

---

# Emulating Aerosol Microphysics with Machine Learning

---

Paula Harder<sup>1 2 3</sup> Duncan Watson-Parris<sup>4</sup> Dominik Strassel<sup>1</sup> Nicolas Gauger<sup>2</sup> Philip Stier<sup>4</sup> Janis Keuper<sup>1 5</sup>

## Abstract

Aerosol particles play an important role in the climate system by absorbing and scattering radiation and influencing cloud properties. They are also one of the biggest sources of uncertainty for climate modeling. Many climate models do not include aerosols in sufficient detail. In order to achieve higher accuracy, aerosol microphysical properties and processes have to be accounted for. This is done in the ECHAM-HAM global climate aerosol model using the M7 microphysics model, but increased computational costs make it very expensive to run at higher resolutions or for a longer time. We aim to use machine learning to approximate the microphysics model at sufficient accuracy and reduce the computational cost by being fast at inference time. The original M7 model is used to generate data of input-output pairs to train a neural network on it. By using a special logarithmic transform we are able to learn the variables tendencies achieving an average  $R^2$  score of 89%. On a GPU we achieve a speed-up of 120 compared to the original model.

## 1. Introduction

Aerosol forcing remains the largest source of uncertainty in the anthropogenic effect on the current climate (Bellouin et al., 2020). The aerosol cooling effect hides some of the positive radiative forcing caused by greenhouse gas emissions and future restrictions to lower air pollution might result in stronger observed warming. Aerosols impact climate change through aerosol-radiation interactions and aerosol-

cloud interactions (IPCC, 2013). They can either scatter or absorb radiation, which depends on the particle's compounds. Black carbon aerosols from fossil fuel burning for example have a warming effect by absorbing radiation and whereas sulphate from volcanic eruptions has a cooling effect by scattering radiation. Clouds strongly influence the earth's radiation budget by reflecting sunlight and aerosols can change cloud properties significantly by acting as cloud condensation nuclei (CCN). A higher concentration of aerosols leads to more CCN which, for a fixed amount of water, results in more but smaller cloud droplets. Smaller droplets increase the cloud's albedo (Twomey, 1974) and can enhance the cloud's lifetime (Albrecht, 1989).

Many climate models often consider aerosols only as external parameters, they are read once by the model, but then kept constant throughout the whole model run. In the case where some aerosol properties are modeled, there might be no distinction between aerosol types, and just an overall mass is considered. To incorporate a more accurate description of aerosols, aerosol-climate modeling systems like ECHAM-HAM (Tegen et al., 2019) have been introduced. It couples the ECHAM General Circulation Model (GCM) with a complex aerosol model called HAM. The microphysical core of HAM is either SALSA (Kokkola et al., 2008) or the M7 model (Vignati et al., 2004). We consider the latter here. M7 uses seven log-normal modes to describe aerosol properties, the particle sizes are represented by four size modes, nucleation, aiten, accumulation, and coarse, of which the aiten, accumulation, and coarse can be either soluble or insoluble. It includes processes like nucleation, coagulation, condensation, and water uptake, which lead to the redistribution of particle numbers and mass among the different modes. In addition, M7 considers five different components - Sea salt (SS), sulfate (SO<sub>4</sub>), black carbon (BC), primary organic carbon (OC), and dust (DU). M7 is applied to each grid box independently, it does not model any spatial relations.

More detailed models come with the cost of increased computational time: ECHAM-HAM can be run at 150km resolution for multiple decades. But to run for example storm-resolving models the goal is ideally a 1km horizontal grid resolution and still be able to produce forecasts up to a few decades. If we want to keep detailed aerosols descriptions for this a massive speedup of the aerosol model is needed.

---

<sup>1</sup>Competence Center High Performance Computing, Fraunhofer Institute for Industrial Mathematics, Kaiserslautern, Germany <sup>2</sup>Scientific Computing, University of Kaiserslautern, Kaiserslautern, Germany <sup>3</sup>Fraunhofer Center Machine Learning, Germany <sup>4</sup>Dpt. of Atmospheric, Oceanic and Planetary Physics, University of Oxford, UK <sup>5</sup>Institute for Machine Learning and Analytics (IMLA), Offenburg University, Germany. Correspondence to: paula.harder@itwm.fraunhofer.de <

>.

Replacing climate model components with machine learning approaches and therefore decreasing a model’s computing time has been shown to be successful in the past. There are several works on emulating convection, both random forest approaches (O’Gorman & Dwyer, 2018) and deep learning models (Rasp et al., 2018), (Gentine et al., 2018), (Beucler et al., 2020) have been explored. Recently, multiple consecutive neural networks have been used to emulate a bin microphysical model for warm rain processes (Gettelman et al., 2021). Silva et al. (Silva et al., 2020) compare several methods, including deep neural networks, XGBoost, and ridge regression as physically regularized emulators for aerosol activation. In addition to the aforementioned approaches random forest approaches have been used to successfully derive the CCN from atmospheric measurements (Nair & Yu, 2020).

Our work shows a machine learning approach to approximate the M7 microphysics module. We investigated different approaches, neural networks as well as ensemble models like random forest and gradient boosting, with the aim of achieving the desired accuracy and computational efficiency, finding the neural network appeared to be the most successful. We use data generated from a realistic ECHAM-HAM model run and train a model offline to predict one time step of the aerosol microphysics. The underlying data distribution is challenging, as the changes in the variables are often zero or very close to zero. We do not predict the full values, but the tendencies, which requires a logarithmic transformation for both positive and negative values. After we considered different hyperparameters, a 2-layer, fully connected network to solve this multivariate regression problem is chosen. To train, validate and test our model in a representative way, we use about 11M data points from separate days spread out through the year. On the test set, an overall regression coefficient of 89% and a root mean squared error (RMSE) of 24.9% are achieved. For 75% of the variables, our model’s prediction has a  $R^2$  value of over 95% on the logarithmically transformed tendencies. Using a GPU we achieve a huge speed-up of 120 compared to the original model, on a single CPU we are still 1.6 faster.

## 2. Methodology

### 2.1. Data

#### 2.1.1. DATA GENERATION

To easily generate data we extract the aerosol microphysics model from the global climate model ECHAM-HAM and develop a stand-alone version. To obtain input data ECHAM-HAM is run for four days within a year. We use a horizontal resolution of 150km at the equator, over all 31 vertical levels, 96 latitudes, and 192 longitudes. A time step length of 450s is chosen. This yields a data set

of over 100M points for each day and we only use a subset of 5 times a day, resulting in about 2.85M points per day. We use the days in January and April for training, a day in July for validation, and the October data for testing the model. Because the two sea salt variables only change in 0.2% of the time we do not model them here. The masses and particle concentrations for different aerosol types and different modes are both inputs and outputs, the output is the value one time step later. Atmospheric state variables like pressure, temperature, and relative humidity are used as inputs only. A full list of input and output values can be found in the appendix in Table 5.

#### 2.1.2. DATA DISTRIBUTION AND TRANSFORMATION

Compared to the full values the tendencies are very small, so we aim to predict the tendencies, not the full values, where it is possible. Depending on the variable we have very different size scales, but also a tendency for a specific variable may span several orders of magnitude. In some modes, variables can only grow, in some only decrease, and in others do both. Often the majority of tendencies are either zero or very close to zero, but a few values might be very high. Usual normalizing would not solve the problem, that the training and evaluation would be dominated by a small number of huge values. As we want to give more weight to the prominent case, the smaller changes, a log transform is necessary, as for example done in (Gettelman et al., 2021). To have a continuous transformation, being able to deal with negative values, and only using one network and transformation we choose the following

$$y = \begin{cases} \log \sqrt{x} + 1 & x \geq 0 \\ -\log \sqrt{-x} + 1 & \text{else.} \end{cases}$$

The additional square root spreads out the values close to zero. The above transformation is applied to all variables. After the logarithmic transformation, we normalize and standardize all input and output values.

### 2.2. Model

For emulating the M7 microphysics model we explored different machine learning approaches, including random forest regression, gradient boosting, and a simple neural network. The neural network approach appears to be the most successful (See Figure 4) for this application and will be presented here.

#### 2.2.1. NETWORK ARCHITECTURE

We employ a fully connected network with a sigmoid activation function and two hidden layers, each hidden layer contains 256 nodes. Using zero hidden layers results in a linear regression and does not have the required expressivity for our task, one hidden layer already achieve a good

performance, after two layers the model we could not see any further improvements. The width of layers improves performance up to about 256 nodes and then stagnates as well. Even though the network's performance is not very sensitive to the choice of activation function, we achieve the best results using a sigmoid function.

### 2.2.2. TRAINING

We train the network using the Adam optimizer with a learning rate of  $10^{-3}$ , a weight decay of  $10^{-9}$ , and a batch size of 4096. Sensitivity analysis to different learning rates, weight decays, and batch size values can be found in the appendix. As a loss function, we choose an MSE loss and use early stopping with a patience of 10 epochs, where our training process stops after about 60 epochs. The training on a single NVIDIA Titan V GPU takes about 2 hours.

## 3. Results

### 3.1. Predictive performance

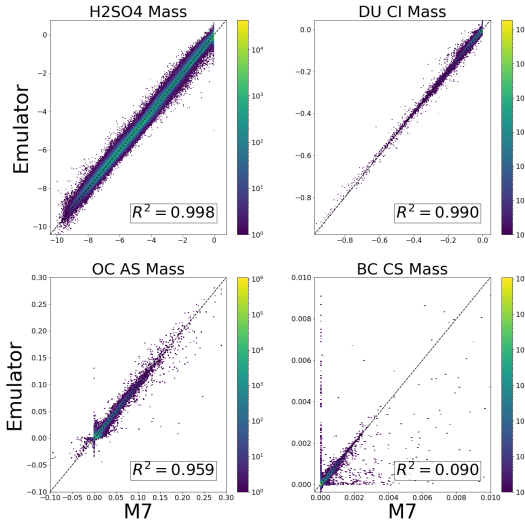


Figure 1. Predicted value by emulator versus true value from M7 model, tendencies, logarithmically transformed.

To evaluate the performance of our emulator we run the neural network on our independent test set. We calculate the regression coefficient, the  $R^2$  score, on the log-transformed tendencies. In order to be more comparable, the different scales for the different variables suggest a normalized RMSE, therefore we calculate this metric on the normalized values. Table 4 shows the average performance over all 28 predicted variables, the masses, and concentration after

Table 1. Regression performance of M7 emulator.

CASE	RMSE	$R^2$
TRAIN	0.230	0.933
VAL	0.254	0.924
TEST	0.249	0.892

Table 2. Regression performance of the M7 emulator for all variables.

VARIABLE	$R^2$	RMSE
H2SO4 MASS	0.9987	0.043
SO4 NS MASS	0.967	0.187
SO4 KS MASS	0.837	0.369
SO4 AS MASS	0.961	0.186
SO4 CS MASS	0.985	0.116
BC KS MASS	0.765	0.457
BC AS MASS	0.969	0.174
BC CS MASS	0.090	0.523
BC KI MASS	0.990	0.010
OC KS MASS	0.477	0.652
OC AS MASS	0.959	0.213
OC CS MASS	0.631	0.220
OC KI MASS	0.989	0.117
DU AS MASS	0.980	0.065
DU CS MASS	0.989	0.063
DU AI MASS	0.988	0.050
DU CI MASS	0.990	0.062
NS CONCENTRATION	0.977	0.154
KS CONCENTRATION	0.823	0.329
AS CONCENTRATION	0.699	0.512
CS CONCENTRATION	0.985	0.074
KI CONCENTRATION	0.993	0.088
AI CONCENTRATION	0.984	0.058
CI CONCENTRATION	0.987	0.067
NS WATER	0.995	0.068
KS WATER	0.997	0.041
AS WATER	0.991	0.096
CS WATER	0.993	0.082

one time step. The performance does not drop significantly compared to the training set, but there is a decrease in the  $R^2$  value from the validation set to the test case. Test scores for all variables separately can be found in Table 6. For 21 out of 28 variables  $R^2$  scores of over 95% are achieved. We found that two variables seem problematic to model, which is organic carbon in the aitenken mode and especially black carbon in the coarse mode. The bad performance for black carbon was not observed on the validation set, which could indicate some seasonal differences. A possible solution could be to train on a full year and evaluate on the next.

In Figure 1 we show the emulator tendency prediction plotted against the M7 module predictions for four variables. We show the best and worst-performing, as well as the variables closest to 25 and 75 percentiles, plots for all variables, can be found in the appendix, in Figure 2. In the plots

Table 3. Runtime comparison for the original M7 model and the M7 emulator.

MODEL	M7	EMULATOR GPU	EMULATOR CPU
TIME (S)	5.781	0.048	3.716
SPEED-UP	-	120.4	1.6

we can observe that often there are stronger deviations at zero, the model struggles with detecting whether a variable is changing at all or not. For black carbon coarse mode, we can see, that even though we applied a log transform, most of the values are very close to zero, whereas a few a many orders of magnitude bigger. This probably makes it hard for the model to learn the dependencies. The scores for back-transformed values show weaker performance for some variables, looking at the scores for the full variables though a perfect  $R^2$  is achieved apart from three variables, see Table 6.

Although the model performs well in an offline evaluation, it still remains to be shown how it performs when plugged back into the GCM and run for multiple time steps, good offline performance is no guarantee for good online performance (Rasp, 2020) in the case of convection parameterization, where a model crash could occur. In our case, it is likely that offline performance is a good indicator for on-line performance, as a model crash is not expected, because aerosols do not strongly affect large-scale dynamics.

### 3.2. Runtime

We conduct a preliminary runtime analysis by comparing the Python runtime for the emulator with the Fortran runtime of the original model. For the M7 model, the 31 vertical levels are calculated simultaneously and for the emulator, we predict one time step at once globally. We use a single NVIDIA Titan V GPU and single Intel Xeon 4108 CPU. As shown in Table 3 we can achieve a massive speed-up of 120 using a GPU compared to the original model, but using a single CPU we are only 1.6 times faster. Further speed-ups could be achieved by using multiple CPUs, a smaller network architecture, and an efficient implementation in Fortran.

## 4. Conclusion and future work

This work shows how neural networks can be used to learn the mapping of an aerosol microphysics model. Our model approximates the log-transformed tendencies well for nearly all variables and excellent for most of them. Using a GPU we achieve a significant speed-up.

How much of a speed-up can be achieved in the end remains

to be shown, when the model is used in a GCM run. For long-time climate modeling, it is important that our model does not show any mass biases and slowly loses or gains mass. To prevent that it has to be investigated if mass conservation is systematically violated or if it is symmetrically distributed around zero. To enforce mass conservation one possibility is to add a penalty term to the network’s loss function. Although it is not guaranteed, it could decrease the violation of mass conservation. In some cases, our model might also predict nonphysical values like negative masses and a mass fixer could be necessary. To employ the full potential of deep learning it could be possible to include spatial relationships in the model and not consider each grid box independently.

## Acknowledgements

DWP and PS acknowledge funding from NERC project NE/S005390/1 (ACRUISE) as well as from the European Union’s Horizon 2020 research and innovation programme iMIRACLI under Marie Skłodowska-Curie grant agreement No 860100. PS additionally acknowledges support from the ERC project RECAP and the FORCeS project under the European Union’s Horizon 2020 research programme with grant agreements 724602 and 821205. We would like to thank the reviewers for their helpful comments.

## References

- Albrecht, B. A. Aerosols, cloud microphysics, and fractional cloudiness. *Science*, 245(4923):1227–1230, 1989. ISSN 0036-8075. doi: 10.1126/science.245.4923.1227. URL <https://science.sciencemag.org/content/245/4923/1227>.
- Bellouin, N., Quaas, J., Gryspeerdt, E., Kinne, S., Stier, P., Watson-Parris, D., Boucher, O., Carslaw, K. S., Christensen, M., Daniau, A.-L., Dufresne, J.-L., Feingold, G., Fiedler, S., Forster, P., Gettelman, A., Haywood, J. M., Lohmann, U., Malavelle, F., Mauritsen, T., McCoy, D. T., Myhre, G., Mülmenstädt, J., Neubauer, D., Possner, A., Rugenstein, M., Sato, Y., Schulz, M., Schwartz, S. E., Sourdeval, O., Storelvmo, T., Toll, V., Winker, D., and Stevens, B. Bounding global aerosol radiative forcing of climate change. *Reviews of Geophysics*, 58(1):e2019RG000660, 2020. doi: <https://doi.org/10.1029/2019RG000660>. URL <https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2019RG000660>. e2019RG000660 10.1029/2019RG000660.
- Beucler, T., Pritchard, M., Gentine, P., and Rasp, S. Towards physically-consistent, data-driven models of convection, 2020.



- Gentine, P., Pritchard, M., Rasp, S., Reinaudi, G., and Yacalis, G. Could machine learning break the convection parameterization deadlock? *Geophysical Research Letters*, 45(11):5742–5751, 2018. doi: <https://doi.org/10.1029/2018GL078202>. URL <https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2018GL078202>.
- Gettelman, A., Gagne, D. J., Chen, C.-C., Christensen, M. W., Lebo, Z. J., Morrison, H., and Gantos, G. Machine learning the warm rain process. *Journal of Advances in Modeling Earth Systems*, 13(2):e2020MS002268, 2021. doi: <https://doi.org/10.1029/2020MS002268>. URL <https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2020MS002268>. e2020MS002268 2020MS002268.
- IPCC. *Climate Change 2013: The Physical Science Basis. Contribution of Working Group I to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change*. Cambridge University Press, Cambridge, United Kingdom and New York, NY, USA, 2013. ISBN 978-1-107-66182-0. doi: 10.1017/CBO9781107415324. URL [www.climatechange2013.org](http://www.climatechange2013.org).
- Kokkola, H., Korhonen, H., Lehtinen, K. E. J., Makkonen, R., Asmi, A., Järvenoja, S., Anttila, T., Partanen, A.-I., Kulmala, M., Järvinen, H., Laaksonen, A., and Kerminen, V.-M. Salsa: a sectional aerosol module for large scale applications. *Atmospheric Chemistry and Physics*, 8(9):2469–2483, 2008. doi: 10.5194/acp-8-2469-2008. URL <https://acp.copernicus.org/articles/8/2469/2008/>.
- Nair, A. A. and Yu, F. Using machine learning to derive cloud condensation nuclei number concentrations from commonly available measurements. *Atmospheric Chemistry and Physics*, 20(21):12853–12869, 2020. doi: 10.5194/acp-20-12853-2020. URL <https://acp.copernicus.org/articles/20/12853/2020/>.
- O’Gorman, P. A. and Dwyer, J. G. Using machine learning to parameterize moist convection: Potential for modeling of climate, climate change, and extreme events. *Journal of Advances in Modeling Earth Systems*, 10(10):2548–2563, 2018. doi: <https://doi.org/10.1029/2018MS001351>. URL <https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2018MS001351>.
- Rasp, S. Coupled online learning as a way to tackle instabilities and biases in neural network parameterizations: general algorithms and lorenz 96 case study (v1.0). *Geoscientific Model Development*, 13(5):2185–2196, 2020. doi: 10.5194/gmd-13-2185-2020. URL <https://gmd.copernicus.org/articles/13/2185/2020/>.
- Rasp, S., Pritchard, M. S., and Gentine, P. Deep learning to represent subgrid processes in climate models. *Proceedings of the National Academy of Sciences*, 115(39):9684–9689, 2018. ISSN 0027-8424. doi: 10.1073/pnas.1810286115. URL <https://www.pnas.org/content/115/39/9684>.
- Silva, S. J., Ma, P.-L., Hardin, J. C., and Rothenberg, D. Physically regularized machine learning emulators of aerosol activation. *Geoscientific Model Development Discussions*, 2020:1–19, 2020. doi: 10.5194/gmd-2020-393. URL <https://gmd.copernicus.org/preprints/gmd-2020-393/>.
- Tegen, I., Neubauer, D., Ferrachat, S., Siegenthaler-Le Drian, C., Bey, I., Schutgens, N., Stier, P., Watson-Parris, D., Stanelle, T., Schmidt, H., Rast, S., Kokkola, H., Schultz, M., Schroeder, S., Daskalakis, N., Barthel, S., Heinold, B., and Lohmann, U. The global aerosol–climate model echam6.3–ham2.3 – part 1: Aerosol evaluation. *Geoscientific Model Development*, 12(4):1643–1677, 2019. doi: 10.5194/gmd-12-1643-2019. URL <https://gmd.copernicus.org/articles/12/1643/2019/>.
- Twomey, S. Pollution and the planetary albedo. *Atmospheric Environment (1967)*, 8(12):1251–1256, 1974. ISSN 0004-6981. doi: [https://doi.org/10.1016/0004-6981\(74\)90004-3](https://doi.org/10.1016/0004-6981(74)90004-3). URL <https://www.sciencedirect.com/science/article/pii/0004698174900043>.
- Vignati, E., Wilson, J., and Stier, P. M7: An efficient size-resolved aerosol microphysics module for large-scale aerosol transport models. *Journal of Geophysical Research: Atmospheres*, 109(D22), 2004. doi: <https://doi.org/10.1029/2003JD004485>. URL <https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2003JD004485>.

## A. Appendix

Table 4. Regression performance of different machine learning approaches. Average scores over all predicted variables

METHOD	RMSE	$R^2$
RANDOM FOREST	0.9570	-2.139
GRADIENT BOOSTING	1.36	-11.197
NEURAL NETWORK	0.249	0.892

Table 5. Modelled variables.

VARIABLE	UNIT	INPUT	OUTPUT
PRESSURE	PA	✓	
TEMPERATURE	K	✓	
REL. HUMIDITY	-	✓	
IONIZATION RATE	-	✓	
CLOUD COVER	-	✓	
BOUNDARY LAYER	-	✓	
FOREST FRACTION	-	✓	
H2SO4 PROD. RATE	$cm^{-3}s^{-1}$	✓	
SS AS MASS	$\mu g m^{-3}$	✓	
SS CS MASS	$\mu g m^{-3}$	✓	
H2SO4 MASS	$\mu g m^{-3}$	✓	✓
SO4 NS MASS	$molec. m^{-3}$	✓	✓
SO4 KS MASS	$molec. m^{-3}$	✓	✓
SO4 AS MASS	$molec. m^{-3}$	✓	✓
SO4 CS MASS	$molec. m^{-3}$	✓	✓
BC KS MASS	$\mu g m^{-3}$	✓	✓
BC AS MASS	$\mu g m^{-3}$	✓	✓
BC CS MASS	$\mu g m^{-3}$	✓	✓
BC KI MASS	$\mu g m^{-3}$	✓	✓
OC KS MASS	$\mu g m^{-3}$	✓	✓
OC AS MASS	$\mu g m^{-3}$	✓	✓
OC CS MASS	$\mu g m^{-3}$	✓	✓
OC KI MASS	$\mu g m^{-3}$	✓	✓
DU AS MASS	$\mu g m^{-3}$	✓	✓
DU CS MASS	$\mu g m^{-3}$	✓	✓
DU AI MASS	$\mu g m^{-3}$	✓	✓
DU CI MASS	$\mu g m^{-3}$	✓	✓
NS CONCENTRATION	$cm^{-3}$	✓	✓
KS CONCENTRATION	$cm^{-3}$	✓	✓
AS CONCENTRATION	$cm^{-3}$	✓	✓
CS CONCENTRATION	$cm^{-3}$	✓	✓
KI CONCENTRATION	$cm^{-3}$	✓	✓
AI CONCENTRATION	$cm^{-3}$	✓	✓
CI CONCENTRATION	$cm^{-3}$	✓	✓
NS WATER	$kg m^{-3}$		✓
KS WATER	$kg m^{-3}$		✓
AS WATER	$kg m^{-3}$		✓
CS WATER	$kg m^{-3}$		✓

Table 6. Regression performance back-transformed values of the M7 emulator for all variables. Second column is the tendency in original units and third row the full values.

VARIABLE	$R^2$ TEND	$R^2$ FULL
H2SO4 MASS	0.683	0.708
SO4 NS MASS	0.000	0.000
SO4 KS MASS	$-3.9 \cdot 10^{42}$	$-7.5 \cdot 10^{37}$
SO4 AS MASS	0.259	1.0
SO4 CS MASS	-0.672	1.0
BC KS MASS	0.938	1.0
BC AS MASS	0.921	1.0
BC CS MASS	-42.842	1.0
BC KI MASS	0.937	1.0
OC KS MASS	0.624	1.0
OC AS MASS	0.744	1.0
OC CS MASS	0.691	1.0
OC KI MASS	0.731	1.0
DU AS MASS	0.898	1.0
DU CS MASS	0.991	1.0
DU AI MASS	0.992	1.0
DU CI MASS	0.992	1.0
NS CONCENTRATION	0.031	0.031
KS CONCENTRATION	0.250	1.0
AS CONCENTRATION	0.869	1.0
CS CONCENTRATION	0.985	1.0
KI CONCENTRATION	0.781	1.0
AI CONCENTRATION	0.984	1.0
CI CONCENTRATION	0.989	1.0
NS WATER	0.863	0.863
KS WATER	0.994	0.994
AS WATER	0.933	0.933
CS WATER	0.975	0.975

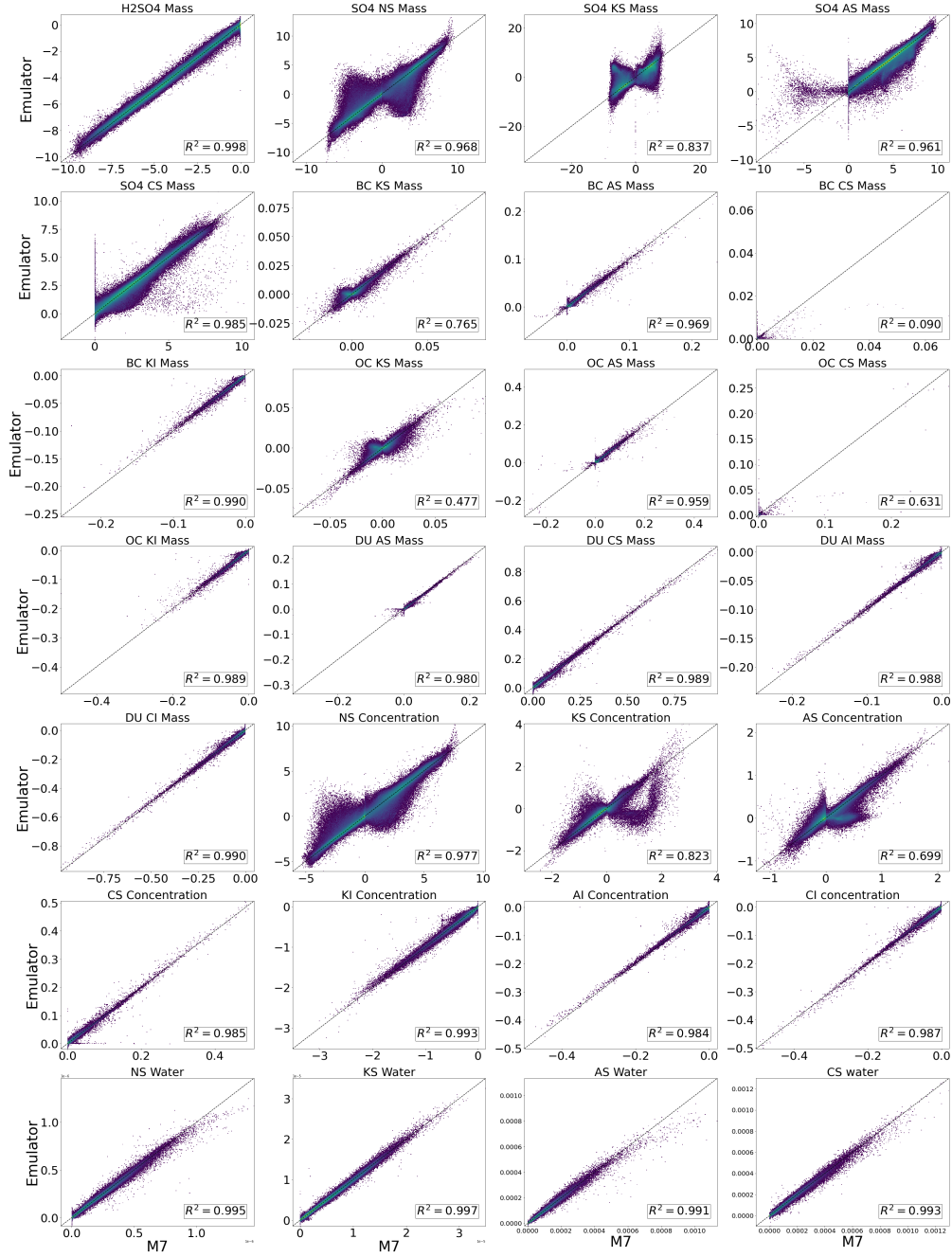


Figure 2. Predicted value by emulator versus true value from M7 model, tendencies, logarithmically transformed.