

JGR Atmospheres

RESEARCH ARTICLE

10.1029/2020JD032759

Key Points:

- We create a machine-learned surrogate model of an atmospheric chemical solver which operates orders of magnitude faster than the original
- We use a recurrent training regime to yield models that improve numerical stability relative to previous efforts
- Our models can reversibly compress the number of modeled chemical species by greater than 80% without decreasing accuracy

Supporting Information:

- Supporting Information S1

Correspondence to:

C. W. Tessum,
ctessum@illinois.edu

Citation:



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

Received 22 MAR 2020

Accepted 22 SEP 2020

Accepted article online 17 NOV 2020

Toward Stable, General Machine-Learned Models of the Atmospheric Chemical System

Makoto M. Kelp¹ , Daniel J. Jacob¹, J. Nathan Kutz², Julian D. Marshall³, and Christopher W. Tessum⁴ 

¹Department of Earth and Planetary Sciences, Harvard University, Cambridge, MA, USA, ²Department of Applied Mathematics, University of Washington, Seattle, WA, USA, ³Department of Civil and Environmental Engineering, University of Washington, Seattle, WA, USA, ⁴Department of Civil and Environmental Engineering, University of Illinois at Urbana-Champaign, Urbana, IL, USA

Abstract Atmospheric chemistry models—components in models that simulate air pollution and climate change—are computationally expensive. Previous studies have shown that machine-learned atmospheric chemical solvers can be orders of magnitude faster than traditional integration methods but tend to suffer from numerical instability. Here, we present a modeling framework that reduces error accumulation compared to previous work while maintaining computational efficiency. Our approach is novel in that it (1) uses a recurrent training regime that results in extended (>1 week) simulations without exponential error accumulation and (2) can reversibly compress the number of modeled chemical species by >80% without further decreasing accuracy. We observe an ~260× speedup (~1,900× with specialized hardware) compared to the traditional solver. We use random initial conditions in training to promote general applicability across a wide range of atmospheric conditions. For ozone (concentrations ranging from 0–70 ppb), our model predictions over a 24-hr simulation period match those of the reference solver with median error of 2.7 and <19 ppb error across 99% of simulations initialized with random noise. Error can be significantly higher in the remaining 1% of simulations, which include extreme concentration fluctuations simulated by the reference model. Results are similar for total particulate matter (median error of 16 and <32 $\mu\text{g}/\text{m}^3$ across 99% of simulations with concentrations ranging from 0–150 $\mu\text{g}/\text{m}^3$). Finally, we discuss practical implications of our modeling framework and next steps for improvements. The machine learning models described here are not yet replacements for traditional chemistry solvers but represent a step toward that goal.

1. Introduction

The U.S. National Research Council has identified the inclusion of atmospheric chemistry in Earth system models (ESMs) as one of its five priority science areas for earth system modeling (NRC, 2012, 2016). Atmospheric chemistry mechanisms for use in large-scale modeling typically include hundreds of gas and aerosol species coupled by chemical reactions on time scales ranging from milliseconds to years. Forward integration of these chemical systems using traditional methods is computationally intensive (Brasseur & Jacob, 2017), typically doubling the cost of an ESM simulation (Hu et al., 2018). This puts the inclusion of atmospheric chemistry in tension with other computationally intensive ESM priorities such as increased spatial resolution. The current slowdown in the rate of increase in the speed of computer central processing units (CPUs)—the potential “end of Moore’s law” (Theis & Wong, 2017)—underscores the need for computationally efficient approaches.

Relatedly, policymakers are currently searching for ways to reduce the computational intensity of estimating the impacts of air quality policies (Industrial Economics, 2019). This type of impact assessment is typically done with regional air quality models (AQMs), of which the chemical mechanism is the most computationally intensive component. A number of alternative approaches exist to streamline the creation of these estimates, including reducing the number of species in chemical mechanisms (Whitehouse et al., 2004); adapting a chemical mechanism locally to neglect minor species (Santillana et al., 2010); or using heuristics (Apte et al., 2012; Bare, 2002), statistical emulators (Fann et al., 2009; Heo et al., 2016; Lu & Ricciuto, 2019), or simplified AQMs (Muller, 2014; Tessum et al., 2017). However, none of these approaches can fully replace AQMs (Industrial Economics, 2019).

The above considerations suggest that the development of a class of algorithms that dramatically reduces the computational intensity of integrating the system of differential equations that represents the atmospheric chemical system without a major loss in accuracy could be a critical enabler for developments in multiple fields.

Although the integration of differential equations is a mature field, a quickly developing area of inquiry—sometimes called “physics-informed” machine learning (ML) (Reichstein et al., 2019)—may prove transformative. For example, Bar-Sinai et al. (2019) trained neural networks to estimate spatial gradients for the solution of partial differential equations, allowing explicit Euler integration of the system at coarser resolutions than possible with a traditional integrator. Champion et al. (2019) used ML to discover efficient governing equations for dynamical systems. Raissi et al. (2017) found that learning Navier-Stokes integration closure parameters using a neural network outperformed standard approaches. Additional advances have been made by enforcing or encouraging physical constraints and symmetries, such as the conservation of mass and energy, temporal coherence, Lyapunov stability, and Newton's laws (Erichson et al., 2019; Karpatne et al., 2017; Kim et al., 2018; Lusch et al., 2018; Reichstein et al., 2019; Stewart & Ermon, 2016; Wang et al., 2017; Xie et al., 2018).

Owing to the high dimensionality of chemical mechanisms in both their inputs and outputs (hundreds of chemical species and relevant physical variables), they are a challenging target for ML surrogate modeling. Keller and Evans (2019) created a random forest-based integrator that could operate accurately in an AQM for relatively short simulations under conditions similar to the simulation on which it was trained. However, the random forest surrogate model was slower than the reference model, and exponential error growth occurred during long-range forecasts and under conditions different than the training simulation. Kelp et al. (2018) trained a neural network to emulate the CBM-Z gas-phase chemical mechanism (Zaveri & Peters, 1999) that can accurately predict changes in concentrations an hour in the future with a single integration time step and can do so orders of magnitude faster than the reference solver described by Zaveri and Peters (1999). However, extended simulations usually ended with exponential error propagation resulting in predictions of near-infinite concentrations.

Here, we explore the opportunities for improvement identified by previous work (Keller & Evans, 2019; Kelp et al., 2018) to create a surrogate model that (1) can make accurate predictions over longer time periods than it was trained on, (2) can make accurate predictions for a wide range of atmospheric environments, and (3) never experiences exponential error propagation within the time period it was trained on. In effect, we create a model that is more “stable” and “general” than previous results, two key prerequisites for routine use in atmospheric modeling. We do this by leveraging three key innovations: (1) the use of a recurrent training regime (Brenowitz & Bretherton, 2018; McGibbon & Bretherton, 2019) to train the model to represent chemical reactions across multiple time scales, (2) the use of an encoder-operator-decoder framework to reduce the dimensionality of the chemical system (Lee & Carlberg, 2018; Pan & Duraisamy, 2020; Regazzoni et al., 2019), and (3) the use of a weighted optimization metric to create surrogate models that specialize in the prediction of key air quality metrics such as ozone and total particulate matter (PM).

2. Materials and Methods

2.1. Reference Model

As a reference chemical mechanism, we use the Carbon Bond Mechanism Z coupled to the Model for Simulating Aerosol Interactions and Chemistry (Zaveri et al., 2008; Zaveri & Peters, 1999)—CBM-Z/MOSAIC—which simulates tropospheric gas-phase (77 species) and aerosol (24 species) chemistry, respectively, including a total of 164 reactions. We use a standalone version of CBM-Z/MOSAIC in this work (a “box model”) which does not include emission, deposition, advection, or any atmospheric processes other than chemistry and microphysics. We configure the CBM-Z/MOSAIC box model as shown in Table S1 in the supporting information, allowing the initial conditions of the 101 chemical species and physical parameters including the temperature, pressure, relative humidity, and the cosine of the solar zenith angle to independently and randomly vary within the specified ranges to create training and testing data for our surrogate model. We use CBM-Z/MOSAIC as a ground truth for exploring methods for surrogate modeling, while acknowledging that CBM-Z/MOSAIC itself may inaccurately represent chemical dynamics under some atmospheric conditions.

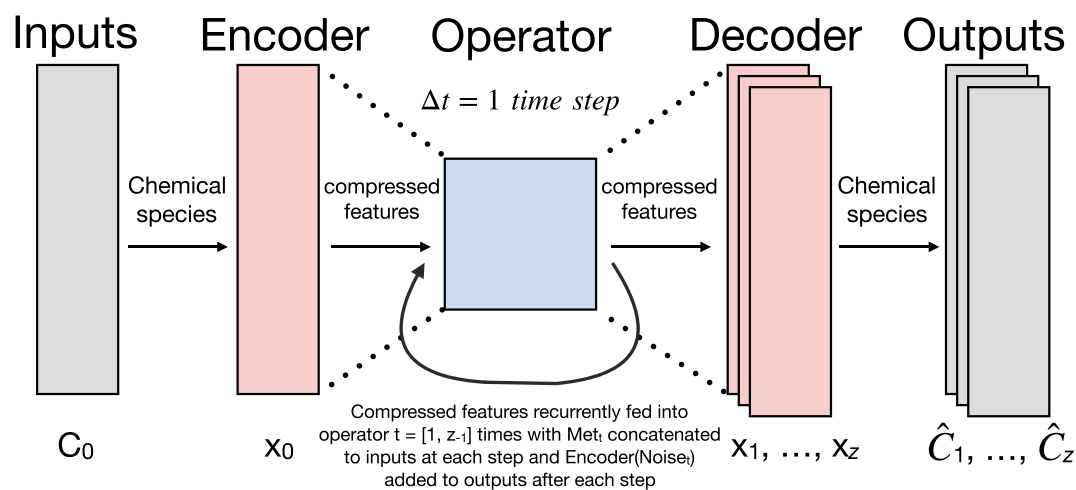


Figure 1. The encoder-operator-decoder neural network framework used here. Initial input concentrations C_0 are compressed by the encoder to a latent representation x_0 , which is recurrently fed through the operator z times—along with meteorological parameters Met_{0-z} —to integrate the system forward over z time steps. The decoder translates predictions from the latent space back to the original chemical species over the z time steps.

2.2. ML Model Architecture

A glossary of ML terminology used in this work is provided in Table S2.

We create a ML surrogate model of CBM-Z/MOSAIC. The CBM-Z/MOSAIC model is a nonlinear function with multiple continuous input variables and multiple continuous output variables. This type of function can be empirically modeled using multitarget regression, where “regression” refers to continuous rather than categorical outputs and “multitarget” refers to more than one output. Neural networks (McCulloch & Pitts, 1943; Schmidhuber, 2015) are one class of algorithm that can perform multitarget regression.

Figure 1 shows the model architecture used in this work. It consists of three main components: encoder, operator, and decoder, each of which are neural networks. Briefly, the encoder creates a reduced-dimension latent representation of the input data, the operator integrates the reduced-dimension system forward in time, and the decoder transforms the system from the reduced-dimension space back to the original list of chemical species. After the initial concentrations are encoded, the operator is applied recurrently to integrate the system forward by multiple steps.

In other words, during training, the encoder learns to map the chemical species to a compressed representation, while the operator jointly learns to integrate the compressed representation forward in time, while the decoder jointly learns to convert the compressed representation back to the original species. This is all driven by gradient descent optimization.

Text S1 contains a detailed description of the model architecture.

2.3. Training Data

We run the CBM-Z/MOSAIC model to create a training data set sampling the 105-dimensional space of input variables including concentrations for the 101 species in the mechanism and four meteorological parameters: temperature, pressure, relative humidity, and cosine of solar zenith angle. The integration time step of the CBM-Z/MOSAIC solver is 5 min, output is archived every hour, and the simulation is conducted over 24 hr ($z = 24$). Temperature, pressure, and relative humidity are held fixed to the randomly determined initial values over the 24-hr simulation, while the cosine of the solar zenith angle varies with time of day and day of year for a latitude of 40° with randomly sampled start dates and times.

To create training examples, we first use Latin hypercube sampling to generate 40 million sets of initial conditions for each chemical species within typical tropospheric ranges (Table S3), as well as values for temperature, pressure, and relative humidity and a starting date and time for calculating the evolution of the solar zenith angle. Figures S1 and S2 show the distributions of the input data.

We also create a separate testing data set where random noise is added to the chemical concentrations at every hourly time step, to investigate possible effects of operator splitting between chemistry and other processes that would occur in an ESM or AQM (Figure S3). The noise is sampled with a boxcar probability density function (PDF) extending $\pm 5\%$ of the maximum initial value for that species as specified by the input ranges and clipped to avoid negative concentrations. The same perturbations are added to the neural network and to the CBM-Z/MOSAIC simulation, with the neural network perturbations being compressed by the encoder before addition.

For each of the training data sets above, we create additional validation and testing data sets of one million examples each. The validation data sets are used for the hyperparameter optimization described below, and the testing data sets are used to produce the results reported below. Finally, we create 1-week (168 hr) and 1-month (720 hr) simulation data sets to examine the neural network model's ability to make predictions over time periods substantially longer than the 24 hr it was trained on.

When training and testing the ozone-specialized model (described in next section), we remove all examples from the training and test data set where CBM-Z/MOSAIC-simulated concentrations go above our ozone concentration initial range of 200 ppb in order to avoid evaluating the model on conditions outside of the range of concentrations on which it was trained.

2.4. Training Procedure

We train the ML surrogate model to emulate the CBM-Z/MOSAIC using supervised learning. Using the Adam (Kingma & Ba, 2014) stochastic gradient descent optimization algorithm, we minimize the mean square error (MSE) between ML and CBM-Z/MOSAIC predictions across the 101 chemical species and the 24 one-hour time steps during training.

For each example at each training step, the encoder is applied to the initial conditions, the operator is recurrently applied 24 times in sequence, and the decoder is applied to the resulting outputs of each operator step for comparison to CBM-Z/MOSAIC results. We refer to this as “recurrent training” because we are training the model to recurrently integrate the chemical system to predict its evolution.

Our MSE training optimization metric includes all CBM-Z/MOSAIC chemical species, but not all chemical species are equally important. Often, ESM and AQM users are interested in predictions of a small number of species for regulatory, health, or climate forcing reasons; the remaining species are only included in the simulation to support predictions of the species of interest. In an ML context, we can represent this by giving errors in predictions of certain species a greater weight in the optimization algorithm than is given to other species. This procedure allows any number of specialized models to be created from a single training data set. Here, we explore three:

1. Nonspecialized: Errors in all species contribute equally to the optimization metric. Typical atmospheric ranges of different species vary by orders of magnitude (Table S3); therefore, this metric results in the algorithm prioritizing error minimization for species with the largest atmospheric concentrations (e.g., CH₄, CO, and ozone).
2. Ozone specialized: Error in predicting ozone concentration is weighted by $10^4\times$ more than errors in predicting other species. This prioritizes performance in predicting ozone concentrations.
3. PM specialized: Error in predicting total PM concentration is weighted by $10^4\times$ more than errors in predicting other species. CBM-Z/MOSAIC includes multiple PM species; total PM is calculated using Equation S2.

We performed hyperparameter optimization using Scikit-optimize Bayesian optimization with the Gaussian Process algorithm (Head et al., 2018), run for 400 iterations with each iteration run for three training epochs, using a fixed random seed across all iterations. Hyperparameters were selected from the search space reported in Table S4. Text S2 contains further detail on the selected hyperparameters in each case.

We independently optimize the hyperparameters for the three specialized models in the preceding paragraph as well as for ozone specialized models with (1) ozone trained on a single prediction step rather than 24 steps and (2) ozone trained on a week of data (168 steps).

After hyperparameters are selected, we fully train each model with an initial learning rate of 1.0×10^{-4} with learning-rate decay (You et al., 2019) occurring every time the test set error plateaus for 10 epochs. We use

early stopping (Li et al., 2019) to halt model training when test set error stops improving for 15 epochs, to prevent overfitting when training the neural network algorithm. We used a stochastic gradient descent batch size of 1,024. Performing hyperparameter optimization took approximately 100 hr, and fully training a single model took approximately 10 hr, on a single Nvidia Tesla K80 graphics processing unit (GPU). We use the Python library Keras (Chollet, 2017) with the TensorFlow 1.12.0 (Abadi et al., 2016) backend for all neural network model experiments. Neural network model code is available at Zenodo (<https://zenodo.org/record/3711059#.Xm0xrhNKjfZ>).

2.5. Effect of Latent Representation Dimensionality

We investigate the effect of compressing CBM-Z/MOSAIC's 101 species into a reduced-dimension representation to test the hypothesis that it is possible to faithfully represent the chemical system with fewer than 101 dimensions. We conduct a set of simulations in which we encode the original 101 chemical species down to 64, 32, 16, 8, and 4 compressed species and compare the testing error against the original 101-dimensional space (i.e., no compression and no autoencoder used). We train the models using a fixed random seed for choosing pseudo-random initial conditions for neural network parameters and using the results of hyperparameter search described above. Even with a fixed random seed, the trained model result is affected by stochastic GPU reduction operations; we train six nominally identical instances of each model to explore this variability.

2.6. Computational Speed

The use of ML has shown potential to solve differential equations orders of magnitude faster than standard numerical integration (Kelp et al., 2018; Kim et al., 2018; Rasp et al., 2018). However, this is not guaranteed (Keller & Evans, 2019). To compare the performance of the ML approach here to the reference model, we compile CBM-Z/MOSAIC with gfortran using the default optimization level of zero. We implement the ML model using the Keras library with the TensorFlow ML backend with the default compiler optimization included in the downloadable binary runtime. For CBM-Z/MOSAIC we use a single CPU core; for the neural network algorithm we test three configurations: one CPU core, eight CPU cores, and one GPU (NVIDIA Tesla V100).

We compare model run times between CBM-Z/MOSAIC and the neural network algorithm for simulating chemistry in one million independent grid cells (initial conditions) during a simulation period of one model time step (CBM-Z/MOSAIC: 5 min; neural network: 1 hr). This number of grid cells approximately corresponds to one vertical layer of a chemical transport model simulation over North America at $0.25^\circ \times 0.3125^\circ$ horizontal resolution or to a global simulation at $2^\circ \times 2.5^\circ$ horizontal resolution with 72 vertical layers.

3. Results

Consistent with opportunities for improvement identified in previous work (Keller & Evans, 2019; Kelp et al., 2018), we are interested in creating a chemical mechanism surrogate model that (1) can make accurate predictions over longer time periods than it was trained on, (2) can make accurate predictions in a wide range of atmospheric environments, and (3) never incurs exponential error growth. We evaluate our models for these characteristics by testing models on (1) data with a wider range of conditions than we would expect to occur in an ESM or AQM simulation by using initial conditions with no correlations among chemical species; (2) data with different characteristics than they were trained on by adding disruptive noise between integration steps; and (3) data representing longer simulations than they were trained on. We use testing data sets with 1 million examples and highlight the highest error experienced among all examples to act as a surrogate for the worst-case example of what might happen if the model was running in an ESM or AQM. Although the model as configured here makes predictions for 101 chemical species, for the sake of concision we focus our presentation and discussion of results on predictions of tropospheric ozone and total PM, which are of health and regulatory interest. Results for additional chemical species are available in the supporting information.

Results of hyperparameter and architecture optimization are found in Table S5.

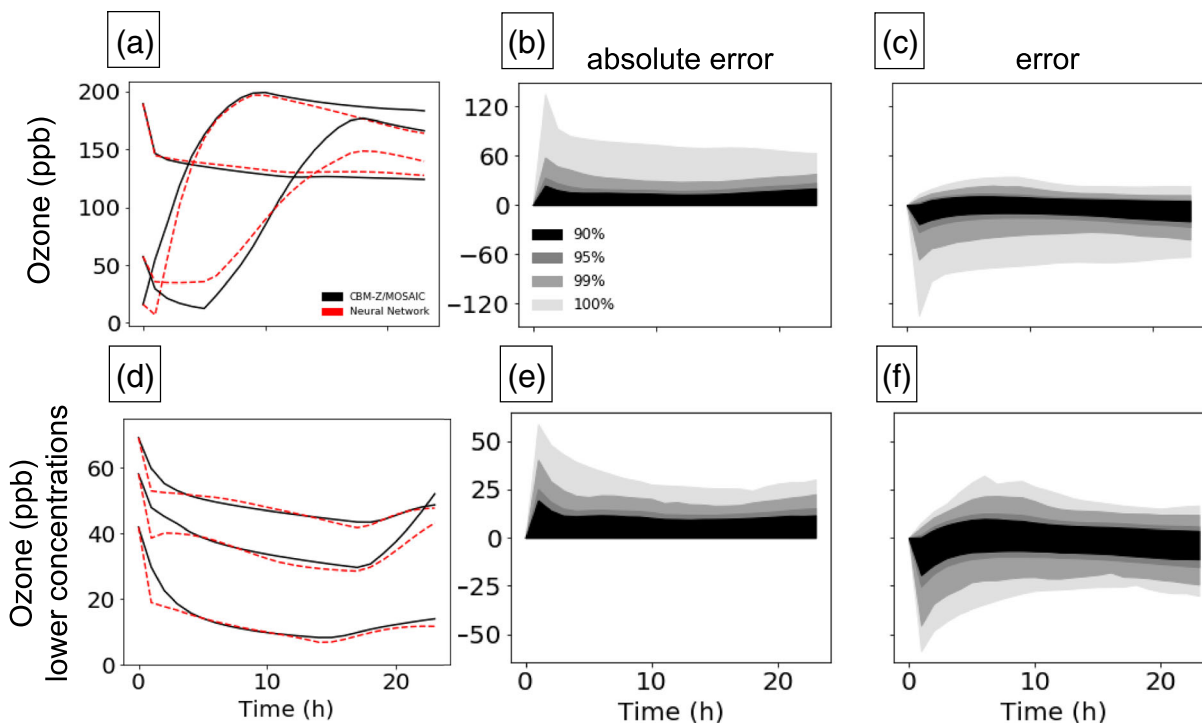


Figure 2. Twenty-four-hour ozone simulations for the full range of concentrations (upper row) and a lower concentration subset (lower row). The left column (panels a and d) shows CBM-Z/MOSAIC (black solid line) and ML model (red dashed line) trajectories for three representative simulations. The middle and right columns show absolute error (panels b and e) and error (panels c and f) percentiles for simulations with random initial conditions. Shaded percentile values represent the fraction of the simulations with lower error than the value shown.

3.1. Ozone Prioritized Model: 24-hr Simulations

Figure 2 shows the evolution of the chemical system over a period of 24 hr as predicted by CBM-Z/MOSAIC and the ozone-prioritized ML model. In Figure 2a, trajectories are shown for three representative individual simulations. These trajectories illustrate that the ML model is qualitatively able to reproduce CBM-Z/MOSAIC predictions of the nonlinear interactions between ozone and the other chemical species and meteorological parameters. The model also makes simultaneous predictions for the 100 other species; trajectories for several species for randomly chosen initial conditions are shown in Figure S4. These additional species are not prioritized in the optimization metric, so performance is expected to be lower.

Figures 2b and 2c show statistical distributions in absolute error (the absolute value of the difference between the ML model and CBM-Z/MOSAIC; Figure 2b) and error (the difference between the ML model and CBM-Z/MOSAIC; Figure 2c) for one million randomly initialized simulations. With uniformly distributed initial ozone concentrations ranging from 0–200 ppb, our ML model predictions differ from CBM-Z/MOSAIC predictions over 24 hr with median absolute error of 4.0 ppb. Absolute error was less than 13.4 ppb for 90% of all comparison points (one million simulations \times 24 time steps = 24 million comparison points), and absolute error was less than 36.3 ppb in 99% of all comparisons. The maximum absolute error was 142 ppb. Statistics are also shown in Table S6.

Because the individual grid cells in an ESM or AQM all interact with each other, a large prediction error in any one grid cell at any one point in time can adversely affect an entire simulation. Therefore, we examine the conditions under which our neural network predictions are the worst. We find that the largest errors are underpredictions that occur when ozone concentrations predicted by CBM-Z/MOSAIC are at their highest (~150 ppb; Figure S5). The fact that the most extreme surrogate model errors are associated with the most extreme reference model predictions suggests that when the surrogate model fails, it does so in a way that is predictable and understandable. If the largest surrogate model error was instead associated with a relatively normal reference model simulation, that would be more concerning. Reducing these errors is an area requiring further research.

Figures 2d–2f represent the subset of test simulations (28%) in Figures 2a–2c where the ozone concentration predicted by CBM-Z/MOSAIC is within the US EPA daily maximum 8-hr ozone standard of 70 ppb (U.S. EPA, 2006) with an average ozone concentration of 24.8 ppb across these simulations. For these 28% of simulations, the ML model predictions differ from CBM-Z/MOSAIC predictions over 24 hr with median absolute error of 2.7 ppb. Absolute error was less than 8.1 ppb in 90% of all comparisons, and absolute error was less than 18.7 ppb in 99% of all simulations. The maximum absolute error observed among the 280,000 simulations was 66.9 ppb (Table S7).

Overall, even in the least accurate simulations, error does not substantially increase after the first ~5 hr (Figures 2b, 2c, 2e, and 2f).

To test if error stabilization over time is the direct result of the use of the recurrent training regime, we train a neural network model using model architecture and training data identical to that above (including its own hyperparameter optimization), but training the model to predict concentrations for only 1 hr, rather than recursively over 24 hr. The corresponding error statistics for this alternative model are in Table S8. In sharp contrast to the model trained with the recursive training regime (Figure 2), the model trained to predict one step ahead accumulates error exponentially when used to predict multiple steps in sequence (Figure S6). This suggests that our recursive training approach is a key factor in stabilizing error accumulation over a 24-hr simulation.

Finally, ML model performance does not significantly decrease ($p = 0.8$) when tested on a data set containing noise between time steps (Figure S7; Table S9). This suggests that the current approach may be robust to distribution shifts induced by operator splitting, but further research with a 3-D atmospheric model is needed to confirm.

3.2. Ozone-Prioritized Model: 7- and 30-Day Simulations

Ultimately, any atmospheric chemistry model suitable for general use in an ESM or AQM needs to make predictions that are accurate for more than 24 hr. We find that the original ozone-prioritized model avoids exponential error accumulation when run forward a week into the future (Figures 3a–3c; Table S10), though error gradually increases after 24 hr. Training the ML model using week-long simulations (168 hr instead of 24 hr) reduces error accumulation over time but results in larger errors at the beginning of simulations (Figures 3d–3f; Table S11).

The model trained on week-long simulations can make accurate predictions for about 1.5 weeks but tends to suffer from exponential error accumulation after 2 weeks (Figure S8). Overall, training the ML model on longer-duration simulations increases numerical stability, but it may also reduce its ability to accurately predict the large concentration changes that often occur at the very beginning of the simulations (Figures 3b and 3e).

3.3. Alternate Model Specialization

In section 2.5 we discussed the relationship between the chosen training optimization metric and the relative performance of the resulting ML model in predicting the concentrations of different chemical species. The results above have focused on an ozone-specialized model; here we explore the difference between a PM-specialized model and a nonspecialized model in predicting total PM concentrations.

Figure 4 compares the performance of neural network models with a training error metric that equally weights all species (Figures 4a–4c) versus specifically prioritizing the 19 aerosol species that comprise total PM (Figures 4d–4f) as calculated in Equation S2. The model architectures and hyperparameters are selected based on the results of the optimization routine in section 3.1. Both models avoid exponential error growth, but error and bias are much lower for the PM-specialized model (Table S12). Similar performance gains are found when restricting comparisons to simulations with total PM concentrations within the US EPA 24-hr average PM_{10} standard of $150 \mu\text{g}/\text{m}^3$ (U.S. EPA, 2009) (Figure S9; Table S12). By altering the weighting of different species in the error metric, we can direct the optimization algorithm to prioritize accuracy in some species at the expense of the accuracy in other species. In practice, it is common for air quality and climate modelers to be interested in predictions of both ozone and PM. We do not include here a model that is specialized for both, but it would be straightforward to create one.

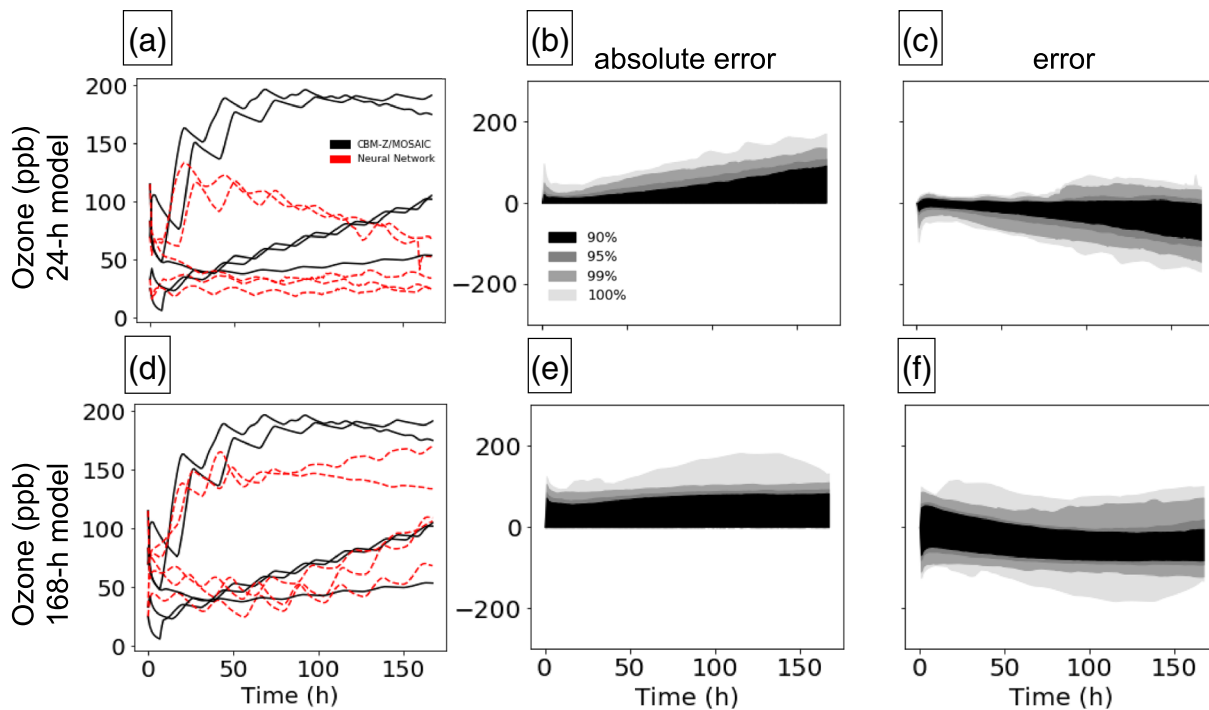


Figure 3. Week-long ozone predictions by models trained on 1-day (upper row) and 7-day (lower row) simulations. The left column (panels a and d) shows CBM-Z/MOSAIC (black solid line) and ML model (red dashed line) trajectories for three representative simulations. The middle and right columns show absolute error (panels b and e) and error percentiles (panels c and f) for simulations with random initial conditions. Shaded percentile values represent the fraction of the simulations with lower error than the value shown.

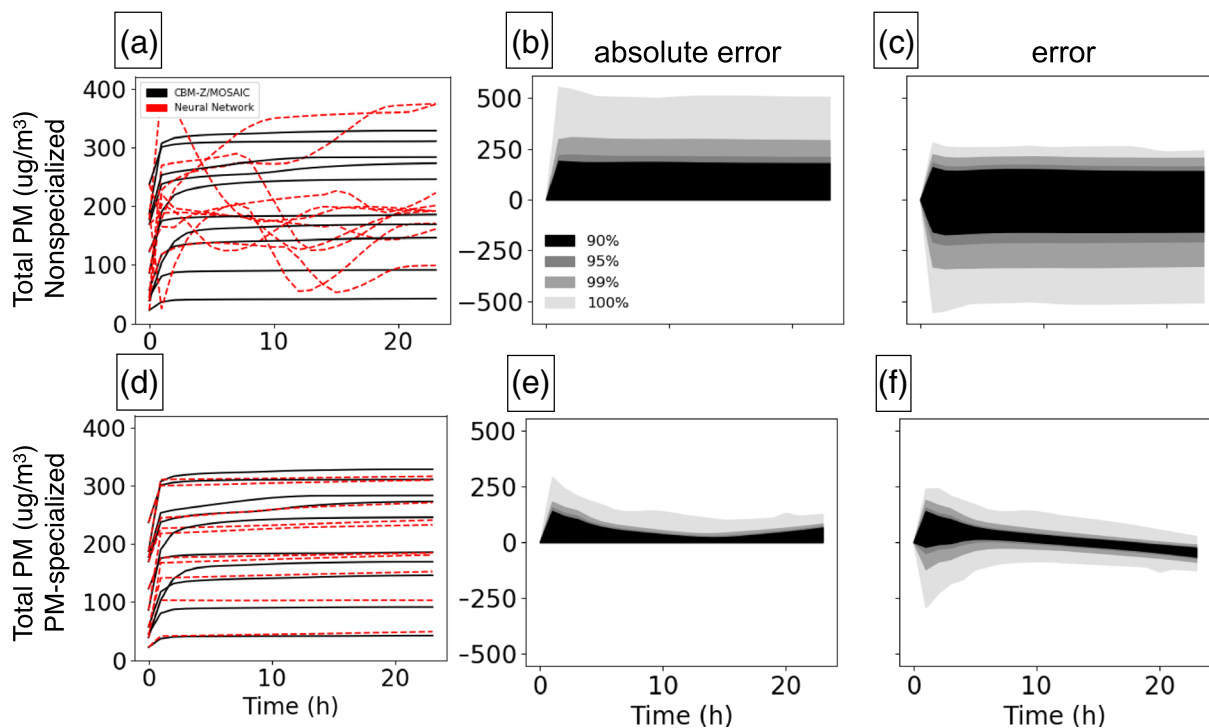


Figure 4. Twenty-four-hour total PM predictions by models trained with a nonspecialized (top row) versus a PM-specialized error metric. The left column (panels a and d) shows CBM-Z/MOSAIC (black solid line) and ML model (red dashed line) trajectories for 10 randomly chosen simulations. The middle and right columns show absolute error (panels b and e) and error percentiles (panels c and f) for simulations with random initial conditions. Shaded percentile values represent the fraction of the simulations with lower error than the value shown.

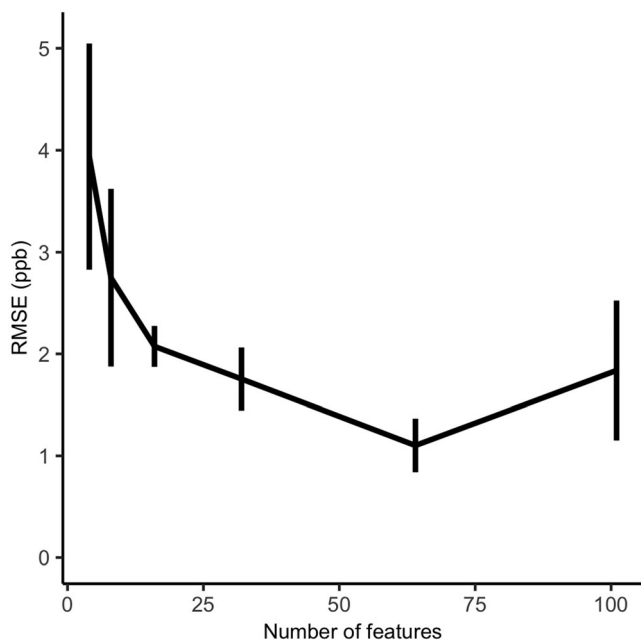


Figure 5. Test set RMSE for ozone as a function of the number of compressed features used by the dynamic operator (shown in Figure 1). Bars represent the median and 95% confidence interval of RMSE for six nominally identical versions of the ML model (results differ owing to stochastic GPU operations). RMSE calculations for all models use the same one million randomly initialized test simulations.

3.4. Effect of Latent Representation Dimensionality

Here we explore the effect of varying the size of the latent representation (i.e., number of compressed chemical species) on model accuracy. Figure 5 shows the relationship between the compressed features and root mean square error (RMSE) between the ML model and CBM-Z/MOSAIC across one million 24-hr test simulations.

Compressing to 64 features produces the lowest training error of the options we tested. Decreasing the number of features below 16 typically increases total error as compared to no compression. With fewer than 16 features, error sharply increases, indicating too few features to faithfully represent the system.

Although the best model performance is for a latent space with 64 features, we are able to compress the 101-dimensional system down into a latent representation of 16 features without considerable loss in performance. Figure S10 and Table S13 show the dominant components of each of the features in our 16-feature ozone specialized model. In some cases, it is possible to physically interpret makeup of these latent features, for example, those containing large contributions from chemical species known to regulate tropospheric ozone formation such as HO and HO₂ (Feature 9) and NO (Feature 12). However, other features are not as readily interpretable.

It is possible that differences in randomly chosen initial weights and biases in the neural network may substantially affect the resulting learned features. Exploring this possibility—and exploring why some features are more interpretable than others—are areas for future research.

3.5. Computational Performance

Figure 6 shows that our ML model performs integration steps ~260× faster than CBM-Z/MOSAIC when running on the same hardware (Table S16). The ML model is also able to perform multiple simulations in parallel in the same computer operating system process using either a multicore CPU or a GPU. When using the GPU, the ML model is ~1,900× faster than CBM-Z/MOSAIC.

Although Figure 6 shows the amount of time required to integrate one time step, time steps for the ML model are much longer than for CBM-Z/MOSAIC—1 hr versus 5 min, respectively. Additionally, part of the time required by the ML model is for running the encoder and decoder, which may not need to be run in every step of an extended simulation. If we compare the time required to perform a 24-hr simulation instead of a single integration step, the ML model is ~3,700× faster than CBM-Z/MOSAIC when using the same hardware.

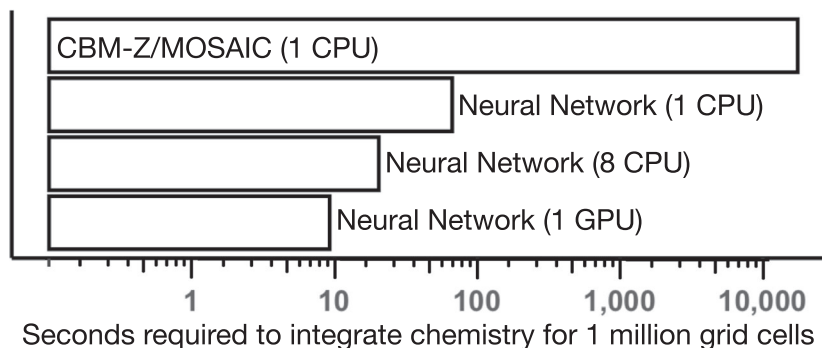


Figure 6. CBM-Z/MOSAIC and neural network timing results. Time required for one million independent integration steps using either CBM-Z/MOSAIC using one CPU core, the neural network model using one or eight CPU cores, or the neural network model using one GPU. Model run times are in Table S16.

4. Conclusions

As discussed above, our overarching goal is to create a fast, accurate, stable, and general machine-learned representation of atmospheric chemistry—one that can make accurate, long-term predictions in a wide range of atmospheric environments. The results here represent a step toward that goal.

Our ML model integrates the CBM-Z/MOSAIC chemical system 260× faster (with a GPU: ~1,900×) than the standard numerical solver. The neural network model contains an encoder that can compress the 101 chemical species in the mechanism to as few as 16 features, which could potentially reduce the working memory requirements of ESMs and AQMs and reduce the computational intensity of the nonchemistry model operations. Future work is needed to confirm whether the compressed features we create can be compatible with the other operators—such as advection—in an ESM or AQM.

Predictive accuracy varies by chemical species; for ozone, with initial concentrations ranging from 0–70 ppb, neural network predictions differed from CBM-Z/MOSAIC predictions over 24 hr with median absolute error of 2.7 ppb. In 90% of all comparisons, absolute errors were less than 8.1 ppb, and in 99% of all comparisons, absolute errors were less than 18.7 ppb. The maximum absolute error encountered is 66.9 ppb, with the largest errors occurring as underpredictions of some of the most extreme concentration fluctuations predicted by CBM-Z/MOSAIC. Perhaps these large prediction errors in a small minority of cases result from the use of random initial conditions at locations in parameter space that would be unlikely to occur in the real atmosphere. Additionally, choosing appropriate concentration ranges for each of the 101 chemical species is a challenge with the approach used here. Using ESM or AQM simulations to generate the training data would avoid these issues, but doing so would also reduce the ability of the model to detect conditions not well represented in the training data, where the neural network architecture used here cannot be expected to work well. Exploring these trade-offs is an area for future research.

Regardless, for simulations of 1.5 weeks or less, error growth is limited. The recurrent training procedure used here is a key factor responsible for increased stability relative to previous work because the dynamics of chemical systems are controlled by various reactions that occur across a wide range of time scales (Lowe & Tomlin, 2000). Adapting the machine-learning approach to account for this type of domain-specific consideration leads to improved performance.

We train and test our ML surrogate models using independent, uniformly distributed noise within typical atmospheric ranges as initial conditions for CBM-Z/MOSAIC simulations. Because of this, if we were to implement the models shown here in an ESM or AQM, we would expect them to perform with accuracy similar to the results shown here as long as model concentrations stay within the ranges of initial conditions in our training data sets. This approach makes clear which points in parameter space are represented in the training and testing data; models could easily be programmed to halt if concentrations ever exceed those ranges. In other words, we could ensure that the training data ranges include all the atmospheric conditions a model would be likely to encounter, and then program the model to throw an error if it ever ended up outside of those conditions. If successful, this approach will create models with behavior that is more predictable than is possible using alternative methods of creating training data sets that include correlations between variables. The use of training and testing data with correlated variables would make it much more difficult for an ESM or AQM to determine whether a given location in the parameter space was well represented in the training and testing data.

Although the results here represent a step forward from previous literature, work still remains to reach the overarching goal described above. First, although we create models that can emulate longer-duration simulations than they are trained on, they still tend to become numerically unstable if the simulation is made long enough (Figure S8). We hypothesize this instability may happen because the simulated concentrations of some species in CBM-Z/MOSAIC tend to drift farther and farther from the densely sampled initial concentration range the longer the simulation continues, thereby requiring the ML model to make predictions in parts of the parameter space where it has not seen many training examples. If this is the case, fixing the issue may be as simple as generating the training data in a way that constrains simulated concentrations within the initial ranges. Nevertheless, this current ML framework curtails error propagation over short integration windows and may offer promise for applications of short-term forecasting and chemical data assimilation.

Second, we find that training our ML models on longer-duration simulations increases their numerical stability but also reduces their ability to accurately predict the large concentration changes that often occur at the very beginning of the simulations (Figures 3b and 3e). It is possible that using a ML model architecture with a specialized inductive bias (i.e., built-in assumptions about the system the model is emulating) may make it easier to learn dynamics across multiple time scales.

Finally, although we hypothesize that our model training procedure will result in chemistry models that work in an ESM or AQM with performance similar to that shown here, future work is required to implement models such as those presented here in ESMs and AQMs and evaluate their performance under more realistic conditions.

Data Availability Statement

The neural network model code and scripts used to generate training and testing data are available at Zenodo (<https://zenodo.org/record/4075312#.X4B-xZNKgUs>). The CBM-Z/MOSAIC box model code is available according to instructions in Zaveri et al. (2008).

Acknowledgments

This publication was developed with support from Assistance Agreements No. R835873 (the Center for Air, Climate and Energy Solutions (CACES)) and No. RD-84001201-0 awarded by the U.S. Environmental Protection Agency. It has not been formally reviewed by EPA. The views expressed in this document are solely those of authors and do not necessarily reflect those of the Agency. EPA does not endorse any products or commercial services mentioned in this publication. The authors thank Rahul Zaveri for use of and advice regarding CBM-Z/MOSAIC.

References

- Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., et al. (2016). TensorFlow: A system for large-scale machine learning. Paper presented at 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16). USENIX Association.
- Apte, J. S., Bombrun, E., Marshall, J. D., & Nazaroff, W. W. (2012). Global intraurban intake fractions for primary air pollutants from vehicles and other distributed sources. *Environmental Science & Technology*, *46*(6), 3415–3423. <https://doi.org/10.1021/es204021h>
- Bare, J. C. (2002). Traci. *Journal of Industrial Ecology*, *6*(3–4), 49–78. <https://doi.org/10.1162/108819802766269539>
- Bar-Sinai, Y., Hoyer, S., Hickey, J., & Brenner, M. P. (2019). Learning data-driven discretizations for partial differential equations. *Proceedings of the National Academy of Sciences*, *116*(31), 15,344–15,349. <https://doi.org/10.1073/pnas.1814058116>
- Brasseur, G. P., & Jacob, D. J. (2017). *Modeling of atmospheric chemistry* (Vol. 1). Cambridge University Press.
- Brenowitz, N. D., & Bretherton, C. S. (2018). Prognostic validation of a neural network unified physics parameterization. *Geophysical Research Letters*, *45*, 6289–6298. <https://doi.org/10.1029/2018GL078510>
- Champion, K., Lusch, B., Kutz, J. N., & Brunton, S. L. (2019). Data-driven discovery of coordinates and governing equations. *Proceedings of the National Academy of Sciences*, *116*(45), 22,445–22,451. <https://doi.org/10.1073/pnas.1906995116>
- Chollet, F. (2017). *Keras*. <https://github.com/fchollet/keras>
- Erichson, N. B., Muehlebach, M., & Mahoney, M. W. (2019). Physics-informed autoencoders for Lyapunov-stable fluid flow prediction. *arXiv*. arXiv:1905.10866.
- Fann, N., Fulcher, C. M., & Hubbell, B. J. (2009). The influence of location, source, and emission type in estimates of the human health benefits of reducing a ton of air pollution. *Air Quality, Atmosphere and Health*, *2*(3), 169–176. <https://doi.org/10.1007/s11869-009-0044-0>
- Head, T., Louppe, G., Shcherbatyi, I., & Vinicius, Z. (2018). scikit-optimize/scikit-optimize: v0.5.2.
- Heo, J., Adams, P. J., & Gao, H. O. (2016). Reduced-form modeling of public health impacts of inorganic PM_{2.5} and precursor emissions. *Atmospheric Environment*, *137*, 80–89. <https://doi.org/10.1016/j.atmosenv.2016.04.026>
- Hu, L., Keller, C. A., Long, M. S., Sherwen, T., Auer, B., Da Silva, A., et al. (2018). Global simulation of tropospheric chemistry at 12.5 km resolution: Performance and evaluation of the GEOS-Chem chemical module (v10-1) within the NASA GEOS Earth system model (GEOS-5 ESM). *Geoscientific Model Development*, *11*(11), 4603–4620. <https://doi.org/10.5194/gmd-11-4603-2018>
- Industrial Economics (2019). Evaluating reduced-form tools for estimating air quality benefits, (technical report).
- Karpatne, A., Watkins, W., Read, J., & Kumar, V. (2017). Physics-guided neural networks (PGNN): An application in lake temperature modeling. *arXiv*. arXiv:1710.11431.
- Keller, C. A., & Evans, M. J. (2019). Application of random forest regression to the calculation of gas-phase chemistry within the GEOS-Chem chemistry model v10. *Geoscientific Model Development*, *12*(3), 1209–1225. <https://doi.org/10.5194/gmd-12-1209-2019>
- Kelp, M. M., Tessum, C. W., & Marshall, J. D. (2018). Orders-of-magnitude speedup in atmospheric chemistry modeling through neural network-based emulation. *arXiv*. arXiv:1808.03874.
- Kim, B., Azevedo, V. C., Thuerey, N., Kim, T., Gross, M., & Solenthaler, B. (2018). Deep fluids: A generative network for parameterized fluid simulations. *arXiv*. arXiv:1806.02071.
- Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv*. arXiv:1412.6980.
- Lee, K., & Carlberg, K. (2018). Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders. *Journal of Computational Physics*, *360*, 108973. <https://doi.org/10.1016/j.jcp.2019.108973>
- Li, M., Soltanolkotabi, M., & Oymak, S. (2019). Gradient descent with early stopping is provably robust to label noise for overparameterized neural networks. *arXiv*. arXiv:1903.11680.
- Lowe, R., & Tomlin, A. (2000). Low-dimensional manifolds and reduced chemical models for tropospheric chemistry simulations. *Atmospheric Environment*, *34*(15), 2425–2436. [https://doi.org/10.1016/S1352-2310\(99\)00447-1](https://doi.org/10.1016/S1352-2310(99)00447-1)
- Lu, D., & Ricciuto, D. (2019). Efficient surrogate modeling methods for large-scale Earth system models based on machine-learning techniques. *Geoscientific Model Development*, *12*(5), 1791–1807. <https://doi.org/10.5194/gmd-12-1791-2019>
- Lusch, B., Kutz, J. N., & Brunton, S. L. (2018). Deep learning for universal linear embeddings of nonlinear dynamics. *Nature Communications*, *9*(1), 4950. <https://doi.org/10.1038/s41467-018-07210-0>
- McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, *5*(4), 115–133. <https://doi.org/10.1007/BF02478259>
- McGibbon, J., & Bretherton, C. S. (2019). Single-column emulation of reanalysis of the northeast Pacific marine boundary layer. *Geophysical Research Letters*, *46*, 10,053–10,060. <https://doi.org/10.1029/2019GL083646>

- Muller, N. Z. (2014). Boosting GDP growth by accounting for the environment. *Science*, *345*(6199), 873–874. <https://doi.org/10.1126/science.1253506>
- National Research Council (2012). *A national strategy for advancing climate modeling*. National Academies Press.
- National Research Council (2016). *The future of atmospheric chemistry research: Remembering yesterday, understanding today, anticipating tomorrow*. National Academies Press.
- Pan, S., & Duraisamy, K. (2020). Physics-informed probabilistic learning of linear embeddings of non-linear dynamics with guaranteed stability. *SIAM Journal on Applied Dynamical Systems*, *19*(1), 480–509. <https://doi.org/10.1137/19m1267246>
- Raissi, M., Perdikaris, P., & Karniadakis, G. E. (2017). Physics informed deep learning (Part II): Data-driven discovery of nonlinear partial differential equations. *arXiv*. arXiv:1711.10566.
- Rasp, S., Pritchard, M. S., & Gentile, P. (2018). Deep learning to represent subgrid processes in climate models. *Proceedings of the National Academy of Sciences*, *115*(39), 9684–9689. <https://doi.org/10.1073/pnas.1810286115>
- Regazzoni, F., Dedè, L., & Quarteroni, A. (2019). Machine learning for fast and reliable solution of time-dependent differential equations. *Journal of Computational Physics*, *397*, 108852. <https://doi.org/10.1016/j.jcp.2019.07.050>
- Reichstein, M., Camps-Valls, G., Stevens, B., Jung, M., Denzler, J., & Carvalhais, N. (2019). Deep learning and process understanding for data-driven Earth system science. *Nature*, *566*(7743), 195–204. <https://doi.org/10.1038/s41586-019-0912-1>
- Santillana, M., Le Sager, P., Jacob, D. J., & Brenner, M. P. (2010). An adaptive reduction algorithm for efficient chemical calculations in global atmospheric chemistry models. *Atmospheric Environment*, *44*(35), 4426–4431. <https://doi.org/10.1016/j.atmosenv.2010.07.044>
- Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks*, *61*, 85–117. <https://doi.org/10.1016/j.neunet.2014.09.003>
- Stewart, R., & Ermon, S. (2016). Label-free supervision of neural networks with physics and domain knowledge. *AAAI*, 2576–2582.
- Tessum, C. W., Hill, J. D., & Marshall, J. D. (2017). InMAP: A model for air pollution interventions. *PLoS ONE*, *12*(4), e0176131. <https://doi.org/10.1371/journal.pone.0176131>
- Theis, T. N., & Wong, H.-S. P. (2017). The end of Moore's law: A new beginning for information technology. *Computing in Science & Engineering*, *19*(2), 41–50. <https://doi.org/10.1109/mcse.2017.29>
- U. S. EPA (2006). *Air quality criteria for ozone and related photochemical oxidants (final report, 2006)*. Washington, DC: U.S. Environmental Protection Agency.
- U. S. EPA (2009). *Integrated Science Assessment (ISA) for particulate matter (final report, Dec 2009)*. Washington DC: U.S. Environmental Protection Agency.
- Wang, J.-X., Wu, J., Ling, J., Iaccarino, G., & Xiao, H. (2017). A comprehensive physics-informed machine learning framework for predictive turbulence modeling. *arXiv*. arXiv:1701.07102.
- Whitehouse, L. E., Tomlin, A. S., & Pilling, M. J. (2004). Systematic reduction of complex tropospheric chemical mechanisms, Part I: Sensitivity and time-scale analyses. *Atmospheric Chemistry and Physics*, *4*(7), 2025–2056. <https://doi.org/10.5194/acp-4-2025-2004>
- Xie, Y., Franz, E., Chu, M., & Thuerey, N. (2018). tempoGAN: A temporally coherent, volumetric GAN for super-resolution fluid flow. *ACM Transactions on Graphics*, *37*. <https://doi.org/10.1145/3197517.3201304>
- You, K., Long, M., Wang, J., & Jordan, M. I. (2019). How does learning rate decay help modern neural networks? *arXiv* arXiv:1908.01878.
- Zaveri, R. A., Easter, R. C., Fast, J. D., & Peters, L. K. (2008). Model for Simulating Aerosol Interactions and Chemistry (MOSAIC). *Journal of Geophysical Research*, *113*, D13204. <https://doi.org/10.1029/2007JD008782>
- Zaveri, R. A., & Peters, L. K. (1999). A new lumped structure photochemical mechanism for large-scale applications. *Journal of Geophysical Research*, *104*(D23), 30,387–30,415. <https://doi.org/10.1029/1999JD900876>

References From the Supporting Information

- Boucher, O., Moulin, C., Belviso, S., Aumont, O., Bopp, L., Cosme, E., et al. (2003). DMS atmospheric concentrations and sulphate aerosol indirect radiative forcing: A sensitivity study to the DMS source representation and oxidation. *Atmospheric Chemistry and Physics*, *3*(1), 49–65. <https://doi.org/10.5194/acp-3-49-2003>
- Bravo, H., Sosa, R., Sánchez, P., Bueno, E., & González, L. (2002). Concentrations of benzene and toluene in the atmosphere of the southwestern area at the Mexico City Metropolitan Zone. *Atmospheric Environment*, *36*(23), 3843–3849. [https://doi.org/10.1016/S1352-2310\(02\)00292-3](https://doi.org/10.1016/S1352-2310(02)00292-3)
- Corrêa, S. M., & Arbilla, G. (2005). Formaldehyde and acetaldehyde associated with the use of natural gas as a fuel for light vehicles. *Atmospheric Environment*, *39*(25), 4513–4518. <https://doi.org/10.1016/j.atmosenv.2005.03.042>
- Davis, S., Talbot, R., Mao, H., & Neuman, J. (2014). Meteorological influences on trace gas transport along the North Atlantic coast during ICARTT 2004. *Atmosphere*, *5*(4), 973. <https://doi.org/10.3390/atmos5040973>
- Delhomme, O., Morville, S., & Millet, M. (2010). Seasonal and diurnal variations of atmospheric concentrations of phenols and nitrophenols measured in the Strasbourg area, France. *Atmospheric Pollution Research*, *1*(1), 16–22. <https://doi.org/10.5094/APR.2010.003>
- González Abad, G., Allen, N. D. C., Bernath, P. F., Boone, C. D., McLeod, S. D., Manney, G. L., et al. (2011). Ethane, ethyne and carbon monoxide concentrations in the upper troposphere and lower stratosphere from ACE and GEOS-Chem: A comparison study. *Atmospheric Chemistry and Physics*, *11*(18), 9927–9941. <https://doi.org/10.5194/acp-11-9927-2011>
- Grosjean, E., & Grosjean, D. (1998). Formation of ozone in urban air by photochemical oxidation of hydrocarbons: Captive air experiments in Porto Alegre, RS. *Journal of the Brazilian Chemical Society*, *9*, 131–143. <https://doi.org/10.1590/S0103-50531998000200004>
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the 29th IEEE Conference on Computer Vision and Pattern Recognition* (pp. 770–778).
- Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
- Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv*. arXiv:1502.03167.
- Kramer, M. A. (1991). Nonlinear principal component analysis using autoassociative neural networks. *AIChE Journal*, *37*(2), 233–243. <https://doi.org/10.1002/aic.690370209>
- McLaren, R., Wojtal, P., Majonis, D., McCourt, J., Halla, J. D., & Brook, J. (2010). NO₃ radical measurements in a polluted marine environment: Links to ozone formation. *Atmospheric Chemistry and Physics*, *10*(9), 4187–4206. <https://doi.org/10.5194/acp-10-4187-2010>
- Mikkonen, S., Romakkaniemi, S., Smith, J. N., Korhonen, H., Petäjä, T., Plass-Duelmer, C., et al. (2011). A statistical proxy for sulphuric acid concentration. *Atmospheric Chemistry and Physics*, *11*(21), 11,319–11,334. <https://doi.org/10.5194/acp-11-11319-2011>

- Müller, J.-F., Liu, Z., Nguyen, V. S., Stavrakou, T., Harvey, J. N., & Peeters, J. (2016). The reaction of methyl peroxy and hydroxyl radicals as a major source of atmospheric methanol. *Nature Communications*, 7, 13213. <https://doi.org/10.1038/ncomms13213>
- Petersson, G. (1988). High ambient concentrations of monoterpenes in a Scandinavian pine forest. *Atmospheric Environment (1967)*, 22(11), 2617–2619. [https://doi.org/10.1016/0004-6981\(88\)90497-0](https://doi.org/10.1016/0004-6981(88)90497-0)
- Santurkar, S., Tsipras, D., Ilyas, A., & Madry, A. (2018). How does batch normalization help optimization? *arXiv arXiv:1805.11604*.
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1), 1929–1958.
- Wang, S., Nan, J., Shi, C., Fu, Q., Gao, S., Wang, D., et al. (2015). Atmospheric ammonia and its impacts on regional air quality over the megacity of Shanghai, China. *Scientific Reports*, 5, 15842. <https://doi.org/10.1038/srep15842>
- Webb, A. J., Bösch, H., Parker, R. J., Gatti, L. V., Gloor, E., Paul, I., et al. (2016). CH₄ concentrations over the Amazon from GOSAT consistent with in situ vertical profile data. *Journal of Geophysical Research: Atmospheres*, 121, 11,006–11,020. <https://doi.org/10.1002/2016JD025263>
- Wennberg, P. O., Hanisco, T. F., Jaeglé, L., Jacob, D. J., Hints, E. J., & Lanzendorf, E. J. (1998). Hydrogen radicals, nitrogen radicals, and the production of O₃ in the upper troposphere. *Science*, 279(5347), 49–53. <https://doi.org/10.1126/science.279.5347.49>