematical Rewriting of the Model and Some Numerical Testing. *Sci*, 6(3):48.

Mishra, A., Habermann, D., Schmitt, M., Radev, S. T., and Bürkner, P.-C. (2025). Robust amortized bayesian inference with self-consistency losses on unlabeled data. *arXiv preprint arXiv:2501.13483*.

Moishin, M., Deo, R. C., Prasad, R., Raj, N., and Abdulla, S. (2021). Designing deep-based learning flood forecast model with ConvLSTM hybrid algorithm. *IEEE Access*, 9:50982–50993.

Moukalled, F., Mangani, L., Darwish, M., Moukalled, F., Mangani, L., and Darwish, M. (2016). *The Finite Volume Method*. Springer.

Murad, J. (2021). Harnessing the power of the cloud-computational fluid dynamics with SimScale. In *Fluids Engineering Division Summer Meeting*, volume 85284, page V001T02A049. American Society of Mechanical Engineers.

Navier, C. L. (1823). Mémoire sur les lois du mouvement des fluides. *Mémoires de l'Académie Royale des Sciences de l'Institut de France*, 6(1823):389–440.

Nemeth, C., Fearnhead, P., Mihaylova, L., and Vorley, D. (2012). Particle learning methods for state and parameter estimation. In *9th IET Data Fusion & Target Tracking Conference (DF&TT 2012): Algorithms & Applications*, pages 12–1. IET.

Nesser, H., Jacob, D. J., Maasakkers, J. D., Lorente, A., Chen, Z., Lu, X., Shen, L., Qu, Z., Sulprizio, M. P., Winter, M., et al. (2024). High-resolution US methane emissions inferred from an inversion of 2019 TROPOMI satellite data: Contributions from individual states, urban areas, and landfills. *Atmospheric Chemistry and Physics*, 24(8):5069–5091.

Newman, T., Nemeth, C., Jones, M., and Jonathan, P. (2025). Probabilistic inversion modeling of gas emissions: A gradient-based MCMC estimation of Gaussian plume parameters. *The Annals of Applied Statistics*, 19(4):2937–2956.

OMGP2.0 (2024). The Oil & Gas Methane Partnership 2.0. https://www.ogmpartnership.org/.

Papamakarios, G., Nalisnick, E., Rezende, D. J., Mohamed, S., and Lakshminarayanan, B. (2021). Normalizing flows for probabilistic modeling and inference. *Journal of Machine Learning Research*, 22(57):1–64.

Pasquill, F. (1961). The estimation of the dispersion of windborne material. *Meteorological Magazine*, 90(1063):33–49.

Pasquill, F. and Smith, F. B. (1983). *Atmospheric Diffusion*, volume 437. Ellis Horwood Chichester.

Pathak, J., Hunt, B., Girvan, M., Lu, Z., and Ott, E. (2018). Model-free prediction of large spatiotemporally chaotic systems from data: A reservoir computing approach. *Physical Review Letters*, 120(2):024102.

Pfaff, T., Fortunato, M., Sanchez-Gonzalez, A., and Battaglia, P. (2020). Learning mesh-based simulation with graph networks. In *International Conference on Learning Representations*.

Pitt, M. K. and Shephard, N. (1999). Filtering via simulation: Auxiliary particle filters. *Journal of the American Statistical Association*, 94(446):590–599.

Plumlee, M. (2019). Computer model calibration with confidence and consistency. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 81(3):519–545.

Prashad, C. D. (2025). State-space modelling for infectious disease surveillance data: Dynamic regression and covariance analysis. *Infectious Disease Modelling*, 10(2):591–627.

Qiu, S., Chen, B., Wang, R., Zhu, Z., Wang, Y., and Qiu, X. (2018). Atmospheric dispersion prediction and source estimation of hazardous gas using artificial neural network, particle swarm optimization and expectation maximization. *Atmospheric Environment*, 178:158–163.

Quiñonero-Candela, J., Sugiyama, M., Schwaighofer, A., and Lawrence, N. D. (2022). *Dataset Shift in Machine Learning*. MIT Press.

Raissi, M., Perdikaris, P., and Karniadakis, G. E. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707.

Ramadan, A., Al-Sudairawi, M., Alhajraf, S., and Khan, A. (2008). Total SO_2 emissions from power stations and evaluation of their impact in Kuwait using a Gaussian plume dispersion model. *American Journal of Environmental Sciences*, 4(1):1–12.

Rasp, S. and Thuerey, N. (2021). Data-driven medium-range weather prediction with a resnet pretrained on climate simulations: A new model for WeatherBench. *Journal of Advances in Modeling Earth Systems*, 13(2):e2020MS002405.

Ricciardi, L. M. and Sacerdote, L. (1979). The Ornstein-Uhlenbeck process as a model for neuronal activity: I. Mean and variance of the firing time. *Biological Cybernetics*, 35(1):1–9.

Robert, C. P., Casella, G., and Casella, G. (1999). *Monte Carlo Statistical Methods*, volume 2. Springer.

Roberts, C., Maasakkers, J., Schuit, B. J., Houweling, S., and Aben, I. (2024). Using a convolutional neural network to improve tropomi super-emitter methane plume emission rate quantification. In *AGU Fall Meeting Abstracts*, volume 2024, pages A53I–2214.

Roberts, G. O. and Rosenthal, J. S. (1998). Optimal scaling of discrete approximations to Langevin diffusions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 60(1):255–268.

Roberts, G. O. and Tweedie, R. L. (1996). Exponential convergence of Langevin distributions and their discrete approximations. *Bernoulli*, 2(4):341–363.

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241. Springer.

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386.

Royer, D. L., Berner, R. A., Montañez, I. P., Tabor, N. J., Beerling, D. J., et al. (2004). Co_2 as a primary driver of Phanerozoic climate. *GSA today*, 14(3):4–10.

Sánchez-García, E., Gorroño, J., Irakulis-Loitxate, I., Varon, D. J., and Guanter, L. (2021). Mapping methane plumes at very high spatial resolution with the WorldView-3 satellite. *Atmospheric Measurement Techniques Discussions*, 2021:1–26.

Saunois, M., Martinez, A., Poulter, B., Zhang, Z., Raymond, P. A., Regnier, P., Canadell, J. G., Jackson, R. B., Patra, P. K., Bousquet, P., et al. (2025). Global methane budget 2000–2020. *Earth System Science Data*, 17(5):1873–1958.

Scheutz, C., Samuelsson, J., Fredenslund, A. M., and Kjeldsen, P. (2011). Quantification of multiple methane emission sources at landfills using a double tracer technique. *Waste Management*, 31(5):1009–1017.

Schuit, B. J., Maasakkers, J. D., Bijl, P., Mahapatra, G., Van den Berg, A.-W., Pandey, S., Lorente, A., Borsdorff, T., Houweling, S., Varon, D. J., et al. (2023). Automated detection and monitoring of methane super-emitters using satellite data. *Atmospheric Chemistry and Physics Discussions*, 2023:1–47.

Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics*, pages 461–464.

Scire, J. S., Strimaitis, D. G., Yamartino, R. J., et al. (2000). A user's guide for the CALPUFF dispersion model. *Earth Tech, Inc*, 521:1–521.

Seinfeld, J. H. and Pandis, S. N. (2016). *Atmospheric Chemistry and Physics: From Air Pollution to Climate Change*. John Wiley & Sons.

Shi, X., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W.-K., and Woo, W.-c. (2015). Convolutional LSTM network: A machine learning approach for precipitation nowcasting. *Advances in Neural Information Processing Systems*, 28.

Sisson, S. A., Fan, Y., and Beaumont, M. (2018). *Handbook of Approximate Bayesian Computation*. CRC press.

Stam, J. (1999). Stable fluids. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '99, page 121–128. ACM Press/Addison-Wesley Publishing Co.

Stockie, J. M. (2011). The mathematics of atmospheric dispersion modeling. *Siam Review*, 53(2):349–372.

Stokes, G. G. (1845). On the theories of the internal friction of fluids in motion, and of the equilibrium and motion of elastic solids. *Transactions of the Cambridge Philosophical Society.*

Stone, L. D., Streit, R. L., Corwin, T. L., and Bell, K. L. (2013). *Bayesian Multiple Target Tracking*. Artech House.

Tan, M. and Le, Q. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114. PMLR.

Temam, R. (1968). Une méthode d'approximation de la solution des équations de Navier-Stokes. *Bulletin de la Société Mathématique de France*, 96:115–152.

Thuerey, N., Weißenow, K., Prantl, L., and Hu, X. (2020). Deep learning methods for Reynolds-averaged Navier–Stokes simulations of airfoil flows. *AIAA Journal*, 58(1):25–36.

Tong, B., Wang, X., Fu, J., Chan, P., and He, Y. (2022). Short-term prediction of the intensity and track of tropical cyclone via ConvLSTM model. *Journal of Wind Engineering and Industrial Aerodynamics*, 226:105026.

Tuo, R. (2019). Adjustments to computer models via projected kernel calibration. *SIAM/ASA Journal on Uncertainty Quantification*, 7(2):553–578.

Uhlenbeck, G. E. and Ornstein, L. S. (1930). On the theory of the Brownian motion. *Physical Review*, 36(5):823.

Van de Kerkhof, B., Jones, M., and Randell, D. (2024). pyELQ: Python Emission Localization and Quantification.

Varon, D. J., Jacob, D. J., McKeever, J., Jervis, D., Durak, B. O., Xia, Y., and Huang, Y. (2018). Quantifying methane point sources from fine-scale satellite observations of atmospheric methane plumes. *Atmospheric Measurement Techniques*, 11(10):5673–5686.

Veefkind, J. P., Aben, I., McMullan, K., Förster, H., De Vries, J., Otter, G., Claas, J., Eskes, H., De Haan, J., Kleipool, Q., et al. (2012). TROPOMI on the ESA Sentinel-5 Precursor: A GMES mission for global observations of the atmospheric composition for climate, air quality and ozone layer applications. *Remote Sensing of Environment*, 120:70–83.

Voss, A., Vänskä, E., Weidmann, D., Pulkkinen, A., and Seppänen, A. (2024). Multi-open-path laser dispersion spectroscopy combined with Bayesian state estimation for localizing and quantifying methane emissions. *Atmospheric Environment: X*, page 100260.

Wang, Y., Chen, B., Zhu, Z., Wang, R., Chen, F., Zhao, Y., and Zhang, L. (2020). A hybrid strategy on combining different optimization algorithms for hazardous gas source term estimation in field cases. *Process Safety and Environmental Protection*, 138:27–38.

Weiss, K., Khoshgoftaar, T. M., and Wang, D. (2016). A survey of transfer learning. *Journal of Big Data*, 3:1–40.

Weller, H. G., Tabor, G., Jasak, H., and Fureby, C. (1998). A tensorial approach to computational continuum mechanics using object-oriented techniques. *Computers in Physics*, 12(6):620–631.

West, M. and Harrison, J. (2006). *Bayesian Forecasting and Dynamic Models*. Springer Science & Business Media.

Western, L. M., Ramsden, A. E., Ganesan, A. L., Boesch, H., Parker, R. J., Scarpelli, T. R., Tunnicliffe, R. L., and Rigby, M. (2021). Estimates of north African methane emissions from 2010 to 2017 using GOSAT observations. *Environmental Science & Technology Letters*, 8(8):626–632.

Wong, R. K., Storlie, C. B., and Lee, T. C. (2017). A frequentist approach to computer model calibration. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 79(2):635–648.

Xifara, T., Sherlock, C., Livingstone, S., Byrne, S., and Girolami, M. (2014). Langevin diffusions and the Metropolis-adjusted Langevin algorithm. *Statistics & Probability Letters*, 91:14–19.

Yang, J., Zhang, T., Zhang, J., Lin, X., Wang, H., and Feng, T. (2024). A ConvLSTM nearshore water level prediction model with integrated attention mechanism. *Frontiers in Marine Science*, 11:1470320.

Yokota, T., Yoshida, Y., Eguchi, N., Ota, Y., Tanaka, T., Watanabe, H., and Maksyutov, S. (2009). Global concentrations of CO_2 and CH_4 retrieved from GOSAT: First preliminary results. *Sola*, 5:160–163.

Zou, Y., Zhao, X., and Chen, Q. (2018). Comparison of STAR-CCM+ and ANSYS Fluent for simulating indoor airflows. In *Building Simulation*, volume 11, pages 165–174. Springer.