Machine Learning-Based Parameterisation of Photolysis in GEOS-Chem: Version 14.2.2

Marcus Eliot Brady

Masters of Science by Research

University of York

Chemistry

September, 2024

Abstract

Photolysis schemes are an integral part of chemical transport models. However, they are computationally intensive. The chemical transport model GEOS-Chem uses the photolysis scheme Fast-JX, which takes a significant amount of time to run. Through the developments of machine learning, parameterised approaches for physical processes have become a common way of speeding up calculations. This study looks to develop a proof-of-concept approach for a machine learning-based parameterisation of the Fast-JX photolysis calculations using a collection of XGBoost models. The machine learning models were integrated into the GEOS-Chem Fortran code base and were quantitatively evaluated against the standard Fast-JX scheme. This work additionally determines the wider impact the photolysis predictions had on the GEOS-Chem simulation in regards to the calculated concentration of key components such as O_3 and NO_2.

Results show high accuracy for most species, with 103 out of 105 unique photolysis rates maintaining an R^2 greater than 0.95 throughout a six month simulation period. While the current implementation is minimally optimised, and hence computationally slower than Fast-JX, it successfully demonstrates that a machine learning parameterisation of photolysis rates is feasible in Fortran based chemical transport models and provides a foundation for future optimisations.

Acknowledgments

First and foremost, I would like to thank to thank my supervisor Prof. Mat Evans. I am additionally grateful to have made acquaintance with everyone in the Evans group: Ryan Pound, Amy Lees, Hansen Cao, Matthew Rowlinson, and Killian Murphy. All the support, ranging from learning Fortran and plotting guidance to the photos of various trips at the end of the Thursday slides, has been greatly appreciated over this year. You have all greatly helped me develop my skills as a researcher.

It would be inappropriate to not thank my family for their continued support, if I mentioned everyone, it would be a very long section of text so I will just say I have an appreciation for all. However, a very special thank you to my Grandpa and my Grandma, whose generosity led me to this masters and hence paved the way for all future research that I will take. Aside from my family there are a couple names I must mention. I must thank Lily, for your discussions, patience, and support throughout the past year not just academically but personally - I'm incredibly grateful. Also Immy and Willow, for keeping my company during my writing period.

For compute power, the Viking cluster was used during this project, which is a high performance compute facility provided by the University of York. I am grateful for computational support from the University of York, IT Services, and the Research IT team. Through everyone's support and following the completion of this thesis I can now go on to the next steps of my research career of completing my PhD over the next four years at UCL using the skills I have built throughout this year.

Declaration

I, Marcus Brady, declare that this thesis is a presentation of original work and I am the sole author. This work has not previously been presented for a degree or other qualification at this University or elsewhere. All sources are acknowledged as references in the Bibliography.

Contents

List of Tables

List of Figures

1 Introduction

1.1 Atmospheric Chemistry, Environmental Problems, and Chemical Transport Models

In the atmosphere, there are numerous complex interactions and processes, many of which are chemical. The study of these chemical processes is considered atmospheric chemistry and through understanding this field, we can address many environmental problems.

One important problem that can addressed through the understanding of atmospheric chemistry is the depletion of stratospheric ozone. This is when chlorofluorocarbons (CFCs) and other ozone-depleting substances break down in the stratosphere, most often caused by ultraviolet (UV) radiation emitted from the sun, to generate halo-atomic radicals, in a process called photodissociation. The generated radicals ($Cl^\bullet$ for example) react with ozone, which forms more radicals ($ClO^\bullet$) that can react with other atmospheric species to regenerate radicals ($Cl^\bullet$) but without releasing ozone, creating a cycle whose net result is the destruction of ozone molecules. This cycle has, over time, weakened the stratospheric ozone layer (Molina and Rowland, 1974). This stratospheric ozone depletion allows increased UV radiation to reach the surface of Earth, and in turn increases skin cancer incidence rates (Slaper et al., 1996).

Climate change and air pollution are two other issues that affect everyday life which can be understood through atmospheric chemistry. An understanding of the mechanisms behind climate change can be developed through studying the distribution and lifetimes of various greenhouse gases (like CO_2, CH_4, N_2O, and water vapour) and aerosols (like black carbon, mineral dust, and sulfate aerosols), as well as their reaction pathways, often producing secondary affects. By researching this, the various warming and cooling effects can be investigated. Similarly, air pollution can be investigated by examining the chemical reactions and photodissociations of primary pollutants (emitted directly from a source into the atmosphere) to form secondary pollutants, including tropospheric ozone. This ground-level ozone can induce respiratory inflammation at ambient levels, especially for younger people (Monks et al., 2015). Having an empirical understanding of these atmospheric processes has helped scientists to inform and guide policy makers to reduce the negative impact of these issues (IPCC et al., 2021; WHO, 2021). For example, the introduction of the Montreal Protocol in 1987 laid out guidelines for phasing out different substances, like CFCs, to reduce the rate of loss of the stratospheric ozone layer (UNEP, 1987).

One of the main tools scientists rely on to understand these atmospheric processes are Chemical Transport Models (CTMs). These are powerful computational tools that involve complex chemistry and other meteorological aspects, allowing for the simulation of the movement and distribution of chemical species both globally and locally. CTMs also allow a modular approach to development, through the collaboration of scientists of different specialities, to work on individual processes contributing to the overall model. This increases accuracy, efficiency, and credibility of the numerical modelling, giving it practical applications for example investigating the health impacts of both stratospheric ozone depletion (Eastham et al., 2018) and air pollution (Zhang et al., 2021; Vohra et al., 2021) on mortality.

Some prominent CTMs include WRF-Chem, short for Weather Research and Forecasting with Chemistry (Grell et al., 2005); MOZART-4, short for Model for OZone and Related Chemical Tracers (Emmons et al., 2010); CAMx, standing for Comprehensive Air Quality Model with Extensions (ENVIRON, 2008); and GEOS-Chem (Bey et al., 2001), which is the focus of this study. These CTMs can be used differently and have more favourability depending on the use case but they all serve as a useful tool for researchers to both analyse and understand atmospheric phenomena.

1.2 Photolysis

The photodissociation of molecules (photolysis) is a fundamental process in atmospheric chemistry. Photolysis occurs when a molecule absorbs a quantum of electromagnetic radiation, with energy $E = h\nu$, where h is Planck's constant ($6.626 \times 10^{-34}\ J\ s$) and ν is the frequency of radiation (Hz or s^{-1}). If the energy absorbed is sufficient to overcome the bond energy, chemical bonds can break, resulting in the formation of two or more products. CTMs rely on precise photolysis schemes to calculate the rates of these atmospheric processes; however, there is a computational burden with solving the high-dimensional problems they present. Most photolysis schemes solve a modified version of the 1-D plane-parallel radiative transfer equation (RTE) to obtain solar fluxes, using approximations for the scattering and absorbance by clouds and aerosols (Wild et al., 2000; Logan et al., 1981; Williams et al., 2006). Upon solving the RTE and obtaining actinic fluxes (F, the number of photons in a spectral range over all directions), the photolysis rate equation (Eq. (1)) is solved. Here, the absorption cross-section (σ) and quantum yield (ϕ) are functions of wavelength, temperature, and pressure. The cross-section defines the probability of the photons being absorbed by the molecules at each wavelength and the quantum yield is the ratio of the number of molecules that undergo photodissociation after absorbing a photon, to the total absorbance, which can vary as a function of wavelength.

$$J(\lambda) = \int_{\lambda_{\min}}^{\lambda_{\max}} \sigma(\lambda,T,P)\Phi(\lambda,T,P)F(\lambda)\,d\lambda \tag{1}$$

The computational inefficiencies caused by the high-dimensionality of the schemes leads to bottlenecks and a longer runtime within CTMs. For example, in GEOS-Chem, photolysis currently takes just under 50% of the time it takes to perform all of the transport of chemical species (GEOS-Chem, 2024). The path propagated by the beams of photons in the photolysis schemes is depicted in Figure 1. The photons are scattered, absorbed, and/or reflected in the atmosphere by (i) clouds, (ii) the surface (quantified by surface albedo), (iii) aerosols, and (iv) the ozone column. For photolysis calculations, this 3-D representation of the atmosphere is discretised into horizontal layers over grid boxes at different vertical levels, and a mix of clouds and aerosols are used. The wavelengths are also discretised into different bins (Figure 1, panel (b)).

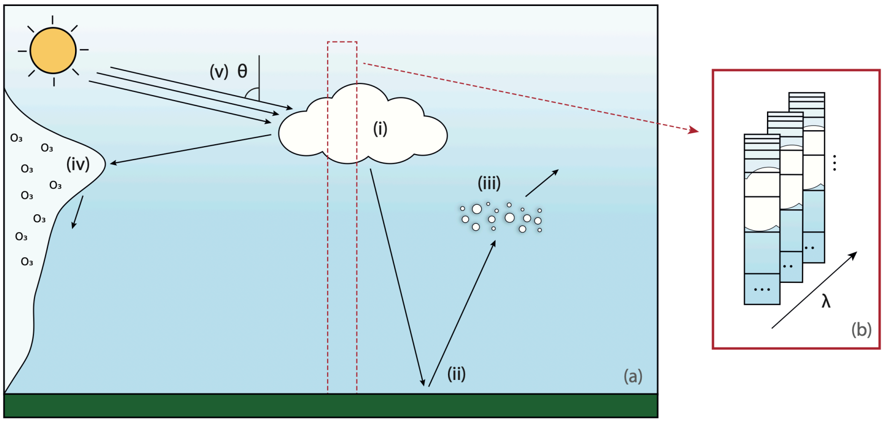

Figure 1: Schematic representation of the path of light solved by photolysis schemes through the RTEs in a 3-D representation with some common scattering, absorption and reflection pathways, panel (a). Where (i) is clouds, (ii) is the surface albedo, (iii) are aerosols and (iv) is the ozone column. Additionally, (v) is the solar zenith angle (SZA), a measure strongly dependent on the solar flux. Panel (b) denotes the vertical representation split into different wavelength bins, using the same meteorological data.

1.3 Different Photolysis Mechanisms

The photolysis scheme used in GEOS-Chem version 14.2.2 is Fast-JX v7.0 (which is discussed in technical detail later in Section 2.5), implemented by Prather (2012) and Eastham et al.

(2014), and is based on the original code for the Fast-J algorithm (Wild et al., 2000). Fast-J included seven wavelength bins in the 291 to 850 nm range, and was later extended to include an additional 11 bins covering 177 to 291 nm to improve resolution and accuracy (Bian and Prather, 2002). There are other commonly used photolysis schemes such as the Tropospheric Ultraviolet extended photolysis calculator (TUV-x) (Madronich and Flocke, 1999), or the faster version with a reduced quantity of bins (FTUV) (Tie et al., 2003). These TUV versions calculate spectral irradiance, actinic flux and photolysis rates, with 156 spectral bins for TUV-x and 17 for FTUV. Some photolysis schemes have larger resolutions for more complexity, such as the UCI Reference Model, which uses 4500 bins (Bian and Prather, 2002). The choice of photolysis scheme largely depends on the context of which it is used: considering the resolution, the computational resources available, as well as the use case.

1.4 Research Aims

In the atmospheric sciences, a host of different machine learning-based methodologies have found common use for both speeding up and parameterising processes. This study aims to leverage these advancements, with the objective being to develop a fully functional machine learning-based photolysis rate prediction system integrated into GEOS-Chem, an alternative to the existing Fast-JX scheme. More specifically, this study aims to investigate how accurately machine learning models can predict photolysis rates (for a wide range of species), how well these machine learning models can be implemented within the Fortran code of GEOS-Chem, and the overall impact the machine learning approach has on GEOS-Chem simulations. It should be noted, this project aims at first developing a proof-of-concept with minimal optimisations in regards to reducing the computational overhead, with potential optimisations discussed in Section 6.1. In this study, the machine learning alternative to Fast-JX is developed using the popular Extreme Gradient Boosting (XGBoost) software (Chen and Guestrin, 2016). This was used to train the collection of models, whose photolysis rates were stored and then fed through to the chemical integrator steps within GEOS-Chem.

The remainder of this thesis is structured as follows: Section 2 provides the theoretical background on machine learning and the photolysis applicable to this study; Section 3 describes the data used for training, as well as the model development methodology; Section 4 presents the results of the performance on the validation data; whilst Section 5 shows the performance of the predictors integrated into GEOS-Chem. Finally, Section 6 discusses possible optimisations, the limitations to the method, and potential use cases. By addressing each mentioned point, this research aims to contribute to the ongoing effort of model development and improving the efficiencies and capabilities of both chemical transport models and atmospheric modelling as a whole.

2 Theoretical Background

This section provides context and a theoretical backdrop to the machine learning approaches used in this study. We first start with the machine learning fundamentals by briefly describing the history and evolution of machine learning as well as some key concepts (Section 2.1). Some of the key algorithms found within machine learning are discussed (Section 2.2), including: linear/multiple regression, decision trees, ensemble methods, and neural networks. For the parameterisation task itself, two methods are considered in detail and the rationale of why one is picked over the other is discussed (Section 2.3). To provide an insight into the machine learning commonly used in atmospheric sciences and why this project is important, an overview is provided in Section 2.4. The original photolysis scheme used in GEOS-Chem (Fast-JX) is described in more detail as well as a description of the actual GEOS-Chem CTM used in this project (Sections 2.5 and 2.6 respectively).

2.1 Machine Learning Foundations

Machine learning, a term first coined in the 1950s by IBM researcher Arthur Samuel (1959), is a subset within the field of artificial intelligence (AI). The formal establishment of AI as a field occurred during a conference known as the Dartmouth Summer Research Project on Artificial Intelligence in 1956 (McCarthy et al., 2006), where the concept of AI was discussed in terms of creating systems capable of ‘intelligent behaviour’. Machine learning, more specifically, is referred to as the study and development of algorithms that enable computers to learn from data patterns and make decisions based on the patterns, rather than being explicitly pre-programmed.

Machine learning tasks can be divided into two main categories: regression or classification. Regression is based on numerical predictions of continuous values such as photolysis rates, bond lengths, and energies. Classification, on the other hand, is used to classify predictions into discrete values, such as binary outcomes (true or false), to recognise handwritten digits, or to categorise images into pre-defined groups. Both of these tasks are powerful when correctly used.

Machine learning for a problem generally follows this pathway: deciding an algorithm appropriate to the task, splitting a dataset into training and validation sections, training the model, and evaluating the performance of the models. A model is trained by learning the relationship between the input and output data, from the training data. The model iteratively improves by minimising the difference between the real and predicted values, achieved by changing the parameters of the chosen algorithm.

Machine learning algorithms generally fall into three main categories: supervised, unsupervised, and reinforcement learning. Supervised learning models rely on being trained on a dataset where a set of input variables, also know as features, $(x_1, x_2, \ldots, x_n)$ can be mapped to one or more output variables $(y_1, y_2, \ldots, y_n)$. This learned relationship is then used to predict outputs for new, unseen inputs. Both regression and classification problems tend to fall in this category. Unsupervised learning is used when there are no labelled target variables and instead the model tries to identifying groupings and relationships within the data. This is commonly used in clustering or dimensionality reduction tasks. Reinforcement learning is commonly used when a model makes decisions and is either rewarded or penalised depending on that decision, creating a feedback loop. In this study, we consider only supervised learning methods and algorithms as the task (parameterising photolysis rates) is a regression problem.

2.2 Key Machine Learning Algorithms

Predating the actual conception of machine learning and AI, the mathematics forming the foundations of the field is considered to be dated well before the 20th century. The concept of regression was first introduced by Galton (1886), focusing on the phenomenon of observations regressing to the mean. This phenomenon describes how extreme measurements in a dataset tend to be closer to the mean in the following, subsequent observations. This work built on the crucial component of the method of least squares, an optimisation technique to find the best-fitting curve/line, first independently developed at the start of the 19th century by both Legendre and Gauss, as per Stigler (1981). These developments laid the groundwork for modern regression analysis which, while not originally a learning algorithm, can derive conclusions from data and lead to predictions, making it a fundamental concept in the modern machine learning landscape. Equation (2) is the basic multiple linear regression model where $\hat{y}$ is the predicted dependent variable, β_0 is the intercept, β are the coefficients/weights, and x are the independent variables. The coefficients (β) are typically estimated through the method of the least squares (Nievergelt, 1994).

$$\hat{y}_i = \beta_0 + \beta_1 x_{1,i} + \beta_2 x_{2,i} + \cdots + \beta_n x_{n,i} \tag{2}$$

Following these early developments, several key algorithms have emerged that shape the field of machine learning. We now discuss the chronological progression of some of the most important ones. One significant advancement was the concept of decision trees. Whilst initially introduced by Morgan and Sonquist in 1963, decision trees emerged as a powerful machine learning algorithm in 1986 with the development of the ID3 algorithm by Quinlan (1986), which could construct trees iteratively. Decision trees work for both regression and classification tasks by making a series of decisions based on the input data. The tree structure consists of what are called nodes: internal nodes are where decisions are made based on the feature values from the input data, and leaf nodes can be seen as the end of the branch and is where the final prediction is made. An example decision tree can be found in Figure 2, used to classify a given atmospheric condition into four different classes of photolytic intensity. The same structure could be used for regression tasks, by training it to predict numerical values instead of classes.

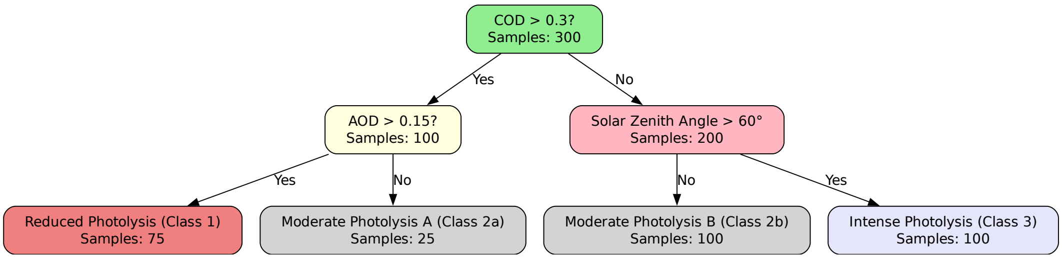

Figure 2: An example decision tree generated in Python for the classification of atmospheric conditions into three different photolysis categories: reduced, moderate, and intense photolysis. The tree is based on key factors: cloud optical depth (COD), aerosol optical depth (AOD), and SZA.

It is rare for a single decision tree to be used in modern problems. They are considered weak predictors and instead a collection of them are used, whose outputs are aggregated to form a final prediction. This approach is known as an ensemble method. The general form of the ensemble method is expressed in Eq. (3), where: $\hat{y}_i$ is the predicted output for the i-th instance, $f_k(x_i)$ is the prediction of the k-th model for the i-th instance, w_k is the weighting of the respective k-th model towards the overall prediction, and M is the total number of models in the ensemble.

$$\hat{y}_i = \sum_{k=1}^{M} w_k f_k(x_i) \tag{3}$$

Both Random Forests (RF) and Gradient Boosting (GB) are examples of this ensemble method, but are used in different ways. These processes are the most relevant to this project and are discussed in more detail in the next section.

A different approach, neural networks (NNs) were first conceptualised by McCulloch and Pitts (1943), stating that the biological neural networks found in living organisms could be replicated through computation. Their conceptual neuron calculated a weighted sum between the input and a fixed weight, similar notation to that in Eq. (3). If that weighted sum exceeded a threshold value (often denoted θ) then the neuron would output 1, and 'fire' like a biological neuron. Otherwise it would return 0. This allowed for the simulation of binary operations just like logic gates. The problem with this concept was that the weights were fixed and had no way to improve, or 'learn'. Building on this original model, the perceptron was introduced by Rosenblatt (1958). This was essentially the same structure as the neuron but with the ability to learn by changing the weights instead of using fixed weights. This was based on the difference (residual) between the output ($\hat{y}$) and the true value (y) as well as the learning rate (denoted η). The learning rate is essentially how much the model weights are adjusted in response to the residual. The learning rule for updating a weight from its often less accurate state ($w_{i,\text{old}}$) to a new state ($w_{i,\text{new}}$), in relation to the respective input feature (x_i) is shown in Eq. (4).

$$w_{i,\text{new}} = w_{i,\text{old}} + \eta \cdot (y - \hat{y}) \cdot x_i \tag{4}$$

Whilst perceptrons were good for classification and linear problems, they had a limitation in capturing complex, non-linear relationships in data. This led researchers to develop a system with multiple layers of perceptrons. These systems had 'hidden' layers of perceptrons, these layers were considered hidden when they were sandwiched between input and output perceptrons, and created what is called the multi-layer perceptron (MLP). However, the earlier MLPs faced the issue of inefficient training and a lack of proper methodology to update the weights in the layers, particularly the hidden layers whose outputs were unknown. This severely limited the application of MLPs and in turn slowed the progress of this field for a few years.

Neural networks really took off in the 1980s following the popularisation of the backpropagation algorithm (Rumelhart et al., 1986), which allowed for the effective training of these complex MLPs by efficiently computing gradients to adjust the internal weights of the perceptrons. This formed the foundations for the neural networks that are widely used today, with the ability to learn complex, non-linear relationships in data. The technicalities of NNs are not discussed further but a more in-depth understanding of their workings can be found in the initial paper on neurons, perceptrons, and backpropagation included in the Bibliography (McCulloch and Pitts, 1943; Rosenblatt, 1958; Rumelhart et al., 1986). Work on NNs has led to the creation of the "deep learning" field, used to study and compute complex interactions and patterns, from natural language processing (NLP) for text analysis (Collobert et al., 2011), to geospatial data predictions (Reichstein et al., 2019), and to computer vision (Krizhevsky et al., 2012), which enables machines to interpret and make decisions based on visual data.

2.3 XGBoost Algorithm over Random Forest Regression

This project considered two main algorithms: Random Forests (RF) and Gradient Boosting (GB) methods. This section discusses the differences between these two approaches.

The RF method was properly introduced by Breiman (2001). It built on the initial random decision forests algorithm, used for classification (Ho, 1995), with a whole host of novel features to construct a ‘forest’ of decision trees to perform both classification and regression analysis. It used the concept of bagging (also developed by Breiman (1996)) which allowed each different tree to be trained on a random, different subset of the total data. Another important feature introduced was random feature selection. This novel concept involves considering a random subset of features (from: $x_1, x_2, \ldots, x_n$) at each decision point in the tree. This approach further reduces correlation between each tree, allowing a more representative, fair prediction. An equation for RF regression (RFR) can be found in Eq. (5).

$$\hat{y}_i = \frac{1}{T} \sum_{k=1}^{T} f_k(x_i) \tag{5}$$

This is a variation of the ensemble model equation, in Eq. (3), where the final prediction ($\hat{y}_i$) is the average of the predictions ($f_k(x_i)$) from each of the T trees. Where the weight here is $1/T$ for all k trees and hence all trees contribute equally. A schematic of this can be found in the bottom half of Figure 3.

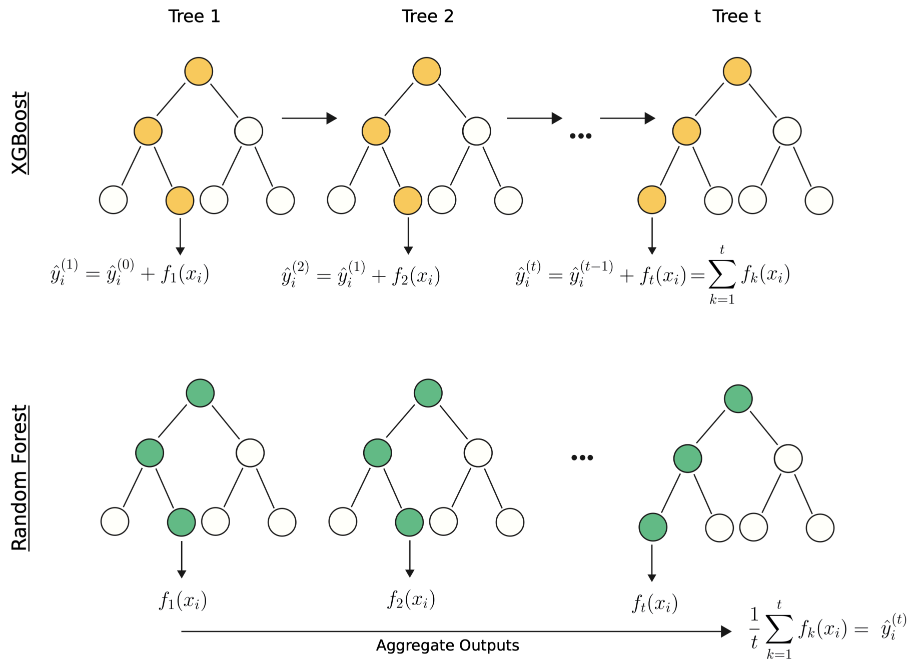
Tree 1
Tree 2
Tree t
XGBoost
$\hat{y}_i^{(1)} = \hat{y}_i^{(0)} + f_1(x_i)$
$\hat{y}_i^{(2)} = \hat{y}_i^{(1)} + f_2(x_i)$
$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + f_t(x_i) = \sum_{k=1}^{t} f_k(x_i)$
Random Forest
$f_1(x_i)$
$f_2(x_i)$
$f_t(x_i)$
Aggregate Outputs
$\frac{1}{t}\sum_{k=1}^{t} f_k(x_i) = \hat{y}_i^{(t)}$

Figure 3: Comparative overview between XGBoost sequential boosting prediction method (top) and the random forest prediction method (bottom). Included are visual representations of the XGBoost and random forest final prediction equations, Eq. (6) and Eq. (5) respectively.

Alternatively, we now consider the gradient boosting methodology, proposed by Friedman (2001). Here, an ensemble of models (usually decision trees) is created by building a predictor after another predictor and so on, in a sequential fashion. The theoretical foundations for GB led to the creation of the popular XGBoost software that is used in this study (Chen and Guestrin, 2016). XGBoost builds predictors sequentially (one after another): the initial prediction is often just the average of all the target values in the dataset, the subsequent models are trained to predict the residual from the previous models predictions. By iteratively summing these model outputs, the overall prediction improves. This is repeated until a stopping criterion is met, for example a maximum number of iterations is reached or there is minimal improvement in the error. The XGBoost methodology is displayed in the top section of Figure 3, where the output of each tree is a function ($f_k(x_i)$) of the i^{th} instance. The sum of the initial prediction ($y_i^{(0)}$) and the first output ($f_1(x_i)$) results in the first prediction ($y_i^{(1)}$), this prediction is then summed with the next models output ($f_2(x_i)$) to get the next prediction ($y_i^{(2)}$). This process is iteratively repeated, with each model's output added to the cumulative prediction, as described in Eq. (6).

$$\hat{y}_i^{(t)} = \sum_{k=1}^{t} f_k(x_i) = y_i^{(t-1)} + f_t(x_i) \tag{6}$$

Both of these algorithms are relatively quick due to their effective use of parallelism, albeit through different approaches. RFR can build each tree in parallel, as they are independent of each other. This allows for the simple distribution of the training process on multiple cores. The approach XGBoost takes is slightly more nuanced due to the sequential nature of the model building. The parallelism occurs within the construction of each individual tree: parallel processing of the features, finding points to split features within tree nodes in parallel, and then parallel construction of each node within the tree. This allows for XGBoost to maintain the benefits of the sequential gradient boosting methodology whilst having the speed of parallelism.

The incremental improvement of XGBoost's predictions is appealing because it adaptively refines and reduces the error, making it suitable for dealing with the large variability often found within photolysis rates and hence is the main factor as to why this algorithm was chosen. Additionally, XGBoost incorporates regularisation techniques (described later in Section 3.4); these help to prevent overfitting. Overfitting occurs when the models learn the patterns of the training data too precisely and cannot generalise well out of that context. Another significant reason why XGBoost was used was the C++ backbone, leading to the ease of implementation within the Fortran source code of GEOS-Chem. This implementation is discussed later in Section 3.5.

2.4 Machine Learning in Atmospheric Chemistry

Given the evolution of these machine learning algorithms, there has been a surge in applied use across all sciences and in other fields. More specifically in atmospheric studies, a review found a 500% increase in the use of machine learning techniques (Zheng et al., 2021). Machine learning has been used for a range of applications in atmospheric sciences, particularly in atmospheric models, for example large-scale technology companies like Microsoft, Google, and NVIDIA have created global deep learning models with the intent of providing rapid and accurate forecasting (Bodnar et al., 2024; Lam et al., 2023; Pathak et al., 2022). While these serve the intended purpose of speedy predictions, they often lack interpretability, potentially obscuring the underlying scientific processes. Instead, alternative methods focus on parameterising various processes ranging from convection to discovering equations for cloud cover, maintaining a connection to the scientific fundamentals (Brenowitz et al., 2020; Grundner et al., 2024).

This is the approach this project takes.

Machine learning has also been used directly in conjunction with GEOS-Chem and we now consider some of these projects, starting with one that used Random Forest Regression (RFR) to simulate gas-phase chemistry (Keller and Evans, 2019). This was one of the first projects involving machine learning in GEOS-Chem and was a direct inspiration for this project. Keller and Evans (2019) successfully created 51 separate random forest predictors for each chemical species that underwent transport in GEOS-Chem. Whilst their implementation was slower than the default, non-machine learning parameterised gas-phase chemistry, due to being minimally optimised, it was accurate for key species and provided a proof-of-concept for future developments. A second study used unsupervised clustering algorithms to reduce computational overhead of the integration of kinetics (Shen et al., 2022). Another study, instead of revolving around an integration into GEOS-Chem, used the outputs in conjunction with an ensemble of prediction methods for air quality forecasting (Fang et al., 2023). Further machine learning projects involving GEOS-Chem can be found in the GEOS-Chem Documentation (2023). While these studies demonstrate the potential and wide applicability of machine learning, certain areas remain unexplored. For example there remains a lack of machine learning-based parameterisation for photolysis calculations. Existing methods mimic the outputs of photo-chemical processes in the form of concentrations (Xing et al., 2022), or directly predict photolysis rates for a small selection of species (without implementation in a CTM) (Pan et al., 2025), yet none act as a suitable replacement to a photolysis scheme like Fast-JX. This is what makes this project important, as it provides the foundations for a machine learning parameterised approach with photolysis rate calculations, which can be a direct replacement for those schemes within Fortran based CTMs.

2.5 Original Photolysis Scheme in GEOS-Chem

The photolysis scheme parameterised here through machine learning is Fast-JX (Section 1.3). Fast-JX accounts for the atmospheric radiative processes by incorporating absorption and scattering from gases, aerosols, and clouds, including both Rayleigh and Mie scattering. It does this by using a multi-stream radiative transfer solver, with eight streams. Each stream represents a ‘direction’ of radiation, where the simplest would be a two-stream approximation with two directions: up and down. The eight streams used in Fast-JX account for different angular directions that light travels with respect to the vertical, this allows for the multiple scattering/absorption scenarios. To accurately account for the vertical aspect of the photolysis calculations, Fast-JX takes the 3-D nature of the atmosphere and transforms it into a discretised collection of horizontal planes. This is called the plane-parallel approximation and makes the calculations tractable. The RTEs are solved using the plane-parallel representation of the atmosphere to quantify actinic flux (F), the flux of photons, at each level in a particular wavelength range. In general, photolysis schemes approach Eq. (1) and discretise it over (k) specific wavelength bins to simplify the calculation, resulting in Eq. (7).

$$J(s^{-1}) = \sum_{i=1}^{k} F(\lambda_i)\sigma(\lambda_i,T,P)\phi(\lambda_i,T,P) \tag{7}$$

The terms in Eq. (7) are the same as those in the continuous equation (Eq. (1)). In the specific case of Fast-JX, it specifically calculates the wavelength bin-resolved fluxes (those seen in Eq. (7)) and multiplies them by the effective cross-section for each bin. For simplicity, the cross-sections used in Fast-JX have the quantum yields incorporated (as a product of the two). Additionally, the temperature and pressure dependence seen in Eq. (7) is accounted for through interpolation of the cross-sections. The wavelength range covered by Fast-JX in GEOS-Chem is 177-850 nm, through 18 wavelength bins. This fewer number of bins, compared to the UCI Reference Model (Bian and Prather, 2002), leads to faster computation, hence the ‘fast’ in Fast-JX. The photolysis rates calculated by Fast-J are stated to have worst-case errors of no more than 10% under various atmospheric conditions (Wild et al., 2000).

2.6 Model Description (GEOS-Chem)

The CTM used in this project is GEOS-Chem Classic version 14.2.2 (GEOS-Chem Community, 2023, `https://doi.org/10.5281/zenodo.10034814`). It is an open source project to facilitate research into atmospheric phenomena and chemistry. GEOS-Chem Classic was configured with a spatial resolution of $4^\circ \times 5^\circ$ and a vertical hybrid-sigma pressure coordinate system using 72 vertical levels. The internal time steps were 40 minutes and 20 minutes for the chemical and transport time steps, respectively. The Modern-Era Retrospective analysis for Research and Applications, Version 2 (MERRA 2) met field was used. MERRA2 is a NASA atmospheric reanalysis tool to provide high resolution meteorological data (Gelaro et al., 2017).

The model chemistry was initially described by Bey et al. (2001) with various additions since. It includes tropospheric chemistry of ozone, NO_x, hydrocarbons, with various updates to halogen chemistry (Parrella et al., 2012; Bates et al., 2021; Eastham et al., 2014). The photochemistry uses a customised version of the Fast-JX code based on Fast-J (Wild et al., 2000), as detailed in the previous section. This scheme calculates photolysis rates for 161 different species, including a mixture of inorganic, nitrogenous, and organic compounds. The primary aim of this study was to emulate and replace the calculation of these photolysis rates, allowing for a seamless transfer of predicted rates to the chemical kinetics calculation, described and solved by the Kinetic PreProcessor (KPP) Rosenbrock solver (Lin et al., 2023).

3 Data and Methodology

3.1 Training and Evaluation Data

This work was performed using the Viking 2 high-performance computing (HPC) cluster, which provided high quantities of memory, CPU, and GPU power. Additionally, the scripts used to prepare the datasets, train the models, including each model instance can be found in the referenced Zenodo repository: (Brady, 2024, `https://doi.org/10.5281/zenodo.12705193`)

For this project, a dataset was produced from the outputs of the GEOS-Chem model, with data ranging from the start of July 2019 to the end of June 2020. The data was collected for every grid box (a spatial setup of $72 \times 46 \times 72$) and was sampled every five hours. The choice of sampling every five hours was made to manage memory use efficiently. It also allowed for the data sample to span all possible hours in a day throughout the month, given 24 hours divided by 5 hours does not give an integer. Despite all data collection and training taking place on the Viking 2 HPC, with large amounts of memory available, using data sampled from every one, two, or four hours drastically increased the time the workflow would take, making it inefficient for model development. This was at the cost of the potential complex interactions missed by reducing the frequency of sampling to every five hours. The primary aim of the dataset was to contain enough atmospheric context to allow the machine learning models to effectively predict in any given scenario. The ‘StateMet’ (meteorological quantities), ‘JValues’ (photolysis rates), and ‘Aerosols’ (aerosol quantities) were the diagnostics enabled for collection in GEOS-Chem for this run. The data was formatted in the Network Common Data Form version 4 (NetCDF4, ‘.nc4’) file format, a common file format in atmospheric sciences for large, multidimensional datasets.

In the training dataset, the collection of variables was designed to match the input data used in the Fast-JX Fortran subroutines, as these were all known to have a clear impact on photolysis rates. Through iterations of model development, features to keep or remove were identified. Physical variables such as temperature, UV surface albedo, overhead ozone column, solar zenith angle (SZA), and the cosine of the SZA were some of the key variables. These variables would directly affect the photolysis rates in different ways. Temperature influences the distribution of energy in molecules and, indirectly, the distribution of aerosols/gases through atmospheric circulation, caused by changes in temperature. Surface albedo indicates how much solar radiation is reflected, whilst the overhead ozone column causes scattering/absorption of photons. The inclusion of SZA allowed for both positional and temporal encoding, as its values are largely dependent on both the seasonality and location. The cosine of the SZA, which is directly related to the SZA, is also a measure of solar flux, as the atmospheric path photons travel is equal to the atmospheric height divided by the cosine of the SZA. SZA also allowed a practical optimisation. When a grid box fulfilled the condition of a SZA greater than 98°, there would be minimal photolytic activity due to an absence of solar flux. The training dataset was filtered with this condition. This was done to remove excess zero values from influencing the training process. An additional two out of seven possible dust aerosol optical depth bins (bins 1 and 7) were used in the input vector. It was important to have aerosol representation due to the scattering and absorption they cause (Liao et al., 1999).

In the final iteration, there were 19 total variables that made up the input vector for the machine learning models: 13 directly from the CTM output data and six calculated from that data (see Table 1 for the collection of variables). The input variables were classified into either 2-D or 3-D variables. Variables were considered 2-D if they had one value for the entire vertical column and were 2-D functions of longitude and latitude, for example surface albedo. 5 of the 19 variables were considered 2-D. These values were broadcasted over the vertical column so the lowest and highest vertical level would have the same value. The remaining 14 variables were classified as 3-D variables, where each grid box had a different value for each level, longitude, and latitude combination.

For some of the 3-D type variables, a mechanism was implemented to provide the machine learning models with vertical context surrounding each instantaneous point. This was done by calculating the cumulative summations above and below the points in the vertical column to get two new variables for each original variable. Three specific 3-D variables underwent this calculation: the water cloud optical depth, ice cloud optical depth, and the 3-D cloud fraction. The six calculated variables seen in the lower part of Table 1. These variables emulated some level of awareness similar to that of what the beam of photons would encounter in terms of cloud behaviour in photolysis schemes. For example, doing this for the water cloud optical depth would represent the amount of water cloud optical depth above or below each point. While this mechanism was beneficial, it was not applied to all 3-D variables.

The aerosol optical depth bins were considered, but this would have led to an additional four variables, hence slowing down the predicting time without substantive increase in the performance. When deciding which variables to keep, it was assumed the impact of clouds would be more significant than aerosols (Lefer et al., 2003), potentially at the cost of a reduction in performance over more aerosol polluted areas like China and North Africa (Li et al., 2022). It must be acknowledged that the exclusion of these aerosols, as well as other aerosol variables such as PM2.5 concentrations, could hinder the performance of some of the machine learning models dictated by aerosol scatterings.

Table 1: Overview of variables that made up the input vector for the machine learning model. The variable symbol, name and unit are included alongside whether they are a 2-D or 3-D quantity. Variable type is included, where collected means a variable from GEOS-Chem output and calculated is using collected data undergoing feature engineering.

Type	Variable Symbol	Variable Name	2-D/3-D Variables	Unit
Collected	CLDF	3-D Cloud Fraction	3-D	-
	SZA	Solar Zenith Angle	2-D	Degree (°)
	SUNCOS	Cosine of SZA	2-D	-
	UVALBEDO	UV Surface Albedo	2-D	-
	lev	Hybrid Level	2-D	level
	TO3	Overhead Ozone Column	2-D	Dobsons (DU)
	T	Temperature	3-D	Kelvin (K)
	PMID	Pressure	3-D	Hectopascal (hPa)
	AIRDEN	Air Density	3-D	kg m^{-3}
	TAUCLI	Ice Cloud Optical Depth	3-D	-
	TAUCLW	Water Cloud Optical Depth	3-D	-
	AODDust1000nm_bin1	Aerosol Optical Depth Bin 1	3-D	-
	AODDust1000nm_bin7	Aerosol Optical Depth Bin 7	3-D	-
Calculated	TAUCLW_above	Water Cloud Optical Depth Above	3-D	-
	TAUCLW_below	Water Cloud Optical Depth Below	3-D	-
	TAUCLI_above	Ice Cloud Optical Depth Above	3-D	-
	TAUCLI_below	Iced Cloud Optical Depth Below	3-D	-
	CLDF_above	3-D Cloud Fraction Above	3-D	-
	CLDF_below	3-D Cloud Fraction Below	3-D	-

3.2 Different Photolysis Rate Machine Learning Models

A model could be made for each photolysis rate in the model. However this would be inefficient. In GEOS-Chem some species had more than one photodissociation pathway, known as channels. This was either implemented through a branching ratio or through the use of different wavelength dependent cross-sections. For the latter, each separate channel was treated as an individual target variable, temporarily added to the 'JValues' diagnostics as its own rate, so a machine learning model could be trained on it. For instance, GEOS-Chem uses two channels for the photolysis of formaldehyde: one producing hydrogen (H) and formyl (CHO) radicals (reaction R1). The other produces carbon monoxide (CO) and diatomic hydrogen (H_2) (reaction R2). This approach allows for the conservation of the complex photochemistry resulting from the separate channels, an important aspect of a photolysis scheme. The total photolysis rate for a species is the sum of the rates of the individual channels, resulting in the quantities found in the GEOS-Chem outputs. This was done for five separate species: formaldehyde (2 channels/models), chlorine nitrate (2 channels/models), acetaldehyde (2 channels/models), glyoxal (3 channels/models), and acetone (2 channels/models). In total these five species account for 11 distinct channels, and hence an additional 11 models.

$$\mathrm{CH_2O + h\nu \rightarrow H + HCO} \tag{R1}$$

$$\mathrm{CH_2O + h\nu \rightarrow H_2 + CO} \tag{R2}$$

When the channels were accounted for by a branching ratio, one model was used for the species and the output was simply multiplied by the different multipliers. This resulted in separate photolysis rates from one machine learning model. For example NO_3 was split via two channels using a ratio of 0.886 to 0.114 (as described in Fast-JX).

When a cross-section was the same for different species, it meant the species were similar enough to have the same values for the photolysis rates calculated by Fast-JX in GEOS-Chem. For the species this was applicable to, a single model could be reused. Table 2 contains the models used more than once. For example in GEOS-Chem, for organic peroxides, the methylhydroperoxide (MP) cross-section is re-used thus there is no need to generate machine learning models for each species as the same one can be reused. Similarly, monoterpene-derived nitrates (MONITS) and hydroperoxethanal (HPETHNL) were used for different biogenically-derived species. When factoring in the re-usability of models and ones needed for separate channels, the 161 photolysis rates used 111 different machine learning models to account for 105 unique photolysis rates. This includes the five species (formaldehyde, chlorine nitrate, acetaldehyde, glyoxal, and acetone) whose photolysis rates are each determined by multiple channels, requiring a total of 11 separate machine learning models to account for all the individual channels.

Table 2: Summary of machine learning models used more than once.

Model Name	Species Description	Frequency of Models
MP	Methylhydroperoxide	22
MONITS	Monoterpene-Derived Nitrates	9
HPETHNL	Hydroperoxyethanal	7
RCHO	> C2 Aldehydes	4
MVK	Methyl Vinyl Ketone	3
NO3	Nitrate Radical	2
HNO4	Pernitric Acid	2
BrNO3	Bromine Nitrate	2
H1211	Halon 1211	2
R4N2	> C3 Alkylnitrates	2
MACR	Methacrolein	2
MPN	Methyl Peroxy Nitrate	2
MCRHNB	A Hydroxynitrate from MACR	2
NITs	Sea-salt Particulate Nitrate	2
NIT	Nitrate	2

3.3 Model Training and Parameters

The machine learning models were trained in Python using XGBoost (Section 2.3), (Chen and Guestrin, 2016). The data was split into training and validation datasets from the start of July 2020 to the end of June 2021. The first 75% of data (from the start of July 2020 to the end of March 2021) was allocated for the training, with the remaining 25% (from the start of April 2021 onwards) saved for validation (the results for which can be found later in Section 4). This split ensured the models could learn from a substantial quantity of data whilst also fairly evaluating the ability to generalise. To optimise the speed of the training process, the machine learning models were trained on NVIDIA H100 GPUs, on Viking 2. Hyperparameters that defined the models were optimised largely through trial and error. The primary focus was on the learning rate and maximum depth, as their interactions most often increase the performance in gradient boosting models (Zuo et al., 2019). Other hyperparameters were tuned, but with minimal sensitivity to the reduction in error. All hyperparameters are shown in Table 3.

An analysis or optimisation of the hyperparameters, through common hyperparameter optimisation techniques like Optuna or Bayesian Optimisation (Akiba et al., 2019; Wu et al., 2019), would undoubtedly increase performance. However, performing this for each species was considered too time consuming for this initial study.

Table 3: Tabulated description of the hyperparameters used for all XGBoost photolysis predictors.

Hyperparameter	Value	Hyperparameter Info.
eta (η)	0.005	Learning rate
max_depth	12	Maximum depth of a tree
colsample_bytree	0.8	Ratio of training data
alpha (α)	1	L1 regularization term
gamma (γ)	1	Minimum loss reduction for partition
lambda (λ)	1	L2 regularization term
num_rounds	2000	Number of training rounds

3.4 Objective Function and Regularisation

In XGBoost, the model is minimised through the objective function. This is the sum of the loss function and the regularisation terms. Equation (8) shows this operation, where ‘Obj’ is the objective function, $L(\theta)$ is the training loss function, and $\Omega(\theta)$ is the regularisation term. The regularisation term controls the model complexity to prevent overfitting.

$$\text{Obj} = L(\theta) + \Omega(\theta) \tag{8}$$

$$\Omega(\theta) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} (w_j)^2 + \alpha \sum_{j=1}^{T} |w_j| \tag{9}$$

The regularisation terms in XGBoost can be seen in Eq. (9). This equation represents the summation of three components: the tree complexity (γT), the L2 regularisation term

($\frac{1}{2}\lambda\sum_{j=1}^{T}(w_j)^2$), and the L1 regularisation term ($\alpha\sum_{j=1}^{T}|w_j|$). Here T is the number of leaves in the tree, w_j are the leaf weights, and α, λ, and γ are the regularisation parameters seen in Table 3. In the context of XGBoost, the weights (w_j) are the contribution to the final prediction from each sequentially constructed tree in the ensemble. For example, if an input sample traverses the tree and ends up in a leaf with a weight $w_j = 0.5$, then that tree contributes 0.5 to the final prediction. This is represented as $f_k(x_i)$, in relation to Eq. (6), where k is the index of the particular tree for the x_i input sample and, as described in Section 2.3, the final prediction is the sum of the output from all trees. The regularisation terms penalise larger leaf weights (through L1 and L2) and complex tree structures (through γT).

There are a number of different loss functions that can be used for the training. Each one is dependent on the nature of the data and task at hand (Naser and Alavi, 2021), with some of these native to the XGBoost API. Due to the task simply being regression, with a large dataset for training, the XGBoost models minimised the Mean Squared Error (MSE), shown in Eq. (10).

$$\mathrm{MSE}(y,\hat{y}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2. \tag{10}$$

The MSE computes the average of the squared differences between the actual (y_i) and the predicted ($\hat{y}_i$) values for n samples. To avoid issues with the logarithmic spread of photolysis rate (spanning multiple orders of magnitudes), the data was log transformed. The addition of the smallest non-zero constant (denoted δ), for each different photolysis rate from a sample of the data. This was to prevent issues with the log transformation of zero values. The constant was usually a magnitude smaller than $10^{-10}\ s^{-1}$, depending on the species and was saved for each photolysis rate for later use, transforming the runtime predictions in GEOS-Chem back to a normal scale. The general forward log transformation for the true values from the linear to log domain, for the i-th instance and the j-th species, using the respective constant (δ_j) is shown in Eq. (11).

$$y_{\mathrm{log},i,j} = \ln(y_{\mathrm{linear},i,j} + \delta_j) \tag{11}$$

3.5 Implementation

One of the largest challenges of this parameterisation was the need to integrate the machine learning models back into GEOS-Chem once trained. Most tasks in this domain are commonly used in the context of Python or C, with much support and documentation. GEOS-Chem however is built in Fortran. Luckily, more people are finding a need to have machine learning support in Fortran and there are numerous libraries now supported. They generally use pre-existing libraries and define a C wrapper. Some examples include FTorch (ICCS Contributors, 2024), a Fortran library to enable the use of PyTorch models, and Fortran-Keras Bridge which is similar to FTorch but for the Keras models (Ott et al., 2020). Both of these enable popular deep learning libraries to be used within Fortran. In this study, since XGBoost was used, its C++ backbone could be manipulated to interact with Fortran through an XGBoost Fortran API, developed by Keller and Manyin (2022). This used a Fortran module called ‘Iso C Binding’, making the XGBoost functions in C callable in GEOS-Chem’s Fortran code. To enable XGBoost to work with the API, XGBoost (version 1.6.0) was pointed to within a Conda environment when compiling GEOS-Chem. The simple version of the code used to implement the prediction process within GEOS-Chem is included in Appendix 8.1.

The data necessary for the input vector was calculated, collected and then passed through to the native data structure of XGBoost (a DMatrix) for all grid boxes. Any grid boxes that were considered dark (with a SZA greater than 98°) were skipped. On the first GEOS-Chem time step, the machine learning models were loaded and initialised. A simple implementation of OpenMP, Open Multi-Processing (Dagum and Menon, 1998), was used to parallelise the predicting process on the CPUs; it enabled different photolysis models to work on different threads of the CPUs.

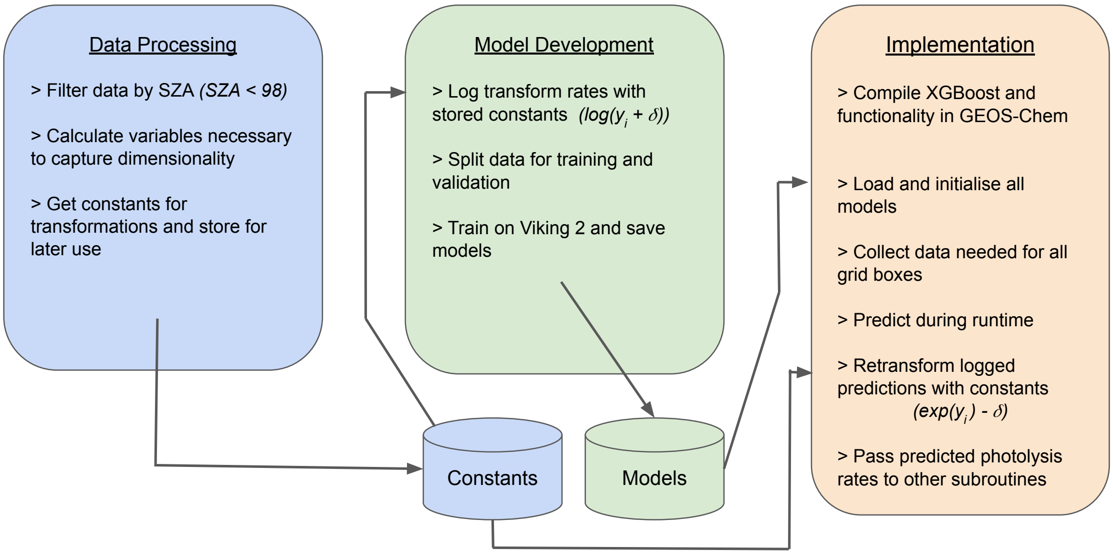
Data Processing
> Filter data by SZA (SZA < 98)
> Calculate variables necessary to capture dimensionality
> Get constants for transformations and store for later use
Model Development
> Log transform rates with stored constants (log(y_i + δ))
> Split data for training and validation
> Train on Viking 2 and save models
Implementation
> Compile XGBoost and functionality in GEOS-Chem
> Load and initialise all models
> Collect data needed for all grid boxes
> Predict during runtime
> Retransform logged predictions with constants (exp(y_i) - δ)
> Pass predicted photolysis rates to other subroutines
Constants
Models

Figure 4: Schematic representation of the model development process from the data processing through to implementation. The constants derived from the data processing step and the models trained from the model development step, were both stored and used within the implementation in GEOS-Chem.

As part of the training process, the photolysis rates were log transformed. Outputs from the machine learning models during runtime needed to be transformed back into linear space by applying the exponential function, and the respective constant (δ_j) subtracted (Eq. (12)). This was done during runtime and the linear photolysis rates were stored in an array containing all photolysis rates for all longitude, latitude, and level combinations and was passed to the KPP module. This was repeated each time step. A schematic showing both the model development and predicting is shown in Figure 4.

$$\hat{y}_{\text{linear},i,j} = \exp(\hat{y}_{\text{log},i,j}) - \delta_j \tag{12}$$

When ran, the machine learning models would occasionally predict negative photolysis rates. This happened when the transformed prediction was smaller than the constant used during transformation (from log space to linear space). In this case, it was assumed the model was attempting to predict zero, and hence a condition was added to set negative photolysis rates to zero. This prevented defiance of conservation laws and hence irrational behaviour within GEOS-Chem, such as an increase of mass in the system.

4 Model Performance on Validation Data

4.1 Evaluation Metrics

The following results are an assessment of the performance on the machine learning models compared to the validation data, calculated by Fast-JX (with the partition described in Section 3.3). To evaluate the performance of the models, three different metrics were used, each giving a different insight into a model's accuracy and reliability. One of the metrics is an adaptation of the loss function (the MSE in Eq. (10)) in the form of the normalised root mean squared error (NRMSE) (Eq. (13)). This metric is sensitive to large errors and in turn outliers (Chai and Draxler, 2014). Additionally, the normalised mean absolute error, NMAE, and the coefficient of determination, R^2, are used (Equations (14) and (15)). The NMAE is simply interpretable as the average error magnitude and is less sensitive to outliers compared to NRMSE, making it a more natural measure of error (Willmott and Matsuura, 2005). The R^2 metric, provides insight into the proportion of variance explained by the model, a powerful metric in measuring predictive power (Alexander et al., 2015). This can however be misleadingly high for poor models if the data has high variability, hence in conjunction with the NRMSE it forms a powerful analysis. The NRMSE and NMAE are normalised with the mean of the true values from Fast-JX ($\bar{y}$) to account for the variation between the photolysis rates of different species. This allows for direct comparison between the different machine learning models and their varying orders of magnitudes. The formulas for the evaluation metrics are as follows:

$$\text{NRMSE}(y,\hat{y}) = \frac{1}{\bar{y}}\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{13}$$

$$\text{NMAE}(y,\hat{y}) = \frac{1}{\bar{y}}\frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{14}$$

$$R^2(y,\hat{y}) = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{15}$$

Here, y_i is the true photolysis rate calculated from Fast-JX, $\hat{y}_i$ is the predicted photolysis rate from the machine learning, and $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$ is the mean of the true values out of n data points.

4.2 The Model Statistics for Validation data

To assess how well the models performed on the validation data, the predicted ozone to $O(^1D)$ and NO_2 photolysis rates (shown in Reactions R3 and R4 and denoted $J(O^1D)$ and $J(NO_2)$, respectively), were plotted against the true values calculated by Fast-JX for the whole of the validation data (Figure 5).

$$O_3 + h\nu \rightarrow O_2 + O(^1D) \tag{R3}$$

$$NO_2 + h\nu \rightarrow NO + O \tag{R4}$$

The $J(O^1D)$ and $J(NO_2)$ rates are considered here as key rates as they represent different partitions of the wavelength spectrum: $J(O^1D)$ rates are determined by short wavelength (largely UV spectrum) conditions where ozone absorption and Rayleigh scattering dictate the amount of photons available. In contrast, NO_2 photolysis is influenced more by the properties of clouds and controls the middle of the wavelength spectrum.

The main density of points in Figure 5 is near, or on, the true-predicted (dashed black) line for both species for the linearly scaled plots, panels (a) and (c). This indicates a generally good performance, especially at the higher magnitudes. The machine learning models are worse at the lower magnitudes shown in the log space (panels (b) and (d)), especially for $J(NO_2)$. In the log space, the $J(O^1D)$ rates appear to have a better performance compared to the $J(NO_2)$ rates, which have a higher NRMSE and NMAE and a lower R^2 value. The linear scale results present a more mixed picture. This range in performance can be explained by the nature of the different distributions of the data. The $J(O^1D)$ rate predictions are generally accurate (high R^2) but have larger relative errors at more extreme, high values, affecting the NRMSE. The $J(NO_2)$ rates, on the other hand, show a more consistent performance across the range of photolysis rates, resulting in the lower NMAE and NRMSE, despite a lower overall correlation.

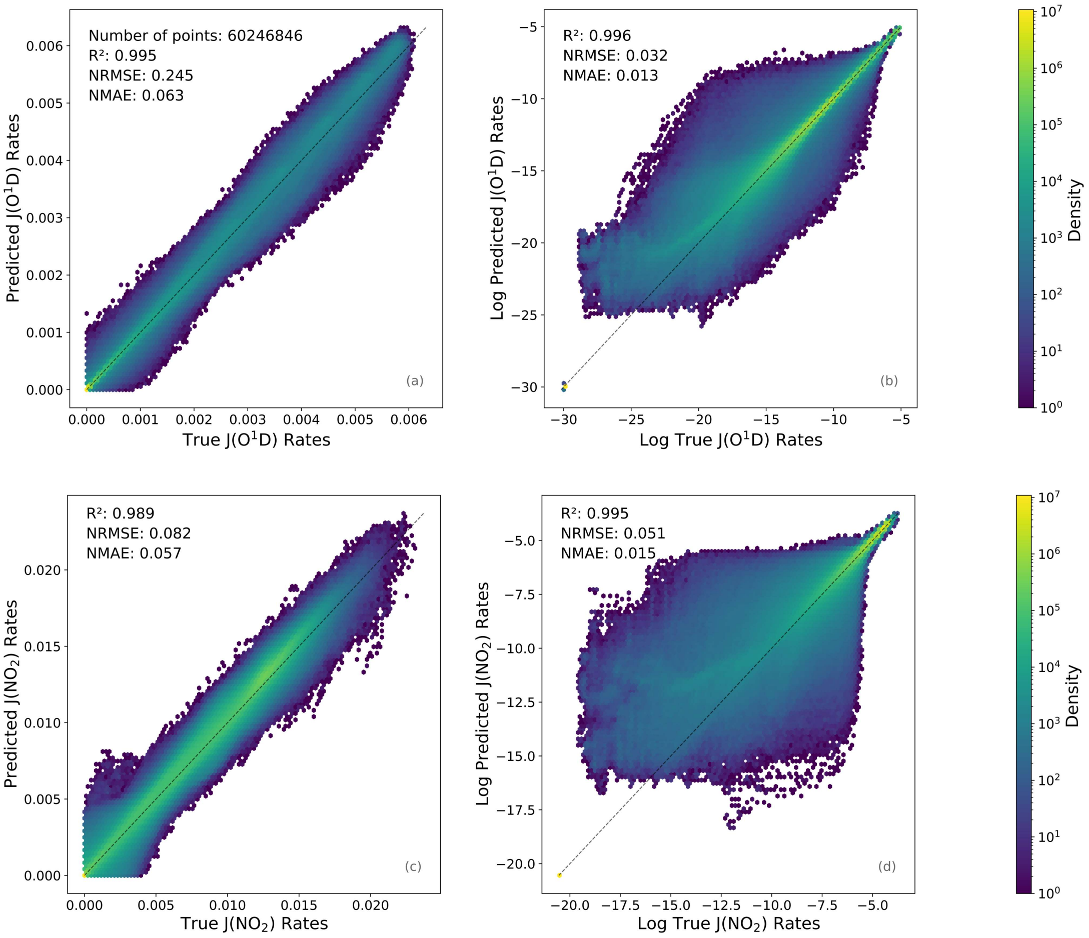

Figure 5: Predicted against true photolysis rates for J(O^1D), panels (a) and (b), and J(NO_2), panels (c) and (d). The hex-bin plots show the density of points that fall under each value in each hexagon where yellow represents roughly 10^7 points and dark purple is 1 to 10 points. Panels (a) and (c) are the normal scale photolysis rates and panels (b) and (d) are photolysis rates in the logarithmic space following predictions. The dashed black line represents when the calculated values are equal to the predicted values and hence indicate minimal error.

Table 4 presents the evaluation metrics for the species found in Figure 5, as well as additional key species: the photodissociation of ozone to atomic oxygen, hydrogen peroxide, nitrogen compounds, organic compounds, and two halogen species. Since the predictions were transformed into the logarithmic space during training, the metrics calculated for the log transformed data, as well as those calculated post-transformation into linear space are shown. The compression of the photolysis rates into logarithmic space allowed the machine learning models to train accurately and reduce the impact of the range of the data. Naturally this reduction in variability makes the metrics show a relatively high degree of accuracy, where all 111 machine learning models had an NRMSE below 0.2 and an NMAE below 0.1, as well as 109 models with an $R^2 > 0.98$. Additionally 94 machine learning models had an $R^2 > 0.995$ in logarithmic space. Despite this reassuring performance, it is the linear photolysis rates that are used within the GEOS-Chem CTM, and hence this becomes the more important space. In the linear photolysis rates, outliers are more heavily penalised in the metric calculations (particularly the NRMSE). For the validation data in the linear space, the machine learning models maintained a strong predicting performance, as 109 out of 111 models had an $R^2 > 0.98$, which indicate little deviation from predicted and real photolysis rates. Although, it should be noted only 48 models had an R^2 greater than 0.995. The NRMSE was below 0.2 for 96 models and 109 models had an NMAE below 0.1. A table containing all 111 machine learning statistics can be found in Appendix 8.2, Table 7. Interestingly, in Table 7 the photolysis predictors with the species IDs 'NIT' and 'NITs' have poor performance in the linear space, but satisfactory performance in the log space. This is most likely due to small errors in the log space translating into errors of several orders of magnitude when transformed into the linear space. The performance of these photolysis rates when incorporated back into the GEOS-Chem model is discussed later.

Table 4: Performance metrics of the R^2, NRMSE, and NMAE for key species calculated from the whole of the validation data. The metrics shown are both in linear and logarithmic space.

		Linear Space			Log Space		
Species ID	Species Info	R^2	NRMSE	NMAE	R^2	NRMSE	NMAE
O3O1D	$O_3 \rightarrow O^1D$	0.995	0.245	0.063	0.996	0.032	0.013
RCHO	>C2 Aldehydes	0.993	0.090	0.057	0.996	0.031	0.010
HNO_3	Nitric Acid	0.996	0.171	0.054	0.997	0.024	0.009
BrO	Bromine Monoxide	0.990	0.082	0.057	0.996	0.060	0.019
NO_2	Nitrogen Dioxide	0.989	0.082	0.057	0.995	0.051	0.015
NO_3	Nitrate Radical	0.984	0.093	0.064	0.995	0.095	0.028
Cl_2	Diatomic Chlorine	0.990	0.082	0.057	0.995	0.044	0.013
HPETHNL	Hydroperoxyethanal	0.994	0.089	0.056	0.996	0.031	0.010
MP	Methylhydroperoxide	0.995	0.096	0.054	0.996	0.027	0.009
MVK	Methyl Vinyl Ketone	0.993	0.086	0.054	0.996	0.027	0.009

4.3 Feature Importance

To provide an understanding of how the XGBoost models were influenced by the input variables, the SHapley Additive exPlanations (SHAP) values were used. SHAP values, founded in game theory, provide a consistent method to interpret model predictions and provide an insight into the importance of the different input variables, in this study the ‘TreeExplainer’ functionality in the SHAP Python package was used (Lundberg et al., 2020). Figure 6 shows the feature importance calculated from a sample of the validation data (n=500,000 points), for the photolysis rates J(O^1D) and J(NO_2) (panels (a) and (b) respectively). These SHAP plots, known as beeswarm plots, highlight the most impactful variables, with the top nine features shown individually and the remaining features’ summed importance as the tenth category.

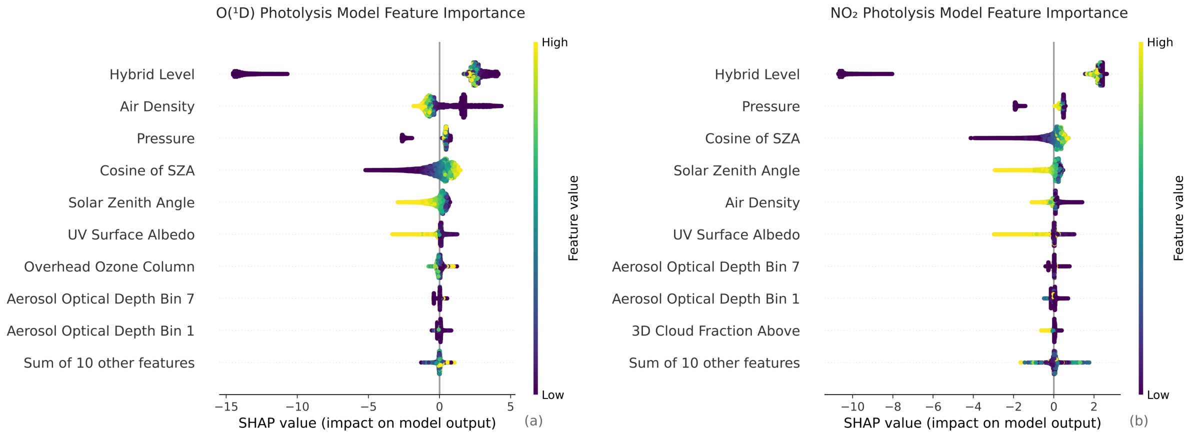

Figure 6: Feature importance plot in the form of a SHAP summary showing the top nine features and the cumulative impact of the remaining 10 features. This is for the species J(O^1D) in panel (a) and for J(NO_2) in panel (b).

Model level, air density, and pressure were important variables for predicting photolysis rates. This is unsurprising as photolysis rates decrease from the edge of space to the surface, due to scattering and absorbance in the atmosphere. This means there is a direct correlation between photolysis rates and height. The SZA and the cosine of the SZA were expected to be highly important variables for predicting photolysis rates due to their direct relation to solar flux and were found to be important for all species. Other significant contributors included the aerosol optical depth bins and the surface albedo, these variables were frequently ranked in the top 10 of feature importance. The aerosol optical depths affect the scattering absorption by various aerosols (Figure 1, part (iii)), and the surface albedo determines the reflectivity of

Earth's surface (Figure 1, part (ii)), influencing the amount of light available for photolysis. From the variables calculated and not directly collected from GEOS-Chem used in the input vector, the cloud fraction above was often the most impactful. The remaining variables were combined together into a single importance category due to their small individual impact. Given the satisfactory performance of the predictors on the validation data and that the rationale behind the predictions are inline with the science that governs photolysis, the implementation of the machine learning models within GEOS-Chem was investigated.

5 Model performance when integrated into GEOS-Chem

The following Section presents the results from the machine learning predictors after implementation in GEOS-Chem. The model is run from the start of July 2020 to the end of June 2021. The initial six months (July to December 2020) served as the spin-up period and the subsequent six months (January to June 2021) were used for analysis and evaluation. Here we analyse the performance of the same two photolysis rate predictors analysed in Section 4: the photolysis of ozone to atomic oxygen ($O(^1D)$) and the photolysis of nitrogen dioxide (NO_2), as well as the performance of other key photolysis rates (Section 5.1). Next, the performance of the photolysis rates at different altitudes in the CTM is evaluated (Section 5.2). Attention is then turned to calculated concentrations, comparing the default GEOS-Chem concentrations calculated using both the Fast-JX (denoted FJX, for brevity, in these results and most of the discussion) and machine learning (denoted ML, for brevity, in these results and the discussion) photolysis rates in Section 5.3.

5.1 Prediction of Photolysis Rates

Figure 7 shows the $J(O^1D)$ surface photolysis rates calculated by GEOS-Chem from Fast-JX (top row), the ML model (middle row), and the relative difference between the two (bottom row) at 00:00 UTC on the first day of the first, third, and sixth month post spin-up. This provides context as to where most error arises. Errors were consistently large around the terminator (the point at which day becomes night and night becomes day), where the SZA is around 90°, and there is much less photolytic activity. This is consistent with Figure 5 for the validation data, where there was more error at lower photolysis rates. The terminator is a source of complexity studied in CTMs due to the steep gradients of chemical concentrations resulting from rapid changes in sunlight available for photolysis (Lauritzen et al., 2015). The ML models largely over-predicted (in excess of 200%) at the terminator and generally under-predicted everywhere else. We learn that the third month is not unique, but instead shows that the error around the terminator steepens. This is consistent with the increase in the gradient of the SZA and light as the simulation approaches the equinox (March). This behaviour is consistent for other species.

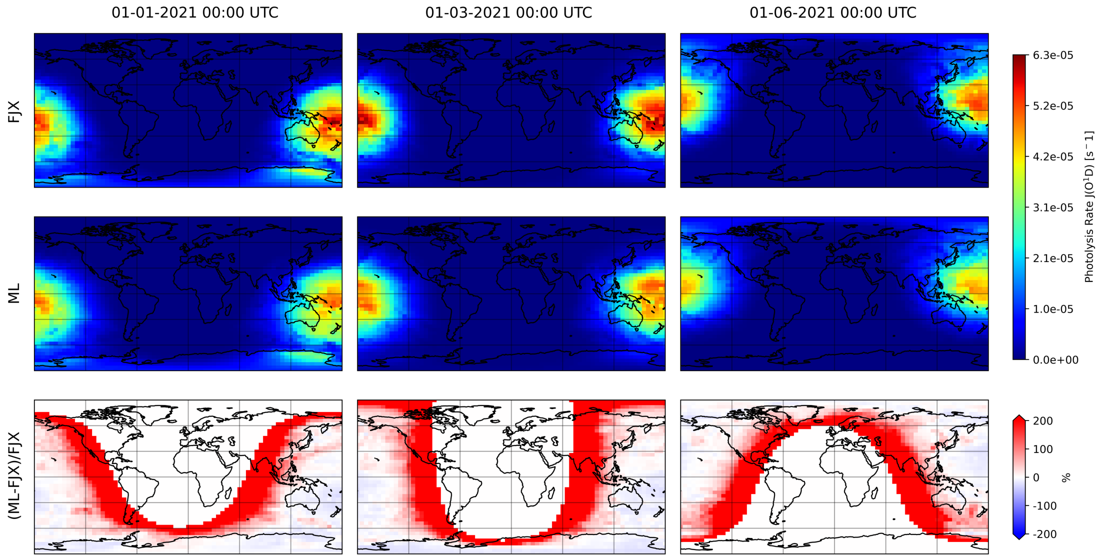

Figure 7: Photolysis rate maps of surface J(O^{1}D) rates following the sixth month spin-up on the first day of the first month (01 January 2021, column 1), the third month (01 March 2021, column 2), and the sixth month (01 June 2021, column 3) calculated by the Fast-JX photolysis scheme (FJX, row 1), and the machine learning approach (ML, row 2). The percentage relative difference between row 1 and row 2 is shown are shown in row 3.

Figure 8 shows the time evolution of the R^2 metric between the FJX and ML rates for J(O^{1}D), J(NO$_2$), and J(NO$_3$); these were calculated for both the troposphere (panel (a)) and the whole vertical column (panel (b)) for the post spin-up integration period. From other ML parameterisations in CTMs, it is found that the error can accumulate through each internal time step and so propagate within the transport model (Keller and Evans, 2019). There are some cases where the error grows so large that predictions become unusable (Kelp et al., 2018). The stability of the photolysis predictors can be analysed by their respective R^2 values over time. The tropospheric photolysis rates, panel (a), are mostly stable and generally high (consistently greater than 0.97), this accuracy was consistent for the predictors over the entire vertical column, Figure 8b.

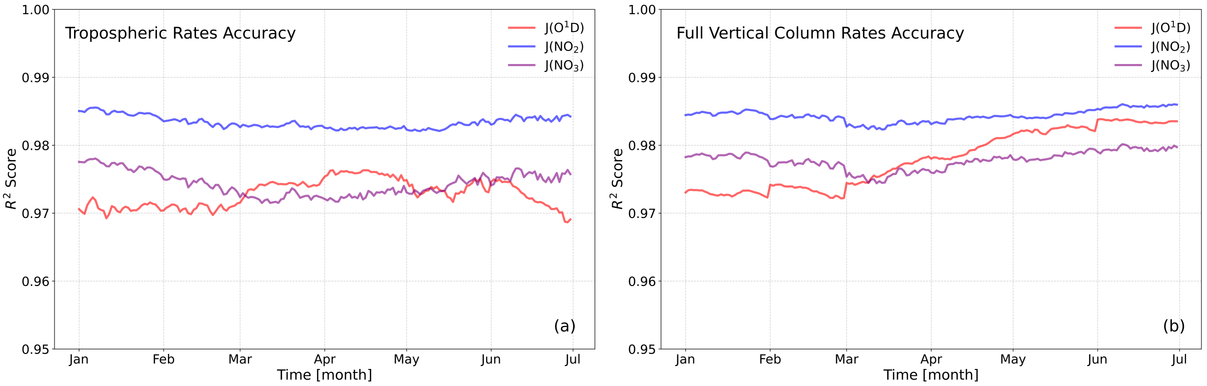

Figure 8: Error propagation and evolution of the R^2 metric between Fast-JX and machine learning photolysis rates: J(O^1D) (red), J(NO_2) (blue), J(NO_3) (purple). Panel (a) is the metrics for the troposphere and panel (b) is the metrics calculated for the entire vertical column. The time period is from the start of January 2021 to the end of June 2021 (post the 6 months spin up).

In the troposphere in January 2021, after six months spin-up, 68 out of the 105 ML predicted photolysis rates maintain an R^2 larger than 0.98, with an additional 81 ML rates consistently having an R^2 greater than 0.95. By June 2021 (six months into the analysis period), the amount of ML predicted rates with an R^2 greater than 0.98 slightly reduces down to 67, but 81 maintain an R^2 greater than 0.95. Table 5 summarises these findings for key species, and is a tabulation of Figure 8. It is noted that for some species, the performance is better in the first month, compared to the sixth, and other species show opposite behaviour. Combining the tabulated data and the plotted metrics, we conclude that the species appear to be equilibrated and in a steady state of prediction. The performance metrics for all 105 tropospheric photolysis rates can be found in Appendix 8.3, Table 8. When the entire vertical column is considered, including the stratosphere, the metrics are similar to the troposphere. From the 105 ML predicted photolysis rates, 73 and 97 maintained an R^2 over 0.98 in the first and sixth months respectively, post spin-up, and 103 ML rates in both these months had an R^2 over 0.95. Most stratospheric and hence full vertical photolysis rates show satisfactory accuracy. A table containing all 105 metrics for the full vertical column photolysis rates calculated between the ML and FJX rates can be found in Appendix 8.3, Table 9.

It should be noted for the tables in the appendix containing both tropospheric and full vertical column metrics, all R^2 are generally good except for the photolysis rate with the species ID ‘SO4’. This discrepancy occurs because this photodissociation is considered a special reaction and is later adjusted by a different module to the photolysis rate calculations (Prather, 2012).

Therefore the poor performance between the predicted and FJX photolysis rates is not of concern. Additionally, in the validation data there was concern for the photolysis rates with species IDs ‘NIT’ and ‘NITs’. These have satisfactory performance in both Table 8 and Table 9.

Table 5: Performance metrics of the R^2, NRMSE, and NMAE for key species photolysis rates within the troposphere from the first day of the first month (01 January 2021) and comparatively the first day of the sixth month (01 June 2021) post the six month spin-up. Calculated between predicted rates and Fast-JX rates

		01 January 2021			01 June 2021		
Species ID	Species Info	R^2	NRMSE	NMAE	R^2	NRMSE	NMAE
O3O1D	$O_3 \rightarrow O^1D$	0.971	0.315	0.155	0.975	0.303	0.149
RCHO	>C2 Aldehydes	0.985	0.173	0.095	0.878	0.689	0.371
HNO_3	Nitric Acid	0.981	0.211	0.111	0.983	0.203	0.110
BrO	Bromine Monoxide	0.986	0.154	0.085	0.985	0.161	0.088
NO_2	Nitrogen Dioxide	0.985	0.155	0.086	0.974	0.193	0.108
NO_3	Nitrate Radical	0.978	0.182	0.099	0.983	0.164	0.092
Cl_2	Diatomic Chlorine	0.986	0.152	0.084	0.985	0.160	0.088
HPETHNL	Hydroperoxyethanal	0.986	0.171	0.094	0.986	0.171	0.092
MP	Methylhydroperoxide	0.986	0.159	0.088	0.986	0.162	0.088
MVK	Methyl Vinyl Ketone	0.987	0.178	0.088	0.987	0.181	0.086

5.2 Predictive Ability in the Vertical Columns

Figure 9 shows the relative difference (%) between the Fast-JX and ML rates for O(^{1}D) (top row), NO_2 (middle row), and NO_3 (bottom row) at the surface (0 km elevation), the 35th (16 km elevation), and the 55th (42 km elevation) vertical level for the columns from left to right for 00:00 UTC on 01 January 2021 (the first time step post spin-up). We find that the error around the terminator, in Figure 7, is not limited to just the surface level, nor to the photolysis of ozone. This is present at all levels and for many species, just at varying intensities. For the photolysis rates shown in Figure 9, the number of over-predicted values around the terminator decreases with higher vertical levels in GEOS-Chem; these are instead replaced by a higher incidence of under-predicted values, albeit at a lower magnitude of error. The reduction in error around the terminator, as the vertical level increases, can most likely be attributed to the differences in tropospheric and stratospheric dynamics and properties. Sources of scattering, like clouds and aerosols, are reduced in the stratosphere and hence contribute to more straight-forward predictions. This behaviour largely accounts for the higher quantity of ML models performance in the entire vertical column, than in just the troposphere.

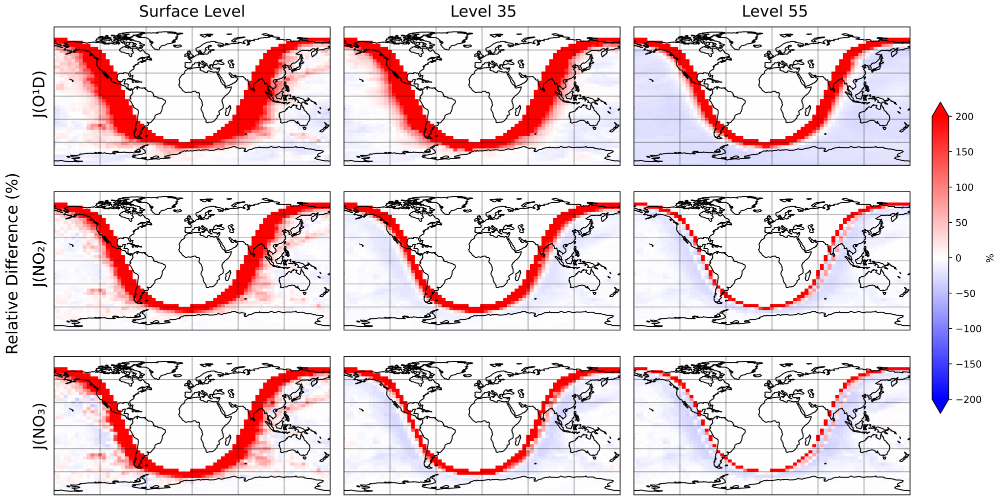

Figure 9: Percentage relative difference map for the photolysis rates: J(O^{1}D) (row 1), J(NO_2) (row 2), and J(NO_3) (row 3) for 01 January 2021 00:00 UTC at 3 different atmospheric vertical levels. Column 1 is surface level (0 km elevation, 1000 hPa), column 2 is level 35 (16 km elevation, 100 hPa), and column 3 is level 55 (42 km elevation, 2.3 hPa).

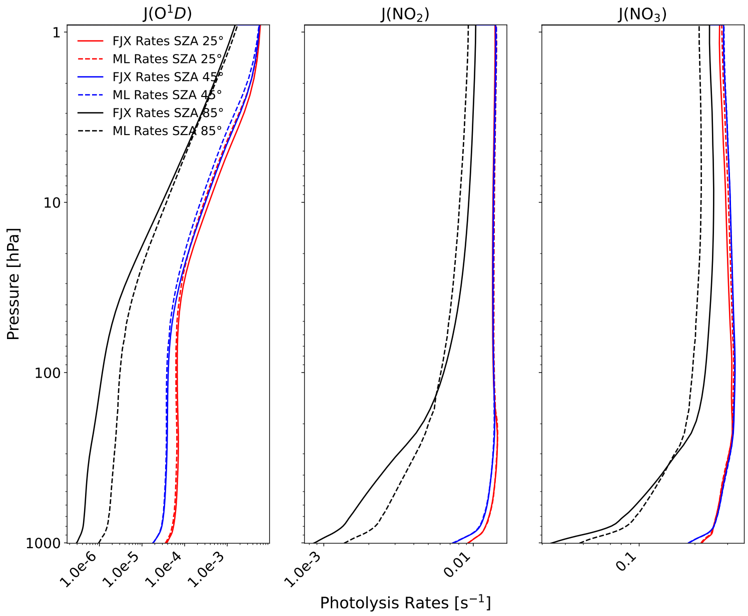

Figure 10: Vertical profile for the photolysis rates for J(O^{1}D), J(NO$_2$), and J(NO$_3$) comparing the Fast-JX rates (solid line) and the machine learning rates (dashed line) at 3 different solar zenith angles: 25°(red), 45°(blue), and 85°(black). This is averaged over all longitudes, latitudes, and times for the respect SZA and level.

Figure 10 shows the vertical profile for three photolysis rates (on a log scale) at three different solar zenith angles (25°, 45°, and 85°) averaged over longitude, latitude, and time, from the start of January to the end of June 2021, comparing the Fast-JX rates (solid line) and ML rates (dashed line). The SZAs used increase with the photolytic intensity, hence it is expected the photolysis rates for lower SZA (25°and 45°) to be higher than a larger, and darker, SZA (at 85°). We find that the vertical levels at which over-predictions and under-predictions at 85° happen, are similar for the J(NO$_2$) and J(NO$_3$) rates, but different for J(O^{1}D) rates. Over-predictions occur below 110 hPa (around 15 km elevation) and 130 hPa (around 14.5 km elevation) for J(NO$_2$) and J(NO$_3$) respectively. The J(O^{1}D) rates are constantly over-predicted at 85°, but become closer to the true rates higher in the atmosphere. Specifically for the nitrogen compounds, when the over-predictions change direction in under-predictions at 85°, is roughly the change between the troposphere to the stratosphere. This mirrors the decrease in over-predictions by level around the terminator in Figure 9. Both of the nitrogen compounds' photolysis rates in Figure 10 show little deviation when the SZA is 25°and 45°, except for the slight over-predictions for the J(NO$_3$) rates above 110 hPa (for a SZA of 25°). In contrast, the J(O^{1}D) rates are minimally under-predicted for these SZAs.

Although this analysis does not cover every ML model, it highlights that key photolysis rates, in general, are accurately predicted over a variety of altitudes. There are points where the ML models are less accurate (such as around the terminator), but the overall magnitude of error remains small and within acceptable limits, as indicated by the 103 out of 105 ML models with an R^2 above 0.95 in the entire vertical column (in Table 9 in the Appendix). Now that the photolysis rates predicted by the ML models implemented in GEOS-Chem has been evaluated, the difference between the concentrations of species calculated using the ML and FJX photolysis rates can be analysed.

5.3 Concentration Calculations using the Predicted Rates

Within GEOS-Chem, the predicted photolysis rates are stored within an array that is later used within the Kinetics PreProcessor (KPP) to calculate concentrations. The concentration of ozone is a key concentration in transport models due to its radiative properties influencing temperature distribution and climate patterns, as well as its chemical interactions with other species (Monks et al., 2015). In this parameterisation it requires a low margin of error, especially considering the total overhead ozone column is a quantity used in the photolysis ML input vector. Here, the effect photolysis rate predictions have on the ozone column, the nitrogen dioxide and ozone surface concentrations are investigated, as well as the potential for error propagation. Figures 11 and 12 illustrate the total ozone column and the total tropospheric column, respectively. These figures compare the concentrations calculated using the FJX rates (top row), the ML rates (middle row), and the relative difference between the two (bottom row) at 00:00 UTC on the first day of the first, third and sixth months after spin-up. In January, the divide between over and under calculations in the total overhead column (Figure 11) is split equatorially, with an over estimate in the Northern Hemisphere and under estimate in the South. By the sixth month, the error in the total overhead ozone grows from ±2% to +5% and the excess ozone concentration moves southwards to the equator. This likely reflects difference in the ozone chemistry between the North and South Hemispheres. The tropospheric column remains largely under-calculated when using the ML rates compared to the FJX rates, growing from -2% generalised globally to -10% localised to the Southern Hemisphere by the sixth month (post spin-up). Since the Southern Hemisphere is entering winter in June, it could be the case of over-predicting ozone photolysis rates due to the equivalent of the ML models predicting as if there is too much solar flux. This reduces the amount of ozone concentration in comparison to what is calculated with the FJX rates. This, in turn, creates a significant hemispheric contrast. Ultimately, these relative differences are small considering the time it took for this error to build up (6 months, following a 6 month spin-up).

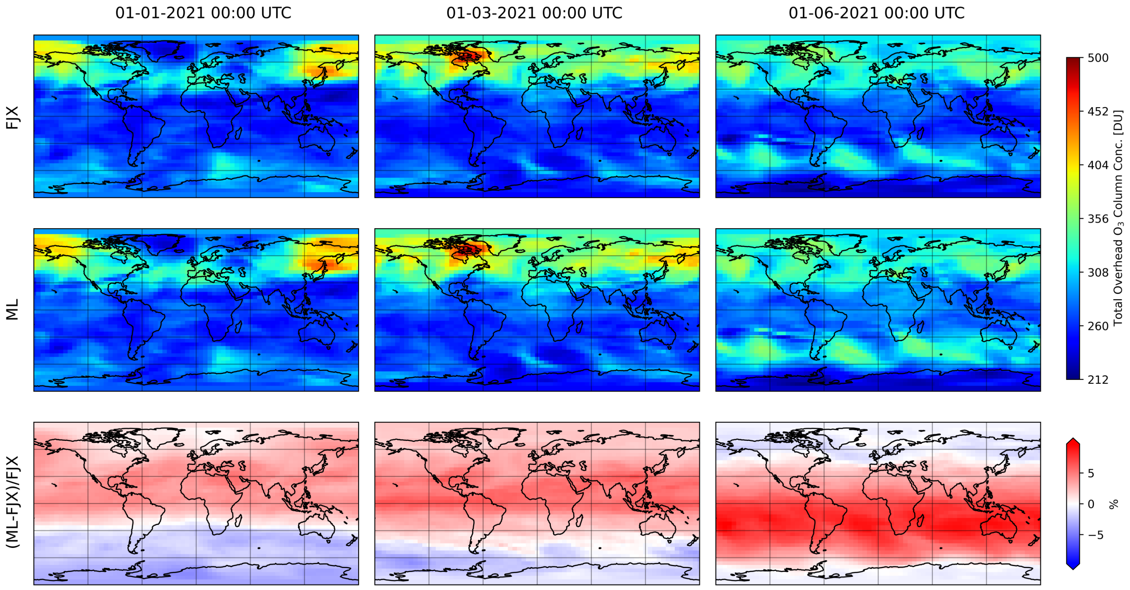

Figure 11: Total overhead ozone column maps (DU) for all vertical levels averaged over the first day of the first month (01 January 2021, column 1), the third month (01 March 2021, column 2), and the sixth month (01 June 2021, column 3) as calculated using the Fast-JX rates (FJX, row 1), and the machine learning rates (ML, row 2). The percentage relative difference between row 1 and row 2 is shown in row 3.

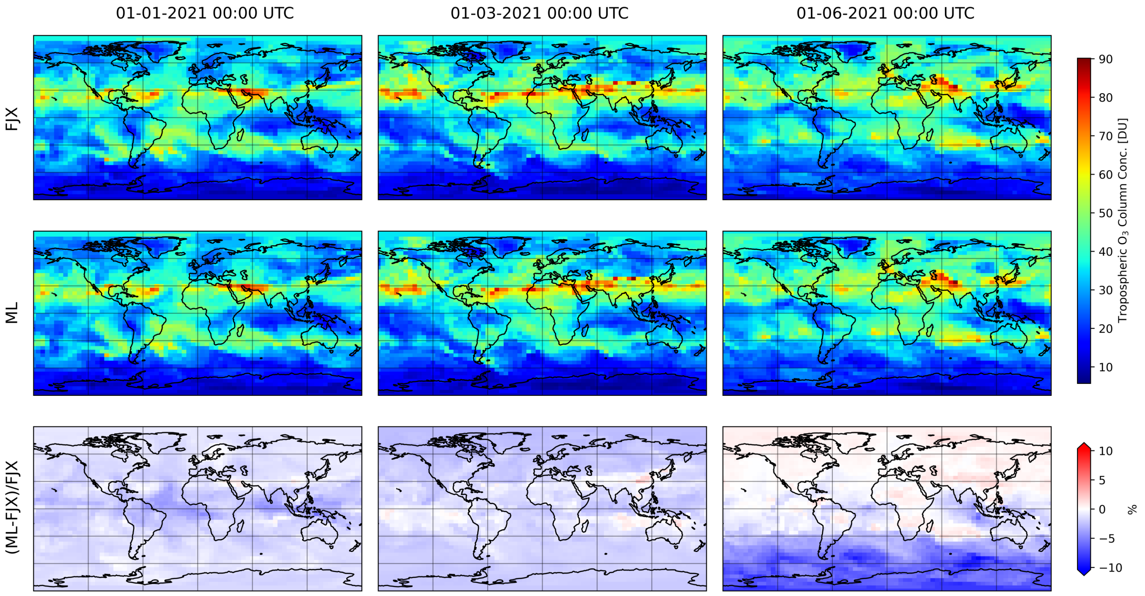

Figure 12: The same as Figure 11 but showing the total tropospheric ozone column (DU), calculated up to the tropopause, instead of total ozone column.

Figure 13 and Figure 14 show time series plots for surface concentrations for O_3 and NO_2 at four different locations, three polluted (London, Shanghai, and New York City) and one clean (Tahiti, French Polynesia) using the default GEOS-Chem (denoted GC as the red line), one using the ML photolysis rates (blue), and one without photolysis enabled at all (black). The ML concentrations closely follow the GC concentrations, demonstrating the reliability of the integration over this period. The fluctuations in O_3 surface concentration in the clean location are closely matched but are slightly under-calculated, particularly in June and July. In contrast, the more polluted locations show the opposite. The minima and maxima are modelled accurately by the ML rates, suggesting the ML models are appropriately calculating photolysis rates. For comparison, the lack of photolysis leads to concentrations that rapidly diverge from the base model into a highly different state. The similarities between the GC and ML NO_2 surface concentrations, combined with the model failure of the concentrations without photolysis rate calculations in Figure 14 further highlights the success of the implementation of the predictors into GEOS-Chem. The concentrations, for both O_3 and NO_2, specifically in Shanghai, panel (b), further support that minimal aerosol representation in the ML models does not cause significantly faulty predictions, even when propagating through to the concentrations. Shanghai provides a good reference point due to its historically polluted conditions in terms of aerosols (Shen et al., 2020).

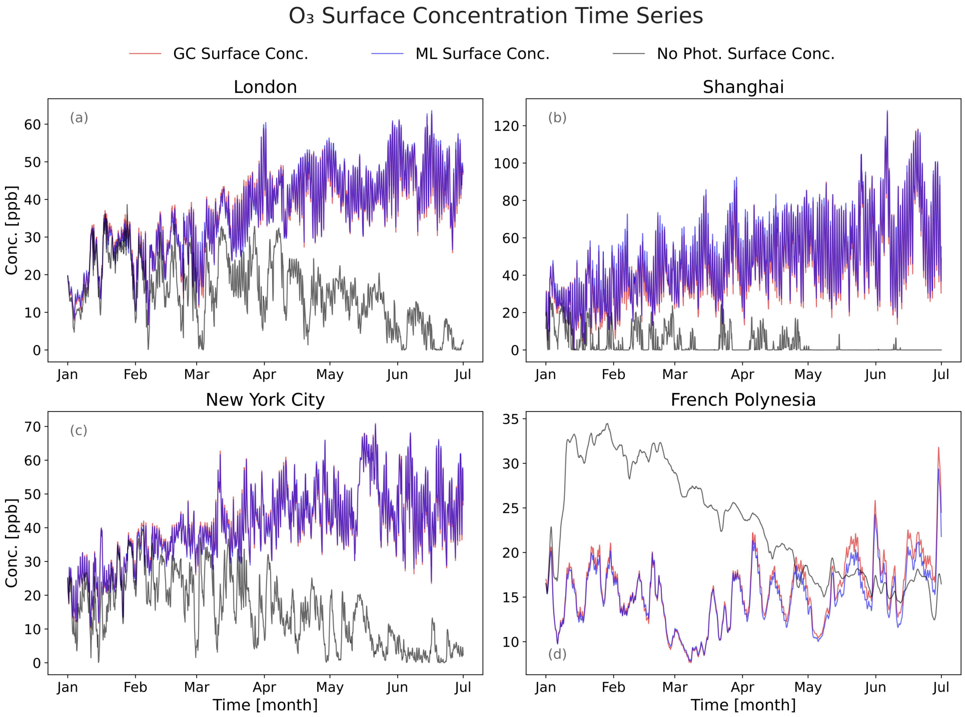

Figure 13: Time series plot showing the comparison of O_3 surface concentrations using the default GEOS-Chem concentration (red), using the machine learning rates (blue), and without photolysis rate calculations (black) at four locations: London (51.5074° N, 0.1278° W), Shanghai (31.2304° N, 121.4737° E), New York City (40.7128° N, 74.0060° W), and French Polynesia, specifically Tahiti (17.6797° S, 149.4068° W).

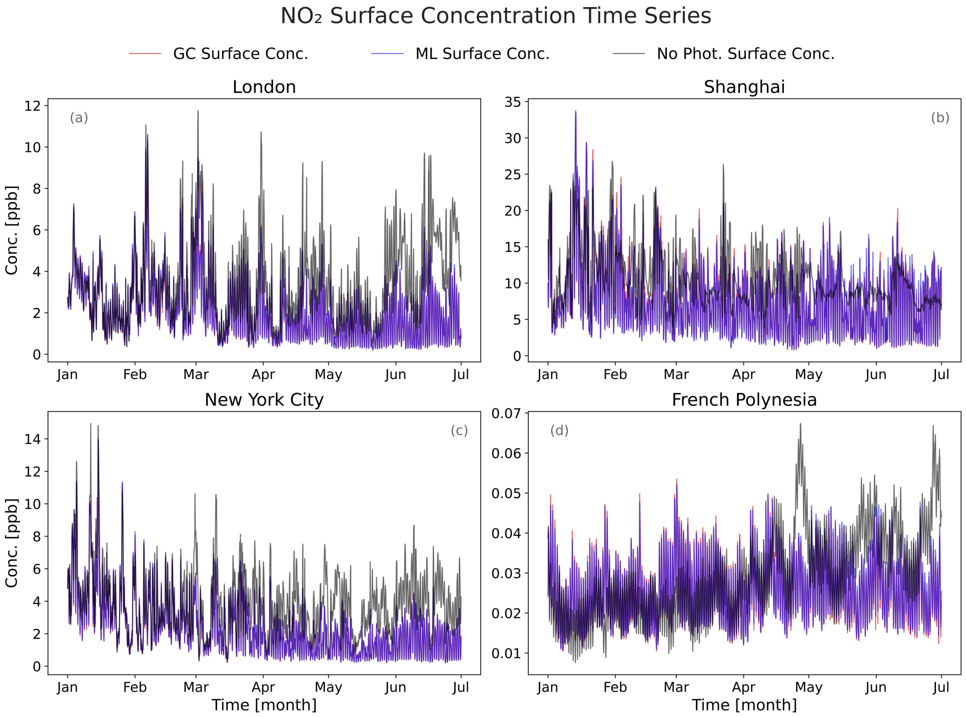

Figure 14: The same as Figure 13 but with NO_2 surface concentration instead of for ozone.

The error propagation (R^2) in terms of the concentrations, is shown in Figure 15. In both panels, the O_3 and NO_2 concentrations appear stable, although the ozone accuracy has a slight deterioration from May onwards in panel (a) and April in panel (b). NO_3 concentrations on the other hand shows more intense fluctuations, and a downwards trend. In Figure 15a, the accuracy for the different species is related to the lifetime, O_3 is the longest lived and most accurate, whilst NO_3 is the shortest and least accurate. The changes in error, and overall reduction in accuracy can largely be attributed to the accumulation of errors from the photolysis rate predictions contributing to the error of the concentrations. These exist longer than the instantaneous quantity of the photolysis rates, that do not rely on the previous time steps values. Whilst many photodissociations contribute to these concentrations, these patterns generally match those found in Figure 8, with NO_2 being the best performing in both.

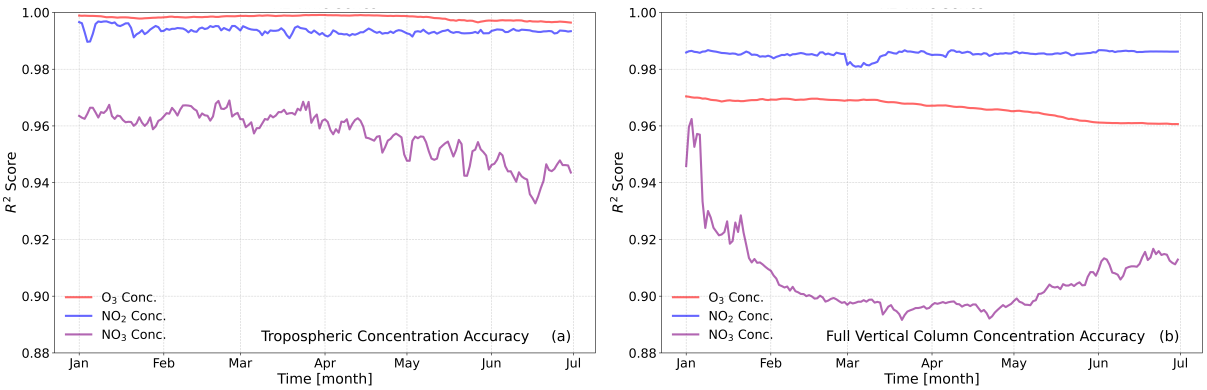

Figure 15: Error propagation and evolution of the R^2 between GEOS-Chem concentrations using Fast-JX rates and predicted rates in both the troposphere (a) and in the full vertical column (b), for O_3 conc. (red), NO_2 conc. (blue), and NO_3 conc. (purple).

Table 6 is a tabulated view of the metrics shown in Figure 15 panel (b), including additional key species. The performance of the concentrations for all species, whether organic, inorganic bromine or nitrogenous compounds, is robust and reassuring when using the predicted photolysis rates. It can be noted species that do not rely on photolysis rates remain unaffected by errors in the ML models, such as N_2.

Table 6: Same as Table 5 but for key concentrations over all vertical levels.

		01 January 2021			01 June 2021		
Species ID	Species Info	R^2	NRMSE	NMAE	R^2	NRMSE	NMAE
O_3	Ozone	0.970	0.241	0.105	0.961	0.275	0.122
OH	Hydroxyl Radical	0.974	0.677	0.141	0.966	0.793	0.154
ROH	>C2 Alcohols	0.993	0.936	0.080	0.990	1.843	0.128
RCHO	>C2 Aldehydes	0.995	0.336	0.073	0.992	0.818	0.096
NO_3	Nitrate Radical	0.946	1.347	0.185	0.909	1.124	0.217
NO_2	Nitrogen Dioxide	0.986	0.246	0.071	0.987	0.239	0.074
N_2	Nitrogen	1.000	0.000	0.000	1.000	0.000	0.000
BrO	Bromine Monoxide	0.984	0.196	0.067	0.985	0.195	0.067
CH_2O	Formaldehyde	0.996	0.114	0.048	0.996	0.123	0.045
ACET	Acetone	0.990	0.165	0.077	0.984	0.199	0.120

6 Discussion

6.1 Hardware and Software Optimisations

The predicting time for the photolysis ML models (161 models, including the reused ones) is significantly longer than the current optimised FJX calculations. The ML models implemented in GEOS-Chem (the 161 ML models) take over 8500 seconds for 135072 grid boxes (on average over 6 time steps), but only 3.4 seconds with FJX (with the same compute resource for the similar time steps). 2 ML models predicting in the same environment took around 35 seconds on average, 4 predictors took around 77 seconds and 8 took 137 seconds. The prediction time scales non linearly in length with the number of predictors, more than doubling in time when doubling the amount of predictors. Table 10 in Appendix 8.3 shows the timings, as well as the individual times for the 6 time steps for the 6 different amounts of predictors and the FJX times. Additionally, for each ML model, the time to predict was determined by how many grid boxes were undergoing prediction and not considered dark. There are a few reasons that this could be significantly slow, and one of the ways the bottlenecks could be found an optimised is through a profiler. Simply put, the profiler would measure the execution of a code, finding particular parts that are inefficient and slow. Profilers are available in many languages and relevant to this project there are some available for Fortran (Shende and Malony, 2006).

The performance issues may occur due to factors involving both the implementation and execution environment. The collection of predictors were stored on personal scratch storage on the Viking 2 HPC, instead of the optimised flash storage. This storage is not optimised for use in bigger projects, potentially affecting the model loading and execution speeds. Additionally the exact memory usage by the models is uncertain and could cause a large overhead. The prediction process itself could be a source of the slow prediction time, potentially showing the machine learning approach is over-complicated compared to the Fast-JX calculations. This is likely not the case, but the source of increased computation time is hard to determine without profiling the specific issues.

Despite the prediction time being more computationally intensive and substantially longer than the Fast-JX calculation, this project was purely a proof-of-concept with three main objectives, previously mentioned: firstly, to evaluate how well machine learning models could predict a broad range of photolysis rates; secondly, to assess the feasibility and effectiveness of implementing these ML models in GEOS-Chem; and thirdly, to analyse the impact of these implemented photolysis rate on the entire GEOS-Chem simulation. Given this has been achieved and the initial hurdles of implementing it in Fortran to the high degree of accuracy has been shown, the optimisations can be of focus. Other aspects of GEOS-Chem, including Fast-JX, are highly optimised and the potential optimisations for the ML approach are now discussed.

The hardware optimisations includes the use of OpenMP to parallelise predictions, with CPUs. However, in machine learning tasks GPU hardware is commonly used due to its inherently high levels of parallelism for the linear algebra operations. The same hardware could be used for predicting in this project. XGBoost has an implementation of CUDA (Compute Unified Device Architecture), a parallel computing platform developed by NVIDIA for computing on GPUs (Mitchell and Frank, 2017). CUDA allows for high-performance computing by leveraging the massive parallelism of GPUs, similar to how OpenMP allows parallel processing on CPUs. For the training process, H100 GPUs drastically reduced the training time, taking just over 28 hours of compute time for the training of all 111 ML models. The difficulty arises when trying to enable the CUDA platform on Fortran, which was beyond the scope for this project. The H100 GPU, used for the training and potentially available for the predicting, is state of the art and fast (Choquette, 2023), and hence would possibly reduce the time for the photolysis predictions. Without benchmarks it is hard to determine the reduction in time precisely but hence provides a direction for future work and, if true, would make the comparison in time taken between FJX and this parameterisation task more competitive.

Whilst the hardware optimisations offer the most significant speed ups to the existing workflow, some software and algorithmic optimisations could provide alternate paths of achieving the same goal. Firstly, XGBoost was tuned for accuracy. This yielded a complex and deep model with a maximum depth of 12 (seen in Table 3). This complexity increases the predicting time. In a similar fashion, the quantity of input variables contributes to a longer predicting time: a higher quantity of variables means more data has to traverse the depth of the XGBoost models; more data is loaded into the memory for prediction as well. Performing a more thorough variable importance analysis to reduce the quantity of variables would cut prediction time, as well as most likely increasing the accuracy of the predictions by removing highly correlated variables. For example, in the feature importance plots (Figure 6) the level, air density, and pressure were all found to be highly important but all correlated with height. This could perhaps be reduced to just one variable.

It was previously mentioned that the time for the predicting process within GEOS-Chem is largely scaled by the number of ML models with a significant number of machine learning models are reused. The most common being organic peroxides (22 times). An obvious solution would be to predict once and share the photolysis rates post prediction, reducing the amount of predictors from 161 to the amount of unique predictors (111).

An alternative to changing the algorithm and instead keeping a similar structure to the current work flow would be to consider a linear combination of photolysis models. Given a small set of photolysis machine learning models, their outputs could be used in a multiple linear regression to get every other photolysis rate. Storing the combination of coefficients for the other photolysis rates and simply referencing them in the linear combination would inherently be faster than predicting for all 161 species with XGBoost and faster than the numerical solver for FJX. This would not only work with XGBoost models but theoretically with any machine learning approach given a good enough predicting accuracy. Additionally, the Fast-JX code approach could be mimicked more closely instead of predicting each photolysis rate, the 18 different respective solar fluxes would be predicted. This would be similar to the process in Fast-JX, where the predicted solar fluxes are then used with the respective cross-sections to calculate the photolysis rates. This would reduce the number of predictions down from 161 to 18, whilst maintaining the scientific integrity behind the predictions.

On the deep learning front, neural networks (NNs) are increasingly popular due to their adaptability and capture complex interactions. One possible future approach is to use a physics-informed neural network (PINN) (Raissi et al., 2019). PINNs are a type of neural network that incorporate physical laws into their architecture, allowing them to learn the underlying physics of a system. In this case, PINNs could be used to predict the photolytic intensity or radiation, since they are heavily defined by physical laws. The predicted photolytic intensity would then be used in conjunction with the quantum yield and absorption cross-section to calculate photolysis rates. This would reduce the amount of predictions to just one, that of the photolytic intensity, and would be the same for all species. This could be achieved by taking advantage of the inherently high dimensional 3-D convolutional layers, found in many imaging problems, for the spatial data in the CTM and combining it with long short-term memory (LSTM) layers, which are optimal for temporal data and forecasting due to their ‘memory’. There are many caveats to this approach including that the inherent black box nature of NNs makes it harder to interpret feature importance (Castelvecchi, 2016), something that should always be considered when handling machine learning inside a scientific domain. Arguably a bigger challenge for this approach would be the integration within GEOS-Chem, as deep learning libraries have varying levels of Fortran integration and complexity.

6.2 Limitations

Whilst the data used to train the model is collected from a year long GEOS-Chem run, the training-validation split is done temporally. The first 9 months were for training and he subsequent 3 months were reserved for validation. This approach, although common in machine learning, leaves the model with no training data from April to the start of July (the 25% validation period). While this could theoretically make it harder for the machine learning models to generalise predictions for this period, this study found no evidence of this limitation. There was no diminishing predictive ability, even in model statistics for June 2021 (the last month of the validation period). Despite these encouraging results, this may not be the case for future runs and different simulations. An alternative approach to address this issue would be to simply split the data randomly, ensuring the models train on data points from all over the year. In hindsight, this would have been the most sensible approach during model development.

The effects of changing the model spatial and temporal resolution and internal time step is unexplored but their effects would likely impact the accuracy of the photolysis rates. Some of the variables used like the SZA, the temperature, the level, and surface albedo would largely not differ between the resolutions and time steps and, according to the feature importance, are some of the most important variables. This could mean there is potential versatility between different time steps and resolutions. However, other variables, like cloud cover and dust, are likely to vary strongly.

There are two main issues regarding accuracy of predictors. First, the negative photolysis rates found in the model development, where predictions were smaller than the constant used for log transformation. Reducing the size of the constant to be even smaller during training would most likely fix this issue, this could be done by increasing the sample size from which the constant was selected. As this issue was small and relatively limited, it did not significantly impact the overall performance. Another notable issue arising from the implementation of the photolysis predictors in GEOS-Chem, from Section 5, was the large error around the terminator. This happened when the machine learning models over-predicted the very small values that were calculated by Fast-JX under lower light conditions. This was an edge-case where the models fall short. A potential solution would be to train a separate model (or collection of models) for the specific solar zenith angle ranges where these issues occur. While there are performance issues arising from the terminator, it reflects a common challenge in atmospheric modeling and addressing this can be overlooked at the moment due to satisfactory overall performance.

7 Conclusions

This project's primary objective was to develop a machine learning-based parameterisation of the Fast-JX photolysis scheme used in GEOS-Chem, and that was successfully achieved. Most photolysis rates (103 out of 105) maintained an R^2 greater than 0.95 for the duration of the GEOS-Chem simulation. Additionally, the effect of the predictors has a minimal effect on other aspects of the GEOS-Chem simulation, with a high accuracy when using the predicted photolysis rates for concentration calculations. This proof-of-concept study is significant as it opens avenues for parameterising more complex processes in CTMs and, more significantly, it is the first machine learning-based parameterisation of photolysis rates for CTMs. Whilst this process shows promise, it is currently limited by the prediction speed, and future work should focus on addressing this. The use of GPUs as a hardware alternative to the current CPUs should be the main focus in accelerating the prediction process. Additionally, whilst the machine learning models show strong performance across a broad range of conditions, there are areas for improvement, such as around the terminator. The terminator, as described in Section 5.1, is a known challenge when modelling photolysis rates, and in atmospheric models more generally, due to the rapid changes in light. Further work in regards to accuracy improvement could explore specialised models/approaches for these edge-cases. Despite this challenge, which affects a relatively small proportion of the global predictions, the machine learning models demonstrate a generally robust performance across the majority of atmospheric conditions.

More broadly, this work contributes to the growing field of machine learning applied in the context of atmospheric sciences. By demonstrating the feasibility of a purely machine learning-based photolysis prediction system to potentially replace photolysis schemes like Fast-JX, it paves the way for potentially more efficient, flexible, and eventually cheaper atmospheric modelling. This could result in more detailed, higher resolution or longer-term simulations due to the additional computational resources available. The success of this machine learning approach in parameterising the complex physical process of photodissociation could inspire similar efforts in other areas involving physical processes. With additional compute, the fast development of algorithms, and more attention, machine learning has the chance to greatly improve scientific computing as a whole.

8 Appendix

8.1 FORTRAN code for predicting in GEOS-Chem

Listing 1: XGBoost Prediction Subroutine

```
SUBROUTINE xgb_pred_J(State_Chm, State_Grid, State_Met)

  USE State_Chm_Mod,    ONLY : ChmState
  USE State_Met_Mod,    ONLY : MetState
  USE State_Grid_Mod,   ONLY : GrdState
  USE CMN_FJX_Mod,      ONLY : NRATJ, L_
  USE TIME_MOD,         ONLY : GET_MONTH, GET_DAY, GET_DAY_OF_YEAR
  USE TIME_MOD,         ONLY : GET_TAU,   GET_YEAR
  USE TOMS_MOD,         ONLY : GET_OVERHEAD_O3
  USE CMN_SIZE_MOD,     ONLY : NDUST
  USE Grid_Registry_Mod
  USE Pressure_Mod
  USE xgb_fortran_api
  USE iso_c_binding
  IMPLICIT NONE

  TYPE(ChmState), INTENT(IN)   :: State_Chm
  TYPE(GrdState), INTENT(IN)   :: State_Grid
  TYPE(MetState), INTENT(IN)   :: State_Met


  ! FOR INIT
  LOGICAL, SAVE                   :: first_time = .TRUE.
  REAL(c_float), ALLOCATABLE      :: xx_carr_small(:,:)
  INTEGER(c_int64_t)              :: xx_dmtrx_len, nrow_dummy
  CHARACTER(LEN=255)              :: xx_fname

  ! LOCAL VARS
  INTEGER(c_int64_t)              :: xx_param_count
  INTEGER(c_int)                  :: xx_option_mask, xx_ntree_limit,
      xx_training
  INTEGER(c_int)                  :: xx_rc
  REAL(c_float), parameter        :: missing_value = -999.0
  TYPE(c_ptr)                     :: xx_dmtrx
TYPE(c_ptr), SAVE             :: xx_booster
INTEGER(c_int64_t)            :: xx_prediction_count, xx_count ! How many grid boxes: NZ * NX * NY
REAL(c_float), ALLOCATABLE    :: xx_carr(:,:)
REAL(fp) :: xx_u0, xx_sza, xx_solf
INTEGER  :: xx_prediction_index, DAY_OF_YR
REAL(fp), POINTER :: ODMDUST  (:,:,:,:,:)

! FOR PREDICTION

INTEGER(c_int64_t)            :: xx_pred_len
TYPE(c_ptr)                   :: xx_cpred

REAL(c_float), POINTER     :: xx_pred(:)

! TEMPORARY LOCAL VARS
INTEGER                          :: xx_lon, xx_lat, xx_lev, i, J, xx_index, i_sum, largest_lon, largest_lat, largest_lev
INTEGER                          :: xx_n
REAL, ALLOCATABLE                :: J_ML(:,:,:,:)
REAL                             :: P0, HyAm, HyBm, Lev, XMID, YMID, POS_ENC, start, finish, largest_value
REAL                             :: TAUCLW_sum_above, TAUCLW_sum_below, TAUCLI_sum_above, TAUCLI_sum_below, CLDF_sum_above, CLDF_sum_below

REAL(fp), POINTER                :: ZPJ      (:,:,:,:)


integer(c_int64_t) :: xx_params, xx_total_preds, xx_nrows, xx_ncols

! FOR FIRST TIME INIT
TYPE(c_ptr), DIMENSION(166), SAVE :: xx_boosters

TYPE ModelInfo
    INTEGER :: modelID
    CHARACTER(LEN=255) :: filePath
    REAL(c_float) :: constant
    CHARACTER(LEN=255) :: species
    CHARACTER(LEN=255) :: predictor
```

```fortran
    REAL(c_float)      :: factor
END TYPE ModelInfo
TYPE(ModelInfo), DIMENSION(166) :: models
ZPJ       => State_Chm%Phot%ZPJ
ODMDUST   => State_Chm%Phot%ODMDUST
xx_param_count = 19

IF (ALLOCATED (J_ML)) DEALLOCATE(J_ML)
IF (ALLOCATED (xx_carr)) DEALLOCATE(xx_carr)

xx_count = 0
DO xx_lon = 1, State_Grid%NX
DO xx_lat = 1, State_Grid%NY
DO xx_lev = 1, State_Grid%NZ
    DAY_OF_YR = GET_DAY_OF_YEAR()
    xx_u0 = State_Met%SUNCOSmid(xx_lon, xx_lat)
    CALL SOLAR_JX(DAY_OF_YR, xx_u0, xx_sza, xx_solf)

    IF (xx_sza < 98) THEN
            xx_count = xx_count + 1
    END IF
END DO
END DO
END DO

WRITE(6,*)'xx_count: ',xx_count
ALLOCATE(xx_carr(xx_param_count, xx_count))
xx_index = 1
DO xx_lon = 1, State_Grid%NX
    DO xx_lat = 1, State_Grid%NY
        DO xx_lev = 1, State_Grid%NZ
            P0       = 1000.0_f8
            HyAm = ( Get_Ap( xx_lev ) + Get_Ap( xx_lev+1 ) ) * 0.5_f8
            HyBm = ( Get_Bp( xx_lev ) + Get_Bp(xx_lev+1 ) ) * 0.5_f8
            Lev = (HyAm/P0) +HyBm
            !WRITE(6,*)'TEST LEV ',Lev
            DAY_OF_YR = GET_DAY_OF_YEAR()
            xx_u0 = State_Met%SUNCOSmid(xx_lon, xx_lat)
            CALL SOLAR_JX(DAY_OF_YR, xx_u0, xx_sza, xx_solf)
```

```

TAUCLI_sum_above = 0.0
TAUCLI_sum_below = 0.0
TAUCLW_sum_above = 0.0
TAUCLW_sum_below = 0.0
CLDF_sum_above = 0.0
CLDF_sum_below = 0.0

DO i_sum = 1, xx_lev - 1
    TAUCLI_sum_below = TAUCLI_sum_below + State_Met%TAUCLI(
       xx_lon, xx_lat, i_sum)
    TAUCLW_sum_below = TAUCLW_sum_below + State_Met%TAUCLW(
       xx_lon, xx_lat, i_sum)
    CLDF_sum_below = CLDF_sum_below + State_Met%CLDF(xx_lon
       , xx_lat, i_sum)
END DO

DO i_sum = xx_lev + 1, State_Grid%NZ
    TAUCLI_sum_above = TAUCLI_sum_above + State_Met%TAUCLI(
       xx_lon, xx_lat, i_sum)
    TAUCLW_sum_above = TAUCLW_sum_above + State_Met%TAUCLW(
       xx_lon, xx_lat, i_sum)
    CLDF_sum_above = CLDF_sum_above + State_Met%CLDF(xx_lon
       , xx_lat, i_sum)
END DO
IF (xx_sza < 98) THEN
    xx_carr(1, xx_index) = Lev
    xx_carr(2, xx_index) = State_Met%SUNCOSmid(xx_lon,
       xx_lat)
    xx_carr(3, xx_index) = State_Met%UVALBEDO(xx_lon,
       xx_lat)
    xx_carr(4, xx_index) = GET_OVERHEAD_O3(State_Chm,
       xx_lon, xx_lat)
    xx_carr(5, xx_index) = xx_sza
    xx_carr(6, xx_index) = State_Met%PMid(xx_lon, xx_lat,
       xx_lev)
    xx_carr(7, xx_index) = State_Met%T(xx_lon, xx_lat,
       xx_lev)
    xx_carr(8, xx_index) = TAUCLW_sum_above
```

```
                xx_carr(9, xx_index) = TAUCLW_sum_below
                xx_carr(10, xx_index) = TAUCLI_sum_above
                xx_carr(11, xx_index) = TAUCLI_sum_below
                xx_carr(12, xx_index) = CLDF_sum_above
                xx_carr(13, xx_index) = CLDF_sum_below
                xx_carr(14, xx_index) = State_Met%AIRDEN(xx_lon, xx_lat, xx_lev)
                xx_carr(15, xx_index) = State_Met%CLDF(xx_lon, xx_lat, xx_lev)
                xx_carr(16, xx_index) = State_Met%TAUCLI(xx_lon, xx_lat, xx_lev)
                xx_carr(17, xx_index) = State_Met%TAUCLW(xx_lon, xx_lat, xx_lev)
                xx_carr(18, xx_index) =  ODMDUST(xx_lon, xx_lat, xx_lev, State_Chm%Phot%IWV1000, 1)
                xx_carr(19, xx_index) =  ODMDUST(xx_lon, xx_lat, xx_lev, State_Chm%Phot%IWV1000, 7)
                xx_index = xx_index + 1
            END IF
        END DO
    END DO
END DO

ALLOCATE(J_ML(State_Grid%NX, State_Grid%NY, State_Grid%NZ, 166))
J_ML = 0.0

xx_rc = XGDMatrixCreateFromMat_f(xx_carr, xx_count, xx_param_count, missing_value, xx_dmtrx)
IF (xx_rc/= 0) THEN
    WRITE(6,*) 'Error in creating DMatrix in Run', xx_rc
END IF

IF (ALLOCATED (xx_carr)) DEALLOCATE(xx_carr)



IF (first_time) THEN
    first_time = .FALSE.
    DO J = 1, 166
```

```
        xx_fname = TRIM(models(J)%filePath)

        xx_rc = XGBoosterCreate_f(c_null_ptr, xx_dmtrx_len, xx_boosters(J))
        IF (xx_rc/= 0) THEN
                WRITE(6,*) 'Error in Creating Booster in Initialising', xx_rc
        END IF
        !WRITE(6,*) 'Reading File for XGBoost: ', xx_fname
        xx_rc = XGBoosterLoadModel_f(xx_boosters(J), xx_fname)
        WRITE(6,*) 'Initialised Model: ', models(J)%species

        IF (xx_rc/= 0) THEN
                WRITE(6,*) 'Error in Loading Model in Initialising', xx_rc
                WRITE(6,*) 'Error at: ', xx_fname
                WRITE(6,*) 'Error for species: ', models(J)%species
        END IF
    END DO
END IF


CALL cpu_time(start)
!$OMP PARALLEL DO SHARED(J_ML, xx_dmtrx, models, State_Grid, State_Met, ZPJ, xx_boosters) &
!$OMP PRIVATE(J, xx_lon, xx_lat, xx_lev, xx_index, DAY_OF_YR, xx_u0, xx_sza, xx_solf) &
!$OMP PRIVATE(xx_fname, xx_param_count, xx_option_mask, xx_ntree_limit, xx_training) &
!$OMP PRIVATE(xx_dmtrx_len, xx_pred, xx_cpred, xx_booster, xx_rc) &
!$OMP SCHEDULE(DYNAMIC)

DO J = 1, 166

    xx_option_mask = 0
    xx_ntree_limit = 0
    xx_training = 0
    xx_dmtrx_len = 0

```

```
xx_rc = XGBoosterPredict_f(xx_boosters(J), xx_dmtrx, xx_option_mask
   , xx_ntree_limit, xx_training, xx_pred_len, xx_cpred)
IF (xx_rc/= 0) THEN
        WRITE(6,*) 'Error in XGBooster Predicting in Run', xx_rc
END IF

IF (ASSOCIATED(xx_pred)) NULLIFY(xx_pred)

!write(6,*) 'PREDICTION LENGHT: ', xx_pred_len
call c_f_pointer(xx_cpred, xx_pred, [xx_pred_len])

IF (.NOT. ASSOCIATED(xx_pred)) THEN
    WRITE(6,*) 'Error: xx_pred is not associated.'
END IF

xx_index = 1
DO xx_lon = 1, State_Grid%NX
DO xx_lat = 1, State_Grid%NY
DO xx_lev = 1, State_Grid%NZ
    DAY_OF_YR = GET_DAY_OF_YEAR()
    xx_u0 = State_Met%SUNCOSmid(xx_lon, xx_lat)
    CALL SOLAR_JX(DAY_OF_YR, xx_u0, xx_sza, xx_solf)

    IF (xx_sza < 98) THEN

        J_ML(xx_lon, xx_lat, xx_lev, J) = (EXP(xx_pred(xx_index)) -
            models(J)%constant) * models(J)%factor

        xx_index=xx_index+1

    END IF
END DO
END DO
END DO

IF (ASSOCIATED(xx_pred)) NULLIFY(xx_pred)
!IF (ASSOCIATED(xx_cpred)) NULLIFY(xx_cpred)
xx_cpred = c_null_ptr

```

```
      !xx_rc = XGBoosterFree_f(xx_booster)



      DO xx_lon = 1, State_Grid%NX
      DO xx_lat = 1, State_Grid%NY
      DO xx_lev = 1, State_Grid%NZ
              ZPJ(xx_lev, J, xx_lon, xx_lat) = J_ML(xx_lon, xx_lat,
                 xx_lev, J)
      END DO
      END DO
      END DO

  END DO

  !$OMP END PARALLEL DO
  DO xx_lon = 1, State_Grid%NX
  DO xx_lat = 1, State_Grid%NY
  DO xx_lev = 1, State_Grid%NZ
  DO J = 1, 166
      IF (ZPJ(xx_lev, J, xx_lon, xx_lat) < 0.0) THEN
              ZPJ(xx_lev, J, xx_lon, xx_lat) = 0.0
      END IF
  END DO
  END DO
  END DO
  END DO


  CALL cpu_time(finish)
  WRITE(6,*)'Time to Predict: ',finish-start
  xx_rc = XGDMatrixFree_f(xx_dmtrx)
  IF (ALLOCATED (J_ML)) DEALLOCATE(J_ML)

  ODMDUST   => NULL()
  ZPJ       => NULL()

END SUBROUTINE xgb_pred_J
```

8.2 Tables Containing Model Statistics

Table 7: The same as Table 4 but for all 111 photolysis rate models including the separate channel machine learning models, where the metrics are calculated between Fast-JX and predicted rates for the validation data. In descending order for the R^2 in the linear space.

	Linear Space			Log Space		
Species ID	R^2	NRMSE	NMAE	R^2	NRMSE	NMAE
NO	0.997	0.217	0.056	1.000	0.013	0.007
HCFC142b	0.996	0.188	0.055	0.999	0.013	0.007
CFC12	0.996	0.187	0.055	0.999	0.014	0.007
CFC114	0.996	0.186	0.055	0.999	0.014	0.007
HNO3	0.996	0.171	0.054	0.997	0.024	0.009
CH3Cl	0.996	0.185	0.055	0.999	0.014	0.007
HCFC22	0.996	0.186	0.055	0.999	0.011	0.006
R4N2	0.996	0.178	0.055	0.997	0.026	0.010
CH2Cl2	0.996	0.185	0.055	0.999	0.015	0.008
HCFC123	0.996	0.187	0.055	0.999	0.015	0.008
ETNO3	0.996	0.176	0.055	0.997	0.025	0.010
NPRNO3	0.996	0.173	0.055	0.997	0.026	0.009
MENO3	0.996	0.177	0.055	0.997	0.025	0.010
IPRNO3	0.996	0.175	0.055	0.997	0.026	0.010
PAN	0.996	0.189	0.056	0.997	0.024	0.009
HCFC141b	0.996	0.185	0.056	0.999	0.015	0.008
CFC113	0.996	0.184	0.056	0.999	0.015	0.008
CFC115	0.996	0.185	0.056	0.999	0.013	0.007
O2	0.996	0.207	0.057	0.999	0.014	0.008
N2O5	0.996	0.121	0.054	0.996	0.030	0.010
N2O	0.996	0.187	0.056	0.999	0.014	0.007
HNO4	0.996	0.128	0.054	0.995	0.031	0.009
CFC11	0.996	0.187	0.057	0.999	0.016	0.008

Continued on next page

Table 7 – validation continued from previous page

	Linear Space			Log Space		
Species ID	R^2	NRMSE	NMAE	R^2	NRMSE	NMAE
Acet-b	0.996	0.196	0.065	0.997	0.020	0.008
H2402	0.996	0.206	0.058	0.997	0.021	0.009
ClO	0.996	0.149	0.059	0.997	0.036	0.014
H1211	0.996	0.207	0.058	0.997	0.021	0.009
HAC	0.996	0.128	0.056	0.996	0.025	0.009
CH3CCl3	0.996	0.188	0.057	0.999	0.016	0.008
H1301	0.996	0.203	0.058	0.999	0.017	0.009
CH3Br	0.996	0.204	0.058	0.999	0.017	0.010
CHBr3	0.996	0.231	0.059	0.997	0.025	0.010
CCl4	0.996	0.199	0.058	0.999	0.017	0.009
H2O2	0.996	0.110	0.054	0.996	0.027	0.009
PIP	0.996	0.110	0.054	0.996	0.027	0.009
CH2Br2	0.996	0.223	0.059	0.999	0.021	0.011
CH2ICl	0.996	0.109	0.055	0.996	0.033	0.011
OCS	0.996	0.237	0.061	0.999	0.017	0.009
GLYC	0.995	0.113	0.058	0.996	0.029	0.010
MEK	0.995	0.112	0.057	0.996	0.028	0.010
O3O1D	0.995	0.245	0.063	0.996	0.032	0.013
O3	0.995	0.245	0.063	0.996	0.032	0.013
IDHDP	0.995	0.096	0.054	0.996	0.028	0.009
HMHP	0.995	0.096	0.054	0.959	0.026	0.008
MP	0.995	0.096	0.054	0.996	0.027	0.009
ClNO3a	0.995	0.106	0.056	0.996	0.029	0.009
INPB	0.995	0.095	0.054	0.996	0.027	0.009
ClNO3b	0.995	0.162	0.064	0.996	0.027	0.009
CF3I	0.995	0.152	0.066	0.997	0.029	0.010
CH2IBr	0.995	0.093	0.055	0.996	0.037	0.012

Continued on next page

Table 7 – validation continued from previous page

	Linear Space			Log Space		
Species ID	R^2	NRMSE	NMAE	R^2	NRMSE	NMAE
CH3I	0.995	0.250	0.063	0.996	0.028	0.010
MPN	0.994	0.111	0.060	0.996	0.027	0.009
ActAld	0.994	0.122	0.063	0.997	0.028	0.010
IDN	0.994	0.098	0.057	0.996	0.027	0.009
MONITS	0.994	0.098	0.057	0.996	0.026	0.008
HPETHNL	0.994	0.089	0.056	0.996	0.031	0.010
Acet-a	0.994	0.191	0.076	0.996	0.026	0.011
MVK	0.993	0.086	0.054	0.996	0.027	0.009
ClNO2	0.993	0.084	0.055	0.996	0.036	0.011
RCHO	0.993	0.090	0.057	0.996	0.031	0.010
MVKN	0.993	0.093	0.058	0.996	0.031	0.010
PROPNN	0.993	0.091	0.058	0.996	0.031	0.010
NPHEN	0.993	0.091	0.058	0.996	0.031	0.010
Glyxlb	0.993	0.083	0.054	0.996	0.028	0.009
Glyxlc	0.993	0.079	0.053	0.996	0.029	0.009
H2COa	0.992	0.084	0.055	0.996	0.030	0.010
Cl2O2	0.992	0.081	0.054	0.996	0.043	0.013
CH2I2	0.992	0.083	0.056	0.996	0.049	0.016
ETHLN	0.992	0.085	0.057	0.996	0.033	0.011
MCRHNB	0.991	0.084	0.057	0.996	0.032	0.010
H2COb	0.991	0.079	0.054	0.996	0.030	0.009
MCRHN	0.991	0.084	0.057	0.996	0.036	0.011
BALD	0.991	0.083	0.057	0.996	0.029	0.009
HOCl	0.991	0.081	0.056	0.996	0.035	0.011
MGLY	0.991	0.086	0.056	0.996	0.036	0.011
PYAC	0.991	0.086	0.056	0.996	0.036	0.011
MVKHC	0.991	0.086	0.056	0.996	0.036	0.011

Continued on next page

Table 7 – validation continued from previous page

	Linear Space			Log Space		
Species ID	R^2	NRMSE	NMAE	R^2	NRMSE	NMAE
IONO	0.991	0.081	0.056	0.996	0.044	0.014
Glyxla	0.991	0.081	0.055	0.996	0.031	0.010
BrO	0.990	0.082	0.057	0.996	0.060	0.019
SO4	0.990	0.312	0.078	0.989	0.023	0.003
ICN	0.990	0.082	0.057	0.996	0.036	0.011
BrNO3	0.990	0.081	0.056	0.995	0.042	0.012
HPALD1	0.990	0.082	0.057	0.996	0.036	0.011
HPALD2	0.990	0.082	0.057	0.996	0.035	0.011
IONO2	0.990	0.081	0.056	0.996	0.051	0.016
MACR	0.990	0.082	0.057	0.996	0.024	0.008
MCRENOL	0.990	0.082	0.057	0.996	0.035	0.011
Cl2	0.990	0.082	0.057	0.995	0.044	0.013
I2O2	0.990	0.082	0.057	0.995	0.065	0.019
I2O4	0.990	0.082	0.057	0.995	0.065	0.019
HNO2	0.989	0.082	0.057	0.995	0.042	0.013
I2O3	0.989	0.082	0.057	0.995	0.065	0.019
NO2	0.989	0.082	0.057	0.995	0.051	0.015
OClO	0.989	0.082	0.057	0.995	0.071	0.021
HOBr	0.989	0.082	0.057	0.995	0.045	0.013
O3O3P	0.988	0.087	0.060	0.994	0.041	0.012
BrNO2	0.988	0.084	0.058	0.995	0.052	0.015
HOI	0.988	0.084	0.059	0.995	0.054	0.016
INO	0.987	0.085	0.059	0.995	0.066	0.019
BrCl	0.987	0.086	0.060	0.995	0.055	0.016
IO	0.986	0.087	0.060	0.995	0.090	0.026
Br2	0.985	0.090	0.062	0.994	0.067	0.019
ClOO	0.984	0.091	0.063	0.994	0.114	0.033

Continued on next page

Table 7 – validation continued from previous page

	Linear Space			Log Space		
Species ID	R^2	NRMSE	NMAE	R^2	NRMSE	NMAE
ICl	0.984	0.091	0.063	0.994	0.065	0.019
IBr	0.984	0.092	0.064	0.994	0.078	0.023
NO3	0.984	0.093	0.064	0.995	0.095	0.028
I2	0.984	0.092	0.064	0.994	0.092	0.027
OIO	0.983	0.092	0.064	0.994	0.095	0.028
NIT	-2.678	4.292	0.333	0.979	0.063	0.018
NITs	-4.104	4.188	0.127	0.981	0.061	0.013

Table 8: The same as Table 5 but for all 105 photolysis rates representing all 161 photolysed species, where the metrics are calculated between Fast-JX and predicted rates from implementation in GEOS-Chem. In descending order for the R^2 for the January, where species marked with a star (*) are combined from separate channels

	01 January 2021			01 June 2021		
Species ID	R^2	NRMSE	NMAE	R^2	NRMSE	NMAE
CH2O*	0.988	0.149	0.078	0.981	0.188	0.110
GLYX*	0.988	0.146	0.075	0.984	0.171	0.095
ClNO3*	0.988	0.144	0.076	0.982	0.174	0.101
MVK	0.987	0.178	0.088	0.987	0.181	0.086
Cl2O2	0.987	0.151	0.084	0.986	0.156	0.086
HOCl	0.987	0.153	0.085	0.985	0.160	0.087
BrO	0.986	0.154	0.085	0.985	0.161	0.088
CH2I2	0.986	0.159	0.088	0.986	0.163	0.088
BALD	0.986	0.158	0.087	0.985	0.165	0.090
MCRHN	0.986	0.160	0.088	0.986	0.166	0.090
I2O2	0.986	0.150	0.084	0.985	0.155	0.085
I2O4	0.986	0.150	0.084	0.985	0.155	0.085

Continued on next page

Table 8 – tropospheric metrics continued from previous page

	01 January 2021			01 June 2021		
Species ID	R^2	NRMSE	NMAE	R^2	NRMSE	NMAE
MCRHNB	0.986	0.160	0.088	0.986	0.166	0.090
ICN	0.986	0.153	0.084	0.985	0.161	0.088
ClNO2	0.986	0.156	0.086	0.985	0.160	0.088
HPALD1	0.986	0.153	0.084	0.985	0.161	0.088
MACR	0.986	0.153	0.084	0.985	0.161	0.088
HPALD2	0.986	0.153	0.084	0.985	0.161	0.088
HMHP	0.986	0.159	0.088	0.986	0.162	0.088
MP	0.986	0.159	0.088	0.986	0.162	0.088
IDHDP	0.986	0.159	0.088	0.986	0.162	0.088
MCRENOL	0.986	0.153	0.084	0.985	0.161	0.088
IONO	0.986	0.153	0.084	0.985	0.161	0.089
ETHLN	0.986	0.163	0.090	0.986	0.167	0.090
INPB	0.986	0.163	0.089	0.986	0.164	0.089
Cl2	0.986	0.152	0.084	0.985	0.160	0.088
IONO2	0.986	0.153	0.085	0.984	0.161	0.089
H2O2	0.986	0.164	0.089	0.986	0.166	0.091
PIP	0.986	0.164	0.089	0.986	0.166	0.091
CH2IBr	0.986	0.166	0.091	0.986	0.167	0.091
HPETHNL	0.986	0.171	0.094	0.986	0.171	0.092
BrNO3	0.986	0.152	0.083	0.984	0.158	0.088
RCHO	0.985	0.173	0.095	0.986	0.172	0.093
HNO2	0.985	0.155	0.085	0.983	0.164	0.091
I2O3	0.985	0.153	0.085	0.984	0.159	0.089
OClO	0.985	0.154	0.085	0.983	0.162	0.091
PROPNN	0.985	0.175	0.096	0.986	0.174	0.094
NPHEN	0.985	0.175	0.096	0.986	0.174	0.094
MONITS	0.985	0.176	0.095	0.986	0.173	0.094

Continued on next page

Table 8 – tropospheric metrics continued from previous page

	01 January 2021			01 June 2021		
Species ID	R^2	NRMSE	NMAE	R^2	NRMSE	NMAE
IDN	0.985	0.176	0.095	0.986	0.173	0.094
CH2ICl	0.985	0.172	0.093	0.985	0.171	0.094
NO2	0.985	0.155	0.086	0.983	0.164	0.092
HOBr	0.985	0.154	0.086	0.983	0.161	0.090
MVKN	0.985	0.180	0.098	0.986	0.177	0.096
N2O5	0.985	0.172	0.093	0.985	0.169	0.092
HNO4	0.985	0.155	0.087	0.984	0.158	0.088
HAC	0.985	0.180	0.098	0.986	0.176	0.096
HOI	0.984	0.157	0.087	0.982	0.165	0.092
MEK	0.984	0.209	0.107	0.986	0.203	0.101
NPRNO3	0.984	0.187	0.101	0.985	0.182	0.099
PAN	0.984	0.184	0.099	0.984	0.181	0.098
MGLY	0.984	0.175	0.086	0.983	0.179	0.090
PYAC	0.984	0.175	0.086	0.983	0.179	0.090
MVKHC	0.984	0.175	0.086	0.983	0.179	0.090
CH3I	0.983	0.191	0.102	0.984	0.186	0.102
BrNO2	0.983	0.161	0.089	0.981	0.168	0.095
BrCl	0.983	0.161	0.090	0.981	0.169	0.095
IPRNO3	0.982	0.200	0.106	0.984	0.195	0.105
INO	0.982	0.164	0.091	0.980	0.172	0.097
GLYC	0.982	0.206	0.111	0.984	0.197	0.106
MPN	0.982	0.208	0.112	0.984	0.199	0.107
ETNO3	0.982	0.207	0.109	0.983	0.201	0.108
IO	0.982	0.166	0.092	0.979	0.174	0.098
MENO3	0.981	0.211	0.111	0.983	0.205	0.111
NITs	0.981	0.215	0.112	0.983	0.207	0.111
HNO3	0.981	0.211	0.111	0.983	0.203	0.110

Continued on next page

Table 8 – tropospheric metrics continued from previous page

	01 January 2021			01 June 2021		
Species ID	R^2	NRMSE	NMAE	R^2	NRMSE	NMAE
R4N2	0.981	0.214	0.113	0.982	0.206	0.112
NIT	0.981	0.228	0.112	0.977	0.308	0.131
CHBr3	0.980	0.225	0.118	0.982	0.220	0.117
Br2	0.979	0.174	0.096	0.977	0.183	0.103
ClOO	0.978	0.178	0.098	0.975	0.187	0.105
ClO	0.978	0.245	0.129	0.982	0.231	0.121
ICl	0.978	0.179	0.099	0.975	0.188	0.105
NO3	0.978	0.182	0.099	0.974	0.193	0.108
IBr	0.977	0.181	0.100	0.974	0.190	0.107
I2	0.977	0.182	0.100	0.974	0.191	0.108
OIO	0.977	0.183	0.100	0.974	0.192	0.108
ALD2*	0.973	0.298	0.168	0.959	0.391	0.221
O3O1D	0.971	0.315	0.155	0.975	0.303	0.149
O3	0.971	0.315	0.155	0.975	0.303	0.149
O3O3P	0.970	0.211	0.129	0.975	0.189	0.111
H1211	0.917	0.727	0.173	0.929	0.724	0.171
H2402	0.905	0.912	0.178	0.920	0.902	0.177
CFC12	0.899	4.745	0.393	0.916	4.698	0.397
ACET*	0.899	0.604	0.336	0.878	0.689	0.371
CFC114	0.898	4.765	0.394	0.915	4.710	0.397
CH3Cl	0.898	4.764	0.394	0.916	4.697	0.397
HCFC22	0.898	4.764	0.394	0.916	4.695	0.369
HCFC142b	0.898	4.768	0.394	0.915	4.725	0.399
CH2Cl2	0.898	4.768	0.937	0.915	4.703	0.398
CFC113	0.897	4.785	0.394	0.915	4.715	0.397
HCFC141b	0.897	4.773	0.394	0.915	4.691	0.398
N2O	0.897	4.755	0.394	0.916	4.663	0.397

Continued on next page

Table 8 – tropospheric metrics continued from previous page

	01 January 2021			01 June 2021		
Species ID	R^2	NRMSE	NMAE	R^2	NRMSE	NMAE
HCFC123	0.896	4.794	0.395	0.914	4.722	0.398
CFC115	0.896	4.798	0.395	0.914	4.731	0.399
CFC11	0.896	4.797	0.395	0.914	4.722	0.398
CH3CCl3	0.894	4.827	0.396	0.913	4.746	0.399
CCl4	0.887	5.007	0.407	0.908	4.883	0.406
O2	0.883	5.119	0.410	0.905	4.982	0.408
H1301	0.880	4.981	0.404	0.902	4.895	0.407
CH3Br	0.880	5.156	0.412	0.903	5.008	0.411
CH2Br2	0.876	2.137	0.294	0.901	2.216	0.296
OCS	0.854	5.415	0.423	0.884	5.230	0.421
NO	0.761	7.845	0.514	0.811	7.513	0.502
SO4	-0.697	1.560	1.000	-0.711	1.551	1.000

Table 9: The same as Table 8 but for the full vertical column.

	01 January 2021			01 June 2021		
Species ID	R^2	NRMSE	NMAE	R^2	NRMSE	NMAE
NO	0.992	0.442	0.095	0.993	0.414	0.076
HCFC142b	0.988	0.440	0.116	0.993	0.356	0.079
CFC12	0.988	0.441	0.117	0.992	0.355	0.079
CH2O*	0.988	0.161	0.074	0.986	0.171	0.087
HCFC22	0.987	0.449	0.120	0.992	0.361	0.082
GLYX*	0.987	0.160	0.074	0.988	0.160	0.080
CFC114	0.987	0.452	0.121	0.992	0.359	0.081
CH3Cl	0.987	0.450	0.121	0.992	0.359	0.081
ClNO3*	0.987	0.280	0.087	0.992	0.224	0.079
ALD2*	0.986	0.273	0.101	0.989	0.256	0.103

Continued on next page

Table 9 – full column metrics continued from previous page

	01 January 2021			01 June 2021		
Species ID	R^2	NRMSE	NMAE	R^2	NRMSE	NMAE
CH2Cl2	0.986	0.464	0.125	0.992	0.366	0.084
HCFC123	0.986	0.468	0.126	0.992	0.369	0.085
HCFC141b	0.986	0.468	0.127	0.991	0.368	0.085
CFC113	0.985	0.469	0.127	0.991	0.369	0.086
IONO	0.985	0.168	0.087	0.988	0.153	0.075
HOCl	0.985	0.170	0.088	0.988	0.153	0.074
BrO	0.985	0.168	0.085	0.987	0.158	0.077
MACR	0.985	0.167	0.084	0.987	0.158	0.078
ICN	0.985	0.167	0.085	0.987	0.158	0.078
HPALD2	0.985	0.167	0.085	0.987	0.158	0.078
HPALD1	0.985	0.167	0.085	0.987	0.158	0.078
MCRENOL	0.985	0.167	0.085	0.987	0.158	0.078
IONO2	0.985	0.167	0.085	0.987	0.156	0.077
BrNO3	0.985	0.167	0.087	0.988	0.151	0.075
Cl2O2	0.985	0.181	0.092	0.991	0.146	0.070
CFC115	0.985	0.476	0.129	0.991	0.373	0.088
Cl2	0.985	0.167	0.084	0.987	0.158	0.079
I2O2	0.985	0.167	0.085	0.987	0.155	0.076
I2O4	0.985	0.167	0.085	0.987	0.155	0.076
ClNO2	0.985	0.196	0.097	0.991	0.153	0.071
ACET*	0.985	0.419	0.121	0.989	0.370	0.108
BALD	0.985	0.177	0.090	0.988	0.160	0.076
NITs	0.985	0.322	0.112	0.986	0.311	0.111
HNO2	0.985	0.167	0.084	0.986	0.161	0.081
MVK	0.985	0.200	0.094	0.990	0.167	0.071
CH2I2	0.985	0.182	0.093	0.989	0.157	0.073
MCRHNB	0.984	0.181	0.092	0.988	0.161	0.075

Continued on next page

Table 9 – full column metrics continued from previous page

	01 January 2021			01 June 2021		
Species ID	R^2	NRMSE	NMAE	R^2	NRMSE	NMAE
NO2	0.984	0.168	0.085	0.985	0.163	0.082
OClO	0.984	0.168	0.085	0.985	0.162	0.082
I2O3	0.984	0.168	0.085	0.986	0.158	0.080
MCRHN	0.984	0.181	0.092	0.988	0.162	0.076
N2O	0.984	0.485	0.132	0.990	0.382	0.089
HOBr	0.984	0.168	0.086	0.986	0.159	0.080
NIT	0.984	0.340	0.112	0.980	0.462	0.131
ETHLN	0.984	0.187	0.094	0.988	0.164	0.077
MGLY	0.984	0.184	0.087	0.987	0.169	0.079
PYAC	0.984	0.184	0.087	0.987	0.169	0.079
MVKHC	0.984	0.184	0.087	0.987	0.169	0.079
HOI	0.983	0.171	0.086	0.984	0.165	0.084
BrNO2	0.983	0.173	0.088	0.985	0.162	0.082
CFC11	0.983	0.503	0.138	0.990	0.390	0.093
HNO3	0.983	0.485	0.137	0.990	0.377	0.094
BrCl	0.982	0.175	0.088	0.983	0.169	0.087
HPETHNL	0.982	0.223	0.106	0.990	0.175	0.078
INO	0.982	0.176	0.090	0.984	0.166	0.084
RCHO	0.982	0.220	0.106	0.989	0.177	0.079
CH2IBr	0.982	0.255	0.113	0.991	0.186	0.080
N2O5	0.982	0.360	0.125	0.990	0.264	0.088
R4N2	0.981	0.515	0.142	0.989	0.403	0.099
IO	0.981	0.179	0.090	0.982	0.173	0.089
INPB	0.981	0.264	0.113	0.991	0.193	0.079
HMHP	0.981	0.272	0.113	0.991	0.195	0.079
IDHDP	0.981	0.272	0.113	0.991	0.195	0.080
MP	0.981	0.272	0.113	0.991	0.195	0.079

Continued on next page

Table 9 – full column metrics continued from previous page

	01 January 2021			01 June 2021		
Species ID	R^2	NRMSE	NMAE	R^2	NRMSE	NMAE
CH3CCl3	0.981	0.522	0.144	0.989	0.402	0.097
PROPNN	0.981	0.225	0.106	0.988	0.187	0.082
NPHEN	0.981	0.225	0.106	0.988	0.187	0.082
MENO3	0.981	0.519	0.144	0.989	0.405	0.101
H2O2	0.981	0.323	0.121	0.991	0.231	0.085
PIP	0.981	0.323	0.121	0.991	0.231	0.085
CH2ICl	0.980	0.319	0.124	0.990	0.229	0.088
ETNO3	0.980	0.524	0.146	0.988	0.409	0.103
MVKN	0.980	0.238	0.109	0.987	0.197	0.085
O2	0.980	0.598	0.159	0.989	0.448	0.108
IPRNO3	0.980	0.524	0.146	0.988	0.407	0.103
NPRNO3	0.980	0.518	0.146	0.988	0.402	0.102
Br2	0.980	0.185	0.093	0.981	0.180	0.092
HNO4	0.979	0.395	0.129	0.990	0.284	0.090
MEK	0.979	0.326	0.129	0.989	0.243	0.092
ClOO	0.979	0.188	0.094	0.980	0.182	0.094
ICl	0.979	0.189	0.095	0.980	0.183	0.094
GLYC	0.979	0.333	0.133	0.989	0.250	0.096
IBr	0.978	0.191	0.096	0.979	0.186	0.095
NO3	0.978	0.192	0.095	0.979	0.188	0.096
CCl4	0.978	0.581	0.158	0.988	0.439	0.107
IDN	0.978	0.273	0.115	0.987	0.219	0.088
I2	0.978	0.191	0.096	0.979	0.186	0.096
MONITS	0.978	0.274	0.115	0.987	0.219	0.088
PAN	0.978	0.586	0.159	0.988	0.444	0.112
OIO	0.978	0.192	0.096	0.979	0.187	0.096
HAC	0.978	0.396	0.138	0.989	0.287	0.098

Continued on next page

Table 9 – full column metrics continued from previous page

	01 January 2021			01 June 2021		
Species ID	R^2	NRMSE	NMAE	R^2	NRMSE	NMAE
ClO	0.977	0.462	0.155	0.988	0.341	0.113
H2402	0.976	0.636	0.168	0.987	0.486	0.118
CH3Br	0.976	0.627	0.167	0.987	0.474	0.116
MPN	0.976	0.319	0.129	0.985	0.258	0.099
H1301	0.975	0.633	0.169	0.986	0.480	0.118
H1211	0.975	0.654	0.173	0.986	0.496	0.122
CHBr3	0.974	0.721	0.181	0.985	0.555	0.132
O3O1D	0.973	0.736	0.184	0.984	0.582	0.140
O3	0.973	0.736	0.184	0.984	0.582	0.140
CH2Br2	0.973	0.713	0.182	0.985	0.541	0.129
OCS	0.972	0.743	0.185	0.984	0.573	0.134
CH3I	0.972	0.738	0.175	0.982	0.599	0.137
O3O3P	0.341	1.930	0.464	0.365	1.921	0.441
SO4	-0.305	1.478	0.845	-0.342	1.492	0.864

8.3 Timings Predictions

Table 10: Time taken (in seconds) for each time step with varying numbers of ML models from 01-06-2021. These were computed on Viking 2 using 430 GB memory and 86 CPUs.

Time Step	FJX	2 Models	4 Models	8 Models	16 Models	32 Models	161 Models
1	3.38	32.14	78.84	156.61	374.28	941.69	8718.19
2	3.42	34.60	73.15	133.59	316.80	929.27	8860.46
3	3.43	41.04	79.58	119.65	356.33	1034.57	8452.78
4	3.39	40.07	87.91	125.19	325.08	941.12	8335.84
5	3.40	35.12	76.87	151.99	311.40	968.34	8470.33
6	3.41	29.80	65.05	137.58	385.16	940.74	8403.83
Avg.	3.41	35.46	76.90	137.44	344.84	959.29	8540.24

References

Akiba, T., Sano, S., Yanase, T., Ohta, T., and Koyama, M. (2019). Optuna: A next-generation hyperparameter optimization framework. *CoRR*, abs/1907.10902.

Alexander, D. L. J., Tropsha, A., and Winkler, D. A. (2015). Beware of r2: Simple, unambiguous assessment of the prediction accuracy of qsar and qspr models. *Journal of Chemical Information and Modeling*, 55(7):1316–1322. PMID: 26099013.

Bates, K. H., Jacob, D. J., Li, K., Ivatt, P. D., Evans, M. J., Yan, Y., and Lin, J. (2021). Development and evaluation of a new compact mechanism for aromatic oxidation in atmospheric models. *Atmospheric Chemistry and Physics*, 21(24):18351–18374.

Bey, I., Jacob, D. J., Yantosca, R. M., Logan, J. A., Field, B. D., Fiore, A. M., Li, Q., Liu, H. Y., Mickley, L. J., and Schultz, M. G. (2001). Global modeling of tropospheric chemistry with assimilated meteorology: Model description and evaluation. *Journal of Geophysical Research Atmospheres*, 106:23073–23095. Original model description.

Bian, H. and Prather, M. J. (2002). Fast-j2: Accurate simulation of stratospheric photolysis in global chemical models.

Bodnar, C., Bruinsma, W. P., Lucic, A., Stanley, M., Brandstetter, J., Garvan, P., Riechert, M., Weyn, J., Dong, H., Vaughan, A., et al. (2024). Aurora: A foundation model of the atmosphere. *arXiv preprint arXiv:2405.13063*.

Brady, M. (2024). XGBoost and GEOS-Chem photolysis parameterisation models and scripts. Software.

Breiman, L. (1996). Bagging predictors. *Machine Learning*, 24(2):123–140.

Breiman, L. (2001). Random forests.

Brenowitz, N. D., Beucler, T., Pritchard, M., and Bretherton, C. S. (2020). Interpreting and stabilizing machine-learning parametrizations of convection. *Journal of the Atmospheric Sciences*, 77(12):4357 – 4375.

Castelvecchi, D. (2016). Can we open the black box of AI? *Nature*, 538(7623):20–23.

Chai, T. and Draxler, R. R. (2014). Root mean square error (rmse) or mean absolute error (mae)?–arguments against avoiding rmse in the literature. *Geoscientific model development*, 7(3):1247–1250.

Chen, T. and Guestrin, C. (2016). Xgboost. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794. ACM.

Choquette, J. (2023). Nvidia hopper h100 gpu: Scaling performance. *IEEE Micro*, 43(3):9–17.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P. (2011). Natural language processing (almost) from scratch. *J. Mach. Learn. Res.*, 12(null):2493–2537.

Dagum, L. and Menon, R. (1998). Openmp: An industry-standard api for shared-memory programming. *IEEE Computational Science and Engineering*, 5(1):46–55.

Eastham, S. D., Keith, D. W., and Barrett, S. R. (2018). Mortality tradeoff between air quality and skin cancer from changes in stratospheric ozone. *Environmental Research Letters*, 13(3):034035.

Eastham, S. D., Weisenstein, D. K., and Barrett, S. R. (2014). Development and evaluation of the unified tropospheric–stratospheric chemistry extension (ucx) for the global chemistry-transport model geos-chem. *Atmospheric Environment*, 89:52–63.

Emmons, L. K., Walters, S., Hess, P. G., Lamarque, J.-F., Pfister, G. G., Fillmore, D., Granier, C., Guenther, A., Kinnison, D., Laepple, T., et al. (2010). Description and evaluation of the model for ozone and related chemical tracers, version 4 (mozart-4). *Geoscientific Model Development*, 3(1):43–67.

ENVIRON, U. G. (2008). Comprehensive air quality model with extensions (camx). version 4.50. *ENVIRON International Corporation, Novato.*

Fang, L., Jin, J., Segers, A., Liao, H., Li, K., Xu, B., Han, W., Pang, M., and Lin, H. X. (2023). A gridded air quality forecast through fusing site-available machine learning predictions from rfsml v1.0 and chemical transport model results from geos-chem v13.1.0 using the ensemble kalman filter. *Geoscientific Model Development*, 16:4867–4882.

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5):1189 – 1232.

Galton, F. S. (1886). Regression towards mediocrity in hereditary stature. *The Journal of the Anthropological Institute of Great Britain and Ireland*, 15:246.

Gelaro, R., McCarty, W., Suárez, M. J., Todling, R., Molod, A., Takacs, L., Randles, C. A., Darmenov, A., Bosilovich, M. G., Reichle, R., Wargan, K., Coy, L., Cullather, R., Draper, C., Akella, S., Buchard, V., Conaty, A., da Silva, A. M., Gu, W., Kim, G.-K., Koster, R., Lucchesi, R., Merkova, D., Nielsen, J. E., Partyka, G., Pawson, S., Putman, W., Rienecker, M., Schubert, S. D., Sienkiewicz, M., and Zhao, B. (2017). The modern-era retrospective analysis for research and applications, version 2 (merra-2). *Journal of Climate*, 30(14):5419 – 5454.

GEOS-Chem (2024). Geos-chem classic benchmark timing information. Text file.

GEOS-Chem Community (2023). geoschem/gcclassic: Gcclassic 14.2.2.

GEOS-Chem Documentation (2023). Machine learning related to geos-chem. `https://wiki.seas.harvard.edu/geos-chem/index.php/Machine_learning_related_to_GEOS-Chem`. Accessed: 2024-08-06.

Grell, G. A., Peckham, S. E., Schmitz, R., McKeen, S. A., Frost, G., Skamarock, W. C., and Eder, B. (2005). Fully coupled "online" chemistry within the wrf model. *Atmospheric Environment*, 39(37):6957–6975.

Grundner, A., Beucler, T., Gentine, P., and Eyring, V. (2024). Data-driven equation discovery of a cloud cover parameterization. *Journal of Advances in Modeling Earth Systems*, 16(3):e2023MS003763. e2023MS003763 2023MS003763.

Ho, T. K. (1995). Random decision forests. In *Proceedings of 3rd International Conference on Document Analysis and Recognition*, volume 1, pages 278–282 vol.1.

ICCS Contributors (2024). Ftorch: A library for coupling (py)torch machine learning models to fortran. `https://github.com/Cambridge-ICCS/FTorch`. Accessed: 07/05/2024.

IPCC, Masson-Delmotte, V., Zhai, P., Pirani, A., Connors, S. L., P'ean, C., Berger, S., Caud, N., Chen, Y., Goldfarb, L., Gomis, M. I., Huang, M., Leitzell, K., Lonnoy, E., Matthews, J. B. R., Maycock, T. K., Waterfield, T., Yelekçi, O., Yu, R., and Zhou, B. (2021). *Climate Change 2021: The Physical Science Basis. Contribution of Working Group I to the Sixth Assessment*

Report of the Intergovernmental Panel on Climate Change. Cambridge University Press, Cambridge, United Kingdom and New York, NY, USA. In press.

Keller, C. A. and 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*, 12:1209–1225.

Keller, C. A. and Manyin, M. E. (2022). Xgboost fortran api. contents of the module used to implement xgboost in Fortran.

Kelp, M. M., Tessum, C. W., and Marshall, J. D. (2018). Orders-of-magnitude speedup in atmospheric chemistry modeling through neural network-based emulation.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Pereira, F., Burges, C., Bottou, L., and Weinberger, K., editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc.

Lam, R., Sanchez-Gonzalez, A., Willson, M., Wirnsberger, P., Fortunato, M., Alet, F., Ravuri, S., Ewalds, T., Eaton-Rosen, Z., Hu, W., Merose, A., Hoyer, S., Holland, G., Vinyals, O., Stott, J., Pritzel, A., Mohamed, S., and Battaglia, P. (2023). Learning skillful medium-range global weather forecasting. *Science*, 382(6677):1416–1421.

Lauritzen, P. H., Conley, A. J., Lamarque, J. F., Vitt, F., and Taylor, M. A. (2015). The terminator "toy" chemistry test: A simple tool to assess errors in transport schemes. *Geoscientific Model Development*, 8:1299–1313.

Lefer, B. L., Shetter, R. E., Hall, S. R., Crawford, J. H., and Olson, J. R. (2003). Impact of clouds and aerosols on photolysis frequencies and photochemistry during trace-p: 1. analysis using radiative transfer and photochemical box models. *Journal of Geophysical Research: Atmospheres*, 108.

Li, J., Carlson, B. E., Yung, Y. L., Lv, D., Hansen, J., Penner, J. E., Liao, H., Ramaswamy, V., Kahn, R. A., Zhang, P., Dubovik, O., Ding, A., Lacis, A. A., Zhang, L., and Dong, Y. (2022). Scattering and absorbing aerosols in the climate system.

Liao, H., Yung, Y. L., and Seinfeld, J. H. (1999). Effects of aerosols on tropospheric photolysis rates in clear and cloudy atmospheres. *Journal of Geophysical Research: Atmospheres*, 104:23697–23707. For info on the effects aerosols have on rates.

Lin, H., Long, M. S., Sander, R., Sandu, A., Yantosca, R. M., Estrada, L. A., Shen, L., and Jacob, D. J. (2023). An adaptive auto-reduction solver for speeding up integration of chemical kinetics in atmospheric chemistry models: Implementation and evaluation in the kinetic pre-processor (kpp) version 3.0.0. *Journal of Advances in Modeling Earth Systems*, 15(2):e2022MS003293. e2022MS003293 2022MS003293.

Logan, J. A., Prather, M. J., Wofsy, S. C., and McElroy, M. B. (1981). Tropospheric chemistry: a global perspective. *Journal of Geophysical Research*, 86:7210–7254.

Lundberg, S. M., Erion, G., Chen, H., DeGrave, A., Prutkin, J. M., Nair, B., Katz, R., Himmelfarb, J., Bansal, N., and Lee, S.-I. (2020). From local explanations to global understanding with explainable ai for trees. *Nature Machine Intelligence*, 2(1):2522–5839.

Madronich, S. and Flocke, S. (1999). The Role of Solar Radiation in Atmospheric Chemistry. In Boule, P., editor, *Environmental Photochemistry*, The Handbook of Environmental Chemistry, pages 1–26. Springer, Berlin, Heidelberg.

McCarthy, J., Minsky, M. L., Rochester, N., and Shannon, C. E. (2006). A proposal for the dartmouth summer research project on artificial intelligence, august 31, 1955. *AI Magazine*, 27(4):12.

McCulloch, W. S. and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133.

Mitchell, R. and Frank, E. (2017). Accelerating the xgboost algorithm using gpu computing. *PeerJ Computer Science*, 3:e127.

Molina, M. J. and Rowland, F. S. (1974). Stratospheric sink for chlorofluoromethanes: chlorine atom-catalysed destruction of ozone. *Nature*, 249:810–812.

Monks, P. S., Archibald, A., Colette, A., Cooper, O., Coyle, M., Derwent, R., Fowler, D., Granier, C., Law, K. S., Mills, G., et al. (2015). Tropospheric ozone and its precursors from the urban to the global scale from air quality to short-lived climate forcer. *Atmospheric chemistry and physics*, 15(15):8889–8973.

Morgan, J. N. and Sonquist, J. A. (1963). Problems in the analysis of survey data, and a proposal. *Journal of the American Statistical Association*, 58:415–434.

Naser, M. Z. and Alavi, A. H. (2021). Error metrics and performance fitness indicators for artificial intelligence and machine learning in engineering and sciences. *Architecture, Structures and Construction*, 3(4):499–517.

Nievergelt, Y. (1994). Total least squares: State-of-the-art regression in numerical analysis. *SIAM Review*, 36(2):258–264.

Ott, J., Pritchard, M., Best, N., Linstead, E., Curcic, M., and Baldi, P. (2020). A fortran-keras deep learning bridge for scientific computing. *CoRR*, abs/2004.10652.

Pan, W., Gong, S., Ke, H., Li, X., Chen, D., Huang, C., and Song, D. (2025). Development of an automated photolysis rates prediction system based on machine learning. *Journal of Environmental Sciences*, 151:211–224.

Parrella, J. P., Jacob, D. J., Liang, Q., Zhang, Y., Mickley, L. J., Miller, B., Evans, M. J., Yang, X., Pyle, J. A., Theys, N., and Roozendael, M. V. (2012). Tropospheric bromine chemistry: Implications for present and pre-industrial ozone and mercury. *Atmospheric Chemistry and Physics*, 12:6723–6740.

Pathak, J., Subramanian, S., Harrington, P., Raja, S., Chattopadhyay, A., Mardani, M., Kurth, T., Hall, D., Li, Z., Azizzadenesheli, K., et al. (2022). Fourcastnet: A global data-driven high-resolution weather model using adaptive fourier neural operators. *arXiv preprint arXiv:2202.11214*.

Prather, M. (2012). Fast-jx v7.0a photolysis mechanism. `https://wiki.seas.harvard.edu/geos-chem/index.php/FAST-JX_v7.0_photolysis_mechanism`.

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, 1:81–106.

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

Reichstein, M., Camps-Valls, G., Stevens, B., Jung, M., Denzler, J., Carvalhais, N., and Prabhat (2019). Deep learning and process understanding for data-driven Earth system science. *Nature*, 566(7743):195–204.

Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65 6:386–408.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, 323:533–536.

Samuel, A. L. (1959). Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, 3(3):210–229.

Shen, J., Zhao, Q., Cheng, Z., Huo, J., Zhu, W., Zhang, Y., Duan, Y., Wang, X., Antony Chen, L.-W., and Fu, Q. (2020). Evolution of source contributions during heavy fine particulate matter (pm2.5) pollution episodes in eastern china through online measurements. *Atmospheric Environment*, 232:117569.

Shen, L., Jacob, D. J., Santillana, M., Bates, K., Zhuang, J., and Chen, W. (2022). A machine-learning-guided adaptive algorithm to reduce the computational cost of integrating kinetics in global atmospheric chemistry models: Application to geos-chem versions 12.0.0 and 12.9.1. *Geoscientific Model Development*, 15:1677–1687.

Shende, S. S. and Malony, A. D. (2006). The tau parallel performance system. *The International Journal of High Performance Computing Applications*, 20(2):287–311.

Slaper, H., Velders, G. J. M., Daniel, J. S., de Gruijl, F. R., and van der Leun, J. C. (1996). Estimates of ozone depletion and skin cancer incidence to examine the Vienna Convention achievements. *Nature*, 384(6606):256–258.

Stigler, S. M. (1981). Gauss and the Invention of Least Squares. *The Annals of Statistics*, 9(3):465 – 474.

Tie, X., Madronich, S., Walters, S., Zhang, R., Rasch, P., and Collins, W. (2003). Effect of clouds on photolysis and oxidants in the troposphere. *Journal of Geophysical Research: Atmospheres*, 108(D20).

UNEP (1987). Montreal protocol on substances that deplete the ozone layer.

Vohra, K., Vodonos, A., Schwartz, J., Marais, E. A., Sulprizio, M. P., and Mickley, L. J. (2021). Global mortality from outdoor fine particle pollution generated by fossil fuel combustion: Results from geos-chem. *Environmental Research*, 195:110754.

WHO (2021). Who global air quality guidelines: Particulate matter (pm2.5 and pm10), ozone, nitrogen dioxide, sulfur dioxide and carbon monoxide.

Wild, O., Zhu, X., and Prather, M. J. (2000). Fast-j: Accurate simulation of in-and below-cloud photolysis in tropospheric chemical models. *Journal of Atmospheric Chemistry*, 37:245–282.

Williams, J. E., Landgraf, J., Bregman, A., and Walter, H. H. (2006). Atmospheric chemistry and physics a modified band approach for the accurate calculation of online photolysis rates in stratospheric-tropospheric chemical transport models.

Willmott, C. J. and Matsuura, K. (2005). Advantages of the mean absolute error (mae) over the root mean square error (rmse) in assessing average model performance. *Climate research*, 30(1):79–82.

Wu, J., Chen, X. Y., Zhang, H., Xiong, L. D., Lei, H., and Deng, S. H. (2019). Hyperparameter optimization for machine learning models based on bayesian optimization. *Journal of Electronic Science and Technology*, 17:26–40. Bayesian Optimisation for XGB.

Xing, J., Zheng, S., Li, S., Huang, L., Wang, X., Kelly, J. T., Wang, S., Liu, C., Jang, C., Zhu, Y., Zhang, J., Bian, J., Liu, T.-Y., and Hao, J. (2022). Mimicking atmospheric photochemical modeling with a deep neural network. *Atmospheric Research*, 265:105919.

Zhang, X., Wang, Z., Cheng, M., Wu, X., Zhan, N., and Xu, J. (2021). Long-term ambient so2 concentration and its exposure risk across china inferred from omi observations from 2005 to 2018. *Atmospheric Research*, 247:105150.

Zheng, L., Lin, R., Wang, X., and Chen, W. (2021). The development and application of machine learning in atmospheric environment studies. Statistics about machine learning in atmospheric models.

Zuo, X., Yang, X., Dou, Z., and Wen, J. R. (2019). To tune or not to tune? an approach for recommending important hyperparameters. In *28th Text REtrieval Conference, TREC 2019 - Proceedings*. National Institute of Standards and Technology (NIST). States for gradient boosting, most important hyperparameter is the learning rate.