Advecting Superspecies

Reduced order modeling of organic aerosols in LOTOS-EUROS using machine learning

by

P. Obin Sturm

to obtain the degree of Master of Science
in Applied Mathematics
specialization Computer Simulations for Science & Engineering,
at the Delft University of Technology,
to be defended publicly on Monday August 30, 2021 at 10:00 AM.

Student number:	4780582	
Project duration:	January 1, 2021 – August 30, 2021	
Thesis committee:	Prof. dr. ir. H. X. Lin,	TU Delft, supervisor
	Dr. ir. A. M. M. Manders-Groot,	TNO, supervisor
	Dr. ir. A. J. Segers ,	TNO
	Prof. dr. ir. C. Vuik,	TU Delft

TU Delft

Abstract

Chemical transport models (CTMs) are used to improve our understanding of the complex processes influencing atmospheric composition, as well as provide operational air quality forecasts and model potential future air quality scenarios. Numerical tracers in CTMs track the concentration of chemical species, while operators simulate various physical processes such as advection. One such CTM, LOTOS-EUROS, uses a volatility basis set (VBS) approach to represent the formation of organic aerosol (OA) in the atmosphere, which contributes to the concentration of total particulate matter. The added dimensionality of the VBS tracers in LOTOS-EUROS slowed down computation of the advection operator by a factor of two, limiting their representation in operational forecasts.

To keep the detailed process representation of OA formation, while reducing the computational costs, we develop an unsupervised machine learning method to compress the VBS tracers to a set of superspecies for use in advection, and subsequently decompress superspecies back to the tracer space for OA-relevant calculations. The focus of this machine learning method is physical interpretability, allowing for operators to resolve equations using the superspecies. This method conserves mass to machine precision and retains important information like phase (gas or aerosol) on compression. This data-driven approach reduces the dimensionality of the system more than a second proposed approach based on partitioning theory. The ML superspecies approach was integrated into LOTOS-EUROS for online calculations, showing numerical stability over a model simulation time of two weeks under various conditions. With the superspecies, the computation time for advection is reduced by 56% to 66% of the time for advection of the VBS tracers. The results of this approach show potential for use in accelerating air quality operational forecasts, as well as pathways forward for integration of ML box models of atmospheric chemistry into CTMs.

Keywords – aerosols, machine learning, atmospheric chemistry, LOTOS-EUROS, air quality

Acknowledgements

This thesis was written as part of a dual MSc program between TU Delft and TU Berlin, Computer Simulations for Science and Engineering (COSSE). I am very grateful to Kees Vuik, the lead organizer for COSSE at TU Delft, who took the time to meet me in person when I was just a prospective student. I appreciate your supportive and communicative presence along the way, including letting me take your scientific computing course, and pointing me in the direction of my TU Delft advisor Hai Xiang Lin. Hai Xiang was instrumental in initiating the project in September 2020, including the collaboration with TNO, all while I was in California, everything was online, and nothing was certain in terms of project timeline. I am lucky to have found a supervisor at TU Delft who shares so many of my interests, including air quality and physics guided machine learning. Thanks Hai Xiang! I also learned a lot in your high performance computing course. Hai Xiang introduced me to Arjo Segers and Astrid Manders-Groot at TNO. Arjo was great support with all things technical, including kicking the project off with some initial benchmarking runs, helping rediagnose the true cause of slowdown. Astrid provided daily guidance and motivation, walking me through code, helping me with physical interpretations of machine learning results, and helping me keep the big picture in mind. Astrid, your support and perspective inspired me to work through the challenging points throughout the project. Ultimately, it was an ambitious project and your advising was essential. I am glad to have had you as an advisor every step of the way.

Ruud Janssen's feedback and guidance was immensely helpful for my developing understanding of organic aerosols and their modeling approaches: thank you for your insightful, detailed comments, your willingness to talk about any bugs, and for fun coffee discussions. Astrid, Arjo, Ruud – I am grateful to all three of you for patiently answering the thousands of questions I asked this summer.

I would also like to acknowledge Tony Wexler, who prompted me to take a closer look at the LOTOS-EUROS benchmarking and pointed me in the direction of low-dimensional manifolds, and who has been an inspiring mentor to me. Shravan, your companionship and positive outlook was my favorite part of having a cohort – thanks for proofreading buddy. Thanks Mom and Dad for your suggestions and support, and being excellent examples of how to balance research with well rounded lives. Love you guys! Lastly, I want to thank Emma for talking through the mass balancing approaches with me the night that I had the idea, for answering my questions about the sky, for your edits, suggestions and constant support, and for helping me somehow find time for a wonderful summer full of swimming, sailing, music, and running.

P. Obin Sturm
Utrecht, August 2021

Figure 1: Obin in front of the KNMI-mast Cabauw in the Netherlands, after a long run from Utrecht. Photo credit Emma Ware.

Contents

1 Introduction

1.1. Particulate Matter, Public Health, and Climate Change

The chemical composition of the atmosphere affects human lives and ecosystems directly and indirectly. Pollutants in ambient air, including particulate matter (PM), are a public health concern (Dockery et al., 1994). Of particular concern to public health are fine *aerosols* that have diameters less than 2.5 micrometers, called $PM_{2.5}$ and even more so ultrafine particles, with diameters less than 0.1 micrometers. These very small particles are considered unhealthy for humans due to their ability to penetrate deep into the lungs (Nel, 2005). PM can be emitted directly or formed in the atmosphere. Directly emitted PM can come from anthropogenic (human-caused) sources, like combustion products from vehicles, as well as from natural sources like sea spray causing airborne salt particles. Gases emitted from biogenic or anthropogenic sources can react in the atmosphere to form secondary organic aerosols (SOA), increasing total PM (Robinson et al., 2007).

Air quality has improved over the last few decades in some parts of the world, such as Southern California (Parrish et al., 2016) and Europe (Colette et al., 2020; Colette et al., 2017) in response to changes in emission patterns. However, air pollution remains a global problem, exacerbated by climate change (Jacob & Winner, 2009). Warmer temperatures in some possible future climate scenarios are projected to lead to increased biogenic emissions of gases that are SOA precursors, resulting in higher secondary organic PM concentrations (Heald et al., 2008). Hotter and dryer conditions lead to longer wildfire seasons in some parts of the world, like California (Williams et al., 2019) and Australia (van Oldenborgh et al., 2021). Longer wildfire season and larger wildfires increase the amount of airborne particulate matter, and affect regional air quality adversely (Jaffe et al., 2020; Rooney et al., 2020).

In turn, PM also contributes to the total energy budget of the Earth's atmosphere: higher concentration of aerosols in the atmosphere have shown to have a non-negligible impact on radiative forcing, influencing the global energy budget. Reflection and absorption of radiation, as well as cloud formation, ultimately influence global mean temperatures (Forster, 2007; Jacobson, 2001). Understanding and forecasting atmospheric composition is important for understanding both ambient air quality and the changing climate of the earth.

1.2. Modeling Atmospheric Composition

The atmosphere is a complex system, and models are important tools for furthering our understanding of it. Models allow for synthesis of the most recent scientific understanding of atmospheric phenomena. Atmospheric models also have applications in operational forecasting, and supporting science-informed environmental policies.

Chemical transport models (CTMs) in particular simulate the chemical composition of the atmosphere, combining theory from the fields of physics, chemistry, and meteorology with techniques from scientific computing and data assimilation. Meteorological conditions from numerical weather prediction models are supplied as input to CTMs, which then numerically solve the continuity equation for all chemical

species of interest. In order to run such sophisticated simulations, these models utilize parallelization techniques from high-performance computing. Recently, machine learning approaches have been used as surrogate models for the most computationally expensive subroutines. Improving computational performance of CTMs is an active area of research (Keller & Evans, 2019; Kelp et al., 2020). Another area of research is improving representation of organic aerosol, as its contribution to PM has been underestimated in CTMs (Heald et al., 2005; Mircea et al., 2019).

CTMs focus on atmospheric composition and solve the continuity equation (see section 2.1) for chemical species of interest, called tracers. CTMs do not solve the conservation equations for momentum or energy in the atmosphere, which is done in general circulation models (GCMs), regional climate models (RCMs), and the more general earth system models (ESMs) (Brasseur & Jacob, 2017; Golaz et al., 2019). CTMs are sometimes coupled with these other dynamic atmosphere models, for example, when assessing the impact of climate change on air quality, or the effect of aerosols or greenhouse gases on radiative forcing. Coupled meteorology-chemistry models aim to capture the two-way interactions of weather and chemical composition (Baklanov et al., 2014). These feedback effects aren't possible to model when a CTM is given meteorological conditions as input. Further representation of atmospheric chemistry in earth system modeling is viewed as a future research priority for the field (National Academies of Sciences, Medicine, et al., 2016). Improved computational efficiency of atmospheric chemistry models will help realize that priority.

1.3. LOTOS-EUROS

LOTOS-EUROS is a CTM originally developed for the European continent that is used for both research and policy support purposes (Manders et al., 2017). It is the fusion of two independently developed models. LOTOS (LOng Term Ozone Simulation) was developed by TNO in collaboration with SAI (Systems Applications Incorporated) and Free University Berlin, based off of predecessors Urban Airshed Model and Regional Transport Model. Originally developed to model ozone, LOTOS incorporated aerosol modeling in 1995. In 2004, LOTOS was combined with EUROS (EURopean Operation Smog model). Prior to this, EUROS had been independently developed by the Dutch National Institute for Public Health and the Environment, RIVM, to simulate winter smog.

The current iteration of the model, LOTOS-EUROS (v2.2), is used to inform air quality regulation and model scenarios that include new energy policies and land use change. As it is relatively computationally efficient for a CTM, LOTOS-EUROS has been used to model longer term scenarios, and has been coupled with both regional and global climate models to assess long term climate change impacts on air quality (Manders et al., 2017). In addition, the model has been extended to other areas besides Europe: LOTOS-EUROS is currently used for operational forecasts for both China and northern Africa. A development priority for LOTOS-EUROS is inclusion of organic aerosols in its operational forecasts for air quality.

1.4. Organic aerosols in the atmosphere

Organic aerosol can be directly emitted to the atmosphere by vehicle exhaust, smoke from wildfires, or residential wood combustion, to name a few sources. These are considered primary organic aerosol (POA). Bioaerosol like pollen and fungal spores are excluded from the classical definition of POA. Organic aerosol can also be formed in the atmosphere via gas-phase chemistry forming lower volatility compounds, that partition to the aerosol phase more readily. Gaseous volatile organic compounds (VOCs) are emitted by anthropogenic sources, like solvents, refineries and other industrial activity, or biogenic sources, such as forests. These VOCs subsequently react with oxidants like the hydroxyl radical and ozone. As these VOCs are oxidized, they tend to become less volatile, becoming semi-volatile or intermediate volatile organic compounds (siVOCs). Some fraction of the mass of the siVOCs partitions into the aerosol phase, forming secondary organic aerosol (SOA). This SOA can make a significant contribution to total organic aerosol concentration (De Gouw et al., 2005; Heald et al., 2005), abbreviated in this thesis as TOA. POA has been shown to partially evaporate into siVOC, which can react and age via gas-phase reactions, subsequently becoming less volatile and forming SOA (Robinson et al., 2007). This SOA can be chemically distinct from the POA it came from: for example, it is often more oxidized (Jimenez et al., 2009). For this reason, it is sometimes treated separately from POA in models like LOTOS-EUROS, where it is called siSOA (Manders-Groot et al., 2021).

Organic aerosols are rarely explicitly modeled as speciated molecules, due to the complex nature of chemical processing of aerosols in the atmosphere and the huge number of distinct organic species. One method aiming to capture the range of volatilities is by lumping them into distinct volatility bins (Donahue et al., 2006; Jimenez et al., 2009). This volatility basis set (VBS) modeling approach has been shown to capture the variability of volatilities, from siVOCs to VOCs, as well as atmospheric ageing into less volatile compounds with increased partitioning into SOA.

1.5. Organic aerosols in LOTOS-EUROS

LOTOS-EUROS uses 4 volatility basis sets to represent OA. POA, siSOA, and SOA from both anthropogenic and biogenic precursors are all handled differently. This is due to the fact that VOCs and siVOCs from these 4 VBS classes have different gas-phase reaction rates. An added benefit of the four different VBS classes is that LOTOS-EUROS is able to assess the contribution to TOA from different sources, for example, urban sources versus forests.

SOA formation improves the description of organic aerosol from treating all OA as primary, which can lead to underestimations of OA when compared to observations. For this reason, the VBS classes are important to include in LOTOS-EUROS operational forecasts. However, its implementation leads to a substantial decrease in computational performance of LOTOS-EUROS, doubling the runtime. A relevant research question is then how to make this recently implemented, more sophisticated VBS more computationally efficient.

To illustrate the computational burden of OA modeling, Figure 1.1 shows overall clock time runs with the VBS module switched on and off. This benchmarking study was performed in both a parallel and sequential setting. The domain chosen is used in operational forecasting for the Copernicus Atmosphere Monitoring Service (CAMS): 0.1 degree resolution, 700x420 cells over Europe, for 1 day. The domain was split into 24 sub-domains, which were parallelized over 24 central processing unit (CPU) nodes. The fully sequential run with just one CPU node used 1/24 of the domain used in the implementation.

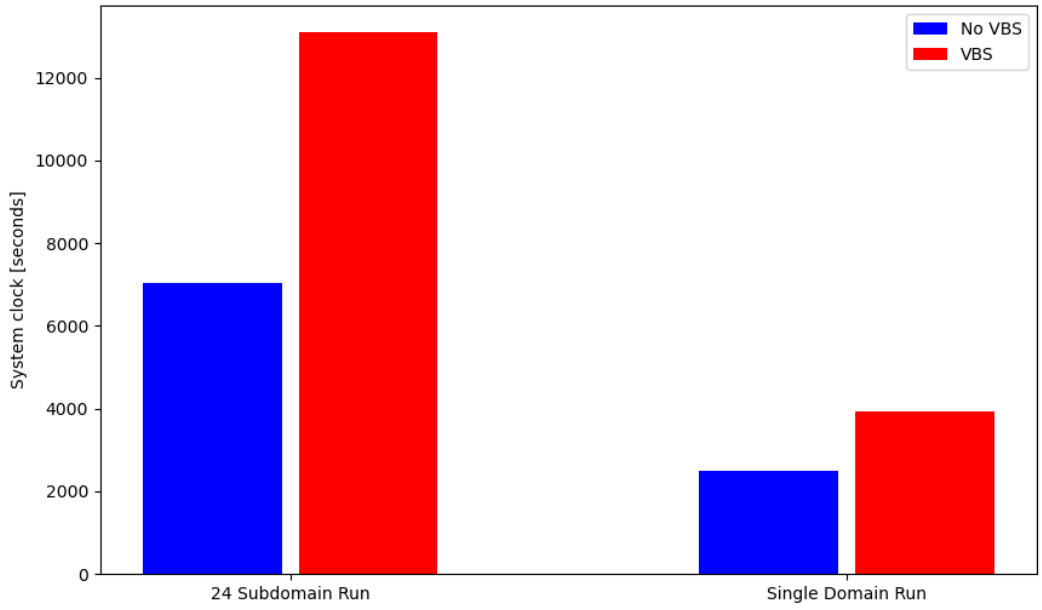


Figure 1.1: Benchmarking the clock speed effect of the VBS operator. The single domain run was performed on one CPU with 1/24th of the domain. The 24 subdomain run was parallelized on 24 CPUs.

Performance can vary depending on how busy each computing node is. However, the above figure

serves as an illustrative example of the VBS-caused slowdown. It should be noted that the performance slowdown with VBS was exacerbated by the domain decomposition parallelization strategy, doubling the system clock time, as opposed to an approximate two-thirds increase for the single domain run. Detailed reports on the timing of these runs are available in Appendices A.1 through A.4.

1.6. Machine learning for aerosols and atmospheric chemistry

The atmospheric sciences have used statistical and machine learning (ML) methods for decades, including for aerosol modeling applications. In the Netherlands, linear regression models PROZON and PROPART started being used for operational forecasting by RIVM in 1992 and 1998 to predict next-day maximum ozone and PM_{10} concentrations respectively, given current concentrations and meteorological conditions (Noordijk, 2003). In the 1990s, an unsupervised machine learning method called positive (non-negative) matrix factorization was developed to find latent factors in aerosol dynamics (Paatero & Tapper, 1994; Paatero et al., 1991). A few years later, a supervised machine learning method, a neural network, was used to estimate surface vapor concentrations of inorganic aerosols (Potukuchi & Wexler, 1997).

Machine learning in atmospheric modeling is still an active area of research and has benefited from recent advances in the field of machine learning. In atmospheric chemistry, ML surrogate models have been developed (Keller & Evans, 2019; Kelp et al., 2020; Kelp et al., 2018) in an attempt to replace the computationally intensive step of the integration of the coupled system of ordinary differential equations representing production and loss rates of chemical species. Both Keller and Evans, 2019 and Kelp et al., 2018 reported instability when using ML models for sequential predictions over longer time periods. ML models do not respect certain symmetries inherent to classical models like conservation of mass, and can systematically add or remove mass to the model, exiting the solution and input space that they were optimized to predict values for. Error compounds on recurrent ML predictions without deterministic constraints from classical models to keep the ML in their solution space, which can lead to runaway exponential error growth. Keller and Evans, 2019 indicated hybridizing deterministic constraints and ML as a future research direction, with a specific example of stoichiometric information. Sturm and Wexler, 2020 provide a framework to incorporate stoichiometric balances into ML surrogate models for systems of gas-phase reactions, conserving mass to machine precision. This is also generalizable to other processes that involve fluxes between properties, such as condensation/evaporation in aerosol microphysics models and radiative energy flux in climate models.

Kelp et al., 2020 fixed the runaway error propagation problem from previous work, with a neural network approach optimized to minimize long-term prediction error. The neural network predicts future concentrations recurrently in a latent space using a recurrent autoencoder architecture. Though physical constraints aren't integrated into this architecture, this is indicated as a future research direction. Another future research question posed by Kelp et al., 2020 is whether other processes like advection can be performed in the reduced-order latent space. This thesis explores that research direction.

1.7. Research questions

The computational burden of the VBS discussed in section 1.5 and current state of research inform the following research questions.

Research question 1: What parts of LOTOS-EUROS are slowed down by inclusion of the volatility basis set? Can they be accelerated using machine learning?

Research question 2: Can a machine learning approach maintain desired accuracy of total organic aerosol, as well as volatility distributions, sources, spatial and temporal patterns?

Research question 3: In what ways can physical information be incorporated into machine learning methods to improve interpretability and/or respect important physical properties?

Research question 4: How does a machine learning parameterization perform when implemented online in LOTOS-EUROS?

1.8. Overview of report

The research questions detailed in section 1.7 guide the rest of the thesis. Chapter 2 introduces the governing equations in LOTOS-EUROS, how organic aerosols are modeled using four volatility basis sets, and how they interact with different processes, including meteorological phenomena like advection. A more granular benchmarking analysis gives insight into which parts of LOTOS-EUROS are most affected by including the four VBS classes and provides an answer to the first part of research question 1. This motivates a parameterization strategy to accelerate parts of LOTOS-EUROS that experience the largest slowdown. The strategy involves machine learning to find latent patterns in model output from LOTOS-EUROS runs, and forming a set of superspecies from VBS tracers. With machine learning (ML) parameterizations in mind, chapter 3 details several different ML approaches appropriate for this problem, and summarizes some recent exploration of ML in modeling of the atmosphere. Chapter 4 uses a linear ML method to explore research question 2 in depth, laying out a framework of the approach, including its hyperparameters, capabilities, and limitations. Chapter 5 builds off of the ideas explored in chapter 4, assessing a complex and nonlinear ML method, a neural network autoencoder that ensures non-negativity, to see if nonlinear methods are more appropriate for this problem. After showing that linear methods are sufficient and in fact a more appropriate choice for this problem, the rest of chapter 5 is dedicated to incorporating physical information into the linear methods. This includes mass conservation and constraining the methods to conserve phase (aerosol or gas). Chapter 5 concludes with a judgement on the most promising method. This machine learning method is then integrated into a customized version of LOTOS-EUROS in chapter 6, where its online accuracy, stability, robustness, and computational benefit are all assessed. Chapter 7 ends with conclusions gained from this study, returning to the research questions and outlining several potential directions for future research.

2

Organic aerosols in LOTOS-EUROS

2.1. The continuity equation

Chemical transport models (CTMs) numerically model pollutants and other chemical species of interest in the atmosphere (Brasseur & Jacob, 2017). These species, referred to in the model as tracers, can exist as gas, liquid, or solids. Tracers can chemically react with other tracers, get blown by the wind (advection), be emitted by various sources into the atmosphere, drop back out of the atmosphere (deposition), diffuse along concentration gradients, and move through different vertical layers of the atmosphere. All of the phenomena mentioned can be combined in the continuity equation, representing how the concentration C of a tracer changes over time t and space (x, y, z):

$$\frac{\partial C}{\partial t} + \underbrace{\nabla \bullet (C\mathbf{U})}_{advection} = \underbrace{\frac{\partial}{\partial x}\left(K_h \frac{\partial C}{\partial x}\right) + \frac{\partial}{\partial y}\left(K_h \frac{\partial C}{\partial y}\right) + \frac{\partial}{\partial z}\left(K_z \frac{\partial C}{\partial z}\right)}_{diffusion} + R + Q + E - D - W \tag{2.1}$$

where $\mathbf{U}$ is a vector containing the bulk wind velocities in the east-west, north-south, and up-down direction: this term represents advection. K_h and K_z are horizontal and vertical diffusion coefficients, and are used in terms that represent diffusion of C from high concentrations to low concentrations. The R term represents net formation rate of the chemical species C. Q is the emission rate of that compound from various sources (for example anthropogenic or biogenic emissions). E represents entrainment (or detrainment), which is dependent on the atmospheric mixing layer. D and W represent dry and wet deposition of particles onto surfaces, ultimately decreasing their concentration in the air (Manders-Groot et al., 2021).

Eq.(2.1) is supplied with boundary and initial conditions. These boundary and initial conditions can be taken from observations, hypothetical cases, model output on larger domains, or a mixture of all three. Vertical boundary conditions are generally Neumann type conditions, where flux at the surface of the Earth is dependent on emissions and deposition rates, and where there is zero flux at the top of the atmosphere. In a global model, horizontal boundary conditions are periodic (Brasseur & Jacob, 2017).

The R term can generally be described through a net sum of overall production rate P and loss rate L corresponding to C:

$$R = P - L \tag{2.2}$$

This can also be represented through a net sum of individual reaction rates r:

$$R = \sum_p r_p - \sum_l r_l \tag{2.3}$$

Where p and l correspond to individual reactions where C is produced or consumed. Each rate r is often controlled by the concentrations of its reactants and an (often empirically) determined rate constant k. For example, the rate equation for the bimolecular reaction $A + B \rightarrow C$ can be expressed as

$$r = kC_A C_B \tag{2.4}$$

This shows that often r, and therefore R, is dependent on chemical species other than the species represented by tracer C in equation (2.1). This leads to coupling of the continuity equation across species.

The domain in the continuity equation (2.1) is discretized in time and space so that it can be solved numerically. With N spatial gridpoints, this results in N coupled equations in space at time t, for the tracer C. An additional complexity is in the R term, which can be dependent on other chemical species. Moreover, R is often nonlinear, in the case of multimolecular reactions or more complex rate laws. As shown by equations (2.3) and (2.4), R couples the single-species Eq.(2.1) for all the chemical tracers. This complexity can be addressed through a numerical method known as operator splitting.

2.2. Operator splitting

CTMs employ the method of operator splitting, also known as the method of fractional steps, to solve Eq. (2.1) in pieces for every timestep (Brasseur & Jacob, 2017; Janenko, 1971). Operator splitting is the practice of separating and solving a differential equation according to its terms, or *operators*, solving each operator over a time step in a given order, and passing each previous operator's solutions as input to the subsequent operator. When all done using the same timestep, this is called Lie-Trotter splitting. An introductory example of Lie-Trotter operator splitting can be given by

$$\frac{\partial u}{\partial t} = Au + Bu \tag{2.5}$$

Where A and B are differential operators. The actual solution to Eq. (2.5) over a timestep Δt is

$$u(t + \Delta t) = e^{\Delta t(A+B)} \tag{2.6}$$

First solving for the Au operator, we get

$$u_A(t + \Delta t) = e^{\Delta t A} u(t) \tag{2.7}$$

where u_A is the solution from the Au operator. The next step is solving the Bu operator with u_A as an initial condition. This gives

$$u_{A,B}(t + \Delta t) = e^{\Delta t B} u_A(t + \Delta t). \tag{2.8}$$

Where the A, B subscript for u indicates the order of the separately solved operators. Combining equations (2.7) and (2.8), we get

$$u_{A,B}(t + \Delta t) = e^{\Delta t B} e^{\Delta t A} u(t) \tag{2.9}$$

If A and B commute, equation (2.9) is equivalent to equation (2.6). If not, the product of the exponential factors in $u_{A,B}$ can be related to exponential sum of $u(t + \Delta t)$ with a first order error term $O(\Delta t)$ via the Baker-Campbell-Hausdorff formula. This error is called the splitting error.

An alternative splitting with second order error is called Strang splitting (MacNamara & Strang, 2016). The Strang splitting approach for this example would take timesteps of $\frac{\Delta t}{2}$ once to solve subproblem A, one full timestep Δt to solve subproblem B, then another step of $\frac{\Delta t}{2}$ for subproblem A. It can be shown with more terms in the Baker-Campbell-Hausdorff formula to have a second order splitting error.

Operator splitting methods are generalizable to more than two operators, as well as multiple timesteps, in which case each operator passes output to the next, cyclically. This is utilized in chemical transport models, which model many complex and coupled phenomena.

2.3. CTM operator splitting

In a chemical transport model, each operator is tasked with solving a portion of equation (2.1) over a certain time interval Δt. The common approach is to split up the different physical processes represented by the terms of the continuity equation. This approach to solve subproblems has computational benefits. For example, though the chemistry operator R and the advection equation are coupled, advection of one substance C is not explicitly dependent on other chemical species, and chemistry is not explicitly dependent on space (at least on the spatial scale of one CTM grid cell – molecular interactions are represented in the form of rate equations). Chemistry operators without spatial dependence are called box models. The assumption is that over the operator splitting timestep Δt, the operators can be approximated as decoupled. At the cost of a splitting error, the degrees of freedom of both chemistry and advection operators can be greatly reduced (Brasseur & Jacob, 2017).

For an illustrative example of how this looks in a CTM, assume a simplified example of equation (2.1) that only accounts for advection and chemistry (though this can be generalized to include diffusion and other processes). The example problem looks like

$$\frac{\partial C}{\partial t} = \underbrace{-\nabla \bullet (C\mathbf{U})}_{advection} + \underbrace{R}_{chemistry} \tag{2.10}$$

The left hand side can be split into two terms that each handle the terms of the right hand side:

$$\frac{\partial C}{\partial t} = \left[\frac{\partial C}{\partial t}\right]_{advection} + \left[\frac{\partial C}{\partial t}\right]_{chemistry} \tag{2.11}$$

which leads to two subproblems, one for chemistry:

$$\left[\frac{dC}{dt}\right]_{chemistry} = R \tag{2.12}$$

and one for advection:

$$\left[\frac{\partial C}{\partial t}\right]_{advection} = \nabla \bullet (C\mathbf{U}) \tag{2.13}$$

Given an initial condition $C(t)$, equation (2.12) is then solved over Δt. The chemistry operator is often a nonlinear system of ordinary differential equations; the advection operator a linear system of partial differential equations.The resulting solution is given as input to equation (2.13), which then is solved over Δt to complete the final solution $C(t + \Delta t)$.

When elements of **U** are large (east-west winds, conventionally U_x,are often highest), a numerical stability requirement places an upper bound on the operator splitting timestep Δt, as described by the Cauchy-Friedrichs-Lewy criterion (see section 2.4, equation (2.14)). Strang or other higher order operator splitting can be used to further decompose the advection operator into the various spatial dimensions. With this, U_x will be solved at a fraction of Δt several times in a cycle through all the operators.

Operator splitting allows for a modular approach to CTMs, which is beneficial to both model development and flexibility. Decomposing the continuity equation into different operators reveals that certain processes consume the majority of computing resources. Acceleration of the more computationally intensive operators is therefore both an established and active area of research, for example aerosol microphysics and thermodynamics (Potukuchi & Wexler, 1997; Silva et al., 2020; Zaveri et al., 2008), gas-phase chemistry (Keller & Evans, 2019; Lowe & Tomlin, 2000; Whitehouse et al., 2004), as well as full mechanisms for aerosol and gas-phase chemistry (Kelp et al., 2020; Santillana et al., 2010).

2.4. LOTOS-EUROS operators

LOTOS-EUROS solves the continuity equation in equation (2.1). Each term is treated in separate operators. Advection, emissions, diffusion and entrainment, dry deposition, wet deposition, and chemistry (including treatment of organic aerosol partitioning) are all resolved within the time splitting step Δt. Strang splitting is utilized to give a second order splitting error. Figure 2.1 illustrates the Strang splitting strategy in LOTOS-EUROS: resolve advection, vertical diffusion (vdif in the figure) and other operators on half of an operator splitting timestep until chemistry or emission. After resolving chemistry or emissions on the full operator time splitting step, the previous operators are called over half of a timestep in reverse order.

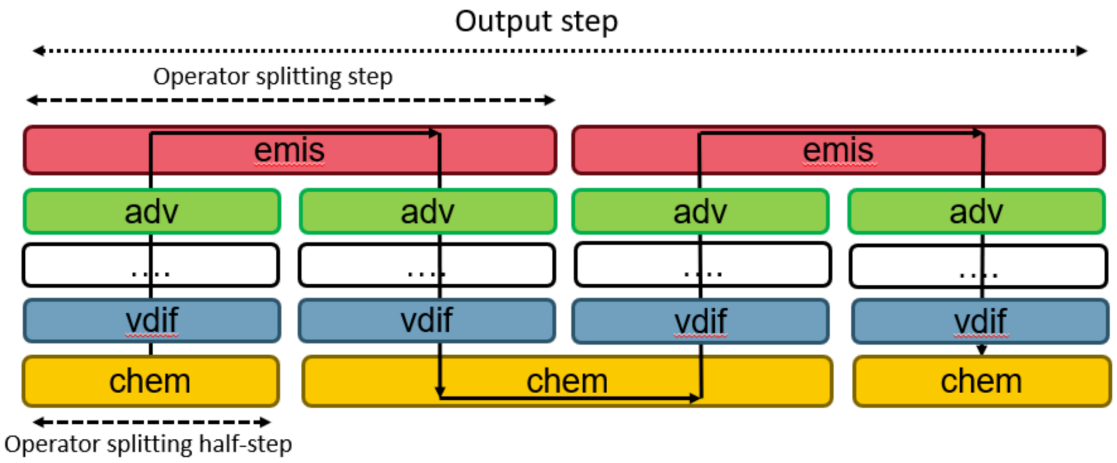


Figure 2.1: A visual of the splitting scheme for various operators in LOTOS-EUROS. Reproduced from the LOTOS-EUROS reference guide (Manders-Groot et al., 2021) with permission.

The advection operator generally limits the length of Δt as governed by the Courant-Friedrichs-Lewy (CFL) condition. The CFL condition puts an upper limit Co_{max} on the Courant number Co in order to ensure numerical stability:

$$Co = \Delta t \left(\frac{U_x}{\Delta x} + \frac{U_y}{\Delta y} + \frac{U_z}{\Delta z} \right) < Co_{max} \tag{2.14}$$

Where $\mathbf{U} = [U_x, U_y, U_z]$ from Eq. (2.1) is shown in each spatial direction and the denominators are the sizes of the corresponding discretizations. The Courant number is a dimensionless quantity used often in discretizing fluid flows: in our case, the physical interpretation of the CFL condition is that an air "puff" carried by bulk transport should not be able to cross a full grid cell within Δt. Though the winds $[U_x, U_y]$ in advection require a minimum Δt, the advection operator is actually chosen in the current version of LOTOS-EUROS to be computed for a half timestep to allow for longer Δt for the chemistry and emission processes while still satisfying (2.14).

2.5. Organic aerosol representation in LOTOS-EUROS

2.5.1. The volatility basis set

The volatility basis set as originally introduced by Donahue et al., 2006 models the partitioning of semi-volatile and intermediate volatility organic compounds (sometimes lumped together in the acronym S/IVOCs) between the aerosol and gas phases. This is determined for each compound by their saturation vapor concentration – the atmospheric concentration at which evaporation rate from a particle is equal to its condensation rate. Donahue et al., 2006 define S/IVOCs as compounds that have saturation vapor concentrations in between 0.01 and 100,000 $\mu g\ m^{-3}$ at 300 K. The VBS approach organizes S/IVOCs into a basis set of logarithmically distributed saturation concentrations, where each S/IVOC is assigned the closest effective saturation concentration C_i^*. This results in a mass loading distributed over volatility bins (Donahue et al., 2006).

It is useful when formulating the volatility basis set approach to define a partitioning coefficient ξ_i for a substance i. This can be used to relate the concentration of a certain organic component C_i in all phases with the total organic aerosol concentration C_{aer}:

$$C_{aer} = \sum_i \xi_i C_i \tag{2.15}$$

The partitioning coefficient can be calculated using the effective saturation concentration C_i^*:

$$\xi_i = \left(1 + \frac{C_i^*}{C_{aer}}\right)^{-1} \tag{2.16}$$

The set of effective concentrations C^* is called the volatility basis set. In order to cover the wide range of volatility of organic compounds, each element of C^* represents a different order of magnitude of saturation vapor concentration defined as:

$$C^* = \{0.01, 0.1, 1, 10, 100, 1000, 10^4, 10^5\} \tag{2.17}$$

where the values within the basis set are in units of $\mu g\ m^{-3}$ and defined at 300 K.

The VBS approach can model oxidative ageing as well as volatility. More specifically, the relationship between SOA oxidation level and volatility is modeled as monotonically decreasing, leading to a ”zombie” effect of all bins marching to the lowest volatility bin (Bergström et al., 2012). This behavior can be expressed via the general reaction

$$CG_x + OH \rightarrow CG_{x-1} \tag{2.18}$$

for $x = 2, \ldots, dim(C^*)$, where CG_x represents the amount of condensable gas-phase material in bin x of C^*. A frequent assumption is that the reactions in (2.18) have a uniform rate constant k for all x, for example $k = 4 \times 10^{11}\ cm^3\ molecule^{-1}\ s^{-1}$ (Robinson et al., 2007). For a 9-bin volatility basis set, this leads to the system of coupled ordinary differential equations:

$$\begin{aligned}
\frac{d}{dt}CG_9 &= -kC_{OH}CG_9 \\
\frac{d}{dt}CG_8 &= kC_{OH}CG_9 - kC_{OH}CG_8 \\
&\vdots \\
\frac{d}{dt}CG_2 &= kC_{OH}CG_3 - kC_{OH}CG_2 \\
\frac{d}{dt}CG_1 &= kC_{OH}CG_2
\end{aligned}$$

(2.19)

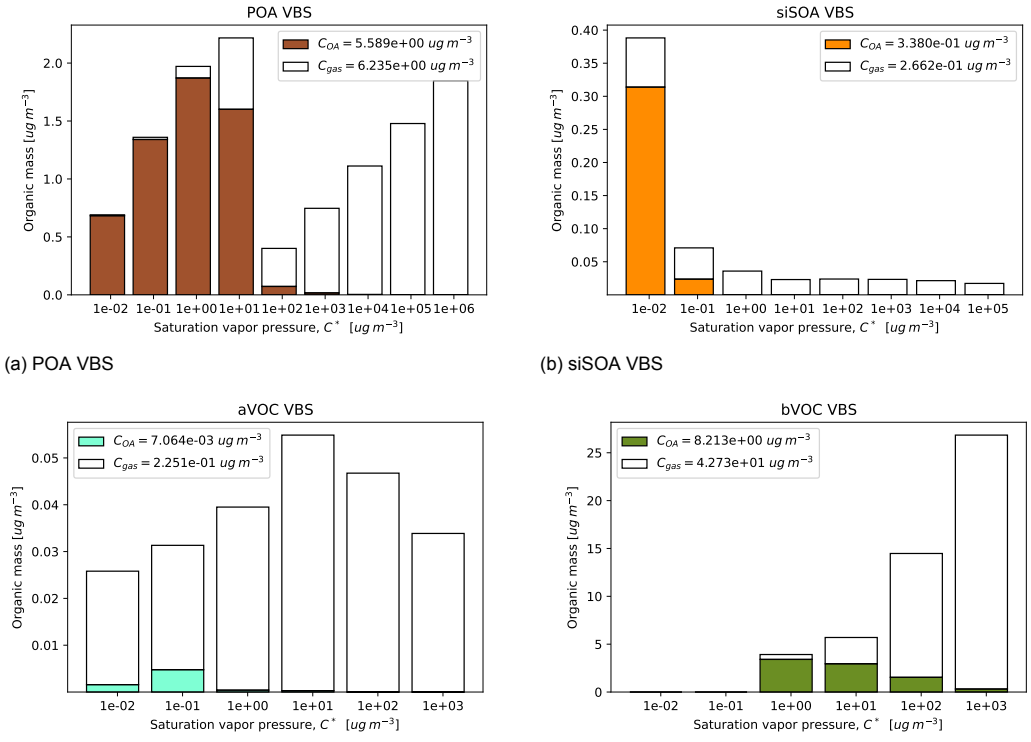
where C_{OH} is the concentration of the hydroxyl radical.

The volatility basis set approach was implemented in a CTM over the United States soon after its introduction. Shrivastava et al., 2008 implemented the VBS scheme in PMCAMx, accounting for gas-particle partitioning of primary organic aerosol (POA). Prior to this, primary emissions were modeled as nonvolatile and did not evaporate. Considering only aerosol phase primary emissions neglects a substantial amount of material that evaporates quickly (Robinson et al., 2007). This material only becomes detectable after ageing and becoming more volatile. The approach used by Robinson et al., 2007 and Shrivastava et al., 2008 is distributing the POA emissions across the lowest 4 volatility bins, and additional material from a fit factor over the highest 5 volatility bins to represent the evaporated fraction. This factor was fit from data to be 1.5 times the original emissions, ultimately multiplying condensable primary emissions by a factor of 2.5. One important conclusion of Shrivastava et al., 2008 was that improved representation of SVOC emissions, including allowing evaporation of POA, could increase predictions of total organic aerosol concentration (C_{OA} from the previous section), up to 50% in an urban summertime setting (Shrivastava et al., 2008).
The above result was an important conclusion, in light of the systematic underestimation of organic aerosols by CTMs (Mircea et al., 2019). The ability of the volatility basis set to better capture organic PM led to its implementation in other CTMs and regions, for example Mexico City (Tsimpidi et al., 2010) and Europe (Bergström et al., 2012). The work of Bergström et al., 2012 was influential to the current VBS implementation in LOTOS-EUROS. Bergström et al., 2012 conducted a comparison study with four different VBS classes, implemented into the EMEP (European Monitoring and Evaluation Programme) chemical transport model in long-term simulations over Europe, modeling years 2002-2007. This study included comparisons to measurements and other models. The four different basis set approaches differ in how they treat partitioning of POA, aging of primary semi-volatile and intermediate volatility organic compounds (together siVOC), and production of SOA. No judgement on best model configuration was made. One conclusion was that that bSOA is a major contributor to TOA in the summer, even if ageing reactions are turned off for that VBS class. However, uncertainty in bVOC emissions was large, and this study concluded that more data on bVOC emission rates is required for meaningful summertime SOA modeling by CTMs.

2.5.2. Four VBS classes in LOTOS-EUROS

LOTOS-EUROS uses several 1D volatility basis sets to model different sources of organic aerosols (OA), including secondary organic aerosols (SOA) from gaseous precursors, primary organic aerosols (POA), and SOA from volatile POA emissions (Manders-Groot et al., 2021). Volatile organic compounds such as xylene (XYL), toluene (TOL), paraffins (PAR), and olefins (OLE) are considered to be anthropogenic precursors of secondary organic aerosols, though they themselves are too volatile to partition to the aerosol phase appreciably. Formation of SOA from anthropogenic VOCs (aVOC) is represented with a 6-bin VBS defined from 10^{-2} to 10^{3} $\mu g\ m^{-3}$ at 298 K. An analogous 6-bin VBS is used to model SOA formation from biogenic VOC precursors (bVOC), namely monoterpenes (TERP) and isoprene (ISO).

Figure 2.2 shows examples of mass distributions over volatility bins for the 4 VBS classes. Primary organic material (POM) emissions are modeled using a 9-bin VBS approach, logarithmically distributed from 10^{-2} to 10^{5} $\mu g\ m^{-3}$ at 298 K. The reported mass is distributed over the lower 4 volatility bins. An additional 1.5 times this mass is distributed over the higher 5 volatility bins, representing non-reported semi and intermediate volatility organic compounds (S/IVOCs) that will age to form siSOA. Only a fraction remains in the aerosol phase – the fraction that evaporates is assumed to be semi-volatile VOC, $1 < C^{*} < 10^{3} \mu g\ m^{-3}$ or intermediate volatility VOC, $10^{3} < C^{*} < 10^{6}\ \mu g\ m^{-3}$, defined at 298 K. These S/IVOCs are treated separately from POA in LOTOS-EUROS, as they undergo secondary oxidation and form SOA, denoted as siSOA for semi/intermediate volatility. Subsequent oxidation leads to siSOA represented by lower volatility bins, down to $10^{-2} \mu g\ m^{-3}$. The total siSOA is represented by an 8-bin VBS from 10^{-2} to 10^{3} $\mu g\ m^{-3}$ (defined at 298 K). POA and siSOA are represented using different volatility basis sets, but as they originate from the same sources, show similar spatial patterns in comparison to the other basis sets. However, as siSOA is aged, it has more time to be transported in the LOTOS-EUROS, spreading out from the emission source.



(a) POA VBS

(b) siSOA VBS

(c) aVOC VBS

(d) bVOC VBS

Figure 2.2: Representative examples from a LOTOS-EUROS run of mass distribution over the volatility bins, and phase partitioning within each bin, for all 4 VBS classes. The POA example in 2.2a is in the surface gridcell containing the Cabauw Experimental Site for Atmospheric Research in the Netherlands, on February 26 at 21:00. The other three VBS classes are for a surface gridcell over a forest in southern Germany, Schönbuch Nature Reserve, on July 30 at 04:00.

The tables below provide a summary of the VBS classes and their tracers, as detailed in the LOTOS-EUROS reference guide (Manders-Groot et al., 2021).

Table 2.1: List of VBS aerosol tracers in LOTOS-EUROS.

C_i^*	10^{-2}	10^{-1}	1	10	10^2	10^3	10^4	10^5	10^6
aVOC	asoa1	asoa2	asoa3	asoa4	asoa5	asoa6	-	-	-
bVOC	bsoa1	bsoa2	bsoa3	bsoa4	bsoa5	bsoa6	-	-	-
POA	poa1	poa2	poa3	poa4	poa5	poa6	poa7	poa8	poa9
siSOA	sisoa1	sisoa2	sisoa3	sisoa4	sisoa5	sisoa6	sisoa7	sisoa8	-

Table 2.2: List of VBS gas tracers in LOTOS-EUROS.

C_i^*	10^{-2}	10^{-1}	1	10	10^2	10^3	10^4	10^5	10^6
aVOC	asog1	asog2	asog3	asog4	asog5	asog6	-	-	-
bVOC	bsog1	bsog2	bsog3	bsog4	bsog5	bsog6	-	-	-
POA	pog1	pog2	pog3	pog4	pog5	pog6	pog7	pog8	pog9
siSOA	sisog1	sisog2	sisog3	sisog4	sisog5	sisog6	sisog7	sisog8	-

On top of tracers for the SOA species, there are also tracers required for the secondary organic gases (SOG). This results in 12 tracers required to model SOA formation from aVOC: 6 aerosol phase tracers (aSOA) and 6 aerosol phase tracers (aSOG). Similarly, bVOC formation of SOA requires 12 tracers for bSOA/bSOG. The 9-bin POA/POG VBS requires 18 tracers, and the 8-bin siSOA/siSOG VBS requires 16 tracers. In total, this adds 58 VBS-specific tracers to the 64 non-VBS tracers in LOTOS-EUROS.

2.5.3. VBS tracers in other operators

The VBS module falls within the chemistry operator in LOTOS-EUROS. The module calculates the saturation vapor pressures of the volatility bins via the Clausius-Clapeyron equation. It then calculates the new partitioning for each bin in each VBS class via equations (2.15) and (2.16) (Manders-Groot et al., 2021).

Also within the chemistry operator is gas-phase reactions of the condensable gas tracers: asog_i,pog_j, and sisog_k, where i = 1, ... ,6, j = 1,..,9, and k = 1,...,8. These reactons shift material to lower volatility bins in the form of equation (2.19). Gaseous precursors to aSOA (OLE, PAR, TOL, and XYL) and bSOA (ISO and TERP) react to form aSOG and bSOG in the upper four volatility bins for those classes, from saturation concentrations of 10 $\mu g\ m^{-3}$ to 1000 $\mu g\ m^{-3}$. The yields of each bin from the VOC precursors is parameterized into two cases, high and low NO_x conditions, as suggested by Lane et al., 2008 (Manders-Groot et al., 2021). Note that ageing between bins in the biogenic VBS is currently off in LOTOS-EUROS, so all contributions to the biogenic VBS bins come from biogenic gaseous precursors. As a result the lowest 2 volatility bins do not recieve any material and are effectively off, though they are still passed between processes.

The emissions operator directly interacts with only the POA VBS tracers. Different emission inventories can be used, which determine how POA is distributed over the bins. The base inventory for emissions is the TNO-MACC II inventory (Kuenen et al., 2014). Deposition handles gases and aerosols differently, but works with each tracer individually. Transport processes, like advection and vertical diffusion, act on each tracer individually (unless explicitly turned off – extremely quickly reacting radicals, for example, are not advected). All non-chemistry operators treat tracers as being independent of each other, as described in the section on CTM operator splitting. However, all must resolve their equations for each tracer: the high-level structure is therefore a for loop over all tracers, unless the tracers are specifically turned off. For example, the advection operator still needs to calculate the concentration gradient for each relevant tracer as shown by (2.13). The computation time for all other operators is therefore expected to increase linearly with number of tracers to be advected: $\mathcal{O}(n)$ where n is the number of overall tracers.

2.6. Diagnosing the source of slowdown

In the introductory chapter, Figure 1.1 showed that slowdown from a VBS operator is amplified when using domain parallelization: VBS inclusion in the parallel run doubled computation time, compared to about a 2/3 increase in computation time for the sequential run. This is striking because the VBS operator is a box model process, where adjacent gridcells don't interact. Domain parallelization splits up processes in space, and section 2.3 describes how chemistry operators (of which VBS is considered one of) are treated as spatially independent within the operator splitting timestep. The VBS operator is not spatially dependent and does not require concentrations from domains on other CPUs, so why does its inclusion slow down a parallelized run more than a sequential run?

The above question motivates a more granular benchmarking analysis to find the source of the slowdown. Figures 2.3 and 2.4 report the proportion of clock time within the inner time loop of LOTOS-EUROS of sequential and parallelized runs.

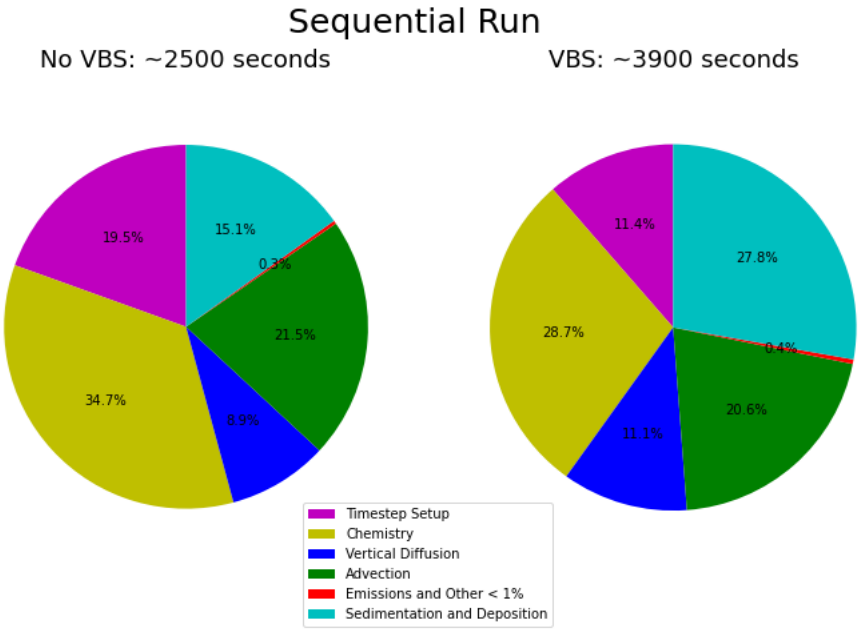


Figure 2.3: A sequential run benchmarking the various operators in LOTOS-EUROS on 1/24th of the CAMS domain, for 1 day. The typical resolution and domain is 0.1 degrees for 700x420 cells over Europe.

The sequential run shows that the relative time of the overall chemistry operator decreases. Computations from OA tracers in the overall chemistry operator include partitioning of vbs tracers and gas-phase reaction with hydroxyl radical. With the VBS tracers, overall time spent on chemistry increased from ~830 seconds to ~1122 seconds. This 35% slowdown in the time loop is disproportionate to the overall slowdown from ~2396 to ~3914 seconds: a proportional slowdown of 63%. Advection, on the other hand, goes from (.215×2500 seconds) ≈ 538 seconds to (.206×2900 seconds) ≈ 803 seconds: nearly a 50% increase in wall time. Similarly, and yet more extreme, the wall-times for deposition operators increase from ~378 seconds to ~1084 seconds, corresponding to a 187% increase in wall time. Both advection and dry deposition perform for loops over all tracers. The takeaway from the sequential run is that addition of organic aerosol tracers via the volatility basis set adds minimal computation time for partitioning and organic chemistry, but rather slows down other processes such as transport.

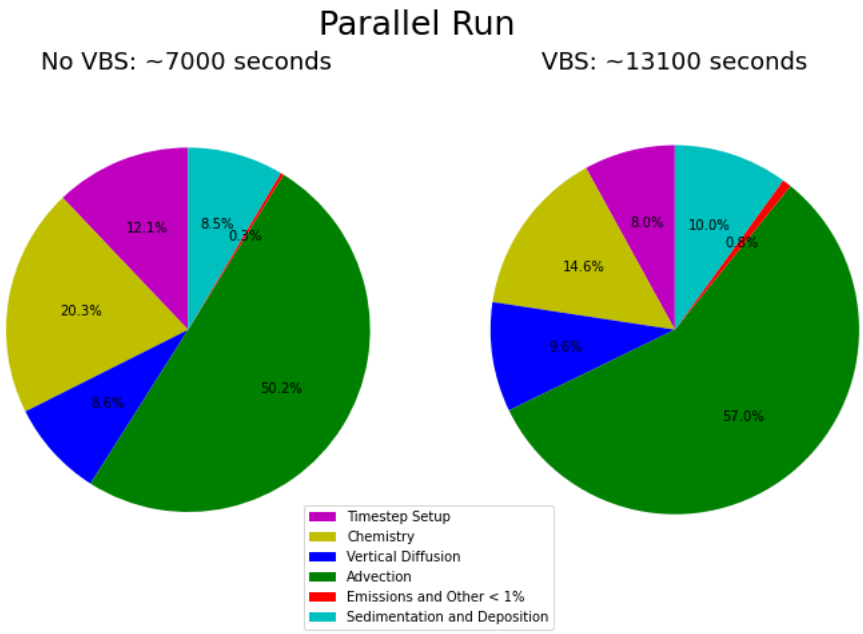


Figure 2.4: A parallel run benchmarking the various operators in LOTOS-EUROS using 24 CPUs on the CAMS domain, using the standard operational settings: 0.1 degree resolution, 700x420 cells over Europe.

Comparing the proportion of advection (green) of Figure 2.4 with that of 2.3, it can be seen that advection becomes the dominant operator in terms of computation time. Inclusion of the VBS nearly doubles the total amount of tracers, and this is reflected in the timing for the advection operator: Figures 2.3 and 2.4 show that advection costs increase by up to 2 times when the VBS is implemented. This problem is exacerbated by the parallelization. The time required to communicate data between computing nodes is known as overhead. This overhead is often represented by a communication time function. This communication time function is linearly dependent on message size m in bytes of data to be communicated

$$t_{com} = \alpha + \beta m \tag{2.20}$$

where α is a setup/startup time of communication, and β is the transmission speed in seconds per byte. When $\alpha << \beta m$, t_{com} can be regarded as only dependent on the size of the data. However, when α is not negligible, the amount of communication instances becomes a factor. This is the case for TNO's home system running LOTOS-EUROS. In the current advection scheme, the row of the whole domain is first scanned for minima and maxima. Advection is then resolved left-right and then right-left over the row of gridcells for the whole domain. When the domain is decomposed into subdomains for processors to solve in parallel, each processor has to communicate data to other computing nodes. The advection operator resolves equation (2.13) for each species, so this process must be repeated for every chemical species.

2.7. Motivation for reduced-order modeling

Inclusion of the VBS tracers slows down processing time for the advection operator, which does not perform calculations specific to OA chemistry and partitioning. In fact, advection is a bulk process that doesn't require tracer-specific parameters. This problem motivates investigation of whether the high dimensional VBS tracer space could be represented by a lower-dimensional manifold for use in other processes. Underlying equations of these processes could be solved for a latent space representation of the tracers rather than the tracers themselves. This latent space representation could be interpreted as a set of superspecies that are formed from combinations the original tracers. If there is a mapping from tracers to superspecies and superspecies back to the tracers, then the most computationally expensive operators can resolve their equations using the superspecies, while VBS-specific operators such as emissions and chemistry can be resolved in the full VBS tracer space.

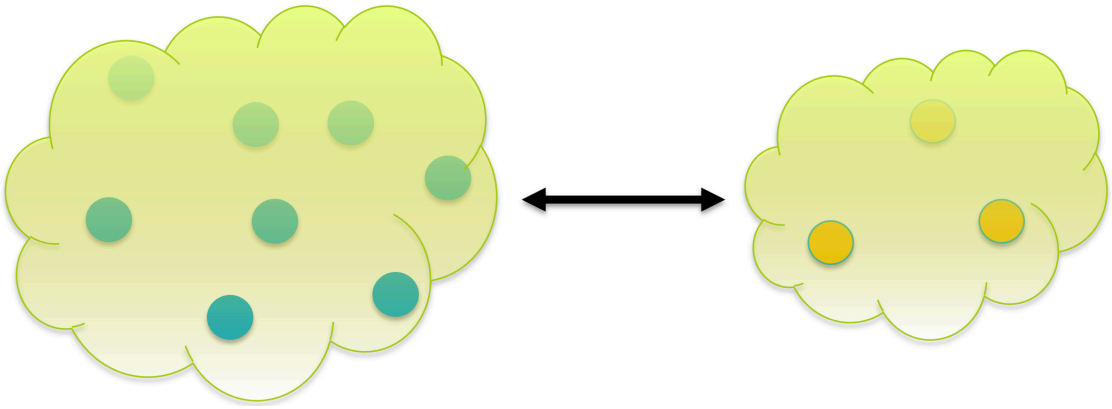


Figure 2.5: We seek a mapping from the original VBS tracers to lower dimensional set of superspecies, and vice versa.

The following subsections build towards the approach focused on for the rest of the thesis: a data driven machine learning approach to find a representative set of superspecies for the VBS tracers. This superspecies set might not be specific to any one process but rather inferred from the large amount of model output for VBS tracers that can be obtained from LOTOS-EUROS runs. Section 2.7.1 briefly summarizes previous work in reduced-complexity techniques in atmospheric science, including collapsing reactions, lumping species, and finding lower-dimensional manifolds of chemical systems. Section 2.7.2 proposes a non-data driven technique that takes advantage of partitioning theory to combine tracers, but could only halve the number of tracers and would not be compatible with phase-specific processes like dry deposition. Section 2.7.3 motivates a data-driven approach to find a representative set of superspecies from the large amount of model output.

2.7.1. Model order reduction in the atmospheric sciences

Reduced complexity approaches have been developed for other atmospheric modeling applications. Chemical mechanisms in CTMs often lump species together, such as larger alkanes: a result of this assumption is that mass is not balanced in chemical reactions (Heald & Kroll, 2020). A well known approximation to reduce the complexity of chemical kinetics (the nonlinear system of differential equation representing chemistry) is the quasi-steady state assumption. The quasi-steady state assumption is applied to chemical species that react on much faster timescales than the reactions of other species, or the timescales of advective transport. Assuming that these fast-reacting species are in equilibrium (or quasi steady-state) implies zero net rates of formation which allows for their concentrations to be represented by algebraic expressions of slower-reacting species and other parameters (Turanyi et al., 1993). This reduces the complexity of the nonlinear ordinary differential chemistry operator, as well as removes their direct dependence on transport. For this reason, some radical tracers in LOTOS-EUROS are skipped in the advection for loop.

There can be more fast reactions than those dictated by the deliberate selection of fast-reacting species in the quasi-steady state approximation. Chemical kinetics can be further simplified using a geometric-based technique called intrinsic low-dimensional manifolds (ILDM), introduced by Maas and Pope, 1992

for combustion applications. This manifold is a lower-dimension surface approximating the reaction space and is determined by the slower timescales of the reactions, neglecting fast timescales. Interest in of low-dimensional manifolds or ”slow manifolds” in the atmospheric sciences was first raised for reduced-order modeling of atmospheric flows (Lorenz & Krishnamurthy, 1987). Lowe and Tomlin, 2000 adapted the ILDM approach from combustion chemistry for tropospheric chemistry simulations. Calculations were done via a repro-modeling approach, representing concentrations of the important variables via polynomial functions of previous concentration and other kinetic parameters, rather than a lookup table for manifold values. In this application, the dimension of manifold showed temporal variation, with day and night modes. Whitehouse et al., 2004 lumped species with similar lifetimes/timescales together, an approach distinct from lumping species by their reactivity or bond types.

More recently, Kelp et al., 2020 used machine learning, specifically a recurrent neural network autoencoder, to find a low-dimensional manifold to model time evolution of a chemical system. Unlike previous attempts to use machine learning approaches for the atmospheric chemical system, this technique did not exhibit exponential error growth upon recurrent calculations over longer-term timescales than optimized for. The training procedure minimized error of recurrent predictions, rather than predictions after after a single timestep. This model, like the LOTOS-EUROS chemistry operator, is a box-model: it is zero dimensional in space, and can be applied to each gridcell individually in a 3-dimensional CTM. Incorporation of this machine learning technique into a CTM was indicated as a future direction of research, including how the compressed features in the hidden layers of the neural network could be used in other operators, like advection.

2.7.2. Zero-order compression technique

The more granular benchmarking in 2.6 concluded that operators like advection and dry deposition are slowed down proportionally more by VBS tracers than the chemistry operator. The chemistry operator includes gas-phase reactions of VOC precursors to OA as well as VBS-specific partitioning calculations. An appropriate reduced order modeling technique would find a set of superspecies that could be used by multiple processes that do not need the full detail of all VBS tracers. This method should provide a mapping back to the original tracer space for processes specific to the organic aerosols: emissions of POA, VBS partitioning, and gas-phase reactions of aVOC and bVOC to the top four volatility bins of the anthropogenic and biogenic VBS classes.

An immmediate parameterization that would reduce the number of tracers to be advected could be realized without losing information (not lossy compression). The minimum information for the VBS to calculate the OA gas/aerosol partitioning is the total concentration, or mixing ratio, of material in every bin, along with the concentration of total organic aerosol (TOA). Using these values would require about half of the information carried by all 4 VBS classes. Instead of the 58 tracers organized by phase, VBS class, and bin, this parameterization would require 30 tracers in total: 29 superspecies corresponding to total concentration (gas and aerosol) in the volatility bins of all classes, as well as a tracer corresponding to TOA.

This can be regarded as a zero-order compression technique, with no information lost on compression and decompression. However, this would only halve the number of tracers added by the VBS. Moreover, superspecies representing total bin concentration would be limited in the processes they could be used in: for example, the dry deposition operator handles aerosol and gas tracers differently.

2.7.3. Machine learning superspecies

In search of a set of superspecies to represent the VBS tracers with a compression factor of more than 2, we turn to unsupervised machine learning methods. The focus of the rest of this thesis is machine learning methods that find patterns in the large amount of concentration data for VBS tracers generated by LOTOS-EUROS simulations. Whenever possible, we will hybridize scientific knowledge of aerosols with machine learning results for physical interpretability. This thesis will aim to develop a machine learned parameterization that can represent the VBS tracers in a lower dimensional space, while assessing:

- accuracy on reconstruction (decompression) to the original tracer space
- physical interpretability of the superspecies and their decompressed tracers
- stability of the ML parameterization when implemented in LOTOS-EUROS, when applied recurrently in longer term simulations on the order of weeks

Chapter 3 explores the main concepts of several unsupervised machine learning methods in the literature of the fields of computer science, applied math, and atmospheric modeling, including non-negative matrix factorization (NMF) and neural network autoencoders. Chapter 4 develops a linear method for the specific application of fast online superspecies compression (and decompression), assessing its accuracy and limitations. Chapter 5 compares a class of linear methods, matrix factorization, to a neural network autoencoder, a nonlinear approach that aims to represent the tracers in a lower dimensional nonlinear manifold. Chapter 5 also explores ways of bringing physical consistency to the data-driven machine learning methods, such as guaranteeing non-negative concentrations, conserving mass, and keeping track of the phase of the superspecies. Chapter 6 assesses accuracy, stability, and robustness of the machine learning superspecies when replacing VBS tracers in the advection operator, implemented online in LOTOS-EUROS. In every operator-splitting timestep before the advection process, the VBS tracers are compressed to superspecies. The advection operator is then applied to the superspecies and relevant non-VBS tracers. After advection, superspecies are decompressed to tracers.

3

Machine Learning

3.1. Overview and applications

The expansive field of machine learning (ML) can be broken down into two broad categories: supervised and unsupervised learning. Supervised ML is a broad category of predicting known targets or labels from corresponding data: examples in this category include linear regression, K-nearest neighbor, random forests, and neural networks. Neural network regression is often used for its ability to emulate nonlinear input-output relationships, for example, how atmospheric concentration of pollutants depends on previous concentrations and meteorological parameters. Creation of *surrogate models* to accelerate computationally intensive classical models using supervised machine learning is an active field of research in atmospheric science (Keller & Evans, 2019; Kelp et al., 2020). Unsupervised ML aims to glean patterns from data without targets – rather than memorize an input-output relationship, instead determine some structure of the data. Examples in this category include projection methods like principle component analysis (PCA), locality preserving projections (He & Niyogi, 2004), and non-negative matrix factorization (NMF) (D. D. Lee & Seung, 1999; Paatero & Tapper, 1994), clustering methods like K-means, and specific types of neural networks like self organizing maps and autoencoders (Marsland, 2014). While unsupervised learning can be used only to find patterns in data, it is also an essential part of data-driven reduced order models, which aim to represent high dimensional systems by a *latent space* with fewer dimensions. A reduced order model can still resolve the original model's fundamental equations, but in the latent space.

Optimizing machine learning approaches on large datasets, often referred to as *training*, has become more and more accessible. Some of this is due to increased processor speed, though processor speeds have largely stabilized over the last decade due to heat dissipation problems (Keyes, 2001). A related but distinct advancement is alternative computing frameworks: for example, exploitation of easily parallelizable operations like matrix multiplication over distributed memory CPU clusters, or the SIMT (single instruction multiple thread) computing paradigm of modern graphics processing units (GPUs). Increased accessibility and speed have led to widespread application of machine learning methods.

Machine learning applications in the atmospheric sciences have existed for decades. The use of multilayer perceptrons in atmospheric modeling to predict the behavior of nonlinear systems emerged in 1990s (Gardner & Dorling, 1998; Potukuchi & Wexler, 1997). Around that time, an unsupervised linear machine learning method called positive matrix factorization was invented by aerosol researchers in Finland (Paatero & Tapper, 1994; Paatero et al., 1991). This technique was later used in image compression, and became widely known in the machine learning community as non-negative matrix factorization, or NMF (D. D. Lee & Seung, 1999). Recent atmospheric modeling applications of machine learning include bias correction to improve model accuracy (Cho et al., 2020; Xu et al., 2021), dimensionality reduction (Drosatou et al., 2019; Kelp et al., 2020), and surrogate modeling of computationally expensive operators (Kelp et al., 2020).

3.2. Matrix factorization methods

This section discusses a linear unsupervised machine learning method and how it can be used to find superspecies that are a linear combination of VBS tracers.

3.2.1. Non-negative Matrix Factorization

Non-negative matrix factorization (NMF), known in some fields as positive matrix factorization, is a form of unsupervised learning used for dimension reduction (D. D. Lee & Seung, 1999; Paatero & Tapper, 1994). Unlike another well known related linear method for dimensionality reduction, principal component analysis (PCA), which is mean centered and can give negative values in principle components, NMF returns only positive values in its reduced dimensions. For this reason, NMF is often chosen in applications where both the lower dimension representation and reconstructed data must stay positive, for example in image compression (D. D. Lee & Seung, 1999), or many mathematical physics problems. NMF has been been used in environmental and atmospheric modeling applications for this reason, where negative values are nonphysical, for example concentration (Paatero & Tapper, 1994). NMF is frequently used in the aerosol community to attribute contributions of source sectors using data from aerosol mass spectrometers (Drosatou et al., 2019). Given a matrix of non-negative data $V \in \mathbb{R}^{m \times n}$ with m dimensions and n data points, NMF aims to approximate V with two non-negative factors:

$$V \approx WH \tag{3.1}$$

where $W \in \mathbb{R}_{\geq 0}^{m \times r}$ is a mapping from the d dimensional space to a lower dimensional latent space with r features, and $H \in \mathbb{R}_{\geq 0}^{r \times n}$ is a representation of each data point in the latent space representation. Though the conventions for matrix shapes and names are inconsistent across the literature, the definitions here have been chosen to be consistent with the seminal paper on NMF for image compression by D. D. Lee and Seung, 1999. D. D. Lee and Seung, 1999 interpret the positive nature of W and H, showing that NMF creates a parts-based representation, where different parts are combined through purely additive operations. W and H are optimized to approximate V based on the minimization of an objective function. One common objective function (called a cost function or loss function) $\mathcal{L}$ measuring the error between WH and V is the square of the Frobenius norm:

$$\mathcal{L} = \|V - WH\|_F^2 = \sum_{i=1}^{m} \sum_{j=1}^{n} (V_{ij} - (WH)_{ij})^2 \tag{3.2}$$

W and H that minimize equation (3.2) are preferred. Sometimes, regularization terms are included in the loss function to prevent either W or H from getting too large. This extends equation 3.2 to

$$\mathcal{L}_{reg} = \|V - WH\|_F^2 + \alpha\left(\|W\|_F^2 + \|H\|_F^2\right) \tag{3.3}$$

where α is a hyperparameter controlling the relative weight of the regularization terms. However, the problem

$$\underset{W,H}{\operatorname{argmin}} \|V - WH\|_F^2 \qquad s.t. \quad W, H \geq 0 \tag{3.4}$$

is NP-hard and ill-posed due to the non-negative constraints (Gillis, 2014). For this reason, local minimization approaches like gradient descent of $\mathcal{L}$ with respect to W and H are often chosen to minimize the difference between X and WH in the Frobenius norm. Gradient descent is an iterative process where, each iteration, W is updated as

$$W \leftarrow W - \eta \frac{\partial \mathcal{L}}{\partial W} \tag{3.5}$$

where the learning rate η is some positive real-valued scalar. H is updated analogously, alternating with the updates for W. An adaptive η often helps with convergence of gradient descent.

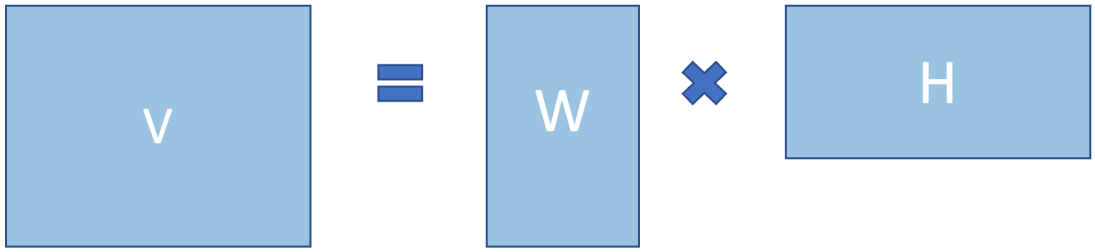


Figure 3.1: Schematic of non-negative matrix factorization.

3.2.2. Pseudoinverse approach

NMF is a matrix factorization method and for that reason operates on batches of data, often using local optimization methods like gradient descent to minimize $V - WH$ in the Frobenius norm, as in equation (3.4). For the purpose of speeding computations, it might be counterproductive to optimize W and H for every new online data point **v**, or synchronize W between different computing nodes when running in parallel.

Assuming that W was optimized on representative data, it can be fixed for subsequent observations, rather than requiring NMF to find new factors every time. This fixed W is a decompression matrix mapping the latent space, which can be thought of a collection of superspecies composed of different amounts of tracers, to original tracer space. However, finding a way to get **h** without using gradient descent also requires a fixed compression matrix B. The compression step becomes

$$\mathbf{h} = B\mathbf{v} \tag{3.6}$$

where $\mathbf{v} \in \mathbb{R}^m_{\geq 0}$ is a new vector of tracer concentrations and $\mathbf{h} \in \mathbb{R}^r_{\geq 0}$ is the latent space representation, a vector of superspecies concentrations. The decompression step is

$$\mathbf{v}_{dec} = W\mathbf{h} \tag{3.7}$$

where $\mathbf{v}_{dec} \in \mathbb{R}^m_{\geq 0}$ is the decompressed vector of tracer concentrations, ideally as close of an approximation to **v** as possible.

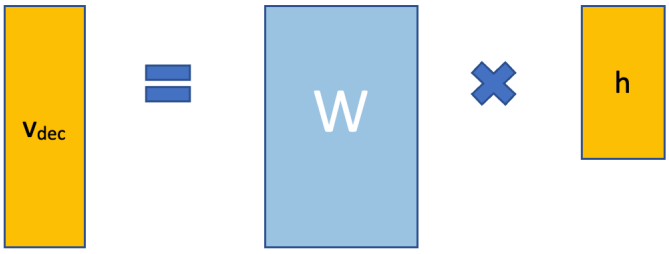


Figure 3.2: Schematic of online decompression from a latent space **h** to decompressed vector $\mathbf{v}_{dec}$, using a fixed W.

Given that W has independent columns (and if it didn't, we could simply remove a column and have one less latent dimension), W^TW is invertible and a left pseudoinverse W^+ can be calculated:

$$W^+ = (W^TW)^{-1}W^T \tag{3.8}$$

which, applied to equation (3.7), gives the desired latent space representation

$$\mathbf{h} = W^+\mathbf{v} \tag{3.9}$$

Provided that W is a suitable mapping, W^+ can be calculated once, requiring only a single matrix-vector multiplication for every **h**. Online NMF updating is also a field of research, and if W needs to be updated in an online fashion, this route could be further explored (Kim et al., 2014) as an extension of the methods developed in this thesis.

3.2.3. Non-negative compression

If tracers are compressed to a single superspecies, W is a vector, and W^TW is a scalar that will be at least zero, with equality only when W is zero (which wouldn't be very useful in our application). The inverse of positive scalar W^TW will also be positive. However, this does not generalize to larger matrices with more latent features. It is possible that some elements of $(W^TW)^{-1}$ are negative, which would lead to negative elements in the Moore-Penrose inverse W^+, and potentially **h**. It must be seen if the pseudoinverse approach leads to negative values for either the superspecies or the decompressed tracers, and if such negative values are large enough in magnitude to cause problems in the model, like systematic removal of mass.

The NMF approach can be adapted to give a non-negative compression matrix B. First, NMF can be applied to find a decompression matrix W mapping superspecies to decompressed tracers and a matrix of data H corresponding to the latent space representation of the original data X. Then, the compression matrix B can be gained from the objective function

$$\underset{B}{\operatorname{argmin}} \|H - BV\|_F^2 \qquad s.t. \quad B \geq 0 \tag{3.10}$$

where V is given, and H is found in the first step. These two steps together result in a compression matrix B optimized to transform V to an approximation of H, and a decompression matrix W optimized to transform H to an approximation of V. Expression (3.10) is a less common objective than the normal NMF goal (3.4), but fixing V and H is possible within the scikit-learn Python package for non-negative factorization (Pedregosa et al., 2011). It should be noted that scikit-learn uses different shapes and definitions for the matrices, but is equivalent.

The two steps could be combined to yield a single objective function

$$\underset{B,W}{\operatorname{argmin}} \|V - WBV\|_F^2 \qquad s.t. \quad B, W \geq 0 \tag{3.11}$$

although this objective is not readily found in the literature, and not possible to do with the widespread scikit-learn NMF package. This objective has similarities to a technique known as archetypal analysis (Cutler & Breiman, 1994). Archetypal analysis combines data points in each dimension rather than dimensions of each data point, and has additional constraints on B and W.

3.3. Neural Networks

There exist many sources detailing neural networks: Chapter 4 from Marsland, 2014 is used to inform the summary below. Artificial neural networks are comprised of interconnected perceptrons, or nodes. These nodes are arranged in layers – layers that are in between the input and output are called hidden layers. In a fully connected feed forward neural network, each node receives as input a vector which is the collection of outputs from all the nodes in the previous layer.

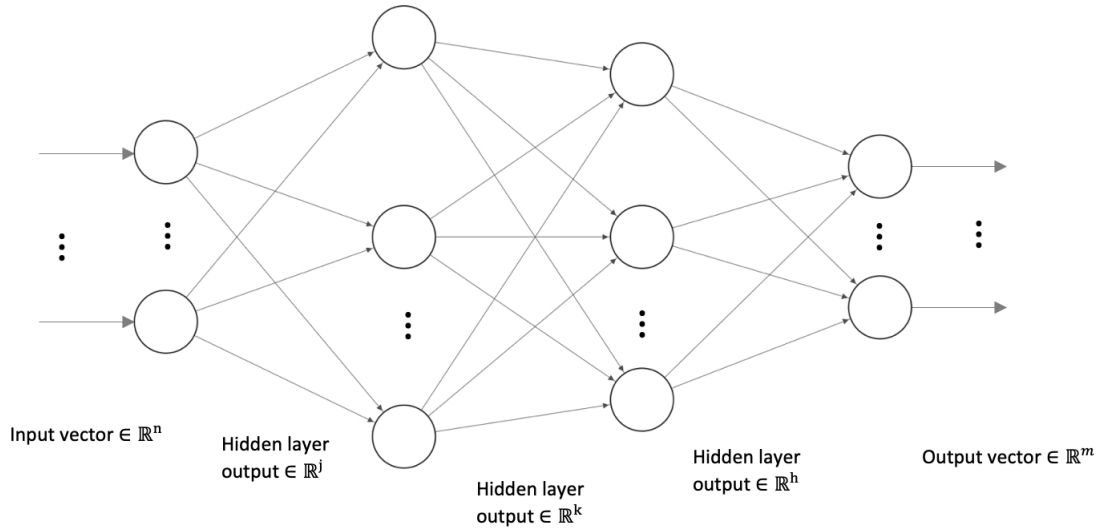


Figure 3.3: A sketch of a fully connected feedforward neural network predicting an m-dimensional output from n-dimensional input. Nodes are represented by circles. The hidden layers output vectors of dimensions j, k and h to all the nodes of the subsequent layers. This figure was in part developed using the tool from LeNail, 2020.

Within each node, an inner product between weight parameter vector **w** and input **x** is calculated. Then another weight parameter b known as the bias is added. Finally, this scalar $s = \mathbf{w}^T\mathbf{x} + b$ is given to some activation function $a(s)$. An expression for the full transformation of the data in a single node therefore looks like:

$$a(\mathbf{w}^T\mathbf{x} + b) \tag{3.12}$$

The activation function a can be identity, but is often chosen to be a non-linear function. One common class of functions chosen is the sigmoid or ”S”-shaped functions, for example, the logistic function:

$$a(s) = \frac{1}{1 + e^{-s}} \tag{3.13}$$

ranging from 0 to 1. Another frequently used sigmoid function is hyperbolic tangent, which ranges from -1 to 1. Sigmoid functions in this context were originally chosen to mimic action potentials in brains that govern whether neurons fire. The scalar values of all the nodes in one layer are fed to the next layer of nodes, with new weights.

The values of weight parameters **w** and b can be varied to minimize a loss function $\mathcal{L}$, which is a measure of the error between the NN output **y** and a target $\mathbf{y}_{true}$. Typical loss functions are mean squared error:

$$\mathcal{L}_{MSE} = \|\mathbf{y}_{true} - \mathbf{y}\|_2^2 \tag{3.14}$$

or regularized mean squared error (used in ridge regression), where a term $\lambda\|\mathbf{w}\|$ is added to equation (3.14) with some norm $\|\cdot\|$ and positive-valued scalar λ, that aims to constrain the weight parameters $\mathbf{w}$.

Parameter adjustment is usually done through gradient descent methods, though other minimization functions exist. Stochastic gradient descent updates all weights for a single data point, rather than for all of the training data, and is often chosen. Mini-batch gradient descent updates weights for a set of the training data at a time. As mentioned in the previous section, adaptive learning rates have been shown to improve convergence, such as the *Adam* algorithm (Kingma & Ba, 2014). The process of performing gradient descent from the output layer backwards through the hidden layers, adjusting the parameters along the way, is known as backpropagation (Marsland, 2014).

Neural networks' useful property of being able to approximate any non-linear smooth function with an unrestricted amount of nodes in a single hidden layer is known as the Universal Approximation Theorem. Cybenko, 1989 showed this property using sigmoid activation functions, and it was later shown to hold true for general non-polynomial activation functions (Leshno et al., 1993).

3.4. Autoencoders and latent space representation

A neural network can be trained to reproduce the input it receives, in a process called auto-associative learning (Marsland, 2014). This class of neural network is called an autoencoder. As an autoencoder does not require learning targets beyond its input data, it is considered an unsupervised learning algorithm.

The architecture of an autoencoder is often chosen to perform compression in a hidden layer. This hidden layer has fewer nodes than the input and output layer. If the neural network in Figure 3.4 were an autoencoder with hidden layer compression, then the output layer size m would be equal to input layer size n, and at least one hidden layer size j, k, h would be less than m; more concisely, $min(j, k, h) < m$. An example of autoencoding a vector of length 9 into a latent space representation with only four elements is shown below.

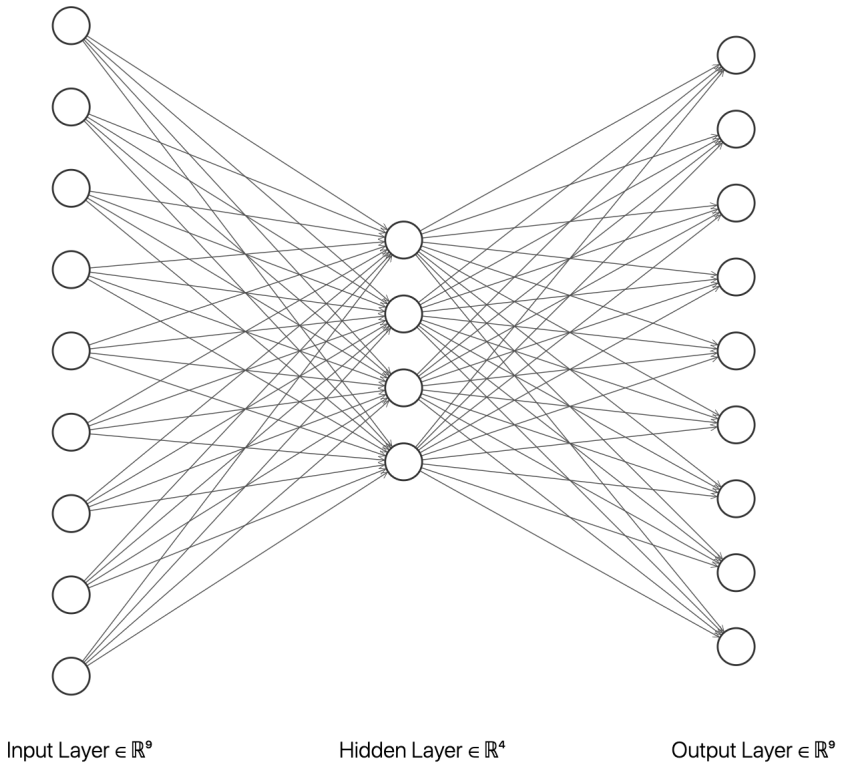


Figure 3.4: A sketch of an autoencoder encoding a vector with 9 elements into a reduced dimension representation with only four elements. This figure was developed using the tool from LeNail, 2020.

Though the autoencoder approximates the identity function, it does so through compression then decompression. The first set of hidden layers transform the input to a smaller vector, representing the data in a smaller latent space. The second set of hidden layers reconstructs the original data. Both layers have weight parameters that are trained (often via gradient descent methods) to reconstruct the original data in a way that minimizes some cost function (often mean square error). These layers are called the encoder, and decoder, respectively: the latent space representation is called the code. Activation functions in the hidden layers autoencoder can be chosen to be nonlinear, making the output nonlinear with respect to the weights **w**. This allows the latent space to be something other than a linear combination of the original variables, unlike NMF.

Autoencoders have been used in reduced-order models in fluid mechanics applications, where advection calculations are the computational bottleneck (K. Lee & Carlberg, 2020; Maulik et al., 2021). More specifically, K. Lee and Carlberg, 2020 introduced a convolutional autoencoder to create a lower dimension nonlinear manifold on which to solve general governing equations. An example given was advection-dominated problems, where linear subspace methods like proper orthogonal decomposition

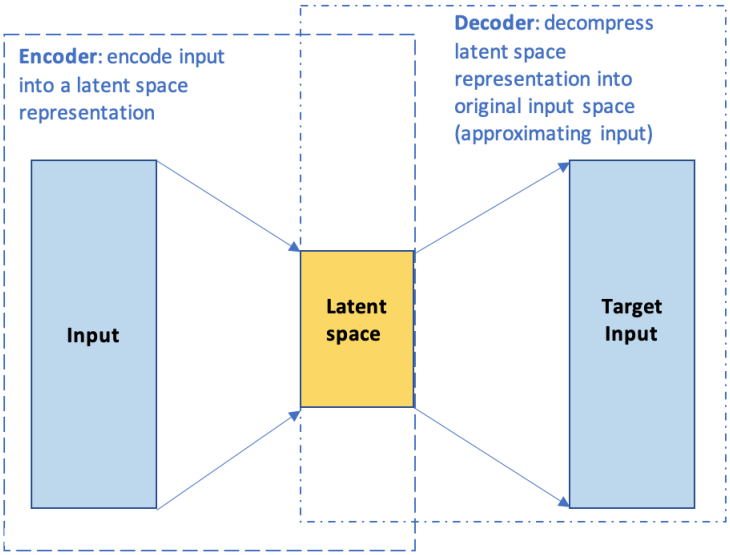


Figure 3.5: The parts of an autoencoder: encoder, code, and decoder.

(POD, equivalent to PCA) are often insufficient. Maulik et al., 2021 used a similar convolutional autoencoder, but within a recurrent neural network to step a nonlinear latent space forward in time, rather than solving the governing equations in the latent space. This approach was found to be more suitable for advection-dominated systems than a linear subspace approach like POD: examples given were the viscous Burgers equation with an advecting shock, and the shallow water equations.

In chemical transport models, literature on using autoencoders for reduced order modeling for advection dominated systems is sparse. A recent PhD thesis focusing on reduced order methods for data assimilation in an urban air quality model only briefly mentions neural networks – however, concludes similarly to K. Lee and Carlberg, 2020 and Maulik et al., 2021 that projection-based model order reduction methods have limited efficacy in advection dominated cases (Hammond, 2017).

Kelp et al., 2020 use a combination somewhat analogous to the Maulik et al., 2021 approach, using autoencoding layers within a recurrent neural network, in a surrogate model for a paired box model of gas-phase chemistry and aerosol microphysics, MOSAIC/CBM-Z (Zaveri et al., 2008). They applied this approach to estimate the computationally intensive time integration step. The surrogate model is orders of magnitude faster, and remains stable even when running on longer timescales than it was optimized for. In order to incorporate a surrogate model of this architecture in a CTM or ESM, Kelp et al., 2020 point to a future research direction: assessing how other processes, like advection, handle the compressed latent space tracers. Methods developed in chapters 4 and 5 will be applied in chapter 6 to answer a related question: can NMF or a neural network autoencoder find a latent space representation of volatility basis set tracers to be advected? Is the chosen linear or nonlinear method accurate, over longer simulation times of several weeks?

4

Offline Machine Learning: NMF/Pseudoinverse Reconstruction

Non-negative matrix factorization (NMF) as detailed in chapter 3 was performed on winter data over the default LOTOS-EUROS domain for 4 volatility basis sets. This resulted in a mapping from original tracers to a lower dimensional latent space, which can be physically interpreted as a set of superspecies. From the NMF, the mapping W and its Moore-Penrose left pseudoinverse W^+ are used to transform between the volatility basis set and the superspecies on new test data without performing NMF. This chapter develops the approach for one superspecies, then evaluates it on differing amounts of superspecies, making a preliminary judgement on a reasonable extent of compression. This chapter goes on to explore the limitations of this approach both quantitatively and qualitatively.

4.1. Model settings and data

The online NMF approach was tested on a LOTOS-EUROS run from February 15th through 28th, 2018. Hourly surface concentrations for 58 tracers from the four volatility basis sets were reported on the Monitoring Atmospheric Composition and Climate (MACC) domain, at a resolution of 0.50 degrees by 0.25 degrees, which is about 36 km by 25 degrees at 50 degrees North (Manders-Groot et al., 2021).

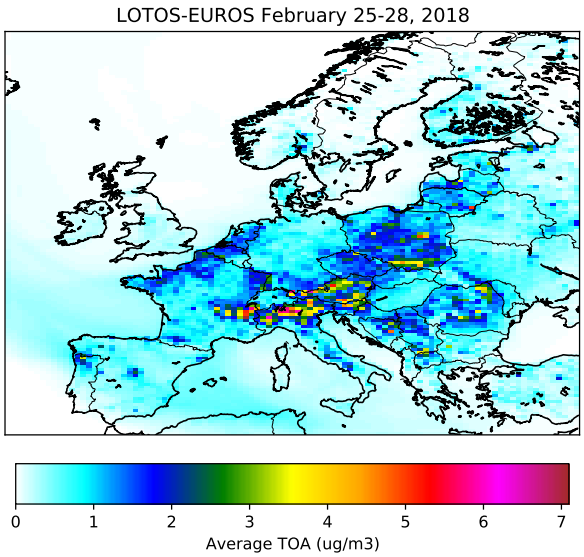


Figure 4.1: Total organic aerosol simulated in LOTOS-EUROS for the test days February 25-28, 2018.

The days of February 15th through 19th were used as model spin-up: data from these days was disregarded, leaving 9 days for training and validation. With 216 hours, 100 latitudinal gridlines, and 140 longitudinal gridlines, there are about 3 million data points for each volatility basis set. The data points range from 12-dimensional (from anthropogenic and biogenic gaseous precursors) to 18-dimensional, in the case of primary organic emissions. Data over the domain from February 20th through 24th, approximately 1.7 million data points, were used as training data to generate an optimal W for each VBS. The subsequent four days February 25th through 28th, approximately 1.3 million data points, were used for testing how well the tracers-to-features mapping matrices W and their Moore-Penrose left pseudoinverses W^+ work on data that was not used in their optimization.

Figure 4.1 shows the LOTOS-EUROS run, and will be taken as ground truth for the sake of this thesis: all methods will be judged on how well they replicate the LOTOS-EUROS data.

4.2. Recap of non-negative matrix factorization

NMF is an unsupervised machine learning approach used in this application on data $V \in \mathbb{R}_{\geq 0}^{m \times n}$ with m rows corresponding to VBS tracers and n columns corresponding to data points. Two optimal, non-negative factors, $W \in \mathbb{R}_{\geq 0}^{m \times r}$ and $H \in \mathbb{R}_{\geq 0}^{r \times n}$ are found via gradient descent methods, to approximate V via the product WH. On a high level, W can be thought of intuitively as a mapping from the original tracer space in m dimensions into a lower dimensional latent space with r dimensions. The latent space can further be interpreted of as some sort of *superspecies* space, where each dimension is some sort of combination of the original tracers. H is each data point represented in the superspecies space.

4.3. Pseudoinverse approach

One ultimate goal is to be able to perform the compression to a set of superspecies online, with new predictions. Using gradient descent methods to find an optimal W and H might be too time-consuming to do for each data point **v**, where **v** is the set of tracers for a specific VBS class at a gridcell at a given time. Online optimization of W would bring an added complication of when to communicate different W across gridcells. Instead, we can use representative training data to find an optimal W. A Moore-Penrose left pseudoinverse W^+, defined as

$$W^+ := (W^T W)^{-1} W^T \tag{4.1}$$

can be used to calculate a latent space data point **h** from a data point **v** as follows:

$$\mathbf{h} = W^+ \mathbf{v} \tag{4.2}$$

The latent vector **h**, which can be thought of as a superspecies, is a lower dimensional representation of **v**, that can be computationally more efficient to perform other operations on (e.g. parallel communication between processing nodes in domain decomposition, advection, deposition). Finally, a decompressed data point $\mathbf{v}_{dec}$ can be obtained via

$$\mathbf{v}_{dec} = W\mathbf{h} \tag{4.3}$$

4.4. A single superspecies

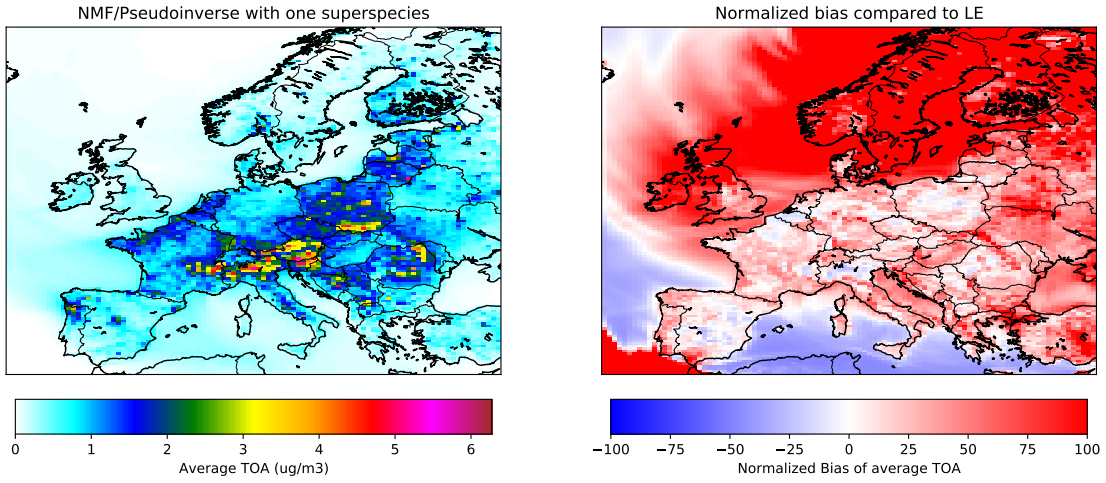
4.4.1. Spatial patterns

NMF is considered *lossy* compression, meaning information is lost during the linear transformation to the latent space and back again. To get an estimate for a lower bound of accuracy in compression, NMF with one single latent feature (one *superspecies*) was performed on each of the volatility basis sets. This approach is extremely limited as it compresses the volatility basis set into a characteristic vector of rank one. The following results quantify and visualize the extent to which all the partitioning distributions over 4 days and an entire domain can be represented with one direction. Table 4.1 shows error metrics on the test data from February 25th through 28th. Mean value for each test set, as well as total organic aerosol (TOA) and total organic material (TOM) are reported to give a sense of the magnitudes of error. It can be seen that error values of decompression from a single superspecies, especially RMSE, are quite high relative to average values.

Table 4.1: Evaluation data set metrics when compressing to a single latent feature

	RMSE [$\mu g\ m^{-3}$]	Bias [$\mu g\ m^{-3}$]	Mean [$\mu g\ m^{-3}$]
aVOC	0.0021	-3.9×10^{-6}	0.0043
bVOC	0.0061	2.9×10^{-4}	0.0262
POA	0.0441	-0.0021	0.0558
siSOA	0.0205	6.4×10^{-5}	0.0153
TOA	0.266	0.094	0.386
TOM	0.0978	-0.0328	1.61

Figure 4.1 visualizes organic aerosol (TOA) over the domain averaged over the 4 test days. Figure 4.2a shows the corresponding reconstruction using the optimal W from the training data, and Figure 4.2b the relative bias between the two. The compression and decompression with a single vector for W is able to visually reproduce areas of average higher concentrations on the regional scale, for example the Po Basin in northern Italy.



(a) NMF/Pseudoinverse with one superspecies

(b) Relative bias compared to LOTOS-EUROS test data (Figure 4.1)

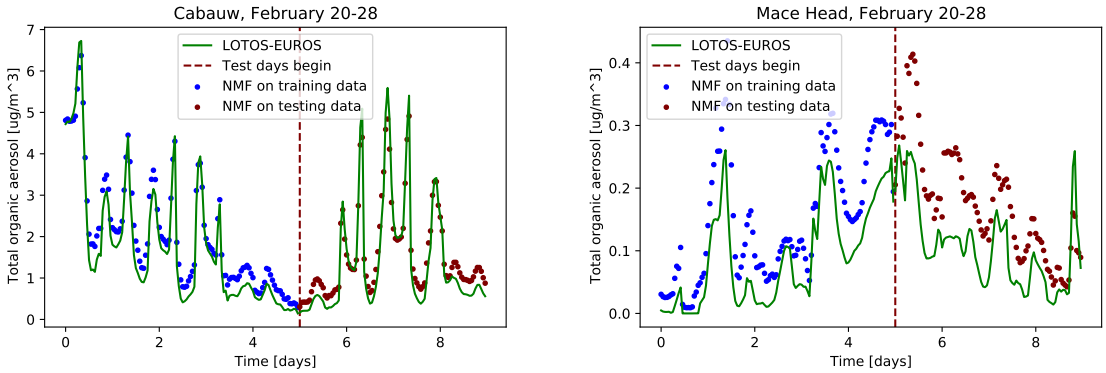
Figure 4.2: Average TOA (a) for the test days Febraury 25-28, 2018, via compression with a pseudoinverse W^+ and decompression with NMF matrix W. It can be seen that this image has higher amounts of average TOA compared to Figure 4.1. This difference is illustrated in figure (b) by the relative bias between (a) and Figure 4.1

Figures 4.1 and 4.2a are visually similar, especially in areas where there is appreciable TOA. Figure 4.2b shows large relative errors only occurring in places with negligible (near zero) organic aerosol in LOTOS-EUROS. Though the color bar scale is 1 to 100, relative error near the Arctic (north border) and Atlantic (western border) is sometimes much larger, but since the concentrations are so small, this might

not be an issue in terms of model prediction. However, it could be problematic if the NMF approach artificially introduces a significant amount of mass into the system near the edge of the domain. This reconstruction of time-averaged TOA for the test days indicates that a single superspecies is able, albeit somewhat limitedly, to capture spatial variability over much of continental Europe.

4.4.2. Temporal patterns

Figure 4.3 investigates the temporal behavior of TOA, which is quite periodic, at two stations: the Cabauw Experimental Site for Atmospheric Research in the Netherlands, and Mace Head Atmospheric Research Station in Ireland. In green (dots before the test days begin) are the sum of the concentrations of VBS aerosol tracers of the training days using an NMF reconstruction. The red dots after the test days begin are reconstructions using the left inverse to project each VBS into a one-dimensional space, then multiplication by the vector W to get an approximation of each VBS. The total organic aerosol concentration is then summed up and compared to the LOTOS-EUROS predictions, the solid black line. Total organic aerosol seems to be quite well predicted for Cabauw. This was also the case for other areas in the domain including Melplitz, Helsinki, and Barcelona. Mace Head, which is regarded as a more pristine station (O'Dowd et al., 2014), shows systematic overestimation of TOA using 1 feature to represent each bin. However, the TOA estimated using decompressed aerosols manages to capture the diurnal pattern of TOA for both stations.



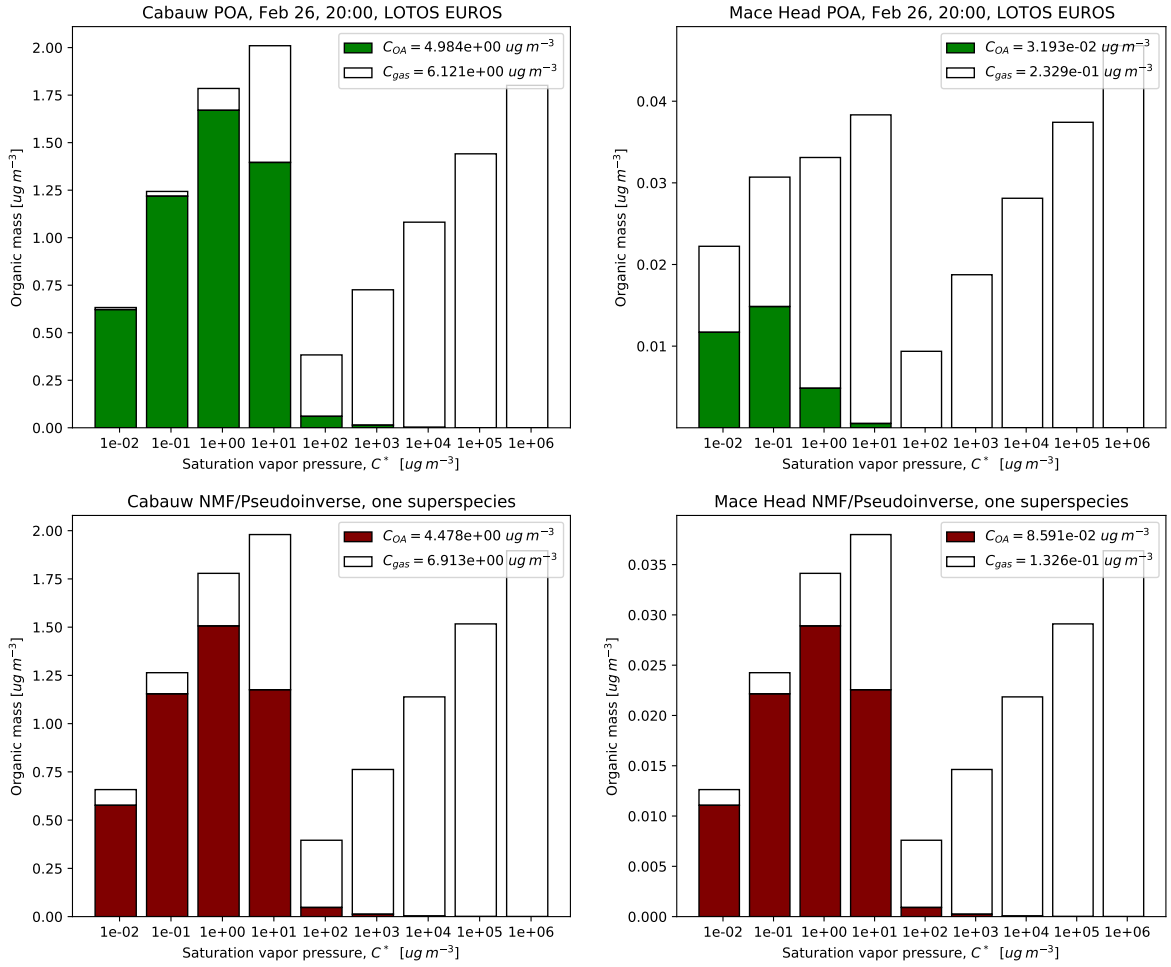
(a) Cabauw, single superspecies

(b) Mace Head, single superspecies

Figure 4.3: Simulated TOA over the training and test period, after compressing each VBS class to a single superspecies using W^+ and subsequent decompression via W for Cabauw and Mace Head. Note that TOA is about an order of magnitude higher at Cabauw than Mace Head.

4.4.3. Mass distribution over volatility bins

The ability of a single characteristic vector W per VBS to model TOA is not necessarily indicative of accurate VBS reconstructions. Using a single characteristic vector will fix the shape of the distribution, and a single latent point h will scale the distribution. W is of rank one, so the VBS reconstructed from the projection via W^+ into a 1D latent dimension (one single superspecies) will always be one shape. Put another way, each NMF feature, which can be thought of as a representative superspecies, has a certain composition of each original tracer. If we only use one single feature, the relative compositions will not change, only the magnitude of the distribution (which can be thought of as increasing or decreasing the concentration of the single superspecies).



(a) Cabauw (b) Mace Head

Figure 4.4: Plots of the POA volatility basis set at (a) Cabauw and (b) Mace Head at 20:00 on February 26th. The upper row in green is the VBS distribution from LOTOS-EUROS, and the bottom row in maroon is the reconstruction after compression into one superspecies and subsequent decompression. Note that this method, when using a single superspecies, is limited to reconstructing the VBS with one fixed shape as determined by the characteristic vector W. This results in the decompressed POA VBS having the same shape for both Mace Head and Cabauw, despite very different magnitudes and target shapes.

The bottom row of Figure 4.4 shows the limitation of using a single superspecies in a linear approach, in that decompression is locked into a single shape. Cabauw and Mace Head show different proportions of C_{OA} to total concentration in the POA VBS tracers, as well as different relative amounts of semivolatile gas phase tracers. The use of a single superspecies manages to capture magnitude: this information is communicated by its concentration. However, its decompression is limited to a single characteristic vector, resulting in the same shape per class of VBS. This will fail in capturing the spatio-temporal variability of the shape of the VBS across the whole domain and time period.

While a single superspecies per VBS can somewhat capture surface TOA, this is very limited in capturing the distribution. This implies that offline reconstruction of total organic aerosol is fairly robust to the unsupervised machine learning approach, and alone not the best metric in which to evaluate methods for compressing the volatility basis sets. Using a single shape for reconstruction might not be completely meaningless, especially for the POA VBS: primary emissions are distributed in an invariant shape over the volatility bins, and a single superspecies captures this to a certain extent. Figure 4.5 shows that the shape of the superspecies is quite similar to the fractions that determine how primary emissions are distributed over the bins of the POA VBS. This indicates that NMF, which starts with a randomly initialized W, is able to learn a realistic and physically interpretable pattern from the data and include that in the superspecies shape.

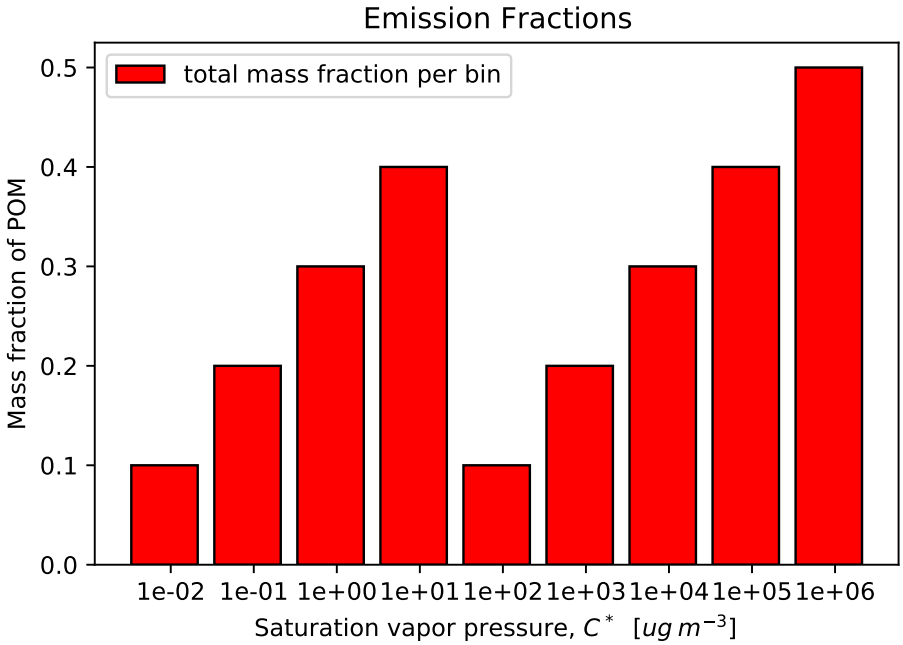


Figure 4.5: Mass fractions of primary emissions distributed over the volatility bins. Note that partitioning is not considered here, just the relative amounts to each bin, summing to a total of 2.5 as introduced in section 2.5.1.

The single basis direction of 1 superspecies might work well in areas dominated by primary emissions, but will not be able to capture variations of distributions after subsequent ageing. Offline error metrics of compressing to a single species are not able to provide insight into possible effects of fixing the shape of the distribution recurrently online in LOTOS-EUROS. This is expected to counteract the model's aim to capture oxidative aging of organic aerosol and its tendency to become less volatile, by redistributing material in the exact same way upon each decompression. The desire to allow for different shapes in a VBS motivates the use of multiple superspecies: NMF with more than 1 latent dimension. Section 4.5 explores the effect of increasing number of superspecies (and thus characteristic vectors) on the RMSE and bias accuracy metrics.

4.5. Compression factor and accuracy

The number of NMF features was varied to explore effect of compression on accuracy. NMF for 1 through 6 features was performed for each of the volatility basis sets. Bias and RMSE were calculated for the tracers from each VBS class, as well as total organic aerosol. An overview of these results can be found in Figure 4.6.

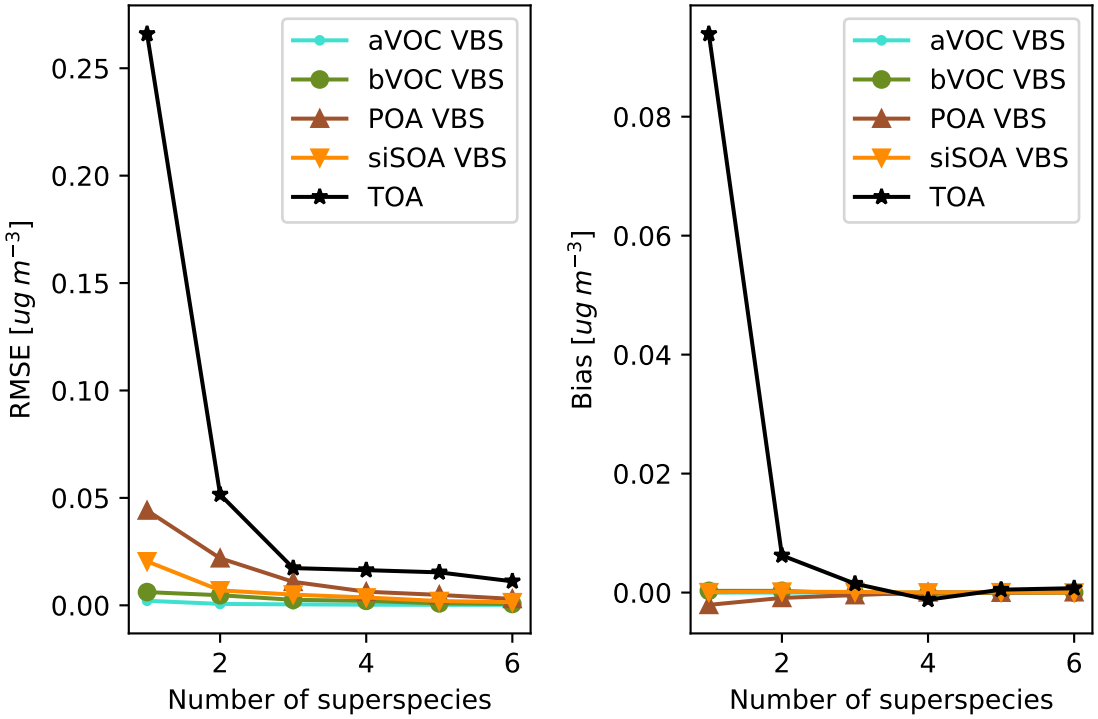


Figure 4.6: Plots of RMSE and bias for the 4 VBS classes, as well as TOA.

These results are dependent on the (seeded) randomly initialized weights. For a confidence interval of accuracy, this experiment could be run several times for each amount of superspecies and then averages of RMSE and bias metrics reported. However, analyzing results from a single run, the steep improvement between 1 and 3 is visually apparent. RMSE monotonically decreases with increasing number of superspecies, though there are some diminishing returns to accuracy when increasing the number of superspecies after 3. However, more superspecies mean more compressed tracers to advect, which will affect online performance (computational speed) in LOTOS-EUROS. This inevitable tradeoff between accuracy and computational speed is a canonical problem not only in atmospheric chemistry modeling, but more generally in simulation of complex systems. In this case, 3 superspecies seems to strike a good balance, ranging from a compression factor of 4 (aVOC and bVOC basis sets) to 6 (POA basis set) with a significant improvement from 2 superspecies and minimal improvement when using 4 or more superspecies.

4.6. Three superspecies

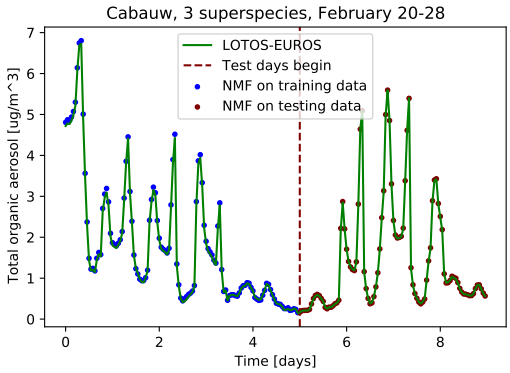
The analysis in section 4.4 was repeated for the NMF/pseudoinverse approach using 3 superspecies. Additionally, the condition numbers $\kappa(W^TW)$ for the symmetric W^TW matrix for each VBS are given. This is an important thing to check as poorly conditioned matrices can lead to numerical instability in inversion and creation of the pseudoinverse W^+. This was not reported when using 1 feature, as W^TW was a scalar, and its inverse therefore its reciprocal.

It can be seen that the symmetric matrix W^TW is well conditioned for all VBS classes, and is not ex-

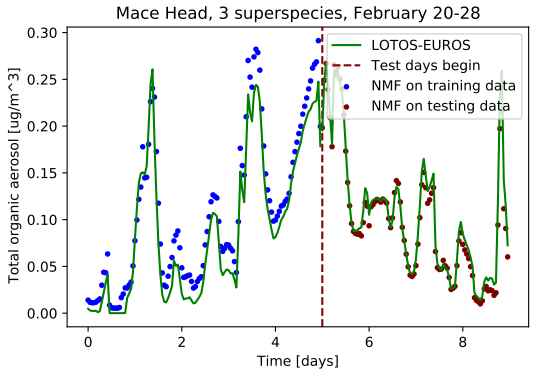
Table 4.2: Evaluation data set metrics when compressing to 3 latent features

	RMSE [$\mu g\ m^{-3}$]	Bias [$\mu g\ m^{-3}$]	$\kappa(W^T W)$
aVOC	4.4×10^{-4}	2.6×10^{-5}	43.5
bVOC	0.0026	-1.6×10^{-4}	48.2
POA	0.0109	-4.2×10^{-4}	16.9
siSOA	0.0050	-9.9×10^{-5}	20.9
TOA	0.0173	0.0015	-
TOM	0.0547	-0.00763	-

pected to cause issues with numerical stability. It should be noted that the exact values are sensitive to differences in the random initialization of W and H in NMF, a general issue with gradient descent optimization and stochastic methods. For reproducibility, random seed values were specified in the code, but more informative would be conducting multiple different runs with different seed values to obtain a confidence interval for the metrics. However, a single arbitrarily seeded run can already provide some insight. Bias values of some of the volatility basis sets are reduced by a factor of approximately 5. RMSE for TOA is reduced by a factor of 15 and bias by a factor of 60 from using a single superspecies. This improvement in accuracy can be seen in the timeseries for both Cabauw and Mace Head in Figure 4.7, where Mace Head shows a significant decrease in overestimation of TOA and Cabauw's decompressed TOA is visually indistinguishable from the LOTOS-EUROS predictions of TOA.



(a) Cabauw, single superspecies



(b) Mace Head, single superspecies

Figure 4.7: Simulated TOA over the training and test period, after compressing each VBS class to 3 superspecies using W^+ and subsequent decompression via W for Cabauw and Mace Head. This shows improved predictions of TOA over using just one single superspecies in Figure 4.3.

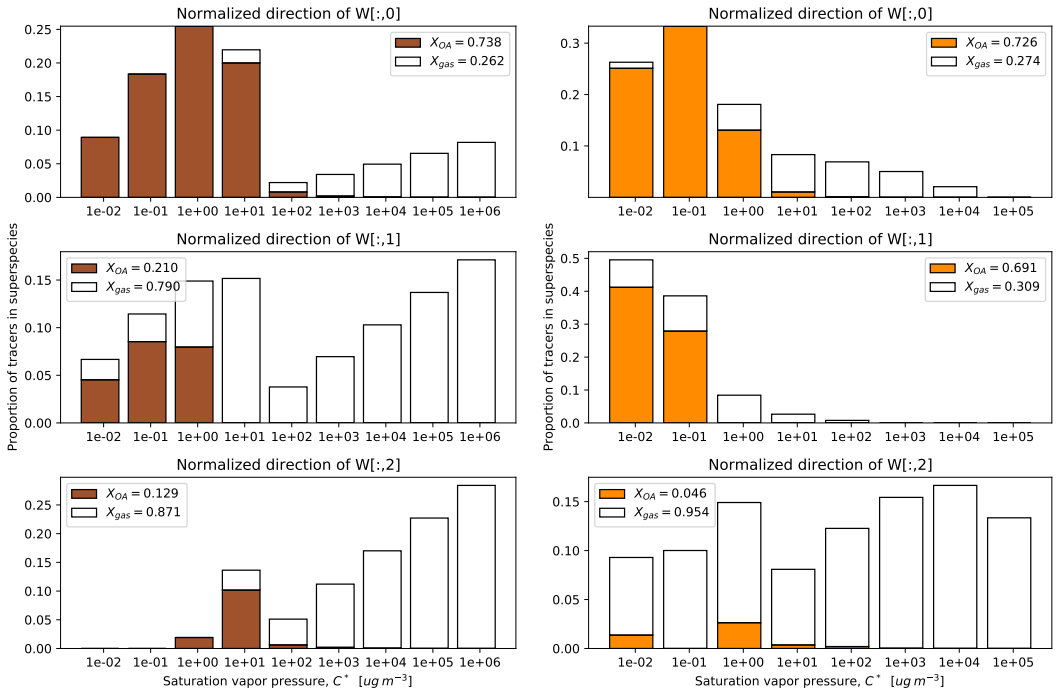
The improvement in accuracy can be attributed to increased degrees of freedom: the distribution for each VBS class can be reconstructed by 3 characteristic vectors, whose magnitudes are determined by the value of their corresponding superspecies. The more superspecies, the more degrees of freedom. Figure 4.8 in section 4.7 shows the three characteristic shapes, in the context of developing physical intuition for the results of this unsupervised machine learning approach.

4.7. Towards physical interpretability

The improvement in some of the VBS metrics when using 3 superspecies is likely due to the additional degrees of freedom available to reconstruct the distributions. Rather than fixing the shape of the distribution, and using the scalar concentration of a single superspecies to scale the distribution, a combination of 3 several superspecies of different compositions is added. The relative amount of each superspecies determines the shape of the final distribution. Figure 4.8 plots the three columns of the decompression matrices W optimized for the POA VBS and siSOA VBS. Each column is scaled by its sum. Upon normalization, each column can be interpreted as the composition of 1 superspecies in terms of the original tracers. The concentration of each superspecies is then given by the corresponding

element in the superspecies vector $h = [h_1, h_2, h_3]^T$.

Degrees of freedom added by superspecies allow modeling of specific sources, or regimes that have specific distributions. Freshly emitted aerosol will have a different mass distribution over volatility bins than aged aerosol. Use of multiple superspecies allows the distribution to be a linear combination of the specific regimes.



(a) 3 POA VBS superspecies

(b) 3 siSOA VBS superspecies

Figure 4.8: Plots of the 3 columns of W for the POA and siSOA VBS, each normalized by their column sum, when using NMF with three features. These vectors are physically interpretable as the relative composition the three superspecies.

The legends in Figure 4.8 indicate the fraction of each superspecies (column of W) that is in the aerosol phase, X_{OA}, and gas phase X_{gas}. It can be seen that the first superspecies (column W[:,0]) is composed primarily of aerosol phase material in the semivolatile bins, and the third superspecies contains most of the gas phase material, and is mostly gaseous. The physical intuition here is that W and B become matrices of composition. Each column of W is the tracer composition of a superspecies. Similarly, for W^+, each tracer is fully distributed over each superspecies, no more and no less. One potential drawback is the manipulation of columns in W^+, which might detract from the optimal directions found in NMF by gradient descent. Another limitation of this strategy is that if W^+ has negative values, which is a potential issue raised in section 3.2.3, we lose the physical intuition of composition matrices.

If instead of the pseudoinverse W^+ with a positive compression matrix B is used with the same constraint on the columns, then mass would be conserved. This strategy, as well as other mass-conserving approaches, is explored in Chapter 5.

4.8. Negative concentrations

Section 3.2.3 raised the theoretical possibility of the pseudoinverse compression approach leading to negative values, even when the decompression matrix W from NMF contains all non-negative elements. This problem is valid when compressing to more than one superspecies, which may be desired if we want to represent the VBS distribution with more than one characteristic vector. Returning to this theoretical problem, it is seen that in practice negative values are indeed created from this method.

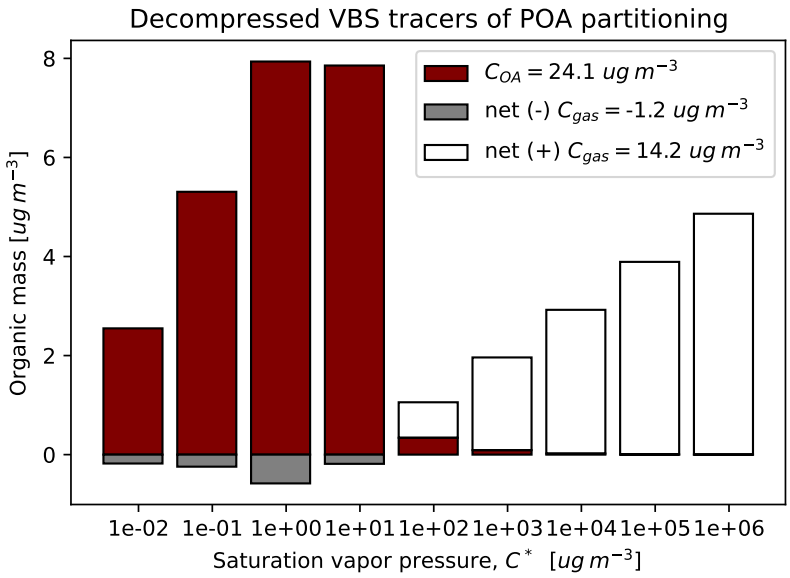


Figure 4.9: Negative values of POA VBS tracers after decompression. The gridcell containing Bergamo in northern Italy was chosen as a representative example, at 8 am EST, February 26, 2018. Outside the scope of this thesis, but relevant in the intersection of air quality and public health: the city of Bergamo has the second highest mortality burden from $PM_{2.5}$ exceedances in all of Europe (Khomenko et al., 2021).

Negative concentrations that are extremely small in magnitude can be approximated as zero. This tolerance can of course be set, for example $1 \times 10^{-8}\ \mu g\ m^{-3}$. However, over the entire test data, there are over 4.7 million cases in the test data where a POA VBS tracer is below $-1 \times 10^{-8}\ \mu g\ m^{-3}$, more than 19% of the 24 million values in the test data for the POA VBS.

A more relative tolerance can be chosen: for instance, all concentrations that are more negative than the bias for each VBS. These ”significantly negative” concentrations would be negative even after an additive bias correction. For the POA VBS, there were 855083 such concentrations, about 3.5% of the total test data. The other VBS classes showed similar proportions of ”significantly negative” concentrations: 4.2%, 5.6%, and 7.0% respectively for the siVOC, aVOC, and bVOC VBS classes (for the anthropogenic VBS and siVOC VBS, which had positive biases, the tolerance was chosen to be the negative absolute value of the bias). Since a bias correction would not make such concentrations positive, they could be set to zero. However, these aren't near-zero negative concentrations. In the example in Figure 4.9, the minimum value is -0.58 $\mu g\ m^{-3}$. Artificial reassignment of significantly negative values removes material from the model systematically. Moreover, this would occur at least once every operator splitting timestep, as tracers are compressed for advection, deposition, or other processes. This recurrent behavior may compound on itself in an online setting. Additionally, negative values for superspecies (before decompression) have potential to cause numerical problems when passed to other processes.

The spatio-temporal variability of negative concentrations from this method could be analyzed. Some information could already be gained from looking at the elements of W^+ and W, seeing which tracers correspond to the most negative elements of W^+, and to which superspecies that corresponds to. Such interpretability is an advantage of the linear method. The spatio-temporal behavior of said tracers could be explored on the domain and time period, and perhaps give insight into failure modes of the NMF/pseudoinverse approach: under what conditions the parameterization should not be used. Rather than further exploring the limitations of the NMF/pseudoinverse approach, we choose instead to develop methods that allow for multiple superspecies while preventing negative values entirely. The following chapter explores methods that restrict both the superspecies and the decompressed tracers to their non-negative half spaces. The introduction of more physical knowledge into the machine learning approaches are explored, including mass conservation and phase-specific compression into purely gas and purely aerosol species for improved online integration into LOTOS-EUROS.

5

Offline machine learning: Incorporating physical constraints

This chapter extends the methods developed in the last chapter, exploring machine learning approaches for physically consistent reduced order modeling of the tracers from the four volatility basis sets. An emphasis is placed on creating a hybrid machine learning method that finds latent patterns in a large amount of data while adhering to important physical properties, such as non-negative concentrations, conservation of mass, and phase (in this application, condensed or gaseous).

5.1. Preventing Negative Concentrations

The previous chapter showed that using the pseudoinverse of the decompression matrix W when compressing tracers for specific VBS classes could lead to negative concentrations for both superspecies and the decompressed tracers. These negative values are non-negligible in magnitude, exceeding the magnitude of the bias values for each VBS class about 1 in 20 times. This is a critical flaw in the pseudoinverse/NMF approach, which otherwise showed promise in a) recreating total organic aerosol, and b) reproducing distribution of the masses across volatility bins. The methods and perspectives developed in the last chapter serve as a basis for developing the following modifications, that have non-negativity built into them. One modification uses the same NMF algorithm with gradient descent to find a positive valued compression matrix to replace the pseudoinverse W^+. The second modification involves connecting a series of matrix multiplications together, followed by addition of a scalar and element-wise application of non-linear functions, chosen to output only non-negative values. This structure is a neural network autoencoder. Autoencoders have shown promise in many fields, including image compression, reduced order modeling in fluids, and a recent atmospheric chemistry modeling application (Kelp et al., 2020).

5.1.1. Non-negative compression and decompression

Non-negative matrix factorization results in a superspecies-to-tracer mapping W, which can decompress the latent superspecies space H to the original tracer data V. A limitation of NMF for this application is that the superspecies H are also found by gradient descent, and the standard NMF algorithm does not give an optimized compression matrix for transformation from V to H. Online methods for NMF do exist (Cao et al., 2007; Guan et al., 2012) that can find an optimal superspecies vector **h** for each sequential data point **v**. Note that online in this context doesn't mean coupling CTMs with meteorological or climate models, but rather in a machine learning context, parameter optimization performed as new training data comes in. However, with the ultimate goal of maximizing performance improvement, it would be better to develop a fully trained method offline that doesn't require online optimization. For this reason, the Moore-Penrose left pseudoinverse approach was investigated in the last chapter, with the finding that it caused a large number of significantly negative values for both the superspecies concentration vector and decompressed tracer concentrations.

Using a same approach as NMF, A positive compression matrix B can be found that best approximates

$$H \approx BV \tag{5.1}$$

Analogously to the relation of V, W, and H in equation 3.1. In this case, both V and H are given. While V is the tracer data, H can be found in a prior step using a standard NMF approach. The full approach on a high level then looks like

1. Given V, find H, W that minimizes $V - WH$
2. Given V, and using H from the previous step, find B that minimizes $H - BV$.
3. Use B to compress new observations **v** to a non-negative vector of superspecies **h**, and W to decompress **h** to the original tracer space $\mathbf{v}_{dec}$.

Both the NMF/pseudoinverse and the non-negative compression approaches limit the superspecies to a linear combination of the original tracers. This is not necessarily a weakness, as linear combinations have several advantages, including commutativity and easy interpretation. However, nonlinear methods such as a neural network autoencoder are also worth exploring in this application.

5.1.2. Neural network autoencoder

Unlike matrix factorization methods, a neural network autoencoder can represent its input layer by a nonlinear manifold in a hidden layer. In this application the first half of an autoencoder creates superspecies from tracers using nonlinear transformations. The first half uses nonlinear transformations to reconstruct the original tracers as closely as possible.

Analogously to the compression/decompression matrices, one autoencoder was designed for each VBS class. Figure 5.1 illustrates the structure for the bVOC and aVOC VBS classes: an input layer is fed to a 10-dimensional hidden layer, that is fed to a 3-dimensional hidden ”superspecies” layer. The superspecies layer is fed back to a 10-dimensional hidden layer that subsequently is fed to the output layer in tracer space. In the case of aVOC and bVOC VBS classes, this tracer space is 12-dimensional; for siSOA and POA the input/output layers are 16-dimensional and 18-dimensional, respectively. The activation function of the first hidden layer is chosen to be hyperbolic tangent, to allow for nonlinearity that is not piece-wise linear. The activation function of the superspecies layer is chosen to be a rectified linear unit (ReLU). This choice was purposeful and acts as a non-negative filter, setting all negative values resulting from linear combinations of the previous layer to zero. This constrains the output of the superspecies layer to be non-negative. Unlike the logistic function, which also restricts its output to positive real numbers between 0 and 1, the ReLU has no upper limit. The ReLU function is linear, and in fact identity, for all nonzero input, not constraining superspecies values by an upper bound. The 10-dimensional layer after the superspecies layer has a hyperbolic tangent activation function. The output from that layer is fed to the output layer, which has a ReLU activation function to constrain the decompressed output to non-negative values in tracer space.

A canonical theme in machine learning is the bias-variance tradeoff when increasing model complexity. ML models that are simple often show high bias to approximate the range of their target output. They are not too sensitive to changes in their input, showing little variance. Models that are too complex for their problem often show very high variance in their solution surface when their input has been shifted slightly, for instance when the test data is slightly different from the training data. This is often a sign of overfitting, when an overly complex model learns irrelevant features of the data at the cost of being a good approximator of the surface of the underlying function that generated the original data.

There exist several ways to prevent overfitting in neural networks. One is using dropout layers, which selectively turn off randomized nodes in the layers, reducing complexity. A dropout layer is applied after every feed-forward layer, besides the 3-node superspecies layer. Validation data is also used to assess whether the neural network is overfitting on the training data. The autoencoders are given 100 epochs to converge, with an early stopping criterion if the validation loss error does not improve after a certain amount of iterations (10 iterations were chosen). These two strategies are used to prevent the autoencoders from overfitting on the training data.

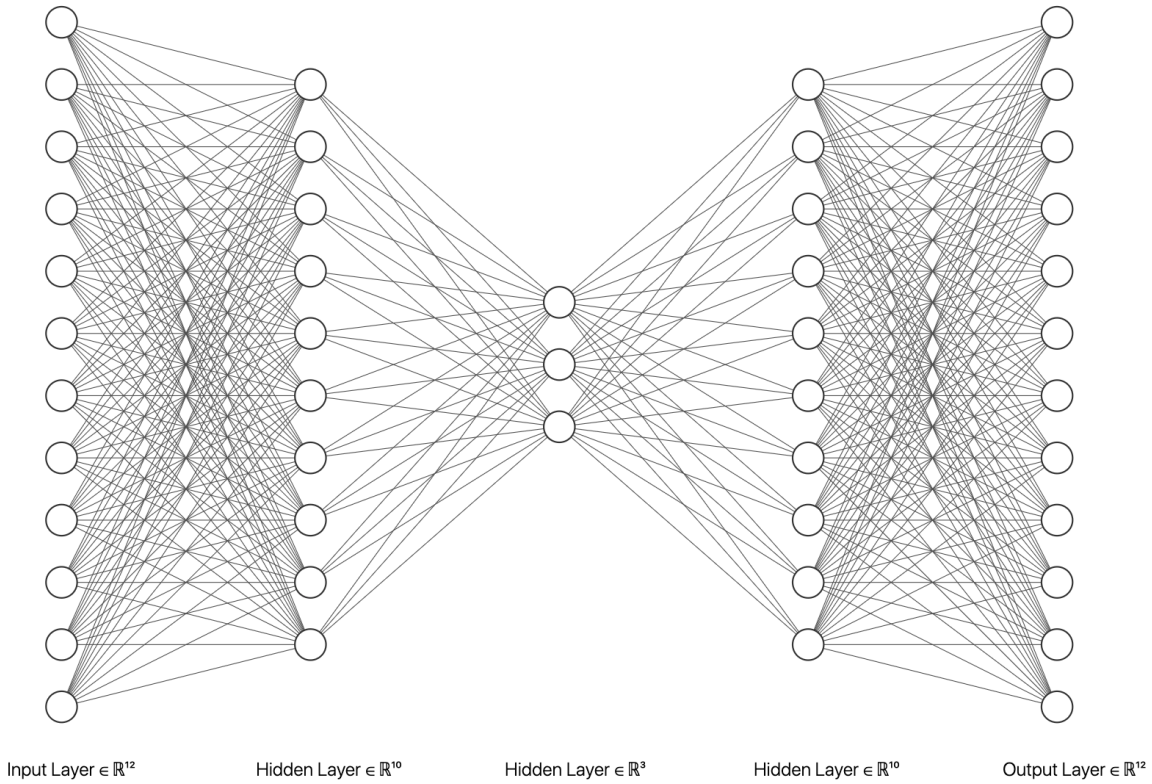


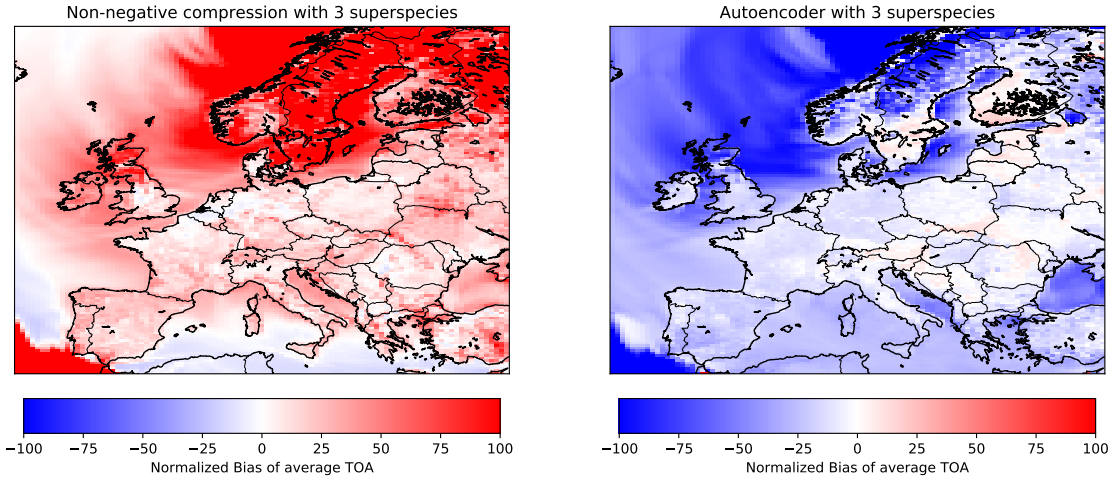
Figure 5.1: Autoencoder architecture for the anthropogenic and biogenic VBS, compressing 12 tracers to 3 superspecies and back again. The autoencoders for the other VBS classes have analogous architectures, differing only in the number of input and output nodes: 18 for POA and 16 for siSOA. Figure generated using the tool from LeNail, 2020.

No automated hyperparameter tuning was conducted for the neural network approach, for several reasons. One reason is that some hyperparameters were purposeful design choices, such as ReLU activation functions for non-negativity and the choice of 3 nodes in the hidden middle layer for comparison to the linear methods, where 3 latent features were chosen. The second is that it is not clear if a more complex model than linear non-negative compression and decompression matrices is needed: if it shows promise, hyperparameter tuning to find the most appropriate neural network model would help decide on the best autoencoder configuration.

Without trying these methods, is not clear whether a linear or nonlinear compression technique is most appropriate for the VBS tracers. A nonlinear method might be able to better handle the diverse conditions found across the LOTOS-EUROS domain, while linear method is simpler to implement and is easily interpretable. The next section compares the accuracy of the two approaches on winter test data from February 25th through 28th.

5.1.3. Comparison of the linear and nonlinear method

Figure 5.2 shows the normalized bias of average TOA from Febraury 25th through 28th, compared to the test data from LOTOS-EUROS. The autoencoder shows a negative bias for TOA in many parts of the domain, whereas the non-negative linear compression method shows positive bias. The average TOA bias for the autoencoder is -0.0346 $\mu g\, m^{-3}$, compared to 0.0657 $\mu g\, m^{-3}$ for non-negative compression. The autoencoder also showed somewhat lower RMSE values for TOA, 0.101 $\mu g\ m^{-3}$ compared to 0.133 $\mu g\ m^{-3}$. However, RMSE values for all individual VBS classes, which as an absolute metric indicates error in the shape of the volatility distribution, is higher for the autoencoder than the linear method. Moreover, the autoencoder showed higher RMSE for total organic mass (TOM), meaning its compression and decompression did not conserve mass as well as the linear method. RMSE and bias values for both approaches are summarized in Table 5.4 and 5.5 at the end of the chapter.



(a) Non-negative compression/decompression

(b) Autoencoder predictions

Figure 5.2: Normalized bias of the two non-negative compression techniques.

Since non-negative compression and decompression using the W from NMF was not outperformed by a neural network model of much higher complexity, it can be concluded that a linear method is indeed adequate for this problem. Design attempts were made to prevent overfitting of the more complex autoencoders, including dropout layers and early stopping. The neural networks all reached the early stopping criterion fairly early on, after fewer than 30 epochs. Future research, especially in a more complex problem, may want to revisit the potential of autoencoders to find a superspecies representation. More tests could also be run on the sensitivity of convergence of these models on parameter initialization. However, for the VBS tracers, there is no clear benefit to using a much more complex model. One advantage of the current linear compression/decompression method is that only two matrix-vector multiplications are required to transform a data point of tracer values to a data point of superspecies and back again. Another advantage of the linear method is that an entire latent space data point can be scaled without changing relative amounts of decompressed tracers. This means that this method can be augmented with correction factors, for example to make sure that the sum of superspecies is equal to the sum of original tracers. The following section develops several ways to conserve properties, like total concentration of tracers or total concentration of aerosol tracers.

5.2. Mass conservation

The bias values from the previous sections show that there is potential for a small amount of mass to be removed or introduced upon compression. This removed/introduced mass is nonphysical and simply an artifact of the compression strategy. Systematic removal or addition of mass may become problematic in the overall model, especially with a compression/decompression routine done once or more every time step. Buildup or removal of mass has potential to create instability for longer-term forecasting as shown in previous work (Keller & Evans, 2019; Kelp et al., 2018). The potential instability of a machine learning method that recurrently adds or subtracts mass cannot be assessed offline, and would only become apparent when doing recurrent predictions online in LOTOS-EUROS. With the above problem anticipated, this section explores several strategies that conserve properties such as total organic matter (TOM) and total organic aerosol (TOA), and aim to improve both physical interpretability and online stability.

5.2.1. Strategy 1: Conserve total organic material (TOM)

In this section, total organic material (TOM) is used to only refer to the VBS-specific tracers which are compressed, and doesn't include other gas-phase organic tracers, even those that are precursors to SOA. With this definition in mind, total organic material (TOM) for VBS is simply the sum of all tracers for each VBS. This quantity can be preserved with some rescaling of the superspecies vector **h**, when projecting the tracer space onto the latent space via W^+, and rescaling of the tracers upon decompression. After creating **h** using W^+, **h** can be scaled by a scaling factor for compression, s_{com}, given by the ratio of the total mass of the tracers and the total mass of the compressed superspecies

$$s_{com} = \frac{\sum_{i=1}^{m} v_i}{\sum_{j=1}^{r} h_j} \tag{5.2}$$

where m is the dimensionality of the tracer space and r the dimensinoality of the superspecies space. Scaling of **h** by s_{com} ensures that the sum of concentrations in the latent space is equal to the sum of concentrations in the tracer space. A similar strategy can be in decompression to the tracer space, with the decompressed set of tracers $\mathbf{v}_{dec}$. The scaling factor for $\mathbf{v}_{dec}$, let's call it s_{dec}, can be calculated by

$$s_{dec} = \frac{\sum_{j=1}^{r} h_j}{\sum_{i=1}^{m} v_{dec,i}} \tag{5.3}$$

The use of these two scaling factors adds minimal computations, and allows communication of physically relevant information without adding additional tracers.

5.2.2. Strategy 2: Conserve total organic aerosol (TOA)

In assessing LOTOS-EUROS, often measurements of surface TOA are taken as a ground truth. If TOA is the most important metric, perhaps its conservation should be prioritized. Adding another tracer to keep track of TOA would limit the effectiveness of compression, because it would add another variable. However, the approach from strategy 1 can be augmented to scale **h** by the sum of all the aerosol tracers. This adds information without adding another variable. The scaling factor for compression, s_{com} is then

$$s_{com} = \frac{\sum_{i \in T} v_i}{\sum_{j=1}^{r} h_j} \tag{5.4}$$

$$s_{dec} = \frac{\sum_{j=1}^{r} h_j}{\sum_{i \in T} v_{dec,i}} \tag{5.5}$$

where T is the set of tracer indices corresponding to aerosol tracers. If s_{dec} is applied to the entire VBS, then the relative amount of aerosol to gas is maintained. Though TOA will be conserved in this approach, it may not conserve total mass of the VBS tracers, and has potential to add or remove material in every compression and decompression step.

5.2.3. Strategy 3: Composition matrices

This idea was inspired by the methods used to generate Figure 4.8 and the corresponding discussion of physical interpretability. The strategy scales each column of W by its sum. The physical intuition here is that W becomes a matrix of composition. For a given column of W corresponding to a given superspecies, each element represents the percentage of a corresponding tracer found in the superspecies.

Interpreting the columns of this scaled W as a sufficiently representative set of superspecies, a compression matrix B can be created from normalizing the columns of W^T analogously to the column normalization of W, so that 100% of each tracer is distributed across the superspecies – no more, and no less.

5.2.4. Comparison of mass-conserving strategies

Strategy 1 is not error free, but shows near-zero bias for every VBS in Table 5.2. This is because each set of tracers and superspecies conserve mass. Despite not introducing mass to the system, this approach still overestimates TOA, which is the only standout in bias in the Strategy 1 column. Oppositely, strategy 2 only performs well for TOA, showing zero bias. Though TOA is conserved, mass is also removed from the system as shown by the negative bias for TOM. Strategy 3 performs in a similar manner to strategy 1, though shows higher RMSE for all VBS. It does conserve overall mass of each volatility basis set to machine precision.

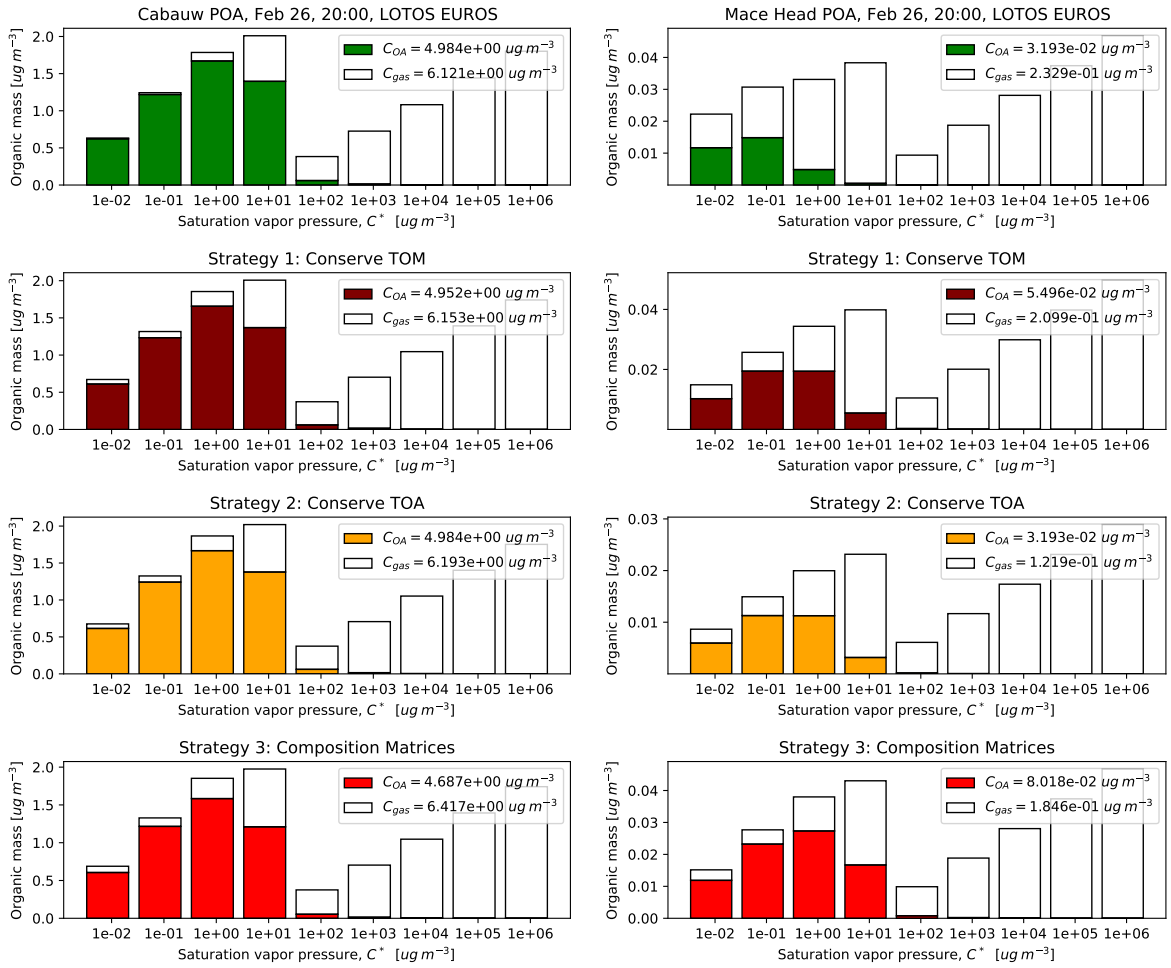
Table 5.1: RMSE of every VBS, TOA and TOM for several mass conservation strategies. All values reported in $\mu g\ m^{-3}$.

	Non-negative compression	Strategy 1	Strategy 2	Strategy 3
aVOC VBS	4.4×10^{-4}	8.2×10^{-4}	0.0056	0.0017
bVOC VBS	0.0026	0.0043	0.0427	0.0106
POA	0.0109	0.0228	0.0340	0.0348
siSOA	0.0050	0.0086	0.0108	0.0132
TOA	0.0173	0.123	4.5×10^{-9}	0.220
TOM	0.0547	1.6×10^{-7}	0.537	3.9×10^{-16}

Table 5.2: Bias of every VBS, TOA and TOM for several mass conservation strategies. All values reported in $\mu g\ m^{-3}$.

	Non-negative compression	Strategy 1	Strategy 2	Strategy 3
aVOC VBS	2.6×10^{-5}	5.7×10^{-20}	-0.0020	-3.5×10^{-19}
bVOC VBS	-1.6×10^{-4}	-1.2×10^{-18}	-0.0131	-6.0×10^{-16}
POA	4.2×10^{-4}	-1.9×10^{-11}	-0.0075	1.7×10^{-17}
siSOA	-9.9×10^{-5}	-1.6×10^{-19}	0.0023	-1.2×10^{-18}
TOA	0.0015	0.038	-8.2×10^{-12}	0.117
TOM	-0.0076	-3.4×10^{-10}	-0.352	-3.0×10^{-18}

Figure 5.3 shows the reconstruction error of the POA VBS when using the 3 different mass conserving strategies. The sum of legend entries for Strategies 1 and 3 are equal to the sum of legend entries from the LOTOS-EUROS distributions, conserving TOM. Strategy 2, on the other hand, reproduces the C_{OA} (specific total aerosol of the VBS class) to machine precision upon compression, and thus TOA when summing C_{OA} of all the VBS classes. Strategy 1 does not conserve TOA. However, this is not a necessary compromise. The following section designs a method that conserves both TOM and TOA, further refining the mass conservation techniques by adding additional information about the phase of the superspecies.



(a) Cabauw POA VBS

(b) Mace Head POA VBS

Figure 5.3: Comparison of the volatility basis set distribution for POA near (a) Cabauw and (b) Mace Head on February 26th, which have very different conditions and magnitudes of POA. The top row in green shows the distributions as modeled by LOTOS-EUROS. The second row in maroon shows the distributions after the non-negative compression/decompression outlined in Strategy 1, using 3 superspecies to represent aerosol and gas tracers. The third row in orange shows the decompressed distribution after the TOA conservation, strategy 2. The bottom row in red shows the compressed/decompressed distribution using the composition matrices constructed using strategy 3.

5.3. Phase-specific Compression

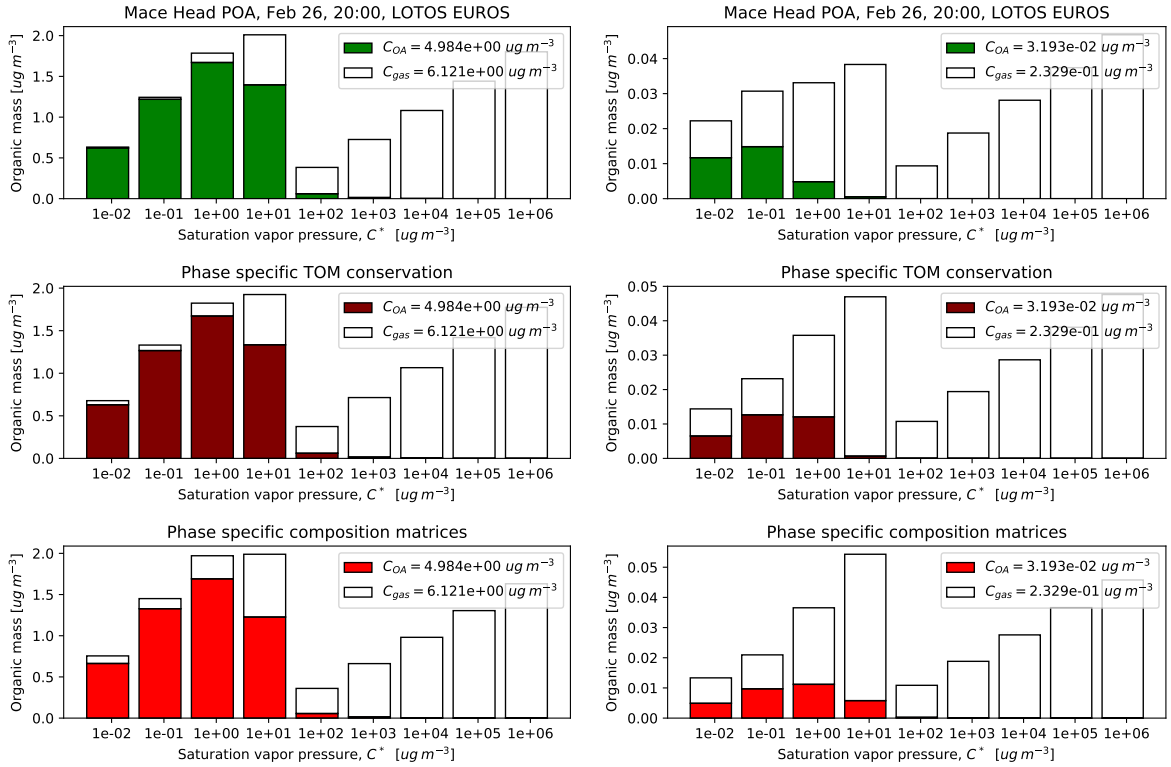
An assumption made so far is that a superspecies can be a mixture of tracers from both gas and aerosol phases. This is a result of superspecies being created by linear (NMF compression and decompression matrices) or non-linear (NN autoencoder) combinations of tracers from both the gas and aerosol phases. However, some processes in LOTOS-EUROS, like dry deposition, handle gases and aerosols differently. This limits the usefulness of mixed-phase superspecies, as they either need to be assumed as only being in one phase for these processes, or not be used at all.

An alternative approach is to instead repeat methods explored above to gas tracers and aerosol tracers separately. This has additional benefits. One immediate result is that either of the TOM-conserving methods explored in the previous section can be used to conserve both TOA and TOM. It is also expected that this will generalize to dilute conditions with very low TOA, by decoupling aerosol and gas relative compositions in the superspecies. The linear mixed phase superspecies explored so far have fixed relative values of aerosol to gas tracers, and their shapes form a basis for the reconstructed distribution. High gas concentrations could increase the concentration of superspecies, which would then redistribute that mass across the other tracers upon decompression. If there are regions in the domain with high siVOC content but little OA, that will be reflected in the concentrations of the pure aerosol/gas superspecies.

Adding another cross-section, phase, splits the tracers of each VBS class in half based on their phase. Using 3 superspecies per class per phase thus halves the compression factor; in the case of the aVOC and bVOC classes, this would be a factor of 2, e.g. compressing 6 bSOA tracers to 3 superspecies. This compression factor is about as beneficial as the compression factor technique proposed in section 2.7. In that case, the zero-order compression technique might be a better option, as it theorized to be non-lossy compression: it takes advantage of the partitioning equation to decompress total mass of each bin into equilibrium partitions, as long as an additional superspecies tracks TOA. Fewer than 3 superspecies per class should be used for a desirable compression factor, but as demonstrated in section 4.4, 1 superspecies per class is also extremely limited. A single superspecies is only 1 degree of freedom as dictated by its concentration. This fixes the distribution of decompressed tracers to a single shape and cannot capture any variability in volatility distribution. For this reason, 2 superspecies finds a middle ground between compression factor and flexibility. The total reduced dimensionality is then 2 superspecies for each of the 2 phases, for each of the 4 VBS classes leading to 16 superspecies for 58 tracers, a compression factor of approximately 3.6.

Figure 5.4 shows the reconstruction as in Figure 5.3, but with the 2-superspecies, phase-specific approach. The legends showing C_{OA} (total POA in this case) and C_{gas} demonstrate that mass is conserved completely within phase. This contrasts with the mixed phase approach used in Figure 5.3, which was capable of either conserving TOM (mass conserving strategies 1 and 3) or conserving TOA (strategy 2). With phase-specific superspecies, this compromise is no longer necessary, and strategy 1 can be used for each VBS class and phase to conserve TOA and TOM.

The compression/decompression matrices in phase-specific compression become smaller. For example, the bSOA decompression matrix W is 6 rows and 2 columns, compared to the bVOC VBS W in the previous section of 12 rows and 3 columns. Adding the phase constraint results in many simple models, which have disadvantages and advantages. Two disadvantages is that each model has fewer degrees of freedom, and the overall system has a lower compression factor of 3.6, compared to 4.8 when using 3 mixed-phase superspecies. However there are also advantages. One advantage is directly related to compression extent, in that the single-phase superspecies approach has more overall degrees of freedom (16) than the approach with mixed-phase superspecies (12). Another benefit is that small matrices allow for easy scanning and interpretation of the superspecies components. TOA and TOM can be conserved simultaneously, using the same scaling factor approach from strategy 1. Finally, this approach was designed to create superspecies that are compatible with phase-specific processes, like dry deposition. This opens a door for future use of superspecies by more operators in LOTOS-EUROS than just advection.



(a) Cabauw (b) Mace Head

Figure 5.4: Comparison of the volatility basis set distribution for POA near (a) Cabauw and (b) Mace Head on February 26. The top row in green shows the distributions as modeled by LOTOS-EUROS. The middle row in maroon shows the distributions after the non-negative compression/decompression outlined in Strategy 1, using 2 superspecies to represent aerosol tracers and 2 to represent gas tracers. The bottom row in red shows the distribution after compression/decompression using composition matrices.

Phase-specific superspecies are able reproduce the mass distribution over volatility bins for Cabauw and Mace Head, stations with very different conditions, as shown in Figure 5.4. An absolute metric, RMSE, can give insight in their ability to reproduce mass distribution over volatility bins across the whole domain. The following section compares metrics for the phase-specific, 2 superspecies approach using mass conserving strategy 1, with several other approaches that were developed and tested so far.

5.4. Comparison of Selected Approaches

Tables 5.3 and 5.4 provide an overview of some methods explored so far: the NMF/Pseudoinverse approach from the previous chapter, the non-negative compression and autoencoder methods compared in section 5.1.3. Finally, the phase-specific approach using mass conserving strategy 1 developed in the last section is shown.

Table 5.3: Evaluation RMSE of selected approaches. All values reported in $\mu g\ m^{-3}$.

	3 feature NMF/ pseudoinverse	3 feature non-negative compression	3 feature NN autoencoder	2 feature, single phase non-negative compression
aVOC VBS	4.4×10^{-4}	0.0010	0.0021	0.0011
bVOC VBS	0.0026	0.0078	0.0181	0.0042
POA	0.0109	0.0285	0.0306	0.0142
siSOA	0.0050	0.0086	0.0094	0.0057
TOA	0.0173	0.133	0.101	6.9×10^{-13}
TOM	0.0547	0.240	0.328	1.0×10^{-12}

Table 5.4: Evaluation bias of selected approaches. All values reported in $\mu g\ m^{-3}$.

	3 feature NMF/ pseudoinverse	3 feature non-negative compression	3 feature NN autoencoder	2 feature, single phase non-negative compression
aVOC VBS	2.6×10^{-5}	1.2×10^{-4}	-3.9×10^{-4}	2.8×10^{-20}
bVOC VBS	-1.6×10^{-4}	3.8×10^{-4}	-0.0051	-1.6×10^{-16}
POA	-4.2×10^{-4}	0.0050	-0.0075	-8.8×10^{-18}
siSOA	-9.9×10^{-5}	7.7×10^{-4}	-0.0022	1.2×10^{-19}
TOA	0.0015	0.0657	-0.0346	-1.3×10^{-15}
TOM	-0.00763	0.108	-0.237	-2.1×10^{-15}

It can be seen that phase-specific compression using strategy 1 is able to conserve TOM and TOA, as well as total mass in each individual VBS class and each phase, to machine precision upon decompression. However, it also shows higher accuracy of RMSE for all the VBS classes compared to both the (not conservative) non-negative compression and neural network autoencoder developed at the beginning of this chapter. As RMSE values are absolute metrics, they are indicative of distribution error, showing that phase-specific mass conserving compression is able to reproduce the distributions most accurately.

Studying reconstruction accuracy offline is not able to fully capture how useful these methods will be online in LOTOS-EUROS. The machine learning methods take the VBS tracers away from equilibrium – differences in total mass for each bin, as well as TOA, will ultimately cause different partitioning when the VBS module is called, changing TOA. It must be assessed if this machine learning method can remain stable and accurate when this compression is done recurrently every operator splitting timestep. Chapter 6 implements the phase-specific, mass conserving superspecies into LOTOS-EUROS.

So far, much effort has been spent in developing a machine learning method that does not require online optimization. However, another question is how robust this method is to different conditions. If it needs to be changed/updated to handle different conditions, for example in different parts of the year, or for different areas, how often should this be done? The phase-specific superspecies are tested for summer conditions to see how well they can represent very different TOA composition and spatial patterns.

6

Online Implementation

Phase-specific and VBS class-specific matrices generated by using non-negative compression were chosen to be implemented in LOTOS-EUROS using a scaling factor to conserve mass on compression and decompression (strategy 1). These were demonstrated in the previous chapter to conserve mass within each volatility class and phase (aerosol or gas). This method reproduced the mass distribution across volatility bins better than other methods, including the other mass balancing approach, strategy 3, of creating composition matrices from W. Given that non-negative, mass-conserving, phase-specific linear compression is the most appropriate approach for transforming the VBS tracers into superspecies and back again, this chapter does not continue comparing machine learning methods. Instead the focus is on investigating how the phase-specific, mass conserving method performs online in LOTOS-EUROS when used to accelerate advection calculations.

6.1. Implementing superspecies into LOTOS-EUROS

The unsupervised machine learning approach developed in previous chapters was optimized (trained) and evaluated (tested) on model output from LOTOS-EUROS. The ultimate goal of the superspecies parameterization is use online in LOTOS-EUROS to reduce model dimensionality of several operators. This is a research challenge, but also a development challenge: integration of superspecies into the LOTOS-EUROS source code requires restructuring of various elements. This section very briefly summarizes a) the scheme to use superspecies for advection and b) some other adjustments required to make the minimal ML superspecies extension of LOTOS-EUROS.

To advect superspecies, the scheme in the LOTOS-EUROS driver program was adjusted. Initialization, compression, and decompression subroutines were added to the VBS module. These subroutines are then called in the driver program:

1. The initialization subroutine to load offline-optimized compression and decompression matrices is called during operator initialization early in the LOTOS-EUROS driver program, before the time loop starts.
2. Within the time loop, the compression subroutine is called right before advection to transform VBS tracers into superspecies, overwriting the current superspecies values. The advection operator then acts on these superspecies instead of the VBS tracers.
3. Also within the time loop, directly after advection, the decompression routine is called to transform superspecies into VBS tracers, overwriting the VBS tracer values.

Many other adjustments need to be made in order to make this work, only several important ones are mentioned here. First, the set of 16 superspecies were added to the LOTOS-EUROS tracer list (superspecies tracers to be added to ultimately reduce total tracers in advection). The superspecies were assigned similar tracer groups to their VBS counterparts: vbs, and soa, poa, cg (condensable gas), and fine mode (aerosol) when applicable. They were additionally designated as a new group, superspecies. With those group definitions, the for loop over tracers in the advection operator could be adjusted to

skip all tracers belonging to the vbs group that did not also belong to the superspecies group. It should be noted that some other operators have analogous loops over all tracers (though sometimes the structure is different), like vertical diffusion and dry deposition, but no adjustments were made to them. For that reason, these processes in fact had to deal with both VBS tracers and superspecies, performing meaningless operations on superspecies that would be overwritten in the compression step, and slowing down more than if there were only VBS tracers. There is a nontrivial amount of development work required to adjust all operators and processes to disregard certain groups of tracers. Future code optimization will have to make decisions on which processes handle superspecies, and which processes handle the VBS tracers, at minimum the emissions operator and the chemistry operator (which includes the VBS module).

The superspecies extension of LOTOS-EUROS was used to perform various experiments comparing ”superspecies runs” to a control run with the VBS tracers. The rest of this chapter investigates the behavior of VBS tracers when their superspecies representation is advected. Main focuses are accuracy and limitations of using the superspecies for advection under a variety of conditions, as well as the speedup benefit resulting from reduced order modeling.

Specific questions explored are

- Is this method stable when running recurrently?
- How does advecting superspecies perform under conditions it hasn't been optimized for?
- Does this parameterization yield a speed-up in LOTOS-EUROS runs?

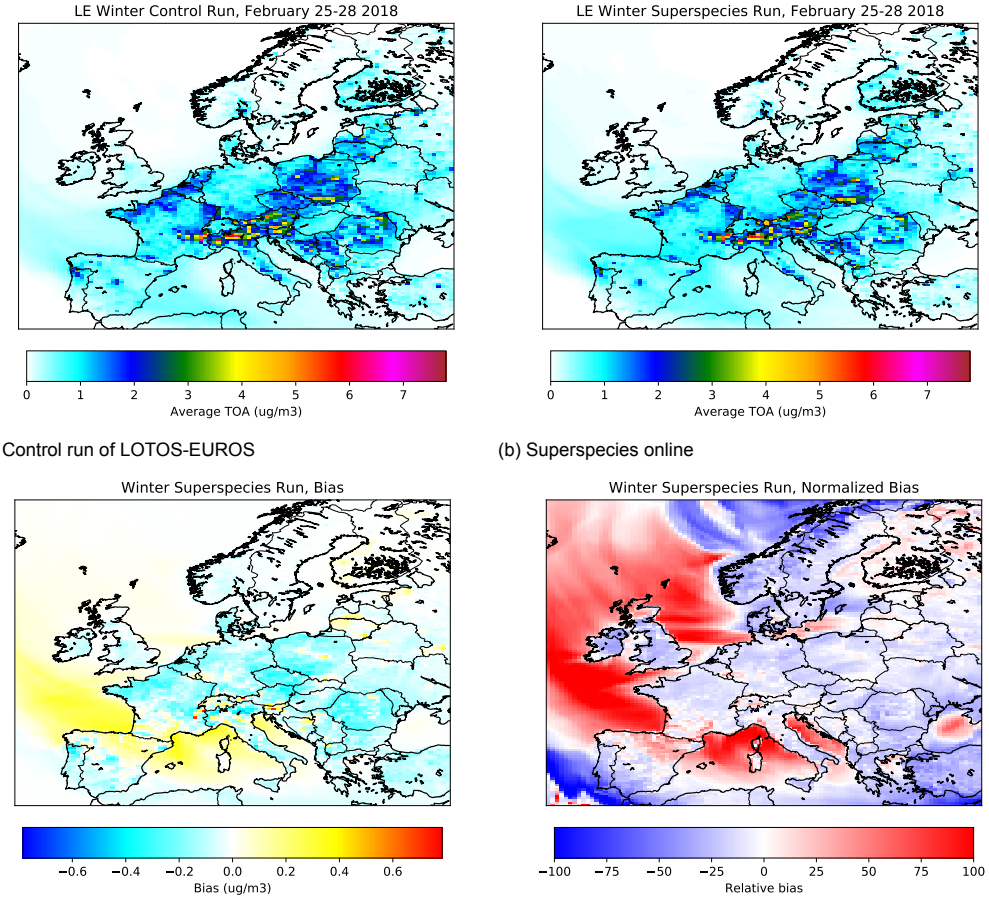
6.2. Accuracy of online implementation in winter

After offline training of the compression and decompression matrices on data from February 20th through 24th, the superspecies parameterization was implemented into LOTOS-EUROS, and used in the advection operator for a run from February 15th through 28th. In the offline training regime, the days of February 15th through 19th were used for model spin up time, data from February 20th through 24th used for training, and data from February 25th through 28th used for model evaluation. In an online setting, it can be argued that even small errors caused by advecting superspecies changes the VBS tracer concentrations so that the time period February 20th through 24th becomes new data, and should not be regarded as ”previously seen” data that the superspecies parameterization was trained on. That being said, meteorological conditions and other processes independent of the VBS and superspecies parameterization are identical to that of the offline training data. To maximize comparability, the superspecies run and control run are still evaluated on the same time period as the offline methods in chapters 4 and 5, February 25th through 28th, even though the superspecies run has the chance to accumulate error since the begin of the simulation. As in previous chapters, model output from February 15th through 19th is regarded as model spin-up time and disregarded.

The ML parameterization remains stable throughout the entire superspecies run, showing low error even after 14 days. Figure 6.1 shows average TOA of the control run and the superspecies run, from February 25 through February 28. This test time period is well into the model run, 10 days after the begin of the simulation. During this time period and over the entire domain, average bias of TOA of the superspecies run compared to the control run is small and slightly negative, -0.0095 $\mu g\, m^{-3}$. Small average bias is not in itself indicative of low error, as positive and negative bias cancellations throughout the domain and time period are possible. RMSE, an absolute metric, was larger at 0.217 $\mu g\ m^{-3}$. However, relatively low error metrics show that the recurrent use of a machine learning technique was stable, in any case not causing exponential error accumulation even after 14 days: more than twice the length of the period used for training data. Superspecies advection on higher vertical levels that they were not optimized for did not appear to cause problems in surface data, even after allowing enough time for mixing. At higher vertical levels, conditions tend to be more dilute, and different temperature and pressure regimes impact partitioning.

There are a few potential reasons for the ease of generalization to higher vertical levels. One is that concentrations are highest near the surface anyway. Another is the large amount of dilute conditions in the surface training data on the MACC domain. The use of scaling factors to conserve total aerosol and gas upon compression and decompression might be important for this successful generalization,

preventing material from being introduced unrealistically in these dilute conditions. More experiments involving training on other layers, as well as not enforcing mass conservation, would be necessary before drawing conclusions about whether only surface data is indeed sufficient training data, or if more vertical levels are needed in the training set. Preliminary experiments using a different vertical level scheme suggest that training data should match the vertical level scheme chosen, but future experiments will have to investigate this in more detail before conclusions can be made. The following section explores how the superspecies, optimized on training data from winter conditions, perform in a summer run.



(a) Control run of LOTOS-EUROS

(b) Superspecies online

(c) Bias of TOA predictions

(d) Relative bias of TOA predictions

Figure 6.1: Average TOA for February 25th through 28th using superspecies matrices optimized offline on winter conditions from February 20th through 24th.

6.3. Generalizing to summer conditions

Advecting superspecies was shown to run stably in winter, in a time period different from the time period the machine learning approach was trained on. The test time period from February 25th through 28th was directly after the training test period February 20 through 24th, and had relatively similar conditions to what the superspecies transformation matrices were optimized for.

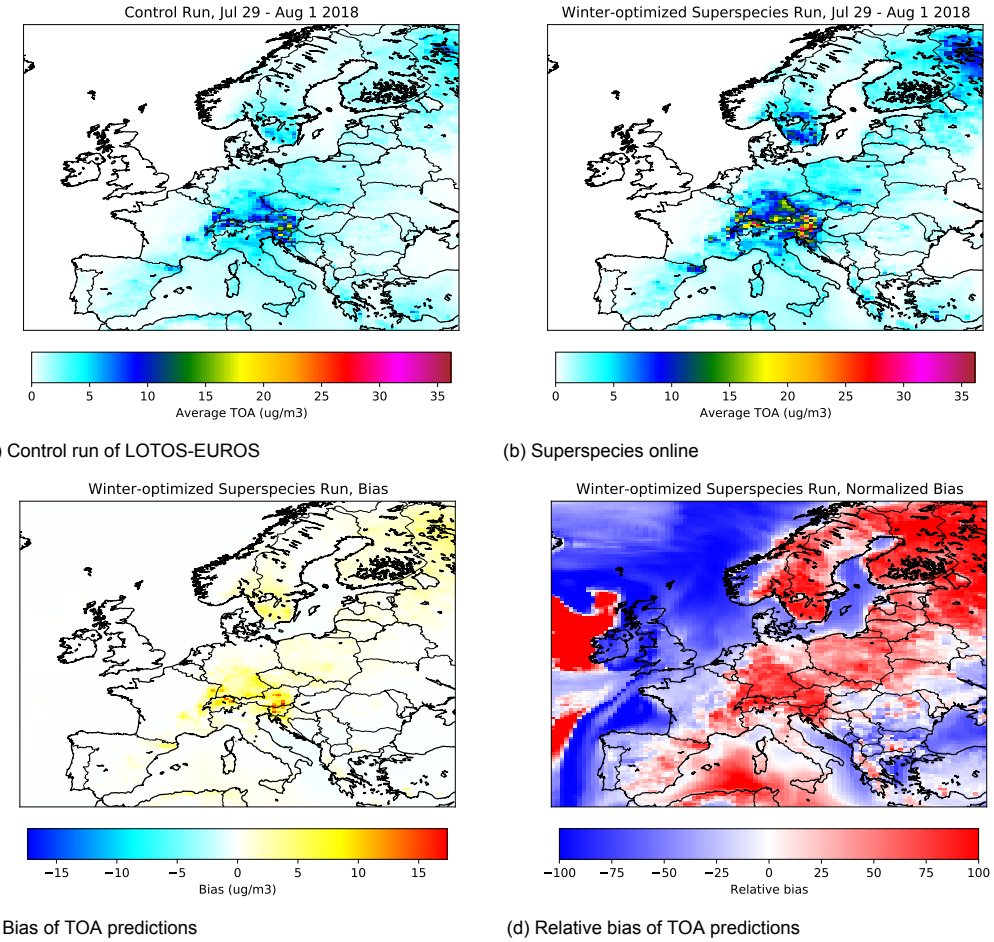
A run in summer from July 20th through August 1st was chosen to assess the robustness of this method to very different conditions. Summer conditions differ from winter conditions for several reasons. One, biogenic precursor gases make a larger contribution to formation of secondary organic aerosol in the summer, partially due to emissions from forests. Two, average temperatures in Europe are higher, affecting the partitioning of the VBS by changing the saturation pressure values C^* over volatility bins. Three, due to increased heat flux there is more vertical mixing, increasing exchange of material between the surface and higher altitudes, which tend to be more dilute than the surface concentrations (which the superspecies ML approach was trained on). The different conditions lead to different modeled compositions of total organic aerosol (TOA). Table 6.1 shows the modeled average composition of TOA for February 25th through 28th with TOA composition for July 29th through August 1st.

OA Type	February	July
aSOA	0.8%	9.5%
bSOA	4.5%	34.8%
POA	61.2%	12.5%
siSOA	33.5%	43.2%

Table 6.1: Average TOA composition for the LOTOS-EUROS runs for February 25-28 and July 29-August 1.

Though siSOA is on average the largest component of TOA in the run from July 29th through August 1st this is not the full picture, and underscores the importance of bSOA under some conditions. The maximum concentration of surface siSOA over the entire domain over the entire period from July 29th through August 1st was 15.0 $\mu g\, m^{-3}$, and 99th percentile 1.3 $\mu g\, m^{-3}$ compared to the maximum bSOA concentration of 100.3 $\mu g\, m^{-3}$ and 99th percentile 9.4 $\mu g\, m^{-3}$. This indicates that although siSOA may dominate in background conditions and when TOA is low, bSOA is the dominant component of TOA in other conditions.

Figure 6.2 shows average surface TOA, as predicted by the control run (Figure 6.2a), the run with superspecies advected (Figure 6.2b), and the bias and relative bias of the superspecies run with regards to the control (Figures 6.2c and 6.2d). When compared to Figure 6.1, it is clear that the spatial pattern of TOA is very different in the summer than the winter. Primary organic emissions (POA) are often the largest contributor to winter TOA, and for this time period TOA is most concentrated the Po Valley, Czechia, and Poland. The winter superspecies run is able to recreate these large regions of high TOA, as well as other smaller but distinct pockets of TOA, such as Madrid (the most populous city in Spain) and northwest Portugal, a region with a lot of industrial activity. In contrast, summer TOA is concentrated around southern Germany, Switzerland and Austria. Many places in this region are forested, and contribute to TOA via biogenic precursors of SOA. Biogenic SOA is a significant contributor to TOA in summer conditions, while bSOA is only a minor contributor to the winter TOA. The superspecies run shown in Figure 6.2b is able to capture these spatial patterns, but with a strong bias. For this reason, other areas with high biogenic emissions emerge in Figure 6.2b, such as Slovenia, southern Sweden, and Finland Proper, as well as northwestern Russia, which is more than 50% forested. These highly forested areas are modeled in LOTOS-EUROS via land use maps, with corresponding emissions the type of forest (Manders-Groot et al., 2021).



(a) Control run of LOTOS-EUROS

(b) Superspecies online

(c) Bias of TOA predictions

(d) Relative bias of TOA predictions

Figure 6.2: Average TOA on the MACC domain for July 29th through August 1st using superspecies matrices that were optimized for winter conditions.

Despite reproducing the general spatial pattern of TOA in the domain, the superspecies, optimized for winter conditions and evaluated on summer conditions, show a large positive bias over the areas with high average TOA, especially heavily forested regions. RMSE for TOA over the whole domain and time period is 2.12 $\mu g\ m^{-3}$, with an average bias of 0.321 $\mu g\ m^{-3}$. RMSE of the tracers from the biogenic VBS for all times and gridcells is 0.66 $\mu g\, m^{-3}$, an order of magnitude higher than tracers from the other VBS classes: the class of tracers with the next highest RMSE value is the siSOA VBS class, at 0.062 $\mu g\, m^{-3}$. The average bSOA bias (neglecting gaseous tracers) is 0.068 $\mu g\, m^{-3}$, 3 orders of magnitude smaller than the maximum bSOA bias of 82.9 $\mu g\ m^{-3}$. Overestimation of bSOA in the superspecies run under some conditions is likely due to errors in decompression, artificially shifting mass to lower volatility bins. However, the large large positive bias in parts of the domain (Figures 6.2b and 6.2c) indicate that this tendency to overestimate bSOA only happens in certain conditions: namely, forested regions. The following section analyzes one gridcell in a forested region, and finds additional temporal patterns where bSOA is significantly overestimated, leading to overestimation of TOA.

6.4. Case study: Summer night in Schönbuch

The superspecies optimized on winter conditions show high bias in the night-time summer conditions over forests. This section is a case study to illustrate the limitations of the winter-optimized superspecies in such conditions. The LOTOS-EUROS gridcell containing Schönbuch Natural Reserve in southwest Germany, which is 156 square kilometers and 85% forested, was chosen. Figure 6.3 shows temporal variation of TOA from July 29th through August 1st. This overestimation is systematic, with the night of July 30th to July 31st a particularly high TOA event showing the highest bias.

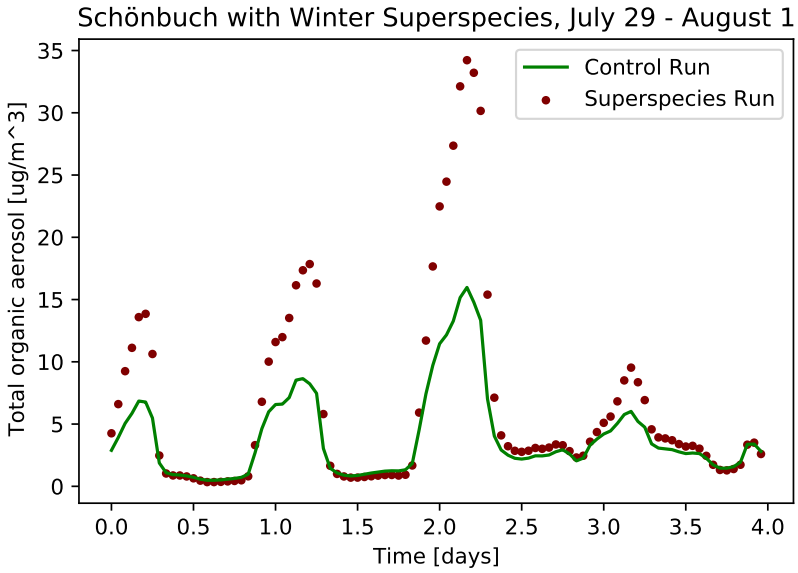
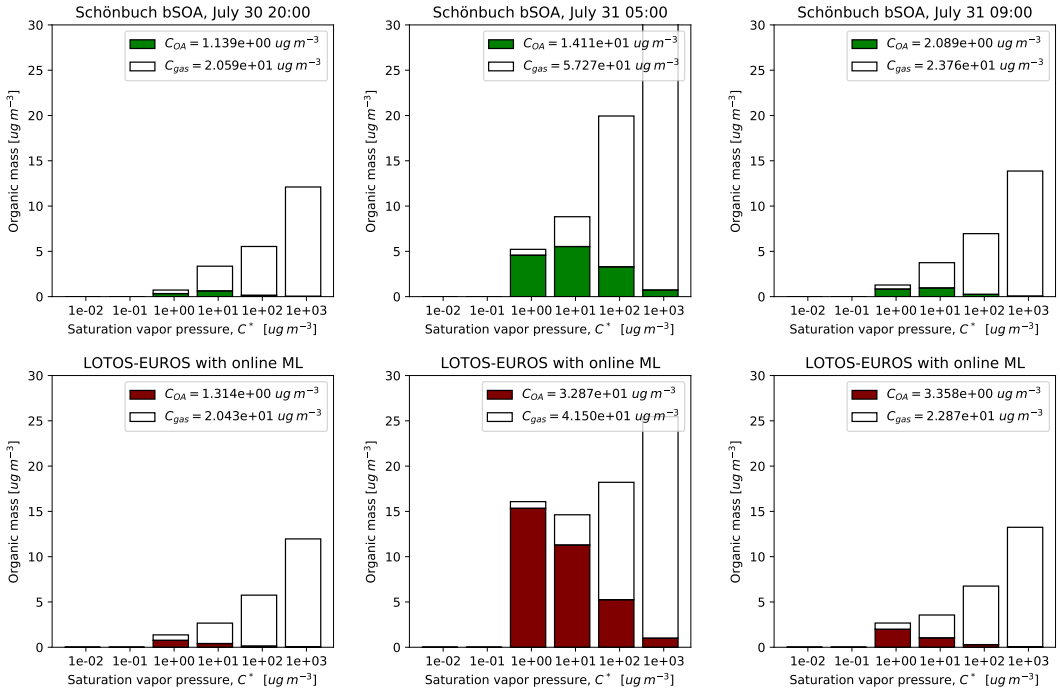


Figure 6.3: Temporal variation of TOA over Schönbuch. The maroon points of TOA as predicted with the online run using winter-optimized superspecies are compared to the green line of TOA as modeled by the LE control run.

Given that TOA is overestimated in the summer run when using winter-optimized superspecies, and the main contributor to TOA in the summertime is bSOA, it is good to compare the biogenic VBS distributions of the control run and the superspecies run. An episode of overnight high bSOA between July 30th and July 31st is shown in Figure 6.4. This night showed the highest overestimation of bSOA, C_{OA}, over the entire summer test period. Note that C_{OA} is used to refer to total concentration over all bins *within* the VBS class, in this case bSOA, whereas TOA is used for the total organic aerosol concentration over all VBS classes.

Figure 6.4a shows only a slight bias in C_{OA} at 20:00 on July 30th. In this nightly episode, C_{OA} begins increasing at 21:00 in both the superspecies run and the control run, but at a faster rate in the superspecies run, culminating at 05:00 July 31st in Figure 6.4b and overestimating total bSOA with a factor between 2 and 2.5 times that of the control. The superspecies run bSOA concentration is 32.9 $\mu g\, m^{-3}$, 99% of total TOA for the superspecies at that gridcell and time. The control run concentration of bSOA is 14.1 $\mu g\, m^{-3}$, about 95% of TOA. By 09:00, Figure 6.4c shows that both runs return to total bSOA of less than 3.5 $\mu g\, m^{-3}$. This night episode of high bSOA contains the largest overpredictions for that particular gridcell in the whole time period. However, it is illustrative of a failure mode of the winter-optimized superspecies to capture the volatility distribution and total concentration (C_{OA} in the figures) of bSOA, and ultimately TOA due to the importance of bSOA contributions in this example. The spatial patterns and temporal patterns of the superspecies run compared to the control run show that the superspecies are limited in their ability to model conditions over forested areas on summer nights.



(a) 20:00 (b) 05:00 (c) 09:00

Figure 6.4: Case study of a high night-time bSOA event over a forest in southern Germany. The superspecies run captured the temporal variability in terms of magnitudes, but overpredicted total bSOA and mass in lower volatility bins.

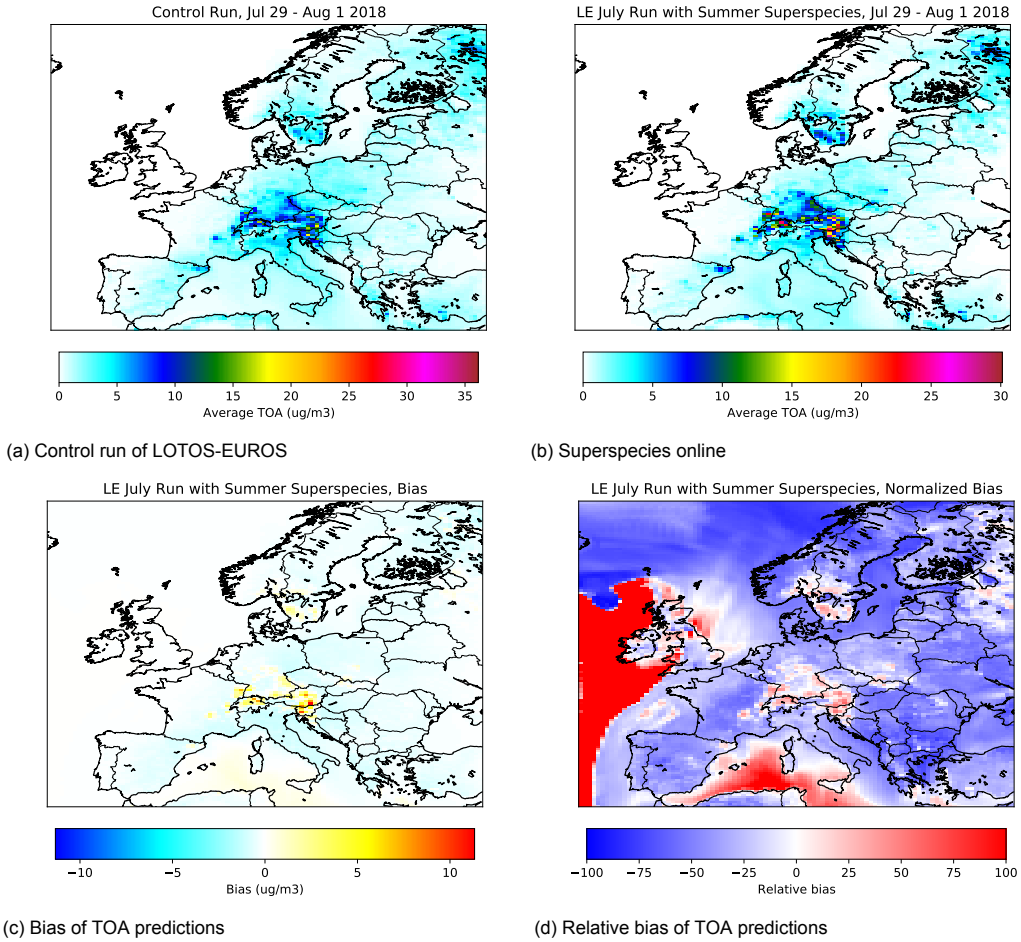
6.5. Superspecies optimized on summer conditions

6.5.1. Summer-optimized superspecies in summer runs

Given that winter-optimized superspecies showed limitations in capturing high bSOA events over forested areas, a followup question is: can superspecies optimized on summer conditions and implemented online reproduce high bSOA conditions from the LOTOS-EUROS control run with higher accuracy?

The mass balancing strategies 1 and 3 from the previous chapter (sections 5.2.1 and 5.2.3) were repeated to get compression and decompression matrices, using data from July 23rd through 28th, 2018. The reconstruction of the VBS sets using these matrices was assessed using test data from July 29th through August 1st. As both strategies conserve mass within class and phase upon compression, average bias for each reconstructed volatility basis set was near zero. However, online error arises from inaccurate reconstruction of the mass distribution across volatility bins. An absolute metric like RMSE can capture error in the reconstructed distribution. Using strategy 1, RMSE for the biogenic VBS tracers after decompression was 0.099 $\mu g\ m^{-3}$, an order of magnitude higher than that of the other VBS classes. Using strategy 3, RMSE for the biogenic VBS tracers was 0.307 $\mu g\ m^{-3}$, also an order of magnitude higher than RMSE of other VBS classes using the same approach.

The matrices from strategy 1 were then used in online runs to generate superspecies to be advected. The same winter and summer periods from sections 6.2 and 6.3 were respectively used, with the only difference being that the compression and decompression matrices have been optimized on summer conditions.



(a) Control run of LOTOS-EUROS

(b) Superspecies online

(c) Bias of TOA predictions

(d) Relative bias of TOA predictions

Figure 6.5: Average TOA on the MACC domain for July 29th through August 1st using superspecies matrices that were optimized for summer conditions.

Figure 6.5 shows that using summer-optimized superspecies can reduce overall bias across the do-

main. Interestingly, bSOA (neglecting the gaseous tracers) demonstrates a slightly negative average bias over the domain and time period of -0.023 $\mu g\ m^{-3}$. However, peaks of high TOA bias up to a maximum of around 10 $\mu g\ m^{-3}$ can be seen in some parts of the domain. These pockets of TOA overestimation occur in the same conditions noted in the previous section, over highly forested areas. For example, some of the highest positive bias is shown in Slovenia, where over half of the land area is forested. Quantitatively, the superspecies optimized on summer conditions result in more accurate runs. The RMSE over the whole domain of time-averaged average TOA was 0.98 $\mu g\ m^{-3}$ when using summer-optimized superspecies, reduced by over a factor of 2 when compared to the 2.12 $\mu g\, m^{-3}$ when using winter-optimized superspecies. RMSE of the tracers from the biogenic VBS for all times and gridcells is also reduced by a factor of 2, at 0.32 $\mu g\, m^{-3}$ compared to 0.66 $\mu g\, m^{-3}$. However, as in the run using winter-optimized superspecies, the biogenic VBS tracers show significantly higher error than the tracers of the other VBS classes, with the siSOA VBS class having the next highest RMSE value at 0.050 $\mu g\, m^{-3}$. The temporal pattern of nightly TOA overestimation is diminished as can be seen when comparing Figure 6.6 to 6.3, though still shows a strong overestimation on the night of August 30th to 31st. However, this is the only incidence of significant overestimation from February 20th-28th in that gridcell, on a night of particularly high TOA, whereas the winter-optimized superspecies run systematically showed nightly overestimations. The results of this case study indicate that summer-optimized superspecies are better suited in handling night-time conditions over forests in the the summer. More generally, seasonal-specific superspecies might result in higher accuracy.

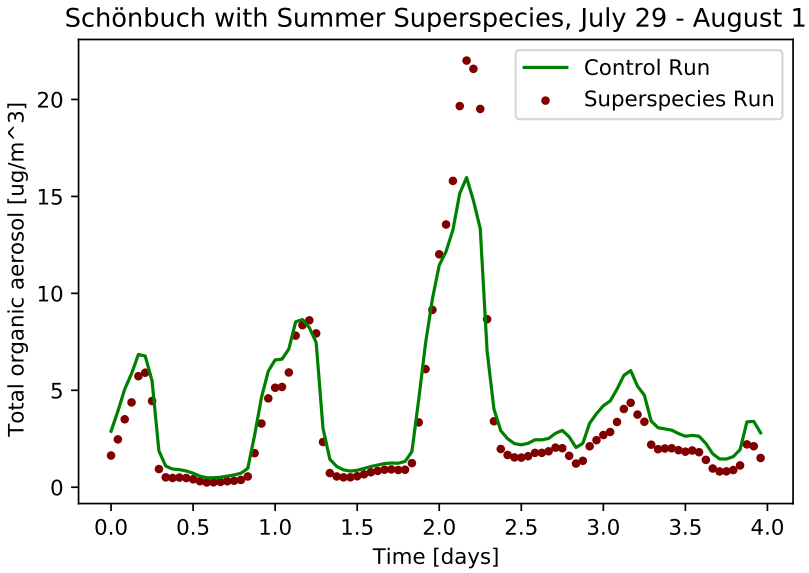
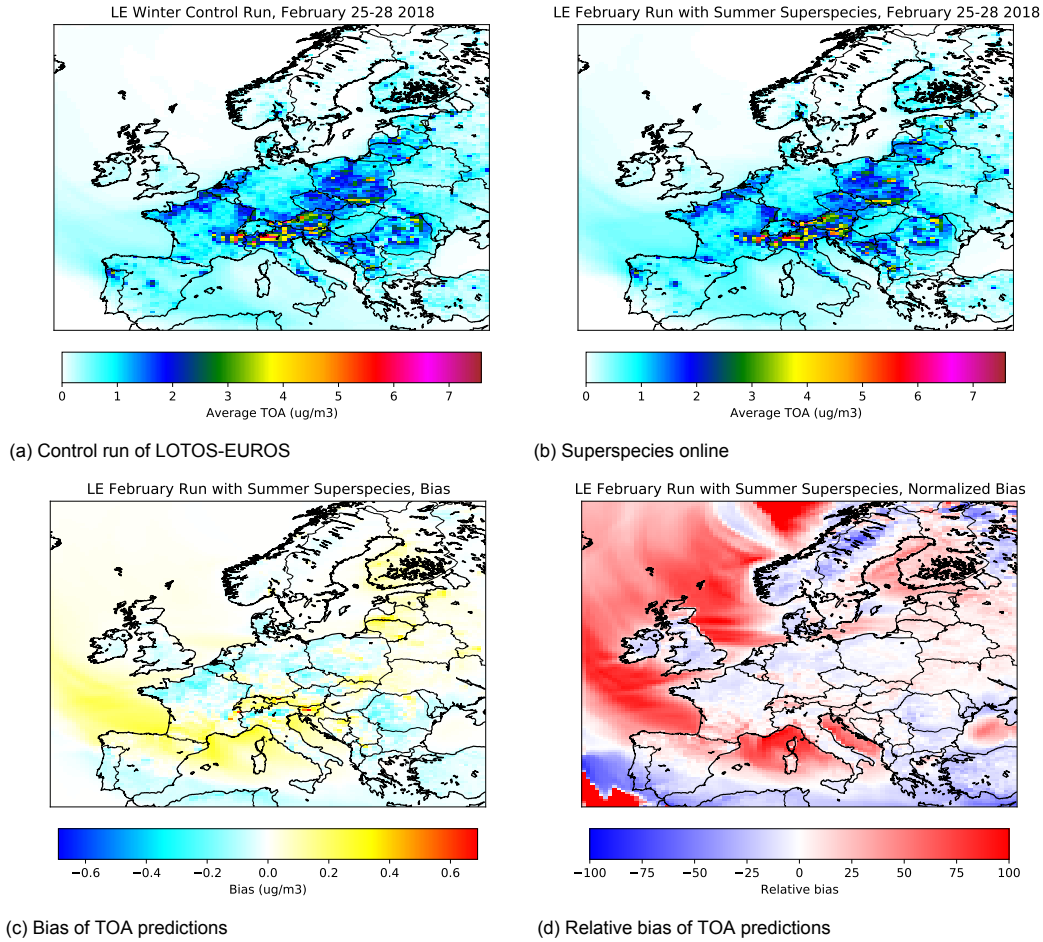


Figure 6.6: Temporal variation of TOA over Schönbuch. The maroon points of TOA as predicted with the online run using summer-optimized superspecies are compared to the green line of TOA as modeled by the LE control run.

6.5.2. Summer-optimized superspecies in winter runs

The spatial patterns of time-averaged TOA in Figure 6.5 show that summer-optimized superspecies still result in high average TOA bias in forested areas in the summer. However, summer-optimized superspecies can be used stably and with little error in winter runs. In the test period of February 25th through 28th, TOA from the superspecies run (Figure 6.7b) shows the same spatial pattern as that of the control run (Figure 6.7a). Quantitatively, there is little average bias (Figure 6.7c, with a maximum of around 0.6 $\mu g\ m^{-3}$ average TOA bias. There is only high relative bias as in areas with extremely low (near zero) concentrations, as can be seen when comparing the relative bias in Figure 6.7d with the the control run or the superspecies run. RMSE of TOA over the whole domain and time period is 0.15 $\mu g\ m^{-3}$. The siSOA VBS tracers show the highest RMSE of 0.040 $\mu g\ m^{-3}$. The tracers with the next highest error are the POA VBS tracers, with an RMSE of 0.024 $\mu g\ m^{-3}$. The biogenic VBS tracers show a much smaller error, but are a significantly smaller contributor to TOA during wintertime conditions, as seen in Table 6.1.



(a) Control run of LOTOS-EUROS

(b) Superspecies online

(c) Bias of TOA predictions

(d) Relative bias of TOA predictions

Figure 6.7: Average TOA for February 25th through 28th using superspecies matrices optimized on summer conditions.

Winter-optimized superspecies perform well in winter conditions, but generalize poorly to summer conditions, whereas summer-optimized superspecies generalize well to winter conditions while performing significantly better than winter-optimized super species in summer. This asymmetric inability to generalize to other seasonal conditions encourages use of summer-optimized superspecies if a choice has to be made about a general year long superspecies approach, rather than season-specific superspecies. The underlying cause of this asymmetry could be that winter conditions are governed by a simpler model: POA is the main contributor to winter TOA. Primary organic emissions are of a constant shape, as introduced in section 2.5.2, and ageing of the primary VBS gas phase tracers only removes material from the VBS rather than shifting it to bins within the VBS. This contrasts with the biogenic VBS, whose yields are dependent on concentration of NOx, linearly interpolated between two extremes of high and low NOx. Moreover, POA emissions in LOTOS-EUROS have a prescribed shape in emissions year round, though amount may vary. Primary VBS superspecies optimized on summer conditions are therefore optimized to capture the same emissions effects as in winter.

6.5.3. Biogenic SOA in the summer

Online runs, performing the compression-advection-decompression step recurrently multiple times per hour, for several weeks, are stable and fairly accurate for modeling winter conditions. This approach for winter runs is robust to the conditions the superspecies (specifically the compression and decompression matrices for each VBS class) were trained on. Accuracy does not change much if the compression/decompression matrices are optimized for summer conditions and ran for the winter. However, this method is less accurate for summer conditions. The error for summer conditions is systematic, with the highest overestimation occuring over forested areas at night. Superspecies optimized for summer conditions can reduce the spatial pattern of bias over forests, but cannot remove it. For a case study over a southern Germany forest, optimizing superspecies on summer data removed the temporal pattern of overnight bias when implemented online, for the entire period except one night with a high TOA event.

In the summer runs, tracers from the biogenic VBS showed the highest error, which is in line with the spatial patterns of highest error in TOA occuring over forests. Given the relative ease of superspecies optimized on either winter or summer conditions to model winter conditions accurately, it can be concluded that the bSOA is the hardest OA type to model using the superspecies for the test periods. However, it can not yet be concluded exactly why. One underlying reason could be that the bVOC VBS has more complex chemistry than siSOA and POA, resulting in more variation in the mass distribution across volatility bins than 4 superspecies (2 aerosol and 2 gas superspecies) can capture. Gaseous tracers in the POA VBS react only with hydroxyl radical to form siSOA VBS tracers, and gaseous tracers in the siSOA VBS react to form material in lower volatility siSOA bins. The biogenic gaseous precursors have more complex chemistry, including dependence on NO for the branching ratio that determines the yields of each precursor onto the top 4 volatility bins. However, there are effectively only 4 bins (8 tracers) in the biogenic VBS, whereas the siSOA and POA basis sets have higher dimensionality: 8 and 9 bins (16 and 18 tracers) respectively. The anthropogenic VBS also has similar chemistry of its gaseous precursors. Though the aSOA contribution to TOA is usually small, its normalized bias can be compared with that of bSOA. The normalized bias of aSOA is in fact larger in magnitude than that of bSOA: -0.46 to -0.21. This might indicate that if aSOA was a more significant contributor to TOA in the time periods studied, it might contribute detrimentally to accuracy. However, the negative bias values for bSOA contrast with the spatial and temporal patterns of high bSOA overestimation under some conditions. This indicates that the bSOA overestimation using superspecies is specific to certain conditions, and that domain-wide and time period error metrics fail to capture this phenomenon. Future experiments will have further explore bSOA overestimation with the superspecies under these conditions in order to understand the underlying causes. This might provide general insight into the limitations of the superspecies approach, and when a different approach (like the zero-order compression proposed in section 2.7.2) might be more appropriate.

Despite showing high bias, the superspecies approach did not accumulate error exponentially, staying in the realm of realistic concentrations. The machine learning step remains stable even after being run independently from the control run for 14 days (July 19th through August 1st), about 288 hours. Given that the compression/advection/decompression step was done around 6 times per hour, this is after more than 2000 sequential compression/decompression steps (including the matrix multiplications for aerosol and gas vectors as one step) for that grid cell alone. However, grid cells influence each other via transport between grid cells, via vertical diffusion and advection. Over all grid cells for 12 days, dividing up the longitude by 100 gridpoints, and latitude by 140, with 5 vertical layers, the compression/decompression step is done over 140 million times on the biogenic VBS alone, without leading to runaway error.

6.6. Speedup on the MACC domain

The advection operator has an outer for loop over all tracers that are transported. Advecting superspecies that represent combinations of tracers reduces the overall amount of variables in the outer for loop. With the superspecies approach, SOA modeling requires 16 superspecies (two gas and two aerosol superspecies for each of the four VBS classes) rather than the 58 VBS tracers that would be advected.

Timing output files are contained in Appendix A, and summarized in the following sections. Wall times for various processes and subprocesses are reported in seconds for consistency with the .prf files in the appendix, though hours are also given to aid in interpretation. Relative timing factors are also given in the following discussion.

The winter simulation on the MACC domain used in the previous analysis ran for 14 days, from midnight on February 15th through February 28th 2018, ending on March 1st, 2018. The first 5 days were used as model spin-up, the next 5 for training the ML approach, and the last 4 for evaluating its accuracy.

The total time taken was 40140 seconds, of which 99.7% was spent in the time loop: 40031 seconds, just over 11 hours. Further analysis of the time loop gives insight into which processes take significant amounts of wall time. For this reason, Figure 6.8 includes the time loop duration, and subsequent discussion focuses on time loop duration. Without the VBS tracers, an otherwise identical simulation ran for 22985 seconds, about 6.4 hours. Advection also takes a larger proportion of the timeloop when using the VBS tracers, 30.2% compared to 20.2% without using VBS tracers. This corresponds to 12073 seconds to advect all tracers compared to 4639 seconds to advect only the non-VBS tracers, taking a factor of 2.6 more time in calculations. This is larger than the slowdown factor for chemistry, 5766 seconds to 11256 seconds, a slowdown of slightly less than 2.

Advecting superspecies that represent combinations of tracers yields a significant speedup. Total time for the advection operator when advecting superspecies rather than VBS tracers was 6790 seconds, 1.8 times faster than the 12073 seconds to advect all tracers. 6790 seconds is slower than the run without any VBS tracers by a factor of approximately 1.46, which can be attributed to the addition of 16 superspecies that need to be advected.

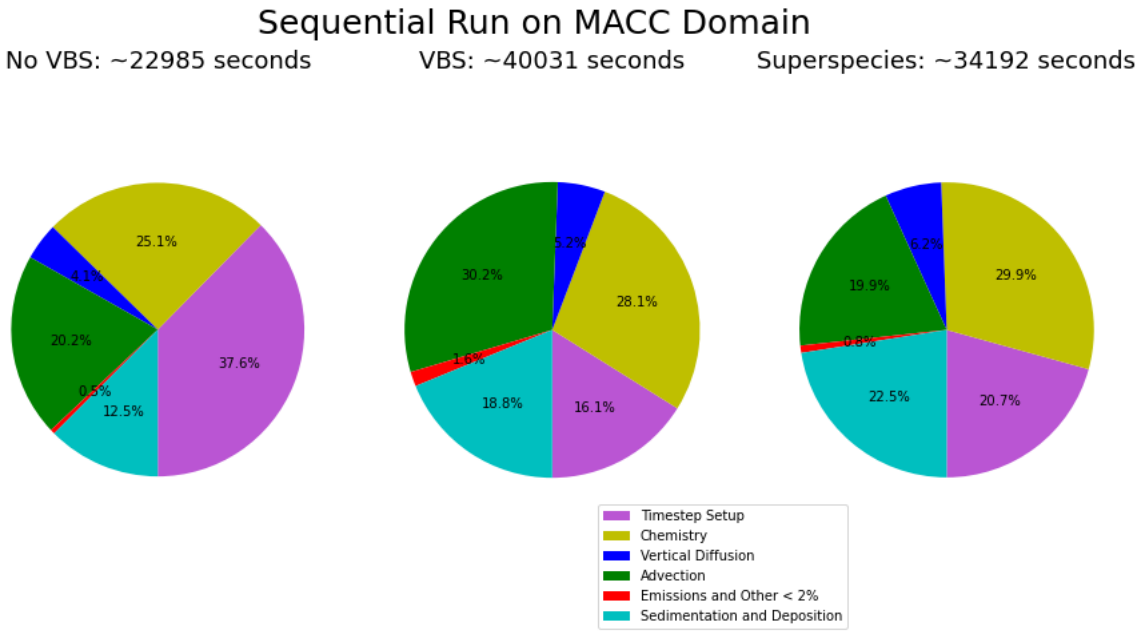


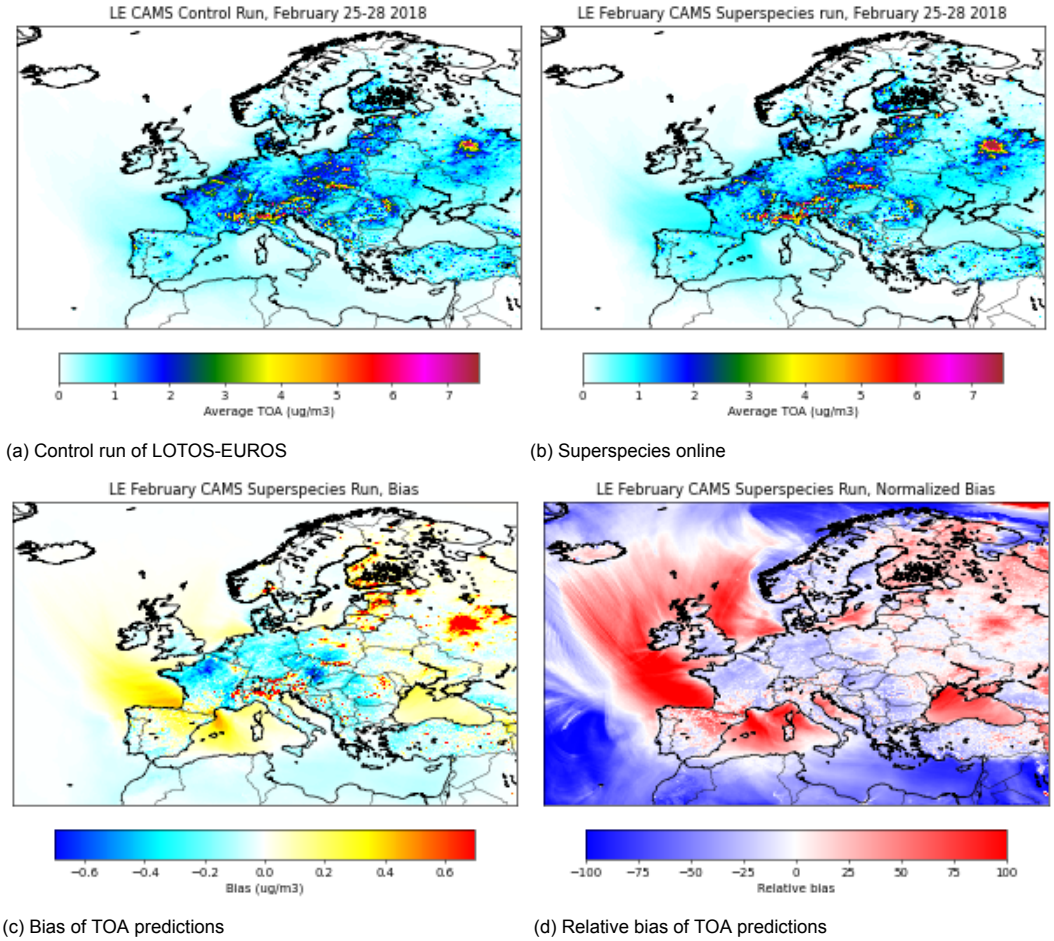
Figure 6.8: Benchmarking of various processes in the time loop of a LOTOS-EUROS run from February 15 through 28 on the MACC grid. The domain is not decomposed, this run was performed fully sequentially on one computing node.

6.7. Towards operational forecasting: CAMS Domain

The domain used in CAMS operational forecasts has a much higher resolution than the domain used by MACC: 0.1 degrees for 420 by 700 gridcells compared to the 0.50 by 0.25 degrees used in the MACC domain on previous experiments. The change of resolution and domain increases the number of gridcells by a factor of 20. One result of this, beyond much more data, is that the operator splitting timestep Δt needs to decrease in order to satisfy the Courant-Friedrichs-Lewy criterion in equation (2.14), as the gridcell distance is smaller. Advection is therefore done more times per hour, as well as the compression of tracers into superspecies and decompression of superspecies back into tracers.

Due to the increased requirement in computing power, no sequential run is performed using the CAMS operational forecasting domain. The following CAMS runs for both the control and superspecies runs are performed using domain decomposition over 24 computing nodes with each node computing a subdomain of 175 by 70 gridcells. The compression and decompression matrices optimized on the MACC February run were used in the run on the CAMS domain. This tests not only timing differences in the operational configuration of the CAMS domain, but also how the superspecies optimized on a coarse grid can do on a finer resolution. Moreover, the CAMS domain is over a wider area than the MACC grid, extending past Moscow, Russia. This experiment tests the limitations of using the superspecies on areas not included in the training data.

6.7.1. Accuracy on the CAMS domain



(a) Control run of LOTOS-EUROS

(b) Superspecies online

(c) Bias of TOA predictions

(d) Relative bias of TOA predictions

Figure 6.9: Time averaged TOA for the period of February 25th through 28th on the CAMS domain used in operational forecasting, from control and superspecies runs, as well as bias and relative bias.

Figure 6.9a shows the time-averaged TOA values for the CAMS run with VBS tracers advected, and Figure 6.9b the time-averaged TOA values for the CAMS run with the superspecies advected. The

superspecies run has a positive bias for TOA of 0.019 $\mu g\ m^{-3}$, with visible overestimation in the area near Moscow, which is not in the MACC grid used to optimize the compression/decompression matrices. Though this indicates that the superspecies approach might perform better on areas it has been trained on, further experiments would have to verify this, including applying the approach to a completely different area than the one it was trained on. The colorbar limits of Figures 6.9a, 6.9b, and 6.9c were adjusted for visual comparison with Figure 6.1. For this reason, colors at the upper or lower limits should be interpreted as greater or equal to the limit. Though the maximum of time-averaged TOA from both the superspecies run and the control run was 28.2 $\mu g\ m^{-3}$, 99.85% of the gridcells had a time-averaged TOA under 7.6 $\mu g\ m^{-3}$, which was chosen as the upper limit of the colorbar. This means that only 0.15% of the gridcells in Figures 6.9a and 6.9b exceed the limit shown in the colorbar. Neglecting the highest 0.15% of average TOA, the spatial patterns of the that the CAMS superspecies run become apparent and look very similar to the spatial patterns to the CAMS control run. Both exhibit spatial patterns similar to the simulations performed on the MACC grid for the same time period, in Figure 6.1. Analogously, the maximum absolute error of time-averaged TOA between the superspecies run and the control run was 8.9 $\mu g\ m^{-3}$, but 99.2% of all gridcells had an absolute error of less than 0.70 $\mu g\ m^{-3}$. Less than 1% of the gridcells in Figure 6.9c exceed the colorbar limit. Biases on the CAMS domain superspecies run are larger in magnitude than the those on MACC domain in Figure 6.1c, but only a few gridcells exceed the maximum error of time-averaged TOA on the MACC grid.

The largest bias for TOA over all cells and the entire test time period (not time averaged) was 89 $\mu g\ m^{-3}$, corresponding to a gridcell in northwestern Spain, near Ponferrada. This gridcell also showed the highest time-averaged TOA concentration of 32.0 $\mu g\ m^{-3}$ for the superspecies run, compared to 19.4 $\mu g\ m^{-3}$ for the control run. This difference corresponds to the highest overestimation for time-averaged TOA of any gridcell of 12.6 $\mu g\ m^{-3}$, which dictates the colorbar range of Figure 6.9c. The overestimation peak occurred simultaneously with a high TOA event on February 25th at 19:00 . At the highest bias of 89 $\mu g\ m^{-3}$, TOA concentration as modeled by the superspecies run is 206.4 $\mu g\ m^{-3}$ and the control run predicts a TOA concentration of 117.4 $\mu g\ m^{-3}$. TOA during this event is composed almost wholly of primary material: the superspecies run models a POA concentration of 205.9 $\mu g\ m^{-3}$ (99.78% of TOA concentration) while the control run POA concentration is 117.1 $\mu g\ m^{-3}$ (99.75 %). Figure 6.10 shows the timeseries behavior of TOA for both the runs during the high event and subsequent days. Rather than error compounding and leading to divergence from the control run, the superspecies run restabilized for the rest of the simulation. This indicates that in an online context, other processes in LOTOS-EUROS can correct temporary overpredictions from the superspecies as the simulation progresses.

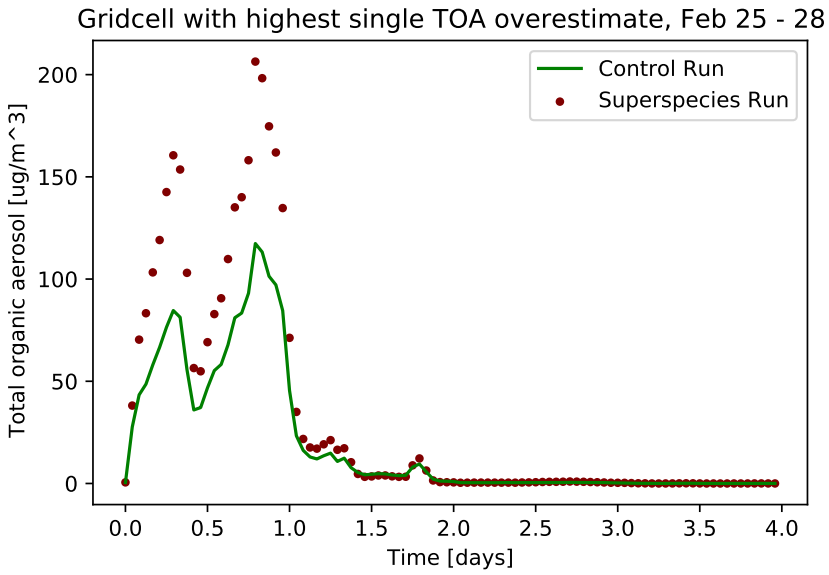
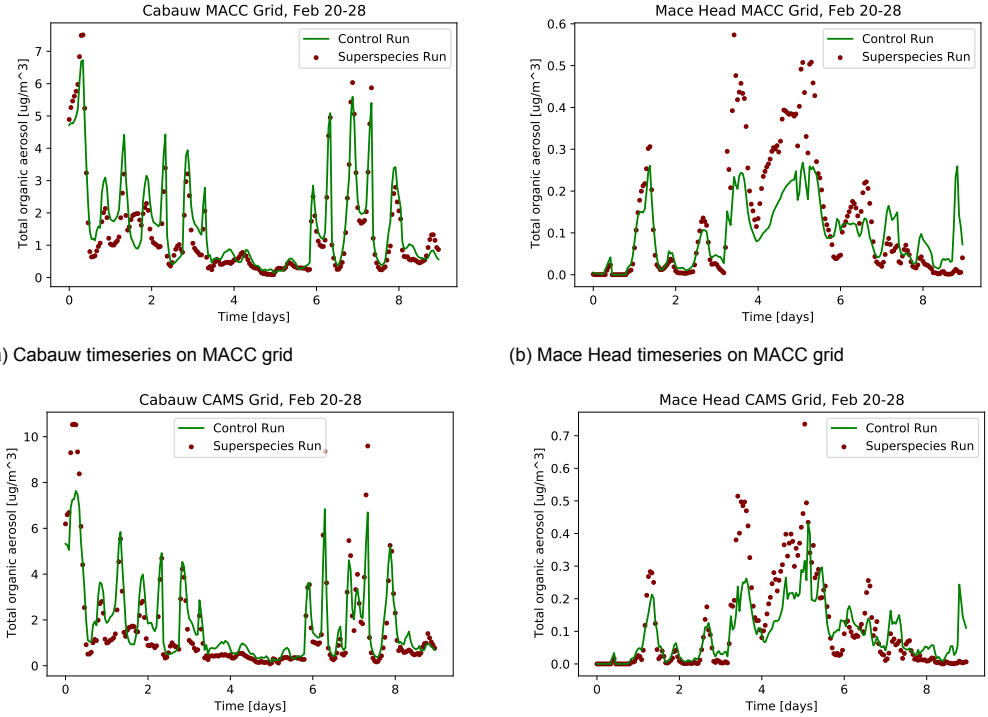


Figure 6.10: Temporal variation of TOA at the gridcell with the highest TOA overestimate in the entire CAMS domain from February 25th through 28th.

Failure modes of the superspecies approach will have to be further assessed to gain a better understanding of how such a high overestimation of TOA can occur when advecting superspecies instead of VBS tracers. However, this extreme overestimation is a rare occurrence, with 99% of bias values under 0.94 $\mu g\, m^{-3}$ and 95% of bias values under 0.33 $\mu g\, m^{-3}$. This provides an argument that in most cases, the superspecies approach is stable when integrated online in LOTOS-EUROS, even when running on finer grid resolutions than it was optimized for.

Figure 6.11 returns to the two atmospheric research stations, Cabauw and Mace Head, used in offline evaluations in chapter 4. Here, the timeseries of TOA predictions when advecting winter-optimized superspecies online is compared to the control run, for both the MACC domain and the CAMS domain.



(a) Cabauw timeseries on MACC grid

(b) Mace Head timeseries on MACC grid

(c) Cabauw timeseries on CAMS grid

(d) Mace Head timeseries on CAMS grid

Figure 6.11: Timeseries of TOA at Cabauw and Mace Head, on the MACC grid and CAMS grid.

Mace Head, a remote station, shows TOA concentrations an order of magnitude lower than Cabauw. Though both stations show error, especially Mace Head about halfway through the run, this is not too surprising, especially given the dilute conditions at Mace Head. What is actually more striking is when TOA from the superspecies run tracks TOA of the control run, at some points following the curve for hours, even after advecting the superspecies for over a week. This happens in the middle of day 6 and 7 at Cabauw for both the MACC and CAMS domains. For Mace Head, periods of high accuracy are day 2 and parts of day 5, visibly tracking the control TOA despite the small scale of the concentration axis. Another point of interest for Mace Head is that after a period of high overestimation around February 24th and 25th in both the MACC and CAMS runs, TOA restabilizes, even tracking the control TOA again. This is qualitatively similar to the restabilization shown in Figure 6.10, though several orders of magnitude more dilute. The existence of both cases indicates the existence of a phenomenon or multiple phenomena that act to dampen error caused by the superspecies parameterization of advection, preventing runaway error and divergence of results when using the superspecies parameterization. This also suggests that superspecies formed by mass-conserving combinations of tracers might behave physically similar in processes. The ability of the LOTOS-EUROS superspecies configuration to correct its error over subsequent timesteps, in different conditions and over many magnitudes, suggests that this is a robust approach.

6.7.2. Speedup on the CAMS domain

Figure 6.12 shows the breakdown of different processes within the time loop for different experiments on the CAMS domain. With the VBS, overall wall time nearly doubles, and advection time more than doubles from 34959 seconds to 74762 seconds. With superspecies advected instead of VBS tracers, wall time for the advection operator is 49473 seconds. The run advecting superspecies takes about 40% more time than with all the VBS tracers switched off, but is able to model secondary organic aerosol, decompressing to the VBS tracer space after every advection time step, and taking about 66% of the time that it would with all the VBS tracers advected.

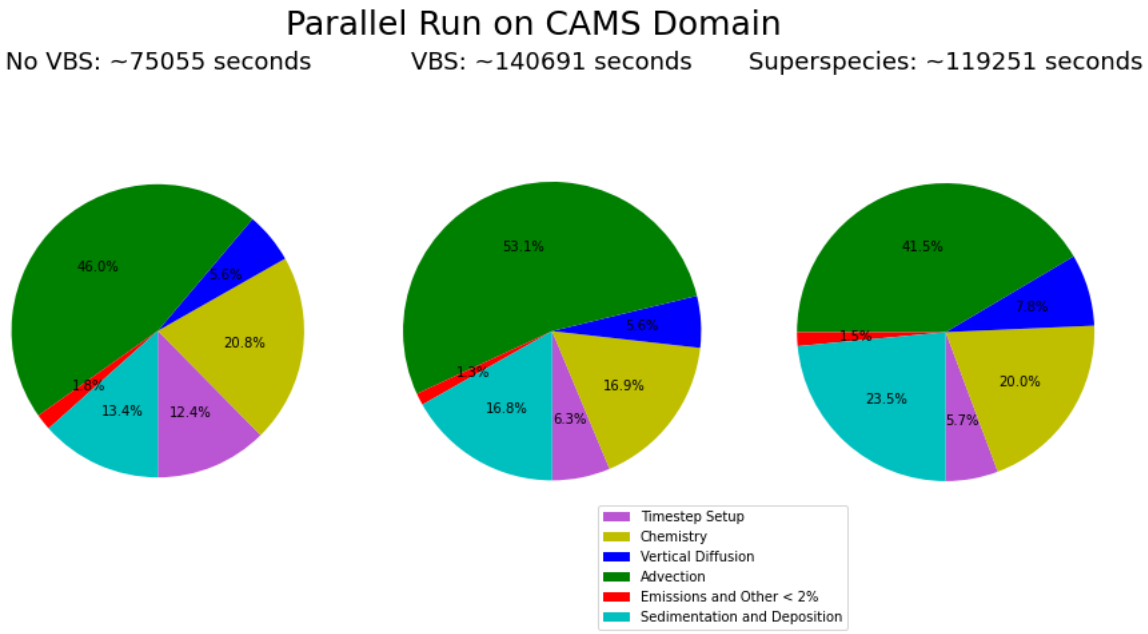


Figure 6.12: Timing of various processes in the time loop of a LOTOS-EUROS run from February 15 through 28 on the CAMS grid. The parallelization technique involved domain decomposition into 24 subdomains, parallelized on 24 computing nodes.

The full potential speedup benefit of superspecies is not yet realized in the current implementation. The time loop of the LOTOS-EUROS superspecies run takes approximately 85% of the time that the control run does. This is because other processes actually have an increase in time, handling both tracers and superspecies, though the superspecies values are subsequently overwritten in the compression step right before advection. The decision of which processes should use superspecies, and subsequent implementation, is a future development goal outlined in the conclusion. The purpose of this experiment was to demonstrate a significant reduction in the runtime of the advection operator when handling superspecies instead of VBS tracers: this reduction varies between slightly over half to two thirds.

This chapter assessed the accuracy and speedup benefit of replacing VBS tracers with machine learning superspecies in the advection operator. An implementation of LOTOS-EUROS that includes these superspecies was developed and shortly summarized in section 6.1. This is the first instance of a machine learning parameterization running online in LOTOS-EUROS. The strategy of advecting superspecies was found to run stably for a model simulation of 2 weeks under various seasonal conditions, for example in sections 6.3 and 6.5. Superspecies optimized on winter conditions showed systematic spatial and temporal bias during summer simulations in section 6.3. However, optimizing superspecies on summer conditions in section 6.5 significantly reduced this systematic error. Machine learning superspecies optimized on the coarse MACC domain show potential to generalize to the finer CAMS domain used in operational forecasting without experiencing runaway error. Timing experiments in sections 6.6 and 6.7.2 found that the advection operator using superspecies took 56% to 66% of the time that it took when using VBS tracers.

7

Conclusions

This thesis developed a machine-learning parameterization to reduce the number of tracers to be advected in LOTOS-EUROS. This technique creates a set of 16 superspecies with linear combinations of 58 tracers from the 4 volatility basis set (VBS) classes. After advecting the superspecies instead of the original tracers, the superspecies can be decompressed to determine new concentrations of the VBS tracers.

A phase-specific superspecies approach was chosen to represent the tracers from each VBS class with 2 aerosol and 2 gas superspecies. Use of scaling factors allowed for mass conservation when compressing and decompressing tracers. This superspecies approach was trained on 4 days of data and then implemented into LOTOS-EUROS for online calculations. Results using this superspecies implementation are stable on model simulations of 2 weeks. The advection operator using superspecies takes 56% to 66% of the time it would take using original VBS tracers. This approach shows potential to generalize to other conditions and finer grid-sizes than it was trained on.

Section 7.1 returns to the research questions formulated in section 1.7 using the results of this research to make some conclusions about the limitations and potential of advecting a superspecies representation of tracers. Future research questions in section 7.2 suggest possible directions for application and exploration of the approach introduced in this thesis. Section 7.2.1 discusses next steps for integration of the superspecies in LOTOS-EUROS operational forcasting, and limitations of this approach found in some conditions. Section 7.2.2 broadens the scope to other applications beyond the LOTOS-EUROS community, relating the machine learning superspecies approach to other current research and discussing its potential to be generalized beyond the development goal it was designed for.

7.1. Return to the research questions

The four core research questions introduced in section 1.7 informed the experiments and results throughout this thesis. This section summarizes key results related to each.

Research question 1: What parts of LOTOS-EUROS are slowed down by inclusion of the volatility basis set? Can they be accelerated using machine learning parameterizations? Section 1.5 shows that the volatility basis set approach which can double the runtime of LOTOS-EUROS simulations under certain conditions.The inclusion of the 4 volatility basis sets requires 58 additional tracers on top of the 64 default tracers. A more granular benchmarking in section 2.6 found that other processes slowed down, most notably advection, sometimes by a factor of 2. One underlying cause is not VBS-specific computations, such as calculating partitioning, but modeling advection of the 58 VBS tracers. The advection operator is one of the more computationally intensive processes in LOTOS-EUROS. Its computation time depends linearly on number of tracers due to an outer for loop over all tracers. VBS inclusion approximately doubles the tracers and therefore should be expected to double the computation time of the advection operator.

Though the tracer slowdown was identified relatively early on in the project, this discovery itself should

not be taken for granted. In fact, the goal at the outset of this project was a machine learning surrogate model for the VBS gas-phase reactions and partitioning. The benchmarking in 2.6 showed that this would have saved minimal time and not addressed the problem, as calculating the partitioning is not a significant source of slowdown. Identifying the additional tracers as the source of slowdown in 2.6 motivated reduced order modeling. Though a zero-order compression technique advecting total bin concentration and TOA was proposed, this would require 30 superspecies, a compression factor of around 2. The desire for a larger compression factor led to the proposition of unsupervised machine learning techniques, which have shown success in atmospheric modeling applications, to find lower dimensional, latent patterns in the large amount of model output.

Chapters 3, 4, and 5 developed a machine learning approach to combine VBS tracers into superspecies, which can replace the tracers in bulk processes like advection. These superspecies can then be decompressed into original tracers for VBS-specific calculations. Chapter 6 evaluated the speedup of this method when integrated into LOTOS-EUROS. On the MACC domain that the method was trained on, advection took approximately 56% of the time that it would without the superspecies parameterization. In a run using the CAMS domain used in operational forecasting, advection with superspecies took approximately 66% of the time that it would with all 58 VBS tracers. These speed improvements are in agreement with the expected linear dependence of advection time on number of superspecies. The set of 16 superspecies is slightly less than a quarter the size of the 58 VBS tracers. The VBS tracers are about half of the advected tracers (some of the 64 tracers, like radicals, aren't advected), so the theoretical estimate of speedup is 63% when using 16 superspecies.

Research question 2: Can a machine learning approach maintain desired accuracy of total organic aerosol, as well as mass distributions over volatility bins, sources, spatial and temporal patterns?

A linear unsupervised machine learning approach using non-negative matrix factorization (NMF) was introduced in chapter 3. Through a series of experiments and comparisons in chapters 4 and 5, the NMF approach was refined to create optimized compression and decompression matrices that did not require additional optimization for new data points. Bias, a standard metric, was used to give insight into potential under or overestimation resulting from compression and decompression. Root mean squared error (RMSE) was used as a second key metric. RMSE is an absolute metric of error between decompressed tracers and target tracers and gives insight into reconstruction error of the mass distribution over volatility bins, even if overall bias is low. Chapters 4 demonstrated that this approach can reconstruct the mass distribution across the volatility bins to an acceptable extent, outperforming more complex methods like a nonlinear neural network autoencoder in chapter 5. This compression and decompression technique is able to reproduce spatial patterns of average TOA over the LOTOS-EUROS domain on test data it was not optimized for (trained on). Temporal variation of TOA at two atmospheric research stations, Cabauw and Mace Head, was studied to find that the compression technique is able to track the variation in conditions over time. Section 4.5 tested the lossy reconstruction error of this method as a function of compression extent. Three superspecies per class were chosen as a reasonable trade-off between accuracy and compression extent. With three superspecies, the approach in chapter 4, showed an average bias for TOA across the whole evaluation data was 0.0015 $\mu g\ m^{-3}$, and RMSE 0.0173 $\mu g\ m^{-3}$. Little marginal improvement of these accuracy metrics was shown with subsequent increase of the latent dimension.

Research question 3: In what ways can classic modeling approaches and machine learning be hybridized to improve physical interpretability and/or respect important physical properties?

The unsupervised learning strategy finds a lower dimensional latent space to represent the original tracer space. The latent space representation found using non-negative matrix factorization (NMF) can be physically interpreted as a set of superspecies, formed by linear combinations of tracers. Superspecies can represent different OA regimes, like freshly emitted aerosol or aged aerosol, which have different mass distributions across volatility classes. Subsequent combinations of these superspecies can create distributions that are linear combinations of different regimes. The non-negativity of both the compression and decompression matrices ensures non-negative concentrations of both superspecies and tracers. An advantage of a linear method is the invariance of the distribution shape in tracer space to scaling of the superspecies space. Mass conserving strategy 1 in section 5.2.1 uses scaling factors after compression and decompression to conserve total organic concentration to ma-

chine precision. Superspecies are specific to VBS class, so material is conserved within the class. Additionally, total mass is conserved for each phase (gas or aerosol), as well as the two cross-sections together: VBS class and phase. Here, two superspecies per class and phase were chosen, rather than 3 or 1, to maintain a desired compression factor while allowing for different distribution shapes upon decompression, as determined by the concentrations of the 2 superspecies. With this approach, both total organic aerosol and total organic matter are conserved to machine precision on compression and decompression. On an evaluation dataset, RMSE for TOA was $6.9 \times 10^{-13}\ \mu g\ m^{-3}$, and TOM $1.0 \times 10^{-12}\ \mu g\ m^{-3}$.

Strategy 3 in section 5.2.3 extended the physical interpretation of superspecies by constraining the columns of the compression and decompression matrices to sum to 1. Each column of the superspecies-to-tracers matrix W can be interpreted as the composition of a superspecies, with each element corresponding to the fractional contribution of a tracer in forming that superspecies. With that in mind, analogous column normalization of W^T could be interpreted as how each tracer is distributed over the superspecies. Another result of these composition matrices is that total mass is conserved to machine precision upon both compression and decompression.

Research question 4: How does a machine learning parameterization perform when implemented online in LOTOS-EUROS?
Results in chapter 6 show that the machine learning species optimized to reconstruct model output offline can run stably online, without accumulating error. The compression/advection/decompression step was run recurrently online in LOTOS-EUROS for a simulated 2 weeks from February 15th through 28th on the MACC domain. Error did not compound or propagate, and was relatively low (time averaged bias below 10%) even after simulating 14 days of online superspecies, completely independent from the control run. Superspecies were advected at all other levels, despite the ML method only being optimized on surface data.

However, superspecies optimized on winter data did not capture TOA as accurately for summer conditions, simulated on the MACC domain for July 19th through August 1st. A case study over a forest where bSOA was the dominant component of TOA showed that the winter-optimized superspecies did not capture the variability of the mass distribution over the bVOC volatility basis set, overestimating bSOA. A case study over a forest in southern Germany found that winter-optimized superspecies consistently overestimated high bSOA events during summer nights. Heavily forested regions of the domain showed a strong positive bias of time-averaged TOA when advecting winter-optimized superspecies. Overestimation of biogenic SOA brought up a secondary question: under what conditions matrices should be updated. A different set of superspecies were trained on data from July 25th through 29th, and assessed for the same summer dates as the winter-optimized superspecies. With summer-optimized superspecies the biogenic VBS still showed the highest RMSE of the superspecies approach, in the summer, despite having the fewest tracers (8) with non-negligible concentrations. The spatial patterns of error for time-averaged TOA remained, with large bias over highly forested areas in the domain. Preliminary experiments to optimize superspecies on selected spatial data over forests did not improve results, but more analysis is needed for a definitive assessment. However, superspecies optimized on summer conditions captured temporal variation much better than winter-optimized superspecies. A case study of a forest over southern Germany found that using summer-optimized superspecies removed the nightly overestimation shown when using winter-optimized superspecies. The limitations of the machine learned superspecies to model biogenic SOA indicate that the superspecies approach might be suitable for only the other three VBS classes. However, the biogenic VBS class is smaller and therefore has less speedup potential from compression. The biogenic VBS uses 6 volatility bins (12 tracers), and only the 4 highest volatility bins in the model have nonzero concentrations from formation via gas-phase isoprene reactions, as ageing between bins is currently off for the biogenic VBS in LOTOS-EUROS. Removing the unused two volatility bins of the biogenic VBS from advection and other processes would reduce the number of tracers from 12 to 8. The zero-order approach proposed in section 2.7 of advecting TOA and total mass of each bin across phase, with decompression done via partitioning, could further reduce 8 tracers to 5 superspecies (4 bin totals and then either TOA or total biogenic SOA). Improved accuracy is expected from such an approach, as it is not lossy compression: no information will be lost. However, this is a limited compression factor that will limit the speedup potential.

A simulation on the finer-resolution CAMS domain assessed how this method might perform in an operational setting. Winter-optimized superspecies optimized on the MACC domain were used in a run from February 15th through 28th on the CAMS domain. Though an instance of extreme error of TOA predictions from the control run was observed with a positive error of 89 $\mu g\, m^{-3}$ at a high POA event in the superspecies run, this was not representative of the majority of predicted TOA across the domain in this time period: the 99th percentile of TOA error was less than 1 $\mu g\, m^{-3}$ over the whole domain and time period, with an average bias of 0.019 $\mu g\, m^{-3}$ and RMSE of 0.43 $\mu g\, m^{-3}$. Additionally, the gridcell with the highest overestimation event restabilized as the superspecies run continued, converging to values of TOA more similar to that of the control run. The ability of the LOTOS-EUROS superspecies configuration to dampen error and converge back to the behavior of the control configuration indicates that the superspecies approach may be robust to temporary inaccuracies.

7.2. Future directions

The method of machine-learned superspecies for dimension reduction of tracers shows potential, but much remains to explore. One category of future directions is inward looking: exploring VBS superspecies in LOTOS-EUROS in further depth. A second, more outward-looking category broadens the scope to other models and applications, perhaps requiring different unsupervised machine learning algorithms to find an appropriate lower dimensional manifold.

7.2.1. Looking inward

This study developed a parameterization based on machine learning superspecies and took steps towards its practical implementation, including the first extension of LOTOS-EUROS that includes machine learning parameterizations. This LOTOS-EUROS superspecies implementation was tested on the domain used in CAMS operational forecasting and found to remain stable in model simulations of 2 weeks. However, there remain steps in between the results presented in this thesis and use of machine learning parameterizations in LOTOS-EUROS operational forecasts. One of the last and essential steps would be longer simulation periods using the superspecies parameterization, and subsequent comparison of model output to the control simulation for a full validation year.

The failure modes of the superspecies parameterization must be further explored and better understood. This includes continued investigation of the limitations of summer conditions, including the high overestimation of bSOA, in particular over forested areas at night. The biogenic VBS has effectively the smallest number of tracers: though it technically has 6 bins, only the 4 highest volatility bins (8 tracers) can receive material. It cannot yet be concluded that bSOA should not be modeled using superspecies, but it has the least benefit from compression to 4 superspecies via machine learning. The zero-order compression technique proposed in section 2.7 might be worth exploring. At the very least, it is recommended to remove the lowest volatility bins for the biogenic VBS, whose tracers are passed around the model despite never having any mass.

Another path of interest is exploring how other operators in LOTOS-EUROS handle superspecies. All processes besides advection currently deal with both the superspecies and the VBS tracers: this in fact adds the burden of even more extra tracers to most of the model. For this reason, overall runtime in the superspecies version was not as low as it could be, with some processes like vertical diffusion and dry deposition taking more time than the control run just with VBS tracers. Moreover, dealing with superspecies outside of the compression/decompression steps is meaningless and unnecessary, as superspecies concentrations will be overwritten at these steps according to the values of their corresponding tracers. This is in part a development problem and not just a research problem. The timing experiments in chapter 6 should be repeated after this model optimization to give a more realistic idea of the benefit of using superspecies. Decisions should be made on which process should handle the original tracers (at minimum emissions and chemistry) and which processes can use superspecies. Possible contenders are dry deposition and vertical transport. The superspecies approach in this thesis has been designed to be phase specific for future compatibility with dry deposition, another operator that shows significant slowdown with addition of VBS tracers. The unsupervised machine learning method is a process-independent, learning latent characteristics of the VBS tracers themselves. Using the same superspecies in other operators shows promise for this reason.

7.2.2. Looking outward

The previous section points to using this approach for a superspecies representation of VBS tracers in other processes of LOTOS-EUROS, for example, dry deposition and vertical diffusion. However, other models may not have such a large tracer space dedicated to SOA modeling, and may benefit from a superspecies representation of other types of tracers. One potential class of tracers could be inorganic aerosols, especially when modeled by a sectional approach with many size bins.

Future applications of a superspecies approach might want to give consideration to the choice of unsupervised machine learning algorithm. The linear method in this thesis uses non-negative matrix factorization to obtain compression and decompression matrices, and outperformed a nonlinear neural network autoencoder for VBS tracers. Other sets of tracers may be more appropriately modeled by the nonlinear manifold in the hidden layer of an autoencoder. Further refinement of the linear approach is a future direction itself. The objective function and parameters can be adjusted to optimize compression matrix B and decompression matrix W simultaneously, which has similarities to the objective function of archetypal analysis (Cutler & Breiman, 1994). Constraints like column normalization for the composition in strategy 3 could also be imposed during optimization, rather than transforming the matrices after convergence, potentially compromising optimality. Though out of scope for this thesis, dependence of the superspecies on randomly initialized parameters before training should be assessed.

The mass conservation strategy will have to be altered for a nonlinear approach like the autoencoder explored briefly in Chapter 5. This is because the shape of the decompressed tracer distribution is no longer invariant to the two scaling factors introduced in strategy 1. A potential pathway for ensuring mass conservation lies in the activation function of the superspecies layer. One possibility is an activation function that divides the linear combination of the layer before it by the sum of the input layer. If the input to the superspecies layer is non-negative, then the only additional constraint to the the layer weights is non-negativity (purely additive) and this compression step can be viewed as analogous to the compression and scaling step in strategy 1. Decompression and scaling could be done with an activation function that divides the linear combination of the layer input by the sum of the superspecies. Though linear in the superspecies layer, an autoencoder can achieve nonlinear transformations with respect to its input with other layers in its architecture. One result of nonlinearity is dependence on scaling – but a neural network trained with such an activation function and weight constraints would still have parameters optimized to minimize reconstruction error, while conserving mass on compression and decompression. The effect of such an architecture on convergence and stability of long-term predictions would have to be assessed.

A related research question posed by Kelp et al., 2020 is how the latent space of the recurrent autoencoder surrogate model of MOSAIC/CBM-Z, that includes gas-phase and inorganic aerosol tracers, could be configured to interact with processes such as advection. The autoencoder is a surrogate model of computationally intensive chemistry integration, where timestepping is done in the latent space. This thesis focused on advecting a lower dimension, latent space representation of tracers, and could illuminate a path forward for integrating the recurrent autoencoder surrogate model into a chemical transport model or a larger earth system model. The neural network autoencoder might need to change some hyperparameters, such as moving from linear activation functions in the hidden recurrent layer to an activation function and weight parameters that constrain the latent space to non-negative, real numbers. Interpretation of the recurrent autoencoder latent space as a superspecies representation of tracers is important for their use in other processes. Incorporating mass conservation with the autoencoder is made difficult not by the autoencoder, but by the nature of the tracers. The VBS tracers were not speciated, but rather concentration of organic material lumped into volatility bins, a modeling approach developed because of large uncertainty in the chemical makeup of organic aerosol. Tracers in a chemical mechanism, on the other hand, are defined by their unique atomic makeup, and conserving mass would involve conserving atoms. Though some total metric could be conserved with an activation function with a scaling factor, it is not clear if this is an essential or relevant property, or even possible, as chemical mechanisms are not necessarily designed to conserve mass (Heald & Kroll, 2020). However, the recurrent autoencoder explored by Kelp et al., 2020 showed long-term stability without explicit mass conservation in hidden layers. A potential direction for further research could be whether lack of mass conservation can lead to runaway error when superspecies interact with other processes, and if so, ways to maintain stability while interacting with other processes.

This study developed machine learning methods to represent organic aerosol tracers with a smaller number of superspecies, reducing the computational burden of advection in LOTOS-EUROS. Applying the advection operator on superspecies took approximately 56% through 66% of the time it took using the original tracers. The first implementation of a machine learning parameterization in LOTOS-EUROS was developed, which ran stably on model simulations of 2 weeks without experiencing runaway error. Future directions for this work include further development of the superspecies parameterization in LOTOS EUROS, and broadening the scope to other potential uses of the superspecies to accelerate 3D atmospheric models.

A

Timing

Every LOTOS-EUROS run that finishes successfully writes a file ”lotos-euros.prf” as part of its output, which includes information on timing of various parts of the model. Excerpts of these .prf files used in the experiments in the thesis are included to supplement the figures shown in timing.

A.1. 1/24 of the CAMS domain, sequential run without VBS

```
# ----------------------------------------  ------------ ---------
# timer                                     system_clock       (%)
# ----------------------------------------  ------------ ---------
#
# root                                          2497.41
#   model init                                    99.30 (  4.0 %)
#   model time loop                             2396.09 ( 95.9 %)
#   model done                                     0.75 (  0.0 %)
#   time step first                                1.27 (  0.1 %)
#   other                                          0.00 (  0.0 %)
#
# model init                                      99.30
#
# model time loop                               2396.09
#   time step output                              11.96 (  0.5 %)
#   time step save                                 2.10 (  0.1 %)
#   time step setup                              453.43 ( 18.9 %)
#   particle update                                1.49 (  0.1 %)
#   chemistry                                    831.32 ( 34.7 %)
#   vertical diffusion                           213.00 (  8.9 %)
#   sedimentation                                 48.43 (  2.0 %)
#   dry deposition                               151.09 (  6.3 %)
#   wet deposition                               315.60 ( 13.2 %)
#   advection                                    360.90 ( 15.1 %)
#   emission                                       2.73 (  0.1 %)
#   dry deposition velocities                      1.10 (  0.0 %)
#   other                                          2.95 (  0.1 %)
#
# model done                                      0.75
# ----------------------------------------  ------------ ---------
```

A.2. 1/24th of CAMS domain, sequential run with VBS

```
    # ----------------------------------------  ------------ ---------
```

```
# timer                                        system_clock        (%)
# -------------------------------------------  ------------ ---------
#
# root                                              3935.69
#   model init                                        19.56 (  0.5 %)
#   model time loop                                 3914.15 ( 99.5 %)
#   model done                                         0.47 (  0.0 %)
#   time step first                                    1.50 (  0.0 %)
#
# model init                                          19.56
#
# model time loop                                   3914.15
#   time step output                                  11.10 (  0.3 %)
#   time step save                                     1.43 (  0.0 %)
#   time step setup                                  435.28 ( 11.1 %)
#   particle update                                    3.05 (  0.1 %)
#   chemistry                                       1121.99 ( 28.7 %)
#   vertical diffusion                               434.11 ( 11.1 %)
#   sedimentation                                     83.72 (  2.1 %)
#   dry deposition                                   372.92 (  9.5 %)
#   wet deposition                                   632.65 ( 16.2 %)
#   advection                                        805.56 ( 20.6 %)
#   emission                                           5.64 (  0.1 %)
#   dry deposition velocities                          1.14 (  0.0 %)
#   other                                              5.58 (  0.1 %)
#
# model done                                          0.47
# gas-phase chemistry                               689.81
#   vbs chemistry                                    14.26 (  2.1 %)
#   other                                           675.54 ( 97.9 %)
# -------------------------------------------  ------------ ---------
```

A.3. Entire CAMS Domain, parallel run without VBS

```
    # -------------------------------------------  ------------ ---------
# timer                                        system_clock        (%)
# -------------------------------------------  ------------ ---------
#
# root                                              7024.28
#   model init                                        32.67 (  0.5 %)
#   model time loop                                 6979.00 ( 99.4 %)
#   model done                                         9.86 (  0.1 %)
#   time step first                                    2.50 (  0.0 %)
#   other                                              0.24 (  0.0 %)
#
# model init                                          32.67
#
# model time loop                                   6979.00
#   time step output                                 138.90 (  2.0 %)
#   time step save                                    31.12 (  0.4 %)
#   time step setup                                  680.01 (  9.7 %)
#   particle update                                    1.80 (  0.0 %)
#   chemistry                                       1413.30 ( 20.3 %)
#   vertical diffusion                               602.81 (  8.6 %)
#   sedimentation                                    111.97 (  1.6 %)
#   dry deposition                                   289.69 (  4.2 %)
#   wet deposition                                   191.36 (  2.7 %)
```

```
#   advection                                  3501.89 ( 50.2 %)
#   emission                                      7.42 (  0.1 %)
#   dry deposition velocities                     0.68 (  0.0 %)
#   other                                         8.04 (  0.1 %)
#
# model done                                      9.86
# ----------------------------------------  ------------ ---------
```

A.4. Entire CAMS domain, parallel run, with VBS

```
    # ----------------------------------------  ------------ ---------
# timer                                     system_clock        (%)
# ----------------------------------------  ------------ ---------
#
# root                                        13095.98
#   model init                                   37.73 (  0.3 %)
#   model time loop                           13046.01 ( 99.6 %)
#   model done                                    8.54 (  0.1 %)
#   time step first                               3.68 (  0.0 %)
#   other                                         0.02 (  0.0 %)
#
# model init                                      37.73
#
# model time loop                             13046.01
#   time step output                             198.88 (  1.5 %)
#   time step save                                61.59 (  0.5 %)
#   time step setup                              784.02 (  6.0 %)
#   particle update                                3.64 (  0.0 %)
#   chemistry                                   1906.46 ( 14.6 %)
#   vertical diffusion                          1253.76 (  9.6 %)
#   sedimentation                                194.76 (  1.5 %)
#   dry deposition                               702.60 (  5.4 %)
#   wet deposition                               404.38 (  3.1 %)
#   advection                                   7434.81 ( 57.0 %)
#   emission                                      15.71 (  0.1 %)
#   dry deposition velocities                      0.70 (  0.0 %)
#   other                                         84.68 (  0.6 %)
#
# model done                                      8.54
# gas-phase chemistry                         1185.36
#   vbs chemistry                                 36.03 (  3.0 %)
#   other                                       1149.32 ( 97.0 %)
# ----------------------------------------  ------------ ---------
```

A.5. MACC domain without VBS

```
# ----------------------------------------  ------------ ---------
# timer                                     system_clock        (%)
# ----------------------------------------  ------------ ---------
#
# root                                        23093.64
#   model init                                  107.49 (  0.5 %)
#   model time loop                           22984.23 ( 99.5 %)
#   model done                                    0.53 (  0.0 %)
#   time step first                               1.39 (  0.0 %)
#   other                                         0.00 (  0.0 %)
#
```

```
# model init                                    107.49
#
# model time loop                             22984.23
#   time step output                            155.16 (  0.7 %)
#   time step save                                6.38 (  0.0 %)
#   time step setup                            8642.48 ( 37.6 %)
#   adjust                                      144.85 (  0.6 %)
#   particle update                              10.23 (  0.0 %)
#   chemistry                                  5766.36 ( 25.1 %)
#   vertical diffusion                          933.95 (  4.1 %)
#   sedimentation                               187.26 (  0.8 %)
#   dry deposition                             2035.03 (  8.9 %)
#   wet deposition                              346.71 (  1.5 %)
#   advection                                  4638.53 ( 20.2 %)
#   emission                                      9.33 (  0.0 %)
#   dry deposition velocities                    50.37 (  0.2 %)
#   other                                        57.59 (  0.3 %)
#
# model done                                      0.53
# ----------------------------------------  ------------ ---------
```

A.6. MACC domain with VBS

```
# ----------------------------------------  ------------ ---------
# timer                                     system_clock        (%)
# ----------------------------------------  ------------ ---------
#
# root                                        40140.00
#   model init                                  104.02 (  0.3 %)
#   model time loop                           40030.97 ( 99.7 %)
#   model done                                    0.38 (  0.0 %)
#   time step first                               4.64 (  0.0 %)
#   other                                         0.01 (  0.0 %)
#
# model init                                    104.02
#
# model time loop                             40030.97
#   time step output                             83.69 (  0.2 %)
#   time step save                               10.27 (  0.0 %)
#   time step setup                            6427.40 ( 16.1 %)
#   adjust                                      325.94 (  0.8 %)
#   particle update                              27.02 (  0.1 %)
#   chemistry                                 11255.61 ( 28.1 %)
#   vertical diffusion                         2069.52 (  5.2 %)
#   sedimentation                               293.77 (  0.7 %)
#   dry deposition                             6417.42 ( 16.0 %)
#   wet deposition                              817.53 (  2.0 %)
#   advection                                 12073.01 ( 30.2 %)
#   emission                                     22.56 (  0.1 %)
#   dry deposition velocities                    57.58 (  0.1 %)
#   other                                       149.66 (  0.4 %)
#
# model done                                      0.38
# ----------------------------------------  ------------ ---------
```

A.7. MACC domain with superspecies

Note the breakdown of advection timing into ”vbs machine learning”, which includes the compression and decompression, takes about 9% of the total time required for the advection operator.

```
# ----------------------------------------  ------------ ---------
# timer                                    system_clock       (%)
# ----------------------------------------  ------------ ---------
#
# root                                         34287.40
#   model init                                    91.28 (  0.3 %)
#   model time loop                            34191.84 ( 99.7 %)
#   model done                                     0.29 (  0.0 %)
#   time step first                                3.98 (  0.0 %)
#   other                                          0.01 (  0.0 %)
#
# model init                                       91.28
#
# model time loop                              34191.84
#   time step output                             102.24 (  0.3 %)
#   time step save                                16.27 (  0.0 %)
#   time step setup                             7071.29 ( 20.7 %)
#   adjust                                       332.61 (  1.0 %)
#   particle update                               27.89 (  0.1 %)
#   chemistry                                  10215.30 ( 29.9 %)
#   vertical diffusion                          2123.27 (  6.2 %)
#   sedimentation                                276.60 (  0.8 %)
#   dry deposition                              6193.60 ( 18.1 %)
#   wet deposition                               827.09 (  2.4 %)
#   advection                                   6789.65 ( 19.9 %)
#   emission                                      21.72 (  0.1 %)
#   dry deposition velocities                     51.29 (  0.2 %)
#   other                                        143.04 (  0.4 %)
#
# model done                                        0.29
#
# advection                                    6789.65
#   vbs machine learning                        604.37 (  8.9 %)
#   other                                      6185.28 ( 91.1 %)
# ----------------------------------------  ------------ ---------
```

A.8. CAMS domain without VBS

Below is an excerpt of the .prf file for the LOTOS-EUROS run from February 15th through 28th, using the CAMS domain without the VBS tracers.

```
# ----------------------------------------  ------------ ---------
# timer                                    system_clock       (%)
# ----------------------------------------  ------------ ---------
#
# root                                         76124.10
#   model init                                    62.40 (  0.1 %)
#   model time loop                            76055.44 ( 99.9 %)
#   model done                                     4.21 (  0.0 %)
#   time step first                                2.05 (  0.0 %)
#   other                                          0.01 (  0.0 %)
#
# model init                                       62.40
```

```
#
# model time loop                                76055.44
#   time step output                               870.41 (  1.1 %)
#   time step save                                 111.83 (  0.1 %)
#   time step setup                               9458.12 ( 12.4 %)
#   adjust                                         152.46 (  0.2 %)
#   particle update                                  9.93 (  0.0 %)
#   chemistry                                    15844.97 ( 20.8 %)
#   vertical diffusion                            4264.43 (  5.6 %)
#   sedimentation                                  797.35 (  1.0 %)
#   dry deposition                                7484.25 (  9.8 %)
#   wet deposition                                1901.20 (  2.5 %)
#   advection                                    34958.76 ( 46.0 %)
#   emission                                        40.74 (  0.1 %)
#   dry deposition velocities                       57.87 (  0.1 %)
#   other                                          103.10 (  0.1 %)
#
# model done                                         4.21
# ----------------------------------------  ------------ ---------
```

A.9. CAMS domain with VBS

Below is an excerpt of the .prf timing file for the LOTOS-EUROS control run from February 15th through 28th, using the CAMS domain with the VBS tracers.

```
# ----------------------------------------  ------------ ---------
# timer                                     system_clock        (%)
# ----------------------------------------  ------------ ---------
#
# root                                        140733.18
#   model init                                   36.69 (  0.0 %)
#   model time loop                           140690.83 (100.0 %)
#   model done                                    1.10 (  0.0 %)
#   time step first                               4.55 (  0.0 %)
#   other                                         0.01 (  0.0 %)
#
# model init                                      36.69
#
# model time loop                             140690.83
#   time step output                             806.73 (  0.6 %)
#   time step save                               298.29 (  0.2 %)
#   time step setup                             8868.14 (  6.3 %)
#   adjust                                       304.12 (  0.2 %)
#   particle update                               22.06 (  0.0 %)
#   chemistry                                  23727.95 ( 16.9 %)
#   vertical diffusion                          7891.11 (  5.6 %)
#   sedimentation                               1036.56 (  0.7 %)
#   dry deposition                             18710.86 ( 13.3 %)
#   wet deposition                              3905.99 (  2.8 %)
#   advection                                  74762.33 ( 53.1 %)
#   emission                                      85.60 (  0.1 %)
#   dry deposition velocities                     54.89 (  0.0 %)
#   other                                        216.21 (  0.2 %)
#
# model done                                        1.10
```

A.10. CAMS domain with superspecies

Below is an excerpt of the .prf timing file for the LOTOS-EUROS control run from February 15th through 28th, using the CAMS domain with the superspecies advected. Note the breakdown of advection timing into ”vbs machine learning”, which includes the compression and decompression, takes about 5% of the total time required for the advection operator

```
# ----------------------------------------  ------------ ---------
# timer                                     system_clock        (%)
# ----------------------------------------  ------------ ---------
#
# root                                        119300.65
#   model init                                    43.27 (  0.0 %)
#   model time loop                           119251.07 (100.0 %)
#   model done                                     1.59 (  0.0 %)
#   time step first                                4.72 (  0.0 %)
#   other                                          0.00 (  0.0 %)
#
# model init                                       43.27
#
# model time loop                             119251.07
#   time step output                             974.53 (  0.8 %)
#   time step save                               121.95 (  0.1 %)
#   time step setup                             6818.19 (  5.7 %)
#   adjust                                       345.05 (  0.3 %)
#   particle update                               26.53 (  0.0 %)
#   chemistry                                  23848.70 ( 20.0 %)
#   vertical diffusion                          9266.85 (  7.8 %)
#   sedimentation                               1144.30 (  1.0 %)
#   dry deposition                             20101.16 ( 16.9 %)
#   wet deposition                              6735.99 (  5.6 %)
#   advection                                  49472.83 ( 41.5 %)
#   emission                                     90.80 (  0.1 %)
#   dry deposition velocities                    52.26 (  0.0 %)
#   other                                       251.94 (  0.2 %)
#
# model done                                        1.59
#
# advection                                   49472.83
#   vbs machine learning                       2633.92 (  5.3 %)
#   other                                     46838.91 ( 94.7 %)
# ----------------------------------------  ------------ ---------
```

B

Superspecies Matrices

B.1. Winter Superspecies

This section reports the aerosol and gas phase compression matrices B_{aer} and B_{gas} as well as aerosol and gas phase decompression matrices W_{aer} and W_{gas}, using 2 phase-specific superspecies per phase per class, optimized on winter conditions. They were developed in chapter 5 and used with mass balancing strategy 1 from section 5.2.1. Their performance was assessed offline in chapter 5 and online in chapter 6.

The results shown here can be adjusted to become composition matrices for mass balancing strategy 3 from section 5.2.3. To get compression matrices for this method, set B_{aer} and B_{gas} to the respective transposes of W_{aer} and W_{gas}, then scale each column to sum to 1. For decompression matrices, just scale each column of W_{aer} and W_{gas} to sum to 1.

B.1.1. Anthropogenic matrices

$$B_{aer} = \begin{bmatrix} 0.00e+00 & 0.00e+00 & 0.00e+00 & 5.91e-01 & 3.83e-02 & 9.39e+00 \\ 6.29e-01 & 1.90e-01 & 6.21e-01 & 0.00e+00 & 0.00e+00 & 0.00e+00 \end{bmatrix} \quad \text{(B.1)}$$

$$B_{gas} = \begin{bmatrix} 0.00e+00 & 5.26e-01 & 2.31e-01 & 0.00e+00 & 0.00e+00 & 0.00e+00 \\ 0.00e+00 & 0.00e+00 & 0.00e+00 & 0.00e+00 & 0.00e+00 & 2.34e-01 \end{bmatrix} \quad \text{(B.2)}$$

$$W_{aer} = \begin{bmatrix} 0.00e+00 & 9.24e-02 \\ 0.00e+00 & 2.40e-01 \\ 2.06e-01 & 1.27e+00 \\ 1.12e+00 & 2.97e-01 \\ 2.40e-01 & 0.00e+00 \\ 1.62e-02 & 0.00e+00 \end{bmatrix} \quad \text{(B.3)}$$

$$W_{gas} = \begin{bmatrix} 4.44e-01 & 0.00e+00 \\ 8.40e-01 & 0.00e+00 \\ 2.00e+00 & 2.37e-01 \\ 3.30e+00 & 2.01e+00 \\ 3.14e+00 & 3.06e+00 \\ 1.63e+00 & 3.75e+00 \end{bmatrix} \quad \text{(B.4)}$$

B.1.2. Biogenic matrices

Note that the lowest two volatility bins do not recieve material in LOTOS-EUROS, though they are not turned off by default in many processes, including advection.

$$B_{aer} = \begin{bmatrix} 0.00e+00 & 0.00e+00 & 1.99e-01 & 1.25e-02 & 0.00e+00 & 0.00e+00 \\ 0.00e+00 & 0.00e+00 & 0.00e+00 & 3.05e-02 & 1.97e-01 & 1.09e-01 \end{bmatrix} \quad \text{(B.5)}$$

$$B_{gas} = \begin{bmatrix} 0.00e+00 & 0.00e+00 & 0.00e+00 & 0.00e+00 & 2.89e-01 & 6.85e-03 \\ 1.05e+02 & 7.36e+01 & 0.00e+00 & 3.23e-01 & 0.00e+00 & 0.00e+00 \end{bmatrix} \quad \text{(B.6)}$$

$$W_{aer} = \begin{bmatrix} 1.54e-10 & 0.00e+00 \\ 7.77e-11 & 0.00e+00 \\ 4.79e+00 & 0.00e+00 \\ 1.94e+00 & 2.63e+00 \\ 1.57e-01 & 3.79e+00 \\ 0.00e+00 & 9.64e-01 \end{bmatrix} \quad \text{(B.7)}$$

$$W_{gas} = \begin{bmatrix} 5.04e-07 & 4.27e-06 \\ 5.04e-07 & 4.26e-06 \\ 4.73e-01 & 1.74e-02 \\ 7.99e-01 & 1.17e+00 \\ 3.29e+00 & 0.00e+00 \\ 5.86e+00 & 2.53e+00 \end{bmatrix} \quad \text{(B.8)}$$

B.1.3. POA matrices

Note that for this VBS class the transposes of the compression matrices B_{aer} and B_{gas} are given for ease of printing.

$$B_{aer}^{T} = \begin{bmatrix} 0.00e+00 & 3.10e-01 \\ 0.00e+00 & 0.00e+00 \\ 3.14e-03 & 0.00e+00 \\ 1.98e-01 & 0.00e+00 \\ 0.00e+00 & 0.00e+00 \\ 0.00e+00 & 0.00e+00 \\ 0.00e+00 & 0.00e+00 \\ 0.00e+00 & 0.00e+00 \\ 0.00e+00 & 0.00e+00 \end{bmatrix} \tag{B.9}$$

$$B_{gas}^{T} = \begin{bmatrix} 0.00e+00 & 3.11e-03 \\ 0.00e+00 & 0.00e+00 \\ 0.00e+00 & 2.88e-01 \\ 0.00e+00 & 4.16e-02 \\ 4.78e-02 & 0.00e+00 \\ 4.60e-01 & 0.00e+00 \\ 4.49e-02 & 0.00e+00 \\ 4.94e-03 & 0.00e+00 \\ 3.95e-04 & 0.00e+00 \end{bmatrix} \tag{B.10}$$

$$W_{aer} = \begin{bmatrix} 1.28e+00 & 1.54e+00 \\ 2.65e+00 & 2.99e+00 \\ 4.29e+00 & 2.79e+00 \\ 4.95e+00 & 0.00e+00 \\ 2.24e-01 & 0.00e+00 \\ 6.07e-02 & 0.00e+00 \\ 1.48e-02 & 0.00e+00 \\ 2.01e-03 & 0.00e+00 \\ 2.60e-04 & 0.00e+00 \end{bmatrix} \tag{B.11}$$

$$W_{gas} = \begin{bmatrix} 0.00e+00 & 8.32e-01 \\ 0.00e+00 & 1.11e+00 \\ 1.54e-02 & 2.49e+00 \\ 9.14e-01 & 3.90e+00 \\ 7.41e-01 & 3.31e-01 \\ 1.73e+00 & 1.78e-01 \\ 2.65e+00 & 1.63e-01 \\ 3.54e+00 & 1.95e-01 \\ 4.43e+00 & 2.41e-01 \end{bmatrix} \tag{B.12}$$

B.1.4. siSOA matrices

Note that for this VBS class the transposes of the compression matrices B_{aer} and B_{gas} are given for ease of printing.

$$B_{aer}^{T} = \begin{bmatrix} 1.45e-01 & 0.00e+00 \\ 4.18e-02 & 3.96e-02 \\ 0.00e+00 & 1.87e-01 \\ 0.00e+00 & 0.00e+00 \\ 0.00e+00 & 0.00e+00 \\ 0.00e+00 & 0.00e+00 \\ 0.00e+00 & 0.00e+00 \\ 0.00e+00 & 0.00e+00 \end{bmatrix} \tag{B.13}$$

$$B_{gas}^{T} = \begin{bmatrix} 0.00e+00 & 1.81e-01 \\ 0.00e+00 & 0.00e+00 \\ 0.00e+00 & 0.00e+00 \\ 0.00e+00 & 1.79e-01 \\ 0.00e+00 & 0.00e+00 \\ 0.00e+00 & 0.00e+00 \\ 1.03e-01 & 0.00e+00 \\ 1.61e-01 & 0.00e+00 \end{bmatrix} \tag{B.14}$$

$$W_{aer} = \begin{bmatrix} 5.31e+00 & 0.00e+00 \\ 3.72e+00 & 3.15e+00 \\ 0.00e+00 & 3.37e+00 \\ 0.00e+00 & 3.01e-01 \\ 0.00e+00 & 5.58e-02 \\ 0.00e+00 & 4.43e-03 \\ 0.00e+00 & 9.35e-04 \\ 0.00e+00 & 4.52e-05 \end{bmatrix} \tag{B.15}$$

$$W_{gas} = \begin{bmatrix} 9.26e-01 & 1.73e+00 \\ 1.41e+00 & 1.76e+00 \\ 1.85e+00 & 2.32e+00 \\ 1.00e+00 & 1.93e+00 \\ 2.13e+00 & 1.79e+00 \\ 3.15e+00 & 1.39e+00 \\ 3.81e+00 & 7.13e-01 \\ 3.40e+00 & 0.00e+00 \end{bmatrix} \tag{B.16}$$

B.2. Summer Superspecies

This section reports the aerosol and gas phase compression matrices B_{aer} and B_{gas} as well as aerosol and gas phase decompression matrices W_{aer} and W_{gas}, using 2 phase-specific superspecies per phase per class, optimized on summer conditions. They were used with mass balancing strategy 1 from section 5.2.1. They were created in chapter 6 to assess whether superspecies optimized on the seasonal conditions they are evaluated on increases accuracy.

B.2.1. Anthropogenic matrices

$$B_{aer} = \begin{bmatrix} 0.00e+00 & 1.59e-01 & 0.00e+00 & 0.00e+00 & 0.00e+00 & 0.00e+00 \\ 2.05e-01 & 0.00e+00 & 3.60e-02 & 1.34e-01 & 0.00e+00 & 0.00e+00 \end{bmatrix} \tag{B.17}$$

$$B_{gas} = \begin{bmatrix} 1.53e-01 & 2.62e-02 & 4.12e-03 & 0.00e+00 & 0.00e+00 & 0.00e+00 \\ 0.00e+00 & 0.00e+00 & 0.00e+00 & 6.73e-02 & 1.83e-01 & 4.00e-02 \end{bmatrix} \tag{B.18}$$

$$W_{aer} = \begin{bmatrix} 2.19e-01 & 3.86e+00 \\ 6.21e+00 & 3.72e-01 \\ 1.18e-01 & 1.03e+00 \\ 0.00e+00 & 1.51e-01 \\ 0.00e+00 & 8.27e-03 \\ 0.00e+00 & 2.31e-04 \end{bmatrix} \tag{B.19}$$

$$W_{aer} = \begin{bmatrix} 6.03e+00 & 0.00e+00 \\ 1.60e+00 & 9.60e-01 \\ 1.67e+00 & 1.97e+00 \\ 9.08e-01 & 3.33e+00 \\ 1.47e-01 & 3.34e+00 \\ 0.00e+00 & 2.57e+00 \end{bmatrix} \tag{B.20}$$

B.2.2. Biogenic matrices

Note that the NMF approach converged to a very high value for the second tracer for the second superspecies of B_{aer}. A common limitation of machine learning algorithms is their sensitivity to randomly initialized parameters, which sometimes can lead to poor convergence. In light of negligible concentrations for the lowest 2 volatility bins of the biogenic VBS, perhaps adjusting this to zero would improve results. As these matrices are small, they can be checked by hand before online implementation, to correct values arising from poor convergence

$$B_{aer} = \begin{bmatrix} 0.00e+00 & 0.00e+00 & 0.00e+00 & 2.87e-02 & 1.17e-01 & 1.77e-01 \\ 0.00e+00 & 1.35e+07 & 1.61e-01 & 0.00e+00 & 0.00e+00 & 0.00e+00 \end{bmatrix} \tag{B.21}$$

$$B_{gas} = \begin{bmatrix} 0.00e+00 & 0.00e+00 & 0.00e+00 & 0.00e+00 & 0.00e+00 & 1.26e-01 \\ 0.00e+00 & 0.00e+00 & 2.18e-01 & 3.30e-01 & 5.39e-03 & 0.00e+00 \end{bmatrix} \tag{B.22}$$

$$W_{aer} = \begin{bmatrix} 0.00e+00 & 1.23e-11 \\ 0.00e+00 & 1.28e-11 \\ 2.37e+00 & 4.58e+00 \\ 4.79e+00 & 2.47e+00 \\ 4.87e+00 & 8.34e-02 \\ 1.19e+00 & 0.00e+00 \end{bmatrix} \tag{B.23}$$

$$W_{gas} = \begin{bmatrix} 0.00e+00 & 6.69e-07 \\ 0.00e+00 & 6.69e-07 \\ 0.00e+00 & 5.40e-01 \\ 6.35e-02 & 2.40e+00 \\ 2.98e+00 & 1.96e+00 \\ 7.08e+00 & 1.93e+00 \end{bmatrix} \tag{B.24}$$

B.2.3. POA matrices

Note that for this VBS class the transposes of the compression matrices B_{aer} and B_{gas} are given for ease of printing.

$$B_{aer}^{T} = \begin{bmatrix} 0.00e+00 & 0.00e+00 \\ 0.00e+00 & 1.52e-01 \\ 2.32e-02 & 0.00e+00 \\ 1.63e-01 & 0.00e+00 \\ 4.41e-02 & 0.00e+00 \\ 0.00e+00 & 0.00e+00 \\ 0.00e+00 & 0.00e+00 \\ 0.00e+00 & 0.00e+00 \\ 0.00e+00 & 0.00e+00 \end{bmatrix} \tag{B.25}$$

$$B_{gas}^{T} = \begin{bmatrix} 0.00e+00 & 4.05e-02 \\ 0.00e+00 & 0.00e+00 \\ 0.00e+00 & 1.54e-01 \\ 0.00e+00 & 7.38e-02 \\ 0.00e+00 & 0.00e+00 \\ 2.86e-01 & 0.00e+00 \\ 1.75e-01 & 0.00e+00 \\ 1.57e-02 & 0.00e+00 \\ 1.31e-03 & 0.00e+00 \end{bmatrix} \tag{B.26}$$

$$W_{aer} = \begin{bmatrix} 9.48e-01 & 1.19e+00 \\ 1.62e+00 & 3.43e+00 \\ 3.50e+00 & 2.22e+00 \\ 5.36e+00 & 0.00e+00 \\ 5.26e-01 & 0.00e+00 \\ 2.62e-01 & 0.00e+00 \\ 5.08e-02 & 0.00e+00 \\ 7.20e-03 & 0.00e+00 \\ 1.08e-03 & 0.00e+00 \end{bmatrix} \tag{B.27}$$

$$W_{gas} = \begin{bmatrix} 8.31e-02 & 1.39e+00 \\ 1.11e-01 & 1.24e+00 \\ 1.71e-01 & 3.10e+00 \\ 5.04e-01 & 4.45e+00 \\ 5.89e-01 & 4.85e-01 \\ 1.61e+00 & 2.43e-01 \\ 2.59e+00 & 4.58e-02 \\ 3.49e+00 & 4.63e-03 \\ 4.37e+00 & 0.00e+00 \end{bmatrix} \tag{B.28}$$

B.2.4. siSOA matrices

Note that for this VBS class the transposes of the compression matrices B_{aer} and B_{gas} are given for ease of printing.

$$B_{aer}^{T} = \begin{bmatrix} 1.50e-01 & 0.00e+00 \\ 0.00e+00 & 1.65e-01 \\ 0.00e+00 & 4.66e-02 \\ 0.00e+00 & 7.27e-03 \\ 0.00e+00 & 0.00e+00 \\ 0.00e+00 & 5.23e-04 \\ 0.00e+00 & 0.00e+00 \\ 0.00e+00 & 0.00e+00 \end{bmatrix} \tag{B.29}$$

$$B_{gas}^{T} = \begin{bmatrix} 1.56e-01 & 0.00e+00 \\ 1.66e-02 & 0.00e+00 \\ 0.00e+00 & 3.81e-02 \\ 0.00e+00 & 9.42e-03 \\ 0.00e+00 & 9.49e-02 \\ 0.00e+00 & 9.50e-02 \\ 0.00e+00 & 7.02e-02 \\ 0.00e+00 & 6.58e-02 \end{bmatrix} \tag{B.30}$$

$$W_{aer} = \begin{bmatrix} 6.52e+00 & 7.07e-01 \\ 6.64e-02 & 5.55e+00 \\ 0.00e+00 & 1.47e+00 \\ 0.00e+00 & 2.13e-01 \\ 0.00e+00 & 5.36e-02 \\ 0.00e+00 & 5.02e-03 \\ 0.00e+00 & 7.82e-04 \\ 0.00e+00 & 4.19e-05 \end{bmatrix} \tag{B.31}$$

$$W_{gas} = \begin{bmatrix} 6.18e+00 & 9.37e-03 \\ 1.26e+00 & 1.15e+00 \\ 8.64e-01 & 1.96e+00 \\ 4.69e-01 & 1.68e+00 \\ 2.51e-01 & 2.41e+00 \\ 4.00e-02 & 2.95e+00 \\ 0.00e+00 & 3.03e+00 \\ 0.00e+00 & 2.38e+00 \end{bmatrix} \tag{B.32}$$

Bibliography

Baklanov, A., Schlünzen, K., Suppan, P., Baldasano, J., Brunner, D., Aksoyoglu, S., Carmichael, G., Douros, J., Flemming, J., Forkel, R., et al. (2014). Online coupled regional meteorology chemistry models in europe: Current status and prospects. *Atmospheric Chemistry and Physics*, *14*(1), 317–398.

Bergström, R., Denier van der Gon, H. A. C., Prévôt, A. S. H., Yttri, K. E., & Simpson, D. (2012). Modelling of organic aerosols over europe (2002ndash;2007) using a volatility basis set (vbs) framework: Application of different assumptions regarding the formation of secondary organic aerosol. *Atmospheric Chemistry and Physics*, *12*(18), 8499–8527. https://doi.org/10.5194/acp-12-8499-2012

Brasseur, G. P., & Jacob, D. J. (2017). *Modeling of atmospheric chemistry*. Cambridge University Press. https://doi.org/10.1017/9781316544754

Cao, B., Shen, D., Sun, J.-T., Wang, X., Yang, Q., & Chen, Z. (2007). Detect and track latent factors with online nonnegative matrix factorization. *IJCAI*, *7*, 2689–2694.

Cho, D., Yoo, C., Im, J., & Cha, D.-H. (2020). Comparative assessment of various machine learning-based bias correction methods for numerical weather prediction model forecasts of extreme air temperatures in urban areas [e2019EA000740 2019EA000740]. *Earth and Space Science*, *7*(4), e2019EA000740. https://doi.org/https://doi.org/10.1029/2019EA000740

Colette, A., Solberg, S., Aas, W., & Walker, S. (2020). Understanding air quality trends in europe. *Focus on the relative contribution of changes in emission of activity sectors, natural fraction and meteorological variability, European Topic Centre on Air pollution, transport, noise and industrial pollution, Kjeller, Norway, EIONET Report–ETC/ATNI*, *8*.

Colette, A., Solberg, S., Beauchamp, M., Bessagnet, B., Malherbe, L., Guerreiro, C., Andersson, A., Cuvelier, C., Manders, A., Mar, K. A., et al. (2017). Long term air quality trends in europe: Contribution of meteorological variability, natural factors and emissions.

Cutler, A., & Breiman, L. (1994). Archetypal analysis. *Technometrics*, *36*(4), 338–347.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, *2*(4), 303–314.

De Gouw, J., Middlebrook, A., Warneke, C., Goldan, P., Kuster, W., Roberts, J., Fehsenfeld, F., Worsnop, D., Canagaratna, M., Pszenny, A., et al. (2005). Budget of organic carbon in a polluted atmosphere: Results from the new england air quality study in 2002. *Journal of Geophysical Research: Atmospheres*, *110*(D16).

Dockery, D., Pope, C., Xu, X., Spengler, J., Ware, J., Fay, M., Ferris, B., & Speizer, F. (1994). An association between air pollution and mortality in six u.s. cities. *The New England journal of medicine*, *329*, 1753–9. https://doi.org/10.1056/NEJM199312093292401

Donahue, N. M., Robinson, A., Stanier, C., & Pandis, S. (2006). Coupled partitioning, dilution, and chemical aging of semivolatile organics. *Environmental science & technology*, *40*(8), 2635–2643.

Drosatou, A. D., Skyllakou, K., Theodoritsi, G. N., & Pandis, S. N. (2019). Positive matrix factorization of organic aerosol: Insights from a chemical transport model. *Atmospheric Chemistry and Physics*, *19*(2), 973–986.

Forster, P. (2007). Changes in atmospheric constituents and in radiative forcing, in climate change 2007: The physical science basis. *Contribution of Working Group i to the Fourth Assessment Report of the Intergovernmental Panel on Climate Change*, *2007*.

Gardner, M. W., & Dorling, S. (1998). Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. *Atmospheric environment*, *32*(14-15), 2627–2636.

Gillis, N. (2014). The why and how of nonnegative matrix factorization.

Golaz, J.-C., Caldwell, P. M., Van Roekel, L. P., Petersen, M. R., Tang, Q., Wolfe, J. D., Abeshu, G., Anantharaj, V., Asay-Davis, X. S., Bader, D. C., Baldwin, S. A., Bisht, G., Bogenschutz, P. A., Branstetter, M., Brunke, M. A., Brus, S. R., Burrows, S. M., Cameron-Smith, P. J., Donahue,

A. S., … Zhu, Q. (2019). The doe e3sm coupled model version 1: Overview and evaluation at standard resolution. *Journal of Advances in Modeling Earth Systems*, *11*(7), 2089–2129. https://doi.org/https://doi.org/10.1029/2018MS001603

Guan, N., Tao, D., Luo, Z., & Yuan, B. (2012). Online nonnegative matrix factorization with robust stochastic approximation. *IEEE Transactions on Neural Networks and Learning Systems*, *23*(7), 1087–1099.

Hammond, J. K. (2017). *Reduced basis methods for urban air quality modeling* (Doctoral dissertation). Université Paris Est-Marne-la-Vallée; IFSTTAR-Institut Français des Sciences …

He, X., & Niyogi, P. (2004). Locality preserving projections. *Advances in neural information processing systems*, *16*(16), 153–160.

Heald, C. L., Henze, D., Horowitz, L., Feddema, J., Lamarque, J.-F., Guenther, A., Hess, P., Vitt, F., Seinfeld, J., Goldstein, A., et al. (2008). Predicted change in global secondary organic aerosol concentrations in response to future climate, emissions, and land use change. *Journal of Geophysical Research: Atmospheres*, *113*(D5).

Heald, C. L., Jacob, D. J., Park, R. J., Russell, L. M., Huebert, B. J., Seinfeld, J. H., Liao, H., & Weber, R. J. (2005). A large organic aerosol source in the free troposphere missing from current models. *Geophysical Research Letters*, *32*(18). https://doi.org/https://doi.org/10.1029/2005GL023831

Heald, C. L., & Kroll, J. (2020). The fuel of atmospheric chemistry: Toward a complete description of reactive organic carbon. *Science advances*, *6*(6), eaay8967.

Jacob, D. J., & Winner, D. A. (2009). Effect of climate change on air quality. *Atmospheric environment*, *43*(1), 51–63.

Jacobson, M. Z. (2001). Global direct radiative forcing due to multicomponent anthropogenic and natural aerosols. *Journal of Geophysical Research: Atmospheres*, *106*(D2), 1551–1568. https://doi.org/https://doi.org/10.1029/2000JD900514

Jaffe, D. A., O'Neill, S. M., Larkin, N. K., Holder, A. L., Peterson, D. L., Halofsky, J. E., & Rappold, A. G. (2020). Wildfire and prescribed burning impacts on air quality in the united states [PMID: 32240055]. *Journal of the Air & Waste Management Association*, *70*(6), 583–615. https://doi.org/10.1080/10962247.2020.1749731

Janenko, N. N. (1971). *The method of fractional steps* (Vol. 160). Springer.

Jimenez, J. L., Canagaratna, M., Donahue, N., Prevot, A., Zhang, Q., Kroll, J. H., DeCarlo, P. F., Allan, J. D., Coe, H., Ng, N., et al. (2009). Evolution of organic aerosols in the atmosphere. *Science*, *326*(5959), 1525–1529.

Keller, C. A., & Evans, M. J. (2019). Application of random forest regression to the calculation of gas-phase chemistry within the geos-chem chemistry model v10. *Geoscientific Model Development*, *12*(3), 1209–1225. https://doi.org/10.5194/gmd-12-1209-2019

Kelp, M. M., Jacob, D. J., Kutz, J. N., Marshall, J. D., & Tessum, C. W. (2020). Toward stable, general machine-learned models of the atmospheric chemical system [e2020JD032759 2020JD032759]. *Journal of Geophysical Research: Atmospheres*, *125*(23), e2020JD032759. https://doi.org/https://doi.org/10.1029/2020JD032759

Kelp, M. M., Tessum, C. W., & Marshall, J. D. (2018). Orders-of-magnitude speedup in atmospheric chemistry modeling through neural network-based emulation. *arXiv preprint arXiv:1808.03874*.

Keyes, R. W. (2001). Fundamental limits of silicon technology. *Proceedings of the IEEE*, *89*(3), 227–239.

Khomenko, S., Cirach, M., Pereira-Barboza, E., Mueller, N., Barrera-Gómez, J., Rojas-Rueda, D., de Hoogh, K., Hoek, G., & Nieuwenhuijsen, M. (2021). Premature mortality due to air pollution in european cities: A health impact assessment. *The Lancet Planetary Health*, *5*(3), e121–e134.

Kim, J., He, Y., & Park, H. (2014). Algorithms for nonnegative matrix and tensor factorizations: A unified view based on block coordinate descent framework. *Journal of Global Optimization*, *58*. https://doi.org/10.1007/s10898-013-0035-4

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kuenen, J., Visschedijk, A., Jozwicka, M., & Denier Van Der Gon, H. (2014). Tno-macc_ii emission inventory; a multi-year (2003–2009) consistent high-resolution european emission inventory for air quality modelling. *Atmospheric Chemistry and Physics*, *14*(20), 10963–10976.

Lane, T. E., Donahue, N. M., & Pandis, S. N. (2008). Effect of no x on secondary organic aerosol concentrations. *Environmental science & technology*, *42*(16), 6022–6027.

Lee, D. D., & Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature*, *401*(6755), 788–791.

Lee, K., & Carlberg, K. T. (2020). Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders. *Journal of Computational Physics*, *404*, 108973.

LeNail, A. (2020). Publication-ready nn-architecture schematics. *NN SVG*.

Leshno, M., Lin, V. Y., Pinkus, A., & Schocken, S. (1993). Multilayer feedforward networks with a non-polynomial activation function can approximate any function. *Neural Networks*, *6*(6), 861–867. https://doi.org/https://doi.org/10.1016/S0893-6080(05)80131-5

Lorenz, E. N., & Krishnamurthy, V. (1987). On the nonexistence of a slow manifold. *Journal of Atmospheric Sciences*, *44*(20), 2940–2950.

Lowe, R., & Tomlin, A. (2000). Low-dimensional manifolds and reduced chemical models for tropospheric chemistry simulations. *Atmospheric Environment*, *34*(15), 2425–2436.

Maas, U., & Pope, S. B. (1992). Simplifying chemical kinetics: Intrinsic low-dimensional manifolds in composition space. *Combustion and flame*, *88*(3-4), 239–264.

MacNamara, S., & Strang, G. (2016). Operator splitting. *Splitting methods in communication, imaging, science, and engineering* (pp. 95–114). Springer.

Manders, A. M. M., Builtjes, P. J. H., Curier, L., Denier van der Gon, H. A. C., Hendriks, C., Jonkers, S., Kranenburg, R., Kuenen, J. J. P., Segers, A. J., Timmermans, R. M. A., Visschedijk, A. J. H., Wichink Kruit, R. J., van Pul, W. A. J., Sauter, F. J., van der Swaluw, E., Swart, D. P. J., Douros, J., Eskes, H., van Meijgaard, E., … Schaap, M. (2017). Curriculum vitae of the lotos–euros (v2.0) chemistry transport model. *Geoscientific Model Development*, *10*(11), 4145–4173. https://doi.org/10.5194/gmd-10-4145-2017

Manders-Groot, A. M. M., Segers, A. J., & Jonkers, S. (2021). Lotos-euros v2.2.000 reference guide. *TNO Reports*.

Marsland, S. (2014). *Machine learning: An algorithmic perspective, second edition* (2nd). Chapman Hall/CRC.

Maulik, R., Lusch, B., & Balaprakash, P. (2021). Reduced-order modeling of advection-dominated systems with recurrent neural networks and convolutional autoencoders. *Physics of Fluids*, *33*(3), 037106.

Mircea, M., Bessagnet, B., D'Isidoro, M., Pirovano, G., Aksoyoglu, S., Ciarelli, G., Tsyro, S., Manders, A., Bieser, J., Stern, R., et al. (2019). Eurodelta iii exercise: An evaluation of air quality models' capacity to reproduce the carbonaceous aerosol. *Atmospheric Environment: X*, *2*, 100018.

National Academies of Sciences, E., Medicine et al. (2016). *The future of atmospheric chemistry research: Remembering yesterday, understanding today, anticipating tomorrow*. National Academies Press.

Nel, A. (2005). Air pollution-related illness: Effects of particles. *Science*, *308*(5723), 804–806.

Noordijk, H. (2003). Prozon en propart; statistische modellen voor smogprognose.

O'Dowd, C., Ceburnis, D., Ovadnevaite, J., Vaishya, A., Rinaldi, M., & Facchini, M. (2014). Do anthropogenic, continental or coastal aerosol sources impact on a marine aerosol signature at mace head? *Atmospheric Chemistry and Physics*, *14*(19), 10687–10704.

Paatero, P., & Tapper, U. (1994). Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, *5*(2), 111–126.

Paatero, P., Tapper, U., Aalto, P., & Kulmala, M. (1991). Matrix factorization methods for analysing diffusion battery data. *Journal of Aerosol Science*, *22*, S273–S276.

Parrish, D. D., Xu, J., Croes, B., & Shao, M. (2016). Air quality improvement in los angeles—perspectives for developing cities. *Frontiers of Environmental Science & Engineering*, *10*(5), 1–13.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Potukuchi, S., & Wexler, A. S. (1997). Predicting vapor pressures using neural networks. *Atmospheric Environment*, *31*(5), 741–753.

Robinson, A. L., Donahue, N. M., Shrivastava, M. K., Weitkamp, E. A., Sage, A. M., Grieshop, A. P., Lane, T. E., Pierce, J. R., & Pandis, S. N. (2007). Rethinking organic aerosols: Semivolatile emissions and photochemical aging. *Science*, *315*(5816), 1259–1262.

Rooney, B., Wang, Y., Jiang, J. H., Zhao, B., Zeng, Z.-C., & Seinfeld, J. H. (2020). Air quality impact of the northern california camp fire of november 2018. *Atmospheric Chemistry and Physics*, *20*(23), 14597–14616. https://doi.org/10.5194/acp-20-14597-2020

Santillana, M., Le Sager, P., Jacob, D. J., & Brenner, M. P. (2010). An adaptive reduction algorithm for efficient chemical calculations in global atmospheric chemistry models. *Atmospheric Environment*, *44*(35), 4426–4431. https://doi.org/https://doi.org/10.1016/j.atmosenv.2010.07.044

Shrivastava, M. K., Lane, T. E., Donahue, N. M., Pandis, S. N., & Robinson, A. L. (2008). Effects of gas particle partitioning and aging of primary emissions on urban and regional organic aerosol concentrations. *Journal of Geophysical Research: Atmospheres*, *113*(D18).

Silva, S. J., Ma, P.-L., Hardin, J. C., & Rothenberg, D. (2020). Physically regularized machine learning emulators of aerosol activation. *Geoscientific Model Development Discussions*, *2020*, 1–19. https://doi.org/10.5194/gmd-2020-393

Sturm, P. O., & Wexler, A. S. (2020). A mass- and energy-conserving framework for using machine learning to speed computations: A photochemistry example. *Geoscientific Model Development*, *13*(9), 4435–4442. https://doi.org/10.5194/gmd-13-4435-2020

Tsimpidi, A. P., Karydis, V. A., Zavala, M., Lei, W., Molina, L., Ulbrich, I. M., Jimenez, J. L., & Pandis, S. N. (2010). Evaluation of the volatility basis-set approach for the simulation of organic aerosol formation in the mexico city metropolitan area. *Atmospheric Chemistry and Physics*, *10*(2), 525–546. https://doi.org/10.5194/acp-10-525-2010

Turanyi, T., Tomlin, A., & Pilling, M. (1993). On the error of the quasi-steady-state approximation. *The Journal of Physical Chemistry*, *97*(1), 163–172.

van Oldenborgh, G. J., Krikken, F., Lewis, S., Leach, N. J., Lehner, F., Saunders, K. R., van Weele, M., Haustein, K., Li, S., Wallom, D., Sparrow, S., Arrighi, J., Singh, R. K., van Aalst, M. K., Philip, S. Y., Vautard, R., & Otto, F. E. L. (2021). Attribution of the australian bushfire risk to anthropogenic climate change. *Natural Hazards and Earth System Sciences*, *21*(3), 941–960. https://doi.org/10.5194/nhess-21-941-2021

Whitehouse, L., Tomlin, A., & Pilling, M. (2004). Systematic reduction of complex tropospheric chemical mechanisms, part ii: Lumping using a time-scale based approach. *Atmospheric Chemistry and Physics*, *4*(7), 2057–2081.

Williams, A. P., Abatzoglou, J. T., Gershunov, A., Guzman-Morales, J., Bishop, D. A., Balch, J. K., & Lettenmaier, D. P. (2019). Observed impacts of anthropogenic climate change on wildfire in california. *Earth's Future*, *7*(8), 892–910. https://doi.org/https://doi.org/10.1029/2019EF001210

Xu, M., Jin, J., Wang, G., Segers, A., Deng, T., & Lin, H. X. (2021). Machine learning based bias correction for numerical chemical transport models. *Atmospheric Environment*, *248*, 118022. https://doi.org/https://doi.org/10.1016/j.atmosenv.2020.118022

Zaveri, R. A., Easter, R. C., Fast, J. D., & Peters, L. K. (2008). Model for simulating aerosol interactions and chemistry (mosaic). *Journal of Geophysical Research: Atmospheres*, *113*(D13). https://doi.org/https://doi.org/10.1029/2007JD008782