Geosci. Model Dev., 14, 3067–3077, 2021
https://doi.org/10.5194/gmd-14-3067-2021

Geoscientific
Model Development
Open Access

Physically regularized machine learning emulators of aerosol activation

Sam J. Silva[1], **Po-Lun Ma**[1], **Joseph C. Hardin**[1], **and Daniel Rothenberg**[2]

[1]Pacific Northwest National Laboratory, Richland, WA, USA
[2]ClimaCell, Boston, MA, USA

Correspondence: Sam J. Silva (sam.silva@pnnl.gov)

Received: 24 November 2020 – Discussion started: 21 December 2020
Revised: 11 April 2021 – Accepted: 14 April 2021 – Published: 28 May 2021

Abstract. The activation of aerosol into cloud droplets is an important step in the formation of clouds and strongly influences the radiative budget of the Earth. Explicitly simulating aerosol activation in Earth system models is challenging due to the computational complexity required to resolve the necessary chemical and physical processes and their interactions. As such, various parameterizations have been developed to approximate these details at reduced computational cost and accuracy. Here, we explore how machine learning emulators can be used to bridge this gap in computational cost and parameterization accuracy. We evaluate a set of emulators of a detailed cloud parcel model using physically regularized machine learning regression techniques. We find that the emulators can reproduce the parcel model at higher accuracy than many existing parameterizations. Furthermore, physical regularization tends to improve emulator accuracy, most significantly when emulating very low activation fractions. This work demonstrates the value of physical constraints in machine learning model development and enables the implementation of improved hybrid physical and machine learning models of aerosol activation into next-generation Earth system models.

1 Introduction

Aerosols are important components of the Earth system, where they play a critical role in cloud processes and strongly modulate the radiative budget. Aerosols impact radiation through directly absorbing and scattering light (Wallace and Hobbs, 2006), and by changing the radiative characteristics, lifetime, and abundance of clouds through a wide array of aerosol–cloud interactions (e.g., Albrecht, 1989; Twomey, 1977). This large influence is in part because cloud formation through the direct condensation of atmospheric water vapor into cloud droplets is thermodynamically unfavorable in the atmosphere. Instead, cloud formation is largely initiated by the nucleation of cloud droplets through heterogenous interactions between water vapor and aerosol (Seinfeld and Pandis, 2016). These aerosol–cloud interactions are the dominant radiative impact of aerosol in the Anthropocene, attributable to a large portion of the anthropogenic radiative forcing of aerosol (Bellouin et al., 2020). This influence on the global radiative budget is relatively large and potentially offsets much of the warming associated with anthropogenic greenhouse gas emissions. Despite the importance of aerosol–cloud interactions, modern Earth system models struggle to accurately represent these processes (Seinfeld et al., 2016). In total, the accurate simulation of aerosol–cloud interactions is one of the largest uncertainties in modern Earth system models and is also a limiting factor in developing a predictive capability for the Earth system (Bellouin et al., 2020; Boucher et al., 2013; Committee on the Future of Atmospheric Chemistry Research et al., 2016).

Aerosol activation, also known as droplet nucleation, is a necessary step in the processes driving aerosol–cloud interactions. Once activated, aerosol can directly influence cloud properties (Wallace and Hobbs, 2006). For example, the addition of activated aerosol to existing clouds can change the number concentration of cloud droplets, which impacts cloud brightness and lifetime, and the resulting net radiative impact of clouds (e.g., Christensen et al., 2020; Twomey, 1974, 1977). The aerosol activation process occurs at a scale much smaller than Earth system model grid spacing and interacts with a variety of other sub-grid-scale processes relating to clouds (e.g., turbulent mixing and convection).

Current methods for simulating aerosol activation require trade-offs between model fidelity and computational efficiency. The most accurate models of aerosol activation simulate the thermodynamic and chemical conditions within a zero-dimensional parcel of air to analytically predict the fraction of aerosol activated into cloud droplets (e.g., Ghan et al., 2011; Rothenberg and Wang, 2015). These so-called "parcel models" explicitly resolve the condensational processes across a size-resolved distribution of an aerosol population for a specified amount of time to calculate both the maximum supersaturation of the local atmosphere and the total number of aerosols activated into cloud droplets. These parcel models are too computationally expensive to be used in global Earth system models; thus, various parameterizations have been developed with reduced computational cost.

Early parameterizations of aerosol activation were based on a few observations (e.g., Twomey, 1959) and were generally simple functions of a limited number of parameters. As computational and observational capabilities increased, these parameterizations increased in complexity (e.g., Abdul-Razzak and Ghan, 2000; Fountoukis and Nenes, 2005; Ming et al., 2006). Although these parameterizations all generally aim to calculate similar quantities, there are key differences in their implementation. These differences are largely based around the level of explicit versus approximated process-level details and the degree of tuning within the parameterizations, described further for a variety of popularly used parameterizations in Ghan et al. (2011). The majority of these modern parameterizations perform similarly well for common atmospheric conditions, although relatively large differences ($\sim 30\,\%$) can be found in certain scenarios (Ghan et al., 2011). Despite their increased computational complexity, these parameterizations are still unable to fully reproduce the results of detailed parcel models, with errors on the order of $\sim 10\,\%$ (Rothenberg and Wang, 2015). While small, these errors can potentially compound in models with longer run times, further motivating the development of emulators with improved predictive skill.

Recent work applying machine learning techniques to the emulation of computationally expensive systems has shown promise toward developing emulators that are both fast and accurate (e.g., Brenowitz and Bretherton, 2018; Gentine et al., 2018; Rasp et al., 2018; Silva et al., 2020a). This is particularly the case for the class of so-called "physically informed" machine learning emulators (e.g., Raissi et al., 2019; Reichstein et al., 2019). Physically informed machine learning algorithms directly incorporate physical information into their construction and/or training with the aim of creating emulators that are more performant in terms of ease of training or resulting accuracy. For example, physical information can be encoded through penalizing emulators that violate known constraints (e.g., conservation of energy) in the cost function that is optimized during model parameter optimization (e.g., Beucler et al., 2019). More complex methods of including physical information can be achieved through directly altering the architecture of a machine learning model to analytically enforce various constraints or follow a physically based model system (e.g., Zhao et al., 2019). These approaches of including physical information in machine learning model development ultimately have what is known as a "regularizing effect" on the model, wherein they help reduce model overfitting.

In this study, we explore how a hybrid physical and machine learning modeling approach can be used to develop improved parameterizations of aerosol activation. We demonstrate that applying a very simple constraint to commonly used machine learning techniques improves their predictive skill and can lead to more accurate and trustworthy emulators of computationally expensive models. We build on previous work (e.g., Rothenberg and Wang, 2015; Lipponen et al., 2013) by considering a wider array of emulator design methods and physical constraints.

2 Modeling approach

We develop emulators of a detailed parcel model for aerosol activation. Specifically, we train several classes of machine learning models to emulate the "Pyrcel" parcel model, reproducing the fraction of aerosol particles activated (Rothenberg and Wang, 2015). Pyrcel simulates the activated fraction of an initial population of aerosol in a zero-dimensional parcel of air as it adiabatically rises in the atmosphere. Here, we use a single aerosol mode with 250 bins and an initial supersaturation of zero. This broadly assumes that we are making these calculations directly at the base or edge of a cloud. Other atmospheric conditions used as inputs to the Pyrcel model are varied to generate the emulator development datasets and are summarized in Table 1. For more details on the Pyrcel model, see Rothenberg and Wang (2015).

2.1 Machine learning techniques

We assess three commonly used machine learning regression models: ridge regression, gradient boosted regression trees, and deep neural networks. All models take the quantities in Table 1 as inputs and predict the activated fraction of aerosol, which ranges from 0 to 1.

Ridge regression is a linear prediction technique that optimizes coefficients using a penalized cost function that aims to account for and reduce the impact of collinearity in the training dataset. This is done through the addition of an L2 penalty to the commonly used sum of square residual minimization from ordinary least squares fitting. Ridge regression has been used in a variety of prediction tasks in the Earth system sciences, including ozone chemistry (Nowack et al., 2018) and estimating the climate sensitivity (Bretherton and Caldwell, 2020). In this study, we use the implementation of

Table 1. Pyrcel parcel model input parameters and sampling range used for emulator training.

Quantity	Units	Range	Name
$\mathrm{Log}_{10}N$	$\mathrm{Log}_{10}\mathrm{cm}^{-3}$	[1, 4]	Mode number concentration
$\mathrm{Log}_{10}\mu_\mathrm{g}$	$\mathrm{Log}_{10}\mu\mathrm{m}$	[−3, 1]	Mode geometric mean radius
Sigma_g	–	1.6 or 1.8	Mode standard deviation
Kappa	–	[0, 1.2]	Mode hygroscopicity
$\mathrm{Log}_{10}\mathrm{V}$	$\mathrm{Log}_{10}\mathrm{ms}^{-1}$	[−2, 1]	Updraft velocity
T	K	[248, 310]	Air temperature
P	Pa	[50 000, 105 000]	Air pressure
a_c	–	[0.1, 1.0]	Accommodation coefficient

ridge regression in the "glmnet" package in the R language (Friedman et al., 2010). For more information on ridge regression, see Hastie et al. (2001).

Gradient boosted regression trees are a class of machine learning algorithm that trains an ensemble of small tree-based regression models. After the first ensemble member is trained, each following member is fit to the residuals of the previous model, and this process is iteratively completed until satisfactory model performance is achieved (e.g., no additional improvement in prediction skill on the validation dataset is gained by adding ensemble members). We specifically use the XGBoost library, as implemented in the "XGBoost" package in the R language (Chen and Guestrin, 2016). XGBoost has been shown to effectively train useful models in the Earth sciences, including applications to atmospheric composition (Ivatt and Evans, 2020; Silva et al., 2020b) and evapotranspiration (Fan et al., 2018). For more information on boosted trees and XGBoost, see Chen and Guestrin (2016).

Deep neural networks (DNNs) are the third class of machine learning algorithm that we explore in this work. DNNs are a type of artificial neural network, with multiple layers between the input and output nodes. In recent years, DNNs have seen widespread use in the Earth system sciences as they perform quite well in estimation tasks and scale well on large supercomputing systems, making them ideal candidates for process emulation in models of the Earth system (e.g., Rasp et al., 2018; Silva et al., 2019). We use the Keras library and the TensorFlow implementation in the Python programming language to design and train the DNNs used in this work (Chollet et al., 2015; Martín Abadi et al., 2015). All DNNs here are feed-forward neural networks, with each densely connected layer followed by a dropout layer. For more information on DNNs, see Goodfellow et al. (2016).

2.2 Physical regularization

We investigate the improvement to emulator performance achieved by the application of physical regularization. In the context of this work, physical regularization is the process of adding physical information into an otherwise physically naïve machine learning model to help reduce overfitting. The governing hypothesis here is that by including additional physical information, the model should perform better on an unknown test dataset. To that end, we use the maximum supersaturation and activation fraction parameterizations described in Twomey (1959) (hereafter, Twomey) and Abdul-Razzak and Ghan (2000) (hereafter, ARG) as regularizing terms for all three machine learning methods described here.

The Twomey scheme was developed as a simple expression of only updraft velocity, where the maximum supersaturation in an air parcel is defined as

$$S_\mathrm{max} = \left(\frac{1.63 \times 10^{-3} V^{\frac{3}{2}}}{ck\beta\left(\frac{3}{2},\ \frac{k}{2}\right)} \right)^{\frac{1}{k+2}}, \tag{1}$$

and the activated fraction is

$$\mathrm{ActFrac}_\mathrm{Twomey} = \frac{cS_\mathrm{max}^k}{N}. \tag{2}$$

Here, V is the vertical velocity (cm/s), c and k are fitted parameters (here $c = 2000$ and $k = 0.4$), β is the beta function (evaluated here as 4.48), and N is the aerosol number concentration in the air parcel (see Twomey, 1959, for more details). We bound Eq. (2) within the range from 0 to 1 in order to account for known limitations in the scheme (e.g., Ghan et al., 2011). We use Eqs. (1) and (2) as regularizing terms through a simple hybrid modeling approach where the machine learning emulator is optimized to calculate the residual of the parcel model from the original Twomey (1959) estimates. This is visualized in the flowchart in Fig. 1. Stated mathematically, we calculate

$$\mathrm{ActFrac} = \mathrm{ActFrac}_\mathrm{Twomey} + f_\mathrm{ActFrac}(x), \tag{3}$$

where ActFrac is the target parcel model activation fraction to emulate, $\mathrm{ActFrac}_\mathrm{Twomey}$ is the estimate from the Twomey scheme, $f_\mathrm{ActFrac}(x)$ is the function that the machine learning emulators will be trained to learn, and x is the set of input parameters. This method allows for some of the nonlinear behavior of ActFrac to be encoded into the estimation prior to any machine learning optimization and, thus, should potentially allow for a better solution to this ill-posed estimation task. We additionally feed the emulators with the Twomey-predicted $S_{\max}$ and $\text{ActFrac}_{\text{Twomey}}$ as additional input variables for the prediction tasks.

The calculation of the maximum supersaturation and activated fraction in the ARG scheme is more involved than the Twomey scheme and is described in detail in Abdul-Razzak and Ghan (2000). We incorporate the ARG-estimated activation fraction identically to the Twomey regularization, through learning the residual of the ARG scheme and the parcel model. As with the Twomey regularization, we feed the ARG-regularized emulators the ARG-predicted $S_{\max}$ and activated fraction as additional input variables for the prediction task. The impact of including the ARG or Twomey parameterization predicted $S_{\max}$ and activated fraction is marginal in terms of net performance of the emulator, although as the information is already calculated in the regularization step, including it in the model input space adds extra information for little computational cost.

3 Emulator training

We generated a training and evaluation dataset of 20 000 realizations of the detailed Pyrcel parcel model using the range of environmental conditions summarized in Table 1, sampled using a Latin hypercube sampling (LHS) technique as implemented in the "SMT" Python package (Bouhlel et al., 2019). A total of 10 000 simulations were completed with LHS given the limits shown in Table 1. Another 10 000 samples were completed using the same LHS limits shown in Table 1 without the logarithmic transformation applied to the vertical velocity, aerosol number density, or aerosol mean diameter. This sampling method and input parameter space is similar to previous aerosol activation parameterization development datasets using the Pyrcel parcel model (Rothenberg and Wang, 2015). The Pyrcel model fails to converge in the numerical solver for one case out of the 20 000 total simulations during certain conditions unlikely to occur in the real atmosphere (very low pressure with high temperatures and updraft velocities); this case was removed from the training dataset. The full dataset was randomly split into training (70 %), validation (10 %), and testing (20 %) datasets. The training dataset was used to optimize the machine learning model parameters and hyperparameters as assessed by evaluation against the validation datasets. Final model performance was assessed based on performance on the test dataset. We completed an additional set of 1000 simulations using the same input space as shown in Table 1 and Latin hypercube sampling, but ranging from 310 to 314 K, which is 4 K warmer than the training dataset. This is intended to assess model generalizability, or performance on out-of-sample training data. All features were standardized through a Z-score normalization where the mean was subtracted from each feature, followed by dividing each feature by its standard deviation.

3.1 Hyperparameter selection

All three of the emulator methods used in this work require some degree of hyperparameter selection within the model architecture. Unless otherwise stated, we used package default values for all hyperparameters. Hyperparameters were selected separately for each application of the emulators in this work based on validation dataset performance and are summarized in Table 2. In general, we found that the emulator performance was not particularly sensitive to the hyperparameter tuning; the performance metrics only improved marginally after more optimal parameters were selected (not shown).

In ridge regression, the strength of the L2 norm penalty is controlled by a hyperparameter, commonly written as "lambda". We selected this lambda exponential value by directly searching across a range of 101 values from -2 to 3 in increments of 0.05. For the cases investigated here, the validation error tended to asymptotically decrease with smaller lambda values below approximately -1.1.

The XGBoost hyperparameters chosen here were the learning rate, the maximum depth of each tree, and the total number of trees included in the emulator. We searched across these parameters using a grid search of the learning rate and the maximum depth, spanning from 0.1 to 0.9 in steps of 0.1 for the learning rate and from 2 to 24 in steps of 2 for the maximum tree depth. For all trees, we allowed the trees to continuously grow with 25 early stopping rounds determining the final depth. Once adding trees did not improve the performance of the emulator on the validation dataset for 25 tree additions, the model training was stopped.

The DNN hyperparameter tuning requires a more careful approach than direct grid search due to the very large hyperparameter optimization space. Our approach involves using automated hyperparameter tuning software to suggest candidate model hyperparameters and then fully evaluating the top candidates individually. For automated hyperparameter tuning, we used the Keras Tuner software package, which searches over a wide range of possible hyperparameters more efficiently than random or grid search techniques (O'Malley et al., 2019). Keras Tuner parameters included the hyperband search algorithm, a validation loss objective, with maximum tuner and training epochs of 100, and a factor value of 3. We chose to search across the number of layers, the number of nodes per layer, the dropout rate following each dense layer, and the DNN learning rate. We allowed up to five network hidden layers, each with between 10 and 100 nodes, a dropout rate of 0.1 to 0.9, and a learning rate of 10^{-2}, 10^{-3}, or 10^{-4}. We used the Adam optimizer and the rectified linear unit (ReLu) activation function for all but the output layer, which was set to linear function for all prediction tasks. We then took the top 15 suggested model configurations from the Keras Tuner search, fully trained them, and selected the best performer from that subset. For these fully trained models, we use a batch size of 64 and optimize for the number of

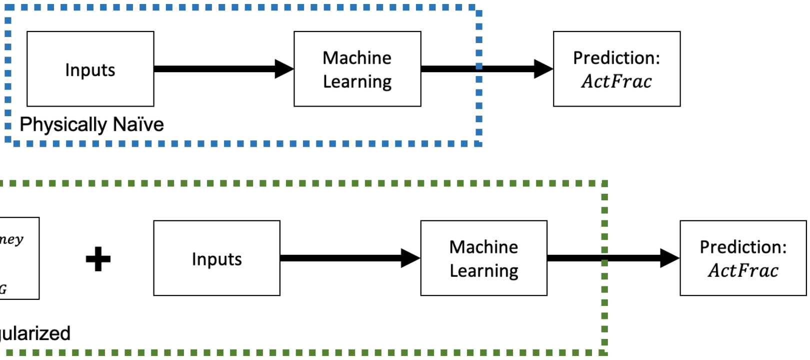

Figure 1. Schematic diagram of the emulator construction for the physically naïve and physically regularized emulators.

training epochs using early stopping, with 25 early stopping epochs.

4 Emulator evaluation

We evaluate the skill of these emulators in reproducing the activation fraction prediction within the test set, as described in Sect. 3. As machine learning predictive skill on the training set is not always an indicator of predictive skill on the test set, we discuss only test set performance here as a more strict evaluation criteria.

4.1 Physically naïve machine learning emulators

The test set performance of the activation fraction machine learning emulators without any physical regularization, here referred to as "physically naïve", is summarized in Fig. 2 along with the ARG parameterization predictions. In general, the emulators perform well, with the majority of points clustered around the 1 : 1 line, mean squared errors (MSE) below 0.05, and high R^2 values (above ~ 0.7). The best performance comes from the DNN and the XGBoost regressions, followed by the ridge regression. We additionally include the ARG activation parameterization in Fig. 2 as a baseline performance comparison with a commonly used existing activation parameterization. Both the DNN and the XGBoost regression outperform the ARG benchmark parameterization, with lower mean squared errors and higher R^2 values. The Twomey scheme is not shown in Fig. 2, as it performs relatively poorly (MSE $=0.29$, $R^2=0.03$) and is ,thus, not a particularly useful benchmark compared with the relatively skillful ARG parameterization.

For certain cases very near the mass-conserving bounds from 0 to 1 ($\sim 10\,\%$ of the test data), the emulators predict activation fraction values that extend beyond those bounds. Other than for the linear ridge regression, these deviations outside of the mass-conserving bounds are all very small (less than 0.01). Although imposing that range as an upper and lower bound on those regressions would be a sensible choice if the emulators were implemented into an Earth system model, the imposition does not substantially impact performance metrics (MSE and R^2).

For the DNN emulator, we chose a linear activation function for the final model layer. As the activation fraction varies from 0 to 1, a sigmoid activation function would also be a logical choice and would encode a small amount of prior information into the system. However, in this case, the linear activation function has slightly better predictive skill. Using a sigmoid activation function does not appreciably change the results shown here and actually leads to slightly worse emulator performance.

Machine learning emulators tend to improve performance with larger training datasets. In this case, training using only half of the available data still leads to relatively skillful emulators. For example, the same DNN trained on 50 % of the training samples has an MSE of 0.0017 and an R^2 of 0.99. This is worse than the DNN in Fig. 2, which is fully trained, but does still outperform the commonly used and physically based ARG parameterization.

4.2 Physically regularized emulators

Including physical regularization generally improves model performance on the test set. Performance for the Twomey- and ARG-regularized models is shown in the scatterplots in Fig. 3. Ultimately, the poor performance of the Twomey scheme prior to implementation as a regularizing term limits the added value it provides to the emulators. The performance gains by Twomey regularization compared with the physically naïve emulators are generally $\sim 10\,\%$ or less in terms of mean squared error for the emulators, with differences in R^2 values of generally less than a few percent. While this specific performance gain is not large, the Twomey scheme can be calculated as a simple power of vertical velocity and is thus a computationally simple technique for improving emulator accuracy.

The benefits of the ARG regularization are larger and more consistent across emulators, as can be seen in Fig. 3. For all three machine learning model types, the ARG regularization performs the best, with the lowest mean squared error

Table 2. Emulator hyperparameters chosen in this study.

	Naïve	Twomey regularized	ARG regularized
Ridge			
Lambda	−1.9	−2.0	−2.0
XGBoost			
Max depth	8	8	6
Eta	0.1	0.1	0.1
DNN			
Learning rate	1.00×10^{-3}	1.00×10^{-4}	1.00×10^{-3}
Training epochs	40	147	125
No. of layers	3	3	3
No. of nodes	[100, 80, 40]	[100, 40, 70]	[50, 100, 30]
Dropout fraction	[0.3, 0.1, 0.5]	[0.1, 0.1, 0.2]	[0.1, 0.3, 0.2]

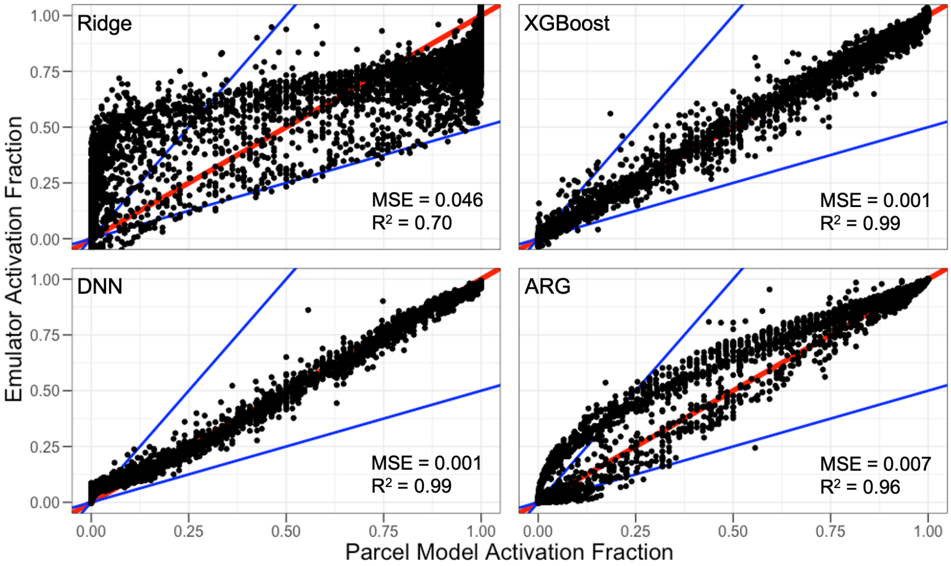

Figure 2. Scatterplot comparisons of the three physically naïve machine learning emulators and the ARG scheme predicted activation fraction with the detailed parcel model. The 1 : 1 line is in red, and the blue lines represent a factor of 2 difference. Performance statistics are given in each panel.

and highest R^2 values within each emulator category. While all model types improve with the additional information provided by the ARG scheme, the smallest relative improvement is that of the DNN and XGBoost emulators, and the largest improvement is for the ridge regression. The linear ridge regression, when regularized by the ARG scheme, outperforms the standard ARG parameterization with a 40 % relative reduction in the mean squared error. Framing this finding from the perspective of the ARG scheme, a linear correction term with ridge-calculated coefficients could reduce the parameterization error by 40 %, and nonlinear corrections (i.e., XGBoost or DNNs) could further reduce that error by an order of magnitude. As with the naïve emulators, for predictions very near the bounds of 0 to 1, the physically regularized emulators do tend to predict variables outside of that range. The linear ridge regression predicts the largest deviations, where the physically regularized XGBoost and DNN models typically predict deviations within 0.01 of the bounds.

Physical regularization particularly improves the emulator behavior for very low activation fractions. As an example, Fig. 4 shows all three versions of the DNN emulator performance for cases with activation fractions below 0.1. The physically naïve emulator substantially overestimates most very low parcel model simulated activation fractions. The regularization from the Twomey scheme improves upon this issue but increases the emulator scatter in this range. The more detailed and general ARG regularization reduces the overprediction issue from the naïve scheme even further,

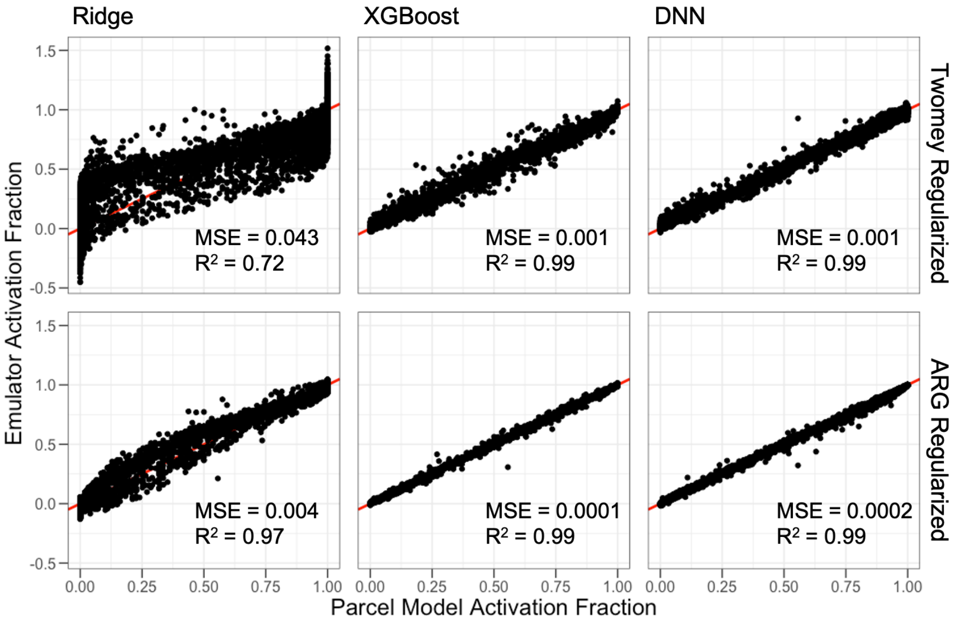

Figure 3. Scatterplots for the various emulator types against the parcel model activation fraction. The 1 : 1 line is shown in red, and emulator specific performance statistics are shown in each panel.

with the best overall performance. This potentially has implications for the impact of these emulators when implemented into an Earth system model, where low activation fractions can be common.

The capacity of these emulators (their ability to emulate arbitrarily complex functions) increases with the emulator complexity and number of parameters. As the capacity increases, the benefit of the Twomey and ARG physical regularization decreases. This is evident by the large gains in accuracy when the ridge regression, which is fundamentally a linear model, is physically regularized as compared with the very modest absolute accuracy gains in the largely nonlinear DNNs. To further illustrate this point, we ran sensitivity experiments with the XGBoost emulators, evaluating the prediction error on the validation dataset as a function of the number of boosting iterations for a naïve and ARG physically regularized emulators, both with the same hyperparameters. Each boosting iteration has the potential to add trees to the model and, thus, increases the emulator capacity. Results are shown in Fig. 5. As expected, additional trees (boosting iterations) reduce the mean squared prediction error of both the naïve and ARG-regularized emulators. When the emulator capacity is relatively low (the number of trees is low), the physically regularized emulator is much more skillful in terms of MSE. As the capacity increases, this regularization accuracy benefit is reduced substantially, although it is always present to some extent. For a given machine learning technique, increased capacity typically comes with increased computational cost. Thus, including physical information through physical regularization can be a computationally efficient strategy for achieving a given model accuracy with lower capacity.

4.3 Emulator generalizability

We further evaluate the emulators using a dataset with up to 4 K warmer temperatures than used in the training data. This evaluation on input data outside of the training parameter space can provide useful information on the generalizability of the emulators as well as their performance when used in scenarios that may be not well characterized by the training dataset (potentially likely in a climate model simulation). The generalizability test dataset was generated using the same parameter space as the training dataset described in Table 1, except the temperature range was from 310 to 314 K.

The emulators tend to perform fairly well in this generalizability test, with MSE and R^2 values similar to the performance shown on the test dataset. Summary results are shown for the DNN emulator for all three emulator designs in Fig. 6. The results in Fig. 6 are qualitatively consistent for the ridge and XGBoost emulators. The best emulator performance is from the ARG-regularized emulator, followed by the Twomey regularization, and then the physically naïve DNN. Ultimately, the limited conditions under which the original Twomey (1959) formulation is derived (e.g., Ghan et al., 2011) limit the generalizability performance of the scheme as a regularizing term for these emulators. It is important to note that although the emulators perform well in this generalizability test, there is no guarantee that they will perform well for all extrapolation cases, particularly those that deviate very far from the training data parameter space.

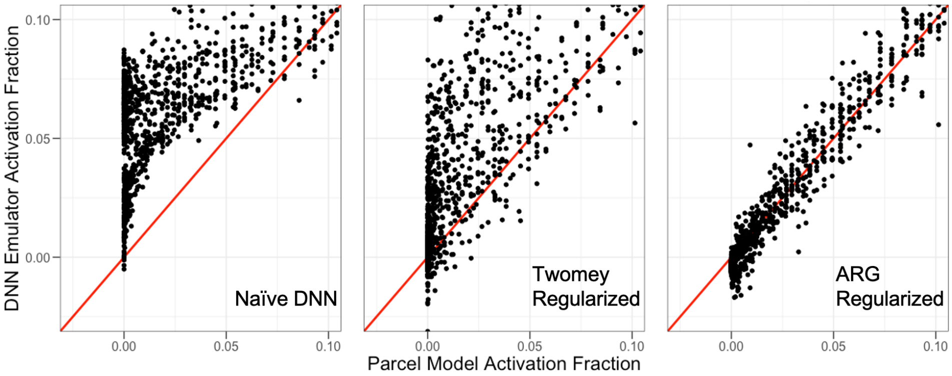

Figure 4. Scatterplots for the various DNN emulator types against the parcel model activation fraction for cases with activation fractions below 0.1. The 1 : 1 line is shown in red.

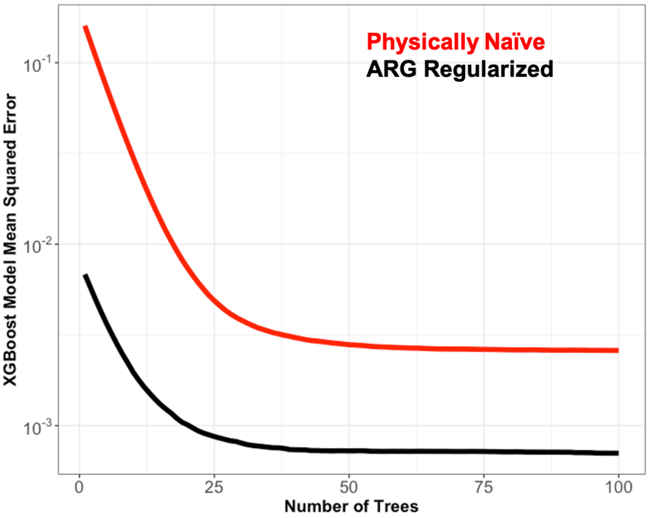

Figure 5. Mean squared error as a function of the XGBoost number of trees for both the naïve non-regularized emulator (red) and the ARG-regularized emulator (black).

4.4 Emulator sensitivity

We additionally evaluate the emulators as a function of variability in their input parameter space. Analogous to Rothenberg and Wang (2015), we fix all but one input parameter and explore the variability in the emulator as a function of one single input parameter. Results for the DNN emulators as a function of number concentration, vertical velocity, mean radius, and hygroscopicity are shown in Fig. 7. Other emulators are generally consistent, with worse overall skill for the ridge regression emulators. Generally, the emulators all perform well. The best performance is associated with the ARG-regularized scheme, and the most aberrant performance is associated with the Twomey regularization.

The emulators are all within ∼ 10 % for all predictions as a function of number concentration, mean radius, and vertical velocity. Much larger errors are apparent for emulator performance in cases with very low aerosol hygroscopicity, where the only skillful emulator is the ARG-regularized model. In the real atmosphere, very low hygroscopicity values are reasonably common, and activation overestimates by nearly a factor of 4 for the naïve and Twomey-regularized schemes would likely have a substantial impact on climate, producing too many cloud droplets by activating hydrophobic aerosols. These issues are consistent with the large overestimates seen at low activations in Fig. 4. Although the specific issue of the poor performance of the Twomey-regularized and naïve emulators in this low hygroscopicity range could potentially be somewhat resolved with additional model training data and other training optimization techniques (e.g., transfer learning on a subsample of the data and optimizing in log space), initial tests suggest that none of these issues completely solve the performance issues. This strongly motivates the use of sufficient physical regularization to address other potentially unknown biases in emulator performance.

5 Summary and future directions

Although aerosol activation can be challenging to simulate with high accuracy and computational speed, machine learning techniques provide a potential path forward. We demonstrate that several classes of machine learning models can produce accurate emulations of a detailed parcel model, competitive with existing model parameterizations. We evaluate the performance of three machine learning regression models: ridge, XGBoost, and DNNs. Both the XGBoost and the DNN regression outperform the commonly used ARG parameterization, with the best overall performance observed from the DNN.

We show that including physical information in the construction and training of these machine learning models can yield improved emulators through physical regularization with the Twomey (1959) and Abdul-Razzak and Ghan (2000) aerosol activation parameterizations. In particular, improved performance through physical regularization is apparent in

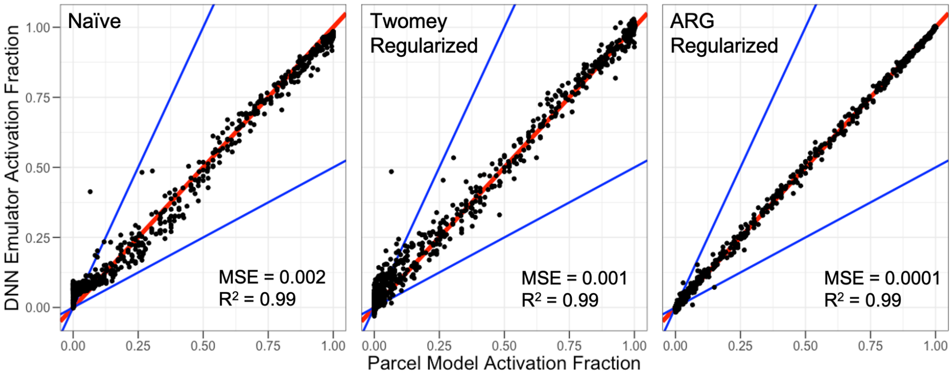

Figure 6. Activation fraction performance of the DNN emulators used here on the +4 K generalizability test dataset. The 1 : 1 line is in red, and the blue lines represent a factor of 2 difference. Performance statistics are given in each panel.

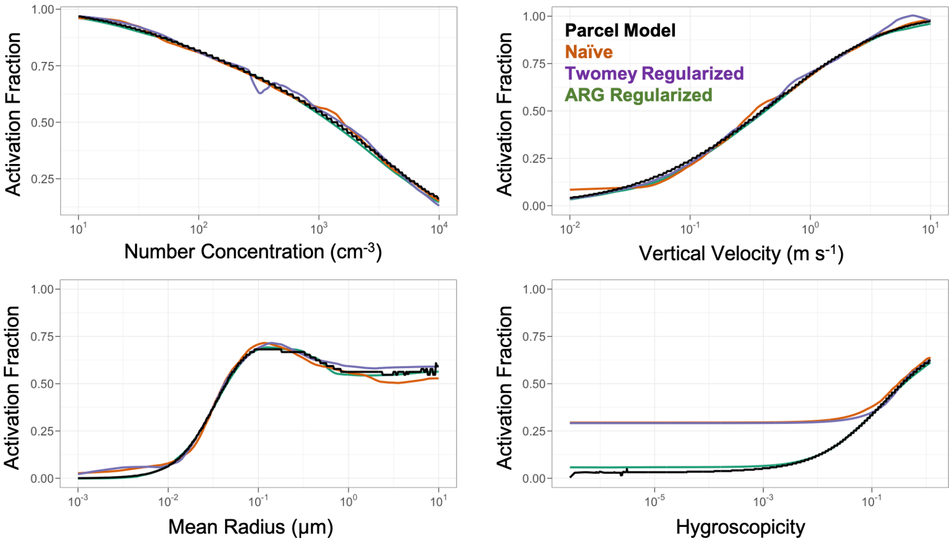

Figure 7. Variability in the predicted activation fraction from the DNN emulators and the parcel model as a function of input parameters. Number concentration, vertical velocity, mean radius, and aerosol hygroscopicity are shown here. The parcel model is shown in black, the naïve DNN emulator is shown in orange, the Twomey-regularized emulator is shown in purple, and the ARG-regularized emulator is shown in green. For each panel, all other parameters are fixed at the following settings: number concentration of 1000 cm^{-3}, mean radius of 0.05, aerosol mode standard deviation of 1.8, hygroscopicity of 0.54, vertical velocity of 0.5 $\mathrm{m\,s}^{-1}$, temperature of 283 K, pressure of 85 000 Pa, and an accommodation coefficient of 0.95.

emulator edge cases and cases that are poorly represented in the emulator training data. These accuracy gains are dependent on the quality of the physical information provided in the regularization step as well as the capacity of the machine learning model. The original Twomey (1959) activation scheme is limited in scope and is only applicable under certain atmospheric conditions. This leads to reduced performance of the Twomey-regularized emulators over those regularized by the globally applicable ARG parameterization. The improved performance from physical regularization is somewhat dependent on emulator capacity: once sufficient emulator capacity is available, the accuracy differences between physically informed and physically naïve models become small.

Machine learning techniques have been shown to scale quite well on large-scale supercomputing systems, particularly for feed-forward deep neural networks like those applied here. This good computational speed scaling lends support to the applicability of machine learning emulators in computationally expensive Earth system models, like the Energy Exascale Earth System Model (E3SM) from the United States Department of Energy. Additionally, the algorithms investigated here (XGBoost and DNNs) have efficient graphics processing unit (GPU) implementations and are, thus, directly applicable to next-generation high-performance computing architectures that may rely more on GPUs. As the representation of processes in Earth system models grows more complex and computationally expensive, the development and application of novel emulation techniques becomes continually more useful as an important step in model development.

Code and data availability. The current version of the Pyrcel model is available from the project website: http://pyrcel.readthedocs.io (last access: 1 December 2020) (under the New BSD (3-clause) license). The exact version of the Pyrcel model used to produce the results used in this paper, along with the analysis code and model output, is archived on Zenodo: https://doi.org/10.5281/zenodo.4319145 (Silva, 2020).

Author contributions. SJS, PM, and JCH designed the study. SJS developed and implemented the emulator techniques. DR developed the Pyrcel code. All authors contributed to the paper preparation.

Competing interests. The authors declare that they have no conflict of interest.

Acknowledgements. A portion of the research described in this paper was conducted in the framework of the "Laboratory Directed Research and Development" program at Pacific Northwest National Laboratory (PNNL), a multi-program national laboratory operated by Battelle (under contract no. DE-AC05-76RL01830) for the U.S. Department of Energy. Sam J. Silva is grateful for the support of the Linus Pauling Distinguished Postdoctoral Fellowship program. He would also like to acknowledge his parents for providing childcare during the COVID-19 pandemic, which enabled the writing of this paper. This study was partly supported by the "Enabling Aerosol–cloud interactions at GLobal convection-permitting scalES (EAGLES)" project (project no. 74358), funded by the U.S. Department of Energy, Office of Science, Office of Biological and Environmental Research, Earth System Model Development (ESMD) program.

Financial support. This research has been supported by the Pacific Northwest National Laboratory (Laboratory Directed Research and Development program) and the U.S. Department of Energy (Enabling Aerosol–cloud interactions at GLobal convection-permitting scalES (EAGLES)" project (project no. 74358).

Review statement. This paper was edited by David Topping and reviewed by two anonymous referees.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M. Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X.: TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems, available at: https://www.tensorflow.org/ (last access: 8 October 2020), 2015.

Abdul-Razzak, H. and Ghan, S. J.: A parameterization of aerosol activation: 2. Multiple aerosol types, J. Geophys. Res.-Atmos., 105, 6837–6844, https://doi.org/10.1029/1999JD901161, 2000.

Albrecht, B. A.: Aerosols, Cloud Microphysics, and Fractional Cloudiness, Science, 245, 1227–1230, https://doi.org/10.1126/science.245.4923.1227, 1989.

Bellouin, N., Quaas, J., Gryspeerdt, E., Kinne, S., Stier, P., Watson-Parris, D., Boucher, O., Carslaw, K. S., Christensen, M., Daniau, A.-L., Dufresne, J.-L., Feingold, G., Fiedler, S., Forster, P., Gettelman, A., Haywood, J. M., Lohmann, U., Malavelle, F., Mauritsen, T., McCoy, D. T., Myhre, G., Mülmenstädt, J., Neubauer, D., Possner, A., Rugenstein, M., Sato, Y., Schulz, M., Schwartz, S. E., Sourdeval, O., Storelvmo, T., Toll, V., Winker, D., and Stevens, B.: Bounding Global Aerosol Radiative Forcing of Climate Change, Rev. Geophys., 58, e2019RG000660, https://doi.org/10.1029/2019RG000660, 2020.

Beucler, T., Pritchard, M., Rasp, S., Gentine, P., Ott, J., Baldi, P., and Gentine, P.: Enforcing Analytic Constraints in Neural Networks Emulating Physical Systems, Phys. Rev. Lett., 126, 098302, https://doi.org/10.1103/PhysRevLett.126.098302, 2021.

Boucher, O., Randall, D., Artaxo, P., Bretherton, C., Feingold, G., Forster, P., Kerminen, V.-M., Kondo, Y., Liao, H., Lohmann, U., Rasch, P., Satheesh, S. K., Sherwood, S., Stevens, B., and Zhang, X. Y.: Clouds and Aerosols, in: Climate Change 2013: The Physical Science Basis. Contribution of Working Group I to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change, edited by: Stocker, T. F., Qin, D., Plattner, G.-K., Tignor, M., Allen, S. K., Boschung, J., Nauels, A., Xia, Y., Bex, V., and Midgley, P. M., Cambridge University Press, Cambridge, United Kingdom and New York, NY, USA, 571–658, https://doi.org/10.1017/CBO9781107415324.016, 2013.

Bouhlel, M. A., Hwang, J. T., Bartoli, N., Lafage, R., Morlier, J., and Martins, J. R. R. A.: A Python surrogate modeling framework with derivatives, Adv. Eng. Softw., 102662, https://doi.org/10.1016/j.advengsoft.2019.03.005, 2019.

Brenowitz, N. D. and Bretherton, C. S.: Prognostic Validation of a Neural Network Unified Physics Parameterization, Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 45, 6289–6298, https://doi.org/10.1029/2018GL078510, 2018.

Bretherton, C. S. and Caldwell, P. M.: Combining Emergent Constraints for Climate Sensitivity, J. Climate, 33, 7413–7430, https://doi.org/10.1175/JCLI-D-19-0911.1, 2020.

Chen, T. and Guestrin, C.: XGBoost: A Scalable Tree Boosting System, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 785–794, https://doi.org/10.1145/2939672.2939785, 2016.

Chollet, F. and others: Keras, available at: https://keras.io (last access: 12 September 2020), 2015.

Christensen, M. W., Jones, W. K., and Stier, P.: Aerosols enhance cloud lifetime and brightness along the stratus-to-cumulus transition, P. Natl. Acad. Sci. USA, 117, 17591–17598, https://doi.org/10.1073/pnas.1921231117, 2020.

Committee on the Future of Atmospheric Chemistry Research, Board on Atmospheric Sciences and Climate, Division on Earth and Life Studies, and National Academies of Sciences, Engineering, and Medicine: The Future of Atmospheric Chemistry Research: Remembering Yesterday, Understanding Today, Anticipating Tomorrow, National Academies Press, Washington, D.C., https://doi.org/10.17226/23573, 2016.

Fan, J., Yue, W., Wu, L., Zhang, F., Cai, H., Wang, X., Lu, X., and Xiang, Y.: Evaluation of SVM, ELM and four tree-based ensemble models for predicting daily reference evapotranspiration using limited meteorological data in different climates of China, Agr. Forest Meteorol., 263, 225–241, https://doi.org/10.1016/j.agrformet.2018.08.019, 2018.

Fountoukis, C. and Nenes, A.: Continued development of a cloud droplet formation parameterization for global climate models, J Geophys. Res.-Atmos., 110, D11212, https://doi.org/10.1029/2004JD005591, 2005.

Friedman, J., Hastie, T., and Tibshirani, R.: Regularization Paths for Generalized Linear Models via Coordinate Descent, J. Stat. Softw., 33, 1–22, 2010.

Gentine, P., Pritchard, M., Rasp, S., Reinaudi, G., and Yacalis, G.: Could Machine Learning Break the Convection Parameterization Deadlock?, Geopys. Res. Lett., 45, 5742–5751, https://doi.org/10.1029/2018GL078202, 2018.

Ghan, S. J., Abdul-Razzak, H., Nenes, A., Ming, Y., Liu, X., Ovchinnikov, M., Shipway, B., Meskhidze, N., Xu, J., and Shi, X.: Droplet nucleation: Physically-based parameterizations and comparative evaluation, J. Adv. Model. Earth Sy., 3, 4, https://doi.org/10.1029/2011MS000074, 2011.

Goodfellow, I., Bengio, Y., and Courville, A.: Deep Learning, MIT Press, 2016.

Hastie, T., Tibshirani, R., and Friedman, J.: The Elements of Statistical Learning, Springer New York Inc., New York, NY, USA, 2001.

Ivatt, P. D. and Evans, M. J.: Improving the prediction of an atmospheric chemistry transport model using gradient-boosted regression trees, Atmos. Chem. Phys., 20, 8063–8082, https://doi.org/10.5194/acp-20-8063-2020, 2020.

Lipponen, A., Kolehmainen, V., Romakkaniemi, S., and Kokkola, H.: Correction of approximation errors with Random Forests applied to modelling of cloud droplet formation, Geosci. Model Dev., 6, 2087–2098, https://doi.org/10.5194/gmd-6-2087-2013, 2013.

Ming, Y., Ramaswamy, V., Donner, L. J., and Phillips, V. T. J.: A New Parameterization of Cloud Droplet Activation Applicable to General Circulation Models, J. Atmos. Sci., 63, 1348–1356, https://doi.org/10.1175/JAS3686.1, 2006.

Nowack, P., Braesicke, P., Haigh, J., Abraham, N. L., Pyle, J., and Voulgarakis, A.: Using machine learning to build temperature-based ozone parameterizations for climate sensitivity simulations, Environ. Res. Lett., 13, 104016, https://doi.org/10.1088/1748-9326/aae2be, 2018.

O'Malley, T., Bursztein, E., Long, J., Chollet, F., Jin, H., Invernizzi, L., and others: Keras Tuner, available at: https://github.com/keras-team/keras-tuner (last access: 13 August 2020), 2019.

Raissi, M., Perdikaris, P., and Karniadakis, G. E.: Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, J. Computat. Phys., 378, 686–707, https://doi.org/10.1016/j.jcp.2018.10.045, 2019.

Rasp, S., Pritchard, M. S., and Gentine, P.: Deep learning to represent subgrid processes in climate models, P. Natl. Acad. Sci. USA, 115, 9684–19689, https://doi.org/10.1073/pnas.1810286115, 2018.

Reichstein, M., Camps-Valls, G., Stevens, B., Jung, M., Denzler, J., Carvalhais, N., and Prabhat: Deep learning and process understanding for data-driven Earth system science, Geophys. Res. Lett., 566, 195–204, https://doi.org/10.1038/s41586-019-0912-1, 2019.

Rothenberg, D. and Wang, C.: Metamodeling of Droplet Activation for Global Climate Models, J. Atmos. Sci., 73, 1255–1272, https://doi.org/10.1175/JAS-D-15-0223.1, 2015.

Seinfeld, J. H. and Pandis, S. N.: Atmospheric chemistry and physics: from air pollution to climate change, 3rd Edn., Wiley, Hoboken, New Jersey, 1120 pp., 2016.

Seinfeld, J. H., Bretherton, C., Carslaw, K. S., Coe, H., DeMott, P. J., Dunlea, E. J., Feingold, G., Ghan, S., Guenther, A. B., Kahn, R., Kraucunas, I., Kreidenweis, S. M., Molina, M. J., Nenes, A., Penner, J. E., Prather, K. A., Ramanathan, V., Ramaswamy, V., Rasch, P. J., Ravishankara, A. R., Rosenfeld, D., Stephens, G., and Wood, R.: Improving our fundamental understanding of the role of aerosol-cloud interactions in the climate system, P. Natl. Acad. Sci. USA, 113, 5781–5790, https://doi.org/10.1073/pnas.1514043113, 2016.

Silva, S. J.: Code for Silva et al. Aerosol Activation, Zenodo, https://doi.org/10.5281/zenodo.4319145, 2020.

Silva, S. J., Heald, C. L., Ravela, S., Mammarella, I., and Munger, J. W.: A Deep Learning Parameterization for Ozone Dry Deposition Velocities, Geophys. Res. Lett., 46, 983–989, https://doi.org/10.1029/2018GL081049, 2019.

Silva, S. J., Heald, C. L., and Guenther, A. B.: Development of a reduced-complexity plant canopy physics surrogate model for use in chemical transport models: a case study with GEOS-Chem v12.3.0, Geosci. Model Dev., 13, 2569–2585, https://doi.org/10.5194/gmd-13-2569-2020, 2020a.

Silva, S. J., Ridley, D. A., and Heald, C. L.: Exploring the Constraints on Simulated Aerosol Sources and Transport Across the North Atlantic With Island-Based Sun Photometers, Earth Space Sci., 7, e2020EA001392, https://doi.org/10.1029/2020EA001392, 2020b.

Twomey, S.: The nuclei of natural cloud formation part II: The supersaturation in natural clouds and the variation of cloud droplet concentration, Geofis. Pur. Appl., 43, 243–249, https://doi.org/10.1007/BF01993560, 1959.

Twomey, S.: Pollution and the planetary albedo, Atmos. Environ., 8, 1251–1256, https://doi.org/10.1016/0004-6981(74)90004-3, 1974.

Twomey, S.: The Influence of Pollution on the Shortwave Albedo of Clouds, J. Atmos. Sci., 34, 1149–1152, https://doi.org/10.1175/1520-0469(1977)034<1149:TIOPOT>2.0.CO;2, 1977.

Wallace, J. M. and Hobbs, P. V.: Atmospheric Science: An Introductory Survey, Elsevier Academic Press, 2006.

Zhao, W. L., Gentine, P., Reichstein, M., Zhang, Y., Zhou, S., Wen, Y., Lin, C., Li, X., and Qiu, G. Y.: Physics-Constrained Machine Learning of Evapotranspiration, Geophys. Res. Lett., 46, 14496–14507, https://doi.org/10.1029/2019GL085291, 2019.