University of Reading
School of Mathematics and Statistics

Machine Learning, Emulation and Bayesian Dimension Reduction for Climate Change Projection

Laura Anne Mansfield

Submitted in part fulfilment of the requirements for the degree of
Doctor of Philosophy in Mathematics of Planet Earth
of the University of Reading, June 2021

Declaration of Originality

I confirm that this is my own work and the use of all material from other sources has been properly and fully acknowledged.

Laura Anne Mansfield.

Abstract

Climate change projection under different greenhouse gas and aerosol emission scenarios is crucial for informing societal adaptation and mitigation measures. This traditionally relies on computationally expensive global climate models (GCMs) run on decadal to centennial timescales. One of the goals of this thesis is in exploring machine learning models and emulators trained on the output of global climate models, that can assist in this endeavour by providing rapid estimations of the climate response. Two statistical models are developed, one of which emulates the global short-term climate response to an emissions perturbation and one which learns the mapping from the short-term climate response to the long-term climate response. Different perspectives are taken so that the short-term response is predicted with a probabilistic emulator which interpolates between known and unknown data points, while the global patterns of long-term response are predicted with machine learning methods. Both models are shown to accelerate climate change projections and also provide new insights into the main drivers of climate change through sensitivity analysis to different emission perturbations and by uncovering consistent early indicators of long-term climate response.

Discovering structures in climate data that can explain patterns and behaviour is another focus of this thesis, addressed through a dimension reduction technique to simplify large datasets. This is approached from a Bayesian perspective which could allow a complete quantification of uncertainty when making predictions through an emulator trained on a reduced dataset. Reversible jump Markov chain Monte Carlo and Sequential Monte Carlo algorithms are developed for a latent factor model to infer the probability distribution on both the number of underlying dimensions and the structure of these. Sequential Monte Carlo is found to be significantly more effective at determining these and is demonstrated on weather observations to reveal underlying factors governing the weather behaviour.

Acknowledgements

I would like to express my gratitude to all of my supervisors who have been so supportive over the last four years. I am thankful to Apostolos Voulgarakis who initiated the project idea and for all his insight and advice on atmospheric physics, and to Richard Everitt for his extensive supervision on the Bayesian computation side of this project. Although not officially a supervisor, I would like to thank Peer Nowack for his insight in both machine learning and atmospheric science and for introducing me to many of the concepts in this thesis. I am also grateful to Sarah Dance for taking me on as a student half-way through the PhD, and for all the support she has provided over this period. My thanks also go out to Ed Ryan and Oliver Wild for all the useful discussions about emulator design and analysis that are crucial to the second results chapter.

I am deeply grateful to the Met Office computing team for the access and help with the Monsoon computing system that was used in this project. I would also like to give a special thank you to the Mathematics of Planet Earth CDT who gave me the opportunity to work on this PhD and have provided so much support along the way, including the additional training, conferences and social activities. I am incredibly grateful to have undertaken two unforgettable summer placements at the Met Office and the Department for Work and Pensions.

A special thank you to all my colleagues both from Reading and Imperial who have also become friends during this time. I also appreciate everyone from the Space and Atmospheric Physics Department at Imperial who have welcomed me and kept me involved in activities.

Finally, I would like to thank my family who have always been so supportive, all my friends who have kept me sane and most of all my partner Robbie without whom this work would not have been possible in normal times, let alone during a pandemic.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Climate change is one of the most pressing issues of our time given its potential to impact everyone on the Earth. The emissions released today can shape our future climate which has major implications for society as a whole. This makes climate change projection an essential task. Predicting the climate response to emissions can influence policy decisions, potentially leading to a different world, in terms of both climate and human socioeconomics. A wide range of state-of-the-art methods are constantly being explored and employed for climate change projection. This thesis delves into the potential of Bayesian statistics and machine learning to assist with this endeavour.

Climate projection has always benefited from a wide array of statistical methods, in both data analysis, where the goal is to gain information from a dataset, and climate modelling, where the state of the climate is simulated through time under different conditions (Collins et al., 2013a). The choice of statistical methods is constantly growing with increased computing power and recently, the term 'machine learning' has become highly integrated into this set of tools. This includes supervised learning, which can be applied to learn information about relationships between variables and to make predictions from these learned relationships, and unsupervised learning, which aims to uncover common factors or patterns in a dataset.

The main focus of this thesis will be on the use of statistical methods for climate change projection, specifically to predict the surface temperature response to different emission scenarios. When the overall climate response is of interest, long integrations of complex global climate models (GCMs) are typically required to A) reach an approximate equilibrium state and B) average over the effects of internal climate variability. This is highly expensive due to the complexity of GCMs, providing motivation for surrogate models, which can quickly estimate the response of a complex model to a perturbed input. While this type of prediction will not provide the same level of detail or accuracy as a GCM, it can provide almost instantaneous estimations, making it a useful tool for immediate policy decision or to as a complement to GCMs in more detailed policy studies. Developing surrogate models for this use is one of the key goals of this thesis.

Although climate change projection is concerned mainly with determining relationships between the emissions and the climate response, crucial to this is understanding the relationships between different aspects of the climate. Patterns in the climate response often reflect this and hence there is value in seeking out underlying features that govern the behaviour of the climate. Determining these features is therefore another focus of this thesis. This will be approached by modelling the data with a latent factor model, where each factor describes an aspect of the data's behaviour. A Bayesian computation perspective will be taken to develop Monte Carlo simulation methods for inference on the probability distribution on the latent factor model, including on the number of underlying factors. Understanding these factors is not only useful from a scientific point of view, but can also aid prediction, by simplifying the variables involved in prediction. Before outlining the main goals of this work, the introduction will provide some background to the topic of climate change projection and the statistical tools used throughout, including machine learning, emulation, dimension reduction, Bayesian statistics and Monte Carlo.

1.1 Climate Response

1.1.1 Climate Concepts

First, we will outline some basic climate terminology and concepts that will be used throughout this thesis. The **climate** can generally be thought of as the long-term statistics of the weather and is usually defined by the mean and variability of relevant quantities, such as surface temperature, precipitation and wind. Typically, these averages are taken over 30 years, a standard defined by the World Meteorological Organization (Cubasch et al., 2019). The climate system includes the atmosphere, hydrosphere, cryosphere, land surface and biosphere (WMO, 1975). Climate change refers to a change in this state that can be identified and that persists for an extended period, typically decades or longer.

The Earth system is powered by energy from incoming solar shortwave radiation (SWR), of which some is absorbed by the surface, some is absorbed by the atmosphere and the remaining is reflected back to space (Figure 1.1, Cubasch et al. (2019)). Most of the SWR meets the Earth in the tropics and subtropics, but atmospheric and oceanic transport processes act to redistribute this energy across the Earth to middle and high latitudes. The Earth's surface also emits energy in the form of longwave radiation (LWR). This is mostly absorbed by greenhouse gases and clouds, which themselves emit LWR in all directions. The downward component of this contributes to heating the lower atmosphere, called the troposphere, in the **greenhouse effect**, while the upward component leaves the top of the atmosphere as outgoing longwave radiation (OLR).

At the top of the atmosphere, the energy from incoming solar radiation can be compared with the energy from outgoing radiation (OLR plus reflected SWR). This is called the Earth's energy budget. Over long timescales, the incoming and outgoing radiation balance each other out to reach an equilibrium state, but a sudden change in the Earth system can lead to an imbalance between these (Murphy et al., 2009). This leads us to the concept of **radiative forcing** (Hansen et al., 1997).

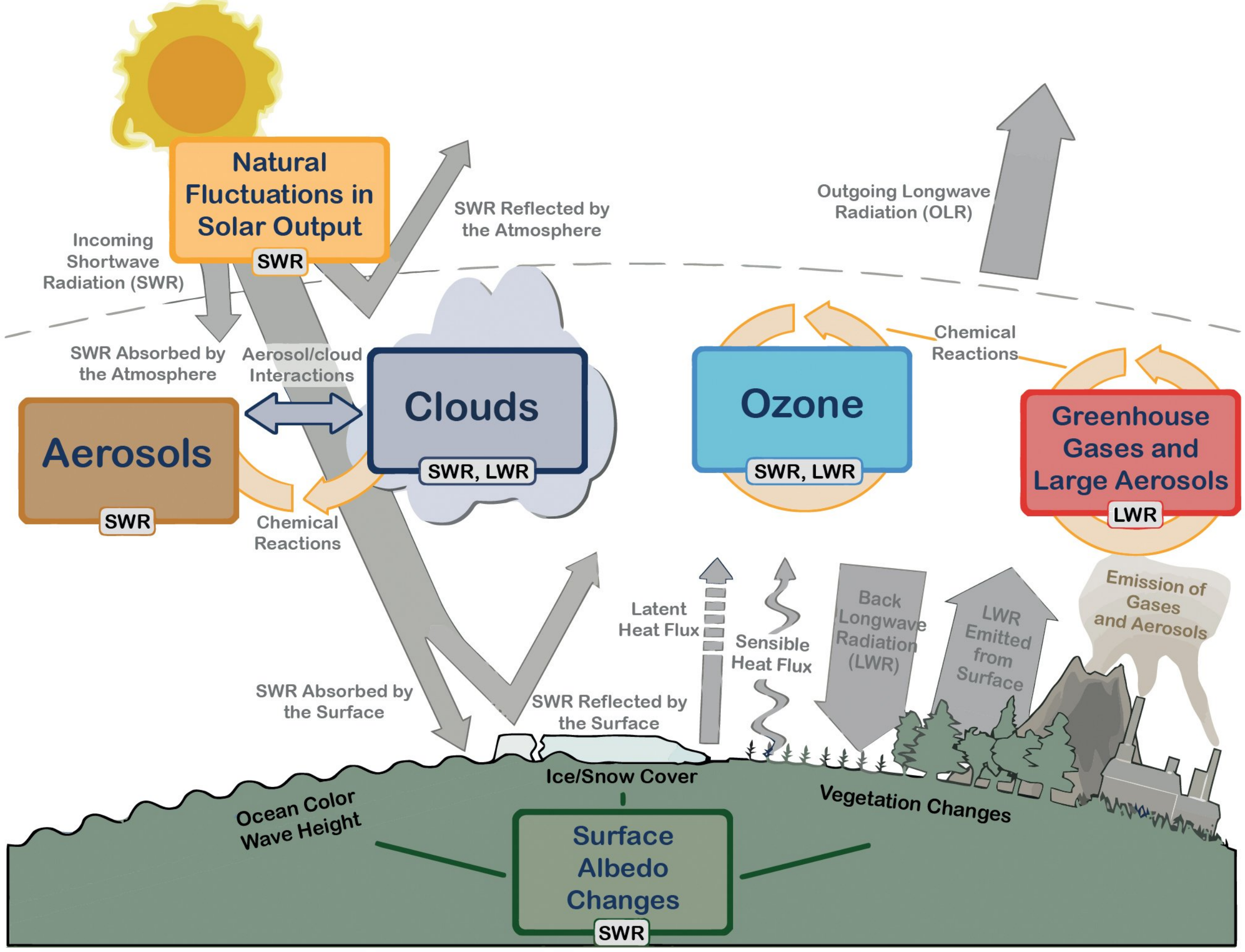

Natural Fluctuations in Solar Output
SWR
Incoming Shortwave Radiation (SWR)
SWR Reflected by the Atmosphere
Outgoing Longwave Radiation (OLR)
SWR Absorbed by the Atmosphere
Aerosol/cloud Interactions
Aerosols
SWR
Clouds
SWR, LWR
Chemical Reactions
Ozone
SWR, LWR
Chemical Reactions
Greenhouse Gases and Large Aerosols
LWR
Emission of Gases and Aerosols
Latent Heat Flux
Sensible Heat Flux
Back Longwave Radiation (LWR)
LWR Emitted from Surface
SWR Absorbed by the Surface
SWR Reflected by the Surface
Ice/Snow Cover
Ocean Color Wave Height
Vegetation Changes
Surface Albedo Changes
SWR

Figure 1.1: Reproduced from Cubasch et al. (2019) showing the Earth's energy budget and the main drivers of climate change.

We define the **instantaneous radiative forcing** as the change in net downward energy flux incident on the Earth at the top of the atmosphere. This can be in response to a natural perturbation, such as fluctuations in the solar output, or to an anthropogenic (human induced) perturbation, such as the emission of radiatively active gases or aerosols. We will be mostly concerned with the anthropogenic influences, as we are interested in understanding future climate change under various scenarios defined by societal changes. This definition, however, does not account for any interactions that occur between perturbed quantities and atmospheric properties, for example, how aerosols change the atmospheric conditions for clouds to form (semi-direct effect, Section 1.1.2) (Gregory et al., 2004; Shine et al., 2003) or fast stratospheric temperature adjustments to changes in atmospheric carbon dioxide concentrations (Smith et al., 2018b). We call these interactions **rapid adjustments** to the atmosphere. To account for this, we define **effective radiative forcing** (ERF) as the change in net downward flux at the top of the atmosphere, after allowing for any rapid adjustments without any large-scale changes of surface temperature. This is usually calculated by running a climate model given a perturbation and allowing rapid atmospheric adjustments to occur while fixing sea surface temperature and ice cover constant (Myhre et al., 2013). This allows us to better quantify the total radiative effect of a perturbation and tends to be a better indicator of temperature response.

As well as being concerned with external forcings, we will also refer to internal processes that can affect the climate. **Climate feedbacks** occur when a change in one quantity affects another, which in turn leads to an additional change in the original quantity (Intergovernmental Panel on Climate Change (IPCC), 2014). A feedback can be positive, in which the initial perturbation is enhanced, or negative, in which the initial perturbation is weakened. These are particularly relevant because they can amplify or diminish the effect of a forcing.

1.1.2 Types of Anthropogenic Forcings

Anthropogenic perturbations are human-induced forcings that affect the Earth's radiative imbalance and can therefore modify the Earth's climate. These perturbations include surface changes such as changes to vegetation and surface albedo (the fraction of light reflected from the surface) and the release of pollutants (Figure 1.1). Globally, most of the anthropogenic forcing is attributed to greenhouse gases, mainly in the form of carbon dioxide which was first noticed to be accumulating in the atmosphere by Keeling (1961). However, there are other relevant pollutants, such as other gases and aerosols, that also have a substantial effect on the radiative imbalance (Myhre et al., 2013). This section will outline some of the key pollutants and their properties.

Pollutant Properties

A pollutant can be a gas or an aerosol (a liquid or solid particle) and can alter the climate in different ways, such as causing a surface warming or cooling, affecting cloud properties or changing precipitation levels. The pollutant can have varying degrees of local and remote response, ranging from primarily impacting regions close to the emission region compared to causing a global climate impact. Furthermore, the pollutant can have a short or a long lifetime, which will determine the duration of the pollutant's effect on the climate and can affect their ability to move across the Earth and cause changes to the climate further afield from the region of emission. We will categorise pollutants into short- and long-lived pollutants, where short-lived pollutants have lifetimes $\ll 50$ days and long-lived pollutants have lifetimes beyond this that allow them to disperse throughout the troposphere (Heicklen, 1982).

The most common long-lived pollutants are well-mixed greenhouse gases, such as carbon dioxide (CO_2) and methane (CH_4). These pollutants remain in the atmosphere for long periods of time after emission (centuries to thousands of years for CO_2 (Archer and Brovkin, 2008) and decades for CH_4 (Dentener et al., 2013)). This time period is greater than the timescale for atmospheric mixing (a few years (Myhre et al., 2013)), and therefore they are able to disperse throughout the troposphere and become well-mixed over the globe. This means they have a fairly spatially homogeneous effect on the Earth's climate.

In contrast, short-lived pollutants, also known as near-term climate forcers, have short lifetimes meaning they do not accumulate in the atmosphere on decadal to centennial time scales (Myhre et al., 2013). These can be gases or aerosols. Examples include the gas ozone (O_3), the sulfate aerosol (SO_4) and black or organic carbon aerosol (BC/OC). Short-lived gases tend to be highly reactive with other atmospheric gases and therefore quickly undergo chemical processes, removing the pollutant from the atmosphere. Aerosols typically survive in the atmosphere for a few days or weeks because they are removed from the atmosphere through dry deposition (as they stick to the surface at the rate of turbulent diffusion), wet deposition (through precipitation) or re-evaporation. These pollutants are not transported far during their short lifetime which means their main effects take place predominantly in the near term and close to the region of emission.

Greenhouse Gases

Well-mixed greenhouse gases (GHG) like CO_2 are the most high profile of anthropogenic forcings. The term ‘well-mixed’ is used as they are sufficiently mixed throughout the troposphere due to their long lifetime (although local and hemispheric variations may still exist due to sources and sinks). Increases in GHGs since pre-industrial times has given rise to a positive radiative forcing by enhancing the greenhouse effect (Myhre et al., 2013). The most important well-mixed GHGs that contribute to the anthropogenic forcing from 1750-2013 are CO_2, CH_4, nitrous oxide (N_2O) and dichlorodifluoromethane (CFC-12) (Myhre et al., 2013).

GHGs absorb longwave radiation emitted by the Earth’s surface and emit longwave radiation in all directions. Some of this longwave radiation is emitted in the downwards direction, which causes further warming of the atmosphere below and the surface. This is the greenhouse effect. As well as directly affecting the Earth’s temperature, GHGs also affect other properties of the atmosphere such as hydrological cycle, which is affected in two ways on differing timescales. Firstly, the presence of GHGs in the atmosphere decreases precipitation rates as atmospheric stability is increased. This is a fast response that occurs before a change in surface temperature is realised (Dong et al., 2009; Andrews et al., 2010). Secondly, the surface temperature warming caused by GHGs leads to additional surface evaporation, since the Clausius-Clapeyron relationship predicts an increase in water vapour saturation pressure as temperature increases(Held and Soden, 2006). This adds additional water vapour (another greenhouse gas) to the atmosphere, which further enhances the greenhouse effect.

Aerosols

The next largest contributors to the anthropogenic forcing after GHGs are aerosols (Myhre et al., 2013). Aerosols are solid or liquid particles suspended in the air, typically 0.001μm-10μm in diameter (Haywood and Boucher, 2000). Aerosol particles can be emitted into the atmosphere directly, but they also often form from gaseous precursors. For instance, sulfate aerosols (SO_4) are usually formed from the oxidation of sulfur dioxide (SO_2), of which the main anthropogenic

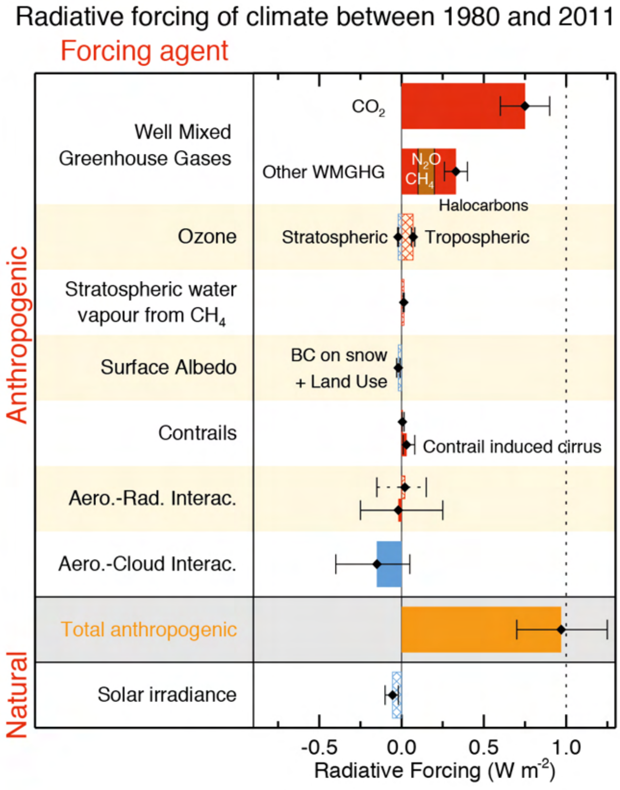


Figure 1.2: Reproduced from Myhre et al. (2013) showing breakdown of radiative forcing in 1980-2011 and total anthropogenic forcing.

source is burning fossil fuels.

Aerosols can have a wide range of effects on the climate. Firstly, there are aerosol-radiation interactions shown on the left-hand panel of Figure 1.3. Aerosols can directly interact with radiation, by scattering or absorbing SWR and emitting LWR. This is often called the direct effect (Boucher et al., 2013). By heating or cooling the air directly, aerosols change the conditions for clouds. For example, black carbon absorbs SWR which heats the atmosphere and decreases the relative humidity. This can evaporate clouds in the region and stabilise the atmosphere to prevent new clouds forming. A decrease in cloud cover leads to a decrease in scattered SWR and therefore has a warming effect. This can be viewed as an adjustment to the aerosol-radiation interactions and is sometimes called the semi-direct effect (Johnson et al., 2004).

Aerosol particles can also affect the climate through the aerosol-cloud interaction shown in Figure 1.3, also known as indirect effects (Haywood and Boucher, 2000). These interactions differ from the semi-direct effects as they involve changing cloud microphysics. Aerosol particles act as cloud condensation nuclei (CCN) which means that water vapour condenses on them to form cloud droplets. Areas rich in aerosols have more available CCN for the formation of clouds. This creates clouds with more water droplets of a smaller size, making them more reflective with an increased surface area. This enhances their cooling effect and is known as the cloud albedo effect. It also causes further adjustments, such as modifying the cloud lifetime. As precipitation occurs only when the water droplets are large enough, a cloud with smaller water droplets has an increased lifetime before precipitating.

The net effect of aerosol particles is therefore highly complex and often a result of a trade-off between different effects. Furthermore, we often see different effects depending on the region of emission and the time of day, season or year. Because of this, there are still large uncertainties in the estimates of aerosol effective radiative forcings, as demonstrated by the large error bars in Figure 1.2 (Boucher et al., 2013). Most aerosols tend to create a negative radiative forcing (and surface cooling) due to the direct radiative effect, i.e. because the aerosol particles reflect SWR back to space. Sulfate aerosols are one of the main anthropogenic contributers to this. The negative radiative forcing is partially offset by the absorbing properties of aerosols such as black carbon, which contributes a smaller but substantial positive radiative forcing. Overall, however, the net effect of anthropogenic aerosol emission is a negative radiative forcing, as seen in Figure 1.2, and thus a surface cooling.

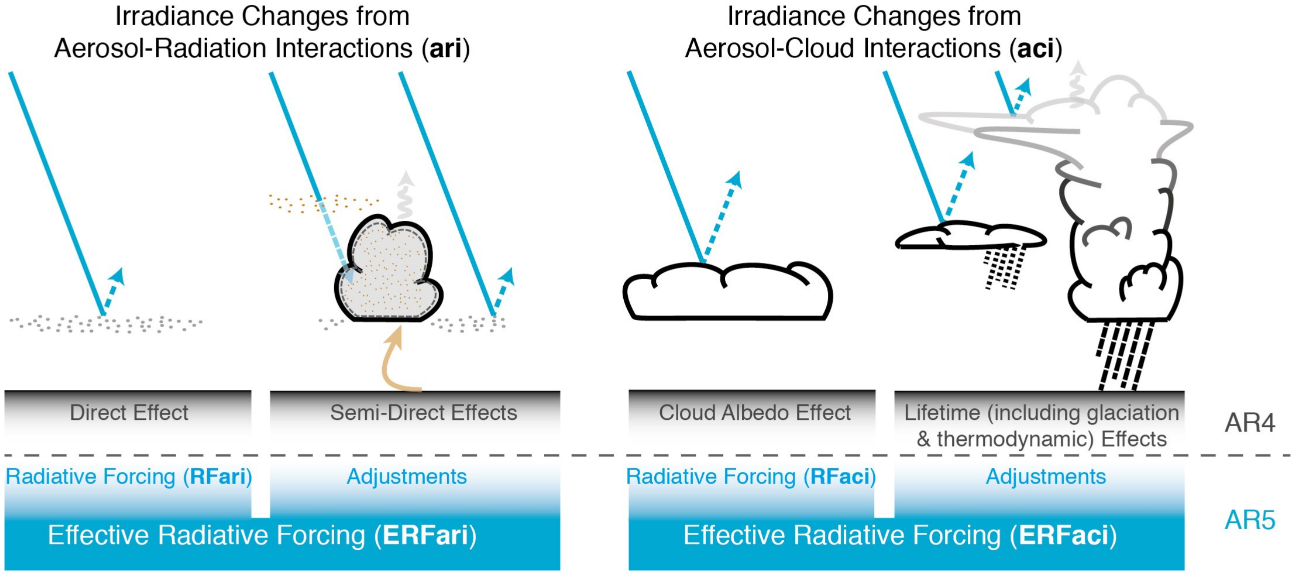
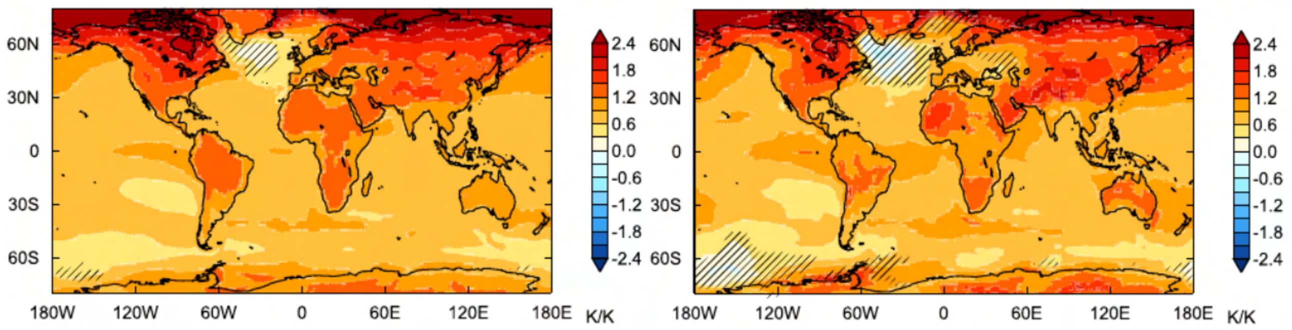

Irradiance Changes from
Aerosol-Radiation Interactions (ari)
Irradiance Changes from
Aerosol-Cloud Interactions (aci)
Direct Effect
Semi-Direct Effects
Cloud Albedo Effect
Lifetime (including glaciation & thermodynamic) Effects
AR4
Radiative Forcing (RFari)
Adjustments
Radiative Forcing (RFaci)
Adjustments
Effective Radiative Forcing (ERFari)
Effective Radiative Forcing (ERFaci)
AR5

Figure 1.3: Reproduced from Boucher et al. (2013) showing the aerosol-radiation interactions and aerosol cloud interactions.

1.1.3 Global Response to Forcings

Understanding the individual and net effect of different anthropogenic forcings on the climate system has long been of interest to climate scientists. For instance, the contribution of forcing agents to the net radiative forcing in Figure 1.2 is constantly being refined as new studies come to light (Myhre et al., 2013). GCMs have provided robust methods to estimate the net radiative forcing and the associated climate response. These have led to extensive research in uncovering properties of the climate system, such as the **equilibrium climate sensitivity** which describes the equilibrium change in global mean temperature per unit forcing. It is typically assumed that the relationship between global mean forcing and global mean temperature response is linear. Based on this assumption, several studies have constrained the climate sensitivity for different climate forcings and in different climate models, in order to make robust predictions on future climate change given a set of forcings (Hansen et al., 2000, 2005; Gregory et al., 2004; Andrews et al., 2012).

Beyond this, it is not just the global mean but the complete spatial response that is of interest to scientists and policy-makers worldwide. Estimating this has become more accessible in recent years, with the increase in computational budget and complex climate models available. A wide range of model intercomparison projects have constrained the projected global response to different pollutant perturbations and scenarios, (e.g. Richardson et al., 2019; Liu et al., 2018b; Shindell et al., 2013; Taylor et al., 2012; Meehl et al., 2007). For instance, the multi-model mean surface temperature response relative to the global mean is shown in Figure 1.4 for a doubling of CO_2 and tripling of CH_4 (Richardson et al., 2019). There is a general consensus in the pattern of global warming amongst projections of GHGs, that includes enhanced warming over land relative to the oceans, with a roughly consistent ratio of 1.4-1.7x, regardless of the mean response magnitude (Collins et al., 2013a). This is thought to be predominantly due to enhanced evaporation and latent heat fluxes over the ocean, rather than sensible heat fluxes that occur over land (Sutton et al., 2007) and due to differences in lapse rate (the rate of change of temperature with decreasing height) in moist air and dry air (Joshi et al., 2008). Another common feature across GHG warming projections is strong warming over the Arctic,



(a) 2xCO2, mean response 2.44K ± 0.75 (b) 3xCH4, mean response 0.67K± 0.33K

Figure 1.4: Reproduced from Richardson et al. (2019) showing multi-model mean surface temperature response for core PDRMIP forcing experiments (years 81–100 of coupled runs), normalized by the global mean surface temperature response given in caption. Hatching shows where the multimodel mean is less than the intermodel standard deviation.

known as Arctic amplification. This occurs mainly because of two postive feedback processes, the ice-albedo feedback and the lapse rate feedback. The former is because the melting of ice darkens the surface of the Earth and reduces reflectivity, called the ice-albedo feedback. The latter occurs because of the positive lapse rate at the poles (warmer temperatures closer to the surface) which is enhanced with additional warming (Masson-Delmotte, 2012). These spatial features are not exclusive to GHG perturbations and are also present in other projections, such as aerosol removal forcings (Kasoar et al., 2018; Richardson et al., 2019; Persad et al., 2018; Xie et al., 2013).

Both GHGs and aerosols perturbations have spatially dependent responses but the short-lived nature of aerosols, leading to highly inhomogeneous responses, makes a spatial analysis even more relevant. Shindell and Faluvegi (2009) were one of the first to investigate the climate response to regional aerosol forcing, by applying aerosol perturbations in latitudinal bands in a GCM. Importantly, this study found the response to forcings to be highly dependent on the region, with enhanced responses at higher latitudes, even due to forcings in tropical regions. This is due to transport of heat from the tropics to higher latitudes, as well as positive feedbacks caused by clouds and water vapour (of which there are more at higher latitude climates) and changes to surface albedo.

More recent studies have taken this further through more localised aerosol perturbations on continental scales (e.g. Kasoar et al., 2018; Aamaas et al., 2017; Collins et al., 2013b; Myhre et al., 2013). Kasoar et al. (2018) show spatially similar patterns of response to the removal of sulfate aerosol from different regions in a climate model, as demonstrated in Figure 1.5. In particular, the boxes highlight regions of similarities over the North America extending into the North Atlantic ocean, central Russia and over the North Pacific ocean. Furthermore, these patterns are also present within leading spatial modes of variability within the long climate model simulations. This indicates the response is a projection onto these existing modes, which has been suggested in previous studies based on both simulations (Shindell et al., 1999; Ring and Plumb, 2008) and observations (Corti et al., 1999; Palmer, 1999). These consistencies between emissions, regardless of region, and response patterns provide a strong basis for learning a simplified model of the relationship between emissions and response pattern which will be approached in this thesis through machine learning and emulation methods (Chapter 3 and 4). This has implications for policy-relevant studies, as spatial response projections can be made more rapidly using these simplified models.

1.1.4 Policy Relevance

The United Nations Framework Convention on Climate Change (UNFCCC) is the part of the United Nations that is tasked with supporting the global response to the threat of climate change (UNFCCC, 2021b). It consists of 197 countries that have agreed to limit human-induced climate change. Every year since 1995, a Conference of Parties (COP) is held where all countries of the UNFCCC are represented and decisions are made. For example, at the 2015 Paris Climate Conference (COP21) an agreement was made to 'keep the global temperature rise this century well below 2 degrees Celsius above pre-industrial levels and to pursue efforts to limit the temperature increase even further to 1.5 degrees Celsius' (UNFCCC, 2021a). Decisions such as these are made based on the outcome of the assessment reports from the Intergovernmental Panel on Climate Change (IPCC), which are a combined effort of thousands of scientists globally who present the scientific basis of climate change, its impacts and future risks, and options for adaptation and mitigation (IPCC, 2021). The IPCC assess thousands of scientific research papers on this topic and present the results in an objective way, while also reflecting the degree

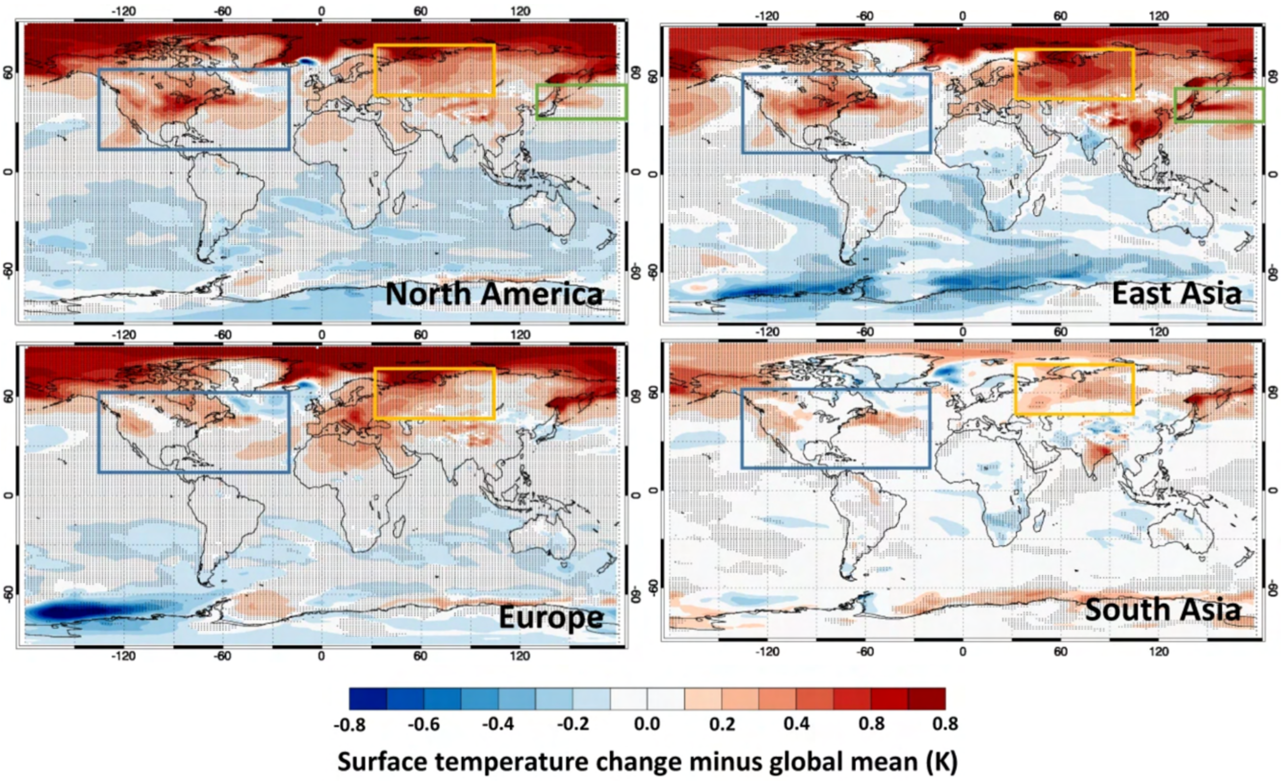


Figure 1.5: Reproduced from Kasoar et al. (2018) showing surface temperature changes minus the global mean change for 150 year annual. a–e show the geographic pattern of 150-year annual mean surface temperature change, with the global mean temperature change subtracted off in each case, due to SO2 emissions being removed from North America, c Europe, d East Asia, and e South Asia. The rectangles highlight regions with consistent regional patterns of stronger temperature change, as discussed in the text. stippling indicates that the change at that grid-point exceeded 2 standard deviations of the 150-year mean in six different control simulations.

of certainty in results.

One of the points of interest of these reports is to make projections on climate change at a global and regional level, based on various emission scenarios (or pathways) (Brock and Xepapadeas, 2019). The most recent Assessment Report (AR5) by the IPCC stated that an overall warming of the climate is unequivocal based on observed changes to the climate and that this warming will continue with certainty under future projections from large-scale models (Stocker et al., 2013). However, this is difficult to quantify precisely because we are uncertain of the path that future emissions will take, which depends on societal, technological, and economic developments. The Shared Socioeconomic Pathways (SSPs) provide different pathways that we may take with different levels of development, mitigation and adaptation strategies, which fall under the categories of 'Sustainability', 'Middle of the Road', 'Regional Rivalry', 'Inequality' and 'Fossil-fueled Development'. GCMs predict the climate response to these different pathways which are used in a large range of independent studies. For example, there were 1,378 published studies between 2014-2019 that made use of SSPs, for applications in climate impacts or adaptation, (e.g. agriculture) as well as for understanding drivers or mitigating effects of climate change (e.g. changing energy use) (O'Neill et al., 2020). These studies not only influence IPCC reports and ultimately decisions made by world leaders, but they also have a wider reach by improving our understanding of the relationships between emissions and climate response.

Predictions of climate response to different scenarios will continue to be of importance in the near future, both as part of independent studies and of the UNFCCC's efforts to limit the effects of climate change. Climate projection is an important aspect of COP meetings and for informing policy decisions. State-of-the-art global climate models are, and always will be, crucial in this role. At the same time, other approaches to quantitative climate change assessment could be beneficial in this endeavour. Currently, multi-decadal climate model simulations typically take weeks to complete which motivates the development of alternative models that can estimate climate response on the order of minutes. Therefore, we can turn to the use of machine learning and emulation for climate change projection, which have the potential to assist in rapid prediction and assessment of climate change risks under different scenarios.

Furthermore, recent and future reports have started to focus not just on global mean climate change but also on the regional pattern of climate change with, for example, the spatial distribution of societal impacts being one of the key risks for concern highlighted by AR5 (IPCC, 2013). There is a trend towards an increasing demand for climate projections at regional and local scales which are most relevant for decision-making (O'Neill et al., 2020). Estimating the climate response under different scenarios at a global level requires high resolution GCMs which are computationally expensive. This is further motivation for the development of prediction methods that can drastically reduce computational time such as emulation of the global response.

1.2 Climate Change Projection

1.2.1 Global Climate Models

The use of computer simulations to model the climate dates back to the 1960s (McGuffie and Henderson-Sellers, 2001). Advances in computer capabilities at the time enabled short-term weather forecasting to be run for longer integrations on climate timescales. This led to the development of atmospheric general circulation models (AGCMs) (e.g. Smagorinsky et al., 1965), which differ slightly to weather models since they have a global domain and are run for decadal to centennial timescales, rather than days to weeks. This also has relevance for the equations of motion that are solved, such as energy, mass and moisture conservation over long periods of time, which are not major concerns in short-term weather forecasting (McGuffie and Henderson-Sellers, 2001).

AGCMs allow the atmosphere to be simulated under different conditions, such as changes to the greenhouse gas forcing. They discretize the atmosphere into a 3D grid to represent the horizontal and vertical directions, typically labelled longitude, latitude and vertical height. On this grid, the model holds information about the state of atmosphere in the form of prognostic variables, which includes all relevant variables such as air temperature, humidity, greenhouse gas concentrations, and so on. Given some initial conditions that describe the state of the atmosphere at time $t = 0$, the equations of motion for each component are solved within each grid cell, to calculate the state of the climate at the next timestep. The equations of motion for the atmosphere include Navier-Stokes (fluid dynamics) with thermodynamic source terms (e.g. radiation and latent heating). Some physical processes occur on spatial scales smaller than the climate model resolution, such as convection. These are instead represented with simplified schemes called parameterisations (McGuffie and Henderson-Sellers, 2001).

Soon after the development of AGCMs the effects of the ocean were incorporated, by coupling the atmosphere with an oceanic component, that also solved the equations of motion for the ocean on a discretized grid (Manabe and Bryan, 1969). The inclusion of the ocean is crucial for accurate long-term climate modelling, because it acts as a large heat reservoir and responds much more slowly to climate perturbations than the atmosphere (100s-1000s of years). The components interact by exchanging relevant quantities, for example, the heat flux from the atmosphere into the ocean. In the following decades, GCMs were improved further with validation and calibration due to data from satellites and in-situ measurements (Oort and Peixóto, 1983; Boer et al., 1992). Furthermore, multi-model comparison projects began in Gates et al. (1999), where GCMs from different modelling institutes were tested under the same settings and compared.

The addition of more components, such as atmospheric chemistry, land and sea-ice, has led to coupled GCMs in the form that we see today, also known as Earth system models (ESMs). They aim to model the entire behaviour of the climate under various conditions on decadal to multi-centennial timescales. These GCMs are continuously calibrated to observations and compared under large scale projects (e.g. Lamarque et al., 2013; Taylor et al., 2012; Eyring et al., 2016; Myhre et al., 2017; Andrews et al., 2020).

GCMs provide an invaluable tool for exploring climate response for scientific and policy related purposes. However, they come at a considerable computational cost. For example, to estimate the climate response to a particular perturbation (an 'idealised' perturbation), the change in forcing is applied in an instant (e.g. a sudden doubling of CO_2 concentrations) and integrated forwards in time until an approximate equilibrium is reached (e.g. Manabe and Bryan, 1969; Hansen et al., 1997; Held et al., 2010; Kasoar et al., 2016; Liu et al., 2018b). The transient process is called 'spin-up' and often must also be performed to reach an equilibrium state suitable for starting conditions of a simulation. Furthermore, to accurately assess climate response, long integrations are required for two reasons: firstly, to capture the slow, long-term effects of mixing in the ocean which occur on the timescales of 100s years, and secondly to isolate the climate response from the effects of internal variability, which create large fluctuations in the year-to-year output. This requires us to run the model for longer time periods and to average over several decades in order to define a 'climate' response. These long-term simulations become expensive, especially because of the short timesteps required to capture small-scale climate processes under high resolutions. Fundamentally, it is the wide range in spatial scales affecting the climate system that demands the GCM to model both short and long timescales.

As well as running GCMs for long integrations, a robust analysis of climate response also often requires multiple simulations. A 'control' run with no perturbations is usually performed as a baseline to compare against the results of a perturbation run. Multiple control and perturbation runs may also be used to create an 'ensemble' of runs as another method to diagnose internal variability which arises from slightly different initial conditions of the system. Modelling climate at increasingly high spatial resolutions has significantly increased the computational complexity of GCMs (Collins et al., 2012), a tendency that has also been accelerated by the incorporation and enhancement of a number of new Earth system model components and processes (e.g. Williams et al., 2018; Walters et al., 2019; Ridley et al., 2018). All these factors contribute to the high computational cost of climate modelling, which provides a case for alternative, quicker climate projection methods. This is a topic of growing research, particularly with the recent rise in machine learning. However, quick approaches for estimating climate model responses are not entirely new. In the following section, we will look at a simplistic method that has carried out this task for several decades.

1.2.2 Pattern Scaling

Pattern scaling is a traditional method for obtaining future spatial patterns of climate change without running a full GCM. First proposed by Santer et al. (1990), it has been widely used for many years for regional climate change projections (e.g. Hulme et al., 1995; Murphy et al., 2007; Watterson, 2008), in impact studies (e.g. Huntingford and Cox, 2000), in policy focused studies (e.g. Moss et al., 2010), and to extend simplified models to predict spatial outputs (e.g. Harris et al., 2006; Castruccio et al., 2014a).

In pattern scaling, an estimate of the global mean response to a forcing is projected onto a normalised pattern from a reference scenario, such as the GCM response to a doubling of CO_2 concentration. It therefore requires one previous run of a GCM to obtain the equilibrium response of the variable of interest for a reference scenario (e.g. the 2X CO_2 scenario). This is used to find a normalised response pattern on the longitude latitude grid, $V_{\text{ref}}(\text{lat}, \text{lon})$. Then, the variable of interest can be predicted at each grid point for a new scenario, $V^*(\text{lat}, \text{lon})$ by multiplying the reference pattern by a scaler value, s, i.e.

$$V^*(\text{lat}, \text{lon}) = s\, V_{\text{ref}}(\text{lat}, \text{lon}) \tag{1.1}$$

The scaler value s can be derived from either a mathematical relationship, simplified climate model or a statistical model. To calculate the equilibrium temperature response, commonly used scalers are the annual global-mean temperature in a simple climate model (Mitchell, 2003; Castruccio et al., 2014a), or the global mean ERF from an energy-balance model (Huntingford and Cox, 2000; Zelazowski et al., 2018).

Pattern scaling gives relatively accurate results under many settings, particularly when scaling responses to well-mixed GHGs and strong forcings (Mitchell, 2003; Tebaldi and Arblaster, 2014; Ishizaki et al., 2012). However, there can be different spatial response patterns in weakly and strongly forced scenarios, due to the effect of changing temperatures in the ocean (Manabe and Wetherald, 1980), which means that often perturbation patterns can be quite different from the reference patterns. Reference patterns with strong forcings are typically preferred in order to maintain high signal-to-noise ratios (Mitchell, 2003). Pattern scaling is therefore known to be less accurate for predicting strongly mitigated scenarios with weak forcings (May, 2012). Furthermore, the response of the predicted variable to a radiative forcing is assumed to scale linearly, which is not necessarily the case, e.g. Mitchell (2003) finds a sub-linear dependency for the temperature response to radiative forcing. Although pattern scaling is an invaluable policy tool for estimating response patterns under different forcings (Moss et al., 2010; Holden and Edwards, 2010), the logical next step is to explore how well machine learning algorithms can perform on the same task.

1.2.3 Machine Learning and Emulation

In recent years, **machine learning** methods have been gaining traction in the climate science community (Reichstein et al., 2019). The term ‘machine learning’ encompasses methods that can automatically detect patterns in data, and then use this to predict future data or perform other kinds of decision making (Murphy, 2012). Searching for patterns in data has long been an endeavour by scientists, but the development of computers that can handle large quantities of data at high speed has allowed these methods to be automated. One of the fundamental goals of machine learning is to develop algorithms that learn from experience, thereby reducing the burden on humans to explicitly program detailed instructional laws (Samuel, 1959). The idea is to learn an adaptive model based on a **training dataset**. In this thesis we are mostly concerned with **supervised learning** which learns output vectors from corresponding input vectors (e.g. regression on continuous data and classification on categorical data); but other forms include unsupervised learning which learns from input vectors alone (e.g. clustering or density estimation) and reinforcement learning (learning actions to take in order to maximise a reward) (Bishop, 2006).

Machine learning algorithms have found popularity in climate science as a tool to aid impact studies (Crane-Droesch, 2018; Huntingford et al., 2019; Abbot and Marohasy, 2017); climate change detection and attribution (Sippel et al., 2019; Wills et al., 2020b); to uncover patterns in data (Toms et al., 2019; Runge et al., 2019; Kretschmer et al., 2016; Thomas et al., 2021) which can then be used for prediction (Kretschmer et al., 2017; Nowack et al., 2020); and to expose relationships in processes within climate models through interpretable machine learning methods (Barnes et al., 2020; Toms et al., 2020; Kuhn-Régnier et al., 2020). Machine learning is becoming of interest to climate modelling groups (Scher, 2018; Yadav et al., 2020; Weber et al.,

2020) and has already proven to be a reliable approach for developing parameterisations within GCMs (Nowack et al., 2018; Gentine et al., 2018; Rasp et al., 2018; Bolton and Zanna, 2019; O'Gorman and Dwyer, 2018). Furthermore, rather than viewing a machine learning method as a 'black-box', research into physics-informed machine learning which satisfy certain physical constraints (such as momentum conservation) has also gained traction in recent years (Raissi et al., 2019; Wu et al., 2018; Dueben and Bauer, 2018; Lutter et al., 2019; Wang et al., 2019; Kashinath et al., 2021). One of the commonly noted challenges in the application of machine learning to climate science is that typical learning algorithms require large quantities of data ('big data'), particularly deep learning such as deep neural networks, while climate data is often small in terms of number of samples available ('small data') (Karpatne and Kumar, 2017; Watson-Parris, 2021; Yadav et al., 2020). This is a challenge encountered in Chapter 3, due to the high cost of obtaining GCM simulations.

The theme of this thesis is predicting climate response via both machine learning regression and **emulation**. 'Emulation' refers to the use of an **emulator**, which is a statistical approximation to an expensive computer model that can make predictions much more rapidly (O'Hagan, 2006). It is also often called 'metamodelling' or 'surrogate modelling' (Ratto et al., 2012). More strictly, an emulator differs from simply approximating an expensive model, as it provides a probability distribution of the predicted output, rather than just a single point estimate. This is particularly convenient for understanding the uncertainty associated with predictions. An emulator is most often used to refer to a Gaussian process emulator, which we will use in Chapter 4.

Because of the increases in speed and the uncertainty provided, the main uses of emulators are in **calibration** (Kennedy and O'Hagan, 2001; Wilkinson, 2010), **sensitivity analysis** (Saltelli et al., 2008), **uncertainty quantification** (O'Hagan et al., 1999; Oakley and O'Hagan, 2002) and **prediction** (also called model reduction) (O'Hagan, 1978; Currin et al., 1991; Ratto et al., 2012). In these uses, emulators are trained on a carefully selected set of simulation data that aims to give the best possible results for predicting new unseen data. This is called **emulator design** (O'Hagan, 1978; Currin et al., 1991; Santner et al., 2003). The training data can be selected to be space-filling over the parameter ranges with a **Latin hypercube sampler** (LHS). Emulator studies typically use a 'maximin' LHS that maximises the minimum distance between new samples, or in other words, maximises how well the parameter space is filled, given a set number of training samples. Given a limited number of samples, the input-output relationships can be learnt across more of the parameter space compared to that learnt from randomly sampled training data (O'Hagan, 2006; Santner et al., 2003; McKay et al., 1979).

In climate science, emulators are already a popular tool for calibration or tuning of climate model output to agree with observational data, in order to reduce any biases in the model (e.g. Williamson et al., 2015; Salter and Williamson, 2016; McNeall et al., 2019; Goldstein and Rougier, 2006; McNeall et al., 2013). They also have widespread use in sensitivity analysis, to assess how sensitive certain climate variables are to different relevant inputs (e.g. Wild et al., 2020; Ryan et al., 2018; Bounceur et al., 2015; Lee et al., 2012; Rougier et al., 2009; Marrel et al., 2011). Taking this a step further, climate model emulators are also designed for uncertainty quantification, which aims to identify the greatest sources of uncertainty in the model output (e.g. Carslaw et al., 2013; Lee et al., 2013, 2016; Edwards et al., 2019).

However, the application of emulation for predicting climate response to forcings is still fairly uncommon. Tran et al. (2016) have built an emulator for prediction of a steady-state atmosphere based on various climate input parameters and configurations, one of which is the CO_2 forcing. Holden and Edwards (2010), Miftakhova et al. (2020), Castruccio et al. (2014a), Bao et al. (2016) and Foley et al. (2016) also develop emulators of GCMs for predicting the global mean temperature response to transient global CO_2 forcings and MacMartin and Kravitz (2016) extend this to emulate the climate under different solar forcing as well as CO_2 scenarios. However, there is only one study so far that we are aware of that emulates climate response to other anthropogenic forcings, such as aerosols, through the FAIR model (Smith et al., 2018a). As with the previous climate model emulators, FAIR predicts the global mean temperature response to transient emissions but does not explore the regional pattern of response, which is highly important in policy-related studies. Building an emulator that predicts the entire pattern of temperature response to a range of anthropogenic forcing scenarios is one of the main outcomes of this thesis and is the focus of Chapter 4.

1.2.4 Dimension Reduction

When building a climate model emulator, it is difficult to ignore the size of the dataset. Although climate model data is usually small in terms of the number of samples available, it is often large in terms of the number of features such as the number of variables or outputs (this could be labelled ‘wide data’). This is particularly common if the data is spatial or temporal, with climate model data typically being both. Unfortunately this makes building machine learning models or emulators more complicated. Gaussian process emulators are often built independently for each output, which ignores any correlations in the output and can lead to higher training costs. Dimension reduction methods such as principal component analysis is therefore frequently used in climate model emulation, particularly for spatial predictions (Salter and Williamson, 2016; Holden and Edwards, 2010; Tran et al., 2016; Ryan et al., 2018).

Principal component analysis (PCA) is a linear statistical method that assumes that many of the variables in the dataset are correlated and that the relationships between them can be used to reduce its dimensionality while retaining as much variation as possible (Jolliffe, 2002). This involves transforming the data to a new set of uncorrelated, orthogonal variables, called **principal components**. These are ordered so that first few contain most of the variation in the data and a cut-off can be selected to discard principal components that contribute little variation to the data set, thereby reducing the dimensionality of the dataset. PCA is usually viewed as traditional statistical method and was first described in this form in Pearson (1901) and Hotelling (1933). However, it is in fact a form of unsupervised learning as it relies only on input variables without any information about a target variable and is now commonly used by the machine learning and emulation communities (Bishop, 2006; Holden and Edwards, 2010; Lawrence, 2004). Although we focus on PCA as an approach to reduce spatial data, it also frequently applied to climate timeseries to identify dominant patterns of temporal variability in the data, known as Empirical Orthogonal Functions (EOFs) (Hannachi et al., 2006).

An alternative approach to dimension reduction is **factor analysis**, which models the dataset as a linear combination of common structures, called **latent factors** (Joliffe and Morgan, 1992) and some residual unique structures. It is assumed that these latent factors cannot be measured directly, but that they explain a large portion of the behaviour of the data. The unique components contribute to any remaining variations in each measured variable.

Unlike principal components, the latent factors are not necessarily orthogonal but rather, they can take any form specified by choice. The latent factors can often be interpreted as relevant drivers governing the behaviour of a dataset. For instance, one factor behind a spatial temperature dataset could be latitude. Factor analysis can ultimately be viewed as deconstructing the dataset into a few physically meaningful factors, rather than simply transforming the data into a set of uncorrelated variables, as done in PCA. Factor analysis is often used in applications for finance (Geweke and Zhou, 1996; Aguilar and West, 2000; Lopes and West, 2004), psychology (Fabrigar et al., 1999; Comrey, 1978; Conti et al., 2014; Fava and Velicer, 1992; Cui and Dunson, 2014) and epidemiology (Ghosh and Dunson, 2009; Chen et al., 2010; Bhattacharya and Dunson, 2011). Although not commonly used in climate science currently, it may prove to be a useful tool in revealing hidden structures in spatial climate data alongside other discovery algorithms (e.g. Runge et al. (2019); Nowack et al. (2020); Kuhn-Régnier et al. (2020); Thomas et al. (2021)).

1.3 Bayesian Statistics

1.3.1 Bayesian Viewpoint

One of the focal points of this work takes a Bayesian approach to uncover the underlying structure of a dataset, which has an interesting application for high dimensional climate data, such as high resolution GCM output. A Bayesian analysis is particularly relevant because it provides a complete probabilistic analysis, with a quantification of uncertainty, rather than just a point-based estimate. In this section, we will outline the Bayesian viewpoint, which will be taken throughout Chapter 5.

In the Bayesian philosophy, we can view probability as a 'degree of belief', rather than just relative frequencies of events (the frequentist viewpoint) (Cox, 1946). This allows us to make probability statements, and even set a probability distribution on parameters which are, in reality, fixed constants. With this, we can carry out **Bayesian inference**, meaning we can infer what the probability distribution is, using a combination of data and our prior degree of belief. Bayesian inference on parameter θ, involves first defining a **prior probability distribution** $p(\theta)$ that expresses our initial belief about this parameter, before observing any data (Wasserman, 2004). We must also define a statistical model, called the **likelihood**, that describes the probability of making an observation, y, given parameter θ. We write the likelihood as $f(y|\theta)$. After observing data y we update our beliefs and calculate the **posterior probability distribution**, $p(\theta|y)$. This can be done via **Bayes theorem** (Bayes and Price, 1763),

$$p(\theta|y) = \frac{f(y|\theta)p(\theta)}{\int f(y|\theta)p(\theta)d\theta} \tag{1.2}$$

where the denominator, $\int f(y|\theta)p(\theta)d\theta$ is the **marginal likelihood**, **normalising constant** or **evidence** which is required to ensure that the posterior probability distribution sums to 1. This term describes the likelihood function of the data, once all parameter values θ have been integrated out.

The Bayesian philosophy allows us to discuss probabilities of outcomes that only occur once (e.g. the probability of rain falling at a specific point in time and space) and by describing parameters as probability distributions, we can quantify uncertainties on them. The use of Bayesian statistics is extensive in climate science and appears in climate model ensemble analysis (Smith et al., 2009; Murphy et al., 2007), impact studies (Meinshausen et al., 2009), climate change detection (Leroy, 1998), remote sensing (Gorte and Stein, 1998) and data assimilation studies (Reich and Cotter, 2015), as well as being the basis for Gaussian process emulators as discussed above. We will take a Bayesian viewpoint with regards to the factor analysis problem presented in Chapter 5, as we will carry out Bayesian inference on the number of factors and the factors themselves. This includes carrying out model selection, as we compare different factor models for a dataset, each with a different number of underlying factors. Model selection often relies on the calculating the marginal likelihood for a given model (often called the **model evidence**) which can be difficult to calculate when θ is high dimensional. This is another topic also encountered in climate and weather science (e.g. Carrassi et al., 2017; Aanonsen et al., 2019). In the following section, we introduce the simulation based tool, Monte Carlo, which is typically used in Bayesian methods and is fundamental to the problem presented in Chapter 5.

1.3.2 Monte Carlo

Monte Carlo is the name given to the simulation of random processes. They are commonly used to evaluate integrals and to find and sample from a distribution of interest. Halton (1970) defines the Monte Carlo method, in general, as “representing the solution of a problem as a parameter of a hypothetical population, and using a random sequence of numbers to construct a sample of the population, from which statistical estimates of the parameter can be obtained”. It was the invention and development of computers that first allowed for the simulation of random processes at a large scale (Brooks et al., 2011). These ideas were set in motion in the 1940s at the Los Alamos National Laboratory, New Mexico, during research into the random physical process within the atomic bomb (Eckhardt, 1987). Stanislaw Ulam, a mathematician and physicist, first recognised that neutron diffusion can be viewed as a succession of random operations and that fast computers could be used to repeat random experiments and evaluate many possible outcomes. He compared these random operations to a game of solitaire, leading to the initially secret name ‘Monte Carlo’ after the world-famous casino in Monaco. The ability to simulate many possible outcomes provides an overall picture of a physical system, which has made these methods popular in the field of physics ever since.

One particular approach that we will encounter in Chapter 5, is **Markov chain Monte Carlo** (MCMC), which simulates a system via a chain of samples that follows the overall behaviour of the physical system, according to its probability distribution. MCMC methods were born shortly after the invention of Monte Carlo, again at Los Alamos National Laboratory which had one of the fastest computers of the time. They were developed based on the realisation that learning about thermodynamic equilibrium of a system did not require simulating the exact dynamics, but rather simulating a Markov chain with the same equilibrium distribution. This was first carried out by Metropolis et al. (1953) through the simulation of a liquid in equilibrium with its gas phase, using what is now called the **Metropolis algorithm**. This was followed by a more general version, called the **Metropolis-Hastings algorithm** (Hastings, 1970).

MCMC methods quickly became ubiquitous in many areas of chemistry and physics, driven by the increase in availability of computers, but it was not until the 1990s when they became popular amongst statisticians (Brooks et al., 2011). As computers have become more accessible and increased in power, more sophisticated Monte Carlo algorithms have been developed. One example of this is **Sequential Monte Carlo** (SMC) (or **particle filters**), which were developed in the 1990s to estimate the evolution of dynamical systems by simulating a sequence of probability distributions (Del Moral, 1997; Liu and Chen, 1998).

The complete assessment of probabilities provided by Monte Carlo methods explains their growing uses in climate science, in terms of both modelling and analysis. They are also an invaluable tool for optimisation and therefore frequently used in the background of machine learning algorithms, as well as taking the foreground in their probabilistic counterparts which frame such algorithms in terms of probability distributions rather than point-estimates (Murphy, 2012). In Chapter 5, we will see how various Monte Carlo methods can be applied to factor analysis, to reveal the underlying structure of a dataset consisting of weather observations made at different spatial locations.

1.4 Aims of Thesis

This thesis has two overarching aims: firstly, to develop a surrogate model to predict climate response at a lower computational cost than running a full GCM, and secondly, to explore how Bayesian statistics can uncover the underlying structure of a high dimensional dataset, that can be applied to climate data.

The motivation for the former is the high cost of running complex GCMs to predict climate response to different emission perturbations, including global perturbations of long-lived GHGs (CO_2, CH_4) and regional perturbations short-lived aerosols (SO_2, OC, BC). For climate timescales, we would typically run a GCM with a sudden perturbation on the order of 100 years. These timescales will be termed ‘long-term’ throughout the thesis. However, even obtaining training data to build a surrogate model is expensive, with each 100 year simulation on HadGEM3, developed by the UK Met Office (Walters et al., 2019) costing on the order of 2000 node hours for a 100 year simulations, which can take several weeks to months to run on the Met Office resources available for research (e.g. 4 nodes of 64 cores on Cray XC40 Met Office). Typically, to build a probabilistic emulator that interpolates between known data, the estimated number of samples required is around 10x the number of inputs (Loeppky et al., 2009). To cover a range of key climate forcers and including regional perturbations for the short-lived aerosols, one may aim to build an emulator with around 10 inputs (discussed in Chapter 4). This would require around 100 training simulations and cost up to 10 years of computing time on the today's Met Office research nodes. To reduce this problem, the surrogate model could be approached in two stages.

In the first step, I propose an emulator designed to predict the intermediate stage of the climate response rather than the full long-term response, to a range of emission perturbations. I will refer to this intermediate response as the 'short-term response'. This can be obtained by running the GCM for a shorter time period of, for instance, 5 or 10 years, for which the order of 100 training simulations is feasible in a few months. Then, the second step is to build another surrogate model for predicting the long-term response based on the short-term response, trained on a smaller but more expensive dataset. For this step, I make use of existing simulations run for previous studies, where there are 21 data points available.

This is similar but not identical to a procedure carried out in previous emulation studies, where the Bayesian approach is used to update the results of the first emulator given the second dataset, which is typically obtained with a more expensive model of, for instance, higher resolution (Tran et al., 2016; Cumming and Goldstein, 2009). This is often called 'multi-level' or 'multi-fidelity' emulation. In these studies, the first dataset is assumed to be an approximation to the second, which is where the approach derived here differs slightly. Rather than assuming that the short-term climate response is an **approximation** to the long-term climate response, I present a methodology that **learns a mapping** from short-term to long-term response. This is proposed based on the idea that the short-term climate response contains indicators of the long-term climate response, even if statistical rather than causal (Ceppi et al., 2017; Persad et al., 2018).

The two surrogate models will be treated separately, as highlighted by the schematic in Figure 1.6, and furthermore they will be approached with two differing perspectives. While the first surrogate model uses a typical emulation approach involving probabilistic predictions, the second takes a machine learning viewpoint of exploring different methods and their accuracy, based on point estimates. Both surrogate models are valuable as standalone tools, carrying out two different tasks: (1) for rapid short-term climate change projection based on emission scenarios, and (2) for mapping the short-term to the long-term climate response. The latter will be learned based on existing GCM data in Chapter 3, while the former will be trained on newly designed simulations in Chapter 4.

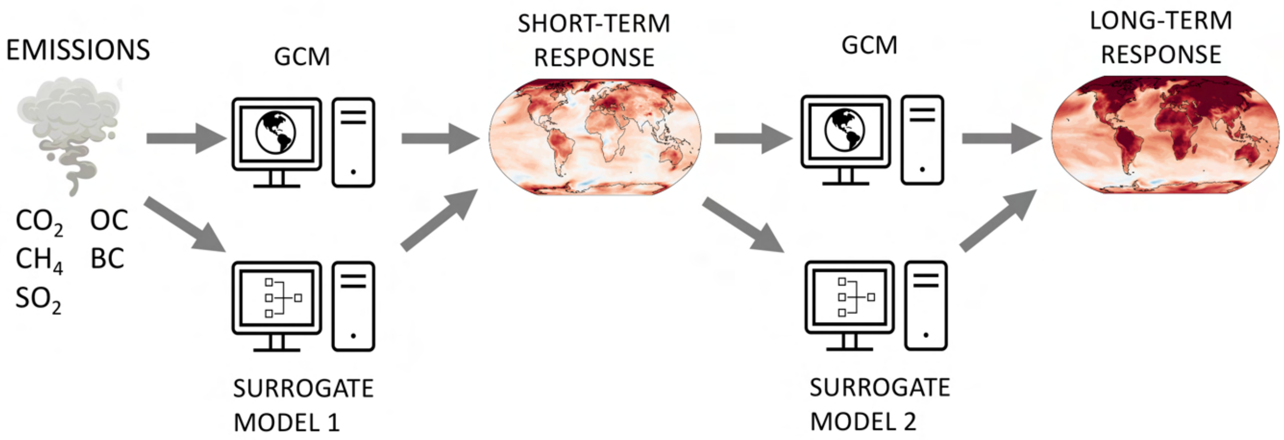


Figure 1.6: The two separate surrogate models explored in this thesis. The first learns the short-term climate response based on emission perturbations, while the second learns the long-term climate response based on the short-term climate response.

One of the characteristics of the climate data encountered in both surrogate modelling tasks is its high dimensionality. This brings us to the remaining task of the thesis, which explores a Bayesian approach to learn the underlying dimension of a dataset, using a factor analysis model. This will involve Monte Carlo simulations to infer the probability distribution on, not only the factors, but also the number of factors required to accurately describe the data. This is motivated by the common use of dimension reduction techniques in emulation studies, to reduce the dimension of the input and/or output space and to enforce correlations in the output (Higdon et al., 2008; Ryan et al., 2018; Lawrence, 2005). The application of these methods introduces additional uncertainty in the emulator, but this is not accounted for in these studies. A fully Bayesian approach to dimension reduction would allow the uncertainty in the reduced model to be quantified, which can be propagated into the emulator to give a more accurate picture of the uncertainty associated with its predictions. Even on its own, however, dimension reduction can be a valuable tool in reducing a dataset to a smaller set of variables that can often be interpreted to improve understanding of the data. For example, applying dimension reduction to spatial data may reveal underlying behaviour of teleconnections between remote regions, while applying such a method to a temporal dataset may highlight reccuring oscillations on different timescales (Hannachi, 2004). Chapter 5 will approach dimension reduction with factor analysis and will focus predominantly on the development of Monte Carlo methods that can be used to infer the probability distribution on both the underlying factors and the number of them. This will be demonstrated on a dataset of weather observations to highlight how the underlying factors can be interpreted meaninfully.

1.4.1 Thesis Outline

In Chapter 2, the methods of the thesis will be described, including: machine learning and emulation relevant to the first two results chapters, the dimension reduction methods used throughout the thesis and the Monte Carlo methods utilised in the final results chapter. Then the subsequent chapters are as follows:

Chapter 3: Long-term Climate Response Prediction with Machine Learning

This chapter is concerned with the task of mapping the long-term climate response given the short-term climate response of a GCM, i.e. the second surrogate model in Figure 1.6. Existing simulations from a range of previous long-term climate studies will be used to train and test the model, which will be developed from a machine learning perspective as multiple different methods are explored. This work has been published in Mansfield et al. (2020).

Chapter 4: Short-term Climate Response Prediction with an Emulator

The first surrogate model in Figure 1.6 is developed in this chapter to estimate the short-term climate response given a range of emission perturbations. This focuses on a probabilistic prediction, including both the mean function and standard deviation of the estimate. The Gaussian process emulator is carefully designed for a selection of emission-based input variables, including global greenhouse gas concentrations and regional aerosol scaling factors. This is followed by a sensitivity analysis, to explore how sensitive the short-term response is to the different emission perturbations. An example of how this emulator could be used as a policy tool is also demonstrated.

Chapter 5: A Bayesian Approach for Dimension Reduction

The final chapter of the thesis revolves around reducing a high dimensional dataset to a lower dimensional representation, through factor analysis. This is approached in a fully Bayesian way, with a focus on the different Monte Carlo methods that can be applied to infer the probability distribution on both the number of factors and the factors themselves. For this task, I will develop and compare reversible jump MCMC and Sequential Monte Carlo algorithms that allow transitions between different factor models. Finally, this chapter will conclude with an application of these methods in the context of climate and weather science.

Chapter 2

Methods

This chapter will cover the techniques used throughout this thesis, beginning with statistical methods that will be used to estimate the climate response based on training simulations from GCMs in Chapters 3 and 4. Then I will introduce methods of dimension reduction, namely principal component analysis and factor analysis. The remainder of this chapter will focus on Monte Carlo methods, including Markov chain Monte Carlo (MCMC), Reversible Jump MCMC and Sequential Monte Carlo (SMC) which build the basis of Chapter 5, where the goal is to learn the underlying structure of a high dimensional dataset through factor analysis.

2.1 Machine Learning and Emulation

2.1.1 Regression

Regression is a supervised learning technique used to determine the relationship between dependent and independent variables (Wasserman, 2004). It is commonly used to analyse and understand characteristics of the relationship between variables as well as in prediction to new unseen situations.

We start in the simple case that the dependent variable, y, is univariate. The aim of regression is to model y from independent variable $\mathbf{x}$ which we allow to be multivariate of size p so that

$\mathbf{x} = (x_1, \cdots, x_p)$. We assume that y is a function of $\mathbf{x}$ with some additional residual error ϵ,

$$y = f(\mathbf{x}) + \epsilon \,. \tag{2.1}$$

Usually ϵ is assumed to be normally distributed with zero mean and constant variance σ^2. The independent variables $\mathbf{x}$ are often referred to as **inputs**, **covariates**, **predictors** or **features** and the dependent variables y as **outputs** or **response variables** (Wasserman, 2004).

In this section, we have a dataset of input-output pairs, $(\mathbf{x}, y)_i$, of size N i.e. $i = 1, 2, \cdots, N$. This dataset is split into a **training set** and a **test set**. The training set is used to learn the function $f(\mathbf{x})$ and the test set is only introduced at the end of training, to assess how well the model generalises to new unseen data (Hastie et al., 2001).

2.1.2 Linear Regression

When f is assumed to be linear, we write

$$f(\mathbf{x}) = \beta_0 + \sum_{j=1}^{p} \beta_j x_j \tag{2.2}$$

where j indexes the independent variables x_j for $j \in 1, \cdots, p$. We wish to estimate the parameters, β_0 (the intercept) and β_j (the slope associated with variable x_j).

These unknown parameters can be estimated from the training data. For a single observation, indexed by i, we have

$$y_i = \beta_0 + \sum_{j=1}^{p} \beta_j x_{ij} + \epsilon_i \,. \tag{2.3}$$

We denote estimated parameters $\hat{\beta}_0$ and $\hat{\beta}_j$ for $j = 1, \cdots, p$, which give **predicted** or **fitted** values of y, denoted $\hat{y}$

$$\hat{y}_i = \hat{\beta}_0 + \sum_{j=1}^{p} \hat{\beta}_j x_{ij} \,. \tag{2.4}$$

The difference between the observed data y_i and the predicted value $\hat{y}_i$ gives the residual error $\hat{\epsilon}_i$

$$\hat{\epsilon}_i = |y_i - \hat{y}_i| = \left| y_i - \Big(\hat{\beta}_0 + \sum_{j=1}^{p} \hat{\beta}_j x_{ij}\Big) \right| . \tag{2.5}$$

This is used to estimate the **training error**, which is minimised in the training process. The same equation can be applied for the test data, to calculate the **test error** on data not seen in the training process.

2.1.3 Least Squares Regression

There are a total of $p + 1$ parameters in Equation (2.2) to estimate, denoted as $\hat{\beta}_0$ and $\hat{\beta}_j$ for $j = 1, \cdots, p$. If the number of training data points is greater than or equal to the number of variables to learn ($N > p$), we can estimate $(\hat{\beta}_0, \hat{\beta}_j)$ with **least squares regression** by minimising the **residual sum of squares**, $RSS = \sum_{i=1}^{N} \epsilon_i^2$ (Hastie et al., 2001), i.e.

$$\arg\min_{\beta} \left\{ \sum_{i=1}^{N} \Big(y_i - (\beta_0 + \sum_{j=1}^{p} \beta_j x_{ij})\Big)^2 \right\} . \tag{2.6}$$

We can interpret the values of $\hat{\beta}_j$ as how much each covariate x_j contributes to the output response variable and the intercept $\hat{\beta}_0$ as a baseline when all covariates are set $= 0$.

Once we have estimated $(\hat{\beta}_0, \hat{\beta}_j)$, we often wish to use our regression model for prediction. Suppose we want to predict the output y_* for a given input with new unseen value $\mathbf{x}_* = (x_{*1}, \cdots, x_{*p})$. This output can be estimated with

$$\hat{y}_* = \hat{\beta}_0 + \sum_{j=1}^{p} \hat{\beta}_j x_{*j} . \tag{2.7}$$

Matrix formulation

It is typical to write this in matrix form (Hastie et al., 2001). We can include the intercept term β_0 in the vector containing β_j by including additional index $j = 0$ by fixing $x_0 = 1$. This gives vector β, of length $p+1$ and covariate matrix $\mathbf{X}$, of size $(N \times p+1)$, where each row is one observation and each column is one covariate:

$$\beta = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{pmatrix} \quad \text{and} \quad \mathbf{X} = \begin{pmatrix} 1 & x_{11} & x_{12} & \dots & x_{1p} \\ 1 & x_{21} & x_{22} & \dots & x_{2p} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{N1} & x_{N2} & \dots & x_{Np} \end{pmatrix}. \tag{2.8}$$

We denote the observed outputs and the residuals both in a vector of size N

$$\mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix} \quad \text{and} \quad \epsilon = \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_N \end{pmatrix}. \tag{2.9}$$

In this notation, Equation (2.3) becomes

$$\mathbf{y} = \beta\mathbf{X} + \epsilon \tag{2.10}$$

and Equation (2.6) is

$$\arg\min_{\beta} \Big\{ (\mathbf{y} - \beta\mathbf{X})^T (\mathbf{y} - \beta\mathbf{X}) \Big\}. \tag{2.11}$$

This is minimised by differentiating with respect to β and setting this to zero, giving the following least squares estimate

$$\hat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}. \tag{2.12}$$

An unbiased estimator of the variance of the residuals, σ^2, is given by

$$\hat{\sigma}^2 = \Big(\frac{1}{N-p}\Big) \sum_{i-1}^{N} \hat{\epsilon}_i{}^2. \tag{2.13}$$

Finally, we can make predictions of outputs at new input values $\mathbf{X}_*$ with

$$\hat{\mathbf{y}}_* = \hat{\beta}\mathbf{X}_* + \epsilon\,. \tag{2.14}$$

2.1.4 Ridge Regression

Equation (2.6) equal to zero can be solved exactly when the number of degrees of freedom in the parameters exactly equals the number of constraints from the data available. This is when $N = p$. Then $\hat{y} = y$ and the variance $\hat{\sigma}^2 = 0$. However, this case is rare and even if so, it does not necessarily mean each data point uniquely corresponds to knowledge of a single parameter (i.e. the solution may be unique, but we would not know each parameter exactly and would still favour increasing the size of the data in order to refine our knowledge of the parameters).

In the case that $N > p$, the β parameters are **over-determined**: there are more data constraints than there are parameters. We can minimise the RSS (equation 2.2), making this an optimisation problem in β, given the available data constraints.

In contrast, when $N < p$ (sometimes called supercollinearity) there are more free parameters (β) than there are observed data points to constrain these parameters meaning there are many possible β values that satisfy Equation (2.6) equal to zero. The matrix $\mathbf{X}^T\mathbf{X}$ in Equation (2.12) becomes singular and so we cannot solve for $\hat{\beta}$ uniquely. This is an ill-posed problem because it is **under-determined**.

We can introduce additional constraints to make this problem unique and solvable. In ridge and LASSO regression, the β coefficients are constrained to be small using a method called **regularisation** (Bishop, 2006). This is done by introducing a penalty term that penalises β coefficients with large magnitudes.

First, we introduce **ridge** or **Tikhonov** regression in which Equation (2.6) becomes

$$\arg\min_{\beta}\left\{\sum_{i=1}^{N}\Big(y_i - (\beta_{0i} + \sum_{j=1}^{p} x_{ij}\beta_j)\Big)^2 + \lambda\sum_{j=1}^{p}\beta_j^2\right\} \tag{2.15}$$

where the term highlighted in red is the **regularisation** or **penalisation** term, with regularisation parameter, λ (Hoerl and Kennard, 1970). This term penalises large magnitudes of β^2 and favours lower magnitudes, often called 'shrinkage'. A larger value of λ gives increased regularisation, meaning the magnitude of β coefficients will shrink more. The choice of λ is discussed later in Section 2.1.6.

The penalisation term in Equation (2.15) imposes the constraint that $\sum_{j=1}^{p} \beta_j^2 < C$ for a constant C, which is proportional to λ. The solution to minimising Equation (2.15) with respect to β is in closed-form giving

$$\hat{\beta} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y}\,. \tag{2.16}$$

Regularisation is used to reduce the problem of **overfitting**, which occurs when the model fits too well for the training data, but cannot be generalised to new datasets. This is a topic encountered frequently in machine learning, both when $N < p$ and $N \geq p$ (Hastie et al., 2001).

The least squares estimates when $\hat{\beta}$ from minimising Equation (2.11) would then give a low bias but a high variance. By adding the penalisation term, a bias is introduced to $\hat{\beta}$ but the variance is reduced, so the model is less dependent on the training dataset. This often improves prediction accuracy for new data outside the training dataset. This is known as the bias-variance trade-off (Tibshirani, 1996).

2.1.5 LASSO Regression

An alternative to ridge regression is LASSO regression (or Least Absolute Shrinkage and Selection Operator), which involves regularisation by constraining $|\beta|$, rather than $|\beta|^2$, i.e.

$$\arg\min_{\beta} \left\{ \sum_{i=1}^{N} \Big(y_i - (\beta_0 + \sum_{j=1}^{p} x_{ij}\beta_j)\Big)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\} \tag{2.17}$$

One of the key differences between ridge regression and LASSO regression is that ridge regression shrinks many β terms close to zero, while LASSO regression shrinks β terms exactly to zero.

This makes LASSO regression a variable selection method, suitable when the y is dependent only on some of the predictors x. The predictors for which $\beta_j = 0$ can be removed from the model completely, improving the user interpretability of the model (Tibshirani, 1996). In comparison, ridge regression is more suitable in cases when the predictors x all make small contributions to the output y, such as when the predictors are highly correlated.

2.1.6 Cross Validation

In the training process of ridge and LASSO regression, we minimise Equations (2.15) or (2.17) to find estimates of the coefficients, β. However, we have not yet determined the regularisation parameter, λ, which controls the amount of shrinkage. For this we could use a process called **validation** which involves setting aside a **validation set** that is not used in training, but is instead used to estimate how well the trained model will perform on new data (Hastie et al., 2001). At this stage, choices are made on any additional parameters, that are not optimised when minimising the residual error, such as the regularisation parameter λ. This means the data is then split into three sets: the training, validation and test set. A typical split suggested by Hastie et al. (2001) would be 50% for training and 25% each for validation and testing. However, keeping aside such a relatively large validation set can only be done when there is sufficient availability of data.

One way around this is to carry out **cross validation** (Stone, 1974). In cross validation, multiple rounds of training and validation are carried out, using one subset of the training data as the training set and a different part as the validation set, repeating this multiple times. One of the more commonly used forms is k-fold cross validation (Hastie et al., 2001). The training set is split into K subsets at random, called 'folds'. We iterate through $k = 1, \cdots, K$ and for each fold, training is carried out using all samples except those in fold k. The trained model is then used to predict the samples in fold k and prediction errors are estimated. Then, the mean prediction error can be calculated across all folds. This can be repeated for different settings, such as different values of the regularisation parameter λ. Finally, the particular setting or value of λ is selected so that the mean prediction error across folds (or other choice of loss function) is minimised.

2.1.7 Gaussian Processes

Gaussian processes provide another way to learn the relationships between input variables $\mathbf{x}$ and output variables $\mathbf{y}$. A Gaussian process is a **stochastic process**, which can be viewed as an extension of a probability distribution. While a probability distribution describes random variables, a stochastic process describes functions. A Gaussian process is therefore a generalisation of the Gaussian probability distribution (Rasmussen and Williams, 2006).

Mathematically, we define a Gaussian process as a collection of random variables, $f = \{f(x) : x \in \mathcal{X}\}$, such that all finite sets of $f(x)$ have a joint multivariate normal distribution,

$$(f(x_1), f(x_2) \cdots, f(x_n)) \sim \mathcal{N}_n(\mu, \Sigma) \tag{2.18}$$

In other words, the relationship between the function value at x_1 and at x_2 can be described by a Gaussian distribution. This is the case for all possible combinations of x. Gaussian distributions are extensively used in statistical problems because of the central limit theorem, which makes the Gaussian distribution a natural choice for many data rich scenarios, and because of their unique properties that make calculations tractable and simple. These benefits carry through to Gaussian processes, making them a computationally tractable and versatile to a wide range of problems.

A Gaussian process is defined by its mean function, $m(\mathbf{x})$, and covariance function, $k(\mathbf{x}, \mathbf{x}')$, specified by the user:

$$f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')) \,. \tag{2.19}$$

The mean function is usually chosen to be zero, $m(\mathbf{x}) = 0$, for simplicity and because the Gaussian process can usually model the mean behaviour from the data and covariance function alone (Murphy, 2012; Rasmussen, 2004). The covariance function, or 'kernel', describes the covariance between pairs of random variables, i.e.

$$k(\mathbf{x}, \mathbf{x}') = \operatorname{cov}\left(f(\mathbf{x}), f(\mathbf{x}')\right) \tag{2.20}$$

This choice, before introducing any data, defines the prior Gaussian process. We can generate samples from this prior, such as those plotted in Figure 2.1a, for an example of a Gaussian process with a univariate input x and univariate output $f(x)$. These samples reflect properties of the chosen covariance functions and any hyperparameters already set, such as lengthscales over which things vary.

We want to learn the posterior Gaussian process, which is done by observing datapoints. We label the dataset, $\{\mathbf{x}, \mathbf{f}\}$. To make predictions of the Gaussian process at new datapoints, labelled $\{\mathbf{x}^*, \mathbf{f}^*\}$, we use the definition that the relation between $\mathbf{f}$ and $\mathbf{f}^*$ must follow a joint multivariate normal distribution. This gives

$$\begin{pmatrix} \mathbf{f} \\ \mathbf{f}^* \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \begin{pmatrix} k(\mathbf{x}, \mathbf{x}) & k(\mathbf{x}, \mathbf{x}^*) \\ k(\mathbf{x}^*, \mathbf{x}) & k(\mathbf{x}^*, \mathbf{x}^*) \end{pmatrix} \right) \tag{2.21}$$

where the notation $k(\mathbf{x}, \mathbf{x}^*)$ is used to mean the matrix of covariances of all pairs of training and test points (of size $N \times N^*$ if there are N training points and N^* test points). The properties of conjugacy in the Gaussian distribution make conditioning on the observations tractable and leads to a posterior Gaussian process,

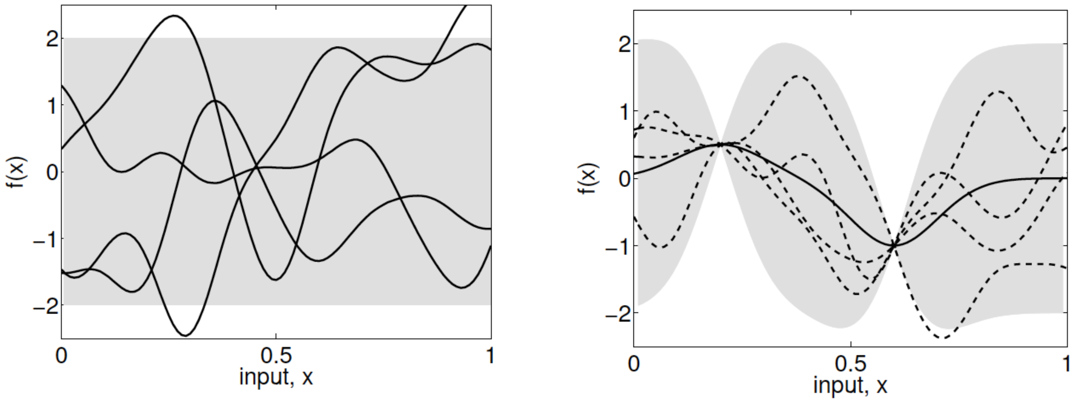
$$f(\mathbf{x}) \sim \mathcal{GP}(m^*(\mathbf{x}), k^*(\mathbf{x}, \mathbf{x}')) \tag{2.22}$$

with mean and covariance functions given by m^* and k^*,

$$\begin{aligned} m^* =& m(\mathbf{x}^*) + k(\mathbf{x}^*, \mathbf{x}) k(\mathbf{x}, \mathbf{x})^{-1} (\mathbf{x} - m(\mathbf{x})) \\ k^* =& k(\mathbf{x}^*, \mathbf{x}^*) - k(\mathbf{x}^*, \mathbf{x}) k(\mathbf{x}, \mathbf{x})^{-1} k(\mathbf{x}, \mathbf{x}^*) \end{aligned} \tag{2.23}$$

This proof involves conditioning on Gaussians distributions (see e.g. Murphy (2012), Section

4.3). We can then sample from this at new unseen inputs $\mathbf{x}^*$. The dotted lines in Figure 2.1b shows different samples from the posterior Gaussian process. The solid line shows the new mean function and the grey shading shows the standard deviation or Gaussian process uncertainty as a consequence of the covariance function. At the training datapoints, $\mathbf{x}$, the value of the output $\mathbf{f}$ is known exactly and therefore the standard deviation shrinks to zero here.



(a) Prior

(b) Posterior

Figure 2.1: Reproduced from Rasmussen and Williams (2006) showing 4 samples from (a) the prior and (b) the posterior after 2 points have been observed. Shading shows 2 standard deviations.

One of the appealing properties of a Gaussian process is that $f(\mathbf{x})$ can take any form allowed by its covariance function and it is not limited to a functional form (e.g. linear, sinusoidal). It is therefore a non-parametric method, as we do not need to learn any parameters (with the exception of any hyperparameters of the covariance function that can be learned). The covariance function almost entirely determines how the Gaussian process behaves. If two points x_1 and x_2 are deemed close or similar by the covariance function, the resulting $f(x_1)$ and $f(x_2)$ is also deemed similar. Therefore the choice of covariance function is important. There are some choices that generally work well for many datasets. For example, for a smoothly varying output, we often use the squared exponential covariance function (also called the Gaussian or radial basis function (RBF)),

$$k(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \exp\left(\frac{|\mathbf{x} - \mathbf{x}'|^2}{2\mathbf{l}^2}\right) \tag{2.24}$$

where $\mathbf{l}$ is a horizontal length scale over which the function varies (and may take different values for different dimensions of $\mathbf{x}$) and σ_f controls the vertical variation. These hyperparameters can be specified by the user before building the Gaussian process, but in many situations we do not know what the best hyperparameters are. Instead, we can find the optimal hyperparameters through cross validation or maximum likelihood estimation (MLE). To use MLE, we maximise the likelihood,

$$\begin{aligned} \log p(\mathbf{x}|\mathbf{y}) &= \log \int p(\mathbf{x}|\mathbf{f},\mathbf{y})p(\mathbf{f}|\mathbf{y})d\mathbf{f} \\ &= -\frac{1}{2}\mathbf{y}^T\mathbf{K}^{-1}\mathbf{y} - \frac{1}{2}\log|\mathbf{K}| - \frac{n}{2}\log 2\pi \end{aligned} \tag{2.25}$$

with respect to the hyperparameters l and σ_f, where $\mathbf{K} = k(\mathbf{x}, \mathbf{x})$ i.e. the covariance function evaluated at the training data points (Rasmussen and Williams, 2006).

The covariance function can be used to incorporate relevant knowledge about the system, for example, periodicity can be represented through a periodic covariance function while linear relationships can be represented through a linear covariance function. The covariance function can also be constructed as linear combinations of multiple covariance functions, to represent many properties. For example, in Chapters 3 and 4, we will make use of a covariance function constructed from adding together the effects of a linear and a squared exponential covariance function. Finally, we can also construct a Gaussian process to allow a fixed noise component, such as measurement error. This can be viewed as including a fixed white noise covariance function. As an outcome, the uncertainty associated with any prediction includes this noise or measurement error. This means the shaded area showing the uncertainty at the two data points in Figure 2.1b would not reduce to zero exactly, but would instead reduce to the measurement error. Furthermore, introducing a fixed noise term, σ_n, also changes Equation (2.25) as $\mathbf{K} = k(\mathbf{x}, \mathbf{x}) + \sigma_n^2\mathbf{I}$ (Rasmussen and Williams, 2006). We will see an example of this in Chapter 4, where the internal variability of the system contributes to the total variance on any prediction.

2.2 Dimension reduction

2.2.1 Principal Component Analysis

In dimension reduction, our goal is to project a data set $\mathbf{x}$ of size m to a smaller space with dimension $k < m$. Principal component analysis (PCA) does this by finding the orthogonal projection where the variance is maximised (Hotelling, 1933).

We define a new vector $\mathbf{u}$ of size k, which we call the 'principal components' (Bishop, 2006). The first principal component, $\mathbf{u}_1$, is a vector that points in the direction of the data that varies the most, such as $\mathbf{u}_1$ in Figure 2.2. The second principal component must be orthogonal to u_1 and is the direction that contains most of the variance subject to this constraint.

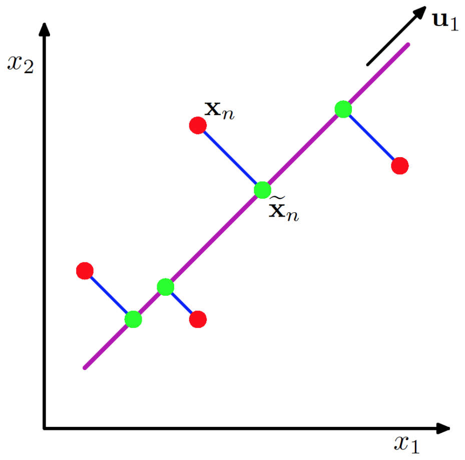


Figure 2.2: Reproduced from Bishop (2006) showing data points (red) projected onto the first principal component (green). The principal axis $\mathbf{u}_1$ contains most of the variance in the data.

To carry out PCA, we project the data onto the first principal component, $\mathbf{u}_1^T\mathbf{x}$. The variance of this projected data is

$$\frac{1}{N}\sum_{n=1}^{N}\left(\mathbf{u}_1^T\mathbf{x}_n - \mathbf{u}_1^T\bar{\mathbf{x}}\right)^2 = \mathbf{u}_1^T\mathbf{S}\mathbf{u}_1 \tag{2.26}$$

where $\bar{\mathbf{x}}$ is the mean of the original data and $\mathbf{S}$ is the covariance matrix of the original data

$$\mathbf{S} = \frac{1}{N}\sum_{n=1}^{N}\left(\mathbf{x}_n - \bar{\mathbf{x}}\right)\left(\mathbf{x}_n - \bar{\mathbf{x}}\right)^T . \tag{2.27}$$

The projected variance, $\mathbf{u}_1^T\mathbf{S}\mathbf{u}_1$, is maximised with respect to $\mathbf{u}_1$. We must also constrain $\mathbf{u}_1^T\mathbf{u}_1$

to prevent $||\mathbf{u}_1|| \to \infty$ which is included in the maximisation with a Lagrange multiplier, λ_1,

$$\mathbf{u}_1^T \mathbf{S} \mathbf{u}_1 + \lambda_1 (1 - \mathbf{u}_1^T \mathbf{u}_1)\,. \tag{2.28}$$

This is maximised when

$$\mathbf{S}\mathbf{u}_1 = \lambda_1 \mathbf{u}_1 \tag{2.29}$$

meaning that the first principal component is the eigenvector with the largest eigenvalue λ_1. The next principal components are derived in the same way. PCA can alternatively be viewed as the projection that minimises the distance between the data points and their projections (Pearson, 1901) and this can be used to write the problem in terms of a singular value decomposition of $\mathbf{x}$ (Hastie et al., 2001).

PCA is used as a dimension reduction method by taking only the first few principal components which contain most of the variance of the data. This is done by sorting eigenvalues from largest to smallest and choosing a sensible 'cut-off', for example, taking the principal components associated with the eigenvalues that explain 95% of the variance. This leaves us with a smaller dataset on which analysis can be carried out and the remaining principal components are discarded. At the end of analysis, the principal components can be projected back into original space. As some variables are discarded, the full dataset will not be recovered entirely, meaning this is a lossy method.

2.2.2 Factor Analysis

Factor analysis is a latent variable method that views the data as a linear function of **latent factors** (also called **latent scores** (Murphy, 2012) or **common factors** (Fabrigar et al., 1999)) which have influence over multiple variables. These are denoted $\boldsymbol{\eta}$ and it is assumed that each variable is a linear combination of these latent factors, with an additional residual variance, ϵ. This residual variance is unique to each variable and is typically assumed to have distribution $\epsilon \sim \mathcal{N}(0, \Sigma^2)$, where the zero mean assumes there is no systematic unique factor and $\Sigma^2 = \text{diag}(\sigma_1^2, \cdots, \sigma_m^2)$ is an error of measurement component, specific to each of the m

variables (Fabrigar et al., 1999; Tipping and Bishop, 1999). The data, $\mathbf{y}$, can be written as

$$\mathbf{y} = \boldsymbol{\eta}\boldsymbol{\Lambda}^T + \epsilon\,. \tag{2.30}$$

where $\boldsymbol{\Lambda}$ is the **factor loading matrix** which describes structure of correlations among measured variables and latent factors, of size $m \times k$ when there are k latent factors and m variables. This factor analysis model is flexible and allows the user to select the way in which the factors are defined. For instance, we could choose the factors to be orthonormal, as in PCA (Figure 2.2). The fundamental difference between factor analysis and PCA is that factor analysis differentiates between the common variances $\boldsymbol{\eta}$ and the unique variances ϵ, whereas standard PCA treats these identically[1] (Tipping and Bishop, 1999).

As an example, assume we have a dataset of temperature observations measured at different weather stations. Factor analysis allows us to break this dataset into a selection of factors that describe this dataset that could represent meaningful quantities (e.g. latitude, altitude, closeness to sea) plus a unique variance associated with each individual weather station. In contrast, PCA provides a method to break the dataset into orthogonal principal components that describe the total variance in the data, which may or may not have an interpretable meaning, and without a unique variance from individual weather stations.

The goal of factor analysis is to estimate η, ϵ and $\boldsymbol{\Lambda}$. In the biology and psychology literature (e.g. Fabrigar et al., 1999), this is typically done in a similar way to PCA, by carrying out an eigenvalue decomposition so that the factors are defined by the amount of variance described by the data. However, Chapter 5 will focus on Bayesian inference of η, ϵ and $\boldsymbol{\Lambda}$, as carried out in Lopes and West (2004) and Press and Shigemasu (1989).

Firstly, note that Equation (2.30) is non-identifiable. There are various methods to make this identifiable, such as requiring $\boldsymbol{\Lambda}$ to be orthonormal as done in PCA; using sparsity promoting methods (much like ridge and LASSO regression described previously); or forcing $\boldsymbol{\Lambda}$ to be a lower triangular matrix, as usually done in the Bayesian community (Murphy, 2012). The latter

[1] Note that the unique variance is included in 'Probabalistic PCA' which is factor analysis with $\epsilon \sim \mathcal{N}(0, \sigma^2\mathbf{I})$, i.e. all latent variables have the same unique variance.

means that the first measured variable is only generated by the first latent factor, the second by the first two latent factors, and so on. This can make the latent factors more interpretable, although it means that the first k variables can affect the interpretation of the latent factors and should therefore be chosen carefully. However, in order to estimate the factors, we generally need to know the number of factors in advance. The number of factors can have a substantial effect on the result and several studies have found that specifying too few factors gives a more severe error than specifying too many factors (Wood et al., 1996; Fava and Velicer, 1992; Comrey, 1978). This is because a variable that depends on a factor not included in the model will appear to depend falsely on a factor included in the model, giving poor estimates of the factor loadings matrix. This can give rotated solutions (where two common factors are combined into one, obscuring the true structure) or solutions that are difficult to interpret. In comparison, factor analysis with more factors than necessary leads to rotated solutions where the main factors are accurately represented but additional factors are associated with weak factor loadings or only one measured variable. In Chapter 5, I will outline various Bayesian approaches to determining the number of factors k, with an application to a set of weather observations.

2.3 Introduction to Monte Carlo

One of the goals of this thesis is to infer properties of an unknown probability distribution that describes the latent factors governing the behaviour of a dataset, such as temperature observations from a collection of weather stations scattered over a region. This will be approached from a Bayesian viewpoint, where prior assumptions are combined with data to learn about the posterior probability distributions, in what is known as **Bayesian inference**. Chapter 5 will be concerned with developing algorithms to do this through stochastic simulation, more commonly known as Monte Carlo simulation. This section will therefore take a more mathematically rigorous tone in preparation for the algorithms discussed in Chapter 5. First, I will outline Monte Carlo integration and importance sampling as methods to approximate an integral, followed by Markov chain Monte Carlo (MCMC) for sampling from an unknown distribution. Building on these basics, I will outline the key methods used in Chapter 5, reversible jump

MCMC (RJMCMC) and Sequential Monte Carlo (SMC), which are used to learn not just the latent factors, but also the number of latent factors that are required to describe a dataset via factor analysis.

2.3.1 Notation

Before describing these methods, I will introduce the notation used throughout this section. I will use X to refer to a random variable and x to be the value it takes, with probability density function generally represented by a Greek letter, such as $\pi(x)$. Any data will be denoted y. I will often refer to the 'target distribution' being the distribution of interest that we wish to sample from. In this thesis, the target distribution is the posterior distribution of x given the data y, which we will denote $\pi(x|y)$. For clarity, we will always use $p(x)$ to denote the prior distribution of x and $f(y|x)$ to denote the likelihood. Bayes theorem is therefore written as (Bayes and Price, 1763)

$$\pi(x|y) = \frac{f(y|x)p(x)}{\int f(y|x)p(x)dx}\,. \tag{2.31}$$

The denominator is the marginal likelihood or normalising constant, which we will often denote Z, i.e.

$$Z = \int f(y|x)p(x)dx\,. \tag{2.32}$$

Later in this section, we will be interested in cases when the variable of interest consists of a target variable, k, and a latent variable, θ, i.e. $x = (k, \theta)$. In particular, in RJMCMC (Section 2.5) and Transformation SMC (Section 2.6.4), the variable k will be a discrete variable describing the model, which determines the number of dimensions of the latent variable, denoting this explicitly with $\theta^{(k)}$.

2.3.2 Monte Carlo Integration

We will start with a discussion of Monte Carlo integration. Suppose we want to evaluate the integral

$$I = \int_{a}^{b} h(x)\mathrm{d}x \tag{2.33}$$

where $h(x)$ is a complicated function with no known closed form expression for its integral, I. We can estimate I using a variety of numerical methods (e.g. Riemann integration, Simpson's rule, trapezoidal rule, Gaussian quadrature) however, these approaches suffer the curse of dimensionality in which the number of points needed to control the error scales exponentially as the dimension increases (Robert and Casella, 2004). For integrals in large dimensions, we can take a simulation based approach, called **Monte Carlo** integration (Wasserman, 2004).

In Monte Carlo integration, we write this integral in terms of a function, $w(x)$, and a probability distribution that we can sample from, $\pi(x)$,

$$I = \int_{a}^{b} w(x)\pi(x)\mathrm{d}x\,. \tag{2.34}$$

For example, we can take π to be a probability density of a uniform random variable over the interval (a, b), i.e. $\pi(x) = \mathrm{Unif}(a, b) = {}^{1}\!/\!{\scriptstyle (b-a)}$ and $w(x) = h(x)\,(b - a)$. This means I is the expectation of $w(x)$, over probability density $\pi(x)$,

$$I = \mathbb{E}_{\pi}(w(X)) \tag{2.35}$$

where samples $X \sim \pi(\cdot)$. We can generate many samples $X_1, \cdots, X_N$ and using the law of large numbers, we can approximate I with

$$\hat{I} = \frac{1}{N}\sum_{i=1}^{N} w(X_i)\,. \tag{2.36}$$

This converges to I in the limit $N \to \infty$ (Wasserman, 2004). The variance of the estimate is

$$\sigma_{\hat{I}}^2 = \mathrm{Var}_{\pi}(\hat{I}) = \frac{1}{N}\sum_{i=1}^{N}(w(X_i) - \hat{I})^2\,. \tag{2.37}$$

Using the relations for variance that state that, for constant a, $\mathrm{Var}(aX) = a^2\mathrm{Var}(X)$, and the summation property, $\mathrm{Var}(\sum_i X_i) = \sum_i \mathrm{Var}(X_i)$, we can show that

$$\begin{aligned}\sigma_{\hat{I}}^2 = \mathrm{Var}_\pi(\hat{I}) &= \mathrm{Var}_\pi\left(\frac{1}{N}\sum_{i=1}^{N} w(X_i)\right) \\ &= \frac{1}{N^2}\sum_{i=1}^{N} \mathrm{Var}_\pi(w(X_i)) \\ &= \frac{1}{N}\mathrm{Var}_\pi(w(X)) = \frac{1}{N}\sigma_w^2\,.\end{aligned} \tag{2.38}$$

Hence the standard error on the estimator $\sigma_{\hat{I}}$, scales with $\frac{1}{\sqrt{N}}$. To decrease the error by a factor of 2, we need 4 times as many points. However, unlike alternative numerical methods of estimating I, this does not depend on the dimension of x. This makes Monte Carlo integration a popular tool for high-dimensional integrals.

The same Monte Carlo integration method can be applied to evaluating the integral of the function $h(x)$ with respect to a known probability density $\pi(x)$, from which we can sample $X_1, \cdots X_N \sim \pi(\cdot)$, when

$$\begin{aligned}I &= \int h(x)\pi(x)\mathrm{d}x \\ &= \mathbb{E}_\pi\left[h\left(X\right)\right]\,.\end{aligned}$$

Using the same logic, we could then estimate I with

$$\hat{I} = \frac{1}{N}\sum_{i=1}^{N} h(X_i)\,. \tag{2.39}$$

2.3.3 Importance Sampling

In many instances, we cannot sample directly from $\pi(x)$. Instead, we can modify how we write the integral, so that it becomes an integral against a given probability density that we can sample from, $\rho(x)$, which could be, for instance, a normal distribution (Robert and Casella, 2004). Our integral becomes

$$\begin{aligned} I &= \int h(x)\pi(x)\mathrm{d}x = \int \frac{h(x)\pi(x)}{\rho(x)}\rho(x)\mathrm{d}x \\ &= \mathbb{E}_\rho\left[\frac{h\left(x\right)\pi\left(x\right)}{\rho\left(x\right)}\right] . \end{aligned}$$

Then, we can simulate $X_1, \cdots X_N \sim \rho(\cdot)$ and estimate I with

$$\hat{I} = \frac{1}{N}\sum_{i=1}^{N}\frac{h(X_i)\pi(X_i)}{\rho(X_i)} . \tag{2.40}$$

We rewrite this as

$$\hat{I} = \frac{1}{N}\sum_{i}^{N} w(x_i)h(x_i) \tag{2.41}$$

where we have defined

$$w(x_i) = \frac{\pi(x_i)}{\rho(x_i)} . \tag{2.42}$$

These are weights or 'importance functions', giving rise to the name importance sampling (Robert and Casella, 2004).

Importance sampling is used when we either cannot sample directly from $\pi(x)$ or when doing so is inefficient because the density $\pi(x)$ is low where the function we are interested $h(x)$ is high (Robert and Casella, 2004). Ripley (1987) highlights this through the example that if the function $h(x)$ is focused over a region in the tail of a distribution, such as the Cauchy distribution. Most of the samples generated from $\pi(x)$ will fall under the main body of the distribution, rather than the tails which are the region of interest for $h(x)$. Instead, it would be preferable to sample from a distribution that favours the regions in which $h(x)$ is greater, in this case, the tails of the Cauchy distribution. Doing so means we can achieve a lower variance given a fixed N and therefore a greater efficiency (Ripley, 1987). Consequently, we generally want $\rho(x)$ to have heavier tails than $\pi(x)$. The opposite being true can cause problems as the ratio π/ρ can be unbounded if the tails of ρ are lighter than the tails of π. This means the weights can vary greatly, giving too much importance to a few values x_i in the tails and for some functions $h(x)$, unbounded π/ρ can lead to unbounded variance on the estimator $\mathbb{E}_\rho(\frac{h(x)\pi(x)}{\rho(x)})$

(Robert and Casella, 2004).

Importance sampling is also a valuable tool for estimating the normalising constant (Equation (2.32)), which can be difficult as it is often a high dimensional integral (Gelman and Meng, 1998). The normalising constant can simply be estimated as an average over the weights, i.e.

$$\hat{Z} = \frac{1}{N}\sum_{i=1}^{N} w_i \,. \tag{2.43}$$

2.4 Markov Chain Monte Carlo

We will now discuss Markov chain Monte Carlo, another simulation based approach for estimating integrals such as Equation (2.34), that also provides samples from a probability distribution that we cannot simulate from directly.

We start with the definition of a **stochastic process** as a collection of random variables, $\{X_n : n \in N\}$ indexed by an index set $n = \{0, 1, 2, \cdots\}$. This index could represent, for example, time or space (historically the index referred to time, hence the term 'process'). This stochastic process is a **Markov chain** if the distribution of X_n depends only on the value at the previous index, X_{n-1} (Brooks et al., 2011), i.e.

$$\mathbb{P}(X_n = x | X_0, X_1, \cdots, X_{n-1}) = \mathbb{P}(X_n = x | X_{n-1}) \,. \tag{2.44}$$

Markov chains are defined by a **transition kernel** K, a conditional probability density that gives

$$X_{n+1} = K(X_n, X_{n+1}) \tag{2.45}$$

(Robert and Casella, 2004). This is the function that determines the transition from X_n to X_{n+1}. One of the most common forms is the random walk process, where

$$X_{n+1} = X_n + \epsilon_n \tag{2.46}$$

and ϵ_n is independent of $X_n, X_{n-1}, \cdots$, for example $\epsilon \sim \mathcal{N}(0,1)$.

2.4.1 Properties of MCMC

A stochastic process is **stationary** if the distribution of $\{X_{n+1}, \cdots, X_{n+k}\}$ is independent of X_n for every positive integer k (Brooks et al., 2011). An initial distribution is **stationary** (or **invariant** or **equilibrium**) if a Markov chain specified by it is stationary for some transition kernel. In this case, the transition kernel **preserves** the initial distribution. These properties are required in MCMC, as we aim to construct a method that generates values from a stationary distribution of interest.

One way to achieve stationarity is through a property called **reversibility** (Brooks et al., 2011). A Markov chain is said to be reversible with respect to an initial distribution if its transition kernel is reversible with respect to its distribution i.e.

$$K(X_n, X_{n+1}) = K(X_{n+1}, X_n) \tag{2.47}$$

In other words, the laws running forward and backwards in time are the same. For a kernel that satisfies **detailed balance**,

$$K(x,y)\pi(x) = K(y,x)\pi(y) \tag{2.48}$$

with a probability density function π, for every (x, y), it is simple to prove that π is the invariant distribution by integrating over x and y (Robert and Casella, 2004). This gives

$$\int_{\mathcal{X}} K(x,B)\pi(x)\mathrm{d}x = \int_{\mathcal{X}}\int_B K(x,y)\pi(x)\mathrm{d}x\mathrm{d}y = \int_{\mathcal{X}}\int_B K(y,x)\pi(y)\mathrm{d}x\mathrm{d}y = \int_B \pi(y)\mathrm{d}y$$

for any measureable set B where we have used $\int K(x,y)dx = 1$ in the last equality. The middle equality shows the transition kernel is reversible with respect to x and y and the final equality shows π is the invariant probability density of the chain.

This condition is sufficient but not necessary for stationarity i.e. reversibility implies stationarity, but not vice-versa. Furthermore, reversibility simplifies Markov chain central limit theory (CLT) and asymptotic variance estimation (Brooks et al., 2011). This leads to reversible MCMC methods being a popular choice, but it is not required for stationarity (e.g. Andrieu et al., 2018).

The stationarity condition ensures that the probability of visiting each part of the state space remains constant throughout the chain, regardless of current position in the state space. To do this, we require that the chain is **irreducible**. This means that the kernel K allows for free moves all over the state space. There must exist finite n where $\mathbb{P}(X_n \in A|X_0) > 0$ for all A such that $\pi(A) > 0$. Irreducibility also implies **recurrence** of a chain, meaning the average number of visits to an arbitrary set A is infinite. Recurrence ensures that the chain has the same limiting behaviour, regardless of starting point, which brings us onto the topic of convergence.

Convergence is concerned with the limiting behaviour of the chain X_n. There are several conditions that should be met for the convergence of the chain's distribution to the invariant distribution, π. One of the most important conditions for the chain to be convergent is that it must be **ergodic**, meaning it must be independent of initial conditions (Robert and Casella, 2004). This is important in practice because we often do not know a suitable starting point for the chain such that $X_0 \sim \pi$ (this would be **perfect simulation**). For a chain to be independent of initial conditions, we require recurrence (the same limiting behaviour under different initial conditions) and aperiodicity (the chain does not re-visit any set A with a fixed period). Examples and details regarding these conditions can be found in Roberts and Rosenthal (2004).

Although in theory, one may construct an MCMC algorithm that satisfies all the relevant criteria for convergence, within finite computation we cannot be completely sure that convergence has been achieved. Therefore, we typically monitor the chain through a **trace plot** of the chain, such as the example provided in Figure 2.3 (Robert and Casella, 2004). In this example, the parameter is initialised outside of the stationary distribution. The first ~ 50 samples from the chain do not appear to be realistic samples from π, based on the remaining behaviour of the chain, and therefore we discard these from our sample. This is known as **burn-in** or **warm-up**. There is not one universal approach to choosing a sensible value to define when a

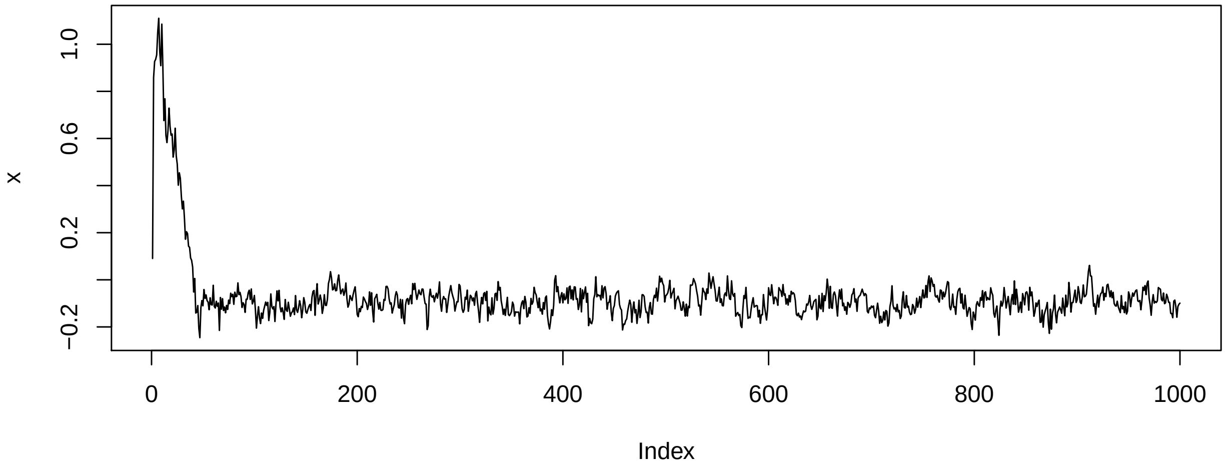


Figure 2.3: Example of an MCMC trace plot

chain has definitely ‘burned in’. A typical value to use is half the total number of simulations, but ultimately we rely on assessment of these plots and any expert knowledge of the system to make these decisions. We can look for any deviant or non-stationary behaviours that indicate convergence has **not** been achieved, but the absence of these does not prove that convergence has been achieved. Simulating a chain for longer than deemed necessary can help us check for any non-stationary behaviour, although this does not eliminate the possibility that the chain is ‘stuck’ in a local minima, rather than converged to the global minima.

To assess convergence, we may simulate multiple independent chains with different starting points x_m where $m \in (1, M)$ for M different chains. There are several benefits to this approach. Firstly, it reduces variability and dependence on initial values. Secondly, multiple chains also provide more opportunities to explore the full distribution, whereas a single chain may not visit the full support, or may over-explore a particular region, which can be very difficult to detect with a single chain. Finally, comparing quantities of interest across multiple chains makes it easier to assess the convergence to the stationary distribution (Robert and Casella, 2004). Furthermore, these chains are completely independent and can therefore be run in parallel, allowing us to use several computational nodes, which are often readily available, over the same time period. This makes running M shorter independent chains more efficient than running one single long chain as the same number of samples can be achieved in almost $1/M$th time. The exception to this is the iterations that are discarded due to burn-in, before convergence of each chain is reached.

A second requirement of a Monte Carlo sampler is the convergence of averages. The empirical average

$$\frac{1}{N}\sum_{i=1}^{N} h(x_i) \tag{2.49}$$

should converge to $E_\pi[h(x)]$ for arbitrary function h (Robert and Casella, 2004). This type of convergence assesses whether the chain exhibits the same features of the limiting distribution π, e.g. all the modes. If the chain follows the limiting distribution π then we would expect this to be the case, but in practice, a chain may not explore the full distribution π in the finite time for which we run the chain. We must ensure that we choose an appropriate minimal value of N that achieves a given level of accuracy for the convergence of the empirical averages $E_\pi[h(x)]$ (Robert and Casella, 2004). This type of convergence is related to the mixing speed of the chain (Brooks et al., 2011), i.e. how quickly the full support of π is explored and how dependent this is on initial conditions. We will call this **efficiency**. We may find that some aspects of the target distribution are explored more quickly than others which indicates the fundamental problem is mixing. A chain with poor mixing can produce highly correlated samples. This leads to an overestimate of the variance of $h(x)$.

2.4.2 Metropolis-Hastings

Now that we have covered some of the theory behind MCMC, we will introduce some of the common approaches to constructing a valid MCMC algorithm, starting with the Metropolis-Hastings algorithm. In Section 2.4, we defined the transition kernel K which describes the transition from x_n to x_{n+1}. We typically construct this kernel by proposing a new parameter value, x' from a proposal distribution $q(x'|x_i)$, where x_i is the parameter value at the current iteration i. We then decide whether or not to accept this based on the **acceptance probability** $\alpha(x \to x')$, given by

$$\alpha(x \to x') = \min\Big\{1, r(x \to x')\Big\} \tag{2.50}$$

where $r(x \to x')$ is the **acceptance rate**.

For the Metropolis-Hastings algorithm, the acceptance rate is

$$r_{MH}(x \to x') = \frac{\pi(x'|y)}{\pi(x|y)} \frac{q(x|x')}{q(x'|x)} \tag{2.51}$$

$$= \frac{f(y|x')p(x')}{f(y|x)p(x)} \frac{q(x|x')}{q(x'|x)} \tag{2.52}$$

where $\pi(x|y)$ is the posterior probability for x given the data y. The second equality uses Bayes' theorem (Equation (2.31)) to write this in terms of the likelihood $f(y|x)$ and the prior $p(x)$.

We then accept the proposed state, x' with probability α. If the state is accepted, we set $x_{i+1} = x'$. Otherwise, we reject the state and set $x_{i+1} = x$. Algorithm 1 describes these steps.

Algorithm 1 Metropolis-Hastings algorithm

```
1. Start with $x_0$ from prior
2. For $x_i$ for $i \in 0, \cdots, N_{iter}$
    (a) Propose $x' \sim q(\cdot|x_i)$
    (b) Evaluate acceptance rate, $r_{MH}$ with Equation (2.52)
    (c) With acceptance probability $\alpha = \min(1, r_{MH})$ either accept and set $x_{i+1} = x'$ or
        otherwise reject and set $x_{i+1} = x_i$
```

This form of MCMC is the most well known and used across many disciplines, since there is freedom to pick a proposal distribution $q(x|x')$ that is suitable for the problem at hand. When this algorithm was first introduced as the 'Metropolis algorithm' in Metropolis et al. (1953), the proposal distribution was defined to be symmetric so that $q(x|x') = q(x'|x)$, but this was later extended to the Metropolis-Hastings algorithm where symmetry is not required (Hastings, 1970). Choosing a sensible proposal distribution is one of the most important parts when constructing an MCMC and it usually requires some expert knowledge of the target distribution. One of the minimal requirements for convergence is that the support of the proposal distribution must cover the support of the target distribution, since points that lie outside the proposal distribution can never be proposed. The choice of proposal distribution also strongly determines how quickly the full space will be explored and thus the efficiency. The closer to the proposal is to the target distribution, the more efficient the chain will be.

This is highlighted when we take the proposal distribution to be an approximation to the target, say $\rho(x) \approx \pi(x)$. Setting $q(x|x') = \rho(x)$, the acceptance rate is

$$r_{MH}(x \to x) = \frac{\pi(x'|y)}{\pi(x|y)} \frac{\rho(x)}{\rho(x')} = \frac{\pi(x'|y)}{\rho(x')} \frac{\rho(x)}{\pi(x|y)} = \frac{w(x')}{w(x)} \tag{2.53}$$

showing that the acceptance rate is the ratio of the importance function weights w defined in Equation (2.42). This highlights the relationship between MCMC and importance sampling. We can see that the better the proposal approximates to the target, the higher the acceptance rate.

One of the most common styles of proposal distributions is the **random walk**. In the case that the proposal is of the form $x' = x_i + \epsilon_i$ where ϵ_i comes from a distribution such as the $\mathcal{N}(0, V)$ where V is a fixed value for the variance on the proposed points (Wasserman, 2004). Without considering acceptance rate, this would form a random walk simulation.

The value V should be chosen carefully, since if it is too small, the proposed points are mostly accepted but in taking small steps, the chain does not explore the sample space well (or it takes much longer to do so, i.e. we have low efficiency). In contrast, setting the proposal variance too large results in high rejection rates as the algorithm proposes new points far away from the current position. This can lead to the chain appearing 'stuck' in the same position for many iterations. Both these algorithms result in poor approximations for the posterior distributions because of poor mixing and even though the theory implies the chain should converge, in practice this will not be seen in finite computation. Therefore, careful tuning of the proposal variance is required, both to obtain the correct posterior distribution and to have an efficient chain.

The Metropolis-Hastings algorithm can be applied when we have a multivariate target variable, $\mathbf{x} = (x_1, x_2, \cdots x_n)$. If we can construct a valid proposal distribution, we can update these parameters together and evaluate the acceptance rate in the same way. However, in high-dimensions it can become difficult to propose sensible parameters across all dimensions, which can lead to low acceptance rates. Instead, we may be able to turn to another popular MCMC algorithm, **Gibbs sampling**, that does not suffer from this problem.

2.4.3 Gibbs Sampling

First proposed in Geman and Geman (1984), Gibbs sampling is a specific case of the Metropolis-Hastings algorithm in which it is possible to derive and sample from the conditional posterior distribution $\pi(x|y)$. Then, we can propose new points from this conditional posterior distribution $x' \sim q(\cdot|x) = \pi(x|y)$. When this is the case, the numerator and denominator of Equation (2.52) cancel and the acceptance probability $r_G = 1$ (Geman and Geman, 1984).

This is still true, even when we have more than one parameter as we can update each parameter at a time. For example, when $x = (x_1, x_2)$ we can update x'_1 first based on the full conditional distribution e.g. $q(x'_1|x_1, x_2) = \pi(x'_1|y)$. Then we can update x'_2 based on the new full conditional and our estimate of x'_1, e.g. $q(x'_2|x'_1, x_1, x_2) = \pi(x'_2|y)$. This gives

$$\begin{aligned} r_G(x, x') &= \frac{\pi(x'_2|x'_1, y)\pi(x'_1|y)}{\pi(x_2|x_1, y)\pi(x_1|y)} \frac{q(x_2, x_1|x'_1, x'_2)}{q(x'_2, x'_1|x_1, x_2)} \\ &= \frac{\pi(x'_2|x'_1, y)\pi(x'_1|y)}{\pi(x_2|x_1, y)\pi(x_1|y)} \frac{q(x_2|x_1, x'_1, x'_2)q(x_1|x'_1, x'_2)}{q(x'_2|x'_1, x_1, x_2)q(x'_1|x_1, x_2)} \\ &= \frac{\pi(x'_2|x'_1, y)\pi(x'_1|y)}{\pi(x_2|x_1, y)\pi(x_1|y)} \frac{\pi(x_2|x_1, y)\pi(x_1|y)}{\pi(x'_2|x'_1, y)\pi(x'_1|y)} \\ &= 1\,. \end{aligned}$$

This can be extended to more parameters with the same logic.

Since we do not need to evaluate the acceptance rate, this makes an efficient algorithm and it tends to perform well in high dimensional situations. However, we are forced to move each parameter independently, which means if there are many highly correlated parameters, we cannot use this knowledge to move towards a region of higher density, by moving multiple parameters together. Furthermore, we do not always know the conditional distribution or we may be unable to sample it from it.

2.4.4 Metropolis within Gibbs

We are often concerned with a high dimensional parameter space where the Metropolis-Hastings algorithm suffers from low acceptance rates, but we may be unable to sample directly from the conditional distributions. Instead, we can use the structure of the Gibbs sampling algorithm but replace the proposal distribution with the Metropolis-Hastings proposal and evaluate the Metropolis-Hastings acceptance rate at each step. In other words, we can update individual parameters using separate Metropolis-Hastings steps. This approach is sometimes called 'Variable-at-a-Time Metropolis-Hastings' (Brooks et al., 2011). By doing this, we can improve the acceptance rates and thus improve efficiency compared to the Metropolis-Hastings algorithm. This does, however, come at a higher cost as we have to evaluate the acceptance rate for each dimension. Furthermore, we cannot use strong correlations between parameters to move efficiently to high density regions of the space. Interestingly, the original sampler presented in Metropolis et al. (1953) was in fact of this form.

2.4.5 MCMC with latent variables

Often, the likelihood $f(y|x)$ is intractable, meaning it cannot be evaluated pointwise. These situations arise when the dataset is large (and the likelihood involves a product of many terms), when the likelihood is known only up to a normalising constant, when the likelihood can be sampled from but not evaluated (e.g. the data follows a complex stochastic computer model) or when there are latent variables that require marginalising out of the likelihood through a high dimensional integral (Everitt et al., 2017). In this section, we will be concerned with the case in which there are a large number of latent variables, in preparation for the reversible jump algorithm introduced in the following section.

In this section, the target variable will be k, with probability density $\pi(k)$ known up to the normalising constant. We also have θ, a random variable with probability density function $\psi(\theta)$. This is a latent variable needed to fully describe the system. If the primary interest is modelling probability distribution $\pi(k)$ alone, (and assuming this is known up to an unknown constant) we can use the Algorithm 1 simply replacing $x = k$ as our target variable. This is called the standard algorithm (Nicholls et al., 2012), the idealised algorithm (Karagiannis and Andrieu, 2013) or the marginal algorithm, since θ is marginalised out of the likelihood, $f(y|k) = \int_\theta f(y|k,\theta)\psi(\theta)d\theta$.

However, when $f(y|k)$ (and therefore $\pi(k|y)$) is intractable, introducing and sampling from the additional variable θ can make this problem simpler and possibly even tractable. It also allows us to construct chains that are more efficient and therefore allows faster simulation (Higdon, 1998). The inclusion of latent variables θ in the algorithm is called the latent variable method (Besag and Green, 1993; Higdon, 1998).

We can do this by specifying the conditional distribution $\pi(\theta|k)$ so that the joint distribution is

$$\pi(k,\theta) = \pi(k)\,\pi(\theta|k)\,. \tag{2.54}$$

This joint distribution admits the same marginal distribution, $\pi(k)$ as before (Besag and Green, 1993). We can target the joint distribution by letting $x = (k, \theta)$ in Algorithm 1 and in the Metropolis-Hastings acceptance rate (Equation (2.50)), which gives

$$r_{joint}(k,\theta \to k',\theta') = \frac{f(y|k',\theta')\,p(\theta')\,p(k')}{f(y|k,\theta)\,p(\theta)\,p(k)}\,\frac{q(k,\theta|k',\theta')}{q(k',\theta'|k,\theta)} \tag{2.55}$$

In practise, this algorithm would involve alternating between sampling k' and sampling θ' in step 2(a) of Algorithm 1 (in the same way as Gibbs sampling alternates between two different variables) (Higdon, 1998).

However, we aim to target the marginal distribution, $\pi(k|y)$ and we assume that this and the likelihood $f(y|k)$ is unknown in exact form. Instead, we can use an unbiased estimator in the acceptance ratio to achieve the correct target distribution $\pi(k|y)$. This is called the 'pseudo marginal algorithm'. There are two types of algorithms that fit into this class that will be used in this thesis: the pseudo marginal target and the pseudo marginal ratio algorithm.

2.4.6 Pseudo Marginal Target Algorithm

In many situations, the marginal likelihood $f(y|k)$ is unknown but can be approximated with an unbiased estimate, $\widehat{f(y|k)}$. For example, we can use an estimate with a single importance sample point:

$$\widehat{f(y|k)} = \frac{f(y|k,\theta)p(\theta|k)}{q(\theta|k,\theta',k')} \tag{2.56}$$

where $\theta \sim q(\cdot|k,\theta',k')$. The acceptance ratio is

$$r_{pseudo-marginal}(k \rightarrow k') = \frac{\widehat{f(y|k')}\,p(k')}{\widehat{f(y|k)}\,p(k)}\,\frac{q(k|k')}{q(k'|k)}\,. \tag{2.57}$$

Because we are estimating $f(y|k)$ with Equation (2.56), we are effectively targeting the joint posterior $\pi(k,\theta|y)$ (as in Equation (2.55)) rather than the marginal posterior $\pi(k|y)$. However, since $\widehat{f(y|k)}$ is an unbiased estimator, it can be shown that samples from this algorithm converges to $\pi(k|y)$ (Andrieu and Vihola, 2015; Beaumont, 2003). As well as being used for intractable marginal likelihoods (Andrieu and Vihola, 2015), this approach can achieve more efficient mixing (Beaumont, 2003). This is because exploring θ space often leads to improved mixing in k-space.

Note that at each iteration, we re-use the estimate of the marginal likelihood $\widehat{f(y|k)}$ from the previous iteration. This has a small computational benefit and means that we only need to calculate the likelihood once per iteration. However, it brings a greater disadvantage in the case that it can lead to poor performance when $\widehat{f(y|k)}$ is a poor estimate. If $\widehat{f(y|k)}$ overestimates $f(y|k)$, it is difficult to propose a new point that will be accepted when compared to the overestimated $\widehat{f(y|k)}$. Because $\widehat{f(y|k)}$ is re-used at each iteration, we keep proposing new points to compare to this estimate until a proposal is accepted. This can lead to many rejections and the MCMC algorithm getting 'stuck' at a certain point, purely due to the poor estimate of the marginal likelihood.

2.4.7 Pseudo Marginal Ratio Algorithm

An alternative approach when the marginal likelihood is not known exactly, is to estimate the entire ratio, $\frac{f(y|k')}{f(y|k)}$, i.e.

$$\begin{aligned} r_{pseudo-ratio}(k \to k') &= \left(\widehat{\frac{\pi(y|k')}{\pi(y|k)}}\right)\frac{q(k|k')}{q(k'|k)} \\ &= \left(\widehat{\frac{f(y|k')}{f(y|k)}}\right)\frac{p(k')}{p(k)}\frac{q(k|k')}{q(k'|k)} \, . \end{aligned}$$

There are various ways to do this. One is to treat the problem as estimating the ratio of normalising constants, $Z(k')/Z(k)$,

$$\left(\widehat{\frac{f(y|k')}{f(y|k)}}\right) = \frac{\widehat{Z(k')}}{Z(k)} \tag{2.58}$$

which can be achieved by choosing the proposal distribution so that it has the same normalising constant and these ratios cancel. This is called the exchange algorithm and is not used in this thesis, but further details can be found in Murray et al. (2006) and Møller et al. (2004).

To estimate the ratio, we will use an importance sampling estimate of the ratio with samples from q

$$\left(\widehat{\frac{f(y|k')}{f(y|k)}}\right) = \frac{f(y|k',\theta')p(\theta')\,q(\theta|k,k',\theta')}{f(y|k,\theta)p(\theta)\,q(\theta'|k',k,\theta)} \tag{2.59}$$

which gives an acceptance rate,

$$r_{pseudo-ratio}(k \to k') = \frac{f(y|k',\theta')\,p(\theta')\,p(k')}{f(y|k,\theta)\,p(\theta)\,p(k)}\,\frac{q(k|k')}{q(k'|k)}\frac{q(\theta|k,k',\theta')}{q(\theta'|k',k,\theta)} \, .$$

Unlike the pseudo marginal target algorithm, we re-calculate the full ratio of likelihoods at each step (rather than just the numerator). This means we do not encounter the problem of an overestimate of the marginal likelihood leading to many rejections. This is called the pseudo ratio or the pseudo marginal ratio (Andrieu and Vihola, 2015).

2.5 Reversible Jump MCMC

In this section, we will cover the case in which the latent variables θ associated with target variable k can live on spaces that can be of a completely different nature, including a different dimension. We define k to be a model for the data, $k \in \mathcal{K}$. Within model k, we write latent variables as $\theta^{(k)} \in \mathcal{R}^{n_k}$ where the dimension of random variable vector $\theta^{(k)}$ is n_k, which differs for each model k.

If the primary interest is model probability distribution alone, $\pi(k)$ (and assuming this is known up to an unknown constant) an idealised MCMC algorithm can be used (Karagiannis and Andrieu, 2013). However, in most scenarios $\pi(k)$ is intractable and instead we extend this algorithm to sample from both the model and the parameter values within that model, $\pi(k, \theta^{(k)})$. In many cases, we may also be interested in model parameters $\theta^{(k)}$, such as the application to factor analysis which I describe here. The joint distribution of variables is

$$\pi(k, \theta^{(k)}, y) = \pi(k)\, \pi(\theta^{(k)}|k)\, f(y|\theta^{(k)}, k) \tag{2.60}$$

which describes the hierarchical structure of the model, where $\theta^{(k)}$ is defined for each model k (Green, 1995).

The parameters $\theta^{(k)}$ lie on a different space to $\theta^{(k')}$ which can involve different dimensions ($n_k \neq n_{k'}$). The Reversible Jump (RJ) MCMC algorithm, first proposed in (Green, 1995), allows a move or a ‘jump’ from one model k to another model k'. These inter-dimensional moves are constructed with reversible transformations, from $(k, \theta^{(k)})$ to $(k', \theta^{(k')})$. As before, this involves proposing the new model, k', and parameter values, $\theta^{(k')}$, and evaluating an acceptance rate. The key difference is that the transitions from one model to another must be defined correctly to ensure that it satisfies the conditions outlined in Section 2.4. These inter-dimensional moves are combined with the typical MCMC moves within a specified dimension and hence gives the algorithm the name RJMCMC (Green, 1995).

RJMCMC targets the joint posterior, $\pi(k, \theta^{(k)}|y)$ which can be factorised as

$$\pi(k, \theta^{(k)}|y) = \pi(k|y)\, \pi(\theta^{(k)}|k, y)\,. \tag{2.61}$$

These two terms can be treated separately with moves on k followed by moves on $\theta^{(k)}$, rather than carrying out model averaging over $\theta^{(k)}$ (Green, 1995).

We need to define an appropriate Markov transition kernel, $K(x, x')$, that satisfies the conditions described in Section 2.4. In MCMC, the acceptance rate is $r = \frac{\pi(x'|y)q(x|x')}{\pi(x|y)q(x'|x)}$, but when $x = (k, \theta^{(k)})$ and $x' = (k', \theta^{(k')})$ exist on different spaces, the ratio of $\pi(x'|y)q(x|x')$ to $\pi(x|y)q(x'|x)$ must be defined rigorously on a common dominating measure. This is ensured by defining a 'dimension balancing' or 'dimension matching' condition on $q(x'|x)$ so that the degrees of freedom match (Richardson and Green, 1997). This is also often described as extending the space in which the MCMC is carried out to an 'augmented space' which includes the maximum number of dimensions.

In particular, we are concerned with the moves on $\theta^{(k)}$, which exist on different spaces, while the moves on k can be considered in the same way as before. This separation is written in the proposal distribution with

$$q(x'|x) = q(k', \theta'^{(k')}|k, \theta^{(k)}) = q(k'|k)\, q(\theta'^{(k')}|k', k, \theta^{(k)})\,. \tag{2.62}$$

In reversible jump, it is typical to propose a move from k to either $k+1$ or $k-1$ with probability $\frac{1}{2}$ on each of these. The exception to this is when $k = k_{\max}$ or $k = k_{\min}$ and therefore the proposed move is $k_{\max} - 1$ or $k_{\min} + 1$ with probability 1. The proposal distribution on k is therefore

$$q(k'|k) = \begin{cases} \frac{1}{2} & k' = k+1, \qquad k \neq k_{\max} \\ \frac{1}{2} & k' = k-1, \qquad k \neq k_{\min} \\ 1 & \begin{cases} k' = k_{\max} - 1, & k = k_{\max} \\ k' = k_{\min} + 1, & k = k_{\min} \end{cases} \end{cases} \tag{2.63}$$

which is used in all RJMCMC algorithms in this thesis.

We also require a suitable proposal for $\theta'^{(k')}$, which is more complicated due to the different spaces that $\theta^{(k)}$ and $\theta'^{(k')}$ lie on. Dimension matching requires that the two subspaces have the same number of dimensions during the transition. This is typically done by introducing new auxilliary variables, $u^{(k)}$, which I outline here by transitioning from $k = 1$ to $k' = 2$ (Green, 1995). To transition from $\theta^{(1)}$ of size n_1 to $\theta^{(2)}$ of size n_2, first we generate a vector of continuous random variables $u^{(1)}$ of length m_1. We will denote the probability density function of this as $\psi_{(1\to 2)}(u^{(1)})$. Then, a deterministic function of $\theta^{(1)}$ and $u^{(1)}$ is used to determine $\theta^{(2)}$. Similarly, in the reverse process of transition from $\theta^{(2)}$ of size n_2 to $\theta^{(1)}$ of size n_1, we generate a vector of continuous random variables $u^{(2)}$ of length m_2 with probability density function $\psi_{(2\to 1)}(u^{(2)})$. Then we apply a deterministic function of $\theta^{(2)}$ and $u^{(2)}$ to obtain $\theta^{(1)}$. The dimension matching requirement is there must be a bijection between $(\theta^{(1)}, u^{(1)})$ and $(\theta^{(2)}, u^{(2)})$, so that the lengths of these random vectors are equal, i.e. $n_1 + m_1 = n_2 + m_2$.

For clarity, in the following section we will denote the probability distribution on $(\theta^{(1)}, u^{(1)})$ with

$$\pi(\theta^{(1)}, u^{(1)}) = \varphi_1(\theta^{(1)})\, \psi_{1\to 2}(u^{(1)})\,. \tag{2.64}$$

When we introduce the new auxiliary variables, the detailed balance condition (Equation (2.48)) on the transition from $(\theta^{(1)}, u^{(1)})$ to $(\theta^{(2)}, u^{(2)})$ requires that

$$\begin{aligned}\int \varphi_1(\theta^{(1)})\psi_{(1\to 2)}(u^{(1)})K(\theta^{(1)}, u^{(1)}, \theta^{(2)}, u^{(2)})d\theta^{(1)}du^{(1)} \\ = \int \varphi_2(\theta^{(2)})\psi_{(2\to 1)}(u^{(2)})K(\theta^{(2)}, u^{(2)}, \theta^{(1)}, u^{(1)})d\theta^{(2)}du^{(2)}\,.\end{aligned} \tag{2.65}$$

Here we can see that both $(\theta^{(1)}, u^{(1)})$ and $(\theta^{(2)}, u^{(2)})$ must lie on the same space. This allows us to write

$$\begin{aligned}\varphi_1(\theta^{(1)})\psi_{(1\to 2)}(u^{(1)})K(\theta^{(1)}, u^{(1)}, \theta^{(2)}, u^{(2)}) \\ = \varphi_2(\theta^{(2)})\psi_{(2\to 1)}(u^{(2)})K(\theta^{(2)}, u^{(2)}, \theta^{(1)}, u^{(1)}) \left| \frac{\partial\left(\theta^{(2)}, u^{(2)}\right)}{\partial\left(\theta^{(1)}, u^{(1)}\right)} \right|\end{aligned} \tag{2.66}$$

where the term $\left|\frac{\partial\left(\theta^{(2)},u^{(2)}\right)}{\partial\left(\theta^{(1)},u^{(1)}\right)}\right|$ is the Jacobian.

This defines the proposal distributions $q(\theta'^{(k')}|k',k,\theta^{(k)})$ between $k=1$ and $k=2$ to be

$$q(\theta^{(1)}|1,2,\theta^{(2)}) = \varphi_1(\theta^{(1)})\,\psi_{(1\to 2)}(u^{(1)}) \tag{2.67}$$

$$q(\theta^{(2)}|2,1,\theta^{(1)}) = \varphi_2(\theta^{(2)})\,\psi_{(2\to 1)}(u^{(2)}) \left|\frac{\partial(\theta^{(2)},u^{(2)})}{\partial(\theta^{(1)},u^{(1)})}\right| . \tag{2.68}$$

This can be written more generally for k and k' and gives the following acceptance rate:

$$\begin{aligned} r_{RJ}(k\to k') =& \frac{\pi(k',\theta'^{(k')}|y)}{\pi(k,\theta^{(k)}|y)}\,\frac{q(k,\theta^{(k)}|k',\theta'^{(k')})}{q(k',\theta'^{(k')}|k,\theta^{(k)})} \\ =& \frac{\pi(k',\theta'^{(k')}|y)}{\pi(k,\theta^{(k)}|y)}\,\frac{q(k|k')}{q(k'|k)}\,\frac{q(\theta^{(k)}|k,k',\theta'^{(k')})}{q(\theta'^{(k')}|k',k,\theta^{(k)})} \\ =& \frac{\pi(k',\theta'^{(k')}|y)}{\pi(k,\theta^{(k)}|y)}\,\frac{q(k|k')}{q(k'|k)}\,\frac{\varphi_k(\theta^{(k)})\,\psi_{(k\to k')}(u^{(k)})}{\varphi_{k'}(\theta'^{(k')})\,\psi_{(k'\to k)}(u'^{(k')})\left|\frac{\partial\left(\theta'^{(k')},u'^{(k')}\right)}{\partial\left(\theta^{(k)},u^{(k)}\right)}\right|} . \end{aligned} \tag{2.69}$$

Usually, one of m_1 or m_2 these will be zero, meaning only one auxiliary variable is needed, which simplifies Equation (2.69). The construction of the RJ move can be difficult and we do not know the likelihood in exact form. As before, both the pseudo-marginal and the pseudo-ratio methods could be applied to the RJMCMC algorithm, both of which are explored for factor analysis in Chapter 5. The general form of RJMCMC is given by Algorithm 2.

Algorithm 2

```
1. Set initial $k$ and draw prior $\theta^{(k)}$. Set $X_0 = (k,\theta^{(k)})$
2. For $i = 1,\cdots,N$: current model is $X_i = (k,\theta^{(k)})$
   (a) Within model: Do 1 MCMC step (e.g. Metropolis-Hastings, Gibbs)
   (b) Between model: Do 1 RJ step
       i. Propose $k' \sim q(\cdot,k)$ with Equation (2.63)
       ii. Propose $(\theta'^{(k')},u^{(k')}) \sim q(\cdot|k',k,\theta^{(k)},u^{(k)})$.
       iii. Evaluate acceptance rate $r_{RJ}$ with equation 2.69.
       iv. With acceptance probability $\alpha = \min(1,r_{RJ})$ either accept and set $X_i = (k',\theta'^{(k')})$
           or reject and set $X_i = (k,\theta^{(k)})$.
```

2.6 Sequential Monte Carlo Methods

In this section, we will outline Sequential Monte Carlo (SMC), which allows us to sample from a probability distribution of interest, by building a sequence of intermediate distributions starting from a known distribution. In particular, this section builds up to a method called Transformation SMC that allows us to transition across probability distributions defined on different spaces, a concept which we have seen already when describing RJMCMC in Section 2.5. We will first outline annealed importance sampling, which has similar properties to SMC, starting with a brief recap of importance sampling.

2.6.1 Annealed Importance Sampling

Section 2.3.3 introduces importance sampling as a method for estimating an integral over $\pi(x)$, by taking samples from a different distribution $\rho(x)$. We estimated the integral $I = \int h(x)\pi(x)\mathrm{d}x$ with

$$\hat{I} = \frac{1}{N}\sum_{i}^{N} w(x_i)h(x_i) \tag{2.70}$$

where

$$w(x) = \frac{\pi(x)}{\rho(x)} \tag{2.71}$$

are the weights and points are generated from $\rho(\cdot)$. This can also be used to estimate the normalising constant with

$$\hat{Z} = \frac{1}{N}\sum_{i=1}^{N} w(x_i)\,. \tag{2.72}$$

The normalised weights, W_i, which can be estimated with $\hat{W}_i = w_i/\hat{Z}$, provide a useful indication of the cost of sampling from ρ rather than π (Neal, 2001). The sample size is effectively reduced by a factor of $1 + \mathrm{Var}(\hat{W})$. Furthermore, large variances also indicate that only a few of the large importance weights are dominating Equation (2.70), which can lead to inaccurate estimates of the integral.

Importance sampling works well if $\rho(x)$ is similar to the target distribution $\pi(x)$ everywhere, as a more 'similar' distribution will give large and equal weights across all x. However, if the proposal distribution, $\rho(x)$ differs greatly from $\pi(x)$, for example, if the peak of $\pi(x)$ lies in the tails of $\rho(x)$ where few points are proposed, it can give larger errors in the estimated integral. This is more problematic in higher dimensional spaces, because the distance between distributions grows exponentially with dimension. Given a fixed number of samples, the error in the estimated integral also grows exponentially with dimension (Agapiou et al., 2017). To control this error, the number of samples must be increased exponentially with dimension. This means the computational cost, typically measured as the number of samples required to control the error below a fixed bound, grows exponentially with dimension. One way around this is to reduce the distance between the proposal and target distributions, by introducing intermediate proposal distributions. This is called **annealed importance sampling** (Neal, 2001).

Annealed importance sampling makes use of **simulated annealing**, an approach used to bridge two distributions. Simulated annealing was first done in Kirkpatrick et al. (1983) for a combinatorics optimisation problem, using ideas from statistical mechanics that relate macroscopic properties to microscopic averages. The process of optimisation is viewed as analogous to the process of cooling a physical system to 0 K to reach a minimum energy state. This can be done in stages, broken into a controlled schedule called the annealing schedule. At each step, the system is is cooled to a controlled temperature and reaches a minimum energy state for this temperature. This allows the true minimum energy state to be found when the temperature is 0 K. In contrast, if cooled rapidly, the system may not reach the true minimum energy state because it can become stuck in local optima at intermediate steps and as it is further cooled the possibility of the system transitioning out of this local optimum is reduced. The annealing process avoids this problem by ensuring global minima are reached at each stage. In optimisation, the same approach can be applied where at each stage of the annealing schedule an intermediate distribution is chosen. Kirkpatrick et al. (1983) describe annealing as an evolutionary process, modelled purely through stochastic means.

Annealing effectively breaks down the progression from one distribution to another into multiple transitions from one intermediate distributions to the next (Neal, 2001). Here our target distribution is $\pi(x)$ and and we define the proposal distribution, $\pi_0(x)$, which we can sample from.

We define a sequence of **intermediate distributions**, $\pi_1, \cdots, \pi_{T-1}$. For each $t \in 1, \cdots, T-1$, we must be able to compute a function proportional to $\pi_t(x)$. This sequence of distributions can be constructed to suit the problem. A common definition is to use

$$\pi_t(x) = \pi(x)^{\gamma_t} \pi_0(x)^{1-\gamma_t} \tag{2.73}$$

where $\gamma_0 = 0$, $\gamma_T = 1$ and $0 = \gamma_0 < \gamma_1 < \gamma_2 < \cdots < \gamma_T = 1$. With each transition, the intermediate distribution becomes closer to the target distribution and further from the initial proposal distribution. A simple choice of γ_t is to choose $\gamma_t = \frac{t}{T}$. We must also define a Markov chain transition, K_t, that we can simulate from that leaves π_t invariant. This can involve either a single or multiple MCMC updates, for instance, a Metropolis-Hastings update.

We start with N draws from the proposal distribution, $x_0^{(i)} \sim \pi_0(\cdot)$, where we use superscripts to indicate each independent sample, $i = 1, \cdots, N$, and subscripts to indicate each intermediate distribution defined by Equation (2.73), $t = 0, 1, \cdots, T-1$. We want to use π_0 as an approximation to the next distribution, π_1. We can calculate the importance sampling weight, $\frac{\pi_1(x_0)}{\pi_0(x_0)}$ (from Equation (2.71)), which compares how well π_1 approximates π_0 evaluated at parameter value $x_0^{(i)}$.

For each sample, i, we then use the transition kernel $K_1(x_0, \cdot)$ to update x_1 with the invariant distribution π_1. This will be used as an estimate for a point drawn from the next distribution, π_2. Following the same steps, we calculate the importance sampling weight, $\frac{\pi_2(x_1)}{\pi_1(x_1)}$. This process is repeated until we have generated x_{T-1} from transition kernel $K_{T-1}(x_{T-2}, \cdot)$ which contributes importance sampling weight $\frac{\pi_T(x_{T-1})}{\pi_{T-1}(x_{T-1})}$. This gives a sequence of points $\{x_0, x_1, \cdots, x_{T-1}\}$, often called a trajectory, all drawn from from intermediate distributions $\{\pi_0, \pi_1, \cdots, \pi_{T-1}\}$. The final value, $x_T \sim K_T(x_{T-1}, \cdot)$, generates the sample from the target distribution i.e. $x^{(i)} = x_T \sim \pi_T$. We evaluate the **annealed importance weight** by multiplying the importance weights calculated in each transition,

$$w_{AIS}^{(i)} = \frac{\pi_1(x_0^{(i)})}{\pi_0(x_0^{(i)})} \frac{\pi_2(x_1^{(i)})}{\pi_1(x_1^{(i)})} \cdots \frac{\pi_T(x_{T-1}^{(i)})}{\pi_{T-1}(x_{T-1}^{(i)})} \tag{2.74}$$

$$= \prod_{t=1}^{T} \frac{\pi_t(x_{t-1}^{(i)})}{\pi_{t-1}(x_{t-1}^{(i)})} \tag{2.75}$$

Then, the integral can be estimated with

$$\hat{I} = \frac{1}{N} \sum_{i}^{N} w_{AIS}^{(i)}(x^{(i)}) h(x^{(i)}) = \frac{1}{N} \sum_{i}^{N} \prod_{t=1}^{T} \frac{\pi_t(x_{t-1}^{(i)})}{\pi_{t-1}(x_{t-1}^{(i)})} h(x^{(i)})\,. \tag{2.76}$$

Furthermore, we can still estimate the normalising constant with

$$\hat{Z} = \frac{1}{N} \sum_{i=1}^{N} w_{AIS}^{(i)}(x^{(i)}) = \frac{1}{N} \sum_{i=1}^{N} \prod_{t=1}^{T} \frac{\pi_t(x_{t-1}^{(i)})}{\pi_{t-1}(x_{t-1}^{(i)})}\,. \tag{2.77}$$

Introducing intermediate distributions to importance sampling is useful when the best proposal distribution that can be found and sampled from is still a poor approximation to the target distribution. For example, if the distance between the prior and target distributions is high, the importance sampling weights alone, $\pi(x)/\rho(x)$ can be low and the error on the estimate $\hat{I}$ may be high. If we introduce an intermediate distribution that lies between these distributions, we can effectively bridge the gap between the two. As noted for importance sampling, the efficiency of the estimation is reduced by a factor of $1 + \mathrm{Var}(\hat{W})$. Neal (2001) show that the variance of the normalised weights falls as the number of intermediate distributions are increased. Furthermore, as the dimensionality of the distribution is increased, the error in the importance sampling estimated integral grows exponentially (Agapiou et al., 2017). However, when annealed distributions are introduced the variance is only increased by a factor of d, the number of dimensions (Neal, 2001). This is because in annealed importance sampling, it is the distance between adjacent intermediate distributions $\pi_t(x_{t-1})$ and $\pi_{t-1}(x_{t-1})$ that are important. For independent states, $x_{t-1} \sim \pi_{t-1}$ each weight, $\frac{\pi_t(x_{t-1})}{\pi_{t-1}(x_{t-1})}$ is composed of d independent terms, and therefore the variance of the weights scales with a constant factor of d. Although the states x_{t-1}

may not be strictly independent, the scaling can be assumed to be close to linear, which is still much less severe than the exponential scaling obtained without annealing distributions.

Algorithm 3 Annealed Importance Sampling

```
1. For all samples, $i = 1, \cdots, N$, initialise $x_0^{(i)} \sim \pi_0$
   For each $i \in 1, \cdots N$, let with $x = x^{(i)}$
   (a) For $t \in 1, \cdots, T$:
       i. Annealing procedure: generate $x_t$ using transition distribution $K_t(\cdot, x_{t-1})$
       ii. Calculate the importance weight at transition $t$,
           $$w_t = \frac{\pi_t(x_{t-1})}{\pi_{t-1}(x_{t-1})} \quad (2.78)$$
           where
           $$\pi_t(x) = \pi_T(x)^{\frac{t}{T}} \pi_0(x)^{\frac{T-t}{T}} \quad (2.79)$$
   (b) Calculate the annealed importance weight
       $$w_{AIS}^{(i)} = \prod_{t}^{T-1} w_t \quad (2.80)$$
   (c) Let $x^{(i)} = x_T$
2. For $i = 1, \cdots, N$, we now have samples $x^{(i)} \sim \pi_T$, and their associated annealed importance
   weights, $w_{AIS}^{(i)}$. The normalising constant can be calculated with $\hat{Z} = \sum_{i=1}^{N} w_{AIS}^{(i)}$.
```

2.6.2 Sequential Monte Carlo

Sequential Monte Carlo (SMC) methods are a type of Monte Carlo algorithm designed for sampling from a sequence of probability distributions. They take a similar approach to AIS, but where we introduce intermediate distributions to slowly transition from one distribution to another. One of the key differences between the two is that SMC introduces the idea of resampling, where appropriate, to ensure the samples are spread fairly across the distribution of interest.

SMC originates from **particle filter** methods, designed to estimate an unknown state vector, given partial and/or noisy observations (Gordon et al., 1993). Particle filters are based on the idea that the sequence of probability distributions describes the evolution of a system in time, leading to the index notation $t = 1, \cdots, T$ (Liu and Chen, 1998). The target probability distribution is the state at the most recent time $t = T$, given all observations up until this point, $\pi_T(x_T|y_1, \cdots, y_T)$. The evolution of the states can be filtered or smoothed, given available observations, and furthermore, successive predictions can be made by extending the time series to $T+1$ and sampling $x_{T+1} \sim \pi_{T+1}(\cdot)$. SMC adapts the methods of particle filtering to sample from a probability distribution of fixed dimension, π_T, where the sequence of probability distributions are defined by an MCMC transition kernel, as seen in AIS above (Del Moral et al., 2006).

We will use the Sequential Monte Carlo sampler presented in Del Moral et al. (2006). We start by defining the sequence of probability distributions, on a common space, $\pi_0, \cdots, \pi_T$ where π_0 is the prior distribution and π_T is the target distribution. Each of these is known only up to a normalising constant. First, at time $t = 0$, we sample N_P independent points from π_0 and we label these $x_0^{(p)} \sim \pi_0$, where $p \in 1, \cdots, N_P$. Based on the origins in particle filters, it is common to refer to these samples as 'particles'. Throughout this, we will use superscripts to refer to the particles and subscripts to refer to the intermediate distributions. Each particle is assigned a weight that will be tracked throughout this algorithm. We will use w to indicate an **unnormalised weight** and W to indicate the **normalised weight**,

$$W^{(p)} = \frac{w^{(p)}}{\sum_{p=1}^{N_P} w^{(p)}} \, . \tag{2.81}$$

The weights are typically initialised from a uniform distribution, so that all particles p have normalised weight $W_0^{(p)} = \frac{1}{N_P}$. We then carry out the following steps for iteration $t = 1, \cdots, T$.

The path of all particles is extended to the next iteration at time t by letting samples from π_{t-1} approximate the next distribution π_t. The **incremental importance weight**, meaning the contribution of the importance weight at this iteration, is calculated as the ratio of the distributions π_{t-1} and π_t

$$\widetilde{w}_t^{(p)} = \frac{\pi_t(x_{t-1}^{(p)})}{\pi_{t-1}(x_{t-1}^{(p)})} \tag{2.82}$$

for each particle p. This is used to update the unnormalised weight $w_t = \widetilde{w}_t w_{t-1}$. This is equivalent to the product of incremental weights calculated at each transition in AIS (Equation

(2.75)). This first step is called **reweighting**.

Before carrying out the MCMC move to update x_t, we consider the variance of the normalised weights which tends to increase at each iteration as the distance from the initial distribution increases (i.e. the discrepancy between π_t and π_0 increases with t). This can lead to a potential degeneracy of particle approximation as some particles may be assigned very low weights and few (or even just one) may have significant weighting, dominating the results. We measure this with the Effective Sample Size (ESS), defined as

$$ESS = \frac{1}{\sum_{p=1}^{N_P}(W^{(p)})^2} \tag{2.83}$$

where $W^{(p)}$ is the normalised weight of particle p calculated from Equation (2.81). The ESS describes how many of the N_P samples are different enough from each other to be relevant and should ideally be close to N_P. For a value much lower than the total number of samples, the degeneracy of particles is high. If $ESS < \alpha$ for a pre-specified threshold, say $\alpha = N_P/2$ we carry out a process called **resampling**. This involves sampling N_P new particles from the weighted distribution $\pi_t W_t^{(p)}$ so that the set of particles are distributed according to the weights. The result is that there are multiple copies of particles with high weights while particles with low weights are discarded. Different approaches can be taken but we will use stratified sampling which ensures a roughly even spread of particles across the cumulative distribution of weights (Kitagawa, 1996). Then all resampled particles are assigned even weights $1/N_P$. At the moment of resampling, the number of distinct particles is decreased and therefore the Monte Carlo variance of the estimator is increased. However, resampling is carried out to enable a more diverse sample and a lower Monte Carlo variance in future iterations (Johansen and Evers, 2010). It is therefore not desirable to carry out resampling at every iteration, which is why we monitor the ESS at each step, and resample only if it falls below a certain threshold.

The third step is to **move** the particles using the MCMC move defined by the transition kernel $K_t(x_{t-1}, x_t)$, as done in annealed importance sampling. This gives a new set of particles $x_t^{(p)}$ distributed according to π_t. These three steps to 'reweight', 'resample' and 'move' all particles are repeated until $t = T$. Figure 2.4 shows how these three steps play out on 10 particles that evolve from step $t-1$ to step t (Doucet et al., 2001). This highlights the importance weights, represented in blue, which are calculated according to the intermediate distribution and used to redistribute particles in the resampling step, before the MCMC move is carried out. Algorithm 4 describes this SMC sampler.

We also have an estimate of the normalising constant at each iteration with $\sum_{p=1}^{N_P} w_t^{(p)}$, calculated in the renormalisation step. The final value of this is an estimate of the normalising constant,

$$\widehat{Z} = \sum_{p=1}^{N_P} w_T^{(p)} = \sum_{p=1}^{N_P} \prod_{t=0}^{T} w_t^{(p)} \tag{2.84}$$

which is equivalent to Equation (2.77) in AIS.

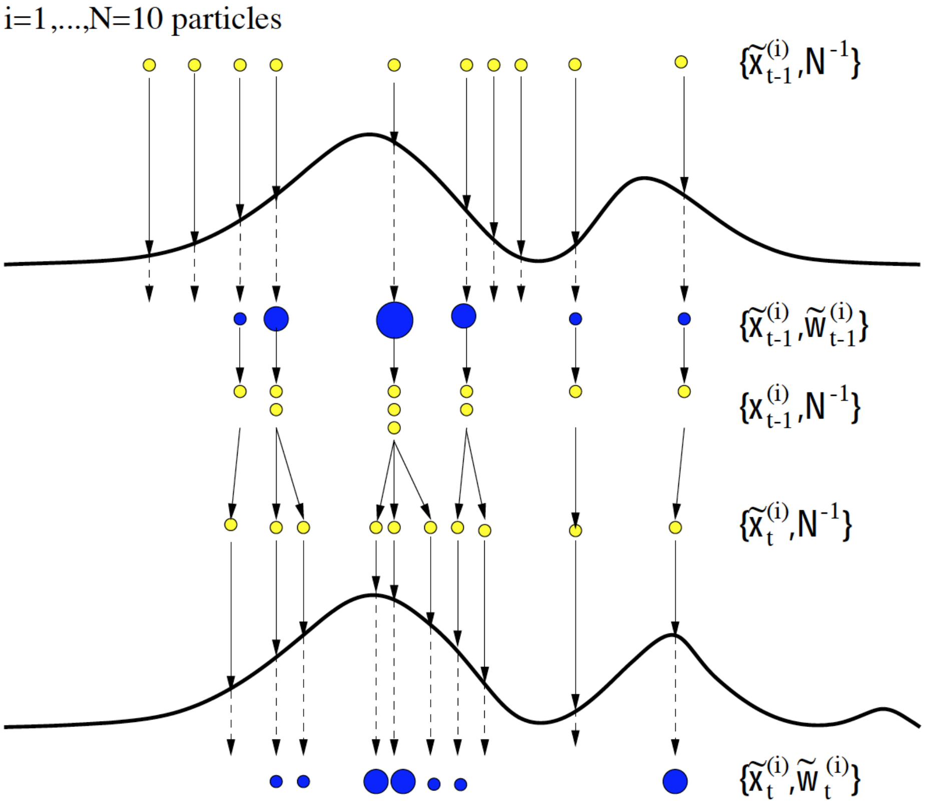

i=1,...,N=10 particles
$\{\widetilde{x}_{t-1}^{(i)}, N^{-1}\}$
$\{\widetilde{x}_{t-1}^{(i)}, \widetilde{w}_{t-1}^{(i)}\}$
$\{x_{t-1}^{(i)}, N^{-1}\}$
$\{\widetilde{x}_{t}^{(i)}, N^{-1}\}$
$\{\widetilde{x}_{t}^{(i)}, \widetilde{w}_{t}^{(i)}\}$

Figure 2.4: Reproduced from Doucet et al. (2001) showing SMC for $N = 10$ particles in yellow with weights in blue. Starting from the top at time $t-1$, these particles are weighted according the intermediate distribution, then resampled according to these weights and then moved according to an MCMC move. This process is then repeated.

SMC algorithms share many similar concepts to AIS since they both rely on constructing a sequence of probability distributions, which can be chosen so that we start with samples from

Algorithm 4 SMC sampler for sequence of probability distributions defined on common space

```
1. For all particles, $p = 1, \cdots, N_P$, initialise $x_0^{(p)} \sim \pi_0$ and $W_0^{(p)} = \frac{1}{N_P}$
2. For each $t \in 1, \cdots, T$:
   (a) Reweight all particles $p = 1, \cdots, N_P$ with
           $w_t^{(p)} = \widetilde{w}_t^{(p)} w_{t-1}^{(p)} = \frac{\pi_t(x_{t-1}^{(p)})}{\pi_{t-1}(x_{t-1}^{(p)})} w_{t-1}^{(p)}$
   (b) Resample if necessary
       i. Renormalise all particles $p = 1, \cdots, N_P$ with
           $W_t^{(p)} = \frac{w_t^{(p)}}{\sum_{p=1}^{N_P} w_t^{(p)}}$
       ii. Calculate ESS with Equation (2.83)
           $ESS = \frac{1}{\sum_{p=1}^{N_P} (W_t^{(p)})^2}$
       iii. If $ESS < \alpha$ resample with stratified sampling and set $W_t^{(p)} = \frac{1}{N_P}$
   (c) Move all particles $p = 1, \cdots, N_P$
           $x_t^{(p)} \sim K_t(x_{t-1}, \cdot)$
```

a prior distribution, π_0 and slowly move transition to a posterior distribution, π_T. The main difference is resampling step in SMC, which aims to keep diversity in the particles and reduce degeneracy in the final sample at $t = T$. However, this degeneracy is not eliminated entirely, but rather shifted to an earlier point in the trajectory (Johansen and Evers, 2010). This is because the process of resampling at time t involves replicating particles, which then initially follow similar paths, creating a degeneracy at time t. This can be a problem if the full history of the particles is of interest, but overall it is beneficial when the goal is sampling from π_T.

The complexity of the SMC algorithm is $O(N)$ and the reweighting and move steps can be paralellised easily. The bottleneck for a completely parallelised algorithm is the process of resampling as this requires summing all the particle weights in renormalisation which cannot be parallelised (Del Moral et al., 2006).

2.6.3 Adaptive SMC

Constructing the intermediate probability distributions $\pi_1, \cdots, \pi_{T-1}$ can be a challenge. We can use the same form as AIS, i.e.

$$\pi_t(x) = \pi_T(x)^{\gamma_t} \pi_0(x)^{1-\gamma_t} \tag{2.85}$$

where π_0 is the prior distribution, π_T is the posterior distribution and γ_t takes values increasing from 0 to 1. However, there is often little information available as to how many intermediate distributions should be taken (T) or how to choose the spacing of distributions (γ_t).

We can choose an **adaptive** sequence of intermediate distributions to select the value of γ_t at each iteration (Zhou et al., 2016). The aim of the intermediate distributions, π_t is to bridge the gap between the previous distribution π_{t-1} and π_T, with as few as distributions as possible to reduce computational cost. Choosing γ_t too far apart leads to a reduction in ESS as many points can become degenerate, while choosing them to be too close together is more accurate but more computationally costly. In adaptive SMC we can choose γ_t based on a trade-off between moving as far as possible from π_{t-1} (and as close as possible to our target distribution π_T)

without allowing the ESS to drop below a certain level. We do this based on the conditional ESS (CESS), defined by

$$CESS(\pi_t) = \frac{N_P\Big(\sum_p (w_{t-1}^{(p)} \widetilde{w}_t^{(p)})\Big)^2}{\sum_p \Big((w_{t-1}^{(p)})(\widetilde{w}_t^{(p)})^2\Big)} \tag{2.86}$$

The CESS describes how close neighbouring distributions are. When the CESS is close to N_P, the neighbouring distributions are close to each other, which should give higher and more equal weights. We set a pre-specified minimum threshold for the CESS, βN_P so that β is close to 1, such as 0.9. We then choose the γ_t as large as possible so that $CESS \geq \beta N_P$.

As before, $\widetilde{w}_t$ is the incremental importance sampling weight that weights points based on the discrepancy between π_t and π_{t-1}. This can be written in terms of γ_t and γ_{t-1} as

$$\begin{aligned}
\widetilde{w}_t^{(p)} &= \frac{\pi_t(x_{t-1}^{(p)})}{\pi_{t-1}(x_{t-1}^{(p)})} \\
&= \frac{\Big(\pi_0(x_{t-1}^{(p)})\Big)^{\gamma_t}\Big(\pi_T(x_{t-1}^{(p)})\Big)^{1-\gamma_t}}{\Big(\pi_0(x_{t-1}^{(p)})\Big)^{\gamma_{t-1}}\Big(\pi_T(x_{t-1}^{(p)})\Big)^{1-\gamma_{t-1}}} \\
&= \Big(\pi_0(x_{t-1}^{(p)})\Big)^{\gamma_t-\gamma_{t-1}}\Big(\pi_T(x_{t-1}^{(p)})\Big)^{-\gamma_t+\gamma_{t-1}}
\end{aligned}$$

Since we require $CESS(\gamma_t) - \beta N_P \geq 0$ in practice, we find for the value of γ_t that gives $CESS(\gamma_t) - \beta N_P$.

This process is cheap, as we do not need to evaluate expensive likelihood functions in π_t at each step. We can simply evaluate π_0 and π_T once, and then repeatedly calculate $\widetilde{w}_t^{(p)}$ for different γ_t, to find the value that satisfies $CESS(\gamma_t) - \beta N_P$. This only slightly changes Algorithm 4. Instead of pre-specifying the number of iterations in step 2, we evaluate the equation and choose a new γ_t. This is carried out at each step until $\gamma_t = 1$, at which point we have achieved the target distribution. This equates to replacing the loop 'for each $t \in 1, \cdots, T$' with 'while $\gamma_t < 1$' and adding an additional step 2(d) where the next value γ_{t+1} is calculated. This step is included in the algorithm presented in the following section (Algorithm 5).

The idea behind choosing γ_t adaptively is similar to the way in which we carry out adaptive resampling to keep ESS above a pre-specified threshold in Equation (2.83), as both are concerned with maintaining a large effective sample size. Note that the $CESS$ is equal to the ESS when resampling is carried out at every iteration (Zhou et al., 2016).

2.6.4 Transformation SMC

In RJMCMC (Section 2.5), I introduced the concept of the probability distributions of interest that exist on different spaces of different dimensions. The same can be true in an SMC algorithm, as it can be designed to transition from probability distribution π_k to a new probability distribution π_{k+1} that lies in a different space. Chapter 5 makes use of a method called **transformation SMC** (TSMC) presented in Everitt et al. (2020), where the transition from one probability distribution to another is defined by transformations, as done in RJMCMC. This is done with an SMC sampler as discussed above (Del Moral et al., 2006) set up on the **extended space** which is the space with the maximum dimensions of the variable of interest.

Here, we will start when $t = 0$ in model k, with parameter values $(\theta^{(k)}, u^{(k)})$. The target distribution of interest at $t = T$ is model $k+1$, with parameter values $(\theta^{(k+1)}, u^{(k+1)})$. In other words, $\pi_0 = \pi_k$ and $\pi_T = \pi_{k+1}$. As before, we will construct an SMC sampler to bridge between these two distributions, π_t for $t = 1, \cdots, T-1$. The key difference, however, is that we must introduce a first step to **transform** the initial particle, with parameter value $(\theta^{(k)}, u^{(k)})$, from model k to a projection into the same space as model $k+1$. This involves sampling $u^{(k+1)} \sim \psi_{k \to k+1}(\cdot)$ followed by applying a deterministic function to obtain $\theta^{(k+1)}$, as described in RJMCMC, Section 2.5.

The remaining steps of reweighting, resampling and moving particles remains the same, but must be outlined carefully to ensure dimension matching between the spaces. Following the same notation as RJMCMC in Section 2.5, the goal is move parameter values from model k, with probability distribution

$$\pi_k(\theta^{(k)}, u^{(k)}) = \varphi_k(\theta^{(k)})\, \psi_{(k \to k+1)}(u^{(k)}) \tag{2.87}$$

to model $k+1$, with probability distribution

$$\pi_{k+1}(\theta^{(k+1)}, u^{(k+1)}) = \varphi_{k+1}(\theta^{(k+1)})\, \psi_{(k+1\to k)}(u^{(k+1)}) \left| \frac{\partial(\theta^{(k+1)}, u^{(k+1)})}{\partial(\theta^{(k)}, u^{(k)})} \right| \tag{2.88}$$

where the last term is the Jacobian which ensures dimension matching between the two distributions. If we were to move directly from π_k to π_{k+1}, the incremental weight update would be

$$\begin{aligned} \widetilde{w}_{k+1} =& \frac{\pi_{k+1}(\theta^{(k)}, u^{(k)})}{\pi_k(\theta^{(k)}, u^{(k)})} \\ =& \frac{\varphi_{k+1}(\theta^{(k+1)})\, \psi_{(k+1\to k)}(u^{(k+1)}) \left| \frac{\partial(\theta^{(k+1)}, u^{(k+1)})}{\partial(\theta^{(k)}, u^{(k)})} \right|}{\varphi_k(\theta^{(k)})\, \psi_{(k\to k+1)}(u^{(k)})} \,. \end{aligned} \tag{2.89}$$

Here, we have dropped the notation indicating the particle index p.

As before, we introduce the sequence of probability distributions designed to bridge the gap between π_k and π_{k+1}. These intermediate distributions are indexed t. These must be constructed on a space of fixed dimension, so we will use the larger dimensional space, $(\theta^{(k+1)}, u^{(k+1)})$ to define these. Following equation 2.73 where we defined intermediate distributions $\pi_t(x)$ this gives

$$\pi_{k\to k+1;t}(\theta^{(k+1)}, u^{(k+1)}) = \left(\pi_{k+1}\left(\theta^{(k+1)}, u^{(k+1)}\right)\right)^{\gamma_t} \left(\pi_k\left(\theta^{(k)}, u^{(k)}\right)\right)^{1-\gamma_t} \tag{2.90}$$

Combining this with Equation (2.82), the incremental weight update when moving from intermediate distribution $t-1$ to t, represented entirely in the common space $(\theta^{(k+1)}, u^{(k+1)})$ is

$$\widetilde{w}_t = \frac{\pi_{k\to k+1;t}(\theta_{t-1}^{(k+1)}, u_{t-1}^{(k+1)})}{\pi_{k\to k+1;t-1}(\theta_{t-1}^{(k+1)}, u_{t-1}^{(k+1)})} \,. \tag{2.91}$$

Writing this out in full, using the definition of π_k, π_{k+1} and $\pi_{k\to k+1;t}$ (Equations (2.87), (2.88) and (2.90) respectively), gives

$$\begin{aligned}
\widetilde{w}_t =& \frac{\left(\pi_{k+1}\left(\theta_{t-1}^{(k+1)}, u_{t-1}^{(k+1)}\right)\right)^{\gamma_t} \left(\pi_k\left(\theta_{t-1}^{(k)}, u_{t-1}^{(k)}\right)\right)^{1-\gamma_t}}{\left(\pi_{k+1}\left(\theta_{t-1}^{(k+1)}, u_{t-1}^{(k+1)}\right)\right)^{\gamma_{t-1}} \left(\pi_k\left(\theta_{t-1}^{(k)}, u_{t-1}^{(k)}\right)\right)^{1-\gamma_{t-1}}} \\
=& \left(\frac{\pi_{k+1}\left(\theta_{t-1}^{(k+1)}, u_{t-1}^{(k+1)}\right)}{\pi_k\left(\theta_{t-1}^{(k)}, u_{t-1}^{(k)}\right)}\right)^{\gamma_t-\gamma_{t-1}} \\
=& \left(\frac{\varphi_{k+1}(\theta_{t-1}^{(k+1)})\, \psi_{(k+1\to k)}(u_{t-1}^{(k+1)}) \left|\frac{\partial(\theta_{t-1}^{(k+1)}, u_{t-1}^{(k+1)})}{\partial(\theta_{t-1}^{(k)}, u_{t-1}^{(k)})}\right|}{\varphi_k(\theta_{t-1}^{(k)})\, \psi_{(k\to k+1)}(u_{t-1}^{(k)})}\right)^{\gamma_t-\gamma_{t-1}} .
\end{aligned} \tag{2.92}$$

This is used in the reweighting step of the algorithm. Resampling is carried out, if necessary, in the same manner as Algorithm 4. Finally, an MCMC move is applied to the particles that targets $\pi_{k+1;t}$. We write this as $K_{k+1;t}((\theta_{t-1}^{(k+1)}, u_{t-1}^{(k+1)}), (\theta_t^{(k+1)}, u_t^{(k+1)}))$. Algorithm 5 outlines the complete process in moving from k to $k+1$, where we also use the adaptive sampling method described above. This can be repeated, to move through different dimensional spaces where $k = 1, \cdots, K$. This algorithm can also be reversed to move particles down in dimension, i.e. $k = K, \cdots, 1$. Furthermore, the normalising constant, Equation (2.77) can be tracked through the entire process, which means we can compare $\hat{Z}$ for different values of k, even when the dimensions differ. This will be used in Chapter 5 as a method of model selection.

Algorithm 5 tSMC sampler increasing in dimension from k to $k+1$, with adaptive intermediate distributions

```
1. Start with particles, $p = 1, \cdots, N_P$, $(\theta_0^{(k)}, u_0^{(k)})^{(p)} \sim \pi_k$ and $W_0^{(p)} = \frac{1}{N_P}$
2. **Transform** all particles $p = 1, \cdots, N_P$, transform $(\theta_0^{(k)}, u_0^{(k)})^{(p)}$ into $k+1$ model space,
   with $G_{k \to k+1}$ to obtain $(\theta_0^{(k+1)}, u_0^{(k+1)})^{(p)}$.
3. Set $\gamma_t = 0$ and $t = 0$
4. While $\gamma_t < 1$:
   (a) **Reweight** all particles $p = 1, \cdots, N_P$ with
           $w_t^{(p)} = \widetilde{w}_t^{(p)} w_{t-1}^{(p)}$
       where $\widetilde{w}_t^{(p)}$ is given by Equation (2.89).
   (b) **Resample** if necessary
       i. Renormalise all particles $p = 1, \cdots, N_P$ with
              $W_t^{(p)} = \frac{w_t^{(p)}}{\sum_{p=1}^{N_P} w_t^{(p)}}$
       ii. Calculate ESS with Equation (2.83)
              $ESS = \frac{1}{\sum_{p=1}^{N_P} (W_t^{(p)})^2}$
       iii. If $ESS < \alpha$ resample with stratified sampling and set $W_t^{(p)} = \frac{1}{N_P}$
   (c) **Move** all particles $p = 1, \cdots, N_P$
           $(\theta_t^{(k+1)}, u_t^{(k+1)})^{(p)} \sim K_t((\theta_{t-1}^{(k+1)}, u_{t-1}^{(k+1)})^{(p)}, \cdot)$
   (d) Calculate next $\gamma_t \in (0, 1]$ such that $CESS(\gamma_t) = \beta N_P$ and set $t = t + 1$.
```

Chapter 3

Long-term Climate Response Prediction with Machine Learning

3.1 Introduction

In this chapter, I will discuss the construction of a machine learning approach for predicting the long-term climate response patterns from the short-term climate response based on GCM simulations. This work is published in Mansfield et al. (2020).

The aim of this chapter is to explore the use of data-driven methods to predict the climate response to a variety of greenhouse gas and aerosol forcing scenarios, rather than the conventional approach of running highly expensive GCMs (Section 1.2.1). To achieve long-term climate change mitigation and adaptation goals, such as limiting global warming to 1.5 or 2 °C, there must be a global effort to decide and act upon effective but realistic emission pathways (Pachauri et al., 2014). This requires an understanding of the consequences of such pathways, which are often diverse and involve changes in multiple climate forcers (Pachauri et al., 2014; Collins et al., 2012; Rogelj et al., 2013). In particular, different emission scenarios of, for example, greenhouse gases and aerosols are responsible for diverse changes in regional climate, which are not always well captured by a metric such as global temperature-change potential (Shindell and Faluvegi, 2009; Shine et al., 2005; Aamaas et al., 2017; Collins et al., 2013b).

Exploring more detailed relationships between emissions and multi-regional climate responses still requires the application of Global Climate Models (GCMs) that allow the behaviour of the climate to be simulated under various conditions on decadal to multi-centennial timescales (e.g. different atmospheric greenhouse gas and aerosol concentrations or emissions fields, Section 1.2.1). However, modelling climate at increasingly high spatial resolutions has significantly increased the computational complexity of GCMs (Collins et al., 2012), a tendency that has been accelerated by the incorporation and enhancement of a number of new Earth system model components and processes (Williams et al., 2018; Walters et al., 2019; Storkey et al., 2018; Ridley et al., 2018). This high computational cost drives this investigation into how machine learning methods can help accelerate estimates of global and regional climate change under different climate forcing scenarios.

Here, we seek a fast ‘surrogate model’ to find a mapping from short-term to long-term response patterns within a given GCM. Once learned, this surrogate model can be used to rapidly predict other outputs (long-term responses) given new unseen inputs (short-term responses i.e. the results of easier to perform short-term simulations). Data science methods are increasingly used within climate science (e.g. Bracco et al., 2018; Kretschmer et al., 2017; Nowack et al., 2018; Sippel et al., 2019; Runge et al., 2019; Knüsel et al., 2019; Nowack et al., 2020), and furthermore, emulators that approximate model output given specific inputs, are already a popular tool of choice for prediction, sensitivity analysis, uncertainty quantification and calibration (Ryan et al., 2018; Wild et al., 2020; Williamson et al., 2015; Salter and Williamson, 2016; Lee et al., 2011, 2016; Rougier et al., 2009; McNeall et al., 2013; Edwards et al., 2019; Castruccio et al., 2014b). These methods have great potential for climate change projection and impact studies. Yet long-term, spatially resolved climate prediction for diverse forcings has not yet been addressed, predominantly because of the lack of appropriate training simulations, which in the past were expensive to obtain. I use a unique dataset of existing climate model simulations to train machine learning models and learn relationships between short-term and long-term temperature responses to different climate forcing scenarios. This promises to drastically accelerate climate change projections, by reducing the costs of additional scenario computations as it requires only the short-term response from a GCM. This is motivated by previous studies that find that the temperature response is often projected onto the same pattern (Kasoar et al., 2018; Levy et al., 2013; Xie et al., 2013) and that links can be made between short- and long-term response (Ceppi et al., 2017; Persad et al., 2018). Furthermore, the machine learning methods discussed here can also be coupled with the emulator presented in Chapter 4, which would further increase the speed of climate change projections, from emissions to global patterns of long-term response.

Here, I present two approaches to building the surrogate model, namely ridge regression and Gaussian process regression. Throughout this section, I will discuss the successes and challenges of these methods and compare them against pattern scaling (described in Section 1.2.2). I also present some of the different set-ups for building the surrogate model, such as dimension reduction and different inputs. Rather than treating machine learning methods purely as a 'black-box', one may wish to uncover how the outputs are learned, a topic of interest in the climate community in recent years (Barnes et al., 2019, 2020; Runge et al., 2019; Kuhn-Régnier et al., 2020). I explore this by analysis of ridge regression coefficients in Section 3.7. Finally, I will outline some of the challenges for the future of emulation of long-term climate response, particularly with respect to potential performance gains by incorporating larger training datasets, as highlighted in Mansfield et al. (2020).

3.2 Data

To train our learning algorithms, I take advantage of a unique set of GCM simulations performed in recent years using the Hadley Centre Global Environment Model 3 (HadGEM3), described in Section 1.2.1. The simulations were run in previous academic studies and model intercomparison projects, namely the Precipitation Driver and Response Model Intercomparison Project (PDRMIP) (Myhre et al., 2017; Liu et al., 2018b; Samset et al., 2016), Evaluating the Climate and Air Quality Impacts of Short-lived pollutants (ECLIPSE) (Stohl et al., 2015; Aamaas et al., 2017; Baker et al., 2015) and Kasoar et al. (2018, 2016); Shawki et al. (2018). In these, step-wise perturbations were applied to various forcing agents to explore characteristic short- and long-term climate responses to them. There are 21 simulations in total for a range of forcings, including long-lived greenhouse gas perturbations (e.g. carbon dioxide (CO_2), methane (CH_4), CFC-12), short-lived pollutant perturbations (e.g. sulfur dioxide emissions (SO_2, the precursor to sulfate aerosol (SO_4)), black carbon (BC), organic carbon (OC)) and a solar forcing perturbation. As described in Section 1.1.2, the forcing types differ in that the long-lived forcers are homogeneously distributed in the atmosphere so that the region of emission is effectively inconsequential for the global temperature response pattern, while the response to short-lived forcers is highly dependent on the emission region. Therefore the long-lived forcing simulations are global perturbations, while for the short-term pollutants, both global and regional perturbations exist, to account for the influence of emission region to the response (Shindell and Faluvegi, 2009; Shindell et al., 2012). Table 3.1 shows all the scenarios with their source, name and description.

The long-lived greenhouse gas (CO_2, CH_4, CFC-12) simulations were performed by altering the global atmospheric concentrations, since these pollutants are well-mixed throughout the atmosphere. However, the short-lived pollutant concentration can vary significantly across the globe and therefore the model relies on spatial fields as inputs (either concentration fields or emissions fields). For these perturbations, emission fields were scaled abruptly from the present-day baseline in experiments performed by ECLIPSE (Stohl et al., 2015; Aamaas et al., 2017; Baker et al., 2015) and Kasoar et al. (2018, 2016); Shawki et al. (2018), while the atmospheric concentration fields were scaled abruptly in the PDRMIP experiments (Myhre et al., 2017; Liu et al., 2018b; Samset et al., 2016). The solar forcing experiment was performed by changing the solar irradiance constant (Myhre et al., 2017). The GCM was run until it converges towards a new climate state, to reach an approximate equilibrium (70-100 years[1]). Independent control simulations were run for each project and the 'response' is calculated by differencing the outputs from the perturbation simulations with the outputs from their corresponding control simulation. The evolution of the GCM's global mean temperature response to some example forcing scenarios is highlighted in Figure 3.1a. All scenarios show an initial sudden response in the first few years, which we label the 'short-term response'. The global mean temperature then converges towards a new (approximately) equilibrated steady state, which we label the 'long-term response'.

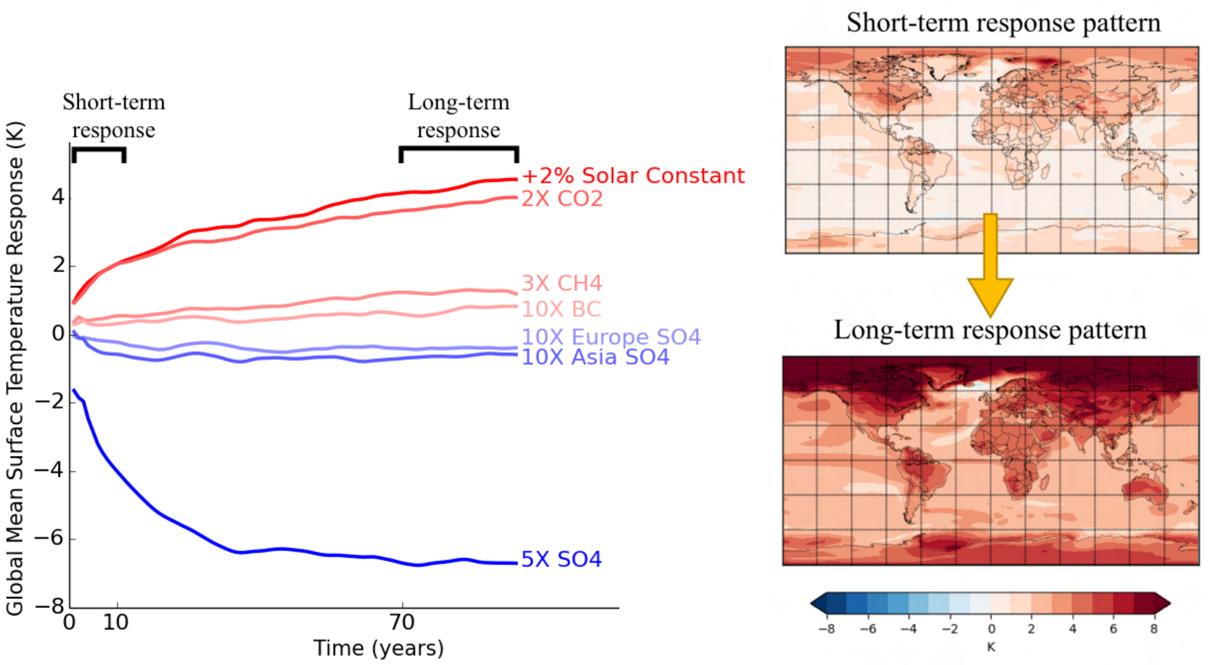
[1] Simulations produced by Kasoar et al. (2018, 2016); Shawki et al. (2018) were run for up to 200 years, but 70-100 were taken here for consistency

Table 3.1: Table of scenarios by short name and description, categorised by the source which generated the data.

Short name	Description
Source: PDRMIP, Myhre et al. (2017); Liu et al. (2018b); Samset et al. (2016)	
$2xCO_2$_PDRMIP	Global doubling of carbon dioxide concentration (PDRMIP)
$3xCH_4$	Global tripling of methane concentration
10xCFC-12	10x increase in global chloroflorocarbon-12 concentration
+2%_Solar_Constant	2% increase in solar forcing
$5xSO_4$_Global	5x increase in sulfate aerosol concentration globally
10xBC_Global	10x increase in black carbon concentration globally
$10xSO_4$_Europe	10x increase in sulfate aerosol concentration over Europe
$10xSO_4$_Asia	10x increase in sulfate aerosol concentration over Asia
10xBC_Asia	10x increase in black carbon concentration over Asia
SO_4_pre-industrial	Pre-industrial sulfate levels
Source: ECLIPSE, Stohl et al. (2015); Aamaas et al. (2017); Baker et al. (2015)	
$2xCO_2$_ECLIPSE	Global doubling of carbon dioxide concentration (ECLIPSE)
20%_CH_4	20% reduction in methane emissions globally
No_BC_Global	100% reduction in black carbon emissions globally
No_SO_2_Global	100% reduction in sulfur dioxide emissions globally
No_CO_Global	100% reduction in carbon monoxide emissions globally
Source: Kasoar et al. (2018, 2016); Shawki et al. (2018)	
No_SO_2_NHML	100% reduction in sulfur dioxide emissions over the Northern Hemisphere mid-latitudes
No_SO_2_China	100% reduction in sulfur dioxide emissions over China
No_SO_2_East_Asia	100% reduction in sulfur dioxide emissions over East Asia
No_SO_2_Europe	100% reduction in sulfur dioxide emissions over Europe
No_SO_2_US	100% reduction in sulfur dioxide emissions over the US
No_BC_NHML	100% reduction in black carbon emissions over the Northern Hemisphere mid-latitudes

For the long-term response, we discard the transient response and average from year 70 to 100 for simulations from PDRMIP and Kasoar et al. (2018) to smooth out internal variability over the 30-year period, as done in previous studies (Kasoar et al., 2016, 2018; Mitchell, 2003). For the 5 ECLIPSE simulations, we average from year 70 to year 80, since this is the full temporal extent of ECLIPSE simulations. For the short-term response, we average over the first 10 years of the simulation to reduce the influence of natural variability of the GCM (Mitchell, 2003). Note that these averages are over time and not space as we are interested in not just the global mean response but, more importantly, in the global response patterns. This results in spatial maps of short-term response and long-term response, such as the example shown in 3.1b for the $2xCO_2$ scenario. The goal of the statistical model presented here is to predict the behaviour of the full

GCM for a specific target climate variable of interest; here we choose surface temperature at each GCM grid-cell, a central variable of interest in climate science and impact studies.



(a) Smoothed timeseries for global mean response

(b) Map of short-term and long-term temperature response of GCM to $2xCO_2$ forcing

Additional simulations had also been performed by Shawki et al. (2018); Stohl et al. (2015); Aamaas et al. (2017); Baker et al. (2015), but we only consider simulations where the global mean temperature response exceeds the natural variability, calculated as the standard deviation among the control simulations. This is to limit the noise in the still relatively small dataset available. Scenarios that were not used for this reason were the global removals of organic carbon, volatile organic compounds and nitrogen oxides (ECLIPSE), the removal of BC in regional perturbations and the removal of SO_2 over India (Shawki et al., 2018).

To motivate the problem, I first consider correlations between the short-term and long-term responses on large scales. Figure 3.2 shows the relationship between short-term responses on the x-axis and long-term responses on the y-axis, both of which are averaged globally and regionally over Europe and East Asia, as example regions. This shows us not only local relationships between the short-term and long-term response over the same fixed region (Europe or East Asia), but also remote relationships between short-term response in Europe and the long-term response

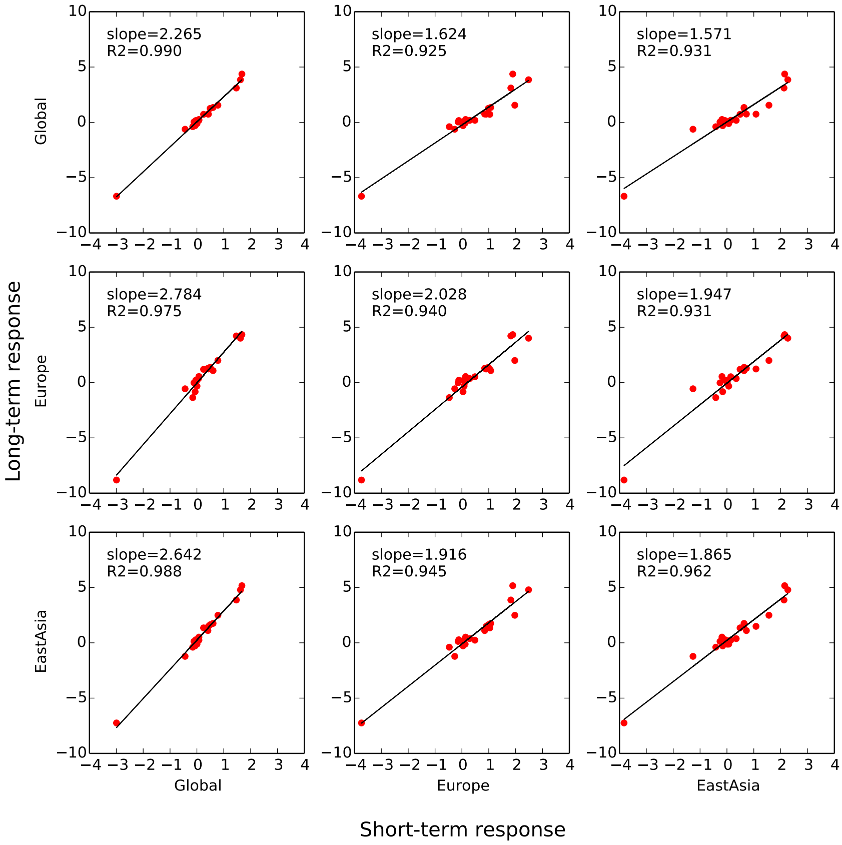


Figure 3.2: Relationships between short- and long-term responses for the data points averaged globally, over Europe and over East Asia. The black line shows linear least squares fit, with the slope and Pearson's coefficient of correlation, R^2, also shown.

in East Asia and vice-versa. Although these regions are distant, we find strong correlations between short- and long-term responses which can be exploited when building the machine learning models. This provides us with some initial evidence that it is sensible to build predictive models in this way, as well as motivation for the linear models presented in the following section.

3.3 Prediction Methods

With these simulations, we aim to learn the function $f(\mathbf{x})$ that maps the short-term responses $\mathbf{x}$, to the long-term responses, $\mathbf{y}$ (Figure 3.3). Both the short- and long-term responses are multi-dimensional, with a contribution from every grid point. We write these in vector form by flattening the longitude-latitude grid, i.e. $\mathbf{x} = (x_1, \cdots, x_p)$ and $\mathbf{y} = (p_1, \cdots, y_p)$, where the index refers to a specific grid-point and p is the total number of grid-points. We use an independent regression model of the long-term response for each grid-cell, i.e. Equation (2.2). Each one depends on the short-term response at all grid-cells, so that predictions are not only based on local information but can draw predictive capability from any changes in surface temperature worldwide. This regression is carried out independently for each output y_i for $i \in (1, p)$. This does not explicitly account for spatial correlations in the outputs as each grid-cell is treated separately, with no sharing of information between adjacent grid-cells. In the following results, we find that correlations between neighbouring grid-cells appear naturally due to the smoothness in the outputs of the training data, as found in other studies (Ryan et al., 2018; Wilkinson, 2010; Conti and O'Hagan, 2010). However, we will later introduce principal component analysis applied to the outputs (Section 3.6.2) which exploits relationships between the output grid and guarantees new predictions contains the same correlations These correlations can be exploited to reduce provides another benefit as it enforces correlations in the outputs, which are otherwise not guaranteed through the independent regression presented previously. For the methods described below, the outputs are scaled prior to training the model. The inputs are not scaled in the results presented below, although we explore the effect of scaling in Section 3.6.

We use Ridge regression and Gaussian Process Regression (GPR) as approaches for constructing this mapping. Ridge regression (Section 2.1.4) penalises large coefficients so that Equation (2.15) can be uniquely solved when there is limited training data available but high number of outputs ($N < p$). Cross-validation is used to learn the regularisation parameter, λ, with 3-folds (Section 2.1.6). We use the scikit-learn package in Python to do this (Pedregosa et al., 2011). As well as using this for prediction, we explore the underlying mechanics of the regression method, by examining the learned coefficients, β_i, corresponding to the short-term response at grid-point i. This indicates which regions most strongly influence the prediction for a particular output. We have also explored the use of LASSO regression (Section 2.1.5) which penalises some coefficients to be zero, although we find poorer performance, as well as a less intuitive model.

Gaussian process regression is implemented with an additive kernel, consisting of a linear kernel and a squared exponential kernel described in Section 2.1.7. The hyperparameters, which correspond to the slopes of the linear fit and the lengthscales and variances for the squared exponential kernel (Equation (2.24)), are fitted by maximising the likelihood with the BFGS

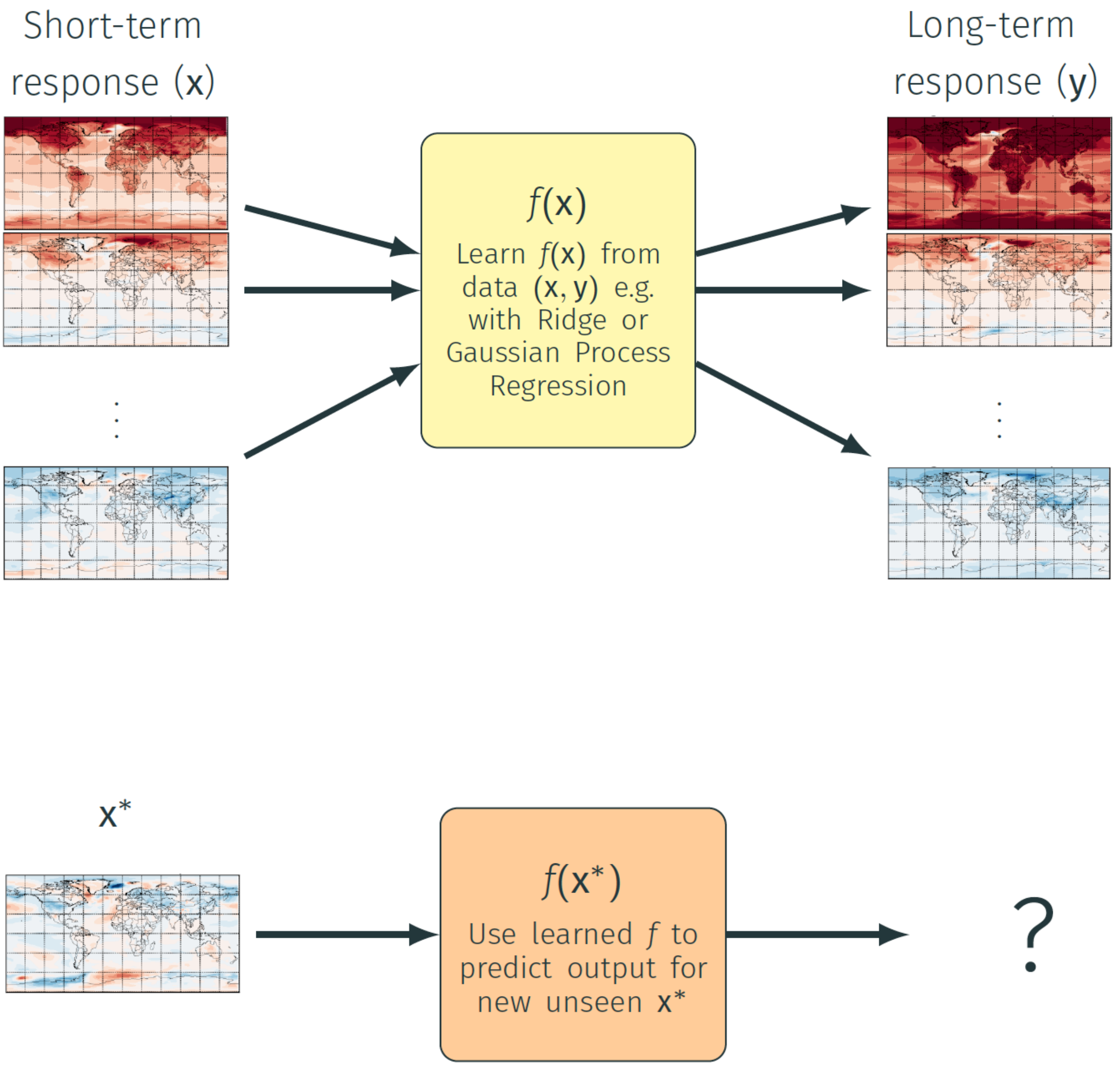

Short-term response (x)
Long-term response (y)
f(x)
Learn f(x) from data (x, y) e.g. with Ridge or Gaussian Process Regression
x*
f(x*)
Use learned f to predict output for new unseen x*
?

Figure 3.3: Training and prediction schematic. The training process learns $f(x)$ to map short-term responses to the long-term responses from all but one of the simulations. The prediction process uses the short-term response of the last simulation, currently unseen to $f(x)$, to predict the long-term response.

optimiser in the GPy package (GPy, 2014).

Both these approaches make the assumption of linearity, which is sensible based on Figures 3.2 and considering the limited data available. These two methods should handle well the limited sample size for training, which also limits how effectively the number of free parameters for other approaches such as deep learning, including convolutional neural networks, could be constrained. However, future data collaborations to increase dataset size could make deep leaning an option.

For the learning process, we use all but one of the available simulations at a time for training. Then, the learned regression functions can be used to predict the long-term response for new, unseen inputs ($\mathbf{x}^*$), (Figure 3.3), corresponding to the one remaining simulation that was left out. This remaining simulation is the test scenario. We assess the prediction skill of our machine learning models by comparing the predicted response maps $f(\mathbf{x}^*)$ to the results of the complex GCM simulations for this test scenario. This training and testing is repeated so that every simulation available is used as test scenario and is predicted based on the information learned from all other independent simulations. This is done to maximise the number of simulations available for training given the limited dataset and to replicate a typical use of predicting the response to a single scenario. In the following section, we show the results of both Ridge regression and GPR and also compare their predictions with pattern scaling predictions.

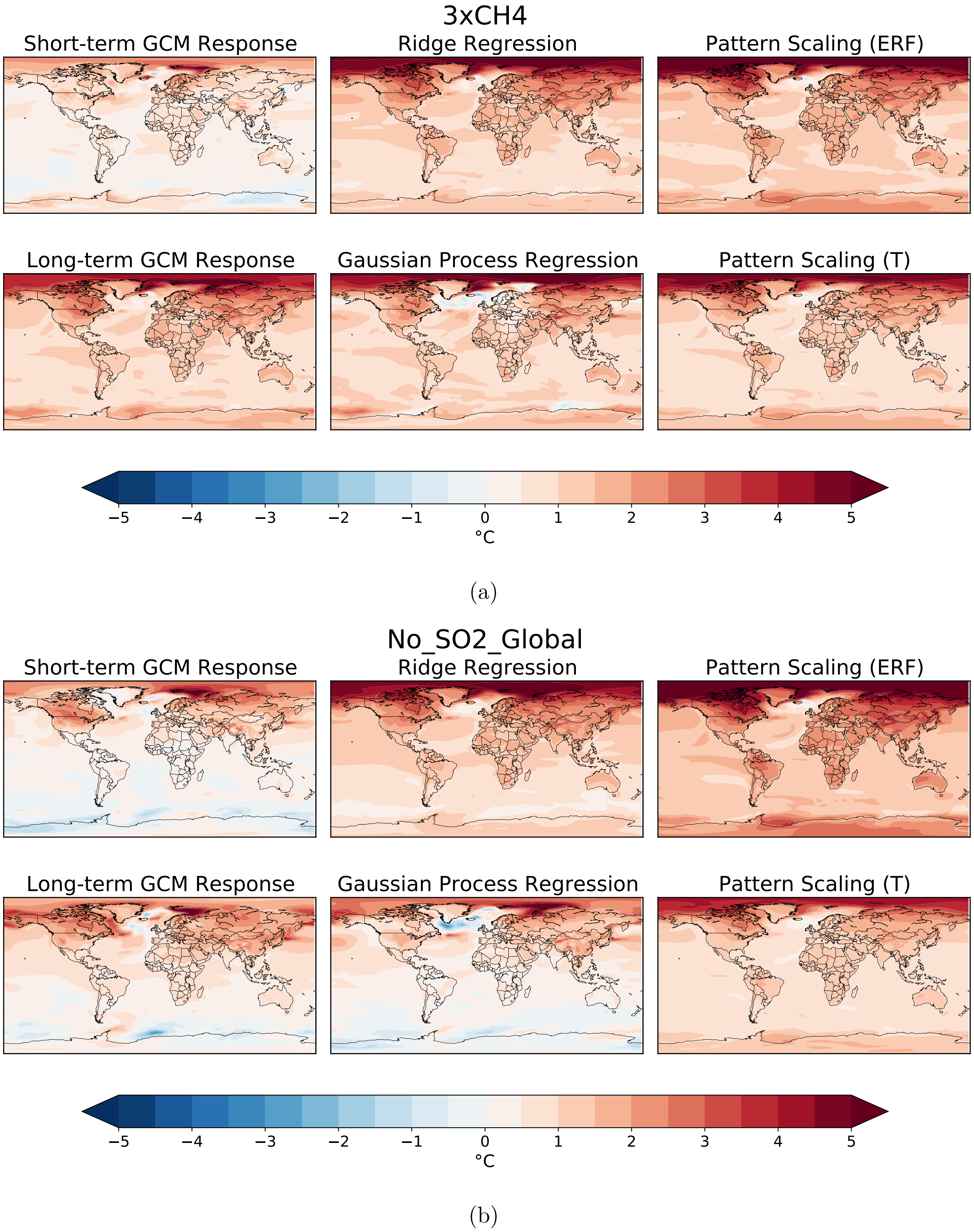
3.4 Performance

We evaluate the performance of the two different machine learning methods (Ridge, GPR) by benchmarking them against the traditional pattern scaling approach described in Section 1.2.2 which is often used for estimating future patterns of climate change (Santer et al., 1990; Mitchell, 2003; Hulme et al., 1995; Murphy et al., 2007; Watterson, 2008). For the reference pattern, the 2xCO_2 scenario (from PDRMIP (Myhre et al., 2017; Liu et al., 2018b; Samset et al., 2016)) is used. Pattern scaling is carried out using both ERF and the global mean short-term response as the scaler value s for determining the magnitude of the response (Equation (1.1)). Although the latter is not the typical approach, it is more comparable to inputs used for Ridge and GPR here. The long-term surface temperature response predictions along with the true output of the complex GCM for some example test scenarios is shown in Figures 3.4 to demonstrate some important features. These figures also show the short-term GCM response, i.e. the inputs. The complete list of figures for all test scenarios are shown in Figure A.1 of Appendix A for reference.

Both Ridge regression and GPR capture some broad features that pattern scaling is also known to predict effectively, such as enhanced warming over the Northern Hemisphere, particularly over land, and Arctic amplification (Tebaldi and Arblaster, 2014). For example, Figure 3.4a shows the output from the 3xCH_4 simulations, for which all methods provide a similar, accurate response when compared against the GCM response.

However, the key advantage of both machine learning methods is that they capture regional patterns and diversity in the response that is not predicted by pattern scaling. This is a feature seen throughout the range of test scenarios (Figure A.1 in Appendix A), but it is particularly important in improving predictions in short-term pollutant scenarios, which feature particularly inhomogeneous responses.

Figure 3.4b, which shows the predictions for the global removal of SO_2 scenario, highlights this in two ways. Firstly, the true long-term GCM response exhibits a strong temperature gradient between the Northern and Southern Hemisphere, because most of the SO_2 emissions are located in the Northern Hemisphere and their removal leads to warming focused over these regions. Both Ridge and GPR recognise this gradient, with GPR more accurately capturing the extent of this with the cooling in the Southern Hemisphere. The pattern scaling methods cannot reflect this enhanced gradient compared against that found in the reference 2xCO_2 scenario. Secondly, there exists response patterns that appear on smaller spatial scales in the GCM response, due to the short-lived nature of the sulfate aerosol, such as the increased warming over a few grid-cells in East Asia which tend to have high emission levels (as sulfate aerosols have a cooling effect on local surface temperature, their removal leads to warming). This increased warming relative to surrounding areas is represented at the grid cell level fairly accurately in Ridge and GPR, but it does not appear in the pattern scaling methods since 1) Ridge/GPR treats individual grid cells as independent regressions and 2) Ridge/GPR makes use of regional patterns in the



(a)

(b)

Figure 3.4: Maps showing short-term GCM response (inputs) and long-term GCM response (aim) compared to the machine learning predictions, Ridge and Gaussian process regression and the benchmark pattern scaling methods estimated with the ERF and the short-term global mean response (T), for (a) a long-lived forcing scenario, 3xCH_4 and (b) a short-lived forcing scenario, No_SO2_Global. All other predictions can be found in Appendix A, Figure A.1

short-term response, such as the enhanced short-term warming over North America the Arctic and parts of East Asia (e.g. the tongue of warming extending into the Pacific ocean). Similar regional responses are also successfully predicted over Asia in the Gaussian process regression prediction of 10xBC_Asia, 10xSO_4_Asia and No_SO_2_China (Appendix Figure A.1). It is the ability to learn these spatial patterns that gives data-driven methods the edge over any pattern scaling method for such predictions.

This is further highlighted by Figure 3.5, which shows the distribution of predicted temperature responses over all individual grid-boxes for all test scenarios. For the long-lived forcings, such as 3xCH_4, all model predictions produce a similar distribution of surface temperature responses to the GCM. However, for the No_SO_2_Global scenario, GPR predicts the distribution of response most accurately, capturing the feature where some grid-points that respond more strongly (up to 2 °C) while the majority of grid-points respond fairly weakly (less than 0.5°C). The pattern scaling methods are particularly poor in representing this as they cannot contain a response with strong variations between regions, such as strong warming in the Northern Hemisphere with weaker warming or even cooling in the Southern Hemisphere. Even though the global mean response estimated from the temperature approach to pattern scaling is fairly close to the true GCM prediction, the variation in response across the grid is underestimated greatly. This is a fairly consistent result across short-lived forcing scenarios because pattern scaling is constrained to the same pattern, regardless of the scaling factor used to estimate the global mean response. Furthermore, the reference pattern used here was the 2xCO_2 scenario and therefore it is not surprising that this pattern does not perform well on the short-lived forcing scenarios. Applying pattern scaling to an alternative reference pattern for the short-lived pollutants, such as a global SO_2 perturbation, may improve performance for these scenarios. Another caveat with pattern scaling occurs in the case that the short-term response contains strong regional warming and cooling, which cancel out to give a weak global mean radiative forcing (e.g. No_CO_Global and No_BC_NHML) or a weak global mean short-term temperature response (e.g. 0.005°C in No_BC_Global). This can potentially lead to weak predictions in terms of both response and spatial variability, e.g. No_BC_Global. Therefore spatial variability in the short-term response can be a valuable predictor of long-term response, which is exploited only with the machine

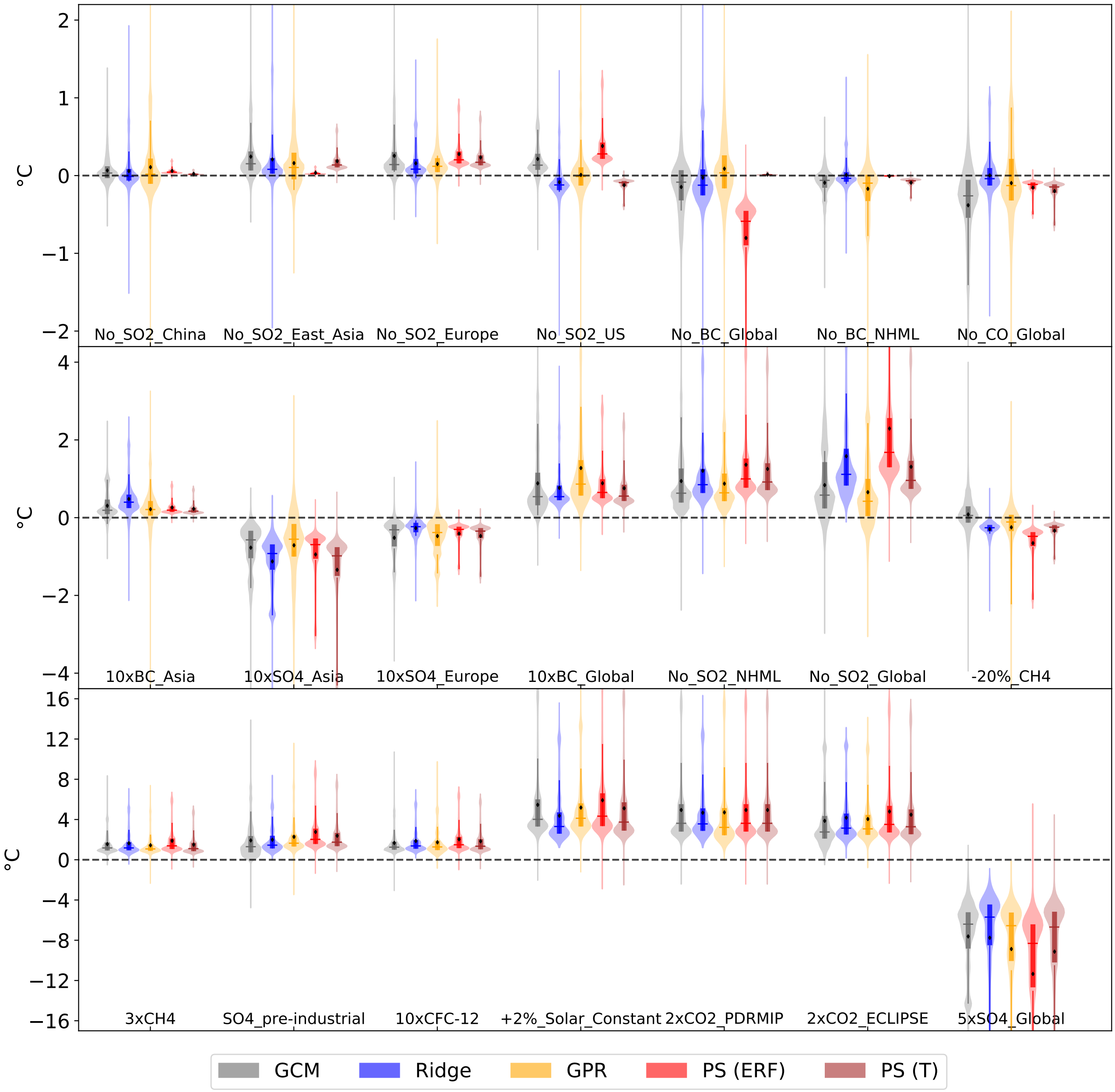


Figure 3.5: Spatial variability of long-term surface temperature response in °C for all prediction methods and for all scenarios. The distribution of predicted surface temperature responses constructed from all spatially weighted grid-points is shown along the vertical axis for each prediction method. From left to right the plots show the prediction from the general circulation model (GCM), Ridge regression prediction, Gaussian Process Regression (GPR) prediction and Pattern Scaling (PS) using effective radiative forcing (ERF) and using the short term global mean surface temperature (T). The central vertical boxes indicate the interquartile range shown on a standard box plot, the horizontal line shows the median and the black point shows the mean. Note the different vertical scales for each row.

learning methods.

In the following, we quantify how well the two machine learning models and pattern scaling perform on different spatial scales. At the grid-scale level, we calculate the Root Mean Squared Error (RMSE) by comparing the prediction and GCM response at every grid-point. We highlight that grid-scale error metrics need to be interpreted with care because they can present misleading results, particularly for higher resolution models. For example, they penalize patterns that - as broad features - are predicted correctly but are displaced marginally on the spatial grid (Rougier et al., 2009). This issue is, by nature, more prevalent for the machine learning approaches where smaller scale patterns are more frequently predicted, while pattern scaling predicts more consistently smooth, cautious patterns with reduced spatial variability. This consideration is a key reason why predictions for larger scale domains are often selected in impact studies (Nowack et al., 2017; Hartmann et al., 2019). We therefore also compare the absolute errors in global mean temperature and in regional mean temperature over ten broad regions, four of which are the main emission regions (North America, Europe, South Asia and East Asia) and the remaining cover primarily land areas where responses affect the majority of the world's population.

The boxplots in Figure 3.6 show how these errors are distributed over all predicted scenarios for each regression method. Note that scenario-specific pattern scaling errors are dependent on the approach chosen to scale the global CO_2 response pattern, but all pattern scaling approaches share their fundamental limitation in predicting spatial variability as described above. Overall, both Ridge and GPR generally outperform the pattern scaling approach, but we find that, in most cases, it is GPR errors that are lowest. The main exception to this is over Europe, which is poorly predicted with GPR. Figure A.1 in Appendix A shows that many scenarios are predicted with a cool bias over Europe with GPR which appears to be learned due to the $10xSO_4$_Europe scenario in the training data. For example, the $10xSO_4$_Asia and 10xBC_Asia scenarios are both predicted with similar patterns to the $10xSO_4$_Europe scenario, due to similarities in the short-term GCM responses. Furthermore, Europe and surrounding regions also exhibit higher levels of noise in the training data, discussed further in Section 3.5, which may hinder GPR prediction.

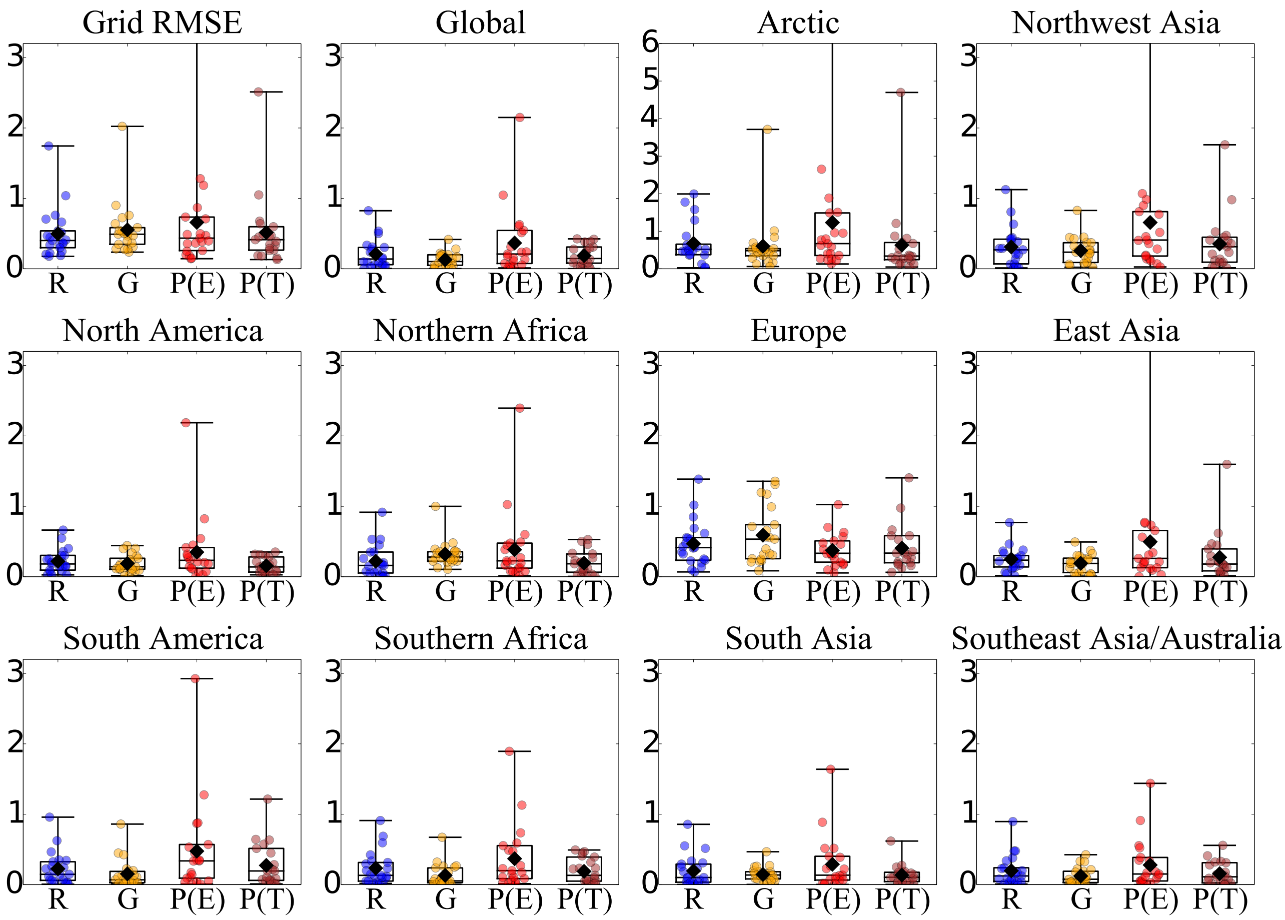
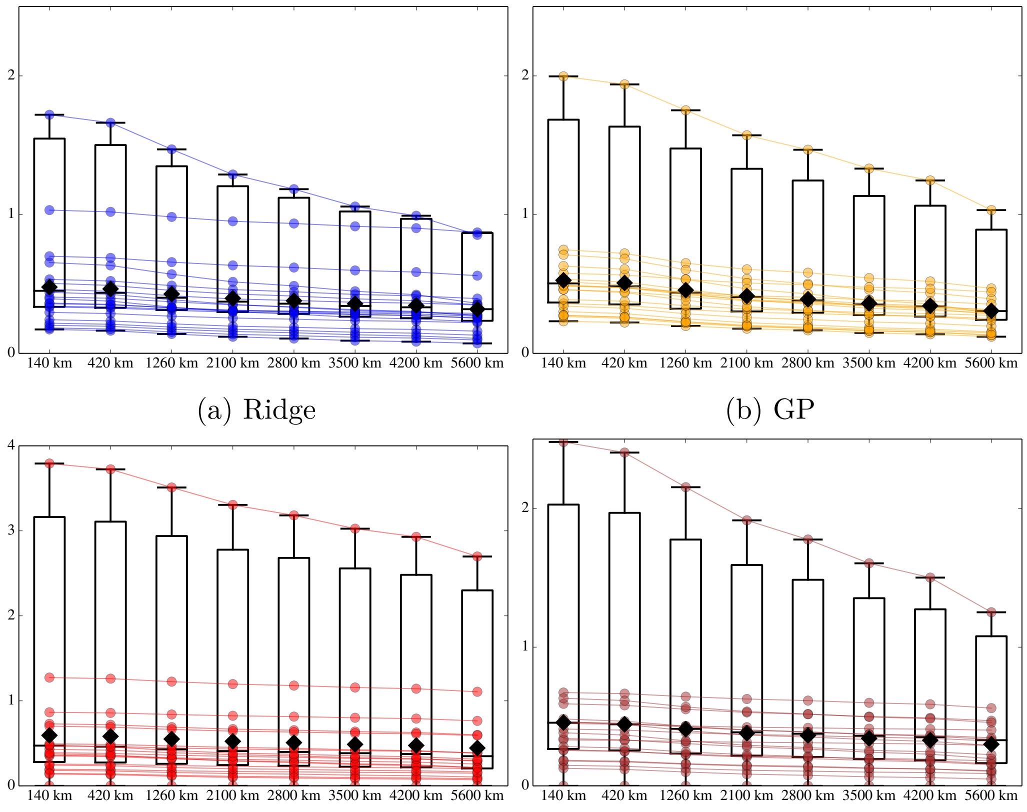


Figure 3.6: Prediction skill comparison for RMSE at grid-cell level, global mean errors and regional mean errors in ten major world regions, in °C for all scenarios. R = Ridge regression, G = Gaussian Process Regression, P(E) = Pattern scaling calculated from the ERF and P(T) = Pattern scaling calculated from the short-term global mean temperature response. Boxplots show the distribution of errors across scenario predictions. Boxes show the interquartile range, whiskers show the extrema, lines show the medians and black diamonds show the mean. The dots indicate the errors for each individual scenario. Note the different scale for the Arctic and that points exceed the scale in Arctic (9.5), Northwest Asia (4.7), East Asia (3.7) and the Grid RMSE (3.8).

Figure 3.7 further highlights the point regarding the performance in different spatial scales, as the RMSE is calculated for boxes of different sizes, starting at the standard grid-box of the GCM (140km) and increasing in size by taking a sliding window average over multiple grid-boxes. The performance of the machine learning does not degrade as poorly when the grid-box size is decreased compared to the pattern scaling methods, indicating they are better at predicting on smaller spatial scales. Although making predictions at the grid-scale level is the ultimate goal, care should be taken when interpreting results of individual grid-cells, as mentioned earlier.



(a) Ridge

(b) GP

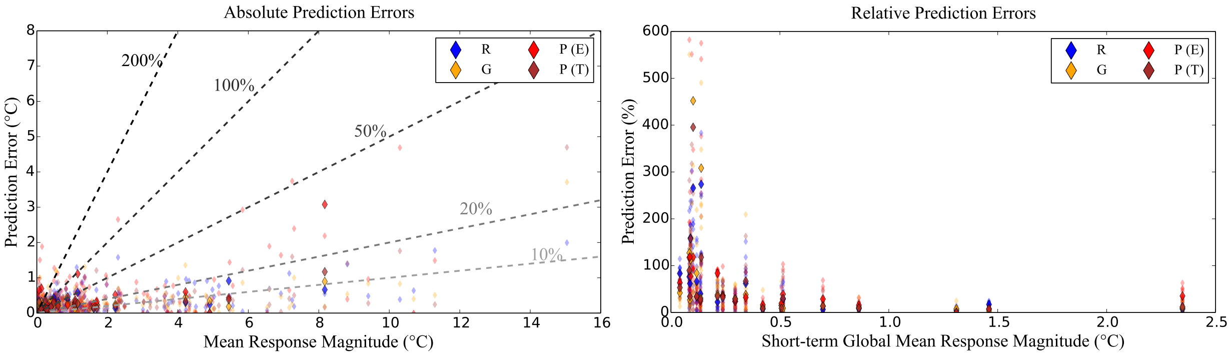
(c) PS (ERF)

(d) PS (T)

Figure 3.7: RMSE in °C calculated on different spatial scales, spanning increasing sizes, each point is the error for one test simulation. R = Ridge regression, G = Gaussian Process Regression, P(E) = Pattern scaling calculated from the ERF and P(T) = Pattern scaling calculated from the short-term global mean temperature response.

3.5 Noise and Internal Variability

The large spread in absolute errors in Figure 3.6 is due to the large spread in response magnitude for the different scenarios. Specifically, the large errors, $\gtrsim 1$ °C arise mostly from regions/scenarios with a large magnitude of response, e.g. $\gtrsim 6$ °C. This is demonstrated by Figure 3.8a, which shows the mean response magnitude against the absolute prediction error for each method, region and scenario. This feature tends to be more pronounced for the scenarios with strong forcings (e.g. strong solar or greenhouse gas forcings). However, these errors are often small relative to the overall magnitude of scenario response, typically falling below the 20% prediction error line. However, the opposite tends to be true for scenarios with a weak mean response magnitude, e.g. < 1 °C in Figure 3.8a. Although the absolute prediction error may not be particularly large these errors can be large relative to the magnitude of response, some of which are greater than 200%.



(a) Absolute prediction error against mean long-term GCM response magnitude

(b) Relative prediction error against the short-term global mean response magnitude

Figure 3.8: Absolute and relative prediction errors for all methods, all scenarios and all regional averages. The methods are ridge regression (R, blue), Gaussian process regression (G, yellow), pattern scaling estimated with ERF (P(E), red) and pattern scaling estimated with short-term global mean temperature response (P(T), dark red). Global mean averages are represented by the points in bold. The relative prediction errors are calculated as a percentage of the magnitude of the true GCM long-term response.

Figure 3.8b shows the relative error for all scenarios over all regions compared to the global mean short-term response magnitude, i.e. the magnitude of the predictor variables. The large relative prediction errors mentioned previously tend to come from predictions of scenarios with weaker short-term responses, which will have lower signal-to-noise ratios. These tend to be the short-lived pollutant scenarios, (BC, SO_4, SO_2, CO). These scenarios are less likely have strong and clear signals that exceed the internal variability. This means predictions may be made based on noisy inputs, leading to larger prediction errors relative to the true response.

This is motivation for a training dataset with more strongly forced scenarios with greater signal-to-noise ratios in the short-term response. It also explains some of the features in Figures A.1 and 3.5, in which weak forcing short-lived pollutants are less accurately predicted in all methods, e.g. No_BC_NHML, No_BC_Global and No_SO_2_US. This is a key limitation for any method, whether it makes use of individual grid-cells as predictors or not, since weaker scenarios often suffer low signal-to-noise ratios, making them inherently more difficult to predict.

There is also a difference in magnitude of response between regions, which has an effect on the magnitude of errors in Figure 3.6. The mean absolute error maps for all prediction methods are shown in Figure 3.9. High latitude regions typically have larger errors, in particular the

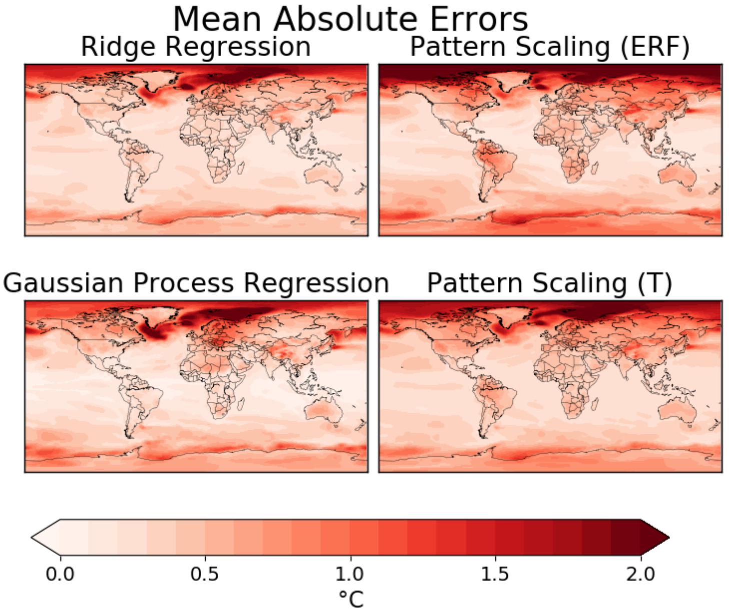
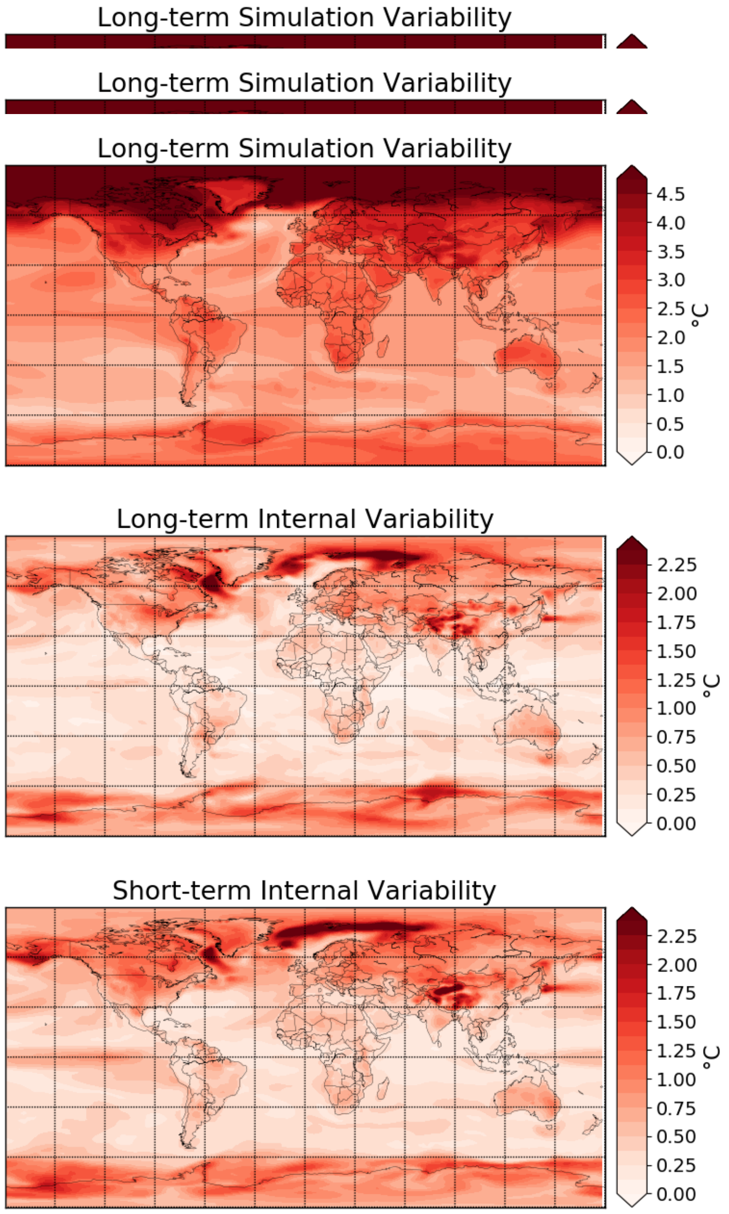


Figure 3.9: Mean absolute prediction errors in °C for all four methods over all test simulations

Arctic and Europe, indicating these are more difficult to predict with any of these methods, based on the available data. These regions are also noted as regions of high variability amongst the simulation data available, shown Figure 3.10a, and of high internal variability within the GCM itself, Figures 3.10b-3.10c which are calculated as the standard deviation in the control run responses over the short and long timescales, respectively. There are particularly similar signatures over the Arctic and North of Europe that appear in both the error maps of the machine learning models and the internal variability maps, suggesting that this internal noise is one of the limitations for machine learning methods. This is a sign that care should be taken when attempting to predict these regions, given the implicit noise in the GCM data. This could further motivate research into the benefits of reducing, or even eliminating, noise in climate model datasets prior to building surrogate models. Recent studies have used machine learning methods to disentangle the signal associated with a forcing from the internal variability in climate data (Sippel et al., 2019; Wills et al., 2018, 2020b). These methods could potentially be used to build a training dataset that is not obscured with internal variability, which could then be combined with methods presented here to build a surrogate model that learns on signals in the data alone.


Long-term Simulation Variability
Long-term Simulation Variability
Long-term Simulation Variability
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
°C
Long-term Internal Variability
2.25
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
°C
Short-term Internal Variability
2.25
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
°C

(c)

Figure 3.10: (a) Variability in long-term response across simulations available for training, calculated as standard deviation across available data (b) internal variability in long-term response and (c) internal variability in short-term response, calculated as standard deviation across control run simulations

3.6 Variations on Regression Inputs

In this section, we will cover the variations we have explored in the regression inputs, by scaling the inputs, applying dimension reduction techniques, working with different predictor variables and using a different timescale to define the ‘short-term’ response.

3.6.1 Scaling Inputs

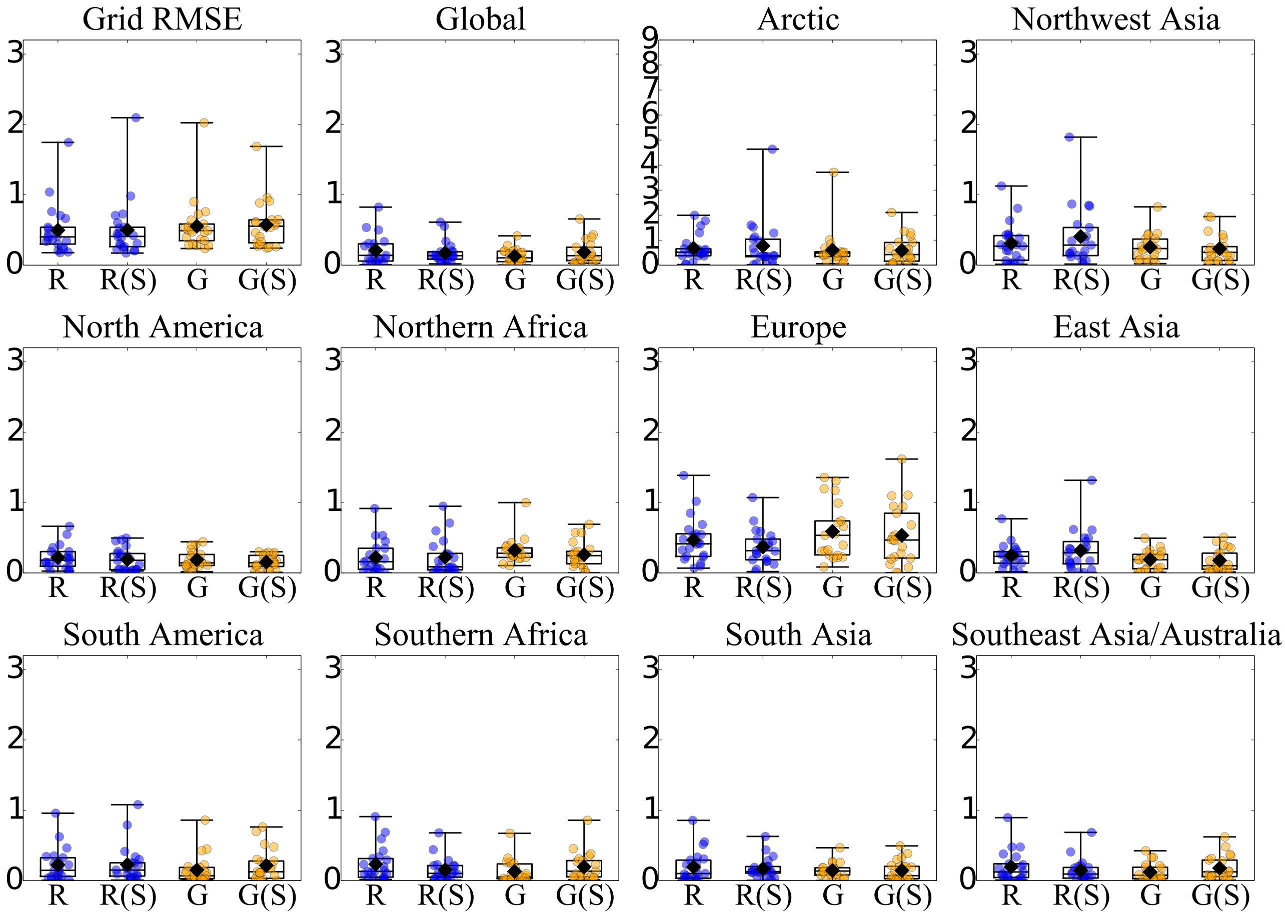


Figure 3.11: Absolute errors in °C for scaled and unscaled inputs for Ridge regression and Gaussian process regression, where the (S) indicates each individual input has been scaled to between between 0 and 1 before training.

The results presented so far did not involve scaling of the inputs independently. This maintains the temperature structure of the grid of inputs and gives greater weighting to regions that typically respond more strongly. However, we have also explored the effect of scaling the inputs independently between 0 and 1, so that the short-term response at every grid cell is relative to the training dataset, rather than the grid itself. This removes the dependence on the magnitude of response, but can introduce dependence on the types of training data available when there are limited simulations available (e.g. if there is a bias towards more warming in a particular region in all training simulations). Figure 3.11 shows the results of both Ridge and GPR predictions when the inputs are scaled, compared against the unscaled approach. There are very little differences between these choices, so for interpretibility we use unscaled inputs unless otherwise stated.

3.6.2 Dimension Reduction

A key challenge of working with the climate model information here is its high dimensionality (27,840 grid-cells) given the small scenario sample size of 21 simulations. We have explored sensible approaches to dimension reduction, both statistical and physical, on both the short-term and long-term temperature response. For the statistical approach, we use principal component analysis (PCA) described in Section 2.2.1. Since the number of components is limited by the number of simulations available (20), we use all components here (Hastie et al., 2001). PCA exploits correlations between the outputs in the training data, which reduces the cost of training, as well as enforcing correlations in the outputs in new predictions, which are otherwise not guaranteed through the independent regression presented previously. Although we do not see improvements in the performance of the response, this choice can make resulting predictions easier to interpret because of the en For physical dimension reduction we make predictions on key regions, informed by knowledge of coherent climate characteristics rather than statistical relationships. We use the regions over land described previously along with additional regions over the oceans (divided into North Atlantic, South Atlantic, North Pacific, South Pacific, Indian Ocean, Southern Ocean and the Antarctic) to cover the full grid. Figure 3.12 shows the absolute GPR prediction errors in each region using these dimension reduction methods on the inputs (the short-term response), on the outputs (the long-term response) and on both the inputs and outputs. By using dimension reduction on the short-term responses, the problem becomes better constrained as there are fewer parameters to optimise. However, we do not find significant improvements in the predictions with either approaches.

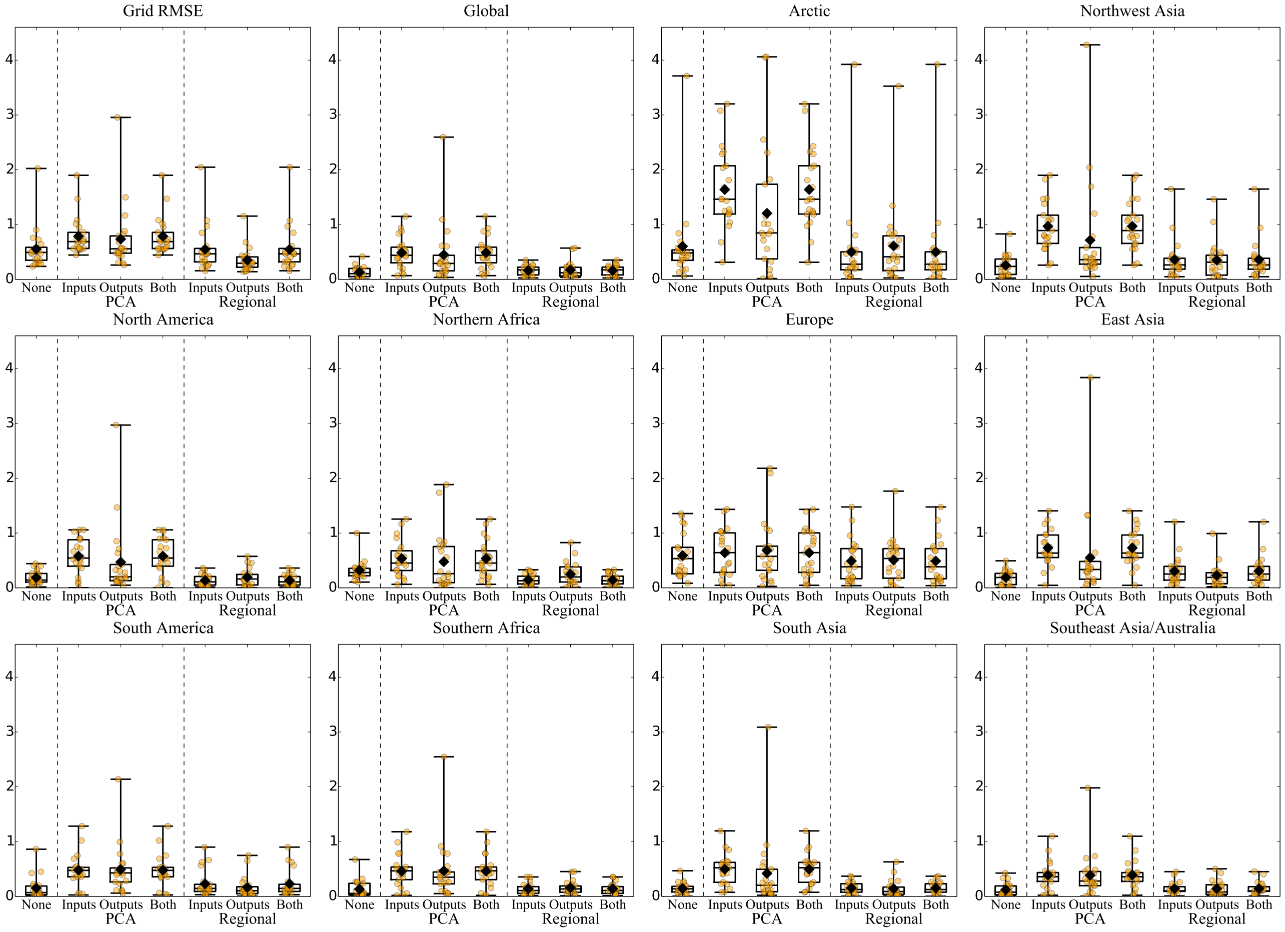


Figure 3.12: Absolute errors in °C for Gaussian process regression where different approaches to dimension reduction are used. The first column shows no dimension reduction, the next three columns use principal component analysis (PCA), on the inputs, the outputs and both inputs and outputs respectively and the last two columns use regional dimension reduction on the inputs, the outputs and both inputs and outputs respectively.

3.6.3 Different Predictors

We have also explored the use of different variables as the short-term predictors, as there are a range of sensible GCM variables that could be chosen as inputs to the regression for predicting the long-term temperature response. Figure 3.13 shows the prediction errors when using various predictor variables in the regression, where errors are calculated from the absolute error over the same regions. These predictor variables are surface air temperature, air temperature at 500 hPa, geopotential height at 500 hPa (as an indicator of the large-scale dynamical responses), ERF and sea level air pressure. Both sea level pressure and ERF produce large absolute errors suggesting these are not suitable predictors for long-term surface temperature response patterns. However, air temperature and geopotential height at 500 hPa offer predictions to a similar degree of accuracy as the surface temperature response. This suggests there is similar information encoded in these variables and their patterns. Still, surface air temperature appears to be the predictor variable with consistently lower prediction errors and is most interpretable for predicting the long-term surface temperature response.

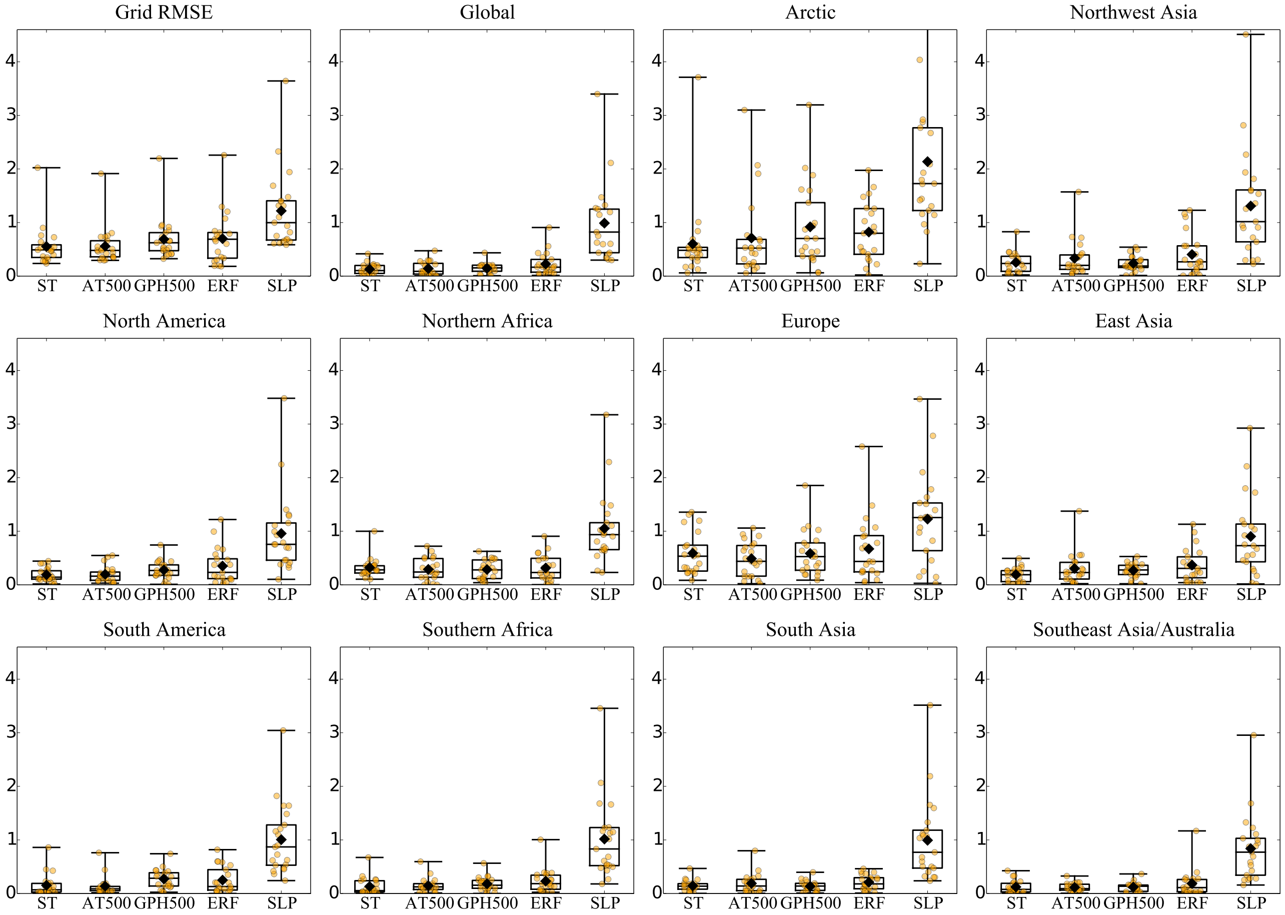


Figure 3.13: Absolute errors in °C for Gaussian process regression where different predictor variables are used as inputs for all key regions: ST=Surface Temperature, AT500=Air Temperature at 500hPa, GPH500=Geopotential Height at 500hPa, ERF=Effective Radiative Forcing, SLP=Sea Level Pressure. Note that when using SLP as a predictor for the response for the Arctic, two points exceed the axis maximum (5.0 and 5.6 °C).

3.6.4 Five year Inputs

Throughout, we have defined the short-term response to be the first 10 years to allow the GCM some time to respond to the forcings to remove as much natural variability as possible, with 10 years deemed to be sufficient for pattern scaling problems in Mitchell (2003). However, we find that using a shorter time period of only 5 years already shows promise. Figure 3.14 shows the absolute prediction errors when the short-term response is defined as the first 5 years of the GCM response. The prediction errors for Ridge regression and Gaussian process regression are increased compared to Figure 3.6, but in most regions are competitive when compared against both pattern scaling methods, particularly the ERF approach. As before, it is expected that an increased training dataset will further reduce prediction error, when using this 5 year approach. This would make a strong enhancement to the speed of prediction of new unseen scenarios, as fewer years of the GCM would be required.

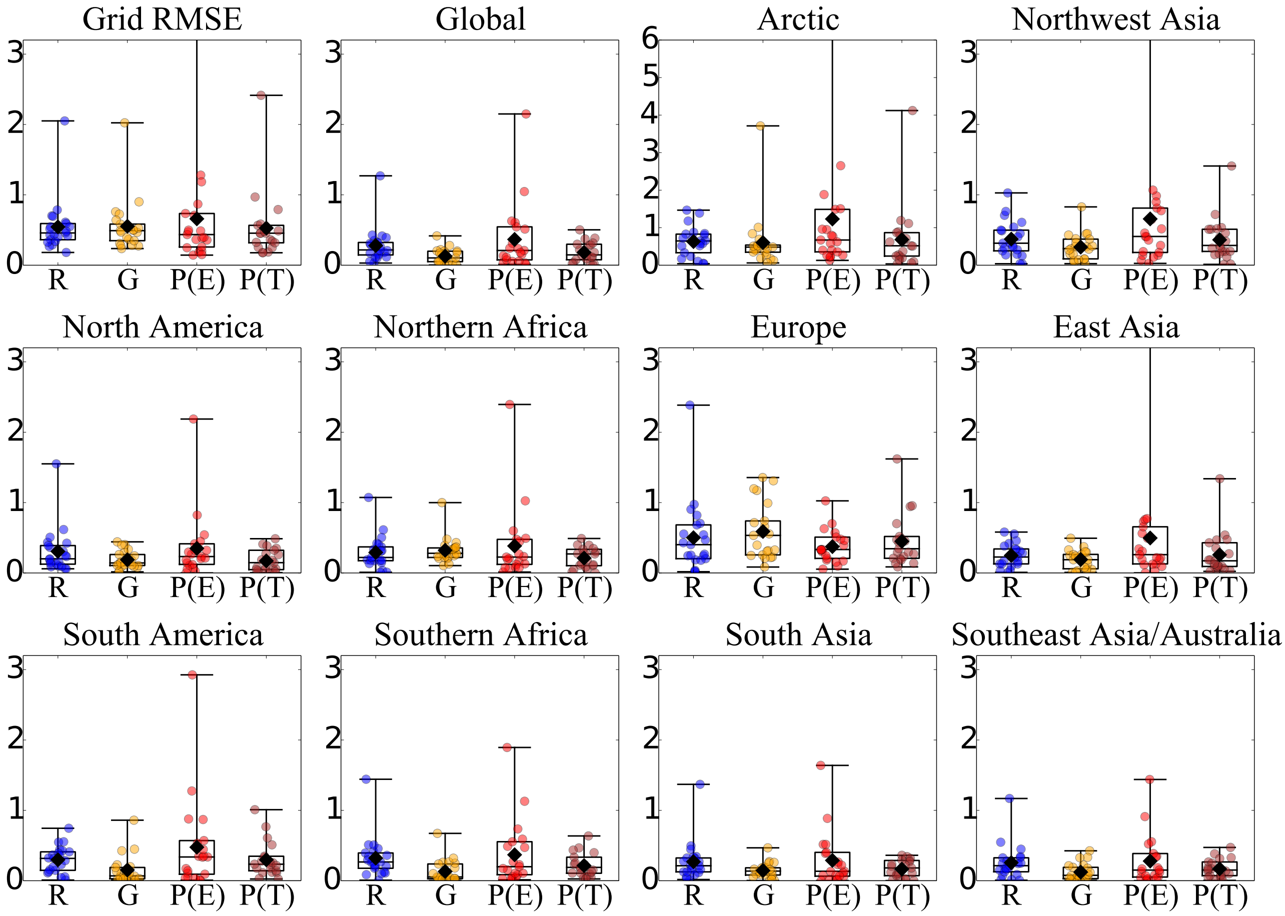


Figure 3.14: Absolute error in °C but calculating using 5-year mean in definition of short-term response.

3.7 Learning Early Indicators

As well as advancing our predictability skills, the machine learning methods inform us about regions that experience the earliest indicators of long-term climate change in the GCM. By assessing the structure of learned Ridge regression coefficients, we find patterns in the short-term response that consistently indicate the long-term temperature response. There are a large number of regression coefficients; specifically, for each of the 27,840 outputs, there are 27,840 regression coefficients as described in Section 2.1.4. We consider the regression coefficients for prediction of the long-term response at a single grid-cell. The nature of ridge regression drives most of these coefficients to close to zero and therefore they have little influence on the predicted response. However, the coefficients of larger magnitudes correspond to regions in short-term response that are the best predictors of the long-term response at the particular grid-cell of interest.

Figure 3.15 show these coefficients for predicting the response a single grid-point, indicated by the star, in (from top left to bottom right) Europe, East Asia, Central Africa, South Africa, North America, South Asia, South America and South-East Asia. Note that these are the coefficients β calculated when the input variables x are normalised between 0 and 1, in order to treat all predictors equally.

In the points selected in Europe, East Asia, Central Africa, South Asia and South America, the dominant coefficients appear in regions close to the predicted grid cell, indicating a strong relationship between the short- and long-term responses in the localized region. This is intuitive as it makes sense for a model to make predictions of long-term response predominantly based on short-term responses in surrounding areas. Still, some patterns in the coefficients exist that indicate influences in the prediction from further afield.

In contrast, some regions draw more predictive power from remote regions. For example, the point in South East Asia and South Africa both maintain fairly weak coefficients in their surrounding area, but rather rely on coefficients located near South America. This is generally seen in Southern Hemisphere predictions. These coefficient patterns are indicative of noise seen

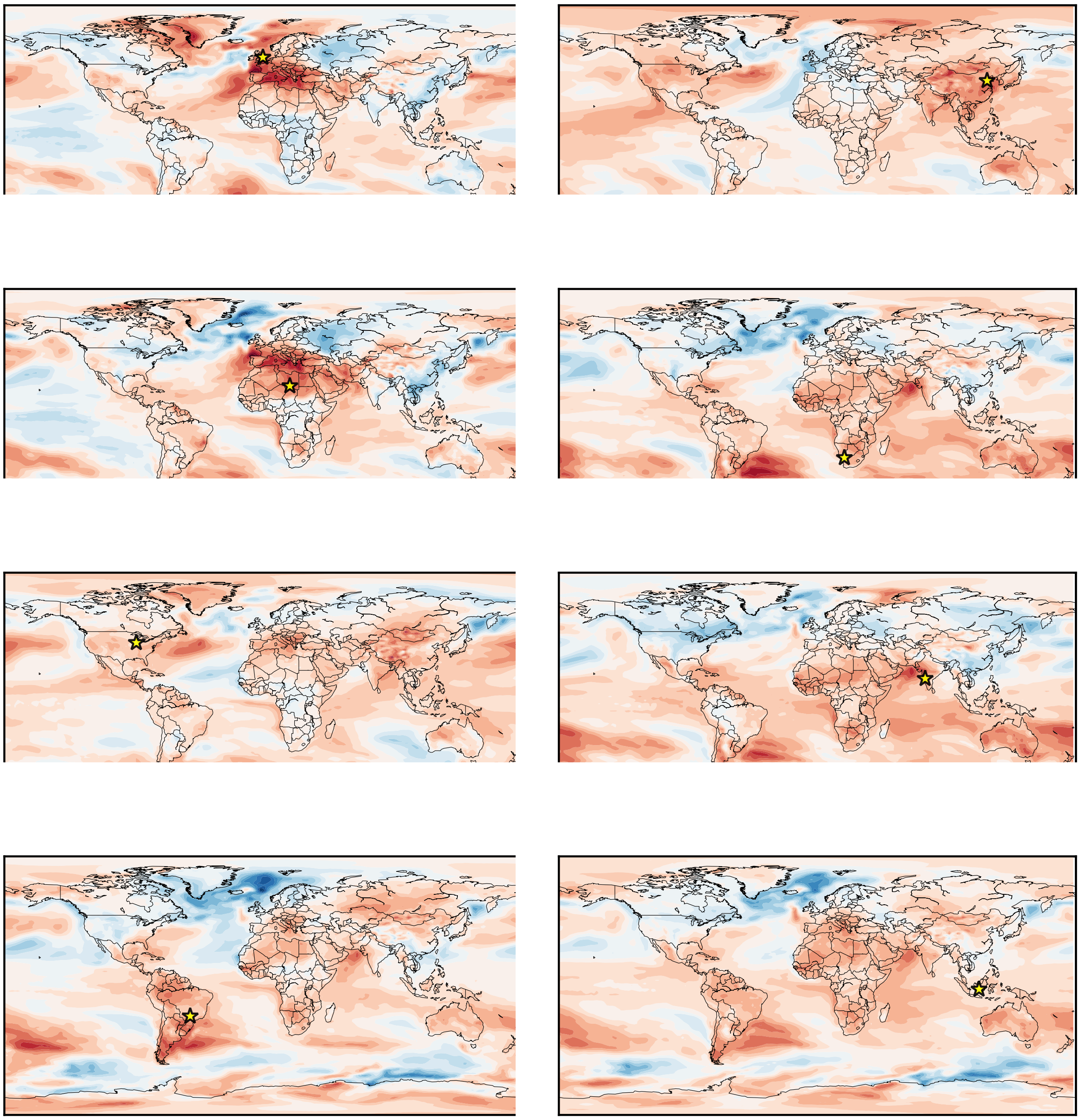
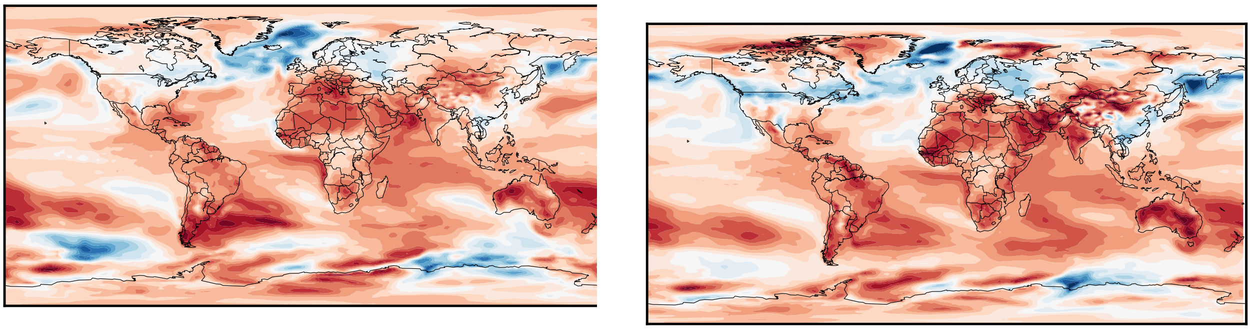


Figure 3.15: Ridge regression coefficients for a single grid-cell regression at the point indicated by the star.

in GCMs, e.g. the Southern Ocean Oscillation. Another example of this is that the coefficients for the European prediction are strongly influenced by the short-term responses in sea ice regions over the Arctic, which is also seen in other high-latitude predictions. As the short-term response in this Arctic region is highly variable (Figure 3.10c) and strongly responding (due to Arctic amplification), this could lead to these regions being picked up as important for the long-term response prediction. This is not necessarily surprising since previous studies find that GCM climate response is often projected onto typical modes of variability (Kasoar et al., 2018), and is not inherently a negative feature of the regression model, since this can be a realistic indicator of how a GCM will project climate response. However, if the training data contains high levels of noise in certain regions, this can lead to inaccurate selection of coefficients. Figure 3.10a showed particularly high variability in the training data in the Arctic region, which may lead to poor predictions if the regression is strongly dependent on these regions. This may explain why European predictions suffer poorer performance relative to other regions, which confirms that



(a) Inputs scaled to between 0 and 1 for each gri point

(b) No scaling applied

Figure 3.16: Ridge regression coefficients globally averaged over all outputs, where (a) scaling has been applied to remove dependence on magnitude of response at each grid cell and (b) no scaling has been applied

There are spatially similar features in the regression coefficient maps that appear regardless of prediction region, such as the larger coefficients over East and South Asia, Northern Africa and over the Southern Hemisphere jet stream. This is further highlighted when an average is taken across all 27,840 outputs to find the global mean regression coefficient map shown in Figure 3.16a. This suggests that these regions are robust early indicators of long-term response in the GCM. These also feature strong coefficients in regions over East Asia which is a typical indicator in the short-term response based on the training data available. Since many of the simulations consist of strong perturbations in that region (SO_2 perturbations in particular), it is not surprising that the coefficients select these regions as important in predicting future response. When the input variables are not scaled, demonstrated by the global mean coefficient map in Figure 3.16b, we find similar patterns in the coefficients, with a slight increase in intensity of the coefficients associated with warming patterns (over the tropics and land). The coefficients also place slightly greater weighting in the ice and snow regions (e.g. Greenland, the Arctic) and high-altitude regions (e.g. Himalayas), both of which are known to warm more rapidly due to ice/snow albedo feedback (Hall et al., 2019) and faster upper tropospheric warming (Nowack et al., 2017; Fu et al., 2011) respectively. The choice not to scale the input variables naturally gives stronger weighting to regions that typically respond more strongly and so it can be used to pick out regions that typically respond more rapidly to perturbations in the model. However, the fact that these patterns are still present in the scaled coefficients, indicates that the accelerated warming in these regions is also a robust harbinger of long-term change within the model. We highlight the implications for future studies that attempt to interpret already observed warming patterns from a climate change perspective.

An alternative approach to ridge regression is the LASSO regression described in Section 2.1.5. In LASSO regression, most coefficients are sent to 0, which leads to strong dependence on just a few grid points (sparsity). This structure does not lead to particularly intuitive coefficient maps, and in fact it was found that often remote regions were picked out. Figure 3.17 shows the coefficient map built using LASSO regression for the two grid-points in Figures 3.16(a) and (b). These both depend entirely on just a few grid-points and furthermore the grid-points are not local to the region of prediction grid-point over Europe. The use of Ridge regression makes more sense over LASSO because Ridge better deals with the collinearity of predictors (Nowack et al., 2018; Dormann et al., 2013).

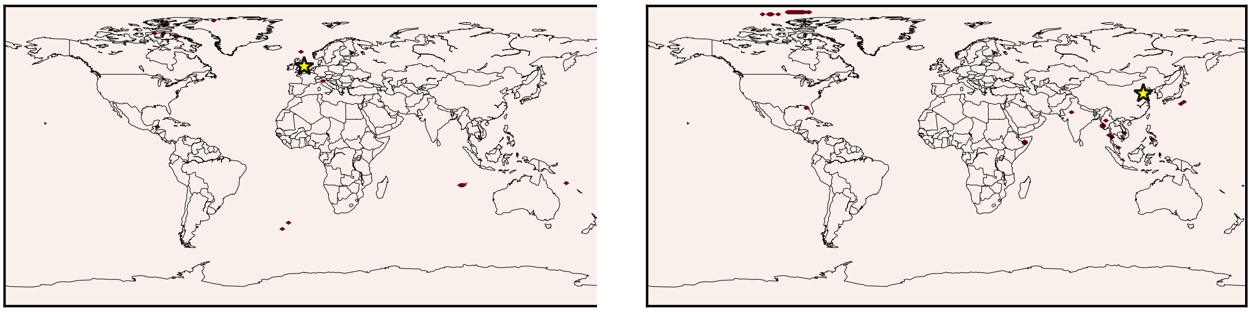


Figure 3.17: Lasso regression coefficients for a single selected grid-cell regression at the point indicated by the star

3.8 Data Constraints

We identify more extensive training data (additional simulations and forcing scenarios) as key to further improving the skill of the machine learning methods. In Figure 3.18 it is demonstrated that as the number of data training samples increases, the mean prediction accuracy significantly increases and becomes more consistent. Note that this is estimated by repeating the training and predicting process over different combinations of training data, which reduces the mean error (c.f. bootstrapping, (Wasserman, 2004)). We would expect significant potential for further improvements in predictions with even more training data, particularly for the RMSE, the Arctic and Europe which show larger errors. More simulations would better constrain parameters of the statistical models and improve the chances that a predicted scenario contains features seen during training of the statistical model.

Since obtaining training data from the GCM is expensive, sensible choices can also be made about how to increase the dataset by choosing which new scenarios will benefit the accuracy of the method the most, e.g. to address some complex regional aspects of the responses to short-lived pollutants. We recommend increasing the dataset to include more short-lived pollutant scenarios, noting that those with large forcings have the potential to reduce the noise in the training data so as to better constrain learned relationships (Figure 3.8b).

Some regions stand out as particularly challenging for our machine learning approaches, with Europe being a prominent example noted above. It is thought this could be down to several factors or a combination of them. Firstly, the large variations in the long-term response across

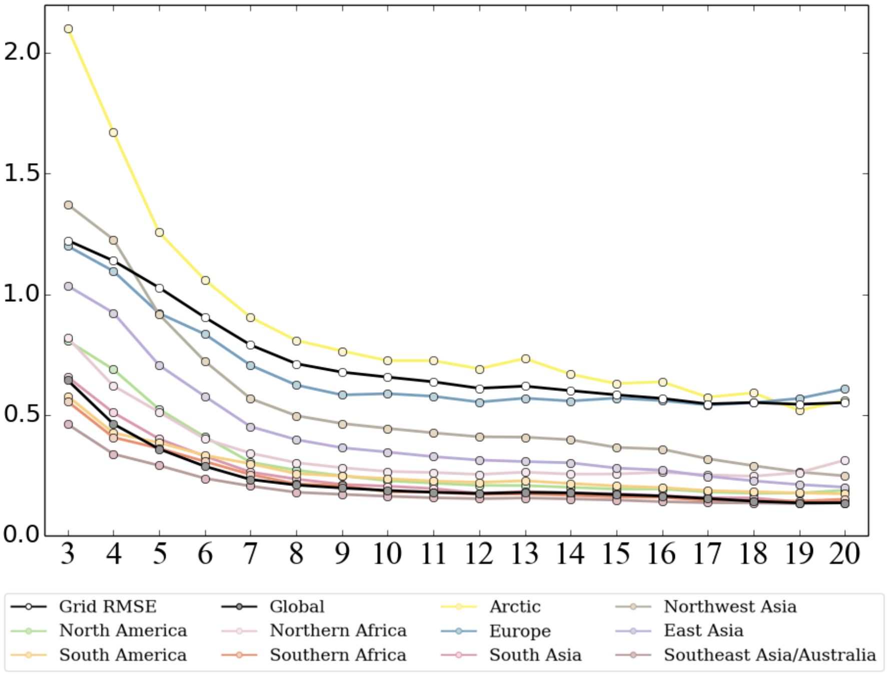


Figure 3.18: Mean of absolute errors in °C across all predicted scenarios against number of training simulations, with each line representing a different region. RMSE at the grid-scale level is also shown in black with white dots. For a fixed number of training data points, the process of training and predicting is repeated several times over different combinations of training data to obtain multiple prediction errors for each scenario.

the training data over Europe and surrounding regions (Figure 3.10a) means predictions are less well constrained and would benefit more from increased training data. Secondly, the internal variability in the GCM-predicted temperature time series is generally larger over Europe compared to other regions in both the control and perturbation simulations, which gives rise to a weaker signal-to-noise ratio for both short- and long-term responses in this region, increasing the difficulty of learning meaningful predictive relationships (Figure 3.10b). And finally it was also noted that Ridge regression predictions for Europe depend strongly on parts of the Arctic (Figure 3.15) where the short-term response is stronger but also highly variable (Figure 3.10c) These reasons point to the issue that internal variability can introduce noise to the inputs and outputs of the regression, which can have unintended consequences. This is partially addressed with temporal averages in the definitions of the short- and long-term responses, under the limitation that we have only a single realization of each simulation available. If, however, future studies have an ensemble of simulations available for each perturbation, an average over ensembles would more effectively separate the internal variability from the response. The use of several diverse simulations in the training dataset also allows the noise in the inputs and outputs to be treated as random noise in the regression, which, however, would be even better determined with increased training data.

3.9 Conclusions

This chapter has covered how Ridge regression and Gaussian process regression can be used to predict long-term temperature response, given short-term temperature response of a particular GCM. These methods outperform the standard approach, pattern scaling, particularly for forcings that are non-homogeneous, such as short-lived pollutants. However, there are still challenges to come for the prediction of climate change using machine learning techniques.

One of these challenges is the high levels of noise due to internal variability in GCMs. There is potential for future machine learning models of this form to be assisted by recent research into tools that help uncover the signal from the noise in climate data (both observations and simulations) (Sippel et al., 2019; Wills et al., 2018, 2020a,b). If these techniques are to become refined and well-used in climate science, machine learning emulators could be built to learn from signals alone, without major interference of unwanted noise.

Another of the challenging aspects of this work is the high-dimensionality, as is common in climate science and which motivates developments made in Chapter 5. In this section, we carried out independent single output Gaussian processes for each grid-point, not accounting for correlations between them. However, recent ongoing literature in the field of Gaussian processes could help exploit correlations to leverage information with this. A review of potential approaches for this can be found in Liu et al. (2018a). A fairly well-used approach is to build co-regionalised models, where multiple outputs are treated like inputs and indexed by the spatial grid. This was not used here because it would be highly expensive for this particular problem as the cost scales as $O(N^3p^3)$ for $p = 27,840$ outputs and N training data simulations.

One of the focal points of Mansfield et al. (2020) is the relevance of this study for the future of data-driven climate modelling, especially concerning the incorporation of even larger model datasets in the future. Although this work made use of existing simulations from a single global climate model, it opens the door for similar approaches to be taken with datasets from other climate models. We therefore encourage widespread data-sharing to test the limits of this approach as an important part of future research efforts in this direction, which can hopefully lead towards more powerful climate response emulators and thus faster climate change projections.

Note that we have explored the use of these methods in the context of predicting temperature responses, however, we leave open the topic of predicting other variables such as precipitation, which we expect to be more challenging due to its spatial and temporal variability (Pendergrass et al., 2017; Pendergrass and Knutti, 2018) but for which pattern scaling approaches are well-known to perform particularly poorly (Mitchell, 2003; Murphy et al., 2007; Tebaldi and Arblaster, 2014; Pendergrass and Knutti, 2018).

This method presented here has the potential to drastically accelerate the process of long-term climate prediction by reducing the length of simulations required from multi-decadal timescales to the order of 5-10 years. However, even further computational savings can be made by coupling this method with the short-term response emulator presented in the following chapter (i.e. multi-level emulation (Cumming and Goldstein, 2009; Tran et al., 2016)). Without the need to repeatedly run expensive GCM simulations, this work provides a starting point for rapid climate prediction as a tool for both scientific and policy purposes.

Chapter 4

Short-term Climate Response Prediction with an Emulator

4.1 Introduction

In Chapter 3, I described a surrogate model built to predict the long-term response of a global climate model (GCM), given the short-term response. The goal of this chapter is to build a surrogate model that predicts the short-term climate response based on emission perturbations. While Chapter 3 focused on a comparison of machine learning regression methods to predict *single point estimates* of the output, given the input and trained on a set of existing data, this chapter will take a more probabilistic approach. The analysis will include a discussion of the probability distributions which are estimated by a Gaussian process. Although both chapters make use of Gaussian processes, I will distinguish the two different viewpoints as 'machine learning' in Chapter 3 and 'emulation' in this chapter. This highlights the focus on probabilistic estimation and the exploration of uncertainty in the current chapter, as well as remaining consistent with the emulation literature. This chapter will also include a complete design of the training data, exploration of sensitivity of the GCM to the different emission perturbations studied and a demonstration of how this type emulator could be useful in climate change studies. As discussed in Section 1.2.3, Gaussian process emulators have been used to assist atmospheric and climate modelling, through sensitivity analyses, uncertainty quantification and calibration (e.g. Carslaw et al., 2013; Lee et al., 2012; Salter and Williamson, 2016; Williamson et al., 2015; Ryan et al., 2018). However, this study is the first to make use of emulators for climate change projection under different emissions scenarios for use in policy-relevant studies.

This emulator is designed for predicting the short-term global surface temperature response to different emission perturbations. We are interested in the climate response to both greenhouse gases and aerosol perturbations and therefore the input variables will include a selection of both, based on their importance shown in previous studies. We take the inputs to be categorical, distinct emission settings (rather than emission fields). For instance, the well-mixed GHGs can be perturbed via the global concentrations. For short-lived pollutants, we use scaling factors over broad regions on continental scales and assume that the relative distribution of emissions within these regions remains the same, following the same approach as the datasets in Chapter 3. The outputs of the emulator are the short-term surface temperature response at every grid cell of the GCM, giving a high-dimensional output space. The short-term temperature response is defined as the average over the first 5 years of the GCM response, in order to reduce the high cost of running the GCM, although ultimately the simulations could be extended to 10 years in future work. As in Chapter 3, each grid cell will be treated independently, with a separate Gaussian process emulator built for each one. Before describing the emulator in further detail, I will outline the set-up and design of the input parameters chosen, starting with an overview of the GCM used in this study.

4.2 Global Climate Model Configuration

We use the same GCM as in Chapter 3, HadGEM3, in the most recent configuration, the third Global Coupled configuration 3.1 (GC3.1) (Williams et al., 2018) which consists of the global atmosphere-land (GA7/GL7.1) (Walters et al., 2019), global ocean (GO6) (Storkey et al., 2018) and sea-ice (GSI8.1) (Ridley et al., 2018). This configuration uses the modal version of the Global Model of Aerosol Processes (GLOMAP-mode) two-moment aerosol microphysics scheme

(Mann et al., 2010), which differs from the aerosol scheme used in Chapter 3 as it simulates aerosol number as well as mass and includes aerosol microphysical processes such as nucleation (Bellouin et al., 2013). The same model resolution is used as Chapter 3, with a grid of 145×192, giving an average grid cell of width 140 km at the equator. The same model configuration is described for historical simulations in Andrews et al. (2020). 'Present-day' conditions are taken from the end of these simulations, using a 2014 timeslice of the climate quantities. This is run for 300 years as 'spin-up', to reach an approximate equilibrium, which is used as a start point for simulations for the emulator.

Before computing any perturbation runs, 6 independent control runs were simulated starting from this equilibrium state. These were run for 5 years each, to be consistent with our definition of 'short-term' climate response in this chapter. These control runs will be used in two ways. Firstly, we aim to predict the surface temperature *response* to any perturbations, which is calculated as the surface temperature of the perturbation run minus the surface temperature of the control run. To reduce the impact of short-term internal variability, we subtract an average over all 6 independent control runs, i.e.

$$\mathbf{y} = \mathbf{T}_{\text{pert}} - \frac{1}{6}\sum_{i=1}^{6} \mathbf{T}_{\text{control},i} \tag{4.1}$$

where $\mathbf{T}_{\text{pert}}$ and $\mathbf{T}_{\text{control}}$ are the spatio-temporal temperature outputs of the perturbation and control runs respectively, i.e. for the entire grid and over the 5 years. For the emulator outputs, the temporal average of $\mathbf{y}$ is taken over these 5 years, to reduce short-term variability. Secondly, the control runs will be used to estimate the internal variability, which will be viewed as a limit of the accuracy of the emulator, later discussed in Sections 4.4 and 4.6.

4.3 Emulator Design

Due to the high cost of running the GCM, we aim to minimise the number of simulations run when building and testing this emulator. This can be done with a careful choice of input parameters and structured design of simulations to be run. In this section, I will outline the steps and decisions made in the design of the emulator.

4.3.1 Choice of Inputs

First, we select a few of the main climate forcers to vary, to limit the number of possible input parameters. These include long-lived climate pollutants, specifically the two dominant greenhouse gases carbon dioxide (CO_2) and methane (CH_4), which are usually regarded as the strongest drivers of climate (Stocker et al., 2013). For the short-lived pollutants, we perturb sulfur dioxide (SO_2, the precursor to sulfate aerosol (SO_4)) and biomass burning carbonaceous aerosols (black carbon (BC) and organic carbon (OC)). Sulfate aerosol is chosen because it has a strong cooling impact on the climate, particularly at a regional level, and because its precursor is emitted mostly from fossil fuel burning, making it is highly relevant for scenario projection studies (Stocker et al., 2013). Biomass burning OC/BC are chosen as they strongly affect the climate through scattering and absorption of radiation and due to aerosol-cloud interactions (Section 1.1.2). In addition, biomass burning is a particularly high source of uncertainty in future scenarios, but has not been studied as extensively as other forcing perturbations (e.g. Tosca et al., 2013; Ward et al., 2012; Lasslop et al., 2019; Voulgarakis and Field, 2015). Other pollutants (e.g. NOx, VOCs, OC/BC from other sources) would require particularly large, unrealistic forcing magnitudes (e.g. 5-10x current emissions) to cause significant climate change and therefore we do not extend the emulator to this regime (e.g. ECLIPSE (2014); Shindell and Faluvegi (2009)).

For the greenhouse gases, we make global perturbations to the GCM, due to their long lifetimes and homogeneous spatial forcing. However, as discussed in Section 1.1.2, the region of emission is paramount when considering the effect of short-lived pollutants such as aerosols. We select broad regions to perturb the SO_2 emissions which will be scaled up with a scaling factor over this region, keeping the distribution of emissions constant. Most of these regions are selected so as to maintain consistency with previous studies, particularly those used to train the emulator described in Chapter 3, (Kasoar et al., 2018, 2016; Shawki et al., 2018). Specifically, I perturb SO_2 in North America, Europe, East Asia and South Asia (the same regions as Kasoar et al.

(2018); Lewinschal et al. (2019); Westervelt et al. (2020)). SO_2 emissions from North America and Europe peaked several decades ago and have been decreasing in the last few decades, whilst East and South Asian emissions were on the rise recently, although emissions from East Asia have started to fall since 2013 (Westervelt et al., 2020). We also extend our study beyond this by perturbing SO_2 in Africa and South America, which currently have low levels of SO_2 relative the to the rest of the world but are likely to see increasing emissions in the coming decades, due to changes in agricultural use in South America (Popp et al., 2017) and rapid growth of cities in Africa (Liousse et al., 2014).

Finally, we perturb the carbonaceous aerosols in the Tropics (between 24°S and 24°N), which covers tropical regions of Africa, South America and South East Asia, where biomass burning due to fires is the largest source of OC/BC (Pechony and Shindell, 2010; Van Der Werf et al., 2010; Tosca et al., 2013; Cachier et al., 1989). We perturb BC and OC in the same proportions, assuming that their ratio remains constant as done in other studies (Tosca et al., 2013; Hodnebrog et al., 2016). This would allow the emulator to be used to explore future scenarios under different levels of tropical biomass burning, whether it be due to anthropogenic suppression/ignition, a consequence of a warmer planet (e.g. alongside increasing GHGs) or under natural variability (Pechony and Shindell, 2010). Table 4.1 shows a summary of the pollutants and regions used, which gives a total of 9 input parameters.

Table 4.1: Summary of chosen emission inputs to emulator

Pollutant	Regions
CO_2	Global
CH_4	Global
SO_2	North America Europe East Asia South Asia Africa South America
Carbonaceous Aerosols	Tropics

In HadGEM3, to run a simulation with perturbed greenhouse gases (CO_2, CH_4), we would change the global mean atmospheric concentrations. To run a simulation with perturbed aerosol (SO_2, OC, BC), we would vary the emissions fields directly. To build the emulator, we first decide on the minimum and maximum values of these concentrations and emissions for each input (pollutant and region). This allows us to define the total parameter space that embodies all the possible scenarios that would be of interest to the user of the emulator (e.g. policy-makers, scientists). In the following section, I present the minimum and maximum values for each input and the justification for these choices.

4.3.2 Input Ranges

First, the range of input values for the emulator to cover is determined. As it is designed for exploring climate change projections, these input values are not necessarily precise or reflect a likely future scenario, rather they are chosen to be broad and to cover a wide range of scenarios. I assess the approximate minimum and maximum plausible values based on previous studies, which include projections of the Shared Socioeconomic Pathways (SSPs) (Riahi et al., 2017; van Vuuren et al., 2017) and Evaluating the Climate and air quality Impacts of short-lived pollutants (ECLIPSE) project (ECLIPSE, 2014; Stohl et al., 2015; Aamaas et al., 2017; Baker et al., 2015), as well as independent studies (Liousse et al., 2014; Pechony and Shindell, 2010). For the greenhouse gases, I take the pre-industrial levels as the minimum value and for the short-lived pollutants I take the minimum value as a scaling factor of zero. For all forcings, I determined the maximum levels to be the worst case estimate from both future projections and historical scenarios (up to pre-industrial) in previous studies. We use different studies to select the maximum values independently for each parameter so that the emulator can be used for the maximum possible range of predictions. Table 4.2 shows the ranges selected for each input and for the reasoning behind the maximum values. CO_2 and CH_4 are treated as global concentrations (for reference, the 'present day' (2014) concentration of CO_2 is 390 ppm and of CH_4 is 1813 ppb (van Vuuren et al., 2017)) and all other pollutant perturbations are treated as scaling factors relative to present day emissions.

Table 4.2: Input ranges for all input perturbations and associated reasoning for maximum value. Minimum is taken to be pre-industrial levels. All are scaling factors of present day emission levels except for CO_2 and CH_4 which are global concentrations. Note that some input parameters have two sources that are roughly equivalent for the maximum value.

Input Parameter	Range	Reasoning
Global CO_2 concentration	282 - 834 ppm	SSP3 baseline projection in 2100 (Riahi et al., 2017)
Global CH_4 concentration	248 - 3238 ppb	SSP3 baseline projection in 2100 (Riahi et al., 2017) and RCP3.4 projection (van Vuuren et al., 2017)
SO_2 North America	0 - 3×	Historical data from 1990 (ECLIPSE, 2014)
SO_2 Europe	0 - 5×	Historical data from 1990 (ECLIPSE, 2014)
SO_2 East Asia	0 - 2×	Historical data from 2005 (ECLIPSE, 2014) and SSP1 projection in 2100 (Riahi et al., 2017)
SO_2 South Asia	0 - 3×	ECLIPSE projection in 2030 (ECLIPSE, 2014)
SO_2 Africa	0 - 7×	Inventory based projection in 2030, (Liousse et al., 2014)
SO_2 South America	0 - 3×	SSP3 baseline projection in 2080 (Riahi et al., 2017)
OC/BC Tropics	0 - 2×	Estimates of fire activity by 2100 (Pechony and Shindell, 2010)

In Table 4.2, most of the maximum perturbation values are selected from either a historical database (ECLIPSE, 2014) or from SSPs (Riahi et al., 2017). The maximum value of SO_2 emissions over Africa is based on a bottom-up approach in Liousse et al. (2014), which suggests that growth in African combustion emissions could be much larger than SSP projections (Riahi et al., 2017) based on projected fuel consumption and emission factors. Future fire emissions are highly uncertain because they depend on a range of factors, including future climate from other pollutants, land use and human activity such as fire suppression. Projections of fire activity by Pechony and Shindell (2010) suggest that it could vary by ±100% relative to present day levels in the Tropics, which we can use to infer an approximate maximum of a 100% increase in biomass burning emissions. Note that these 'maximum' scaling factors do not reflect accurate or realistic scenarios, but instead they are only chosen as an approximate upper limit for scenarios that one may wish to emulate.

4.3.3 Distribution of SO_2 Emissions

The short-lived pollutant scaling factors are applied so that the distribution of emissions over each region remains constant. Previous studies which perturb SO_2 emissions over North America, Europe, East Asia and South Asia do this by multiplying present day emissions fields by a single scaling factor in each grid point (Kasoar et al., 2018; Lewinschal et al., 2019; Westervelt et al., 2020). In other words, they assume that while the level of emissions increases or decreases over the broad continental scales, the relative distribution of emissions within that region (e.g. on country scales) remains constant. The distribution typically attributes most of the emissions in the main urban areas, which is a reasonable assumption for these 4 regions as they have been strong emitters of SO_2 over the past few decades (Westervelt et al., 2020). However, this assumption is not necessarily valid for Africa and South America where present day SO_2 emissions are low and particularly isolated in nature, as shown in Figures 4.1a and 4.1c. Applying a scaling factor to these emission fields results in very high localised emissions in some regions but still virtually no emissions across the rest of the continent, which does not agree with the emissions pathways that we aim to project (e.g. rapid development across the continent). Liousse et al. (2014) suggest that a rapid growth in new cities and megacities in Africa may lead to a rise in SO_2 emissions which will take on a different distribution to the present day levels, as these cities will grow at a more rapid rate relative to regions already urbanised. For Africa and South America, we therefore use the emission distribution from an SSP scenario (specifically the SSP5 baseline scenario with rapid growth (Riahi et al., 2017)) while keeping the same total emissions, shown in Figures 4.1b and 4.1d. This does not have much effect on the response to

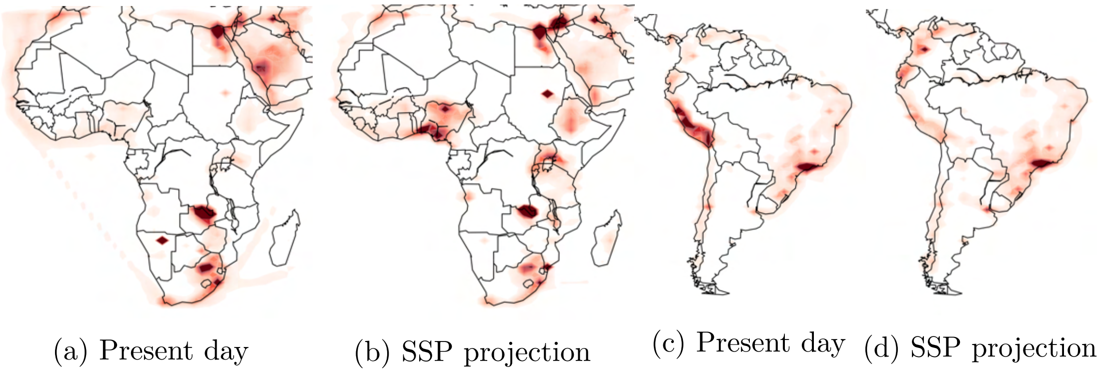
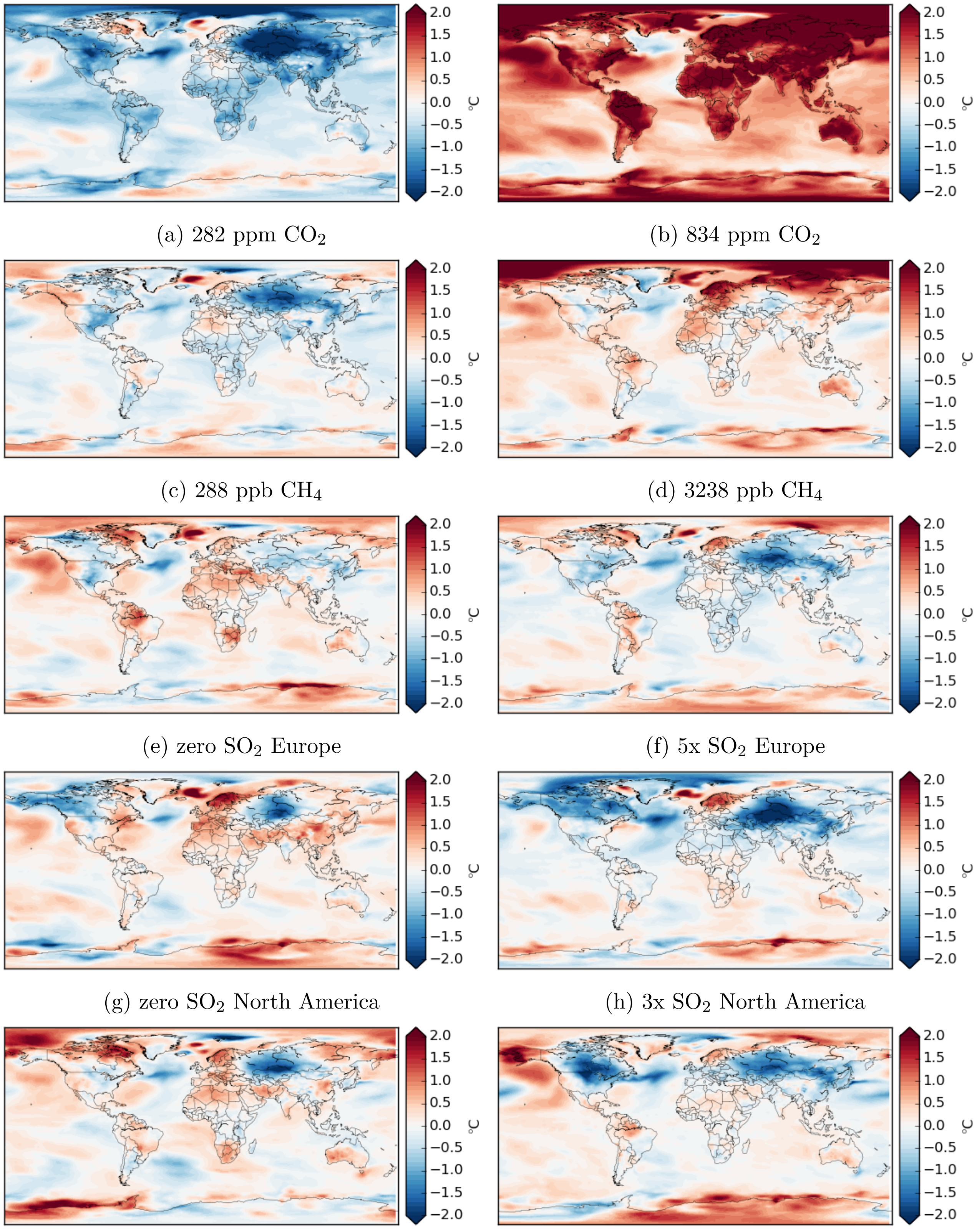


Figure 4.1: Distribution of SO_2 emissions for (a-b) Africa and (c-d) South America using present day estimates and SSP5 projections

present day or reduced emissions (since these perturbations are weak) but for strong positive perturbations, the response is strengthened, particularly for the emissions over Africa. This is because applying a strong scaling factor to present day emissions give very strong isolated sources of SO_2 where emissions already exist, which becomes saturated with sulfate aerosol. As the air is saturated with the aerosol, the climate response also becomes saturated, whereas new emissions over cleaner air are known to have a larger climate effect per unit increase in emission levels (Carslaw et al., 2013). For all other aerosol emission perturbations, we keep the present day distribution of emissions, remaining consistent with the approaches taken in previous studies (Kasoar et al., 2018; Lewinschal et al., 2019; Westervelt et al., 2020; Pechony and Shindell, 2010).

4.3.4 One At a Time Tests

Prior to developing the training simulations for the emulator, it is worthwhile running simulations at the maximum and minimum values of the input ranges decided, while keeping all other input parameters fixed. This is done to make sure that the input ranges are sensible and that they lead to significant changes in the output that would be relevant for an emulator to predict. The results of these are shown for each input parameter in Figure 4.2, with panels on the left showing the responses to the minimum possible values and panels on the right showing the responses to the maximum possible values.



(a) 282 ppm CO_2

(b) 834 ppm CO_2

(c) 288 ppb CH_4

(d) 3238 ppb CH_4

(e) zero SO_2 Europe

(f) 5x SO_2 Europe

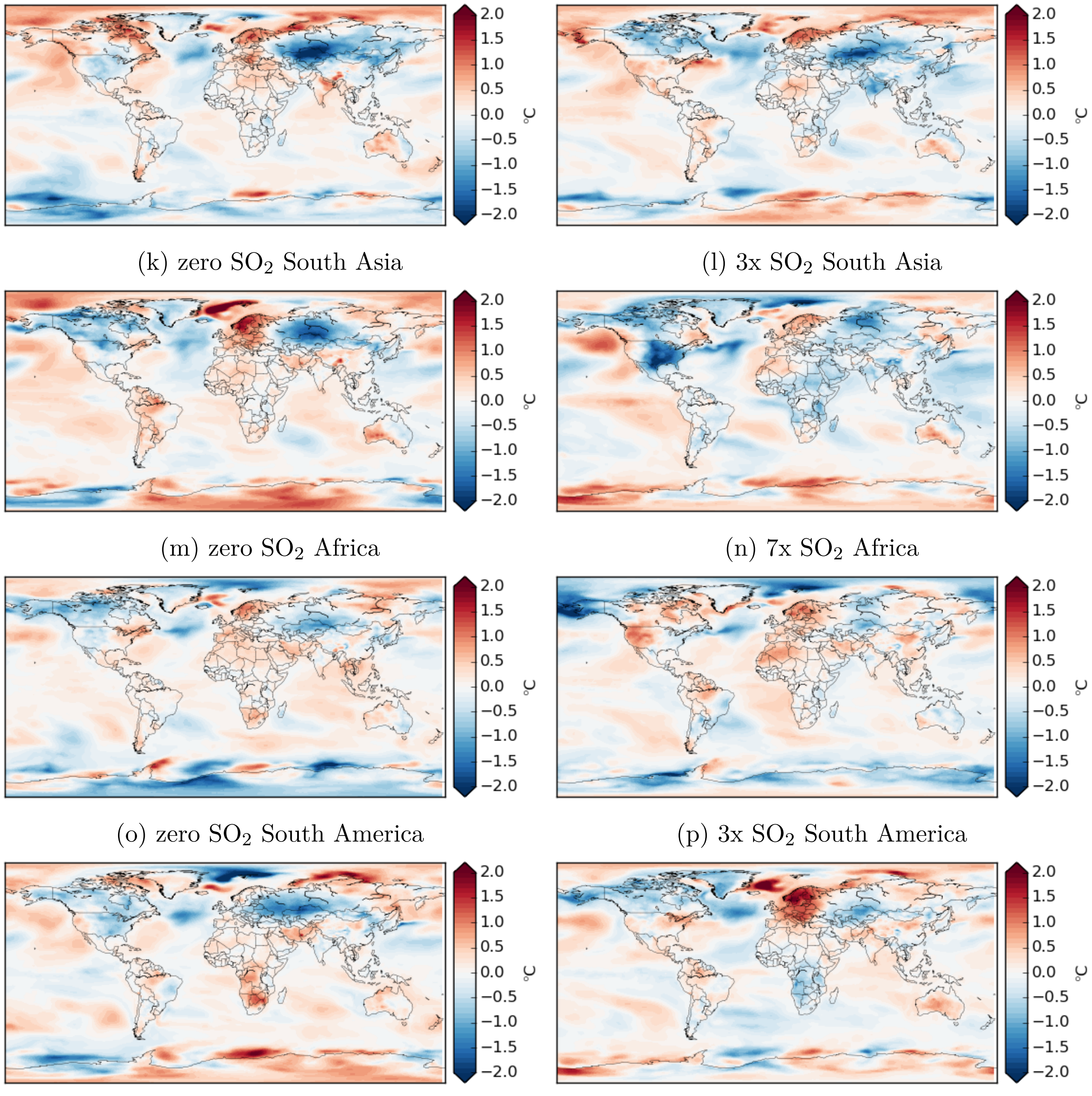
(g) zero SO_2 North America

(h) 3x SO_2 North America

(i) zero SO_2 East Asia

(j) 2x SO_2 East Asia

Figure 4.2: (a-j): 5 year averaged temperature response to one at a time simulations for first five input parameters, with minimum perturbation on left panel and maximum perturbation on right panel. Figure continued on following page for remaining input parameters.



(k) zero SO_2 South Asia

(l) 3x SO_2 South Asia

(m) zero SO_2 Africa

(n) 7x SO_2 Africa

(o) zero SO_2 South America

(p) 3x SO_2 South America

(q) zero OC/BC Tropics

(r) 2x OC/BC Tropics

Figure 4.2: (k-r): 5 year averaged temperature response to one at a time simulations for all input parameters, with minimum perturbation on left panel and maximum perturbation on right panel.

These responses are generally as expected with strong global responses to the long-lived GHGs and strong regional responses to the short-lived aerosol perturbations. They also exhibit similar response patterns over the Northern Hemispheric continents to SO_2 perturbations over North America, Europe and East and South Asia, as also found in Kasoar et al. (2018) (see Figure 1.5). Some of this response, however appears to be a feature of noise, with the strong cooling over Europe appearing in most aerosol perturbations, both positive and negative. Additional noise patterns over the Southern ocean also feature both positive and negative perturbations.

For this analysis, the simulations were also run out to 10 years, to help decide on the ideal length of time to be chosen as an output. 10 year simulations give higher signal-to-noise ratios, however, the choice of 5 years was made because of the high computational savings, allowing double the number of simulations to run given the same computational cost. The current GCM set-up allows these simulations to be extended if desired in the future.

4.3.5 Latin Hypercube Design

A typical approach to designing the set of training simulations is to generate samples that fill the parameter space (O'Hagan, 1978; Currin et al., 1991; Santner et al., 2003). This is a common procedure throughout in climate emulation studies (e.g. Wild et al., 2020; Ryan et al., 2018; Lee et al., 2016). A maximin Latin hypercube sampler (LHS) can be used to generate samples that cover the entire parameter space evenly and maximise distance between new samples (i.e. space-filling) (Santner et al., 2003). This improves the efficiency of samples compared to a random sampler, which is particularly important for this study as obtaining samples is expensive (around 80 node hours per sample on Cray XC40 Met Office). An informal rule of thumb is that the number of runs for training should be around 10 times the number of input dimensions (Loeppky et al., 2009). Given the 9 input parameters of interest in this study but the high computational cost of the simulations, we use 80 simulations for training selected with a maximin LHS design and an additional 18 simulations for testing.

The main parameter space of interest is determined by the input ranges selected in Table 4.2. However, as this emulator is designed for also exploring future unknown and possibly extreme forcing scenarios and there are high uncertainties on the short-lived pollutant estimates, we cannot rule out exploring beyond this input range for the short-lived pollutants. Furthermore, the fact that we break down the short-lived pollutants into smaller than global perturbations inevitably leads to lower abundances and weaker forcings. This can give weaker climate response signals that are difficult to separate from the internal noise (Kasoar et al., 2018; Shawki et al.,

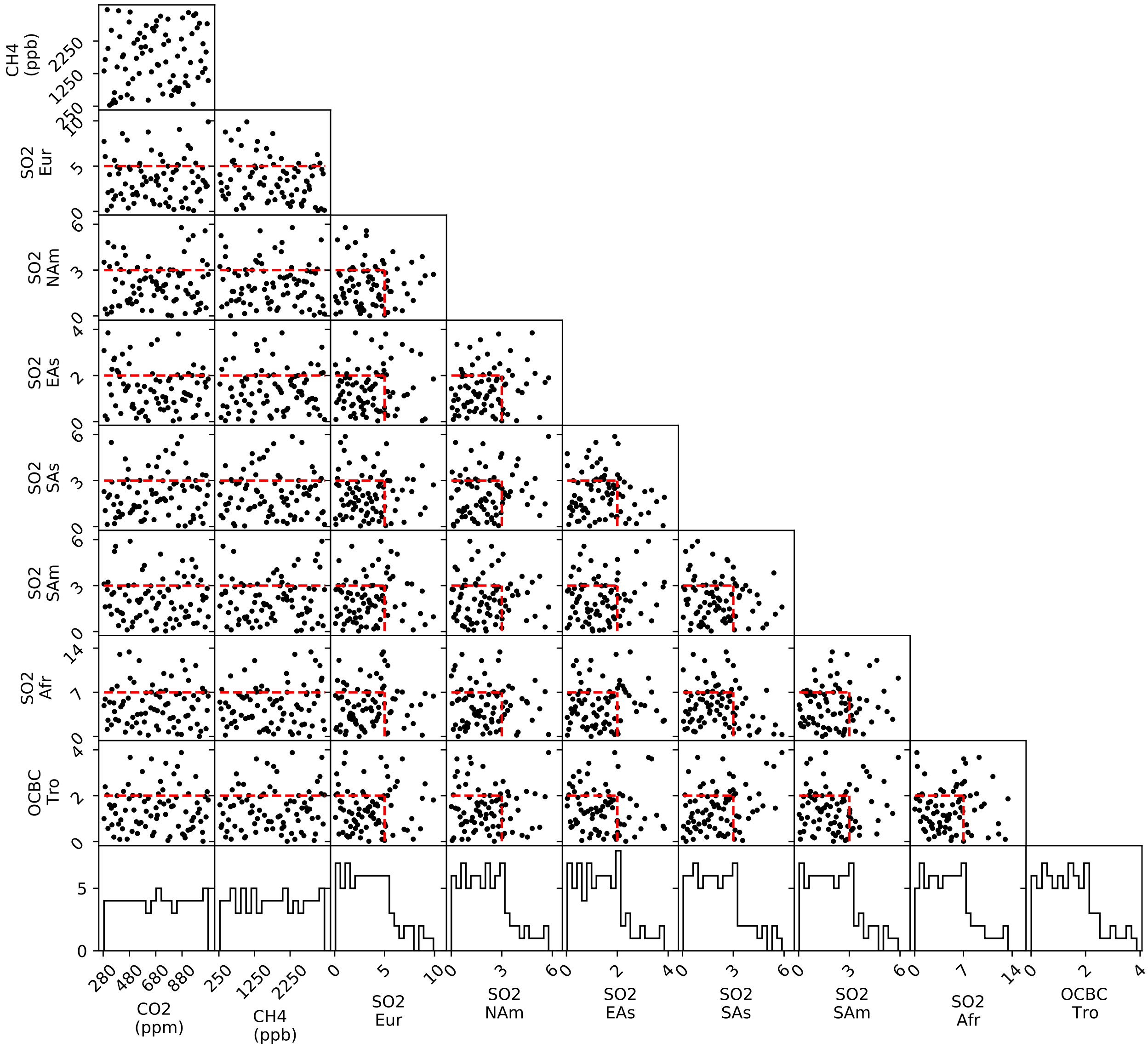


Figure 4.3: Training data generated from Latin Hypercube Sampler for each input against every other input. The bottom row shows the overall histogram for each input parameter independently.

2018). We expect that including some simulations with stronger perturbations, which should have larger signal-to-noise ratios, could be beneficial for constraining the emulator prediction to these perturbations. We therefore extend the parameter space so that 25% of samples are generated within the input range that is double that provided in Table 4.2 for the short-lived pollutants only. This means most of the training simulations fall under the feasible range determined by Table 4.2, with some additional simulations extending beyond this. Figure 4.3 shows the 80 samples generated from the maximin LHS for each combination of input parameters, with the bottom row showing the histogram of points for each input parameter independently.

4.3.6 Input-Output Relationships

As in Chapter 3, we first assess the nature of the data. The emulator output is multivariate as we are interested in emulating the temperature response at every grid cell. However, we first examine the relationships between each input parameter and the global mean temperature response in Figure 4.4. There is a strong linear relationship between the CO_2 concentration and the global mean temperature response, indicating that at the global scale, this input has the greatest effect.

At smaller spatial scales and after removing the first order effect of CO_2, we see regional patterns in the response. The CO_2 concentration is used to estimate the global mean response with a linear regression and the residual from this is shown in Figure 4.5, in each region local to the perturbation. This highlights that, for example, when SO_2 emissions are increased in Europe, the temperature response over Europe is decreased relative to the prediction made from CO_2 alone. The same trends are seen across all SO_2 perturbations to varying degrees, while the global effect of CH_4 appears to be positive, as expected. The effect of OC/BC on the Tropics is not so clear, but this is not surprising as an average is taken over the entire Tropics, a broad area that also includes other emission regions (South Asia, South America and Africa) which may have strong localised responses. Furthermore, this data is created from the LHS design, so each point consists of a combination of perturbations over the 9 input parameters. It should be noted that there are remote effects from the other perturbations that makes it difficult to disentangle the trends from simple scatter plots alone. In the next section, I will proceed to building the Gaussian process emulator and testing its performance on new, unseen test data.

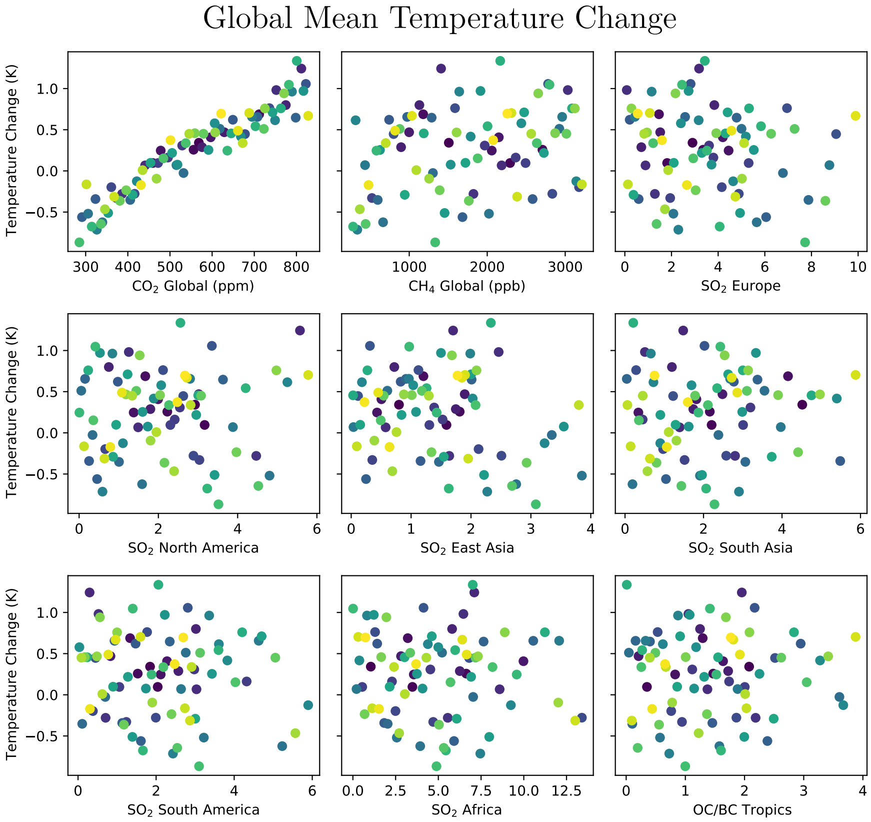


Figure 4.4: Global mean temperature response against each different input parameter. Each point is an independent simulation, with no importance given to the colours.

4.4 Emulator Method

In this set-up, the input vector, $\mathbf{x}$, is the vector of the 9 parameters that control the CO_2 and CH_4 concentrations and the emission scaling factors described above. The output, $\mathbf{y}$, is the multivariate, temperature response of the GCM averaged over the first 5 years. We build a Gaussian process emulator (Section 2.1.7) to estimate each grid-cell's response independently. Prior to doing so, the inputs and outputs are scaled to be centred around 0 with standard deviation of 1. The Gaussian process uses an additive kernel, composed of a linear kernel plus a squared exponential kernel (Equation (2.24)), decided after testing a range of different choices of kernel and finding this gives the best result. The linear component can be viewed as imposing

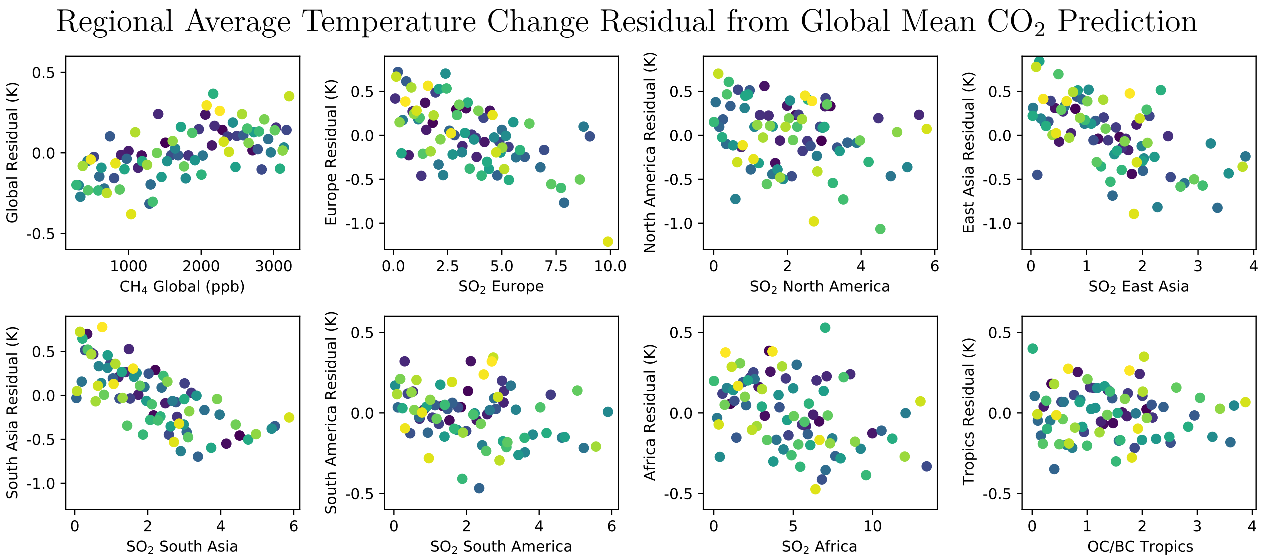


Figure 4.5: Regional mean temperature residual from the estimated response based on a linear regression against the global CO_2 concentrations alone, against each input parameter. Each point is an independent simulation, with no importance given to the colours.

a basic linear relationship between the inputs and outputs, while the squared exponential covers remaining non-linearities. The hyperparameters of these kernels are optimised by maximising the likelihood in the same procedure as Chapter 3 (GPy, 2014).

Additionally, I will treat the uncertainty due to the internal noise in the GCM separately from the Gaussian process uncertainty. The uncertainty due to internal noise, σ_{GCM} is estimated as the standard deviation between as many 5 year control runs as available. We have available 8 segments from one long, 40 year control run after equilibrium had been reached in the spin-up simulation and 6 independent 5-year control runs that branched off from this. Ideally, all control runs would be strictly independent but the 8 segments of the longer control run are included to improve the sample size for estimating the standard deviation. The standard deviation across the 14 control runs is calculated for each grid point independently, giving a map of uncertainty associated with the internal noise. This is an error associated with any prediction from the GCM, due to the limited number of years we have averaged over. Therefore, we will include this as a fixed noise term in our emulator, present for any prediction, even at the training data.

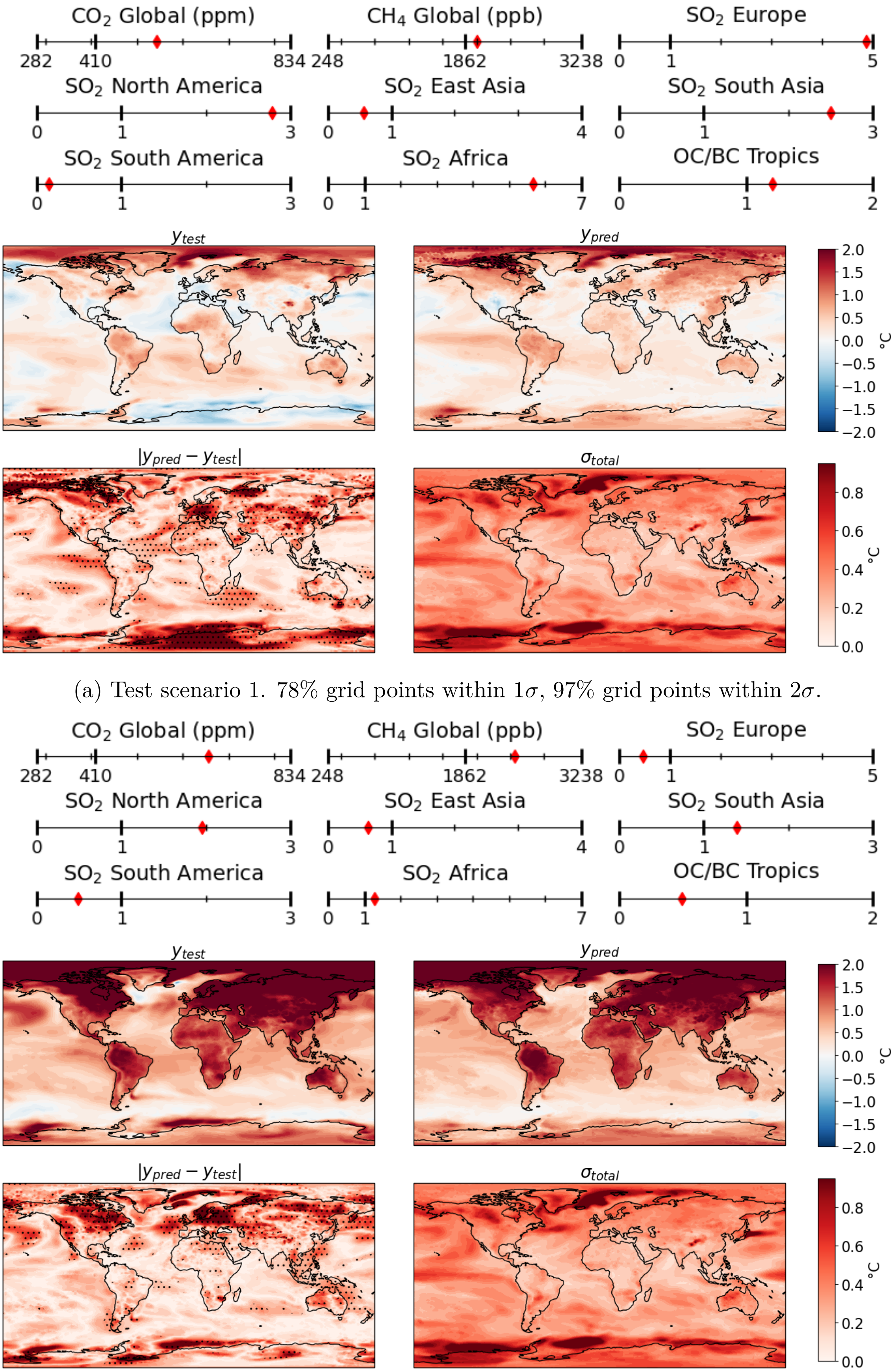
When the emulator predicts the temperature response at new unseen input values, there is an additional uncertainty introduced due to the probabilistic nature of the Gaussian process. This arises due to the distance between the new input values and the training data and is governed by the covariance function (see Section 2.1.7). We will call this σ_{GP}. The total uncertainty arising due to both the Gaussian process fit and the GCM internal noise is denoted σ_{total} and these are related via $\sigma^2_{\mathrm{total}} = \sigma^2_{\mathrm{GP}} + \sigma^2_{\mathrm{GCM}}$. These two forms of uncertainty are presented and compared in Section 4.6.

4.5 Emulator Performance

To test the performance of the emulator, we can predict the output for 18 test simulations, where the input values have been sampled randomly from a Latin Hypercube Sampler (Section 4.3.5), with a uniform distribution across all parameter ranges of interest (Table 4.2). This ensures the emulator is tested on samples from the main distributions of interest.

Figure 4.6 shows a few examples of the emulator performance, with the input values shown along the top panel, relative to the baseline, minimum and maximum possible values. The top left map shows y_{test}, the true prediction from the GCM, the top right shows y_{pred}, the prediction from the Gaussian process, the bottom left shows the absolute difference between these, $|y_{pred} - y_{test}|$ and the bottom right shows the 1 standard deviation error associated with this prediction, σ_{total} (which combines both the internal variability contribution and the GP estimate). The stippling in the bottom right figure shows the regions where the true GCM prediction lies outside of the 1σ range of the Gaussian process prediction, i.e. $|y_{pred} - y_{test}| > \sigma_{total}$. All 18 test simulations can be found in Figure A.2 of Appendix A.

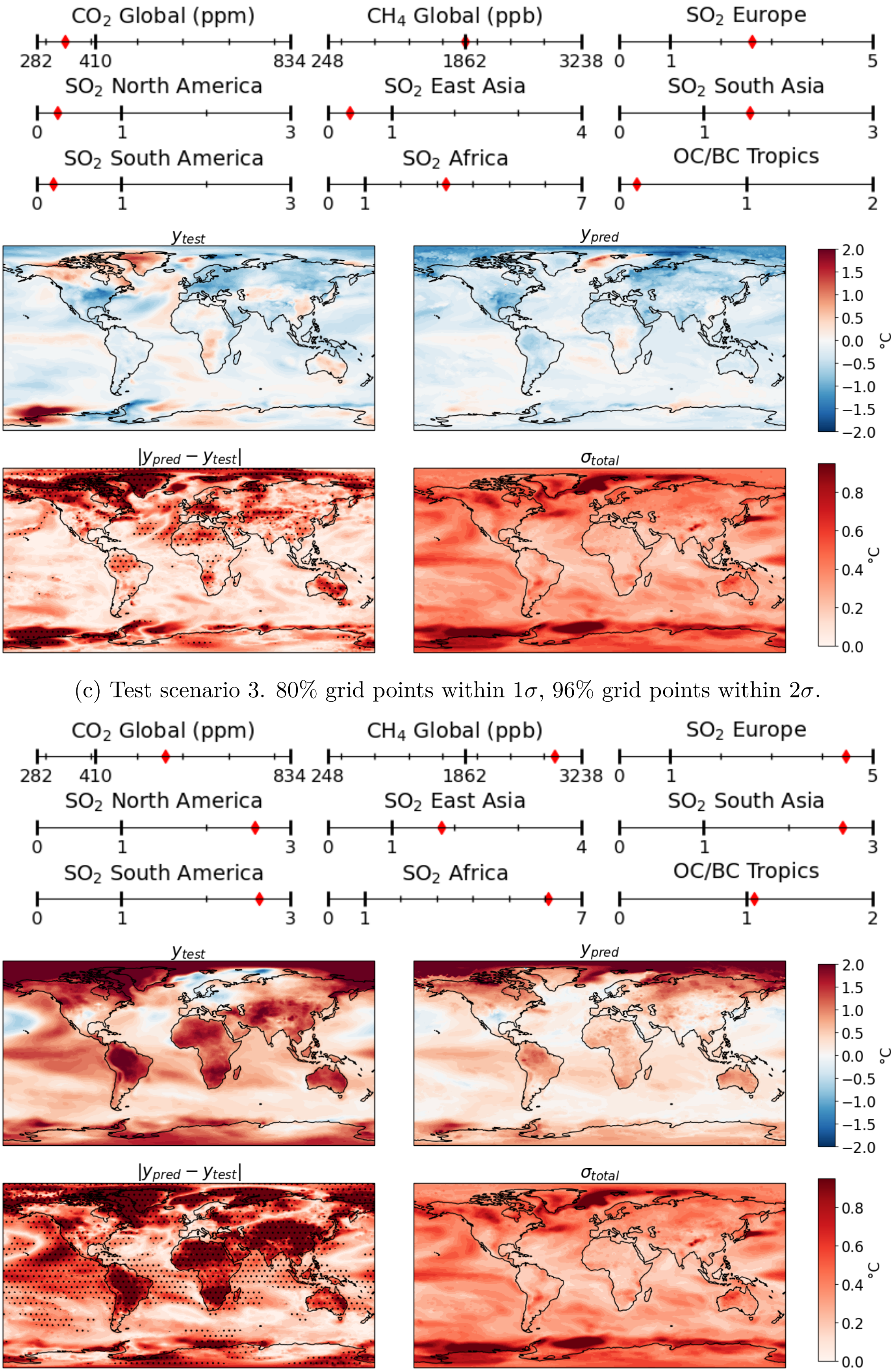
Most test simulations are reasonably accurately predicted by the emulator, as demonstrated in Figures 4.6a-c, which are chosen at random from the 18 test simulations. These show patterns of warming or cooling are typically predicted in the correct spatial location. The absolute difference between the GP and GCM prediction, $|y_{pred} - y_{test}|$ tends to be larger at high latitudes, particularly around the Arctic, North America, Greenland and Northern Europe. Although the magnitude of σ_{total} is slightly larger in these regions than the Tropics, the stippling where $|y_{pred} - y_{test}| > \sigma_{total}$ is more prominent in these high latitude areas, indicating that prediction is still poor here relative to other regions worldwide. Interestingly, these regions are highly



(a) Test scenario 1. 78% grid points within 1σ, 97% grid points within 2σ.

(b) Test scenario 2. 81% grid points within 1σ, 98% grid points within 2σ.

Figure 4.6: (a-b): Example test scenarios. Top panel shows input values, relative to the baseline, minimum and maximum possible values. The top left map shows y_{test}, the true GCM prediction, the top right shows y_{pred}, the GP prediction, the bottom left shows the absolute error, $|y_{pred} - y_{test}|$ and the bottom right shows the 1 standard deviation from GP, σ_{total}. Stippling on the bottom left indicates $|y_{pred} - y_{test}| > \sigma_{total}$.



(c) Test scenario 3. 80% grid points within 1σ, 96% grid points within 2σ.

(d) Test scenario 4. 41% grid points within 1σ, 76% grid points within 2σ.

Figure 4.6: (c-d): Example test scenarios. Top panel shows input values, relative to the baseline, minimum and maximum possible values. The top left map shows y_{test}, the true GCM prediction, the top right shows y_{pred}, the GP prediction, the bottom left shows the absolute error, $|y_{pred} - y_{test}|$ and the bottom right shows the 1 standard deviation from GP, σ_{total}. Stippling on the bottom left indicates $|y_{pred} - y_{test}| > \sigma_{total}$.

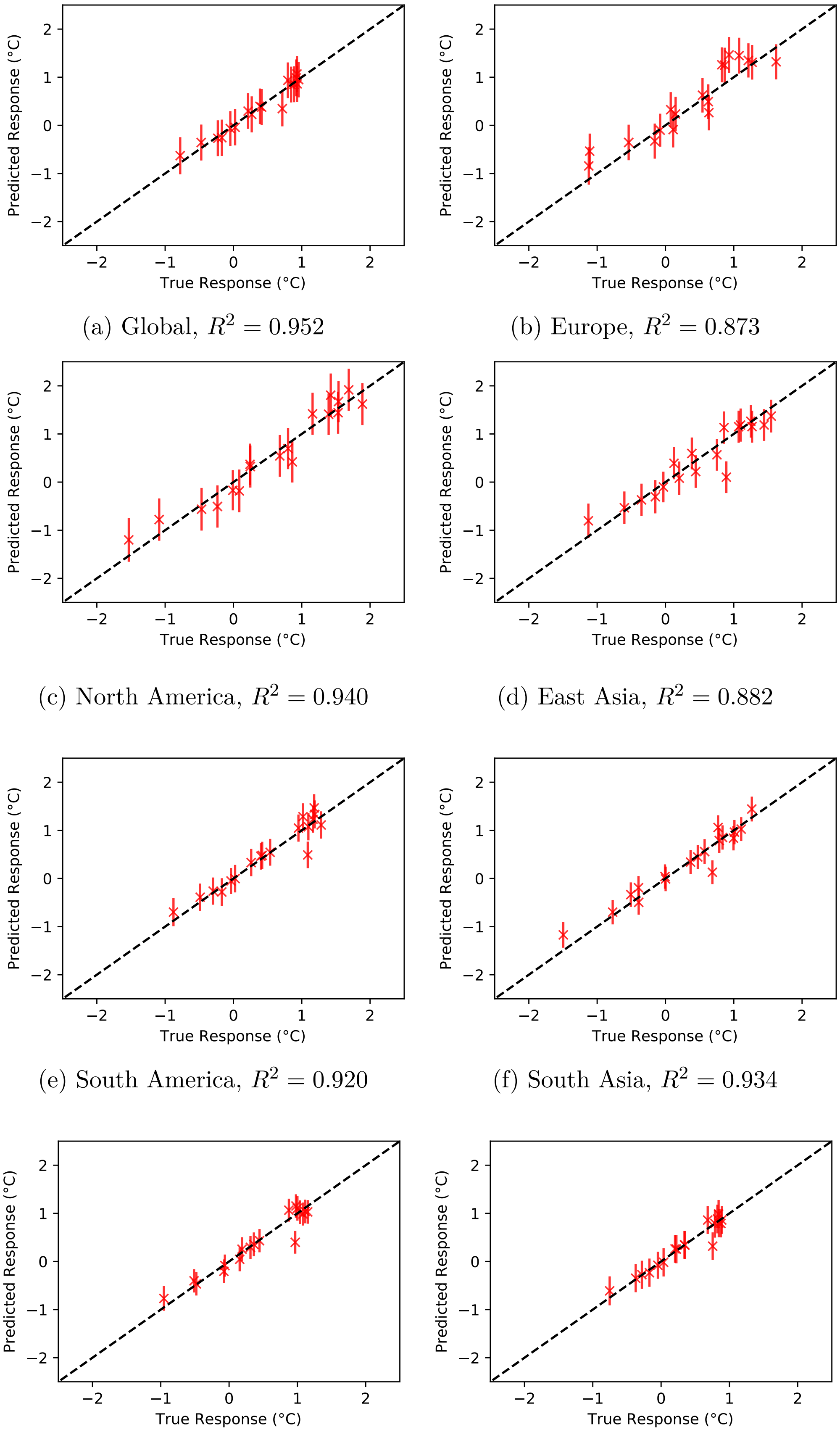
correlated with the regions that experienced larger errors in the regressions presented in Chapter 3, which were typically areas that experience both increased sensitivity to forcings and increased variability (e.g. Figure 3.10). Statistically, we would expect $\sim 32\%$ predicted points to fall outside the 1σ prediction level. However, based on the stippling across all test scenarios (Figure A.2), these points do tend to have this regional pattern and occur more often in the Northern Hemisphere, rather than being distributed randomly across the globe. This indicates a slight bias in these regions being predicted less accurately compared to the Southern Hemisphere, highlighting regions of the response that should be interpreted more cautiously.

In contrast to these example scenarios, which are generally well predicted, the emulator does not predict the test simulation in Figure 4.6d to the same level of accuracy. The input parameters in this scenario are positive relative to the present day levels for all inputs and are fairly strong across most of them. This means these points are fairly close to the corner of the input parameter space, where there are fewer training data points to constrain the prediction. Generally the emulator predicts the correct sign of response in most regions with general warming in most regions, but cooling over North America, the Pacific Ocean and parts of Europe. Furthermore, the relative patterns of warming are fairly consistent, for instance, the increased warming over South America, Africa and Australia relative to the global mean, but again the GP does not predict these patterns as strongly as the GCM. However, it is the magnitude of these warming and cooling contributions that are significantly weaker than the true GCM output, suggesting a global bias in the prediction in this regime of the parameter space, rather than a regional one. Furthermore, as discussed later in Section 4.7, a feature of the emulator is that it reveals a linear relationship between the SO_2 emissions and the temperature response. This is in contrast to GCM studies, that find that as the emissions are increased, there is a saturation of temperature response (Carslaw et al., 2013; Kasoar et al., 2018). The emulator, however, overestimates the cooling attributed to the aerosol pollutants when forced with strong aerosol perturbations, giving a cool bias when all SO_2 input variables are high, as in Figure 4.6d.

This test simulation highlights where caution should be taken when interpreting emulator results and, in particular, that strong perturbations on all input parameters can lead to poor predictions with an underestimated uncertainty estimate. Future studies may wish to refine the accuracy of the emulator in this region of the parameter space, by running additional training simulations with even stronger aerosol perturbations. This would be particularly important if the emulator were to be used to study this regime of the parameter space further. We will take care to avoid this regime in Section 4.8 when demonstrating an application of this emulator. Overall, however, the test simulations indicate that the Gaussian process emulator can predict the response to a range of scenarios under these 9 input parameters. Based on all 18 test simulations, 76.1% of grid points are predicted to within 1 σ of the true response and 95.0% are predicted to within 2 σ. This is roughly in line with expectations based on the definition of σ assuming a Gaussian distribution of responses.

Regional Mean Responses

Next, we explore the response for each region by taking broad averages over the key regions of interest. The response predicted by the GP is plotted against the true GCM response for all test scenarios in Figure 4.7, where the error bars indicate the 1 σ uncertainty levels. A perfect prediction would correspond to the $y = x$ line (black dashed line) and coefficient of determination, $R^2 = 1$. Most regional mean predictions fall either on this line, or within 1 σ from it. The exception to this is the outlier with a positive true response of around 1°C in most regions, that is strongly underestimated, which corresponds to the test scenario presented in Figure 4.6d. Other than this scenario, however, the majority of responses are well predicted and cover a fairly broad range of responses including both cooling and warming of up to 1-2 °C. The high R^2 values, all above 0.87, demonstrates the closeness of the predicted response to the true response across all predictions. These plots also highlight the increased uncertainty associated with some regions, namely North America and Europe, in comparison to the tropical regions such as South Asia, South America and Africa. As well has exhibiting increased uncertainties, these regions also show a wider range of temperature responses in the test data. The mean predictions for these regions are still close to the ground truth, as highlighted by the high R^2 values. This gives confidence to the use of an emulator in this way for quick predictions for instance, in policy-relevant studies. We will demonstrate an example of this in Section 4.8.



(a) Global, $R^2 = 0.952$

(b) Europe, $R^2 = 0.873$

(c) North America, $R^2 = 0.940$

(d) East Asia, $R^2 = 0.882$

(e) South America, $R^2 = 0.920$

(f) South Asia, $R^2 = 0.934$

(g) Africa, $R^2 = 0.929$

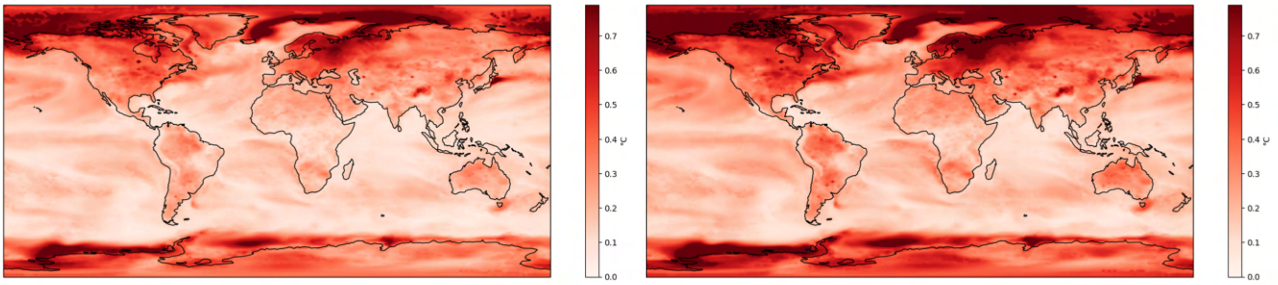
(h) Tropics, $R^2 = 0.935$

Figure 4.7: Regional mean predicted responses of test data against true GCM response, where error bars represent 1σ uncertainty level of Gaussian process prediction.

4.6 Emulator Uncertainty and Errors

I have discussed emulator performance and found accurate predictions on the test data in terms of the mean response and especially when accounting for the Gaussian process uncertainty. In this section, I will explore the patterns of uncertainty and errors further, in order to diagnose any pitfalls of the emulator.

Figure 4.8 shows a) the mean absolute error (MAE) and b) root-mean-squared error (RMSE) across all test simulations, the key difference being that the RMSE gives greater weighting to outliers with larger errors (e.g. the test simulation in Figure 4.6d). Both maps show the errors are greatest close to the poles and also at high latitudes over the continents, particularly on the Eastern side of North America and over Northern Europe, as noted in Section 4.5. Furthermore, the general patterns of MAE are similar to those seen in analysis of the Gaussian process regression in Chapter 3, where Figure 3.9 showed the MAE for the predicted long-term response. The MAE in the Gaussian process prediction (and other statistical approaches to prediction) also showed increased errors over high latitudes and Northern Europe, reduced errors in the Tropics and the highly localised errors over high altitude parts of East Asia. As noted previously, these regions typically respond more strongly to perturbations and exhibit increased variability, making these regions inherently more difficult to predict.

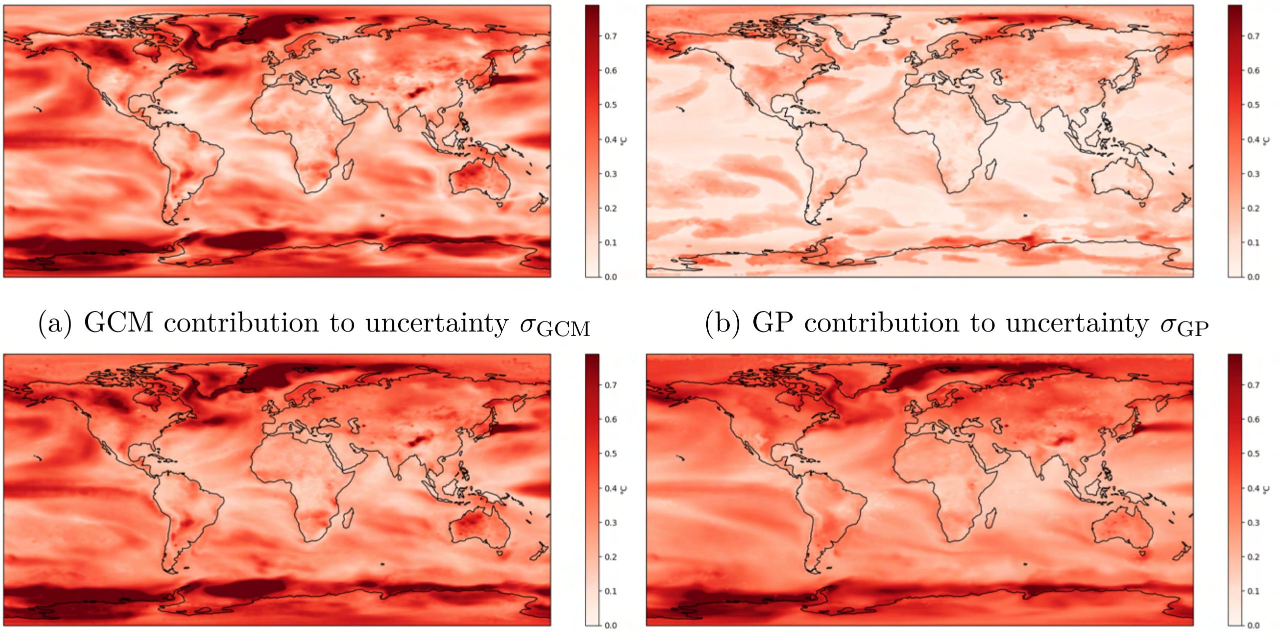


(a) MAE

(b) RMSE

Figure 4.8: Mean Absolute Errors (MAE) and Root Mean Squared Errors (RMSE) across all test data.

We used the internal variability of the GCM as a fixed component of uncertainty when building the emulator, as described in Section 4.4. Figure 4.9a shows this internal variability, calculated over 14 control simulations of 5 year segments. We see patterns associated with noise, such as those around the Southern Ocean, and around sea-ice over the Arctic. The GP provides a complete uncertainty estimate that includes this internal variability as a constant term. For each prediction, the GP uncertainty estimate is calculated from the covariance function (Equation (2.23)) based on the new input parameter values. We can decompose this estimated uncertainty into the fixed term, σ_{GCM} and the residual term which arises from GP covariance function, σ_{GP}. These are related with $\sigma^2_{\text{total}} = \sigma^2_{\text{GP}} + \sigma^2_{\text{GCM}}$. Figure 4.9b and 4.9c show the mean estimates across all test simulations for σ_{GP} and σ_{total} respectively. The majority of the total uncertainty comes from the fixed GCM term, with small additional contributions from the GP predominantly over the Arctic, the Antarctic and Northern Europe/Russia where σ_{GCM} is low.



(a) GCM contribution to uncertainty σ_{GCM}

(b) GP contribution to uncertainty σ_{GP}

(c) Total GP uncertainty σ_{total}

(d) GP uncertainty only σ_{GPonly}

Figure 4.9: Different Gaussian process uncertainty measures (a) GCM contribution to uncertainty which is the GCM internal variability, calculated as the average variance across X simulations of length 5 years, (b) GP contribution to uncertainty, which arises due to uncertainty in dataset (c) Total GP uncertainty, which includes contributions from both (a) and (b), $\sigma^2_{\text{total}} = \sigma^2_{\text{GCM}} + \sigma^2_{\text{GP}}$. (d) GP uncertainty when we do not specify a fixed component from GCM uncertainty in which case $\sigma^2_{\text{total}} = \sigma^2_{\text{GPonly}}$. Note that while (a) is fixed based on GCM simulations, (b-d) are a mean across all 18 test simulations.

As an additional experiment, the Gaussian process emulator was also built without including the fixed contribution σ_{GCM}. This emulator learned almost exactly the same distribution of uncertainties, shown in Figure 4.9d and performs virtually identically to the emulator that does include σ_{GCM} as a fixed component. This result is encouraging, as it suggests that even without knowledge of the GCM internal variability, the same patterns associated with these are learnt through the Gaussian process anyway. The only differences in Figure 4.9c and 4.9d appear to be the higher uncertainty regions over Eastern North America, Greenland, the Northern Hemispheric jet streams and the Inter Tropical Convergence Zone around the equator over the Pacific Ocean. This would then lead to an overestimate in confidence in predictions in these regions. However, these regions are fairly small areas and in most regions showing that the choice is not crucial and that future studies could implement a GP without necessarily needing to specify the error in this way.

4.7 Sensitivity Analysis

The climate modelling community has benefited from a wide range of Gaussian process emulation studies concerned with uncertainty quantification and sensitivity analysis, which aims to characterise the uncertainty on the outputs and identify where it arises from. Both uncertainty quantification and sensitivity analysis require a large set of often computationally heavy simulations, which is why Gaussian process emulators have become a popular tool in these communities to speed up model evaluations. These types of studies have been utilised to diagnose uncertainties and sensitivities in a wide range of GCM components and settings, including atmospheric chemistry models (Lee et al., 2012; Wild et al., 2020; Beddows et al., 2017), sea-ice models (Edwards et al., 2019; Urrego-Blanco et al., 2016), climate-vegetation models (Bounceur et al., 2015) and convection parameterisations (Souza et al., 2020). The inputs to these are typically climate model parameters which are not known perfectly, where gaining an understanding of how sensitive the outputs are to each input parameter is beneficial to the modelling groups. In this study, the input parameters are emission perturbations and we are primarily interested in exploring patterns of potential future climate responses to a range of scenarios. The goal is to develop a deeper understanding of climate response to competing pollutants and to explore the sensitivity of the different world regions to different types of emissions.

A naive approach to sensitivity analysis would involve varying one input parameter while keeping all others fixed at a baseline value, in 'one-at-a-time' experiments similar to those perturbations explored in Section 4.3.4. This does not, however, explore fully the sensitivity to each input under different possible situations in terms of other inputs and it ignores any interaction effects entirely which can lead to biased results (Saltelli and Annoni, 2010; Saltelli et al., 2019). Instead, we will carry out a *global sensitivity analysis*, which quantifies sensitivities to each input parameter averaging over the effects of all other input parameters. The input parameters, $\mathbf{x}$, are indexed from $i = 1, \cdots, 9$ and the output variable is the entire map of responses, $\mathbf{y}$, for grid cells $k = 1, \cdots, m$.

4.7.1 Main Effects

Here, we will explore the *main effect* of each pollutant, by running the emulator with perturbations from the minimum to the maximum value of each input range. This is repeated for many independent realisations where the other input parameters are sampled at random. The mean behaviour of each parameter perturbation is calculated by averaging over the different realisations. Figure 4.10 shows the resulting main effect on each key region (y-axis) to each of the pollutant perturbations (x-axis). Here, 200 different realisations are run with input parameters sampled from a normal distribution centred at the present day levels with a standard deviation 1/4 of the input range, ensuring the average behaviour of all simulations does not deviate significantly from present-day levels. The black line shows the mean over the 200 different realisations, while the red lines show 1 standard deviation across these different realisations (not the GP uncertainty).

We find that the main effect is linear for all pollutants, both GHGs and aerosols, over the ranges explored here. This is intuitive under small forcing perturbations, as we have long known that the relationship between emissions and response is fairly linear at low forcing levels. In fact, it has long been used as a method to estimate climate response given forcing, via the equilibrium climate sensitivity, for GHGs (Hansen et al., 1997; Boer and Yu, 2003; Gregory et al., 2004) and aerosols, using an adjusmtment based on individual aerosol efficiacy (Collins et al., 2013b; Hansen et al., 2005; Richardson et al., 2019). Although many of these studies assume the equilibrium climate sensitivity to be constant to the first order, other studies have demonstrated that it can change under a changing climate (Friedrich et al., 2016; Gregory and Andrews, 2016) or under larger magnitude forcings (Rugenstein et al., 2020; Knutti et al., 2017; Zhu and Poulsen, 2020; Meraner et al., 2013) . One may not expect the emulator to capture large non-linearities due to a changing climate (or only to the degree that they manifest themselves within the first five years) as it is trained on abrupt perturbations from a fixed initial climate. However, based on past studies, we may expect to see non-linearities caused by particularly large forcings in this study. The CO_2 perturbation range selected here gives the largest range of forcing perturbations (up to ∼2xCO_2). However, it does not extend as far as the forcings explored in previous studies where the relationship between forcing and response is found to deviate from linearity (up to 4xCO_2 from pre-industrial levels) (Rugenstein et al., 2020; Knutti et al., 2017; Zhu and Poulsen, 2020; Meraner et al., 2013). Furthermore, many of the non-linearities are explained by transient effects, that would not be captured in the short 5 year simulations used to train the emulator. Knutti et al. (2017) find that the first few years of a model simulation gives a linear relationship between forcing and temperature response and that the slope of this relationship changes at longer timescales of 50-100 year, due to slow climate adjustments, feedbacks and as warming patterns change over time. For instance, CO_2 sinks, such as the ocean, become less efficient as they become saturated (Gregory et al., 2015), which is an effect that would not been seen in the short 5 year timescales predicted here but could lead to increased response at high forcings on longer timescales (Ceppi et al., 2017). Other feedbacks include the positive water vapour feedback in which increasing temperatures allow the atmosphere to hold more water vapour, a greenhouse gas (Meraner et al., 2013), and the negative lapse rate feedback in which surface warming caused by increased CO_2 weakens the lapse rate, making the climate less sensitive to further warming (Colman and McAvaney, 2009). These feedbacks appear after the climate has already warmed significantly, which requires some time even after an abrupt forcing perturbation. The main effects of emulators built on longer timescales may reveal these non-linearities in the long-lived climate pollutants.

Past studies also suggest that we would expect a saturation effect of temperature response as the short-lived pollutants increase. Firstly, the SO_2 perturbations made here are a precursor to the sulfate aerosol SO_4. SO_4 formation begins with nucleation where the precursor gas forms clusters of sulfuric acid (Curtius, 2006). This is more efficient in cleaner air, where new clusters form as opposed to existing clusters growing in size (Carslaw et al., 2013). Therefore increases in SO_2 in regions with lower pollution levels directly leads to a larger increase in SO_4 compared regions with highly polluted air. Secondly, a greater increase in concentration of cloud condensation nuclei (CCN) also leads to a greater increase in cloud droplet number because higher droplet concentrations suppress supersaturation within clouds. Finally, increased water droplet number enhances the cloud albedo effect more when there are fewer water droplets, with the sensitivity of cloud albedo falling with $1/N$ for N cloud droplets.

As a consequence of this, several studies have found that regions with lower SO_2 levels experience greater climate response to increases in SO_2 per unit emission, (Collins et al., 2013b; Liu et al., 2018b; Kasoar et al., 2018). Following this reasoning, as SO_2 emissions are increased, we would expect the decrease in temperature to decelerate, which would lead to a decreasing gradient at high SO_2 perturbations in Figure 4.10 (Carslaw et al., 2013). Unlike the GHG perturbations, these effects are not limited by the short 5 year timescale of the output. However, the main effect of each perturbation appears linear across the entire perturbation range, over all regions. The fact that these non-linearities are not learned by the emulator may explain why the scenario predicted in Figure 4.6d. These non-linearities may be too small to be realised by the emulator given the reduced number of training points at high perturbations and the significant uncertainty associated with the internal variability. Increasing the number of training points at large perturbations and increasing the input ranges to include more extreme forcing may be beneficial for better constraining the response under strong aerosol perturbations.

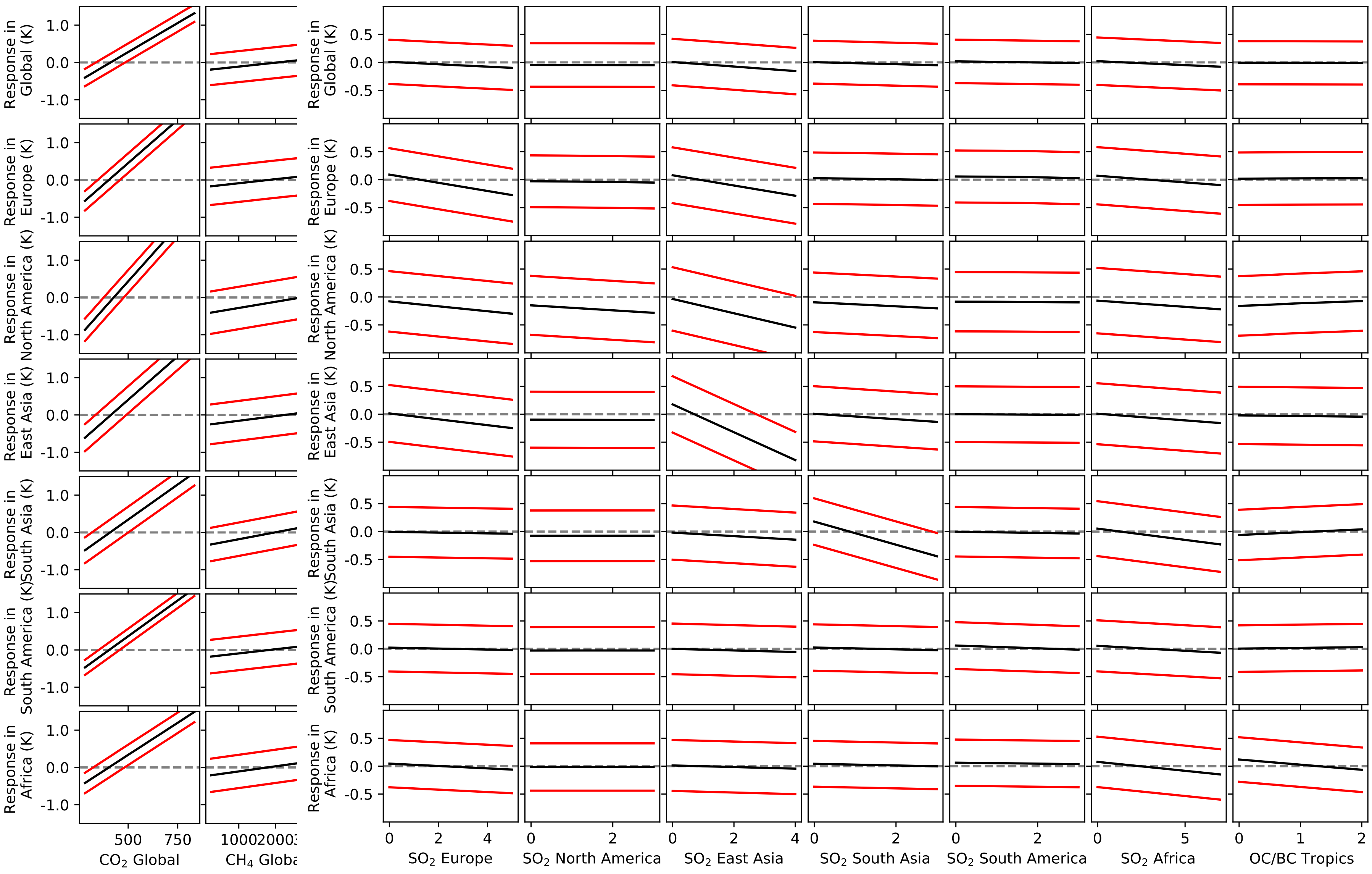


Figure 4.10: Main effects of each pollutant on x-axis on each region of interest on y-axis, averaging over all other pollutants. The black line shows the mean across many realisations run with randomly sampled input values for all other pollutants. The red lines show 1 standard deviation across these different realisations. A steeper gradient indicates stronger change due to that pollutant.

Another key feature of Figure 4.10 is the diversity of the regional response to the regional perturbations. As expected, local perturbations give rise to stronger responses, particularly European response to SO_2 from Europe, East Asian response to SO_2 from East Asia and South Asia response to SO_2 from South Asia. Furthermore, often the second most important contributor is an emissions perturbation in an area that is closer than others. For example, aside from African emissions, the response over Africa is sensitive to SO_2 from Europe, consistent with literature that finds Africa responds to European aerosol perturbations due to interactions with radiation and clouds and changes to the hydrological cycle and atmospheric circulation (Dong et al., 2014). Remote regions show reduced response, often with a gradient of approximately zero. The response to North America SO_2 is particularly low across all regions, including in North America. This perturbation, however, is weaker in terms of total maximum SO_2 perturbation ($\sim$10 Tg) compared to, for instance, the East Asian SO_2 perturbation ($\sim$80 Tg), based on the maximum levels of historical emissions.

The main effects strongly depends on the magnitude of the perturbations. One may wish to interpret the aerosol perturbation main effects in terms of the magnitude of emissions rather than scaling factors, relative to present day emissions. Figure 4.11 shows the main effects of each pollutant per unit Tg of emissions released, in each emission region. This is calculated by estimating the gradients of the main effects in Figure 4.10 assuming linearity. The error bars represent the 1 standard deviation uncertainty propagated from Figure 4.10.

These demonstrate similarities to previous studies, which estimate the temperature change in the first 20 years of response to an abrupt response per unit Tg emissions (Aamaas et al., 2017; Shindell and Faluvegi, 2009; Collins et al., 2013b). For instance, Aamaas et al. (2017) find European SO_2 perturbations to consistently produce larger responses per unit Tg than East Asian SO_2. Shindell and Faluvegi (2009) also find that broad latitudinal SO_2 perturbations in the Tropics give fairly equal responses globally, while Northern hemispheric perturbations responses are much stronger in the latitude band of emission relative to other regions. Figure 4.11 shows consistencies with this as the Northern hemispheric perturbations show much stronger regional effects in other Northern hemispheric response regions while tropical perturbations (SO_2 Africa, South Asia, and South America) show more evenly distributed responses, after accounting for the localised effect. Note the key difference in this study is that we have averaged over the effect of other pollutants, rather than keeping them fixed at present day levels and that these show only the 5 year response, rather than 20 years. However, this is the first study to explore the response to regional tropical perturbations (SO_2 South America, SO_2 Africa and OC/BC Tropics), rather than broad latitudinal perturbations.

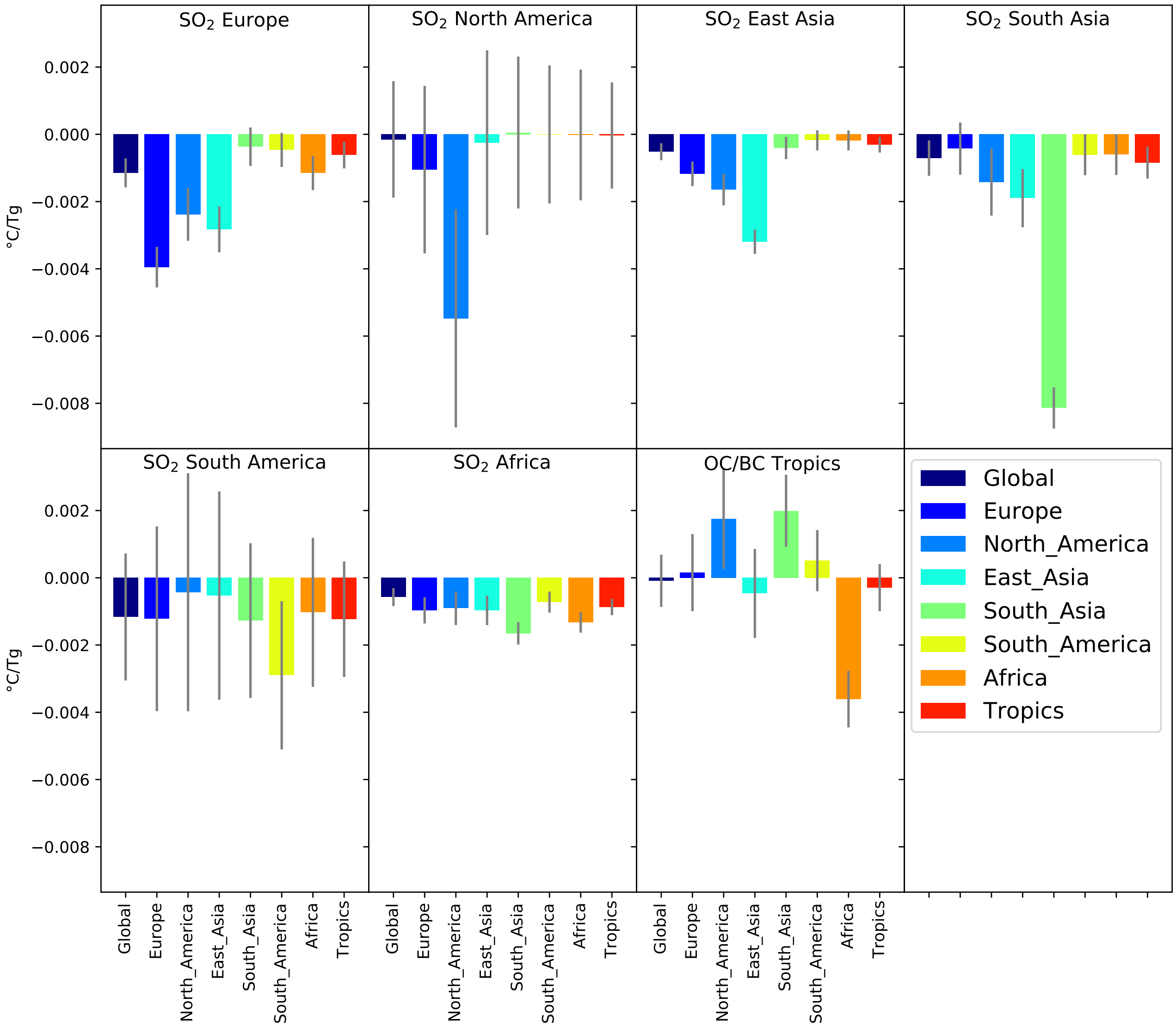


Figure 4.11: Main effects of each aerosol pollutant per unit Tg of emissions in each response region. The error bars represent 1 standard deviation uncertainty estimated by the different realisations in Figure 4.10.

4.7.2 Sensitivity Indices

We will also carry out a variance-based sensitivity analysis, by estimating how much of the variance in the output can be attributed to the variance in the input parameters, while averaging over all other inputs as before. Following the notation of Saltelli et al. (2010), X_i will be used to denote the i-th input, $Y|X_i$ to denote the estimated output due to input variable X_i. We will average over all other input variables and denote this $E_{X_{\sim i}}(Y|X_i)$. The variance due to input variable X_i is then denoted $\text{Var}_{X_i}\left(E_{X_{\sim i}}\left(Y|X_i\right)\right)$. The first order sensitivity index (SI) is given by

$$SI_i = \frac{\text{Var}_{X_i}\left(E_{X_{\sim i}}\left(Y|X_i\right)\right)}{\text{Var}(Y)} \tag{4.2}$$

where $\text{Var}(Y)$ is the total variance over all inputs. Higher order effects can also be estimated from this as the difference between 1 and the sum of the first order indices for all input variables.

Calculating Equation (4.2) exactly is expensive, but we can estimate it using the Sobol method (Saltelli et al., 2008). We introduce two randomly sampled matrices of the input parameters, with 9 columns, indexed by i for each input variable and N rows indexed by n for each independent sample. For this, we uniformly sample across the 'maximum feasible' ranges for all inputs described in Section 4.3.2, although we note that these are only approximate. These matrices are denoted $\mathbf{A}$ and $\mathbf{B}$. From these, we can construct a new matrix, $\mathbf{A_B}^{(i)}$ where all columns are from $\mathbf{A}$ except the i-th column, which is from $\mathbf{B}$. The variance due to input variable i can then be estimated with

$$\frac{\widehat{\text{Var}}_{X_i}\left(E_{X_{\sim i}}\left(Y|X_i\right)\right)}{\text{Var}(Y)} = \frac{1}{N}\sum_{n=1}^{N} f\left(\mathbf{A}_n\right) f\left(\mathbf{A_B}_n^{(i)}\right) - E(Y)^2 \tag{4.3}$$

where the proof for this can be found in Saltelli et al. (2010).

We use the Sobol method to estimate the first order sensitivity indices for each of the 9 input parameters independently for each output variable, i.e. for each grid point. Firstly, we find the that in all grid points, the variance due to the CO_2 concentration is responsible for the majority of the variance in the emulator. It is not necessarily surprising that the GHGs dominate the variance, given the wide input ranges covered by the emulator based on the high uncertainty of future emissions. Figure 4.12 shows the degree to which the above is the case for the major regions of interest, through stacked bar plots showing the sensitivity indices for CO_2, CH_4 and the sum of the aerosol pollutant perturbations studied here. There are significant differences across different regions, with South Asia standing out as the main region that is more sensitive to both CH_4 and aerosols (predominantly the South Asian SO_2 perturbation). This appears part of a tendency in which tropical regions (South Asia, Africa, Tropics) are more sensitive to CH_4 while higher latitude regions (e.g. North America, Europe) are more sensitive to the CO_2 perturbation. This is likely a consequence of enhanced warming at mid- and high-latitudes under CO_2 forcings that is not as strong under CH_4 forcings (e.g. one-at-a-time perturbations in Figure 4.6).

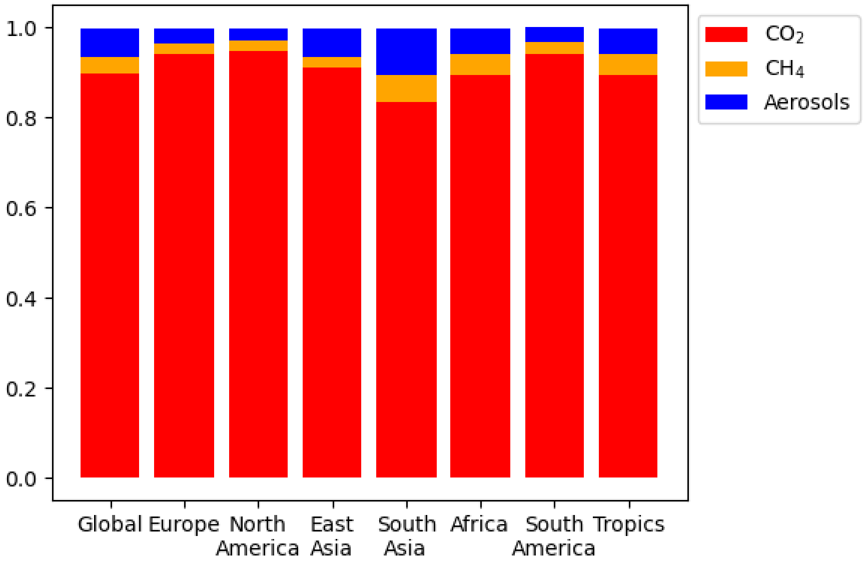


Figure 4.12: Regional mean first order sensitivity indices for CO_2, CH_4 and aerosols.

The first order sensitivity indices in CO_2, CH_4 and the aerosols sum to 1 almost exactly, leaving $< 10^{-4}$ down to higher order terms, which would describe sensitivity due to interactions between multiple perturbations. This falls inline with aerosol perturbation studies that have found the effects of multiple perturbations across different regions to be approximately additive at low levels (Kasoar et al., 2018), but at high levels we may expect the atmosphere to become more saturated with aerosol particles, reducing the efficiency of subsequent perturbations as described in Section 4.7.1. Furthermore, previous studies have found non-linear effects when perturbing GCMs with both GHGs and aerosols (Ming and Ramaswamy, 2009; Feichter et al., 2004; Marvel et al., 2015). The lack of interaction terms between these perturbations indicates these are not significant on the timescales performed here. As with any non-linearities due to GHGs alone, these would take longer than 5 years to affect the temperature response, as climate feedbacks caused by temperature change takes time to be realised.

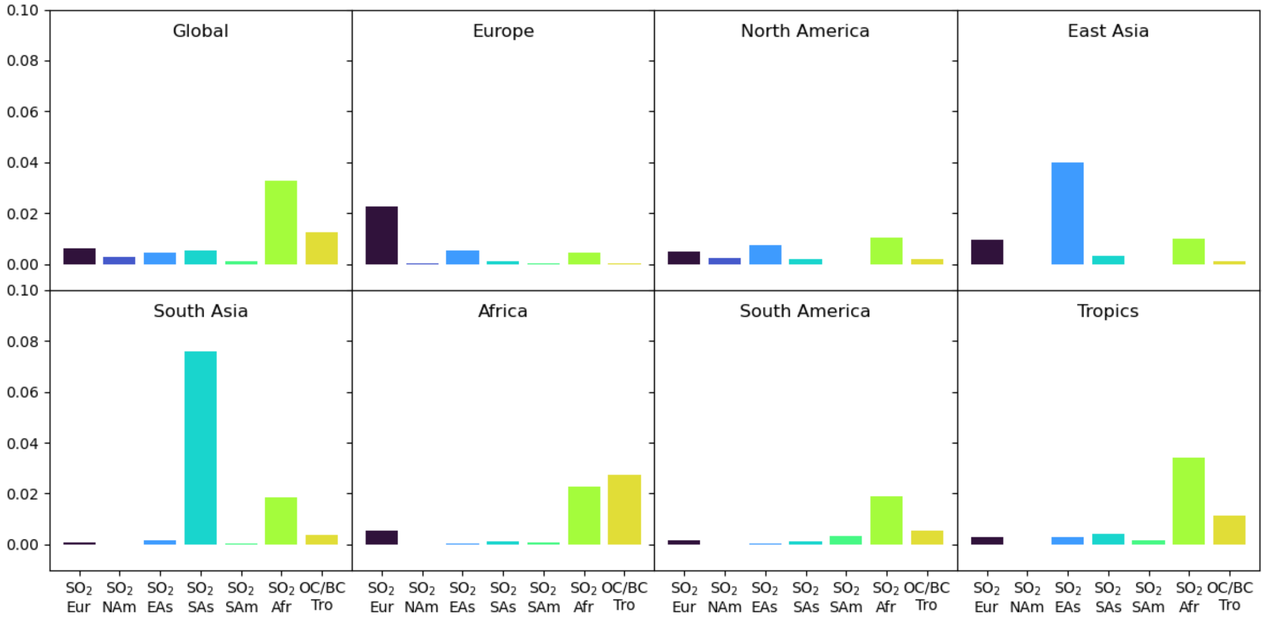


Figure 4.13: Regional mean first order sensitivity indices for aerosols perturbations: the regional SO_2 perturbation and the tropical biomass burning aerosols (OC/BC).

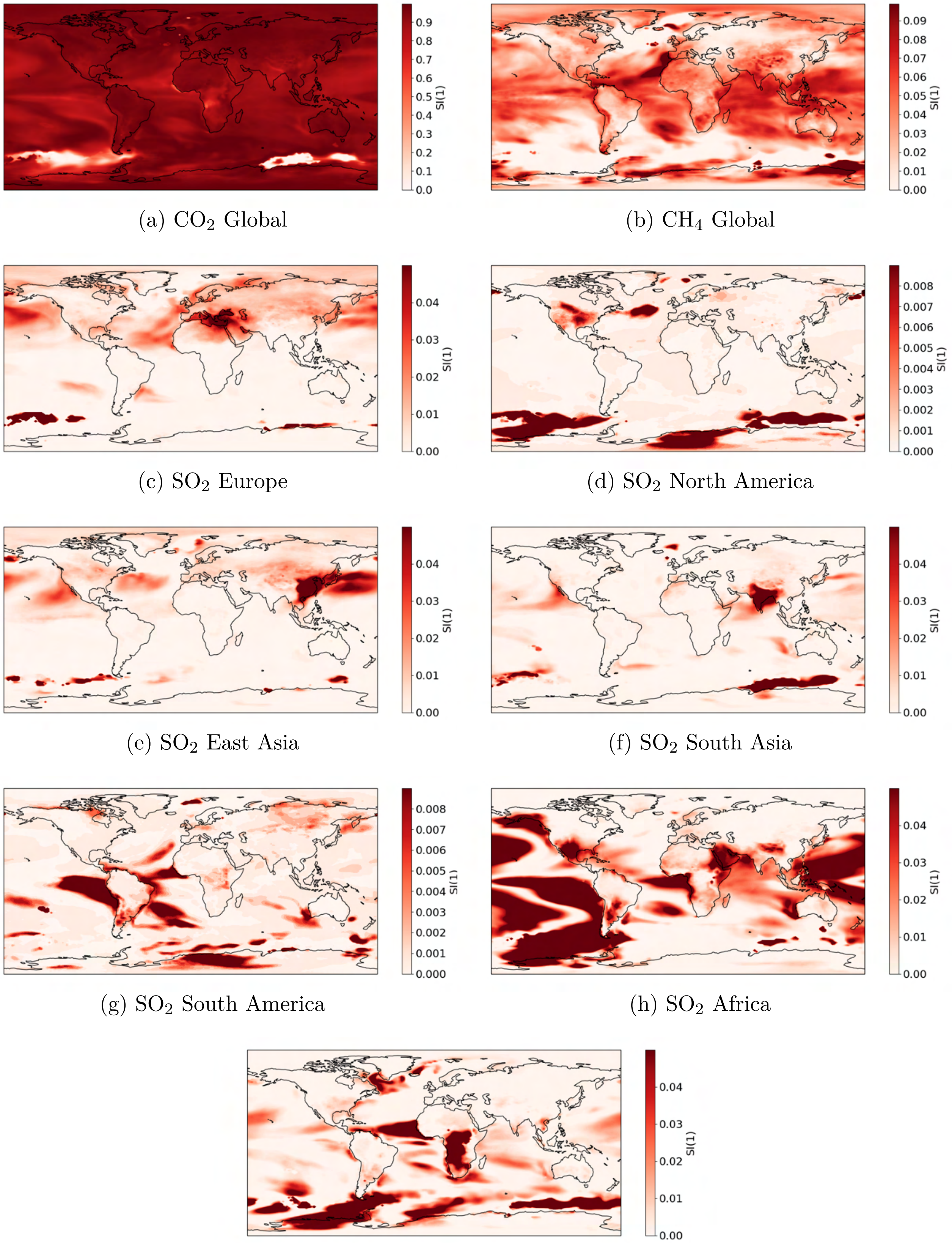
The sensitivity indices for the aerosol perturbations in these regions are shown in Figure 4.13. These show a high dependence on region, particularly, from the local perturbations in Europe, East Asia, South Asia and Africa. Although there are slight increases in sensitivities in North America and South America due to the localised perturbations, these sensitivity indices are still low, even relative to remote perturbations, such as SO_2 over Africa. This is partially because the SO_2 perturbations over North America and South America are considerably weaker than over Africa, due to the high uncertainty in future emissions in Africa (Liousse et al., 2014).

Regardless of the localised effect, Figure 4.13 also highlights significant differences in the sensitivity indices in terms of the latitudinal band of the region of response. In the top row, which consists of sensitivity indices over Northern mid-latitude regions (Europe, North America and East Asia) the sensitivity indices for European and East Asian SO_2 are relatively larger. In the bottom row, showing sensitivity indices over tropical regions (South Asia, Africa, South

America and the Tropics), the sensitivity indices of African SO_2 and tropical biomass burning are most relevant. This is consistent with other aerosol perturbation studies that have found the response to aerosol perturbations strongest in the latitudinal bands of emission, when one at a time perturbations are carried out (Shindell and Faluvegi, 2009).

This is revealed further in Figure 4.14, which shows the complete maps of sensitivity indices for each pollutant, on independent scales based on their relative magnitudes. Firstly, as expected, these show stronger sensitivity indices over the region of emission. These signals are clearer than seen in the one at a time perturbations in Figure 4.2, where short-term fluctuations appear to interfere with the initial response. Ignoring the strong localised effect, the patterns of sensitivity to SO_2 from Europe and East Asia, (Figure 4.14c and d) are remarkably similar. This is reminiscent of results from Kasoar et al. (2018), where removal of SO_2 from different Northern Hemisphere regions gave similar response patterns. As shown in Figure 1.5, this study found enhanced warming over North America and the Northern Atlantic and also over East Asia and the Northern Pacific. Those regions also show enhanced sensitivity in response to European and East Asian SO_2 perturbations. While sensitivity to SO_2 from North America does not follow the same pattern, it should be noted that this perturbation is relatively weaker than the others (Table 4.2), and because of this, the emulator does not attribute a strong response to this pollutant (e.g. Figure 4.10), leading to reduced sensitivity.

Additionally, the tropical perturbations (SO_2 from South America, Africa and South Asia and OC/BC from the Tropics) give rise to similar sensitivity maps. A common feature of these maps is the enhanced sensitivity over the tropical Atlantic and Pacific oceans. This mirrors the enhanced sensitivity over the Northern hemispheric oceans due to the Northern hemispheric perturbations. Kasoar et al. (2018) found that Northern hemispheric pollutant perturbations are projected onto the natural modes of variability of the GCM, based on an Empirical Orthogonal Function (EOF) decomposition. This is a principal component analysis (Section 2.2.1 carried out across the time axis of a long control run, to obtain a map of principal components, which are usually referred to as EOFs. The leading order EOFs represent the areas of maximum variability in the control run. Kasoar et al. (2018) found that the second EOF showed a high degree of similarity to the warming caused by removal of SO_2. Carrying out the same analysis,



(a) CO_2 Global

(b) CH_4 Global

(c) SO_2 Europe

(d) SO_2 North America

(e) SO_2 East Asia

(f) SO_2 South Asia

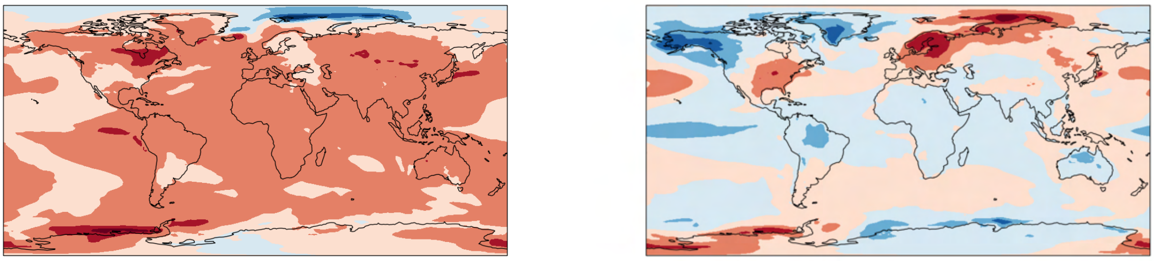
(g) SO_2 South America

(h) SO_2 Africa

(i) OC/BC Tropics

Figure 4.14: First order sensitivity indices (SI(1)) for each input to emulator.

Figure 4.15 shows the first and second EOFs for the model used in this study over the 45 year control run. The second EOF contains elevated features in the Northern Pacific, corresponding to regions particularly sensitive to the Northern hemispheric pollutants. Also, the first EOF shows some similar patterns in the Southern Pacific as those seen in the sensitivity map for SO_2 Africa (Figure 4.14h), suggesting that the tropical perturbations may project onto this mode. This is in line with other studies that suggest the climate response is often projected onto existing natural modes (Kasoar et al., 2018; Shindell et al., 1999; Ring and Plumb, 2008; Corti et al., 1999; Palmer, 1999).



(a) EOF1, 11.8% variance explained

(b) EOF2, 10.2% variance explained

Figure 4.15: First and second EOFs calculated as the modes of variability across the 45 year control run.

4.8 Emulator Application

Th emulator developed in this chapter has proven to be fairly successful at predicting responses to a range of GHG and aerosol perturbations. However, at large aerosol perturbations, it does not capture known effects which would limit additional temperature change, leading to an over-estimate in the response to aerosols under the high forcing regime (Figure 4.6d). In this section, I will demonstrate an example of the emulator use, avoiding this regime of the parameter space.

We will follow two vastly different scenarios from the Shared Socioeconomic Pathways (SSPs) which project socio-economic changes and their associated emissions into the future (Riahi et al., 2017). These are:

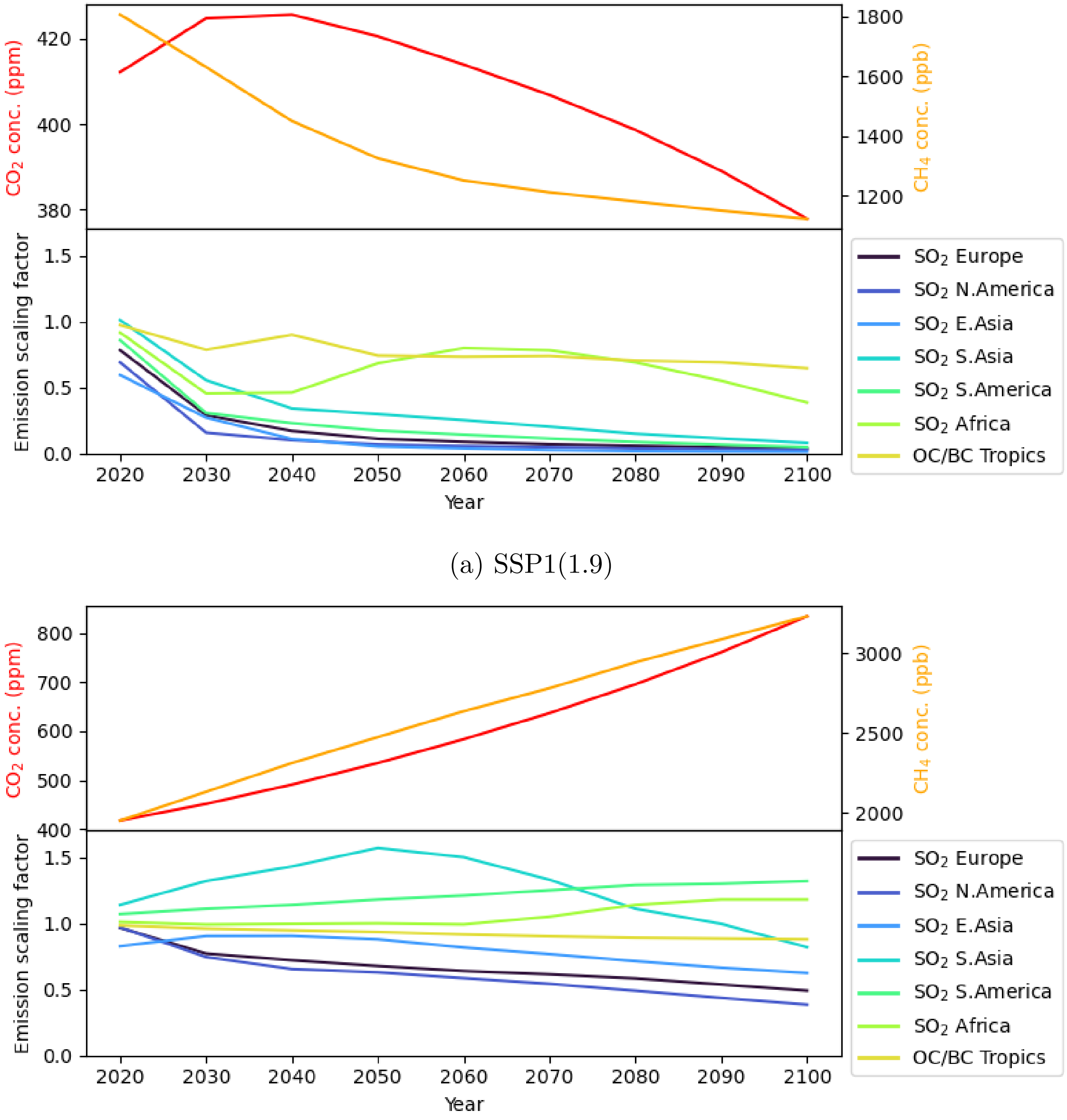
- **SSP1: Taking the Green Road (Low challenges to mitigation and adaptation).**

The world shifts towards a more sustainable path with reductions in consumption and energy use. Environmental goals are respected globally and inequality across and within countries is also reduced.

- **SSP3: Regional Rivalry – A Rocky Road (High challenges to mitigation and adaptation).** Nations focus on domestic or regional development and inequalities worsen with time. Consumption is material-intensive and there is little focus on environmental concerns.

From these pathways, I will use baseline scenarios of projected emissions, which give a radiative forcing of 1.9 W/m^2 for SSP1 (SSP1(1.9))and 7.0 W/m^2 for SSP3 (SSP3(7.0)) by 2100 (van Vuuren et al., 2017). The projected emissions from 2020-2100 in terms of the emulator inputs (GHG concentrations and aerosol scaling factors) are shown in Figure 4.16 for both these scenarios. The two scenarios differ greatly, SSP1(1.9) showing strong reductions in both GHG concentration and the aerosol perturbations, resulting in very small levels of SO_2 emissions by 2100 in all regions. In contrast, SSP3(7.0) projects a steady increase in GHG concentrations and a divergence in SO_2 emissions across the differing regions i.e. the regional rivalry.

To apply the emulator to make prediction under SSP scenarios, we must make several approximations. Firstly, the emulator only covers the main pollutant perturbations, CO_2, CH_4, regional SO_2 and tropical biomass burning. Gidden et al. (2019) show that in terms of radiative forcing, these contribute to $\sim$ 75% of the total forcing in SSP1(1.9) and $\sim$80% of the total forcing in SSP3(7.0). The remaining forcing comes from other sources such as tropospheric ozone, N_2O and Montreal Protocol gases, most of which are positive forcings, so may lead to a slight additional warming effect. Secondly, the regional perturbations make an approximation of a fixed distribution of emissions, although the SSP projections allow the entire emissions field to vary. Finally, we use the emulator to predict the initial short-term response to an abrupt perturbation from present-day conditions, rather than the transient response outlined by the SSPs. For the short-lived pollutants, a change in scaling factor would be realised rapidly due to their short lifetime. However, a sudden change in GHG concentration from present-day levels is a different style of perturbation to the slowly changing transient response given in Figure 4.16, where climate feedbacks and adjustments would have time to be realised in a transient GCM simulation.

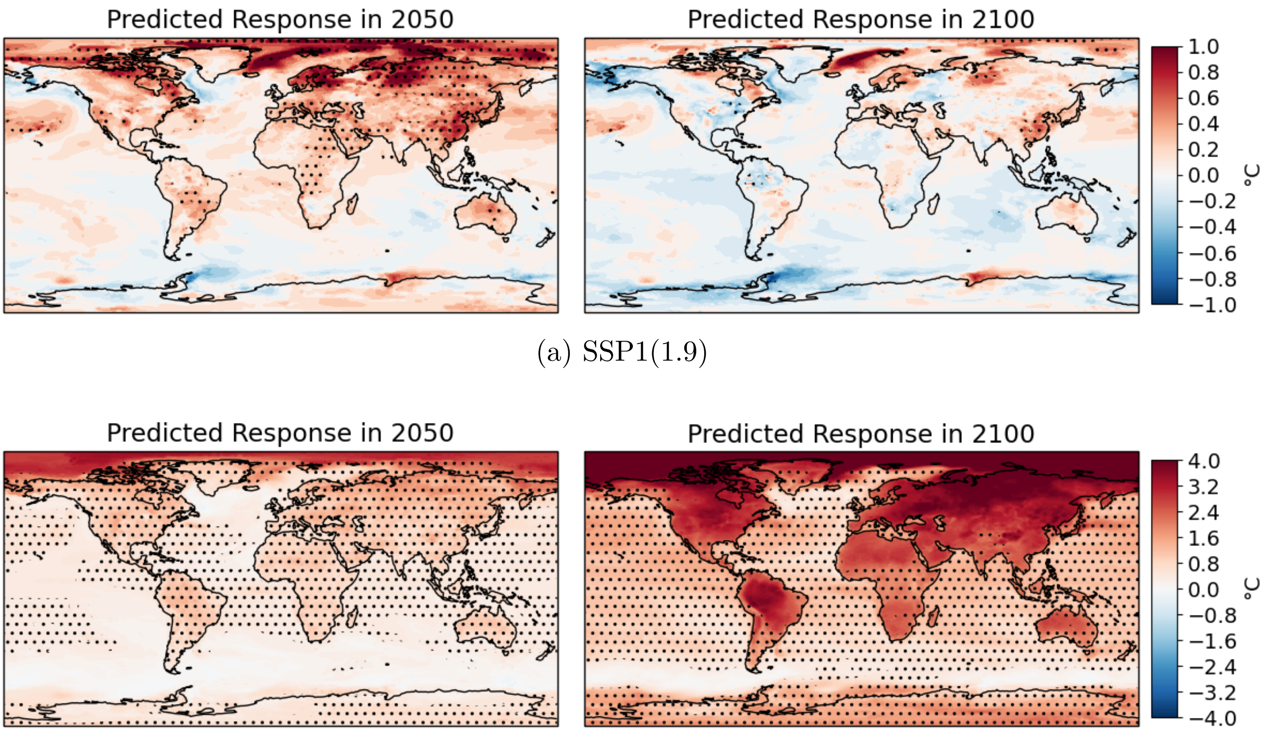


(a) SSP1(1.9)

(b) SSP3(7.0)

Figure 4.16: Projected conditions over time for two Shared Socio-economic Pathways (SCPs) where top panel shows GHG concentrations (CO_2 axis on left, CH_4 axis on right) and bottom panel shows aerosol scaling factors. Scenarios SSP1(1.9) and SSP3(7.0) are described in the text.

Regardless, the emulator can provide a rapid estimate of the short-term climate response to an abrupt jump in emissions and concentrations, without the need to run an expensive, transient GCM simulation. It can therefore be viewed as a ‘fast adjustment’ emulator. The emulator is run with a timeslice of the conditions at year 2050 and 2100 and the response maps are

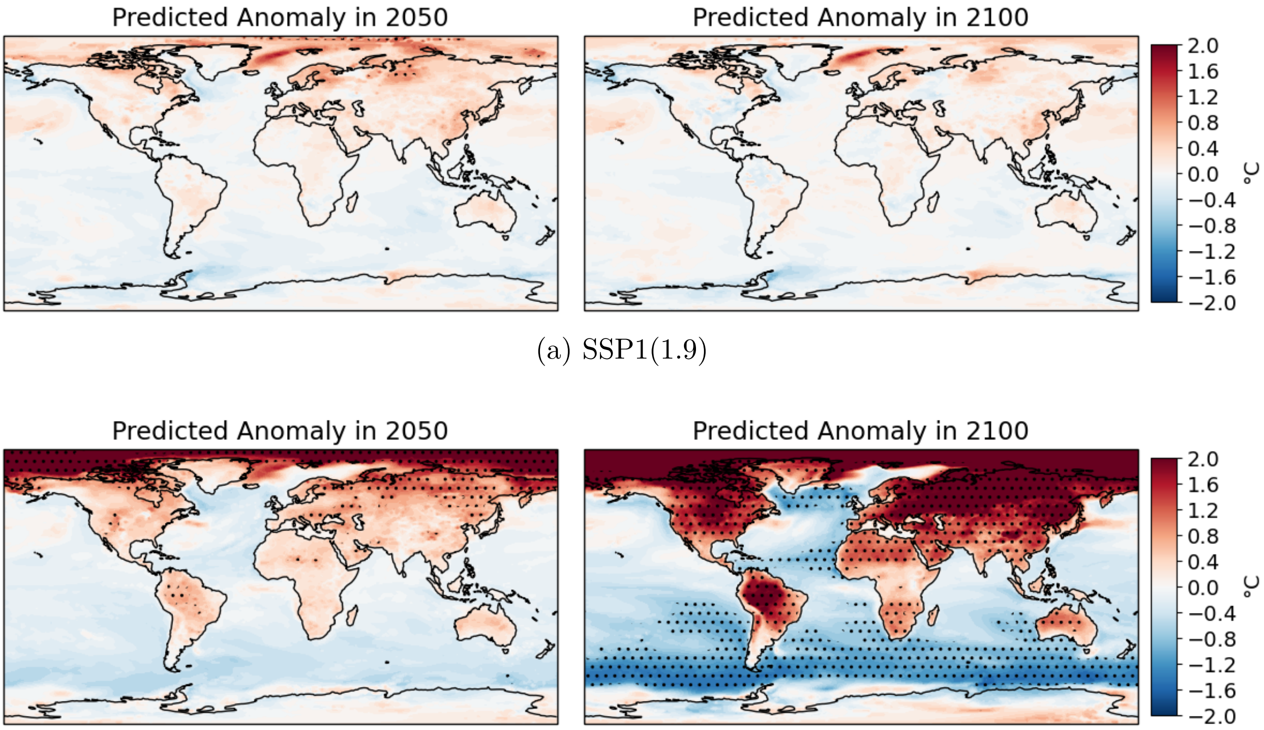

Predicted Response in 2050
Predicted Response in 2100
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.8
−1.0
°C
Predicted Response in 2050
Predicted Response in 2100
4.0
3.2
2.4
1.6
0.8
0.0
−0.8
−1.6
−2.4
−3.2
−4.0
°C

(a) SSP1(1.9)

(b) SSP3(7.0)

Figure 4.17: Emulator predicted short-term response to SSP scenarios in 2050 and 2100.

plotted in Figure 4.17 for both scenarios outlined. The stippling indicates where the response exceeds the 1σ level predicted by the emulator (i.e. the emulator is confident of the sign of response to at least 1σ). Figure 4.17a demonstrates that for SSP1(1.9) there is an expected initial rise in surface temperature, predominantly in the Northern Hemispheric, due to both an increase in GHGs and a decrease in SO_2. This warming is relatively stronger in regions of large SO_2 reductions, particularly East Asia. Following this, by 2100 the decrease in GHGs give rise to a net global cooling, although this does not exceed 1σ of the emulator uncertainty. In the SSP3(7.0) scenario, the dramatic increase in GHGs appears to dominate the prediction, with large increases in surface temperature. In this scenario, the difference in response between regions is large, highlighted by the anomalies with respect to the global mean response, in Figure 4.18. The stippling here indicates where the anomaly exceeds the global mean response to a significance of at least 1σ. While the anomalies in SSP1(1.9) are fairly weak across the globe, the anomalies in SSP3(7.0) show large differences between the different regions, caused by the 'regional rivalry' in this scenario. The emulator prediction under the 2100 conditions shows significantly more warming over land, particularly over the Northern Hemisphere. The temperature response is relatively weaker over tropical regions such as Africa, South Asia and the Southern tip of South America, these being the regions of roughly no change or increases in SO_2 emissions.


Predicted Anomaly in 2050
Predicted Anomaly in 2100
2.0
1.6
1.2
0.8
0.4
0.0
−0.4
−0.8
−1.2
−1.6
−2.0
°C
Predicted Anomaly in 2050
Predicted Anomaly in 2100
2.0
1.6
1.2
0.8
0.4
0.0
−0.4
−0.8
−1.2
−1.6
−2.0
°C

(a) SSP1(1.9)

(b) SSP3(7.0)

Figure 4.18: Emulator predicted anomalies relative to the global mean short-term response for SSP scenarios in 2050 and 2100.

These projections highlight how the emulator can be used to predict a complete map of how a GCM would respond in the first five years to a specific scenario, at a fraction of the cost of the complete GCM. Running the GCM for 5 year timescales takes on the order of days to estimate the climate response, whereas the emulator prediction takes on the order of seconds. This type of tool could benefit policy studies in widening the number of scenarios that could be explored in a limited period of time.

4.9 Conclusions

This chapter presents the first climate emulator with the ability to predict the short-term temperature response to a range of perturbations that include both long-lived greenhouse gases and short-lived aerosol pollutants, including regional perturbations of these. Testing of the emulator on a range of test data shows accurate performance across most of the parameter space, the exception to this being a test scenario with strong aerosol perturbations across all regions in the emulator design. This appears to be a consequence of the emulator learning a linear relationship between aerosol perturbation and temperature response, although in reality one would expect the climate to become less sensitive under high aerosol perturbations (Carslaw et al., 2013). Given this caveat, the emulator is still a useful tool for predictions of scenarios with smaller aerosol perturbations, which are more relevant to climate policy studies as large positive aerosol perturbations are typically deemed less likely in climate projection scenarios (Riahi et al., 2017).

The main source of uncertainty in the emulator predictions is the internal variability of the GCM. Since this is also a source of uncertainty in GCM runs on the same timescales, the choice of simulating climate response with an emulator is a sensible one. Naturally, one of the next steps would be to reduce this uncertainty. This could be done by enhancing the quality of the emulator training data by taking a mean over an ensemble of perturbation simulations in order to reduce internal noise in the data. Alternatively or additionally, the simulations could be averaged over for longer timescales to reduce the effect of year-to-year fluctuations. Although both options add computational cost, it would be fairly simple to extend the existing simulations to achieve either of these methods given the adequate computational resources.

One of the novelties of this emulator is that the inputs include both global GHGs and aerosol perturbations for different regions, allowing a wide range of different future projections to be made. The limitation here is that the aerosol perturbations are scaling factors over continental scales with fixed emission distributions. This means the emulator is limited to increasing or decreasing emissions by the same factor everywhere and cannot introduce new emission patterns. Future studies may be interested in broadening the range of aerosol perturbations to not only include a greater variety of aerosols but also a range of different emission distributions. Not only would this lead to an emulator with increased predictive power, but it could also provide an opportunity for a larger scale sensitivity analysis into the different plausible aerosol perturbations. Given the sensitivity analysis carried out here was based on very approximate feasible input ranges in Section 4.3.2, the sensitivity indices could be refined based on a more accurate range of possible future scenarios, informed by expert knowledge.

Ultimately, to make predictions of the long-term response based on emissions, the emulator presented here could be combined with the surrogate model in Chapter 3. However, doing this requires some consideration of error propagation, as the first emulator will introduce some errors that are then used as inputs to the second surrogate model. In particular, the emulator presented here predicted some regions at a lower accuracy, namely the high latitude continental regions i.e. Greenland, North America and Northern Europe. If coupling these emulators together, errors within these regions could become enhanced under the application of the second surrogate model. However, with further work, this multi-level emulator would be an achievable objective. The choice of a Gaussian process emulator benefits this task because it provides an uncertainty estimate with each prediction and therefore allows error propagation. This could highlight where regions have increased uncertainty and thereby assist the user when interpreting results. To properly test the performance of this multi-level emulator, additional long-term GCM simulations would be crucial. These could also be used to explore how residual errors in the short-term response emulator could be enhanced in the predicted long-term response, which may shine a light on how both these emulators could improved e.g. through additional, carefully selected training simulations.

Regardless, both surrogate models presented here and in Chapter 3 are useful as standalone tools. In particular, an application of the emulator developed here is demonstrated in Section 4.8, for two contrasting scenario projections. This exercise showed how the emulator can provide rapid estimates of the short-term response to perturbations even under a scenario with high inequality and regional rivalry. The scenarios explored here were taken directly from policy related studies, in which the CO_2, CH_4 and SO_2 forcings contribute to 75-80% of the total radiative forcing. Even policy studies interested in the complete range of forcings could benefit from an emulator in this format, by making rapid estimates based on such inputs. This could allow a wider range of scenarios to initially be explored, to help narrow down the choice of inputs to a GCM and thereby reduce the total computational burden. While the purpose of such an emulator is not to replace a GCM entirely, it can complement GCM studies by allowing rapid prediction for a wide range of scenarios and, furthermore, could be accessible to a wider range of users without the need for expensive computer hardware.

Chapter 5

A Bayesian Approach for Dimension Reduction

5.1 Introduction

In previous chapters, one of the noticeable characteristics of the data was the large number of features. This is unavoidable when working with spatio-temporal data, such as the output from climate models, making it common to employ dimension reduction techniques, before carrying out analysis. Previously, in Chapter 3 we made use of principal component analysis (PCA) to reduce the number of dimensions. This is a commonly used tool in prediction studies, particularly for Gaussian process emulation (Salter and Williamson, 2016; Ryan et al., 2018; Williamson et al., 2015; Higdon et al., 2008; Lawrence, 2005). When applied to the quantity that is being predicted, not only does PCA provide a lower dimensional variable to predict, but it also ensures correlations between the output, that are not guaranteed otherwise. However, working with a reduced space, rather than the complete dataset, introduces additional uncertainties that are not accounted for in these studies. In particular, when applying any dimension reduction method, a choice is made on the number of dimensions to keep, which can influence the accuracy of the prediction. It would therefore be useful and, if following a fully Bayesian approach, necessary to quantify the uncertainties associated with a dimension reduction method used, prior to applying a statistical prediction. Bayesian methods allow this to be done by estimating the probability distribution over the reduced space, including the distribution over the size of the reduced space. Then, not only can the optimal size of the reduced space be determined, but the uncertainty associated with it can be quantified and carried through to any subsequent analysis. For instance, an uncertainty on the reduced space can be used to estimate the total uncertainty associated with a prediction via emulation.

In this chapter, we will focus on **factor analysis**, a model for describing a large dataset in terms of underlying common factors, outlined briefly in Section 2.2.2. Factor analysis, like PCA, reduces a dataset to a lower dimensional space, but while PCA treats the variances across all variables equally and aims to maximise the variance in the principal components, factor analysis assumes there are two contributions to this variance. It assumes that the variance in each observed variable can be partitioned into a **common variance** that is equal for all variables (like PCA) and a **unique variance** (i.e. a measurement error, independent for each variables). This leads to a more general model for dimension reduction that allows independent measurement errors on each variable. This model reduces to probabilistic PCA in the case that these measurement errors are equal across all variables.

This chapter will start with an outline of factor analysis and the Bayesian approach to dealing with it for a fixed number of factors (Press and Shigemasu, 1989). However, when the number of underlying factors is unknown, which is typical for high dimensional climate data, Bayesian inference can be carried out on the number of factors as well as the factors themselves. Lopes and West (2004) does this through a reversible jump Markov chain Monte Carlo algorithm, but this particular method becomes computationally infeasible in high dimensions. To address this, I will develop and compare multiple new algorithms in this chapter.

5.2 Factor Analysis

We have a multivariate dataset $\mathbf{y}$ of m dimensions with a total of N separate observations or data points. Factor analysis assumes that there is an underlying structure of lower dimension,

$k < m$, that can be used to describe the dataset (Lopes and West, 2004). First, the data is assumed to be a sample from a zero-mean multivariate normal distribution with covariance matrix $\mathbf{\Omega}$) (size $m \times m$)

$$\mathbf{y} \sim \mathcal{N}_m(\mathbf{0}, \mathbf{\Omega})\,. \tag{5.1}$$

We decompose each data point y_n for $n \in 1, \cdots N$ into a vector of latent factors, η_n of length k and a factor loading matrix, Λ of size $m \times k$, with some residual noise vector ϵ_n of length m, i.e.

$$y_n = \mathbf{\Lambda}\eta_n + \epsilon_n\,. \tag{5.2}$$

The residual noise is a measurement error, unique to each variable dimension $i = 1, \cdots, m$. This equation can be extended for all N observations, by writing each observation as a row in the matrix $\mathbf{Y} = (\mathbf{y}_1, \cdots, \mathbf{y}_N)$, of size $N \times m$. Similarly, we write the k latent factors as a row in the matrix of latent factors $\boldsymbol{\eta} = (\eta_1, \cdots, \eta_k)$, a matrix of size $N \times k$, and each of the m residual noise vectors as a row in a residual noise matrix $\boldsymbol{\epsilon} = (\epsilon_1, \cdots, \epsilon_N)$. In matrix form, Equation (5.2) becomes

$$\mathbf{Y} = \boldsymbol{\eta}\mathbf{\Lambda}^T + \boldsymbol{\epsilon}\,. \tag{5.3}$$

The result of this is that we can describe the data, which has original dimension m, with the latent factor vector which has smaller dimension k. This latent factor vector $\boldsymbol{\eta}$ can then be used instead of the full data in subsequent data analysis, such as regression problems. Then, the matrix $\mathbf{\Lambda}$ and noise vector $\boldsymbol{\epsilon}$ can be used to recover any results in the original size of the data $(N \times m)$.

Firstly, note that this model is non-identifiable, because even with infinite observations, we cannot learn a unique solution for $\mathbf{\Lambda}$ and $\boldsymbol{\eta}$. This is because we can always find an orthogonal transformation that can be applied to $\mathbf{\Lambda}$ and $\boldsymbol{\eta}$ that leaves $\boldsymbol{\eta}\mathbf{\Lambda}$ invariant. For orthogonal matrix $\boldsymbol{P}$ of size $k \times k$, we can let $\mathbf{\Lambda}^* = \mathbf{\Lambda}\boldsymbol{P}^T$ and $\boldsymbol{\eta}^* = \boldsymbol{P}\boldsymbol{\eta}$, which gives $\boldsymbol{\eta}^*\mathbf{\Lambda}^{*T} = \boldsymbol{P}\boldsymbol{\eta}(\mathbf{\Lambda}\boldsymbol{P}^T)^T = \boldsymbol{\eta}\mathbf{P}^T\mathbf{P}\mathbf{\Lambda}^T = \boldsymbol{\eta}\mathbf{\Lambda}^T$, meaning that the choices $(\boldsymbol{\eta}, \mathbf{\Lambda})$ and $(\boldsymbol{\eta}^*, \mathbf{\Lambda}^*)$ are observationally equivalent (Aguilar and West, 2000; Geweke and Zhou, 1996). Following Lopes and West (2004); Geweke and Zhou (1996) and Aguilar and West (2000), we will ensure identifiability with the constraints on $\mathbf{\Lambda}$ so that it is a block lower triangular matrix of full-rank, with positive values on the diagonals. This lower triangular structure does not limit the form of the underlying space. This is because the order of variables do not matter and therefore any rotation matrix $\mathbf{A}$ can be applied to re-order the data, $\mathbf{Ay}$. This gives a factor analysis model with the same latent factors but a rotated loading matrix, $\mathbf{A\Lambda}$ without the lower triangular structure. However, an orthonormal matrix $\mathbf{P}$ can always be found so that the resulting factor loading matrix has a lower triangular structure. This makes the problem identifiable. The order of variables $\mathbf{y}$ has no effect on the resulting model when k is known, however, when determining k a consequence of the lower triangular structure on $\mathbf{A\Lambda}$ is that the latent variables must all appear within the first k measured variables in $\mathbf{y}$ (Lopes and West, 2004). This can be checked by re-arranging the vector $\mathbf{y}$ and repeating the analysis when estimating k, which is done in all analysis here.

Following Lopes and West (2004), we can let the latent factor vector $\boldsymbol{\eta}_n = (\eta_1, \eta_2, \cdots, \eta_k)' \sim \mathcal{N}_k(0, \mathbf{I}_k)$ for $n = 1, \cdots, N$, and then carry out analysis to determine the matrix $\mathbf{\Lambda}$. The residual errors explain the unique variance for each observed variable and it is typical to assume that this has the form $\boldsymbol{\epsilon}_n \sim \mathcal{N}_m(0, \mathbf{\Sigma})$ with diagonal covariance matrix $\Sigma = \text{diag}(\sigma_1^2, \cdots, \sigma_m^2)$ (Lopes and West, 2004). This allows us to write the covariance matrix $\mathbf{\Omega}$ defined in Equation (5.1), as

$$\mathbf{\Omega} = \mathbf{\Lambda}\mathbf{\Lambda}^T + \mathbf{\Sigma}\,. \tag{5.4}$$

The likelihood of the data $\mathbf{y}$, marginalised over latent factors, is

$$\begin{aligned} p(\mathbf{y}|\mathbf{\Lambda}, \mathbf{\Sigma}) &= \mathcal{N}(0, \mathbf{\Omega}) \\ &\propto |\mathbf{\Omega}|^{\frac{-N}{2}} \exp\left(\text{trace}\left(\frac{-1}{2}\mathbf{\Omega}^{-1}\mathbf{y}\mathbf{y}^T\right)\right) \end{aligned} \tag{5.5}$$

The goal is to estimate the underlying parameters with $\mathbf{\Lambda}$ and $\mathbf{\Sigma} = \text{diag}(\sigma_1^2, \cdots, \sigma_m^2)$. In the following section, I will outline a Bayesian method to do this.

5.2.1 Gibbs sampler when k is known

The use of Markov chain Monte Carlo (MCMC, Section 2.4) to estimate parameters $\mathbf{\Lambda}$ and $\mathbf{\Sigma}$ has been around for several decades, first formulated by Press and Shigemasu (1989). This study assumes that the number of factors, k, is known in advance. A Gibbs sampler (Section 2.4.3) can be used to estimate $\mathbf{\Lambda}$ and $\mathbf{\Sigma}$ as we can derive the complete posterior conditional distributions, given the following priors. This method also forms the basis of the algorithms developed by Lopes and West (2004) and Dunson (2006), which will be described subsequently.

Priors

We define the following prior on components of the factor loading matrix $\mathbf{\Lambda}$, labelled λ_{ij} for row i and column j:

$$\lambda_{ij} \sim \begin{cases} \mathcal{N}(\mu_0, C_0) & i \neq j \\ \mathcal{N}(\mu_0, C_0)\mathbf{1}_{\lambda_{ii}>0} & i = j \end{cases} \tag{5.6}$$

with mean and covariance parameters chosen by the user, which are set to $\mu_0 = 0$ and $C_0 = 1$ throughout, following Lopes and West (2004). Note that positive values are satisfied on the diagonals. The prior on σ_i^2 components are:

$$\sigma_i^2 \sim \mathcal{IG}(\nu/2, \nu s^2/2) \tag{5.7}$$

where $\mathcal{IG}$ is an inverse gamma distribution, $\nu = 1$ is the prior degrees of freedom and $s^2 = 0.2$ is the prior mode of the inverse gamma distribution.

Likelihood

Equation (5.3) breaks down the data $\mathbf{y}$ into the mean value $\boldsymbol{\eta}\mathbf{\Lambda}^T$ and some additional normally distributed noise which has variance $\mathbf{\Sigma}$. We can therefore write likelihood of the data,

$f(\mathbf{y}|\boldsymbol{\Lambda}, \boldsymbol{\Sigma}, \boldsymbol{\eta})$, as

$$\begin{aligned} f(\mathbf{y}|\boldsymbol{\Lambda}, \boldsymbol{\Sigma}, \boldsymbol{\eta}) &= \mathcal{N}(\mathbf{y}; \boldsymbol{\Lambda}\boldsymbol{\eta}, \boldsymbol{\Sigma}) \\ &\propto |\boldsymbol{\Sigma}|^{-N/2} \exp\left(-\frac{1}{2}(\boldsymbol{y} - \boldsymbol{\eta}\boldsymbol{\Lambda}^T)^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{y} - \boldsymbol{\eta}\boldsymbol{\Lambda}^T)\right) \\ &\propto |\boldsymbol{\Sigma}|^{-N/2} \exp\left(-\frac{1}{2}\boldsymbol{\epsilon}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\epsilon}\right) \end{aligned} \tag{5.8}$$

where the last equality uses $\boldsymbol{\epsilon} = \mathbf{y} - \boldsymbol{\eta}\boldsymbol{\Lambda}^T$.

Furthermore, we can marginalise out the latent factors $\boldsymbol{\eta}$, to find the marginal likelihood, $f(\mathbf{y}|\boldsymbol{\Lambda}, \boldsymbol{\Sigma})$. Recall that $\boldsymbol{\eta}$ is normally distributed with mean 0 and variance $\mathbf{1}$, meaning $\boldsymbol{\Lambda}\boldsymbol{\eta}$ is also normally distributed with mean 0 and and variance $\boldsymbol{\Lambda}\boldsymbol{\Lambda}^T$. This gives

$$\begin{aligned} f(\mathbf{y}|\boldsymbol{\Lambda}, \boldsymbol{\Sigma}) &= \int \mathcal{N}(\mathbf{y}; \boldsymbol{\Lambda}\boldsymbol{\eta}, \boldsymbol{\Sigma}) \mathcal{N}(\boldsymbol{\eta}; 0, 1) d\boldsymbol{\eta} \\ &\propto |\boldsymbol{\Omega}|^{-N/2} \exp\left(-\frac{1}{2}(\boldsymbol{y}^T \boldsymbol{\Omega}^{-T} \boldsymbol{y})\right) \\ &= \mathcal{N}(\mathbf{y}; 0, \boldsymbol{\Omega}) \end{aligned} \tag{5.9}$$

where this derivation involves completing the square and can be found in e.g. MacKay (1998).

Conditional Posteriors

Full conditional posterior distributions can be derived from the likelihood in Equation (5.8) with the choice of priors provided in Equations (5.6)-(5.7), allowing Gibbs sampling to be used (Section 2.4.3) (Press and Shigemasu, 1989; Lopes and West, 2004). The conditional distribution for $\boldsymbol{\eta}$ for each $n \in 1, \cdots, N$ is

$$\eta_n \sim \mathcal{N}\left(\left(\mathbf{I}_k + \boldsymbol{\Lambda}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\Lambda}\right)^{-1} \boldsymbol{\Lambda}^T \boldsymbol{\Sigma}^{-1} \mathbf{y}_n \,,\, \left(\mathbf{I}_k + \boldsymbol{\Lambda}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\Lambda}\right)^{-1}\right) \tag{5.10}$$

where $\mathbf{I}_k$ is an identity matrix of size $(k \times k)$.

For the non-zero components of $\mathbf{\Lambda}$, the conditional posterior of the values along row i, denoted λ_i, are

$$\lambda_i \sim \mathcal{N}(\mathbf{m}_i, \mathbf{C}_i)\mathbf{1}_{\lambda_{ii}>0} \tag{5.11}$$

where

$$\left.\begin{aligned}\mathbf{m}_i &= \mathbf{C}_i(C_0^{-1}\mu_0\mathbf{1}_i + \sigma_i^{-2}\boldsymbol{\eta}_i^T\mathbf{y}_i)\\ \mathbf{C}_i^{-1} &= \mathbf{C}_0^{-1}\mathbf{I}_i + \sigma_i^{-2}\boldsymbol{\eta}_i^T\boldsymbol{\eta}_i\end{aligned}\right\} i \leq k$$
$$\left.\begin{aligned}\mathbf{m}_i &= \mathbf{C}_i(C_0^{-1}\mu_0\mathbf{1}_k + \sigma_i^{-2}\boldsymbol{\eta}^T\mathbf{y}_i)\\ \mathbf{C}_i^{-1} &= \mathbf{C}_0^{-1}\mathbf{I}_k + \sigma_i^{-2}\boldsymbol{\eta}^T\boldsymbol{\eta}\end{aligned}\right\} i > k\ . \tag{5.12}$$

The subscript i notation on η_i indicates the η vector is truncated at the i-th value for the first k rows. This results in λ_i of length i, with the remaining columns set to zero to maintain the triangular structure. For the remaining rows where $i > k$ the full η vector is used and λ_i is of length m. Note that the diagonal components are fixed to be positive with $\mathbf{1}_{\lambda_{ii}>0}$.

The conditional posterior for σ_i^2 is

$$\sigma_i^2 \sim \mathcal{IG}\left((\nu + N)/2\,,\,\left(\nu s^2 + d_i\right)/2\right) \tag{5.13}$$

where $d_i = (\mathbf{y}_i - \mathbf{\Lambda}\boldsymbol{\eta}_i^T)^T(\mathbf{y}_i - \mathbf{\Lambda}\boldsymbol{\eta}_i^T)$.

Gibbs sampler

Following Lopes and West (2004) and Press and Shigemasu (1989), These conditional posterior distributions can be used to construct a Gibbs sampler, described in Section 2.4.3. This involves updating each model parameter incrementally. The algorithm below shows a single Gibbs step for where we first update η, followed by $\mathbf{\Lambda}$, and then $\mathbf{\Sigma}$.

Iterating over this forms the MCMC algorithm that can be used to infer the probability distributions on $\boldsymbol{\eta}$, $\mathbf{\Lambda}$, and $\mathbf{\Sigma}$. Importantly, this algorithm is an integral part of the methods

Algorithm 6 Gibbs sampling step for factor analysis (Lopes and West, 2004; Press and Shigemasu, 1989)

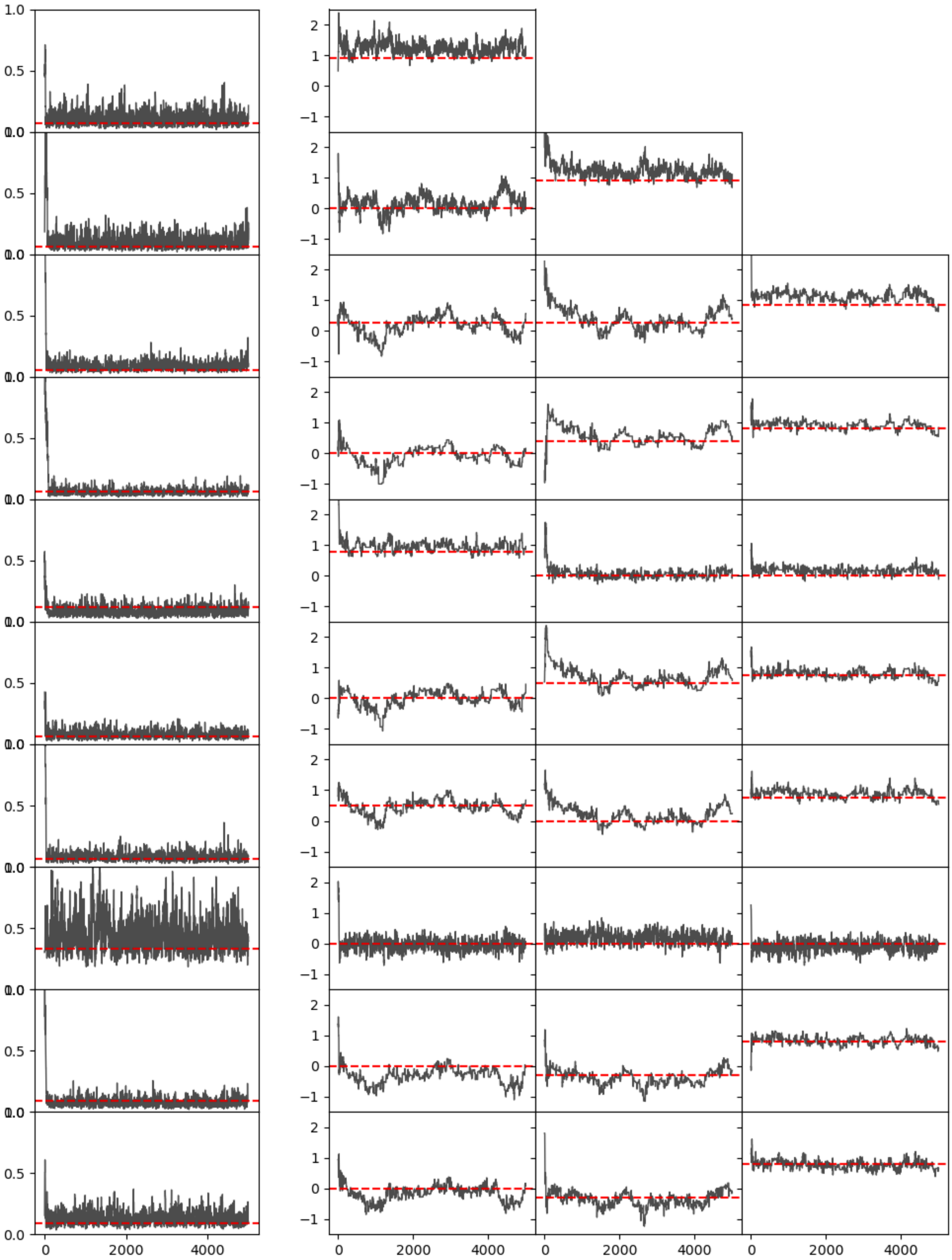
```
Start with $\theta = (\boldsymbol{\eta}, \boldsymbol{\Lambda}, \boldsymbol{\Sigma})$
  1. For $n \in 1, \cdots, N$, draw new $\eta_n$ with Equation (5.10)
  2. For $i \in 1, \cdots, m$, draw the components of $\lambda_i$ with Equations (5.11)-(5.12).
  3. For $i \in 1, \cdots, m$, draw $\sigma_i^2$ with Equation (5.13).
The new parameters are $\theta^* = (\boldsymbol{\eta}^*, \boldsymbol{\Lambda}^*, \boldsymbol{\Sigma}^*)$
```

also used to infer the probability distribution on k presented in Dunson (2006) and Lopes and West (2004) which are explained in the following section.

Example: 3 Factor Model

First, we apply the above MCMC algorithm to a test dataset that we will re-use throughout the following section to demonstrate the results of several algorithms. This test dataset is simulated from a known factor loading matrix and variance vector, where $k = 3$, $m = 10$. The dataset is the same as that presented in Dunson (2006) for $\boldsymbol{\Lambda}$ and with $\boldsymbol{\Sigma}$ divided by a factor of three to make the dataset more informative. This means most of the unique variances are small relative to the magnitude of $\boldsymbol{\Lambda}$, with the exception being the 8th component of $\boldsymbol{\Sigma}$ which is set to be significantly larger than other values, designed to test the methods under higher levels of noise.

These values are shown by the red lines in Figure 5.1 for all 10 components of $\boldsymbol{\Sigma}$ and the 27 non-zero components of $\boldsymbol{\Lambda}$. This plot also shows the trace of the first 5000 iterations of the Gibbs sampler in black (out of 10^4 iterations in total). The chains begin to converge towards the true values at around 2000 iterations, if not earlier, for all components. The diagonal components of $\boldsymbol{\Lambda}$ appear to be over-estimated slightly relative to the ground truth, as does the $\lambda_{8,2}$. This means that the common factor associated with these components appear stronger than they should in the simulated data. Similarly, the unique variances associated with these $(\sigma_1, \sigma_2, \sigma_3, \sigma_8)$ are also over-estimated slightly, as a consequence of this.



(a) $\mathbf{\Sigma}$

(b) $\mathbf{\Lambda}$

Figure 5.1: Trace plots for each non-zero component of $\mathbf{\Sigma}$ and $\mathbf{\Lambda}$ inferred with the Gibbs sampler. The upper triangular components of $\mathbf{\Lambda}$ are zero. Red line indicates true value.

5.2.2 Model selection when k is unknown

When the value of k is also unknown, the goal is to infer both k and $\theta = (\mathbf{\Lambda}, \mathbf{\Sigma})$. Since k determines the model, this is 'model selection'. A naive approach to this would be to explore several different candidate models by repeatedly running MCMC algorithms with different values of k and comparing the output of each one. Ideally, we would compare the marginal likelihood of each model by integrating out all prior values of θ, e.g. for $k = k_1$

$$f(\mathbf{y}|k_1) = \int f(\mathbf{y}|k_1, \theta)p(\theta)d\theta \,. \tag{5.14}$$

To compare two different models, with $k = k_1$ and $k = k_2$, the **Bayes Factor** can be calculated, as the ratio of the marginal likelihoods, i.e.

$$BF_{1:2} = \frac{f(\mathbf{y}|k_1)}{f(\mathbf{y}|k_2)} \,. \tag{5.15}$$

The marginal likelihood tells us how well model k_1 fits the data without assuming particular values of θ. This means models with many latent variables are penalised relative to those with fewer latent variables, since the marginalisation takes place over a greater space (i.e. the integral in Equation (5.14) is over the entire parameter space specified by the prior on θ, which is larger when θ has more dimensions). It is typical to see that increasing the number of latent variables initially increases the marginal likelihood, as the model better fits the data, but then decreases when the number of variables is increased beyond what is needed to describe the data (Everitt et al., 2020). In model selection, we are therefore interested in the model that provides the highest marginal likelihood, i.e. the model that best fits the data without too much additional complexity.

However, the marginal likelihood is an unknown quantity in standard MCMC and evaluating this integral is not a straightforward task, especially when θ is high dimensional. There are various approximation methods that can be used to estimate the marginal likelihood (Chib, 1995; Newton and Raftery, 1994; Gelfand and Dey, 1994; Tierney and Kadane, 1986). These are outlined and compared in Lopes and West (2004) where they are shown to have varying levels of accuracy in different simulation studies but are outperformed by an RJMCMC approach introduced in this paper (discussed in Section 5.2.4). Furthermore, these methods require independent MCMC simulations for each k, which can become expensive in high dimensional spaces, as there are many possible k values to explore. In the following section, we will outline an existing approach that approximately infers k with a single MCMC chain (Dunson, 2006).

5.2.3 Over-parameterised Approximation when k is unknown

Model selection as described above requires an independent MCMC analysis on each possible model. For many dimensions, this can become infeasible. Dunson (2006) uses an over-parameterisation approach to estimate the posterior probabilities for each model k. An MCMC analysis is carried out in model $k = k_{\max}$ using the Gibbs sampler described in Section 5.2.1. This gives the model parameters for model k which we label $\theta^{(k)} = (\boldsymbol{\Lambda}^{(k)}, \boldsymbol{\Sigma}^{(k)}, \eta^{(k)})$. These model parameters are transformed down to model $k' = k - 1$, with model parameters $\theta^{(k')} = (\boldsymbol{\Lambda}^{(k')}, \boldsymbol{\Sigma}^{(k')}, \eta^{(k')})$. Dunson (2006) proposes a transformation in which the last two columns of matrix $\boldsymbol{\Lambda}^{(k)}$ are collapsed to find the last column of matrix $\boldsymbol{\Lambda}^{(k')}$:

$$\lambda_{i,j}^{(k')} = \begin{cases} \lambda_{i,j}^{(k)} & j < k' \\ \operatorname{sign}(\lambda_{i,k-1}^{(k)}) \left({\lambda_{i,k}^{(k)}}^2 + {\lambda_{i,k-1}^{(k)}}^2 \right)^{1/2} & j = k' \end{cases} \tag{5.16}$$

and the variance parameters of $\boldsymbol{\Sigma}$ remain the same:

$$\sigma_i^{(k')} = \sigma_i^{(k)} . \tag{5.17}$$

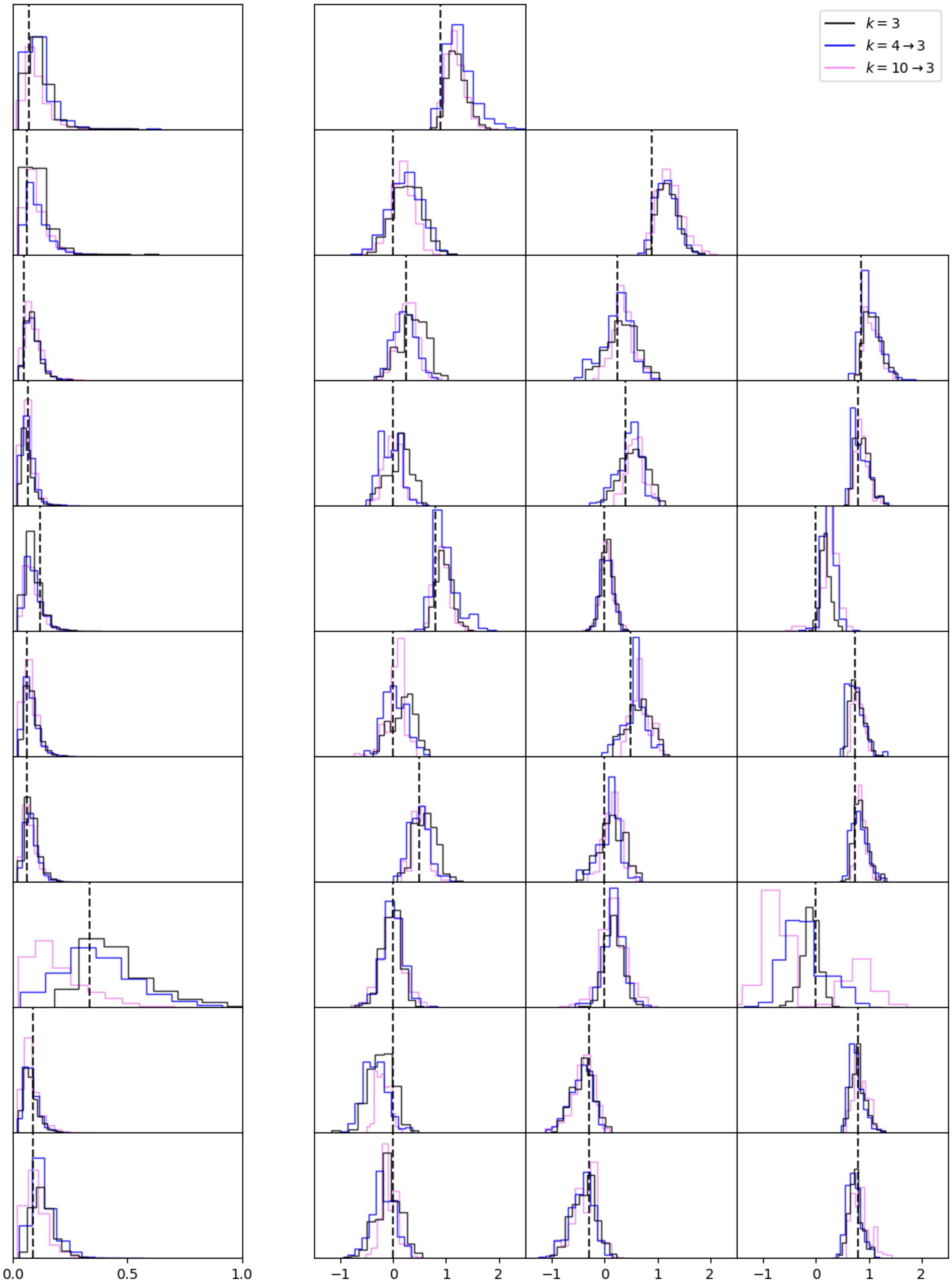
The latent factor vector η is marginalised out here but η can be sampled using Equation (5.10).

This provides us with an MCMC chain in the $k - 1$ model at much lower costs than running a Gibbs sampler. Furthermore, these model parameters can then be transformed down to obtain the parameters of model $k - 2$, $k - 3$, and so on. Dunson (2006) shows that when the model is over-parameterised with $k > k_{\text{true}}$, this transform provides a reasonable approximation to the true posterior and that the MCMC chain in this reduced model exhibits efficient mixing (in fact, more efficient than the original MCMC chain in the over-parameterised model (Ghosh and Dunson, 2009)). An example of this approach is presented in Figure 5.2, where the Gibbs sampler has been run for 10^4 iterations in three separate models, $k = 3$, $k = 4$ and $k = 10$. After removing 5000 iterations for burn-in, the parameters of the $k = 4$ and $k = 10$ models have been transformed down to obtain parameters of the $k = 3$ model. These parameters are plotted as histograms for each component of $\mathbf{\Sigma}$ and $\mathbf{\Lambda}$, where the black dashed line indicates the true model parameters.

The blue histogram showing the the parameter values obtained when run with $k = 4$ and transformed down to $k = 3$ make a very good approximation to the parameter distributions inferred by the $k = 3$ Gibbs sampler. The same is true for the violet histogram, produced by the Gibbs sampler run in the largest possible model $k = 10$ and then transformed down to $k = 3$. Although this approach relies on 6 transformations, most distributions remain similar to those obtained with the $k = 3$ Gibbs sampler. The exception of this is the 8th row, in which both σ_8 and $\lambda_{8,3}$ are poorly represented. Note that this is the row in which the large variance is introduced to test the methods under large residual noise levels. The transformation proposed in Dunson (2006) is a reasonable approximation for the correct model, but can result in inaccuracies, particularly for data with large residual noise and with high dimensions, where more transformations would be required.

The accuracy of the transform in an under-parameterised model, $k = 2$, is shown in Figure 5.3 where the $\mathbf{\Lambda}$ and $\mathbf{\Sigma}$ distributions obtained through the Gibbs sampler with $k = 2$ in black, compared against the Gibbs sampler run with $k = 3$, transformed down to the $k = 2$ model in blue. These distributions differ significantly for many of the transformed $\mathbf{\Lambda}$ and $\mathbf{\Sigma}$ variables. There is a good agreement between some of the $\mathbf{\Sigma}$ distributions but this is overshadowed by the fact that in some cases there are almost no overlap between the distributions (σ_1, σ_5). This transform therefore no longer provides a good approximation to an under-parameterised model, which will be relevant to the methods presented later that use this transformation as a starting point.

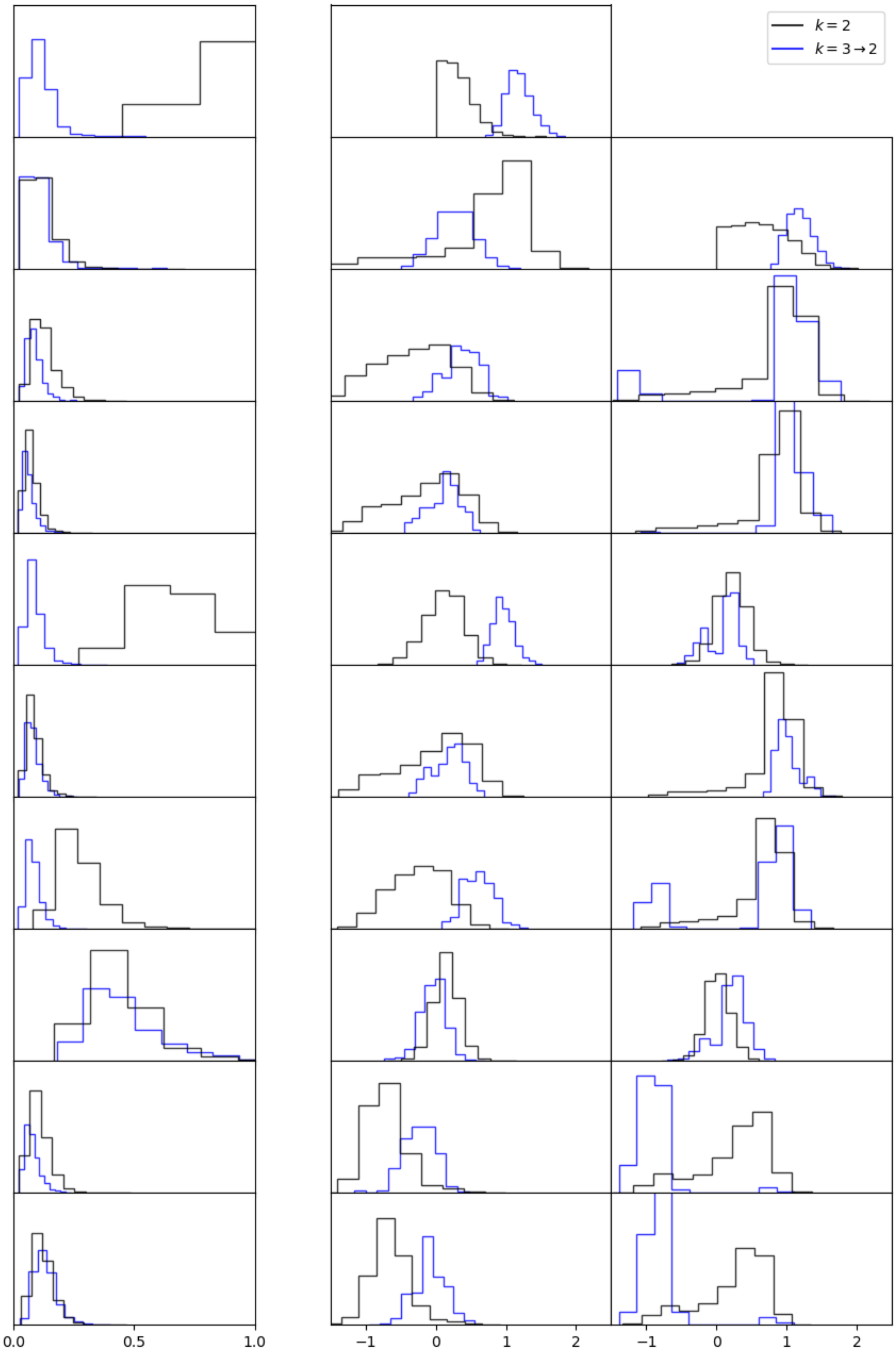
Dunson (2006) use these transformations to carry out model selection by comparing the Bayes



(a) $\boldsymbol{\Sigma}$

(b) $\boldsymbol{\Lambda}$

Figure 5.2: Histograms of the parameters of the $k = 3$ model obtained from a Gibbs sampler run for 5000 iterations (after 5000 iterations for burn-in) in the $k = 3$ model and in over-parameterised models, $k = 4$ and $k = 10$, transformed into the $k = 3$ model space following the method presented in Dunson (2006). The black dashed line shows the true values of which the data is simulated from.



(a) $\boldsymbol{\Sigma}$

(b) $\boldsymbol{\Lambda}$

Figure 5.3: Histograms of the parameters of the $k = 2$ model obtained from a Gibbs sampler run for 5000 iterations (after 5000 iterations for burn-in) in the $k = 2$ model and in the over-parameterised model, $k = 3$ transformed into the $k = 2$ model space following the method presented in Dunson (2006).

factors for the transformed models. This relies on estimating the marginal likelihood, Equation (5.14), which is done by averaging over all iterations obtained through Gibbs sampling in the largest model, $\theta^{(m)} = (\mathbf{\Lambda}^{(m)}, \mathbf{\Sigma})$, called 'model averaging'. The estimated Bayes factors are plotted in blue in Figure 5.4 as boxplots for 20 estimates made with different independent seeds. The same is done using the full Gibbs sampler for every model for comparison in black. We find that the $k = 3$ model is selected in most instances, but in some cases the $k = 4$ model is incorrectly selected as the best model, also found in Ghosh and Dunson (2009). The inaccuracy of the transform in the underparameterised regime, for $k = 1$ and $k = 2$ leads to underestimated marginal likelihoods compared to those obtained with the full Gibbs sampler. This is not problematic when the goal is simply to determine which model is best, but it does not give an accurate distribution over k. For a more accurate Bayesian assessment of this, we propose approaches that makes use of this transform, since it is known to perform reasonably well, but then allow adjustments to find the exact posterior distribution on model parameters. This should be a remedy for the small inaccuracies introduced at each step of the transform, which can build to an overall large inaccuracy as seen in the $k = 10$ transform down to $k = 3$ in Figure 5.2b. It should also improve upon the poor approximation seen in the under-parameterised models (Figure 5.3). We do this through a reversible jump MCMC approach in Section 5.3 and through Sequential Monte Carlo (SMC) in Section 5.4. These methods provide another benefit, as we can carry out Bayesian model comparison without relying on model averaging to estimate the Bayes' factor, as done in Dunson (2006) and in Figure 5.4.

5.2.4 RJMCMC when k is unknown: a psuedo-marginal algorithm

A reversible jump MCMC (RJMCMC) algorithm, described in Section 2.5, is a suitable candidate as a Bayesian method for determining number of underlying factors, as it can learn the true posterior distribution on k. The first use of RJMCMC to do this was proposed by Lopes and West (2004) and is still the only widely known RJMCMC approach in the factor analysis literature to date. We note that the algorithm in Lopes and West (2004) is a form of the pseudo-marginal target algorithm, outlined in Section 2.4.6, in which the marginal likelihood

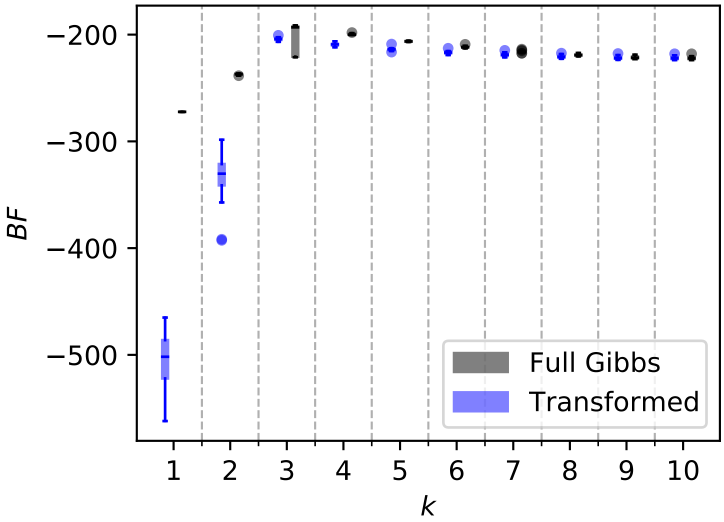


Figure 5.4: Bayes factors estimated using model averaging, presented in Dunson (2006) for simulated data with $k = 3$ underlying factors. Estimates are made across 20 independent seeded runs. In black are the estimates made with the full Gibbs sampler run separately for each k. In blue are the estimates made from the over-parameterised Gibbs sampler run with $k = 10$ transformed down to each model.

$f(y|k)$ is estimated in the acceptance ratio (Equation (2.56)). For the RJ version of this, we require estimating $f(y|k)$ with

$$\widehat{f(y|k)} = \frac{f(y|k, \theta^{(k)})\, p(\theta^{(k)}|k)}{q(\theta^{(k)}|k, \theta'^{(k')}, k')}\,. \tag{5.18}$$

where $q(\theta'^{(k')}|k, \theta^{(k)}, k)$ is a proposal distribution that we can sample $\theta'^{(k')}$ from at each iteration, for model k. This assumes that we have a proposal distribution on k, i.e. $q(k'|k)$, which will follow the standard RJMCMC approach of taking one step up or down in dimension with equal probability ($k' = k + 1$ or $k' = k - 1$). As before, the model parameters are $\theta^{(k)} = (\mathbf{\Lambda}^{(k)}, \mathbf{\Sigma}^{(k)})$. Specifically, Lopes and West (2004) build the proposal distribution based on preliminary MCMC runs when k is fixed, that are used to obtain posterior distributions on $\mathbf{\Lambda}^{(k)}$ and $\mathbf{\Sigma}^{(k)}$ for each k independently. These preliminary runs are used to compose a proposal distribution on the independent components of $\sigma_i^{(k)2}$ and on the matrix $\mathbf{\Lambda}^{(k)}$:

$$q(\sigma^{(k)2}|k) = \prod_{i=1}^{m} \mathcal{IG}(a, av_i^{(k)2}) \tag{5.19}$$

$$q(\mathbf{\Lambda}^{(k)}|k) = \mathcal{N}(\mathbf{b}^{(k)}, b\mathbf{B}^{(k)}) \tag{5.20}$$

where $v_i^{(k)2}$ is the mode of the $\sigma_i^{(k)2}$ MCMC chain for model k (chosen rather than the mean due to the large tails seen in the posterior distributions), $\mathbf{b}^{(k)}$ and $\mathbf{B}^{(k)}$ are the mean and variance matrices of the $\mathbf{\Lambda}^{(k)}$ chain for model k and a and b are positive scale parameters chosen by the user. Following Lopes and West (2004), we let $a = 18$ and $b = 2$.

In this form of RJMCMC, note that the new parameter values $(\mathbf{\Lambda}'^{(k')}, \mathbf{\Sigma}'^{(k')})$ depend only on k and not on the previous parameter values $(\mathbf{\Lambda}^{(k)}, \mathbf{\Sigma}^{(k)})$ i.e.

$$q(\theta^{(k)}|k, k', \theta^{(k')}) = q(\theta^{(k)}|k) = q(\mathbf{\Lambda}^{(k)}|k)\, q(\mathbf{\Sigma}^{(k)}|k) \tag{5.21}$$

The result is an acceptance rate of

$$r_{pm}(k \to k') = \frac{f(y|\theta'^{(k')}, k')\, p(\theta^{(k')}|k')\, p(k')}{f(y|\theta^{(k)}, k)\, p(\theta^{(k)}|k)\, p(k)}\, \frac{q(\theta^{(k)}|k)\, q(k|k')}{q(\theta'^{(k')}|k')\, q(k'|k)}\,. \tag{5.22}$$

As the proposals are independent of θ, the Jacobian term is reduced to $\frac{dk'}{dk} = \frac{d(k\pm1)}{dk} = 1$. Note here that $\frac{f(y|\theta^{(k)},k)\, p(\theta^{(k)},k)}{q(\theta^{(k)}|k)}$ is a single point importance sampling estimate of $\pi(\theta^{(k)})$ with importance weight $q(\theta^{(k)}|k)$ (Equation (2.56)). As this method uses a pseudo-marginal approach to estimate the acceptance ratio, we will call refer to this approach as pseudo-marginal RJMCMC (or pmRJ for short). Algorithm 7 describes the algorithm in full.

The main drawback of this algorithm is that it requires generating a preliminary MCMC for every possible value of $k = (1, \cdots, k_{\max})$, (Step 0) making this an expensive approach for highly multivariate data. For large $k_{\max}$ this becomes infeasible. In particular, data from climate models is often high resolution in time and/or space. Furthermore, for large k and m, these chains may also require many iterations to reach convergence. The total complexity of the algorithm scales with $O(m^3)$, due to the cost of running m independent MCMCs, each scaling

Algorithm 7 Reversible jump MCMC algorithm for factor analysis that uses pseudo-marginal target approximation (pmRJ), first presented in Lopes and West (2004).

```
0. Separately run all MCMC models for $k = 1, 2, \cdots, k_{max}$ . Then for each $k$, find the
   mean and variance of the chain on $\lambda$ to give $b^{(1)}, b^{(2)}, \cdots, b^{(k_{\max})}$ and $B^{(1)}, B^{(2)}, \cdots, B^{(k_{\max})}$
   respectively, and the mode of the chain on $\sigma_i^2$ to give $v_i^{(1)2}, v_i^{(2)2}, \cdots, v_i^{(k_{\max})2}$ . These are
   used in the proposal distributions in the reversible jump method in step 2b.
1. Set initial $k$ and draw prior $\theta^{(k)}$. Set $X_0 = (k, \theta^{(k)})$
2. For $n = 1, \cdots, N$: Current model is $X_n = (k, \theta^{(k)})$
   (a) Do 1 MCMC step (algorithm 6)
   (b) Do 1 RJ step:
       i. Propose new $k = k'$ from proposal $k' \sim q(\cdot|k)$.
       ii. Propose new values for $\theta'^{(k')} = (\mathbf{\Lambda}'^{(k')}, \mathbf{\Sigma}'^{(k')}) \sim q(\theta^{(k)}|k)$ from Equations (5.19)-
           (5.21).
       iii. Calculate acceptance probability $\alpha = \min(1, r_{pm})$ where $r_{pm}$ is given by Equation
            (5.22).
       iv. Accept proposed model with probability $\alpha$ and set $X_{n+1} = (k', \theta'^{(k')})$, otherwise
           reject and set $X_{n+1} = X_n = (k, \theta)$.
```

with $O(m^2)$ based on the required sampling from the truncated multivariate normal distribution, as well as the $O(m^3)$ cost of inverting the matrix $\mathbf{\Omega}$ in the likelihood (Equation (5.8)). In Section 5.3, I present alternative reversible jump algorithms that do not require these independent MCMCs.

Example: 3-Factor Model

This algorithm is implemented on the same simulated data described in Section 5.2.1 with $k = 3$ underlying factors. In MCMC methods, convergence is checked initially through trace plots of each parameter, as done in Figure 5.1. In RJMCMC, there are many parameters to check for, both k and $\theta = (\mathbf{\Lambda}^{(k)}, \mathbf{\Sigma}^{(k)})$. This is complicated by the fact that the nature of θ changes depending on which value k takes. Also, θ is often high-dimensional, such as in this example, where even with only $m = 10$, θ includes at least 20 dimensions and can extend up to 50 dimensions depending on the value of k. To assess convergence, it is simpler to check the trace plot of a scalar value that is a function of all θ and that maintains the same meaning regardless of k (Brooks and Giudici, 2000). A common example is to plot the log likelihood

(Equation (5.8)), alongside k, as shown in Figure 5.12 for the first 1000 iterations. Both k and the log likelihood appear to vary in synchronisation with each other, as the log likelihood jumps between different states along side jumps between different factor models.

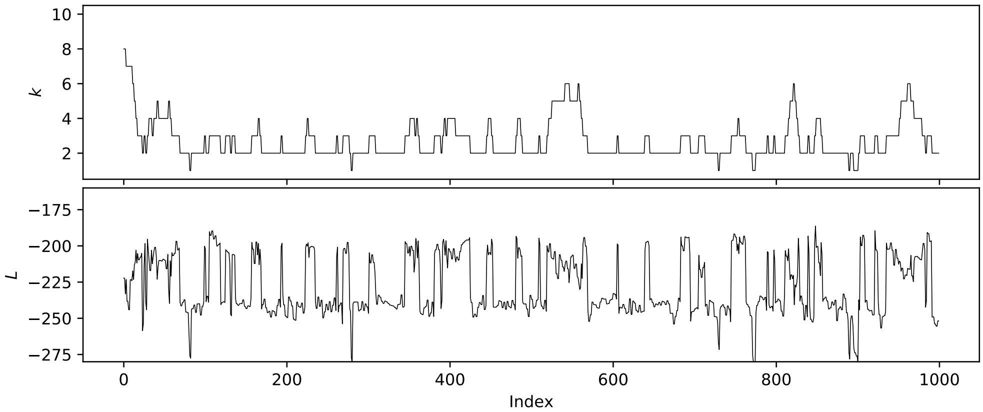


Figure 5.5: Trace plots of (top panel) k and (bottom panel) the log likelihood for the first 1000 iterations of the pmRJ algorithm on the simulated data with $k = 3$ underlying factors.

The trace plot of k shows that the algorithm spends the majority of its time in model $k = 2$ and $k = 3$, indicating that the method is quickly able to find roughly the correct number of factors. However, the algorithm is not perfect, as model $k = 2$ is slightly favoured over the true model $k = 3$. This is shown further in the posterior distribution of k plotted as $\log(p(k))$ in Figure 5.6, calculated as the relative number of iterations spent at each model k after burn-in. The error bars are calculated from repeating this with 20 parallel chains initialised with a different seed.

As this algorithm targets $\pi_k(\theta|y)$, the parameters learned in the model $k = 3$ are roughly consistent with the Gibbs sampler in $k = 3$ model. This is shown by the red histograms in Figure 5.7 where the Gibbs sampler with $k = 3$ fixed are shown for reference in black. The pmRJ algorithm finds the distribution of the parameters to be very similar, which is expected since the proposal distribution is built from the preliminary Gibbs sampler simulations. Some λ parameters have narrower distributions in the pmRJ algorithm. This is likely a consequence of the proposal distribution being independent of the value of θ at the previous iteration. Points at the edges of the distribution are proposed less frequently, and even if they are proposed, they are less likely to be accepted because they are compared against a previous iteration that is often in a high density region of the parameter space.

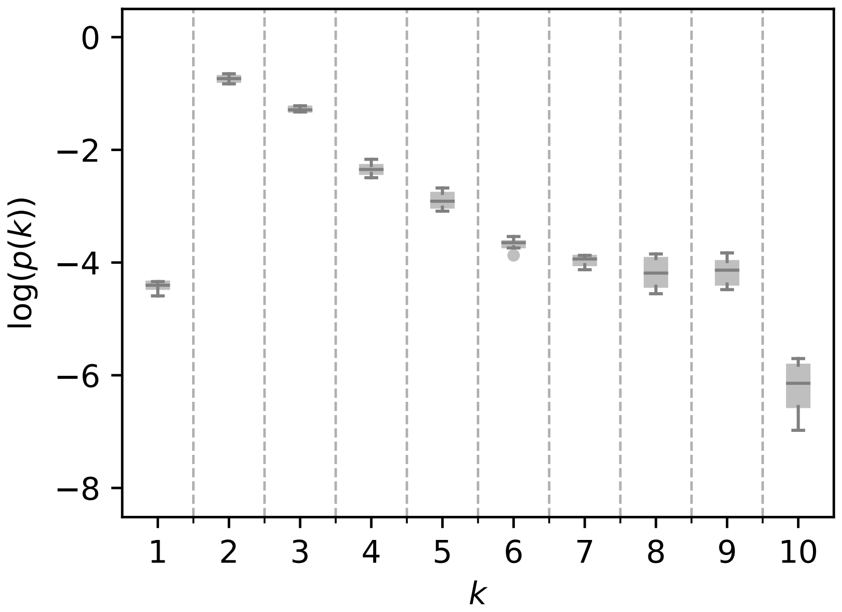
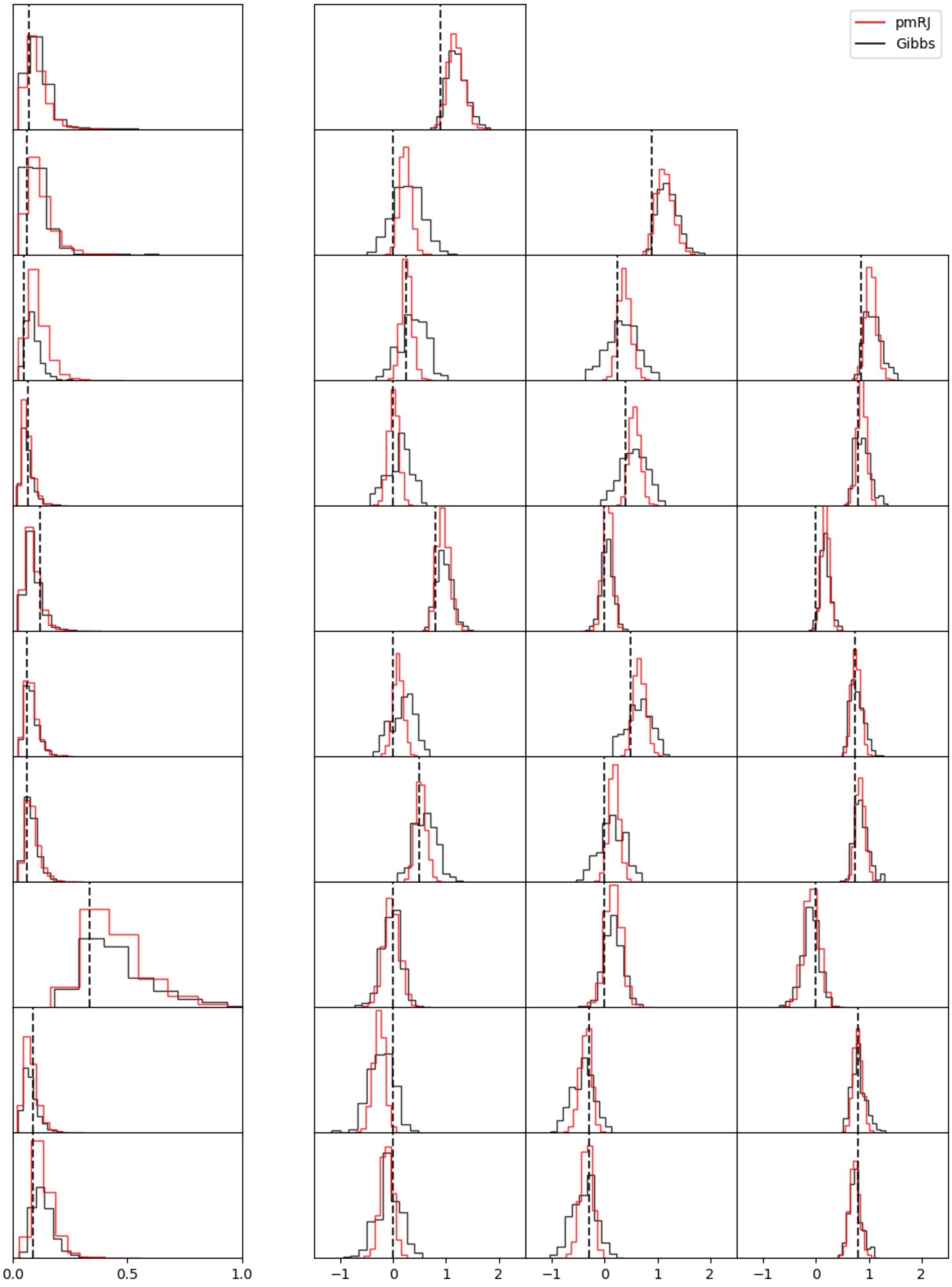


Figure 5.6: Log posterior probabilities of the pmRJ algorithm estimated from 20 independently seeded simulations on the simulated dataset with $k = 3$ underlying factors.

One of the potential problems with this algorithm is that it does not make use of parameter values in previous iterations, which is an unusual feature in MCMC algorithms. Instead, the use of proposals on $\theta^{(k')}$ that depend only on model k' and not on the previous value, $\theta^{(k)}$, can lead to a significant proportion of time and therefore computational effort being spent in undesired regions of the parameter space (e.g. $k = 5, 6, \cdots$ in Figure 5.6). In the next section, we introduce an RJMCMC algorithm that does not use independent proposals which also means it does not require independent preliminary MCMCs. We will build proposals that make use of current parameter values, $\theta^{(k)}$, based on the transformation presented by Dunson (2006).

5.3 A new Reversible Jump MCMC

Existing methods of Bayesian model selection for factor analysis are either inexact (Dunson, 2006) or expensive (Lopes and West, 2004). Here, I will present a novel RJMCMC algorithm to infer the posterior distributions on $(k, \theta^{(k)})$ that avoids the need for preliminary independent MCMCs.



(a) $\mathbf{\Sigma}$ (b) $\mathbf{\Lambda}$

Figure 5.7: Histograms of samples when $k = 3$ in the pmRJ algorithm (Lopes and West, 2004) (red) compared against the Gibbs sampler with fixed $k = 3$ (black). Both algorithms are run for 5000 iterations (after 5000 iterations for burn-in). The black dashed line shows the true values of which the data is simulated from.

5.3.1 Proposal Distributions

The RJMCMC algorithm requires a new proposal distribution, which will be built using the transformation outlined in Dunson (2006), discussed in Section 5.2.2. This transform was shown to propose sensible parameters for the new model when moving down in dimension, from model k to $k' = k - 1$, with Equation (5.16). We call this the *merge* transform as it merges the last two columns of matrix $\mathbf{\Lambda}^{(k)}$.

To move up in dimension from model k to $k' = k+1$, we must build an equivalent *split* transform that is consistent with this merge transformation. As the merge transform involves merging two columns together, there are multiple versions of $\mathbf{\Lambda}^{(k)}$ that can give the same $\mathbf{\Lambda}'^{(k')}$. To do this, we introduce some randomness that determines the new parameters, as done in Richardson and Green (1997). We define the split transform as

$$\lambda_{i,j}^{(k')} = \begin{cases} \lambda_{i,j}^{(k)} & j < k' - 1 \\ \operatorname{sign}(\lambda_{ik}^{(k)})\, u_i^{1/2}\, \lambda_{ik}^{(k)} & j = k' - 1 \\ \gamma_i \left(\lambda_{ik}^{(k)2}\,(1 - u_i)\right)^{1/2} & j = k' \end{cases} \tag{5.23}$$

where u_i and γ_i are additional auxiliary variables. The first k columns of $\mathbf{\Lambda}$ and $\mathbf{\Lambda}^{(k')}$ are equal. The last two columns are then split into two with auxiliary variable, $u_i \in (0, 1)$ determining the proportion. The sign of $\mathbf{\Lambda}^{(k')}$ in column $k' - 1$ is consistent with the sign of $\mathbf{\Lambda}^{(k)}$ and the sign of the column k' is determined at random with auxiliary variable $\gamma = \pm 1$, noting that this must be positive when $i = j$ so that the diagonal values are positive. These choices are made to ensure that the transform down via Equation (5.16) is satisfied. As done in the merge move Dunson (2006), the variance parameters remain the same.

We take

$$u_i \sim \text{Unif}(0, 1) \tag{5.24}$$

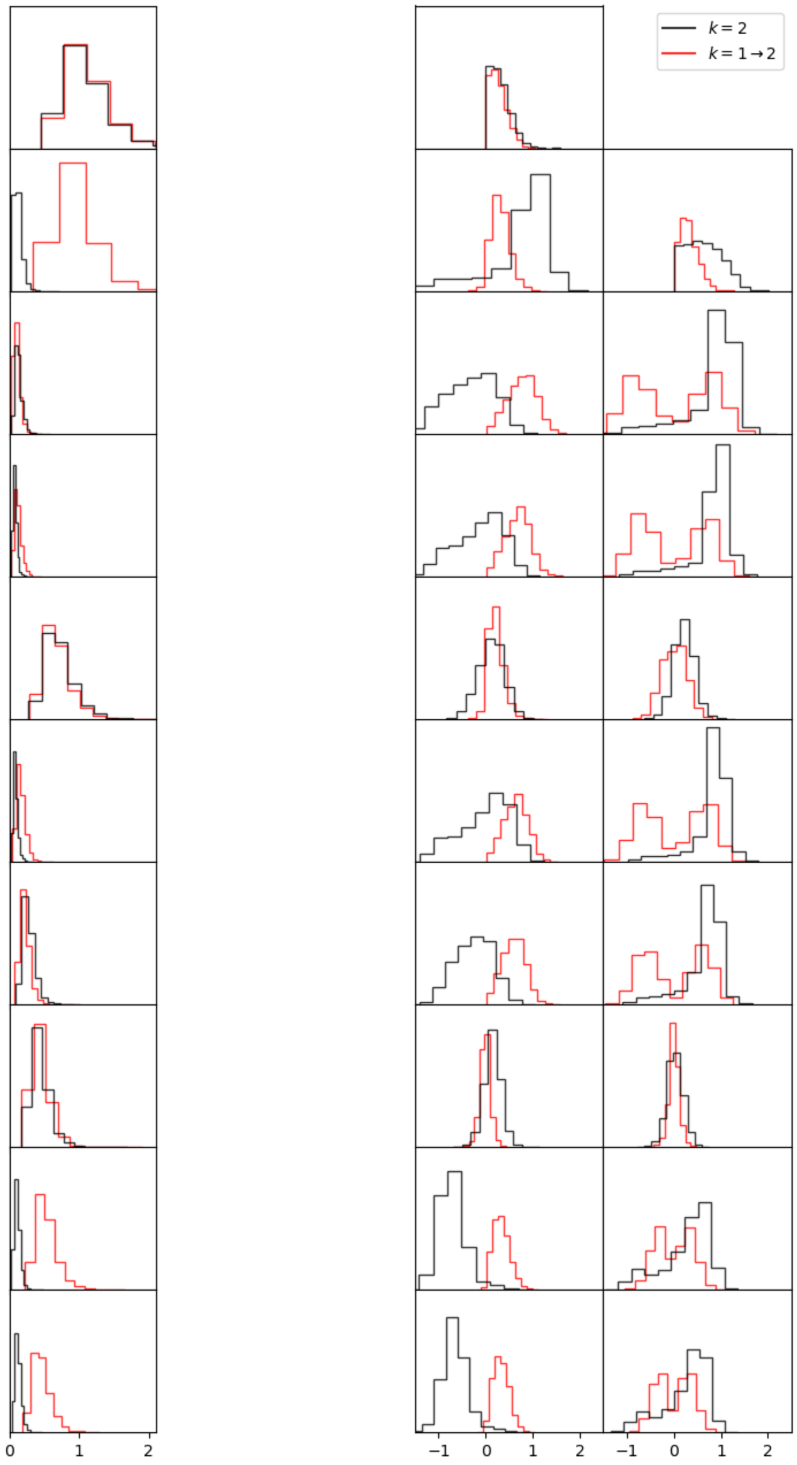
as this choice is general enough to cover most situations, in both under-parameterised and over-parameterised models. For the sign of the last column, we let

$$\gamma_i = \begin{cases} +1 & i = k' \\ \text{Bernoulli}(1, -1) & \text{otherwise}\,. \end{cases} \tag{5.25}$$

We assess the accuracy of the split transform in a similar manner to the merge transform (Figures 5.2-5.3) with the same data with underlying $k = 3$ factors. First, in the under-parameterised case, we run the Gibbs sampler in model $k = 1$ and use the split transform to expand this into the $k = 2$ model. The resulting distributions for $\mathbf{\Lambda}$ and $\mathbf{\Sigma}$ are shown in red in Figure 5.8, compared with the results of the Gibbs sampler for the $k = 2$ model in black. The same is done for $k = 2$ transformed to $k = 3$ in Figure 5.9, $k = 3$ transformed to $k = 4$ in Figure 5.10 and $k = 4$ transformed to $k = 5$ in Figure 5.11. In each of these figures, the red histogram shows how the split move in Equation (5.23) transforms the lower model onto the extended $k + 1$ space, in other words, representing the proposal distribution, while the black histogram represents the target distribution.

For each $\mathbf{\Lambda}$ at a single iteration, the split transform distributes the last column into two based on the auxillary variables u and γ. u determines the second-last column and is selected from a uniform distribution between 0 and 1, which effectively smears the distribution of the first column between 0 and Λ_i for each row i. The final column is determined by the remaining contribution to $\mathbf{\Lambda}$ for each row and the sign is selected at random. This is what leads to the symmetry around $\Lambda_i = 0$ in the second column. This symmetry is seen in all examples but appears more prominent when there are two separate peaks some distance from $\Lambda_i = 0$.

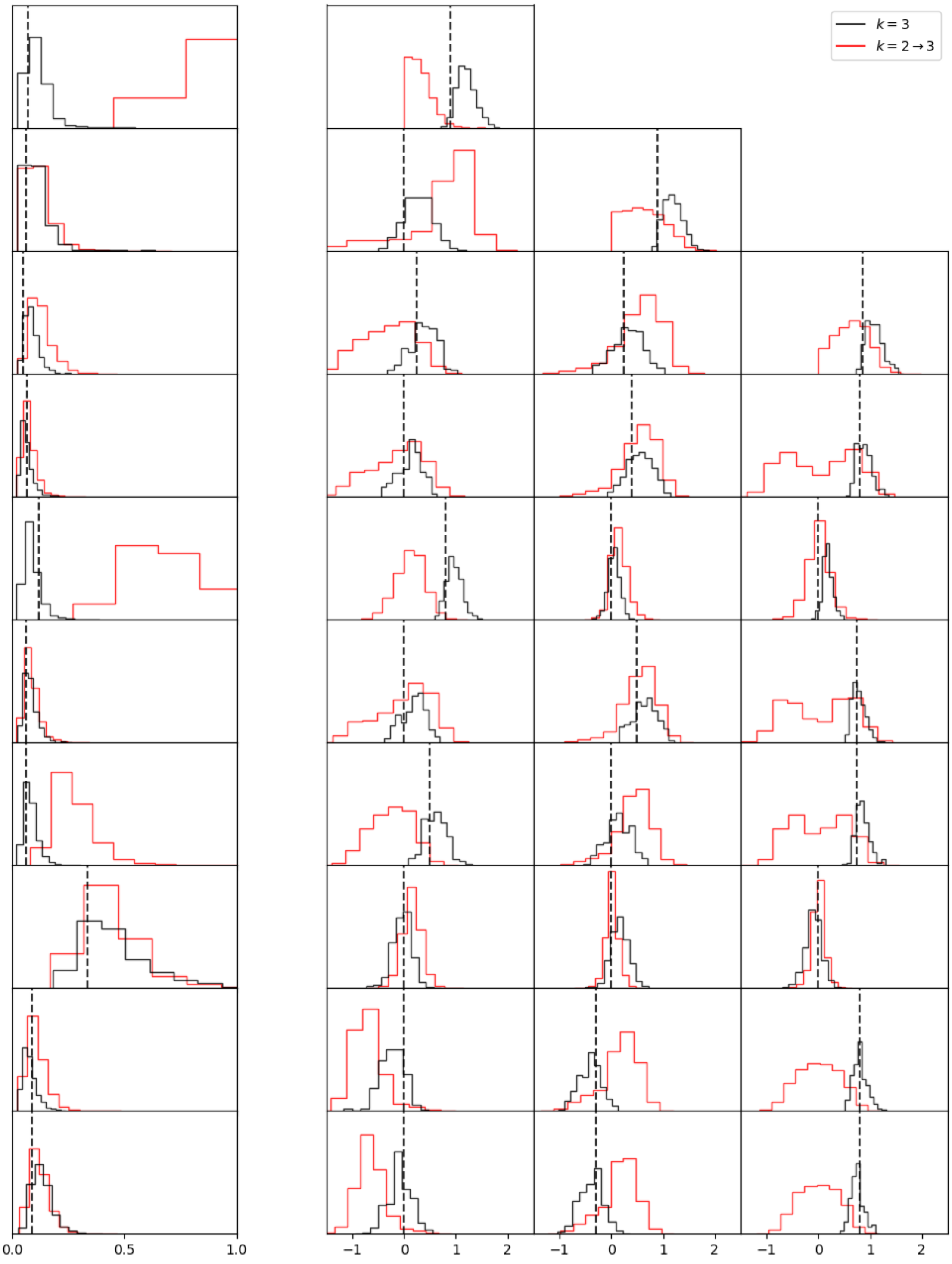
In the under-parameterised regime, the split transform applied to the $k = 1$ and $k = 2$ models give distributions that differ significantly for some of the $\mathbf{\Lambda}$ and $\mathbf{\Sigma}$ variables. This is more problematic for the $k = 1$ model, as this is a poorer approximation to the true $k = 3$ factor model. Most of the $\mathbf{\Lambda}$ variables overlap with the larger models, although this is not always true, for example $\lambda_{91}, \lambda_{10,1}$ in the split $k = 1 \to 2$ and $\lambda_{1,1}$ for $k = 2 \to 3$. This is not ideal for a proposal distribution as it is near-impossible for valid points to be proposed, unless variations on the RJMCMC are made, such as annealing (Section 2.6.1).



(a) $\mathbf{\Sigma}$

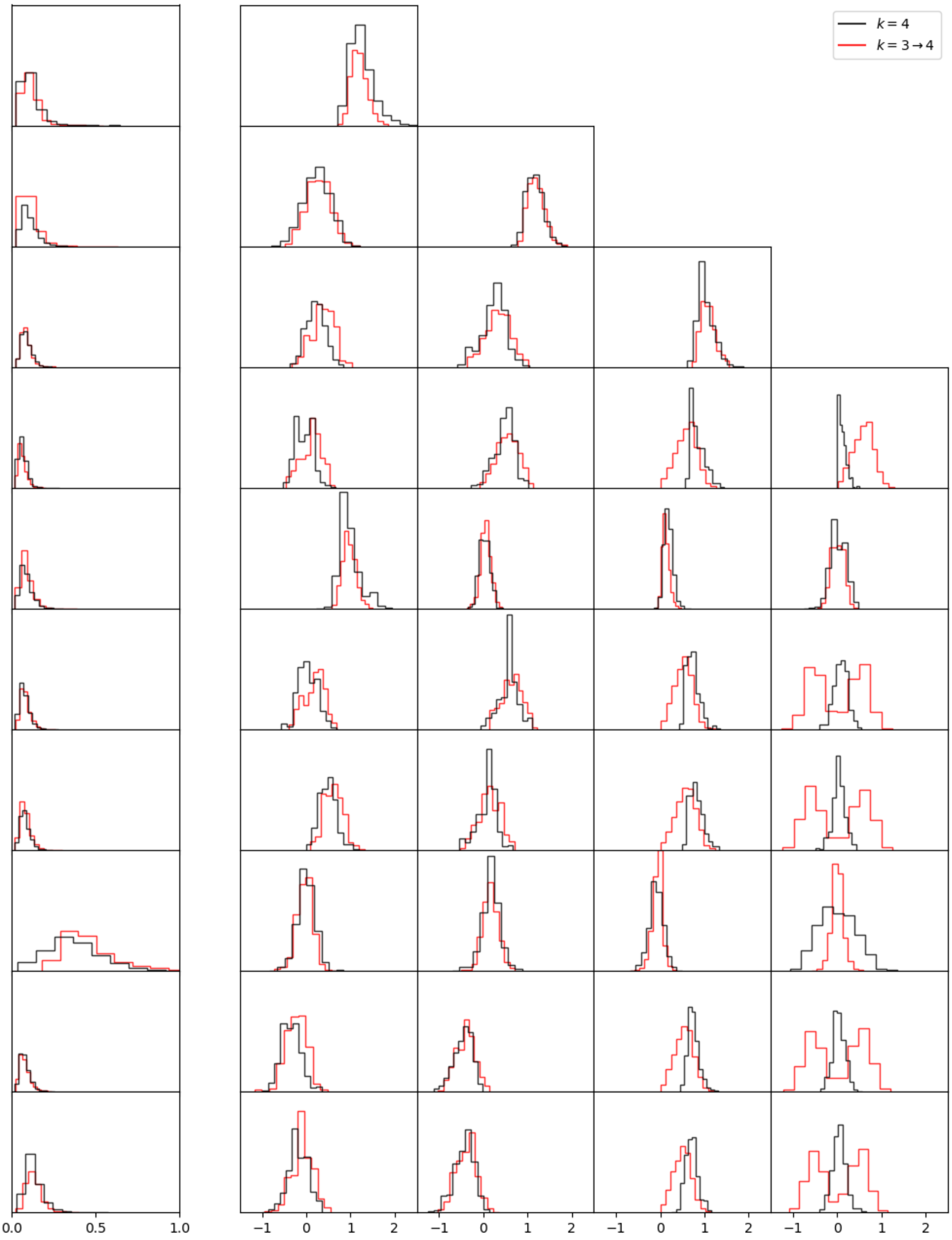
(b) $\mathbf{\Lambda}$

Figure 5.8: Histograms of the parameters of the $k = 2$ model obtained with a Gibbs sampler run for 5000 iterations (after 5000 iterations for burn-in) and in the $k = 1$ model transformed into the $k = 2$ model with the split transform described by Equation (5.23).



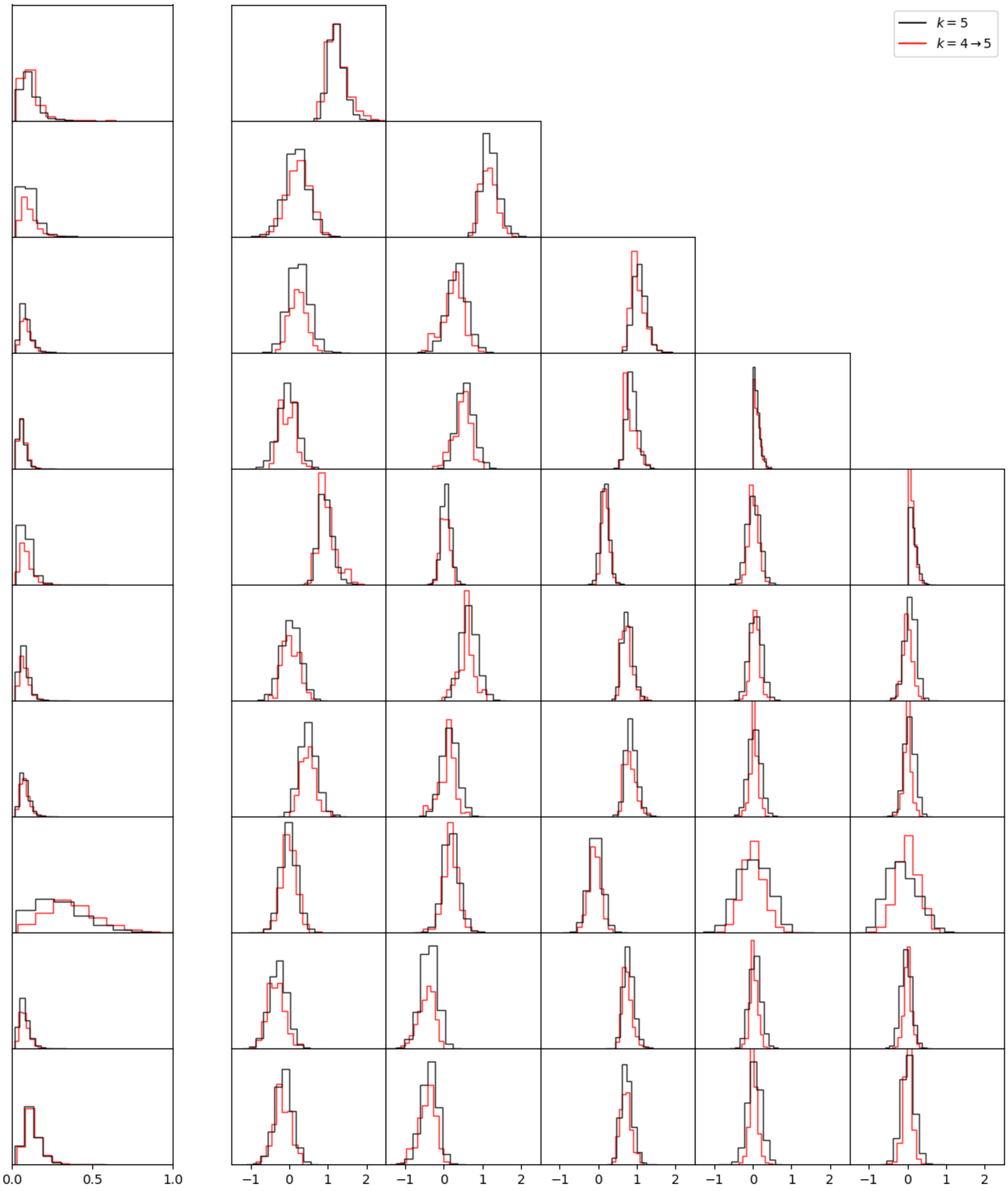
(a) $\mathbf{\Sigma}$ (b) $\mathbf{\Lambda}$

Figure 5.9: Histograms of the parameters of the $k = 3$ model obtained with a Gibbs sampler run for 5000 iterations (after 5000 iterations for burn-in) and in the $k = 2$ model transformed into the $k = 3$ model with the split transform described by Equation (5.23). The black dashed line shows the true values of which the data is simulated from.



(a) $\mathbf{\Sigma}$ (b) $\mathbf{\Lambda}$

Figure 5.10: Histograms of the parameters of the $k = 4$ model obtained with a Gibbs sampler run for 5000 iterations (after 5000 iterations for burn-in) and in the $k = 3$ model transformed into the $k = 4$ model with the split transform described by Equation (5.23).



(a) $\mathbf{\Sigma}$

(b) $\mathbf{\Lambda}$

Figure 5.11: Histograms of the parameters of the $k = 5$ model obtained with a Gibbs sampler run for 5000 iterations (after 5000 iterations for burn-in) and in the $k = 4$ model transformed into the $k = 5$ model with the split transform described by Equation (5.23).

Since the $\mathbf{\Sigma}$ variable is not transformed, it is not surprising that some of these variances do not match when the Gibbs sampler is run in a different model. $\mathbf{\Sigma}$ describes the residual variances which picks up any remaining noise that are not attributed to the underlying factors. When the model is under-parameterised, there are factors that cannot be determined which contributes to the residual variance of this particular row, σ_i^2. This is why the $\mathbf{\Sigma}$ is often over-estimated in the lower model (in row 2, 8 and 9). The same feature is seen when the $k = 2$ model is transformed up to the $k = 3$ space in Figure 5.9 in rows 1, 5 and 7. Again, this makes the proposal suboptimal for targetting the $k + 1$ model when in the under-parameterised regime.

However, the split transform performs much more reasonably in the over-parameterised regime. The transform from $k = 3$ (the correct model) to $k = 4$ (an overparameterised model) shows that the peak of the distribution is sometimes missed in the last column of $\mathbf{\Lambda}$, but there are no regions of the distribution that are not covered by some of the proposal distribution. When the split transform is applied to the $k = 4$ chain which is already over-parameterised, the resulting distributions are very similar to those found directly with the $k = 5$ chain (Figure 5.11). In this over-parameterised regime, the split transform makes a good approximation, particularly since it is only the last column that is added in the transform, of which there are only $m - k = 6$ non-zero elements.

This proposal distribution is likely to be more effective in the over-parameterised regime and therefore result in higher acceptance rates compared to in the under-parameterised regime. It is expected that some adjustments to the RJMCMC may be needed to improve the proposal distribution in the under-parameterised regime, which can be dealt with by introducing annealing distributions (Section 2.6.1). First, however, I will outline how we can construct a reversible jump MCMC algorithm.

5.3.2 RJMCMC Algorithm

In this section, I will discuss the new RJMCMC algorithm that moves between models via these split and merge transforms. Following the same notation as Section 2.5, the proposal distribution for $\theta^{(k')}$ is built by introducing new auxiliary variable, $u^{(k')}$, according to distribution

$\psi_{k\to k'}(u^{(k')})$, i.e.

$$q(\theta^{(k')}, u^{(k')}|k', k, \theta^{(k)}) = \varphi(\theta^{(k')}|u^{(k')}, k', k, \theta^{(k)})\, \psi_{k\to k'}(u^{(k')}|k', k, \theta^{(k)}) \tag{5.26}$$

$$= \varphi(\theta^{(k')})\, \psi_{k\to k'}(u^{(k')}) \tag{5.27}$$

where the second line is used to simplify the notation. These auxiliary variables are those that must be drawn during the split move, u_i for $i = k, \cdots, m$ and γ_i for $i = k+1, \cdots, m$, as described by Equations (5.24) and (5.25) respectively. This gives

$$\psi_{k\to k'}(u^{(k')}|k', k, \theta^{(k)}) = \prod_{i=k}^{m} \psi_{k\to k'}(u_i, \gamma_i) = \prod_{i=k}^{m} \text{Beta}(1,1)\, \text{Bernoulli}_{i\neq k'}(1, -1)\mathbf{1}_{i\neq k}\,. \tag{5.28}$$

Then, to move up in dimension via the split move, we draw these auxiliary variables and apply deterministic functions to calculate $(\mathbf{\Lambda}^{(k')}, \mathbf{\Sigma}^{(k')})$ with Equation (5.23). In contrast, to move down in dimension via the merge move, $(\mathbf{\Lambda}^{(k')}, \mathbf{\Sigma}^{(k')})$ is entirely deterministic and calculated with Equation (5.16). There are no auxiliary variables from model k to $k-1$, but we can calculate the auxiliary variables that would be obtained through the reverse process from model $k-1$ to k. These will be used in the acceptance rate.

We will use this proposal to estimate the ratio $f(y|k')/f(y|k)$, as done in the pseudo marginal ratio algorithm (Section 2.4.7). We estimate this with an importance sampling estimate, drawn from the proposal q described above.

$$\widehat{\left(\frac{f(y|k')}{f(y|k)}\right)} = \frac{f(y|k', \theta'^{(k')})\, p(\theta'^{(k')}|k')}{f(y|k, \theta^{(k)})\, p(\theta^{(k)}|k)} \frac{q(\theta^{(k)}|k, k', \theta'^{(k')})}{q(\theta'^{(k')}|k', k, \theta^{(k)})} \tag{5.29}$$

As we have introduced the auxiliary variables, $u^{(k)}$ and $u^{(k')}$, the ratio becomes

$$\widehat{\left(\frac{f(y|k')}{f(y|k)}\right)} = \frac{f(y|k', \theta'^{(k')})\, p(\theta'^{(k')}|k')}{f(y|k, \theta^{(k)})\, p(\theta^{(k)}|k)} \frac{\varphi(\theta^{(k)})\psi_{k'\to k}(u^{(k)})}{\varphi(\theta^{(k')})\psi_{k\to k'}(u^{(k')})} \left| \frac{\partial\left(\theta^{(k)}, u^{(k)}\right)}{\partial\left(\theta'^{(k')}, u'^{(k')}\right)} \right| \tag{5.30}$$

where the Jacobian is introduced to ensure dimension matching between $q(\theta^{(k)}|k, k', \theta'^{(k')})$ and

$q(\theta'^{(k')}|k', k, \theta^{(k)})$ as described in Equation (2.69). The acceptance ratio from k to k' is

$$r_{RJ}(k \to k') = \frac{f(y|k', \theta'^{(k')})}{f(y|k, \theta^{(k)})} \frac{p(k')\, p(\theta'^{(k')}|k')}{p(k)\, p(\theta^{(k)}|k)} \frac{q(k|k')}{q(k'|k)} \frac{\varphi(\theta^{(k)})\psi_{k' \to k}(u^{(k)})}{\varphi(\theta^{(k')})\psi_{k \to k'}(u^{(k')})} \left| \frac{\partial \left(\theta^{(k)}, u^{(k)}\right)}{\partial \left(\theta'^{(k')}, u'^{(k')}\right)} \right| . \tag{5.31}$$

The algorithm is outlined in Algorithm 8. Like the pseudo marginal target RJMCMC algorithm, presented in (Lopes and West, 2004), the cost of this algorithm scales with $O(m^3)$ due to likelihood (Equation (5.8)) which requires inverting the matrix $\mathbf{\Omega}$ of size $(m \times m)$. However, it does not require the preliminary independent MCMC simulations for each k, making it overall significantly less costly.

Algorithm 8 Reversible jump MCMC algorithm designed for factor analysis that uses split and merge moves.

```
1. Set initial $k$ and draw prior $\theta^{(k)}$. Set $X_0 = (k, \theta^{(k)})$
2. For $n = 1, \cdots, N$, current sample is $X_n = (k, \theta^{(k)})$
    (a) Do 1 MCMC step (Gibbs sampler)
    (b) Do 1 RJ step:
        i. Propose $k' = k + 1$ (split) or $k' = k - 1$ (merge) with probability $q(\cdot|k)$ (Equation (2.63)).
        ii. If $k' = k - 1$: Merge with transform given by Equation (5.16).
        iii. Else $k' = k + 1$: Split by proposing new auxiliary variables, $u, \gamma$ from Equations (5.24)-(5.25) and transform with 5.23.
        iv. Calculate acceptance rate $r_{RJ}(k \to k')$ with Equation (5.31).
        v. With acceptance probability

               $\alpha = \min\left(1, \frac{q(k', k)}{q(k, k')} r_{RJ}(k \to k')\right) ,$        (5.32)

           either accept and set $X_{n+1} = (k', \theta'^{T-1}_{k'})$ or otherwise reject and set $X_{n+1} = X_n = (k, \theta)$.
```

5.3.3 Annealed Reversible Jump

The proposal distribution is built based on the assumption that the merge and split transforms make a reasonable approximation to the posterior distribution for the parameters in model

$k \pm 1$. However, when the model is underparameterised Figure 5.3 and 5.8 showed this not to be the case for the merge and split transforms respectively. This could give low acceptance rates, when k is underparameterised. This leads us to improve the acceptance rate by introducing the method of annealed importance sampling (AIS, Section 2.6.1) embedded within the RJMCMC moves. AIS was first introduced within an MCMC algorithm by Neal (2005) and then applied to RJMCMC by Karagiannis and Andrieu (2013). As described in Section 2.6.1, AIS involves generating a sequence of intermediate probability distributions between the proposal distribution and the target distribution. These are designed to reduce the distance between intermediate distributions so that each intermediate distribution is a better proposal for the next one, leading to higher acceptance rates. We can implement this within a pseudo marginal ratio MCMC algorithm because, as pointed out in Section 2.4.7, the ratio of likelihoods in the acceptance rate can be estimated with an importance sampling estimate, i.e. Equation (2.59). Since the RJMCMC algorithm presented above (Algorithm 8) uses an importance sampling estimate in the acceptance rate (Equation (5.29)), we can apply the concept of AIS to this, in order to improve the acceptance rate . This should reduce the chance of the algorithm becoming 'stuck' due to high rejection rates, which is a common problem in RJMCMC in high dimensions.

We will denote the probability density function of model k by π_k and introduce intermediate probability distributions from $t = 1, \cdots, T-1$ to bridge the gap from π_k to $\pi_{k'}$. Following the same notation as Equation (2.90), these intermediate distribution are

$$\pi_{k \to k';t}(\theta^{(k')}, u^{(k')}) = \left(\pi_{k'}\left(\theta^{(k')}, u^{(k')}\right)\right)^{\gamma_t} \left(\pi_k\left(\theta^{(k)}, u^{(k)}\right)\right)^{1-\gamma_t} \tag{5.33}$$

where γ_t increases from 0 to 1, as t increases from 1 to T. We will use $\gamma_t = \frac{t}{T}$ (Neal, 2001).

Introducing annealing changes Algorithm 8 in two ways. Firstly, the process of proposing new values $\theta^{(k')}$ involves multiple steps across the intermediate distributions, $(\theta_1^{(k')}, u_1^{(k')}), \cdots, (\theta_T^{(k')}, u_T^{(k')})$. Secondly, we must modify the acceptance rate, Equation (5.31), to include this sequence of points, in order to obtain a higher acceptance rate.

For this first task, we require a way to generate samples from $(\pi_{k \to k';1}, \cdots, \pi_{k \to k';T})$. We do this by constructing an MCMC kernel K_t that targets $\pi_{k\to k';t}$ for all t. For simplicity, we will do this by working in the larger dimensional model (k' for the split move or k for the merge move), so that we do not need to build an MCMC move on the auxiliary variables, $u^{(k)}$. This is because in the $k+1$ model, the parameters are $(\theta^{(k+1)}) = (\mathbf{\Lambda}^{(k+1)}, \mathbf{\Sigma}^2)$. Then, the results of this move can then be projected into the smaller model deterministically with the merge transform, to find $(\theta^{(k)}, u^{(k)}) = (\mathbf{\Lambda}^{(k)}, u^{(k)}, \mathbf{\Sigma}^2)$. We use a Metropolis within Gibbs sweep where first we move the $\mathbf{\Lambda}$ parameters incrementally, followed by the $\mathbf{\Sigma}^2$ parameters incrementally (Section 2.4.4). It is logical to update $\mathbf{\Lambda}$ on a row-by-row basis, because the prior and conditional posterior distributions on each row follow a multivariate normal distribution (Equation (5.6) and 5.11). For consistency, we will also use a multivariate normal distribution to update λ_i for each row $i \in 1, \cdots, m$. We will use a normal distribution, truncated at zero, to update σ_i^2 for $i \in 1, \cdots, m$. For both parameters, the proposal distributions are random walks with pre-specified variances Σ_λ and Σ_{σ^2}. As this algorithm targets $\pi_{k\to k';t}$, the acceptance rate is given by

$$r_{MWG,t} = \frac{\pi_{k\to k';t}(\theta^{(k')}, u^{(k')})}{\pi_{k\to k';t-1}(\theta^{(k')}, u^{(k')})} \frac{q\left(\theta^{(k)}, u^{(k)}|\theta^{(k')}, u^{(k')}\right)}{q\left(\theta^{(k')}, u^{(k')}|\theta^{(k)}, u^{(k)}\right)} \tag{5.34}$$

$$= \frac{\left(\pi_{k'}\left(\theta^{(k')}, u^{(k')}\right)\right)^{\gamma_t} \left(\pi_k\left(\theta^{(k)}, u^{(k)}\right)\right)^{1-\gamma_t}}{\left(\pi_{k'}\left(\theta^{(k')}, u^{(k')}\right)\right)^{\gamma_{t-1}} \left(\pi_k\left(\theta^{(k)}, u^{(k)}\right)\right)^{1-\gamma_{t-1}}} \frac{q\left(\theta^{(k)}, u^{(k)}|\theta^{(k')}, u^{(k')}\right)}{q\left(\theta^{(k')}, u^{(k')}|\theta^{(k)}, u^{(k)}\right)} \tag{5.35}$$

$$= \left(\frac{\pi_{k'}\left(\theta^{(k')}, u^{(k')}\right)}{\pi_k\left(\theta^{(k)}, u^{(k)}\right)}\right)^{\gamma_t-\gamma_{t-1}} \frac{q\left(\theta^{(k)}, u^{(k)}|\theta^{(k')}, u^{(k')}\right)}{q\left(\theta^{(k')}, u^{(k')}|\theta^{(k)}, u^{(k)}\right)}. \tag{5.36}$$

This step is described in full in Algorithm 9 in the context of the split move. There is one key difference between this and moving down in dimension via the merge move. Since these two moves must be reversible, we must reverse the order of the updates. Therefore the merge equivalent of this would involve update σ_i^2 first, from $i \in m, \cdots, 1$, followed by updating λ_i from row $i \in m, \cdots, 1$.

We can use Algorithm 9 to generate a path of samples $(\theta_1^{(k)}, u_1^{(k)}), \cdots, (\theta_T^{(k)}, u_T^{(k)})$ from

$$\pi_{k\to k';1}, \cdots, \pi_{k\to k';T}.$$

Algorithm 9 Metropolis within Gibbs algorithm that targets $\pi_{k\to k+1;t}$

```
1. We start with $(\theta^{(k)}_{t-1}, u^{(k)}_{t-1})$ so must project this into the $k+1$ space to obtain $(\theta^{k+1}_{t-1}) = (\mathbf{\Lambda}^{(k+1)}_{t-1}, \mathbf{\Sigma}_{t-1})$
2. $\lambda$ block: For $i \in 1, \cdots, m$
    (a) Propose row $\lambda_i$
        $$\lambda^*_i \sim \begin{cases} \mathcal{N}_i(\cdot; \lambda_i, \Sigma_\lambda)\mathbf{1}_{\lambda^*_{ii}>0} & i < k \\ \mathcal{N}_k(\cdot; \lambda_i, \Sigma_\lambda) & i \geq k \end{cases} \tag{5.37}$$
    (b) Set $(\theta^{*(k+1)}) = (\mathbf{\Lambda}^{*(k+1)}, \mathbf{\Sigma})$ and project back into $k$ space with Equation (5.16) to find $(\theta^{*(k)}, u^{*(k)}) = (\mathbf{\Lambda}^{*(k)}, \mathbf{\Sigma}, u^{*(k)})$.
    (c) Evaluate acceptance rate from Equation (5.36).
    (d) Accept row $\lambda_i = \lambda^*_i$ with probability $\alpha = \min(1, r_{MWG,t})$
3. $\sigma^2$ block: For $i \in 1, \cdots, m$,
    (a) Propose new $\sigma^2_i$
        $$\sigma^{*2}_i \sim \mathcal{N}(\cdot; \sigma^2_i, \Sigma_\sigma)\mathbf{1}_{\sigma^{*2}_i>0} \tag{5.38}$$
    (b) Set $(\theta^{*(k+1)}) = (\mathbf{\Lambda}^{(k+1)}, \mathbf{\Sigma}^*)$ and project back into $k$ space with Equation (5.16) to find $(\mathbf{\Lambda}^{(k)}, \mathbf{\Sigma}^*, u^{*(k)})$.
    (c) Evaluate acceptance rate from Equation (5.36).
    (d) Accept $\sigma^2_i = \sigma^{*2}_i$ with probability $\alpha = \min(1, r_{MWG,t})$
```

These samples are used to calculate the annealed importance weight, first discussed in the context in AIS in Section 2.6.1, where the importance weights are calculated and multiplied at each transition in Equation (2.75). Because we must account for dimension matching, we can follow the same process as the reweighting discussed in Transformation SMC (Section 2.6.4, Equations (2.89)-(2.92)) to obtain annealed importance weights given by

$$w_{AIS} = \prod_{t=1}^{T} \left(\frac{\varphi(\theta_{t-1}^{(k+1)})\, \psi_{(k+1\to k)}(u_{t-1}^{(k+1)}) \left| \frac{\partial(\theta_{t-1}^{(k+1)}, u_{t-1}^{(k+1)})}{\partial(\theta_{t-1}^{(k)}, u_{t-1}^{(k)})} \right|}{\varphi(\theta_{t-1}^{(k)})\, \psi_{(k\to k+1)}(u_{t-1}^{(k)})} \right)^{\gamma_t - \gamma_{t-1}} . \tag{5.39}$$

The final value in the sequence of samples, $(\theta_T^{(k+1)}, u_T^{(k+1)})$, is then a proposed point from the target distribution $\pi_{k\to k+1;T} = \pi_{k+1}$. We then calculate the acceptance rate with

$$r_{AIS} = \frac{q(k|k')}{q(k'|k)} w_{AIS} \tag{5.40}$$

i.e. using annealed importance sampling to estimate the ratio of π_{k+1} to π_k.

5.3.4 Multiple-points Reversible Jump

In the above section, we increased the number of annealing transitions between each proposed point, in order to increase the acceptance rate and thus improve mixing. The algorithm could also be improved in terms of convergence. One way to do this is to increase the number of samples of $(k, \theta^{(k)})$ used to estimate the acceptance ratio, since this would lead to a lower variance. This could be done in the pseudo-marginal example, where we can increase the number of importance sampling points to reduce the variance of the estimate of $f(y|k)$. However, in the pseudo-ratio case, we are estimating the ratio $\frac{f(y|k')}{f(y|k)}$. In this case, increasing the number of points N does not lead to unbiased estimator (Andrieu et al., 2018). Andrieu et al. (2020) introduce a novel a way around this by using averaged acceptance ratios.

The main idea of Andrieu et al. (2020) is to average the acceptance rate, over many possible realisations of the latent variable θ, noting that in the case of reversible jump, each θ exist

Algorithm 10 Annealed reversible jump MCMC designed for factor analysis that uses intermediate distributions between moves.

```
1. Set initial $k$ and draw prior $\theta^{(k)}$. Set $X_0 = (k, \theta^{(k)})$
2. For $n = 1, \cdots, N$, current sample is $X_n = (k, \theta^{(k)})$
   (a) Do 1 MCMC step (algorithm 6)
   (b) Do 1 aRJ step:
       i. Propose $k' = k+1$ (split) or $k' = k-1$ (merge) with probability $q(\cdot|k)$ (Equation (2.63)).
       ii. Dimension match by drawing $u_0^{(k \to k')}$ and compute $(\theta_0^{(k')}, u_0^{(k' \to k)})$ with Equation (5.16) or 5.23)
       iii. Annealing procedure: generate path from $t = 1, \cdots, T$ of samples where $(\theta_t^{(k')}, u_t^{k' \to k}) \sim \pi_{k \to k';t}$ using algorithm 9.
       iv. Calculate annealing importance weight given by Equation (5.39) and acceptance rate $r_{AIS}$ with Equation (5.40).
       v. With acceptance probability
              $\alpha = \min(1, r_{AIS})\,,$    (5.41)
          either accept and set $X_{n+1} = (k', \theta_T^{(k')})$ or otherwise reject and set $X_{n+1} = X_n = (k, \theta^{(k)})$
```

on different spaces defined by k. This is done by sampling many realisations of θ' from proposal distribution $q(\theta'|k', k, \theta)$. Labelling these by p and generating N_P samples, we have $\theta'^{(1)}, \cdots, \theta'^{(N_P)} \sim q(\cdot|k', k, \theta)$, each which has its own acceptance rate, $r_{\theta^{(p)}}(k \to k')$ calculated with Metropolis-Hastings (Equation (2.50)). Based on these acceptance rates, which are effectively weights, we would sample one point to carry through to the next iteration, if accepted. This step would provide us with the 'best' point from the collection of samples, which could be accepted with an averaged acceptance rate given by

$$r_\theta^{(N_P)}(k \to k') = \frac{1}{N_P} \sum_{p=1}^{N_P} r_{\theta^{(p)}}(k \to k')\,. \tag{5.42}$$

Using this would reduce the variance of the estimator, however, it breaks detailed balance with respect to π. Therefore, to preserve reversibility we must add an extra step in which we randomly choose with probability β between two sampling mechanisms for θ':

1. Sample from $q(\theta'|k', k, \theta)$ as described above:

   $$\theta'^{(1)}, \cdots, \theta'^{(P)} \sim q(\cdot|k', k, \theta)\,. \tag{5.43}$$

   From these, sample $s \sim \mathcal{P}\left(r_{\theta^{(1)}}(k \to k')\right) \cdots \left(r_{\theta^{(N_P)}}(k \to k')\right)$. Accept $\theta'^{(s)}$ with averaged acceptance rate given by Equation (5.42).

2. Sample one point from $q(\theta'|k', k, \theta)$, and then sample the remainder $N_P - 1$ points from the reverse proposal distribution given this first sample $q(\theta|k, k', \theta'^{(1)})$, i.e.

   $$\theta'^{(1)} \sim q(\cdot|k', k, \theta) \tag{5.44}$$

   $$\theta'^{(2)}, \cdots, \theta'^{(P)} \sim q(\cdot|k, k', \theta'^{(1)})\,. \tag{5.45}$$

   Accept $\theta'^{(1)}$ with averaged acceptance rate $\frac{1}{r_{\theta'}^{(P)}}(k' \to k)$ where

   $$r_{\theta'}^{(P)}(k' \to k) = \frac{1}{P} \sum_{p=1}^{P} r_{\theta'^{(1)}}(k' \to k)\,. \tag{5.46}$$

This is shown to preserve reversibility in Andrieu et al. (2020). This can also be called an asymmetric acceptance rate, due to the asymmetry that is introduced with these two different sampling mechanisms (Andrieu et al., 2018). β must be chosen so that both sampling methods are selected with equal probability. For MCMC, this is typically a fixed constant $\frac{1}{2}$. However, Andrieu et al. (2020) note that β can be chosen to a be a function of the proposed value k'. This is useful when a particular sampling mechanism is favoured for certain values of k' that can, for instance, reduce cost. For RJMCMC, we can exploit this asymmetry so that sampling mechanism 1 is used when undergoing a split move up in dimension and sampling mechanism 2 is used when undergoing a merge move down in dimension. This means that when moving up in dimension, a greater variety of proposed moves are explored, and the chance that one of these moves is accepted is increased. When moving down in dimension (the deterministic merge move) a variety of reverse split moves are explored to see if the choice to accept a transform down would be favoured relative to other possible parameter values.

Algorithm 11 Multiple point reversible jump designed for factor analysis

```
1. Set initial $k$ and draw prior $\theta^{(k)}$. Set $X_0 = (k, \theta^{(k)})$
2. For $n = 1, \cdots, N$, current sample is $X_n = (k, \theta^{(k)})$
   (a) Do 1 MCMC step (algorithm 6)
   (b) Do 1 mRJ step:
       i. Propose $k' = k + 1$ (split) or $k' = k - 1$ (merge) with probability $q(\cdot|k)$ (Equation (2.63)).
       ii. If split $k' = k + 1$ (sampling mechanism 1)
           A. Sample $N_P$ split moves by sampling auxiliary variables and applying deterministic function to obtain (Equation (5.23))
              $$\left(\theta'^{(k')}, u^{(k')}\right)^{(1)}, \cdots, \left(\theta'^{(k')}, u^{(k')}\right)^{(N_P)} \sim q\left(\cdot|k', k, \left(\theta^{(k)}, u^{(k)}\right)\right)$$
           B. Evaluate acceptance rate for each sample $r_{\theta^{(1)}}(k \to k')), \cdots, r_{\theta^{(N_P)}}(k \to k')$ with Equation (5.31).
           C. Sample
              $$s \sim \mathcal{P}\left(r_{\theta^{(1)}}(k \to k')), \cdots, (r_{\theta^{(N_P)}}(k \to k')\right)$$
           D. Evaluate averaged acceptance rate, Equation (5.42)
           E. With acceptance probability
              $$\alpha = \min\left(1, r_{\theta}^{(N_P)}(k \to k')\right) , \qquad (5.47)$$
              either accept and set $X_{n+1} = (k', \theta'^{(k')(s)})$ or otherwise reject and set $X_{n+1} = X_n = (k, \theta^{(k)})$
       iii. Else merge $k' = k - 1$ (sampling mechanism 2)
           A. Sample 1 merge move with deterministic Equation (5.16) to obtain $(\theta'^{(k')}, u^{(k')(1)}$.
           B. Evaluate acceptance rate for this sample, $r_{\theta^{(1)}}(k \to k')$ with Equation (5.31) and set $r_{\theta'^{(1)}}(k' \to k)) = \frac{1}{r_{\theta^{(1)}}(k \to k')}$
           C. Sample $N_P$ split moves from this by sampling auxiliary variables and applying deterministic function to obtain (Equation (5.23))
              $$\left(\theta'^{(k)}, u^{(k)}\right)^{(2)}, \cdots, \left(\theta'^{(k)}, u^{(k)}\right)^{(P)} \sim q\left(\cdot|k, k', \left(\theta'^{(k')}, u^{(k')}\right)^{(1)}\right)$$
           D. Evaluate acceptance rate for each sample, $r_{\theta'^{(2)}}(k' \to k)), \cdots, r_{\theta'^{(N_P)}}(k' \to k)$ with Equation (5.31).
           E. Evaluate averaged acceptance rate, Equation (5.46)
           F. With acceptance probability
              $$\alpha = \min\left(1, \frac{1}{r_{\theta'}^{(P)}}(k' \to k)\right) , \qquad (5.48)$$
              either accept and set $X_{n+1} = (k', \theta'^{(k')(1)})$ or otherwise reject and set $X_{n+1} = X_n = (k, \theta^{(k)})$
```

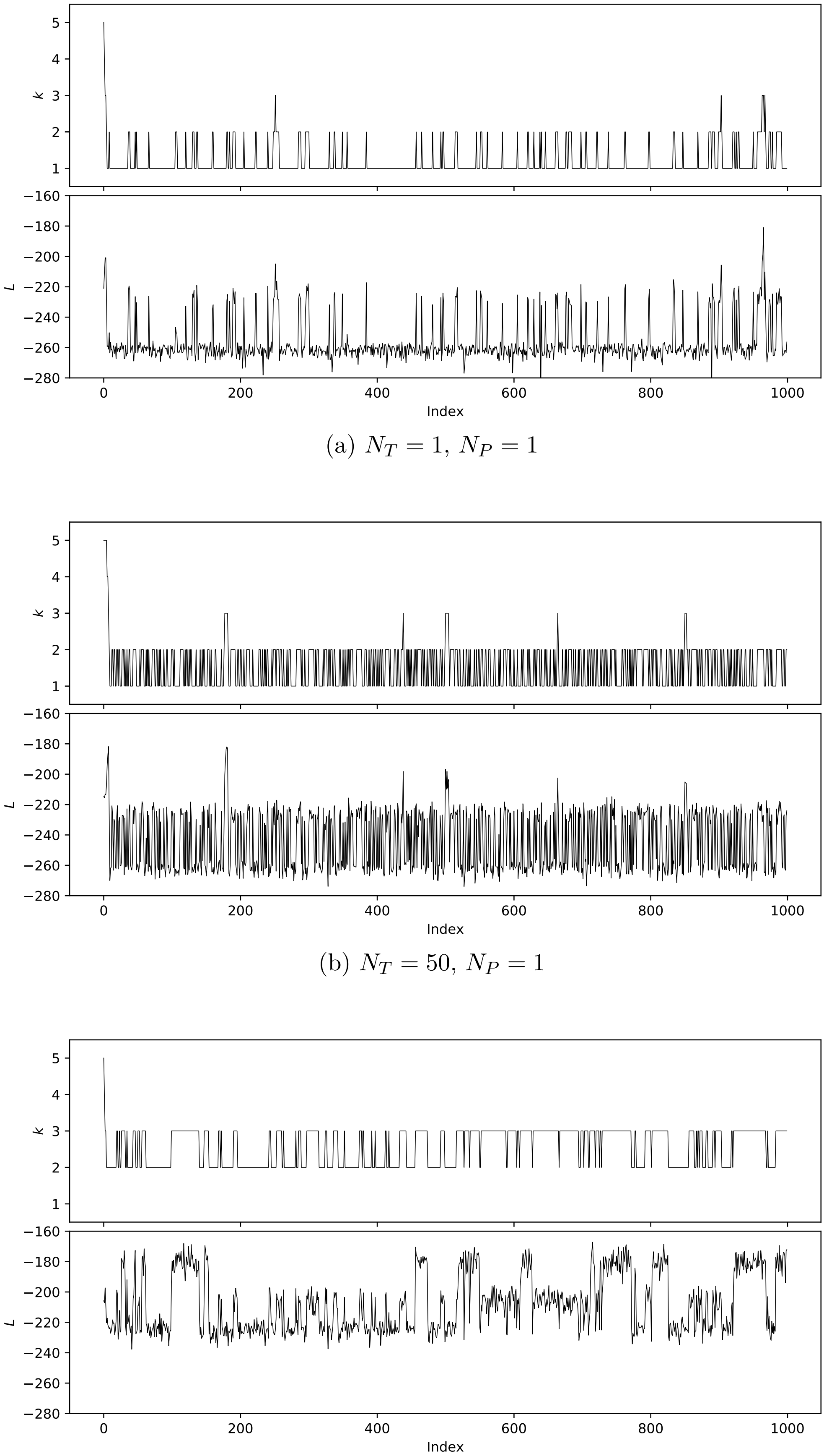
The reversible jump version of this algorithm is outlined in Algorithm 11. This algorithm was designed as an alternative to the Metropolis-Hastings step and in particular, when applied to reversible jump algorithms, it was found to have significant improvements in performance. These improvements were comparable to those found by introducing multiple annealing steps when both were applied to a multiple change-point algorithm (Andrieu et al., 2020). However, these two algorithms have not yet been tested on the factor analysis. Finally, we may wish to combine both of these algorithms, which is possible by replacing the proposal distributions with the annealing procedure outlined in Algorithm 10. This would also require using Equations (5.39)-(5.40) to calculate the acceptance rate for each sample, instead of Equation (5.31).

Next, we will compare the outcomes of these different RJMCMC algorithms, with increasing intermediate steps $N_T > 1$ and increasing number of samples $N_P > 1$. We will compare these methods to those of Dunson (2006) and Lopes and West (2004).

5.3.5 Example: 3 Factor Model

First, the RJMCMC algorithm is demonstrated on the simulated data where there are 3 underlying factors (Section 5.2.1). Following the same process as used on the pmRJ algorithm, we check the trace plots for convergence. Figure 5.12 shows the trace plot of k and the log likelihood. Here we compare the basic RJ algorithm in Figure 5.12a, alongside the results of increasing the number of intermediate distributions to $N_T = 50$ in Figure 5.12b and of also increasing the number of 'particles' to $N_P = 50$ in Figure 5.12c.

When $N_T = 1$ and $N_P = 1$, Figure 5.12a shows that the algorithm spends most of the time in the $k = 1$ model, with several jumps into the $k = 2$. However, once in the $k = 2$ model, the algorithm does not spend long there and quickly jumps back into the $k = 1$ model. It does not move much between these and has a high rejection rate, which is not unusual for RJMCMC as it can be difficult to propose entirely new parameters in the additional dimensions (e.g. Green and Mira, 2001; Al-Awadhi et al., 2004). For this problem in particular, the merge move is successful (particularly in the over-parameterised regime e.g. Figure 5.2) and thus quickly accepted, but once in the under-parameterised regime the split move is a poor proposal for the $k + 1$ model



(a) $N_T = 1$, $N_P = 1$

(b) $N_T = 50$, $N_P = 1$

(c) $N_T = 50$, $N_P = 50$

Figure 5.12: Trace plots of (top panel) k and (bottom panel) the log likelihood for the first 1000 iterations of RJMCMC with (a) $N_T = 1$ and $N_P = 1$, (b) $N_T = 50$ and $N_P = 1$ and (c) $N_T = 50$ and $N_P = 50$ on the simulated data where the true number of factors is $k = 3$.

(e.g. Figure 5.8). This means the algorithm does not explore the parameter space well, further highlighted by the sudden jumps in the log likelihood.

The introduction of more intermediate distributions $N_T > 1$ improves the probability of acceptance of the split transform, as shown in Figure 5.12b. These trace plots explore the same area of the parameter space but with significantly more mixing between models $k = 1$ and $k = 2$. Introducing intermediate distributions has a large effect here, because the proposal is further from the target distribution when in the underparameterised regime (Figure 5.8). However, the split move to $k = 3$ is still rarely accepted, and the merge move appears to be favoured significantly. The asymmetry between these two moves is motivation for introducing more particles $N_P > 1$. As described above, this involves comparing N_P different moves at each step, which should increase the probability of proposing a suitable split move, and ensure that the merge move is accepted when it is actually better than alternative options. This leads the algorithm to have a more even distribution between split and merge moves, and we find that the model favours $k = 2$ and $k = 3$ in Figure 5.12c. Furthermore the likelihood, which summarises the effect of all parameter values, is generally higher, indicating a better model fit has been found under these models.

In short, the stark differences between the quality of the split move and the merge move leads to major asymmetries in acceptance rates when moving up and down in dimension when $N_P = 1$. Increasing $N_P > 1$ is a remedy for this, at each step, a wider range of possible moves are evaluated. This increases the quality of the proposed moves, leading to convergence to the true distribution. Meanwhile, increasing $N_T > 1$ improves the probability of acceptance, by bridging the gap between the proposal and target distribution. Table 5.1 summarises the effects of introducing $N_T > 1$ and $N_P > 1$.

We explore this further by assessing the posterior distribution of k for different combinations of N_T and N_P. The algorithm is run for $N = 10^4$ steps and the first 5000 iterations discarded as burn-in. This is repeated with 20 independent chains initialised with different seeds so that the distributions can be assessed with the box-plots in Figure 5.13.

First, in Figure 5.17a, we vary N_P while fixing $N_T = 100$. When $N_P = 1$, the algorithm spends

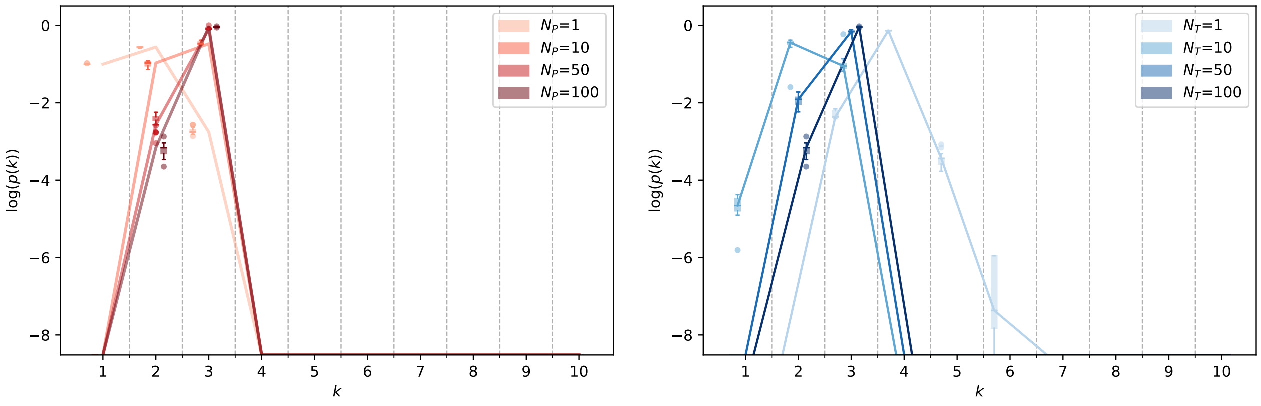
Table 5.1: Summary of behaviour of RJMCMC under increasing the number of annealing distributions (N_T) and the number of points (N_P)

	$N_P = 1$	$N_P > 1$
$N_T = 1$	Split rarely accepted Merge frequently accepted	Split proposal improved Merge acceptance reduced when underparameterised
$N_T > 1$	Split acceptance increased	Split proposal/acceptance improved Merge acceptance reduced when underparameterised

the majority of the time in $k = 1$ and $k = 2$, but the merge move is quickly accepted, as in the trace plot in Figure 5.12b. However, as discussed above, increasing N_P leads to a better quality proposal which leads to quicker convergence to the true distribution. For $N_P = 50$ and $N_P = 100$, the distributions are very similar, with the peak at $k = 3$, which is true for the underlying model.

Next, in Figure 5.17b, we keep N_P fixed at $N_P = 100$ and plot the distribution of k for different values of N_T. When $N_T = 1$, but $N_P = 100$, the algorithm does not accept many merge moves because increasing N_P reduces the acceptance rate of the merge. This gives the peak incorrectly at $k = 4$. Increasing N_T assists with the acceptance of both merge and split moves particularly when the proposal is distant from the posterior, leading to a posterior distribution with a peak at $k = 3$ when $N_T = 50$ and $N_T = 100$. The findings presented here differs slightly from results found by Medina-Aguayo et al. (2020), where adding intermediate distributions did not substantially improve results. However, the problem here is slightly different and the proposal is often far from the truth, especially when in the underparameterised regime, which are the cases that would benefit the most from additional intermediate distributions. The posterior distribution is consistent for different variations where $N_T \geq 50$ and $N_P \geq 50$, indicating that these parameters are suitable to achieve convergence to the posterior distribution.

These distributions on k differ significantly from those found from the pmRJ algorithm (Figure 5.6), with a much sharper peak in the distribution at $k = 3$. Unlike pmRJ, this algorithm never revisits large k after burn-in, giving $k > 4$ small posterior probabilities (less than 1 in 5×10^3 iterations). This is because of the difficulty in proposing a split move to higher dimensional models, via the transform, in contrast to the pmRJ proposal distribution that is



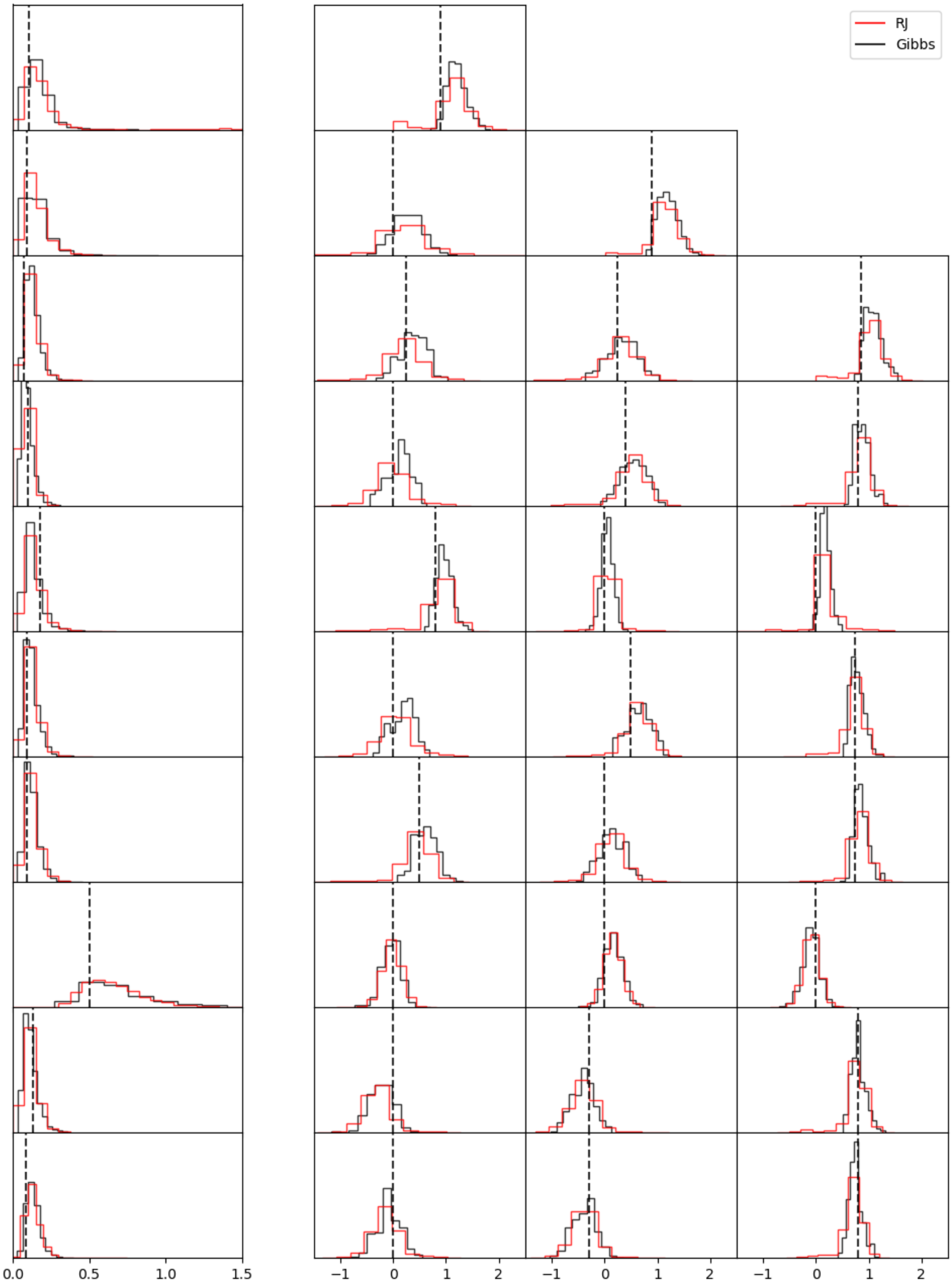
(a) Distributions for $N_P = (1, 10, 50, 100)$, with fixed $N_T = 100$

(b) Distributions for $N_T = (1, 10, 50, 100)$, with fixed $N_P = 100$

Figure 5.13: Log of posterior distributions on k under different settings of (a) number of particles, N_P, and (b) number of intermediate distributions, N_T.

built from preliminary MCMC runs. Furthermore, the pmRJ proposals do not depend on the current $\theta^{(k)}$ values, so new points are proposed without considering all information regarding the current iteration. As a result, pmRJ explores the full parameter space more, while the method presented here spends more time in models closer to the peak of the posterior distribution. This can be beneficial, as computational effort is not spent simulating models with low posterior probabilities.

As before, we should check that the parameters have converged to the correct posterior distribution. Figure 5.14 shows the distribution of the $\theta^{(k)}$ when $k = 3$ in the $N_T = 100$, $N_P = 100$ simulation in red, with the true value of $\mathbf{\Lambda}$ and $\mathbf{\Sigma}$ shown by the black dashed line and the result of the Gibbs sampler with $k = 3$ fixed shown by the black histogram. The Gibbs sampler and RJMCMC with $k = 3$ give very similar distributions, as expected because the RJMCMC uses the Gibbs sampler as within model moves. Although we cannot ever be 100% sure, this is some evidence that the RJMCMC is converging to the correct distribution for θ. Furthermore, the distributions agree with the Gibbs sampler better than the results found from pmRJ in Figure 5.7, which gave a distribution too narrow relative the Gibbs sampler for some components. This indicates that the proposal distribution presented here is better suited as areas of the underlying parameter space are explored better.



(a) $\mathbf{\Sigma}$

(b) $\mathbf{\Lambda}$

Figure 5.14: Histogram of samples where $k = 3$ from RJ algorithm with $N_P = 100$ and $N_T = 100$ (red) compared against Gibbs sampler with fixed $k = 3$ (black). Both algorithms are run for 5000 iterations (after 5000 iterations for burn-in). The black dashed line shows the true values of which the data is simulated from.

5.3.6 Computational Cost

Overall, we see major improvements to the RJ algorithm when adding in both multiple intermediate distributions and multiple points. These improvements appear at around $N_T = 50$ and $N_P = 50$. Increasing N_T and N_P beyond this does not significantly change the distribution of k but is more expensive. In this section, we will explore the cost of these different algorithms.

Figure 5.15 shows the approximate computational cost in seconds per 10000 iterations of each algorithm, run on a single Intel Xeon Processor E5-2680 v3 (Intel®). The first column shows the cost of the MCMC Gibbs sampler, when k is known. The variance on this is high due to the difference in cost between setting $k = 1$ and $k = 10$.

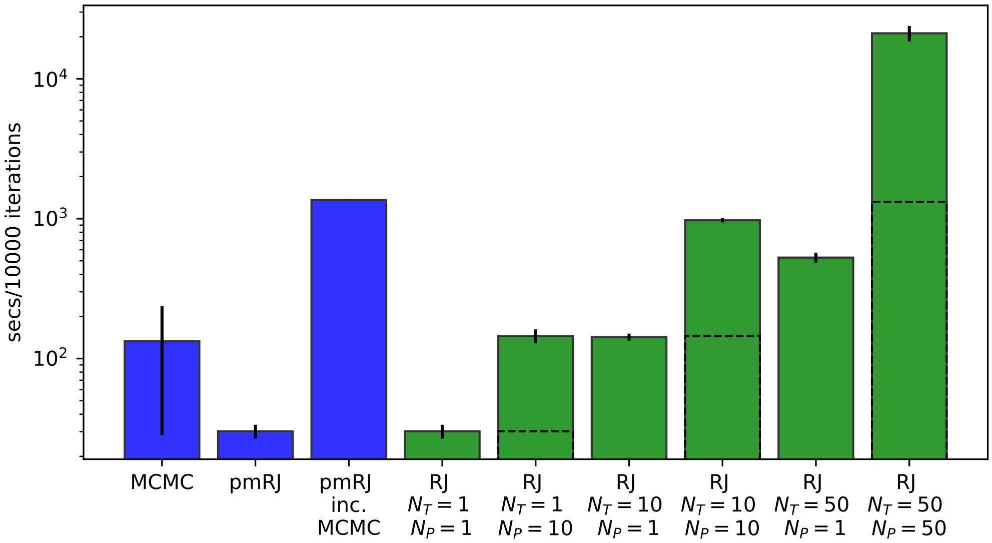


Figure 5.15: Computational cost of different RJMCMC methods measured in time per 10000 iterations run on a single core. Uncertainty bars estimate 1 standard deviation across 20 independent simulations run. The dashed line indicates theoretical cost that could be achieved if parallelised across 24 cores.

The cost of pmRJ and RJ are very similar as they consist of the same structure of within model and between model moves. Both scale with $O(m^3)$ due to the likelihood evaluation at the end of each RJ step. However, the total cost of the m independent MCMC makes pmRJ around an order of magnitude more expensive than the basic RJ method. Here we have used 10^4 iterations of the independent MCMC chains, and assume that the first 5000 iterations are discarded as burnin so the remaining 5000 can be used for obtaining the proposal distributions. Furthermore, the cost of the Gibbs sampler scales with $O(m^2)$, so this additional cost of pmRJ scales with $O(m^3)$, meaning computational savings would be increased with larger m.

However, we found the basic RJ version of the algorithm does not feature ideal convergence or mixing properties. Introducing $N_T > 1$ and $N_P > 1$ improves these properties but increases the computational complexity with $O(N_T N_P)$. Figure 5.15 shows that with $N_T = 10$ or $N_P = 10$, there is a large computational benefit over pmRJ, but extending this to both $N_P = 10$ and $N_T = 10$ under the single node set-up used here becomes comparable to pmRJ in terms of cost. However, the multiple 'particles' can be parallelised by proposing and evaluating the acceptance rates of each particle independently (Steps 2(b)ii.A-B and 2(b)iii.B-C in Algorithm 11). These acceptance rates can be averaged at the end of each loop to find the modified acceptance rate. The dotted lines indicate the theoretical cost that could be achieved with this parallelisation with up to 24 available cores. Therefore, the choice of $N_T = 10$, $N_P = 10$ on parallelised architecture could achieve computational savings of around an order of magnitude relative to the pmRJ algorithm presented in Lopes and West (2004), which would increase further with m as the algorithms both scale with $O(m^3)$.

We found that overall, setting $N_T = 50$ and $N_T = 50$ gave the better convergence properties. As the final column of Figure 5.15 shows, this is computationally more expensive than pmRJ without parallelisation, but has the potential to be on par in terms of cost, if run with 24 cores. This measure of cost, however, does not represent the other key benefits of choosing this algorithm over pmRJ:

1. The difference between the distributions on k differ significantly, with the RJ algorithm presented here finding a peak at the correct value $k = 3$ (Figure 5.13) while pmRJ favours $k = 2$ (Figure 5.6).

2. The algorithm presented here spends less computational effort on low probability models and converges more rapidly to $k = 3$, meaning fewer iterations may be required.

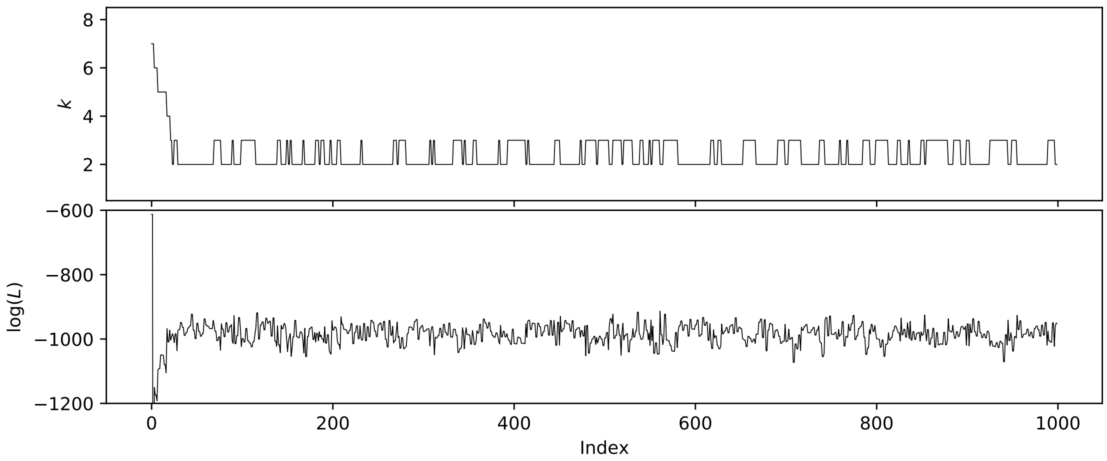
3. The converged parameter distributions on $\mathbf{\Lambda}^{(k)}$ and $\mathbf{\Sigma}^{(k)}$ for $k = 3$ match the Gibbs sampler better (Figure 5.14 compared to Figure 5.7).

4. In pmRJ, each independent MCMC simulation must reach convergence, so a large portion of the computational time and effort is spent simulating points that will be thrown away as a part of burn-in. This issue is enhanced for larger m.

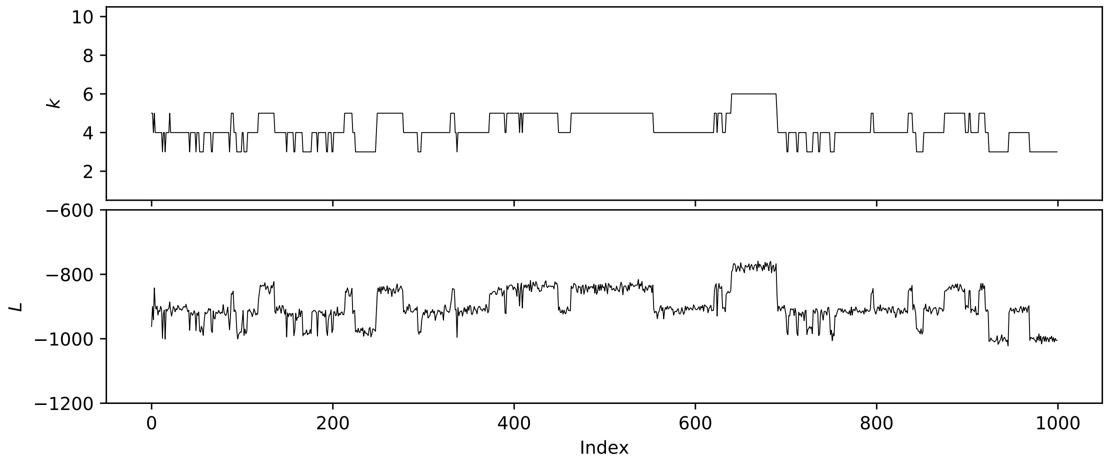
5.3.7 Example 2: 7 Factor Model

These RJMCMC algorithms have also been tested on a larger simulated dataset. This dataset has been simulated in the same way, with $m = 14$ observed variables, $k = 7$ latent factors and $N_{obs} = 100$ observations. First, we check the trace plots of the relevant algorithms, testing this on pmRJ in Figure 5.16a and on the new RJMCMC method developed here, setting $N_P = 100$ and $N_T = 100$ in Figure 5.16b. The pmRJ algorithm suffers from poor mixing and rarely explores beyond $k = 2$, which would lead to the incorrect conclusion that there are $k = 2$ underlying factors. Previously, this algorithm also underestimated the number of factors on the 3 factor model, however, the problem is significantly worse here. This is because one of the inherent difficulties is proposing new variables that are accepted, which is increasingly difficult as the dimensions increase, because more variables must be proposed (Agapiou et al., 2017). In pmRJ, the entire $\mathbf{\Lambda}$ and $\mathbf{\Sigma}$ matrix is proposed at each RJ step, giving up to $m \times (k + 1)/2 + m$ new variables to be proposed. As m increases, the number of variables to be proposed increases linearly and therefore the probability of acceptance decreases exponentially (Agapiou et al., 2017). In contrast, the new approach taken here uses the previous value of θ and only the last column of $\mathbf{\Lambda}$ is proposed, i.e. $m - k$ variables. This still scales linearly with increasing m and so the probability of acceptance still decreases exponentially, but the introduction of annealing intermediate distributions reduces this problem, as discussed in Section 5.3.3.

Figure 5.17 explores the posterior distribution on k for different values of N_T and N_P. As before, we find the same posterior distributions when increasing N_T or N_P beyond 100, suggesting few changes to the inferred distribution when extending N_T and N_P beyond this. However, the algorithm still substantially underestimates the true underlying factors ($k = 7$), suggesting that the optimal choice would be $k = 4$. Previously we had seen some evidence that this RJMCMC favours moves down in dimension, and here we see that as a consequence, this approach is



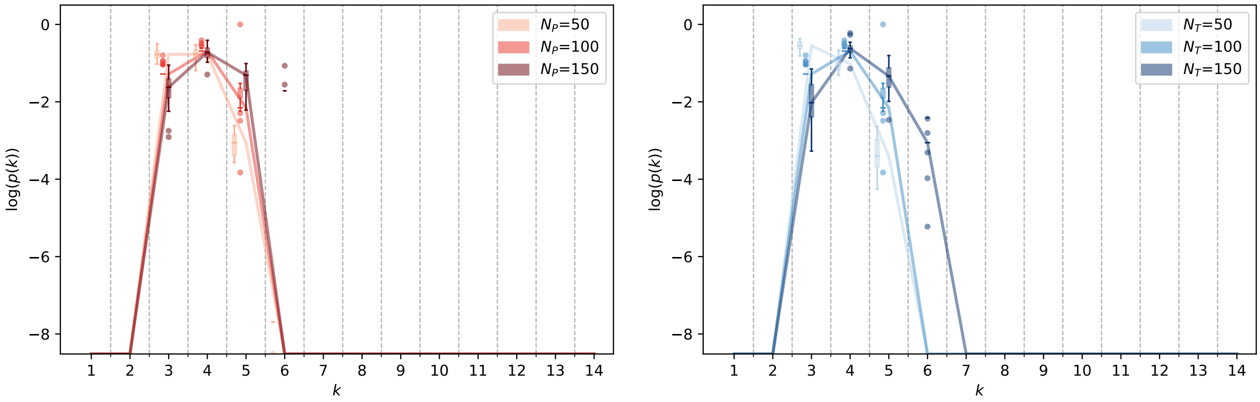
(a) pmRJ



(b) RJ with $N_T = 100$ and $N_P = 100$

Figure 5.16: Trace plots of (top panel) k and (bottom panel) the log likelihood for the first 1000 iterations of (a) pmRJ and (b) RJ with $N_T = 100$ and $N_P = 100$ on the simulated data where the true number of factors is $k = 7$.

prone to underestimating the optimal number of factors. Although this is significantly better than pmRJ, it is still not ideal, particularly as larger datasets would be of interest. In the next section we will explore another approach to estimating the posterior distribution on k.



(a) Distributions for $N_P = (50, 100, 150)$, with fixed $N_T = 100$

(b) Distributions for $N_T = (50, 100, 150)$, with fixed $N_P = 100$

Figure 5.17: Log of posterior distributions on k under different settings of (a) number of particles, N_P, and (b) number of intermediate distributions, N_T.

5.4 Sequential Monte Carlo

This chapter so far has shown how a variety of different reversible jump algorithms can be applied to estimate the posterior distribution on the number of latent factors. However, in high dimensions, these methods appear to have poor mixing and convergence properties making it difficult to know whether or not samples from these algorithms are from the correct posterior distribution. As an alternative, we could make use of Sequential Monte Carlo (SMC, Section 2.6) in this task. With SMC, we can sample from the posterior distribution, and importantly, we can estimate the marginal likelihood in the process. This is typically not easily accessible with MCMC, since it requires integrating over all parameters θ (Equation (5.14)).

As described in Section 2.6, the main idea of SMC is to transition from one distribution to another, via a sequence of intermediate probability distributions (Doucet et al., 2001). One approach to do this is to start with the prior distribution and transition to the posterior distribution for each model k. Alternatively, we can transition between the posterior distribution of each model k, using transformation SMC, discussed in Section 2.6.4. This will make use of the same concepts as RJMCMC, where we transform from one space to another.

5.4.1 Within model SMC

First, I will outline the SMC algorithm designed on the common space of model k (Algorithm 4) (Del Moral et al., 2006). This involves transitioning from the prior distribution, π_0 to the posterior distribution, π, for a given model k. At each transition, the algorithm targets

$$\pi_t(\theta) = (\pi(\theta))^{\gamma_t}(\pi_0(\theta))^{1-\gamma_t} \tag{5.49}$$

where the prior π_0 is given by Equations (5.6)-(5.7) and the posterior is the target distribution, reached at the end of the algorithm. We implement this by selecting γ_t adaptively at each step t (Section 2.6.3). Unlike RJMCMC, we can estimate the normalising constant, $\hat{Z}$ with the SMC algorithm. In this chapter, we will apply this algorithm for each model k independently, to estimate $\hat{Z}_k$ for each value of k. This is outlined in Algorithm 12.

5.4.2 Transformation SMC

In Section 2.6.4 we presented the method of transformation SMC (tSMC), first described in this way in Everitt et al. (2020), which allows us to transition from one distribution, π_k to another distribution that lies in a different space, $\pi_{k'}$ for $k' = k+1$ or $k' = k-1$. This is possible with ideas from RJMCMC, in which we transform the variables $(k, \theta^{(k)}, u^{(k)})$ into the new space $(k', \theta^{(k')}, u^{(k')})$. This involves drawing auxiliary variables $u^{(k)}$ and $u^{(k')}$ in order to meet dimension matching criteria during the reweighting step. We start with samples from π_k, which can be obtained either through the Within model SMC sampler described above or through a previous iteration of the tSMC sampler. We then target intermediate distributions, $\pi_{k\to k'}$, given by from Equation (2.90), at each transition t, using Metropolis within Gibbs sampling moves (Algorithm 9). γ_t is determined adaptively, at each step t. This process is detailed in Algorithm 13, for

Algorithm 12 Within model SMC sampler for factor analysis with fixed k.

```
1. For all particles, $p = 1, \cdots, N_P$, initialise $\theta_0^{(p)} \sim \pi_0$ from Equations (5.6)-(5.7), and $W_0^{(p)} = \frac{1}{N_P}$
2. Set $\gamma_t = 0$ and $t = 0$
3. While $\gamma_t < 1$:
    (a) Reweight all particles $p = 1, \cdots, N_P$ with
            $w_t^{(p)} = \tilde{w}_t^{(p)} w_{t-1}^{(p)} = \frac{\pi_t(\theta_{t-1}^{(p)})}{\pi_{t-1}(\theta_{t-1}^{(p)})} w_{t-1}^{(p)}$
    (b) Resample if necessary
        i. Renormalise all particles $p = 1, \cdots, N_P$ with
            $W_t^{(p)} = \frac{w_t^{(p)}}{\sum_{p=1}^{N_P} w_t^{(p)}}$
        ii. Calculate ESS with Equation (2.83)
            $ESS = \frac{1}{\sum_{p=1}^{N_P} (W_t^{(p)})^2}$
        iii. If $ESS < \alpha$ resample with stratified sampling and set $W_t^{(p)} = \frac{1}{N_P}$
    (c) Move all particles $p = 1, \cdots, N_P$
            $\theta_t^{(p)} \sim K_t(\cdot, \theta_{t-1})$
        where $K_t$ is a Metropolis-within-Gibbs move that targets $\pi_t$, with same proposal distribution outlined in Algorithm 9 but with acceptance rate
            $r_{MWG,t} = \left( \frac{\pi(\theta)}{\pi_0(\theta)} \right)^{\gamma_t - \gamma_{t-1}} \frac{q(\theta|\theta')}{q(\theta'|\theta)}$    (5.50)
    (d) Calculate next $\gamma_t \in (0, 1]$ such that $CESS(\gamma_t) = \beta N_P$ and set $t = t + 1$.
```

tSMC that transitions from k to $k+1$. The reverse process can also be implemented from k to $k-1$.

Algorithm 13 tSMC sampler for factor analysis increasing in dimension from k to $k+1$, with adaptive intermediate distributions.

```
1. Start with particles, $p = 1, \cdots, N_P$, $(\theta_0^{(k)}, u_0^{(k)})^{(p)} \sim \pi_k$ from previous SMC iteration and $W_0^{(p)} = \frac{1}{N_P}$
2. Transform all particles $p = 1, \cdots, N_P$, transform $\left(\theta_0^{(k)}, u_0^{(k)}\right)^{(p)}$ into $k+1$ model space, with split move in Equation (5.23) to obtain $\left(\theta_0^{(k+1)}, u_0^{(k+1)}\right)^{(p)}$.
3. Set $\gamma_t = 0$ and $t = 0$
4. While $\gamma_t < 1$:
    (a) Reweight all particles $p = 1, \cdots, N_P$ with
            $w_t^{(p)} = \widetilde{w}_t^{(p)} w_{t-1}^{(p)}$
        where $\widetilde{w}_t^{(p)}$ is given by Equation (2.89).
    (b) Resample if necessary
        i. Renormalise all particles $p = 1, \cdots, N_P$ with
            $W_t^{(p)} = \frac{w_t^{(p)}}{\sum_{p=1}^{N_P} w_t^{(p)}}$
        ii. Calculate ESS with Equation (2.83)
            $ESS = \frac{1}{\sum_{p=1}^{N_P} (W_t^{(p)})^2}$
        iii. If $ESS < \alpha$ resample with stratified sampling and set $W_t^{(p)} = \frac{1}{N_P}$
    (c) Move all particles $p = 1, \cdots, N_P$
            $(\theta_t^{(k+1)}, u_t^{(k+1)})^{(p)} \sim K_t((\theta_{t-1}^{(k+1)}, u_{t-1}^{(k+1)})^{(p)}, \cdot)$
        where $K_t$ is a Metropolis within Gibbs move that targets $\pi_{k \to k+1;t}$ with Algorithm 9.
    (d) Calculate next $\gamma_t \in (0, 1]$ such that $CESS(\gamma_t) = \beta N_P$ and set $t = t + 1$.
```

In this section, I will describe the results of the various SMC algorithms to determine the number of factors. This includes both the SMC run within model k (Algorithm 12) and tSMC (Algorithm 13) with transforms moving up and down in dimension. Firstly, we will use the same split/merge transforms presented in Section 5.3.1. We will also introduce two new proposals

(Medina-Aguayo et al., 2020). To move down in dimension, rather than merging the last two columns of $\mathbf{\Lambda}$, we could remove the last column of $\mathbf{\Lambda}$ entirely, while keeping the remaining columns the same, i.e.

$$\lambda_{i,j}^{(k')} = \begin{cases} \lambda_{i,j}^{(k)} & j < k \end{cases} \tag{5.51}$$

We will call this the *death* transform as the last column is removed entirely.

The equivalent move up in dimension would be to keep the same matrix $\mathbf{\Lambda}$ but with an additional column, drawn from the prior, i.e.

$$\lambda_{i,j}^{(k')} = \begin{cases} \lambda_{i,j}^{(k)} & j < k \\ \\ \lambda_{i,j}^{(k)} \end{cases} \tag{5.52}$$

We will call this the *birth* transform as the last column is drawn from the prior. We would not expect the death transform to necessarily perform as well as the merge transform presented in 5.2, which was designed specifically to summarise an over-parameterised factor analysis model. However, moving up in dimension has so far proved to be difficult due to the multiple new auxiliary variables that must be drawn appropriately. The birth transform draws new values directly from the prior, as these should always cover the possible space of the prior.

To summarise, all 5 algorithms to be compared in the following section, are as follows:

- **Within SMC**: starting from the prior distribution for model k and run to the posterior for model k, run independently for each model k.
- **Split SMC (tSMC increasing dimension)**: starting from the $k = 1$ model found from the Within SMC and using the split move to find the posterior for model $k = 2, \cdots, K$
- **Birth SMC (tSMC increasing dimension)**: starting from the $k = 1$ model found from the Within SMC and using the birth move to find the posterior for model $k = 2, \cdots, K$
- **Merge SMC (tSMC decreasing dimension)**: starting from the $k = K$ model found from the Within SMC and using the merge move to find the posterior for model $k = K - 1, \cdots, 1$

- **Death SMC (tSMC decreasing dimension)**: starting from the $k = K$ model found from the Within SMC and using the death move to find the posterior for model $k = K-1, \cdots, 1$

5.4.3 Example: 3 Factor Model

Using same dataset simulated from a 3 factor model as described previously, each of the SMC variations has been run, with 20 independent simulations, seeded with a different value. For the following results, I have set $N_P = 1000$ particles and parameters $\beta = 0.9$ and $\alpha = 0.5$ to control the CESS and ESS respectively. First, we compare the log marginal likelihood for each model in figure 5.18, where the distribution indicates the spread across different independent simulations with different seeds (plotted in the same way as Figure 2a of (Everitt et al., 2018)).

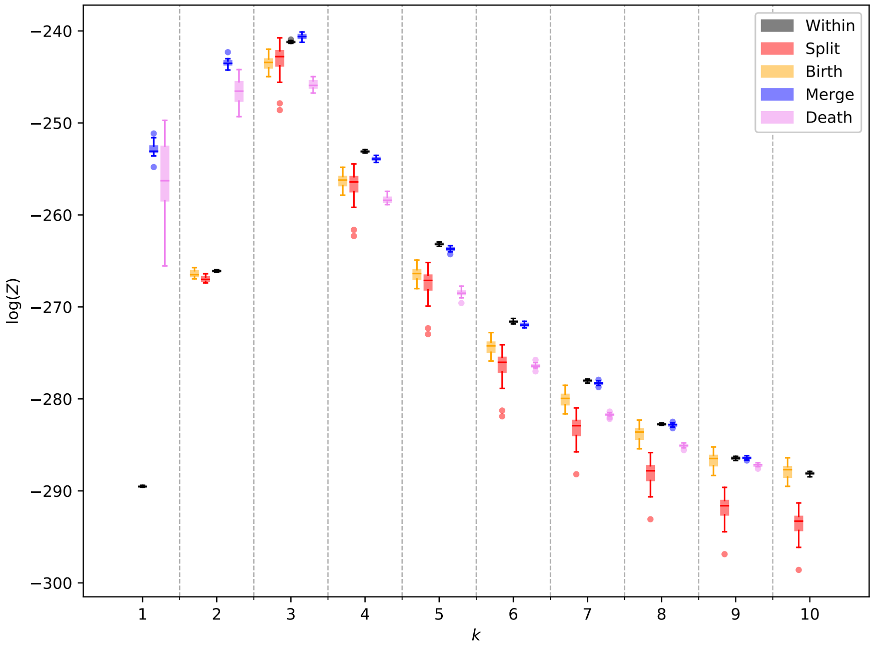


Figure 5.18: Log marginal likelihood estimated by all SMC algorithms with $N_P = 1000$.

Figure 5.18 determines $k = 3$ as the model with greatest marginal likelihood, in all 5 variations of the SMC algorithm. The within model SMC algorithm, which is run independently for each k starting with samples from the prior, has the lowest variance between the independent runs, making it robust against different initialisations. However, it is the most expensive of the algorithms, as each possible model k requires an independent simulation and no information is shared between different k. The relative cost of this and other algorithms is compared in Section 5.4.6.

The tSMC algorithms that increase in dimension (split and birth) start from the within SMC estimate of the $k = 1$ posterior distribution. Figure 5.18 show that beyond the peak at $k = 3$, these algorithms to estimate the log marginal likelihood significantly lower than the within model estimates. Along with this, at $k = 3$ there is an increase in the variance across the different seeded simulations. These factors are an indication of a poor quality SMC (Everitt et al., 2018). If there are only a few particles with high weights but many with low weights, the normalising constant (Equation (2.32)) would be underestimated and there would be a large variance on this calculation as seen here. This could arise if the proposal distribution does not cover the full space of the target distribution. The sudden increase in variance appears during the transition from $k = 2$ to $k = 3$, where we know that the transform does not provide the best proposal distribution (Figure 5.9). For the split algorithm, these large variances and underestimations persist and worsen as k increases, whereas the birth algorithm shows accurate estimates are recovered as k increases, suggesting it to make a better proposal distribution than the split transform.

The SMC algorithms that decrease in dimension (merge and death) start at the $k = 10$ posterior estimated from the within model algorithm. The merge SMC algorithm performs well when in the over-parameterised regime, $k \geq 3$, with marginal likelihoods close to the within SMC model estimates and low variances, in contrast to the death SMC algorithm which consistently underestimates the marginal likelihood. However, the performance of both algorithms degrade rapidly in the under-parameterised regime for $k = 2$ and $k = 1$, where the marginal likelihoods are overestimated and with high variance (particularly for the death SMC algorithm). As noted in Figure 5.3, the merge proposal distribution is too narrow to capture the true distribution, which would lead to an overestimation of the weights and therefore an overestimated marginal likelihood. The high variance also confirms this to be a poor quality SMC proposals in this regime.

5.4.4 Parameter Distributions

Figure 5.19 compares the distributions obtained for $\theta^{(k)}$ for the true number of factors, $k = 3$, for all SMC algorithms. These can also be compared to the Gibbs sampler results in Figure 5.2.

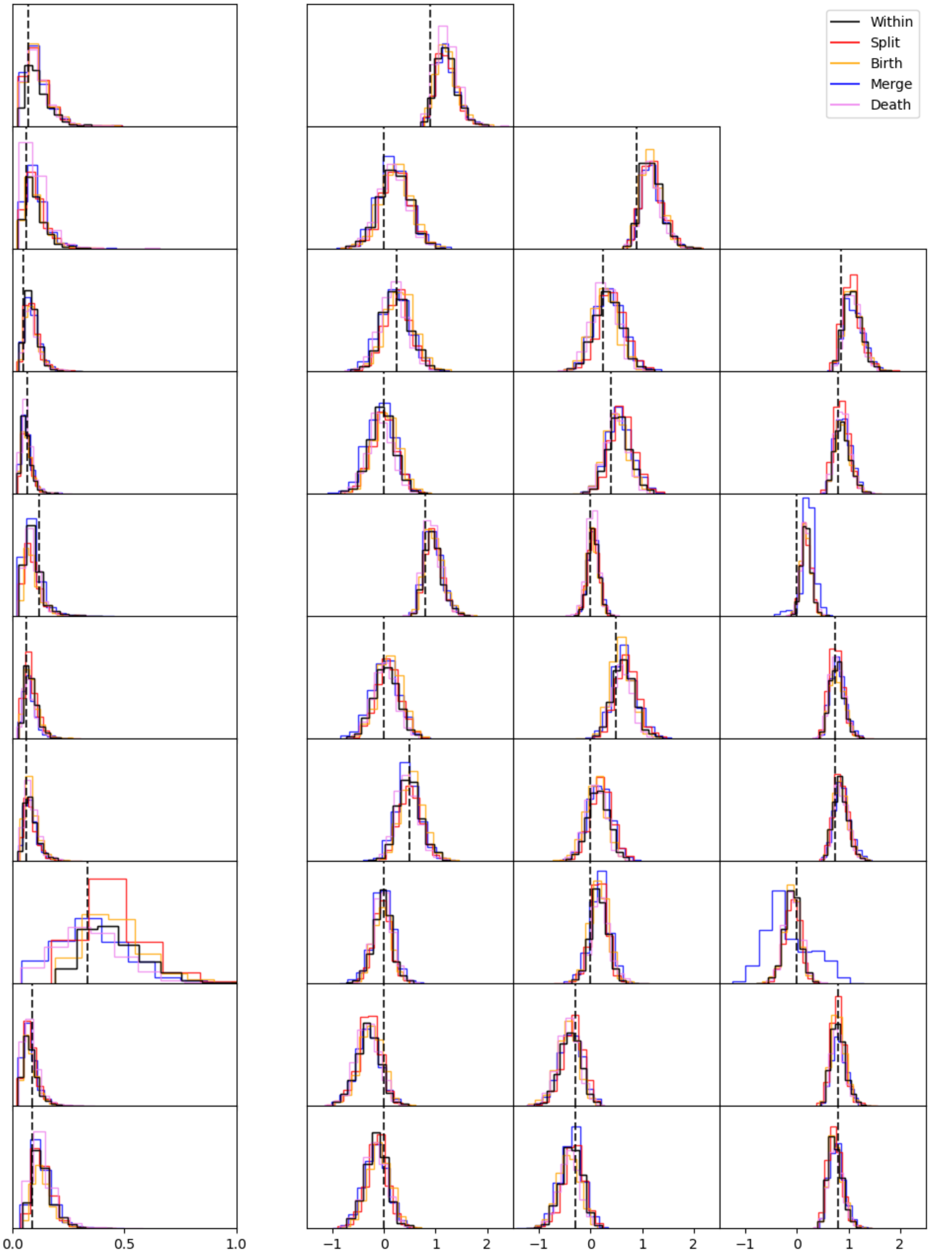
All 5 SMC algorithms produce almost identical distributions centred over the true value of λ_{ij} and σ_j. The exception to this is in the 8th row, which has a particularly large σ_8, designed to test the methods under large variances. The different algorithms give slightly different distributions on σ_8, all of which are quite wide but still centre on the true value. Also, the merge SMC algorithm estimates a wider distribution of $\lambda_{8,3}$ compared to the other methods. A similar result is obtained on row 5, although to a lesser degree, where all methods slightly underestimate σ_5 and overestimate in $\lambda_{5,3}$. However, this is a significant improvement upon the merge transform alone, presented in Figure 5.2 for $k = 4 \to 3$ and $k = 10 \to ... \to 3$.

In general, all SMC algorithms are able to reproduce distributions that are almost identical to the Gibbs sampler, as well as correctly identifying $k = 3$ as the underlying model. Next, we examine in detail different factors that play into the relative performance of these methods, including number of particles, the variance and computational cost.

5.4.5 Number of particles

Figure 5.20 shows the log marginal likelihood for all methods when the number of particles is set at (a) $N_P = 10$ and (b) $N_P = 100$. As the number of particles increase, we expect the estimates of the log marginal likelihoods to become more accurate and the variance between different simulations to decrease. With only 10 particles, there are inconsistencies and high variances in $\log(Z)$, with particularly large errors introduced in the tSMC algorithms. However, with 100 particles, the results are fairly consistent between different seeds and different algorithms, giving almost identical results to setting $N_P = 1000$ as in Figure 5.18.

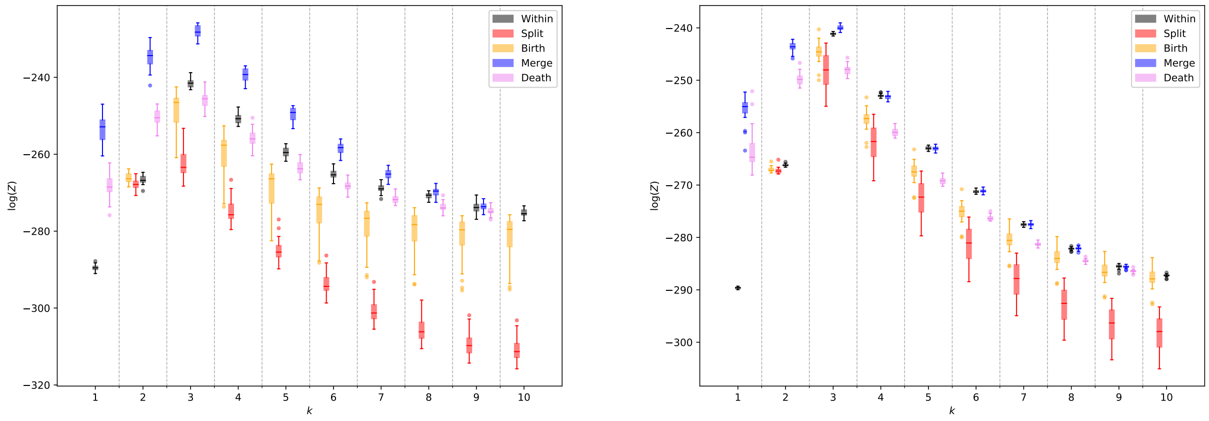
By taking the $\log(Z)$ estimates from the Within SMC with $N_P = 1000$ as the ‘ground truth’, we can directly compare the errors introduced with the transforms and with $N_P < 1000$. Figure



(a) $\boldsymbol{\Sigma}$

(b) $\boldsymbol{\Lambda}$

Figure 5.19: Histogram of samples from $k = 3$ model generated by all SMC algorithms, $N_P = 1000$. The black dashed line shows the true values of which the data is simulated from.



(a) $N_P = 10$

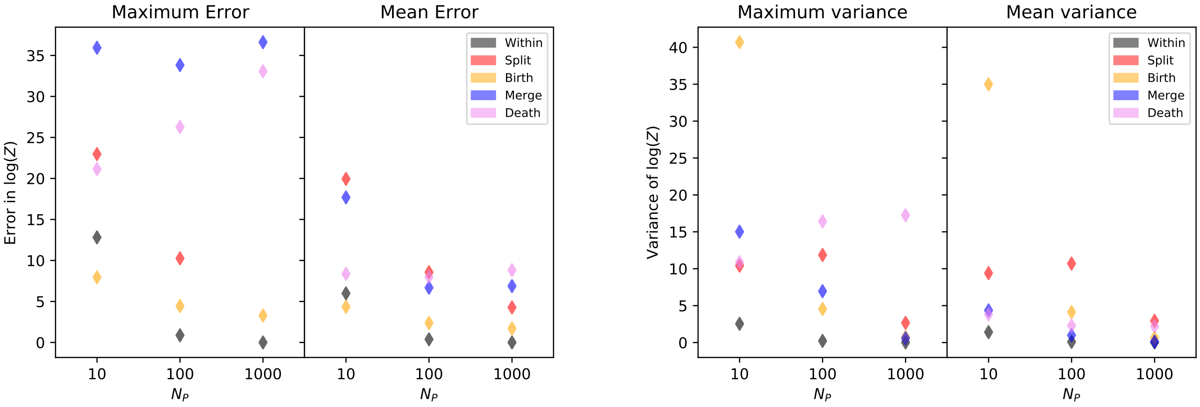
(b) $N_P = 100$

Figure 5.20: Log marginal likelihood estimated by all SMC algorithms with (a) $N_P = 10$ and (b) $N_P = 100$.

5.21a shows the maximum absolute error introduced and and the mean absolute error across all k and all seeds. Here we see the benefit in increasing N_P from 10 to 100, but there is little additional benefit in further increasing this to 1000 for the Within SMC algorithm and for the transforms that move up in dimension. The birth move in particular shows the smallest errors in Figure 5.21a.

Ideally, the algorithm that would be favoured would also achieve a low variance on $\log(Z)$ between different independent simulations. Figure 5.21b shows the maximum and mean variances achieved for these different algorithms with different N_P. This shows significant improvements with increasing N_P in terms of the mean variance and (excluding the death transform) the maximum variance. The death transform appears to maintain a large maximum variance across differently seeded simulations regardless of N_P, which is introduced in the under-parameterised, as seen in $k = 2$ and $k = 1$ in Figure 5.18 and 5.20.

Based on both the errors introduced and the variance, the Within SMC algorithm with only 100 particles appears sufficient, whereas to achieve this low variance with the other tSMC algorithms, more particles are needed. Of the tSMC algorithms, the most reliable is the Birth SMC approach, which maintains fairly low errors and low variances even with $N_P = 100$. In the next section, we will explore the computational costs of these algorithms, which may influence the algorithm selected by a user with a fixed computational budget.



(a) Errors (b) Variances

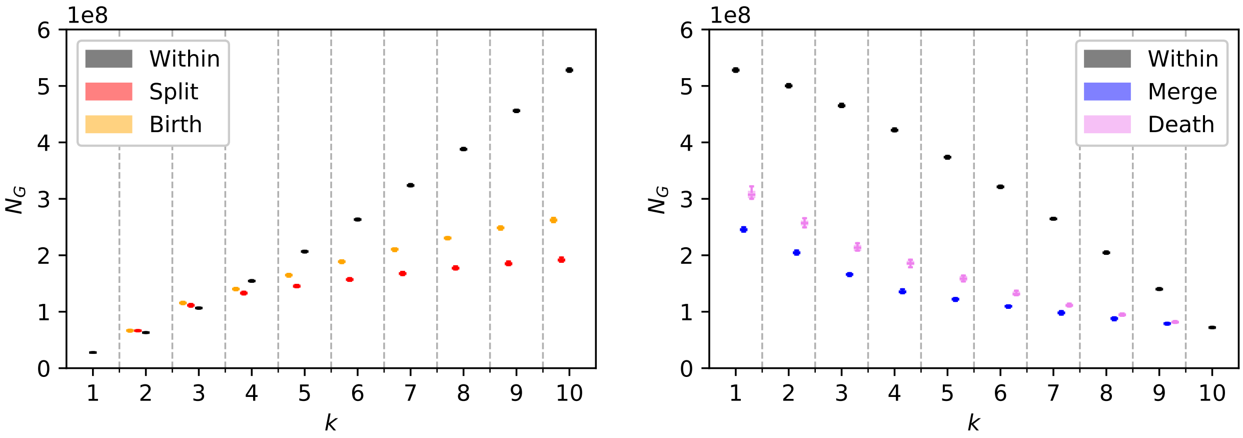
Figure 5.21: Maximum and mean (a) errors of each SMC algorithm relative to the within model SMC with $N_P = 1000$ and (b) variances across independent SMC simulations initialised with different seeds.

5.4.6 Performance in terms of cost

To assess computational cost independent of the machine the algorithm is run on, we compare the number of Gaussian evaluations required to complete the algorithm for all models. This is done by cumulatively summing the number of evaluations for each model, from $k = 1$ to $k = 10$ for the Split/Birth algorithms and from $k = 10$ to $k = 1$ for the Merge/Death algorithms. This calculation requires the number of intermediate steps for each simulation, N_T, the number of particles, N_P, and the number of times the likelihood is evaluated at each step for each particle. The likelihood must be evaluated twice in the reweighting plus the number of times it is required in the Monte Carlo move step (m times in the update for $\mathbf{\Sigma}$ and k times in the update for $\mathbf{\Lambda}_k$).

$$N_G = N_T N_P (2 + (m + k) N_{\text{MCMC}}) \tag{5.53}$$

Note that the factor $(2 + (m + k)N_{\text{MCMC}})$ is roughly constant across models for when potential values for k do not differ greatly and N_{MCMC} is large, meaning the number of evaluations is roughly proportional to the number of intermediate steps. Figure 5.22 shows the number of evaluations needed to reach each model k, where (a) shows the Split/Birth algorithms which start in model $k = 1$ and increase in dimension and (b) shows the Merge/Death algorithms which start in $k = 10$ and decrease in dimension. Both approaches also require the cost of the within model SMC for the first step. We also include the cumulative sum of the number of evaluations for the Within SMC, either increasing or decreasing in dimension. From this, we can see the that the total cost is significantly lower when using tSMC, although note that the within model SMC algorithm is independent for each k and can therefore be processed in parallel if resources are available and time is the limiting factor rather than total cost.



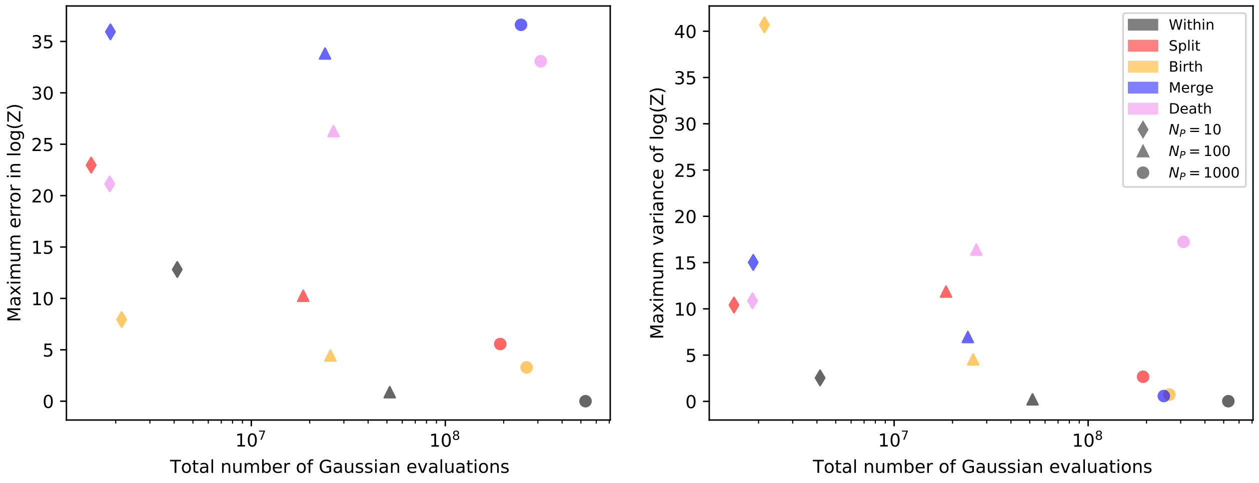
(a) Increasing in dimension

(b) Decreasing in dimension

Figure 5.22: Number of Gaussian Evaluations required to cover entire parameter space (a) starting in model $k = 1$ and increasing in dimension and (b) starting in model $k = 10$ and decreasing in dimension.

Taking the maximum error (measured as the difference between the Within SMC ($N_P = 1000$) algorithm) and the maximum variance as measures of accuracy, we compare the computational cost of running these algorithms to completion in terms of number of Gaussian evaluations in Figure 5.23. Increasing N_P increases the cost linearly, but if given a set computational budget, we can achieve as low as a variance as possible by opting for an algorithm that occupies the lower left-hand corner of both plots. Due to the high maximum errors introduced in the $N_P = 10$ algorithms, we cannot justify these algorithms, even though some achieve low variances. However, the Within SMC and Birth or Split SMC algorithms with $N_P = 100$ would be preferred, based on achieving the lowest error at the lowest relative cost.

Based on the above results, setting $N_P = 100$ appears to be sufficient for achieving reliable results, especially for the Within SMC and the transforms that move up in dimension (Split/Birth). Figure 5.24 shows the total cost of running these algorithms with $N_P = 100$. For reference, we



(a) Maximum error in $\log(Z)$ against cost (b) Maximum variance of $\log(Z)$ against cost

Figure 5.23: (a) Maximum error calculated as the difference relative to the $N_P = 1000$ Within SMC algorithm for each model k and (b) Maximum variance of $\log(Z)$ for each k against total number of Gaussian evaluations required as a measure of computational cost.

also show the total cost of running pmRJ, including all preliminary MCMC runs and RJ with $N_T = 50$ and $N_P = 50$ for 10,000 iterations (previously shown in Figure 5.15.) Although slightly more costly to complete a full sweep over the parameter space with an SMC method, we have found SMC to provide more informative results without the risk of experiencing convergence or mixing issues. Furthermore, these costs could be significantly reduced by introducing parallelisation in the reweighting and move steps of Algorithms 12 and 13. The dashed line shows theoretical computational savings of more than an order of magnitude if run on 24 cores based on the relative time spent in the two parallelisable steps of the algorithms. Finally, the Within SMC relies on independent simulations for each value of k, which means that these can be completely parallelised on separate nodes, for $k = 1, \cdots, m$. As this algorithm also gave the results with the lowest variance and that it appears most consistent with the more expensive simulations run with $N_P = 1000$, this appears to be the algorithm of choice for gaining an accurate picture of the marginal likelihoods across different values of k.

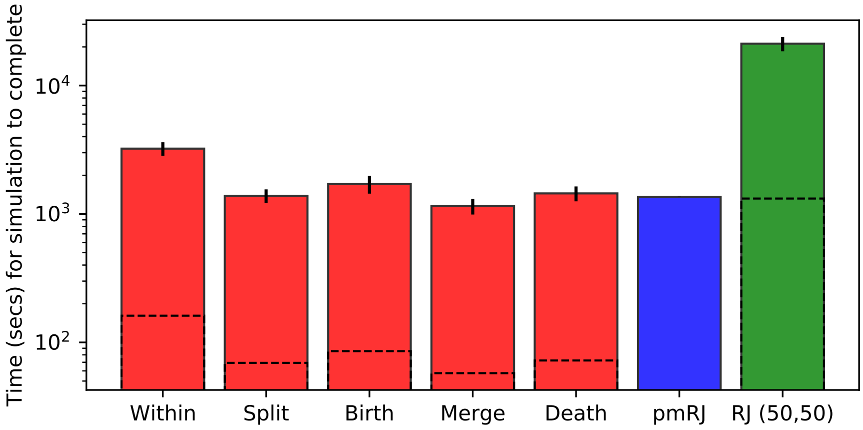


Figure 5.24: Computational cost of different SMC methods measured as the total time for the simulation to complete. All methods are run with $N_P = 1000$ and the uncertainty bars estimate 1 standard deviation across 20 independent simulations run. The dashed line indicates theoretical cost that could be achieved if parallelised across 24 cores. Also shown are the times to run the pmRJ algorithm and RJ with $N_T = 50$ and $N_P = 50$ up to 10000 iterations from Figure 5.15.

5.4.7 Example 2: 7 Factor Model

We explore how well the methods perform on a higher dimensional simulated dataset, with 7 underlying factors and 14 total dimensions, for a dataset of 100 samples ($k = 7, m = 14, N_{\text{obs}} = 100$). Fig 5.25 shows the marginal log likelihoods for all 5 SMC methods, where again, $N_P = 1000$. Most notable is the poor performance of the merge and death algorithms in the under-parameterised region, $k < 6$, where they both overestimate the log marginal likelihoods and give the best number of factors to be around $k = 1$ (death) or $k = 4$ (merge), the latter being in agreement with the RJMCMC algorithms in Section 5.3.7. They also show very high variances between different SMC runs, which is not ideal. As observed before, the split and birth algorithms systematically underestimate the log marginal likelihood in the over-parameterised regime. Overall, however, the split and birth algorithms appear preferable to their merge and death counterparts, because they have significantly lower variances and show a posterior distribution that is closer to the Within SMC estimates. These methods select the number of k to be the same as the within SMC model ($k = 6$), which is fairly close to true underlying number of factors and a substantial improvement upon the RJMCMC methods, which selected $k = 2$ or $k = 4$. Furthermore, both RJMCMC algorithms methods experienced poor mixing, which made it difficult to establish whether or not convergence to the true posterior distribution has been achieved. From this perspective, within SMC method or the tSMC methods that move up in dimension give much more reliable results.

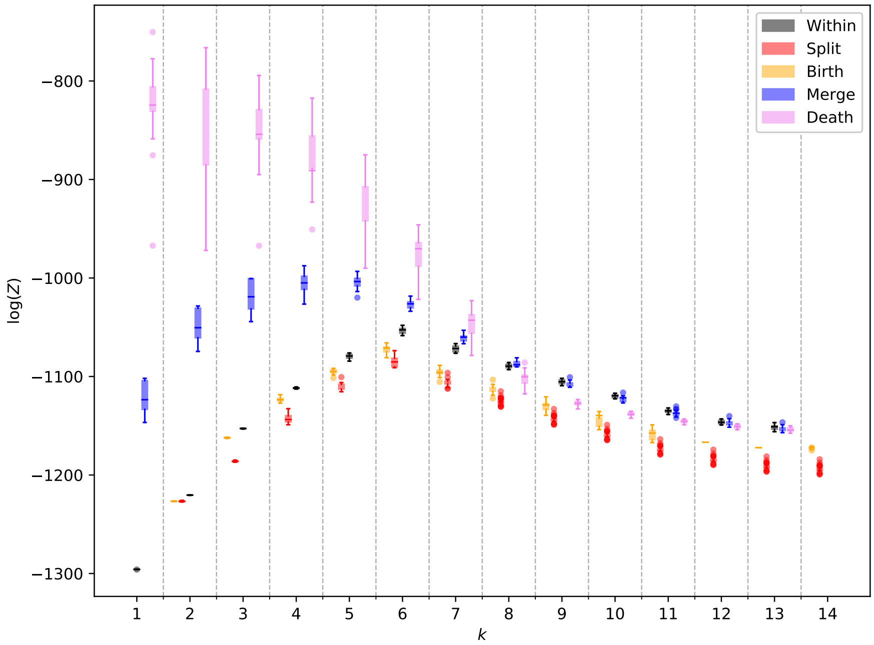


Figure 5.25: Log marginal likelihood estimated by all SMC algorithms with $N_P = 1000$ on simulated dataset with 7 factors.

5.5 Application to Weather Dataset

In this section, I will apply these methods to a weather dataset introduced by Ramsay and Silvermann (1998), that contains temperature observations from 35 different weather stations located across Canada ($m = 35$) (Ramsay et al., 2014). There are 365 observations of daily average temperature ($N = 365$). The data is standardised to remove the annual fluctuations and keep the temperature anomalies relative to the average yearly temperature across all weather stations.

Based on the comparison between RJMCMC and SMC so far, we have found that SMC algorithms tend to perform better, with the RJMCMC algorithms frequently underestimating the total number of factors. In particular, the RJMCMC algorithms in this setting appear to favour the model with $k = 2$, with little exploration of the complete parameter space. An example trace plot is shown for $N_T = 50$, $N_P = 50$ in Figure 5.26.

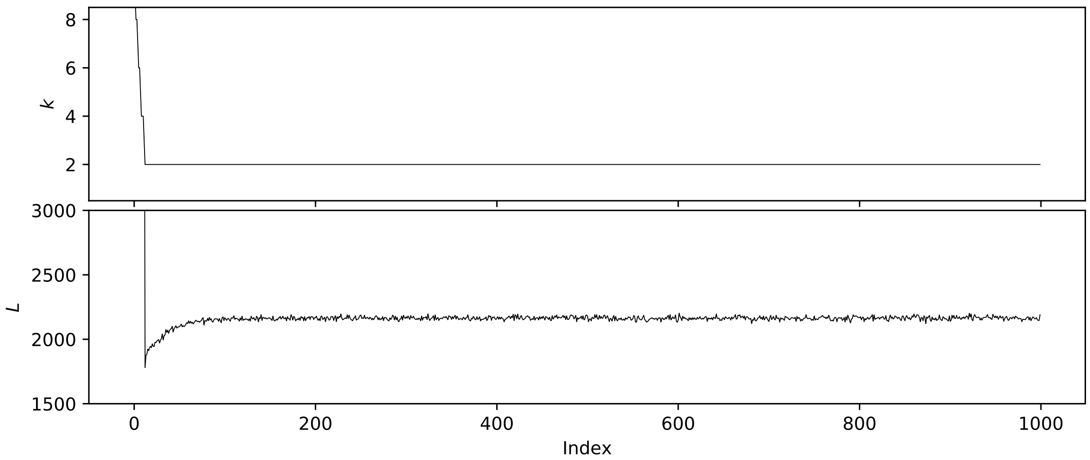


Figure 5.26: Trace plots of (top panel) k and (bottom panel) the log likelihood for the first 1000 iterations of the RJ algorithm with $N_P = 50$ and $N_T = 50$ applied to the observational weather dataset.

For the remainder of this section, we will rely on the SMC methods, which have so far proven to be more suitable as they guarantee an exploration of all possible k values and provide an estimation of the normalising constants. We also found these to computationally more efficient, when setting $N_P = 100$. Furthermore, a tSMC algorithm can be used starting from $k = 1$ and increasing in dimension, until the peak maximum likelihood has passed. This can be used to determine the best value of k, without running the full algorithm. Figure 5.27 shows the resulting log marginal likelihoods for the split and birth tSMC algorithms run with N_100 alongside the within SMC algorithm, which we expect to be the most accurate choice. Using this method, with either the split or birth algorithm, we find the peak marginal likelihood to be $k = 6$, beyond which $\hat{Z}$ starts decreasing. In this observational dataset, the Split SMC algorithm appears to give better a estimation of $\hat{Z}$ for large k, with lower variances and closer to the Within SMC algorithm.

5.5.1 Latent factors

The results of the SMC algorithm suggest that there are $k = 6$ underlying factors in this dataset, that explain the behaviour of the 35 different weather stations. The factors, $\boldsymbol{\eta}^{(6)}$ represent features of the weather stations over the observed time series, relative to the mean behaviour

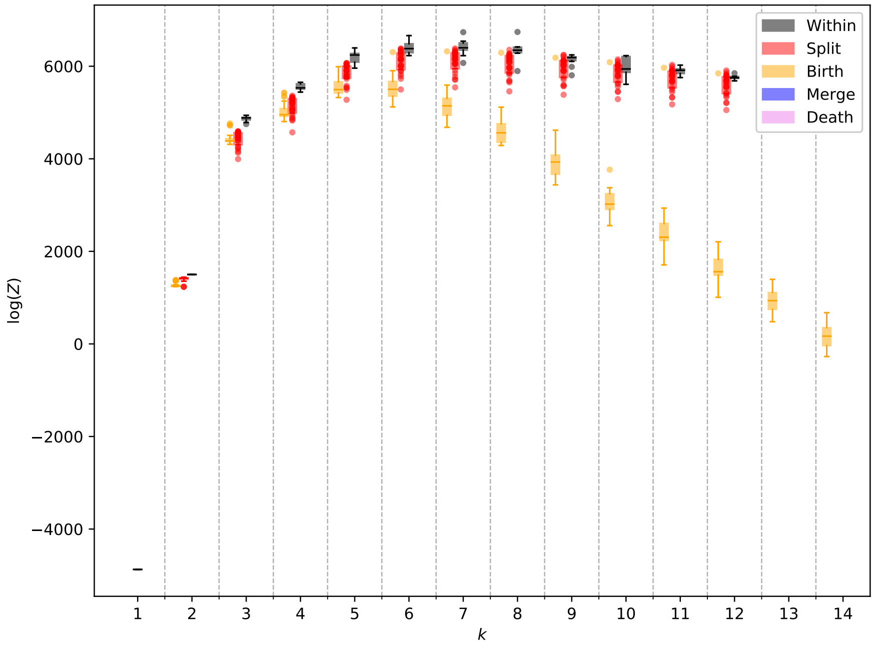
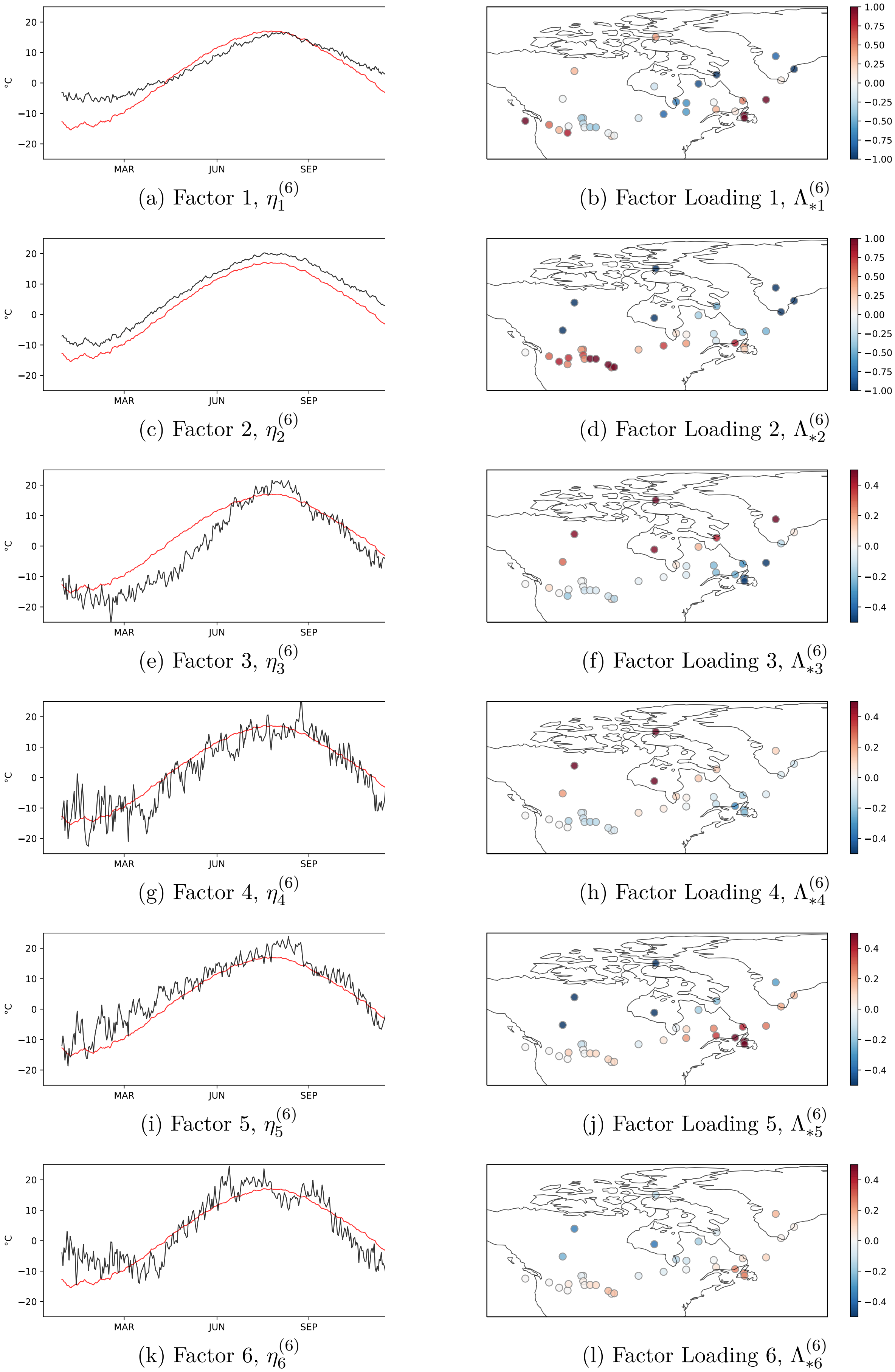


Figure 5.27: Log marginal likelihood estimated by within SMC, birth SMC and split SMC algorithms with $N_P = 100$ on observational weather dataset.

across all stations (as the dataset is normalised by the mean). The factor loading matrix, $\mathbf{\Lambda}^{(6)}$ represents the relationship between the factors and the dataset, which informs us of how each weather station is comprised of the 6 factors. The left-hand panel of Figure 5.28 shows the 6 factors across the time series, $\eta_i^{(6)}$ for $i \in 1, \cdots, 6$, in black, as well as the mean behaviour in red. The right-hand panel shows each column of the factor loading matrix (i.e. following the row-column notation used throughout, the first column, $\Lambda_{*1}^{(6)}$ describes how the data relates to Factor 1). These are shown for on a map that shows the spatial location of each weather stations, with the colour of each point representing the strength of the factor for each station. This allows us to make sense of the factors in relation to their geography. We can also compare against the results of Ramsay and Silvermann (1998) where PCA is applied to the monthly mean of the dataset, assuming a known number of components. There, four principal components are chosen in order to represent the four key regions of Canada that influence the weather: the Atlantic effect, the Pacific effect, the Continental effect and the Arctic effect (Ramsay, 2003).

Figure 5.28a shows that the first factor represents a warmer than average winter. From Figure 5.28b, this effect appears strong in coastal regions, near the Pacific and Atlantic oceans. This effect is identical to the Pacific effect found in Ramsay and Silvermann (1998) defined as a



(a) Factor 1, $\eta_1^{(6)}$

(b) Factor Loading 1, $\Lambda_{*1}^{(6)}$

(c) Factor 2, $\eta_2^{(6)}$

(d) Factor Loading 2, $\Lambda_{*2}^{(6)}$

(e) Factor 3, $\eta_3^{(6)}$

(f) Factor Loading 3, $\Lambda_{*3}^{(6)}$

(g) Factor 4, $\eta_4^{(6)}$

(h) Factor Loading 4, $\Lambda_{*4}^{(6)}$

(i) Factor 5, $\eta_5^{(6)}$

(j) Factor Loading 5, $\Lambda_{*5}^{(6)}$

(k) Factor 6, $\eta_6^{(6)}$

(l) Factor Loading 6, $\Lambda_{*6}^{(6)}$

Figure 5.28: Left hand panel shows the factors in black, compared against the annual mean temperature in red. Right hand panel shows the associated column of the factor loading matrix, the colour indicating the strength of the factor at each weather station. Estimated by the within model SMC with $k = 6$ and $N_P = 1000$.

summer temperature close to the Canadian average, but much warmer in the winter. There is also a cluster of weather stations over the North-East, surrounding the Hudson Bay, which are negatively correlated to this factor, indicating a colder winter relative to the average. This is indicative of the Continental effect defined by Ramsay and Silvermann (1998) as colder than the average in the winter and only slightly warmer than average in the summer.

The second factor, Figure 5.28c, indicates systematically warmer temperature all year round and highlights a clear divide between the North and South in Figure 5.28d, with the Southern weather stations having strong positive components while more Northern weather stations having strong negative components. This is particularly strong in the cluster of weather stations near the South-West on the Atlantic side of Canada. This is in agreement with the Atlantic effect in Ramsay and Silvermann (1998), found to be a nearly constant 5 °C across the year. In contrast, the Arctic regions appear to have strongly negative components of this factor, i.e. systematically cooler all year round, which is a signature of the Arctic effect found in Ramsay and Silvermann (1998) to be a colder than average all year found, even more so in March than December. This latter feature of the Arctic effect appears to be present in Factor 3 (Figure 5.28e) where there is a signicantly colder temperature in Spring months. This is also highly correlated to the Arctic regions in Figure 5.28f.

The remaining factors appear to represent higher frequency fluctuations in the temperature observations, roughly weekly to fortnightly for Factors 4 and 5 and monthly to seasonal for Factor 6. Weather stations close to each other are highly correlated in their relationships to these factors while distant weather stations are anti-correlated, for instance, Factor 5 shows negative factor loadings in the North West and positive factor loadings in the South East. This indicates that on short timescales these regions experience opposing effects as each other, as weather systems take time to pass from one region of the country to the other. Note that these short timescales are not assessed in the approach of Ramsay and Silvermann (1998), as the PCA is carried out over monthly means.

Figure 5.29 shows the values of Σ, the independent noise unique to each weather station. These are significantly higher in the Northern and Arctic regions, which are known to be generally

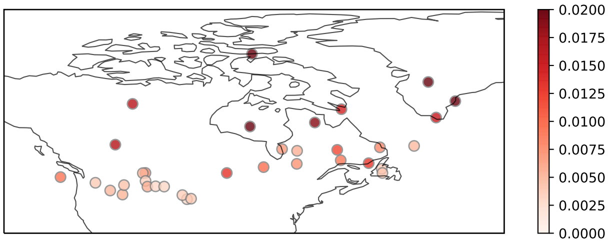


Figure 5.29: Unique variances $\Sigma_*^{(6)}$ for each weather station estimated by the within model SMC with $k = 6$ and $N_P = 1000$.

more variable compared to lower latitudes (e.g. as discussed based on climate model data in Chapter 3, Figure 3.10).

From this, we can work with the reduced vector of only now 6 dimensions instead of 35. After analysis, we can reconstruct the original 35 dimensions through Equation (5.3). Figure 5.30 shows a reconstruction of the factors to represent all 35 weather stations in black compared to the true data in red, which are almost in perfect agreement.

These results have shown a broad agreement with the dimension reduction method applied to monthly means in Ramsay and Silvermann (1998) where the number of principal components is fixed at four, based on expert knowledge of the four key regions. However, the Bayesian factor analysis approach used here does not assume any prior knowledge of the number of factors. This method finds that six factors give a complete representation of the data, going beyond these four assumed effects alone. As well as capturing mean effects, the factor analysis model is able to replicate high frequency fluctuations in the time series, including the different magnitudes of these fluctuations which tend to be larger in winter relative to the summer months.

5.6 Conclusions

This chapter explored various methods of Bayesian model selection, designed to estimate the number of latent factors in a dataset. These were tested on two simulated datasets and on one real observational dataset. Firstly, I presented a newly developed RJMCMC algorithm, based on the transform presented in Dunson (2006), to estimate the posterior probability distribution on

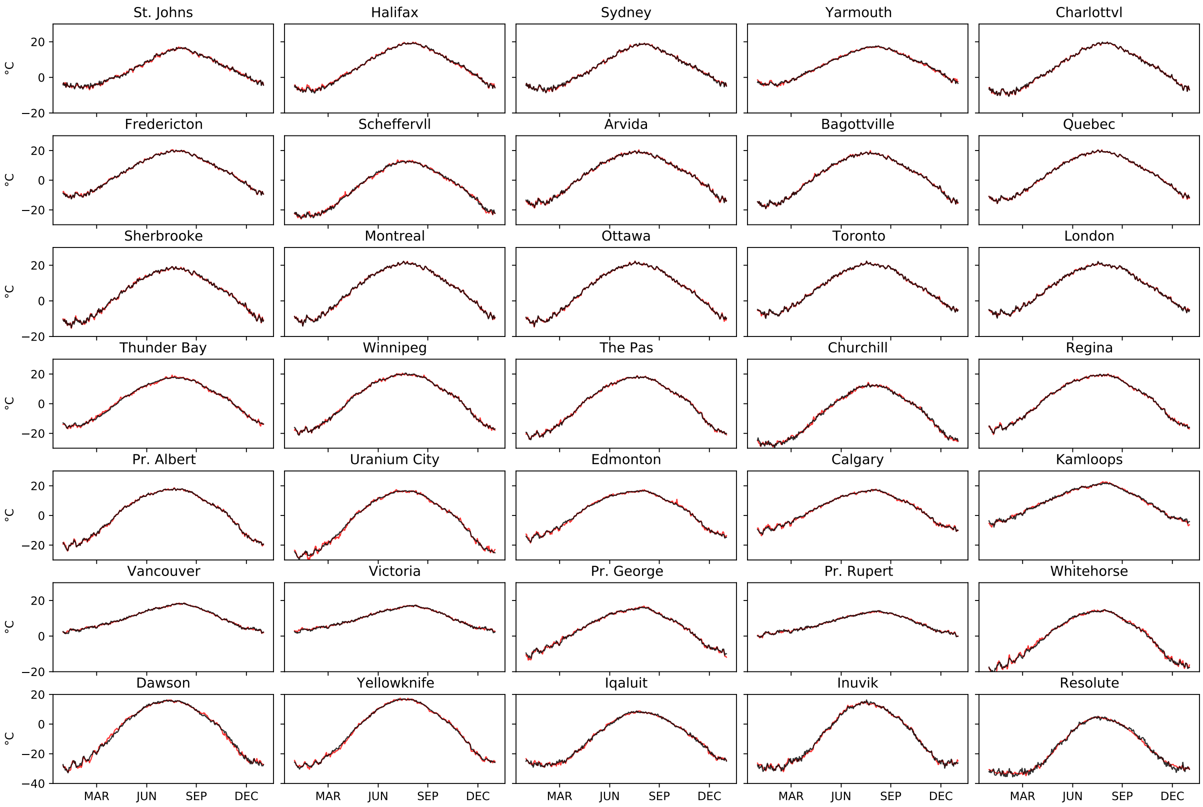


Figure 5.30: Construction of time series based on 6 factor model learned by Within SMC algorithm (black) compared against true time series for all stations (red).

the number of factors. Although much cheaper than the existing RJMCMC algorithm designed for the problem in Lopes and West (2004), this new algorithm was found to show poor mixing and strongly favoured moves that decrease the number of factors. However, improvements to this were seen when introducing additional steps to this algorithm, by adding intermediate distributions between the proposal and target ($N_T > 1$) and by evaluating multiple possible moves at each step ($N_P > 1$) (Karagiannis and Andrieu, 2013; Andrieu et al., 2020). These gave the RJMCMC better mixing and convergence properties at a cost comparable to the existing method of Lopes and West (2004) on a simulated dataset consisting of 10 variables and 3 underlying factors. This benefit is expected to increase with the size of the dataset, however, when applied to a larger simulated dataset of 14 variables and 7 underlying factors, the algorithm suffers from problems with mixing, even when N_P and N_T are large.

The second approach taken to determine the number of latent factors was SMC, which estimates the marginal likelihood directly. This approach is not only more accurate but it also does not suffer from poor mixing and convergence properties, making it significantly cheaper, especially if parallelisation is taken advantage of. An SMC simulation can be run independently within each model of a fixed number of factors, starting with samples from the prior distribution. Alternatively, transformation SMC can be run to move from a $k = 1$ factor model to the maximum number of factors, and vice-verse. Of the possible transforms explored, the transforms that progressively increase the number of factors performed much better than those that decrease the number of factors at each step. This was due to the suboptimal performance of the transforms when underparameterised, which can lead to the algorithm underestimating the optimal number of factors. The most accurate tSMC algorithm uses a birth transform to sample the new factors from the prior, at a lower cost than the within model SMC but achieving a similar accuracy. However, if many nodes are available, the within model SMC is more accurate and can be run independently and in parallel. Both choices can be run while monitoring the marginal likelihood until passing the maximum marginal likelihood value.

Finally, these algorithms were tested on a weather dataset, comprised of 35 different weather stations across Canada. The SMC algorithms suggest 6 as the optimal number of factors that describe the data with a factor analysis model, which is consistent with the geography and climatology of the region (Ramsay and Silvermann, 1998; Ramsay, 2003). This means the dataset could be reduced from 35 dimensions to just 6 dimensions, to carry out subsequent analysis, such as emulation. However, this is still a relatively small dataset compared to some climate data, such as the output of climate models which typically consist of 10000s of datapoints (e.g. the climate model dataset used in Chapter 3). Naturally the next step would involve applying these methods to even larger datasets. With this, we could envisage applying Bayesian factor analysis to spatiotemporal climate model output to uncover relationships between different climate variables, regions or time periods. Given the increasing costs involved with higher dimensional datasets, parallelisation of the algorithms (not done here) would be more beneficial, along with further exploration of the choice of N_P and limits on the ESS (α) and the CESS (β) to infer the number of factors with the most cost-effective approach.

Chapter 6

Conclusions

6.1 Summary of Thesis

This research sits in the context of exploring how human-induced emissions affect the global climate and addresses two key research topics: firstly, climate change projection via statistical methods and secondly, dimension reduction of large datasets that are typical of climate model output. The first of these is approached in two stages, one which predicts the short-term response of a global climate model (GCM) to a sudden emission perturbation and one which predicts the long-term response of a GCM given the short-term response. Both prediction tools use statistical models trained on GCM data.

In Chapter 3, the global pattern of long-term temperature response, averaged over years 70 to 100 of the GCM, is predicted from the global pattern of short-term temperature response, averaged over the first 10 years of the GCM. This chapter finds that generally the two machine learning methods explored (ridge regression and Gaussian process regression) hold greater predictive power relative to a traditional pattern scaling approach. This is particularly true when predicting responses to short-lived pollutant perturbations, which produce more spatially inhomogeneous responses compared to the long-lived GHGs. The developments made here could accelerate long-term climate change projections, requiring only the first 5-10 years of a GCM, or taking this even further, could be combined with the statistical emulator of Chapter 4, removing the need to run an expensive GCM entirely.

This second statistical emulator is built to predict the probability distributions of the first 5 years of the temperature response, given 9 different emission-related input parameters, describing both long-lived greenhouse gases and short-lived aerosol pollutants. The emulator performs well on the test dataset in all but one case where it overestimates the cooling associated with strong positive aerosol forcings. It appears here that the emulator does not capture the non-linearity in temperature response, where a stronger response is expected when new aerosols are added to clean air, compared to additional aerosols being added to an already aerosol-heavy atmosphere (Carslaw et al., 2013). To improve upon this, more strong aerosol perturbations should be added to the training dataset to provide a better picture of the behaviour of the GCM under larger aerosol perturbations.

A recurring theme of the first two results chapters was the high dimensionality of the datasets used, due to the high resolution of the GCM. Chapter 5 tackled dimension reduction with a Bayesian approach to determining the underlying number of factors required to describe a large dataset. An array of different Monte Carlo techniques were developed and applied to infer the probabilistic distribution on the number of factors and on the factors themselves. A reversible jump Markov chain Monte Carlo (RJMCMC) algorithm was developed (Green, 1995), based on previous studies in factor analysis that found a sensible approximation to the underlying factor model, given an over-parameterised version of it (Dunson, 2006). Recent techniques developed for RJMCMC were also incorporated to improve convergence and mixing properties (Andrieu et al., 2018; Karagiannis and Andrieu, 2013). Although this algorithm performed competitively compared against the existing RJMCMC for factor analysis (Lopes and West, 2004), both suffered from poor mixing when in high dimensions, making these approaches infeasible for infering the underlying factors of a large dataset. However, Sequential Monte Carlo (SMC) achieved much more promising results in this task (Del Moral et al., 2006). One of the key benefits of SMC is that it provides an estimate of the marginal likelihood which can be used to determine the number of factors of a dataset. Additionally, SMC can be parallelised which makes it a cheaper option than the RJMCMC methods explored.

6.2 Future Directions

This thesis has taken the first step towards building a surrogate model that emulates climate response to emission perturbations at a cheaper and faster rate than a complex GCM. Although both statistical models presented in Chapters 3 and 4 are useful standalone tools, a natural continuation of this would be to combine the two together. In doing so, the surrogate model could act as a rapid prediction tool for predictions of the long-term climate response to abrupt perturbations that then remains constant in time. This would be the first emulator that could predict the entire global pattern of long-term climate response from a range of scenarios. To achieve this, a surrogate model could be built that takes the emissions as inputs and first predicts the short-term response using the emulator in Chapter 4. The output of this would be used as inputs to the surrogate model in Chapter 3, to estimate the long-term climate response projection. For consistency, it would be sensible to select the Gaussian process option for the second surrogate model, which would also allow for propagation of the uncertainty from the short-term response to the long-term response (Girard et al., 2002; Damianou et al., 2016).

Before putting such a surrogate model into practise, it would, however, be important to carry out further testing on GCM simulations. In particular, there were some differences between the types of training simulations used in Chapters 3 and 4. For instance, Chapter 3 made use of simulations with only single forcing perturbations (i.e. one at a time perturbations), under the assumption that the type of forcing was realised in the first 10 years of the response. To check that this is valid, additional simulations with multiple forcing perturbations should be tested under this surrogate model. Further to this, a more recent version of HadGEM3 was used in the emulator presented in Chapter 4. Ideally, the same version of the model should be used to train both models, or if not, the performance of the surrogate model should be verified with the version one aims to emulate. Finally, the short-term response emulator in Chapter 4 was derived from 5 year simulations, while the timescale used for the short-term definition in Chapter 3 was predominantly 10 years, to reduce the effect of noise. It would therefore be logical to extend the emulator in Chapter 4 to estimating the 10-year averaged response, before coupling the two surrogate models.

Building a hierarchy of surrogate models has been carried out in previous studies, but it is more typical that these use simulations run on a simpler model, or with a lower spatial resolution, as an estimation for a more complex one (Tran et al., 2016; Cumming and Goldstein, 2009; Kennedy and O'Hagan, 2000). Instead, the methodology proposed here is built on the idea of taking the climate response at an early stage can be used to predict the response at a later stage. This becomes a 'dynamic emulator' where, rather than emulating a complete multi-step run of a complex model, the emulator is built to run a simpler, single timestep of the model which is then run iteratively (Conti and O'Hagan, 2010; Conti et al., 2009; Kennedy and O'Hagan, 2000). This shorter timestep could be chosen to be every 5 or 10 years, thereby reducing the complexity that is modelled at each step. Additionally, taking this approach could make the most of the many shorter (e.g. 5-10 years) segments of GCM simulations available for training and testing. However, as discussed previously, each emulator must be highly accurate in order to reduce the chance of errors becoming enhanced after several emulator iterations.

So far, this discussion has focused on an emulation tool that can predict the climate response at a set time in the future to emission perturbations made abruptly today. For a more realistic emulation tool, however, transient response emulators would be of interest. This thesis introduces this idea slightly, via the multi-step emulator discussed above, but further work could potentially take this much further so that time-varying scenarios can be emulated. For example, a dynamic emulator could be built that takes as inputs A) the output of the previous emulator (i.e. the 'current' estimated temperature response), and B) any changes to emission perturbations. This could then account for changing inputs as time progresses, as well as containing some 'memory' based on emission perturbations that occured in the past.

Furthermore, this thesis only emulates temperature response, whereas there are many other relevant properties of the climate that would be of interest in climate change studies. This includes climate variables such as precipitation, air pollution levels, humidity, pressure and wind speeds. Surrogate models could be implemented using the same techniques presented in this thesis, although variables such as precipitation have increased spatial and temporal variability (Pendergrass et al., 2017; Pendergrass and Knutti, 2018), which may bring additional prediction challenges, as it does for pattern scaling (Mitchell, 2003; Murphy et al., 2007; Tebaldi and

Arblaster, 2014). Learning these would provide a deeper understanding of future climate change under different projections. Future work may also be interested in emulating other statistical properties of climate that are relevant to policy decisions, not just the mean response over several years. This could include variability of climate variables and probabilities of extreme events, such as heat waves, droughts and flooding events. The latter poses additional challenges as longer GCM runs are necessary to simulate low probability events. However, with more resources going into machine learning and emulation, these tools are likely to emerge in all areas of climate change projection.

Both chapters find one of the main barriers to building a surrogate model trained on GCM output is the high levels of noise in GCM data, due to internal variability. This was identified as a cause of reduced accuracy when predicting certain regions, such as the Arctic and the Northern Hemisphere mid- and high- latitudes. These are regions are typically more sensitive to forcings, have increased internal variability and increased noise in the data, thus making them more difficult to predict relative to other parts of the globe. Future work on global climate emulation could benefit from limiting the noise present in the training data. This would traditionally require many more expensive GCM simulations, for instance, by taking an ensemble mean for the climate response. Alternatively, recent refinements in statistical methods such as pattern recognition could be employed to disentangle the signal from the noise (Sippel et al., 2019; Wills et al., 2018, 2020a,b). The use of these methods in climate science could prompt future research into emulators built to learn from signals alone, without interference of internal variability. The factor analysis approach taken in Chapter 5 demonstrates an example of how the key underlying factors of a dataset can be found. This work could be continued by emulating the latent factors, rather than the entire high dimensional dataset. Samples could then be generated by projecting the latent factors back into the original space of the data. This could give more meaning behind the emulator, as it is built to learn how the underlying factors behave under different scenarios, rather than the noise. Additionally, the input-output relationships could be explored, for instance, through the regression coefficient analysis carried out in Chapter 3 or the sensitivity analysis carried out in Chapter 4. If these techniques were applied to understand how the climate's latent factors are emulated, it may reveal how they behave under different climate scenarios.

While Chapter 5 focused on and refined a Bayesian methodology to determining the number of factors and their structure, further analysis could be carried out on climate model data. The example at the end of the chapter showed how a 35 dimensional observational dataset could be reduced to just 6 key factors explaining the behaviour of the data. This could be understood through spatial location and geography of the weather stations. These stations were, however, not equally spaced and often sparsely located over the region. The next step would be to apply the method to a high resolution gridded dataset, such as climate model output, which could highlight key patterns and relationships between different climate variables, regions or time periods. One of the difficulties in doing this is the increasing cost of the methods explored, as the data size increases. This provides an opportunity for additional research into how the computational cost could be reduced, aided by the knowledge of the performance of the methods under different situations discussed here.

With signatures of climate change is becoming harder to ignore, climate change projection under different possible future scenarios is becoming a crucial task. The methods explored in this thesis demonstrate some novel examples of how statistical methods can assist in this field, taking varying perspectives from machine learning to Bayesian statistics. In the future, climate science is likely to rely on these types of methods more to complement expensive GCMs. This thesis establishes their potential, in terms of how statistical methods can bring a deeper understanding to the underlying structure of climate data and how machine learning/emulation techniques can aid the depth, breadth and speed of climate change projection.

Bibliography

Aamaas, B., Berntsen, T. K., Fuglestvedt, J. S., Shine, K. P., and Collins, W. J. (2017). Regional temperature change potentials for short-lived climate forcers based on radiative forcing from multiple models. *Atmospheric Chemistry and Physics*, 17(17):10795–10809, DOI: `10.5194/acp-17-10795-2017`, `https://www.atmos-chem-phys.net/17/10795/2017/`.

Aanonsen, S. I., Tveit, S., and Alerini, M. (2019). Using Bayesian model probability for ranking different prior scenarios in reservoir history matching. DOI: `10.2118/194505-PA`.

Abbot, J. and Marohasy, J. (2017). The application of machine learning for evaluating anthropogenic versus natural climate change. *GeoResJ*, DOI: `10.1016/j.grj.2017.08.001`.

Agapiou, S., Papaspiliopoulos, O., Sanz-Alonso, D., and Stuart, A. M. (2017). Importance sampling: Intrinsic dimension and computational cost. *Statistical Science*, DOI: `10.1214/17-STS611`.

Aguilar, O. and West, M. (2000). Bayesian dynamic factor models and portfolio allocation. *Journal of Business and Economic Statistics*, DOI: `10.1080/07350015.2000.10524875`.

Al-Awadhi, F., Hurn, M., and Jennison, C. (2004). Improving the acceptance rate of reversible jump MCMC proposals. *Statistics and Probability Letters*, DOI: `10.1016/j.spl.2004.06.025`.

Andrews, M. B., Ridley, J. K., Wood, R. A., Andrews, T., Blockley, E. W., et al. (2020). Historical Simulations With HadGEM3-GC3.1 for CMIP6. *Journal of Advances in Modeling Earth Systems*, DOI: `10.1029/2019MS001995`.

Andrews, T., Forster, P. M., Boucher, O., Bellouin, N., and Jones, A. (2010). Precipitation, radiative forcing and global temperature change. *Geophysical Research Letters*, DOI: `10.1029/2010GL043991`.

Andrews, T., Gregory, J. M., Webb, M. J., and Taylor, K. E. (2012). Forcing, feedbacks and climate sensitivity in CMIP5 coupled atmosphere-ocean climate models. *Geophysical Research Letters*, DOI: `10.1029/2012GL051607`.

Andrieu, C., Doucet, A., Yldrm, S., and Chopin, N. (2018). On the utility of Metropolis-Hastings with asymmetric acceptance ratio. *ArXiv e-prints*.

Andrieu, C. and Vihola, M. (2015). Convergence properties of pseudo-marginal markov chain monte carlo algorithms. *Annals of Applied Probability*, DOI: `10.1214/14-AAP1022`.

Andrieu, C., Yıldırım, S., Doucet, A., and Chopin, N. (2020). Metropolis-Hastings with Averaged Acceptance Ratios.

Archer, D. and Brovkin, V. (2008). The millennial atmospheric lifetime of anthropogenic CO2. DOI: 10.1007/s10584-008-9413-1.

Baker, L. H., Collins, W. J., Olivié, D. J. L., Cherian, R., Hodnebrog, , et al. (2015). Climate responses to anthropogenic emissions of short-lived climate pollutants. *Atmospheric Chemistry and Physics*, 15(14):8201–8216, DOI: 10.5194/acp-15-8201-2015, https://www.atmos-chem-phys.net/15/8201/2015/.

Bao, J., McInerney, D. J., and Stein, M. L. (2016). A spatial-dependent model for climate emulation. *Environmetrics*, 27(7):396–408, DOI: 10.1002/env.2412.

Barnes, E. A., Hurrell, J. W., Ebert-Uphoff, I., Anderson, C., and Anderson, D. (2019). Viewing Forced Climate Patterns Through an AI Lens. *Geophysical Research Letters*, DOI: 10.1029/2019GL084944.

Barnes, E. A., Toms, B., Hurrell, J. W., Ebert-Uphoff, I., Anderson, C., and Anderson, D. (2020). Indicator Patterns of Forced Change Learned by an Artificial Neural Network. *Journal of Advances in Modeling Earth Systems*, DOI: 10.1029/2020MS002195.

Bayes, T. and Price, R. (1763). An essay towards solving a problem in the doctrine of chances by the Late Rev. Mr. Bayes. *Philosophical Transactions (1683-1775).*

Beaumont, M. A. (2003). Estimation of population growth or decline in genetically monitored populations. *Genetics*, DOI: 10.1016/B978-0-08-057106-5.50009-7.

Beddows, A. V., Kitwiroon, N., Williams, M. L., and Beevers, S. D. (2017). Emulation and Sensitivity Analysis of the Community Multiscale Air Quality Model for a UK Ozone Pollution Episode. *Environmental Science and Technology*, DOI: 10.1021/acs.est.6b05873.

Bellouin, N., Mann, G. W., Woodhouse, M. T., Johnson, C., Carslaw, K. S., and Dalvi, M. (2013). Impact of the modal aerosol scheme GLOMAP-mode on aerosol forcing in the hadley centre global environmental model. *Atmospheric Chemistry and Physics*, DOI: 10.5194/acp-13-3027-2013.

Besag, J. and Green, P. J. (1993). Spatial Statistics and Bayesian Computation. *Journal of the Royal Statistical Society: Series B (Methodological)*, DOI: 10.1111/j.2517-6161.1993.tb01467.x.

Bhattacharya, A. and Dunson, D. B. (2011). Sparse Bayesian infinite factor models. *Biometrika*, DOI: 10.1093/biomet/asr013.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics).* Springer-Verlag, Berlin, Heidelberg, ISBN: 0387310738.

Boer, G. J., Arpe, K., Blackburn, M., Déqué, M., Gates, W. L., et al. (1992). Some results from an intercomparison of the climates simulated by 14 atmospheric general circulation models. *Journal of Geophysical Research: Atmospheres*, DOI: 10.1029/92JD00722.

Boer, G. J. and Yu, B. (2003). Climate sensitivity and response. *Climate Dynamics*, DOI: 10.1007/s00382-002-0283-3.

Bolton, T. and Zanna, L. (2019). Applications of Deep Learning to Ocean Data Inference and Subgrid Parameterization. *Journal of Advances in Modeling Earth Systems*, DOI: 10.1029/2018MS001472.

Boucher, O., Granier, C., Hoose, C., and Jones, A. (2013). IPCC 2013 Chapter 7 - Clouds and aerosols. *Climate Change 2013 the Physical Science Basis: Working Group I Contribution to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change*, ISBN: 9781107415324.

Bounceur, N., Crucifix, M., and Wilkinson, R. D. (2015). Global sensitivity analysis of the climate-vegetation system to astronomical forcing: An emulator-based approach. *Earth System Dynamics*, DOI: 10.5194/esd-6-205-2015.

Bracco, A., Falasca, F., Nenes, A., Fountalis, I., and Dovrolis, C. (2018). Advancing climate science with knowledge-discovery through data mining. *npj Climate and Atmospheric Science*, DOI: 10.1038/s41612-017-0006-4.

Brock, W. and Xepapadeas, A. (2019). Regional Climate Change Policy Under Positive Feedbacks and Strategic Interactions. *Environmental and Resource Economics*, DOI: 10.1007/s10640-018-0254-8.

Brooks, S., Gelman, A., Jones, G. L., and Meng, X. L. (2011). *Handbook of Markov Chain Monte Carlo*. ISBN: 9781420079425, DOI: 10.1201/b10905.

Brooks, S. P. and Giudici, P. (2000). Markov Chain Monte Carlo Convergence Assessment via Two-Way Analysis of Variance. *Journal of Computational and Graphical Statistics*, DOI: 10.2307/1390654.

Cachier, H., Brémond, M. P., and Buat-Ménard, P. (1989). Carbonaceous aerosols from different tropical biomass burning sources. *Nature*, DOI: 10.1038/340371a0.

Carrassi, A., Bocquet, M., Hannart, A., and Ghil, M. (2017). Estimating model evidence using data assimilation. *Quarterly Journal of the Royal Meteorological Society*, DOI: 10.1002/qj.2972.

Carslaw, K. S., Lee, L. A., Reddington, C. L., Pringle, K. J., Rap, A., et al. (2013). Large contribution of natural aerosols to uncertainty in indirect forcing. *Nature*, DOI: 10.1038/nature12674.

Castruccio, S., McInerney, D. J., Stein, M. L., Crouch, F. L., Jacob, R. L., and Moyer, E. J. (2014a). Statistical emulation of climate model projections based on precomputed GCM runs. *Journal of Climate*, DOI: 10.1175/JCLI-D-13-00099.1.

Castruccio, S., McInerney, D. J., Stein, M. L., Liu Crouch, F., Jacob, R. L., and Moyer, E. J. (2014b). Statistical Emulation of Climate Model Projections Based on Precomputed GCM Runs. *Journal of Climate*, 27(5):1829–1844, DOI: 10.1175/JCLI-D-13-00099.1, https://doi.org/10.1175/JCLI-D-13-00099.1.

Ceppi, P., Zappa, G., Shepherd, T. G., and Gregory, J. M. (2017). Fast and Slow Components of the Extratropical Atmospheric Circulation Response to CO2 Forcing. *Journal of Climate*, 31(3):1091–1105, ISBN: 0894-8755, DOI: 10.1175/JCLI-D-17-0323.1, https://doi.org/10.1175/JCLI-D-17-0323.1.

Chen, B., Chen, M., Paisley, J., Zaas, A., Woods, C., et al. (2010). Bayesian inference of the number of factors in gene-expression analysis: Application to human virus challenge studies. *BMC Bioinformatics*, DOI: 10.1186/1471-2105-11-552.

Chib, S. (1995). Marginal likelihood from the Gibbs output. *Journal of the American Statistical Association*, DOI: 10.1080/01621459.1995.10476635.

Collins, M., Chandler, R. E., Cox, P. M., Huthnance, J. M., Rougier, J., and Stephenson, D. B. (2012). Quantifying future climate change. *Nature Climate Change*, 2:403 EP –, https://doi.org/10.1038/nclimate1414.

Collins, M., Knutti, R., Arblaster, J., Dufresne, J., Fichefet, T., et al. (2013a). Long-term Climate Change: Projections, Commitments and Irreversibility. In: Climate Change 2013: The Physical Science. *Climate Change 2013 the Physical Science Basis: Working Group I Contribution to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change*.

Collins, W. J., Fry, M. M., Yu, H., Fuglestvedt, J. S., Shindell, D. T., and West, J. J. (2013b). Global and regional temperature-change potentials for near-term climate forcers. *Atmospheric Chemistry and Physics*, DOI: 10.5194/acp-13-2471-2013.

Colman, R. and McAvaney, B. (2009). Climate feedbacks under a very broad range of forcing. *Geophysical Research Letters*, DOI: 10.1029/2008GL036268.

Comrey, A. L. (1978). Common methodological problems in factor analytic studies. *Journal of Consulting and Clinical Psychology*, DOI: 10.1037/0022-006X.46.4.648.

Conti, G., Frühwirth-Schnatter, S., Heckman, J. J., and Piatek, R. (2014). Bayesian exploratory factor analysis. *Journal of Econometrics*, DOI: 10.1016/j.jeconom.2014.06.008.

Conti, S., Gosling, J. P., Oakley, J. E., and O'Hagan, A. (2009). Gaussian process emulation of dynamic computer codes. *Biometrika*, DOI: 10.1093/biomet/asp028.

Conti, S. and O'Hagan, A. (2010). Bayesian emulation of complex multi-output and dynamic computer models. *Journal of Statistical Planning and Inference*, DOI: 10.1016/j.jspi.2009.08.006.

Corti, S., Molteni, F., and Palmer, T. N. (1999). Signature of recent climate change in frequencies of natural atmospheric circulation regimes. *Nature*, DOI: 10.1038/19745.

Cox, R. T. (1946). Probability, Frequency and Reasonable Expectation. *American Journal of Physics*, DOI: 10.1119/1.1990764.

Crane-Droesch, A. (2018). Machine learning methods for crop yield prediction and climate change impact assessment in agriculture. *Environmental Research Letters*, DOI: 10.1088/1748-9326/aae159.

Cubasch, U., Wuebbles, D., Chen, D., Facchini, M., Frame, D., et al. (2019). IPCC 2013: Introduction. In: Climate Change 2013: The Physical Science Basis. Contribution of Working Group I to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change. Technical report.

Cui, K. and Dunson, D. B. (2014). Generalized dynamic factor models for mixed-measurement time series. *Journal of Computational and Graphical Statistics*, DOI: 10.1080/10618600.2012.729986.

Cumming, J. and Goldstein, M. (2009). Bayes Linear Uncertainty Analysis for Oil Reservoirs Based on Multiscale Computer Experiments. *The Oxford Handbook of Applied Bayesian Analysis.*

Currin, C., Mitchell, T., Morris, M., and Ylvisaker, D. (1991). Bayesian prediction of deterministic functions, with applications to the design and analysis of computer experiments. *Journal of the American Statistical Association*, DOI: 10.1080/01621459.1991.10475138.

Curtius, J. (2006). Nucleation of atmospheric aerosol particles. DOI: 10.1016/j.crhy.2006.10.018.

Damianou, A. C., Titsias, M. K., and Lawrence, N. D. (2016). Variational inference for latent variables and uncertain inputs in Gaussian processes. *Journal of Machine Learning Research.*

Del Moral, P. (1997). Nonlinear filtering: Interacting particle resolution. *Comptes Rendus de l'Académie des Sciences - Series I - Mathematics*, DOI: 10.1016/s0764-4442(97)84778-7.

Del Moral, P., Doucet, A., and Jasra, A. (2006). Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, 68(3):411–436, DOI: 10.1111/j.1467-9868.2006.00553.x.

Dentener, F. J., Easterling, D. R., Uk, R. A., Uk, R. A., Cooper, O., et al. (2013). IPCC Climate Change 2013: The Physical Science Basis. Chapter 2: Observations: Atmosphere and Surface. *Climate Change 2013 the Physical Science Basis: Working Group I Contribution to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change*, ISBN: 9781107415324.

Dong, B., Gregory, J. M., and Sutton, R. T. (2009). Understanding land-sea warming contrast in response to increasing greenhouse gases. Part I: Transient adjustment. *Journal of Climate*, DOI: 10.1175/2009JCLI2652.1.

Dong, B., Sutton, R. T., Highwood, E., and Wilcox, L. (2014). The impacts of European and Asian anthropogenic sulfur dioxide emissions on Sahel rainfall. *Journal of Climate*, DOI: 10.1175/JCLI-D-13-00769.1.

Dormann, C. F., Elith, J., Bacher, S., Buchmann, C., Carl, G., et al. (2013). Collinearity: A review of methods to deal with it and a simulation study evaluating their performance. *Ecography*, DOI: 10.1111/j.1600-0587.2012.07348.x.

Doucet, A., Freitas, N., and Gordon, N. (2001). An Introduction to Sequential Monte Carlo Methods. In *Sequential Monte Carlo Methods in Practice.* DOI:

Dueben, P. D. and Bauer, P. (2018). Challenges and design choices for global weather and climate models based on machine learning. *Geoscientific Model Development*, DOI: 10.5194/gmd-11-3999-2018.

Dunson, D. B. (2006). Efficient Bayesian model averaging in factor analysis. *Duke University.*

Eckhardt, R. (1987). Stan ulam, john von neumann, and the monte carlo method. *Los Alamos Science.*

ECLIPSE (2014). ECLIPSE V5 global emission fields. https://iiasa.ac.at/web/home/research/researchPrograms/air/ECLIPSEv5.html.

Edwards, T. L., Brandon, M. A., Durand, G., Edwards, N. R., Golledge, N. R., et al. (2019). Revisiting Antarctic ice loss due to marine ice-cliff instability. *Nature*, DOI: 10.1038/s41586-019-0901-4.

Everitt, R. G., Culliford, R., Medina-Aguayo, F., and Wilson, D. J. (2018). Sequential Bayesian inference for mixture models and the coalescent using sequential Monte Carlo samplers with transformations. *arXiv*.

Everitt, R. G., Culliford, R., Medina-Aguayo, F., and Wilson, D. J. (2020). Sequential Monte Carlo with transformations. *Statistics and Computing*, DOI: 10.1007/s11222-019-09903-y.

Everitt, R. G., Johansen, A. M., Rowing, E., and Evdemon-Hogan, M. (2017). Bayesian model comparison with un-normalised likelihoods. *Statistics and Computing*, DOI: 10.1007/s11222-016-9629-2.

Eyring, V., Bony, S., Meehl, G. A., Senior, C. A., Stevens, B., et al. (2016). Overview of the Coupled Model Intercomparison Project Phase 6 (CMIP6) experimental design and organization. *Geoscientific Model Development*, DOI: 10.5194/gmd-9-1937-2016.

Fabrigar, L. R., MacCallum, R. C., Wegener, D. T., and Strahan, E. J. (1999). Evaluating the use of exploratory factor analysis in psychological research. DOI: 10.1037/1082-989X.4.3.272.

Fava, J. L. and Velicer, W. F. (1992). The Effects of Overextraction on Factor and Component Analysis. *Multivariate Behavioral Research*, DOI:

Feichter, J., Roeckner, E., Lohmann, U., and Liepert, B. (2004). Nonlinear aspects of the climate response to Greenhouse gas and aerosol forcing. *Journal of Climate*, DOI: 10.1175/1520-0442(2004)017<2384:NAOTCR>2.0.CO;2.

Foley, A. M., Holden, P. B., Edwards, N. R., Mercure, J. F., Salas, P., et al. (2016). Climate model emulation in an integrated assessment framework: A case study for mitigation policies in the electricity sector. *Earth System Dynamics*, DOI: 10.5194/esd-7-119-2016.

Friedrich, T., Timmermann, A., Tigchelaar, M., Timm, O. E., and Ganopolski, A. (2016). Nonlinear climate sensitivity and its implications for future greenhouse warming. *Science Advances*, DOI: 10.1126/sciadv.1501923.

Fu, Q., Manabe, S., and Johanson, C. M. (2011). On the warming in the tropical upper troposphere: Models versus observations. *Geophysical Research Letters*, DOI: 10.1029/2011GL048101.

Gates, W. L., Boyle, J. S., Covey, C., Dease, C. G., Doutriaux, C. M., et al. (1999). An Overview of the Results of the Atmospheric Model Intercomparison Project (AMIP I). DOI: 10.1175/1520-0477(1999)080<0029:AOOTRO>2.0.CO;2.

Gelfand, A. E. and Dey, D. K. (1994). Bayesian Model Choice: Asymptotics and Exact Calculations. *Journal of the Royal Statistical Society: Series B (Methodological)*, DOI: 10.1111/j.2517-6161.1994.tb01996.x.

Gelman, A. and Meng, X. L. (1998). Simulating normalizing constants: From importance sampling to bridge sampling to path sampling. *Statistical Science*, DOI: 10.1214/ss/1028905934.

Geman, S. and Geman, D. (1984). Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, DOI: 10.1109/TPAMI.1984.4767596.

Gentine, P., Pritchard, M., Rasp, S., Reinaudi, G., and Yacalis, G. (2018). Could Machine Learning Break the Convection Parameterization Deadlock? *Geophysical Research Letters*, DOI: 10.1029/2018GL078202.

Geweke, J. and Zhou, G. (1996). Measuring the Pricing Error of the Arbitrage Pricing Theory. *Review of Financial Studies*, DOI: 10.1093/rfs/9.2.557.

Ghosh, J. and Dunson, D. B. (2009). Default prior distributions and efficient posterior computation in Bayesian factor analysis. *Journal of Computational and Graphical Statistics*, DOI: 10.1198/jcgs.2009.07145.

Gidden, M. J., Riahi, K., Smith, S. J., Fujimori, S., Luderer, G., et al. (2019). Global emissions pathways under different socioeconomic scenarios for use in CMIP6: A dataset of harmonized emissions trajectories through the end of the century. *Geoscientific Model Development*, DOI: 10.5194/gmd-12-1443-2019.

Girard, A., Rasmussen, C. E., and Murray-Smith, R. (2002). Gaussian Process priors with Uncertain Inputs : Multiple-Step-Ahead Prediction. *Technical Report TR-2002-119*.

Goldstein, M. and Rougier, J. (2006). Bayes linear calibrated prediction for complex systems. DOI: 10.1198/016214506000000203.

Gordon, N. J., Salmond, D. J., and Smith, A. F. (1993). Novel approach to nonlinear/non-gaussian Bayesian state estimation. *IEE Proceedings, Part F: Radar and Signal Processing*, DOI: 10.1049/ip-f-2.1993.0015.

Gorte, B. and Stein, A. (1998). Bayesian classification and class area estimation of satellite images using stratification. *IEEE Transactions on Geoscience and Remote Sensing*, DOI: 10.1109/36.673673.

GPy (2014). GPy: A gaussian process framework in python. http://github.com/SheffieldML/GPy.

Green, P. J. (1995). Reversible jump Markov chain monte carlo computation and Bayesian model determination. *Biometrika*, DOI: 10.1093/biomet/82.4.711.

Green, P. J. and Mira, A. (2001). Delayed rejection in reversible jump Metropolis-Hastings. *Biometrika*, DOI: 10.1093/biomet/88.4.1035.

Gregory, J. M. and Andrews, T. (2016). Variation in climate sensitivity and feedback parameters during the historical period. *Geophysical Research Letters*, DOI: 10.1002/2016GL068406.

Gregory, J. M., Andrews, T., and Good, P. (2015). The inconstancy of the transient climate response parameter under increasing CO2. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, DOI: 10.1098/rsta.2014.0417.

Gregory, J. M., Ingram, W. J., Palmer, M. A., Jones, G. S., Stott, P. A., et al. (2004). A new method for diagnosing radiative forcing and climate sensitivity. *Geophysical Research Letters*, 31(3), ISBN: 0094-8276, DOI: 10.1029/2003GL018747, https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2003GL018747https://doi.org/10.1029/2003GL018747.

Hall, A., Cox, P., Huntingford, C., and Klein, S. (2019). Progressing emergent constraints on future climate change. DOI: `10.1038/s41558-019-0436-6`.

Halton, J. H. (1970). Retrospective and Prospective Survey of the Monte Carlo Method. *SIAM Review*, DOI: `10.1137/1012001`.

Hannachi, A. (2004). *A primer for EOF analysis of climate data.*

Hannachi, A., Jolliffe, I. T., Stephenson, D. B., and Trendafilov, N. (2006). In search of simple structures in climate: Simplifying EOFS. *International Journal of Climatology*, DOI: `10.1002/joc.1243`.

Hansen, J., Sato, M., and Ruedy, R. (1997). Radiative forcing and climate response. *Journal of Geophysical Research Atmospheres*, DOI: `10.1029/96JD03436`.

Hansen, J., Sato, M., Ruedy, R., Lacis, A., and Oinas, V. (2000). Global warming in the twenty-first century: An alternative scenario. *Proceedings of the National Academy of Sciences of the United States of America*, DOI: `10.1073/pnas.170278997`.

Hansen, J., Sato, M., Ruedy, R., Nazarenko, L., Lacis, A., et al. (2005). Efficacy of climate forcings. DOI: `10.1029/2005JD005776`, `https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2005JD005776`.

Harris, G. R., Sexton, D. M. H., Booth, B. B. B., Collins, M., Murphy, J. M., and Webb, M. J. (2006). Frequency distributions of transient regional climate change from perturbed physics ensembles of general circulation model simulations. *Climate Dynamics*, 27(4):357–375, DOI: `10.1007/s00382-006-0142-8`, `https://doi.org/10.1007/s00382-006-0142-8`.

Hartmann, D. L., Blossey, P. N., and Dygert, B. D. (2019). Convection and Climate: What Have We Learned from Simple Models and Simplified Settings? DOI: `10.1007/s40641-019-00136-9`.

Hastie, T., Tibshirani, R., and Friedman, J. (2001). *The Elements of Statistical Learning.* Springer Series in Statistics. Springer New York Inc., New York, NY, USA.

Hastings, W. K. (1970). Monte carlo sampling methods using Markov chains and their applications. *Biometrika*, DOI: `10.1093/biomet/57.1.97`.

Haywood, J. and Boucher, O. (2000). Estimates of the direct and indirect radiative forcing due to tropospheric aerosols: A review. DOI: `10.1029/1999RG000078`.

Heicklen, J. (1982). Atmospheric lifetimes of pollutants. *Atmospheric Environment (1967)*, DOI: `10.1016/0004-6981(82)90400-0`.

Held, I. M. and Soden, B. J. (2006). Robust responses of the hydrological cycle to global warming. *Journal of Climate*, DOI: `10.1175/JCLI3990.1`.

Held, I. M., Winton, M., Takahashi, K., Delworth, T., Zeng, F., and Vallis, G. K. (2010). Probing the Fast and Slow Components of Global Warming by Returning Abruptly to Preindustrial Forcing. *Journal of Climate*, 23(9):2418–2427, DOI: `10.1175/2009JCLI3466.1`, `https://doi.org/10.1175/2009JCLI3466.1`.

Higdon, D., Gattiker, J., Williams, B., and Rightley, M. (2008). Computer model calibration using high-dimensional output. *Journal of the American Statistical Association*, DOI: 10.1198/016214507000000888.

Higdon, D. M. (1998). Auxiliary variable methods for markov chain monte carlo with applications. *Journal of the American Statistical Association*, DOI: 10.1080/01621459.1998.10473712.

Hodnebrog, O., Myhre, G., Forster, P. M., Sillmann, J., and Samset, B. H. (2016). Local biomass burning is a dominant cause of the observed precipitation reduction in southern Africa. *Nature Communications*, DOI: 10.1038/ncomms11236.

Hoerl, A. E. and Kennard, R. W. (1970). Ridge Regression: Applications to Nonorthogonal Problems. *Technometrics*, DOI: 10.1080/00401706.1970.10488635.

Holden, P. B. and Edwards, N. R. (2010). Dimensionally reduced emulation of an AOGCM for application to integrated assessment modelling. *Geophysical Research Letters*, DOI: 10.1029/2010GL045137.

Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, DOI: 10.1037/h0071325.

Hulme, M., Raper, S. C. B., and Wigley, T. M. L. (1995). An integrated framework to address climate change (ESCAPE) and further developments of the global and regional climate modules (MAGICC). *Energy Policy*, 23(4):347–355, DOI: https://doi.org/10.1016/0301-4215(95)90159-5, http://www.sciencedirect.com/science/article/pii/0301421595901595.

Huntingford, C. and Cox, P. M. (2000). An analogue model to derive additional climate change scenarios from existing GCM simulations. *Climate Dynamics*, 16(8):575–586, DOI: 10.1007/s003820000067, https://doi.org/10.1007/s003820000067.

Huntingford, C., Jeffers, E. S., Bonsall, M. B., Christensen, H. M., Lees, T., and Yang, H. (2019). Machine learning and artificial intelligence to aid climate change research and preparedness. *Environmental Research Letters*, DOI: 10.1088/1748-9326/ab4e55.

Intel® (2021). Xeon® Processor E5-2680 v3. *https://ark.intel.com/content/www/us/en/ark/products/81908/xeon-processor-e5-2680-v3-30m-cache-2-50-ghz.html. Date Accessed: 04/06/2021.*

Intergovernmental Panel on Climate Change (IPCC) (2014). Annex II: Glossary [Mach, K.J., S. Planton and C. von Stechow (eds.)]. *Climate Change 2014: Synthesis Report. Contribution of Working Groups I, II and III to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change*, ISBN: 9789291691432.

IPCC (2013). Working Group 1, Summary for Policymakers. *Climate Change 2013: The Physical Science Basis. Contribution of Working Group I to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change*, ISBN: 9781107661820.

IPCC (2021). About — IPCC. *https://www.ipcc.ch/about/ Date Accessed: 30/03/2021.*

Ishizaki, Y., Shiogama, H., Emori, S., Yokohata, T., Nozawa, T., et al. (2012). Temperature scaling pattern dependence on representative concentration pathway emission scenarios. *Climatic Change*, 112(2):535–546, DOI: 10.1007/s10584-012-0430-8, https://doi.org/10.1007/s10584-012-0430-8.

Johansen, A. M. and Evers, L. (2010). Monte carlo methods. *Lecture notes 200, University of Warwick.*

Johnson, B. T., Shine, K. P., and Forster, P. M. (2004). The semi-direct aerosol effect: Impact of absorbing aerosols on marine stratocumulus. *Quarterly Journal of the Royal Meteorological Society*, DOI: 10.1256/qj.03.61.

Joliffe, I. T. and Morgan, B. (1992). Principal component analysis and exploratory factor analysis. *Statistical Methods in Medical Research*, DOI: 10.1177/096228029200100105.

Jolliffe, I. T. (2002). Principal Component Analysis, Second Edition. *Encyclopedia of Statistics in Behavioral Science*, ISBN: 0387954422, DOI: 10.2307/1270093.

Joshi, M. M., Gregory, J. M., Webb, M. J., Sexton, D. M., and Johns, T. C. (2008). Mechanisms for the land/sea warming contrast exhibited by simulations of climate change. *Climate Dynamics*, DOI: 10.1007/s00382-007-0306-1.

Karagiannis, G. and Andrieu, C. (2013). Annealed Importance Sampling Reversible Jump MCMC Algorithms. *Journal of Computational and Graphical Statistics*, 22(3):623–648, ISBN: 1061-8600, DOI: 10.1080/10618600.2013.805651, https://doi.org/10.1080/10618600.2013.805651.

Karpatne, A. and Kumar, V. (2017). Big data in climate: Opportunities and challenges for machine learning. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* ISBN: 9781450348874, DOI: 10.1145/3097983.3105810.

Kashinath, K., Mustafa, M., Albert, A., Wu, J. L., Jiang, C., et al. (2021). Physics-informed machine learning: Case studies for weather and climate modelling. DOI: 10.1098/rsta.2020.0093.

Kasoar, M., Shawki, D., and Voulgarakis, A. (2018). Similar spatial patterns of global climate response to aerosols from different regions. *npj Climate and Atmospheric Science*, 1(1):12, ISBN: 2397-3722, DOI: 10.1038/s41612-018-0022-z, https://doi.org/10.1038/s41612-018-0022-z.

Kasoar, M., Voulgarakis, A., Lamarque, J. F., Shindell, D. T., Bellouin, N., et al. (2016). Regional and global temperature response to anthropogenic SO2 emissions from China in three climate models. *Atmospheric Chemistry and Physics*, DOI: 10.5194/acp-16-9785-2016.

Keeling, C. D. (1961). The concentration and isotopic abundances of carbon dioxide in rural and marine air. *Geochimica et Cosmochimica Acta*, DOI: 10.1016/0016-7037(61)90023-0.

Kennedy, M. C. and O'Hagan, A. (2000). Predicting the output from a complex computer code when fast approximations are available. *Biometrika*, DOI: 10.1093/biomet/87.1.1.

Kennedy, M. C. and O'Hagan, A. (2001). Bayesian calibration of computer models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, DOI: 10.1111/1467-9868.00294.

Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, DOI: 10.1126/science.220.4598.671.

Kitagawa, G. (1996). Monte Carlo Filter and Smoother for Non-Gaussian Nonlinear State Space Models. *Journal of Computational and Graphical Statistics*, DOI: 10.2307/1390750.

Knüsel, B., Zumwald, M., Baumberger, C., Hirsch Hadorn, G., Fischer, E. M., et al. (2019). Applying big data beyond small problems in climate research. DOI: 10.1038/s41558-019-0404-1.

Knutti, R., Rugenstein, M. A., and Hegerl, G. C. (2017). Beyond equilibrium climate sensitivity. DOI: 10.1038/NGEO3017.

Kretschmer, M., Coumou, D., Donges, J. F., and Runge, J. (2016). Using causal effect networks to analyze different arctic drivers of midlatitude winter circulation. *Journal of Climate*, DOI: 10.1175/JCLI-D-15-0654.1.

Kretschmer, M., Runge, J., and Coumou, D. (2017). Early prediction of extreme stratospheric polar vortex states based on causal precursors. *Geophysical Research Letters*, DOI: 10.1002/2017GL074696.

Kuhn-Régnier, A., Voulgarakis, A., Nowack, P., Forkel, M., Prentice, I. C., and Harrison, S. P. (2020). Quantifying the Importance of Antecedent Fuel-Related Vegetation Properties for Burnt Area using Random Forests. *Biogeosciences Discussions*, 2020:1–24, DOI: 10.5194/bg-2020-409, https://bg.copernicus.org/preprints/bg-2020-409/.

Lamarque, J. F., Shindell, D. T., Josse, B., Young, P. J., Cionni, I., et al. (2013). The atmospheric chemistry and climate model intercomparison Project (ACCMIP): Overview and description of models, simulations and climate diagnostics. *Geoscientific Model Development*, DOI: 10.5194/gmd-6-179-2013.

Lasslop, G., Coppola, A. I., Voulgarakis, A., Yue, C., and Veraverbeke, S. (2019). Influence of Fire on the Carbon Cycle and Climate. DOI: 10.1007/s40641-019-00128-9.

Lawrence, N. (2005). Probabilistic non-linear principal component analysis with Gaussian process latent variable models. *Journal of Machine Learning Research.*

Lawrence, N. D. (2004). Gaussian process latent variable models for visualisation of high dimensional data. In *Advances in Neural Information Processing Systems.* ISBN: 0262201526.

Lee, L. A., Carslaw, K. S., Pringle, K. J., and Mann, G. W. (2012). Mapping the uncertainty in global CCN using emulation. *Atmospheric Chemistry and Physics*, DOI: 10.5194/acp-12-9739-2012.

Lee, L. A., Carslaw, K. S., Pringle, K. J., Mann, G. W., and Spracklen, D. V. (2011). Emulation of a complex global aerosol model to quantify sensitivity to uncertain parameters. *Atmospheric Chemistry and Physics*, DOI: 10.5194/acp-11-12253-2011.

Lee, L. A., Pringle, K. J., Reddington, C. L., Mann, G. W., Stier, P., et al. (2013). The magnitude and causes of uncertainty in global model simulations of cloud condensation nuclei. *Atmospheric Chemistry and Physics*, DOI: 10.5194/acp-13-8879-2013.

Lee, L. A., Reddington, C. L., and Carslaw, K. S. (2016). On the relationship between aerosol model uncertainty and radiative forcing uncertainty. *Proceedings of the National Academy of Sciences of the United States of America*, DOI: 10.1073/pnas.1507050113.

Leroy, S. S. (1998). Detecting climate signals: some Bayesian aspects. *Journal of Climate*, DOI: 10.1175/1520-0442(1998)011<0640:DCSSBA>2.0.CO;2.

Levy, H., Horowitz, L. W., Schwarzkopf, M. D., Ming, Y., Golaz, J. C., et al. (2013). The roles of aerosol direct and indirect effects in past and future climate change. *Journal of Geophysical Research Atmospheres*, DOI: 10.1002/jgrd.50192.

Lewinschal, A., Ekman, A. M., Hansson, H. C., Sand, M., Berntsen, T. K., and Langner, J. (2019). Local and remote temperature response of regional SO2 emissions. *Atmospheric Chemistry and Physics*, DOI: 10.5194/acp-19-2385-2019.

Liousse, C., Assamoi, E., Criqui, P., Granier, C., and Rosset, R. (2014). Explosive growth in African combustion emissions from 2005 to 2030. *Environmental Research Letters*, DOI: 10.1088/1748-9326/9/3/035003.

Liu, H., Cai, J., and Ong, Y. S. (2018a). Remarks on multi-output Gaussian process regression. *Knowledge-Based Systems*, DOI: 10.1016/j.knosys.2017.12.034.

Liu, J. S. and Chen, R. (1998). Sequential monte carlo methods for dynamic systems. *Journal of the American Statistical Association*, DOI: 10.1080/01621459.1998.10473765.

Liu, L., Shawki, D., Voulgarakis, A., Kasoar, M., Samset, B. H., et al. (2018b). A PDRMIP Multimodel study on the impacts of regional aerosol forcings on global and regional precipitation. *Journal of Climate*, DOI: 10.1175/JCLI-D-17-0439.1.

Loeppky, J. L., Sacks, J., and Welch, W. J. (2009). Choosing the sample size of a computer experiment: A practical guide. *Technometrics*, DOI: 10.1198/TECH.2009.08040.

Lopes, H. F. and West, M. (2004). Bayesian model assessment in factor analysis.

Lutter, M., Ritter, C., and Peters, J. (2019). Deep Lagrangian networks: Using physics as model prior for deep learning. In *7th International Conference on Learning Representations, ICLR 2019*.

MacKay, D. (1998). Introduction to {G}aussian Processes. In *Book Neural Networks and Machine Learning, Springer-Verlag*.

MacMartin, D. G. and Kravitz, B. (2016). Dynamic climate emulators for solar geoengineering. *Atmospheric Chemistry and Physics*, DOI: 10.5194/acp-16-15789-2016.

Manabe, S. and Bryan, K. (1969). Climate Calculations with a Combined Ocean-Atmosphere Model. *Journal of the Atmospheric Sciences*, DOI: 10.1175/1520-0469(1969)026<0786:ccwaco>2.0.co;2.

Manabe, S. and Wetherald, R. T. (1980). On the Distribution of Climate Change Resulting from an Increase in CO2 Content of the Atmosphere. *Journal of the Atmospheric Sciences*, 37(1):99–118, DOI: 10.1175/1520-0469(1980)037<0099:OTDOCC>2.0.CO;2, https://doi.org/10.1175/1520-0469(1980)037%3C0099:OTDOCC%3E2.0.CO;2.

Mann, G. W., Carslaw, K. S., Spracklen, D. V., Ridley, D. A., Manktelow, P. T., et al. (2010). Description and evaluation of GLOMAP-mode: A modal global aerosol microphysics model for the UKCA composition-climate model. *Geoscientific Model Development*, DOI: 10.5194/gmd-3-519-2010.

Mansfield, L. A., Nowack, P. J., Kasoar, M., Everitt, R. G., Collins, W. J., and Voulgarakis, A. (2020). Predicting global patterns of long-term climate change from short-term simulations using machine learning. *npj Climate and Atmospheric Science*, DOI: `10.1038/s41612-020-00148-5`.

Marrel, A., Iooss, B., Jullien, M., Laurent, B., and Volkova, E. (2011). Global sensitivity analysis for models with spatially dependent outputs. *Environmetrics*, DOI: `10.1002/env.1071`.

Marvel, K., Schmidt, G. A., Shindell, D., Bonfils, C., Legrande, A. N., et al. (2015). Do responses to different anthropogenic forcings add linearly in climate models? *Environmental Research Letters*, DOI: `10.1088/1748-9326/10/10/104010`.

Masson-Delmotte, V.; Schulz, M. (2012). Chapter 5 : Information from Paleoclimate Archives. *Ipcc.*

May, W. (2012). Assessing the strength of regional changes in near-surface climate associated with a global warming of 2°C. *Climatic Change*, 110(3):619–644, DOI: `10.1007/s10584-011-0076-y`, `https://doi.org/10.1007/s10584-011-0076-y`.

McGuffie, K. and Henderson-Sellers, A. (2001). Forty years of numerical climate modelling. *International Journal of Climatology*, DOI: `10.1002/joc.632`.

McKay, M. D., Beckman, R. J., and Conover, W. J. (1979). Comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, DOI: `10.1080/00401706.1979.10489755`.

McNeall, D., Williams, J., Betts, R., Booth, B., Challenor, P., et al. (2019). Correcting a bias in a climate model with an augmented emulator. *Geoscientific Model Development Discussions*, DOI: `10.5194/gmd-2019-171`.

McNeall, D. J., Challenor, P. G., Gattiker, J. R., and Stone, E. J. (2013). The potential of an observational data set for calibration of a computationally expensive computer model. *Geoscientific Model Development Discussions*, DOI: `10.5194/gmdd-6-2369-2013`.

Medina-Aguayo, F. J., Didelot, X., and Everitt, R. G. (2020). Speeding up Inference of Homologous Recombination in Bacteria. *bioRxiv*, page 2020.05.10.087007, DOI: `10.1101/2020.05.10.087007`, `http://biorxiv.org/content/early/2020/05/10/2020.05.10.087007.abstract`.

Meehl, G. A., Covey, C., Delworth, T., Latif, M., McAvaney, B., et al. (2007). The WCRP CMIP3 multimodel dataset: A new era in climatic change research. *Bulletin of the American Meteorological Society*, DOI: `10.1175/BAMS-88-9-1383`.

Meinshausen, M., Meinshausen, N., Hare, W., Raper, S. C., Frieler, K., et al. (2009). Greenhouse-gas emission targets for limiting global warming to 2°C. *Nature*, DOI: `10.1038/nature08017`.

Meraner, K., Mauritsen, T., and Voigt, A. (2013). Robust increase in equilibrium climate sensitivity under global warming. *Geophysical Research Letters*, DOI: `10.1002/2013GL058118`.

Met Office (2021). The Cray XC40 supercomputing system. *https://www.metoffice.gov.uk/about-us/what/technology/supercomputer. Date Accessed: 21/06/2021.*

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*, DOI: `10.1063/1.1699114`.

Miftakhova, A., Judd, K. L., Lontzek, T. S., and Schmedders, K. (2020). Statistical approximation of high-dimensional climate models. *Journal of Econometrics*, DOI: `10.1016/j.jeconom.2019.05.005`.

Ming, Y. and Ramaswamy, V. (2009). Nonlinear climate and hydrological responses to aerosol effects. *Journal of Climate*, DOI: `10.1175/2008JCLI2362.1`.

Mitchell, T. D. (2003). Pattern Scaling: An Examination of the Accuracy of the Technique for Describing Future Climates. *Climatic Change*, 60(3):217–242, DOI: `10.1023/A:1026035305597`, `https://doi.org/10.1023/A:1026035305597`.

Møller, J., Pettitt, A. N., Berthelsen, K. K., and Reeves, R. W. (2004). An efficient Markov chain Monte Carlo method for distributions with intractable normalising constants. *Biometrika*.

Moss, R. H., Edmonds, J. A., Hibbard, K. A., Manning, M. R., Rose, S. K., et al. (2010). The next generation of scenarios for climate change research and assessment. DOI: `10.1038/nature08823`.

Murphy, D. M., Solomon, S., Portmann, R. W., Rosenlof, K. H., Forster, P. M., and Wong, T. (2009). An observationally based energy balance for the Earth since 1950. *Journal of Geophysical Research Atmospheres*, DOI: `10.1029/2009JD012105`.

Murphy, J. M., Booth, B. B. B., Collins, M., Harris, G. R., Sexton, D. M. H., and Webb, M. J. (2007). A methodology for probabilistic predictions of regional climate change from perturbed physics ensembles. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 365(1857):1993–2028.

Murphy, K. P. (2012). *Machine learning: a probabilistic perspective (adaptive computation and machine learning series)*. ISBN: `0262018020`.

Murray, I., Ghahramani, Z., and MacKay, D. J. (2006). MCMC for doubly-intractable distributions. In *Proceedings of the 22nd Conference on Uncertainty in Artificial Intelligence, UAI 2006*. ISBN: `0974903922`.

Myhre, G., Forster, P. M., Samset, B. H., Hodnebrog, Sillmann, J., et al. (2017). PDRMIP: A precipitation driver and response model intercomparison project-protocol and preliminary results. In *Bulletin of the American Meteorological Society*, volume 98 of *EGU General Assembly Conference Abstracts*, pages 1185–1198. DOI: `10.1175/BAMS-D-16-0019.1`.

Myhre, G., Shindell, D., Bréon, F.-m., Collins, W., Fuglestvedt, J., et al. (2013). *Anthropogenic and natural radiative forcing. In: Climate change 2013: the physical science basis. Contribution of working group I.* ISBN: `9781107415324`.

Neal, R. M. (2001). Annealed importance sampling. *Statistics and Computing*, DOI: `10.1023/A:1008923215028`.

Neal, R. M. (2005). Taking Bigger Metropolis Steps by Dragging Fast Variables. `https://arxiv.org/abs/math/0502099`.

Newton, M. A. and Raftery, A. E. (1994). Approximate Bayesian Inference with the Weighted Likelihood Bootstrap. *Journal of the Royal Statistical Society: Series B (Methodological)*, DOI: `10.1111/j.2517-6161.1994.tb01956.x`.

Nicholls, G., Fox, C., and Watt, A. (2012). Coupled MCMC with a randomized acceptance probability. *ArXiv e-prints*.

Nowack, P., Braesicke, P., Haigh, J., Abraham, N. L., Pyle, J., and Voulgarakis, A. (2018). Using machine learning to build temperature-based ozone parameterizations for climate sensitivity simulations. *Environmental Research Letters*, DOI: `10.1088/1748-9326/aae2be`.

Nowack, P., Runge, J., Eyring, V., and Haigh, J. D. (2020). Causal networks for climate model evaluation and constrained projections. *Nature Communications*, DOI: `10.1038/s41467-020-15195-y`.

Nowack, P. J., Braesicke, P., Luke Abraham, N., and Pyle, J. A. (2017). On the role of ozone feedback in the ENSO amplitude response under global warming. *Geophysical Research Letters*, DOI: `10.1002/2016GL072418`.

Oakley, J. and O'Hagan, A. (2002). Bayesian inference for the uncertainty distribution of computer model outputs. *Biometrika*, DOI: `10.1093/biomet/89.4.769`.

O'Gorman, P. A. and Dwyer, J. G. (2018). Using Machine Learning to Parameterize Moist Convection: Potential for Modeling of Climate, Climate Change, and Extreme Events. *Journal of Advances in Modeling Earth Systems*, DOI: `10.1029/2018MS001351`.

O'Hagan, A. (1978). Curve Fitting and Optimal Design for Prediction. *Journal of the Royal Statistical Society: Series B (Methodological)*, DOI: `10.1111/j.2517-6161.1978.tb01643.x`.

O'Hagan, A. (2006). Bayesian analysis of computer code outputs: A tutorial. *Reliability Engineering and System Safety*, DOI: `10.1016/j.ress.2005.11.025`.

O'Hagan, a., Kennedy, M. C., and Oakley, J. E. (1999). Uncertainty analysis and other inference tools for complex computer codes. *Bayesian Staistics 6*, pages 503–524, `http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.51.446`.

Oort, A. H. and Peixóto, J. P. (1983). Global angular momentum and energy balance requirements from observations. *Advances in Geophysics*, DOI: `10.1016/S0065-2687(08)60177-6`.

O'Neill, B. C., Carter, T. R., Ebi, K., Harrison, P. A., Kemp-Benedict, E., et al. (2020). Achievements and needs for the climate change scenario framework. *Nature Climate Change*, DOI: `10.1038/s41558-020-00952-0`.

Pachauri, R., Allen, M., Barros, V. R., Broome, J., Cramer, W., et al. (2014). IPCC, 2014. *CLIMATE CHANGE 2014 Synthesis Report Summary for Policymakers*, DOI: `citeulike-article-id:2297298`.

Palmer, T. N. (1999). A nonlinear dynamical perspective on climate prediction. *Journal of Climate*, DOI: `10.1175/1520-0442(1999)012<0575:ANDPOC>2.0.CO;2`.

Pearson, K. (1901). On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, DOI: `10.1080/14786440109462720`.

Pechony, O. and Shindell, D. T. (2010). Driving forces of global wildfires over the past millennium and the forthcoming century. *Proceedings of the National Academy of Sciences of the United States of America*, DOI: `10.1073/pnas.1003669107`.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., et al. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Pendergrass, A. G. and Knutti, R. (2018). The Uneven Nature of Daily Precipitation and Its Change. *Geophysical Research Letters*, DOI: `10.1029/2018GL080298`.

Pendergrass, A. G., Knutti, R., Lehner, F., Deser, C., and Sanderson, B. M. (2017). Precipitation variability increases in a warmer climate. *Scientific Reports*, DOI: `10.1038/s41598-017-17966-y`.

Persad, G. G., Ming, Y., Shen, Z., and Ramaswamy, V. (2018). Spatially similar surface energy flux perturbations due to greenhouse gases and aerosols. *Nature Communications*, DOI: `10.1038/s41467-018-05735-y`.

Popp, A., Calvin, K., Fujimori, S., Havlik, P., Humpenöder, F., et al. (2017). Land-use futures in the shared socio-economic pathways. *Global Environmental Change*, DOI: `10.1016/j.gloenvcha.2016.10.002`.

Press, S. J. and Shigemasu, K. (1989). Bayesian Inference in Factor Analysis. In *Contributions to Probability and Statistics*. DOI:

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*, DOI: `10.1016/j.jcp.2018.10.045`.

Ramsay, J. and Silvermann, B. (1998). Functional Data Analysis. Springer Series in Statistics. *Biometrical Journal*, DOI:

Ramsay, J. O. (2003). Functional Data Analysis: Weather Data Analysis. *https://www.psych.mcgill.ca/misc/fda/ex-weather-c1.html, Date Accessed: 13/05/2021.*

Ramsay, J. O., Wickham, H., Graves, S., and Hooker, G. (2014). Package 'fda': Functional data analysis.

Rasmussen, C. E. (2004). Gaussian Processes in machine learning. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, DOI:

Rasmussen, C. E. and Williams, C. K. I. (2006). Gaussian Processes for Machine Learning.

Rasp, S., Pritchard, M. S., and Gentine, P. (2018). Deep learning to represent subgrid processes in climate models. *Proceedings of the National Academy of Sciences of the United States of America*, DOI: `10.1073/pnas.1810286115`.

Ratto, M., Castelletti, A., and Pagano, A. (2012). Emulation techniques for the reduction and sensitivity analysis of complex environmental models. DOI: `10.1016/j.envsoft.2011.11.003`.

Reich, S. and Cotter, C. (2015). *Probabilistic forecasting and bayesian data assimilation.* ISBN: `9781107706804`, DOI: `10.1017/CBO9781107706804`.

Reichstein, M., Camps-Valls, G., Stevens, B., Jung, M., Denzler, J., et al. (2019). Deep learning and process understanding for data-driven Earth system science. *Nature*, DOI: 10.1038/s41586-019-0912-1.

Riahi, K., van Vuuren, D. P., Kriegler, E., Edmonds, J., O'Neill, B. C., et al. (2017). The Shared Socioeconomic Pathways and their energy, land use, and greenhouse gas emissions implications: An overview. *Global Environmental Change*, DOI: 10.1016/j.gloenvcha.2016.05.009.

Richardson, S. and Green, P. J. (1997). On bayesian analysis of mixtures with an unknown number of components. *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, DOI: 10.1111/1467-9868.00095.

Richardson, T. B., Forster, P. M., Smith, C. J., Maycock, A. C., Wood, T., et al. (2019). Efficacy of Climate Forcings in PDRMIP Models. *Journal of Geophysical Research: Atmospheres*, DOI: 10.1029/2019JD030581.

Ridley, J. K., Blockley, E. W., Keen, A. B., Rae, J. G. L., West, A. E., and Schroeder, D. (2018). The sea ice model component of HadGEM3-GC3.1. *Geoscientific Model Development*, 11(2):713–723, DOI: 10.5194/gmd-11-713-2018, https://www.geosci-model-dev.net/11/713/2018/.

Ring, M. J. and Plumb, R. A. (2008). The response of a simplified GCM to axisymmetric forcings: Applicability of the fluctuation-dissipation theorem. *Journal of the Atmospheric Sciences*, DOI: 10.1175/2008JAS2773.1.

Ripley, B. D. (1987). *Stochastic simulation.* Wiley series in probability and mathematical statistics. Applied probability and statistics. Wiley, ISBN: 9780471818847.

Robert, C. P. and Casella, G. (2004). Monte Carlo Statistical Methods Springer-Verlag.

Roberts, G. O. and Rosenthal, J. S. (2004). General state space markov chains and MCMC algorithms. *Probability Surveys*, DOI: 10.1214/154957804100000024.

Rogelj, J., Mccollum, D. L., O'Neill, B. C., and Riahi, K. (2013). 2020 emissions levels required to limit warming to below 2°C. *Nature Climate Change*, DOI: 10.1038/nclimate1758.

Rougier, J., Sexton, D. M., Murphy, J. M., and Stainforth, D. (2009). Analyzing the climate sensitivity of the HadSM3 climate model using ensembles from different but related experiments. *Journal of Climate*, DOI: 10.1175/2008JCLI2533.1.

Rugenstein, M., Bloch-Johnson, J., Gregory, J., Andrews, T., Mauritsen, T., et al. (2020). Equilibrium Climate Sensitivity Estimated by Equilibrating Climate Models. *Geophysical Research Letters*, DOI: 10.1029/2019GL083898.

Runge, J., Nowack, P., Kretschmer, M., Flaxman, S., and Sejdinovic, D. (2019). Detecting and quantifying causal associations in large nonlinear time series datasets. *Science Advances*, DOI: 10.1126/sciadv.aau4996.

Ryan, E., Wild, O., Voulgarakis, A., and Lee, L. (2018). Fast sensitivity analysis methods for computationally expensive models with multi-dimensional output. *Geoscientific Model Development*, 11:3131–3146, DOI: 10.5194/gmd-11-3131-2018.

Saltelli, A., Aleksankina, K., Becker, W., Fennell, P., Ferretti, F., et al. (2019). Why so many published sensitivity analyses are false: A systematic review of sensitivity analysis practices. *Environmental Modelling and Software*, DOI: 10.1016/j.envsoft.2019.01.012.

Saltelli, A. and Annoni, P. (2010). How to avoid a perfunctory sensitivity analysis. *Environmental Modelling and Software*, DOI: 10.1016/j.envsoft.2010.04.012.

Saltelli, A., Annoni, P., Azzini, I., Campolongo, F., Ratto, M., and Tarantola, S. (2010). Variance based sensitivity analysis of model output. Design and estimator for the total sensitivity index. *Computer Physics Communications*, DOI: 10.1016/j.cpc.2009.09.018.

Saltelli, A., Ratto, M., Andres, T., Campolongo, F., Cariboni, J., et al. (2008). *Global Sensitivity Analysis. The Primer.* ISBN: 9780470059975, DOI: 10.1002/9780470725184.

Salter, J. M. and Williamson, D. (2016). A comparison of statistical emulation methodologies for multi-wave calibration of environmental models. *Environmetrics*, DOI: 10.1002/env.2405.

Samset, B. H., Myhre, G., Forster, P. M., Hodnebrog, Andrews, T., et al. (2016). Fast and slow precipitation responses to individual climate forcers: A PDRMIP multimodel study. *Geophysical Research Letters*, 43(6):2782–2791, DOI: 10.1002/2016GL068064, https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1002/2016GL068064.

Samuel, A. L. (1959). Some Studies in Machine Learning. *IBM Journal of Research and Development*.

Santer, B. D., Wigley, T. M. L., Schlesinger, M. E., and Mitchell, J. F. B. (1990). Developing climate scenarios from equilibrium GCM results.

Santner, T. J., Williams, B. J., Notz, W. I., and Williams, B. J. (2003). *The design and analysis of computer experiments*, volume 1. Springer.

Scher, S. (2018). Toward Data-Driven Weather and Climate Forecasting: Approximating a Simple General Circulation Model With Deep Learning. *Geophysical Research Letters*, DOI: 10.1029/2018GL080704.

Shawki, D., Voulgarakis, A., Chakraborty, A., Kasoar, M., and Srinivasan, J. (2018). The South Asian Monsoon Response to Remote Aerosols: Global and Regional Mechanisms. *Journal of Geophysical Research: Atmospheres*, DOI: 10.1029/2018JD028623.

Shindell, D. and Faluvegi, G. (2009). Climate response to regional radiative forcing during the twentieth century. *Nature Geoscience*, 2:294 EP –, DOI: 10.1038/ngeo473, https://doi.org/10.1038/ngeo473.

Shindell, D. T., Lamarque, J. F., Schulz, M., Flanner, M., Jiao, C., et al. (2013). Radiative forcing in the ACCMIP historical and future climate simulations. *Atmospheric Chemistry and Physics*, DOI: 10.5194/acp-13-2939-2013.

Shindell, D. T., Miller, R. L., Schmidt, G. A., and Pandolfo, L. (1999). Simulation of recent northern winter climate trends by greenhouse-gas forcing. *Nature*, DOI: 10.1038/20905.

Shindell, D. T., Voulgarakis, A., Faluvegi, G., and Milly, G. (2012). Precipitation response to regional radiative forcing. *Atmospheric Chemistry and Physics*, 12(15):6969–6982, DOI: 10.5194/acp-12-6969-2012, https://www.atmos-chem-phys.net/12/6969/2012/.

Shine, K. P., Cook, J., Highwood, E. J., and Joshi, M. M. (2003). An alternative to radiative forcing for estimating the relative importance of climate change mechanisms. *Geophysical Research Letters*, DOI: 10.1029/2003GL018141.

Shine, K. P., Fuglestvedt, J. S., Hailemariam, K., and Stuber, N. (2005). Alternatives to the Global Warming Potential for comparing climate impacts of emissions of greenhouse gases. *Climatic Change*, DOI: 10.1007/s10584-005-1146-9.

Sippel, S., Meinshausen, N., Merrifield, A., Lehner, F., Pendergrass, A. G., et al. (2019). Uncovering the Forced Climate Response from a Single Ensemble Member Using Statistical Learning. *Journal of Climate*, DOI: 10.1175/jcli-d-18-0882.1.

Smagorinsky, J., Manabe, S., and Holloway, J. L. (1965). Numerical results from a nine-level general circulation model of the atmosphere. *Monthly Weather Review*, DOI: 10.1175/1520-0493(1965)093<0727:nrfanl>2.3.co;2.

Smith, C. J., Forster, P. M., Allen, M., Leach, N., Millar, R. J., et al. (2018a). FAIR v1.3: A simple emissions-based impulse response and carbon cycle model. *Geoscientific Model Development*, DOI: 10.5194/gmd-11-2273-2018.

Smith, C. J., Kramer, R. J., Myhre, G., Forster, P. M., Soden, B. J., et al. (2018b). Understanding Rapid Adjustments to Diverse Forcing Agents. *Geophysical Research Letters*, DOI: 10.1029/2018GL079826.

Smith, R. L., Tebaldi, C., Nychka, D., and Mearns, L. O. (2009). Bayesian modeling of uncertainty in ensembles of climate models. *Journal of the American Statistical Association*, DOI: 10.1198/jasa.2009.0007.

Souza, A. N., Wagner, G. L., Ramadhan, A., Allen, B., Churavy, V., et al. (2020). Uncertainty Quantification of Ocean Parameterizations: Application to the K-Profile-Parameterization for Penetrative Convection. *Journal of Advances in Modeling Earth Systems*, DOI: 10.1029/2020MS002108.

Stocker, T. F., Qin, D., Plattner, G. K., Tignor, M. M., Allen, S. K., et al. (2013). *Climate change 2013 the physical science basis: Working Group I contribution to the fifth assessment report of the intergovernmental panel on climate change*. ISBN: 9781107415324, DOI: 10.1017/CBO9781107415324.

Stohl, A., Aamaas, B., Amann, M., Baker, L. H., Bellouin, N., et al. (2015). Evaluating the climate and air quality impacts of short-lived pollutants. *Atmospheric Chemistry and Physics*, 15(18):10529–10566, DOI: 10.5194/acp-15-10529-2015, https://www.atmos-chem-phys.net/15/10529/2015/.

Stone, M. (1974). Cross-Validatory Choice and Assessment of Statistical Predictions. *Journal of the Royal Statistical Society: Series B (Methodological)*, DOI: 10.1111/j.2517-6161.1974.tb00994.x.

Storkey, D., Blaker, A. T., Mathiot, P., Megann, A., Aksenov, Y., et al. (2018). UK Global Ocean GO6 and GO7: a traceable hierarchy of model resolutions. *Geoscientific Model Development*, 11(8):3187–3213, DOI: 10.5194/gmd-11-3187-2018, https://www.geosci-model-dev.net/11/3187/2018/.

Sutton, R. T., Dong, B., and Gregory, J. M. (2007). Land/sea warming ratio in response to climate change: IPCC AR4 model results and comparison with observations. *Geophysical Research Letters*, DOI: `10.1029/2006GL028164`.

Taylor, K. E., Stouffer, R. J., and Meehl, G. A. (2012). An overview of CMIP5 and the experiment design. DOI: `10.1175/BAMS-D-11-00094.1`.

Tebaldi, C. and Arblaster, J. M. (2014). Pattern scaling: Its strengths and limitations, and an update on the latest model simulations. *Climatic Change*, 122(3):459–471, DOI: `10.1007/s10584-013-1032-9`, `https://doi.org/10.1007/s10584-013-1032-9`.

Thomas, C., Voulgarakis, A., Lim, G., Haigh, J., and Nowack, P. (2021). An unsupervised learning approach to identifying blocking events: the case of European summer. *Weather and Climate Dynamics Discussions*, 2021:1–34, DOI: `10.5194/wcd-2021-1`.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288.

Tierney, L. and Kadane, J. B. (1986). Accurate approximations for posterior moments and marginal densities. *Journal of the American Statistical Association*, DOI: `10.1080/01621459.1986.10478240`.

Tipping, M. E. and Bishop, C. M. (1999). Probabilistic principal component analysis. *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, DOI: `10.1111/1467-9868.00196`.

Toms, B. A., Kashinath, K., Prabhat, and Yang, D. (2019). Deep Learning for Scientific Inference from Geophysical Data: The Madden-Julian Oscillation as a Test Case. *ArXiv e-prints*, `http://arxiv.org/abs/1902.04621`.

Toms, B. A., Kashinath, K., Prabhat, and Yang, D. (2020). Testing the Reliability of Interpretable Neural Networks in Geoscience Using the Madden-Julian Oscillation. *Geoscientific Model Development Discussions*, 2020:1–22, DOI: `10.5194/gmd-2020-152`, `https://gmd.copernicus.org/preprints/gmd-2020-152/`.

Tosca, M. G., Randerson, J. T., and Zender, C. S. (2013). Global impact of smoke aerosols from landscape fires on climate and the Hadley circulation. *Atmospheric Chemistry and Physics*, DOI: `10.5194/acp-13-5227-2013`.

Tran, G. T., Oliver, K. I. C., Sóbester, A., Toal, D. J. J., Holden, P. B., et al. (2016). Building a traceable climate model hierarchy with multi-level emulators. *Advances in Statistical Climatology, Meteorology and Oceanography*, DOI: `10.5194/ascmo-2-17-2016`.

UNFCCC (2021a). The Paris Agreement — UNFCCC. *https://unfccc.int/process-and-meetings/the-paris-agreement/the-paris-agreement, Date Accessed: 30/03/2021.*

UNFCCC (2021b). What is the United Nations Framework Convention on Climate Change? — UNFCCC. *https://unfccc.int/process-and-meetings/the-convention/what-is-the-united-nations-framework-convention-on-climate-change Date Accessed: 30/03/2021.*

Urrego-Blanco, J. R., Urban, N. M., Hunke, E. C., Turner, A. K., and Jeffery, N. (2016). Uncertainty quantification and global sensitivity analysis of the Los Alamos sea ice model. *Journal of Geophysical Research: Oceans*, DOI: `10.1002/2015JC011558`.

Van Der Werf, G. R., Randerson, J. T., Giglio, L., Collatz, G. J., Mu, M., et al. (2010). Global fire emissions and the contribution of deforestation, savanna, forest, agricultural, and peat fires (1997-2009). *Atmospheric Chemistry and Physics*, DOI: 10.5194/acp-10-11707-2010.

van Vuuren, D. P., Stehfest, E., Gernaat, D. E., Doelman, J. C., van den Berg, M., et al. (2017). Energy, land-use and greenhouse gas emissions trajectories under a green growth paradigm. *Global Environmental Change*, DOI: 10.1016/j.gloenvcha.2016.05.008.

Voulgarakis, A. and Field, R. D. (2015). Fire Influences on Atmospheric Composition, Air Quality and Climate. DOI: 10.1007/s40726-015-0007-z.

Walters, D., Baran, A. J., Boutle, I., Brooks, M., Earnshaw, P., et al. (2019). The Met Office Unified Model Global Atmosphere 7.0/7.1 and JULES Global Land 7.0 configurations. *Geoscientific Model Development*, 12(5):1909–1963, DOI: 10.5194/gmd-12-1909-2019, https://www.geosci-model-dev.net/12/1909/2019/.

Wang, J., Balaprakash, P., and Kotamarthi, R. (2019). Fast domain-aware neural network emulation of a planetary boundary layer parameterization in a numerical weather forecast model. *Geoscientific Model Development*, DOI: 10.5194/gmd-12-4261-2019.

Ward, D. S., Kloster, S., Mahowald, N. M., Rogers, B. M., Randerson, J. T., and Hess, P. G. (2012). The changing radiative forcing of fires: Global model estimates for past, present and future. *Atmospheric Chemistry and Physics*, DOI: 10.5194/acp-12-10857-2012.

Wasserman (2004). All of Statistics : A Concise Course in Statistical Inference Brief Contents. *Simulation*, ISBN: 0387402721, DOI: 10.1007/978-0-387-21736-9.

Watson-Parris, D. (2021). Machine learning for weather and climate are worlds apart. DOI: 10.1098/rsta.2020.0098.

Watterson, I. G. (2008). Calculation of probability density functions for temperature and precipitation change under global warming. *Journal of Geophysical Research: Atmospheres*, 113(D12), DOI: 10.1029/2007JD009254, https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2007JD009254.

Weber, T., Corotan, A., Hutchinson, B., Kravitz, B., and Link, R. (2020). Technical note: Deep learning for creating surrogate models of precipitation in Earth system models. *Atmospheric Chemistry and Physics*, DOI: 10.5194/acp-20-2303-2020.

Westervelt, D. M., Mascioli, N. R., Fiore, A. M., Conley, A. J., Lamarque, J. F., et al. (2020). Local and remote mean and extreme temperature response to regional aerosol emissions reductions. *Atmospheric Chemistry and Physics*, DOI: 10.5194/acp-20-3009-2020.

Wild, O., Voulgarakis, A., O'Connor, F., Lamarque, J. F., Ryan, E. M., and Lee, L. (2020). Global sensitivity analysis of chemistry-climate model budgets of tropospheric ozone and OH: Exploring model diversity. *Atmospheric Chemistry and Physics*, DOI: 10.5194/acp-20-4047-2020.

Wilkinson, R. D. (2010). Bayesian Calibration of Expensive Multivariate Computer Experiments. In *Large-Scale Inverse Problems and Quantification of Uncertainty*. ISBN: 9780470697436, DOI: 10.1002/9780470685853.ch10.

Williams, K. D., Copsey, D., Blockley, E. W., Bodas-Salcedo, A., Calvert, D., et al. (2018). The Met Office Global Coupled Model 3.0 and 3.1 (GC3.0 and GC3.1) Configurations. *Journal of Advances in Modeling Earth Systems*, 10(2):357–380, DOI: 10.1002/2017MS001115, https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1002/2017MS001115.

Williamson, D., Blaker, A. T., Hampton, C., and Salter, J. (2015). Identifying and removing structural biases in climate models with history matching. *Climate Dynamics*, DOI: 10.1007/s00382-014-2378-z.

Wills, R. C., Battisti, D. S., Armour, K. C., Schneider, T., and Deser, C. (2020a). Pattern Recognition Methods to Separate Forced Responses from Internal Variability in Climate Model Ensembles and Observations. *Journal of Climate*, DOI: 10.1175/JCLI-D-19-0855.1.

Wills, R. C., Schneider, T., Wallace, J. M., Battisti, D. S., and Hartmann, D. L. (2018). Disentangling Global Warming, Multidecadal Variability, and El Niño in Pacific Temperatures. *Geophysical Research Letters*, DOI: 10.1002/2017GL076327.

Wills, R. C. J., Sippel, S., and Barnes, E. A. (2020b). Separating forced and unforced components of climate change: The utility of pattern recognition methods in Large Ensembles and observations. *US CLIVAR Variations*, DOI: 10.5065/0DSY-WH17.

WMO (1975). *The Physical Basis of Climate and Climate Modelling*, volume 16. WMO/ICSU, Geneva.

Wood, J. M., Tataryn, D. J., and Gorsuch, R. L. (1996). Effects of under- and overextraction on principal axis factor analysis with varimax rotation. *Psychological Methods*, DOI: 10.1037/1082-989X.1.4.354.

Wu, J. L., Xiao, H., and Paterson, E. (2018). Physics-informed machine learning approach for augmenting turbulence models: A comprehensive framework. *Physical Review Fluids*, DOI: 10.1103/PhysRevFluids.3.074602.

Xie, S. P., Lu, B., and Xiang, B. (2013). Similar spatial patterns of climate responses to aerosol and greenhouse gas changes. *Nature Geoscience*, DOI: 10.1038/ngeo1931.

Yadav, N., Ravela, S., and Ganguly, A. R. (2020). Machine learning for robust identification of complex nonlinear dynamical systems: Applications to earth systems modeling.

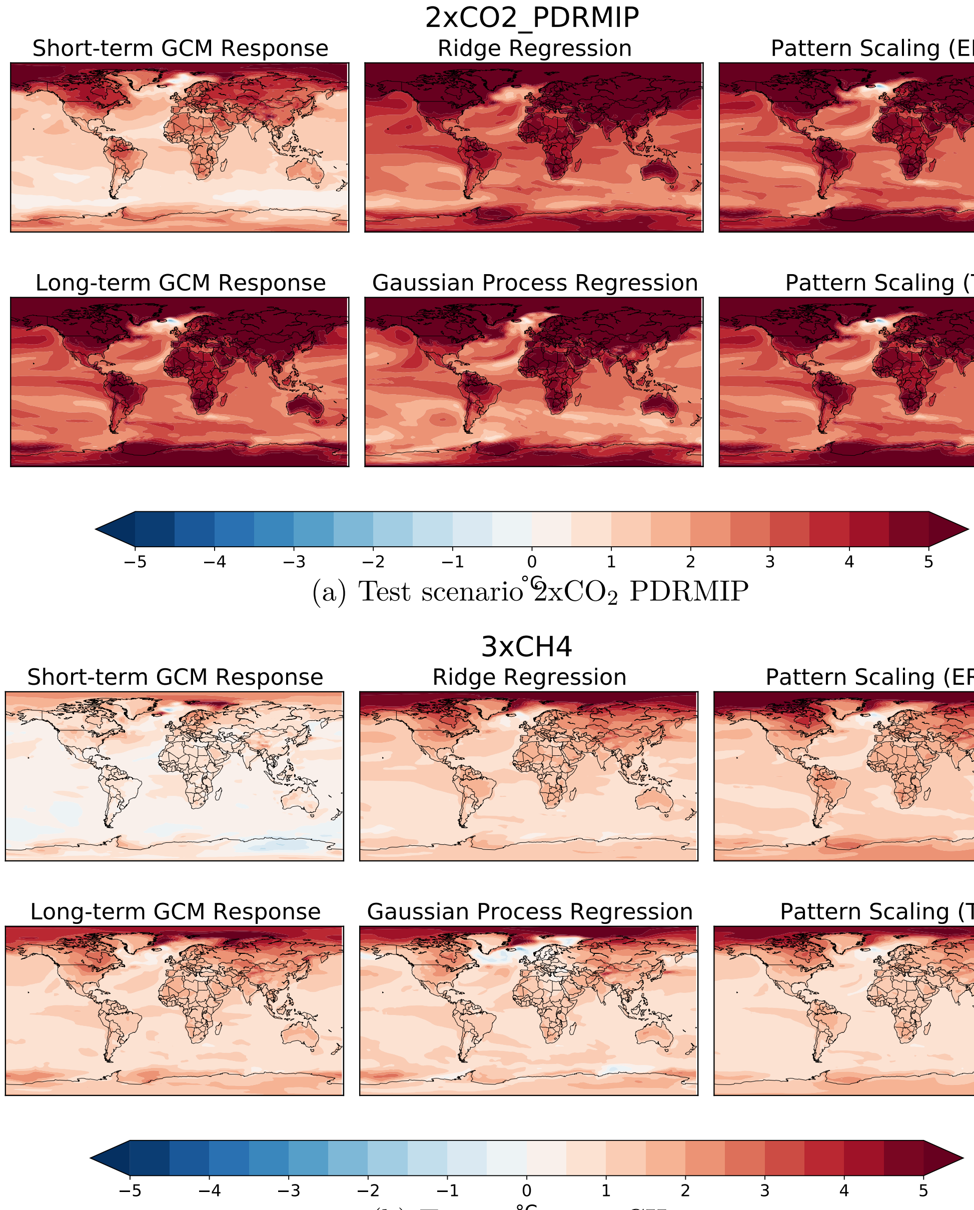
Zelazowski, P., Huntingford, C., Mercado, L. M., and Schaller, N. (2018). Climate pattern-scaling set for an ensemble of 22 GCMs – adding uncertainty to the IMOGEN version 2.0 impact system. *Geoscientific Model Development*, 11(2):541–560, DOI: 10.5194/gmd-11-541-2018, https://www.geosci-model-dev.net/11/541/2018/.

Zhou, Y., Johansen, A. M., and Aston, J. A. (2016). Toward Automatic Model Comparison: An Adaptive Sequential Monte Carlo Approach. *Journal of Computational and Graphical Statistics*, DOI: 10.1080/10618600.2015.1060885.

Zhu, J. and Poulsen, C. J. (2020). On the Increase of Climate Sensitivity and Cloud Feedback With Warming in the Community Atmosphere Models. *Geophysical Research Letters*, DOI: 10.1029/2020GL089143.

Appendix A

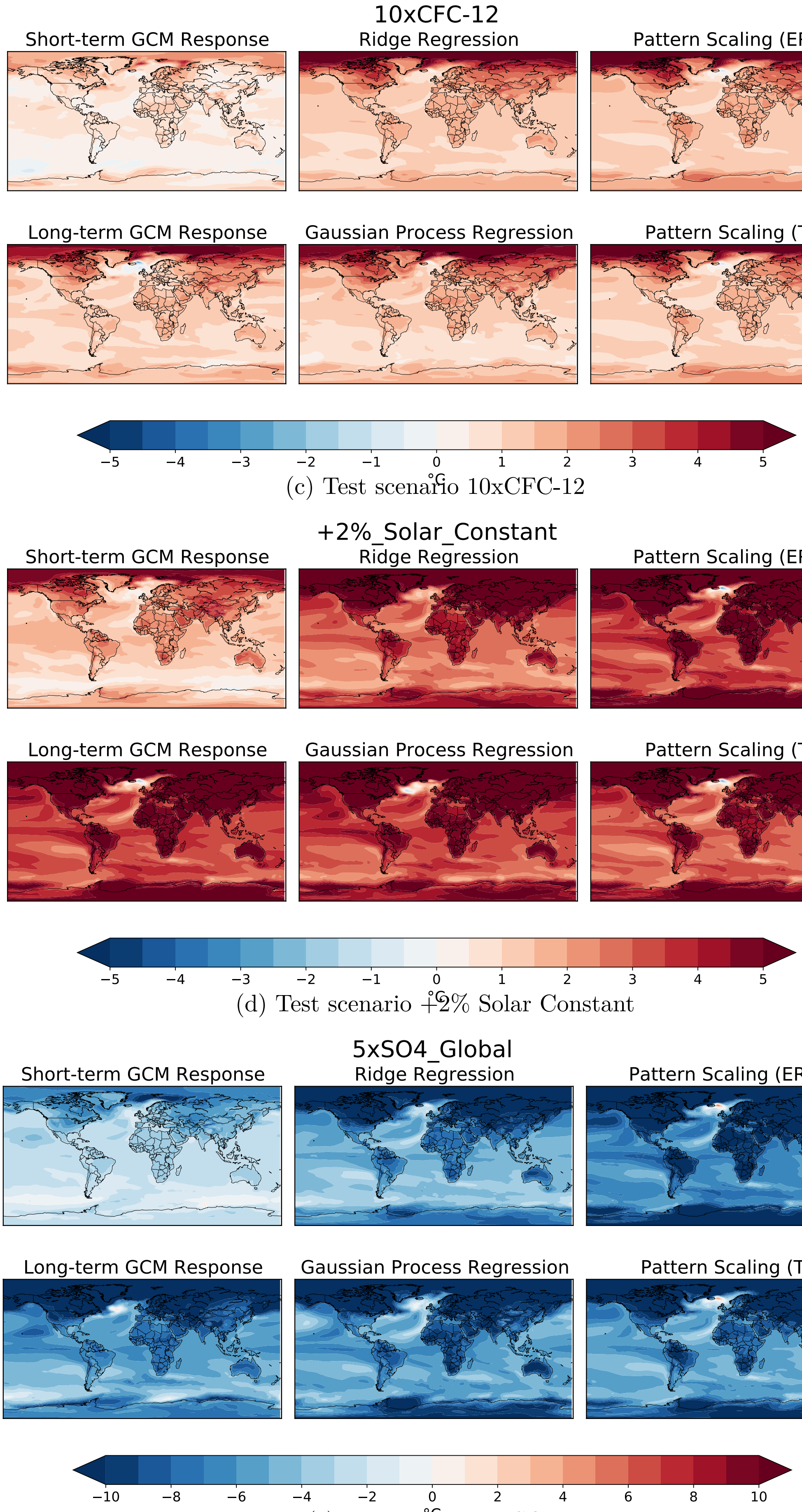
Additional Figures



(a) Test scenario $2xCO_2$ PDRMIP

(b) Test scenario $3xCH_4$

Figure A.1: (a-b): Short-term GCM response, long-term GCM response, machine learning predictions (Ridge and Gaussian process regression) and pattern scaling predictions estimated with ERF and the short-term global mean temperature response (T)

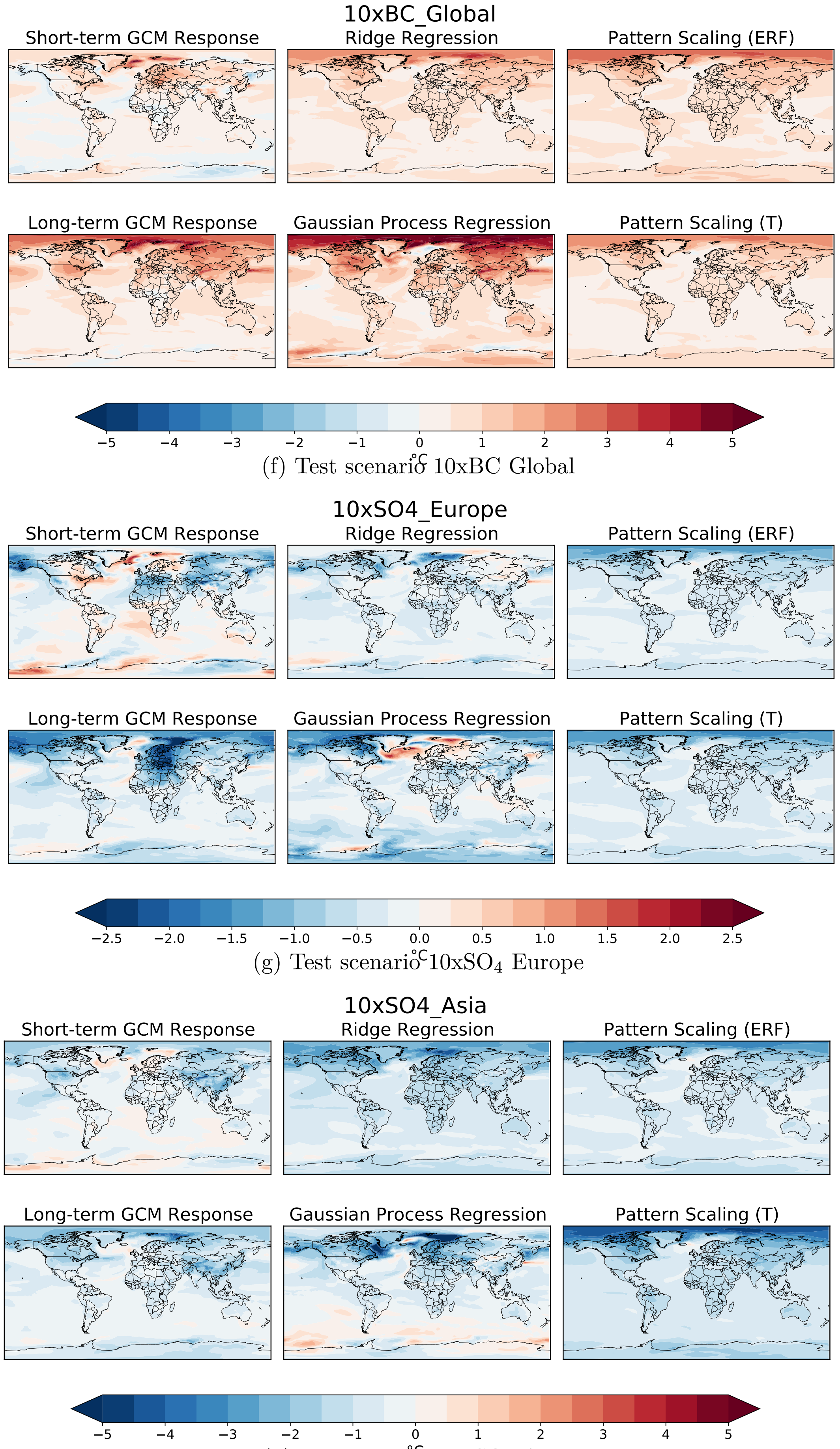


(c) Test scenario 10xCFC-12

(d) Test scenario +2% Solar Constant

(e) Test scenario 5xSO_4

Figure A.1: (c-e): Short-term GCM response, long-term GCM response, machine learning predictions (Ridge and Gaussian process regression) and pattern scaling predictions estimated with ERF and the short-term global mean temperature response (T)

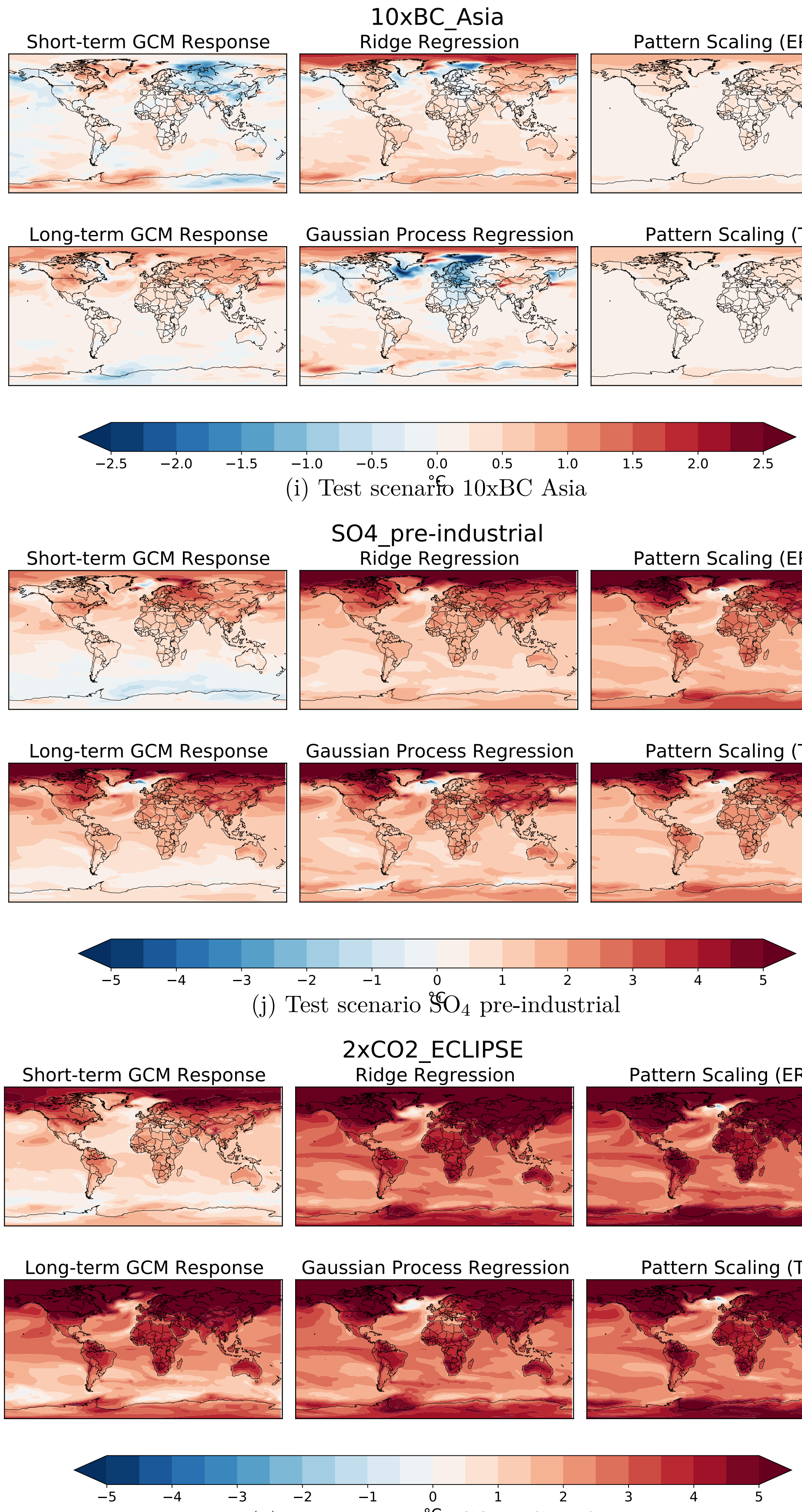


(f) Test scenario 10xBC Global

(g) Test scenario 10xSO$_4$ Europe

(h) Test scenario 10xSO$_4$ Asia

Figure A.1: (f-h): Short-term GCM response, long-term GCM response, machine learning predictions (Ridge and Gaussian process regression) and pattern scaling predictions estimated with ERF and the short-term global mean temperature response (T)

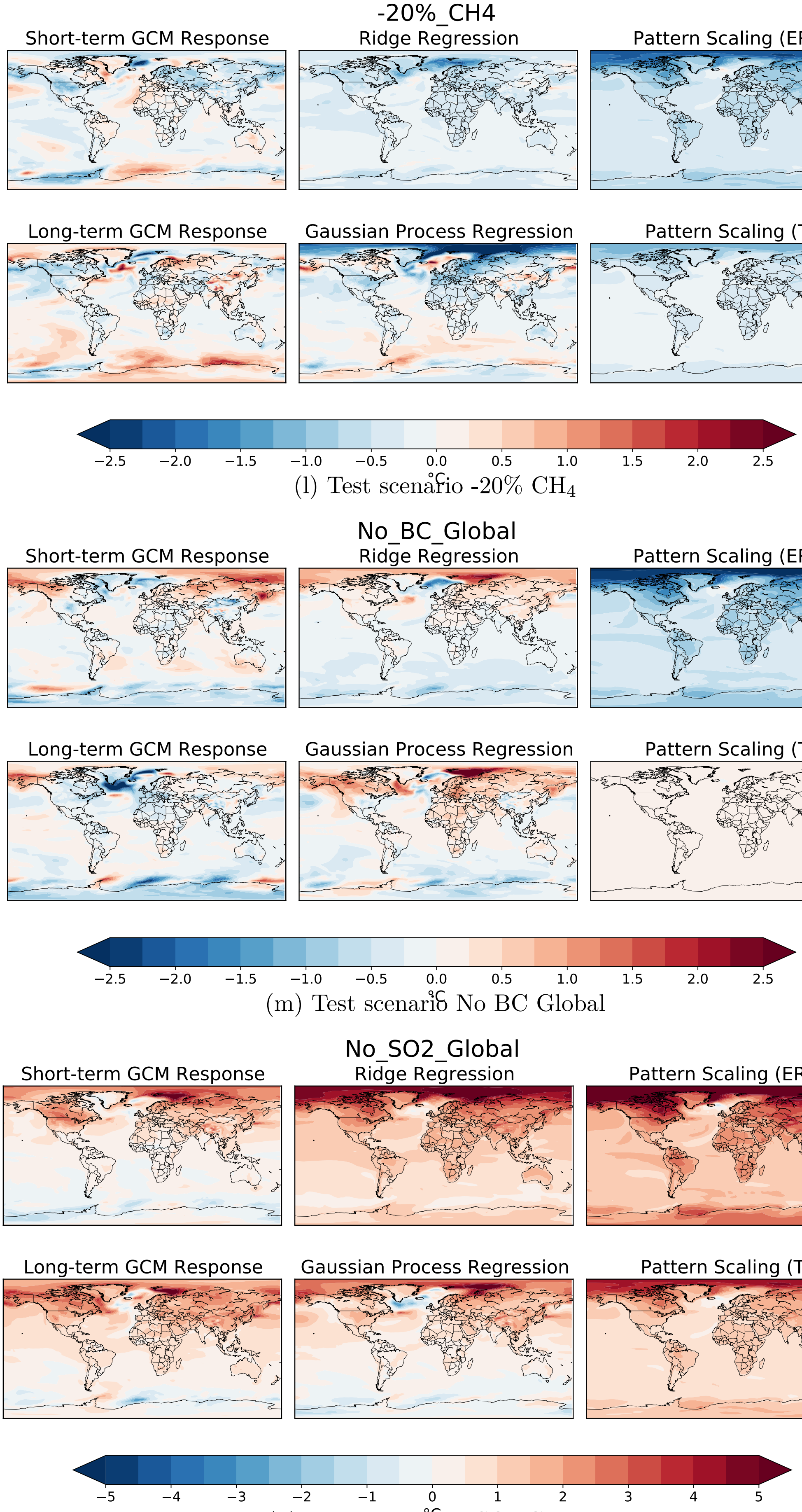


(i) Test scenario 10xBC Asia

(j) Test scenario SO_4 pre-industrial

(k) Test scenario $2xCO_2$ ECLIPSE

Figure A.1: (i-k): Short-term GCM response, long-term GCM response, machine learning predictions (Ridge and Gaussian process regression) and pattern scaling predictions estimated with ERF and the short-term global mean temperature response (T)

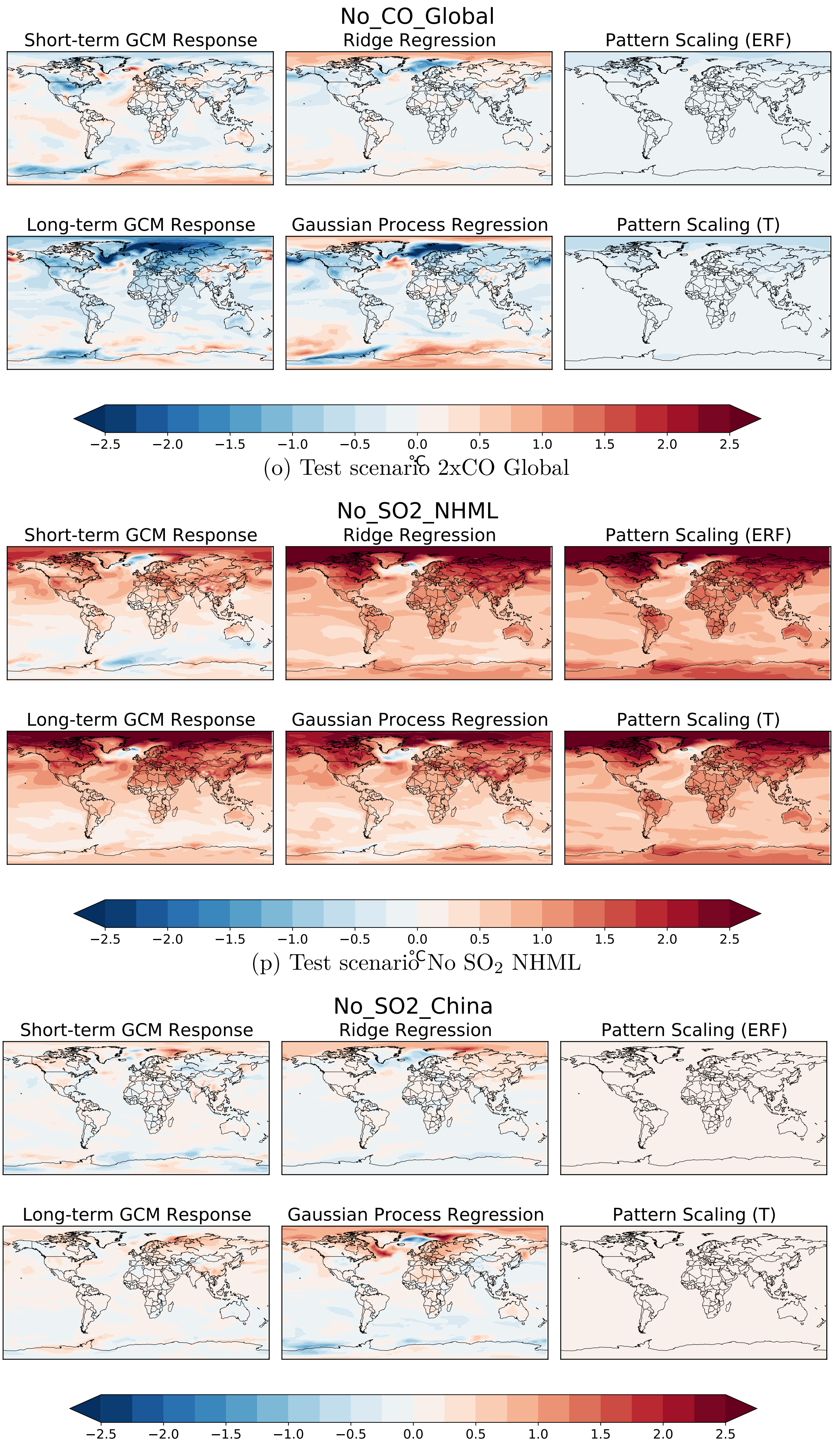


(l) Test scenario -20% CH$_4$

(m) Test scenario No BC Global

(n) Test scenario 2xSO$_2$ Global

Figure A.1: (l-n): Short-term GCM response, long-term GCM response, machine learning predictions (Ridge and Gaussian process regression) and pattern scaling predictions estimated with ERF and the short-term global mean temperature response (T)

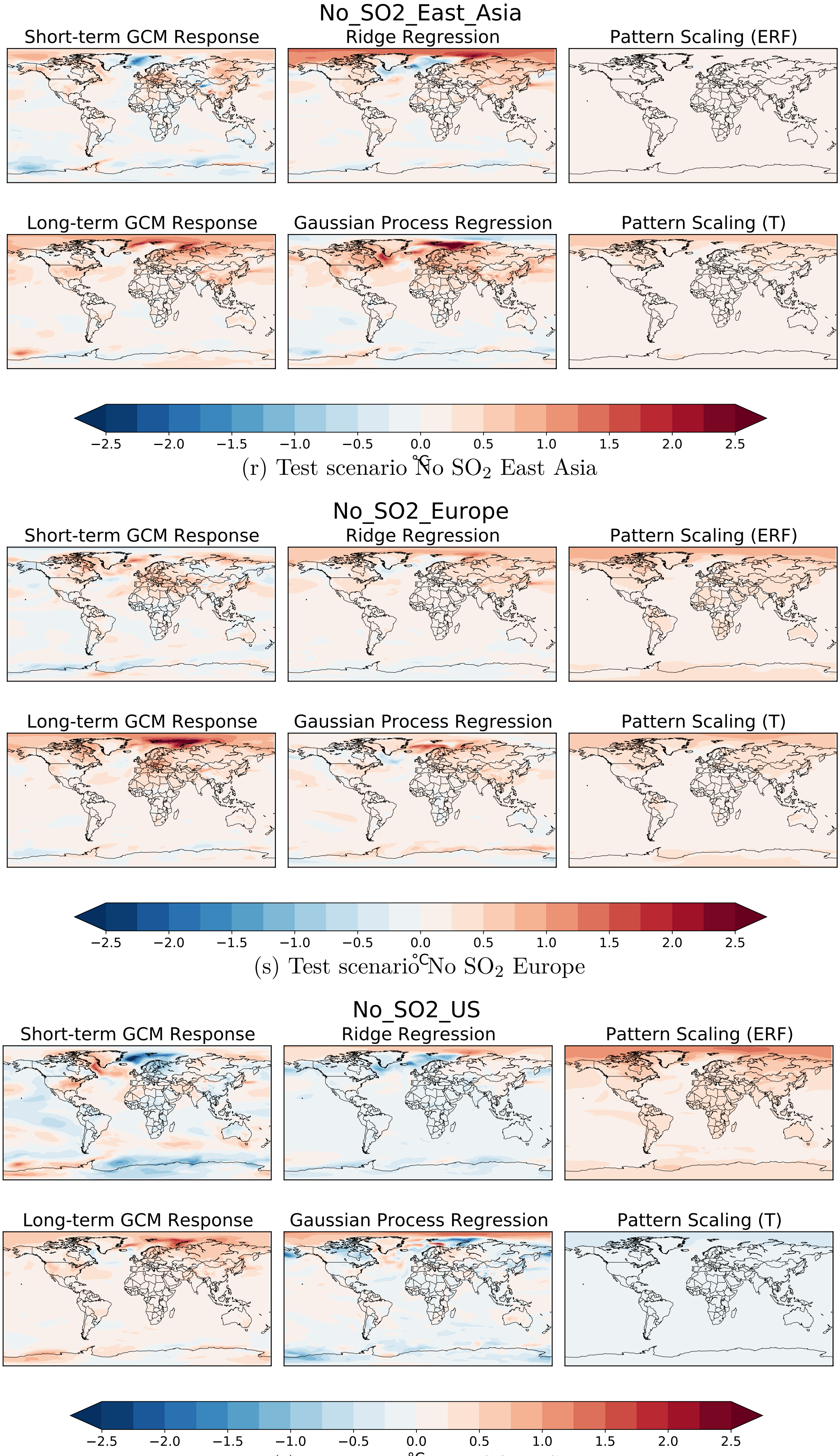


(o) Test scenario 2xCO Global

(p) Test scenario No SO_2 NHML

(q) Test scenario No SO_2 China

Figure A.1: (o-q): Short-term GCM response, long-term GCM response, machine learning predictions (Ridge and Gaussian process regression) and pattern scaling predictions estimated with ERF and the short-term global mean temperature response (T)

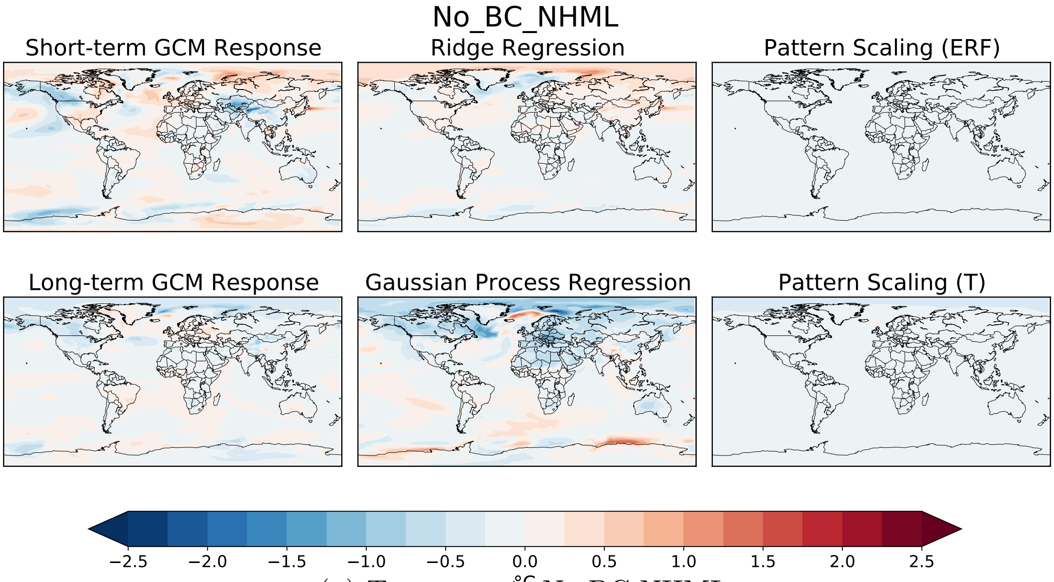


(r) Test scenario No SO_2 East Asia

(s) Test scenario No SO_2 Europe

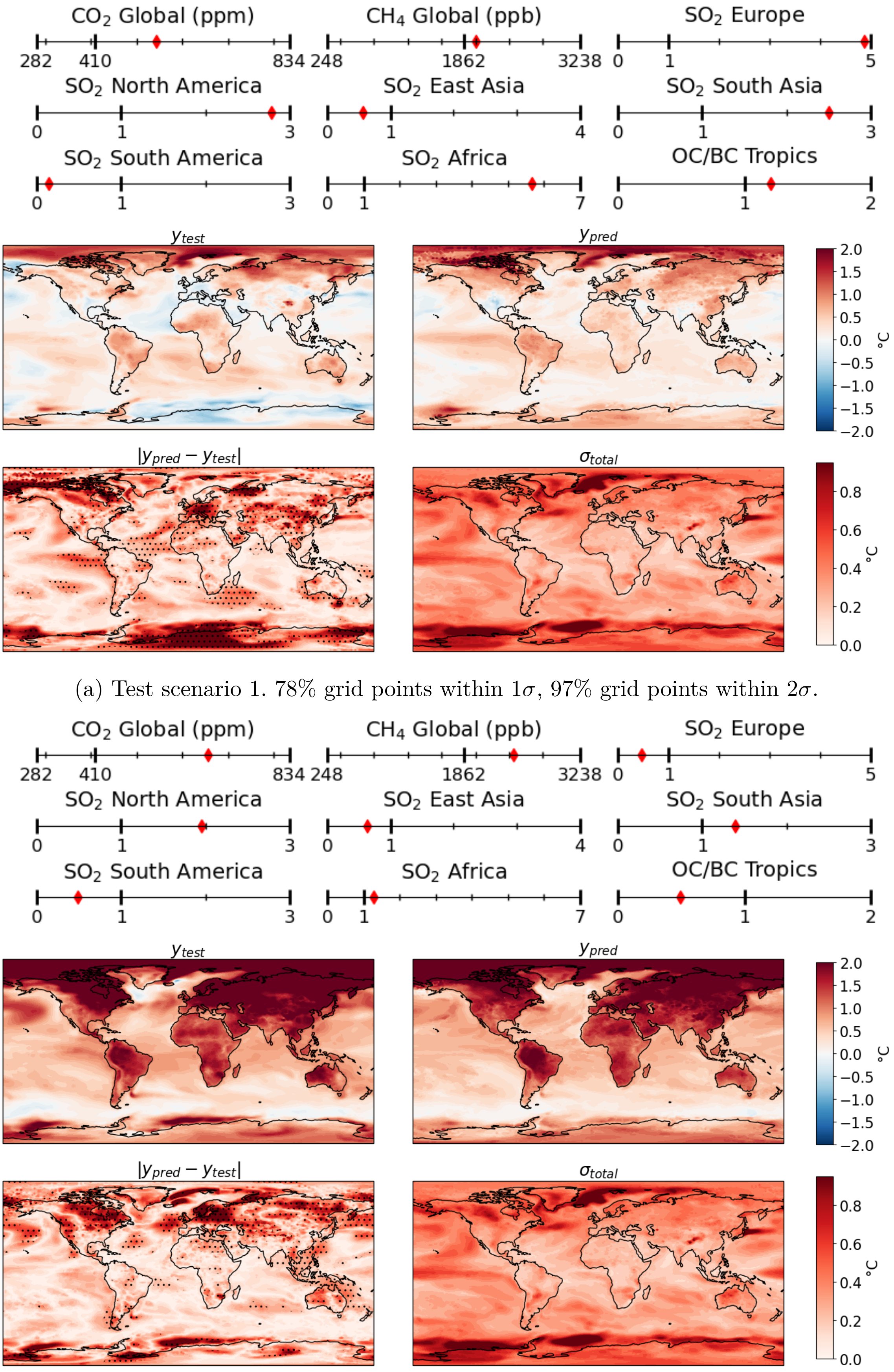
(t) Test scenario No SO_2 US

Figure A.1: (r-t): Short-term GCM response, long-term GCM response, machine learning predictions (Ridge and Gaussian process regression) and pattern scaling predictions estimated with ERF and the short-term global mean temperature response (T)



(u) Test scenario No BC NHML

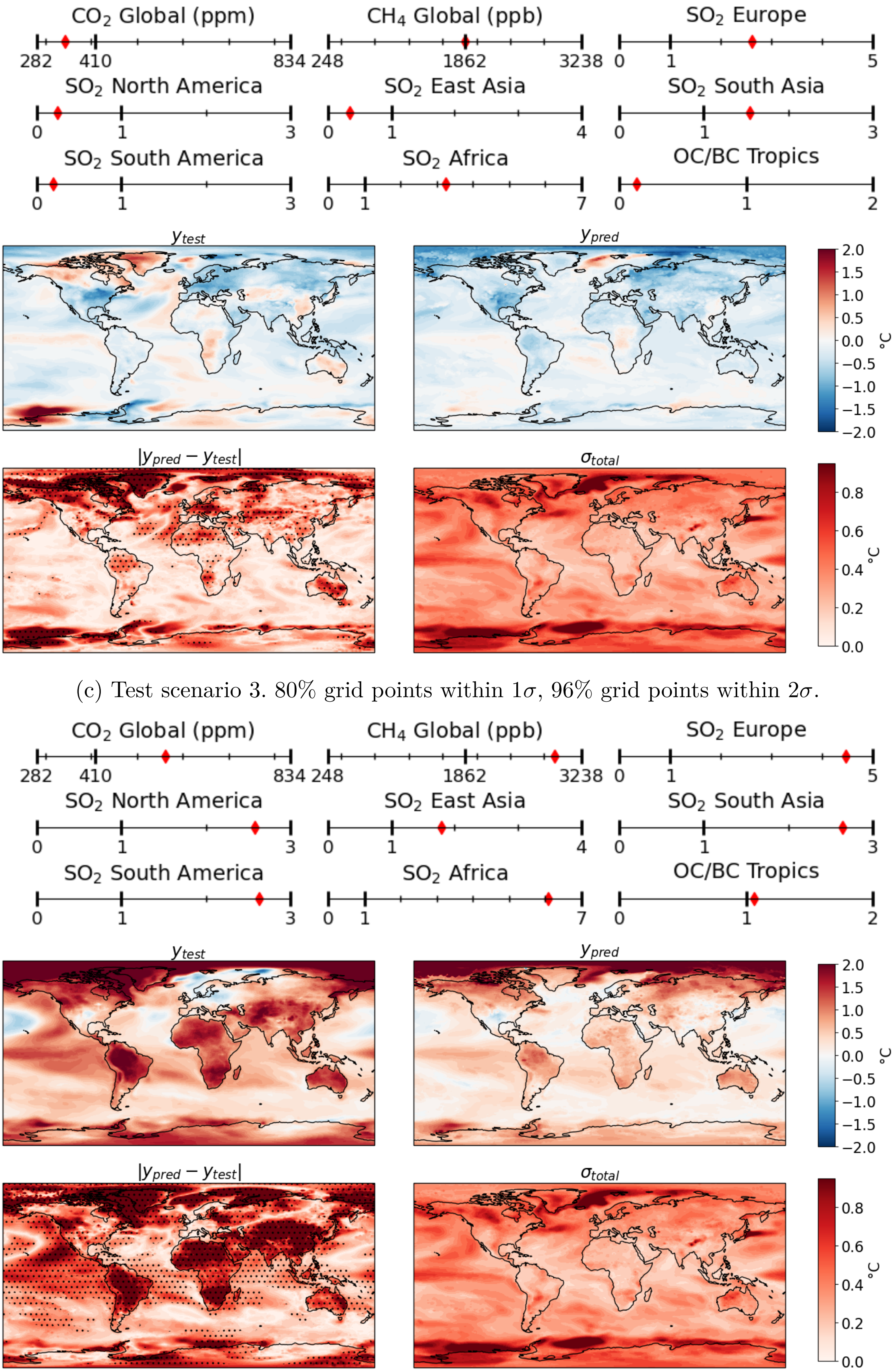
Figure A.1: (u): Short-term GCM response, long-term GCM response, machine learning predictions (Ridge and Gaussian process regression) and pattern scaling predictions estimated with ERF and the short-term global mean temperature response (T)



(a) Test scenario 1. 78% grid points within 1σ, 97% grid points within 2σ.

(b) Test scenario 2. 81% grid points within 1σ, 98% grid points within 2σ.

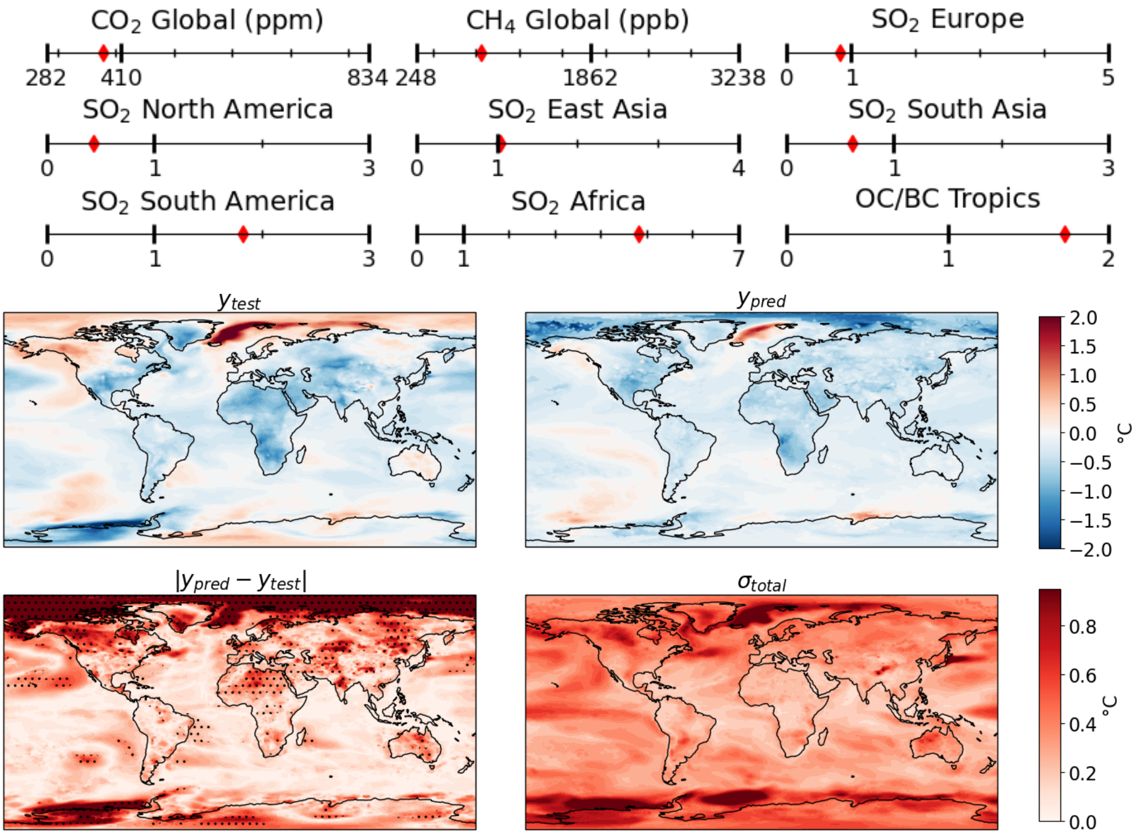
Figure A.2: (a-b): Test scenarios, also shown in Figure 4.6a-b. Top panel shows input values, relative to the baseline, minimum and maximum possible values. The top left map shows y_{test}, the true GCM prediction, the top right shows y_{pred}, the GP prediction, the bottom left shows the absolute error, $|y_{pred} - y_{test}|$ and the bottom right shows the 1 standard deviation from GP, σ_{total}. Stippling on the bottom left indicates $|y_{pred} - y_{test}| > \sigma_{total}$.



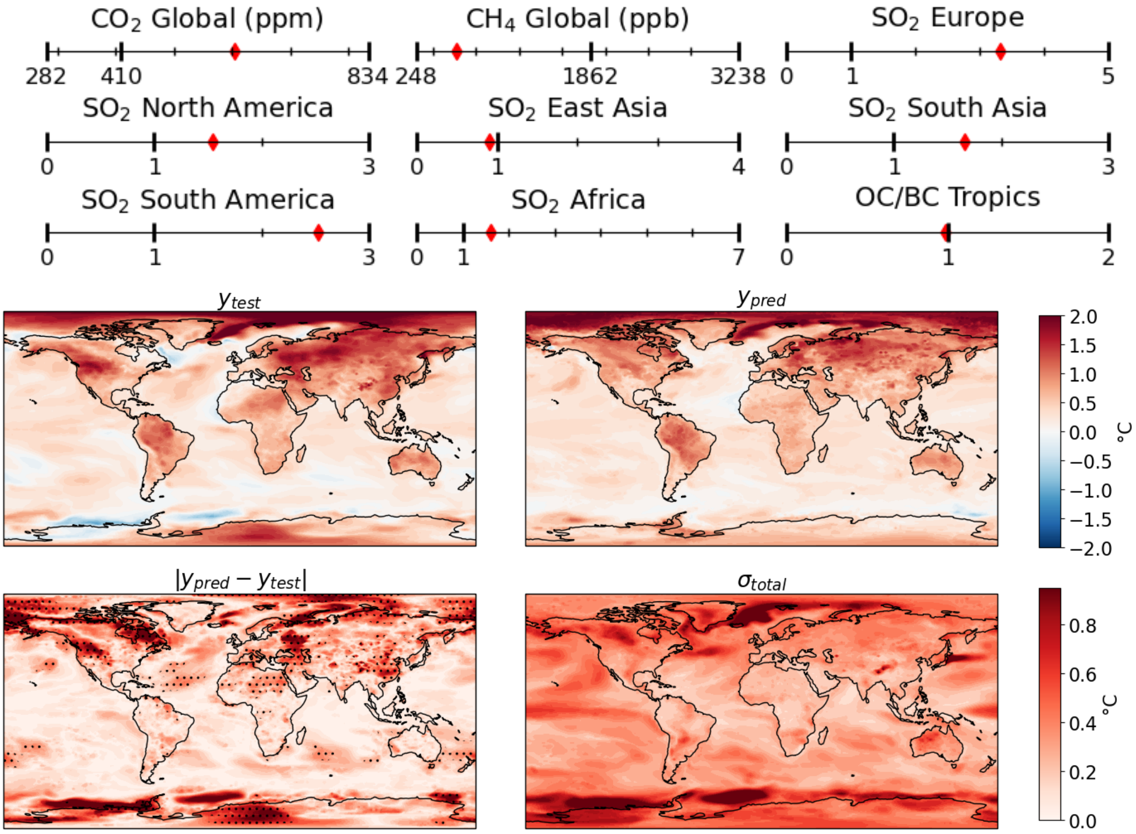
(c) Test scenario 3. 80% grid points within 1σ, 96% grid points within 2σ.

(d) Test scenario 4. 41% grid points within 1σ, 76% grid points within 2σ.

Figure A.2: (c-d): Test scenarios, also shown in Figure 4.6c-d Top panel shows input values, relative to the baseline, minimum and maximum possible values. The top left map shows y_{test}, the true GCM prediction, the top right shows y_{pred}, the GP prediction, the bottom left shows the absolute error, $|y_{pred} - y_{test}|$ and the bottom right shows the 1 standard deviation from GP, σ_{total}. Stippling on the bottom left indicates $|y_{pred} - y_{test}| > \sigma_{total}$.

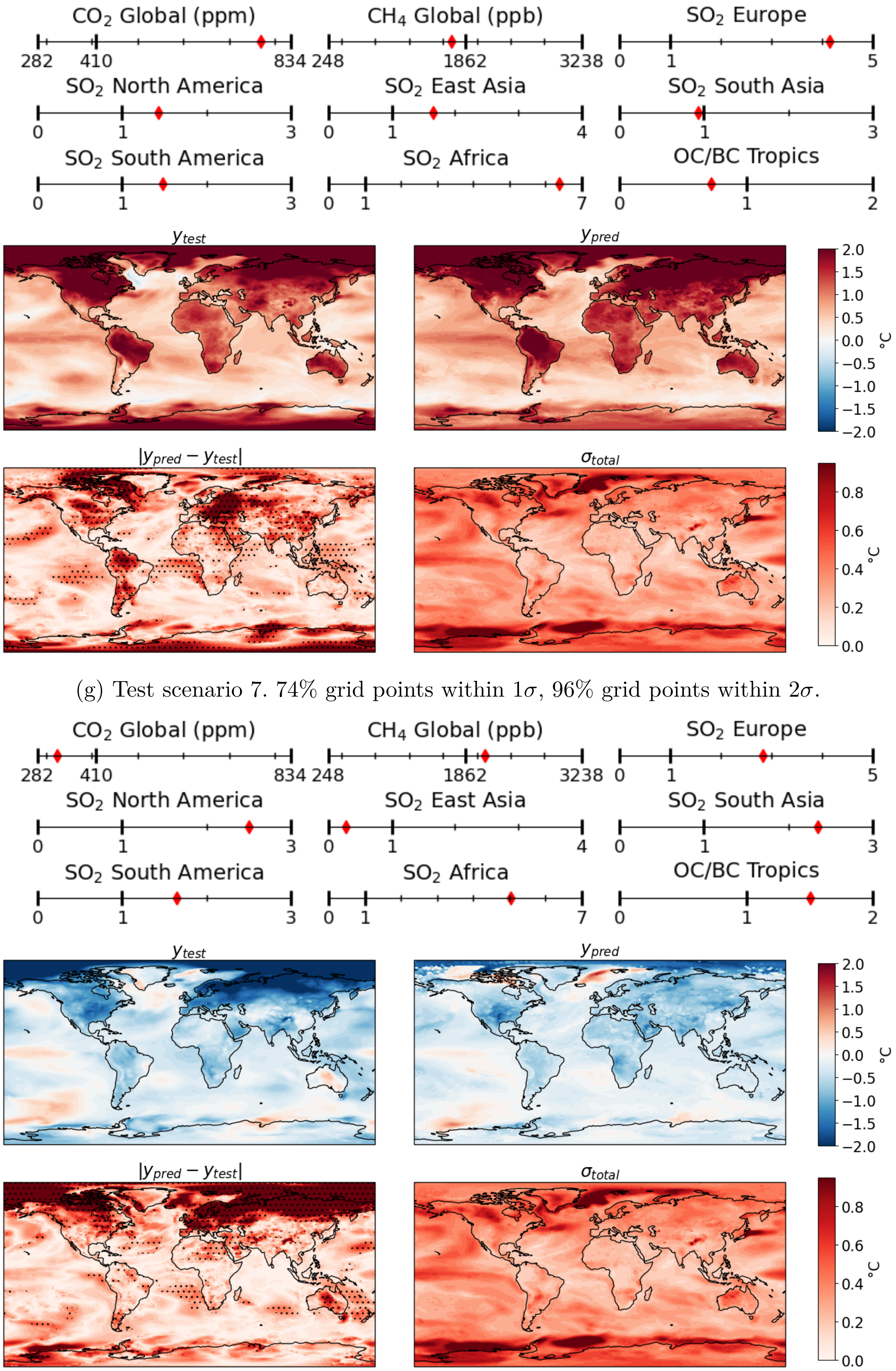


(e) Test scenario 5. 77% grid points within 1σ, 93% grid points within 2σ.



(f) Test scenario 6. 87% grid points within 1σ, 99% grid points within 2σ.

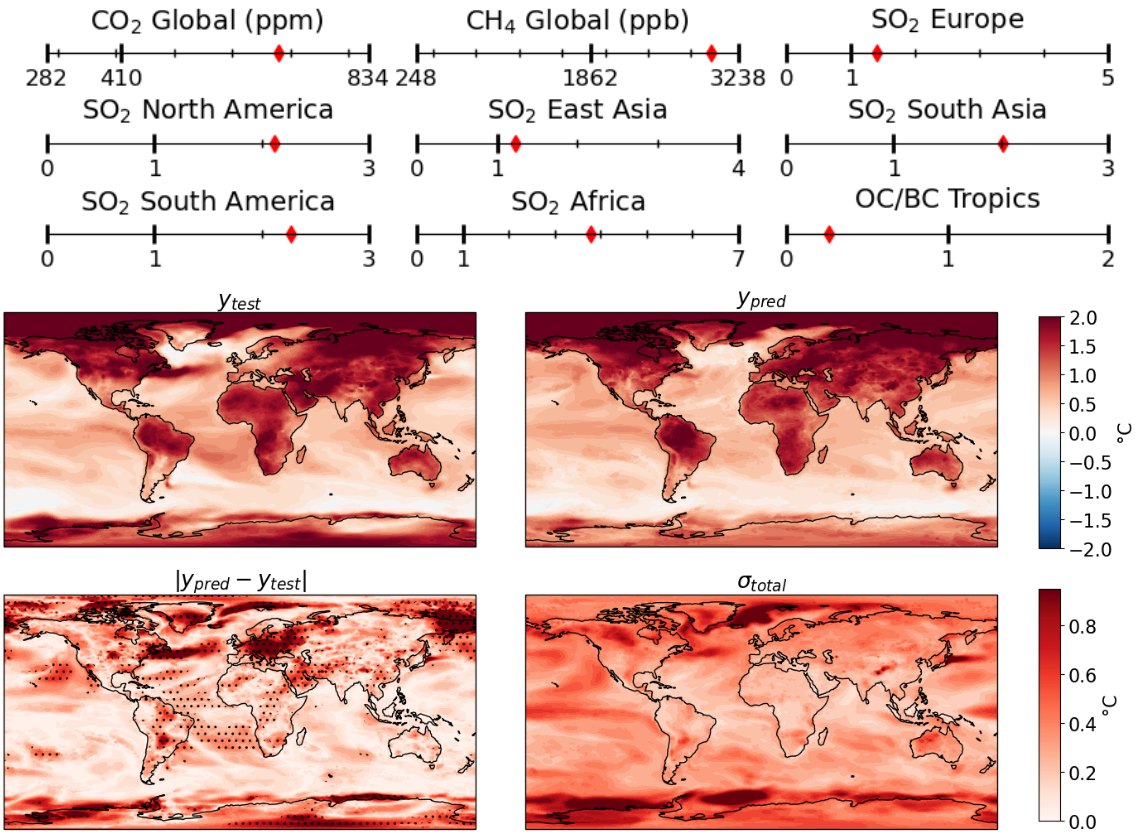
Figure A.2: (e-f): Additional test scenarios. Top panel shows input values, relative to the baseline, minimum and maximum possible values. The top left map shows y_{test}, the true GCM prediction, the top right shows y_{pred}, the GP prediction, the bottom left shows the absolute error, $|y_{pred} - y_{test}|$ and the bottom right shows the 1 standard deviation from GP, σ_{total}. Stippling on the bottom left indicates $|y_{pred} - y_{test}| > \sigma_{total}$.



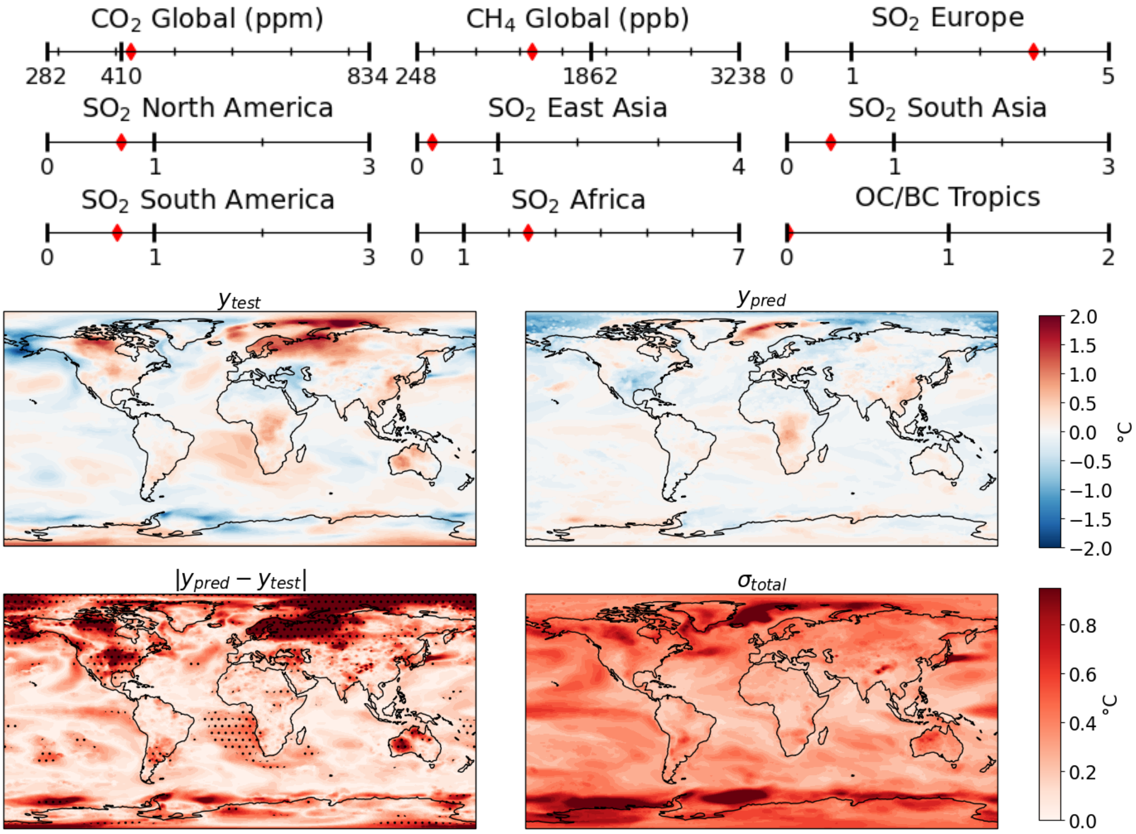
(g) Test scenario 7. 74% grid points within 1σ, 96% grid points within 2σ.

(h) Test scenario 8. 73% grid points within 1σ, 89% grid points within 2σ.

Figure A.2: (g-h): Additional test scenarios. Top panel shows input values, relative to the baseline, minimum and maximum possible values. The top left map shows y_{test}, the true GCM prediction, the top right shows y_{pred}, the GP prediction, the bottom left shows the absolute error, $|y_{pred} - y_{test}|$ and the bottom right shows the 1 standard deviation from GP, σ_{total}. Stippling on the bottom left indicates $|y_{pred} - y_{test}| > \sigma_{total}$.

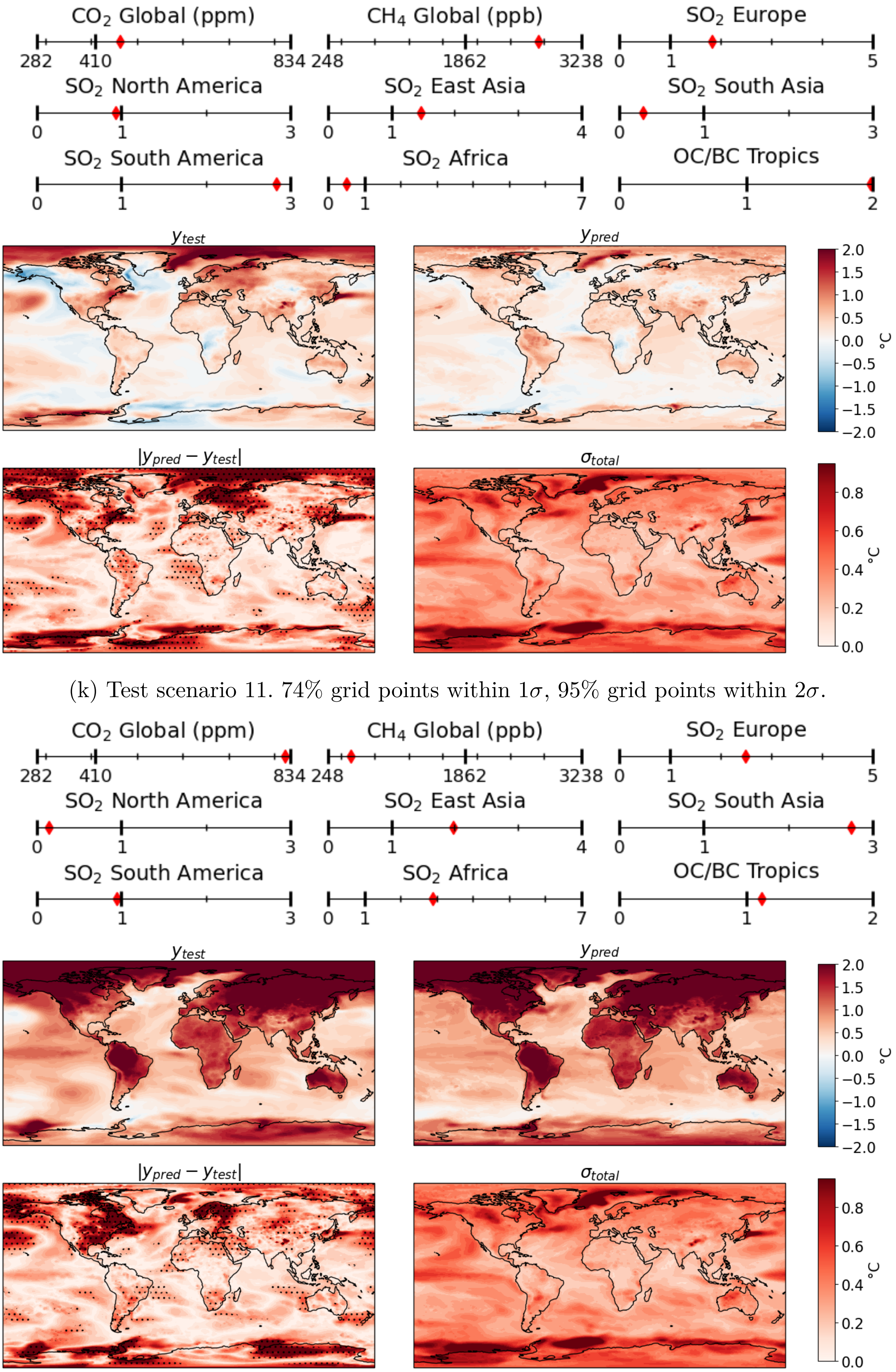


(i) Test scenario 9. 80% grid points within 1σ, 98% grid points within 2σ.



(j) Test scenario 10. 78% grid points within 1σ, 96% grid points within 2σ.

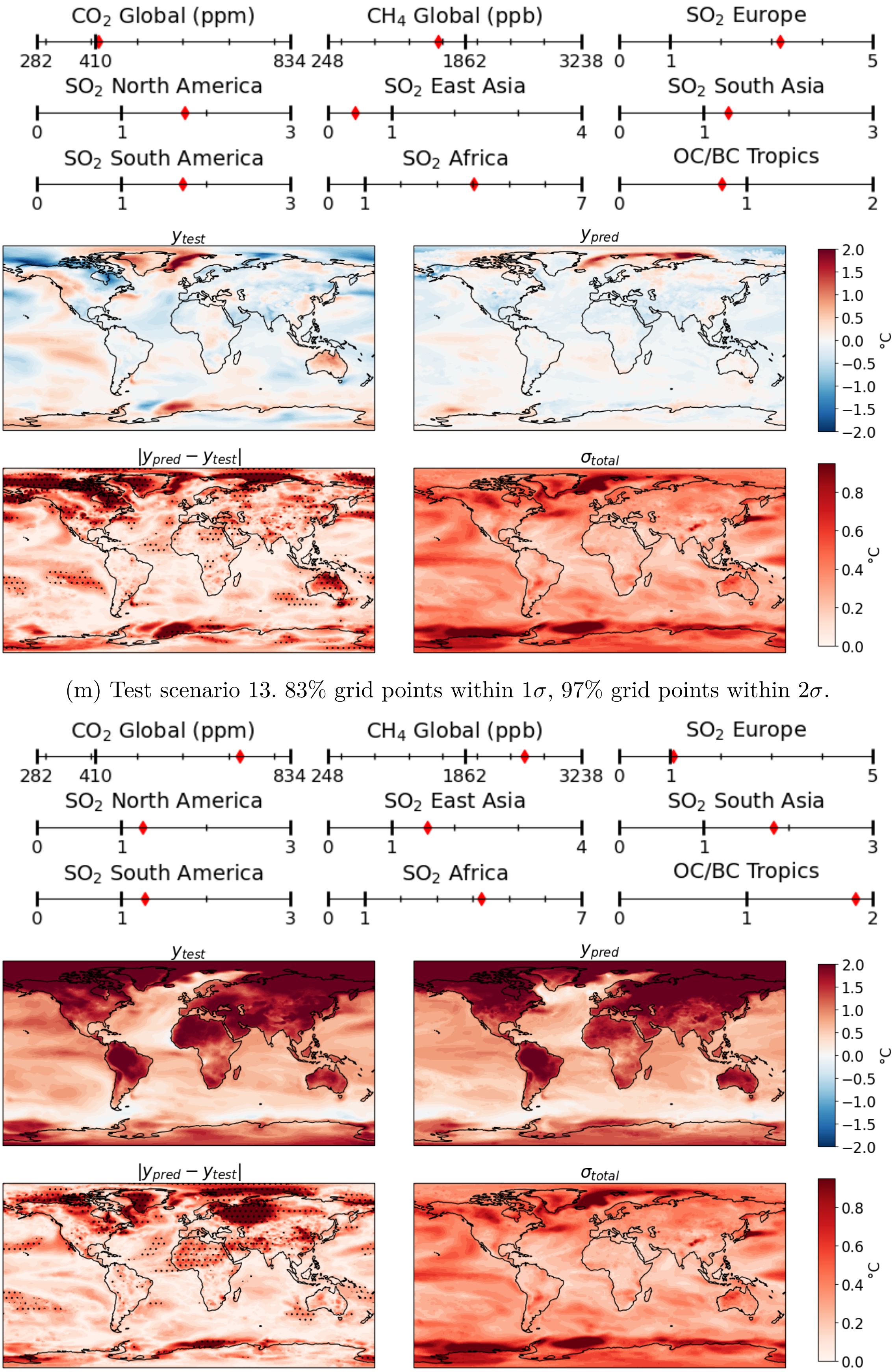
Figure A.2: (i-j): Additional test scenarios. Top panel shows input values, relative to the baseline, minimum and maximum possible values. The top left map shows y_{test}, the true GCM prediction, the top right shows y_{pred}, the GP prediction, the bottom left shows the absolute error, $|y_{pred} - y_{test}|$ and the bottom right shows the 1 standard deviation from GP, σ_{total}. Stippling on the bottom left indicates $|y_{pred} - y_{test}| > \sigma_{total}$.



(k) Test scenario 11. 74% grid points within 1σ, 95% grid points within 2σ.

(l) Test scenario 12. 77% grid points within 1σ, 98% grid points within 2σ.

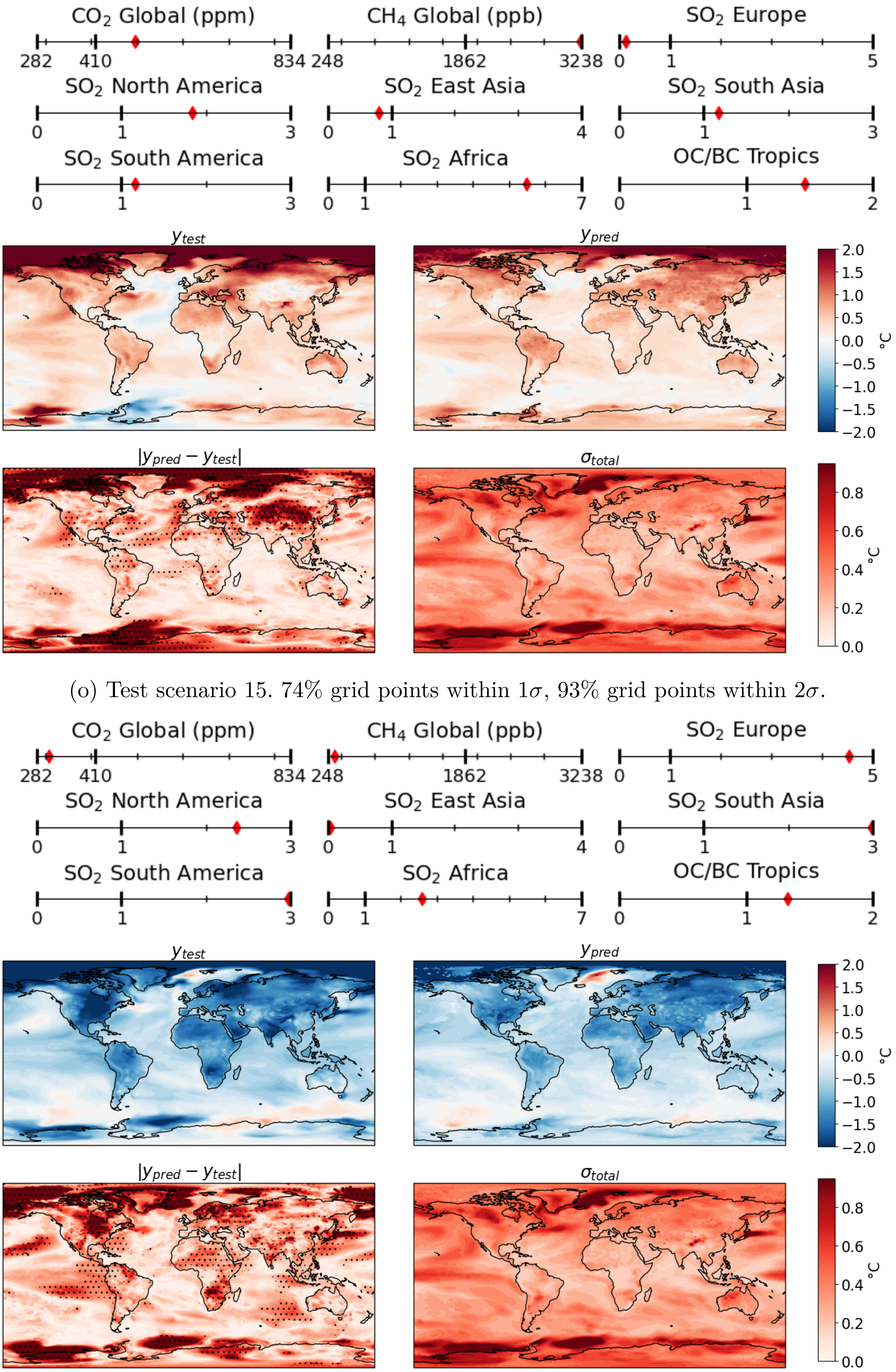
Figure A.2: (k-l): Additional test scenarios. Top panel shows input values, relative to the baseline, minimum and maximum possible values. The top left map shows y_{test}, the true GCM prediction, the top right shows y_{pred}, the GP prediction, the bottom left shows the absolute error, $|y_{pred} - y_{test}|$ and the bottom right shows the 1 standard deviation from GP, σ_{total}. Stippling on the bottom left indicates $|y_{pred} - y_{test}| > \sigma_{total}$.



(m) Test scenario 13. 83% grid points within 1σ, 97% grid points within 2σ.

(n) Test scenario 14. 83% grid points within 1σ, 98% grid points within 2σ.

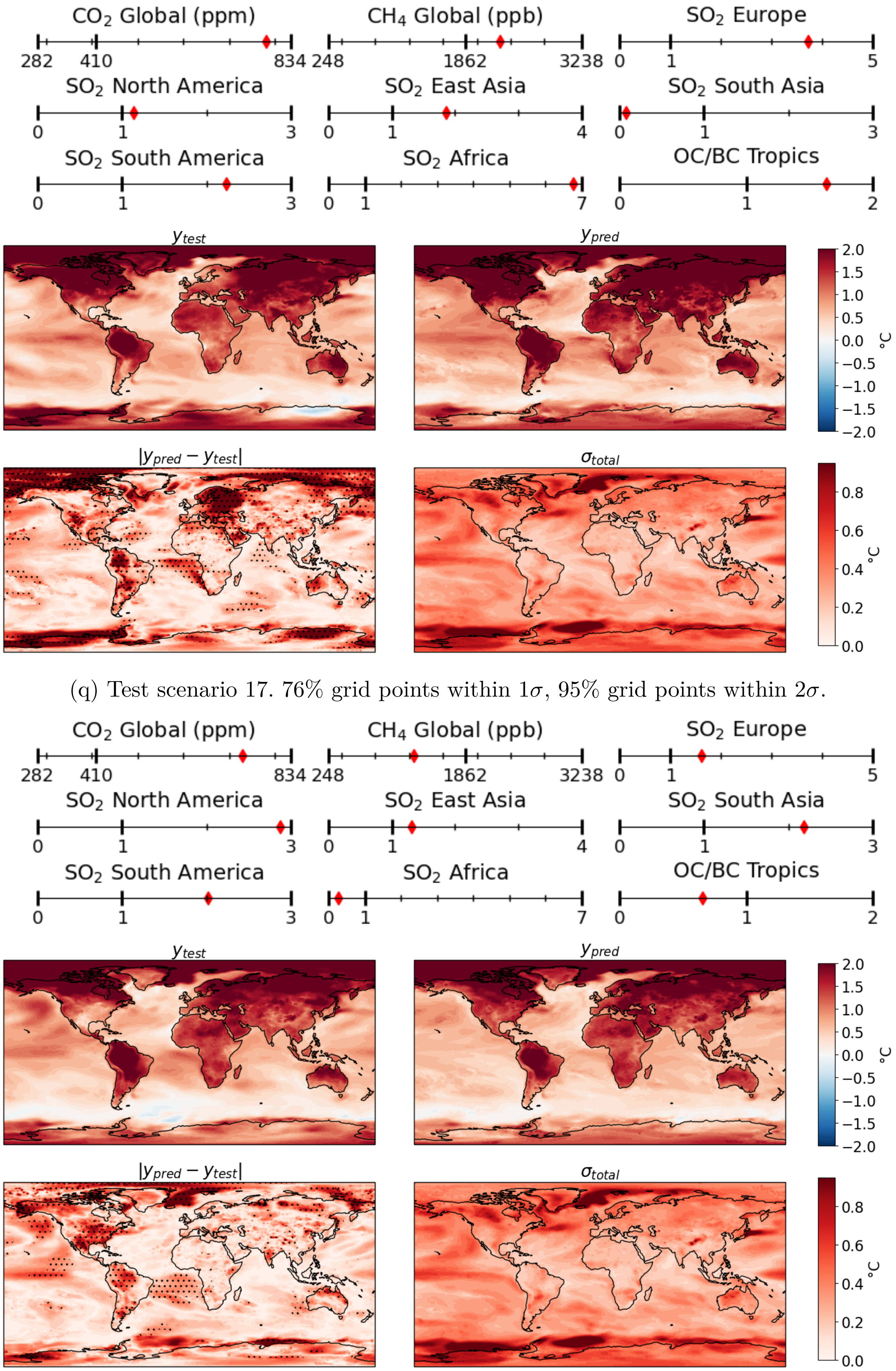
Figure A.2: (m-n): Additional test scenarios. Top panel shows input values, relative to the baseline, minimum and maximum possible values. The top left map shows y_{test}, the true GCM prediction, the top right shows y_{pred}, the GP prediction, the bottom left shows the absolute error, $|y_{pred} - y_{test}|$ and the bottom right shows the 1 standard deviation from GP, σ_{total}. Stippling on the bottom left indicates $|y_{pred} - y_{test}| > \sigma_{total}$.



(o) Test scenario 15. 74% grid points within 1σ, 93% grid points within 2σ.

(p) Test scenario 16. 69% grid points within 1σ, 94% grid points within 2σ.

Figure A.2: (o-p): Additional test scenarios. Top panel shows input values, relative to the baseline, minimum and maximum possible values. The top left map shows y_{test}, the true GCM prediction, the top right shows y_{pred}, the GP prediction, the bottom left shows the absolute error, $|y_{pred} - y_{test}|$ and the bottom right shows the 1 standard deviation from GP, σ_{total}. Stippling on the bottom left indicates $|y_{pred} - y_{test}| > \sigma_{total}$.



(q) Test scenario 17. 76% grid points within 1σ, 95% grid points within 2σ.

(r) Test scenario 18. 84% grid points within 1σ, 99% grid points within 2σ.

Figure A.2: (q-r): Additional test scenarios. Top panel shows input values, relative to the baseline, minimum and maximum possible values. The top left map shows y_{test}, the true GCM prediction, the top right shows y_{pred}, the GP prediction, the bottom left shows the absolute error, $|y_{pred} - y_{test}|$ and the bottom right shows the 1 standard deviation from GP, σ_{total}. Stippling on the bottom left indicates $|y_{pred} - y_{test}| > \sigma_{total}$.