Approximation and Optimization of Global Environmental Simulations with Neural Networks

Elnaz Azmi
Steinbuch Centre for Computing
Karlsruhe Institute of Technology
Karlsruhe, Germany
elnaz.azmi@kit.edu

Jörg Meyer
Steinbuch Centre for Computing
Karlsruhe Institute of Technology
Karlsruhe, Germany
joerg.meyer2@kit.edu

Marcus Strobl
Steinbuch Centre for Computing
Karlsruhe Institute of Technology
Karlsruhe, Germany
marcus.strobl@kit.edu

Michael Weimer
Earth, Atmospheric and Planetary Sciences Massachusetts Institute of Technology
Cambridge, MA, United States
mweimer@mit.edu

Achim Streit
Steinbuch Centre for Computing
Karlsruhe Institute of Technology
Karlsruhe, Germany
achim.streit@kit.edu

ABSTRACT

Solving a system of hundreds of chemical differential equations in environmental simulations has a major computational complexity, and thereby requires high performance computing resources, which is a challenge as the spatio-temporal resolution increases. Machine learning methods and specially deep learning can offer an approximation of simulations with some factor of speed-up while using less compute resources. In this work, we introduce a neural network based approach (ICONET) to forecast trace gas concentrations without executing the traditional compute-intensive atmospheric simulations. ICONET is equipped with a multifeature Long Short Term Memory (LSTM) model to forecast atmospheric chemicals iteratively in time. We generated the training and test dataset, our target dataset for ICONET, by execution of an atmospheric chemistry simulation in ICON-ART. Applying the ICONET trained model to forecast a test dataset results in a good fit of the forecast values to our target dataset. We discussed appropriate metrics to evaluate the quality of models and presented the quality of the ICONET forecasts with RMSE and KGE metrics. The variety in the nature of trace gases limits the model's learning and forecast skills according to the respective trace gas. In addition to the quality of the ICONET forecasts, we described the computational efficiency of ICONET as its run time speed-up in comparison to the run time of the ICON-ART simulation. The ICONET forecast showed a speed-up factor of 3.1 over the run time of the atmospheric chemistry simulation of ICON-ART, which is a significant achievement, especially when considering the importance of ensemble simulation.

PASC '23, June 26–28, 2023, Davos, Switzerland

ACM ISBN 979-8-4007-0190-0/23/06.
https://doi.org/10.1145/3592979.3593418

CCS CONCEPTS

• **Computing methodologies** → **Neural networks**; *Modeling and simulation*; • **Applied computing** → *Environmental sciences*; • **Mathematics of computing** → Time series analysis.

KEYWORDS

Neural networks, LSTM, Environmental simulations, Time series

ACM Reference Format:
Elnaz Azmi, Jörg Meyer, Marcus Strobl, Michael Weimer, and Achim Streit. 2023. Approximation and Optimization of Global Environmental Simulations with Neural Networks. In *Platform for Advanced Scientific Computing Conference (PASC '23), June 26–28, 2023, Davos, Switzerland*. ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3592979.3593418

1 INTRODUCTION

Computational optimization approaches are used in many research fields to achieve the optimal solution corresponding to various criteria or constraints, e.g., reducing costs and computation time, or increasing profits [3, 14]. The ultimate objectives of optimization are to minimize undesirable effects, to provide more useful solutions and an enhanced efficiency and reliability, and to reduce costs [21]. Computer-based modeling and simulation are widely used techniques in scientific research to analyze and understand real-world systems, as well as to design and develop performant products [34]. However, the development and execution of large-scale and complex systems' simulations are time- and energy-consuming. Under the perspective of energy saving, and despite the availability of modern and powerful computing technologies, there is a need "to address issues such as the complexity and scale of the systems that need to be modeled today" [11]. We addressed in this work the exploitation of optimization approaches to compute a plausible approximation of large-scale numerical simulations that is computationally less expensive than the original, and resulting in a simulation output that is acceptable for domain scientists.

Numerical environmental simulations, especially in high spatio-temporal resolution, consisting of large-scale dynamical systems are compute-intensive and require high performance computing (HPC) resources. Our approach to reduce the computational complexity (after [5]) and the high demand of computing resources is

to approximate simulations through supervised machine learning methods focusing on neural networks.

In the field of atmospheric sciences, accurate forecasts of the atmosphere demand large-scale simulations. Atmospheric chemistry modeling, for example, usually requires solving a system of hundreds of coupled ordinary partial differential equations that describe the concentration changes of atmospheric trace gases due to chemical reactions in the atmosphere [8, 27]. In general, the growth of HPC resources over the last decades makes it possible to increase the resolutions (i.e., number of grid cells) of atmospheric models, thus resolving more and more processes directly rather than accounting only for their effect with so-called parametrizations [33]. On the other hand, solving the system of chemical differential equations for each grid cell is a major fraction of the computation time in atmospheric chemistry models, which is a challenge as resolution increases. The goal of this study is to investigate the feasibility, opportunities but also challenges and pitfalls of replacing the compute-intensive chemistry of a state-of-the-art atmospheric chemistry model with a trained neural network model to forecast the concentration of trace gases at each grid cell and to reduce the computational complexity of the simulation. In this work, we introduce a neural network based approach that trains a model using the simulation dataset of twelve trace gas concentrations and three physical input data from past simulations to forecast the twelve trace gas concentrations in the future iteratively. This work is a proof of concept, and a deeper study has to be done to apply it on any operational system.

The remainder of this paper is structured as follows: Sect. 2 provides further information about the study background, Sect. 3 is a survey of related work, the proposed approach is explained in Sect. 4. In Sect. 5, the results are presented and evaluated, Sect. 6 is about the implementation environment and the conclusions are drawn in Sect. 7.

2 BACKGROUND

2.1 The ICON-ART Model

The study case of this work is the ICOsahedral Nonhydrostatic modeling framework with its extension for Aerosols and Reactive Trace gases (ICON-ART) [29, 33, 36]. It has been jointly developed by several German institutions [35] and is a unified model for all time and spatial scales that are relevant for the atmosphere. Since 2015, it has been used for operational weather forecasting at the German Weather Service (DWD) [36]. The chemistry and photolysis rates in ICON-ART are calculated using the box model CAABA/MECCA and CLoudJ for each grid cell [24, 27, 29], i.e., the differential equations for the chemical reactions in ICON-ART are solved separately in each grid cell.

In this study, we use ICON-ART version 2.1 with the 90 vertical model levels from near ground (level 90) up to 75 km (level 1) of the operational setup at DWD and a horizontal resolution of about 160 km, resulting in about 1.8 million grid cells [36]. In order to replace only the atmospheric chemistry solvers of the ICON-ART model by a neural network model, it is needed to switch off the transport of chemical tracers, since this is a process that is not part of the chemistry solver and should not be captured by ICONET either. This is only true for the transport of chemical trace gases, but all variables relevant for the meteorology are transported in the model. Therefore, we ran the simulation without transportation of the trace gases to preserve a closed system in the model without neighborhood interaction for each grid cell. The time step of the simulation is set to six minutes and the output is given at each time step.

We applied the approach to a chemical mechanism for ozone in the stratosphere from vertical model level 45 to 30 (16 levels) [26], using a subset of 23 reactions for oxygen- and nitrogen-related species [29]. From the 14 gases of these reactions, N_2 and O_2 are no trace gases and available in excess, hence constant and not included in this work. Therefore, each output data file contains the volume mixing ratio (VMR) of twelve trace gases and three physical features of all grid cells for one day in six minute resolution. The twelve trace gases are N_2O, N_2O_5, HO_2, H_2O, NO, NO_3, HNO_3, $O(^3P)$, NO_2, OH, O_3, and $O(^1D)$. The physical features are temperature, pressure, and the cosine of the solar zenith angle (cos SZA). The simulation covers the years 2013 and 2014.

2.2 Neural Networks

Neural networks and deep learning are useful programming paradigms with a potential of achieving excellent performance and promising results in many areas such as image and language processing, speech recognition and forecasting [12, 19, 31]. Time series are one of the main input data in forecasting domain, and are defined as sequences of data points indexed in time order. Time series forecasting is the method of predicting future values given the domain knowledge and previously observed values [18]. As our study case data are time series, we used LSTM, that is one of the suitable and widely used neural network models for time series data [1, 20, 32]. LSTM is a special kind of Recurrent Neural Networks (RNNs) that is extended to learn long-term dependencies [16]. RNNs are networks with loops, using the information learned from previous inputs to generate outputs. LSTM benefits from using additional gates and cell state to address the problem of long-term dependencies [15]. LSTM consists of memory cells which contain gates using the sigmoid and hyperbolic tangent activation functions. These functions change the cell's state and decide which information to retain for future forecasts. In LSTM, the model passes the last hidden state (short-term memory) and cell state (long-term memory) to the next step of the sequence, thus holds the information of previously seen data and uses it to forecast the future data [16]. In this work, we use the LSTM from the RNN module in PyTorch (version: 1.10.0+cu113) [22] wrapped by PyTorch Lightning (version: 1.5.2) [10].

3 RELATED WORK

Environmental simulations and forecasting models are usually compute-intensive, in particular when considering ensemble simulations in high spatio-temporal resolution. With the growing capability of HPCs containing GPU resources, there is a potential of benefiting from forecasting methods based on machine learning, particularly neural network models [7, 20, 25, 28, 32]. Once such a model is trained on an HPC system, it can be used several times for forecasting, and consequently the overall process is faster and computationally less expensive than the physical models. Additionally,

there is a potential of forecasting at time scales and in locations where physical models act weakly. However, this is the case when the network is trained with the observation data.

[28] used a deep convolutional neural network to emulate physical models like general circulation model (GCM) used in weather prediction and climate science. The network learns from the dynamics of GCM and forecasts the model weather for several time steps (up to 14 days) ahead. The Root Mean Squared Error (RMSE) of neural network forecast over the true state of GCM is decreased compared to both the persistence and climatological forecasts. The study shows that the neural network learns the time evolution and dynamics of a simple GCM principally, but the studies need to be continued for more complex models, including external forcing.

A fully connected multi-layer neural network model for the atmosphere to forecast global weather is developed by [7]. They showed that the model can make better forecasts than a simple persistence model and the forecasts are competitive with forecasts of coarse-resolution ($6^\circ \approx 668$ km) atmosphere models of similar complexity, at least for short lead times. However, the forecasts are not stable and deteriorate after a few days. According to the study, a close collaboration is required between computer scientists and meteorologists to include the physical knowledge and deep understanding of the earth system into the neural network model.

For medium-range weather forecasting, defined [25] a data-driven method that uses a deep residual convolutional neural network (Resnet). They trained models to forecast geopotential, temperature, and precipitation at $5.625^\circ \approx 626$ km resolution up to five days ahead. Compared to physical models, Resnet achieves comparable scores to a physical model at comparable resolution. They used 150 years dataset of the Coupled Model Intercomparison Project (CMIP [9]) to pretrain the model and fine-tune it using the ERA data. Hence, the current CMIP models run at around 100 km resolution, it can not be used for forecasts at higher resolutions. The goal of this work is exploring the feasibility of data-driven approaches in weather forecast.

In order to overcome the high computational costs while attaining comparable quality of their results, [2] presented a fully connected neural network. They used a dataset generated from the global numerical atmosphere chemistry model (EMAC) to make predictions of chemical tendencies. This work showed a proof of concept that neural networks are able to predict atmospheric chemistry tendencies. However, hyperparameter tuning is required to optimize the model and to overcome modeling problems due to seasonal trends in the data. This is a challenge of using a neural network model in comparison to the use of physical models.

A review and discussion of the opportunities given through deep learning approaches in the field of weather prediction is given in [30]. They focused on the possibility to replace numerical forecast models, and presented models that are limited to short-term forecasting of less than 24 h. In conclusion, they do see potential in using deep learning for weather forecasts, but emphasize that a lot of research is still necessary until deep learning methods can replace traditional numerical models.

4 ICONET ARCHITECTURE

We developed the ICON Neural Network based approach (ICONET), an approach containing a multifeature LSTM model that forecasts atmospheric trace gases. The core process of ICONET is learning a function that maps a sequence of ten past time steps to the next (eleventh) time step. The use case of this work is the ICON-ART chemistry simulation at a spatial resolution of about 160 km and a temporal resolution of six minutes. However, since the chemistry solver of ICON-ART works on a grid-cell basis, and we want to replace it by ICONET, ICONET is applicable to every horizontal and vertical grid cell of ICON-ART, independent of its spatio-temporal resolution. In our experiments, we apply ICONET to a subset of arbitrary grid cells in the stratosphere. Fig. 1 shows an overview of the ICONET architecture containing four main steps, namely preprocessing, training, forecasting and postprocessing. The following describes each step of the ICONET architecture in detail.

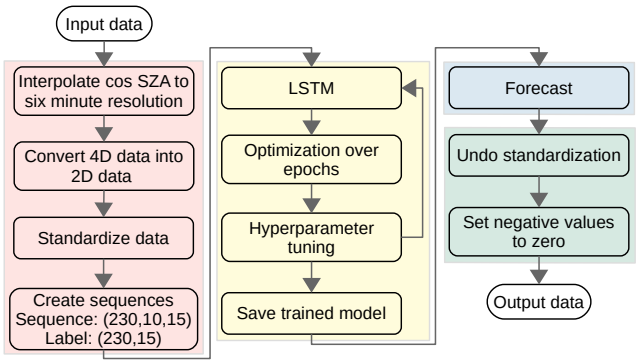


Figure 1: ICONET architecture showing the main steps preprocessing (red), training (yellow), forecasting (blue), and postprocessing (green).

4.1 Input Data and Preprocessing

We generated the input data for training, validating, and testing of ICONET from the ICON-ART simulation output. Every output data file contains the volume mixing ratio (VMR) of twelve trace gases and three physical features of all grid cells for one day in six minute resolution. To improve the manageability of data loading during the training of ICONET, we split the output data files so that the grid cell locations are shuffled, and the grid cells are randomly distributed into 256 files of ~ 94 MiB (Fig. 2). This simplified sampling of the input data. We used two of these sub files for training and one for validation. Due to the importance of time dependency, we keep the temporal dimension unchanged.

All features are available in the same temporal resolution, except for cos SZA. In order to improve the accuracy of ICONET, we interpolate and smooth this feature to the six minute resolution as in the ICON-ART output. Fig. 3 shows the value range and distribution of all features from an example grid cell located above the Indian Ocean on 12 September 2013. In the following text, we refer to this grid cell as an exemplary grid cell. The simulation output is four-dimensional data (vertical model level, grid cell location, time step, and feature), which is not suitable as LSTM input data. As the

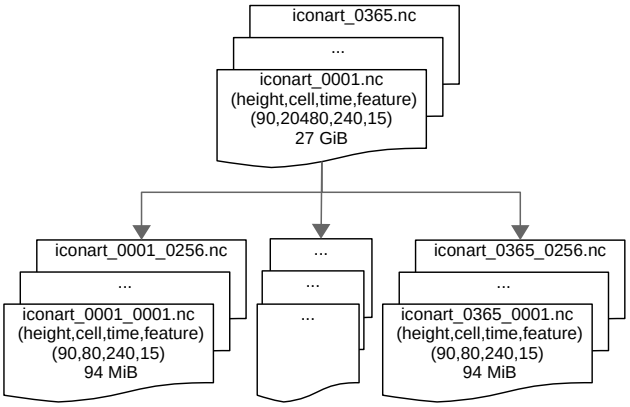


Figure 2: Splitting schema for simulation output data files.

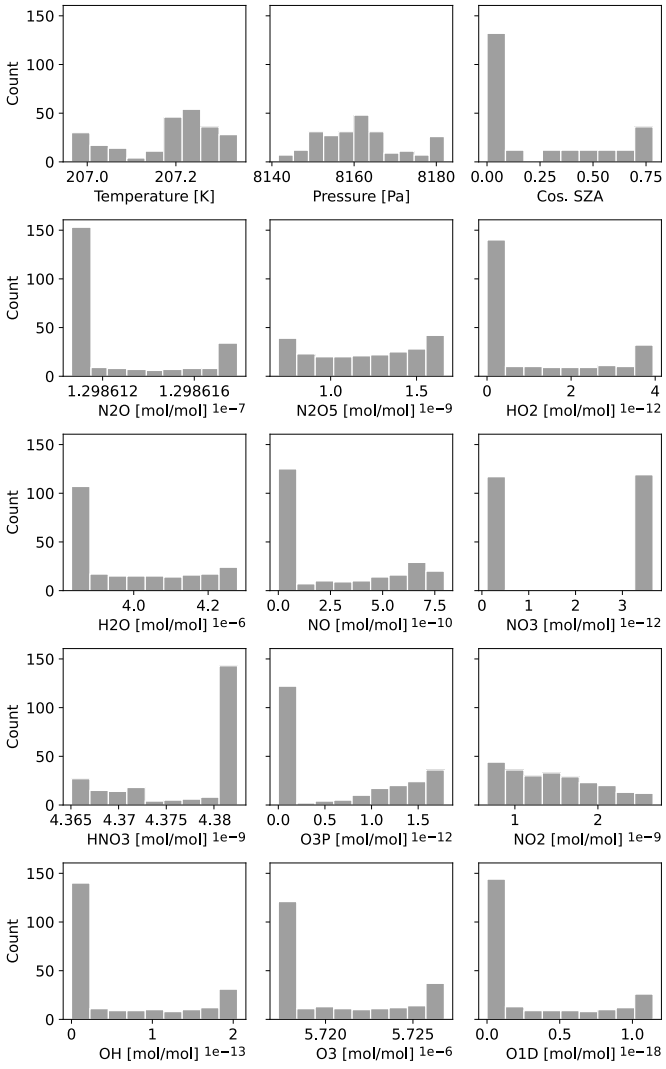


Figure 3: Input data distribution of a grid cell located above the Indian Ocean from 12 September 2013.

vertical levels are defined based on the pressure values, we removed this dimension from the dataset due to its high correlation with the pressure feature. Since we only consider replacing the atmospheric chemistry, there is no need to know the grid cell positions apart from the pressure being an input feature. Thus, we removed the dimension related to the position of the grid cell in the atmospheric column. Principally, the grid cells have no interaction with each other, hence the grid cell coordinates are not informative in this case. Thus, we also removed this dimension from the dataset. Finally, we have two-dimensional data (time step and feature) for each grid cell.

An equally scaled input data is important for the learning performance in machine learning algorithms [6]. Thus, we standardize these multifeature data because the features values vary in several magnitudes, see Fig. 3. Standardization is a scaling method where the values are centered around the mean with a unit standard deviation as follows:

$$X' = \frac{X - \mu}{\sigma}, \tag{1}$$

where X' is the standardized feature set, μ is the mean and σ is the standard deviation of the features X. Our last preprocessing step is rolling, which is the process of shifting a fixed-size window over time series to create smaller time series and extract features for each shortened sequence. To label our data, we utilized a rolling window of ten time steps on all one-day data sets and applied the eleventh time step as the respective label. This results in 230 input sequences of length ten time steps each.

4.2 Training and Validation

4.2.1 LSTM Layers. ICONET consists of a one layer LSTM model with one input layer, 15 hidden states representing 15 chemical and physical input features and one output layer. While training, ICONET reads an input sequence into the LSTM model and gets the final hidden state of the last time step in the input sequence as an output or forecast value. This output is compared with the labels (target) in the loss function, and the process is iterated in some epochs to learn a plausible model.

4.2.2 Mass Conservation. One of the main goals during the development of ICON was an improved mass conservation compared to other meteorological models [35]. Mass conservation should be given for every closed system in chemistry, that is in our case study a single grid cell without transportation of the trace gases. Any replacement model or modification in the simulation has to conserve quantities that are necessary in the model accuracy. In our study case, VMR of Nitrogen (N) in the forecast values is constant and conserved during the simulation time, see the target line at Fig. 4. VMR of Hydrogen (H) and Oxygen (O) is not conserved because we do not have a closed system for H_2O which influences the VMR of H and O. We calculate the VMR of N using Eq. 2.

$$C_N = \sum_{g=1}^{12} n_g \cdot C_g, \tag{2}$$

where C_N is the VMR of N in mol/mol, n_g is the number of N atoms in each molecule of a trace gas and C_g is the VMR of trace gas g in mol/mol. In order to assess the quality of mass conservation in ICONET forecast, we calculated the VMR of N in the forecast values generated by a trained model with a variable λ in the loss

function (Eq. 3).

$$Loss(Y, \hat{Y}) = \lambda \cdot \text{MSE}(Y, \hat{Y}) + (1 - \lambda) \cdot \text{MSE}(C_N, \hat{C_N}), \quad (3)$$

with:

$$\text{MSE}(Y, \hat{Y}) = \frac{1}{n} \sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2, \quad (4)$$

where Y is target, $\hat{Y}$ is forecast, λ is a hyperparameter for penalizing the deviation from mass conservation and MSE is the mean squared error. The loss function consists of two parts, the loss of forecast and mass conservation. We varied the λ values in the range of 0.5 to one and evaluated the test results from two aspects. First, how stable the VMR of N is, second, how the quality of the forecast is. Fig. 4 shows the conservation of N in forecast using the loss

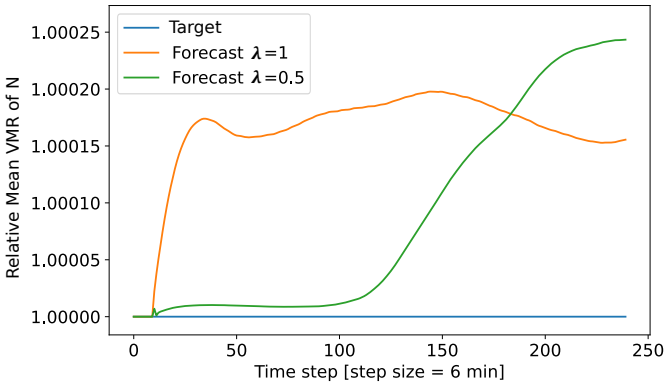


Figure 4: Mean VMR of N for all grid cells relative to the target values of the middle stratospheric vertical model level on 12. September 2014.

function with $\lambda = 0.5$ is close to the target for the first 100 time steps in comparison to the forecast using the loss function with $\lambda = 1$. However, the forecast of trace gases using the loss function with $\lambda = 0.5$ has a greater RMSE than the loss function with $\lambda = 1$. Processing our test results using different λ values showed that the VMR of N is not constant during the test day, when we use other λ values than one. Therefore, we concluded to set λ to one and, consequently, removed the second part of the loss function.

4.2.3 Training Steps and Optimization. Following after the preprocessing steps, the input data are ready for training of ICONET. In order to load the input sequences into the training step, we use Pytorch Lightning Dataloader with four workers (subprocesses) per GPU and the batch size of one. While the Dataloader loads the next sequences and labels into the training step, the multifeature LSTM of ICONET generates an output that is the $\hat{Y}$ in the loss function (Eq. 3). The loss is calculated between the target (Y) and the model output ($\hat{Y}$) with the Mean Squared Error (MSE) metric, see Eq. 4. The loss value is minimized in several iterations of adjusting the weights and biases of the model (training epochs).

Network structure and training configuration of neural network models are defined as hyperparameters, that have to be set before training. We did a greedy search over a range of values to find a reasonable combination of the hyperparameters for a better forecast. In the hyperparameters search, for example, we set the number of LSTM layers between one and four. More layers perform better in reproducing the trend of the curves than a single layer, but on the downside they produce a high oscillation around the target, while the metrics show no significant improvement. Additionally, the run time of the training and forecasting steps are higher than when using only one layer. The final hyperparameters are as following:

- Learning rate: 3×10^{-3}
- Batch size: 230 (number of shortened sequences in one day)
- Sequence length: ten time steps (in total one hour)
- Number of training epochs: 3500
- Number of LSTM layers: one
- Number of LSTM hidden states: 15
- Number of input features in loss function: 15 (all features)
- λ in the loss function: one
- Input data scaling: standardization
- Interpolation and smoothing of cos SZA: yes
- Number of grid cells: 126 for training, 54 for validation
- Simulation days: one in 15 days interval in one year (in total 24 days)

We trained and validated ICONET on a subset of totally 180 randomly located grid cells. We used 70% of this subset for training and 30% for validation. The temporal subset for training and validation contains totally 24 days distributed in one year that is one in 15 days interval. We trained the model with different number of iterations and used early stopping to avoid overfitting. Early stopping is a regularization strategy that determines when to stop training so that a model generalizes well to larger or unseen datasets [23]. We saved the trained model on epoch 3500 where the validation epoch loss starts to converge. Fig. 5 shows the epoch loss (MSE) during the model training on the training and validation dataset. In the zoomed figure of Fig. 5, we show that the distance between the training and the validation fit curves starts to increase from about epoch 3500, which shows that the model is not learning anymore.

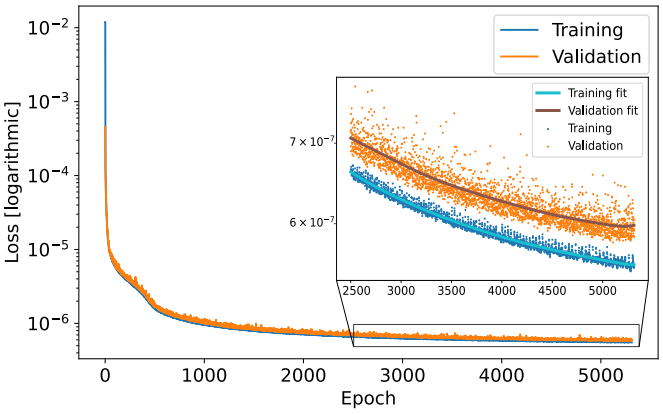


Figure 5: Training and validation loss per epoch.

4.3 Forecast

We used our trained model to forecast all twelve trace gases for all grid cells of one vertical model level in the middle stratosphere on 12. September 2014, hence one year after the training dataset

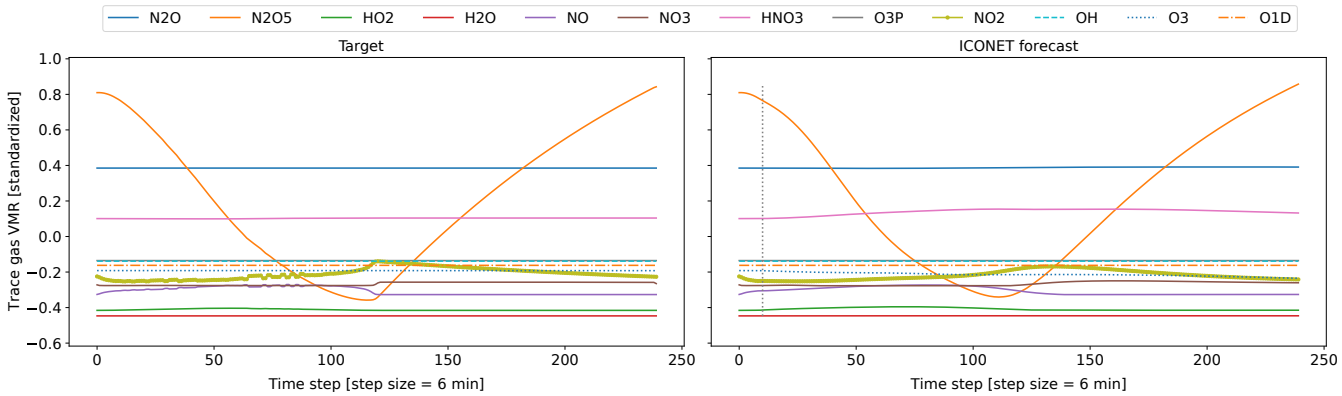


Figure 6: ICON-ART simulation output (target) on the left side and ICONET forecast on the right side. It shows the VMR of all trace gases in the exemplary grid cell on 12 September 2014. Both Y-axes are in the same min-max limit.

(test dataset). We do not forecast the three physical features, hence they are the available input features of the atmospheric chemistry simulation. The preprocessing of the test dataset followed the same procedures as for the training dataset. In the forecast step, we input all features of ten time steps to ICONET and forecast only the trace gases for the next time step. Afterward, we use the forecast values as input, thereby the next time step is predicted. In this work, we use the term "forecast" for this iterative forecast.

4.4 Postprocessing

In the postprocessing step, we transform the standardized output of the neural network back to the physical units by undoing the standardization (Eq. 1) done in the preprocessing. Forecasts with very small values compared to the distribution of the training data might get transformed back to negative values. As VMR are positive numbers, we set those negative values to zero.

5 RESULTS AND DISCUSSION

In this section, we show the results of the ICONET trained on the output data of an atmospheric simulation of ICON-ART. For evaluation of our results, we need an appropriate metric, which we discuss in the following.

5.1 Metrics

Selection of an appropriate quality metric for forecasting models is challenging, because the quality of a model can be considered from different aspects. Here, we discuss the pros and cons of some relevant metrics. One of the basic evaluation metrics is the absolute difference between the forecast and target values. This metric shows directly, how much the forecast values are above or below the target and in the same unit as the calculated values. However, using the absolute difference we could not compare the deviations of different variables with each other. A very simple relative error metric is measuring the absolute difference between the forecast and target values divided by the target value. We could not use this metric, as we have target values that are exactly zero, which causes a divided by zero error.

ICONET is a regression model which outputs continuous variables. Therefore, we consider a metric for gauging regression models. MSE (Eq. 4) is a regression metric that measures the mean squared difference between the forecast and target values, but it is very sensitive to outliers. MSE weights large errors more heavily than the small ones. Another common metric is Root Mean Squared Error (RMSE) defined as the square root of MSE:

$$\mathrm{RMSE}(\mathrm{Y}, \hat{\mathrm{Y}}) = \sqrt{\frac{1}{\mathrm{n}} \sum_{\mathrm{i}=1}^{\mathrm{n}} (\mathrm{Y_i} - \hat{\mathrm{Y}}_\mathrm{i})^2}. \tag{5}$$

It gives less weight to larger deviations compared to MSE. RMSE is commonly preferred to use, due to its interpretability, as it returns the error in the same units as the target value. RMSE values can range from zero to positive infinity, where values closer to zero indicate a better estimation.

While RMSE is doing well in quantifying the distance of two curves, it does not measure how well the forecast follows the trend of the target curves. Kling-Gupta Efficiency (KGE) [13] see Eq. 6 is a measure of the goodness-of-fit, common in hydrological modeling. KGE values can range from negative infinity to one, and the values closer to one indicate better fit.

$$\mathrm{KGE}(\mathrm{Y}, \hat{\mathrm{Y}}) = 1 - \sqrt{(\mathrm{r}-1)^2 + (\alpha-1)^2 + (\beta-1)^2}, \tag{6}$$

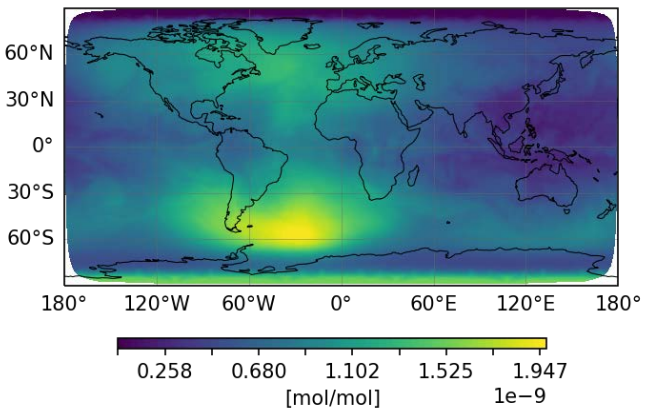
with:

$$\mathrm{r} = \frac{\mathrm{Cov}_{\hat{\mathrm{Y}}\mathrm{Y}}}{\sigma_{\hat{\mathrm{Y}}} \cdot \sigma_{\mathrm{Y}}}, \qquad \alpha = \frac{\sigma_{\hat{\mathrm{Y}}}}{\sigma_{\mathrm{Y}}}, \qquad \beta = \frac{\mu_{\hat{\mathrm{Y}}}}{\mu_{\mathrm{Y}}},$$

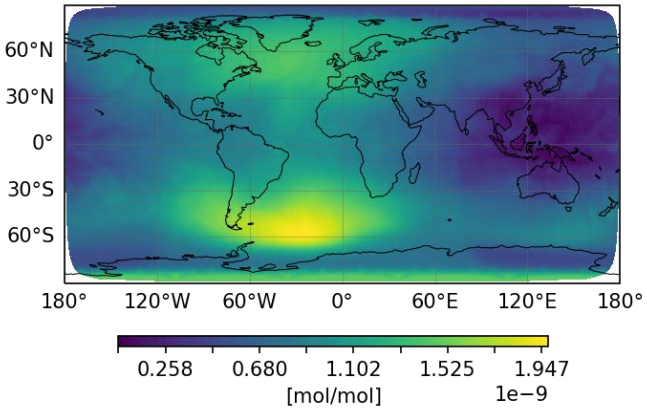
where $\mathrm{Cov}_{\hat{\mathrm{Y}}\mathrm{Y}}$ is the covariance between the forecast and target values, σ is the mean and μ is the standard deviation. In other words, r is the linear cross-correlation coefficient between the forecast and target values, α is a measure of variability in the data values, and β is equal to the mean of the forecast values over the mean of the target values.

Considering the sensitivity to outliers and other disadvantages of the discussed metrics, we used two metrics, alone or in combination, to evaluate the quality of our model, i.e., how close the forecast values are to the target values. The first metric is RMSE and the second one is KGE. We show the RMSE values in both original and standardized scale. The RMSE of the values in original scale are better interpretable for domain scientists. The RMSE of the

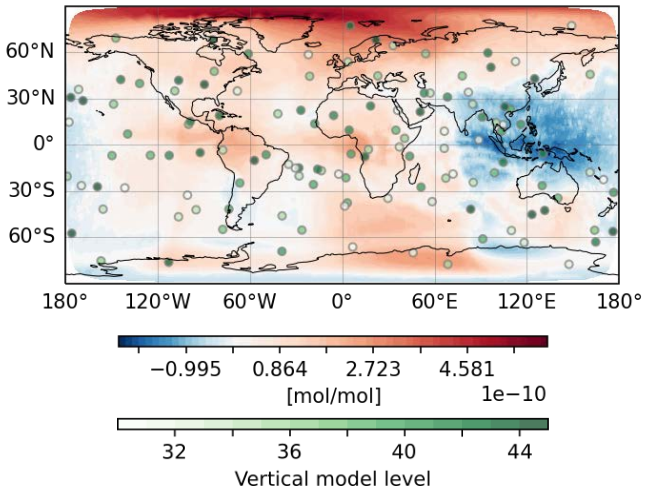
standardized values enables us to compare different features in the same order of magnitude.



(a) Target



(b) ICONET forecast



(c) Difference

Figure 7: N_2O_5 VMR for all grid cells in an arbitrary time step. Points on (c) show the location, and their colors show the vertical model levels (45 to 30) of the trained grid cells.

In addition to the quality metrics, the computational efficiency (performance) of ICONET is presented as its run time speed-up in comparison to the run time of the atmospheric chemistry simulation of ICON-ART using the same computing resources.

5.2 Results Evaluation

After training of our model on a small subset of the stratospheric grid cells, we can forecast all twelve trace gases for all grid cells of any stratospheric vertical model level on any date of a year. As a test case, we forecast all twelve trace gases for all grid cells of one vertical model level in the middle stratosphere on 12 September 2014. In the following, we show the results of this test and its evaluation.

Fig. 6 shows the standardized VMR of all trace gases in the exemplary grid cell during the test day. It shows the ICON-ART simulation output (target) on the left side and ICONET forecast on the right side. All trace gases in the forecast show a plausible fit to the target values. The shapes of the forecast curves are smoothed in comparison to the target. The forecast values show a close fit with the target values.

In order to visualize the deviation of the test results from the target in all grid cells, we demonstrate N_2O_5 values as an example in Fig. 7. The values are VMR of N_2O_5 at each grid cell in an arbitrary time step of the test day. The difference map (Fig. 7c) shows a lot of grid cells where the difference is very close to zero (smaller than $+/- 1e^{-11}$) in white and the highest difference in red at polar regions. The points on Fig. 7c show the location, and their color shows different vertical model levels of the trained grid cells. As the map shows, we trained a subset of randomly distributed grid cells from different stratospheric vertical model levels.

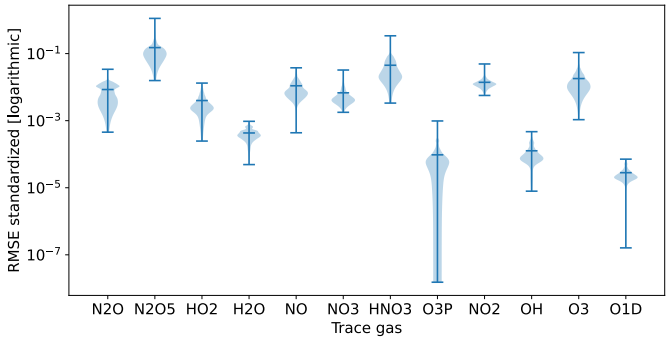


Figure 8: RMSE distribution for all grid cells of the test case

We show and interpret the RMSE values from two perspectives. First, the RMSE distribution of all grid cells for each trace gas during the test day, and second, the RMSE of each grid cell (map view) for each trace gas during the test day. From the first perspective, we compared the quality of ICONET forecast between different trace gases, as shown in Fig. 8. This is a RMSE distribution of all grid cells of the test case. As the RMSE values are calculated between target and forecast values in standardized scale, we could compare the trace gases in this diagram. N_2O_5 shows the highest mean RMSE value (0.15) and $O(^1D)$ has the lowest value ($2.8e^{-5}$) among the others. The RMSE values of $O(^3P)$ are widely distributed, which

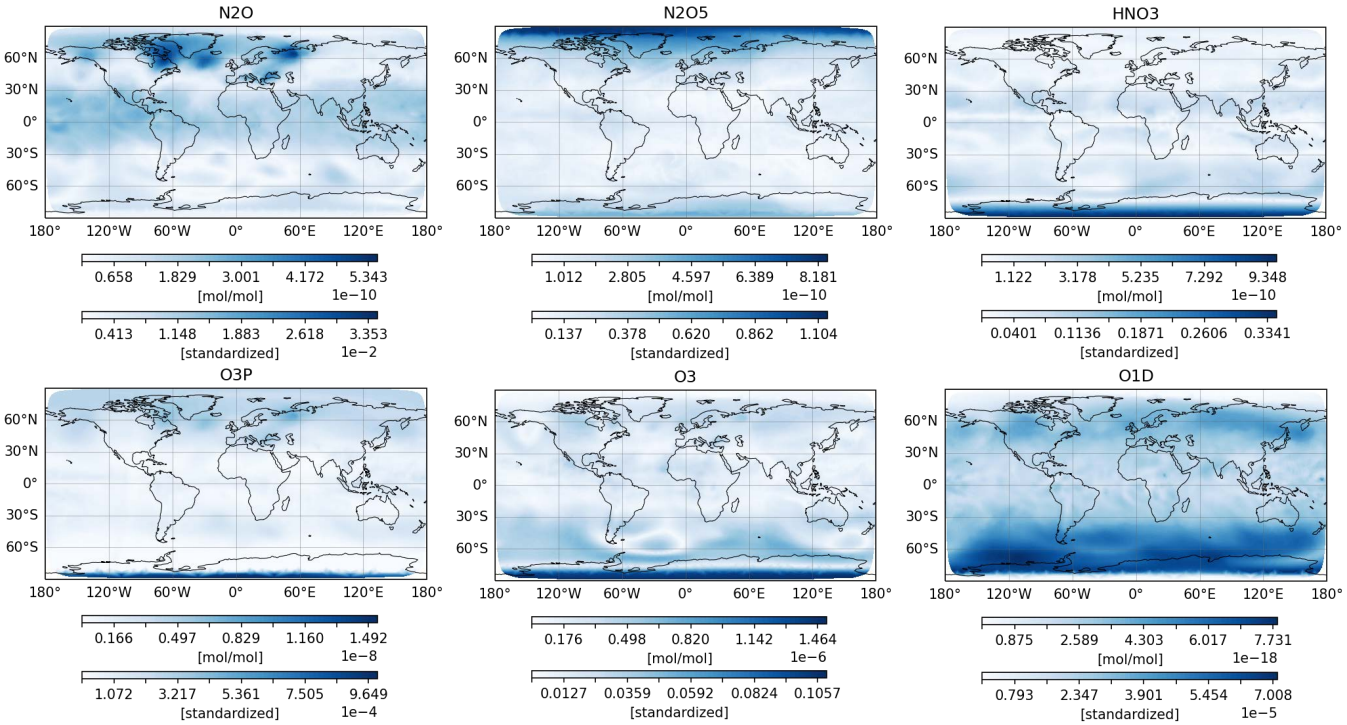


Figure 9: RMSE between target and ICONET forecast for VMR of some exemplary trace gases of the test case.

shows the instability of the model forecast spatio-temporally (Fig. 8). From the second perspective, we compared the quality of ICONET forecast over different locations of the world map. The maps in Fig. 9 visualize the RMSE between the target and ICONET forecast for VMR of some exemplary trace gases of the test case. The maps show mostly higher RMSE in the polar regions. N_2O_5, HNO_3, O_3 and $O(^3P)$ show a similar spatial distribution of the RMSE values. We conclude here, that the model behaves differently in different spatial regions.

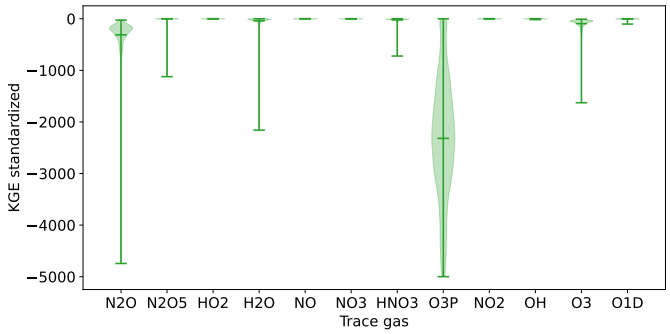
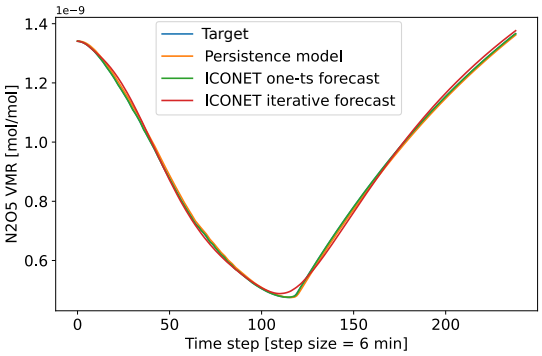


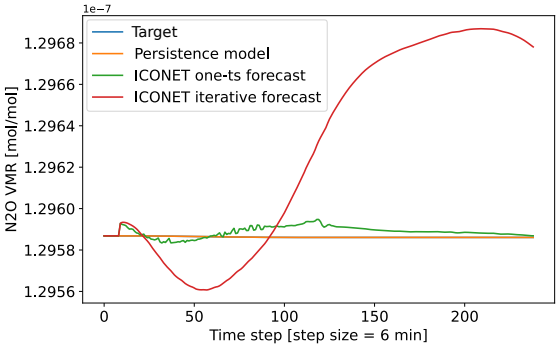
Figure 10: Distribution of KGE for all grid cells of the test case. Some outliers smaller than −5000 are removed from the diagram for better visualization of the values close to one.

Another metric that we used in our quality evaluation is KGE. Fig. 10 shows a similar diagram as Fig. 8 but for KGE values. We interpret the KGE results from different aspects. We compared the KGE and RMSE results and their relation with each other. Both metrics show a plausible forecast for OH and $O(^1D)$ (low RMSE < 0.15 and high KGE close to one), but RMSE values for $O(^1D)$ contain some outliers, which shows the model does not learn well on some grid cells. Some trace gases (HO_2, NO, NO_3, NO_2 and O_3) with high KGE values also have similar mean RMSE values and show the model's ability to learn well and forecast plausibly for these trace gases. Even though the results show overall low RMSE and plausible fit for most of the trace gases, there are two exceptions. N_2O with plausible RMSE values shows in contrast low KGE values containing lots of outliers, which is a sign of a not well-learned model for this trace gas. Additionally, $O(^3P)$ though a very low RMSE ($9.6e^{-5}$), shows very low and highly scattered KGE values, concluding the model's inability in learning the trend of this variable.

During the evaluation of ICONET, we also compare the quality of our model with a very simple persistence model as a reference. The persistence model forecasts the future value of a time series under the assumption that nothing changes between the current time and the forecast time [17]. This means that the values at time step $t + 1$ (forecast) are equal to the values at the current time step t. As the persistence model forecasts account for the next time step only, we generated a comparable ICONET forecast for one time step only and without iteration. We named this forecast as ICONET one-ts forecast. This forecast ensures a fair comparison with the persistence model. Fig. 11 illustrates the target values together with the forecast values of the ICONET and the persistence model. The



(a) N_2O_5 **VMR**



(b) N_2O **VMR**

Figure 11: VMR of two exemplary trace gases from ICONET forecast and persistence model vs. target.

values are VMR of a trace gas from the exemplary grid cell on the test day. For trace gas N_2O_5, Fig. 11a shows that both ICONET forecasts are as good as or even better than the persistence model. Fig. 11b shows that ICONET one-ts forecast has a very close results to the persistence model for trace gas N_2O, but ICONET iterative forecast does not fit well to the other models.

In the evaluation of ICONET, in addition to the model quality, we quantify the model performance with speed-up of ICONET over the ICON-ART simulation run time. We ran the atmospheric chemistry simulation of ICON-ART on one compute node without any parallelization, for forecasting of the same test case as for ICONET. The test of forecasting all grid cells of one vertical model level during one day, resulted in the simulation run time of ~ 106 s, where the ICONET run time was ~ 34 s. In comparison, ICONET forecast showed 3.1x speed-up over the simulation run time.

For evaluation of the stability of ICONET over one week (1670 time steps) we ran the ICONET forecast for seven continuous days in September 2014 and calculated the RMSE of all grid cells from the test case for each day separately. The mean of these RMSE values is shown in Fig. 12. The presented RMSE values are in terms of multiples of standard deviations and not percent. The errors lower than one standard deviation show that the method works in general. The graph shows that the RMSE values of most trace gases are roughly constant after the third day. This means that ICONET forecast remains stable for 66% of tested trace gases. The RMSE values of N_2O, N_2O_5, NO_2 and O_3 trace gases increase linearly until the third day, then the growth of values slows down toward stability. Fig. 13 is an example of ICONET forecast during one week. It is the forecast of the exemplary grid cell from the test case in September 2014. The diagram shows a plausible fit and stable forecast for most trace gases.

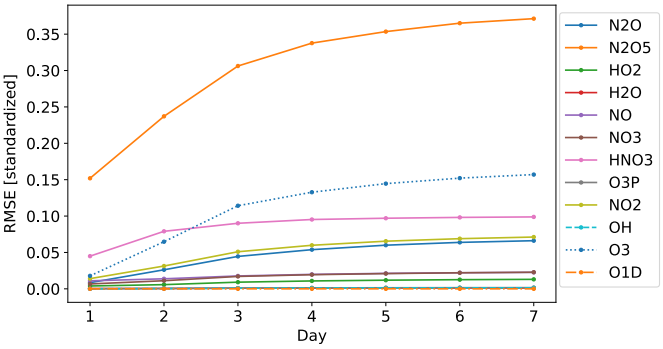


Figure 12: Evolution of mean RMSE value of all grid cells from the test case for all trace gases during one week.

6 IMPLEMENTATION ENVIRONMENT

We implemented ICONET in Python, mainly in PyTorch and PyTorch Lighting. The training and forecast are executed on supercomputer, see Acknowledgments. For ICONET training, we used accelerator nodes containing two Intel Xeon Platinum 8368 processors, 512 GB of main memory, four NVIDIA A100-40 GPUs of 40 GB memory with Red Hat Enterprise Linux (RHEL) 8.x. operating system. All scripts, data files and requirements of the model are available in a GitHub repository named "iconet" [4].

7 CONCLUSIONS AND OUTLOOK

High resolution environmental simulations are compute-intensive and require a lot of HPC resources. In order to approximate such simulations and reduce the computational complexity, and consequently the resource demand, we developed a neural network based approach (ICONET). This multifeature LSTM model was developed on a study case of forecasting atmospheric chemistry. The model is applicable to simulation output dataset in any spatio-temporal resolution. However, the dataset has to be prepared before loading into the training step. ICONET consists of several steps namely, preprocessing, training, forecast and postprocessing. The ICONET one-ts forecast applied on a test case shows for some trace gases an improvement over the persistence model. The ICONET iterative forecast results in a plausible fit with the target, but naturally it is not comparable with the persistence model. In addition to the quality of the ICONET forecast, we describe the computational efficiency of ICONET as its run time speed-up in comparison to the run time of the ICON-ART simulation. ICONET forecast showed a 3.1x speed-up over the run time of the ICON-ART atmospheric chemistry simulation. This work is a proof of concept, and it has not been tested yet in an operational system. The forecasts of the use case shows low RMSE values and partially a plausible fit with

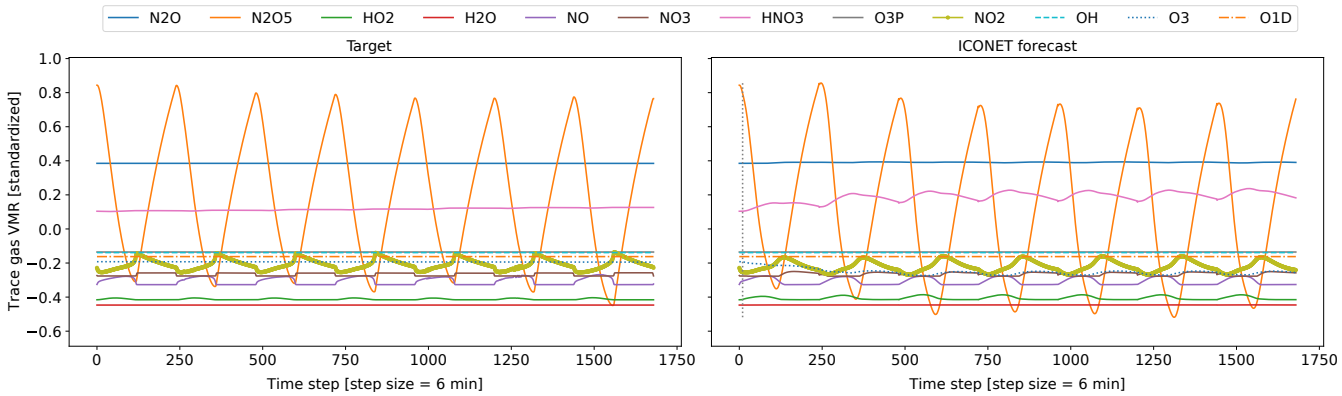


Figure 13: Simulation output (left) and ICONET forecast (right) of VMR of trace gases in the exemplary grid cell in one week.

target. Considering the need of ensemble simulations, even a small speed-up over the original simulation is a significant achievement.

In future works, we should consider that extreme modifications in the input dataset and the study case require redoing the training steps and tuning of the hyperparameters. Additionally, there is a potential of programmatically optimizing the preprocessing and training of ICONET, to be able to train a larger subset of grid cells and improve the accuracy of the forecast. This work can be a base work and study for comparison with future studies like using transformers or physics-informed neural networks on the same use case. The relative mass conservation error seems reasonable on the presented test case, but for longtime simulations, a deeper study should be done.

ACKNOWLEDGMENTS

This work was performed on the HoreKa supercomputer funded by the Ministry of Science, Research and the Arts Baden-Württemberg and by the Federal Ministry of Education and Research. It is also supported partially by the Helmholtz Association Initiative and Networking Fund on the HAICORE@KIT partition. We thank Markus Götz and Uğur Çayoğlu for constructive discussions about neural networks, the Helmholtz AI consultants for Earth and environment from German climate computing center (DKRZ) for discussions about AI in climate simulations, Roland Ruhnke that made the ICON-ART model and its related data files available, and the reviewers for their comments to improve the paper.

REFERENCES

[1] S. Agarwal, N. Tosi, P. Kessel, D. Breuer, and G. Montavon. 2021. Deep learning for surrogate modeling of two-dimensional mantle convection. *Phys. Rev. Fluids* 6 (2021), 113801. Issue 11. https://doi.org/10.1103/PhysRevFluids.6.113801

[2] F. Albrecht, F. Stiehler, B.-M. Sinnhuber, S. Versick, and T. Weigel. 2021. *AI for Fast Atmospheric Chemistry.* EGU General Assembly. https://doi.org/10.5194/egusphere-egu21-4570

[3] G. Alonso, E. Del Valle, and J. R. Ramirez. 2020. *Desalination in Nuclear Power Plants.* Woodhead Publishing, Duxford, United Kingdom. https://doi.org/10.1016/C2019-0-01164-8

[4] E. Azmi. 2023. An ICON Neural Network based approach. https://github.com/elnazazmi/iconet

[5] E. Azmi, U. Ehret, S. V. Weijs, B. L. Ruddell, and R. AP. Perdigão. 2021. Technical note: "Bit by bit": a practical and general approach for evaluating model computational complexity vs. model performance. *Hydrol Earth Syst Sci* 25, 2 (2021), 1103–1115. https://doi.org/10.5194/hess-25-1103-2021

[6] C. M. Bishop. 1995. *Neural networks for pattern recognition.* Oxford university press, New York City, New York.

[7] P. D. Dueben and P. Bauer. 2018. Challenges and design choices for global weather and climate models based on machine learning. *Geosci Model Dev* 11, 10 (2018), 3999–4009. https://doi.org/10.5194/gmd-11-3999-2018

[8] L. K. Emmons, S. Walters, P. G. Hess, J. F. Lamarque, G. G. Pfister, D. Fillmore, et al. 2010. Description and evaluation of the Model for Ozone and Related chemical Tracers, version 4 (MOZART-4). *Geosci. Model Dev.* 3, 1 (2010), 43–67. https://doi.org/10.5194/GMD-3-43-2010

[9] V. Eyring, S. Bony, G. A. Meehl, C. A. Senior, B. Stevens, R. J. Stouffer, and K. E. Taylor. 2016. Overview of the Coupled Model Intercomparison Project Phase 6 (CMIP6) experimental design and organization. *Geoscientific Model Development* 9, 5 (2016), 1937–1958. https://doi.org/10.5194/gmd-9-1937-2016

[10] W. Falcon and The PyTorch Lightning team. 2019. PyTorch Lightning. https://doi.org/10.5281/zenodo.3828935

[11] R. Fujimoto, C. Bock, W. Chen, E. Page, and J. H. Panchal. 2017. *Research Challenges in Modeling and Simulation for Engineering Complex Systems.* Springer, Cham, Switzerland. https://doi.org/10.1007/978-3-319-58544-4

[12] I. Goodfellow, Y. Bengio, and A. Courville. 2016. *Deep learning.* MIT press, Cambridge, Massachusetts. http://www.deeplearningbook.org.

[13] H. V. Gupta, H. Kling, K. K. Yilmaz, and G. F. Martinez. 2009. Decomposition of the mean squared error and NSE performance criteria: Implications for improving hydrological modelling. *J. Hydrol.* 377, 1-2 (2009), 80–91. https://doi.org/10.1016/j.jhydrol.2009.08.003

[14] A. Herzel, S. Ruzika, and C. Thielen. 2021. Approximation Methods for Multiobjective Optimization Problems: A Survey. *INFORMS J Comput* 33, 4 (2021), 1284–1299. https://doi.org/10.1287/ijoc.2020.1028

[15] S. Hochreiter. 1991. Untersuchungen zu dynamischen neuronalen Netzen. *Diploma, Technische Universität München* 91, 1 (1991).

[16] S. Hochreiter and J. Schmidhuber. 1997. Long short-term memory. *Neural comput* 9, 8 (1997), 1735–1780.

[17] Y.-Y. Hong and R. Pula. 2020. Comparative Studies of Different Methods for Short-term Locational Marginal Price Forecasting. In *2020 5th International Conference on Green Technology and Sustainable Development (GTSD).* IEEE, Ho Chi Minh City, Vietnam, 527–532. https://doi.org/10.1109/GTSD50082.2020.9303121

[18] R. J. Hyndman and G. Athanasopoulos. 2018. *Forecasting: principles and practice.* OTexts, Melbourne, Australia.

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton. 2017. Imagenet classification with deep convolutional neural networks. *Commun. ACM* 60, 6 (2017), 84–90. https://doi.org/10.1145/3065386

[20] P. Lara-Benitez, M. Carranza-Garcia, and J. C. Riquelme. 2021. An Experimental Review on Deep Learning Architectures for Time Series Forecasting. *Int J Neural Syst* 31, 3 (2021), 2130001. https://doi.org/10.1142/S0129065721300011

[21] C. Merrill, R. L. Custer, J. Daugherty, M. Westrick, and Y. Zeng. 2008. Delivering Core Engineering Concepts to Secondary Level Students. *J. Technol. Educ.* 20, 1 (2008), 48–64. https://doi.org/10.21061/jte.v20i1.a.4

[22] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, et al. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Adv Neural Inf Process Syst*, Vol. 32. Curran Associates, Inc., Red Hook, New York, 8024–8035. http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[23] J. Patterson and A. Gibson. 2017. *Deep learning: A practitioner's approach.* O'Reilly Media, Inc., Sebastopol, California.

[24] M. J. Prather. 2015. Photolysis rates in correlated overlapping cloud fields: Cloud-J 7.3c. *Geosci. Model Dev.* 8, 8 (2015), 2587–2595. https://doi.org/10.5194/gmd-8-2587-2015

[25] S. Rasp and N. Thuerey. 2021. Data-Driven Medium-Range Weather Prediction With a Resnet Pretrained on Climate Simulations: A New Model for WeatherBench. *J. Adv. Model. Earth Syst.* 13, 2 (2021), e2020MS002405. https://doi.org/10.1029/2020MS002405

[26] D. Reinert, F. Prill, H. Frank, M. Denhard, M. Baldauf, C. Schraff, et al. 2021. DWD Database Reference for the global and Regional ICON and ICON-EPS Forecasting System. https://doi.org/10.5676/DWD_pub/nwv/icon_2.1.7

[27] R. Sander, A. Baumgaertner, S. Gromov, H. Harder, P. Jöckel, A. Kerkweg, et al. 2011. The atmospheric chemistry box model CAABA/MECCA-3.0. *Geosci. Model Dev.* 4, 2 (2011), 373–380. https://doi.org/10.5194/gmd-4-373-2011

[28] S. Scher. 2018. Toward Data-Driven Weather and Climate Forecasting: Approximating a Simple General Circulation Model With Deep Learning. *Geophys. Res. Lett.* 45, 22 (2018), 12–616. https://doi.org/10.1029/2018GL080704

[29] J. Schröter, D. Rieger, C. Stassen, H. Vogel, M. Weimer, S. Werchner, et al. 2018. ICON-ART 2.1: a flexible tracer framework and its application for composition studies in numerical weather forecasting and climate simulations. *Geosci. Model Dev.* 11, 10 (2018), 4043–4068. https://doi.org/10.5194/gmd-11-4043-2018

[30] M. G. Schultz, C. Betancourt, B. Gong, F. Kleinert, M. Langguth, L. H. Leufen, et al. 2021. Can deep learning beat numerical weather prediction? *Philos. Trans. Royal Soc. A* 379, 2194 (2021), 20200097. https://doi.org/10.1098/rsta.2020.0097

[31] C. K. Sønderby, L. Espeholt, J. Heek, M. Dehghani, A. Oliver, T. Salimans, et al. 2020. Metnet: A neural weather model for precipitation forecasting. https://doi.org/10.48550/arXiv.2003.12140 arXiv:2003.12140

[32] G. Van Houdt, C. Mosquera, and G. Napoles. 2020. A review on the long short-term memory model. *Artif Intell Rev* 53, 8 (2020), 5929–5955. https://doi.org/10.1007/s10462-020-09838-1

[33] M. Weimer, J. Buchmüller, L. Hoffmann, O. Kirner, B. Luo, R. Ruhnke, et al. 2021. Mountain-wave-induced polar stratospheric clouds and their representation in the global chemistry model ICON-ART. *Atmos. Chem. Phys.* 21, 12 (2021), 9515–9543. https://doi.org/10.5194/acp-21-9515-2021

[34] C. Yin and A. McKay. 2018. Introduction to Modeling and Simulation Techniques. In *Proceedings of ISCIIA and ITCA*. University of Leeds, White Rose Research Online, Tengzhou, China, 6 pages. https://eprints.whiterose.ac.uk/135646.

[35] G. Zängl, D. Reinert, and F. Prill. 2022. Grid Refinement in ICON v2.6.4. *Geosci. Model Dev.* 15 (2022), 7153–7176. https://doi.org/10.5194/gmd-15-7153-2022

[36] G. Zängl, D. Reinert, P. Rípodas, and M. Baldauf. 2015. The ICON (ICOsahedral Non-hydrostatic) modelling framework of DWD and MPI-M: Description of the non-hydrostatic dynamical core. *Q. J. R. Meteorol. Soc.* 141, 687 (2015), 563–579. https://doi.org/10.1002/qj.2378