ETH zürich

PAUL SCHERRER INSTITUT
PSI

Assessing the Influence of Measurement Errors on the Performance of Neural Networks for the Retrieval of Aerosol Properties from Light Scattering Measurements

Bachelor Thesis

written by

Ryan Ammann

supervised by

Dr. Andreas Adelmann

scientific advisor

Dr. Romana Boiger

Contents

1 Introduction **2**

2 Problem Setup **3**
2.1 Light Scattering Measurements 3
2.2 Macroscopic Aerosol Properties 4
2.3 Synthetic Data 5
2.4 Ill-Posedness 5

3 Solution Approaches and Implementation **6**
3.1 Forward Neural Network with Gaussian Noise Layer 6
3.2 Invertible Neural Network with Perturbed Loss 7
3.3 Data Preprocessing 8

4 Results **8**
4.1 Forward Neural Network Trained without Gaussian Noise Layer 9
4.2 Forward Neural Network Trained with Gaussian Noise Layer 12
4.3 Invertible Neural Network Trained without Perturbed Loss 15
4.4 Invertible Neural Network Trained with Perturbed Loss 18

5 Conclusion **24**

1 Introduction

Aero-solutions, short aerosols, describe any substance in either solid and/or liquid form (except pure water and ice) suspended in the atmosphere. Those, emitted directly as particles through natural or anthropogenic processes are so-called primary aerosols. These primary sources of aerosols include volcanic eruptions, emission of sea spray, release of soil and rock debris as well as fuel combustion and general vehicular traffic dust. Particles created through condensation of atmospheric vapors are secondary aerosols [1]. The distribution of aerosols in the atmosphere influences the blueness of the sky and redness of the sunset. It influences whether it is clear or cloudy and whether it is hot or cold. By controlling the way incoming solar light is scattered or absorbed, aerosols strongly influence the climate. By reflecting away incoming radiation, aerosols that scatter light tend to have a cooling effect on the climate, while those which absorb light warm it up. Furthermore, aerosols can serve as cloud-condensation nuclei and ice nuclei upon which droplets and crystals form. Thus, investigating aerosols is of great importance for understanding and predicting the climate. Aerosol measurements can be performed by satellites, aircraft or even ground based instruments, of which the latter is considered in this work.
Previously, inferring properties of aerosols from measurements could be done using either look-up tables or physically-based algorithms, such as GRASP-OPEN [2]. Look-up tables are fast but inaccurate and physically-based algorithms are slow but accurate. The development of machine learning methods over the last years creates new options, such as the possibility of both fast and accurate retrieval of aerosol properties. A more detailed account of this is given in [3]. Hence, in this work the retrieval of aerosol properties from measurement data using neural networks is considered. Specifically, the influence of measurement errors on the performance of different neural network architectures is investigated. In total, five neural networks are trained with noise-free synthetic measurement data. Their performance is then tested with different levels of noisy input data and compared to each other. In Chapter 2 the general problem is described, including the type of measurement data, the aerosol properties and the synthetic dataset. Furthermore, the challenge of ill-posedness is explained. Chapter 3 discusses the solution methods and implementation and their results are discussed in Chapter 4. Finally, this work is concluded in Chapter 5.

2 Problem Setup

Given the light scattering measurements of aerosol particles, we want to determine their corresponding macroscopic properties using neural networks, especially in the presence of measurement errors.

2.1 Light Scattering Measurements

Each light scattering data-point consists of two quantities measured at different angles and wavelengths, see Fig. 1. The scattering phase function $P_{11}(\theta, \lambda)$ relates the intensity of the scattered light $I_{scat}(\theta, \lambda)$ to the intensity of the incoming light I_{in}, while the polarized phase function $P_{ppf}(\theta, \lambda)$ measures the degree of polarization $Q_{scat}(\theta, \lambda)$.

$$P_{11}(\theta, \lambda) = K^{-1} \frac{I_{scat}(\theta, \lambda)}{I_{in}} , \qquad P_{ppf}(\theta, \lambda) = -\frac{Q_{scat}(\theta, \lambda)}{I_{scat}(\theta, \lambda)}$$

Theoretically, measurements for $\theta \in [0°, 180°]$ carry valuable information. In practice however, the angles at the extreme ends and the middle of the interval are truncated due to physical limitations of the measuring device and stray light effects, respectively. Effectively, measurements are taken for

$$\theta \in \{x \in \mathbb{N} \mid 5° \leq x \leq 84°\} \cup \{x \in \mathbb{N} \mid 96° \leq x \leq 175°\}$$

with wavelengths

$$\lambda \in \{450nm, 532nm, 630nm\}.$$

The measuring device used is the PSI polar nephelometer, modeled after the Polarized Imaging Nephelometer [4], depicted in Fig. 2.

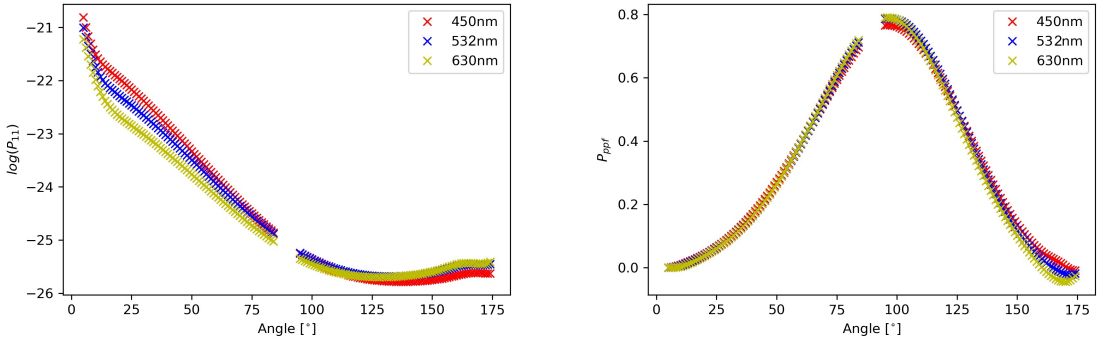


Figure 1: Example of one light scattering data-point. The abscissa contains the angles at which measurements are taken, while the ordinates are $log(P_{11})$ on the left and P_{ppf} on the right.

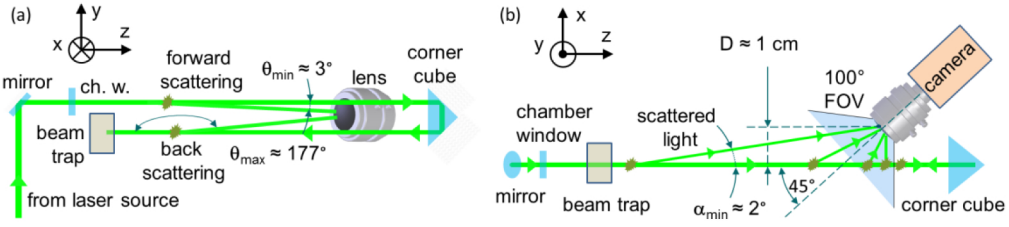


Figure 2: Schematic of the Polarized Imaging Nephelometer [4]

2.2 Macroscopic Aerosol Properties

The macroscopic aerosol properties of interest and their assumed ranges are represented in Table 1.

Property	Unit	Interval
α_{fine}	1	$[1, 7]$
α_{coarse}	1	$[1.5, 2.4]$
$k_{450,fine}$	1	$[0, 0.2]$
$k_{450,coarse}$	1	$[0.0001, 0.02]$
n_{fine}	1	$[1.33, 1.6]$
n_{coarse}	1	$[1.45, 1.6]$
V_{tot}	m^3cm^{-3}	$[0, 100]$
FMF	1	$[0.53, 0.93]$
μ_{fine}	μm	$[0.126, 0.163]$
μ_{coarse}	μm	$[0.87, 1.3]$
σ_{fine}	1	$[0, 3]$
σ_{coarse}	1	$[0, 3]$
sph	1	$[0.17, 0.85]$

Table 1: Macroscopic properties of the aerosols.

Explicitly, these are the Angstrom exponents α_{fine} and α_{coarse}, the imaginary parts of the refractive indices k_{fine} and k_{coarse}, the real parts of the refractive indices n_{fine} and n_{coarse}, the total volume concentration V_{tot}, the fine-mode-fraction FMF, the means of the two modes μ_{fine} and μ_{coarse}, the geometric standard deviations of the two modes σ_{fine} and σ_{coarse} and the spherical particle percentage sph.
Previous research in the field of aerosol size distribution suggests that a multimodal size distribution better represents the atmospheric distribution than a unimodal one [5]. Accordingly, a bi-lognormal distribution is assumed, see Fig. 3. Particles belonging to the left mode are typically called $fine$ and those belonging to the right mode are called $coarse$. The concentration of particles $C(r)$ for a given radius r is given by

$$C(r) = \frac{FMF * V_{tot}}{\sqrt{2\pi}\log(\sigma_{fine})} \exp(-\frac{(\log(r) - log(\mu_{fine}))^2}{2(\log(\sigma_{fine}))^2}) + \frac{(1 - FMF) * V_{tot}}{\sqrt{2\pi}\log(\sigma_{coarse})} \exp(-\frac{(\log(r) - log(\mu_{coarse}))^2}{2(\log(\sigma_{coarse}))^2}).$$

Note that the imaginary parts of the refractive indices for $\lambda \in \{532nm, 630nm\}$ are not explicitly listed as properties of interest. This is because they are implicitly defined through the empirically determined relation [6]

$$k(\lambda_2) = k(\lambda_1)(\frac{\lambda_2}{\lambda_1})^{1-\alpha},$$

which holds for both the coarse and fine mode. The real parts of the refractive index do not vary for the considered wavelengths, which is why only a single real part of the refractive index is given per mode:

$$n(\lambda_1) = n(\lambda_2).$$

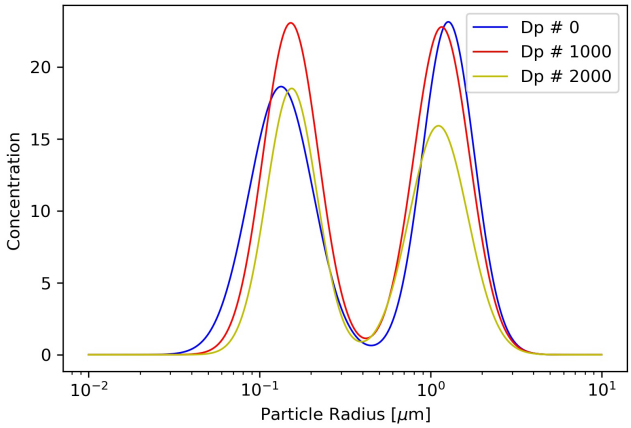


Figure 3: Size distribution of three arbitrarily chosen datapoints

2.3 Synthetic Data

As mentioned in Chapter 2.2, the particles are assumed to be bi-lognormally distributed with respect to their radii. The GRASP-OPEN algorithm [2] was used to generate the scattering data corresponding to the macroscopic properties. In total 100000 samples were provided to train and test the neural network. The dataset itself did not contain any measurement errors. The 100000 samples were randomly split into 80000 samples for training and 20000 samples for testing.

2.4 Ill-Posedness

Given the set of light scattering data $Y = \{P_{11}, P_{ppf}\}$ and the set of macroscopic properties X, $y \in Y$ can be determined from $x \in X$ by

$$y = F(x),$$

where F denotes the Mie scattering theory [7]. Determining $x \in X$ from $y \in Y$ requires solving the inverse problem

$$x = F^{-1}(y).$$

A problem is said to be well-posed if the following condtions hold:

(i) a solution exists

(ii) the solution is unique

(iii) the solution's behaviour changes continuously with the initial conditions

The inverse problem of interest is in violation of (iii) and is thus ill-posed, meaning small perturbations in the light scattering measurements $y \in Y$ can lead to huge errors in the retrieval of macroscopic aerosol properties $x \in X$. This needs to be considered in the applied solution methods.

3 Solution Approaches and Implementation

Methods that can deal with ill-posedness are called regularization methods. Chapters 3.1 and 3.2 discuss the two regularization methods, that are used in this work.

3.1 Forward Neural Network with Gaussian Noise Layer

One possible approach is to train a dense forward neural network, where, during training, the second layer of neurons acts as a regularization layer that adds gaussian noise onto the input (assuming the first layer acts as the input layer). This method is referred to as "jittering" [8]. A detailed mathematical discussion can be found in [8], however I still intend to give the reader an intuition for this method: Without the use of a regularization layer, the training data consists of scattering data-points, $y_i \in Y$, each uniquely mapped to a properties data-point, $x_i \in X$. This way, there is no restriction on the smoothness of the resulting function f. This would lead to behaviour in violation of property (iii) of well-posed problems, see Chapter 2.4.

$$\|f(y_1) - f(y_2)\| \ggg 0 \quad \text{where} \quad \|y_1 - y_2\| \approx 0.$$

Introducing a gaussian noise layer creates distributions of points, N_i, each centered at $y_i \in Y$, respectively,

$$N_i = y_i + \epsilon \ , \ \ \epsilon \sim N(0, \sigma) \ .$$

Each point $n \in N_i$ is mapped to x_i. This imposes a smoothness restriction onto the resulting function, since points in each others close vicinity are mapped to the same output during training.
The neural network is implemented in python with the Tensorflow-Keras API [9]. The Asynchronous Successive Halving Algorithm [10], short ASHA, was used to sample and train the most promising hyperparameter configurations, see Table 2. ASHA was started with 75 concurrently training neural networks. They were trained on the Merlin HPC Cluster at the Paul Scherrer Institute.

Hyperparameter	Space
depth	$\{x \in \mathbb{N} \mid 2 \leq x \leq 10\}$
width	$\{x \in \mathbb{N} \mid 10 \leq x \leq 160\}$
learning rate	$[0.01, 0.001]$
batch size	$\{8, 16, 32, 128, 256\}$
activation function	relu
epochs	50
σ	$\{0.00, 0.05\}$

Table 2: Forward Neural Network Hyperparameters; Note: σ is fixed by user

3.2 Invertible Neural Network with Perturbed Loss

A further possibility is to employ invertible neural networks [11]. These neural networks are trained in the well defined direction (parameters x to measurements y). What differentiates them from normal networks is that, due to their architecture, the reverse direction (measurements y to parameters x) is implicitly learned during training. To deal with the ill-posedness of the problem, an additional latent variable z is introduced. It aims to capture the lost information in the forward process, i.e. to capture the information about x not contained in y, see Fig. 4.

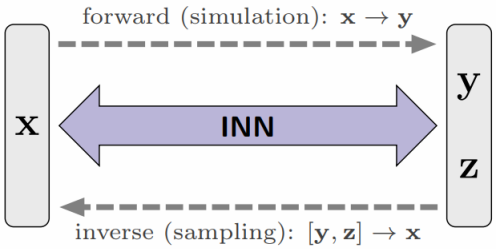


Figure 4: Schematic of an Invertible Neural Network [11]

In a further attempt to increase the neural network's robustness, gaussian noise was introduced during the training process. Concretely, after each training epoch, the current loss is calculated as a weighted sum of different losses, i.e.

$$l_{forward} = |f(x) - [y, z]_{true}|$$

$$l_z = |p(z|y) - p(z_{true})|$$

$$l_{perturbed} = |f^{-1}([y, z] + \epsilon) - x_{true}| \ , \ \ \epsilon \sim N(0, \sigma)$$

$$l_{total} = \frac{w_{forward} * l_{forward} + w_z * l_z + w_{perturbed} * l_{perturbed}}{w_{forward} + w_z + w_{perturbed}}.$$

Note, the forward and reverse evaluations of the network are denoted as f and f^{-1}, respectively. Also, this is a simplified representation of the loss, the full details of which can be found in the original work [11]. The invertible neural network is also implemented in python with the Tensorflow-Keras API [9]. The hyperparameterspace explored by ASHA [10] is shown in Table. 3.

Hyperparameter	Space
blocks	$\{3, 4, 5\}$
depth	$\{2, 3, 4\}$
width	$\{x \in \mathbb{N} \mid 80 \leq x \leq 150\}$
learning rate	$[0.0001, 0.001]$
batch size	$\{8, 16, 32, 128, 256\}$
$w_{forward}$	$\{x \in \mathbb{N} \mid 1 \leq x \leq 350\}$
$w_{reverse}$	$\{x \in \mathbb{N} \mid 1 \leq x \leq 350\}$
$w_{perterbed}$	$\{x \in \mathbb{N} \mid 220 \leq x \leq 350\}$
activation function	{relu, linear}
epochs	50
σ	$\{0.00, 0.01, 0.05\}$

Table 3: Invertible Neural Network Hyperparameters; Note: σ is fixed by user

3.3 Data Preprocessing

Before training the neural networks, the raw synthetic dataset needs to be preprocessed. For instance, instead of P_{11}, $log(P_{11})$ is used. Additionally, the Standard Scaler by scikit-learn [12] is used on the scattering data:

$$stdscaler(x) = \frac{x - u}{s}$$

where u is the mean and s the standard deviation of x.
The aerosol properties are scaled using the Min-Max Scaler by scikit-learn [12], which works as follows:

$$minmax(x) = x_{std} * (max - min) + min$$

$$x_{std} = (x - min(X))/(max(X) - min(X))$$

where min and max are the limits of the interval to which the data is scaled and $min(X)$ and $max(X)$ are the minimum and maximum of dataset X, respectively.
The imaginary parts of the refractive indices k_{450}, k_{532}, k_{630} are macroscopic properties present in the raw data. Since these quantities are strongly correlated to each other, considering them as independent properties is incorrect. A better approach is to express them as a single refractive index and the angstrom exponent, see Chapter 2.2,

$$k_{450}, k_{532}, k_{630} \rightarrow k_{450}, \alpha.$$

4 Results

In this chapter the performance of the neural networks from Chapter 3.3 is discussed. More precisely, how well they infer the macroscopic properties of interest, given different levels of noisy light scattering data.
In total, five different neural network architectures were tested. Namely, a forward neural network, a forward neural network trained with a gaussian noise layer $\epsilon \sim N(0, 0.05)$, an invertible neural network and two invertible neural networks trained with perturbed loss with $\epsilon \sim N(0, 0.01)$ and $\epsilon \sim N(0, 0.05)$. For each architecture, the hyperparameterspaces (see Chapters 3.1 and 3.2) are explored using ASHA [10] and the best performing model is selected. The models are compared using the coefficient of determination R^2. The R^2 value is the proportion of variation in the aerosol properties that is predictable from the scattering measurements. It is defined as

$$R^2 = 1 - \frac{\sum_i (x_i - f_i)^2}{\sum_i (x_i - \bar{x}_i)^2},$$

where f_i is the prediction, x_i the true value of the properties of datapoint i and $\bar{x}_i$ the mean.
Finally, the performance of each architecture's best model is evaluated on the testing dataset, where varying levels of noise are applied each light scattering datapoint i as follows,

$$y_{i,test} = y_i + N_i(0, \sigma),\ i = 1, 2, ..., 20000$$

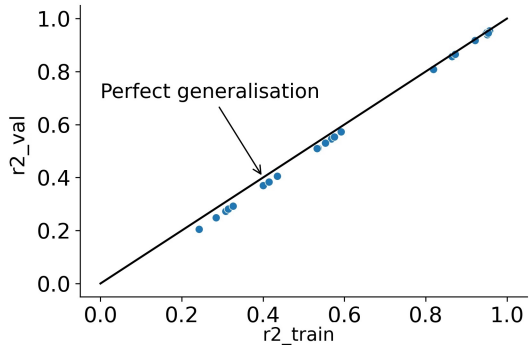
where σ is the standard deviation of the gaussian distribution N_i, centered at 0. The metric used to evaluate the performance is the Weighted Mean Absolute Percentage Error, defined as

$$WMAPE_i = \frac{\sum_{t=1}^{n} |A_{t,i} - F_{t,i}|}{\sum_{t=1}^{n} |A_{t,i}|} * 100\% \ ,$$

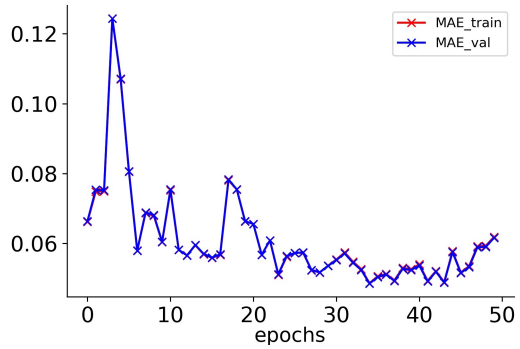
where $A_{t,i}$ and $F_{t,i}$ denote the true and predicted property i of testing datapoint t. Compared to the naively used Mean Absolute Percentage Error, the weighted version avoids division by near-zero on values close to zero.

4.1 Forward Neural Network Trained without Gaussian Noise Layer

The first model was trained without jittering [8]. The R^2 values for these neural networks are plotted in Fig. 5a. Ideally, there would be a cluster at $(1, 1)$, which would indicate that each trained neural network manages to capture the entire variation present in the training and validation data. It is also desirable that $R^2_{val} \geq R^2_{train}$, as that indicates that the neural network is at least as good at capturing the variance of the validation dataset as for the training dataset. $R^2_{val} < R^2_{train}$ is a strong indicator of overfitting. The model with the highest coefficient of determination $R^2_{val} = 0.9578$ is saved and considered for further testing. Fig. 5b shows the Mean Absolute Error of the considered model given the training and validation dataset, plotted over the training epochs. Around epoch 20, a Mean Absolute Error of 0.06 is achieved and roughly maintained untill epoch 50.



(a) Comparison between R^2_{train} and R^2_{val}



(b) MAE_{train} and MAE_{val} of the best model

Figure 5: Training metrics for forward neural networks without gaussian noise layer

The WMAPE for aerosol properties at different levels of noisy scattering data is given in Table 4. Prediction of the aerosol properties from light scattering data without noise works well, since the WMAPE is below 10% for all properties and even below 3.6% for properties, that describe the size distribution. However, the neural network performs significantly worse for light scattering measurements with noise. At the noise-level $N(0, 0.05)$, the WMAPE for properties describing the scattering of light reaches up to 73%. In particular, the Angstrom exponents α_{coarse}, α_{fine} and the imaginary parts of the refractive indices $k_{450nm,coarse}$, $k_{450nm,fine}$ are especially susceptible to noise.

WMAPE[%]	No Noise	Noise = $N(0, 0.01)$	Noise = $N(0, 0.05)$	Noise = $N(0, 0.10)$
α_{fine}	5.15	22.48	67.06	111.27
α_{coarse}	5.11	18.28	48.61	81.40
$k_{450,fine}$	5.88	16.74	73.51	158.06
$k_{450,coarse}$	9.99	31.08	71.34	92.42
n_{fine}	0.60	1.51	5.02	6.73
n_{coarse}	0.33	1.21	4.04	6.62
V_{tot}	2.57	5.29	26.19	63.32
FMF	0.89	3.24	12.58	18.71
μ_{fine}	0.42	1.57	6.58	9.43
μ_{coarse}	1.00	3.82	15.21	28.30
σ_{fine}	0.50	1.68	6.21	9.91
σ_{coarse}	0.36	1.95	5.48	8.25
sph	3.59	11.54	46.03	72.77

Table 4: WMAPE of a neural network trained without noise at different levels of noisy scattering data

Fig. 6 shows a comparison between the predicted and true size distributions for two arbitrarily chosen datapoints. At noise-level $N(0, 0.01)$, all the properties describing the size distribution, with exception of *sph*, have a WMAPE below 5.3%. Accordingly, the predicted distribution still closely follows the true distribution. At higher levels of noise, stronger deviations by properties such as V_{tot} and μ_{coarse}, lead to a larger discrepancy between predicted and true distribution.

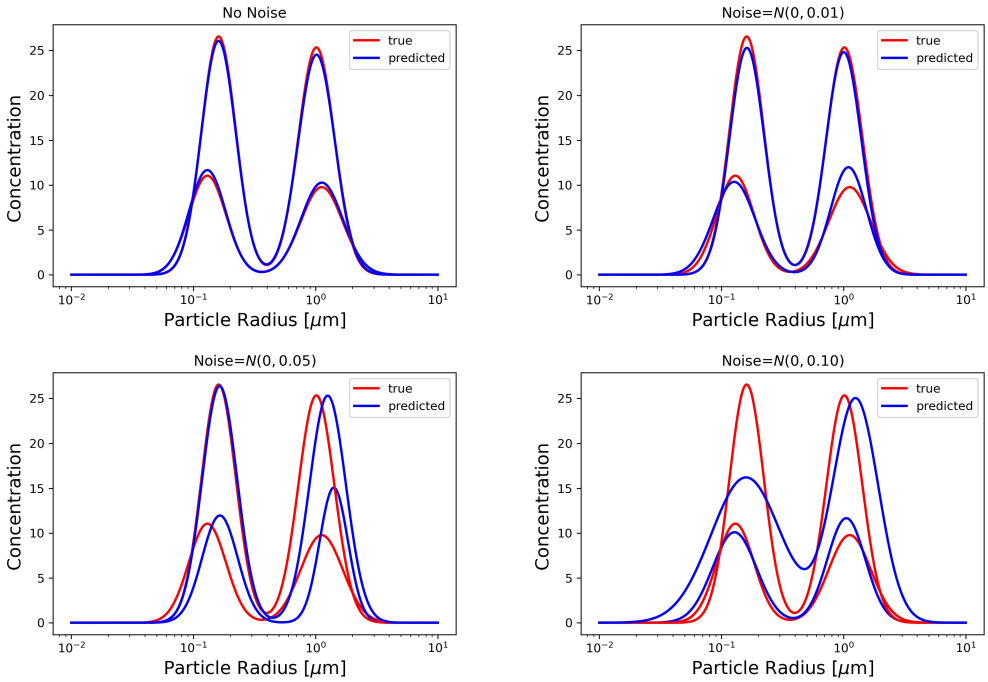


Figure 6: Predicted vs true distributions for two datapoints at different levels of noise

Fig. 7 compares the predicted and true values of the properties describing scattering of light. In accordance with the WMAPEs in Table 4, α_{coarse} and $k_{450nm,coarse}$ are severely impacted by the presence of noise, while n_{coarse} is more stable.

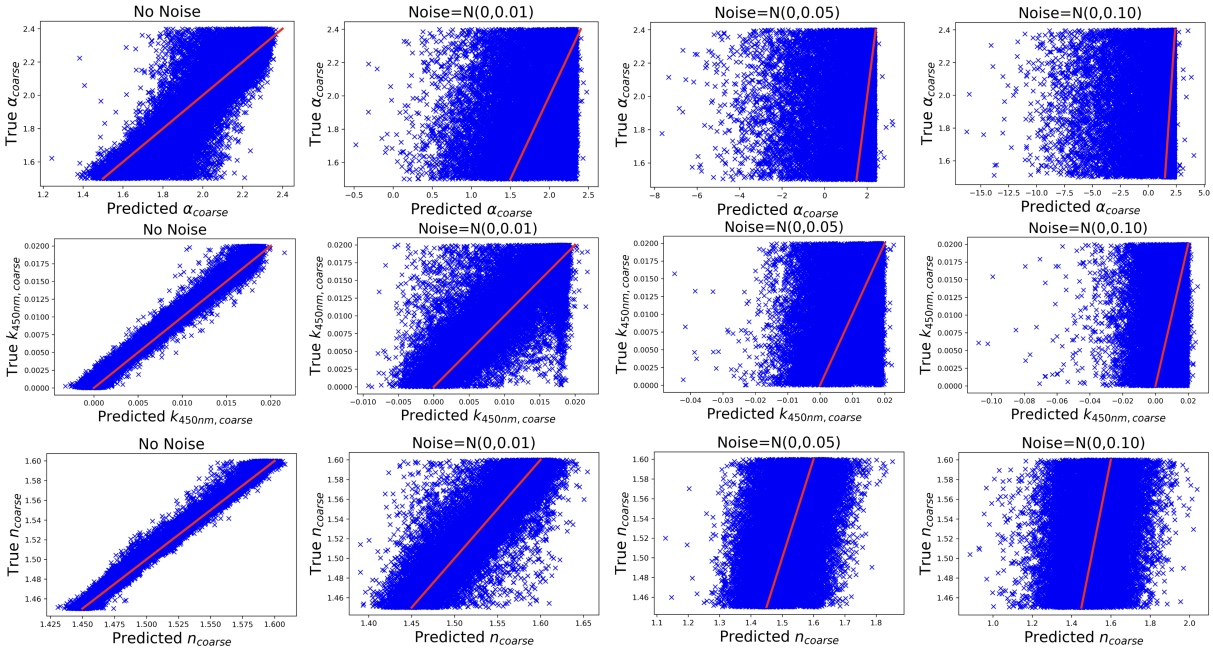
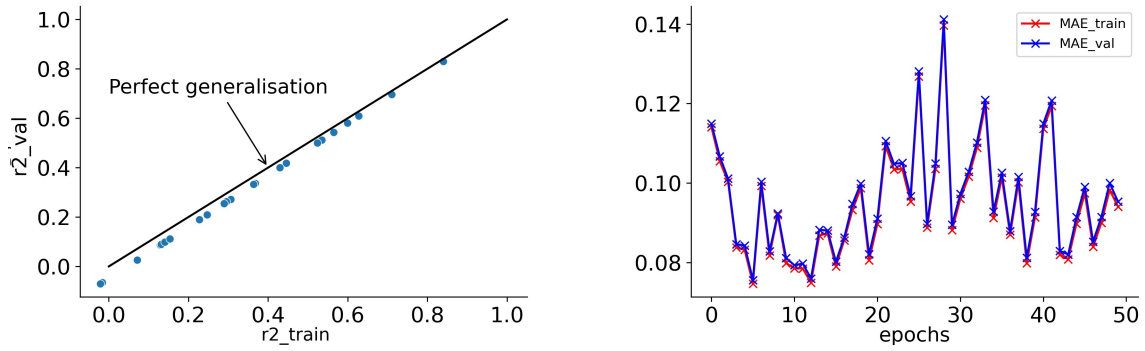


Figure 7: Comparison between predicted and true values of the properties α_{coarse}, $k_{450nm,coarse}$, n_{coarse} at different levels of noise

4.2 Forward Neural Network Trained with Gaussian Noise Layer

The following neural networks were trained with a gaussian noise layer $\epsilon \sim N(0, 0.05)$. Fig. 8a shows that these neural networks were less able to capture the variation present in the data, compared to the forward neural networks trained without jittering [8]. Consequently, the highest coefficient of determination is only $R^2_{val} = 0.8425$. In accordance, the Mean Absolute Error is also worse for the considered neural network. Fig. 8b depicts the Mean Absolute Errors consistently fluctuating around 0.10.



(a) Comparison between R^2_{train} and R^2_{val}

(b) MAE_{train} and MAE_{val} of the best model

Figure 8: Training metrics for jittered forward neural networks

Table 5 shows the performance of the considered model for different levels of noisy scattering data. For no noise, this neural network performs slightly worse than the neural network trained without a gaussian noise layer. Still, the WMAPEs for properties describing the size distribution are below 5%. As expected, the regularized neural network performs significantly better under noisy scattering data than the unregularized forward neural network. Even at noise-level $N(0, 0.10)$, the WMAPEs of most of the properties describing size distribution does not exceed 6%. The prediction of other properties such as Angstrom exponent and refractive indices are also more robust towards noisy scattering data.

WMAPE[%]	No Noise	Noise = $N(0, 0.01)$	Noise = $N(0, 0.05)$	Noise = $N(0, 0.10)$
α_{fine}	9.13	10.66	24.01	35.48
α_{coarse}	11.48	11.48	11.61	12.21
$k_{450,fine}$	5.55	6.26	15.21	27.47
$k_{450,coarse}$	13.31	16.02	37.68	54.43
n_{fine}	0.36	0.43	1.21	2.06
n_{coarse}	0.44	0.49	1.13	1.84
V_{tot}	3.31	3.47	6.28	11.83
FMF	1.08	1.43	4.35	7.51
μ_{fine}	0.77	0.88	2.26	3.94
μ_{coarse}	1.85	1.99	3.72	5.83
σ_{fine}	0.62	0.72	1.80	2.82
σ_{coarse}	1.21	1.22	1.40	1.64
sph	4.28	5.43	15.65	26.32

Table 5: WMAPE of a jittered neural network (see Chapter 3.1) at different levels of noisy scattering data

As mentioned above, the properties describing size distribution are robust with respect to noisy scattering data. This behaviour is reflected in Fig. 9, where the predicted distribution closely follows the true distribution for all but the highest level of noise.

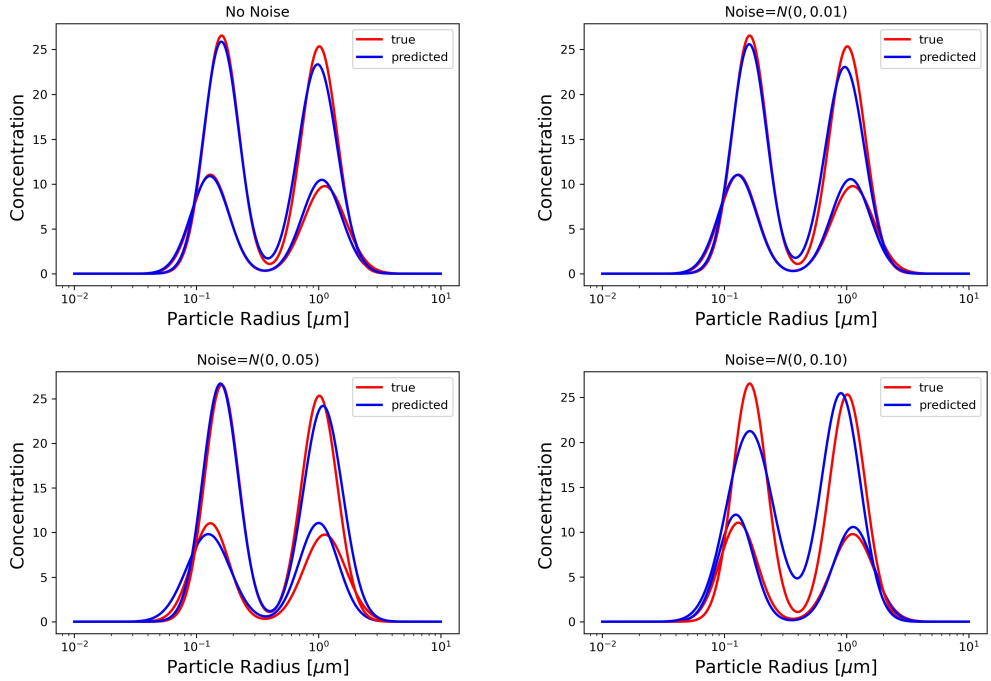


Figure 9: Predicted vs true distributions for two datapoints at different levels of noise

As shown in Fig. 10, the neural network fails to accurately predict the Angstrom exponent for the coarse mode α_{coarse}. Even given noiseless scattering data, the WMAPE still exceeds 10%, see Table 5. For the real and imaginary parts of the refractive index, $k_{450nm,coarse}$ and n_{coarse}, the WMAPE also grows with increasing noise, however at a slower rate than for the neural network trained without noise.

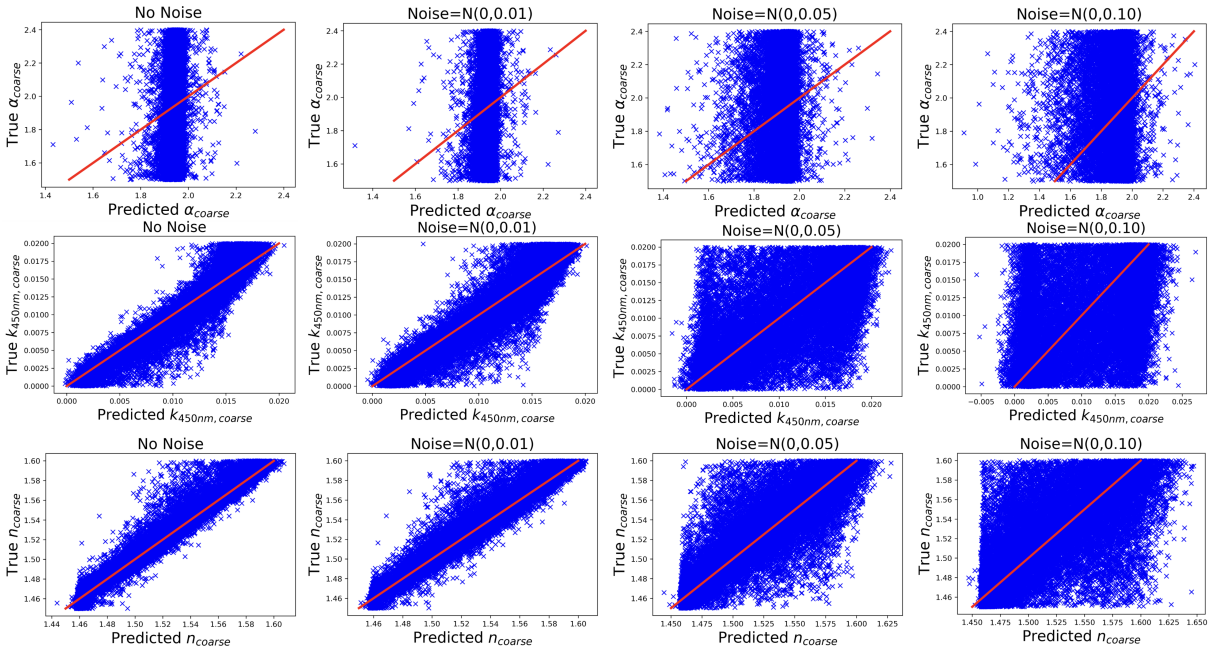
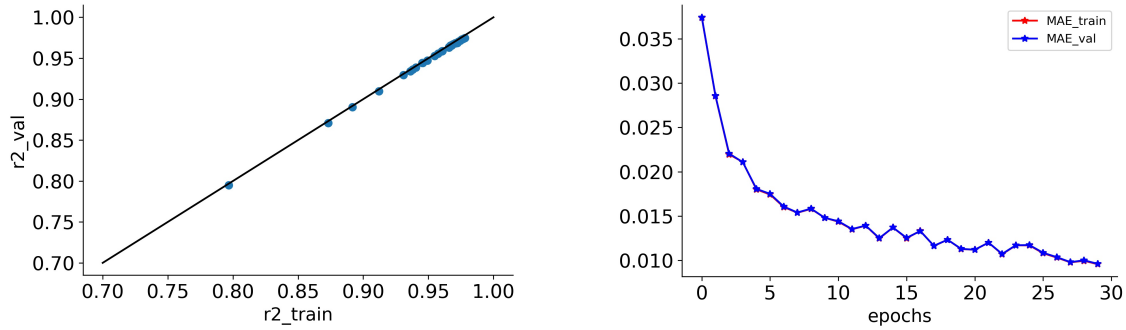


Figure 10: Comparison between predicted and true values of the properties α_{coarse}, $k_{450nm,coarse}$, n_{coarse} at different levels of noise

4.3 Invertible Neural Network Trained without Perturbed Loss

Due to the architecture of invertible neural networks (see Chapter 3.2), they are able to capture more of the variation present in the data than regular neural networks. Accordingly, the coefficients of determination in the desired direction (scattering data to properties) of the trained networks are significantly higher, with $R^2 = 0.996$ for the best one, see Fig. 11a. At 0.01, the MAE of the best network is almost an entire order of magnitude smaller than for the forward neural network, see Fig. 11b.



(a) Comparison between R^2_{train} and R^2_{val}

(b) MAE_{train} and MAE_{val} of the best model

Figure 11: Training metrics for invertible neural networks without perturbed loss

As shown in Table 6, given noiseless scattering data, the invertible neural network predicts the aerosol properties with high accuracy. All properties have a WMAPE under 5% and even under 1.5% for those describing size distribution. However, it is ill-equipped to handle noisy scattering data. Even at the noise-level $N(0, 0.01)$, there is up to a 15-fold increase in the WMAPE.

WMAPE[%]	No Noise	Noise = $N(0, 0.01)$	Noise = $N(0, 0.05)$	Noise = $N(0, 0.10)$
α_{fine}	3.87	33.62	516.32	1324.83
α_{coarse}	4.25	19.98	170.72	432.45
$k_{450,fine}$	1.59	26.13	497.70	1362.28
$k_{450,coarse}$	3.82	32.02	578.91	1584.19
n_{fine}	0.14	1.49	21.78	59.36
n_{coarse}	0.09	1.25	10.08	26.01
V_{tot}	0.78	13.65	234.59	650.94
FMF	0.35	6.55	123.00	335.30
μ_{fine}	0.20	2.59	44.71	130.29
μ_{coarse}	0.38	4.05	56.54	149.73
σ_{fine}	0.13	2.17	26.78	68.78
σ_{coarse}	0.22	2.06	24.64	61.86
sph	1.48	36.22	948.44	2613.20

Table 6: WMAPE of an invertible neural network at different levels of noisy scattering data

With WMAPE of distribution properties below 1.5%, the predicted and true distributions overlap close to perfectly, given noiseless scattering data. At noise-level $N(0, 0.01)$, the increased WMAPE of V_{tot} causes a shift in the height of peaks. At higher levels of noise the prediction falls apart.

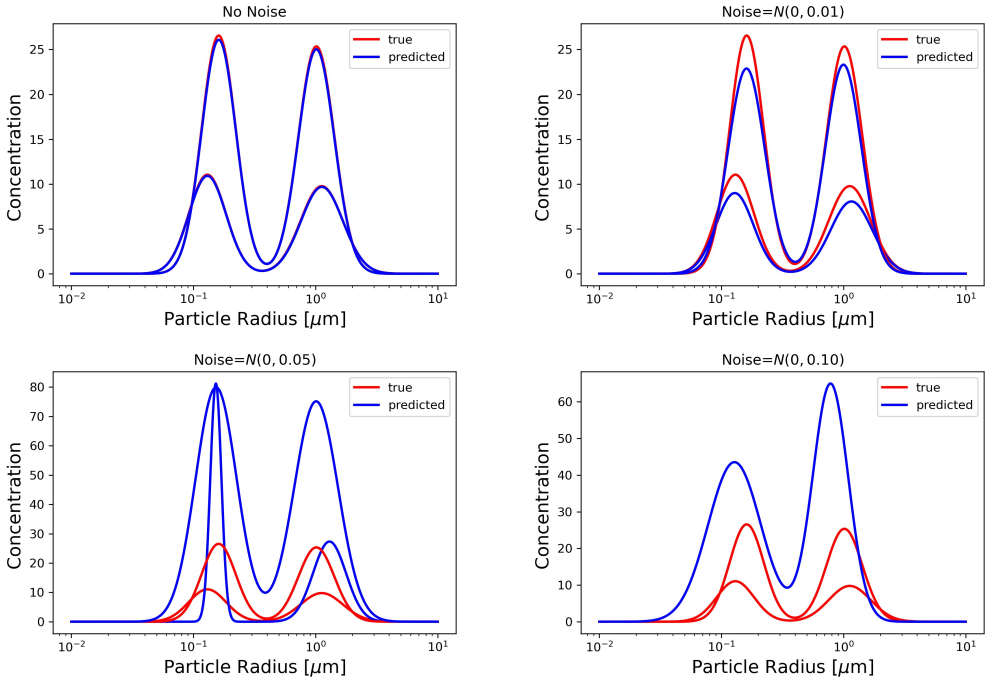


Figure 12: Predicted vs true distributions for two datapoints at different levels of noise

For noiseless scattering data, the imaginary and real part of the refractive indices, $k_{450nm,coarse}$ and n_{coarse}, are more accurately predicted than the Angstrom exponent α_{coarse}, see Fig. 13. These three properties, especially $k_{450nm,coarse}$, are very susceptible to noisy input data, see Table 6. Thus the accuracy of the prediction significantly decreases with noisy scattering data, which can be observed in Fig. 13.

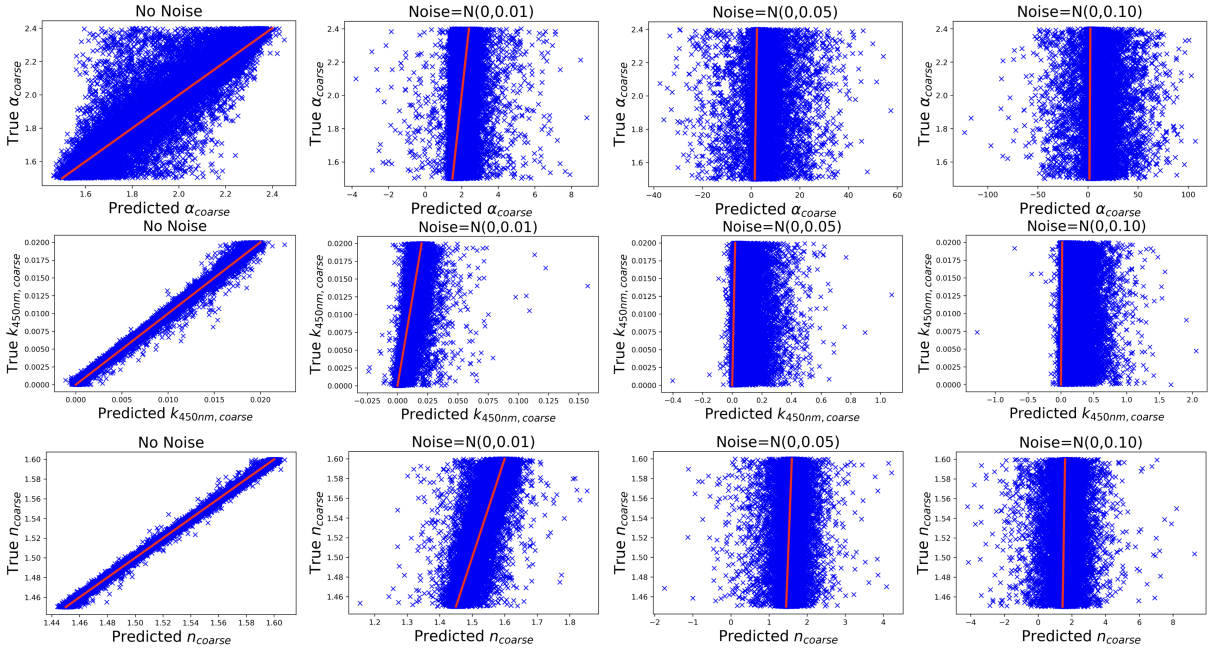
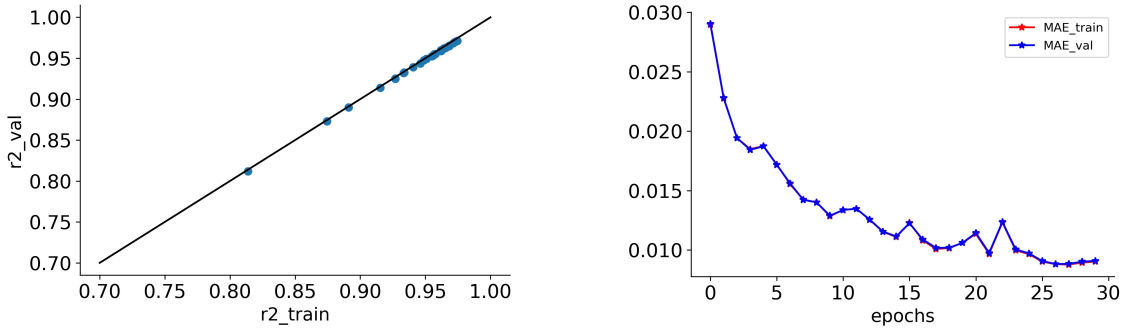


Figure 13: Comparison between predicted and true values of the properties α_{coarse}, $k_{450nm,coarse}$, n_{coarse} at different levels of noise

4.4 Invertible Neural Network Trained with Perturbed Loss

This invertible neural network was trained with perturbation of the loss $\epsilon \sim N(0, 0.01)$. Similarly to the invertible neural network trained without perturbation (Chapter 4.3), more of the variation present in the data is captured, thanks to the network's architecture. Thus a higher overall R^2 value is reached compared to the forward neural networks. The best network has a coefficient of determination of $R^2 = 0.997$ and reaches a MAE of 0.01, see Fig.14a and Fig. 14b.



(a) Comparison between R^2_{train} and R^2_{val}

(b) MAE_{train} and MAE_{val} of the best model

Figure 14: Training metrics for invertible neural networks

The additional perturbation during training, as described in Chapter 3.2, allows the network to better deal with noisy scattering data. Compared to the neural network trained without perturbation, at the highest noise-level the WMAPE reaches a maximum value of 266.18% instead of 2613.20%. For noiseless scattering data, the predictions have a maximum WMAPE of 4.33%, see Table 7.

WMAPE[%]	No Noise	Noise = $N(0, 0.01)$	Noise = $N(0, 0.05)$	Noise = $N(0, 0.10)$
α_{fine}	4.00	19.41	96.80	216.93
α_{coarse}	4.33	15.05	41.56	93.95
$k_{450,fine}$	1.80	13.33	67.21	161.00
$k_{450,coarse}$	3.93	24.30	117.52	325.61
n_{fine}	0.15	0.95	4.35	8.74
n_{coarse}	0.14	1.27	5.39	11.25
V_{tot}	0.84	8.54	41.68	94.21
FMF	0.31	3.67	17.72	43.22
μ_{fine}	0.18	1.95	8.25	16.85
μ_{coarse}	0.44	2.39	11.20	29.18
σ_{fine}	0.17	1.41	6.91	15.10
σ_{coarse}	0.29	0.92	3.29	6.55
sph	1.61	10.14	89.94	266.18

Table 7: WMAPE of an invertible neural network at different levels of noisy scattering data

Fig. 15 shows an accurate prediction of the distribution for the noiseless case. This is reflective of the fact that the properties describing the distribution have a maximum WMAPE of 1.61%. Thanks to the additional regularization in form of the perturbed loss, this network's predicted distribution more closely follows the true distribution for higher levels of noise.

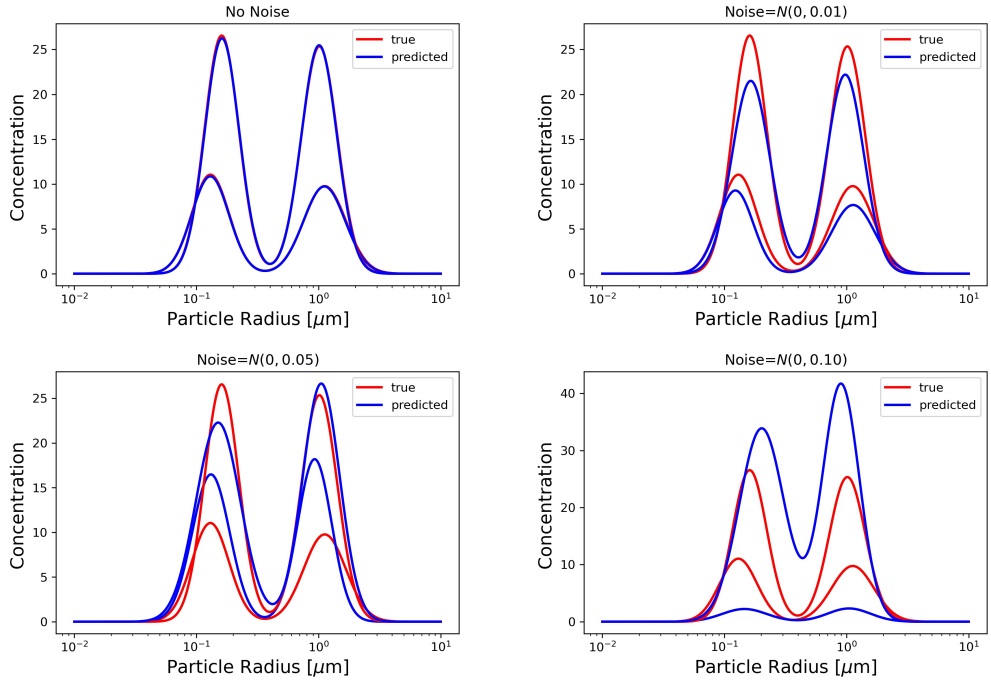


Figure 15: Predicted vs true distributions for two datapoints at different levels of noise

The prediction of the Angstrom exponent α_{coarse} and the real and imaginary parts of the refractive index n_{coarse} and $k_{450nm,coarse}$ are as accurate as the prediction in Chapter 4.3. Comparing Fig. 16 and Fig. 13 also shows that the additional perturbation during training increases robustness with respect to noisy scattering data.

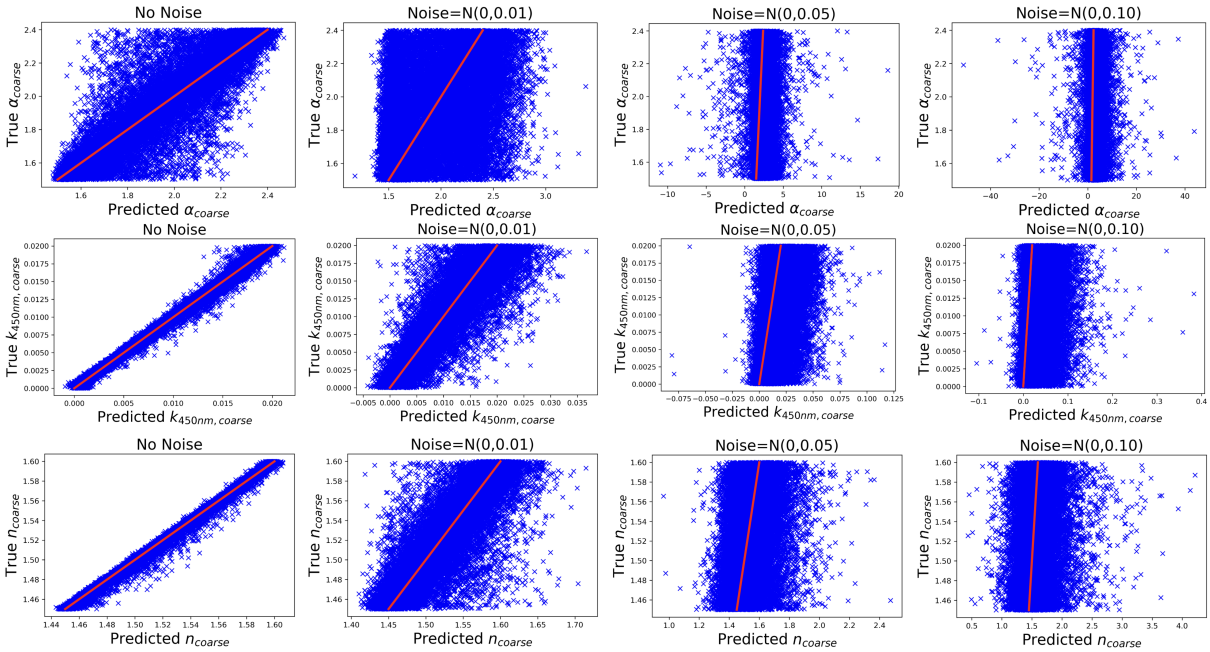
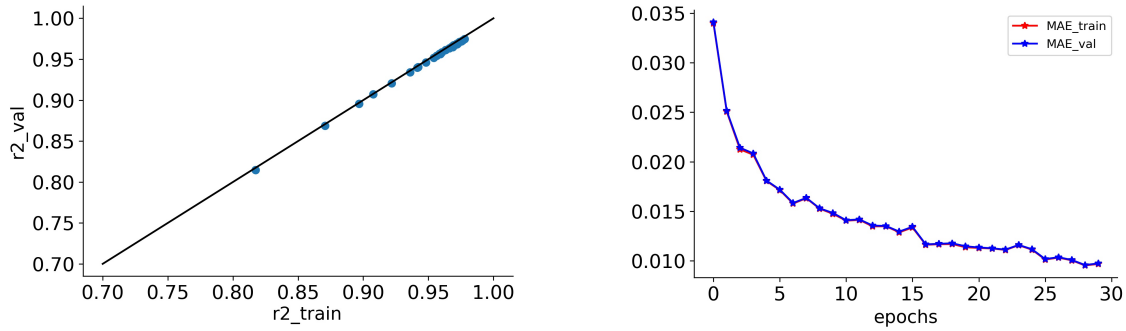


Figure 16: Comparison between predicted and true values of the properties α_{coarse}, $k_{450nm,coarse}$, n_{coarse} at different levels of noise

Followingly, invertible neural networks with a perturbed loss of $\epsilon \sim N(0, 0.05)$ are considered. The best model has a coefficient of determination of $R^2 = 0.996$ and also reaches a MAE of 0.01 during training.



(a) Comparison between R^2_{train} and R^2_{val}

(b) MAE_{train} and MAE_{val} of the best model

Figure 17: Training metrics for invertible neural networks

It performs similarly to the invertible neural network trained with perturbation of $\epsilon \sim N(0, 0.01)$, although it's worse for higher levels of noise. For instance, the WMAPE of the imaginary part of the refractive index of the coarse mode $k_{450nm,coarse}$ is three times as high.

WMAPE[%]	No Noise	Noise = $N(0, 0.01)$	Noise = $N(0, 0.05)$	Noise = $N(0, 0.10)$
α_{fine}	4.62	19.86	173.54	442.89
α_{coarse}	4.02	14.31	52.01	116.08
$k_{450,fine}$	1.63	11.27	75.41	189.85
$k_{450,coarse}$	3.63	27.91	310.16	855.76
n_{fine}	0.13	1.36	7.92	18.42
n_{coarse}	0.11	1.02	4.68	10.12
V_{tot}	0.82	8.55	55.69	140.23
FMF	0.44	2.94	17.67	47.52
μ_{fine}	0.17	2.37	14.13	36.19
μ_{coarse}	0.42	2.81	18.59	48.58
σ_{fine}	0.15	1.13	9.03	24.02
σ_{coarse}	0.28	1.20	5.18	9.27
sph	1.47	11.41	103.50	283.23

Table 8: WMAPE of an invertible neural network at different levels of noisy scattering data

The comparison of the predicted and true distribution and the predicted and true scattering properties is qualitatively the same as for the previously discussed model. They are given in Fig. 18 and Fig. 19.

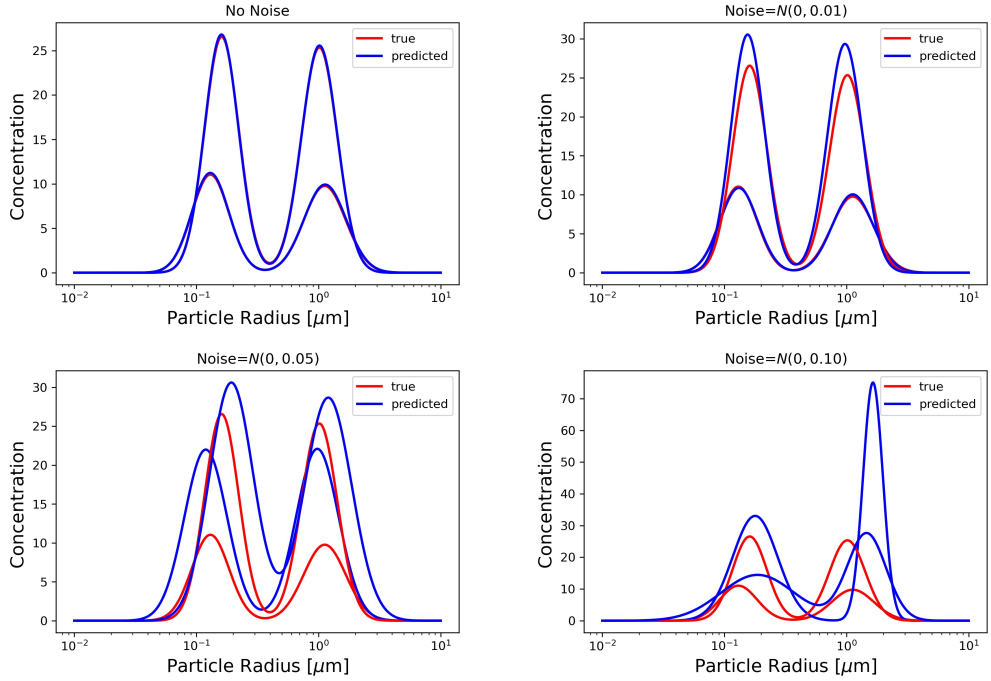


Figure 18: Predicted vs true distributions for two datapoints at different levels of noise

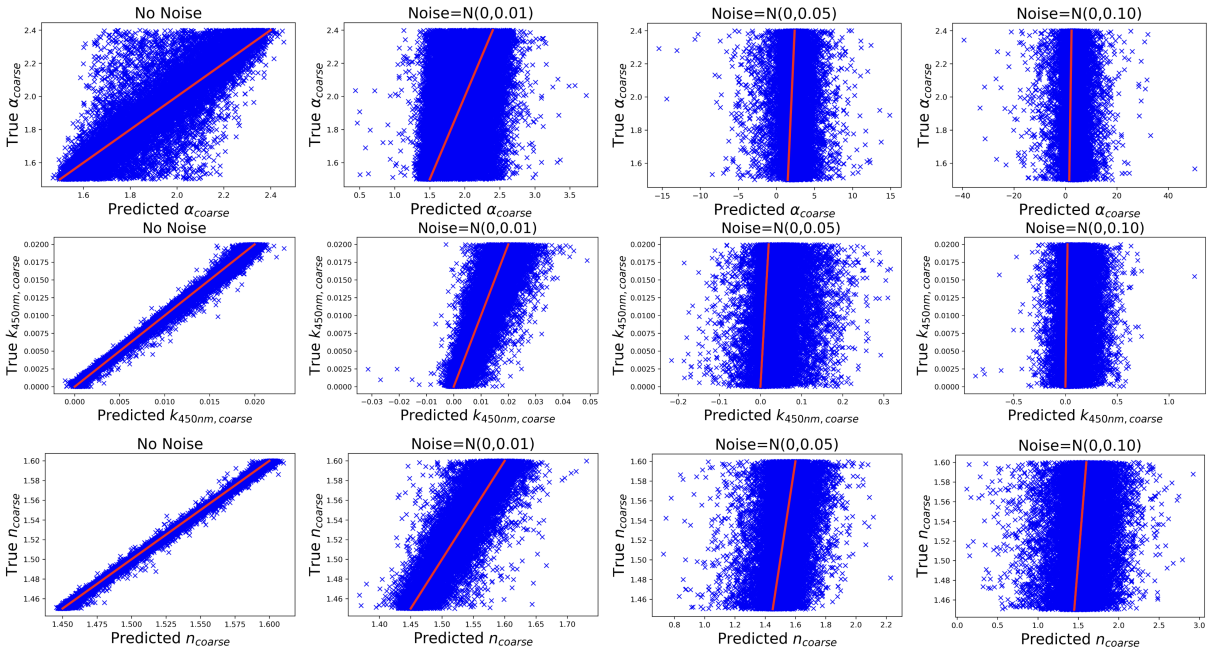


Figure 19: Comparison between predicted and true values of the properties α_{coarse}, $k_{450nm,coarse}$, n_{coarse} at different levels of noise

5 Conclusion

The goal of this work is to infer macroscopic aerosol properties from light scattering measurements using neural networks. In a first step, the raw data had to be preprocessed appropriately. This included scaling parts of the scattering data, as well as removing correlations between properties. Additionally, the problem of ill-posedness needed to be addressed. Specifically, that small perturbations in the scattering data could lead to vastly different predictions of macroscopic aerosol properties. Two different approaches to tackling this problem were then proposed. First, the use of jittered neural networks to impose a smoothness condition on the prediction. The second approach employed invertible neural networks with perturbed loss, which were trained in the well-defined direction, meaning properties to scattering data. Due to their architecture, the reverse direction is then implicitly learned.
The neural networks were then trained on the Merlin HPC Cluster at PSI, using the Asynchronous Successive Halving Algorithm to efficiently sample the best hyperparameter configurations. The neural network which was able to capture most of the variation present in the dataset was then saved and evaluated further. Comparing the error between the forward neural network trained with gaussian noise to the one trained without, a 2-3 fold reduction of the error, given noisy scattering data, can be observed. When given noiseless scattering data, the error of the network trained with gaussian noise is slightly higher for some properties. As expected, it is slightly less accurate given perfect data, but significantly more robust to measurement error. In the case of invertible neural networks, regularization can lead to a 1.2 times, all the way to a 6 times reduction of the error, given noisy scattering data. In the case of noiseless scattering data, there is no indication of a decrease in accuracy. Since physical measurements are necessarily accompanied by measurement error, using the regularized neural networks can significantly improve robustness and thus the quality of prediction.
Future work could include further increasing the robustness through different regularization techniques and neural network architectures. An attempt was made to use physics informed neural networks. Basically, the loss term is adjusted to enforce certain conditions on the predicted data. The idea was to enforce the macroscopic aerosol properties to stay within their physically possible boundaries, with the use of barrier functions. However due to time limitations, this possibility could not be explored further.

References

[1] C. Tomasi and A. Lupi, *Primary and Secondary Sources of Atmospheric Aerosol*, ch. 1, pp. 1–86. John Wiley Sons, Ltd, 2017.

[2] O. Dubovik, T. Lapyonok, P. Litvinov, M. Herman, D. Fuertes, F. Ducos, A. Lopatin, A. Chaikovsky, B. Torres, Y. Derimian, X. Huang, M. Aspetsberger, and C. Federspiel, "Grasp: a versatile algorithm for characterizing the atmosphere," *SPIE Newsroom*, 09 2014.

[3] A. Di Noia and O. P. Hasekamp, "Neural Networks and Support Vector Machines and Their Application to Aerosol and Cloud Remote Sensing: A Review," in *Springer Series in Light Scattering: Volume 1: Multiple Light Scattering, Radiative Transfer and Remote Sensing* (A. Kokhanovsky, ed.), Springer Series in Light Scattering, pp. 279–329, Cham: Springer International Publishing, 2018.

[4] G. Dolgos and J. V. Martins, "Polarized imaging nephelometer for in situ airborne measurements of aerosol light scattering," *Opt. Express*, vol. 22, pp. 21972–21990, Sep 2014.

[5] C. Davies, "Size distribution of atmospheric particles," *Journal of Aerosol Science*, vol. 5, no. 3, pp. 293–300, 1974.

[6] G. L. Schuster, O. Dubovik, and B. N. Holben, "Angstrom exponent and bimodal aerosol size distributions," *Journal of Geophysical Research: Atmospheres*, vol. 111, no. D7, 2006.

[7] D. W. Hahn, "Light scattering theory," *Department of Mechanical and Aerospace Engineering, University of Florida*, pp. 7–9, 2009.

[8] R. Reed, S. Oh, R. Marks, *et al.*, "Regularization using jittered training data," in *International Joint Conference on Neural Networks*, vol. 3, pp. 147–152, 1992.

[9] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015. Software available from tensorflow.org.

[10] L. Li, K. G. Jamieson, A. Rostamizadeh, E. Gonina, M. Hardt, B. Recht, and A. Talwalkar, "Massively parallel hyperparameter tuning," *CoRR*, vol. abs/1810.05934, 2018.

[11] L. Ardizzone, J. Kruse, S. Wirkert, D. Rahner, E. W. Pellegrini, R. S. Klessen, L. Maier-Hein, C. Rother, and U. Köthe, "Analyzing inverse problems with invertible neural networks," *arXiv preprint arXiv:1808.04730*, 2018.

[12] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.