MSc Artificial Intelligence
Faculty of Science

Focus error sensing in lithography patterns by means of machine learning

Master Thesis
ASML Veldhoven

Dirk F.I. Vet

Supervisors:
First Supervisor: dr. ir. Xixi Lu
Daily supervisor: dr. ir. Victor Calado (ASML)
Second Examiner: dr. ing. Georg Krempl

PUBLIC VERSION

20-Feb-2025

Abstract

With the continuous growth of technology (e.g. artificial intelligence, autonomous vehicles) the demand for more advanced computer chips keeps increasing. This boils down to a need for more transistors on a chip while maintaining the chip size constant. One solution that is being pursued by chip manufacturers is to shrink the transistors further by reducing the lithography. Within chip manufacturing, a 3D structure is created by stacking layers containing structures made by means of lithography. A resolution reduction can be achieved by increasing the physical size of the lens, i.e. the numerical aperture (NA), which is currently being developed for ASML's future high-NA extreme ultraviolet (EUV) system. A physical consequence of a higher NA is that the depth of focus (DOF) increases, which leads to focus margins becoming more critical. The error between the intended focus during exposure of wafers and actual focus applied is called the focus error, which should be kept as small as possible. This focus error becomes more significant in high-NA systems. In this research we aim to find a machine learning based method that quantifies these focus errors by using scanning electron microscope (SEM) derived data. The models use as features the measured critical dimension (CD) and placement errors in the X and Y-direction (PE_x and PE_y) for a set of patterns leveraging the optical proximity effect. We further investigate how different aggregation methods influence the performance for focus prediction and how a new feature concept leveraging the optical proximity effect might be beneficial for focus prediction. Results on R^2, root-mean-squared error (RMSE) and 3σ have shown that in general feedforward neural network is the best algorithm compared to the alternative (regularized) linear regression algorithms.

Contents

Chapter 1

Introduction

In this day and age it is hard to imagine a world without computer chips. With the continuous growth of technology (e.g. artificial intelligence, autonomous vehicles) the demand for more advanced computer chips increases. In chip designing lithography is an essential aspect we cannot miss.

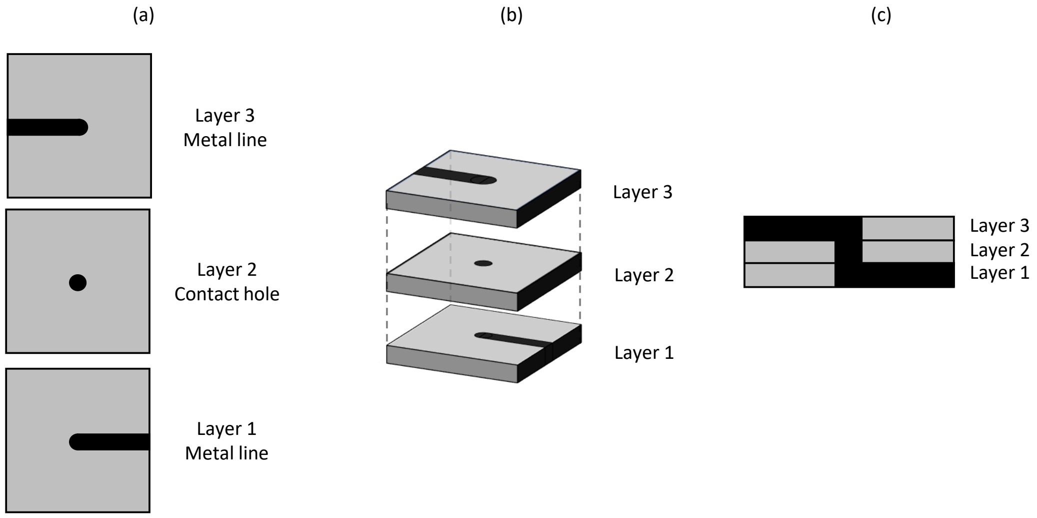


Figure 1.1: Lithography consists of printing layers of metal lines and contact holes (a) to create a 3D structure (b). The resulting cross-section is shown in (c).

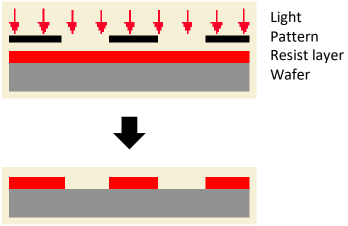


Figure 1.2: Pattern is printed in the resist layer (Adapted from [1])

According to Moore's Law the amount of transistors in a computer chip doubles every two years [2]. The need for more transistors on a chip requires more advanced technology while maintaining the chip size constant. As the amount of transistors increases, the circuit designs within the chips become more dense, thus the error margins are getting smaller. With lithography a 3D structure is printed by printing layers from the bottom up and connecting these individually as shown in Figure 1.1. Printing is done on a wafer, i.e. a silicon disk, which consists of multiple chips once cut up. To create these chips, a pattern of structures is printed in the resist layer on the wafer by exposing it with light as shown in Figure 1.2. In order to achieve a higher chip density, the print resolution has to go down. The print resolution, i.e. the size of the smallest structure that can be printed, is defined by Formula 1.1. Here λ is the wavelength of the light used for exposure and NA the numerical aperture, which is related to the size of the lens. The latest technology has reduced the resolution by going from deep ultraviolet lighting (DUV) with a wavelength of 193 nm to extreme ultraviolet lighting (EUV) with a wavelength of 13 nm. The next step in reducing the resolution is by increasing the NA.

$$\text{Resolution} \sim \frac{\lambda}{NA} \tag{1.1}$$

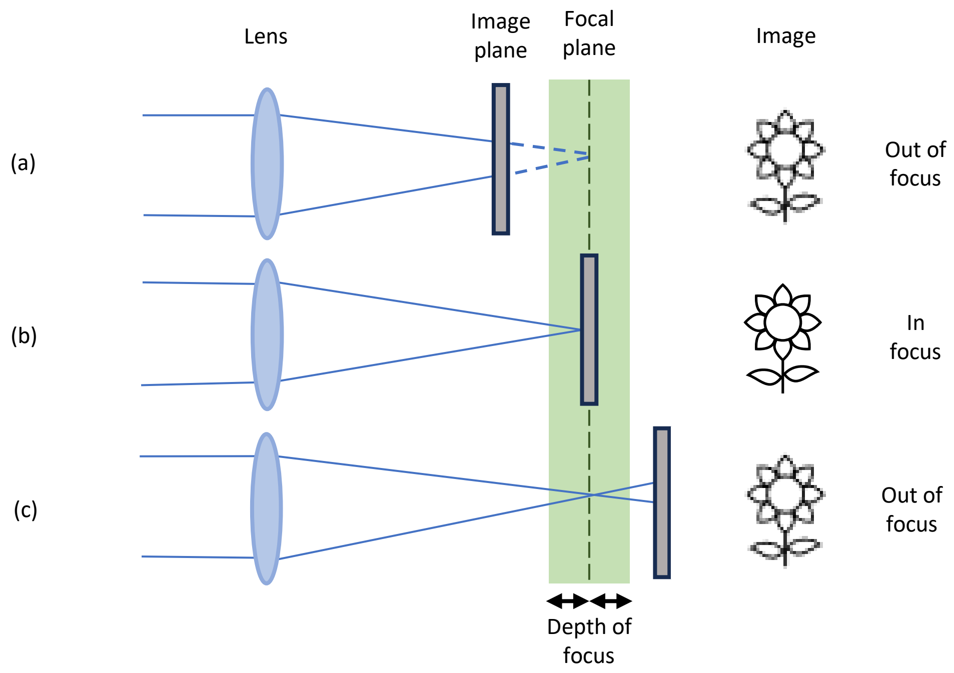


Figure 1.3: A camera gives a clear image when in focus. When out of focus the image gets blurry.

To get optimal yield in the production process and optimal performance in terms of functionality, we need to reduce the printing errors as much as possible. The ultraviolet light is bundled towards one focal point, which is called the focus. The importance of focus will be explained through an analogy: In Figure 1.3 three situations are drawn at which an image is taken of a flower. Light reflected from the object hits the camera lens and gets bundled onto the image plane. In Figure 1.3(b) the image plane and focal plane are the same, so all the light gets bundled towards one focal point. This means that the camera is in focus and results in a sharp image. In Figure 1.3(a) the light bundle hits the image plane before it reaches the focal plane. This means that the camera is out of focus and results in a blurred image. In Figure 1.3(c) the light gets bundled but spreads out again before it hits the image plane. Again, the camera is out of focus with a blurry image as result. The importance of focus for clear camera imaging is similar to that of printing a chip. The light source for printing can be in focus at its optimal setting and out of focus at a suboptimal setting. To reduce the printing errors on a chip the optimal focus needs to be set.

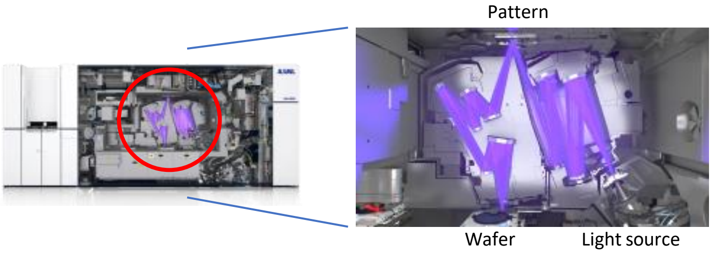


Figure 1.4: Internal components of an EUV machine [3]

In Figure 1.4 an EUV machine is shown together with its internal components. Instead of a lens system similar to a camera an EUV machine uses mirrors to bundle the ultraviolet light. The light indicated in purple gets reflected on a series of mirrors before it hits the mask containing the chip pattern. Next the light with the pattern is reflected onto another series of mirrors before it hits the wafer. This light prints the pattern and has to be in focus when it hits the wafer.

The depth of focus (DOF) specifies the focus error in which the image is still considered of good quality. Returning to the camera analogy, in Figure 1.3 the green area indicates the optimal focus ± DOF range. When the image plane falls in this range clear images are returned, otherwise the images will be blurry. So with a relatively small DOF focus control becomes more important as images are prone to become blurry more easily.

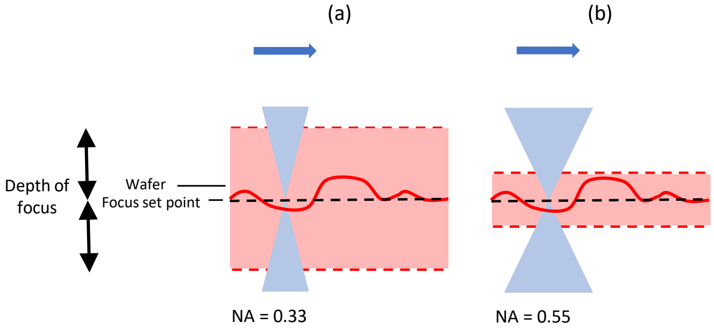


Figure 1.5: Illustration of focus significance. Wafer topology becomes more significant for high-NA (b) than low-NA (a) due to the smaller depth of focus

To print the chip a focus is set in the machine settings. However, a wafer is not perfectly flat, so there is a difference between the focus in the machine settings and the focus that effectively is applied on the wafer. The need for accurate focus measurement becomes evident in Figure 1.5. As mentioned before, the next step in improving the print resolution is by increasing the NA. In Figure 1.5(a) the error range in which the wafer is in focus, i.e. the DOF, is relatively big for a low-NA system. In Figure 1.5(b) the high-NA system comes with a relatively small depth of focus. Similarly to the camera analogy, the high-NA system needs better focus control due to the DOF. Since a wafer is not perfectly flat, inconsistencies in the wafer are proportionally more significant for high-NA systems than low-NA systems. This means that focus becomes more important when evolving from low-NA to high-NA system.

Since development of more advanced computer chip requires proceeding towards high-NA systems, it might be important to be able to measure the applied focus on the wafer as these might differ from the intended focus. There could be a need to post-correct these focus errors. Until today focus errors can be optically measured on dedicated metrology patterns. In this research we explore the possibility of measuring focus by using scanning electron microscope (SEM) data. This might be preferred over the optical approach as the focus measurement is done on real product patterns with small structures, whereas the optical measurement rely on dummy patterns. In previous researches focus measurement methods have been proposed that use regression to infer the effective focus on the wafer and the dose, which is the amount of energy used per surface area [4][5]. Therefore, in this research we aim to find the best algorithm for a prediction based model that functions as a measurement for focus. It is based on previous work from Calado et al., which will be explained in more detail in Section 2.2.1. That work implements a prediction based model for focus and is made by means of linear regression [5].

This algorithm will be fitted for every chip design that goes into production as chip design will heavily influence the performance. Once a model has been created using this algorithm, the predicted dose and focus can enhance the performance of existing focus metrology systems by providing additional information or it can adjust the light source accordingly as a form of feedback correction. Although the problem case is not existing today, the findings of this research might be beneficial to achieve sufficient low focus measures for critical layers for high-NA systems in the future.

This report consists of eight chapters. In Chapter 1 the reasoning behind this research will be explained together with the research question and short introduction to the company at which the research has taken place.

Chapter 2 describes the background knowledge related to chip manufacturing, prior work and machine learning algorithms that are suitable for the research objective. Chapter 3 contains the essentials of chip manufacturing that are key in this research. Chapter 4 explains the data, setup and research method. In Chapter 5 we introduce the concept of a new feature by discussing the drawbacks of prior work. Chapter 6 describes the results of different experiments on focus prediction models, the newly defined feature from Chapter 5 and ends with an analysis. In Chapter 7 a discussion takes place on our findings and on further improvement of the research. Finally, in Chapter 8 we draw our conclusion and summarize the research.

1.1 Research Question

Previously the growing need for better focus control for more advanced lithography machines has been identified. The scope of this research is to find a machine learning model for predicting focus by using SEM derived data on contact holes. This brings us to the main research question:

Main research question

Which machine learning algorithm performs best in a predicting model for focus measuring on SEM derived data?

The primary assumption that is made is that input features heavily depend on the chip design. Therefore our objective is to find the best algorithm or network architecture for a selected set of features. In order to answer the main research question, this research will be split up in subresearch questions:

RQ1 *What available data could be turned into a useful feature for a candidate model?*:
For this subresearch question we need to select which data has a valuable meaning within in our domain. This domain knowledge will be obtained through a background study. Questions that arise from this are:

(a) *Should the available data be adapted in order for it to be used by a candidate model?*:
This involves verifying whether the data can be used as it is or whether a preprocessing step needs to be applied.

(b) *What new features can we define by means of the available data?*:
From the background study we aim to understand the relation between the new input feature and focus as output. The need for

a new feature becomes evident from the drawbacks of prior work, which will be discussed in Section 5.1

RQ2 *What candidate machine learning techniques are there for the given data and our problem?*:
Focus measurement should be on a continuous scale, so a list of candidate algorithms and architectures is made that are suitable for a regression problem.

RQ3 *How does our model perform against other algorithms?*:
A linear regression model will be used as baseline model. Other models following different architectures will be compared against the baseline model by looking at the R^2, root-mean-squared error and the 3σ measure.

The second assumption that is made is that findings on the given data will hold on real applications, even though the data is from a wafer developed for research purposes. This research is based on the work of Calado et al. [5] and aims to extend it by proposing different algorithms and a new feature to improve performance. Furthermore, different input granularity levels for the model input instances are investigated by aggregating different amount of data.

1.2 Method

In this research there will be two data sets used: the first one for model training and to determine the best algorithm, the second one for the generalizability of the trained models. These data sets are from a focus energy matrix (FEM) wafer, which are developed for research purposes only and provided by ASML. From SEM images there is data of roughly 20M contact holes, i.e. each data set has upto 20M instances.

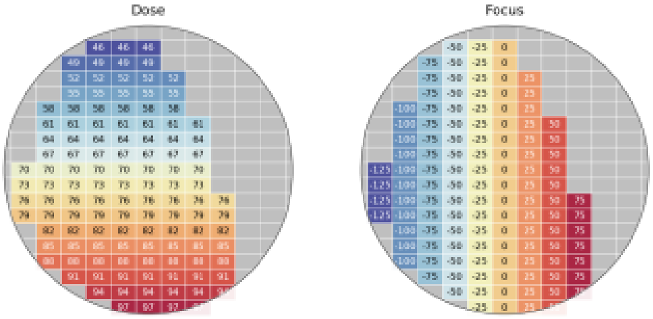


Figure 1.6: Focus energy matrix wafer. Each field has a specific focus and dose value following a matrix structure.

To answer RQ1 subresearch questions RQ1.a and RQ1.b need to be answered, which make use of the same background study. The aim is to find physical phenomena and quantities that are related to focus and dose. This is done by skimming through related work, literature regarding lithography, and through talking with domain experts. These findings will tell which available data can be considered as useful features. In RQ1.a the translation from data to feature is considered. From the background study findings it is determined whether the given data is usable as is, a preprocessing step is needed, or if data needs to be restructured (e.g. a x and y coordinate should be restructured into a location structure (x, y)). To answer RQ1.b understanding the relation between a quantity or physical phenomenon and the focus and dose is needed. It might become possible to derive quantities or physical phenomena used in other researches from the given data.

A second background study is performed to answer RQ2. In this study algorithms and neural architectures are found to solve a regression problem. These findings will be based on a Google Scholar query on existing techniques from the domain and other sources on general theory on algorithms tackling regression problems.

Then a new feature is implemented that explains the physical phenomenon that followed from the domain knowledge gained in RQ1.b. The physical phenomenon that will be described by a new feature is the proximity effect, which will be explained in Chapter 5. Placement errors in contact holes are caused by the proximity effect following the effective focus and dose on wafer level. Hence the strength of the proximity effect is modeled into a new feature by empirically finding out the best relationship between placement error and the proximity effect. The concept of the new feature will be thoroughly explained in Chapter 5 together with the results of this subresearch in section 6.3.

The last part is about RQ3, which involves testing out the different algorithms with different inputs against each other. A baseline model is created by means on linear regression. The performance measures that are being used are R^2 to indicate a sufficient fit, RMSE for quantifying prediction error on values relevant to the usecase and 3σ for focus uniformity.

1.3 Internship host - ASML

This research has been developed through an internship at Advanced Semiconductor Materials Lithography, ASML. ASML is known for their lithography systems that are used for chip manufacturing, in particular their state-of-the-art EUV technology [6].

Chapter 2

Background

2.1 Chip manufacturing

2.1.1 Optics

In lithography the numerical aperture (NA) and the depth of focus (DOF) are of high significance. These determine the minimum size that structures on a chip can have.

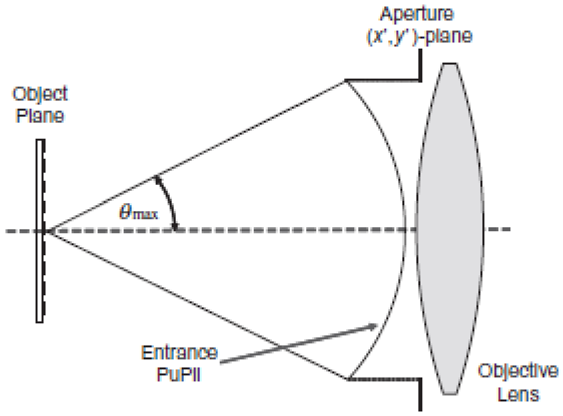


Figure 2.1: The maximum angle of diffraction is determined by the aperture size [1]

For exposure we want the lens to be as big as possible to collect light. The aperture defines the area of the lens that is able to catch diffracted light. Any light that falls outside the aperture will not hit the target on the object plane. From Figure 2.1 we notice that by increasing the aperture size, the maximum angle of diffraction θ_{max} is increased. In Equation 2.1 we define the size of the aperture as the numerical aperture NA based on the refractive index n of the medium between the mask and the lens, and θ_{max}. In this formula we see that an increase in θ_{max} results in a greater NA.

$$NA = n \sin \theta_{max} \tag{2.1}$$

In Equation 2.2 the Raleigh criterion formula is shown for calculating the minimum structure size called the critical dimension (CD). Here the CD is in nanometers, a scaling factor k_1 depending on the manufacturing process, the wavelength of exposure λ in nanometers and the NA. In Equation 2.3 the DOF depends on a process dependent scaling factor k_2, the wavelength of exposure λ and the NA.

$$CD = k_1 \frac{\lambda}{\text{NA}} \tag{2.2}$$

$$DOF = k_2 \frac{\lambda}{\text{NA}^2} \tag{2.3}$$

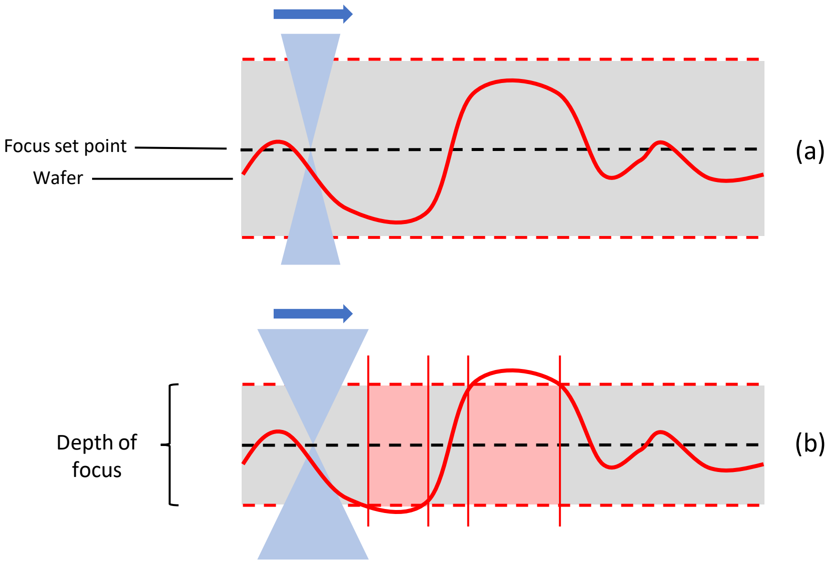


Figure 2.2: Low-NA systems with a high DOF (a) are less prone to focus errors than high-NA systems with low DOF (b). Focus errors are indicated by red areas

From Equation 2.2 can be observed that for constant exposure wavelengths an increase in NA is needed in order to print finer structures. The result of this is that the DOF will decrease. A wafer can be represented as a landscape of mountains and valleys as shown in Figure 2.2. When we set

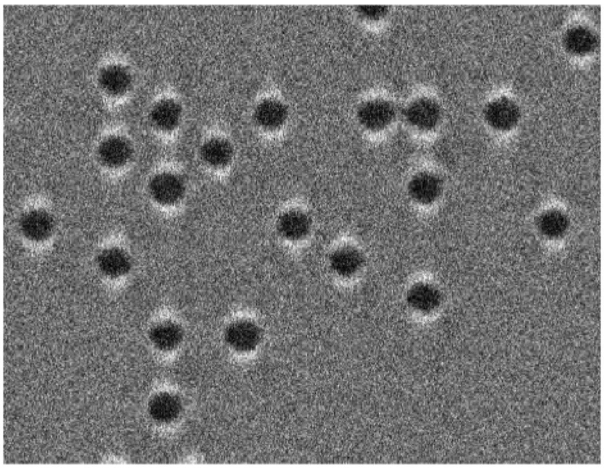


Figure 2.3: SEM image of contact holes [5]

the focus, the scanner (i.e. the light source) will be considered in focus as long it is within the boundaries defined by the DOF. Because of the wafer topography the scanner gets out of focus, resulting in printing errors. The wafer topography is responsible for the difference between the set and the real focus used on the wafer.

2.1.2 Scanning electron microscope

A scanning electron microscope (SEM) is a device that performs microscopy by emitting electrons [7]. There three key differences between optical metrology systems and scanning electron microscopy. First, a SEM uses electrons for measuring, while optical microscopes use visible light. Second, optical microscopes typically achieve a magnification upto 400-1000 times the original size of the observed object, whereas SEM achieves magnification upto 300.000 times. This allows SEM to observe object with 200 nm resolution compared to 1 nm for SEM [8]. Third, with optical microscopy an object can be observed in their original colors, while SEM returns a highly-detailed gray-scale image of the object as shown if Figure 2.3. The SEM used in this research captures is able to create images of 12x12 micron, where each pixel represents 1 nm (i.e. images of 12Kx12K pixels) [9].

2.1.3 SEM vs Optical focus metrology systems

There are a few differences between SEM and optical focus metrology system, which form the driving force of this research. SEM uses electrons to measure the structures, so its operating wavelength is relatively small. On the otherhand, optical focus metrology uses visible light, so it is a relatively big operating wavelength [10]. This difference in wavelength implies that SEM can measure individual contact holes of nanometer size, whereas optical metrology systems cannot. Yet optical metrology systems are able to measurement data by creating a target of micron size. This makes optical focus metrology an 'on-target' approach. SEM does not require any dedicated structure for measuring as it can measure structures of the product directly, hence SEM is an 'on-product' measurement. The major drawback of SEM is that the electrons used in measurement damage the product, which means that it no longer can be used for further production processes. With optical focus metrology, the visible light does not damage the product, so there is no product loss.

2.2 Prior work

2.2.1 Linear regression on orthogonal grouping SEM

This research is based on the research of Calado et al. which implemented a prediction based model for focus and dose [5]. The prediction based model is a linear regression consisting of 48 input features and a bias. The 48 input features are based on the CD and PE for the X and Y direction for 16 patterns as shown in Figure 2.4(a).

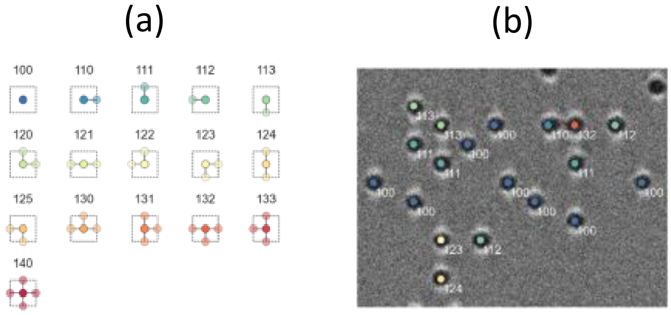


Figure 2.4: Classification scheme used by Calado et al. [5] (a) Orthogonal grouping scheme (b) SEM image with pattern label per contact hole

The patterns indicate the orientation of neighboring contact holes in close proximity. The purpose of the patterns is to describe the orientation of the optical proximity effect, which will be explained more thoroughly in section 3.4, and thus the systematic behaviour for CD and PE. The area in close proximity around a contact hole (i.e. the ambit) is divided into four quadrants: left, right, top and bottom. Depending of the presence of a neighboring contact hole in a quadrant, we can assign a label to a contact hole for 16 different patterns. In Figure 2.4(b) a SEM image is shown with a pattern label for each contact hole. When a neighboring contact hole is present in one of the quadrants, it is expected that the proximity effect will be caused by that quadrant. For each focus and dose setting the proximity effect will be different, so changes in CD and PE are focus and dose dependent. The neighboring contact holes are found by means of the K-nearest neighbor (KNN) algorithm.

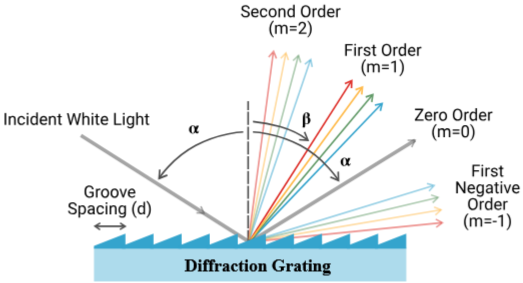


Figure 2.5: Diffraction orders [11]

2.2.2 Diffraction Based Focus metrology

Diffraction based focus metrology is a form of optical metrology based on diffraction. When white light is projected onto a grating it gets separated into a zero order and negative and positive higher order diffractions as shown in Figure 2.5. White light contains all colors of visible light, so the different order diffraction contain information about the intensities of different wavelengths.

For different shapes of grating the intensity of the diffraction orders change. Since the shapes are the result of focus the diffraction order intensities can be used to infer the focus. The inference curve of focus that has been used on the wafer is shown in Figure 2.6[12].

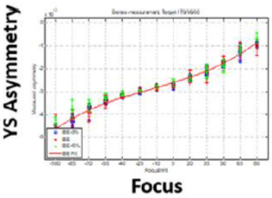


Figure 2.6: Inference curve for focus based on asymmetry in negative and positive diffraction order [12]

2.3 Machine learning algorithms

This research requires machine learning algorithms that are suitable for regression in a prediction-based model. The algorithm should be able to predict focus on a continuous scale, but also not be piece-wise constant.

2.3.1 Linear Regression with regularization

Linear regression aims to find a fit on the given data that reduces the error to a minimum. A common approach is to use ordinary least squares (OLS) to find the best weights of the model. In 2.4 the OLS formula is shown, which is slightly adapted from Bishop [13].

$$E_D(w) = \frac{1}{2}\sum_{n=1}^{N}\{y_n - \mathbf{w}^T X\}^2 \tag{2.4}$$

With OLS there is still the risk of overfitting. By means of restrictions on the weights overfitting can be controlled by minimizing a formula of the form in eq. 2.5. Here $E_D(w)$ represents the data-dependent error, *lambda* the regularization coefficient and $E_W(w)$ the regularization term that forms the restriction on the weights. Once eq. 2.5 is combined with eq. 2.6 for E_W we get eq. 2.7 that should be minimized.

$$E_D(w) + \lambda E_W(w) \tag{2.5}$$

$$E_W(w) = \frac{1}{2}\sum_{j=1}^{M}|w_j|^q \tag{2.6}$$

$$\frac{1}{2}\sum_{n=1}^{N}\{y_n - \mathbf{w}^T X\}^2 + \frac{\lambda}{2}\sum_{j=1}^{M}|w_j|^q \tag{2.7}$$

In 2.6 we see that the type of regularization can be decided by the value of q. When we set $q = 1$ we get the least absolute shrinkage and selection operator, in short Lasso regularization or L1 regularization [14]. $q = 1$ implies that we take the Manhattan distance on our weight vector. When we set $q = 2$ we get Ridge regression or L2 regularization, which means taking the Euclidean distance on our weight vector.

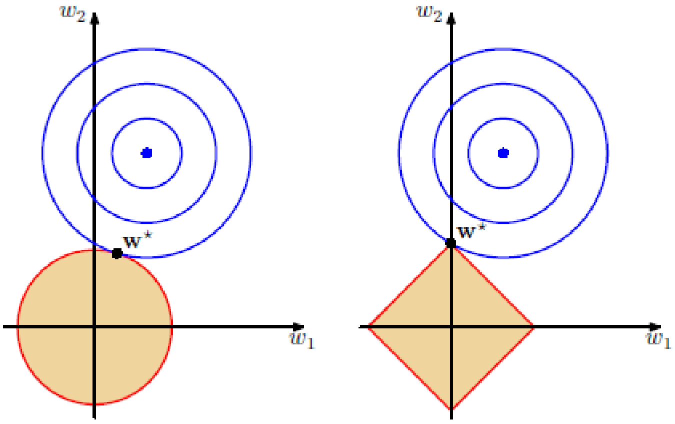


Figure 2.7: Difference between L2 (Ridge) regularization left and L1 (Lasso) on the right [13]

$$\sum_{j=1}^{M} |w_j|^q \leq \eta \tag{2.8}$$

The difference between Lasso and Ridge regression lies in the solution of minimizing Equation 2.7. This comes down to the most optimal OLS solution for Equation 2.4 while also satisfying the constraint in Equation 2.8 for a chose η. In Figure 2.7 we see in blue the contour lines in the parameter space, where the blue dot represent the optimal solution for w_1 and w_2 through unregularized OLS. The orange circle and square represent the parameter space that satisfy Equation 2.8, so the regularized OLS solution fall in these regions. For Ridge regression we see that the optimal regularized OLS solution $w^\star$ gives $w_1 \neq 0$ and $w_2 \neq 0$. For the Lasso regression we have an optimal regularized OLS solution $w^\star$ with $w_1 = 0$ and $w_2 \neq 0$. Lasso is more prone to sparse solutions than Ridge regression for sufficiently large λ, therefore one might prefer Ridge regression over Lasso regression.

$$(1-\alpha)\sum_{j=1}^{M} |w_j| + \alpha \sum_{j=1}^{M} |w_j|^2 \leq \eta \tag{2.9}$$

It is also possible to combine Lasso and Ridge regression together to get the best balance of the methods [15]. This means we have two regularization parameters λ_1 and λ_2 that give an OLS solution that has to satisfy the constraint if eq. 2.9 where $\alpha = \frac{\lambda_2}{\lambda_1+\lambda_2}$. When we set $\alpha = 0$ elasticnet is similar to Lasso regression, similarly for $\alpha = 1$ we have elasticnet being similar to Ridge regression.

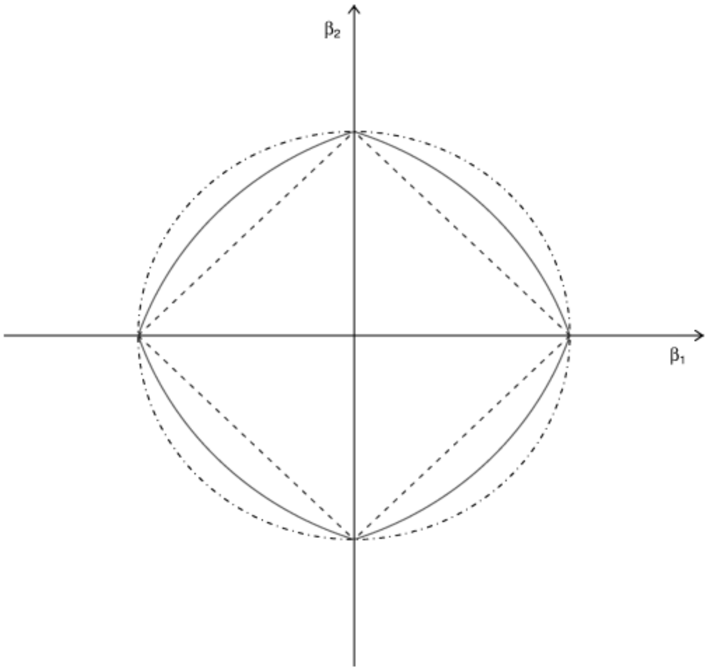


Figure 2.8: Elasticnet (solid line) solution space drawn as a combination of L1 (dashed line) and L2 (dotted line) regularization [15]

Chapter 3

Preliminaries

In Chapter 1 we have explained focus, dose and depth of focus in detail. In addition, in this chapter we will introduce other key aspects in detail in order to understand this research.

3.1 Wafer

A wafer is a silicon disk on which chips are printed by means of lithography. It has a radius of 300mm and can be divided into many fields containing chip design a chip design as shown in Figure 3.1(a). It is possible to have multiple chips (i.e. multiple dies) in a single field as shown in Figure 3.1(b) [16].

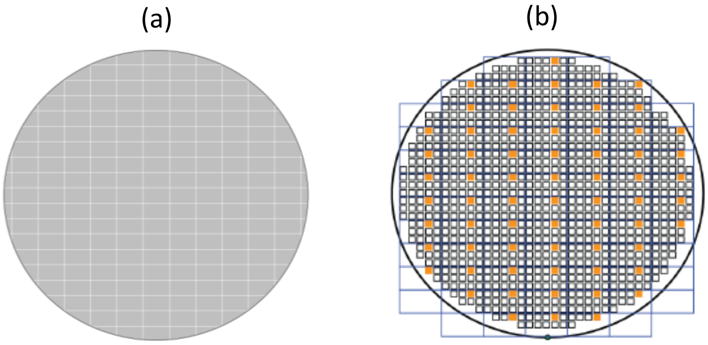


Figure 3.1: Two illustrations of a wafer. (a) Empty wafer consisting of many fields (i.e. rectangular areas). (b) Wafer within each field (blue) a 5x3 die array [16]

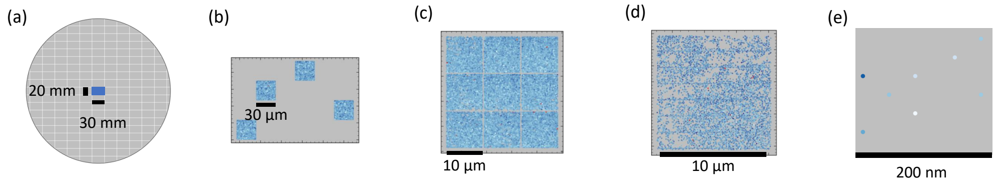


Figure 3.2: Rough dimensions of the data visually presented. (a) One field on a wafer. (b) Four pads within a field. (c) Nine SEM images within a pad. (d) One SEM image.

In Figure 3.2(a) an illustration of a field on a wafer is shown again. Within a field, pads can be defined which are collections of SEM images. In Figure 3.2(b) four pads are shown of typically 30x30 micron with in it nine SEM images (Figure 3.2(c)). Figure 3.2(d) shows a SEM image of 10x10 micron.

On each field there can be millions of contact holes, but it is unfeasible to collect data of each structure individually. SEM measuring these structures is a time consuming process. Besides, the resist layer is getting damaged by the electrons shot by the SEM and leads to product loss as already mentioned in Section 2.1.3.

3.2 Critical Dimension

The critical dimension (CD) is the diameter of a contact hole. This follows in general a quadratic relationship with focus as indicated in Figure 3.3.

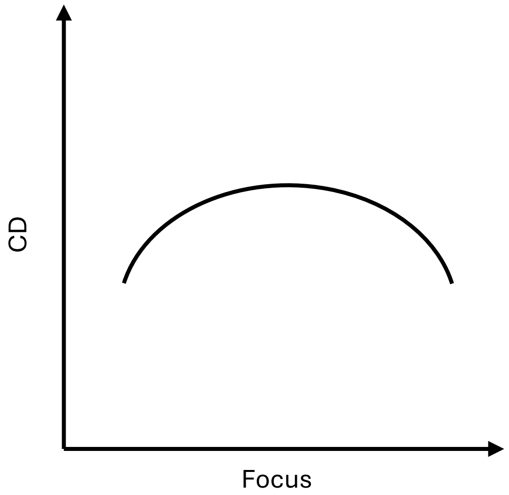


Figure 3.3: Sketch of the quadratic relationship between CD and focus.

Since contact holes change in size and shape when printed, the CD will be affected. To determine the CD of a printed contact hole an ellipse is fitted on SEM images by means of Equation 3.1. Here x and y are the points on the ellipse and a and b form half the width and half the height of the ellipse (see Figure 3.4(a)). In Figure 3.4(b) a SEM image is shown with an ellipse fit on the contact holes.

$$\frac{x^2}{a} + \frac{y^2}{b} = 1 \tag{3.1}$$

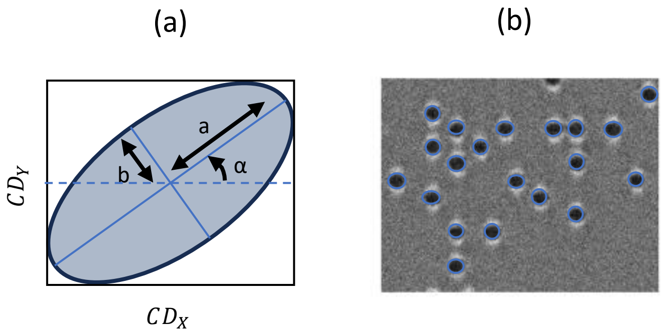


Figure 3.4: (a) Ellipse components for finding the critical dimension. (b) Illustration of ellipse fits on contact holes in a SEM image

$$CD = \sqrt{CD_{major} \cdot CD_{minor}} \tag{3.2}$$

CD_{major} and CD_{minor} are the major and minor axis of the ellipse, so that is the width and height respectively. The CD can be calculated through Equation 3.2 by using the CD_{major} and CD_{minor}. In addition, the X-component CD_X and Y-component CD_Y of the CD can be found by fitting a box around the ellipse as shown in Figure 3.4(a). The formulas used to calculate the CD_X and CD_Y are shown in Formula 3.3 and 3.4 where α is the angle of rotation of the ellipse.

$$CD_X = \sqrt{(CD_{major} \cdot \cos(\alpha))^2 + (CD_{minor} \cdot \sin(\alpha))^2} \tag{3.3}$$

$$CD_Y = \sqrt{(CD_{major} \cdot \sin(\alpha))^2 + (CD_{minor} \cdot \cos(\alpha))^2} \tag{3.4}$$

3.3 Placement Error

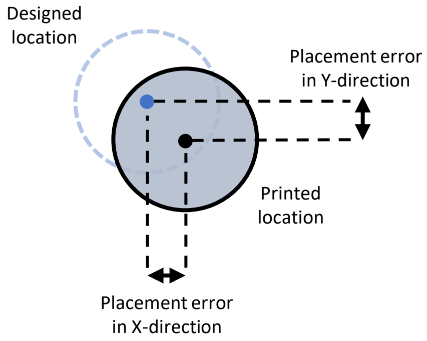

Designed location
Placement error in Y-direction
Printed location
Placement error in X-direction

Figure 3.5: Placement error in the X- and Y-direction

In Figure 3.5 the placement error (PE) of a contact hole is shown. The placement error is the difference between the designed location and the printed location of a structure based on the centroid. This displacement is defined through components in the X- and Y-direction, i.e. PE_X and PE_Y respectively. PE follows a linear relationship with focus, which is shown in Figure 3.6.

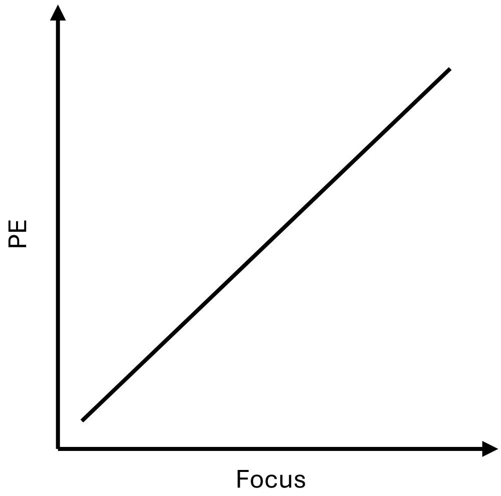


Figure 3.6: Sketch of the linear relationship between placement error and focus.

3.4 Optical Proximity Effect

Imaging relies on the assumption that light is fully propagated through a lens or mirror. However, light will never be perfectly propagated due to the scattering of light.

Following the scattering of light, the effective dose on the wafer gets changed. This is especially a problem in the close proximities of the printed structures. During exposure, the light used for one structure will also be used for neighboring structures in close proximity. This imaging effect is called the (optical) proximity effect and cannot be avoided.

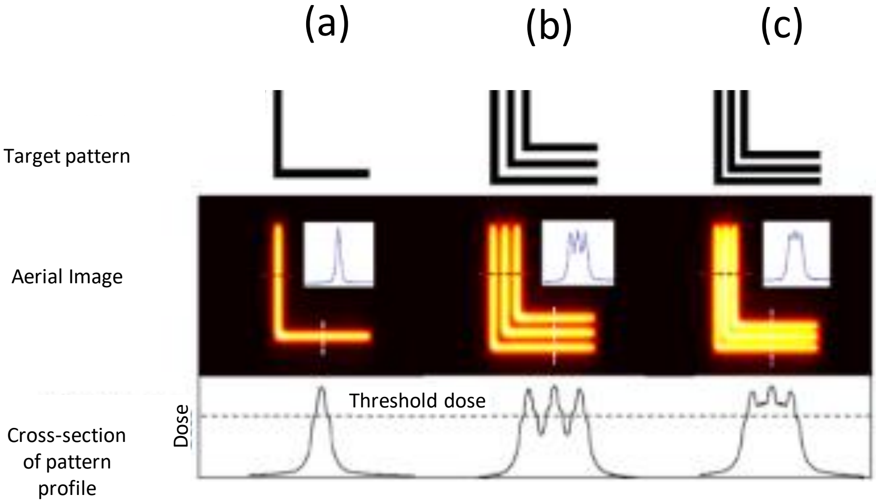


Figure 3.7: Proximity effect. Adapted from [17]

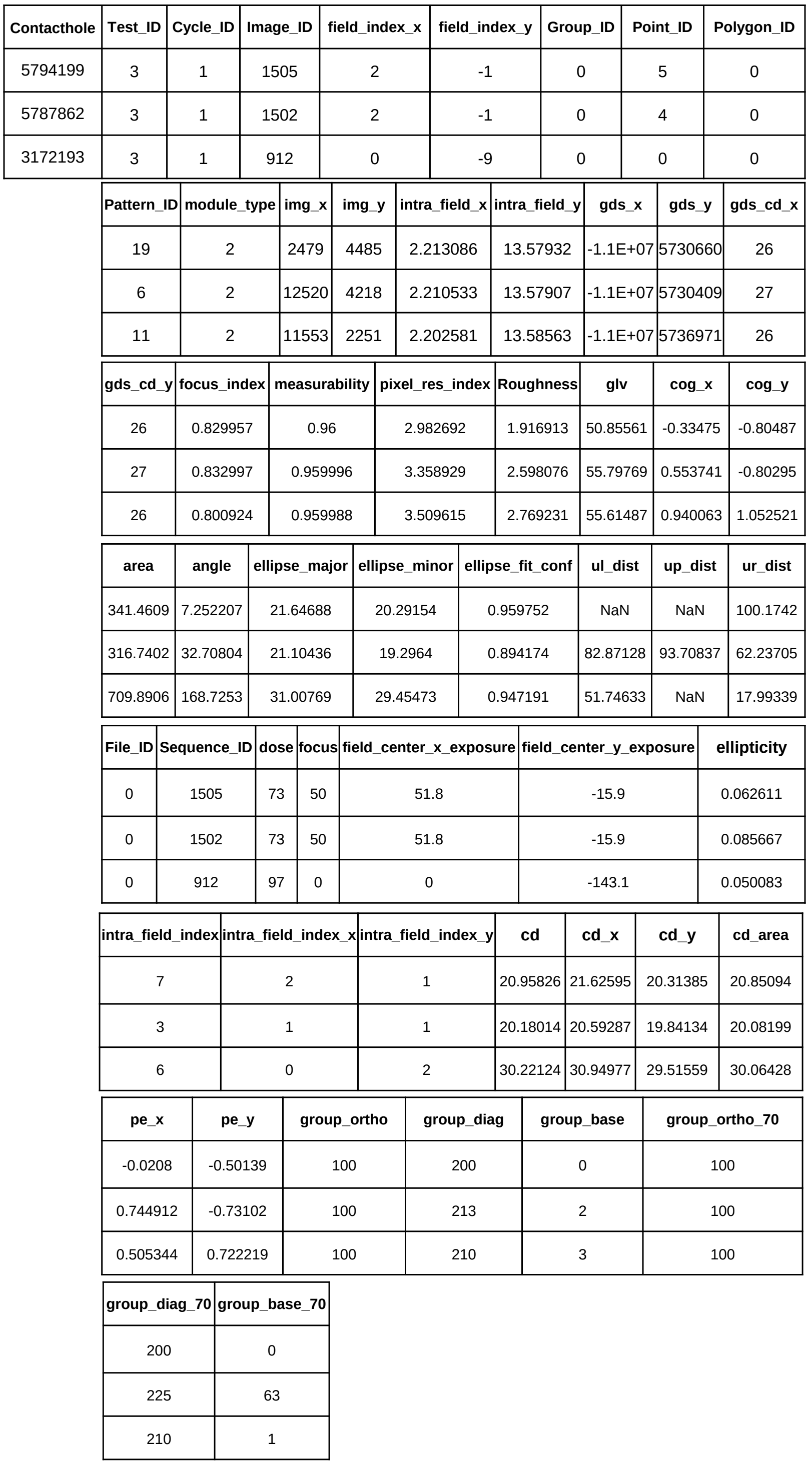
In Figure 3.7 we see the result of the proximity effect when exposing a target pattern consisting of line segments. In Figure 3.7(a) we see a target pattern with a single L-shape with an expected aerial image and one peak above the threshold dose. In Figure 3.7(b) we try to print three L-shaped patterns in close proximity of each other. In the cross-section of the pattern profile we expect to see three distinct dose curves, similar as in Figure 3.7(a). However, due to the close proximity of the L-shapes, the spikes curves are starting to overlap. This phenomenon is the proximity effect. Note that there are still three distinguishable spikes present above the threshold dose, making it a printed pattern consisting of three separate L-shapes. Figure 3.7(c) still has the same three L-shaped patterns as target, but now the patterns are closer to each other. In the cross-section of the pattern profile we see that the proximity effect is so strong by the decreased space between the patterns that the three expected dose curves merge such that only one spike above the threshold dose arises. This means that the printed image also starts to overlap as can be seen in the aerial image. From the aerial images and cross-section profile we observe that target patterns get distorted in the direction of neighboring patterns during printing. This distortion behavior also occurs with contact holes. Besides a shift in location, the shape of contact holes will change, which will affect the CD. The shift and distortion of the contact hole is a function of focus and dose.

Chapter 4

Experimental methods

4.1 Data description

The first data set consists of information about 19M individual contact holes from the same FEM wafer as used by Calado et al [5]. There are 119 fields on this wafer, where each exposure field is distinguishable by their combined dose and focus value. The dose is the intensity of the ultraviolet light during exposure, which is defined by an amount of energy per surface area in mJ/cm^2. For each focus and dose pair a subset consists of 150K contact holes, which can be subdivided into four pads and within each pad nine SEM images similar to Figure 3.2. A pad and SEM images consist of 42K and 5K contact holes respectively (see Table 4.1).



Contacthole	Test_ID	Cycle_ID	Image_ID	field_index_x	field_index_y	Group_ID	Point_ID	Polygon_ID
5794199	3	1	1505	2	-1	0	5	0
5787862	3	1	1502	2	-1	0	4	0
3172193	3	1	912	0	-9	0	0	0

Pattern_ID	module_type	img_x	img_y	intra_field_x	intra_field_y	gds_x	gds_y	gds_cd_x
19	2	2479	4485	2.213086	13.57932	-1.1E+07	5730660	26
6	2	12520	4218	2.210533	13.57907	-1.1E+07	5730409	27
11	2	11553	2251	2.202581	13.58563	-1.1E+07	5736971	26

gds_cd_y	focus_index	measurability	pixel_res_index	Roughness	glv	cog_x	cog_y
26	0.829957	0.96	2.982692	1.916913	50.85561	-0.33475	-0.80487
27	0.832997	0.959996	3.358929	2.598076	55.79769	0.553741	-0.80295
26	0.800924	0.959988	3.509615	2.769231	55.61487	0.940063	1.052521

area	angle	ellipse_major	ellipse_minor	ellipse_fit_conf	ul_dist	up_dist	ur_dist
341.4609	7.252207	21.64688	20.29154	0.959752	NaN	NaN	100.1742
316.7402	32.70804	21.10436	19.2964	0.894174	82.87128	93.70837	62.23705
709.8906	168.7253	31.00769	29.45473	0.947191	51.74633	NaN	17.99339

File_ID	Sequence_ID	dose	focus	field_center_x_exposure	field_center_y_exposure	ellipticity
0	1505	73	50	51.8	-15.9	0.062611
0	1502	73	50	51.8	-15.9	0.085667
0	912	97	0	0	-143.1	0.050083

intra_field_index	intra_field_index_x	intra_field_index_y	cd	cd_x	cd_y	cd_area
7	2	1	20.95826	21.62595	20.31385	20.85094
3	1	1	20.18014	20.59287	19.84134	20.08199
6	0	2	30.22124	30.94977	29.51559	30.06428

pe_x	pe_y	group_ortho	group_diag	group_base	group_ortho_70
-0.0208	-0.50139	100	200	0	100
0.744912	-0.73102	100	213	2	100
0.505344	0.722219	100	210	3	100

group_diag_70	group_base_70
200	0
225	63
210	1

Figure 4.1: Overview of all the available data for three contact tholes.

Examples of the available data are shown in Figure 4.1 for three contact holes. From this data there are some variables that could be of interest. First, gds_x and gds_y describe the location (x, y) of the contact holes in design. The Sequence_ID is the identifier of one single SEM image. Dose and focus are key variables required in this research as they define the FEM wafer structure. Intra_field_index_x and intra_field_index_y describe the index of the SEM image location in the X and Y-direction within a field. The combination of the intrafield indices in the X and Y-direction, describe in which pad a contact hole is located. The Cd indicates the measured CD of the contact holes. Pe_x and pe_y are the measured placement errors PE_x and PE_y in the X and Y-direction respectively. Group_ortho indicates which orthogonal grouping label the contact holes has.

The chip layout of each field on the FEM wafer are identical. The SEM images are taken at the exact locations relative to the fields, but the amount of data per field is not consistent. This is caused by three types of errors: First, the SEM makes a placement error in itself, so the location of the images are not always on the same position. Some contact holes get out of view or get sliced. Second, the SEM does not capture a contact hole properly even though it is present. Third, a contact hole is not present at all, so it is a printing error rather than a measuring error. Since the SEM images are roughly at the same position, the same contact holes in the design are directly comparable albeit with different properties. One important aspect to note is that there is a large amount of data provided (i.e. millions of data points), but the diversity is relatively low as the data comes from one wafer.

Granularity level	Number of contact holes	Instances after aggregation
SEM	5K	3744
Pad	42K	416
Field	150K	104

Table 4.1: Number of instances before and after aggregation at different levels

In our research we use a subset of the data by reducing the focus range between -75 nm and 75 nm as this is closer to values used in practice. In Figure 4.2 are all the 119 focus and dose values shown together with the 104 focus and dose values in the subset.

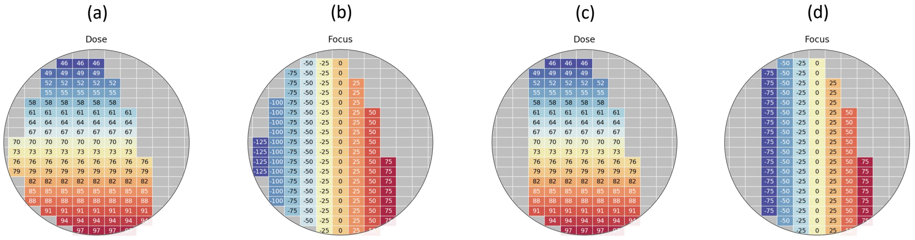


Figure 4.2: Focus and dose values before and after reducing the data to focus ranging between -75 and 75 nm. (a) All dose values (b) All focus values (c) Dose values in the subset (d) Focus values in the subset.

With different granularity levels model input instances are created through aggregation on contact holes per field, pad or SEM image. Since we want to investigate different granularity levels it is important that all groups from the classification scheme in Figure 2.4(a) are present for the different granularities. For SEM granularity level it becomes a problem, because in some SEM images in the wafer not all groups are present. In Figure 4.3 the group count on dataset 1 and dataset 2 is shown. From the figure can be observed that the groups with three or four neighbors combined are less than 0.5% present in the data, while the majority is determined by contact holes with zero, one or two neighbours. Therefore, after applying grouping the contact holes with three or four neighbors are removed from the datasets such that on all granularity levels the same groups as input features are being used.

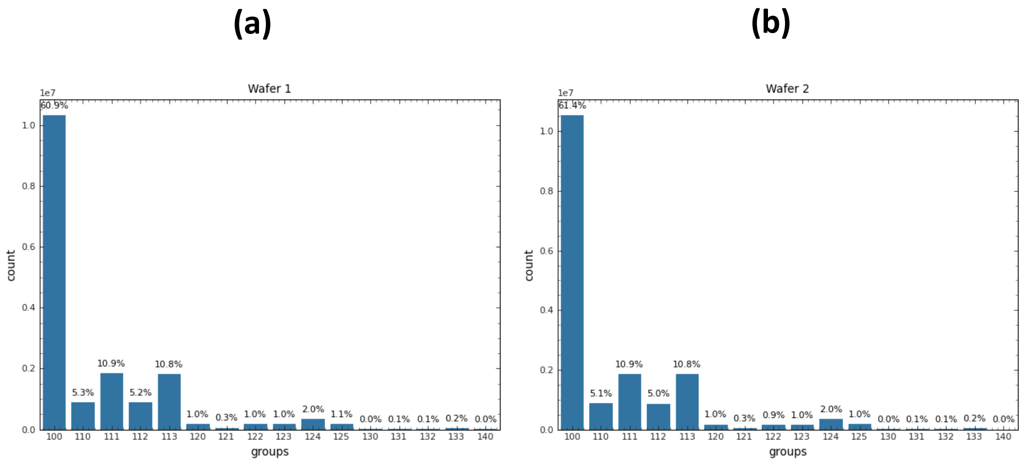
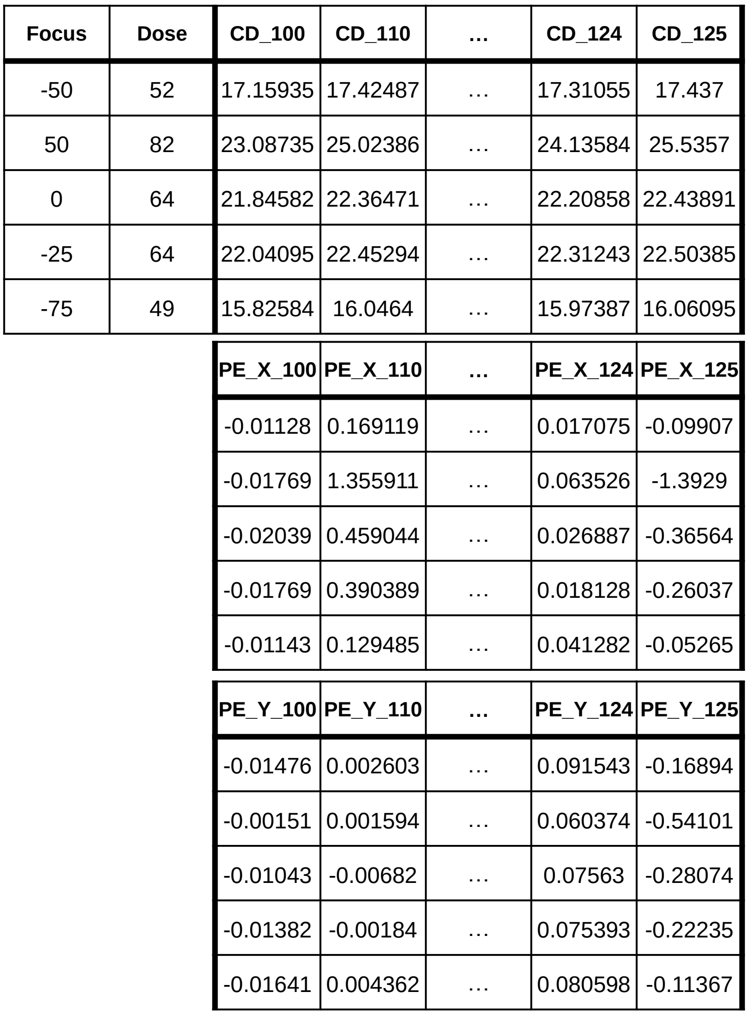


Figure 4.3: Amount of contact holes per group and their percentage of presence in the first dataset (a) and second dataset (b).

Following the grouping of contact holes with upto 2 neighbors and the different granularity levels, the input and output values of the variables of interest are shown in Table 4.2, 4.3 and 4.4 for field, pad and SEM granularity level respectively. At field granularity level, each data instance for the models can be distinguished by a combination of focus and dose as visible in Table 4.2. In Figure 4.3, each data instance at pad granularity can be distinguished by the focus, dose and Pad ID, which indicates on which one of the four pads within a field has been aggregated. At SEM granularity level the combination of focus, dose and Sequence ID makes each data instance unique identifiable as observed in Figure 4.4.



Focus	Dose	CD_100	CD_110	…	CD_124	CD_125	PE_X_100	PE_X_110	…	PE_X_124	PE_X_125	PE_Y_100	PE_Y_110	…	PE_Y_124	PE_Y_125
-50	52	17.15935	17.42487	…	17.31055	17.437	-0.01128	0.169119	…	0.017075	-0.09907	-0.01476	0.002603	…	0.091543	-0.16894
50	82	23.08735	25.02386	…	24.13584	25.5357	-0.01769	1.355911	…	0.063526	-1.3929	-0.00151	0.001594	…	0.060374	-0.54101
0	64	21.84582	22.36471	…	22.20858	22.43891	-0.02039	0.459044	…	0.026887	-0.36564	-0.01043	-0.00682	…	0.07563	-0.28074
-25	64	22.04095	22.45294	…	22.31243	22.50385	-0.01769	0.390389	…	0.018128	-0.26037	-0.01382	-0.00184	…	0.075393	-0.22235
-75	49	15.82584	16.0464	…	15.97387	16.06095	-0.01143	0.129485	…	0.041282	-0.05265	-0.01641	0.004362	…	0.080598	-0.11367

Table 4.2: Focus and dose output values with CD, PE_x and PE_y input values at field granularity level.

Focus	Dose	Pad ID	CD_100	CD_110	…	CD_124	CD_125	PE_X_100	PE_X_110	…	PE_X_124	PE_X_125	PE_Y_100	PE_Y_110	…	PE_Y_124	PE_Y_125
50	94	3	26.6724	30.00806	…	27.4821	30.23771	-0.01779	2.449761	…	0.021173	-2.33899	0.003533	-0.01149	…	0.025546	-0.5177
-25	88	0	27.4824	28.69956	…	27.70889	29.08363	0.002729	1.147498	…	0.013981	-1.30673	-0.00937	0.030846	…	0.045964	-0.32661
25	76	2	23.76163	24.84861	…	24.32439	25.1674	-0.01698	0.878557	…	0.023653	-0.91365	-0.00828	0.001605	…	0.06055	-0.40524
-75	76	3	22.44214	23.15629	…	22.63943	23.39904	-0.01575	0.564452	…	0.048066	-0.56816	-0.01898	0.003862	…	0.069607	-0.14554
0	73	2	24.46486	25.12212	…	24.7762	25.34891	-0.01433	0.632306	…	0.059776	-0.61761	-0.01007	-0.01027	…	0.073485	-0.30507

Table 4.3: Focus and dose output values with CD, PE_x and PE_y input values at pad granularity level.

Focus	Dose	Sequence ID	CD_100	CD_110	…	CD_124	CD_125
-25	64	3797	22.07756	22.41874	…	22.32362	22.40425
-50	82	2142	25.93036	26.64123	…	26.22285	26.82818
25	82	4041	25.40508	26.77849	…	25.84978	27.45844
-25	67	800	22.55568	22.98302	…	22.84598	22.92777
-25	64	819	22.06865	22.46599	…	22.20488	22.41644

PE_X_100	PE_X_110	…	PE_X_124	PE_X_125
-0.01484	0.37262	…	-0.04741	-0.13804
-0.0127	0.764909	…	0.179911	-0.78851
0.007624	1.180715	…	0.041609	-1.55084
-0.01842	0.435641	…	0.070106	-0.37567
-0.02597	0.427132	…	-0.05036	-0.23887

PE_Y_100	PE_Y_110	…	PE_Y_124	PE_Y_125
-0.01838	-0.01792	…	0.104816	-0.19915
-0.01752	0.018831	…	0.138309	-0.1279
-0.00873	0.05002	…	0.02161	-0.34797
-0.01572	-0.06545	…	0.021277	-0.32211
0.001184	0.02231	…	0.004556	-0.21891

Table 4.4: Focus and dose output values with CD, PE_x and PE_y input values at SEM granularity level.

The second dataset comes from a second FEM wafer with the same subset as shown in Figure 4.2, i.e. focus between -75 nm and 75 nm and groups with upto two neighbours. In Figure 4.4 the distribution of CD, PE_x and PE_y for the first and second dataset are given. This dataset differs from the first dataset in their mask biased CD, i.e. the CD in chip layout are the same, but not on the mask. The mask biased CD is 24.31 nm for wafer 1 and 25.245 nm for wafer 2, which results in a different distribution for CD as shown in Figure 4.4(a).

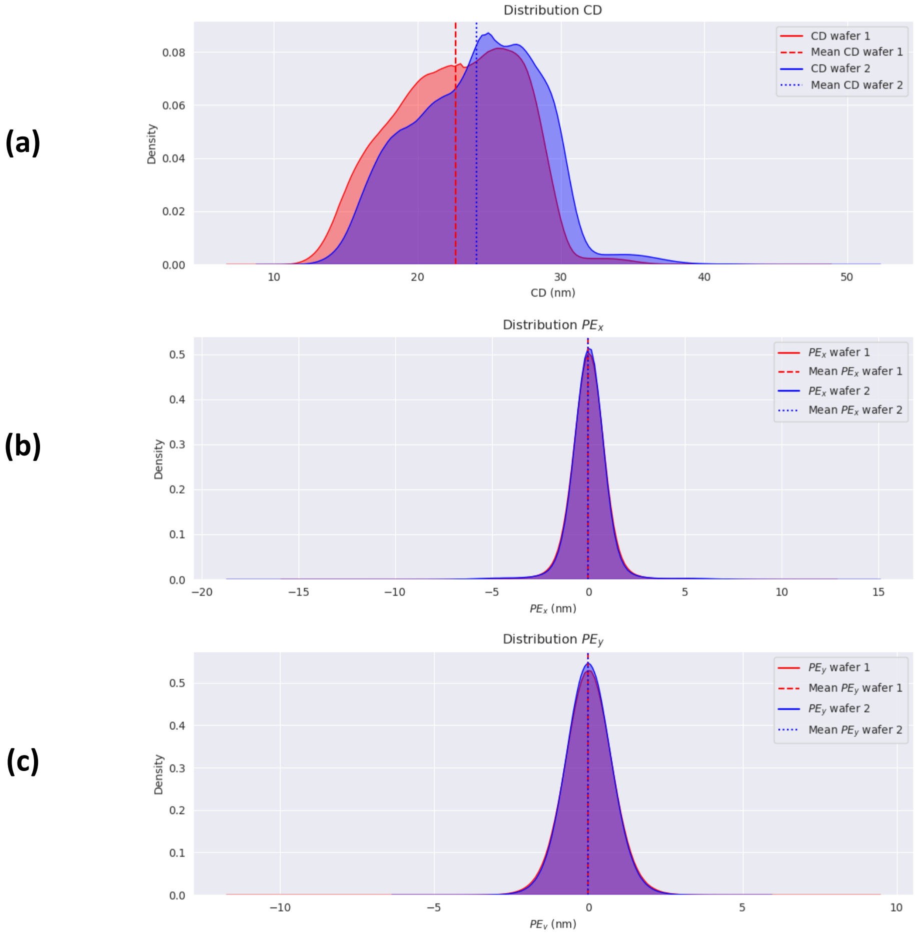


Figure 4.4: Distribution of CD (a), PE_x (b) and PE_y (c) of first and second dataset.

Since switching to different granularity levels involves aggregation on different amount of contact holes, the distribution of CD, PE_x and PE_y of the contact holes within a single input instance might change. In Figure 4.5 the distribution of CD, PE_x and PE_y within a single field, pad and SEM instance of nominal focus are shown.

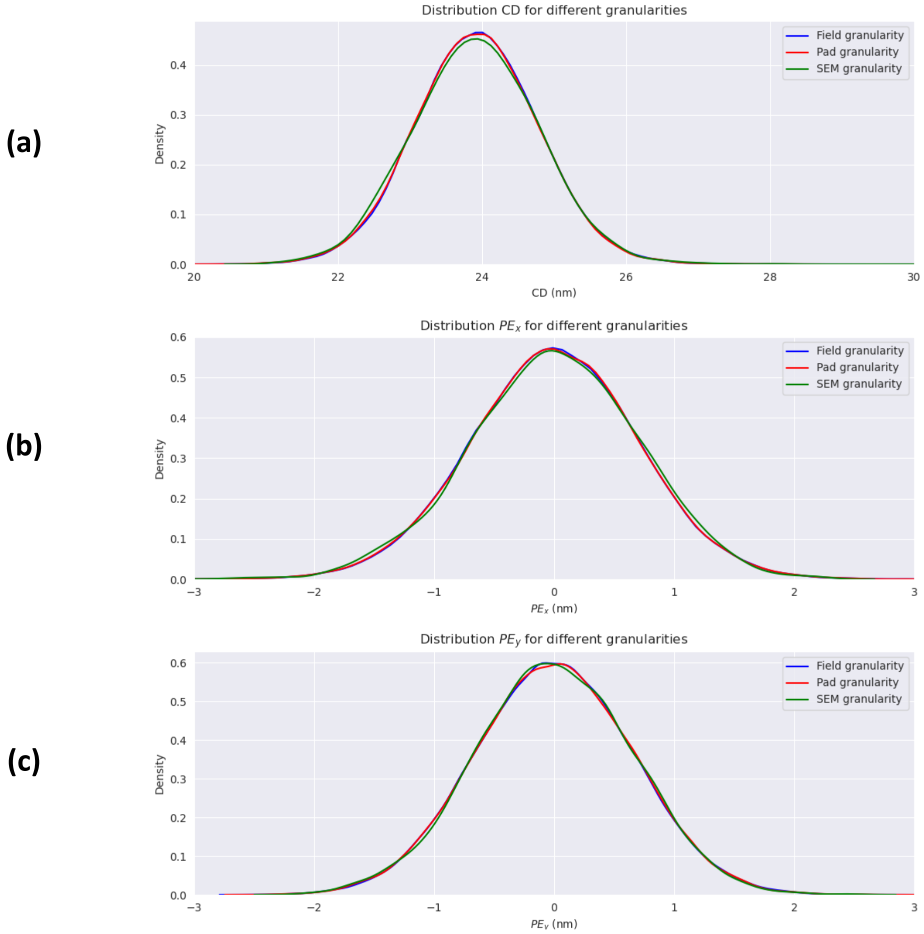


Figure 4.5: Distribution of CD (a), PE_x (b) and PE_y (c) within one field, pad and SEM instance.

From Figure 4.5 can be observed that the distribution of CD, PE_x and PE_y does not change much between field, pad and SEM instances. However, with finer granularity the number of contact holes that are aggregated reduces, which result in higher standard error of the mean. The standard error of the mean describes how precise the sample mean corresponds with the mean of the true distribution. The higher the standard error becomes, the more imprecise the input data of our model gets, so it implies that the input data becomes noisier. The standard error of the mean is defined by $SEM = \frac{\sigma}{\sqrt{N}}$, where σ is the standard deviation and N is the amount of contact holes. As shown in Table 4.1, the amount of contact holes for one input instance deviates a lot among the different the granularity levels, which could reflect in the standard error of the mean. In Table 4.5, Table 4.6 and

Table 4.7 can be seen that the standard error of the mean increases when switching to finer granularity levels due to the lower number of contact holes.

Granularity level	Number of contact holes	Mean	Standard deviation	Standard error of the mean
Field	168757	23.954	1.000	0.002
Pad	42110	23.944	0.998	0.005
SEM	4602	23.940	1.027	0.015

Table 4.5: The standard error of the mean increases for CD as the number of contact holes per granularity level reduces.

Granularity level	Number of contact holes	Mean	Standard deviation	Standard error of the mean
Field	168757	0.000	0.716	0.002
Pad	42110	0.000	0.712	0.003
SEM	4602	0.000	0.714	0.011

Table 4.6: The standard error of the mean increases for PE_x as the number of contact holes per granularity level reduces.

Granularity level	Number of contact holes	Mean	Standard deviation	Standard error of the mean
Field	168757	0.000	0.660	0.002
Pad	42110	0.000	0.661	0.003
SEM	4602	0.000	0.651	0.010

Table 4.7: The standard error of the mean increases for PE_y as the number of contact holes per granularity level reduces.

4.2 Objective

In order to conclude which algorithm is best for focus prediction and how generalizable the trained model is, we will look at multiple metrics on two different data sets. First the R^2 score to see how well the fitted model behaves against the true values. We create a set-get plot, which shows predicted values plotted against the true values as shown in Figure 4.6. Ideally all predictions follow the line $y = x$, giving $R^2 = 1$ while the slope is one. Comparing the R^2 of all models against a baseline will tell whether a fit is sufficient enough.

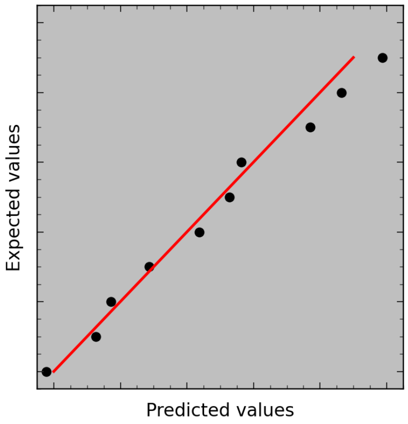


Figure 4.6: Example of a set-get plot, with the red line indicating the ideal fit $y = x$.

The second measure we will look at is the root-mean-squared error (RMSE). This measure takes at first the squared differences between the fitted outcomes and the expected outcomes. Then the mean is taken of the squared differences followed by the root. The RMSE ranges from zero to infinity. Ideally we want the RMSE value to be zero as that implies a perfect fit. This measure will be performed on nominal focus, i.e. data from a common fixed focus value and common dose range used in practice.

The final measure to take into consideration is the 3σ. The 3σ describes three times the standard deviation in our error distribution. The RMSE and 3σ are closely related to eachother as $3 \cdot RMSE \approx 3\sigma$. A 3σ close to zero for the difference in predicted and expected focus would imply that the focus on the entire wafer is close to uniformity. In practice, the expected 3σ is not zero as focus errors also rise due to other stochastics involved. At field granularity level it is known that $3\sigma \approx 10$ nm, meaning that roughly 99.7% of the focus errors are upto 10 nm from the mean error. Therefore we are looking for $3\sigma \approx 10$ nm in this research. This 10 nm will be used also at other granularity levels for comparison.

The most crucial measures for the conclusion are the RMSE and the 3σ as they are relevant to the usecase. To verify generalizability the trained models are applied on a second dataset.

4.3 Setup

We create a new feature that also makes use of the proximity effect, but in a numerical way and with less parameters to be tuned as will be explained in Chapter sec:proxy-force-theory. The proximity effect tells us that there is an inverse relationship between the strength of the effect and the distance between structures. Therefore, we empirically look for an inverse relationship based on the distance to describe the strength of the proximity effect, which we will call '*proximity force*' (PF). The proximity force feature will be defined by empirically trying out different orders of inverse distance functions between a contact hole and its local neighborhood. By finding a relationship between function results and the PE a new feature can be defined in which the feature values are the function parameters

In this research five experiments will be conducted:

(A) Finding the appropriate amount of epochs to train a neural network for focus prediction

(B) Focus prediction with machine learning models on two datasets.

(C) Defining a new feature that builds upon existing work.

(D) Focus prediction with machine learning models on two datasets while using the new defined feature from Experiment (C)

(E) Analysing the difference in results between the two datasets.

In Experiment (A) we will try to estimate an appropriate amount of epochs to train a neural network for focus predictions. This is required as a too small amount of epochs used in training might lead to underfitting. The amount of epochs will be investigated for field, pad and SEM granularity and for the 80%/20%, nominal focus and 3σ datasplits. In Experiment (B) the amount of epochs found in Experiment (A) will be used for neural network training. Together with linear regression, lasso, ridge and elasticnet regression models for focus prediction will be developed. These models will be tested on two datasets on which we will compare the R^2, RMSE and 3σ metrics with their corresponding datasplits. For this a similar approach as Calado et al. is used as explained in Section 2.2.1 [5]. In that research focus and dose were predicted with the CD, and the PE for the X and Y direction per pattern, resulting in 48 input features. In Experiment (C) a new feature will be developed based on the existing work of Calado et al. The approach of orthogonal grouping comes with drawbacks that we try to tackle by defining a new feature. In Experiment (D) focus prediction models will be developed based on Experiment (D) by adding the new feature defined in Experiment (C). Also here will the models be tested on two datasets on which we will compare the R^2, RMSE and 3σ metrics with their corresponding datasplits. In Experiment (E) we try to argue why discrepancies in performance arise between the first and second dataset.

4.4 Hardware and software

Simple machine learning experiments have been executed on a Windows based virtual desktop interface (VDI) with an Intel Xeon Gold 6146 CPU. The neural network approaches have been executed on a Linux based high performance cluster with 4 GPU's and 16 Intel CPU's of Skylake architecture.

In Table 4.8 an overview is given of the software used in this research. The code is developed in python by means of the Pandas and NumPy module for data processing, Matplotlib module for plotting, and scikit-learn and Ray for hyperparameter tuning, PyTorch module for neural networks, and Ray for neural network hyperparameter tuning [18][19][20][21][22][23][24]. The models are made with scikit-learn and Pytorch. We opted for python and these modules because of their simplicity, available support online and compatibility with the existing toolbox of ASML.

Module	Purpose
Pandas	Data processing
NumPy	Data processing
Matplotlib	Plotting
Scikit-learn	Linear models, hyperparameter tuning
PyTorch	Neural Networks
Ray	Hyperparameter tuning

Table 4.8: Modules used during development.

4.5 Execution

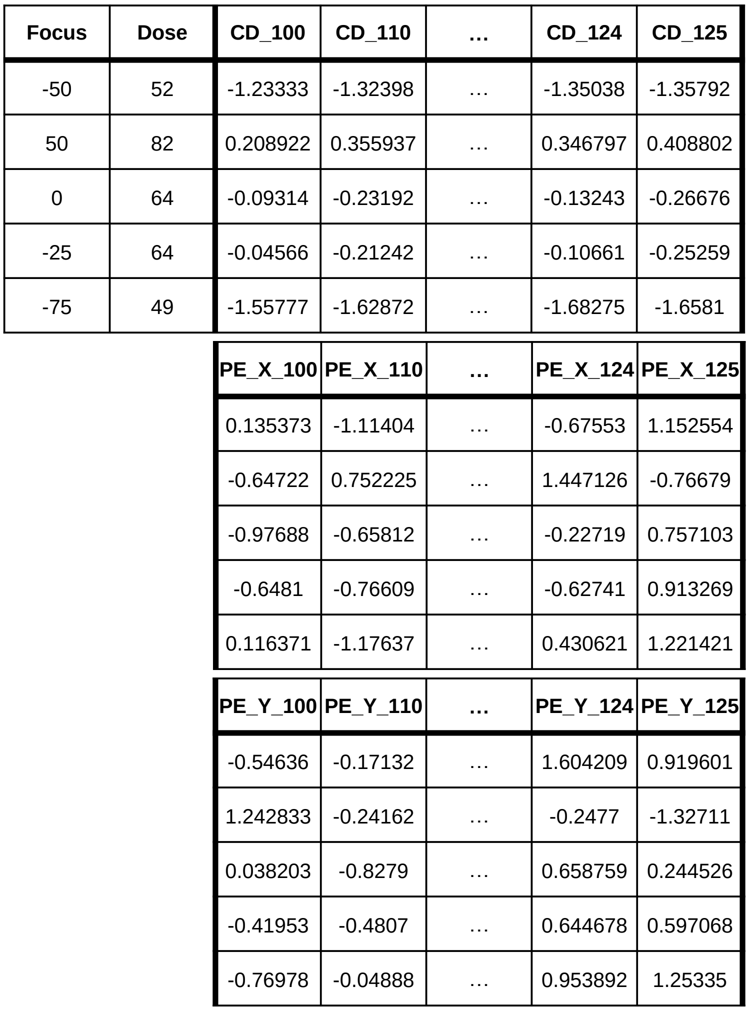
The research has been split up into three methods:

(I) Splitting the data into 80% training set and 20% test set

(II) Taking a nominal focus and a dose range as a test set and taking the rest as training. The nominal focus was set on -25.0 nm with a dose range of 70.0 mJ/cm^2 ± 14%

(III) Calculating the focus uniformity on the wafer by fitting on all the data. This describes how much focus prediction differs from the expected focus value in general on the entire wafer.

A scaler will be fitted on the input data of the trainingsplit, which will standardize the input features by the formula $z = \frac{x-\mu}{\sigma}$. This scaling is done for the training data at field, pad and SEM granularity, which results in Table 4.9, 4.10 and 4.11 respectively.



Focus	Dose	CD_100	CD_110	...	CD_124	CD_125
-50	52	-1.23333	-1.32398	...	-1.35038	-1.35792
50	82	0.208922	0.355937	...	0.346797	0.408802
0	64	-0.09314	-0.23192	...	-0.13243	-0.26676
-25	64	-0.04566	-0.21242	...	-0.10661	-0.25259
-75	49	-1.55777	-1.62872	...	-1.68275	-1.6581

PE_X_100	PE_X_110	...	PE_X_124	PE_X_125
0.135373	-1.11404	...	-0.67553	1.152554
-0.64722	0.752225	...	1.447126	-0.76679
-0.97688	-0.65812	...	-0.22719	0.757103
-0.6481	-0.76609	...	-0.62741	0.913269
0.116371	-1.17637	...	0.430621	1.221421

PE_Y_100	PE_Y_110	...	PE_Y_124	PE_Y_125
-0.54636	-0.17132	...	1.604209	0.919601
1.242833	-0.24162	...	-0.2477	-1.32711
0.038203	-0.8279	...	0.658759	0.244526
-0.41953	-0.4807	...	0.644678	0.597068
-0.76978	-0.04888	...	0.953892	1.25335

Table 4.9: Focus and dose output values with scaled CD, PE_x and PE_y input values at field granularity level.

Focus	Dose	Pad ID	CD_100	CD_110	…	CD_124	CD_125	PE_X_100	PE_X_110	…	PE_X_124	PE_X_125	PE_Y_100	PE_Y_110	…	PE_Y_124	PE_Y_125
50	94	3	1.081006	1.457611	…	1.178727	1.434317	-0.61358	2.471498	…	-0.34784	-2.16722	1.829314	-0.90301	…	-1.42894	-1.16954
-25	88	0	1.278044	1.168377	…	1.235112	1.182597	1.715022	0.424345	…	-0.582	-0.63815	0.172229	1.405764	…	-0.6807	-0.03193
25	76	2	0.372938	0.317151	…	0.393644	0.328417	-0.52169	0.001571	…	-0.26709	-0.05588	0.312502	-0.18898	…	-0.14614	-0.50005
-75	76	3	0.051961	-0.05692	…	-0.02528	-0.05729	-0.38135	-0.4922	…	0.527701	0.455882	-1.06162	-0.06584	…	0.185748	1.046028
0	73	2	0.544003	0.377609	…	0.505975	0.368006	-0.22092	-0.38553	…	0.908941	0.382637	0.082694	-0.83678	…	0.327887	0.096279

Table 4.10: Focus and dose output values with scaled CD, PE_x and PE_y input values at pad granularity level.

Focus	Dose	Sequence ID	CD_100	CD_110	...	CD_124	CD_125	PE_X_100	PE_X_110	...	PE_X_124	PE_X_125	PE_Y_100	PE_Y_110	...	PE_Y_124	PE_Y_125
-25	64	3797	-0.03662	-0.21948	...	-0.1034	-0.27259	-0.18709	-0.78984	...	-0.93344	1.064217	-0.59862	-0.49334	...	0.52053	0.612059
-50	82	2142	0.900439	0.71374	...	0.864866	0.692344	-0.02813	-0.17565	...	1.666373	0.118489	-0.53145	0.308827	...	0.91515	0.970398
25	82	4041	0.772683	0.744075	...	0.772227	0.829814	1.479828	0.475354	...	0.084638	-0.98987	0.157514	0.989609	...	-0.45981	-0.13648
-25	67	800	0.079662	-0.09477	...	0.026316	-0.1584	-0.45302	-0.69117	...	0.410547	0.718728	-0.39025	-1.53083	...	-0.46373	-0.00642
-25	64	819	-0.03879	-0.20904	...	-0.13288	-0.26993	-1.01298	-0.70449	...	-0.96717	0.917627	0.935205	0.384764	...	-0.66074	0.512626

Table 4.11: Focus and dose output values with scaled CD, PE_x and PE_y input values at SEM granularity level.

In method (I) the R^2 is used to indicate whether a fit is sufficient enough compared to the baseline model. The RMSE from method (II) and the 3σ from method (III) will be used to give the final conclusion on which algorithm is the best in each experiment.

4.5.1 Simple machine learning models

Following method (I), (II) and (III) the focus and dose values used in method (I) and method (II) for training and testing of the simple machine learning models are indicated in Figure 4.7 where red fields are from the test set and blue fields are from the training set. After standardizing the input features, the fitted scaler is applied on the test data.

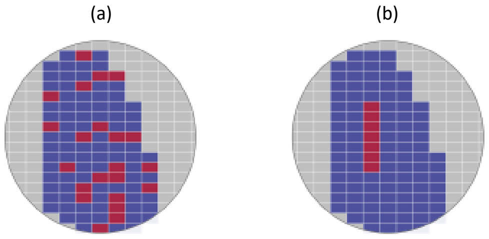


Figure 4.7: Train and test sets used per method in the simple machine learning models. Blue fields are used in training, red fields are used in testing. (a) Method (I) (b) Method (II)

The hyperparameter tuning of the simple machine learning models is done with GridSearch. With this approach all possible combinations of hyperparameter values are tried out. The alternative could be RandomSearch which tries out random combinations of hyperparameter values for a given amount of attempts to find the best hyperparameter values. Since RandomSearch is random it does not ensure finding the best hyperparameter settings. Therefore we use GridSearch, even though it is an exhausting approach.

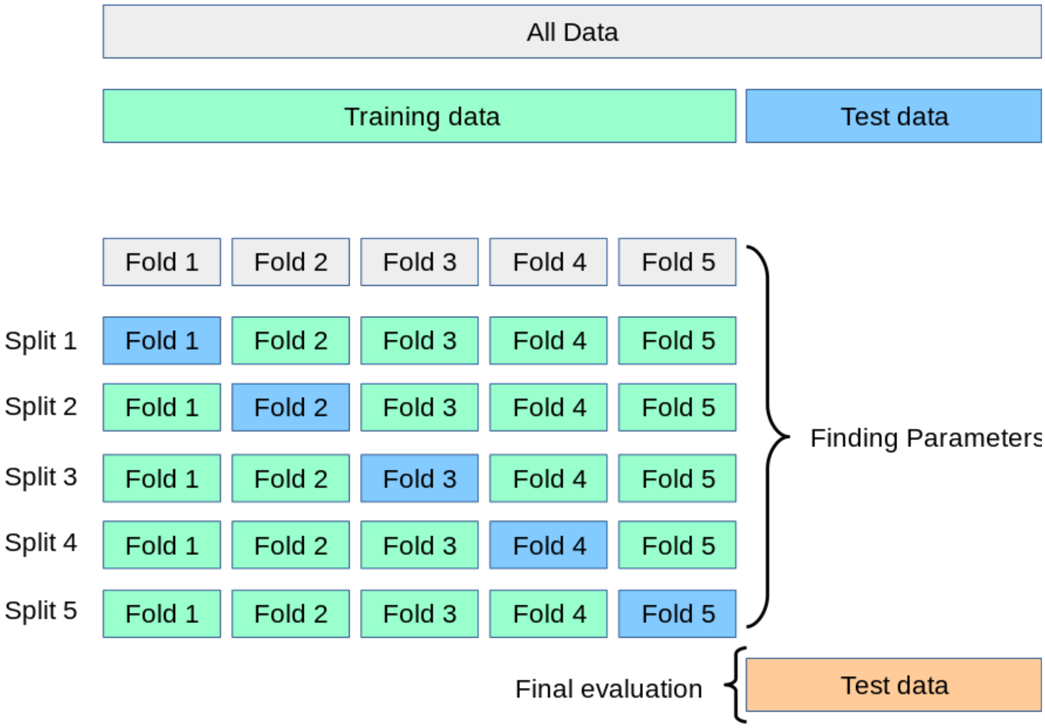


Figure 4.8: Illustration of the testing procedure with KFold cross-validation for $K = 5$ [25]

In both Method (I) and (II) we do a cross-validation by means of GroupKFold, where $K = 10$. With KFold, training data is split into K folds of equal size. One fold is selected for validation, the remaining $K - 1$ folds for training and while all possible hyperparameters combinations. This is done K times for all the different validation folds that can be selected as illustrated in 4.8. From the K procedures a mean R^2 is calculated to select the best hyperparameter settings. The collection of hyperparameter values per model is shown in Figure 4.12.

Algorithm	Hyperparameters	Values
Linear Regression	-	-
Lasso Regression	Alpha	10^{-6}, 10^{-5}, ..., 10^{4}, 10^{5}
	Maximum number of iterations	10000
	Warm start	True
Ridge Regression	Alpha	10^{-6}, 10^{-5}, ..., 10^{4}, 10^{5}
	Maximum number of iterations	1000
Elastic Net	Alpha	10^{-6}, 10^{-5}, ..., 10^{4}, 10^{5}
	L1 ratio	0.1, 0.2, ..., 0.8, 0.9
	Maximum number of iterations	10000
	Warm start	True

Table 4.12: Hyperparameter settings for different machine learning models

When applying KFold cross-validation at pad and SEM granularity level, it might occur that instances coming from the same field are split up into different folds, resulting in a cross-validation in which training and validation happens on the same focus and dose combinations. To prevent this GroupKFold is used rather than KFold. By defining instances with the same focus and dose combinations as a group (i.e. from the same field), instances from the same field no longer can be split up over different folds. This means that at pad granularity level a group has a common focus and dose combination and variable Pad ID and that groups at SEM granularity level have a common focus and dose combination with a variable Sequence ID. Note that when data is aggregated on field granularity level GroupKFold and KFold boil down to the same cross-validation setup.

4.5.2 Neural Network approach

For the neural networks a similar approach as method (I), (II) and (III) is being used. Since a validation set is introduced for the training process in neural networks the percentage of data used in each data subsets is different than for the data subsets in the simple machine learning models. The corresponding focus and dose values and percentages of data used in the training, validation and test set of method (I), (II) and (III) are shown in Figure 4.9(a), (b) and (c) respectively.

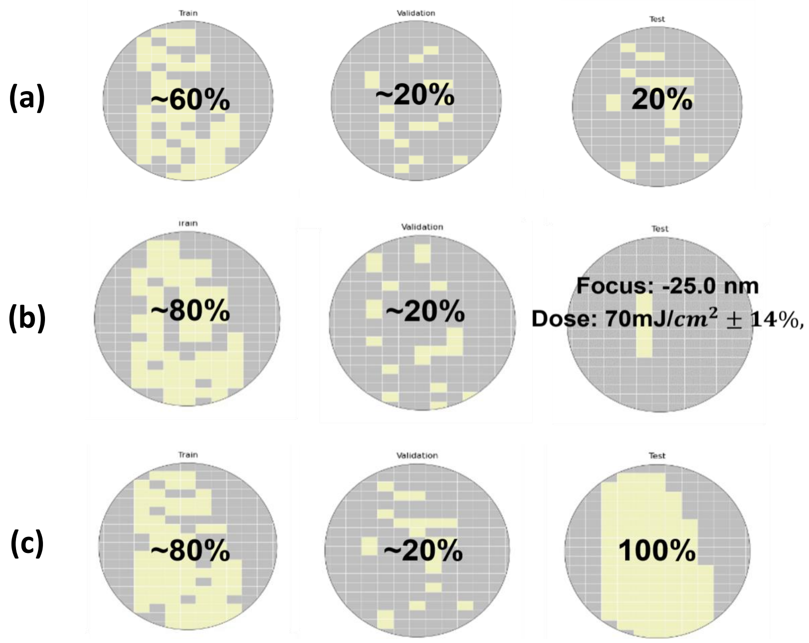


Figure 4.9: Train, validation and test sets used per method in the neural networks indicated in yellow with the percentage of data per subset. (a) Method (I) (b) Method (II) (c) Method (III)

The data will be split up into a training, validation and test split. After standardizing the input features, the fitted scaler is applied on the validation and test set. From the available loss functions of Pytorch we are using the MSE loss as this is closely related to RMSE. With MSE large errors get penalized harder due to the squaring of the error. Further we use the Adam optimizer, which is based on stochastic gradient descent. It chosen as it has shown to perform well in other researches and capable of handling sparse data [26].

For the number of neurons in a hidden layer there is no clear formula as multiple researches propose different solutions [27]. However, there are general rules of thumbs that can help towards finding the optimal number of neurons [28]. The first rule is that the number of neurons for a hidden layer should be between the size of the input layer and the output layer. The second rule is that the number of hidden neurons is smaller than two times the number of output nodes. The third rule states that the minimum number of hidden layers and neurons should capture 70-90% of the variance of the input data. Given that there are 11 orthogonal groups with features CD, PE_x and PE_y, the input has 33 input nodes. There is one output nodes for focus, so by following the first rule of thumb the number of hidden neurons should be between 1 and 33. We deviate slightly from this and select for each hidden layer in the neural network the amount of nodes to be in the range [8, 16, 32, 64] as shown in the list of hyperparameter values in Figure 4.13.

Hyperparameters	Values
Layers	1, 2, 3, 4
Nodes	8, 16, 32, 64
Dropout	0.0, 0.1, 0.2, 0.3, 0.4, 0.5
Batch Normalization	False, True

Table 4.13: Hyperparameter settings used in the prediction models

We assume that the PE and CD are a direct result of only changes in focus, meaning that dose and focus are independent. If we would make one model with two output nodes for focus and dose, the models would share the weights in all hidden layers and would make focus and dose correlated. Therefore we opt for neural networks with only one output node, i.e. a model for focus prediction and a model for dose predictions.

A dataloader is implemented to split the data into batches such that it can fit into memory. A batch size of 16 is being used and samples in the training set and validation set are shuffled during each epoch. The last samples that form a batch smaller than the given batch size are removed from the training epoch. As we want to implement batch normalization, a batch smaller than the given batch size is removed to ensure that a batch has more than one sample, which is a requirement for batch normalization.

A problem in general for neural networks is the overfitting problem. With dropout some neurons are made inactive defined by a ratio, while the remaining neurons become the relevant ones that can describe the training data sufficiently without changing the neurons that are inactive. This is a form of regularization, which helps to prevent overfitting. For dropout we use the probability 0.0 upto 0.5 for the hidden layers with the exception for the input layer, which is set to a maximum dropout probability of 0.2. This means a minimum of 50% of the neurons in the hidden layers is kept and at least 80% of the neurons in the input layer.

The network has to be trained for a sufficient amount of epochs in order to learn the mapping from input to output. If learning has been insufficient it might lead to underfitting. However, too many epochs might result in good performance on the training set but bad performance on the test set, i.e. overfitting. To prevent overfitting early stopping can be applied, which stops the training when the loss on the validation set stops decreasing. In order to prevent underfitting we try to estimate the amount of epochs needed for training in Experiment (A). By setting the maximum amount of epochs of training high and implementing an early stopping criterion an estimate on the appropriate amount of epoch for sufficient learning can be made.

Hyperparameters	Values
Optimizer	Adam
Number of layers	1, 2, 3, 4
Number of nodes	32, 64
Batch size	16
Learning rate	0.01

Table 4.14: Hyperparameter settings used to find the appropriate number of epochs.

Multiple models have been made and trained with fixed hyperparameters, which are shown in table 4.14. By observing the training and validataion curves we can make an estimate on an appropriate amount of epochs based on convergence of the validation loss curve. In general, different hyperparameter settings influence the amount of epochs to achieve optimal performance. So our estimate holds under the assumption that the number of epochs is independent of changes in other hyperparameters. This estimate also depends on the input data, so for field, pad and SEM granularity level input instances we have to find the appropriate amount of epochs separately.

Chapter 5

Proximity Force feature - Alternative to grouping scheme

In this chapter the prior work from section 2.2.1 on which this research is based will be discussed more thoroughly. By discussing the drawbacks of this work we opt for a new feature to address these issues.

5.1 Drawbacks

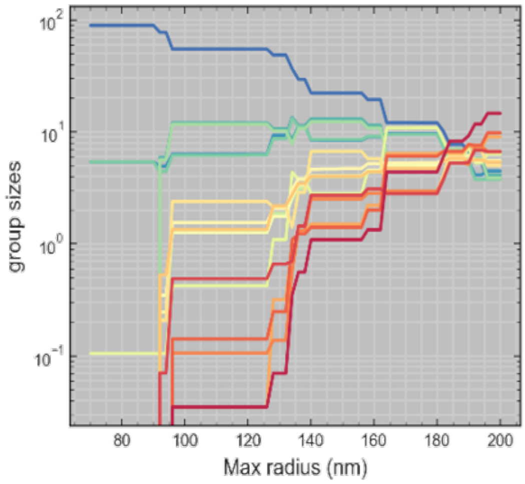


Figure 5.1: Increasing the ambit size leads to saturated grouping. Colors correspond with the groups in Figure 2.4(a)

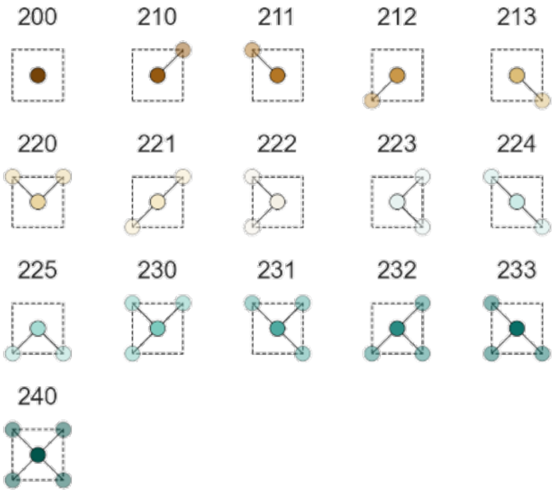


Figure 5.2: Diagonal classification scheme as alternative to the orthogonal classification scheme from Figure 2.4(a)

There are several drawbacks of the classification approach. First, the value K for KNN. The number of contact holes to look for determines how many neighboring contact holes are taken into consideration when assigning a group label. Second, the ambit size can heavily influence the results as it is dependent on the chip layout of interest. Similarly to K in KNN, it influences how the neighboring contact holes will get distributed among the quadrants. In Figure 5.1 the count per grouping within an SEM image for different ambit sizes is shown, where the colors correspond with the groups in Figure 2.4(a). When the ambit sizes increases, the grouping starts to saturate towards a small set of dominant groups, i.e. groups with neighbors in three or four quadrants. The groups with less neighbors will shrink in group size or might even disappear. This results in the wrong labeling of the proximity effect that has occurred. Third, besides the ambit size and K in KNN, the classification scheme in itself is a hyperparameter that can be tuned. In Figure 5.2 an alternative to the orthogonal grouping is shown, which uses diagonal patterns, i.e. the quadrants top-left, top-right, bottom-left and bottom-right. Similarly, the ambit can be divided into more than four areas to get a more refined classification scheme. Lastly, this approach does not quantify the proximity effect, but only describes the direction in which the proximity effect might have an impact on contact holes based on the quadrants.

5.2 Feature Proposal

In order to address the drawbacks mentioned before, we opt for a new feature based on the classification approach. Rather than classifying the direction of the proximity effect based on quadrants, we want to describe the strength of the proximity effect numerically. The proximity effect tells us that there is an inverse relationship between the strength of the effect and the distance between structures. Therefore, we empirically look for a inverse relationship based on the distance to describe the strength of the proximity effect, which we will call '*proximity force*' (PF). With this feature we address the drawbacks of the classification method by making the proximity effect quantifiable and introducing less hyperparameters to be tuned. The strength of the proximity effect will change as function of focus and dose, therefore describing this strength quantitatively might be beneficial for focus and dose prediction. In the following subsections two models are introduced that form the physical reasoning behind our calculations: a vector-based model and a wave-based model.

5.2.1 Vector model

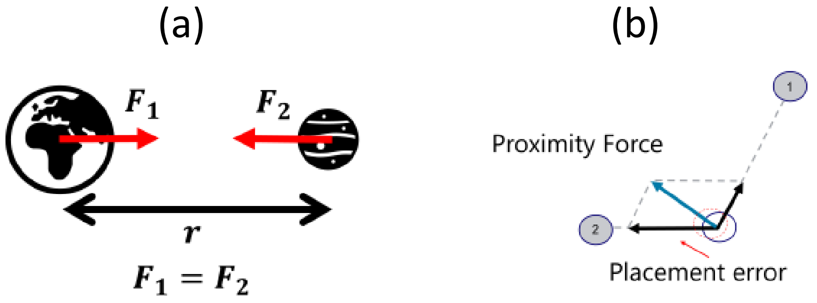


Figure 5.3: Proximity Force is defined as the sum of force vectors proportional to the inversed distance between neighboring contact holes.

In Figure 5.3 the idea of the proximity force by means of vectors is shown. Figure 5.3(a) shows a schematic overview of gravitational force, where two objects attract each other with an increasing force when distance decreases. The gravitational force F is given by $F = G\frac{m_1 m_2}{r^2}$, where G is the gravitational constant, m_1 and m_2 the masses of the object and r the distance

[29]. We know that the proximity effect is strong on relatively short distances and weak on big distances between neighboring contact holes. This corresponds with the gravitational force analogy in which we treat contact holes as objects that attract eachother. This idea is mapped onto force vectors in Figure 5.3(b) from which can be observed that the inverse distance determines the magnitude of the force vectors indicated in black. These vectors describe the strength and direction of the proximity effect with respect to each neighboring contact hole. Once these vectors are summed up, the net vector indicated in blue describes the strength and direction of the total proximity effect, which corresponds with the direction and magnitude of the placement error indicated in red. We call this net vector the proximity force.

The PF calculation for one contact hole i and different order n functions is given in eq. 5.1. Here j is a contact hole in the local neighborhood of i, N the amount of neighboring contact holes in the neighborhood and r the distance between the contact holes i and j measured on the wafer.

$$PF_i = \sum_{j}^{N} \frac{1}{r_{ij}^n} \tag{5.1}$$

With this formula we are able to describe the strength of the proximity effect, while addressing the drawbacks mentioned in section 5.1. Firstly, we are able to describe the strength of the proximity effect. Secondly, we are able to describe the direction of the proximity effect on a 360 circle rather than a rough direction, e.g. a quadrant. So there is no need for defining an orthogonal or diagonal grouping system based on an arbitrary number of quadrants. Furthermore, since we are defining the strength of the proximity effect based on this, there is no need anymore to set up a fixed ambit size and no need to define K for KNN. Lastly, with no grouping system there is also no case of saturated grouping.

5.2.2 Wave model

An alternative mechanism for the proximity force feature might be based on waves. The single slit experiment rests on a fundamental principle in optics, namely diffraction. With a single slit experiment light is being emitted through a small slit as shown in Figure 5.4. Due to the small size of the slit, light is being diffracted such that a pattern of different intensities becomes visible. Since EUV lithography involves developing structures by means of light, contact holes on the mask can be seen as individual slits that diffract light since contact holes are of nanometer scale. For each contact hole a diffraction pattern can be determined based on the CD.

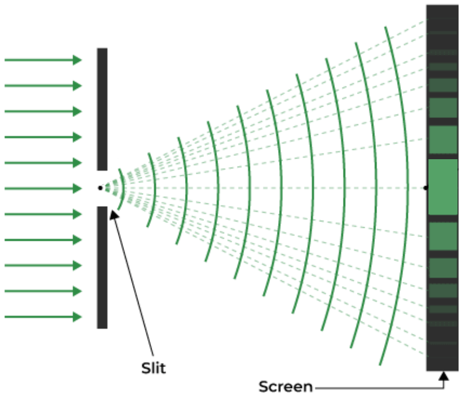


Figure 5.4: Single slit experiment shows the interference pattern on the screen. Adapted from [30]

Note that the diffraction occurs on the mask, so instead of using wafer dimensions as done in the vector model we need to use the mask dimensions. Since our data comes from a low-NA machine with a lens system with a reduction factor of 4x for both the X- and Y-direction, all the dimensions on the mask are 4x bigger than on the wafer. This implies that the CD and distances have to be scaled accordingly.

$$\operatorname{rect}\left(\frac{x}{a}\right) = \begin{cases} 0, & \text{if } |x| > \frac{a}{2} \\ \frac{1}{2}, & \text{if } |x| = \frac{1}{2} \\ 1, & \text{if } |x| < \frac{a}{2} \end{cases} \tag{5.2}$$

Ideally for each contact hole a rectangular function would be used to expose the wafer. In Figure 5.5(a) a plot of a rectangular function is given by Equation 5.2 where width $a = 1$. However, in reality the intensity function will take shape of a diffraction pattern as shown in Figure 5.5(b) [31]. From Figure 5.5(b) can be observed that the highest intensity is found around $x = 0$ between the first positive and negative minima. The intensity in this range will be used for the contact hole itself, but the other intensities observed beyond this range are caused by diffracted light. These intensities will contribute to the intensities of surrounding contact holes.

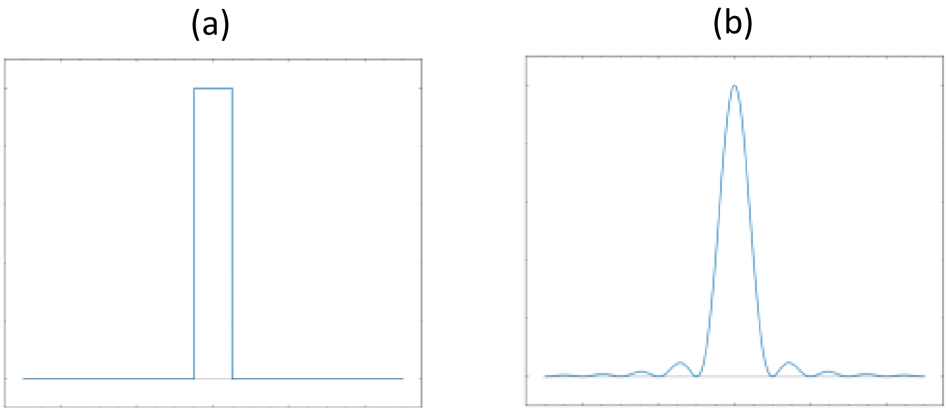


Figure 5.5: Plots of the ideal intensity function (a) and reality (b).

$$y = \left[\frac{2\mathrm{J}_1(x)}{x} \right]^2 \tag{5.3}$$

$$PF_i = \sum_{j}^{N} \left[\frac{2\mathrm{J}_1(r_{ij})}{r_{ij}} \right]^2 \tag{5.4}$$

The function used in Figure 5.5 is the Fraunhofer diffraction formula given by Equation 5.3. It is based on the first kind Bessel function J_α of the first order, i.e. $\alpha = 1$. With Equation 5.4 we define the proximity force of a contact hole i by summing the values following the Fraunhofer diffraction formula on each neighboring contact hole j, where r_{ij} is the distance between contact hole i and a neighboring contact hole j.

Important to note is that the diffraction pattern is based on distance between the aperture plane and the observation plane, i.e. the mask and the wafer respectively. The requirement for Fraunhofer diffraction is that the diffraction is measured in 'far-field'. In Figure 5.6 the difference in diffraction patterns is shown for increasing distances between the aperture plane and the observation plane [32]. The Fresnel number FN, based on the slit width a, distance L and wavelength of light λ, determines which diffraction pattern will be observed. The formula for FN is given in Equation 5.5

$$\mathrm{FN} = \frac{a^2}{L\lambda} \tag{5.5}$$

For EUV lithography we can pick contact holes of 100 nm and wavelength of 13.5 nm, so for $a = 100 \cdot 10^{-9}$ m and $\lambda = 13.5 \cdot 10^{-9}$ m we end up with

$FN = \frac{7.4 \cdot 10^{-7}}{L}$. Given that light in a high-NA machine traverses a series of mirrors between the mask and the wafer, L is in the range of meters, which results in FN $\ll$ 1. Following FN $\ll$ 1, the observation plane is in 'far field' and diffraction follows a Fraunhofer pattern.

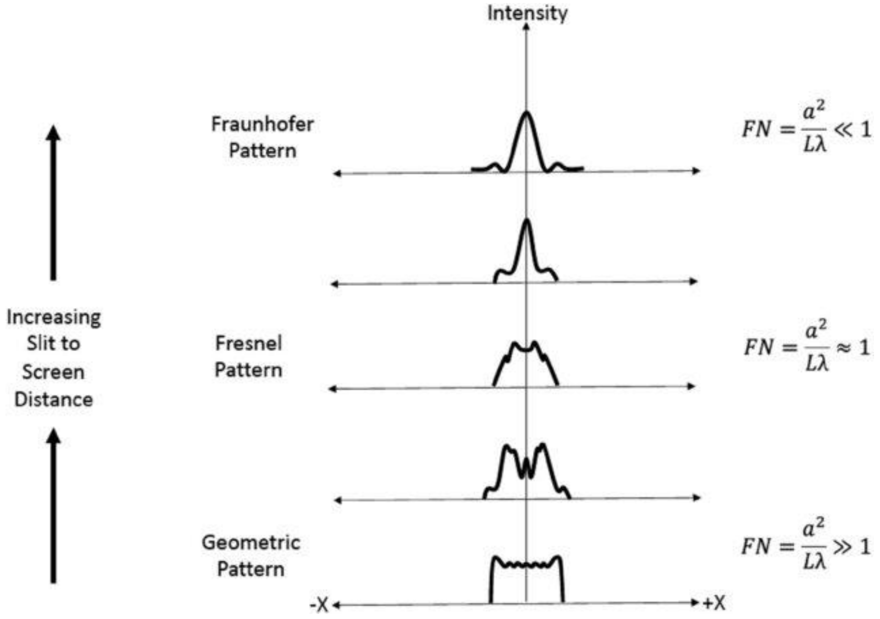

Intensity
Fraunhofer Pattern
$FN = \frac{a^2}{L\lambda} \ll 1$
Increasing Slit to Screen Distance
Fresnel Pattern
$FN = \frac{a^2}{L\lambda} \approx 1$
Geometric Pattern
$FN = \frac{a^2}{L\lambda} \gg 1$
-X
+X

Figure 5.6: Diffraction patterns for increasing distance between aperture plane and observation plane [32]

5.3 Approach

Since the hypothesis is that the proximity effect is dependent on the focus and dose we expect to see a relationship between the PE and the proximity force feature. We take the intended locations of contact holes per field on the wafer and calculate for each contact hole the relative distance to their 250 nearest neighbors by means of KNN. The choice of 250 neighbors is an arbitrary choice, which includes neighboring contact holes that are in practice not considered as in close proximity. Similarly to the ambit and the

K amount of nearest neighbors to select in the work from Calado et al. it is unknown what a proper amount of neighbors is for our calculations. However, since the proximity effect follows an inverse-distance relationship we know that the neighboring contact holes with large distance contribute less to the proximity effect. So by defining the amount of neighbors sufficiently large, the contribution of far distant neighboring contact holes tend to be close to zero, hence the choice for $K = 250$.

For each field we calculate the proximity force and perform a linear fit of PF against the PE. It is expected that this fit on each field gives a different slope following the combination of focus and dose. By using the slope of this linear relationship we can define the sensitivity of the proximity effect on the printing errors for focus and dose values. By observing the slope of the fit, the R^2 of the fit and the cosine similarity between $\overrightarrow{PE} = (PE_x, PE_y)$ and $\overrightarrow{PF} = (PF_x, PF_y)$ for the given combinations of focus and dose we define the most appropriate relationship that describes the proximity effect.

The cosine similarity is a measure that checks similarity between two vectors by mapping them onto their respective normalized vector. So it describes the similarity in direction independent of the magnitude. Following Equation 5.6 the cosine similarity ranges between -1 and 1, where 0 implies vectors being orthogonal, 1 implies vectors having the exact same angle, and -1 vectors being in complete opposite directions. As we want to map the proximity force onto the placement errors we ideally want a cosine similarity of one between PE and PF.

By empirically testing out different functions we determine the most appropriate functions by iteratively excluding functions. We start with the vector models that consist of the $1/r^n$ functions. After determining the best vector-based model(s) we explore the wave model consisting of Fraunhofer diffraction function to potentially get a more appropriate function.

$$\text{cosine similarity}(\overrightarrow{a}, \overrightarrow{b}) = \frac{\overrightarrow{a} \cdot \overrightarrow{b}}{|\overrightarrow{a}||\overrightarrow{b}|} \tag{5.6}$$

Chapter 6

Results

In this research five experiments have been conducted. In Experiment (A) the appropriate amount of epochs to train a neural network for focus prediction is estimated. Following the findings from Experiment (A) neural networks and other regression models are trained in Experiment (B). Through empirical research the new proximity force feature introduced in Chapter 5 will be defined in Experiment (C). Following the new feature definition from Experiment (C) neural networks and other regression models are trained in Experiment (D). In Experiment (E) an analysis is performed to explain the difference in performance between the datasets from Experiment (B). The results of each experiment are shown below.

6.1 Experiment A: Find the appropriate amount of epochs for neural networks

With training a neural network, there is a risk of potential underfitting. This underfitting might be caused by not sufficiently training the model, i.e. having not enough training epochs. Therefore we estimate what the minimal amount of training epochs should be by inspecting the training and validation curves during hyperparameter tuning. For each datasplit on all granularity levels we investigate the loss curves for models with 32 nodes and 64 nodes per hidden layers. By investigating the loss curves for 32 and 64 nodes per hidden layer we can make an assumption on whether the loss curves change substantially with different hyperparameters.

6.1.1 Field granularity

At field granularity level five model fits are being performed with learning rate 0.001, 2000 epochs and 1, 2, 3, 4 hidden layers at 32 and 64 nodes per hidden layer. The mean-squared error (MSE) loss curves of training and validation are given for each datasplit.

I. 80%/20% random data split

In Figure 6.1 and 6.2 the training and validation loss curves for the 80%/20% datasplit and 32 nodes per hidden layer are shown. Similarly for 64 nodes per hidden layer, Figure 6.3 and 6.4 show the training and validation loss curves. Within each figure, the subfigure (a), (b) and (c) show the same loss curves, albeit on different scale. Subfigures (a) show the epochs on a linear x-axis and loss on a linear y-axis, (b) shows the epochs on a linear x-axis and loss on a logarithmic y-axis, and (c) shows the epochs on a logarithmic x-axis and loss on a logarithmic y-axis. In Figure 6.2 and 6.4 the model with the lowest validation loss is indicated by a dashed line.

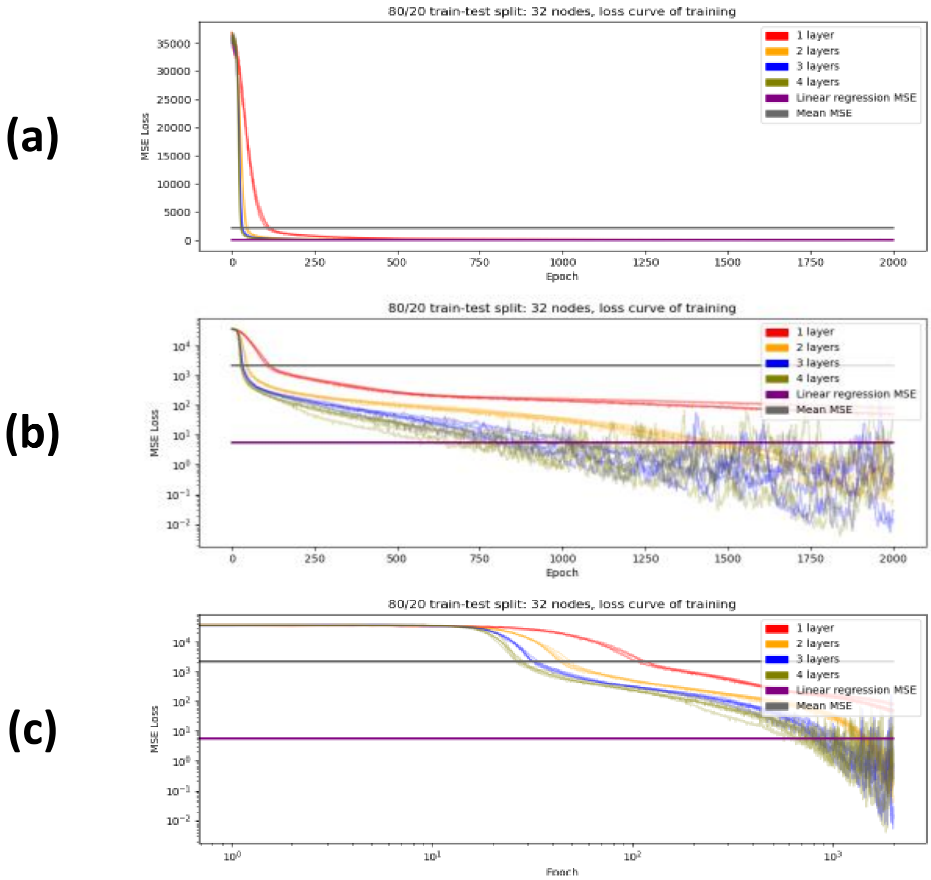


Figure 6.1: MSE training loss of five fits using field granularity inputs with 32 nodes per hidden layer. (a) Epochs and loss on linear scale. (b) Epochs on linear scale and loss on logarithmic scale. (c) Epochs and loss on logarithmic scale.

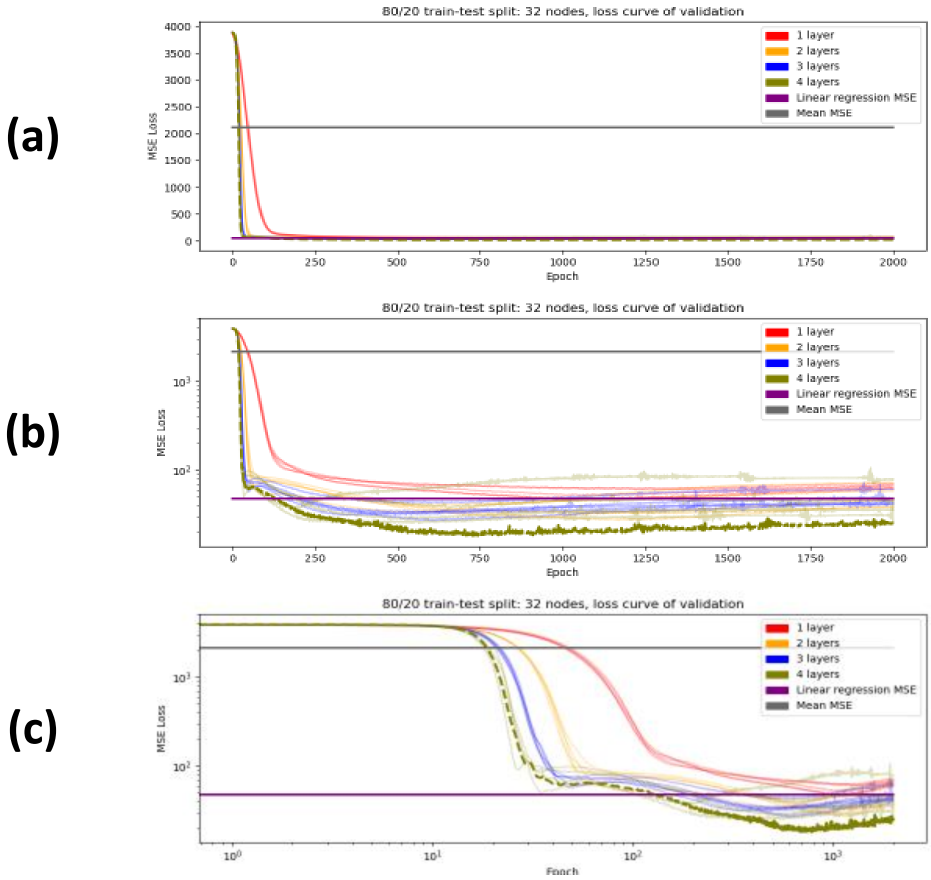


Figure 6.2: MSE validation loss of five fits using field granularity inputs with 32 nodes per hidden layer. (a) Epochs and loss on linear scale. (b) Epochs on linear scale and loss on logarithmic scale. (c) Epochs and loss on logarithmic scale.

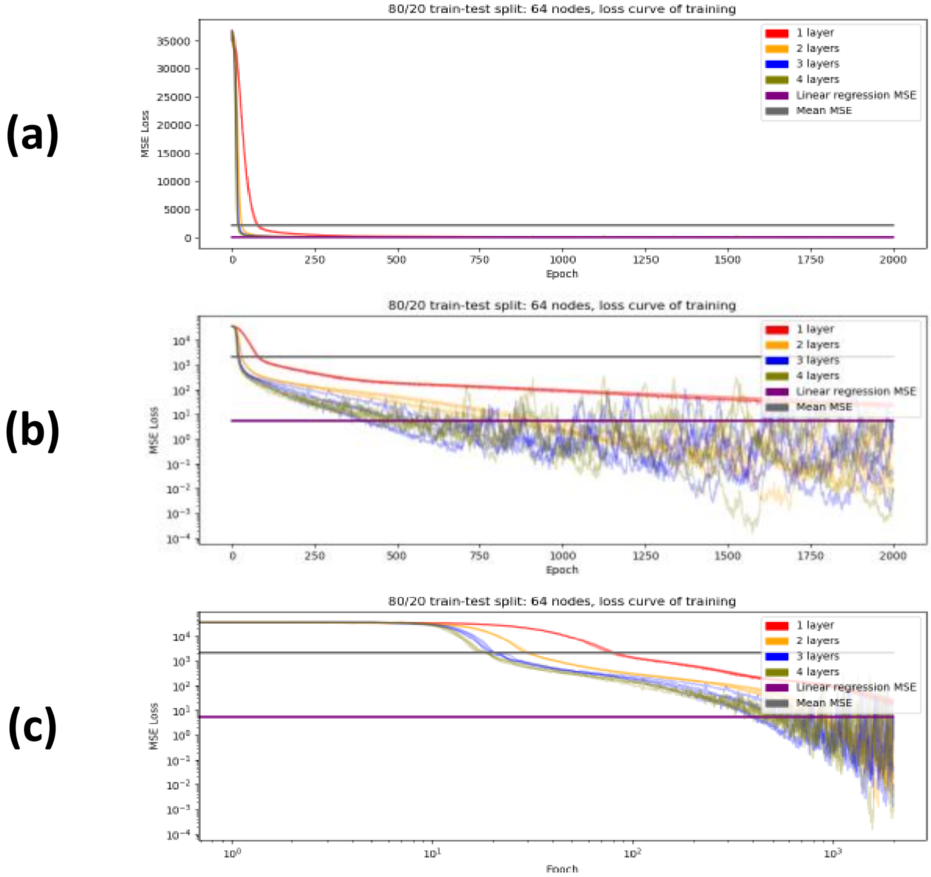


Figure 6.3: MSE training loss of five fits using field granularity inputs with 64 nodes per a hidden layer. (a) Epochs and loss on linear scale. (b) Epochs on linear scale and loss on logarithmic scale. (c) Epochs and loss on logarithmic scale.

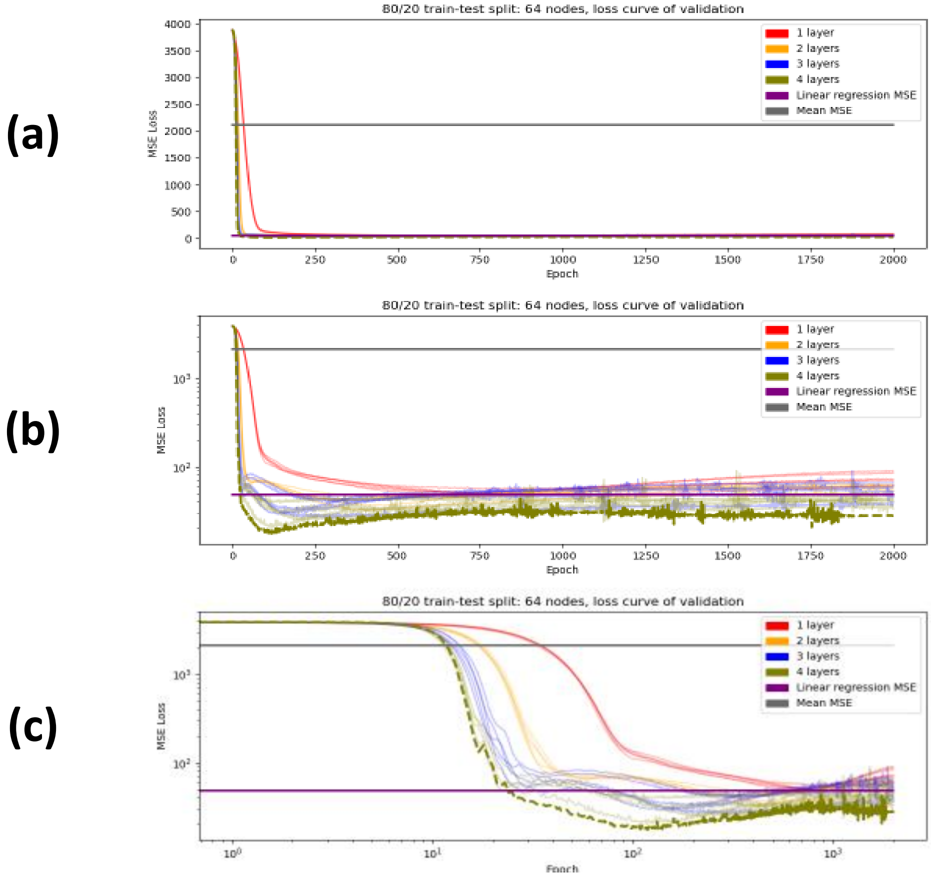


Figure 6.4: MSE validation loss of five fits using field granularity inputs with 32 nodes per hidden layer. (a) Epochs and loss on linear scale. (b) Epochs on linear scale and loss on logarithmic scale. (c) Epochs and loss on logarithmic scale.

In Figure 6.1 and 6.3 we see a continuous decline in the training loss curves, while the validation loss curves in Figure 6.2 and 6.4 converge. The validation loss curves show that the neural networks perform better than predicting the mean and the linear regression baseline.

II. Nominal range

In Figure 6.5 and 6.6 the validation loss curves for the nominal datasplit are shown for 32 and 64 nodes per hidden layer respectively. Within Figure 6.5 and 6.6, the subfigure (a), (b) and (c) show the same loss curves, albeit on different scale. Subfigures (a) show the epochs on a linear x-axis and loss on a linear y-axis, (b) shows the epochs on a linear x-axis and loss on a logarithmic y-axis, and (c) shows the epochs on a logarithmic x-axis and loss on a logarithmic y-axis. The model with the lowest MSE loss is indicated by a dashed line. In Figure 6.5 and 6.6 the model with the lowest validation loss is indicated by a dashed line.

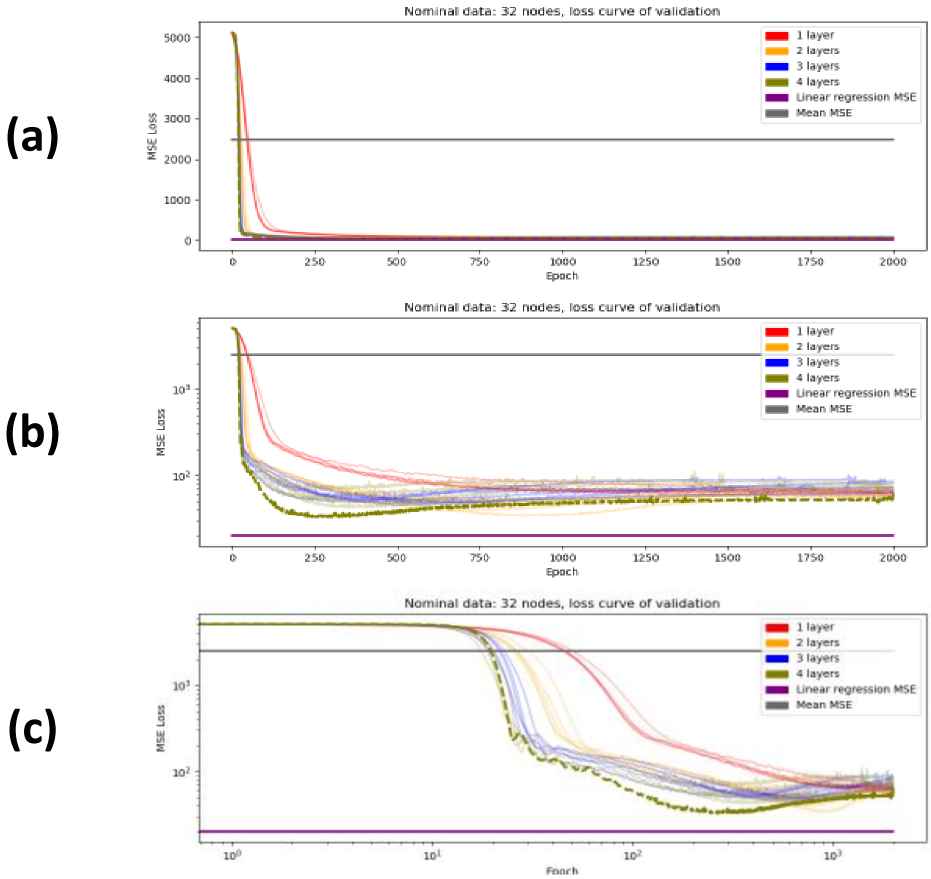


Figure 6.5: MSE validation loss of five fits using field granularity inputs with 32 nodes per hidden layer. (a) Epochs and loss on linear scale. (b) Epochs on linear scale and loss on logarithmic scale. (c) Epochs and loss on logarithmic scale.

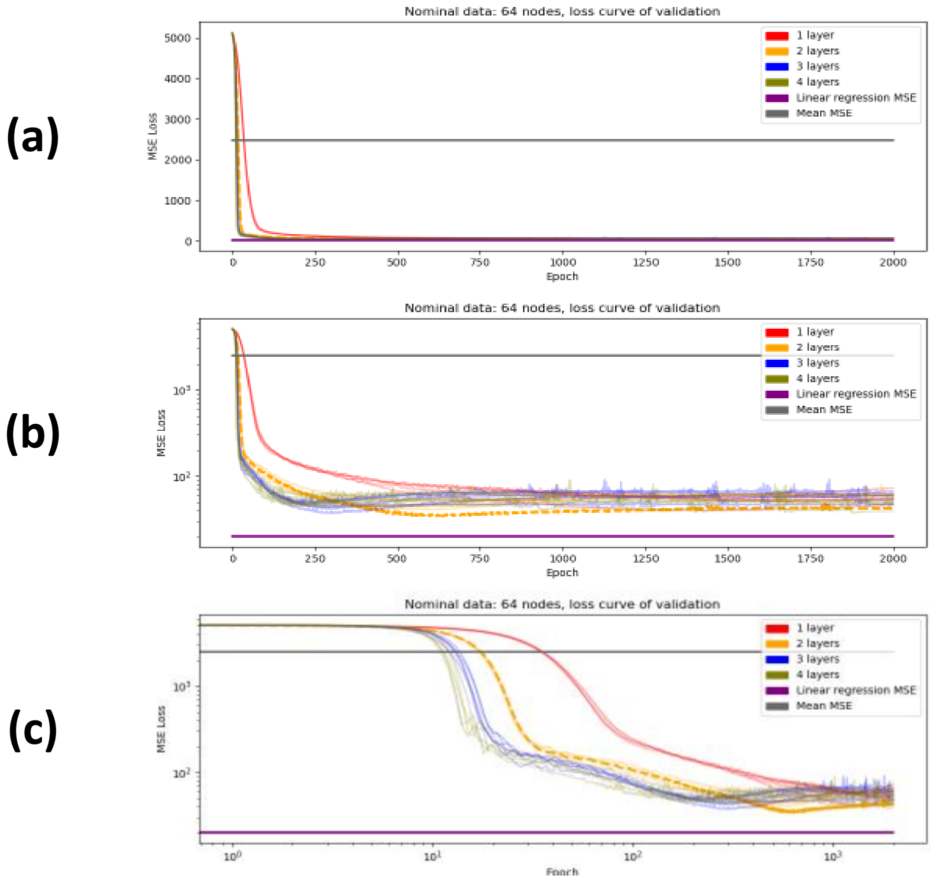


Figure 6.6: MSE validation loss of five fits using field granularity inputs with 64 nodes per hidden layer. (a) Epochs and loss on linear scale. (b) Epochs on linear scale and loss on logarithmic scale. (c) Epochs and loss on logarithmic scale.

In Figure 6.5 and 6.6 the validation loss curves show that the neural networks offer improvement with respect to the mean prediction, but not with respect to the baseline.

III. On all data

In Figure 6.7 and 6.8 the validation loss curves for the 3σ datasplit are shown where (a) shows the epochs on a linear x-axis and loss on a linear y-axis, (b) shows the epochs on a linear x-axis and loss on a logarithmic y-axis, and (c) shows the epochs on a logarithmic x-axis and loss on a logarithmic y-axis. In Figure 6.7 and 6.8 the model with the lowest validation loss is indicated by a dashed line.

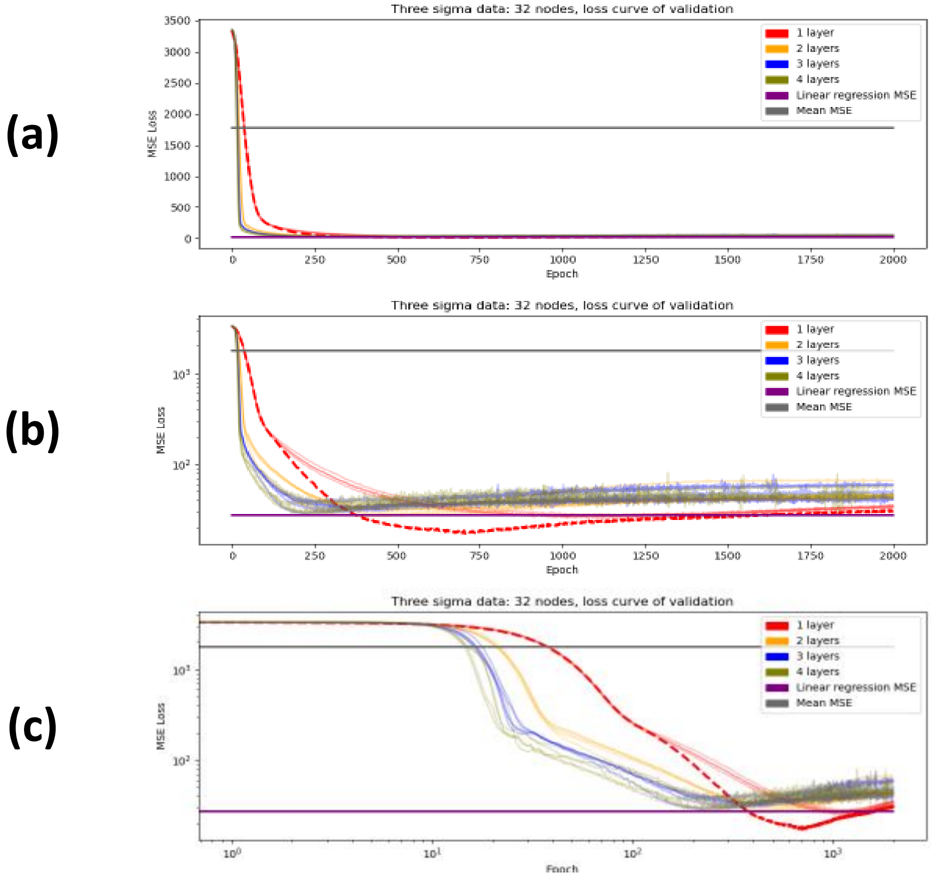


Figure 6.7: MSE validation loss of five fits using field granularity inputs with 32 nodes per a hidden layer. (a) Epochs and loss on linear scale. (b) Epochs on linear scale and loss on logarithmic scale. (c) Epochs and loss on logarithmic scale.

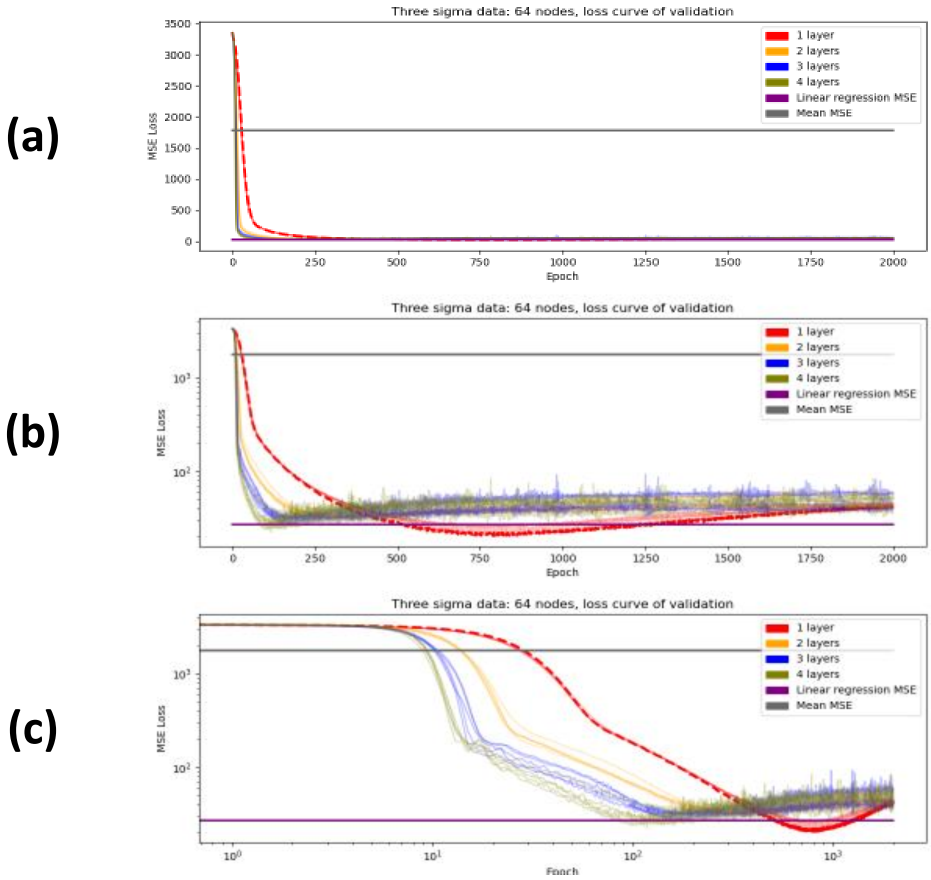


Figure 6.8: MSE validation loss of five fits using field granularity inputs with 64 nodes per a hidden layer. (a) Epochs and loss on linear scale. (b) Epochs on linear scale and loss on logarithmic scale. (c) Epochs and loss on logarithmic scale.

The validation loss curves from Figure 6.7 and 6.8 show that neural networks offer similar performance to the linear regression baseline, while outperforming the mean prediction.

6.1.2 Pad granularity

At pad granularity level five model fits are being performed similar to field granularity, albeit with less epochs. The fits use learning rate 0.001, 1000 epochs and 1, 2, 3, 4 hidden layers with 32 and 64 nodes per hidden layer. The training loss curves are shown in Appendix A. The validation loss curves for 32 and 64 nodes at different datasplits are shown below.

I. 80%/20% random data split

In Figure 6.9 and 6.10 the MSE loss curves for validation on the 80%/20% datasplit are shown where (a) shows the epochs on a linear x-axis and loss on a linear y-axis, (b) shows the epochs on a linear x-axis and loss on a logarithmic y-axis, and (c) shows the epochs on a logarithmic x-axis and loss on a logarithmic y-axis. The model with the lowest validation loss is indicated by a dashed line.

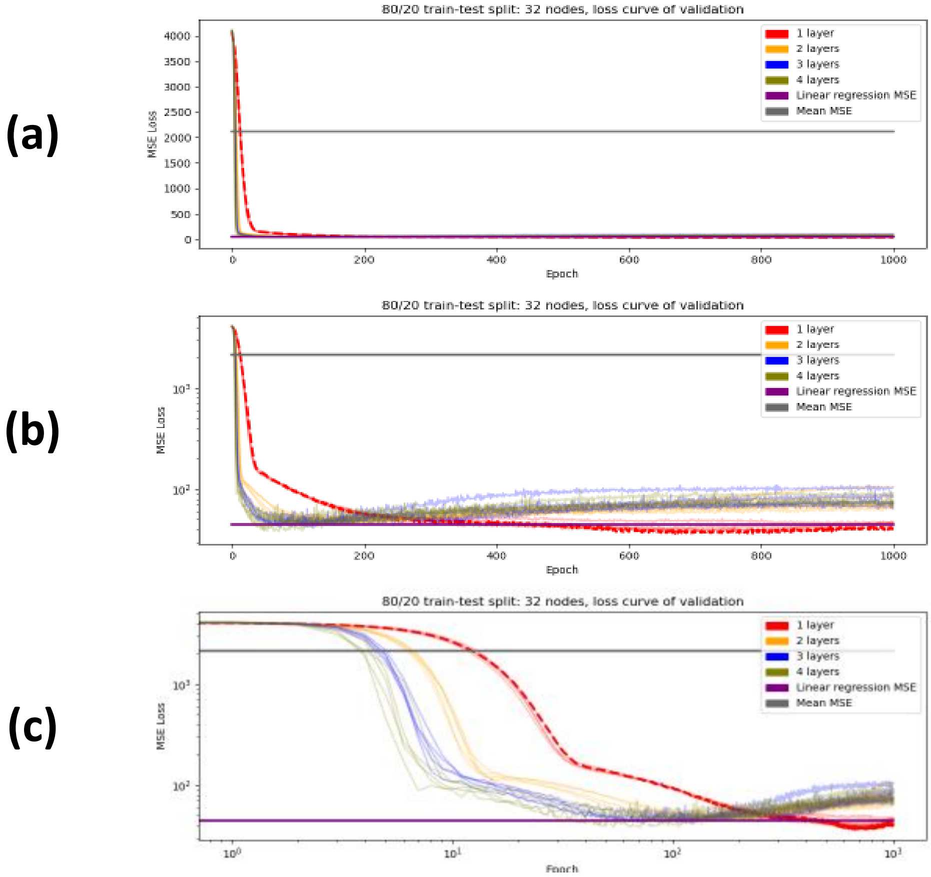


Figure 6.9: MSE validation loss of five fits using pad granularity inputs with 32 nodes per a hidden layer. (a) Epochs and loss on linear scale. (b) Epochs on linear scale and loss on logarithmic scale. (c) Epochs and loss on logarithmic scale.

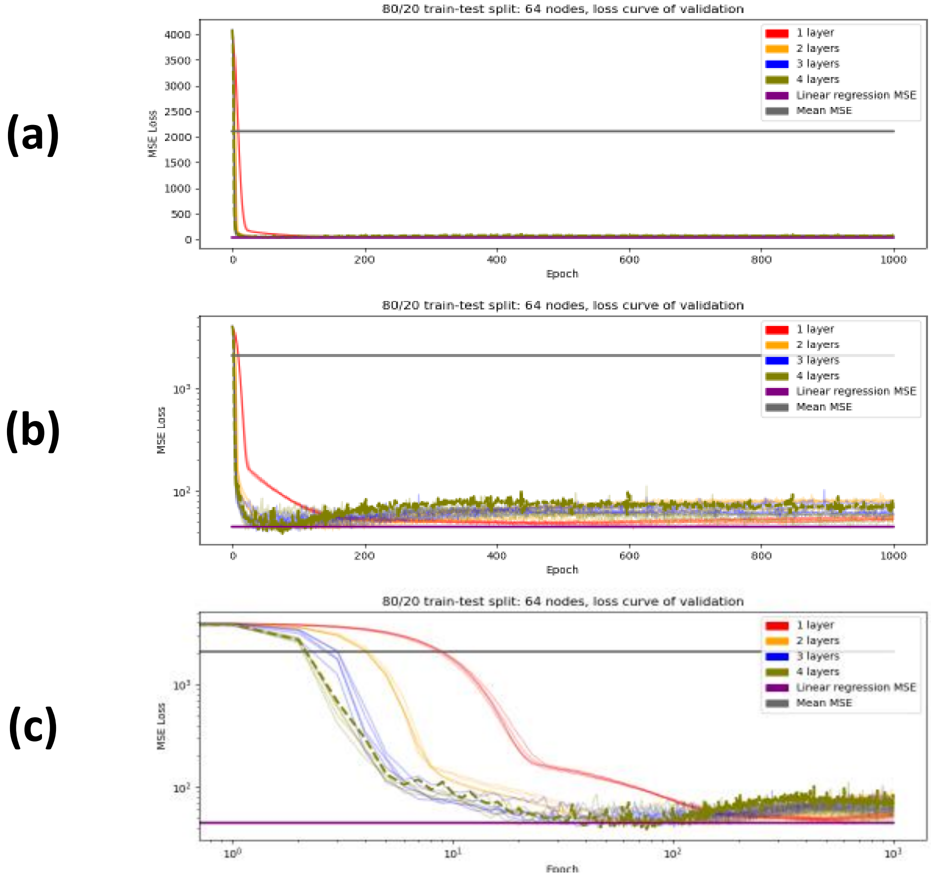


Figure 6.10: MSE validation loss of five fits using pad granularity inputs with 64 nodes per a hidden layer. (a) Epochs and loss on linear scale. (b) Epochs on linear scale and loss on logarithmic scale. (c) Epochs and loss on logarithmic scale.

The validation loss curves from Figure 6.9 and 6.10 show that neural networks offer similar performance to the linear regression baseline, while outperforming the mean prediction.

II. Nominal range

Figure 6.11 and 6.12 show the validation loss curves for the nominal datasplit. Within Figure 6.11 and 6.12 subfigure (a) shows the epochs on a linear x-axis and loss on a linear y-axis, (b) shows the epochs on a linear x-axis and loss on a logarithmic y-axis, and (c) shows the epochs on a logarithmic x-axis and loss on a logarithmic y-axis. The model with the lowest validation loss is indicated by a dashed line.

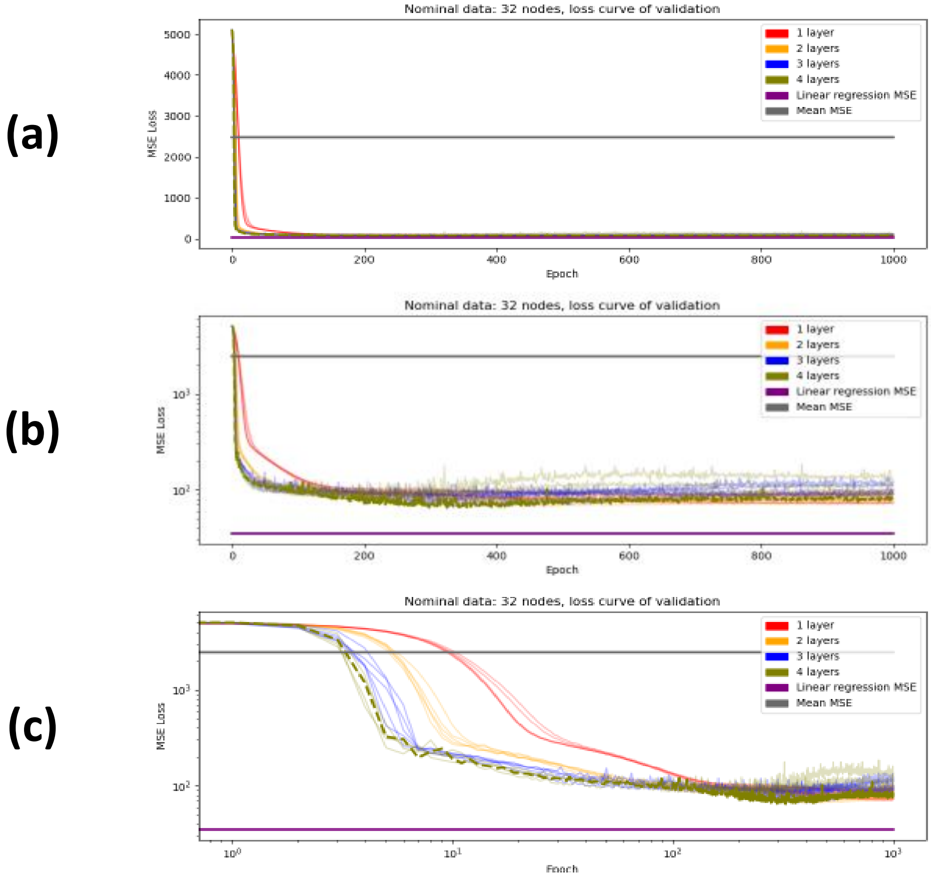


Figure 6.11: MSE validation loss of five fits using pad granularity inputs with 32 nodes per a hidden layer. (a) Epochs and loss on linear scale. (b) Epochs on linear scale and loss on logarithmic scale. (c) Epochs and loss on logarithmic scale.

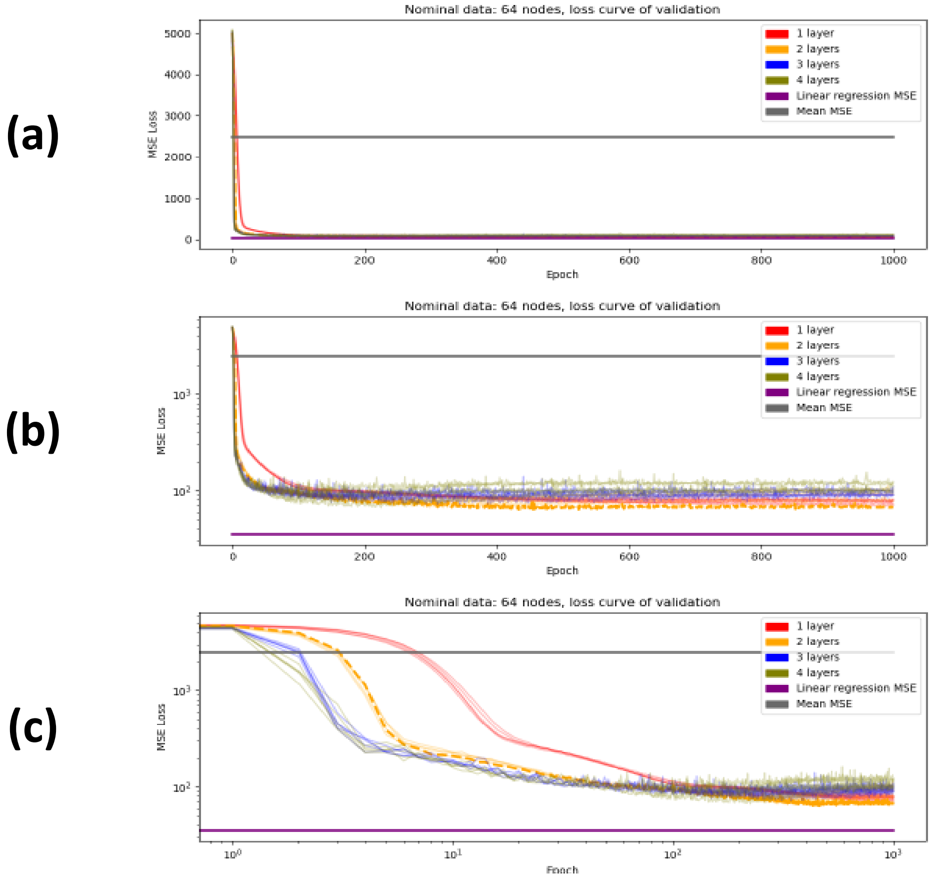


Figure 6.12: MSE validation loss of five fits using pad granularity inputs with 64 nodes per a hidden layer. (a) Epochs and loss on linear scale. (b) Epochs on linear scale and loss on logarithmic scale. (c) Epochs and loss on logarithmic scale.

In Figure 6.11 and 6.12 the validation loss curves show that the neural networks offer improvement with respect to the mean prediction, but not with respect to the baseline.

III. On all data

In Figure 6.13 and 6.14 the validation loss curves for the 3σ datasplit are shown. Within each figure, subfigures (a), (b) and (c) show the same loss curves, albeit on different scale. Subfigure (a) shows the epochs on a linear x-axis and loss on a linear y-axis, (b) shows the epochs on a linear x-axis and loss on a logarithmic y-axis, and (c) shows the epochs on a logarithmic x-axis and loss on a logarithmic y-axis. In Figure 6.13 and 6.14 the model with the lowest validation loss is indicated by a dashed line.

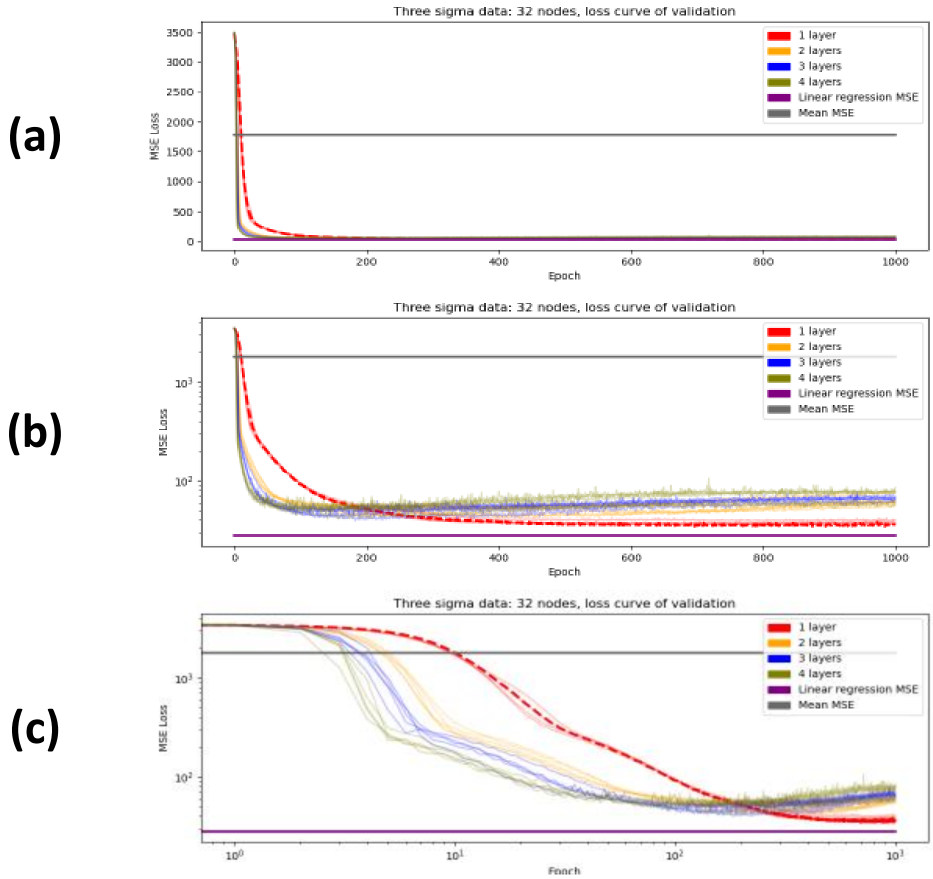


Figure 6.13: MSE validation loss of five fits using pad granularity inputs with 32 nodes per a hidden layer. (a) Epochs and loss on linear scale. (b) Epochs on linear scale and loss on logarithmic scale. (c) Epochs and loss on logarithmic scale.

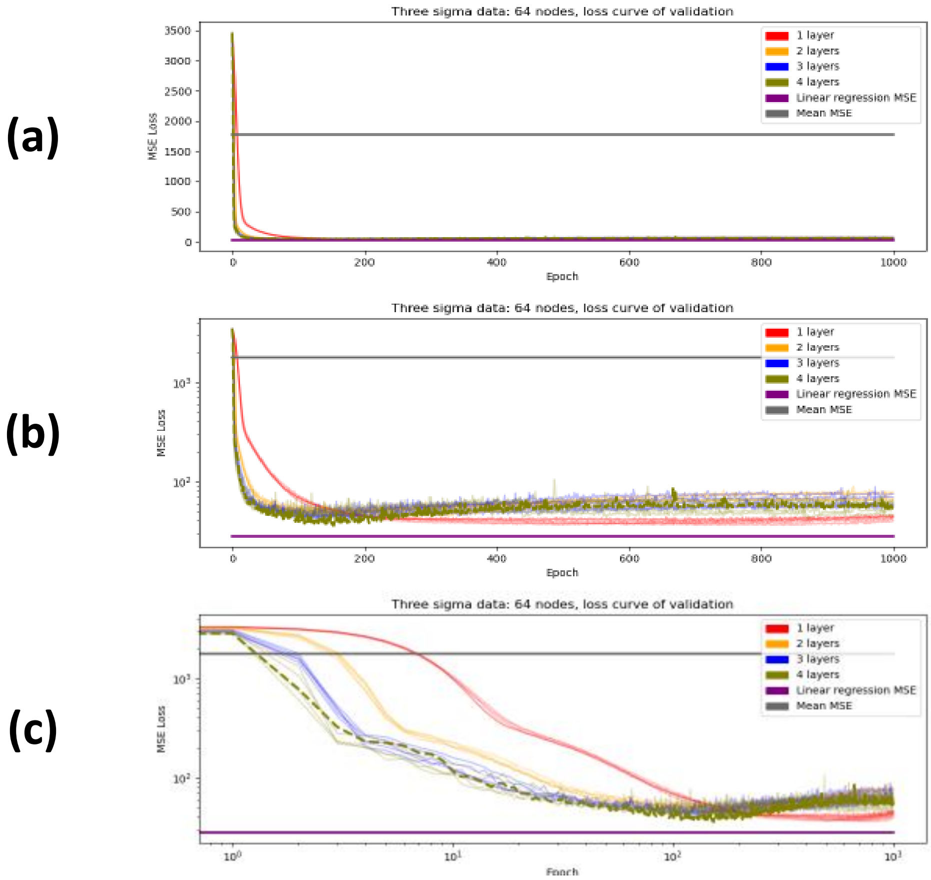


Figure 6.14: MSE validation loss of five fits using pad granularity inputs with 64 nodes per a hidden layer. (a) Epochs and loss on linear scale. (b) Epochs on linear scale and loss on logarithmic scale. (c) Epochs and loss on logarithmic scale.

From Figure 6.13 and 6.14 we observe that neural networks offer similar performance to the linear regression baseline, while showing a big improvement over the mean prediction.

6.1.3 SEM granularity

For the SEM granularity the same approach is used as for pad granularity: Five model fits are being performed with learning rate 0.001, 1000 epochs and 1, 2, 3, 4 hidden layers at 32 and 64 nodes per hidden layer. The training loss curves are shown in Appendix A. The validation loss curves for 32 and 64 nodes per hidden layer are shown below.

I. 80%/20% random data split

In Figure 6.15 and 6.16 the MSE loss curves for validation on the 80%/20% datasplit are shown. Within Figure 6.15 and 6.16, subfigure (a) shows the epochs on a linear x-axis and loss on a linear y-axis, (b) shows the epochs on a linear x-axis and loss on a logarithmic y-axis, and (c) shows the epochs on a logarithmic x-axis and loss on a logarithmic y-axis. In Figure 6.15 and 6.16 the model with the lowest validation loss is indicated by a dashed line.

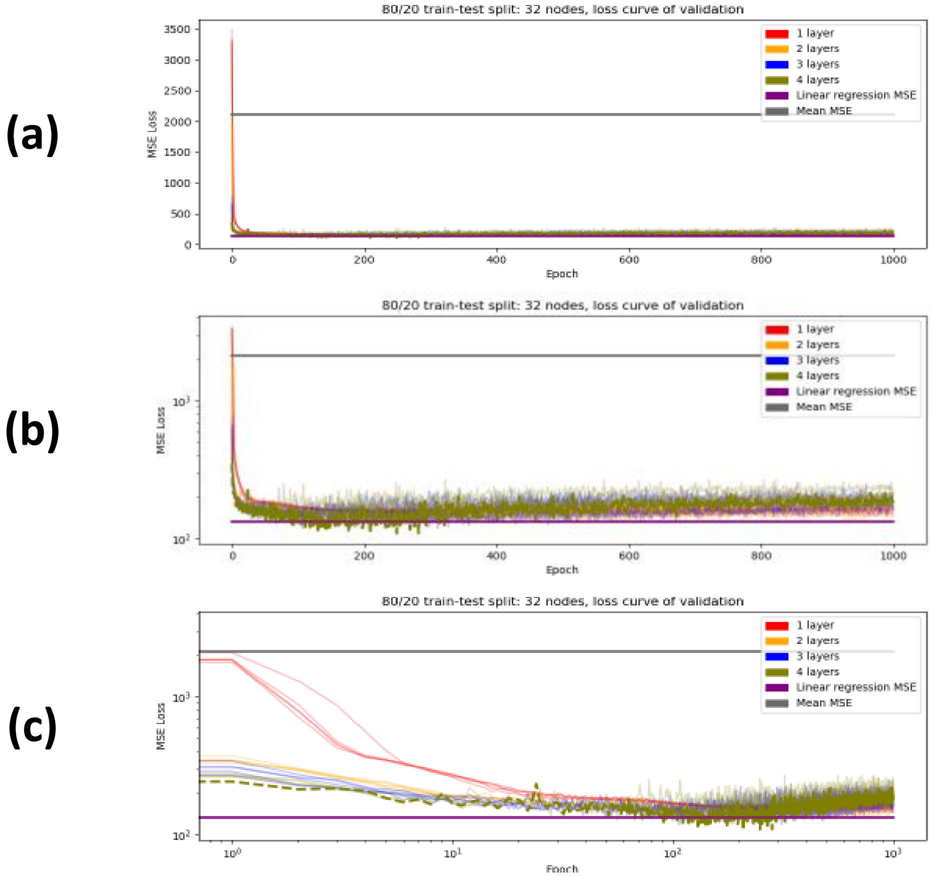


Figure 6.15: MSE validation loss of five fits using SEM granularity inputs with 32 nodes per a hidden layer. (a) Epochs and loss on linear scale. (b) Epochs on linear scale and loss on logarithmic scale. (c) Epochs and loss on logarithmic scale.

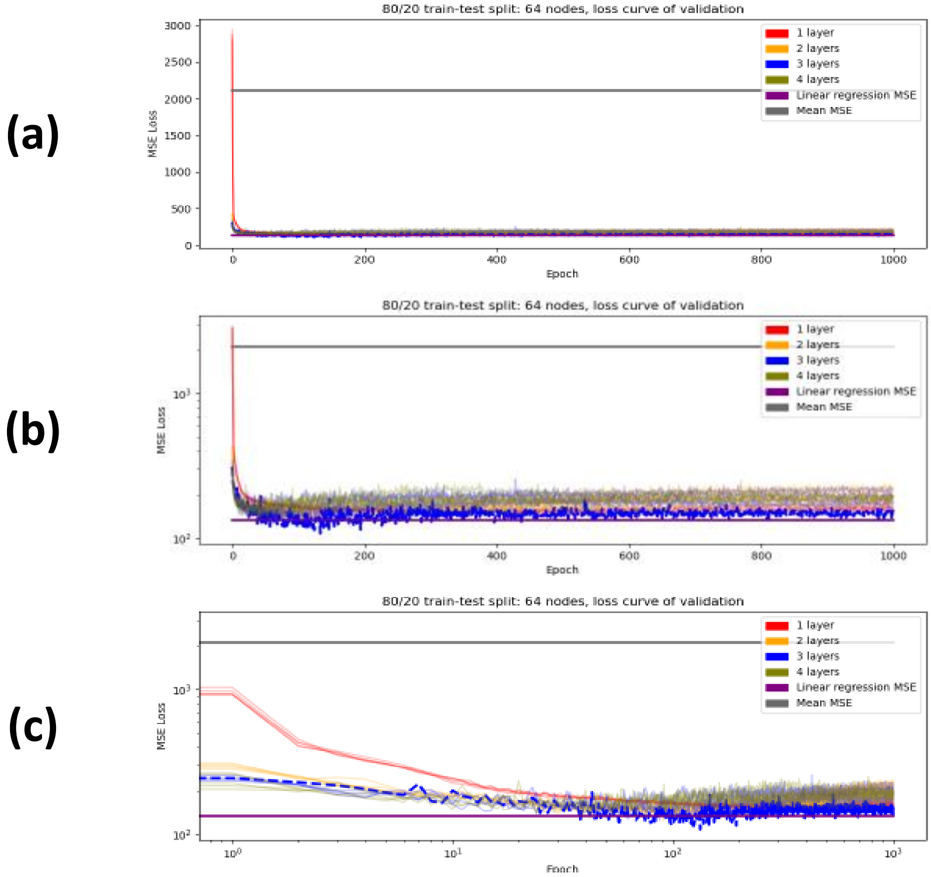


Figure 6.16: MSE validation loss of five fits using SEM granularity inputs with 64 nodes per a hidden layer. (a) Epochs and loss on linear scale. (b) Epochs on linear scale and loss on logarithmic scale. (c) Epochs and loss on logarithmic scale.

From Figure 6.15 and 6.16 we observe that neural networks performs much better than the mean prediction, but offer similar performance to the linear regression baseline.

II. Nominal range

In Figure 6.17 and 6.18 the validation loss curves for the nominal datasplit are shown where subfigure (a) shows the epochs on a linear x-axis and loss on a linear y-axis, (b) shows the epochs on a linear x-axis and loss on a logarithmic y-axis, and (c) shows the epochs on a logarithmic x-axis and loss on a logarithmic y-axis. The model with the lowest validation loss is indicated by a dashed line.

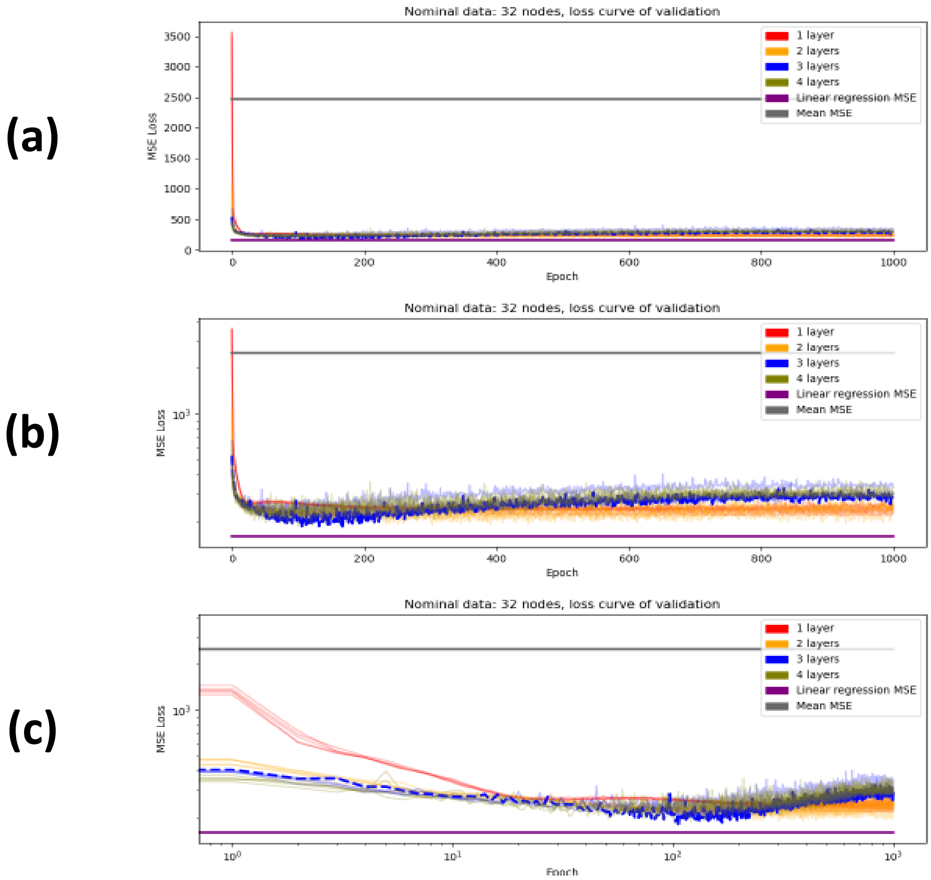


Figure 6.17: MSE validation loss of five fits using SEM granularity inputs with 32 nodes per a hidden layer. (a) Epochs and loss on linear scale. (b) Epochs on linear scale and loss on logarithmic scale. (c) Epochs and loss on logarithmic scale.

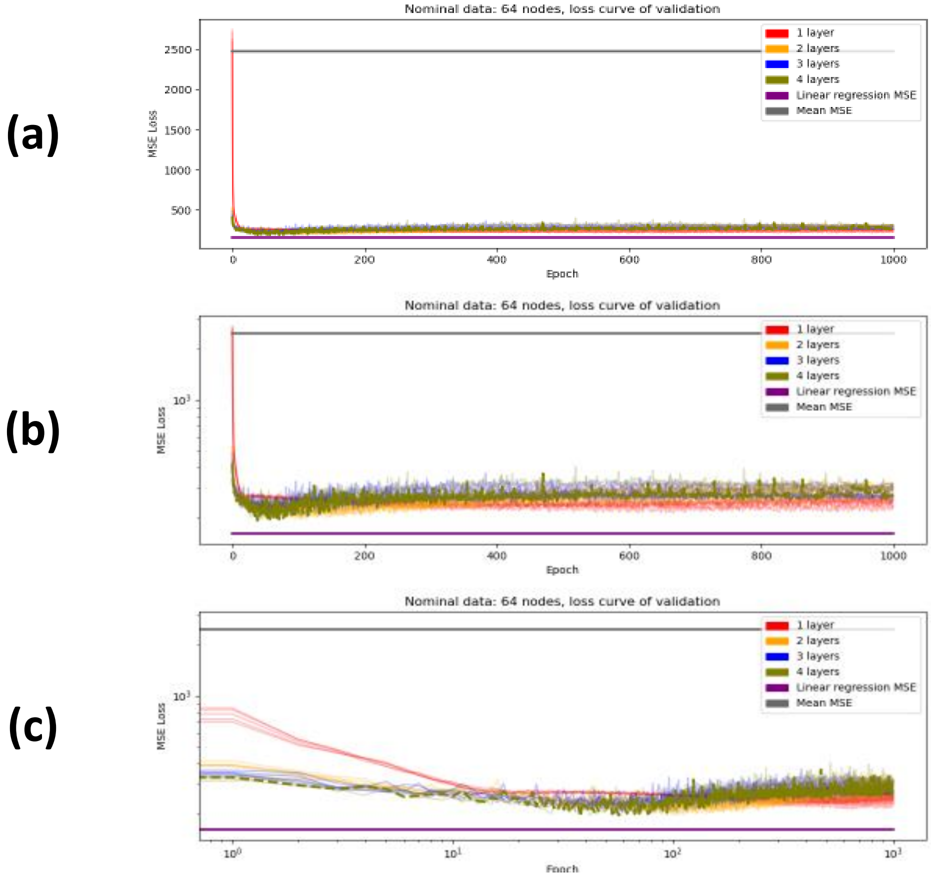


Figure 6.18: MSE validation loss of five fits using SEM granularity inputs with 64 nodes per a hidden layer. (a) Epochs and loss on linear scale. (b) Epochs on linear scale and loss on logarithmic scale. (c) Epochs and loss on logarithmic scale.

Figure 6.17 and 6.18 show that neural networks give much better performance than the mean prediction, but gives worse performance than the baseline.

III. On all data

In Figure 6.19 and 6.20 the validation loss curves for the 3σ datasplit are shown. Within each figure, subfigure (a) shows the epochs on a linear x-axis and loss on a linear y-axis, (b) shows the epochs on a linear x-axis and loss on a logarithmic y-axis, and (c) shows the epochs on a logarithmic x-axis and loss on a logarithmic y-axis. The model with the lowest validation loss is indicated by a dashed line.

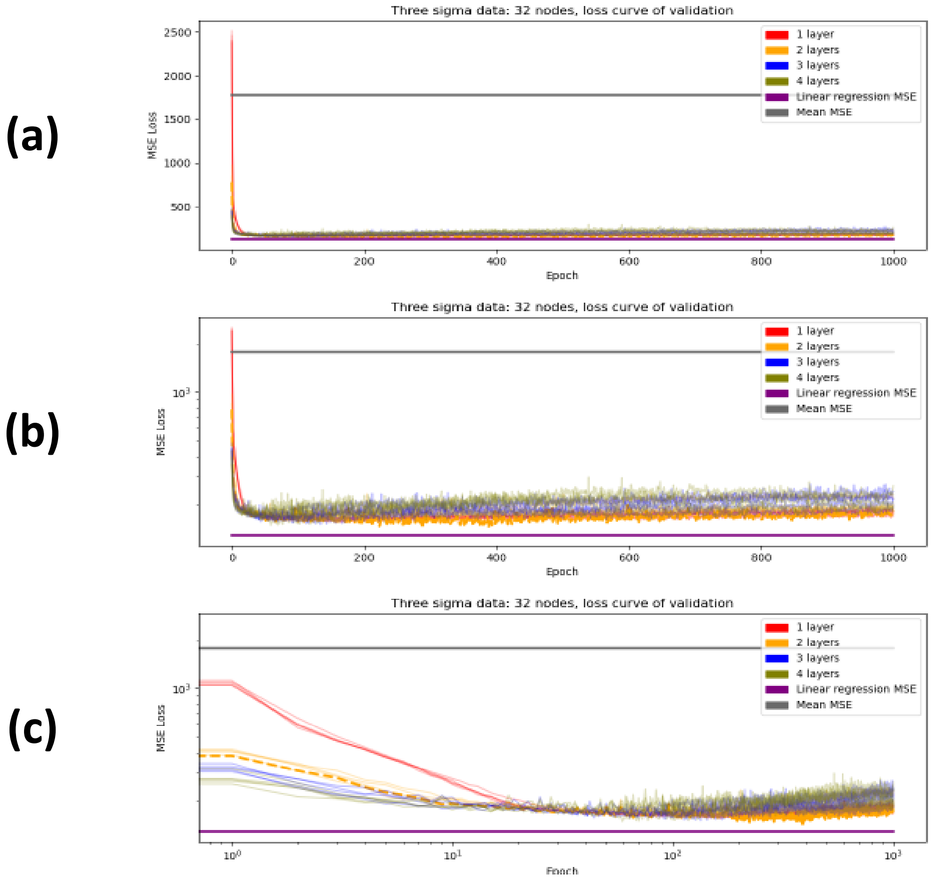


Figure 6.19: MSE validation loss of five fits using SEM granularity inputs with 32 nodes per a hidden layer. (a) Epochs and loss on linear scale. (b) Epochs on linear scale and loss on logarithmic scale. (c) Epochs and loss on logarithmic scale.

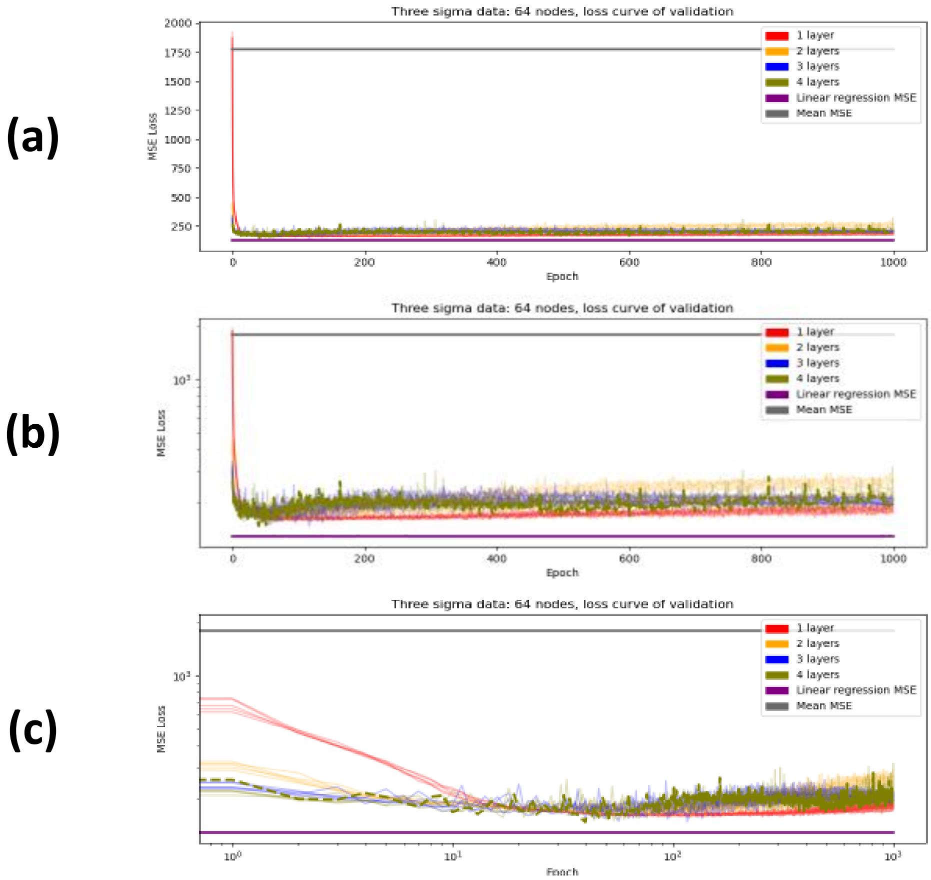


Figure 6.20: MSE validation loss of five fits using SEM granularity inputs with 64 nodes per a hidden layer. (a) Epochs and loss on linear scale. (b) Epochs on linear scale and loss on logarithmic scale. (c) Epochs and loss on logarithmic scale.

The validation loss curves from Figure 6.19 and 6.20 show that the neural networks show better performance than the mean prediction, but worse performance than the linear regression baseline.

6.1.4 Discussion

If the training loss curve would converge, then the learning process would be (almost) finished. This would switch our concerns of potential underfitting to potential overfitting. If the training loss curves together with the validation loss curves keeps improving and does not converge, then it would indicate that the model is likely to be underfitting as more learning would be beneficial. We will discuss our findings from Experiment (A) by investigating the training and validation loss curves to address the underfitting problem.

In section 6.1.1, at field granularity the training loss curves for 32 and 64 nodes per hidden layer for the random 80%/20% split in Figure 6.1 and 6.3 are shown. The figures show that the performance on the training set keeps improving as the MSE loss continues to follow a downward trend. The continuous improvement indicates that the learning process has not yet finished. Further can be observed that on the training set with the given amount of epochs models with 2, 3 and 4 hidden layers perform better than simply predicting the mean and the linear regression baseline. Although models with 1 hidden layer also perform much better than the mean prediction, it does not perform better than the linear regression baseline. Similar behaviour can be observed in the loss curves of the 32 and 64 nodes per hidden layer on the nominal datasplit in Appendix A Figure 1 and 2 and the 32 and 64 nodes per hidden layer on all the data in Appendix A Figure 3 and 4. Given that the training loss curves of these models are still descending, the models might perform better than the baseline if we use a larger training period.

In Figure 6.2(c), 6.5(c) and 6.7(c) we observe that for the neural networks with 32 nodes per hidden layers all validation curves at field granularity converge for the 80%/20%, nominal datasplits and all data. This can be noticed by the loss curves descending to a minimum followed by an upward trend. This implies that the neural networks have learned sufficiently and that there is no underfitting. This similarly holds for models with 64 nodes per hidden layer in Figure 6.4(c), 6.6(c) and 6.8(c).

From the training and validation loss curves we observe that neural networks can learn sufficiently for the given amount of epochs. Although all the validation figures have shown that the validation curves have converged at all granularity levels, there is still a possibility that with other hyperparameter settings underfitting still takes place, i.e. the validation curves are not converging. For this uncertainty we add extra epochs for training in Experiment (B) such that maximum amount of training epochs becomes 2500, 1500 and 1500 for field, pad and SEM granularity level respectively. With this, potential underfitting has been addressed. By preventing underfitting the validation loss curves show a continuous increase in loss, which indicates potential overfitting. However, this problem can be avoided through earlystopping, which stops the training once the validation curves do not show improvement.

The training and validation loss curves also give some insight about the results that might come forward in Experiment (B) section 6.2. All the validation curves show a big improvement over the mean prediction on all granularity levels and datasplits, but not for the linear regression baseline. In particular for the nominal datasplits in Figure 6.5, 6.6, 6.11, 6.12, 6.17

and 6.18 at all different granularity levels we see that the neural networks give worse performance than the linear baseline, which could reflect in the results on the test set in Experiment (B). Further can be observed that for within each datasplit and each granularity levels the validation loss curves have similar shapes. This implies that for 32 and 64 nodes per hidden layer there is little performance difference, albeit that the number of hidden layers might change for the best model.

6.2 Experiment B: Focus prediction based on orthogonal groups

6.2.1 Feedforward Neural Network

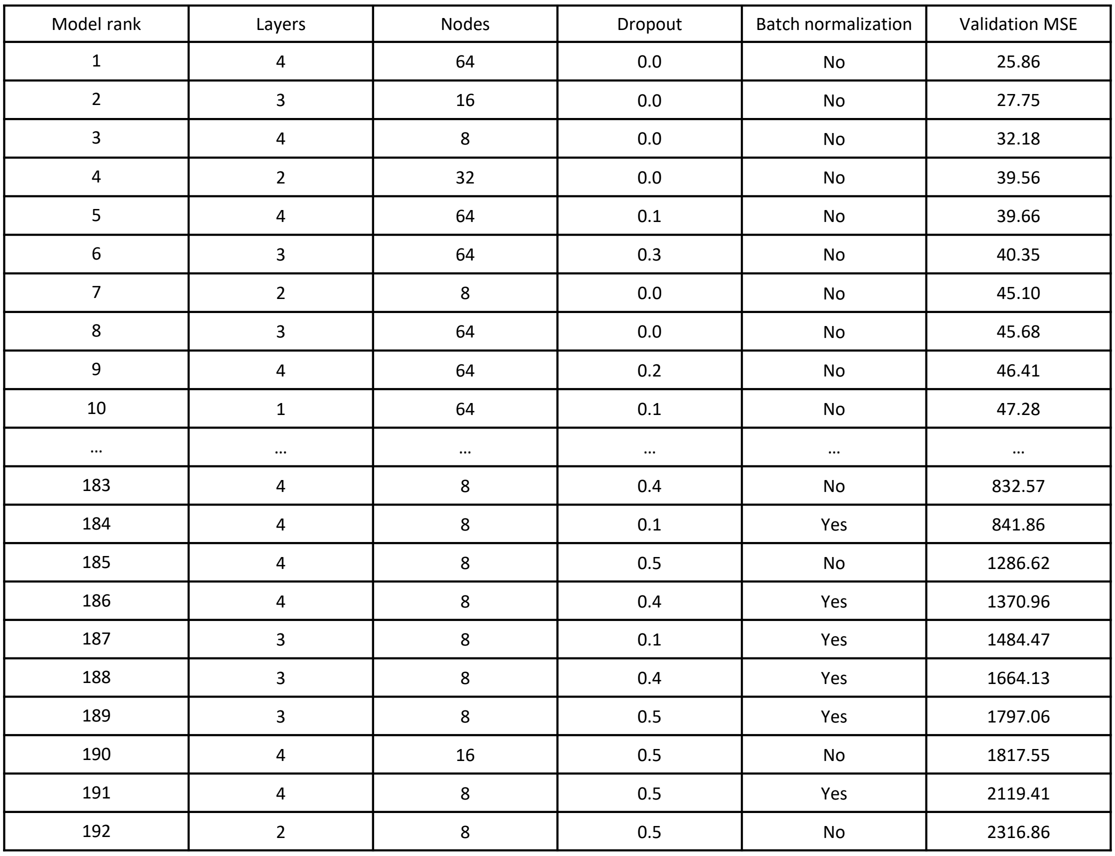
Following the 192 hyperparameter combinations defined by Table 4.13 a model is fitted for each hyperparameter combination. Through Experiment (A) we have set the amount of epochs for training at 2500 epochs at field granularity and 1500 epochs for pad and SEM granularity. The model with the lowest MSE loss will be considered as the best model, which will be used on the test set.

The MSE loss and loss curves on the validation set of the 10 best and 10 worst are given for each datasplit and granularity level. The most important loss curves of the 10 best models are given in the following sections, but not all the loss curves of the 10 worst models per setup are shown

Field granularity

At field granularity level models are trained upto 2500 epochs with earlystopping applied, learning rate 0.001 and the Adam optimizer. In Table 6.1 are the 10 best and worst models given with their hyperparameters and lowest recorded MSE loss. In Figure 6.21 the MSE validation loss of the 10 best models are shown, where Figure 6.21(a), (b) and (c) show the same data, but on different scales to properly see if loss curves converge. Figure 6.21(a) shows the epochs on a linear x-axis and loss on a linear y-axis, (b) shows the epochs on a linear x-axis and loss on a logarithmic y-axis, and (c) shows the epochs on a logarithmic x-axis and loss on a logarithmic y-axis.

I. 80%/20% random data split



Model rank	Layers	Nodes	Dropout	Batch normalization	Validation MSE
1	4	64	0.0	No	25.86
2	3	16	0.0	No	27.75
3	4	8	0.0	No	32.18
4	2	32	0.0	No	39.56
5	4	64	0.1	No	39.66
6	3	64	0.3	No	40.35
7	2	8	0.0	No	45.10
8	3	64	0.0	No	45.68
9	4	64	0.2	No	46.41
10	1	64	0.1	No	47.28
…	…	…	…	…	…
183	4	8	0.4	No	832.57
184	4	8	0.1	Yes	841.86
185	4	8	0.5	No	1286.62
186	4	8	0.4	Yes	1370.96
187	3	8	0.1	Yes	1484.47
188	3	8	0.4	Yes	1664.13
189	3	8	0.5	Yes	1797.06
190	4	16	0.5	No	1817.55
191	4	8	0.5	Yes	2119.41
192	2	8	0.5	No	2316.86

Table 6.1: MSE loss of the 10 best and worst models with field instances on the validation set of the 80%/20% datasplit

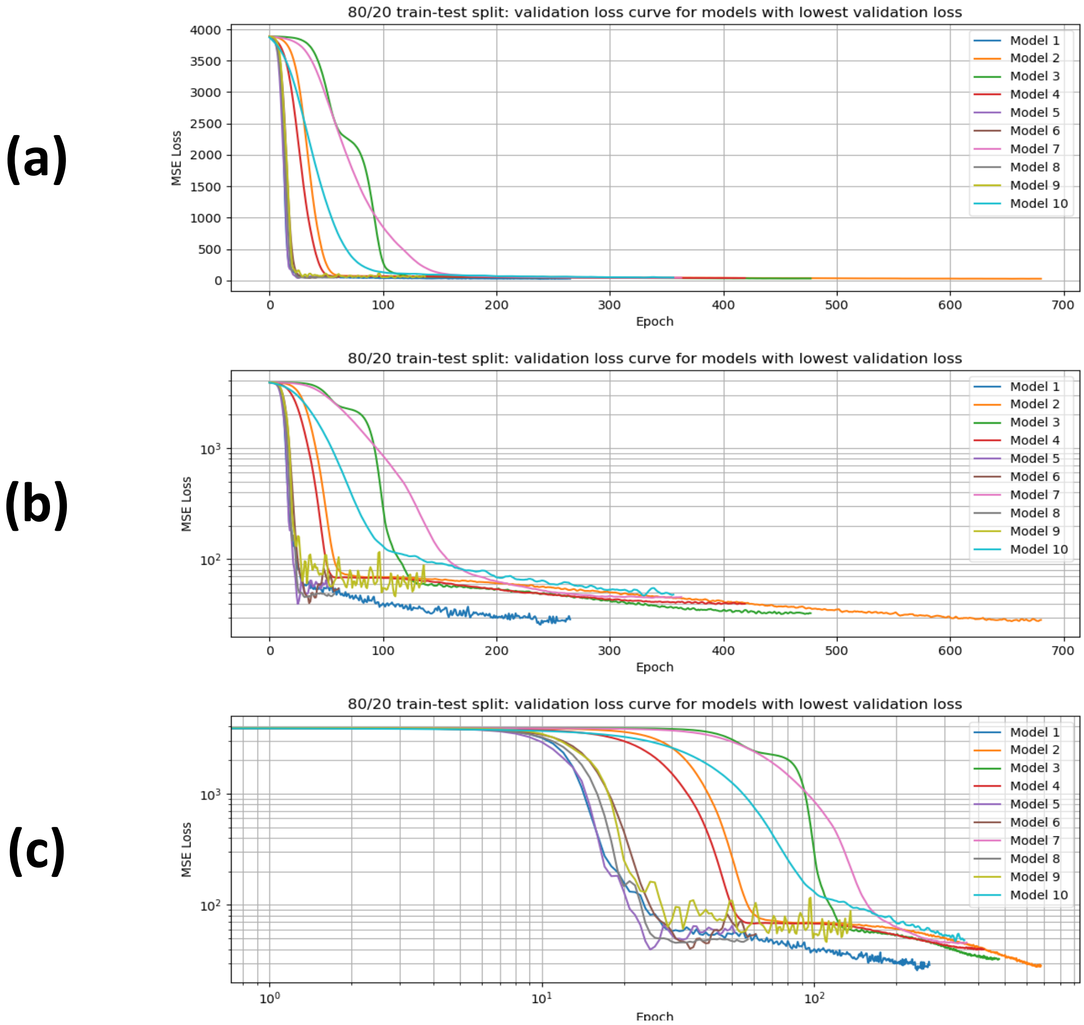


Figure 6.21: MSE validation loss of the 10 best models using field granularity inputs on the 80%/20% datasplit. (a) Epochs and loss on linear scale. (b) Epochs on linear scale and loss on logarithmic scale. (c) Epochs and loss on logarithmic scale.

The validation loss curves of the 10 best models on the 80%/20% datasplit are shown in Figure 6.21. The model numbers in the legends correspond with the model rank in Table 6.1. From Table 6.1 we observe that the 10 best models consists of a large amount of nodes per layer, low dropout rate and without batch normalization. The 10 worst models consists mostly of high dropout rates and a small amount of nodes per layer. From Figure 6.21(b) and (c) can be observed that for model 1 the loss fluctuates slightly, while for model 5 and 9 the loss curves are fluctuating more with lower model rank.

The models have the same hyperparameters, but differ in dropout rate. From these three models we observe that increasing dropout rates give an increase in MSE loss and fluctuations.

II. Nominal range

In Table 6.2 are the 10 best and worst models given with their hyperparameters and lowest recorded MSE loss. In Figure 6.22 the MSE validation loss of the 10 best models are shown, where Figure 6.22(a)(b) and (c) show the same data, but on different scales to properly see if loss curves converge. Figure 6.22(a) shows the epochs on a linear x-axis and loss on a linear y-axis, (b) shows the epochs on a linear x-axis and loss on a logarithmic y-axis, and (c) shows the epochs on a logarithmic x-axis and loss on a logarithmic y-axis. Similarly, the MSE validation loss of the 10 worst models on the nominal focus datasplit are shown in Figure 6.23 with similar scaled axes for (a), (b) and (c).

Model rank	Layers	Nodes	Dropout	Batch normalization	Validation MSE
1	4	32	0.0	No	36.23
2	4	16	0.0	No	44.37
3	2	16	0.0	No	45.63
4	2	32	0.0	No	48.18
5	3	8	0.0	No	50.72
6	3	64	0.1	No	51.49
7	2	64	0.0	No	51.68
8	3	64	0.0	No	54.93
9	3	32	0.0	No	57.10
10	4	64	0.0	No	57.12
…	…	…	…	…	…
183	3	8	0.5	Yes	1107.27
184	4	8	0.4	Yes	1178.54
185	4	8	0.4	No	1398.42
186	4	8	0.0	Yes	1636.56
187	4	8	0.5	No	1649.02
188	4	8	0.5	Yes	1718.59
189	4	8	0.1	Yes	1812.57
190	4	8	0.3	No	2243.73
191	4	8	0.3	Yes	2258.66
192	2	8	0.0	No	2674.44

Table 6.2: MSE loss of the 10 best and worst models with field instances on the validation set of the nominal datasplit

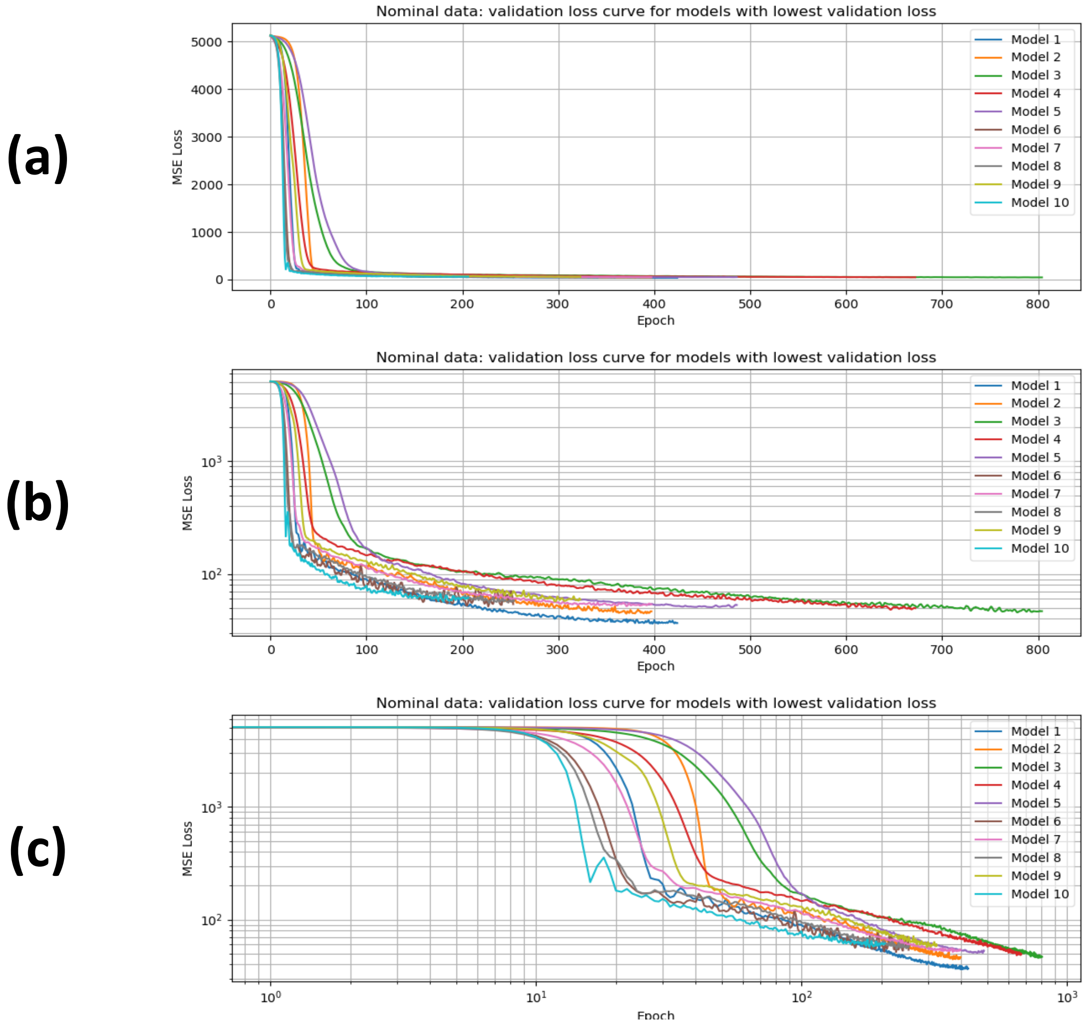


Figure 6.22: MSE validation loss of the 10 best models using field granularity inputs on the nominal datasplit. (a) Epochs and loss on linear scale. (b) Epochs on linear scale and loss on logarithmic scale. (c) Epochs and loss on logarithmic scale.

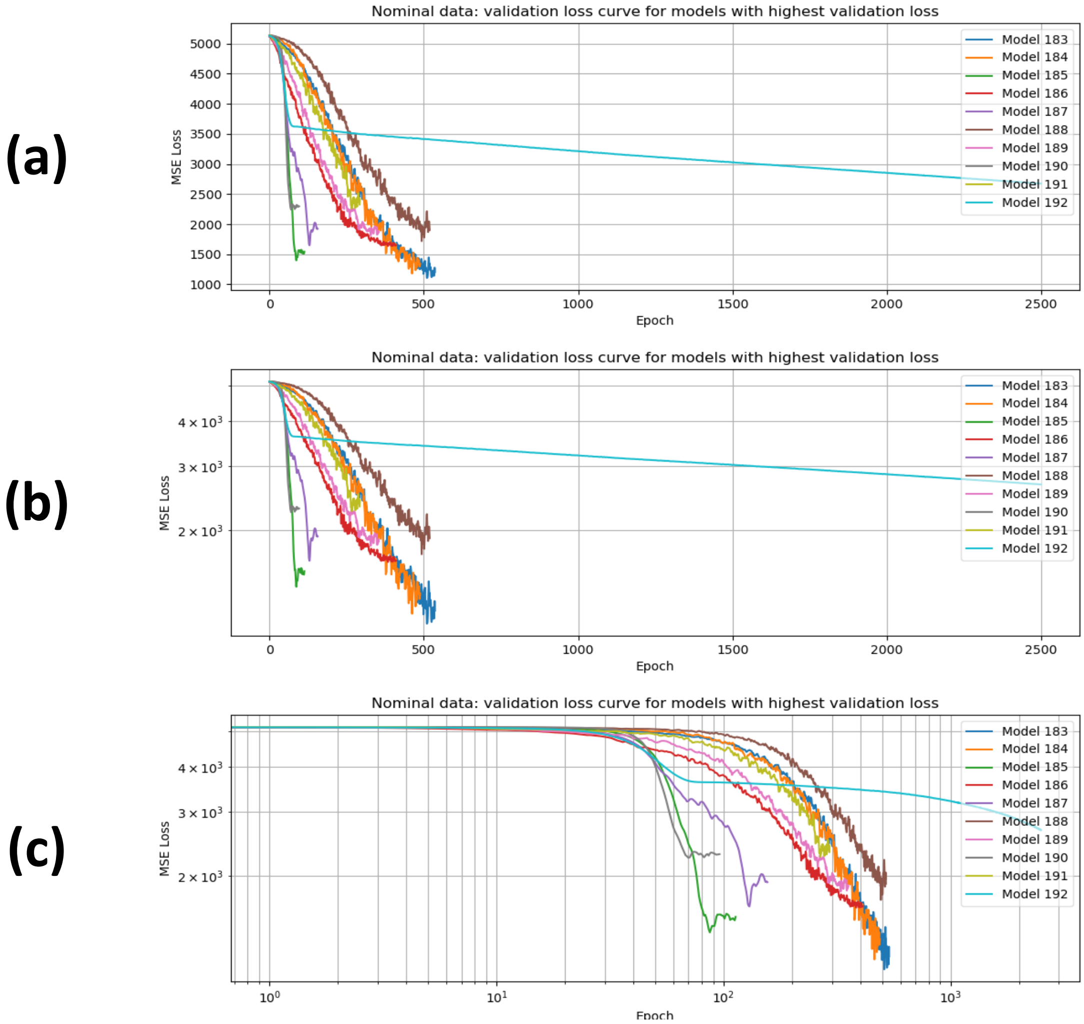


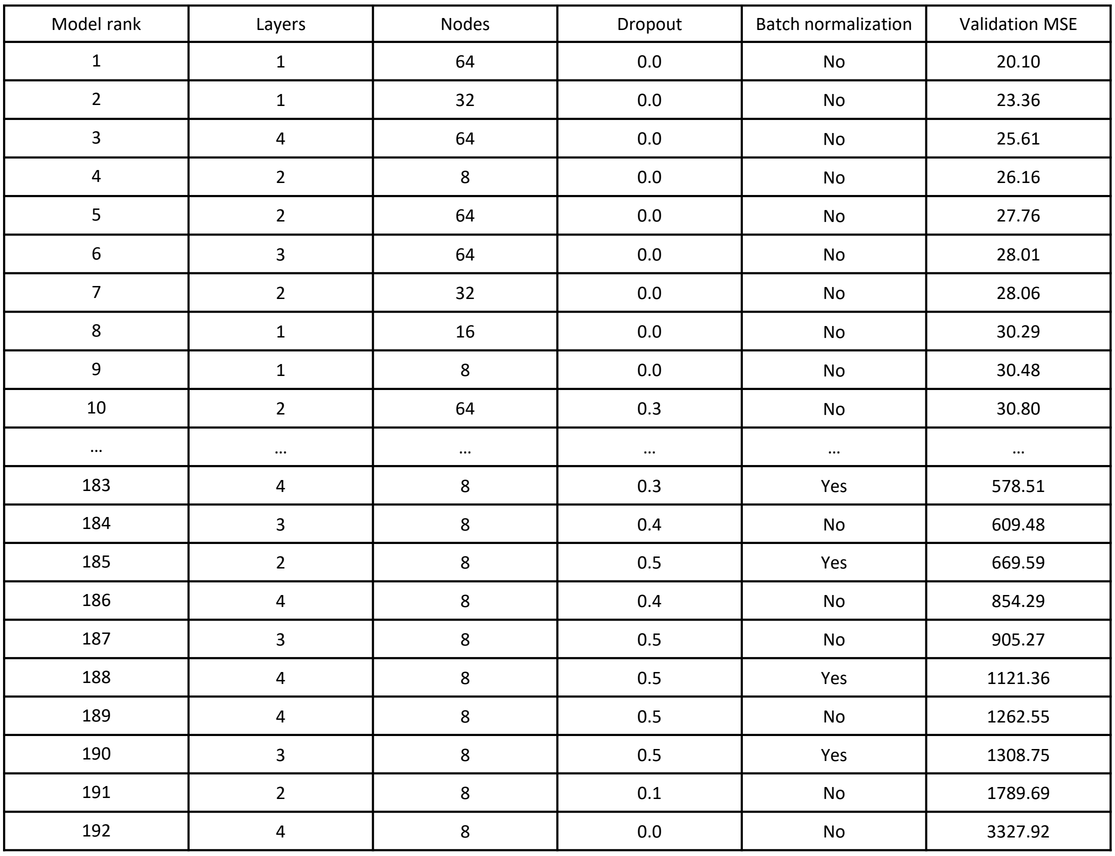
Figure 6.23: MSE validation loss of the 10 worst models using field granularity inputs on the nominal datasplit. (a) Epochs and loss on linear scale. (b) Epochs on linear scale and loss on logarithmic scale. (c) Epochs and loss on logarithmic scale.

The validation loss curves are shown in Figure 6.22 and 6.23. The model numbers in the legends correspond with the model rank in Table 6.2. From Table 6.2 we observe that the 10 best models use a low dropout rate and no batch normalization. The 10 worst models consists mostly of high dropout rates, many layers and small amount of nodes per layer. Figure 6.22(b) and (c) show that all the 10 best models converge as the MSE loss does no longer show any improvement after a specific amount of epochs following earlystopping, i.e. the loss curves do not show a positive trend. We further observe that the loss curves of models 3, 4 and 7 need a relatively large amount of epochs. This is likely due to the relatively low model complexity made up of 2 hidden layers, while the other models in the top 10 contain more hidden layers.

Note that in Figure 6.23 the worst model is using the total amount of epochs and is not converging. The model with the same hyperparameters but with 16 nodes per layer is ranked as the third best model and does converge, although it uses a relatively big amount of epochs compared to other good models. This shows that similar models converge slowly, but can perform well, so ideally the worst model should be given a longer training period, i.e. more epochs.

III. On all data

In Table 6.3 are the 10 best models given with their hyperparameters and lowest recorded MSE loss. In Figure 6.24 the MSE validation loss of the 10 best models are shown, where Figure 6.24(a)(b) and (c) show the same data, but on different scales to properly see if loss curves converge. Figure 6.24(a) shows the epochs on a linear x-axis and loss on a linear y-axis, (b) shows the epochs on a linear x-axis and loss on a logarithmic y-axis, and (c) shows the epochs on a logarithmic x-axis and loss on a logarithmic y-axis.



Model rank	Layers	Nodes	Dropout	Batch normalization	Validation MSE
1	1	64	0.0	No	20.10
2	1	32	0.0	No	23.36
3	4	64	0.0	No	25.61
4	2	8	0.0	No	26.16
5	2	64	0.0	No	27.76
6	3	64	0.0	No	28.01
7	2	32	0.0	No	28.06
8	1	16	0.0	No	30.29
9	1	8	0.0	No	30.48
10	2	64	0.3	No	30.80
…	…	…	…	…	…
183	4	8	0.3	Yes	578.51
184	3	8	0.4	No	609.48
185	2	8	0.5	Yes	669.59
186	4	8	0.4	No	854.29
187	3	8	0.5	No	905.27
188	4	8	0.5	Yes	1121.36
189	4	8	0.5	No	1262.55
190	3	8	0.5	Yes	1308.75
191	2	8	0.1	No	1789.69
192	4	8	0.0	No	3327.92

Table 6.3: MSE loss of the 10 best and worst models with field instances on the validation set of the 3σ datasplit

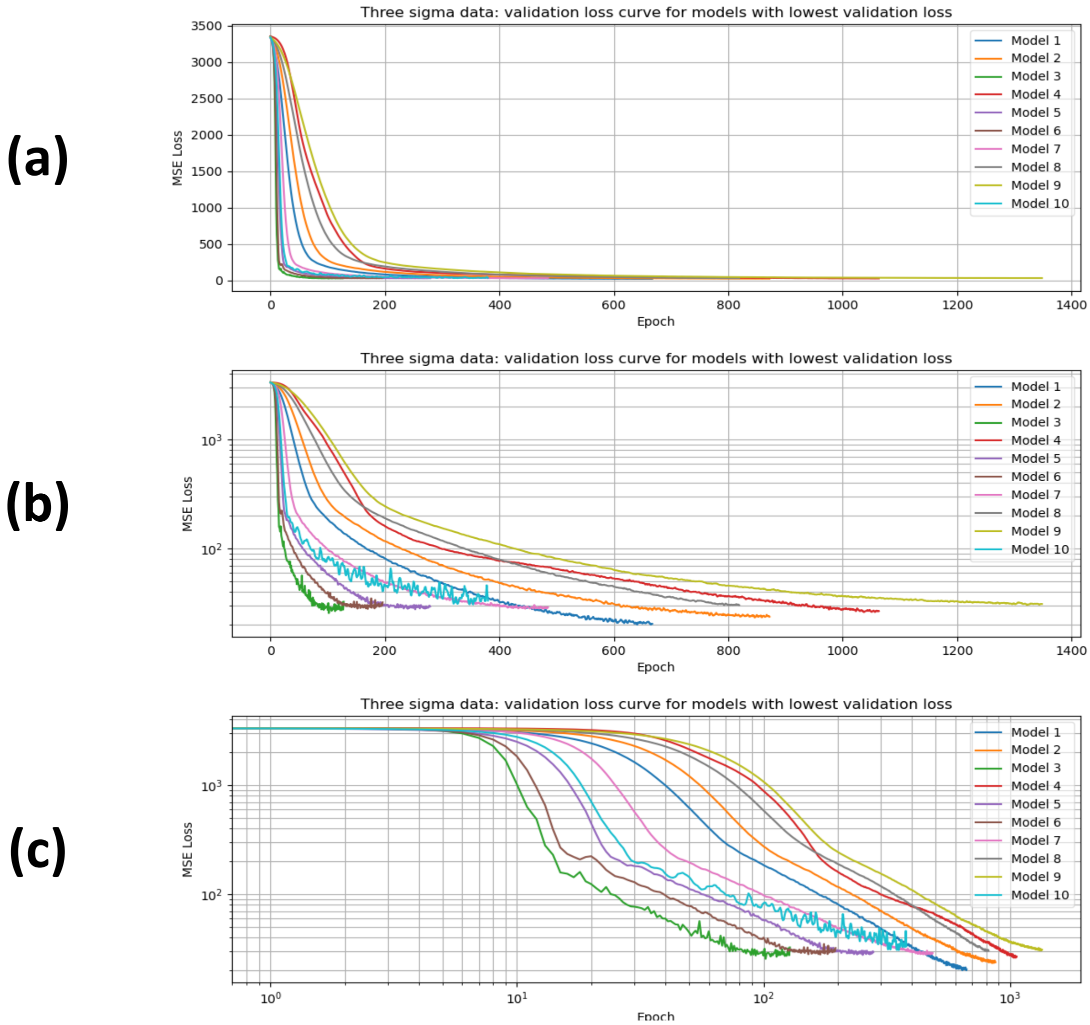


Figure 6.24: MSE validation loss of the 10 best models using field granularity inputs on the 3σ datasplit. (a) Epochs and loss on linear scale. (b) Epochs on linear scale and loss on logarithmic scale. (c) Epochs and loss on logarithmic scale.

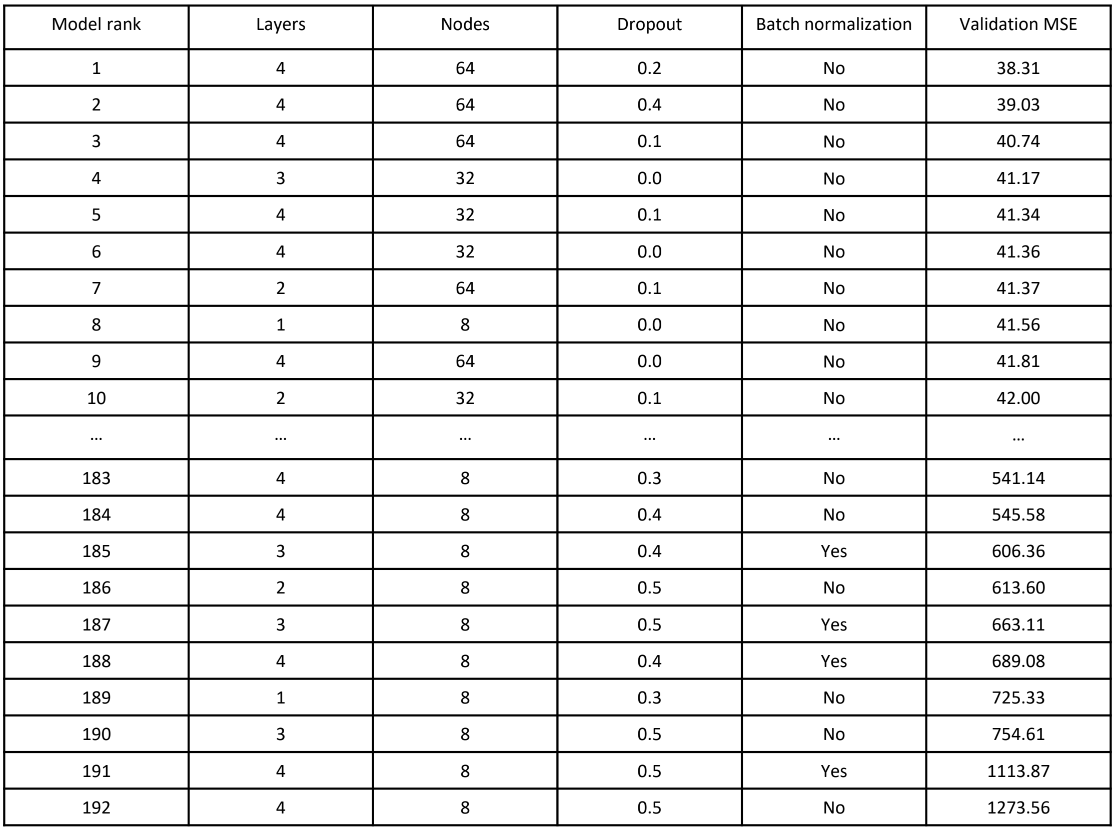
The validation loss curves are shown in Figure 6.24. The model numbers in the legends correspond with the model rank in Table 6.3. From Table 6.3 we observe that the 10 best models use a small amount of layers, a low dropout rate and no batch normalization. The 10 worst models mostly use high dropout rates and a small amount of nodes per layer. Figure 6.24(b) and (c) show that all the models converge with earlystopping activated. However, model 10 shows some fluctuations in the loss curve, likely due to the dropout rate of 0.3 while the other best performing models do not use dropout. Note that most of the best models only consists of one layer, which implies that relatively simple model is enough to capture the complexity in our data.

Pad granularity

At pad granularity level models are trained upto 1500 epochs with earlystopping applied, learning rate 0.001 and the Adam optimizer. In Appendix B are validation loss curves of the 10 best and worst models on each datasplit shown similar to the validation loss curves observed at field granularity level.

I. 80%/20% random data split

In Table 6.4 are the 10 best and worst models given with their hyperparameters and lowest recorded MSE loss. The validation loss curves of the 10 worst models on the 80%/20% datasplit are shown in Figure 6.25. In Figure 6.25 the MSE validation loss of the 10 worst models are shown, where Figure 6.25(a), (b) and (c) show the same data, but on different scales to properly see if loss curves converge. Figure 6.25(a) shows the epochs on a linear x-axis and loss on a linear y-axis, (b) shows the epochs on a linear x-axis and loss on a logarithmic y-axis, and (c) shows the epochs on a logarithmic x-axis and loss on a logarithmic y-axis. Similarly, the MSE validation loss of the 10 best models on the 80%/20% datasplit are shown in Appendix B Figure 20 with similar scaled axes for (a), (b) and (c).



Model rank	Layers	Nodes	Dropout	Batch normalization	Validation MSE
1	4	64	0.2	No	38.31
2	4	64	0.4	No	39.03
3	4	64	0.1	No	40.74
4	3	32	0.0	No	41.17
5	4	32	0.1	No	41.34
6	4	32	0.0	No	41.36
7	2	64	0.1	No	41.37
8	1	8	0.0	No	41.56
9	4	64	0.0	No	41.81
10	2	32	0.1	No	42.00
…	…	…	…	…	…
183	4	8	0.3	No	541.14
184	4	8	0.4	No	545.58
185	3	8	0.4	Yes	606.36
186	2	8	0.5	No	613.60
187	3	8	0.5	Yes	663.11
188	4	8	0.4	Yes	689.08
189	1	8	0.3	No	725.33
190	3	8	0.5	No	754.61
191	4	8	0.5	Yes	1113.87
192	4	8	0.5	No	1273.56

Table 6.4: MSE loss of the 10 best and worst models with pad instances on the validation set of the 80%/20% datasplit

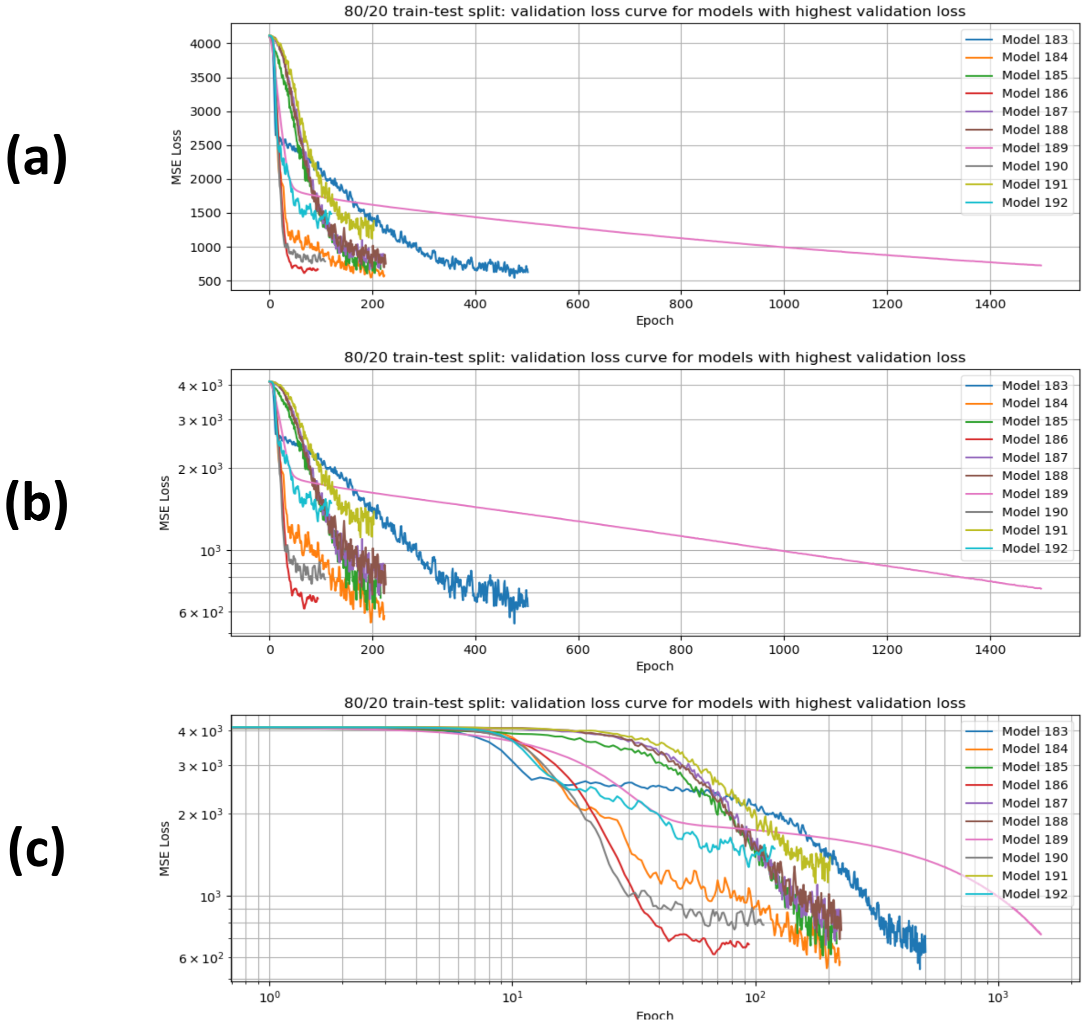


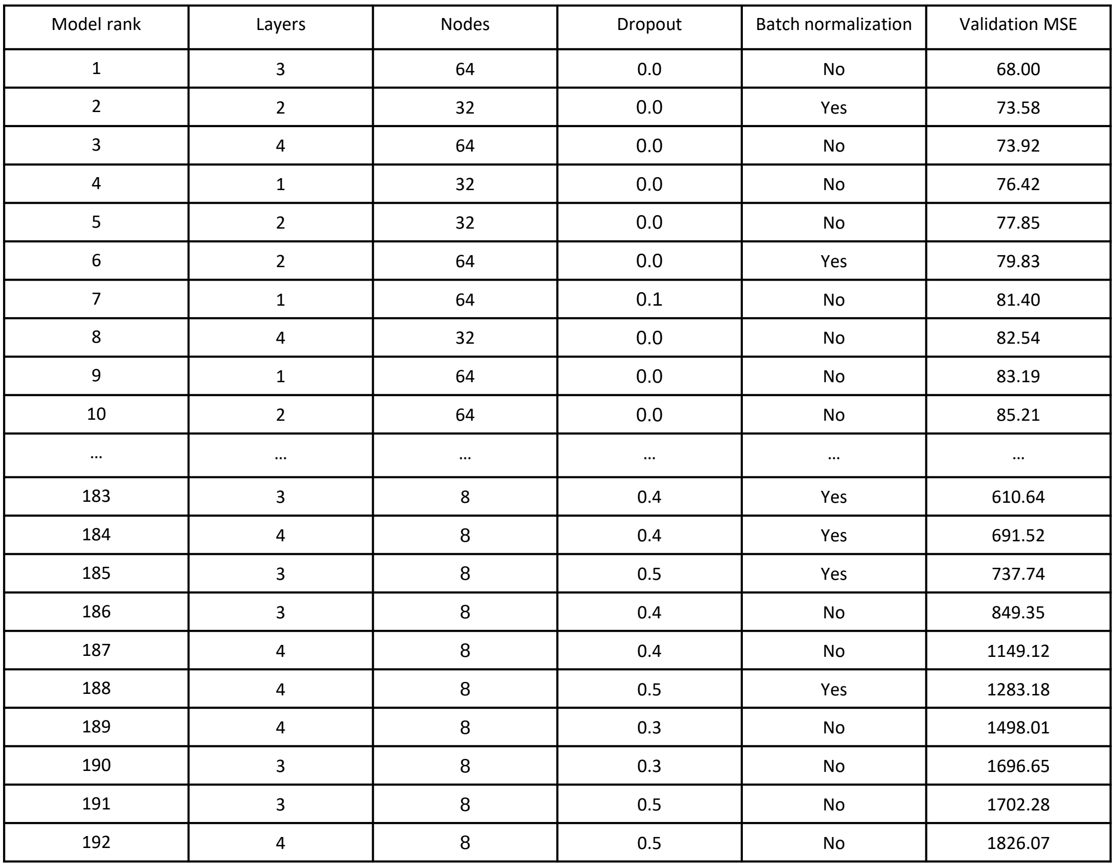
Figure 6.25: MSE validation loss of the 10 worst models using pad granularity inputs on the 80%/20% datasplit. (a) Epochs and loss on linear scale. (b) Epochs on linear scale and loss on logarithmic scale. (c) Epochs and loss on logarithmic scale.

The model numbers in the legend of each figure correspond with the model rank in Table 6.4. From Table 6.4 we observe that the 10 best models use many layers, many nodes per layer and no batch normalization. The 10 worst models mostly use high dropout rates and a small amount of nodes per layer.

From Figure 6.25 can be observed that the loss curve of model 189 does not converge as maximum amount of epochs for training has been reached. The model has the same amount of layers and nodes as model 8, but different dropout rate. Since model 189 has a similar model complexity as model 8, it might perform well when a higher amount of training epochs is applied.

II. Nominal range

In Table 6.5 are the 10 best and worst models given with their hyperparameters and lowest recorded MSE loss. In Appendix B Figure 21 the MSE validation loss of the 10 best models are shown, where Figure 21(a), (b) and (c) show the same data, but on different scales to properly see if loss curves converge. Figure 21(a) shows the epochs on a linear x-axis and loss on a linear y-axis, (b) shows the epochs on a linear x-axis and loss on a logarithmic y-axis, and (c) shows the epochs on a logarithmic x-axis and loss on a logarithmic y-axis. Similarly, the MSE validation loss of the 10 worst models on the nominal focus datasplit are shown in Appendix B Figure 22 with similar scaled axes for (a), (b) and (c). The model number in the legend of each figure correspond with the model rank from Table 6.5.



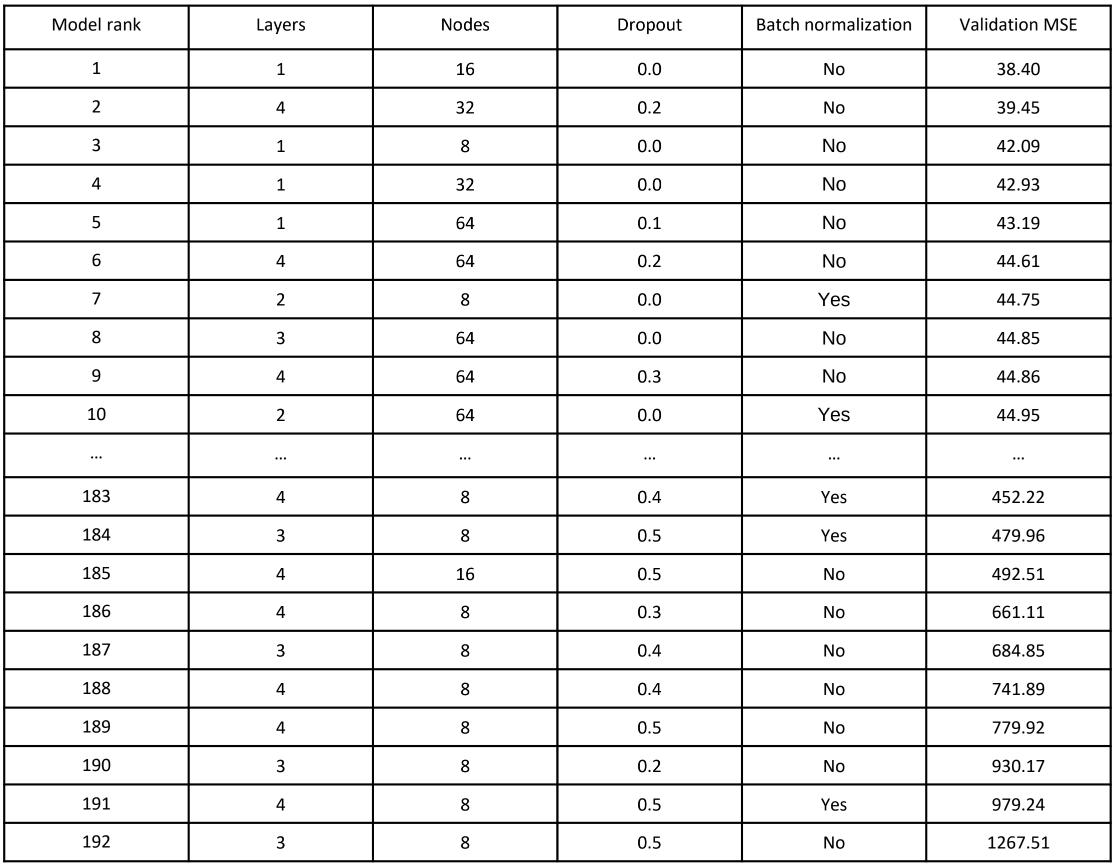
Model rank	Layers	Nodes	Dropout	Batch normalization	Validation MSE
1	3	64	0.0	No	68.00
2	2	32	0.0	Yes	73.58
3	4	64	0.0	No	73.92
4	1	32	0.0	No	76.42
5	2	32	0.0	No	77.85
6	2	64	0.0	Yes	79.83
7	1	64	0.1	No	81.40
8	4	32	0.0	No	82.54
9	1	64	0.0	No	83.19
10	2	64	0.0	No	85.21
…	…	…	…	…	…
183	3	8	0.4	Yes	610.64
184	4	8	0.4	Yes	691.52
185	3	8	0.5	Yes	737.74
186	3	8	0.4	No	849.35
187	4	8	0.4	No	1149.12
188	4	8	0.5	Yes	1283.18
189	4	8	0.3	No	1498.01
190	3	8	0.3	No	1696.65
191	3	8	0.5	No	1702.28
192	4	8	0.5	No	1826.07

Table 6.5: MSE loss of the 10 best and worst models with pad instances on the validation set of the nominal datasplit

From Table 6.5 we observe that the 10 best models use many nodes per layer, a low dropout rate and no batch normalization. The 10 worst models mostly use many layers with small amount of nodes per layer and high dropout rates.

III. On all data

In Table 6.6 are the 10 best and worst models given with their hyperparameters and lowest recorded MSE loss. In Appendix B Figure 23 the MSE validation loss of the 10 best models are shown, where Figure 23(a), (b) and (c) show the same data, but on different scales to properly see if loss curves converge. Figure 23(a) shows the epochs on a linear x-axis and loss on a linear y-axis, (b) shows the epochs on a linear x-axis and loss on a logarithmic y-axis, and (c) shows the epochs on a logarithmic x-axis and loss on a logarithmic y-axis. Similarly, the MSE validation loss of the 10 worst models on all the data are shown in Appendix B Figure 24 with similar scaled axes for (a), (b) and (c). The model number in the legend of each figure correspond with the model rank from Table 6.6.



Model rank	Layers	Nodes	Dropout	Batch normalization	Validation MSE
1	1	16	0.0	No	38.40
2	4	32	0.2	No	39.45
3	1	8	0.0	No	42.09
4	1	32	0.0	No	42.93
5	1	64	0.1	No	43.19
6	4	64	0.2	No	44.61
7	2	8	0.0	Yes	44.75
8	3	64	0.0	No	44.85
9	4	64	0.3	No	44.86
10	2	64	0.0	Yes	44.95
…	…	…	…	…	…
183	4	8	0.4	Yes	452.22
184	3	8	0.5	Yes	479.96
185	4	16	0.5	No	492.51
186	4	8	0.3	No	661.11
187	3	8	0.4	No	684.85
188	4	8	0.4	No	741.89
189	4	8	0.5	No	779.92
190	3	8	0.2	No	930.17
191	4	8	0.5	Yes	979.24
192	3	8	0.5	No	1267.51

Table 6.6: MSE loss of the 10 best and worst models with pad instances on the validation set of the 3σ datasplit

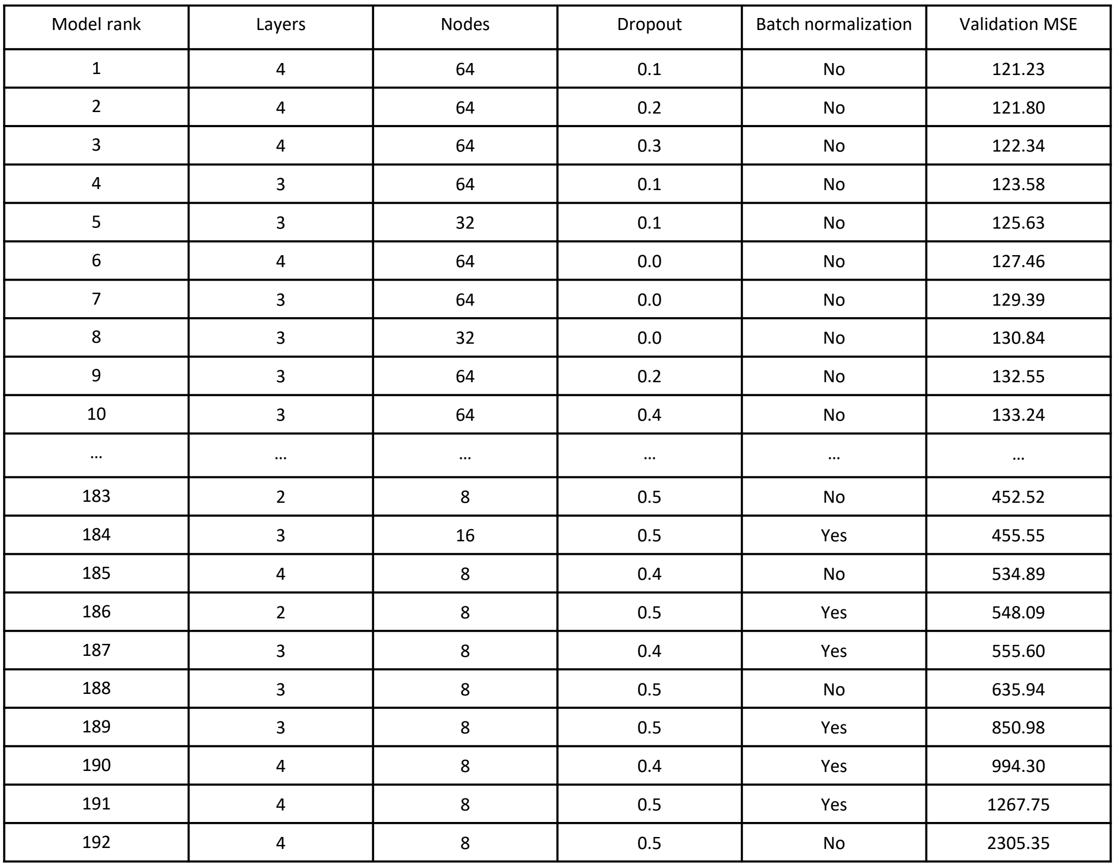
From Table 6.6 we observe that most of the 10 best models only use one layer, with low dropout rates and no batch normalization. The 10 worst models mostly use many layers with a low amount of nodes per layer and high dropout rates. Since most of the best models only consists of one layer, it would imply that a relatively simple model is enough to capture the complexity in our data.

SEM granularity

At SEM granularity level models are trained upto 1500 epochs with earlystopping applied, learning rate 0.001 and the Adam optimizer. In Appendix B are validation loss curves of the 10 best and worst models on each datasplit shown.

I. 80%/20% random data split

In Table 6.7 are the 10 best and worst models given with their hyperparameters and lowest recorded MSE loss. In Figure 6.26 the MSE validation loss of the 10 best models are shown, where Figure 6.26(a), (b) and (c) show the same data, but on different scales to properly see if loss curves converge. Figure 6.26(a) shows the epochs on a linear x-axis and loss on a linear y-axis, (b) shows the epochs on a linear x-axis and loss on a logarithmic y-axis, and (c) shows the epochs on a logarithmic x-axis and loss on a logarithmic y-axis. Similarly, the MSE validation loss of the 10 worst models on the 80%/20% datasplit are shown in Appendix B Figure 25 with similar scaled axes for (a), (b) and (c). The model number in the legend of each figure correspond with the model rank from Table 6.7.



Model rank	Layers	Nodes	Dropout	Batch normalization	Validation MSE
1	4	64	0.1	No	121.23
2	4	64	0.2	No	121.80
3	4	64	0.3	No	122.34
4	3	64	0.1	No	123.58
5	3	32	0.1	No	125.63
6	4	64	0.0	No	127.46
7	3	64	0.0	No	129.39
8	3	32	0.0	No	130.84
9	3	64	0.2	No	132.55
10	3	64	0.4	No	133.24
…	…	…	…	…	…
183	2	8	0.5	No	452.52
184	3	16	0.5	Yes	455.55
185	4	8	0.4	No	534.89
186	2	8	0.5	Yes	548.09
187	3	8	0.4	Yes	555.60
188	3	8	0.5	No	635.94
189	3	8	0.5	Yes	850.98
190	4	8	0.4	Yes	994.30
191	4	8	0.5	Yes	1267.75
192	4	8	0.5	No	2305.35

Table 6.7: MSE loss of the 10 best and worst models with SEM instances on the validation set of the 80%/20% datasplit

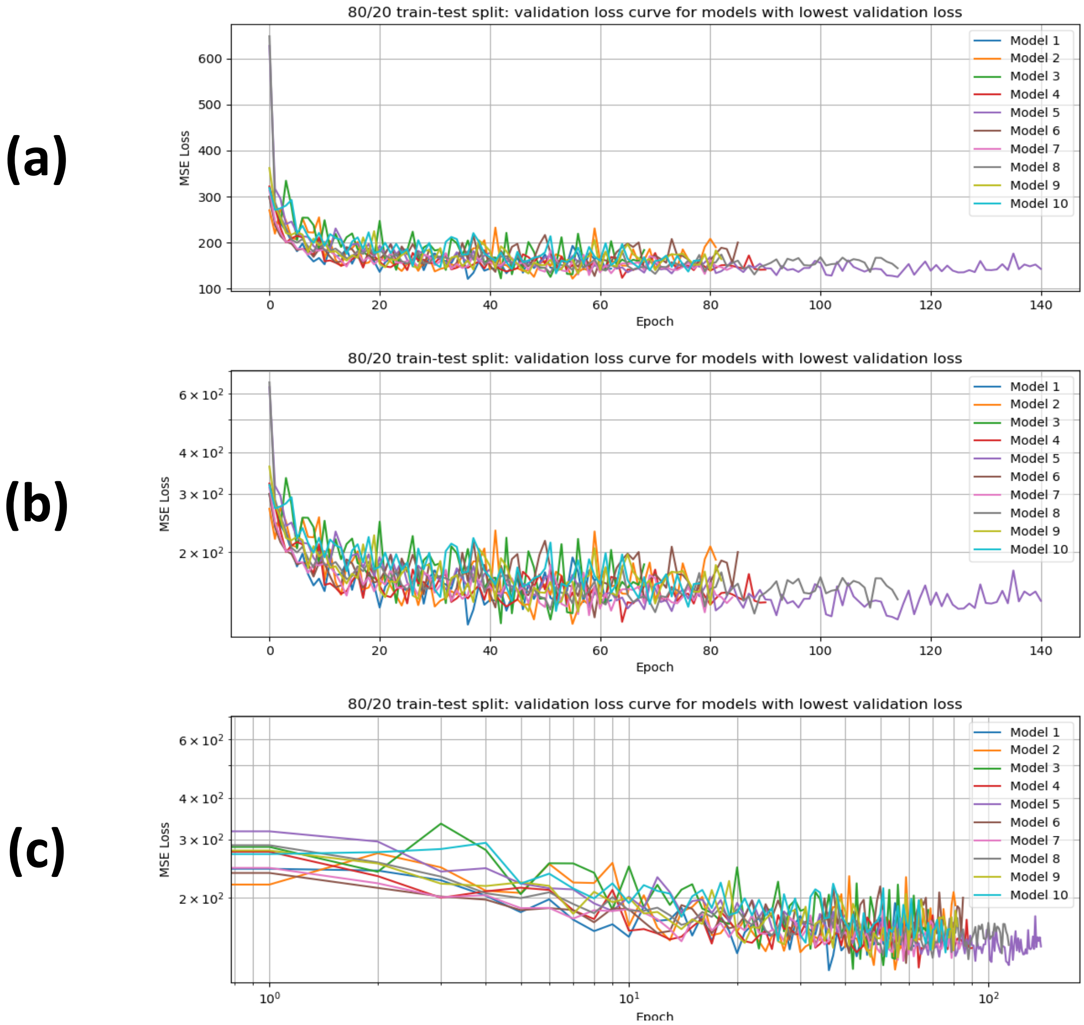
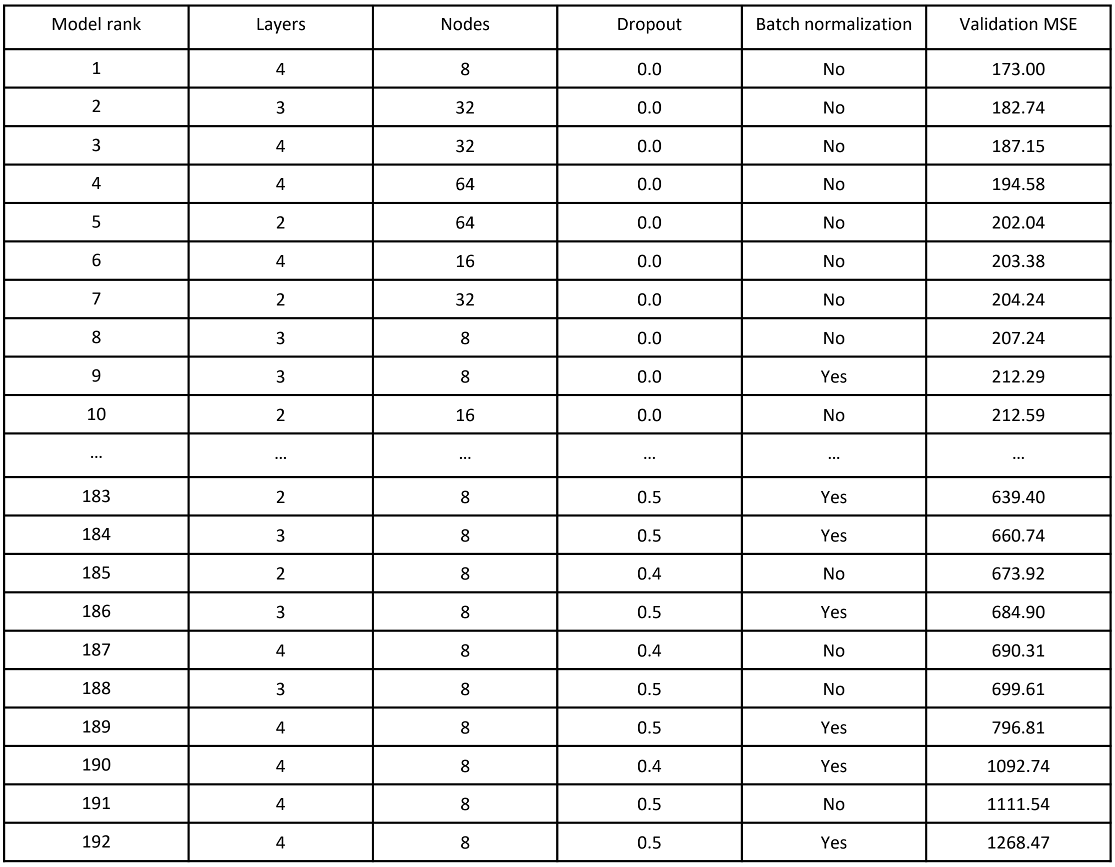


Figure 6.26: MSE validation loss of the 10 best models using SEM granularity inputs on the 80%/20% datasplit. (a) Epochs and loss on linear scale. (b) Epochs on linear scale and loss on logarithmic scale. (c) Epochs and loss on logarithmic scale.

The validation loss curves of the 10 best models are shown in Figure 6.26. From Table 6.7 we observe that the 10 best models use many layers, many nodes per layer and no batch normalization. The 10 worst models mostly use high dropout rates and a small amount of nodes per layer. Since all models consistently use many layers, many nodes and low dropout rates it implies that there is an increase in data complexity compared to data of field and pad granularity. Furthermore, we observe that all models behave roughly the same, i.e. all models show some fluctuations in the loss curves.

II. Nominal range

In Table 6.8 are the 10 best models given with their hyperparameters and lowest recorded MSE loss. In Figure 6.27 the MSE validation loss of the 10 best models are shown, where Figure 6.27(a), (b) and (c) show the same data, but on different scales to properly see if loss curves converge. Figure 6.27(a) shows the epochs on a linear x-axis and loss on a linear y-axis, (b) shows the epochs on a linear x-axis and loss on a logarithmic y-axis, and (c) shows the epochs on a logarithmic x-axis and loss on a logarithmic y-axis. Similarly, the MSE validation loss of the 10 worst models on the nominal datasplit are shown in Appendix B Figure 26 with similar scaled axes for (a), (b) and (c). The model number in the legend of each figure correspond with the model rank from Table 6.8.



Model rank	Layers	Nodes	Dropout	Batch normalization	Validation MSE
1	4	8	0.0	No	173.00
2	3	32	0.0	No	182.74
3	4	32	0.0	No	187.15
4	4	64	0.0	No	194.58
5	2	64	0.0	No	202.04
6	4	16	0.0	No	203.38
7	2	32	0.0	No	204.24
8	3	8	0.0	No	207.24
9	3	8	0.0	Yes	212.29
10	2	16	0.0	No	212.59
…	…	…	…	…	…
183	2	8	0.5	Yes	639.40
184	3	8	0.5	Yes	660.74
185	2	8	0.4	No	673.92
186	3	8	0.5	Yes	684.90
187	4	8	0.4	No	690.31
188	3	8	0.5	No	699.61
189	4	8	0.5	Yes	796.81
190	4	8	0.4	Yes	1092.74
191	4	8	0.5	No	1111.54
192	4	8	0.5	Yes	1268.47

Table 6.8: MSE loss of the 10 best and worst model with SEM instances on the validation set of the nominal datasplit

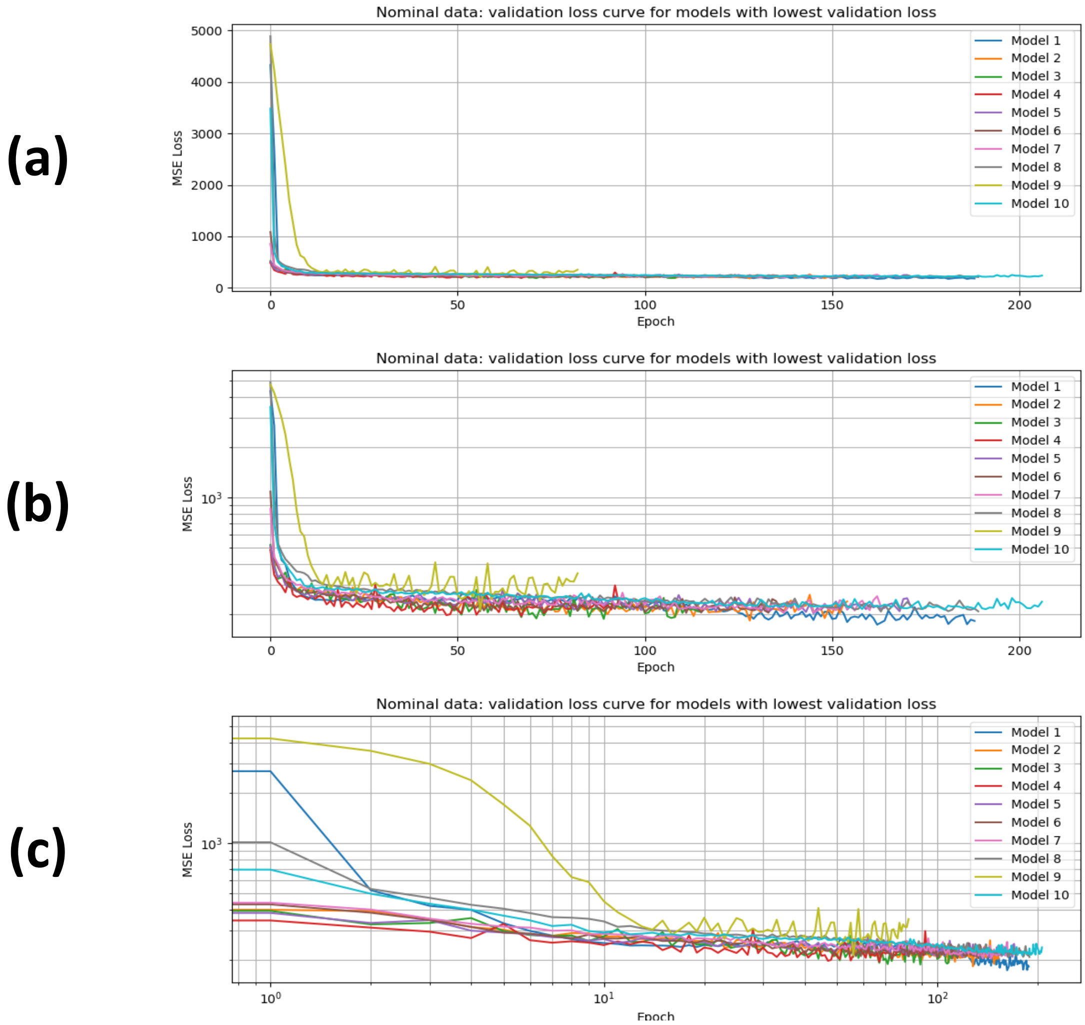
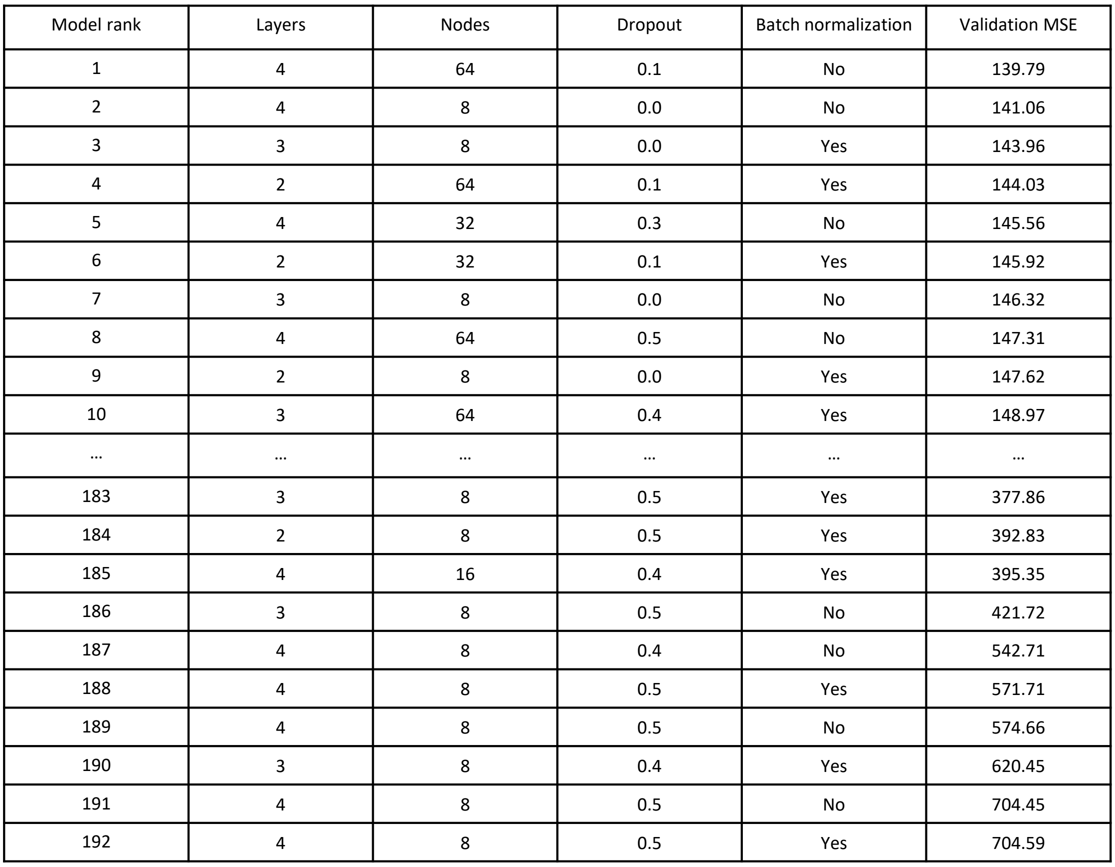


Figure 6.27: MSE validation loss of the 10 best models using SEM granularity inputs on the nominal datasplit. (a) Epochs and loss on linear scale. (b) Epochs on linear scale and loss on logarithmic scale. (c) Epochs and loss on logarithmic scale.

The validation loss curves are shown in Figure 6.27. From Table 6.8 we observe that the 10 best models use no dropout and no batch normalization. The 10 worst models mostly use high dropout rates, batch normalization and a small amount of nodes per layer. Model 9 shows more fluctuations in the loss curves compared to the other models, which is likely due to batch normalization being applied.

III. On all data

In Table 6.9 are the 10 best models given with their hyperparameters and lowest recorded MSE loss. In Figure 6.28 the MSE validation loss of the 10 best models are shown, where Figure 6.28(a), (b) and (c) show the same data, but on different scales to properly see if loss curves converge. Figure 6.28(a) shows the epochs on a linear x-axis and loss on a linear y-axis, (b) shows the epochs on a linear x-axis and loss on a logarithmic y-axis, and (c) shows the epochs on a logarithmic x-axis and loss on a logarithmic y-axis. Similarly, the MSE validation loss of the 10 worst models on all the data are shown in Appendix B Figure 27 with similar scaled axes for (a), (b) and (c). The model numbers in the legend of each figure correspond with the model rank in Table 6.9.



Model rank	Layers	Nodes	Dropout	Batch normalization	Validation MSE
1	4	64	0.1	No	139.79
2	4	8	0.0	No	141.06
3	3	8	0.0	Yes	143.96
4	2	64	0.1	Yes	144.03
5	4	32	0.3	No	145.56
6	2	32	0.1	Yes	145.92
7	3	8	0.0	No	146.32
8	4	64	0.5	No	147.31
9	2	8	0.0	Yes	147.62
10	3	64	0.4	Yes	148.97
…	…	…	…	…	…
183	3	8	0.5	Yes	377.86
184	2	8	0.5	Yes	392.83
185	4	16	0.4	Yes	395.35
186	3	8	0.5	No	421.72
187	4	8	0.4	No	542.71
188	4	8	0.5	Yes	571.71
189	4	8	0.5	No	574.66
190	3	8	0.4	Yes	620.45
191	4	8	0.5	No	704.45
192	4	8	0.5	Yes	704.59

Table 6.9: MSE loss of the 10 best and worst models with SEM instances on the validation set of the 3σ datasplit

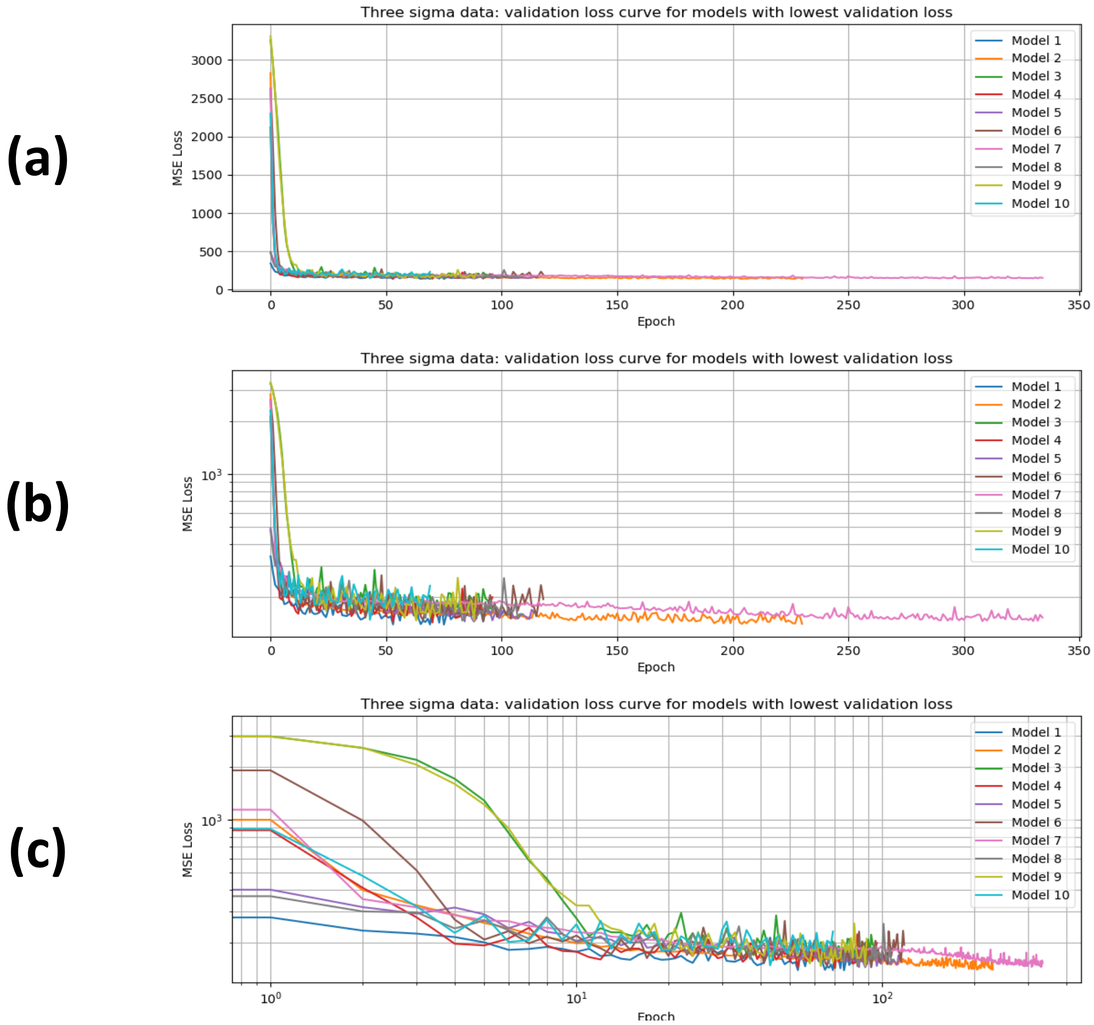


Figure 6.28: MSE validation loss of the 10 best and worst models using SEM granularity inputs on the 3σ datasplit. (a) Epochs and loss on linear scale. (b) Epochs on linear scale and loss on logarithmic scale. (c) Epochs and loss on logarithmic scale.

The validation loss curves are shown in Figure 6.28. From Table 6.9 we observe that the 10 best models mostly use many layers. The 10 worst models mostly use high dropout rates and a small amount of nodes per layer. Note that among the best models high dropout rates and batch normalization is sometimes used, which differs from the best models at field and pad granularity for the same datasplit.

In general we observe that the MSE loss increases with finer granularity, i.e. from field to pad to SEM. This is expected as the finer granularity results in a higher standard error of the mean. A higher standard error of the mean introduces more noise in the data, which explains the increase in model complexity. This is best observed between field and SEM granularity levels on the 80%/20% datasplit in Table 6.1 and 6.7 and on all the data in Table 6.3 and 6.9 by the increasing number of hidden layers and nodes per hidden layer in the 10 best models. In addition we can conclude that lower dropout rates tend to perform better than higher dropout rates and that batch normalization often leads to performance decrease.

6.2.2 Testing the models

Simple machine learning models are being performed with 33 input features based on CD, PE_x, PE_y for 11 orthogonal groups from groups 100-130 in Figure 2.4(a). The neural networks will be compared against simpler models, i.e. linear regression, lasso regression, ridge regression and elastic net regression. Following training, the hyperparameters used for lasso, ridge and elastic net regression with different input granularities on the 80%/20% datasplit are given in Table 6.10. Similarly the hyperparameters for the nominal focus datasplit and the 3σ datasplit are given in Table 6.11 and 6.12 respectively.

Algorithm	Hyperparameters		
	Field	Pad	SEM
Lasso	alpha = 0.01	alpha = 0.001	alpha = 0.001
Ridge	alpha = 0.01	alpha = 0.01	alpha = 0.01
Elastic Net	alpha = 0.001	alpha = 0.001	alpha = 0.0001
	l1_ratio = 0.9	l1_ratio = 0.9	l1_ratio = 0.6

Table 6.10: Hyperparameter values of the regression models for the 80%/20% datasplit at different granularity levels.

Algorithm	Hyperparameters		
	Field	Pad	SEM
Lasso	alpha = 0.001	alpha = 0.001	alpha = 0.001
Ridge	alpha = 0.001	alpha = 0.01	alpha = 10^{-6}
Elastic Net	alpha = 0.001	alpha = 0.001	alpha = 0.0001
	l1_ratio = 0.9	l1_ratio = 0.9	l1_ratio = 0.7

Table 6.11: Hyperparameter values of the regression models for the nominal datasplit at different granularity levels.

Algorithm	Hyperparameters		
	Field	Pad	SEM
Lasso	alpha = 0.001	alpha = 0.001	alpha = 0.001
Ridge	alpha = 0.01	alpha = 0.01	alpha = 10^{-6}
Elastic Net	alpha = 0.001	alpha = 0.001	alpha = 0.0001
	l1_ratio = 0.9	l1_ratio = 0.9	l1_ratio = 0.7

Table 6.12: Hyperparameter values of the regression models for the 3σ datasplit at different granularity levels.

From Table 6.10, 6.11 and 6.12 we observe that the elastic net models have a high L1 ratio. This implies that the models implements a proportionally high L1 penalty and low L2 penalty, which almost boils down to lasso regression.

Next, the models together with the best neural networks from section 6.2.1 are used on the test sets of the 80%/20%, nominal and 3σ dataset. In Table 6.13 the performance metrics are shown for each model and granularity level.

Algorithm	Field			Pad			SEM		
	R^2	RMSE	3σ	R^2	RMSE	3σ	R^2	RMSE	3σ
Linear	0.99	3.18	8.86	0.99	5.23	14.73	0.93	11.11	25.52
Lasso	0.99	3.60	9.86	0.98	5.60	15.31	0.93	11.15	25.67
Ridge	0.99	3.36	9.80	0.99	5.33	14.83	0.93	11.11	25.52
ElasticNet	0.99	3.60	10.11	0.98	5.45	15.26	0.93	11.10	25.54
Neural network	0.98	3.03	8.52	0.98	3.81	11.77	0.95	13.78	19.40

Table 6.13: Test results with R^2 on the 80%/20% datasplit, RMSE on the nominal datasplit, 3σ on all the data at different input granularity levels.

From Table 6.13 we observe that all models give a sufficiently high R^2 value close to zero at all granularity levels, meaning that focus can be properly predicted without too much error. At field granularity neural network performs on RMSE the best with 3.03 nm, i.e. focus errors deviate by 3.03 nm RMSE from the nominal focus -25 nm. Similarly for pad granularity, the RMSE is the best for neural network with 3.81 nm and shows a noticeable improvement in particular for the pad granularity level compared to the other regression models. However, the RMSE at SEM granularity level is worse than the regression models, showing errors deviating by 13.78 nm RMSE from nominal focus -25 nm in contrast with the 11.10 RMSE for the elasticnet model. The 3σ is the best for neural network at field, pad and

SEM with noticeable improvement in particular at pad and SEM granularity level. The 3σ scores show that 99.7% of the data in the neural networks contain focus errors upto 8.52 nm, 11.77 nm and 19.40 nm for field, pad and SEM granularity levels respectively.

6.2.3 Prediction on dataset 2

The models that are trained on the first dataset are applied on the second dataset. The second dataset contains the same output values as the first dataset, but differ in the input feature values. The second dataset differs from the first dataset in the CD distribution, so a performance decrease in the R^2, RMSE and 3σ can be expected. The results are shown in Table 6.14.

Algorithm	Field			Pad			SEM		
	R^2	RMSE	3σ	R^2	RMSE	3σ	R^2	RMSE	3σ
Linear	0.98	4.23	24.17	0.95	8.10	33.47	0.80	18.45	69.96
Lasso	0.98	4.38	26.64	0.95	8.56	35.03	0.80	18.55	69.94
Ridge	0.98	3.92	25.34	0.95	8.24	33.96	0.80	18.45	69.96
ElasticNet	0.98	4.51	25.63	0.95	8.34	34.14	0.80	18.46	69.82
Neural network	0.96	18.11	25.23	0.96	5.07	27.18	0.91	6.84	34.35

Table 6.14: Test results on the second dataset with R^2 on the 80%/20% datasplit, RMSE on the nominal datasplit, 3σ on all the data at different input granularity levels.

From Table 6.14 we observe that at field and pad granularity the R^2 of all models have reduced compared to Table 6.13, but still give a sufficiently high R^2 value close to zero. However, at SEM granularity only the neural network returns a reasonable R^2 value of 0.91, while the R^2 of the other regression models have reduced to 0.80. At field granularity level, the RMSE of the regression models have increased slightly compared to the first dataset, while the RMSE of neural network shows a big increase. For pad and SEM granularity there is also a RMSE increase present for all models, but neural networks then shows a noticeable performance improvement over the regression models. The 3σ of all models show a big increase on all granularity levels compared to Table 6.13. At field granularity the 3σ is the best for linear regression, while neural networks is the best at pad and SEM granularity level with big improvement over the regression models .

6.2.4 Discussion

From section 6.2.2 we observe that the performance metrics R^2, RMSE and 3σ on the testset deteriorate when using finer granularity levels, i.e. from field to pad to SEM granularity level. Although the number of input instances increase when using finer granularity levels, the standard error of the mean increases as demonstrated in Table 4.5, 4.6 and 4.7 for CD, PE_x and PE_y respectively. This increase results in noisier input data, an increase in model complexity and worse performance metrics.

The performance metrics show overall that neural network is the best algorithm as it returns high R^2 and the best 3σ and mostly the best RMSE at all granularity levels. This shows that a neural network is suitable for focus prediction and offers improvement over the linear regression baseline.

The performance metrics worsen with finer granularity, which is most noticeable in the RMSE and 3σ of pad and SEM granularity. The best RMSE for field is 3.03 nm , 3.81 nm for pad and 13.78 nm for SEM granularity. The ideal RMSE should be zero, so the RMSE for field and pad granularity is relatively close compared to the RMSE of SEM granularity. The best 3σ for field is 8.52 nm, 11.77 nm for pad and 19.40 nm for SEM granularity. The expected 3σ at field granularity should be around 10 nm, so 8.52 nm is acceptable. The 11.77 nm 3σ at pad granularity implies that roughly 99.7% of the focus predictions have an error upto 11.77 nm from the mean error. Since this is close to the expected 10 nm of field granularity level and that input data becomes noisier at pad granularity level, the 11.77 nm 3σ could still be considered as acceptable. However, at SEM granularity even more noise is introduced into the input data, which explains the 3σ increase to 19.40 nm. It says that roughly 99.7% of the focus predictions have an error upto 19.40 nm from the mean error, so it might be considered as bad as it might differ too much from the expected 10 nm 3σ at field granularity.

6.3 Experiment C: Defining the Proximity Force feature

To define the proximity force feature we start implementing the vector-based model by empirically trying out $1/r^n$ functions to calculate the proximity force for each contact hole. In Equation 6.1 the inverse-distance function for the proximity force is given for the relative distance r for the amount of neighboring contact holes $N = 250$ and order $n = 1, 2, 3, 4$.

$$PF_i = \sum_{j=1}^{250} \frac{1}{r_{ij}^n} \tag{6.1}$$

In Figure 6.29 a field of view image containing contact holes from a single SEM image are shown. The color of the contact holes indicate the PE and the calculated PF for different orders $1/r$ relationships in the X-direction. To have an indication on what the proper relationship between PE and PF could be, the images for the PF values in Figures 6.29(a),(b),(c) and (d) should follow a similar colorspread as the ones for PE in Figure 6.29(e). In Figure 6.29(a) it is clearly visible that the PF colorspread does not match with the PE of Figure 6.29(e) due to the large amount of PF values colored in red that are less present in the PE figures. In Figure 6.29(b) the colors are spread more evenly, although the higher PF values colored in red are still occurring more frequently. Figures 6.29(c) and (d) show more similar colorspread as the extreme values colored in blue and red are not frequently occurring in both the PF and PE figures.

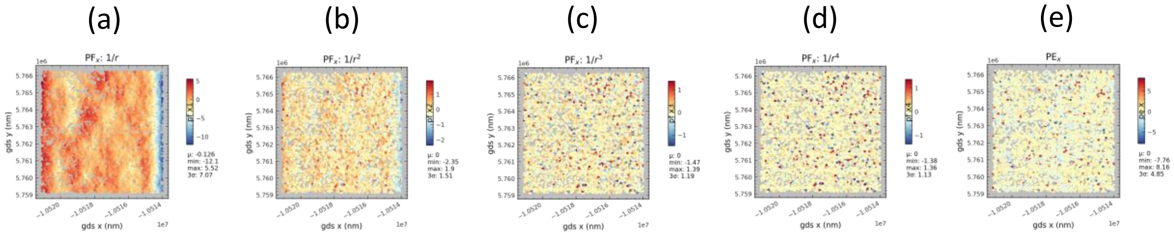


Figure 6.29: SEM image based field of view of calculated PF value per contact hole for different order $1/r$ relationships in the X-direction (a)(b)(c)(d) and the measured PE in the X-direction (e).

From Figure 6.29 it becomes clear how the PF calculations are supposed to be related to the PE values. Based on these observations we can exclude $1/r$ and $1/r^2$ as a possible PF formula. For $1/r^3$ and $1/r^4$ we cannot simply observe one SEM image per field to grasp the relationship between PF and PE considering the SEM image size relative to the field. Therefore all the contact holes from multiple SEM images within a field are taken into consideration by calculating for each contact hole a PF value. For each field a linear fit of PE against PF on each contact hole is performed to expose the relation between PE, focus and dose. In Figure 6.30(a) a linear fit of PE against PF in the X-direction for one field is shown. From this figure we observe that there is likely a linear relationship between PE and PF. The R^2 of the fit is considerably lower than the ideal $R^2 = 1$ due to the large spread of data points. To reduce the spread binning is applied, which results in the linear fit of Figure 6.30(b). Through binning the relationship between PF and PE takes on a more profound linear shape and the R^2 increases. The slope of this fit will indicate the sensitivity of PE with respect to focus and dose. The PF function will be determined by considering the R^2 for directions X and Y individually and the cosine similarity for X and Y combined.

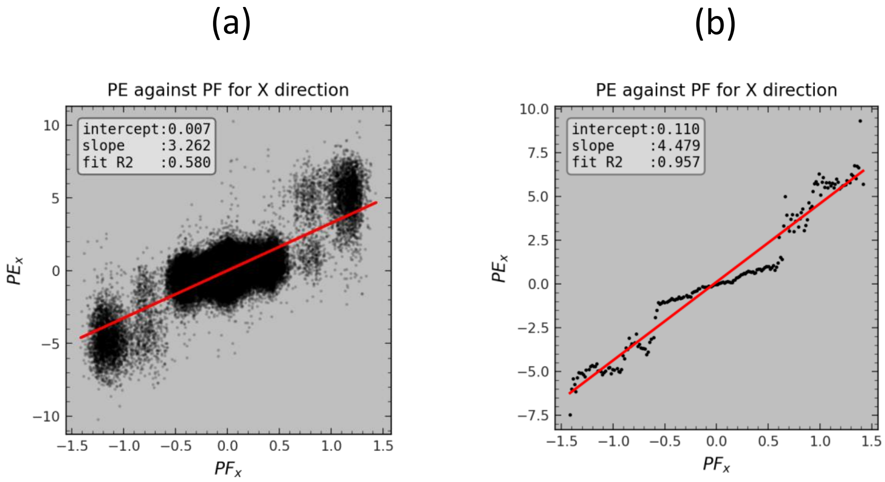


Figure 6.30: Linear fit for PE against PF in the X-direction before and after binning

In Figure 6.31 the linear fit PE against PF for one field in direction X and Y is shown for the $1/r^3$ and $1/r^4$ relationships. The figures for the Y-direction show a clear linearity around $PF = 0$, which gets less profound with more extreme PF values due to outliers. This probably explains why the figures for the X-direction look non-linear. Figure 6.30(a) has shown that the most contactholes fall in the central cluster around $PF = 0$. The most knowledge about the relation between PF and PE can be found in this cluster. Therefore we want to perform a fit on the bins in the central cluster by applying a bincount boundary of 1000, i.e. bins with more than 1000 contact holes.

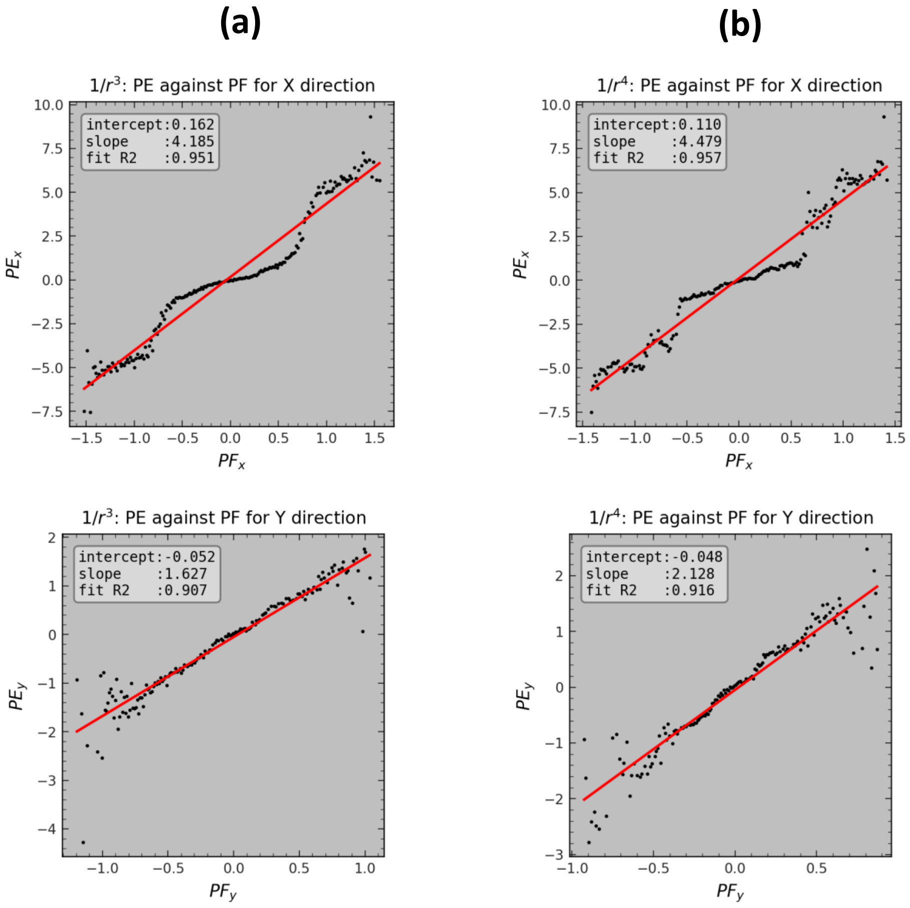


Figure 6.31: Linear fit for PE against PF in the X and Y-direction

After performing a fit with bincount boundary on each all focus and dose combinations, the R^2 of the $1/r^3$ and $1/r^4$ relationship the X and Y-direction are shown in Figure 6.32. We observe that in Figure 6.32(a) and (b) the R^2 is the highest in both the X and Y-direction. Therefore we select $1/r^3$ as the best function for PF. However, as we have observed in Figure 6.31 a non-linear pattern arises following the $1/r^3$ relationship for PF. This might indicate that actually the $1/r^3$ relationship for PF might be too simplistic. Therefore we investigate whether a PF function using a Bessel function might be a more suitable relationship.

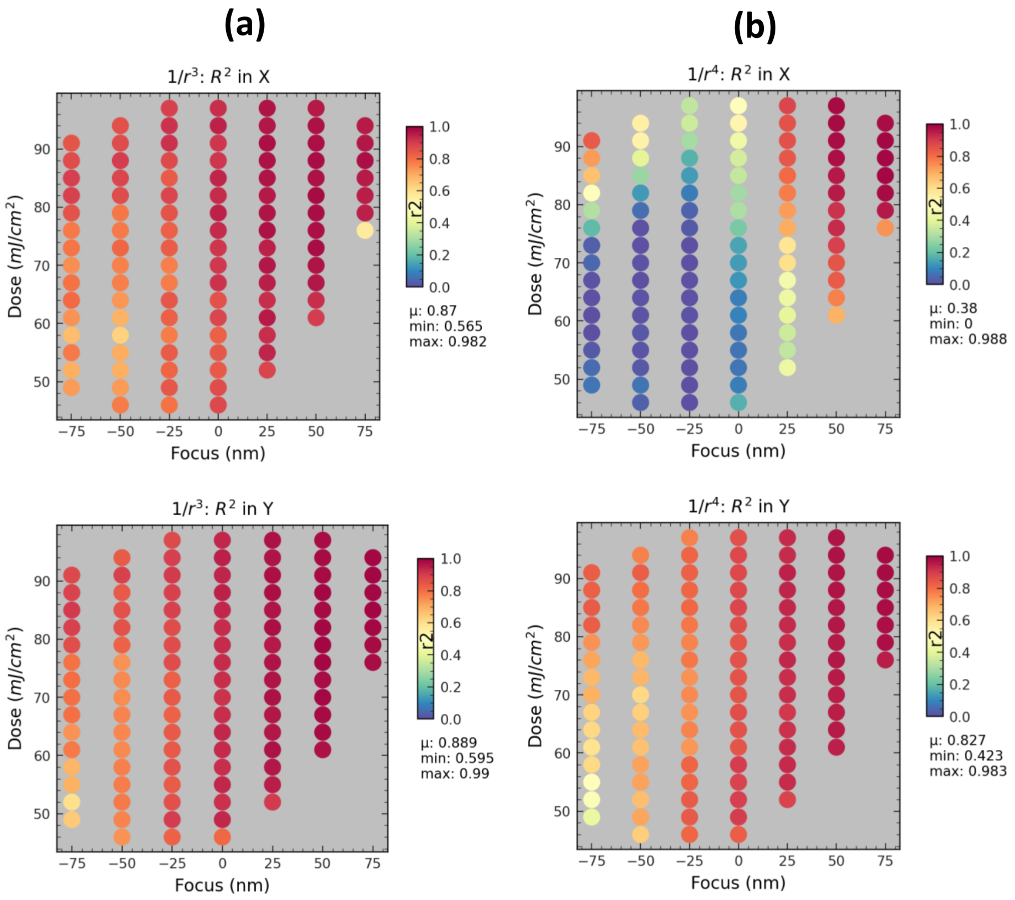


Figure 6.32: R^2 of the linear fit in the X and Y-direction for each focus and dose combination.

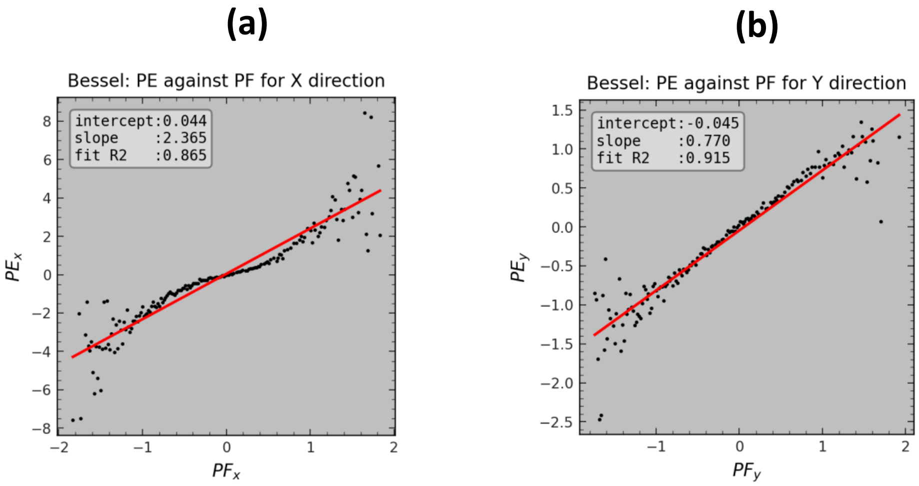


Figure 6.33: Linear fit for PE against PF in the X and Y-direction

In Figure 6.33 the linear fits are shown using the PF function from Equation 5.3 based on the Bessel function. Figure 6.33(b) shows in the Y-direction a similar linearity as in Figure 6.31. However, the linearity in the X-direction seems to be better for the Bessel function than for $1/r^3$. Similarly as before we apply a bincount of 1000 to capture the relationship around $PF = 0$. The R^2 in the X and Y-direction of the Bessel based relationship for each focus and dose combination is shown in Figure 6.34.

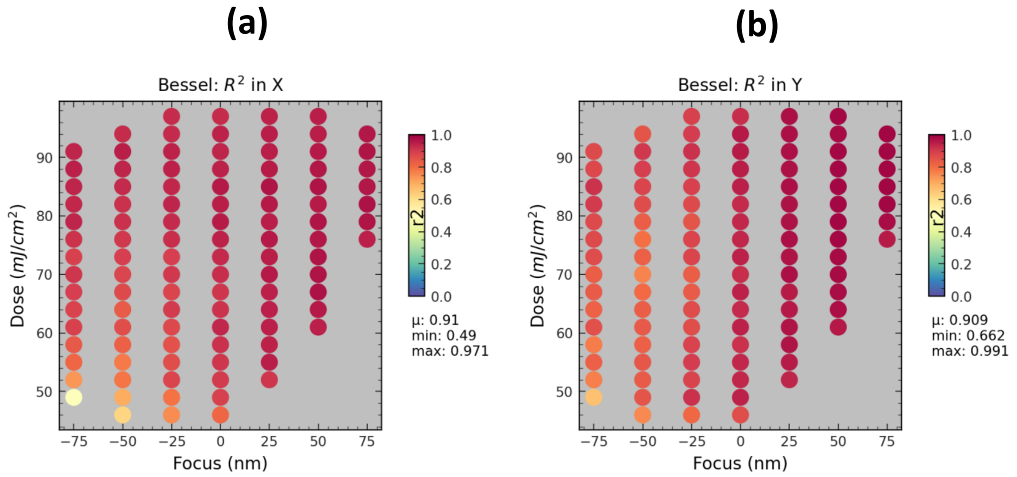


Figure 6.34: R^2 of the Bessel function in the X and Y-direction

From Figure 6.32 and 6.34 we observe that the relationship based on the Bessel function has a higher mean R^2 in the X and Y-direction than $1/r^3$. In Figure 6.35 the cosine similarity scores per focus and dose combinations are given by using the vectors $\overrightarrow{PE} = (PE_x, PE_y)$ and $\overrightarrow{PF} = (PF_x, PF_y)$. When comparing the cosine similarity for the $1/r^3$ and Bessel based relationship, we see that $1/r^3$ performs the best as the mean cosine similarity score is higher. While the Bessel based model has the best R^2 and the $1/r^3$ relationship has the best cosine similarity, we opt for the Bessel based function from Equation 5.3 to describe PF as this formula is heavily related to optics and shows a bigger improvement on R^2 compared to the improvement of $1/r^3$ on cosine similarity. The coefficients of the linear fits in the X and Y-direction are shown in Figure 6.36. These coefficients will be used as the PF values in models from Experiment D.

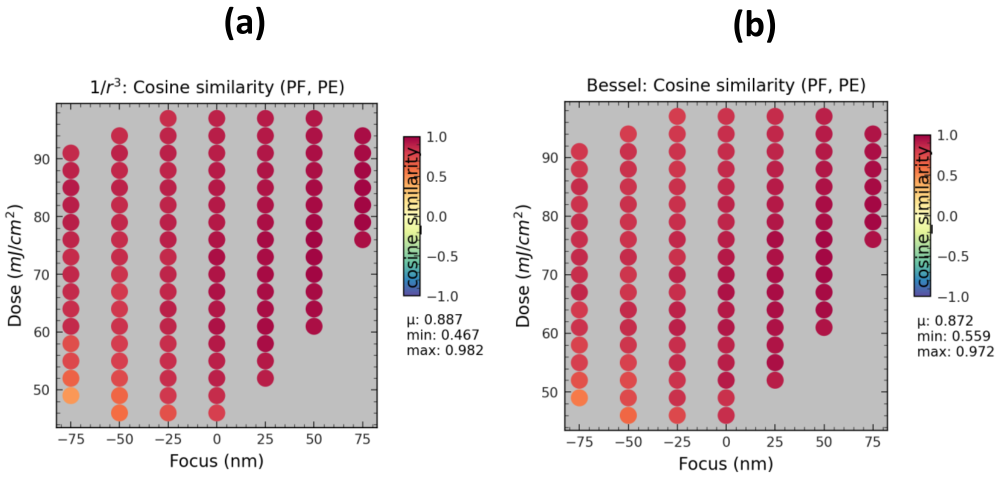


Figure 6.35: Cosine similarity of the $1/r^3$ and Bessel functions

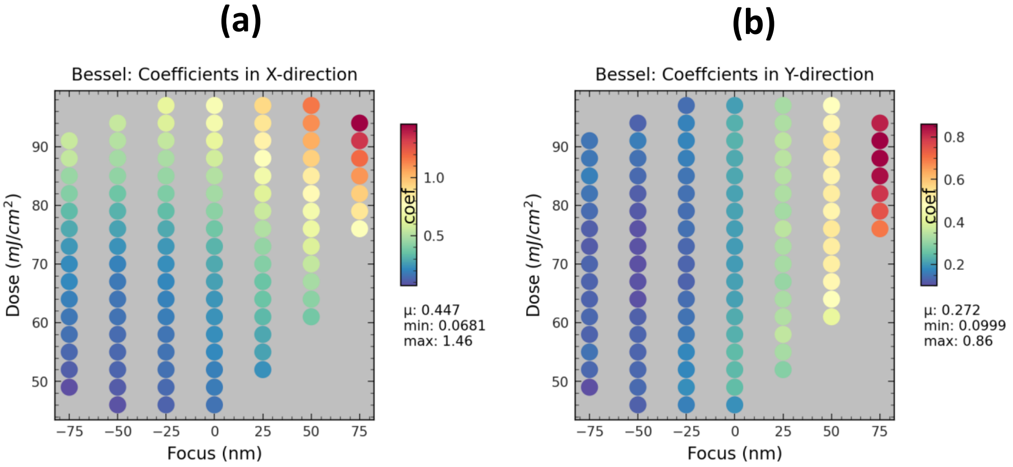


Figure 6.36: Coefficients for PF in the X and Y-direction with the Bessel based relationship.

6.4 Experiment D: Focus prediction based on orthogonal groups with Proximity Force added

With the new proximity force feature defined in Experiment C section 6.3 we aim to improve the previous models from Experiment (B) section 6.2. The models still use the CD, PE_x and PE_y per orthogonal group as before, but also use two additional features PF_x and PF_y, which are the coefficients as shown in Figure 6.36. This brings the number of input features to 35. For hyperparameter tuning the same hyperparameter ranges from Experiment (B) are used as defined by Table 4.12 and 4.13.

6.4.1 Feedforward Neural Network

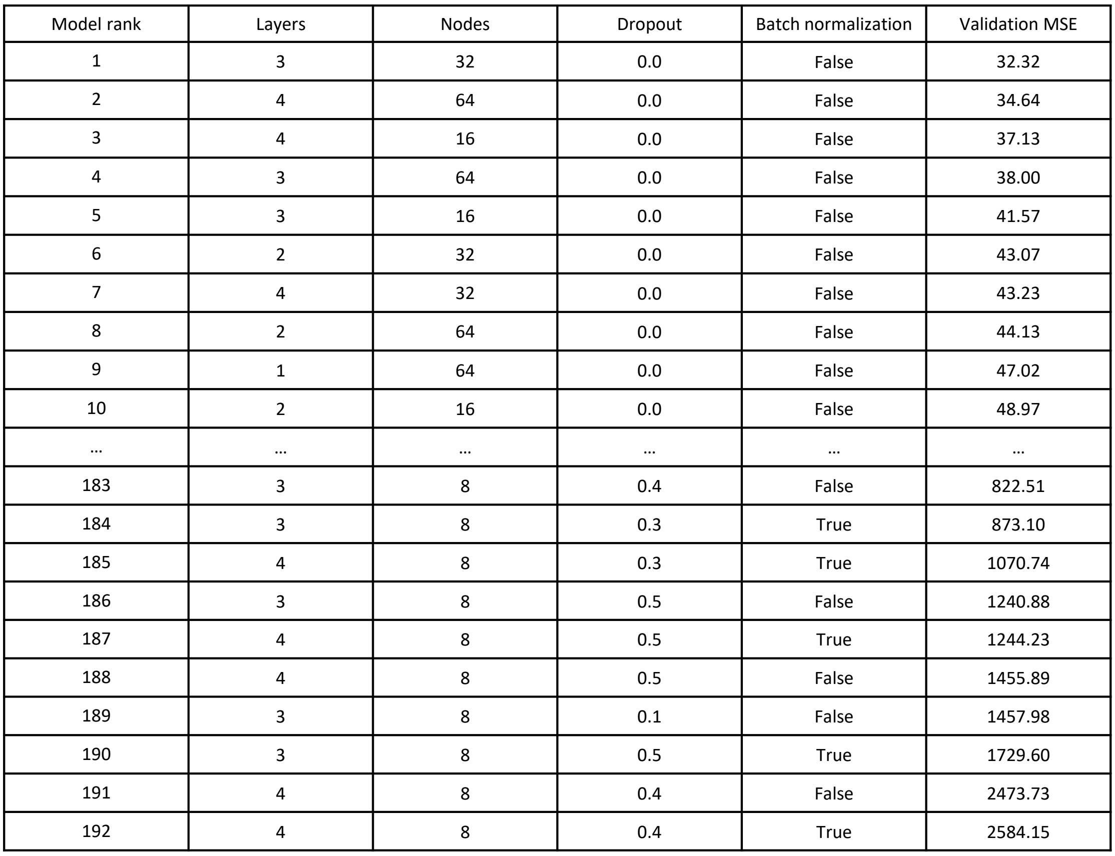
A model is fitted for each hyperparameter combination from Table 4.13 following the gridsearch approach. The MSE loss of the validation set of the 10 best and 10 worst models are given for each datasplit and granularity level. The model with the lowest MSE loss will be considered as the best model, which will be used on the test set. The validation loss curves show similar behavior as the validation curves from Experiment (B). In Appendix C the MSE loss curves of the 10 best and worst models for different input granularity levels and datasplits are shown.

Field granularity

At field granularity the models are trained with learning rate 0.001, 2500 epochs, the Adam optimizer and earlystopping applied. The validation loss of the 10 best and worst models per datasplit are shown below.

I. 80%/20% random data split

In Appendix C Figure 28 and 29 are the validation loss curves of the 10 best and worst models on the 80%/20% random data split given. Subfigures (a), (b) and (c) show the same data, but on different scales to properly see if loss curves converge. Subfigure (a) shows the epochs on a linear x-axis and loss on a linear y-axis, (b) shows the epochs on a linear x-axis and loss on a logarithmic y-axis, and (c) shows the epochs on a logarithmic x-axis and loss on a logarithmic y-axis. The model numbers in the legend of each figure correspond with the model rank in Table 6.15.



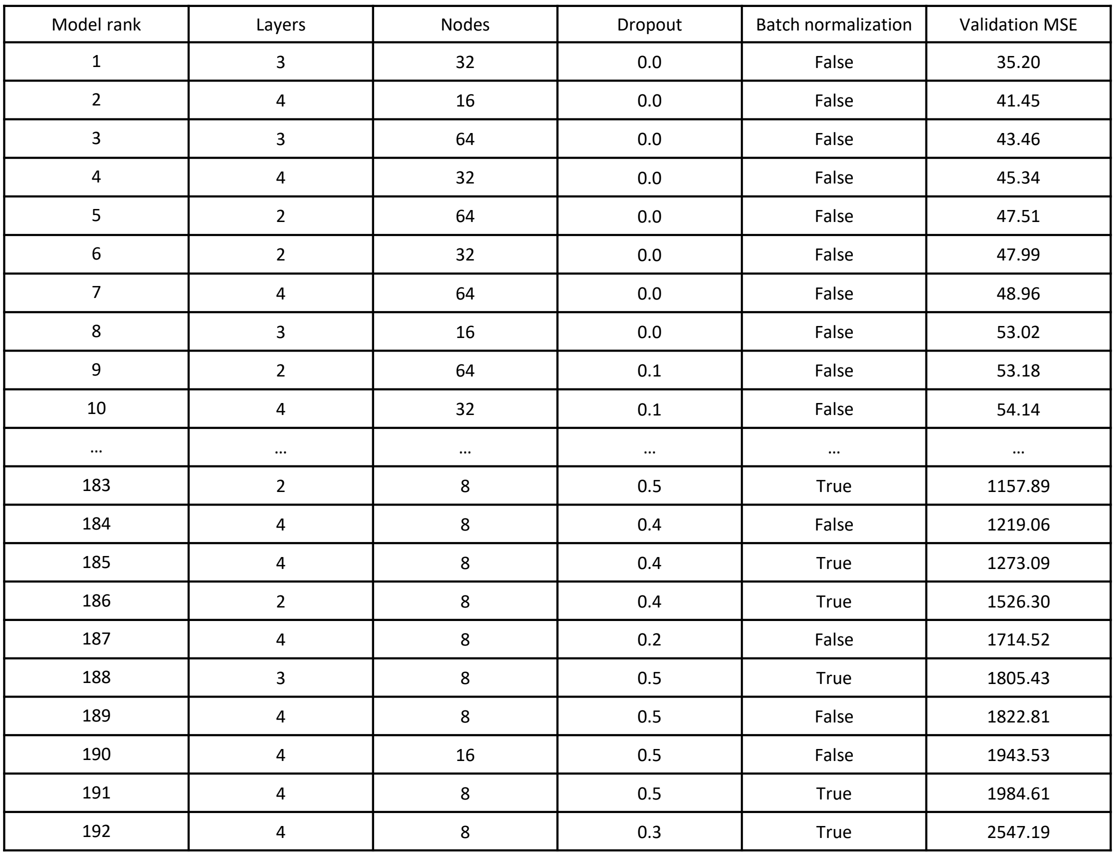
Model rank	Layers	Nodes	Dropout	Batch normalization	Validation MSE
1	3	32	0.0	False	32.32
2	4	64	0.0	False	34.64
3	4	16	0.0	False	37.13
4	3	64	0.0	False	38.00
5	3	16	0.0	False	41.57
6	2	32	0.0	False	43.07
7	4	32	0.0	False	43.23
8	2	64	0.0	False	44.13
9	1	64	0.0	False	47.02
10	2	16	0.0	False	48.97
…	…	…	…	…	…
183	3	8	0.4	False	822.51
184	3	8	0.3	True	873.10
185	4	8	0.3	True	1070.74
186	3	8	0.5	False	1240.88
187	4	8	0.5	True	1244.23
188	4	8	0.5	False	1455.89
189	3	8	0.1	False	1457.98
190	3	8	0.5	True	1729.60
191	4	8	0.4	False	2473.73
192	4	8	0.4	True	2584.15

Table 6.15: MSE loss of the 10 best and worst models with field instances on the validation set of the 80%/20% datasplit

In Table 6.15 are the 10 best and worst models given with their hyperparameters and lowest recorded MSE loss. From Table 6.15 we observe that the 10 best models do not use dropout rate nor batch normalization. The 10 worst models consists mostly of a high amount of layers, a small amount of nodes per layer and a high dropout rate.

II. Nominal range

The validation loss curves of the 10 best and worst models on the nominal datasplit are shown in Appendix C Figure 30 and 31. Subfigures (a), (b) and (c) show the same data, but on different scales to properly see if loss curves converge. Subfigure (a) shows the epochs on a linear x-axis and loss on a linear y-axis, (b) shows the epochs on a linear x-axis and loss on a logarithmic y-axis, and (c) shows the epochs on a logarithmic x-axis and loss on a logarithmic y-axis. The model numbers in the legend of each figure correspond with the model rank in Table 6.16.



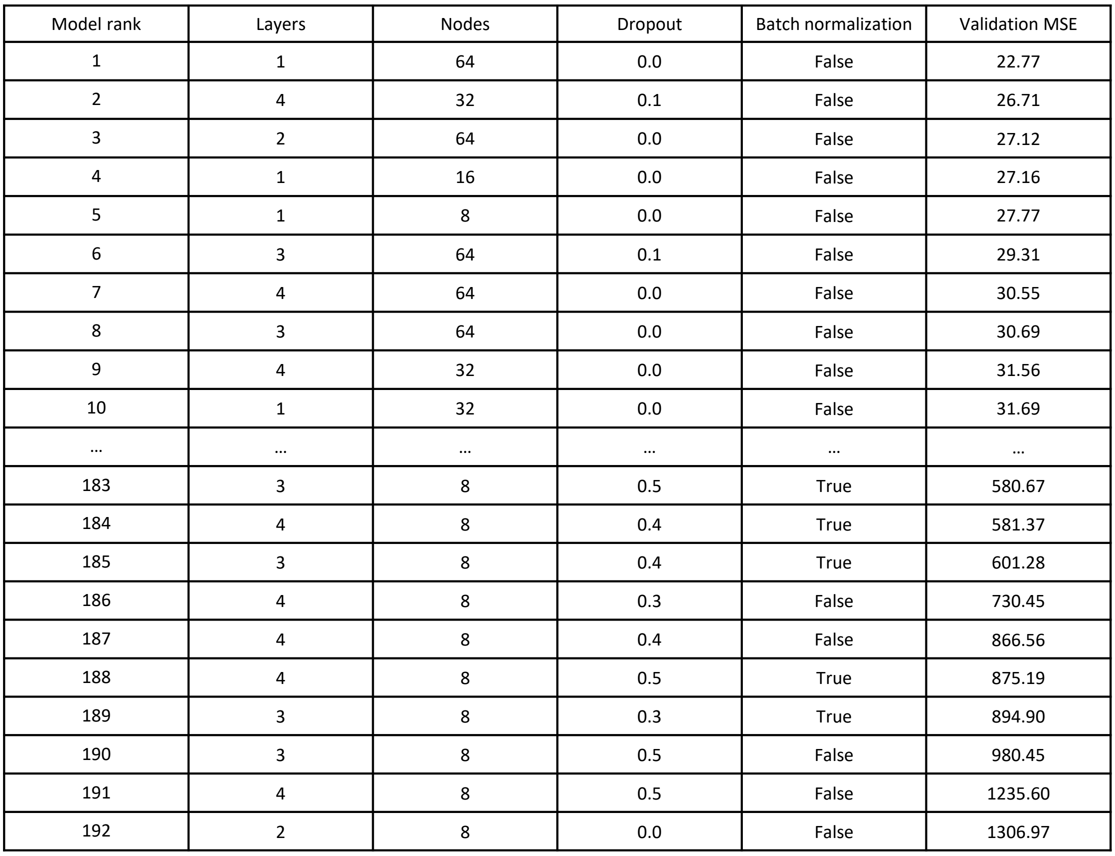
Model rank	Layers	Nodes	Dropout	Batch normalization	Validation MSE
1	3	32	0.0	False	35.20
2	4	16	0.0	False	41.45
3	3	64	0.0	False	43.46
4	4	32	0.0	False	45.34
5	2	64	0.0	False	47.51
6	2	32	0.0	False	47.99
7	4	64	0.0	False	48.96
8	3	16	0.0	False	53.02
9	2	64	0.1	False	53.18
10	4	32	0.1	False	54.14
…	…	…	…	…	…
183	2	8	0.5	True	1157.89
184	4	8	0.4	False	1219.06
185	4	8	0.4	True	1273.09
186	2	8	0.4	True	1526.30
187	4	8	0.2	False	1714.52
188	3	8	0.5	True	1805.43
189	4	8	0.5	False	1822.81
190	4	16	0.5	False	1943.53
191	4	8	0.5	True	1984.61
192	4	8	0.3	True	2547.19

Table 6.16: MSE loss of the 10 best and worst models with field instances on the validation set of the nominal datasplit

In Table 6.16 are the 10 best and worst models given with their hyperparameters and lowest recorded MSE loss. From Table 6.16 we observe that the 10 best models mostly use a large amount of nodes per layer, a low dropout rate and no batch normalization. The 10 worst models consists mostly of high dropout rates, many layers and small amount of nodes per layer. When comparing models 184 with 185 and models 189 with 191 we observe that each pair of models use the same hyperparameters, but differ in batch normalization. It shows that batch normalization has a bad impact on the performance.

III. On all data

The validation loss curves of the 10 best and worst models are shown in Appendix C Figure 32 and 33. To properly see if loss curves converge subfigures (a), (b) and (c) show the same data on different scales. Subfigure (a) shows the epochs on a linear x-axis and loss on a linear y-axis, (b) shows the epochs on a linear x-axis and loss on a logarithmic y-axis, and (c) shows the epochs on a logarithmic x-axis and loss on a logarithmic y-axis. The model numbers in the legend of each figure correspond with the model rank in Table 6.17.



Model rank	Layers	Nodes	Dropout	Batch normalization	Validation MSE
1	1	64	0.0	False	22.77
2	4	32	0.1	False	26.71
3	2	64	0.0	False	27.12
4	1	16	0.0	False	27.16
5	1	8	0.0	False	27.77
6	3	64	0.1	False	29.31
7	4	64	0.0	False	30.55
8	3	64	0.0	False	30.69
9	4	32	0.0	False	31.56
10	1	32	0.0	False	31.69
…	…	…	…	…	…
183	3	8	0.5	True	580.67
184	4	8	0.4	True	581.37
185	3	8	0.4	True	601.28
186	4	8	0.3	False	730.45
187	4	8	0.4	False	866.56
188	4	8	0.5	True	875.19
189	3	8	0.3	True	894.90
190	3	8	0.5	False	980.45
191	4	8	0.5	False	1235.60
192	2	8	0.0	False	1306.97

Table 6.17: MSE loss of the 10 best and worst models with field instances on the validation set of the 3σ datasplit

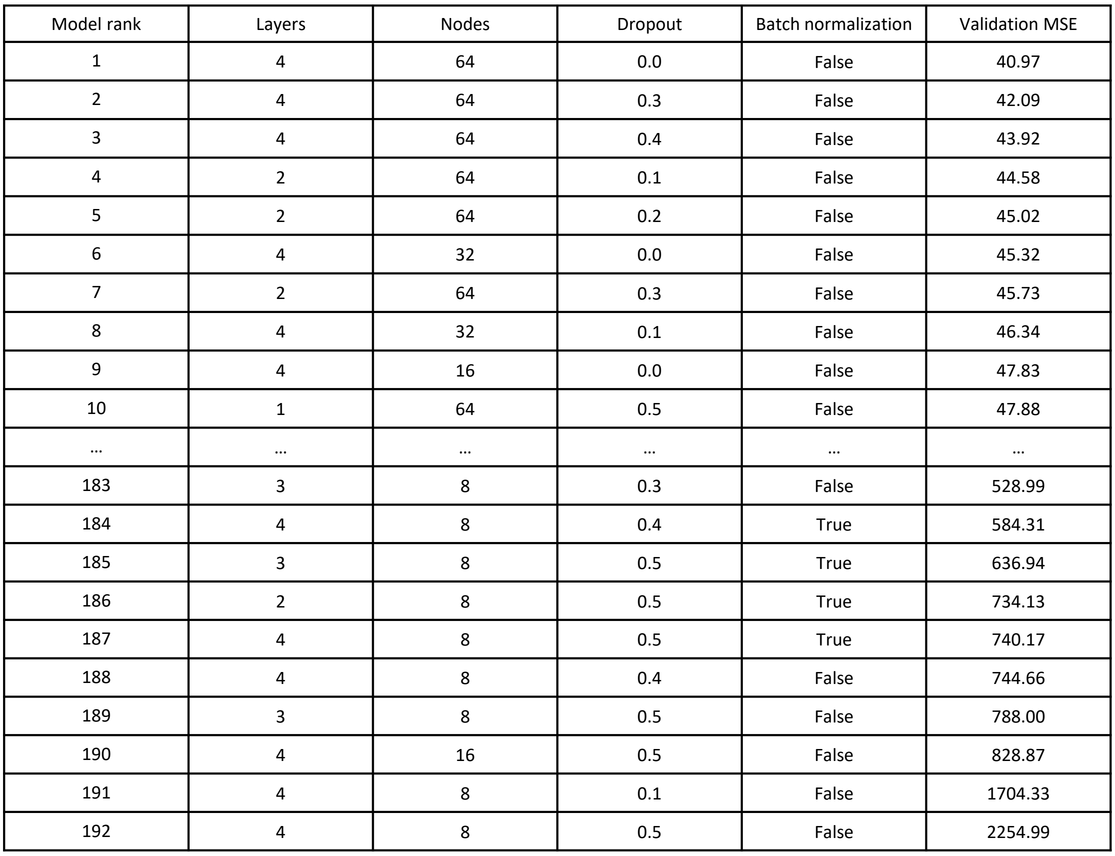
In Table 6.17 are the 10 best and worst models given with their hyperparameters and lowest recorded MSE loss. From Table 6.17 we observe that the 10 best models use a small amount of layers, a low dropout rate and no batch normalization. The 10 worst models mostly use many layers, a small amount of nodes per layer and high dropout rates. Note that most of the best models only consists of one layer, which implies that relatively simple model is enough to capture the complexity in our data.

Pad granularity

At pad granularity the models are fitted with learning rate 0.001 and 1500 epochs. The validation loss curves of the best and worst models per datasplit are shown in Appendix C.

I. 80%/20% random data split

The validation loss curves of the 10 best and worst models on the 80%/20% random data split are shown in Appendix C Figure 34 and 35. Subfigures (a), (b) and (c) show the same data, but on different scales to properly see if loss curves converge. Subfigure (a) shows the epochs on a linear x-axis and loss on a linear y-axis, (b) shows the epochs on a linear x-axis and loss on a logarithmic y-axis, and (c) shows the epochs on a logarithmic x-axis and loss on a logarithmic y-axis. The model numbers in the legend of each figure correspond with the model rank in Table 6.18.



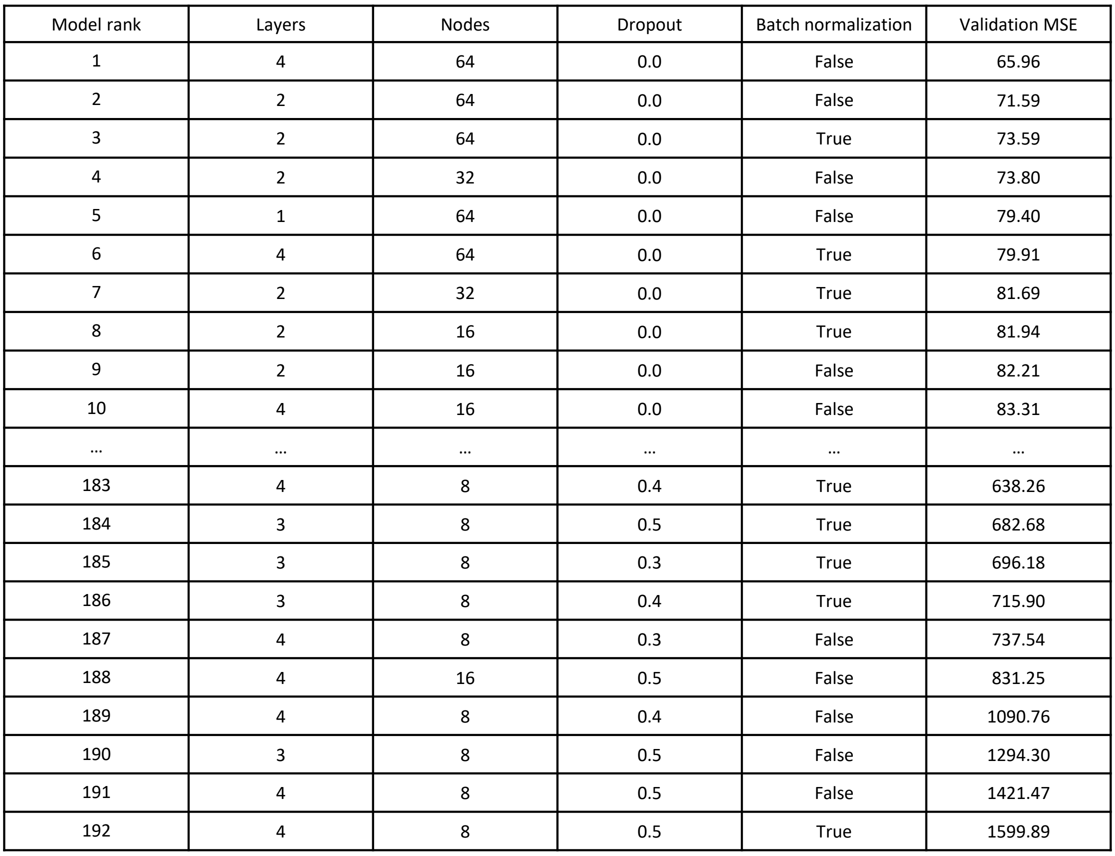
Model rank	Layers	Nodes	Dropout	Batch normalization	Validation MSE
1	4	64	0.0	False	40.97
2	4	64	0.3	False	42.09
3	4	64	0.4	False	43.92
4	2	64	0.1	False	44.58
5	2	64	0.2	False	45.02
6	4	32	0.0	False	45.32
7	2	64	0.3	False	45.73
8	4	32	0.1	False	46.34
9	4	16	0.0	False	47.83
10	1	64	0.5	False	47.88
…	…	…	…	…	…
183	3	8	0.3	False	528.99
184	4	8	0.4	True	584.31
185	3	8	0.5	True	636.94
186	2	8	0.5	True	734.13
187	4	8	0.5	True	740.17
188	4	8	0.4	False	744.66
189	3	8	0.5	False	788.00
190	4	16	0.5	False	828.87
191	4	8	0.1	False	1704.33
192	4	8	0.5	False	2254.99

Table 6.18: MSE loss of the 10 best and worst models with pad instances on the validation set of the 80%/20% datasplit

In Table 6.18 are the 10 best and worst models given with their hyperparameters and lowest recorded MSE loss. From Table 6.18 we observe that the 10 best models use many nodes per layer and no batch normalization. The 10 worst models mostly use many layers with small amount of nodes and high dropout rates.

II. Nominal range

The validation loss curves of the 10 best and worst models on the nominal datasplit are shown in Appendix C Figure 36 and 37. To properly see if loss curves converge subfigures (a), (b) and (c) show the same data on different scales. Subfigure (a) shows the epochs on a linear x-axis and loss on a linear y-axis, (b) shows the epochs on a linear x-axis and loss on a logarithmic y-axis, and (c) shows the epochs on a logarithmic x-axis and loss on a logarithmic y-axis. The model numbers in the legend of each figure correspond with the model rank in Table 6.19.



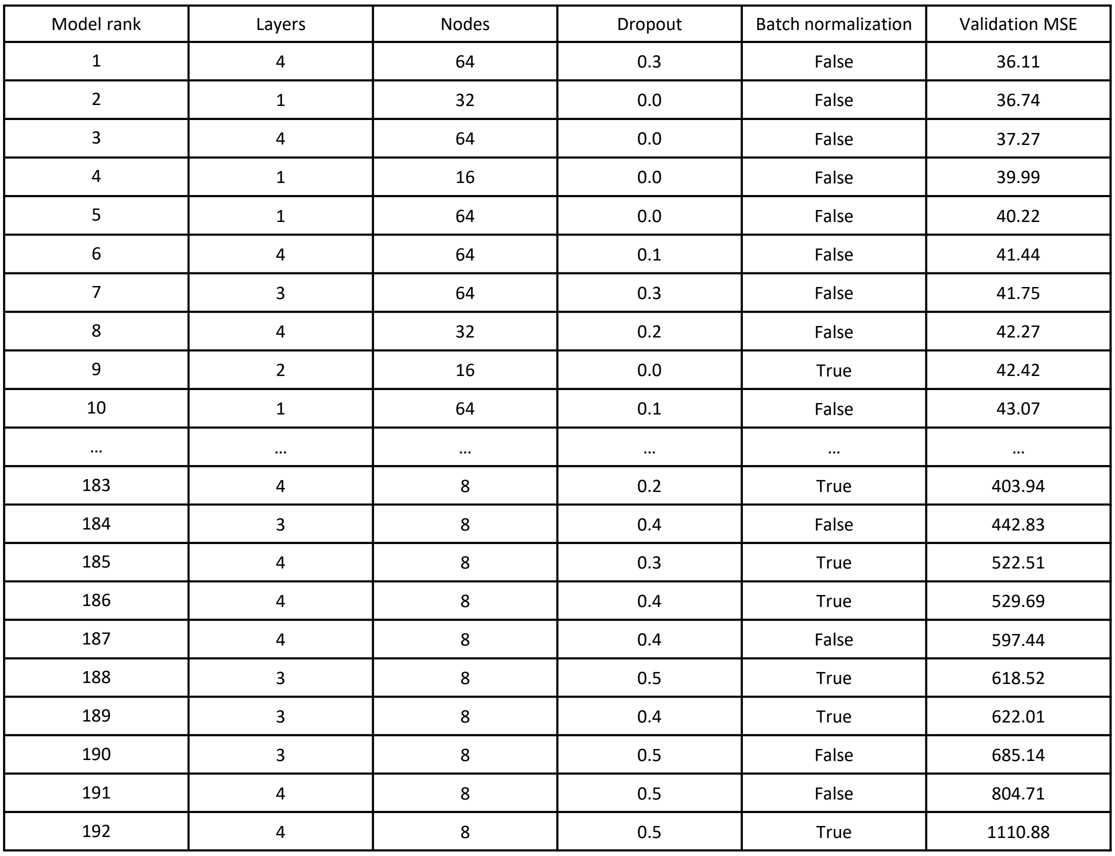
Model rank	Layers	Nodes	Dropout	Batch normalization	Validation MSE
1	4	64	0.0	False	65.96
2	2	64	0.0	False	71.59
3	2	64	0.0	True	73.59
4	2	32	0.0	False	73.80
5	1	64	0.0	False	79.40
6	4	64	0.0	True	79.91
7	2	32	0.0	True	81.69
8	2	16	0.0	True	81.94
9	2	16	0.0	False	82.21
10	4	16	0.0	False	83.31
…	…	…	…	…	…
183	4	8	0.4	True	638.26
184	3	8	0.5	True	682.68
185	3	8	0.3	True	696.18
186	3	8	0.4	True	715.90
187	4	8	0.3	False	737.54
188	4	16	0.5	False	831.25
189	4	8	0.4	False	1090.76
190	3	8	0.5	False	1294.30
191	4	8	0.5	False	1421.47
192	4	8	0.5	True	1599.89

Table 6.19: MSE loss of the 10 best and worst models with pad instances on the validation set of the nominal datasplit

In Table 6.19 are the 10 best and worst models given with their hyperparameters and lowest recorded MSE loss. From Table 6.19 we observe that the 10 best models use many nodes per layer and a low dropout rate. The 10 worst models mostly use many layers with small amount of nodes per layer and high dropout rates.

III. On all data

The validation loss curves of the 10 best and worst models are shown in Appendix C Figure 38 and 39. To properly see if loss curves converge subfigures (a), (b) and (c) show the same data on different scales. Subfigure (a) shows the epochs on a linear x-axis and loss on a linear y-axis, (b) shows the epochs on a linear x-axis and loss on a logarithmic y-axis, and (c) shows the epochs on a logarithmic x-axis and loss on a logarithmic y-axis. The model numbers in the legend of each figure correspond with the model rank in Table 6.20.



Model rank	Layers	Nodes	Dropout	Batch normalization	Validation MSE
1	4	64	0.3	False	36.11
2	1	32	0.0	False	36.74
3	4	64	0.0	False	37.27
4	1	16	0.0	False	39.99
5	1	64	0.0	False	40.22
6	4	64	0.1	False	41.44
7	3	64	0.3	False	41.75
8	4	32	0.2	False	42.27
9	2	16	0.0	True	42.42
10	1	64	0.1	False	43.07
…	…	…	…	…	…
183	4	8	0.2	True	403.94
184	3	8	0.4	False	442.83
185	4	8	0.3	True	522.51
186	4	8	0.4	True	529.69
187	4	8	0.4	False	597.44
188	3	8	0.5	True	618.52
189	3	8	0.4	True	622.01
190	3	8	0.5	False	685.14
191	4	8	0.5	False	804.71
192	4	8	0.5	True	1110.88

Table 6.20: MSE loss of the 10 best and worst models with pad instances on the validation set of the 3σ datasplit

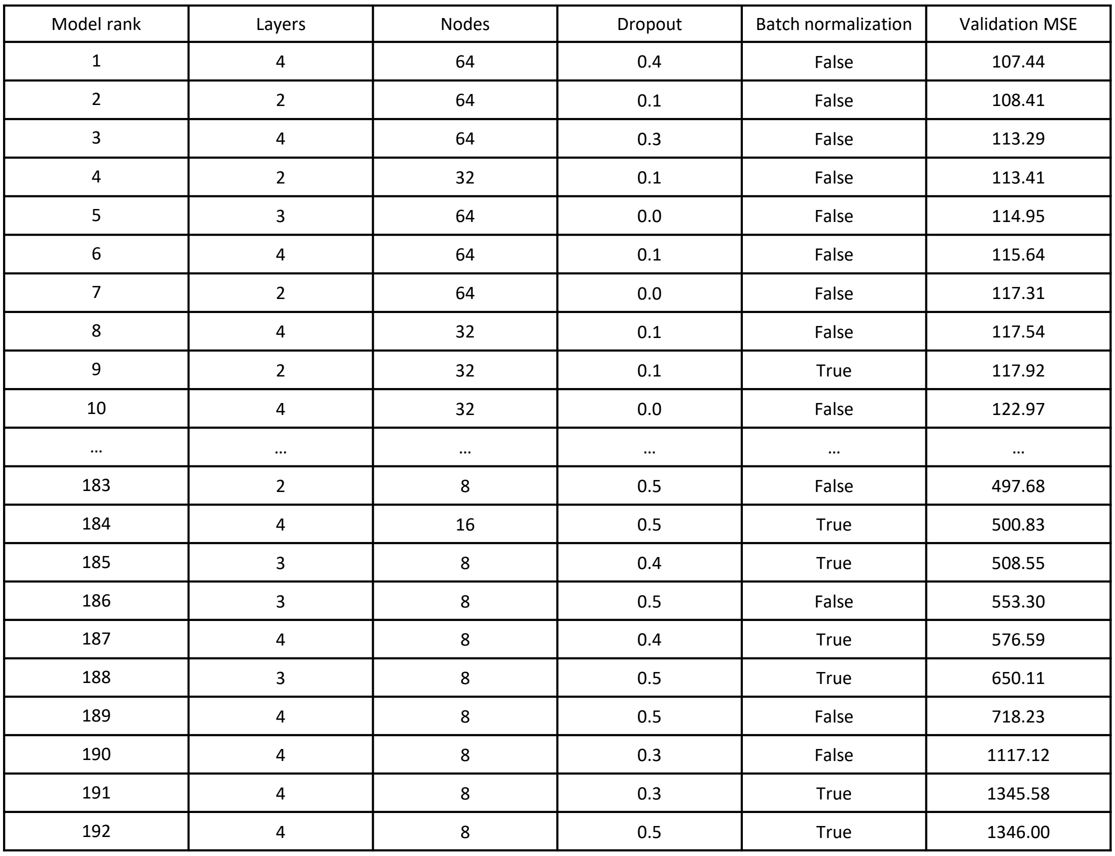
In Table 6.20 are the 10 best and worst models given with their hyperparameters and lowest recorded MSE loss. From Table 6.20 we observe that most of the 10 best models use a large amount of nodes per layer, with low dropout rates and no batch normalization. Since some of the best models only consists of one layer, it would imply that a relatively simple model is enough to capture the complexity in our data. The 10 worst models mostly use many layers with a low amount of nodes per layer and high dropout rates.

SEM granularity

At SEM granularity the models are fitted with learning rate 0.001 and 1500 epochs. The validation loss curves of the best and worst models per datasplit are shown in Appendix C.

I. 80%/20% random data split

The validation loss curves of the 10 best and worst models on the 80%/20% random data split are shown in Appendix C Figure 40 and 41. Subfigures (a), (b) and (c) show the same data, but on different scales to properly see if loss curves converge. Subfigure (a) shows the epochs on a linear x-axis and loss on a linear y-axis, (b) shows the epochs on a linear x-axis and loss on a logarithmic y-axis, and (c) shows the epochs on a logarithmic x-axis and loss on a logarithmic y-axis. The model numbers in the legend of each figure correspond with the model rank in Table 6.21.



Model rank	Layers	Nodes	Dropout	Batch normalization	Validation MSE
1	4	64	0.4	False	107.44
2	2	64	0.1	False	108.41
3	4	64	0.3	False	113.29
4	2	32	0.1	False	113.41
5	3	64	0.0	False	114.95
6	4	64	0.1	False	115.64
7	2	64	0.0	False	117.31
8	4	32	0.1	False	117.54
9	2	32	0.1	True	117.92
10	4	32	0.0	False	122.97
…	…	…	…	…	…
183	2	8	0.5	False	497.68
184	4	16	0.5	True	500.83
185	3	8	0.4	True	508.55
186	3	8	0.5	False	553.30
187	4	8	0.4	True	576.59
188	3	8	0.5	True	650.11
189	4	8	0.5	False	718.23
190	4	8	0.3	False	1117.12
191	4	8	0.3	True	1345.58
192	4	8	0.5	True	1346.00

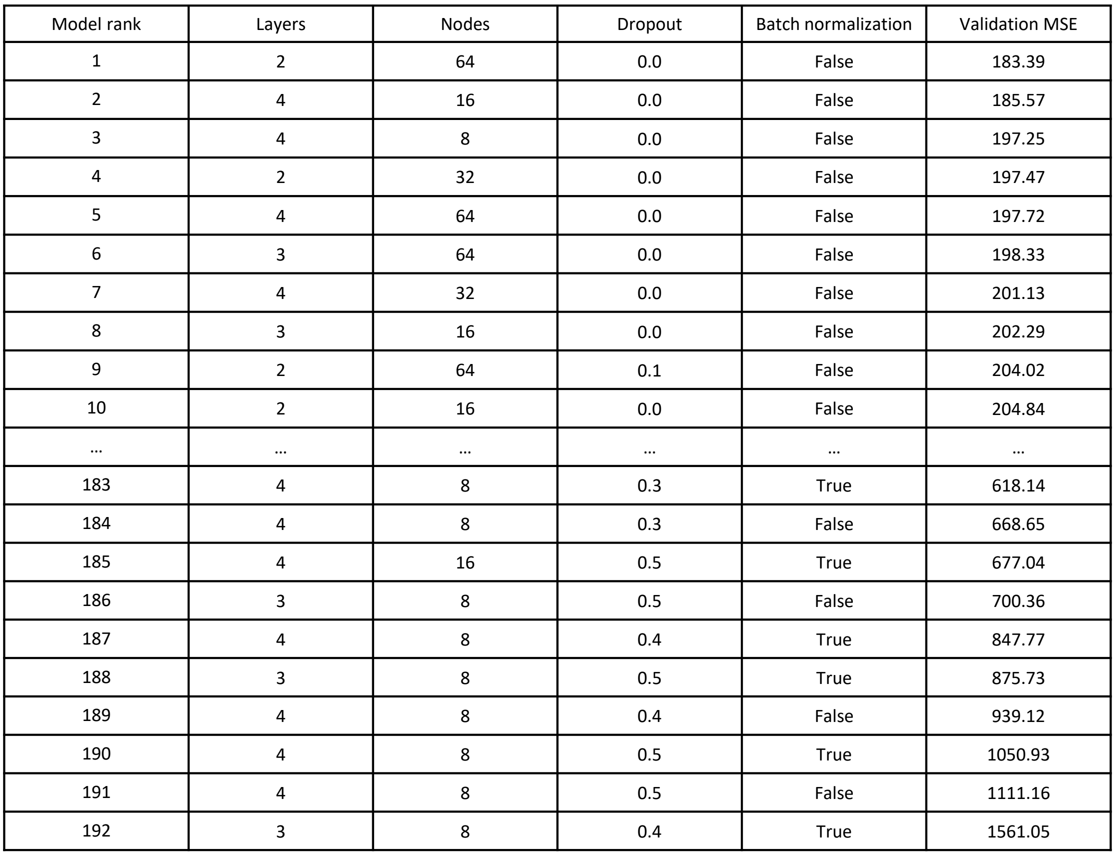
Table 6.21: MSE loss of the 10 best and worst models with SEM instances on the validation set of the 80%/20% datasplit

In Table 6.21 are the 10 best and worst models given with their hyperparameters and lowest recorded MSE loss. From Table 6.21 we observe that the 10 best models use many nodes per layer without batch normalization. The 10 worst models mostly use many layers with a small amount of nodes per layer and high dropout rates. Since the best models primarily use more layers, more nodes and higher dropout rates than the best models in Table 6.15 and 6.18 for field and pad granularity, it implies that there is likely an increase in data complexity.

II. Nominal range

The validation loss curves of the 10 best and worst models on the nominal datasplit are shown in Appendix C Figure 42 and 43. Subfigures (a),

(b) and (c) show the same data, but on different scales to properly see if loss curves converge. Subfigure (a) shows the epochs on a linear x-axis and loss on a linear y-axis, (b) shows the epochs on a linear x-axis and loss on a logarithmic y-axis, and (c) shows the epochs on a logarithmic x-axis and loss on a logarithmic y-axis. The model numbers in the legend of each figure correspond with the model rank in Table 6.22.



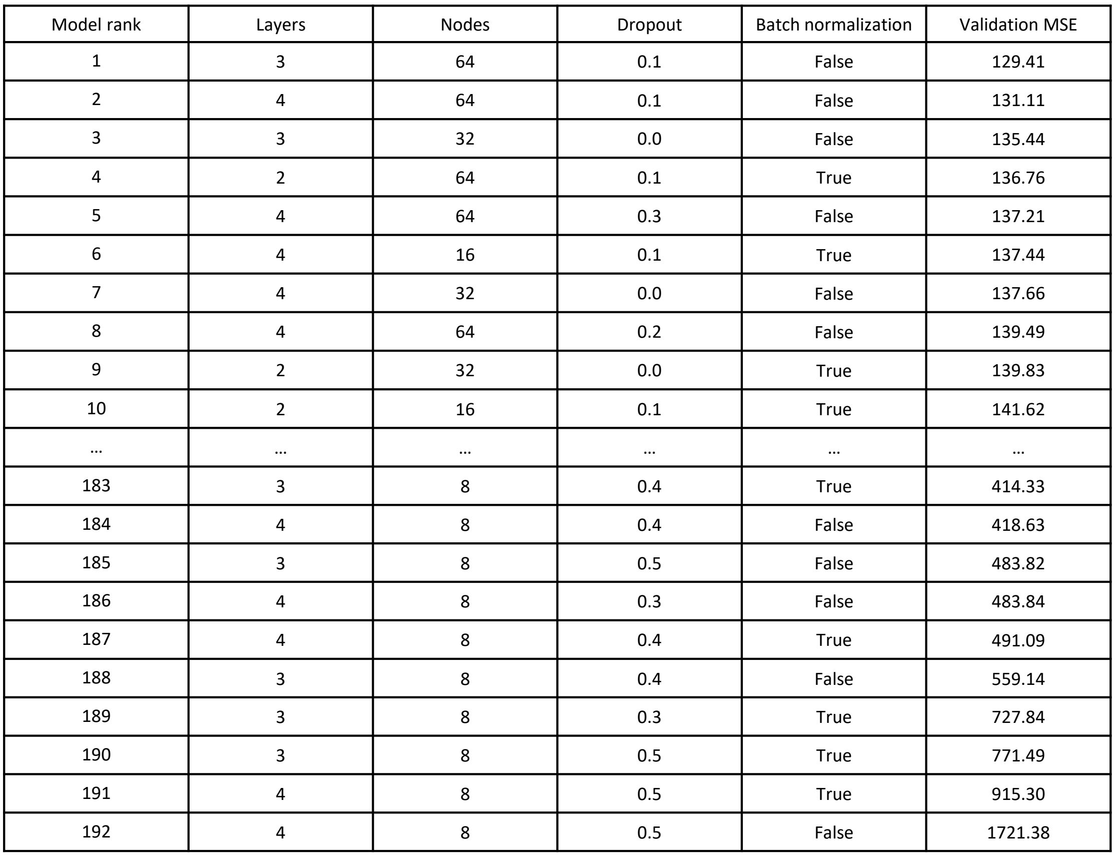
Model rank	Layers	Nodes	Dropout	Batch normalization	Validation MSE
1	2	64	0.0	False	183.39
2	4	16	0.0	False	185.57
3	4	8	0.0	False	197.25
4	2	32	0.0	False	197.47
5	4	64	0.0	False	197.72
6	3	64	0.0	False	198.33
7	4	32	0.0	False	201.13
8	3	16	0.0	False	202.29
9	2	64	0.1	False	204.02
10	2	16	0.0	False	204.84
…	…	…	…	…	…
183	4	8	0.3	True	618.14
184	4	8	0.3	False	668.65
185	4	16	0.5	True	677.04
186	3	8	0.5	False	700.36
187	4	8	0.4	True	847.77
188	3	8	0.5	True	875.73
189	4	8	0.4	False	939.12
190	4	8	0.5	True	1050.93
191	4	8	0.5	False	1111.16
192	3	8	0.4	True	1561.05

Table 6.22: MSE loss of the 10 best and worst model with SEM instancess on the validation set of the nominal datasplit

In Table 6.22 are the 10 best and worst models given with their hyperparameters and lowest recorded MSE loss. From Table 6.22 we observe that most of the 10 best models do not use dropout nor batch normalization. The 10 worst models mostly use high dropout rates, and many layers with a small amount of nodes.

III. On all data

The validation loss curves of the 10 best and worst models are shown in Appendix C Figure 44 and 45. To properly see if loss curves converge subfigures (a), (b) and (c) show the same data on different scales. Subfigure (a) shows the epochs on a linear x-axis and loss on a linear y-axis, (b) shows the epochs on a linear x-axis and loss on a logarithmic y-axis, and (c) shows the epochs on a logarithmic x-axis and loss on a logarithmic y-axis. The model numbers in the legend of each figure correspond with the model rank in Table 6.23.



Model rank	Layers	Nodes	Dropout	Batch normalization	Validation MSE
1	3	64	0.1	False	129.41
2	4	64	0.1	False	131.11
3	3	32	0.0	False	135.44
4	2	64	0.1	True	136.76
5	4	64	0.3	False	137.21
6	4	16	0.1	True	137.44
7	4	32	0.0	False	137.66
8	4	64	0.2	False	139.49
9	2	32	0.0	True	139.83
10	2	16	0.1	True	141.62
…	…	…	…	…	…
183	3	8	0.4	True	414.33
184	4	8	0.4	False	418.63
185	3	8	0.5	False	483.82
186	4	8	0.3	False	483.84
187	4	8	0.4	True	491.09
188	3	8	0.4	False	559.14
189	3	8	0.3	True	727.84
190	3	8	0.5	True	771.49
191	4	8	0.5	True	915.30
192	4	8	0.5	False	1721.38

Table 6.23: MSE loss of the 10 best and worst models with SEM instances on the validation set of the 3σ datasplit

In Table 6.23 are the 10 best and worst models given with their hyperparameters and lowest recorded MSE loss. From Table 6.23 we observe that the 10 best models mostly use many layers with many nodes. The 10 worst models mostly use high dropout rates and many layers with a small amount of nodes per layer.

In general we observe that the MSE loss increases with finer granularity, i.e. from field to pad to SEM. This is caused by the larger standard error of the mean at SEM granularity compared to field and pad granularity. This introduces more noise in the data, which explains the increase in model complexity through the increase in layers, nodes, dropout rate and use of batch normalization as observed in Table 6.21 and Table 6.23. The increase of dropout rates and the use of batch normalization are likely to control overfitting as a result of increasing model complexity. In addition, we can conclude that lower dropout rates tend to perform better than higher dropout rates and that batch normalization often leads to performance decrease.

6.4.2 Testing the models

Simple machine learning models are being performed with 33 input features based on CD, PE_x, PE_y for 11 orthogonal groups from groups 100-130 in Figure 2.4(a). The neural networks will be compared against simpler models, i.e. linear regression, lasso regression, ridge regression and elastic net regression. Following training the hyperparameters used for lasso, ridge and elastic net regression are given in Table 4.12

Algorithm	Hyperparameters		
	Field	Pad	SEM
Lasso	alpha = 0.01	alpha = 0.001	alpha = 0.001
Ridge	alpha = 0.01	alpha = 0.01	alpha = 0.001
Elastic Net	alpha = 0.001	alpha = 0.001	alpha = 0.0001
	l1_ratio = 0.9	l1_ratio = 0.9	l1_ratio = 0.7

Table 6.24: Hyperparameter values of the regression models for the 80%/20% datasplit at different granularity levels.

Algorithm	Hyperparameters		
	Field	Pad	SEM
Lasso	alpha = 0.001	alpha = 0.001	alpha = 0.001
Ridge	alpha = 0.001	alpha = 0.01	alpha = 10^{-6}
Elastic Net	alpha = 0.001	alpha = 0.001	alpha = 0.0001
	l1_ratio = 0.9	l1_ratio = 0.9	l1_ratio = 0.7

Table 6.25: Hyperparameter values of the regression models for the nominal datasplit at different granularity levels.

Algorithm	Hyperparameters		
	Field	Pad	SEM
Lasso	alpha = 0.001	alpha = 0.001	alpha = 0.001
Ridge	alpha = 0.01	alpha = 0.01	alpha = 0.1
Elastic Net	alpha = 0.001	alpha = 0.001	alpha = 0.0001
	l1_ratio = 0.9	l1_ratio = 0.9	l1_ratio = 0.7

Table 6.26: Hyperparameter values of the regression models for the 3σ datasplit at different granularity levels.

From Table 6.24, Table 6.25 and Table 6.26 we observe that the elastic net models have a high L1 ratio. This implies that the models implements a proportionally high L1 penalty and low L2 penalty, which almost boils down to lasso regression.

Next, the models together with the best neural networks from section 6.4.1 are used on the test sets of the 80%/20%, nominal and 3σ dataset. In Table 6.27 the performance metrics are shown for each model and granularity level.

Algorithm	Field			Pad			SEM		
	R^2	RMSE	3σ	R^2	RMSE	3σ	R^2	RMSE	3σ
Linear	0.99	3.34	8.75	0.99	5.24	14.72	0.93	10.90	34.50
Lasso	0.99	3.69	9.69	0.98	5.55	15.26	0.93	10.96	34.55
Ridge	0.99	3.45	9.61	0.99	5.31	14.81	0.93	10.90	34.50
ElasticNet	0.99	3.67	9.95	0.98	5.39	15.23	0.93	10.89	34.53
Neural network	0.98	4.55	8.52	0.98	4.29	11.77	0.95	12.08	19.33

Table 6.27: Test results with R^2 on the 80%/20% datasplit, RMSE on the nominal datasplit, 3σ on all the data at different input granularity levels.

From Table 6.27 we observe that all models give a sufficiently high R^2 value close to zero at all granularity levels, meaning that focus can be properly predicted without too much error. At field and pad granularity the RMSE is the best for neural network and shows a noticeable improvement in particular for the pad granularity level compared to the other regression models. However, the RMSE at SEM granularity level is worse than the regression models. The 3σ is the best for neural network at field, pad and SEM with noticeable improvement in particular at pad and SEM granularity level.

6.4.3 Prediction on dataset 2

The models from section 6.4.2 are now applied on the second dataset. The input values follow from the same output values used on the testset from the first dataset. The results are shown in Table 6.28.

Algorithm	Field			Pad			SEM		
	R^2	RMSE	3σ	R^2	RMSE	3σ	R^2	RMSE	3σ
Linear	0.99	11.09	21.45	0.95	8.15	34.50	0.79	18.31	72.03
Lasso	0.98	4.82	27.25	0.95	8.61	38.36	0.79	18.32	71.91
Ridge	0.99	8.59	24.99	0.95	8.26	36.49	0.79	18.31	72.09
ElasticNet	0.99	3.72	26.19	0.95	8.49	37.64	0.79	18.40	72.24
Neural network	0.95	16.26	26.54	0.93	8.17	25.70	0.85	16.58	53.43

Table 6.28: Test results on the second dataset with R^2 on the 80%/20% datasplit, RMSE on the nominal datasplit, 3σ on all the data at different input granularity levels.

From Table 6.28 we observe that at field granularity the R^2 hardly has changed. At pad granularity the R^2 did change slightly, but the R^2 change is most noticeable at SEM granularity. Compared to Table 6.27 the models at field and pad granularity still give a sufficiently high R^2 value close to zero, but at SEM granularity the neural network has reduced to 0.85 and other regression models have reduced to 0.79. At field granularity level, the RMSE of the linear regression, ridge regression and neural networks models have increased a lot compared to the first dataset, while the RMSE of lasso and elasticnet regression show a relatively small increase. For pad and SEM granularity there is also a RMSE increase present for all models, but on pad granularity level linear regression outperforms the other models, while neural networks is the best at SEM granularity. The 3σ of all models show a big increase on all granularity levels compared to Table 6.27. At field granularity the 3σ is the best for linear regression, neural network performs the best on pad and SEM granularity with big performance improvement with respect to the other models.

6.4.4 Discussion

Similarly to the results in Table 6.13, section 6.4.2 shows that the performance metrics R^2, RMSE and 3σ on the testset deteriorate when using finer granularity levels, i.e. from field to pad to SEM granularity level. Since the standard error of the mean increases, input data becomes noisier, which reflects on the worsening performance metrics.

The performance metrics show that neural network is the best algorithm on field and pad granularity as it returns high R^2 and the best RMSE and 3σ. However the RMSE is higher for neural network at SEM granularity at all granularity levels. This shows that a neural network is suitable for focus prediction and offers improvement over the linear regression baseline for field and pad input instances, but not for SEM due to overfitting on noise.

The performance metrics worsen with finer granularity, which is most noticeable in the RMSE and 3σ of pad and SEM granularity. The best RMSE for field is 3.03 nm , 3.81 nm for pad and 13.78 nm for SEM granularity. The ideal RMSE should be zero, so the RMSE for field and pad granularity is relatively close compared to the RMSE of SEM granularity. The best 3σ for field is 8.52 nm, 11.77 nm for pad and 19.33 nm for SEM granularity. As mentioned before, the expected 3σ at field granularity should be around 10 nm, so 8.52 nm is acceptable. The 11.77 nm 3σ at pad granularity implies that roughly 99.7% of the focus predictions have an error upto 11.77 nm from the mean error. Since this is close to the expected 10 nm of field granularity level and that input data becomes noisier at pad granularity level, the 11.77 nm 3σ could still be considered as acceptable. However, 19.33 nm 3σ at SEM granularity implies that roughly 99.7% of the focus predictions have an error upto 19.40 nm from the mean error, which might be considered as bad as it differs too much from the expected 10 nm 3σ.

When comparing the results from Table 6.13 and 6.27 the R^2 of the two model approaches (i.e. with or without PF features added) give similar performances at all the granularity levels. However, the RMSE and 3σ does not give a solid conclusion on which model is consistently the best among the different granularity levels as the values fluctuate. We have similar observations on the prediction on the second dataset in Table 6.14 and 6.28. Therefore, we conclude that the adding the proximity force features PF_x and PF_y does not improve performance in focus prediction.

6.5 Experiment E: Analysis on discrepancies between dataset 1 and dataset 2

Since the neural network models are not always showing the best performance as expected, we aim to find an explanation for these results. By zooming in on the linear regression models (with regularization) with the 80%/20% data split a possible explanation for the increase in focus prediction errors on the second dataset will be found.

6.5.1 Pearson correlation - Orthogonal groups

There is a possibility that the increase in focus error on the second dataset comes from uncorrelated input features that are part of the model. We investigate this by calculating the Pearson correlation coefficient for each one of the 33 input feature against focus at field, pad and SEM granularity level. The Pearson correlation coefficient ranges between -1 and 1, where -1 means negative correlation, 0 means no correlation and 1 means positive correlation. The correlation coefficients for field, pad and SEM input granularity are shown in Table 6.29, 6.30 and 6.31. In addition, the Pearson correlation coefficient of the proximity force input features in the X and Y-direction for different granularity levels are shown in Table 6.32.

Group	CD	PE_x	PE_y
100	-0.05	-0.07	0.90
110	0.10	0.56	-0.55
111	0.01	-0.49	0.95
112	0.09	-0.54	-0.91
113	-0.01	0.41	-0.96
120	0.16	0.57	0.91
121	0.20	-0.24	-0.71
122	0.16	-0.55	0.88
123	0.15	0.56	-0.98
124	0.05	-0.13	-0.53
125	0.13	-0.53	-0.97

Table 6.29: Pearson correlation between field granularity input features and focus

Group	CD	PE_x	PE_y
100	-0.05	-0.07	0.85
110	0.10	0.56	-0.43
111	0.01	-0.44	0.95
112	0.09	-0.54	-0.82
113	-0.01	0.33	-0.96
120	0.16	0.56	0.88
121	0.20	-0.18	-0.51
122	0.16	-0.55	0.84
123	0.15	0.56	-0.96
124	0.05	-0.09	-0.32
125	0.13	-0.53	-0.95

Table 6.30: Pearson correlation between pad granularity input features and focus

Group	CD	PE_x	PE_y
100	-0.05	-0.04	0.51
110	0.10	0.55	-0.17
111	0.01	-0.23	0.91
112	0.09	-0.53	-0.46
113	-0.01	0.13	-0.92
120	0.16	0.54	0.64
121	0.19	-0.07	-0.14
122	0.16	-0.53	0.55
123	0.15	0.54	-0.79
124	0.05	-0.03	-0.10
125	0.13	-0.51	-0.80

Table 6.31: Pearson correlation between SEM granularity input features and focus

Table 6.29 and Figure 6.37(a) can be observed that the absolute correlation coefficients for CD input features indicated in red are relatively close to zero compared to the absolute correlation coefficients for PE_x shown in green and PE_y shown in blue. Similarly, from Table 6.30 and Figure 6.37(b) for pad granularity input features and from Table 6.31 with Figure 6.37(c) for SEM granularity level we observe that at all granularity levels the absolute correlation coefficients for CD input features are relatively close to zero compared to the absolute correlation coefficients for PE_x and PE_y. This implies that CD is weakly correlated with focus, whereas PE_x and PE_y are stronger correlated with focus. The relatively weak correlation with CD might be explained by the linear relationship that is assumed by the Pearson correlation coefficient, while CD has a quadratic relationship with focus. Note that the correlations becomes weaker when going to finer input granularity, i.e. the correlation between the input features and focus are stronger at field granularity and the weakest at SEM granularity. This is

	PF_x	PF_y
Field	0.61	0.88
Pad	0.61	0.88
SEM	0.61	0.85

Table 6.32: Pearson correlation between proximity force input features and focus at different input granularities.

In Table 6.32 we see that both PF_x and PF_y have a relatively high correlation coefficient among all granularity levels. This means that PF_x and PF_y have a relatively strong correlation with focus. Similarly to input features for CD, PE_x and PE_y we observe that the correlations becomes weaker when going to finer input granularity.

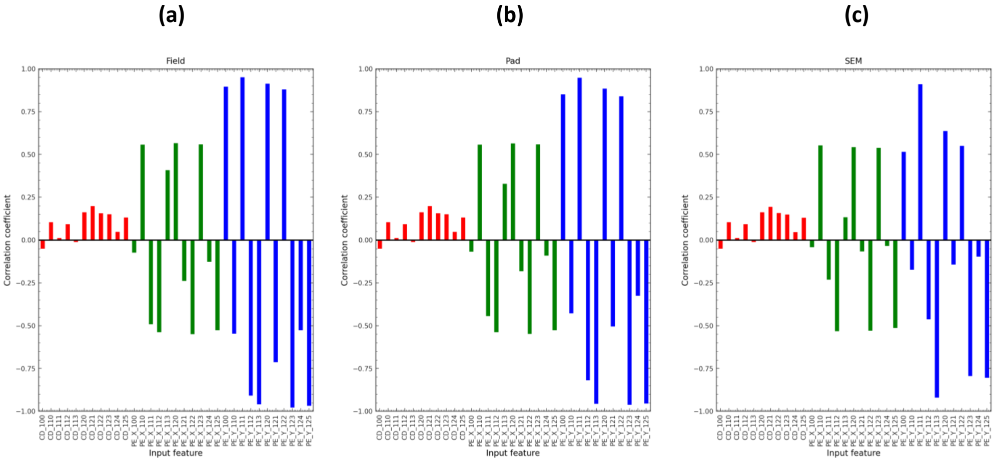


Figure 6.37: Pearson correlation coefficient between input features and focus at different granularity levels. CD features indicated in red, PE_x features indicated in green and PE_y features indicated in blue.

6.5.2 Regression coefficients

The pearson correlation coefficients have shown that CD input variables are weakly correlated with focus. How this reflects in focus prediction will be investigated by looking at the model coefficients. The model coefficients of linear regression, lasso, ridge and elasticnet regression for field, pad and SEM granularity on the 80%/20% random split are shown in Table 6.33, 6.34 and 6.35, respectively.

Input variable	Group	Linear	Lasso	Ridge	Elasticnet
CD	100	187.19	110.41	81.44	116.12
	110	163.51	0	15.23	-10.4
	111	234.33	0	39.75	4.11
	112	64.59	0	13	-17.37
	113	-246.11	0	19.48	-0.16
	120	-166.1	-110.84	-51.88	-70.63
	121	59.02	0	-5.19	-10.92
	122	-87.08	-4.98	-35.32	-21.51
	123	-99.35	0	-31.08	-3.47
	124	-61.15	0	-18.84	11.74
	125	-59.41	-6.74	-41.3	-12.71
PE_x	100	-5.99	-1.51	-1.34	-0.98
	110	-61.63	11.34	6.44	14.18
	111	-7.83	-5.67	-4.6	-4.5
	112	-44.57	-18.15	-27.63	-31.64
	113	-0.05	1.9	1.6	1.69
	120	24.95	-2.88	-9.75	-10.3
	121	-0.18	0.62	0.63	0.68
	122	8.56	0	-3.33	-0.81
	123	20.94	0	8.68	4.88
	124	-0.28	0.35	0.49	0.47
	125	-6.39	7.12	13.26	17.39
PE_y	100	-18.44	-2.53	-9.77	-9.38
	110	-4.19	-2.1	-2.76	-2.77
	111	-32.65	-3.61	-26.94	-26.78
	112	-10.81	-5.93	-8.18	-8.3
	113	-79.59	-21.91	-52.74	-51.29
	120	-4.2	-0.55	-3.98	-4.35
	121	0.88	2	1.75	1.87
	122	0.28	4.19	2.97	3.38
	123	-13.65	-25.15	-24.69	-25.56
	124	-1.41	0.04	-0.36	-0.41
	125	-12.24	-9.25	-10.81	-10.51

Table 6.33: Model coefficients on the 80%/20% random split for field granularity input features.

Input variable	Group	Linear	Lasso	Ridge	Elasticnet
CD	100	208.91	229.15	153.84	174.73
	110	45.66	-64.7	25.84	13.17
	111	113.45	-7.54	71.24	10.17
	112	2.26	-28.15	14.9	-22.96
	113	-128.56	-14.11	-20.08	-10.73
	120	-28.93	-22.46	-33.62	-41.27
	121	16.76	2.68	9.79	-6.15
	122	-51.85	-70.38	-52.83	-59.47
	123	-60.42	-36.65	-57.01	-49.24
	124	-58.27	-1.74	-55.39	-8.39
	125	-75.54	-13.41	-76.25	-22.82
PE_x	100	-4.65	-3.52	-3.75	-2.8
	110	-14.98	16.65	-4.51	8.21
	111	-3.5	-3.39	-3.11	-2.73
	112	-37.38	-31.75	-33.04	-29.75
	113	-1.01	-0.99	-0.88	-0.65
	120	-5.3	-8.33	-5.71	-6.21
	121	-0.1	0.15	0.06	0.29
	122	-4.73	-2.2	-3.8	-2.62
	123	3.26	2.22	2.64	2.24
	124	-0.73	-0.85	-0.7	-0.53
	125	-7.35	-3.21	-6.07	-2.68
PE_y	100	-12.14	-12.76	-12.19	-11.2
	110	-2.79	-2.95	-2.92	-2.8
	111	-13.42	-17.16	-14.84	-13.2
	112	-5.77	-6.41	-5.9	-5.91
	113	-63.45	-63.79	-63.16	-58.57
	120	-2.82	-2.93	-2.71	-2.19
	121	-0.34	-0.45	-0.36	-0.35
	122	-1.8	-1.41	-1.7	-1.17
	123	-12.6	-14.07	-13.35	-13.94
	124	-1.4	-1.5	-1.43	-1.32
	125	-9.01	-10.31	-9.64	-9.85

Table 6.34: Model coefficients on the 80%/20% random split for pad granularity input features.

Input variable	Group	Linear	Lasso	Ridge	Elasticnet
CD	100	214.89	242.81	212.36	211.87
	110	-39.38	-31.55	-39.21	-27.88
	111	74.55	38.48	74.09	53.24
	112	-74.21	-88.14	-73.62	-79.46
	113	-165.56	-166.61	-162.28	-146.09
	120	18.26	19.94	18.01	16.28
	121	7.93	7.38	7.96	7.65
	122	3.36	6.4	3.1	2.95
	123	-31.35	-27	-31.54	-30.99
	124	-4.09	0.56	-4.37	-3.36
	125	-22.01	-18.82	-22.11	-21.09
PE_x	100	3.25	3.07	3.27	3.3
	110	36.95	34.93	36.99	35.31
	111	0.61	0.55	0.62	0.64
	112	6.19	5.02	6.25	5.92
	113	2.43	2.41	2.44	2.48
	120	2.06	1.81	2.07	2.24
	121	1.23	1.29	1.23	1.31
	122	0.87	1	0.84	0.64
	123	1.43	1.47	1.42	1.49
	124	0.4	0.33	0.41	0.45
	125	3.17	3.12	3.19	3.34
PE_y	100	3.71	3.99	3.71	3.93
	110	-0.62	-0.52	-0.62	-0.57
	111	23.38	24.55	23.38	24.15
	112	-1.24	-1.16	-1.24	-1.17
	113	-17.16	-16.93	-17.11	-16.62
	120	2	2.1	2.01	2.1
	121	0.23	0.25	0.24	0.29
	122	1.16	1.24	1.16	1.23
	123	-4.41	-4.35	-4.41	-4.38
	124	0.49	0.52	0.48	0.47
	125	-3.94	-3.89	-3.95	-3.98

Table 6.35: Model coefficients on the 80%/20% random split for SEM granularity input features.

From Table 6.33, 6.34 and 6.35 can be observed that the regression models mostly have coefficients of relatively high absolute magnitude for CD input features compared to PE_x and PE_y input features. This means that a unit increase in the scaled CD input features result in bigger changes in focus prediction than a unit increase in the scaled PE_x and PE_y input features.

6.5.3 Input difference between dataset 1 and dataset 2

When the models are trained, scaling is applied on the input variables based of wafer 1. This scaler is then used to transform the input variables from wafer 2. The difference between the input variables on wafer 1 and wafer 2 might explain why predictions on wafer 2 show big errors. Therefore the mean, maximum and minimum difference in scaled input variables between wafer 1 and wafer 2 is shown in Table 6.36, 6.37 and 6.38 for field, pad and SEM granularity.

Group	CD			PE_x			PE_y		
	Mean	Min.	Max.	Mean	Min.	Max.	Mean	Min.	Max.
100	0.37	0.25	0.59	-0.45	-4.85	1.06	0.64	0.07	1.26
110	0.45	0.26	0.89	0.63	0.09	1.94	-1.86	-4.14	-0.03
111	0.38	0.27	0.61	0.61	-0.3	1.99	0.02	-0.17	0.56
112	0.43	0.25	0.77	-0.57	-1.4	-0.06	-0.19	-0.82	1.12
113	0.38	0.26	0.61	-0.17	-1.74	0.86	-0.07	-0.43	0.09
120	0.44	0.26	0.76	0.57	0.15	1.24	0.41	-0.41	0.95
121	0.48	0.27	0.85	-0.2	-2.27	1.91	-0.3	-2.13	0.73
122	0.45	0.26	0.75	-0.56	-1.15	-0.1	0.53	0.17	0.97
123	0.46	0.27	0.84	0.55	0.01	1.43	-0.41	-0.81	-0.01
124	0.39	0.27	0.62	0.01	-2.01	1.36	0.15	-1.26	1.4
125	0.45	0.27	0.75	-0.58	-1.13	-0.12	-0.29	-0.56	0.08

Table 6.36: Mean, minimum and maximum difference in scaled field input instances between wafer 1 and wafer 2.

Group	CD			PE_x			PE_y		
	Mean	Min.	Max.	Mean	Min.	Max.	Mean	Min.	Max.
100	0.37	0.19	0.65	-0.42	-5.01	1.36	0.62	-0.66	2.02
110	0.45	0.23	0.93	0.63	0.01	2.06	-1.49	-5.16	1.52
111	0.38	0.21	0.67	0.56	-0.86	3.25	0.02	-0.34	0.76
112	0.43	0.23	0.85	-0.57	-1.51	-0.03	-0.17	-2.39	2.28
113	0.38	0.21	0.67	-0.14	-4.56	2.82	-0.07	-0.59	0.29
120	0.44	0.24	0.79	0.57	0.01	1.39	0.4	-1.24	1.64
121	0.47	0.23	0.93	-0.15	-3.34	2.83	-0.22	-2.24	2.39
122	0.44	0.21	0.8	-0.55	-1.21	-0.02	0.51	-0.58	2.28
123	0.46	0.25	0.9	0.55	-0.05	1.49	-0.4	-1.27	0.46
124	0.39	0.21	0.68	0.01	-2.37	2.5	0.1	-2	2.97
125	0.45	0.23	0.84	-0.58	-1.3	0.03	-0.29	-0.93	0.59

Table 6.37: Mean, minimum and maximum difference in scaled pad input instances between wafer 1 and wafer 2.

Group	CD			PE_x			PE_y		
	Mean	Min.	Max.	Mean	Min.	Max.	Mean	Min.	Max.
100	0.37	-0.13	0.7	-0.27	-6.97	3.29	0.39	-3.94	3.87
110	0.45	-0.11	1.15	0.62	-0.25	2.4	-0.6	-4.93	4.44
111	0.38	-0.11	0.71	0.32	-3.81	5.5	0.01	-2.63	1.52
112	0.43	-0.08	1.07	-0.57	-2.04	0.22	-0.07	-4.07	4.13
113	0.38	-0.17	0.74	-0.07	-5.11	3.59	-0.07	-1.62	2.77
120	0.44	-0.13	1.05	0.55	-0.56	2.53	0.33	-2.78	4.44
121	0.47	-0.16	1.27	0.03	-4.58	4.93	-0.01	-3.84	5.92
122	0.45	-0.03	1.06	-0.54	-2.71	0.67	0.32	-4.24	4.49
123	0.46	-0.05	1.28	0.54	-0.63	2.78	-0.36	-2.92	1.92
124	0.39	-0.26	0.83	-0.01	-4.38	4.01	-0.01	-4.58	4.85
125	0.45	-0.12	0.99	-0.55	-2.3	0.65	-0.22	-2.96	2.64

Table 6.38: Mean, minimum and maximum difference in scaled SEM input instances between wafer 1 and wafer 2.

In Table 6.36, 6.37 and 6.38 can be observed that the mean difference between wafer 1 and wafer 2 remains roughly the same for all input variables among different granularities. When going to SEM granularity input variables, the difference between wafer 1 and wafer 2 becomes more profound compared to field and pad granularity as the minima decrease and maxima increase.

6.5.4 Discussion

The pearson correlation coefficients in Table 6.29, 6.30 and 6.31 have shown that the CD input features are weakly correlated with focus. This could mean that CD actually has a negative contribution to focus prediction. As the model coefficients in Table 6.33, 6.34 and 6.35 show that the absolute magnitude is mostly the highest for CD compared to PE_x and PE_y, it means that the weakest correlated input features of the models have the most impact on focus prediction. The mean difference in scaled input features in Table 6.36, 6.37 and 6.38 multiplied by their respective model coefficient confirms that the change in CD between wafer 1 and wafer 2 is likely causing the increase in focus prediction errors on wafer 2.

Chapter 7

Discussion

7.1 Result evaluation

7.1.1 Granularity levels on the first dataset

In section 6.2 Experiment (B) has shown how a neural network performs against other regression models for different input granularity levels. When comparing the neural network performance metrics for the different input granularities we end up with Figure 7.1.

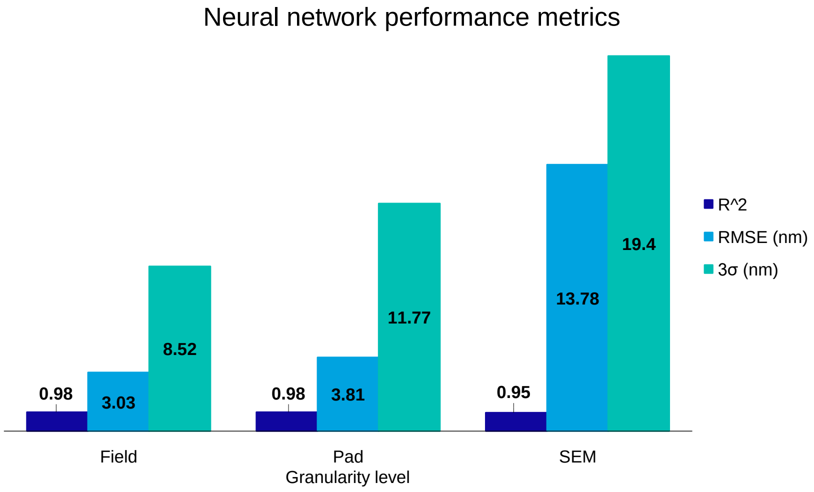


Figure 7.1: Neural network performance metrics at different granularity levels.

In Figure 7.1 the R^2 remains sufficiently large for field, pad and SEM granularity level, implying that a sufficient fit can be made. Further, we see that the RMSE increases slightly between field and pad granularity level and starts to increase a lot with SEM granularity level. Similarly for the 3σ score, there is a relatively small increase between field and pad granularity level, while the increase between pad and SEM is relatively big. This expected behaviour follows from the increase in the standard error of the mean with finer granularity as mentioned in section 6.2.4.

Since the performance is worsening on all three performance metrics when switching to finer granularity levels, we can conclude that in general performance deteriorates with finer granularity levels. In addition, it is likely that SEM granularity input instances are not suitable for proper focus prediction as the RMSE and 3σ are deteriorating too much compared to field granularity level.

7.1.2 Proximity Force feature

In section 6.4 Experiment (D) we have developed models for focus prediction using the new Proximity Force feature defined in section 6.3 Experiment (C). To observe the effect of the new feature as an alternative approach to the orthogonal grouping scheme, the performance metrics of the orthogonal grouping models with and without PF feature added are shown below for different input granularity levels on the first dataset. The R^2, RMSE and 3σ of the best model, i.e. neural network, are shown in Figure 7.2, 7.3 and 7.4 respectively.

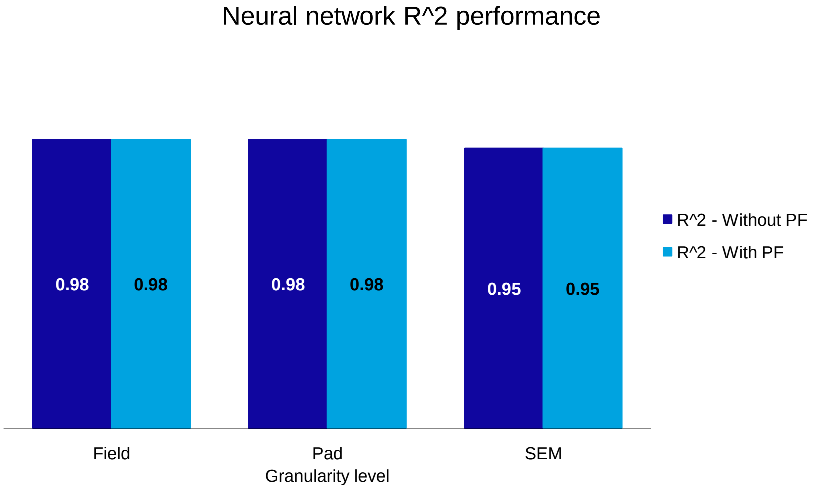


Figure 7.2: R^2 on orthogonal grouping models with and without PF at different input granularity levels.

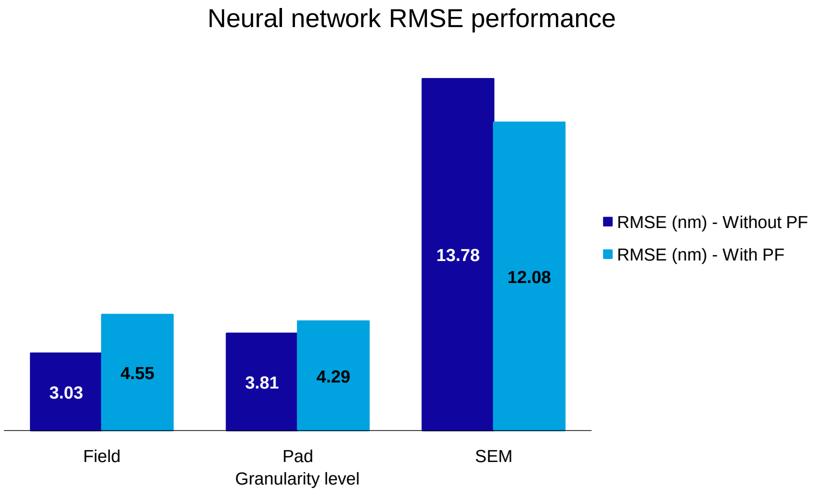


Figure 7.3: RMSE on orthogonal grouping models with and without PF at different input granularity levels.

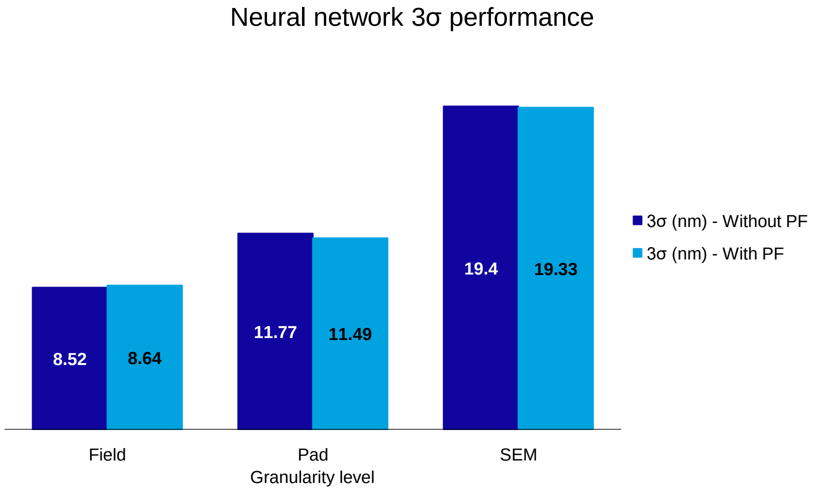


Figure 7.4: 3σ on orthogonal grouping models with and without PF at different input granularity levels.

Figure 7.2 shows that on all three granularity levels there is no change in R^2 between with or without PF feature added to the models. In Figure 7.3 the RMSE on field and pad granularity level increases when adding the PF feature to the models, while on SEM granularity level the RMSE decreases with PF added. We expect to see a trend in increasing RMSE with finer granularity levels as mentioned in section 6.2.4 and 6.4.4. It is noteworthy that the trend of increasing RMSE with finer granularity is visible in the orthogonal grouping models without PF, but not visible with models using the PF feature. The difference in RMSE between the neural network with and without PF might be the result of splitting the dataset with this random seed, so a different seed might give different results in which the increasing RMSE trend does become visible. The increasing RMSE trend is visible on the 3σ in Figure 7.4. The 3σ measure also shows similar performance for the models with or without PF within each granularity level. Given that for the models with PF the R^2 and 3σ show similar behaviour and that RMSE at some granularity levels show worse performance, it shows that the new PF feature does not give added value to the existing orthogonal grouping model without PF.

7.1.3 Comparison between dataset 1 and 2

In Experiment (B) and (D) we have trained models on the first dataset and applied them on both the first and second dataset. We will zoom in on the performance metrics of the best model for both datasets, i.e. neural network. In Figure 7.5 the 3σ on all the data of the first and second dataset for different granularity levels is shown. The figure shows that with finer granularity the 3σ increases on both datasets and that at all granularity levels the 3σ of the second dataset is considerably bigger than for the first dataset. This consistency in increasing 3σ is due to the increasing standard error of the mean with finer granularity level input instances as discussed in section 4.1 and 6.2.4.

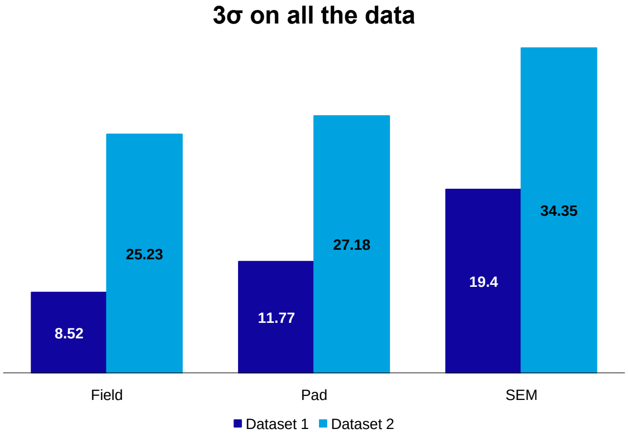


Figure 7.5: 3σ on all the data of the first and second dataset at different granularity levels.

In Figure 7.6 the RMSE on the nominal focus datasplit of the first and second dataset for different granularity levels is shown. On the first dataset the RMSE increases with finer granularity levels, which is due to the increase in the standard error of the mean. As mentioned in section 6.2.4, the increasing RMSE with finer granularity on the first dataset is expected due to the increasing standard error of the mean. The RMSE on the second dataset should ideally follow a similar trend, but from Figure 7.6 this cannot be observed. The RMSE on the second dataset is actually the highest at field granularity level and relatively low at SEM granularity level. From the big differences in 3σ and RMSE between the two datasets at all granularity levels it follows that the neural network models trained on the first dataset cannot be applied on the second dataset.

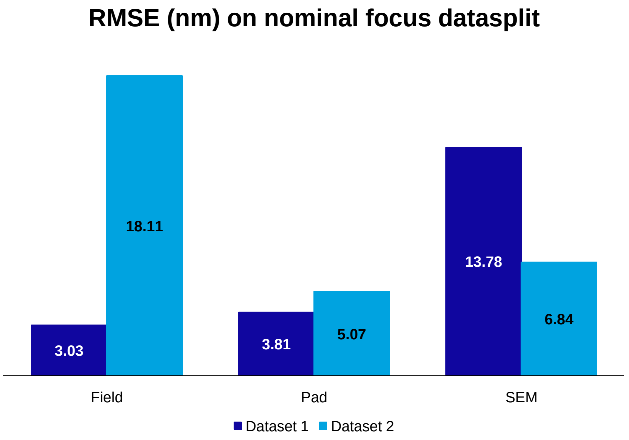


Figure 7.6: RMSE on the nominal focus datasplit of the first and second dataset at different granularity levels.

7.1.4 Explanation on performance decrease on the second dataset

In Experiment (E) we have discussed how the difference in performance between the first and second dataset might come from input features with bad correlation with focus. In this section we further investigate how the coefficients of input features in linear regression, lasso, ridge and elasticnet might explain the performance decrease on the second dataset. In Figure 7.7, coefficients of linear regression, lasso, ridge and elasticnet regression with field input granularity are plotted using the coefficients from Table 6.33. Similar figures for pad and SEM granularity levels are shown in Appendix D Figure 46 and 47 using Table 6.34 and 6.35 respectively.

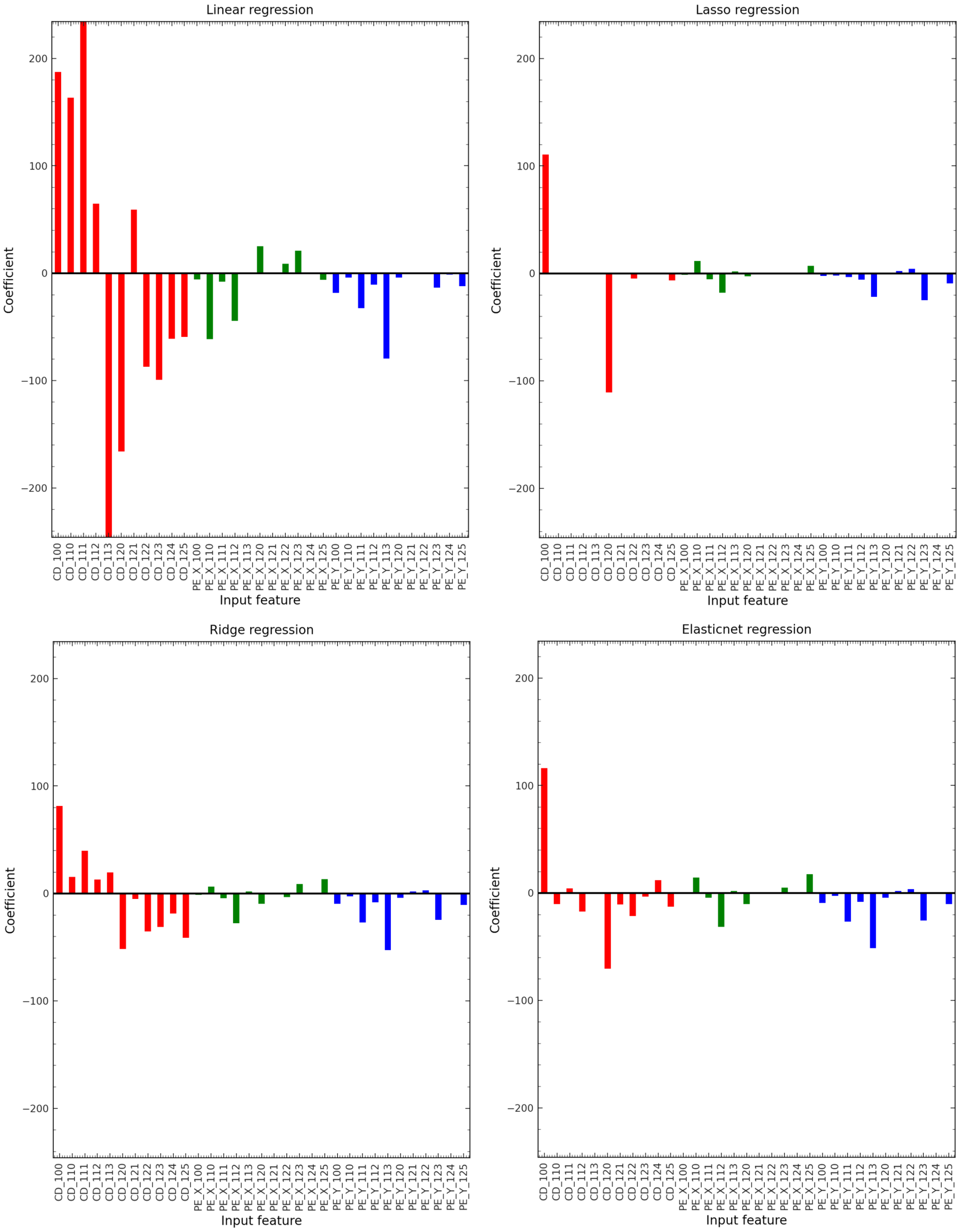


Figure 7.7: Coefficients of each input feature per model using field granularity input instances.

In Figure 7.7 we observe that for linear regression the coefficients for the CD input features on absolute scale are very large meaning that the CD features are dominant in focus prediction. This trend in which CD is more dominant than PE_x and PE_y is also observed in lasso and elasticnet regression. In ridge regression the CD, PE_x and PE_y look more similar in magnitude, but in terms of quantity there is a larger amount of CD features with high coefficient than for PE_x and PE_y. In lasso regression we also observe that some of the features have coefficient zero as lasso regression tends to bring coefficients to zero to minimize the loss. We see then that there are more valuable PE_x and PE_y features than for CD. However, it still shows that CD is very dominant as the coefficients in CD are much larger than for PE_x and PE_y. The same dominance of CD features in the models at pad and SEM granularity level can be observed in Appendix D Figure 46 and 47.

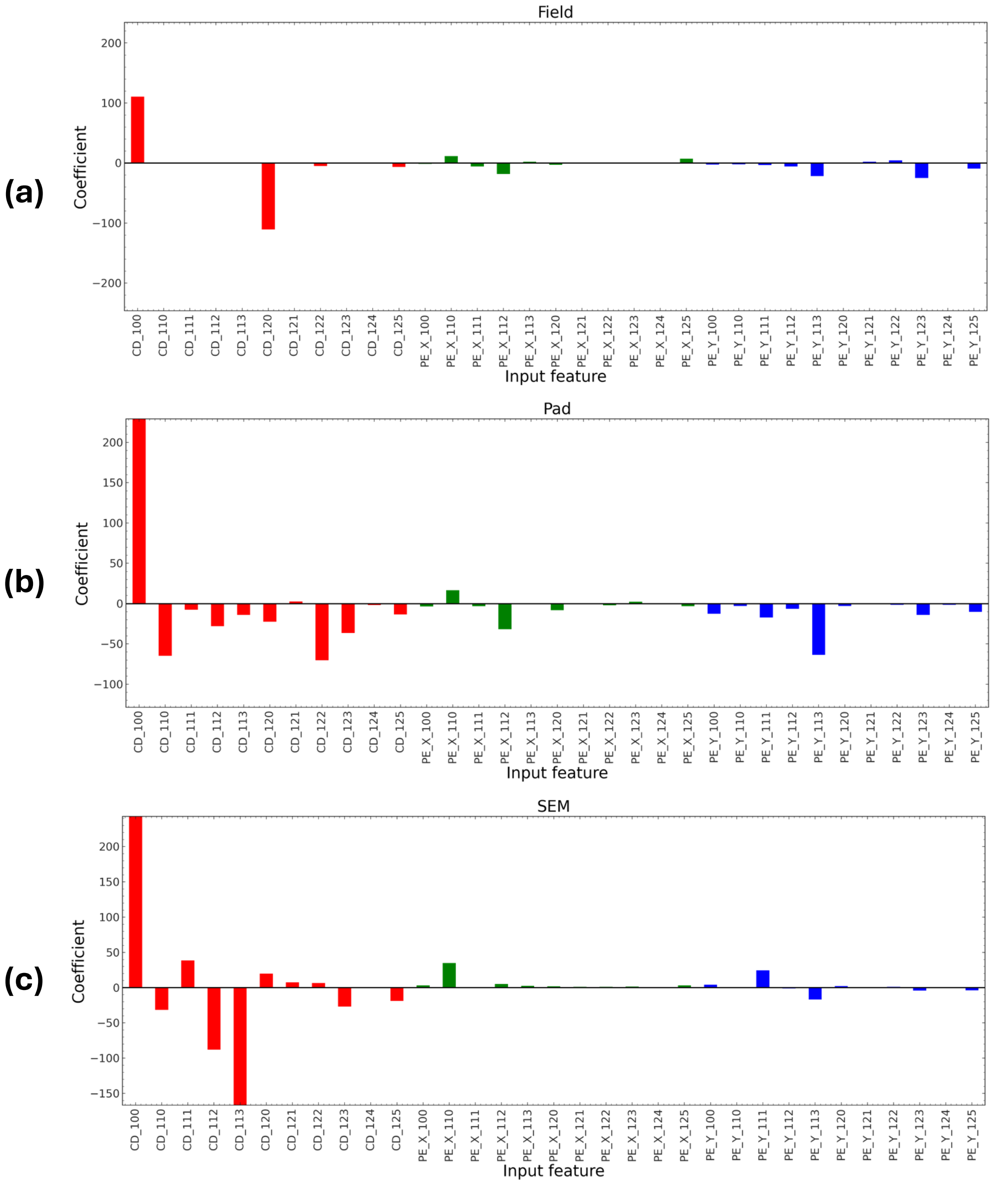


Figure 7.8: Coefficients per input feature of the lasso regression model at field (a), pad (b) and SEM (c) granularity level.

In Figure 7.8 are the coefficients for each feature of the lasso regression models given at field (a), pad (b) and SEM (c) input granularity level. In Figure 7.8(b) and (c) we observe that in lasso regression all the coefficients are now greater than zero compared to field granularity level in Figure 7.8(a). This means that all features are now contributing to focus prediction, which is also visible in the coefficients in Table 6.33, 6.34 and 6.35 of Experiment

(E) for field, pad and SEM granularity respectively. This change in coefficients might be due to the increase in data complexity following the different aggregation.

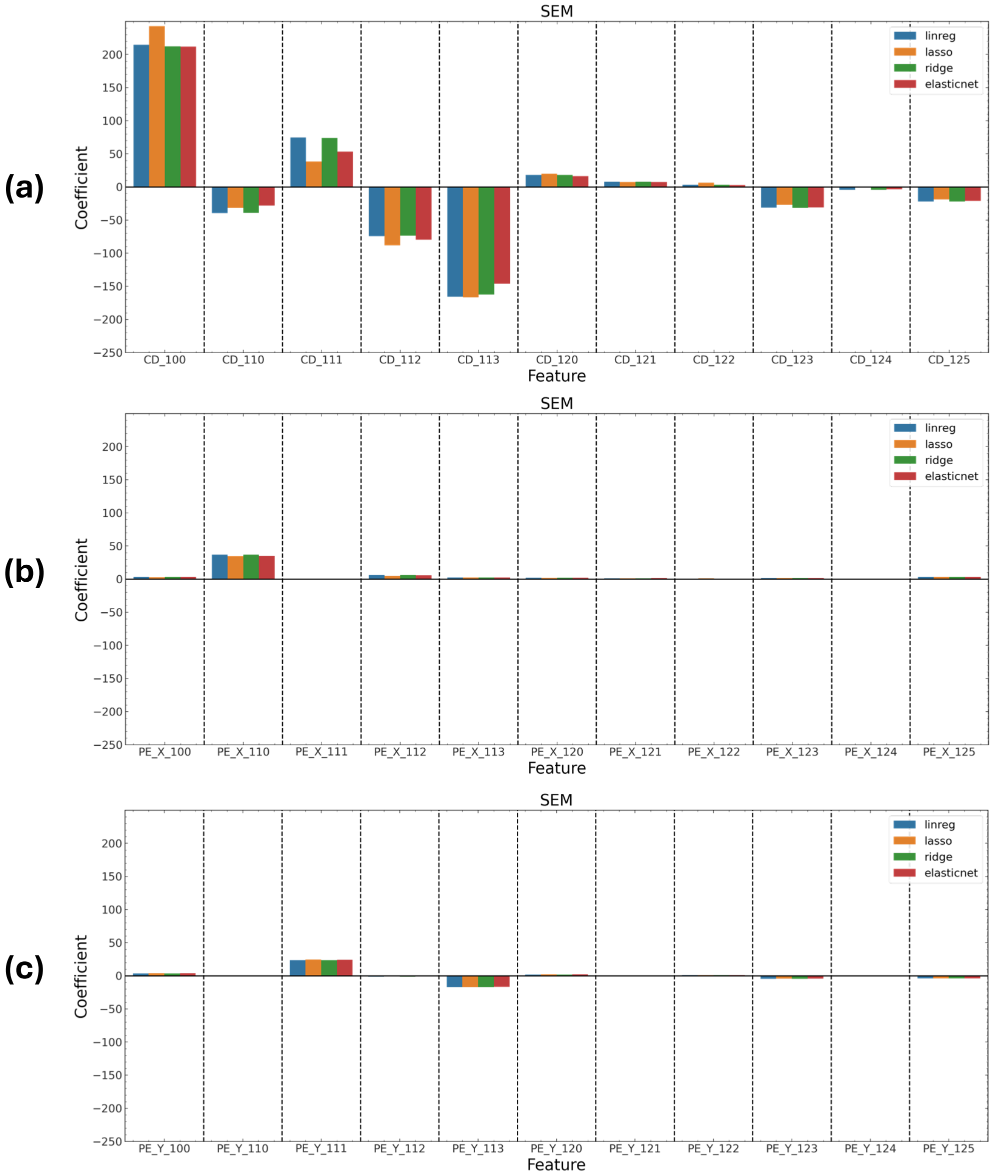


Figure 7.9: Coefficients of CD (a), PE_x (b) and PE_y (c) input features of linear models using SEM granularity input instances.

In Figure 7.9 again we observe that CD features with SEM granularity input instances have the largest coefficients in absolute magnitude. However, all the models are now roughly the same, meaning that there is hardly any gain to achieve with regularization compared to simple linear regression.

In addition we observe a trend in Figure 7.7, Appendix D Figure 46 and 47 in which the influence of PE_x and PE_y among all the three types of granularity are hardly changing, i.e. in all models the coefficients are roughly the same. This implies that the choice of features to be important in all the models are pretty consistent among the models and among different granularity levels. The change between models within and among different granularities are mostly coming from the CD features, which imply that CD features are heavily influencing the performance in focus prediction. Since CD follows a quadratic relationship with focus as mentioned in section 3.2, it might be more difficult for the linear regression models to properly capture this non-linear relationship for focus prediction, while it is easier to properly capture the linear relationship between focus and PE_x and PE_y as mentioned in section 3.3. Together with the different CD in design in the second dataset it explains why the performance on the second dataset deteriorates.

Following Figure 7.7, 7.8 and 7.9 we conclude that the decrease in performance on the second dataset are due to the models being to sensitive to CD features. Given that the second dataset has a different CD distribution it is likely that the CD sensitivity causes the performance decrease.

7.2 Considerations and future work

Since the first and second dataset differ in their CD in design and their CD distribution, it would be better to have two datasets with an equal CD distribution. With the second dataset two variables in our data changed, which are firstly the properties of the wafer during exposure and secondly the change of mask biased CD. The changes in wafer characteristics are always present in the usecase, so the generalizibility of the models for this change in variable of interest should be checked. To properly check for the generalizibility we should keep other variables of interest that influences the input, i.e. the CD in design, constant. Therefore we recommend having two wafers with the same CD in design as our first and second dataset.

In general, it would have been preferred to even have more wafers as then we can train across wafers rather than on only one. Furthermore, the FEM wafer structure contains focus and dose values that are not often used in practice. We therefore opt to create a FEM wafer that contains smaller focus and dose ranges such that the ranges correspond more with practice.

Following the observations of the models being sensitive to CD change and that CD is not correlating well with focus it might be interesting to investigate how the performance changes when CD is excluded from the models. By excluding the CD from the models, only the features PE_x and PE_y are maintained, which are the features that correlate better with focus.

In this research the trained models from the first dataset are directly applied on the second dataset. Since we are interested in the best algorithm as models are heavily dependent on chip layout, it would be worthwhile to retrain a new model on the second dataset for future work. Ideally the same algorithm and the same hyperparameters should be returned. In order to investigate this, the trained hyperparameters from the models on the first dataset could be reused such that only the coefficients of a new model are different, or new model would be developed by redoing hyperparameter tuning.

We have investigated models that take aggregated inputs from different granularity levels, i.e. field, pad and SEM images. Results have shown that field and pad granularity offer performance metrics that are relatively close to eachother compared to SEM images. This has opened the door towards further research into pad granularity input instances for prediction models as it is for metrology a more costeffective alternative to field granularity level. Therefore we opt for further research into finer granularity aggregation methods.

Furthermore, the training, validation and test set for the 80%/20% datasplit and the training and validation set for the nominal and 3σ datasplits involves randomly picking a subset from the data, so that the distribution of the subset in theory follows the same distribution as the original data. In practice, these distributions can vary due to the randomness. Therefore, ideally we should perform the datasplits multiple times with different random splits. With multiple tests we can define a mean and standard error for R^2, RMSE and 3σ to exclude the possibility that results from a single test are a lucky shot.

In section 2.3 we have discussed possible machine learning algorithms for focus error prediction, but we were not able to implement an alternative model that we took into consideration, i.e. LinXGBoost. LinXGBoost is a gradient-boosting machine with linear regressions in the leaf nodes of each boosted tree, making it a piece-wise linear model. Due to software limitations regarding multithreading and GPU support together with the time constraints of this research we were not able to implement this, therefore it might interesting for further research. An additional proposal for further research might be the implementation of transfer learning. With transfer learning we can train any neural network on a collection of wafers with the same chip layout and different CD in design in order to get a pre-trained model. Then the pre-trained model can be finetuned for a specific CD in design for which focus should be predicted.

7.3 Research questions

In Experiment (B) and Experiment (D) we have developed focus predicting models by taking CD, PE_x and PE_y as input features. In addition, the distance between contact holes is used for a new feature as defined in Experiment (C). The new feature describes quantitatively the proximity effect and tackles the drawbacks of the original orthogonal grouping approach used in Experiment (B). With these variables of interest we have answered research question RQ1.

We have answered research question RQ2 in section 2.3 by providing a list of candidate machine learning algorithms that are suitable for a regression problem and are not piece-wise constant. We have proposed besides the linear regression from previous work lasso, ridge and elasticnet regression and feedforward neural network.

Research question RQ3 involves comparing model performance per granularity level and datasplit. The results of these comparisons on both the first and second dataset are given in Table 6.13, 6.27, 6.14 and 6.28. It often shows that the baseline linear regression performs better than the regularized regression models. However, it is frequently less superior than a neural network, primarily at field and pad granularity levels. Neural network has shown that it is able to outperform or to be just as good as the baseline and the remaining models. From the different models at different granularity levels we have observed that focus error tends to increase with finer granularity. This is in particular noticeable at SEM granularity level, therefore we recommend to develop a focus prediction model beyond the level of SEM input instances.

To answer the main research question we look at the results from Experiment (B). From these results we conclude that neural network is the best algorithm on the first dataset as it performs well on R^2 and 3σ on all granularity levels and well on RMSE for field and pad granularity levels. However, the generalizibilty of all the models might not have been thoroughly tested as Experiment (E) has shown that models might be biased towards the uncorrelated input features of CD, while also the second dataset for testing follows a different CD distribution than in the first dataset. This likely leads to the big difference in performance between the two datasets.

Chapter 8

Conclusion

In this research we have found the best machine learning algorithm to predict focus by means of CD, PE_x and PE_y for each group in an orthogonal classification scheme. By investigating simple regression models (linear regression, lasso, ridge, elasticnet regression) and a feedforward neural network on two datasets and quantifying the performance with the R^2, RMSE and 3σ we were able to find the most suitable algorithm for focus prediction. In addition, we have investigated the effect of different input granularity levels in focus prediction, i.e. aggregating on different amount of data.

In Experiment (A) we have estimated the minimum amount of epochs required to properly train a neural network. With this minimum amount of epochs we aim to avoid underfitting.

Following our estimations from Experiment (A), in Experiment (B) we have developed simple regression models and a neural network using field, pad and SEM input granularity. Following the results from the first dataset we conclude that neural network is the best algorithm for focus prediction under the premise that input is at field or pad granularity. Further it is observed that the trained model from the first dataset cannot be applied on the second dataset due to the different CD in design.

In Experiment (C) we have defined a new feature that leverages the optical proximity effect and addresses the drawbacks of the orthogonal grouping approach. By empirically testing out several inverse-distance relationships we selected a relationship based on the Bessel function to describe the strength of the proximity effect.

In Experiment (D) we have duplicated Experiment (B), albeit that the models also use the new feature defined in Experiment (C). Following the results from the first dataset we conclude that using the new feature in the models do not give great performance gain. Again it is observed that the trained model from the first dataset cannot be applied on the second dataset following the big performance decrease.

Experiment (E) investigates the difference between the first and second dataset to find a possible cause for the bad performance on the second dataset in Experiment (B) and (D). It has shown that in the simple linear models CD has a high impact on the focus prediction models.

Following the experiments that have been performed in this research, we can conclude that in general feedforward neural network is the best algorithm to be used for focus prediction as it shows improvement over the linear regression baseline used in previous work. In addition, we have shown that with finer granularity more focus errors in the models occur. Focus prediction models using field and pad input granularity offer reasonable performance, while SEM input granularity gives big declines in performance. Therefore we recommend to develop a focus prediction model beyond the level of SEM input instances.

Bibliography

[1] Chris Mack. *Fundamental principles of optical lithography: the science of microfabrication.* John Wiley & Sons, 2007.

[2] Intel. *Moore's Law.* URL: `https://www.intel.com/content/www/us/en/newsroom/resources/moores-law.html` (visited on 07/24/2024).

[3] ASML. Private photo database of ASML. (Visited on 08/27/2024).

[4] Willem op't Root et al. "Optical diffraction-based methodology to measure on-product EUV exposure focus variations". In: *Metrology, Inspection, and Process Control XXXVIII.* Vol. 12955. SPIE. 2024, pp. 696–708. DOI: `10.1117/12.3010370`.

[5] Victor Calado et al. "Focus sensing using placement and CD variation for high NA EUV lithography". In: *Optical and EUV Nanolithography XXXVII.* Vol. 12953. SPIE. 2024, pp. 62–69. DOI: `10.1117/12.3010835`.

[6] ASML. *History.* URL: `https://www.asml.com/en/company/about-asml/history` (visited on 07/15/2024).

[7] Azad Mohammed and Avin Abdullah. "Scanning electron microscopy (SEM): A review". In: *Proceedings of the 2018 International Conference on Hydraulics and Pneumatics—HERVEX, Băile Govora, Romania.* Vol. 2018. 2018, pp. 7–9.

[8] Anwar Ul-Hamid. *A beginners' guide to scanning electron microscopy.* Vol. 1. Springer, 2018.

[9] ASML. *HMI eP5.* URL: `https://www.asml.com/en/products/metrology-and-inspection-systems/hmi-ep5` (visited on 07/28/2024).

[10] Jan Mulkens et al. "High order field-to-field corrections for imaging and overlay to achieve sub 20-nm lithography requirements". In: *Optical Microlithography XXVI.* Vol. 8683. SPIE. 2013, pp. 500–512. DOI: `10.1117/12.2011550`.

[11] MEETOPTICS. *Groove Density.* URL: https://www.meetoptics.com/academy/groove-density (visited on 07/28/2024).

[12] Amine Lakcher et al. "On-product focus monitoring and control for immersion lithography in 3D-NAND manufacturing". In: *Metrology, Inspection, and Process Control for Microlithography XXXIV*. Vol. 11325. SPIE. 2020, pp. 339–352. DOI: 10.1117/12.2552930.

[13] Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning.* Vol. 4. 4. Springer, 2006.

[14] Robert Tibshirani. "Regression shrinkage and selection via the lasso". In: *Journal of the Royal Statistical Society Series B: Statistical Methodology* 58.1 (1996), pp. 267–288. DOI: 10.1111/j.2517-6161.1996.tb02080.x.

[15] Hui Zou and Trevor Hastie. "Regularization and variable selection via the elastic net". In: *Journal of the Royal Statistical Society Series B: Statistical Methodology* 67.2 (2005), pp. 301–320. DOI: 10.1111/j.1467-9868.2005.00527.x.

[16] Warren Flack et al. "Lithography technique to reduce the alignment errors from die placement in fan-out wafer level packaging applications". In: *2011 IEEE 61st Electronic Components and Technology Conference (ECTC).* IEEE. 2011, pp. 65–70. DOI: 10.1109/ECTC.2011.5898493.

[17] Dandan Han et al. "Enhancement of pattern quality in maskless plasmonic lithography via spatial loss modulation". In: *Microsystems & Nanoengineering* 9.1 (2023), p. 40. DOI: 10.1038/s41378-023-00512-4.

[18] *Python.* URL: https://www.python.org/ (visited on 07/19/2024).

[19] *Pandas.* URL: https://pandas.pydata.org/ (visited on 07/19/2024).

[20] *NumPy.* URL: https://numpy.org/ (visited on 07/22/2024).

[21] *Matplotlib: Visualization with Python.* URL: https://matplotlib.org/ (visited on 07/19/2024).

[22] *Scikit-learn: Machine Learning in Python.* URL: https://scikit-learn.org/stable/ (visited on 07/19/2024).

[23] *PyTorch.* URL: https://pytorch.org/ (visited on 07/19/2024).

[24] *Welcome to Ray.* URL: https://docs.ray.io/en/latest/index.html (visited on 10/22/2024).

[25] Scikit Learn. *3.1. Cross-validation: evaluating estimator performance.* URL: https://scikit-learn.org/stable/modules/cross_validation.html (visited on 07/16/2024).

[26] Sebastian Ruder. “An overview of gradient descent optimization algorithms”. In: *arXiv preprint arXiv:1609.04747* (2016). DOI: 10.48550/arXiv.1609.04747.

[27] K Gnana Sheela and Subramaniam N Deepa. “Review on methods to fix number of hidden neurons in neural networks”. In: *Mathematical problems in engineering* 2013.1 (2013), p. 425740. DOI: 10.1155/2013/425740.

[28] Shuxiang Xu and Ling Chen. “A novel approach for determining the optimal number of hidden layer neurons for FNN’s and its application in data mining”. In: (2008).

[29] Yoshinari Minami. “Another derivation method of the formula of universal gravitation”. In: *Science and Technology Publishing (SCI & TECH)* 4.6 (2020), pp. 291–296.

[30] Geeks for Geeks. *Single-Slit Diffraction.* URL: https://www.geeksforgeeks.org/single-slit-diffraction/ (visited on 12/04/2024).

[31] Max Born and Emil Wolf. *Principles of optics: electromagnetic theory of propagation, interference and diffraction of light.* Elsevier, 2013.

[32] Joseph Ivin Thomas. “Geometrization of the Huygens–Fresnel principle: Applications to Fraunhofer diffraction”. In: *AIP Advances* 14.5 (2024). DOI: https://doi.org/10.1063/5.0191874.

List of Figures

List of Tables

Appendices

Appendix A - Experiment A additional loss curves

Field granularity

II. Nominal range

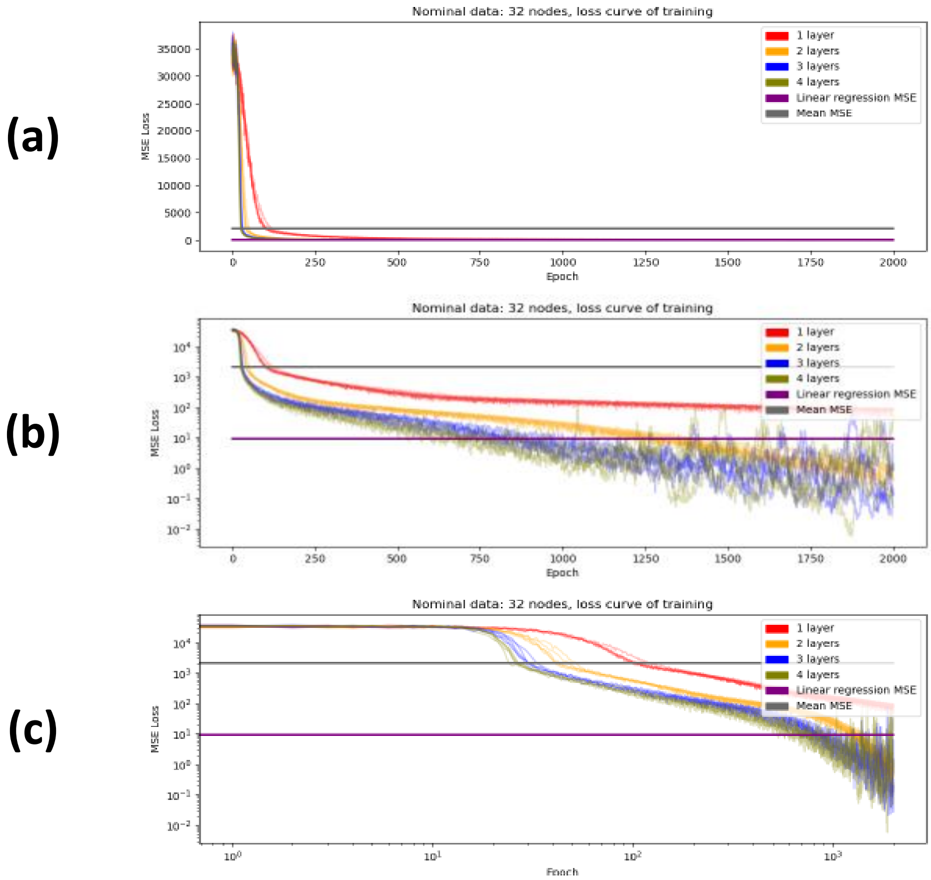


Figure 1: MSE training loss of five fits using field granularity inputs with 32 nodes per a hidden layer. (a) Epochs and loss on linear scale. (b) Epochs on linear scale and loss on logarithmic scale. (c) Epochs and loss on logarithmic scale.

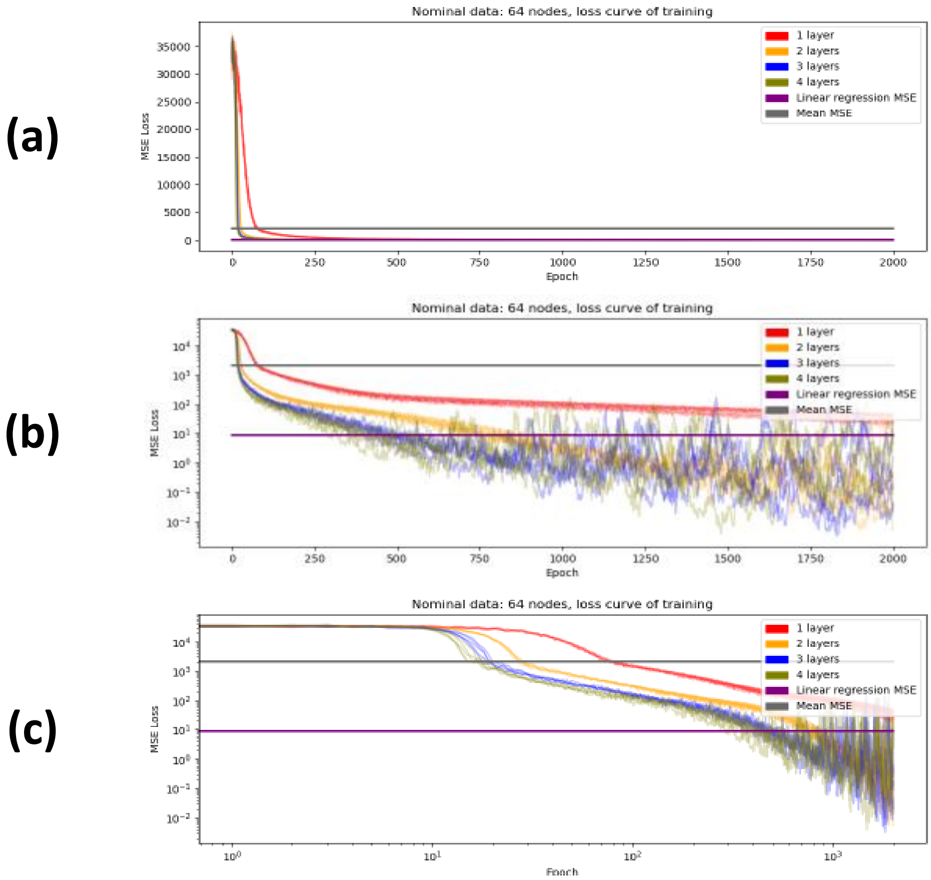


Figure 2: MSE training loss of five fits using field granularity inputs with 64 nodes per a hidden layer. (a) Epochs and loss on linear scale. (b) Epochs on linear scale and loss on logarithmic scale. (c) Epochs and loss on logarithmic scale.

III. On all data

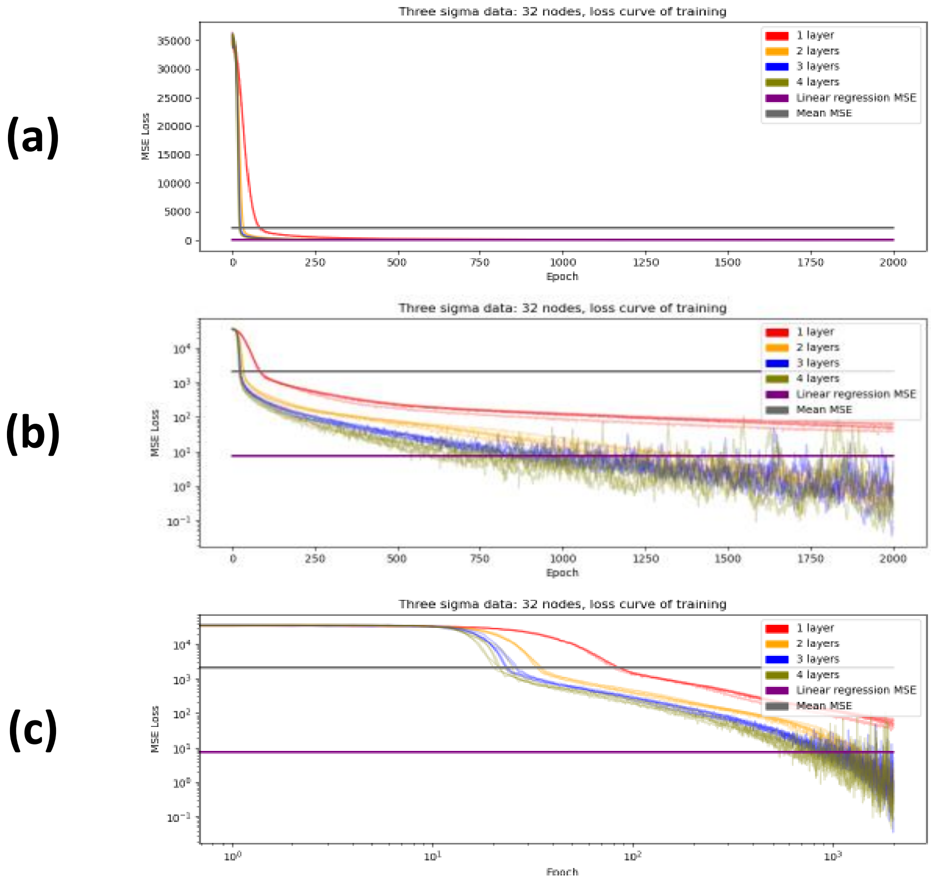


Figure 3: MSE training loss of five fits using field granularity inputs with 32 nodes per a hidden layer. (a) Epochs and loss on linear scale. (b) Epochs on linear scale and loss on logarithmic scale. (c) Epochs and loss on logarithmic scale.

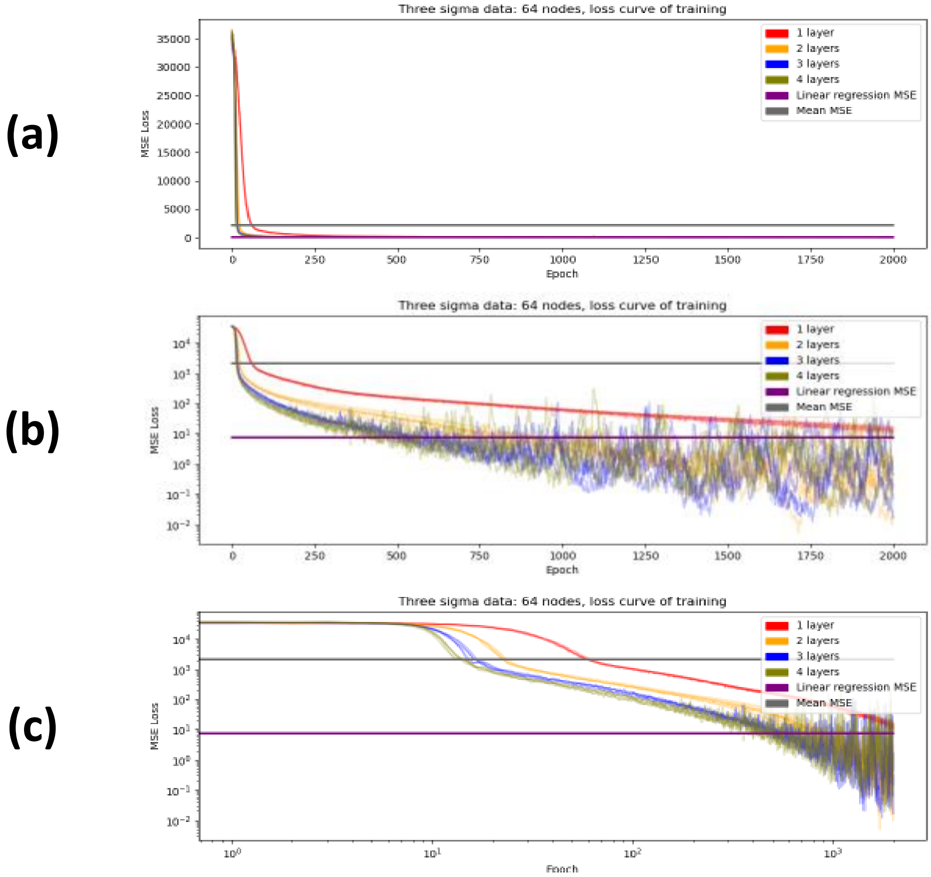


Figure 4: MSE training loss of five fits using field granularity inputs with 64 nodes per a hidden layer. (a) Epochs and loss on linear scale. (b) Epochs on linear scale and loss on logarithmic scale. (c) Epochs and loss on logarithmic scale.

Pad granularity

I. 80%/20% random data split

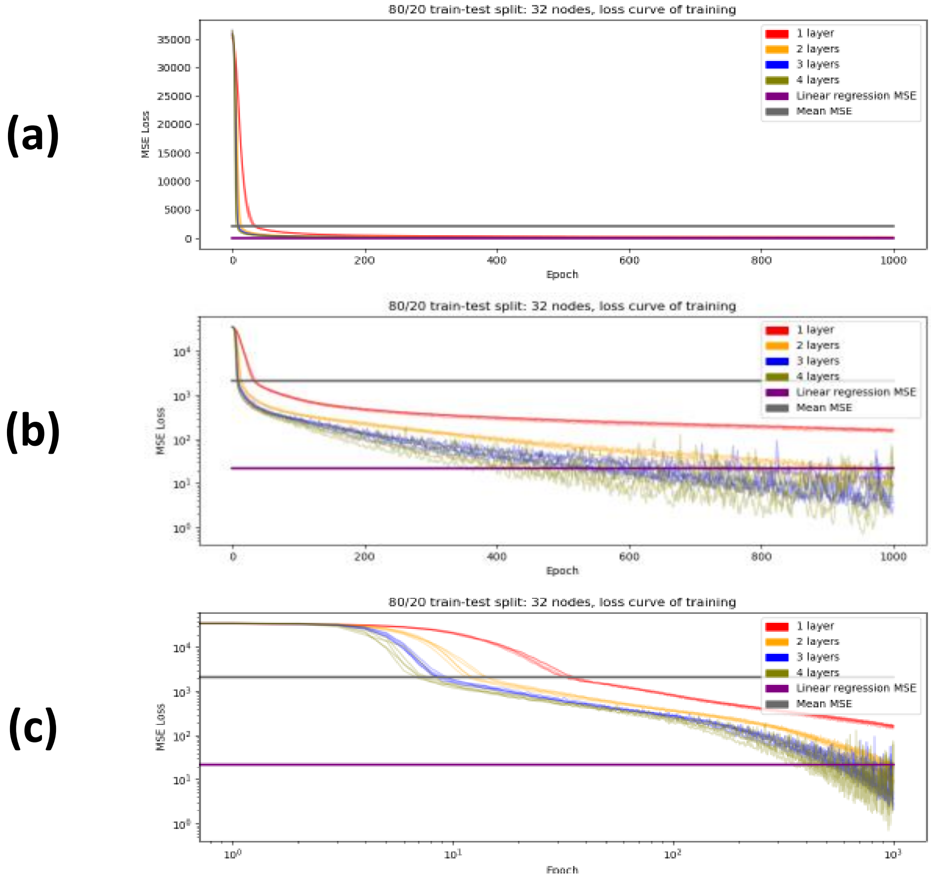


Figure 5: MSE training loss of five fits using pad granularity inputs with 32 nodes per hidden layer. (a) Epochs and loss on linear scale. (b) Epochs on linear scale and loss on logarithmic scale. (c) Epochs and loss on logarithmic scale.

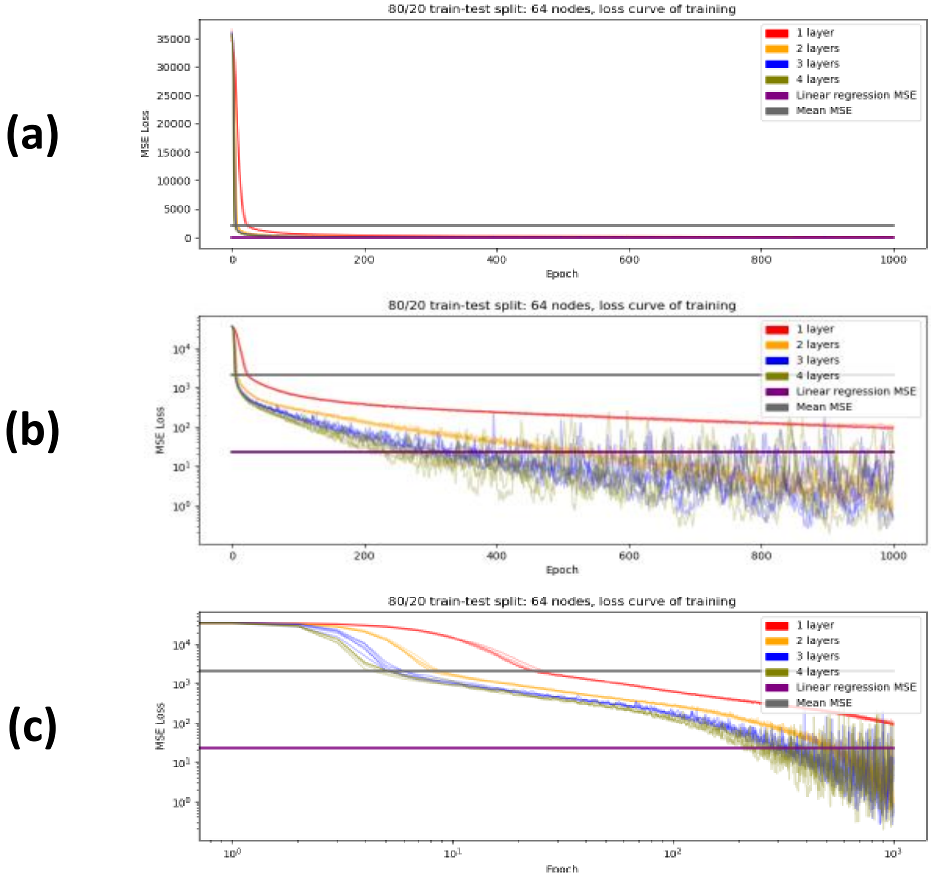


Figure 6: MSE training loss of five fits using pad granularity inputs with 64 nodes per hidden layer. (a) Epochs and loss on linear scale. (b) Epochs on linear scale and loss on logarithmic scale. (c) Epochs and loss on logarithmic scale.

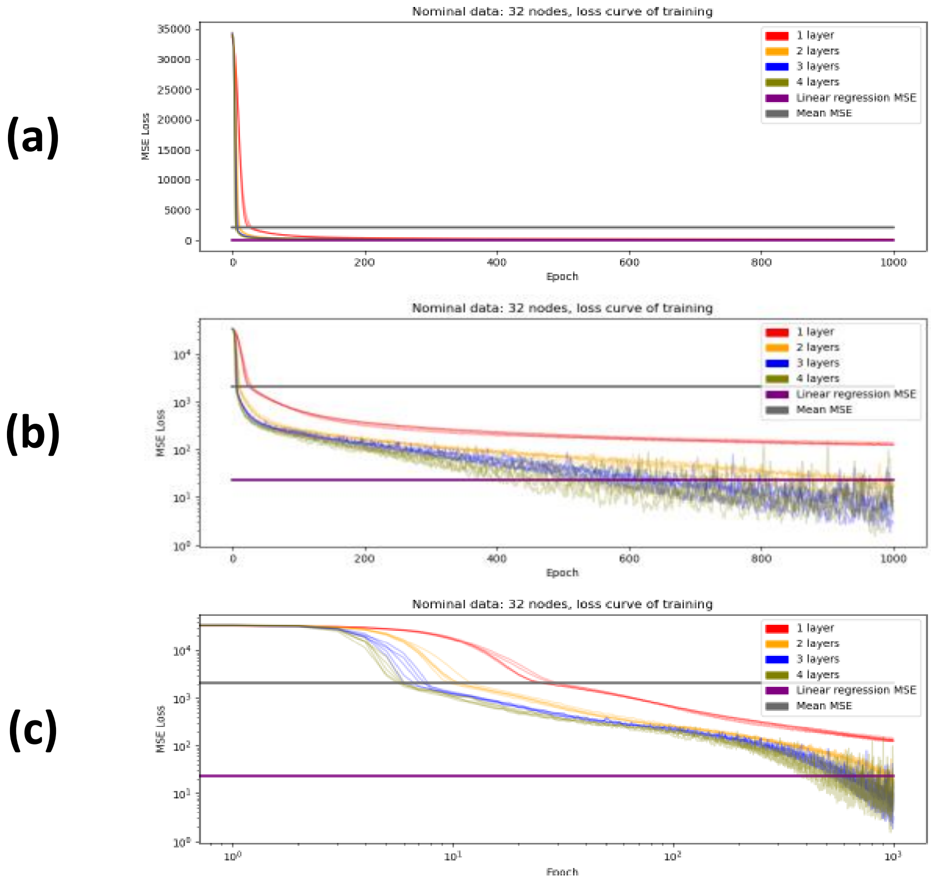


Figure 7: MSE training loss of five fits using pad granularity inputs with 32 nodes per hidden layer. (a) Epochs and loss on linear scale. (b) Epochs on linear scale and loss on logarithmic scale. (c) Epochs and loss on logarithmic scale.

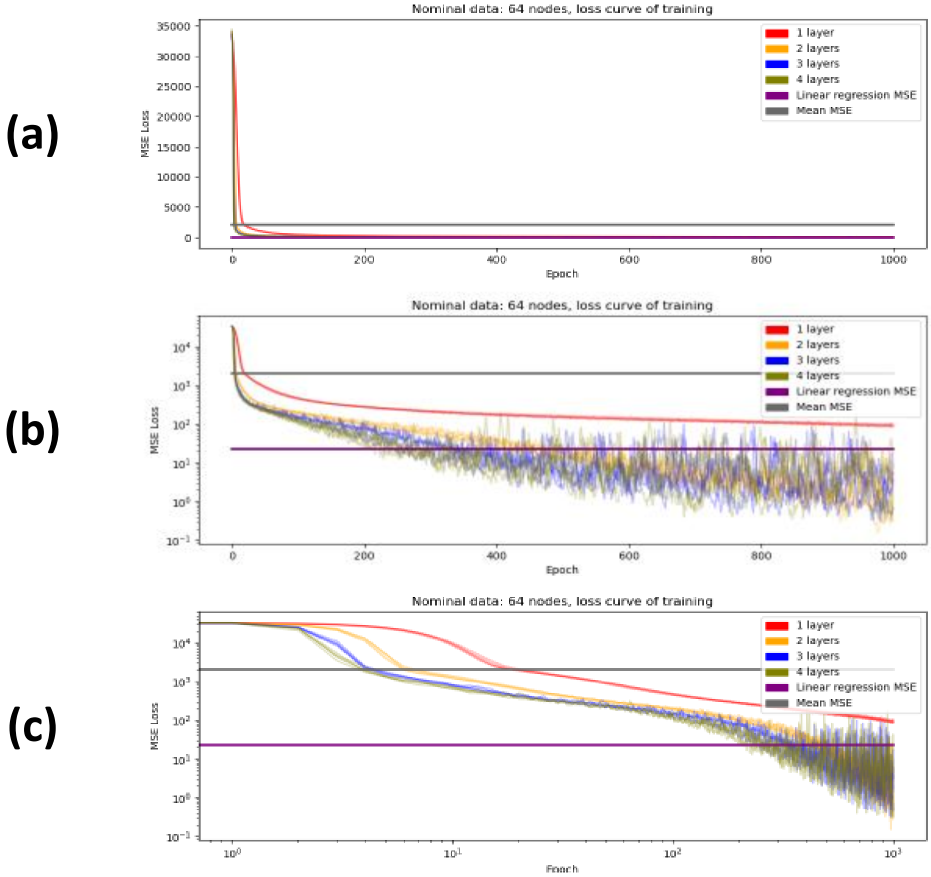


Figure 8: MSE training loss of five fits using pad granularity inputs with 64 nodes per hidden layer. (a) Epochs and loss on linear scale. (b) Epochs on linear scale and loss on logarithmic scale. (c) Epochs and loss on logarithmic scale.

III. On all data

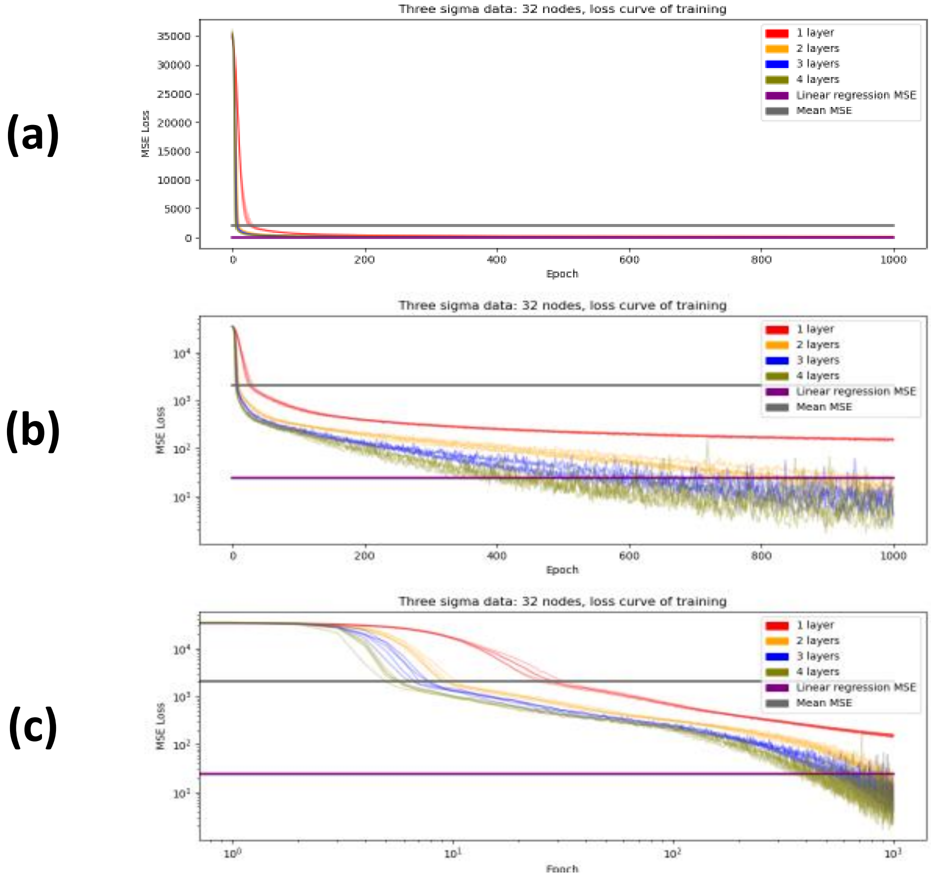


Figure 9: MSE training loss of five fits using pad granularity inputs with 32 nodes per hidden layer. (a) Epochs and loss on linear scale. (b) Epochs on linear scale and loss on logarithmic scale. (c) Epochs and loss on logarithmic scale.

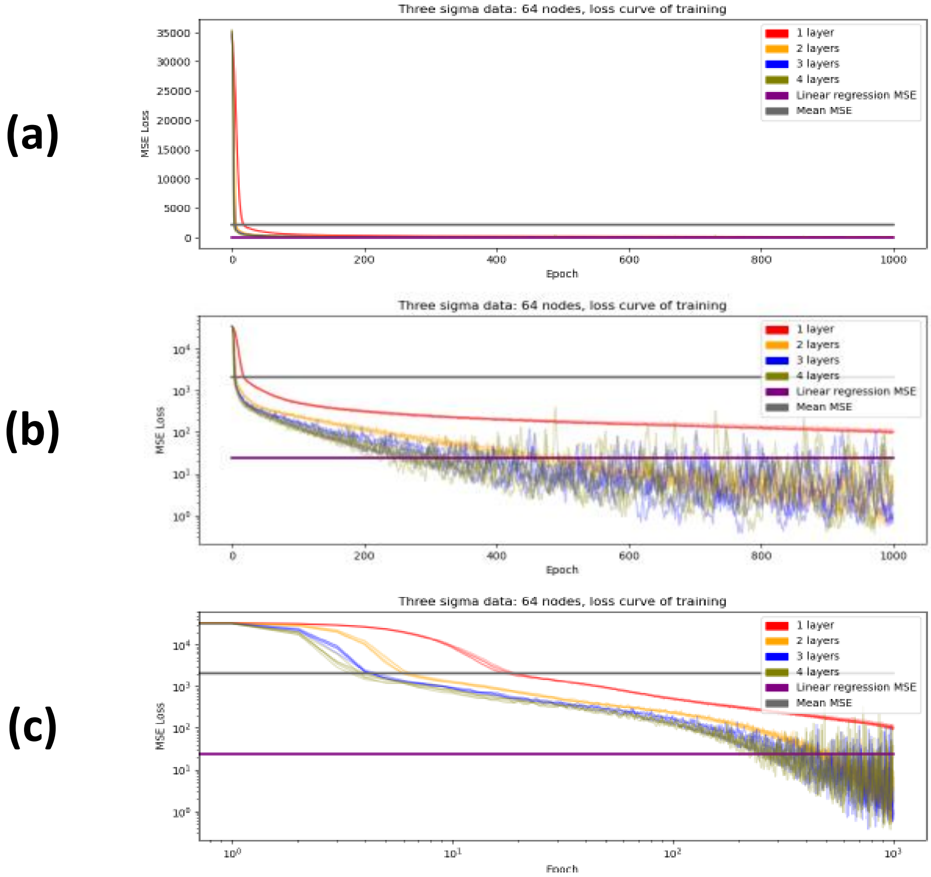


Figure 10: MSE training loss of five fits using pad granularity inputs with 64 nodes per hidden layer. (a) Epochs and loss on linear scale. (b) Epochs on linear scale and loss on logarithmic scale. (c) Epochs and loss on logarithmic scale.

SEM granularity

I. 80%/20% random data split

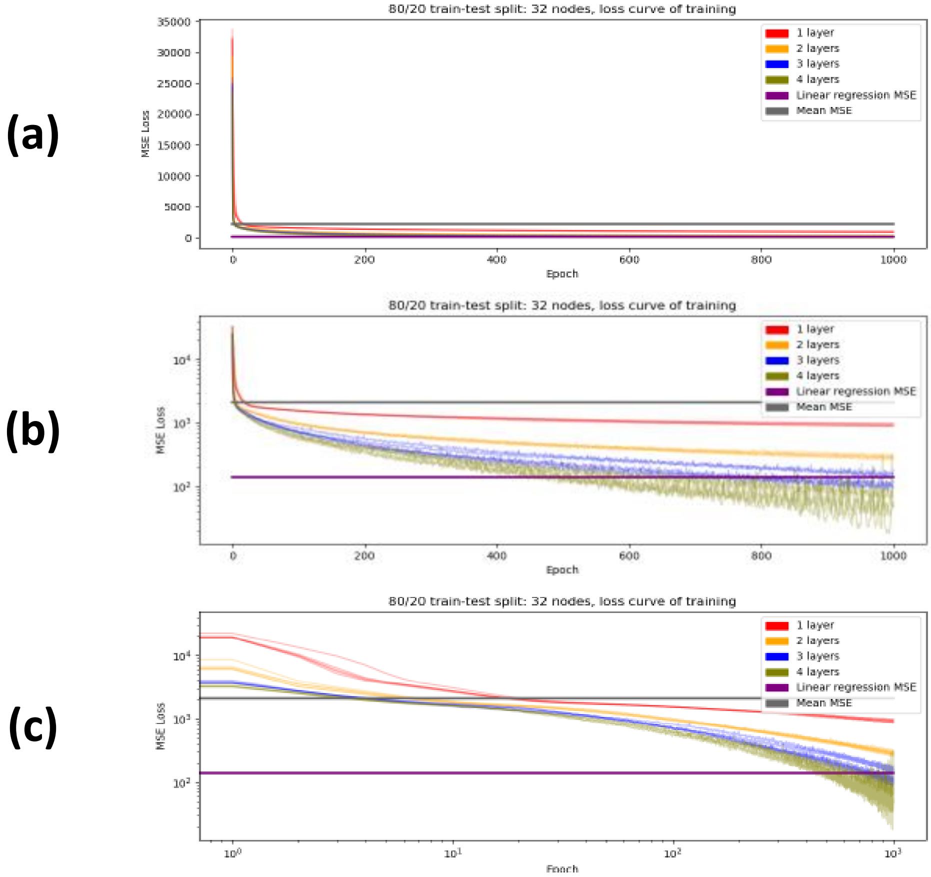


Figure 11: MSE training loss of five fits using SEM granularity inputs with 32 nodes per hidden layer. (a) Epochs and loss on linear scale. (b) Epochs on linear scale and loss on logarithmic scale. (c) Epochs and loss on logarithmic scale.

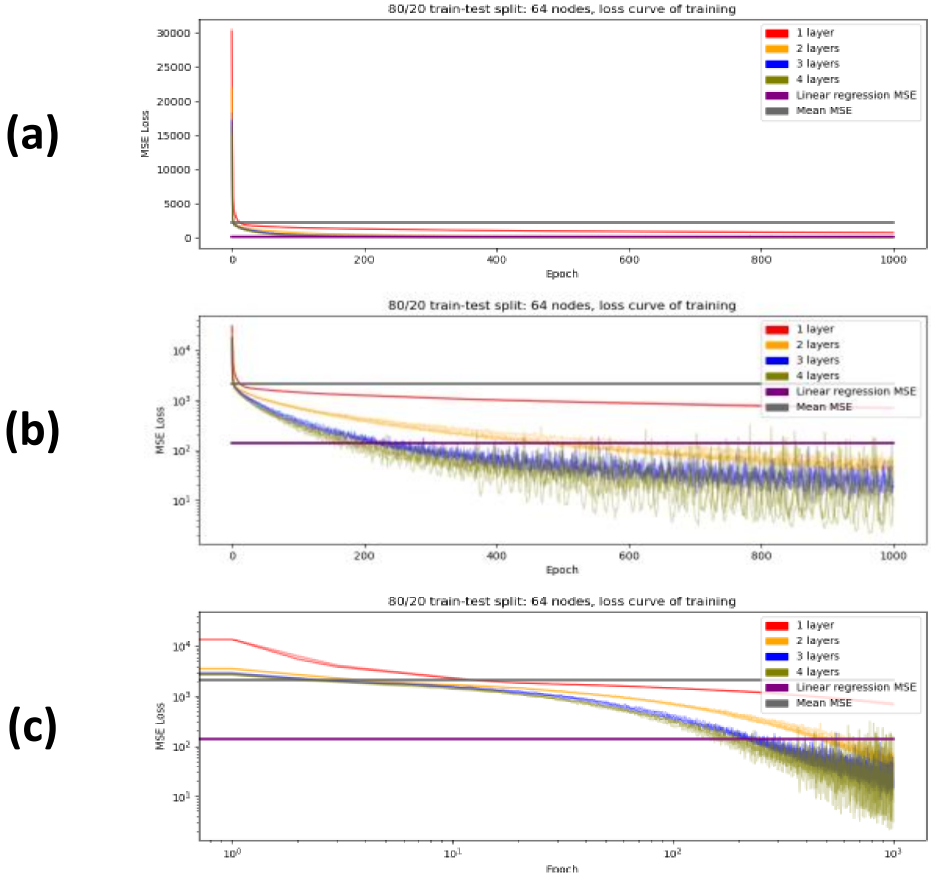


Figure 12: MSE training loss of five fits using SEM granularity inputs with 64 nodes per hidden layer. (a) Epochs and loss on linear scale. (b) Epochs on linear scale and loss on logarithmic scale. (c) Epochs and loss on logarithmic scale.

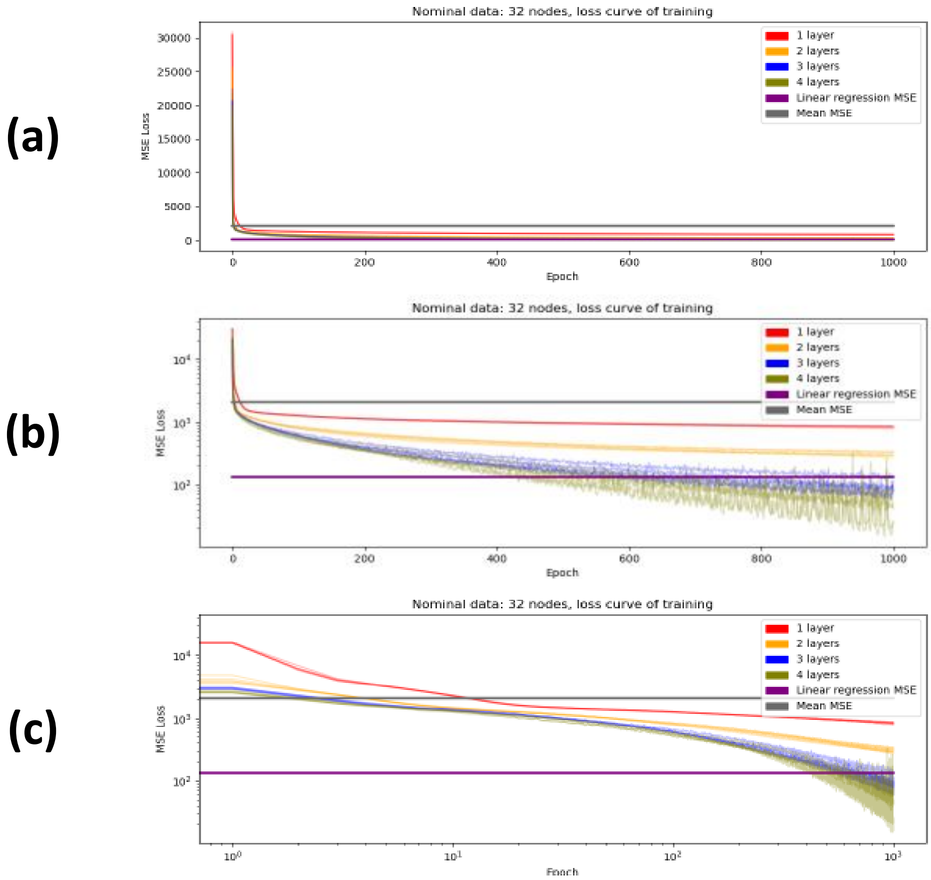


Figure 13: MSE training loss of five fits using SEM granularity inputs with 32 nodes per hidden layer. (a) Epochs and loss on linear scale. (b) Epochs on linear scale and loss on logarithmic scale. (c) Epochs and loss on logarithmic scale.

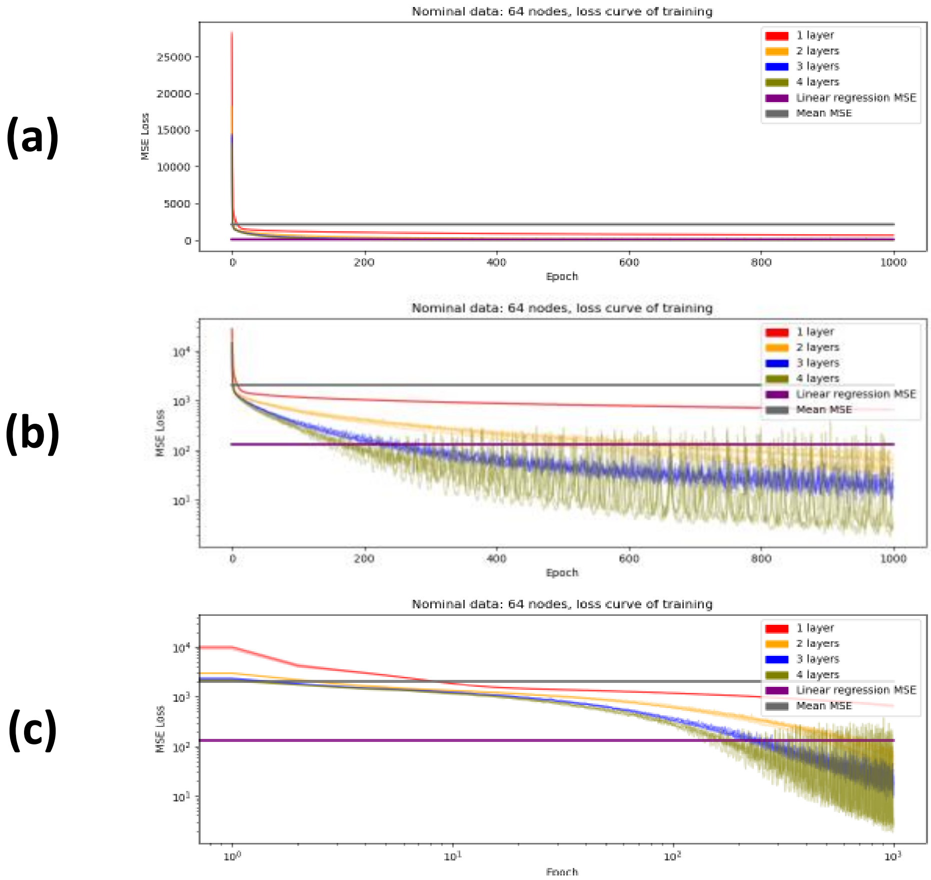


Figure 14: MSE training loss of five fits using SEM granularity inputs with 64 nodes per hidden layer. (a) Epochs and loss on linear scale. (b) Epochs on linear scale and loss on logarithmic scale. (c) Epochs and loss on logarithmic scale.

III. On all data

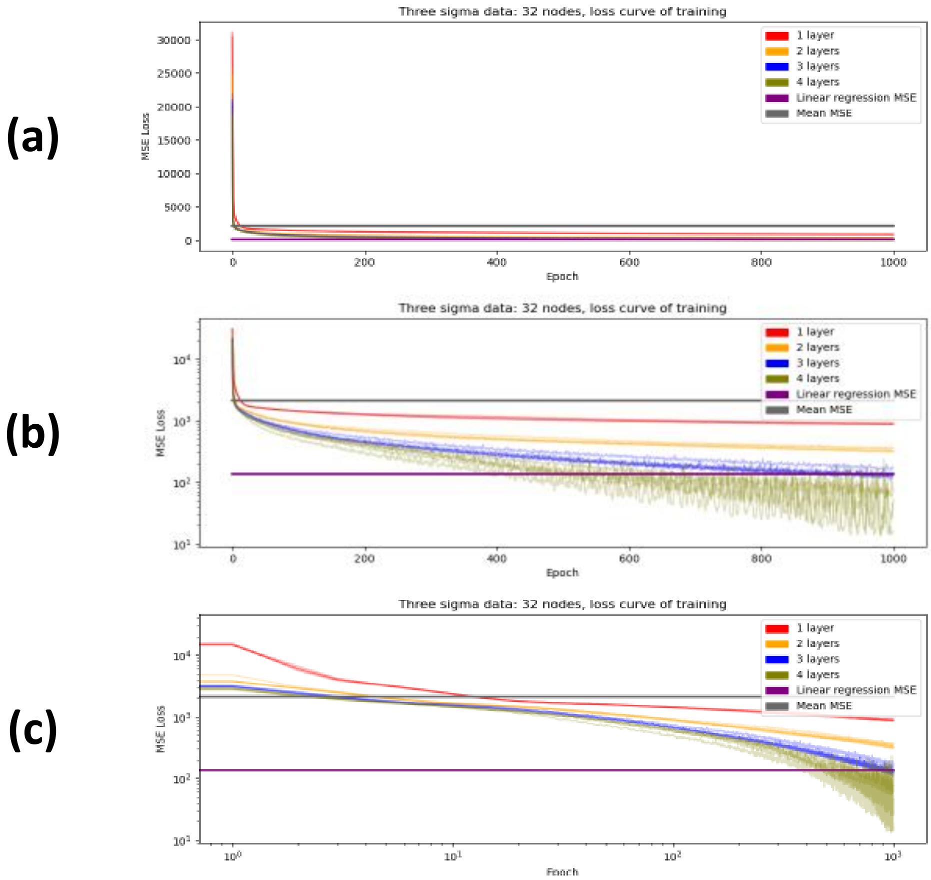


Figure 15: MSE training loss of five fits using SEM granularity inputs with 32 nodes per hidden layer. (a) Epochs and loss on linear scale. (b) Epochs on linear scale and loss on logarithmic scale. (c) Epochs and loss on logarithmic scale.

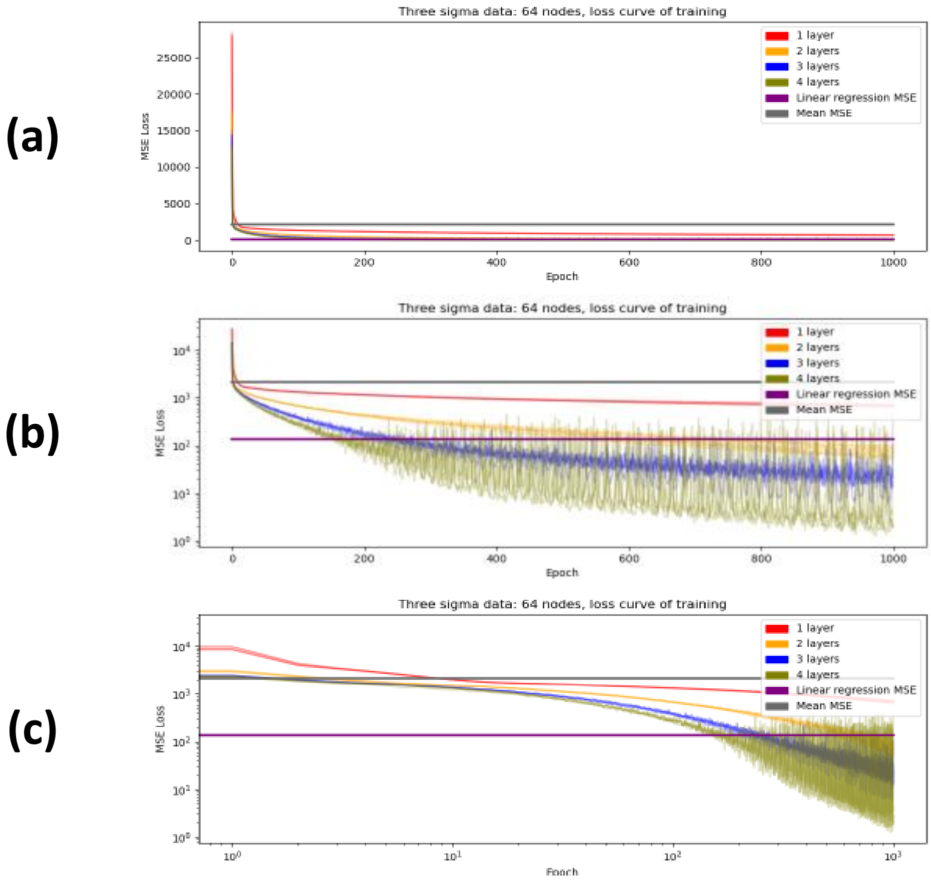


Figure 16: MSE training loss of five fits using SEM granularity inputs with 64 nodes per hidden layer. (a) Epochs and loss on linear scale. (b) Epochs on linear scale and loss on logarithmic scale. (c) Epochs and loss on logarithmic scale.

Appendix B - Experiment B additional loss curves

Field granularity

I. 80%/20% random data split

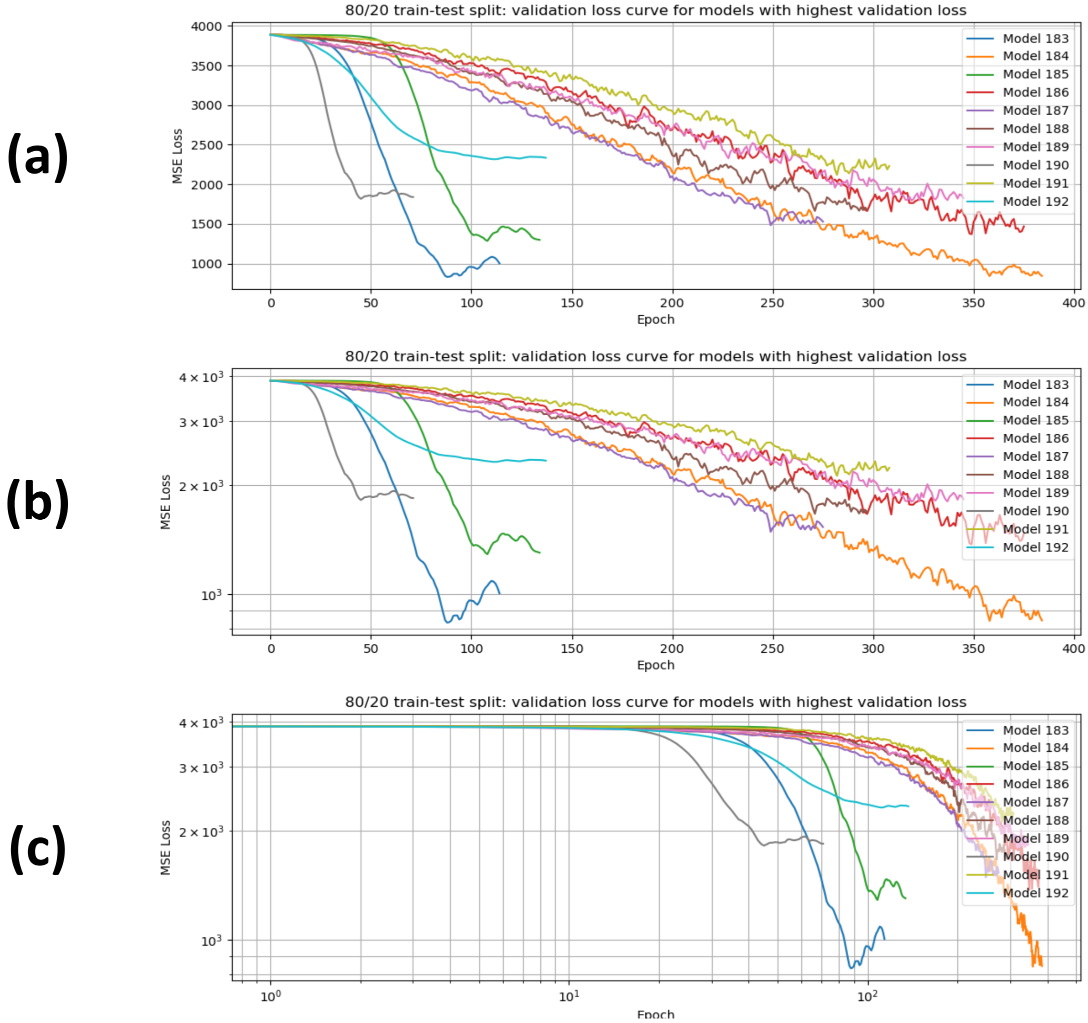


Figure 17: MSE validation loss of the 10 worst models using field granularity inputs on the 80%/20% datasplit. (a) Epochs and loss on linear scale. (b) Epochs on linear scale and loss on logarithmic scale. (c) Epochs and loss on logarithmic scale.

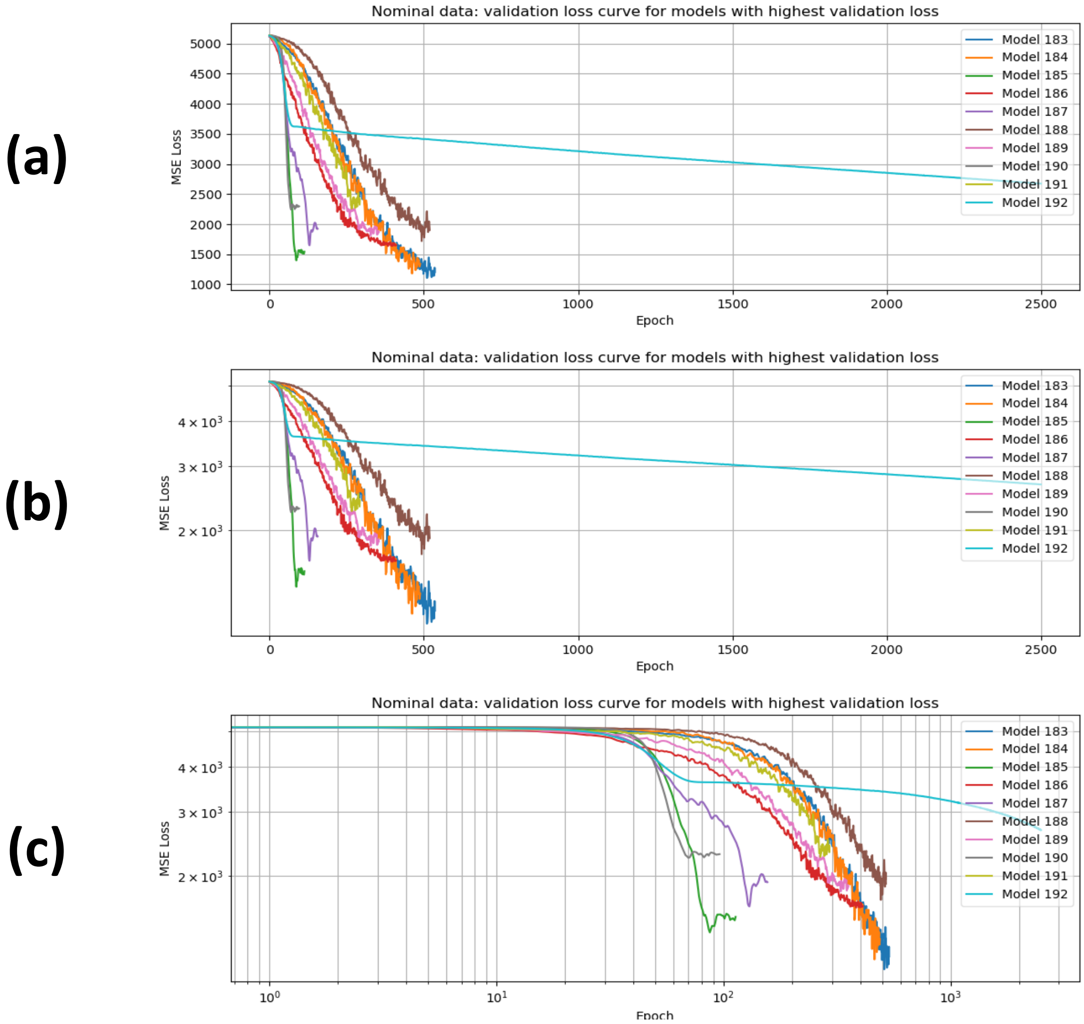


Figure 18: MSE validation loss of the 10 worst models using field granularity inputs on the nominal datasplit. (a) Epochs and loss on linear scale. (b) Epochs on linear scale and loss on logarithmic scale. (c) Epochs and loss on logarithmic scale.

III. On all data

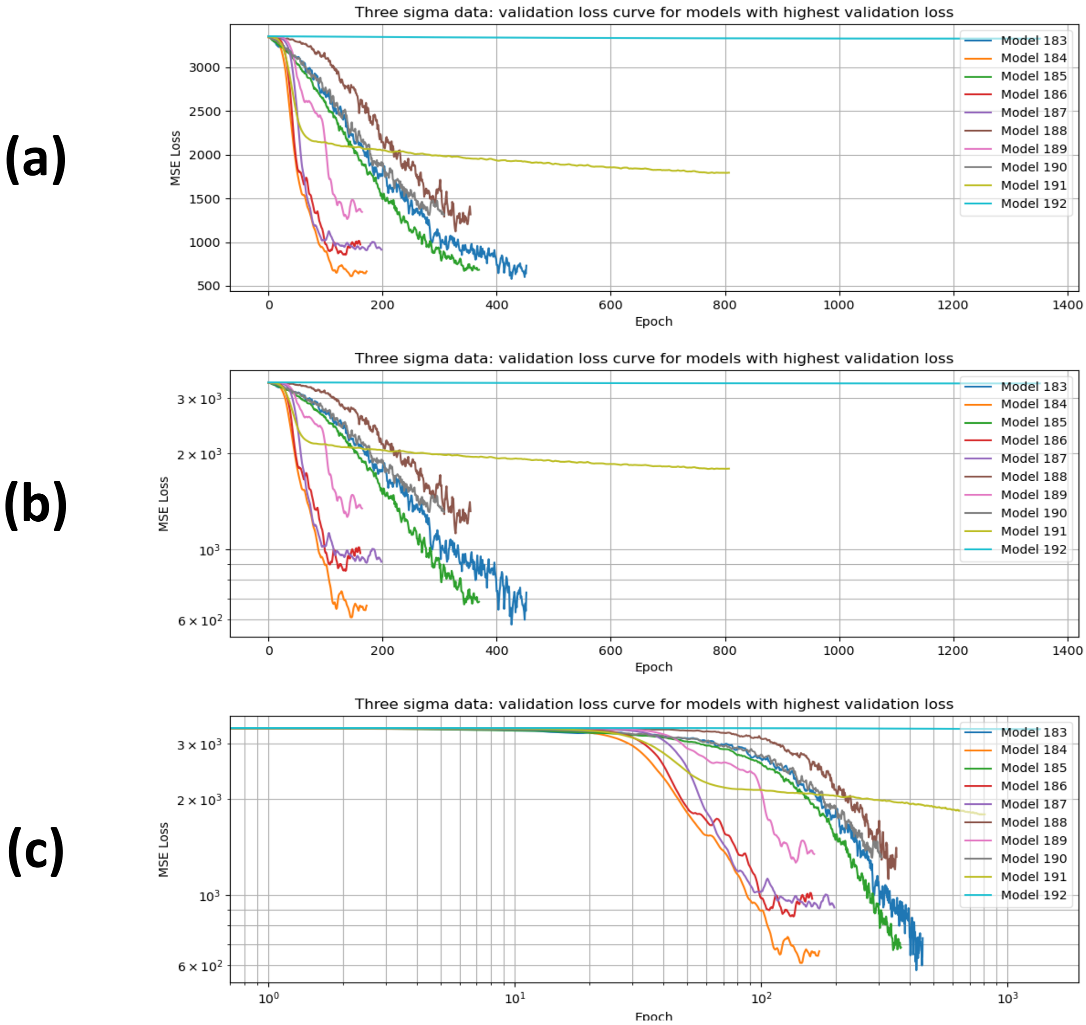


Figure 19: MSE validation loss of the 10 worst models using field granularity inputs on the 3σ datasplit. (a) Epochs and loss on linear scale. (b) Epochs on linear scale and loss on logarithmic scale. (c) Epochs and loss on logarithmic scale.

Pad granularity

I. 80%/20% random data split

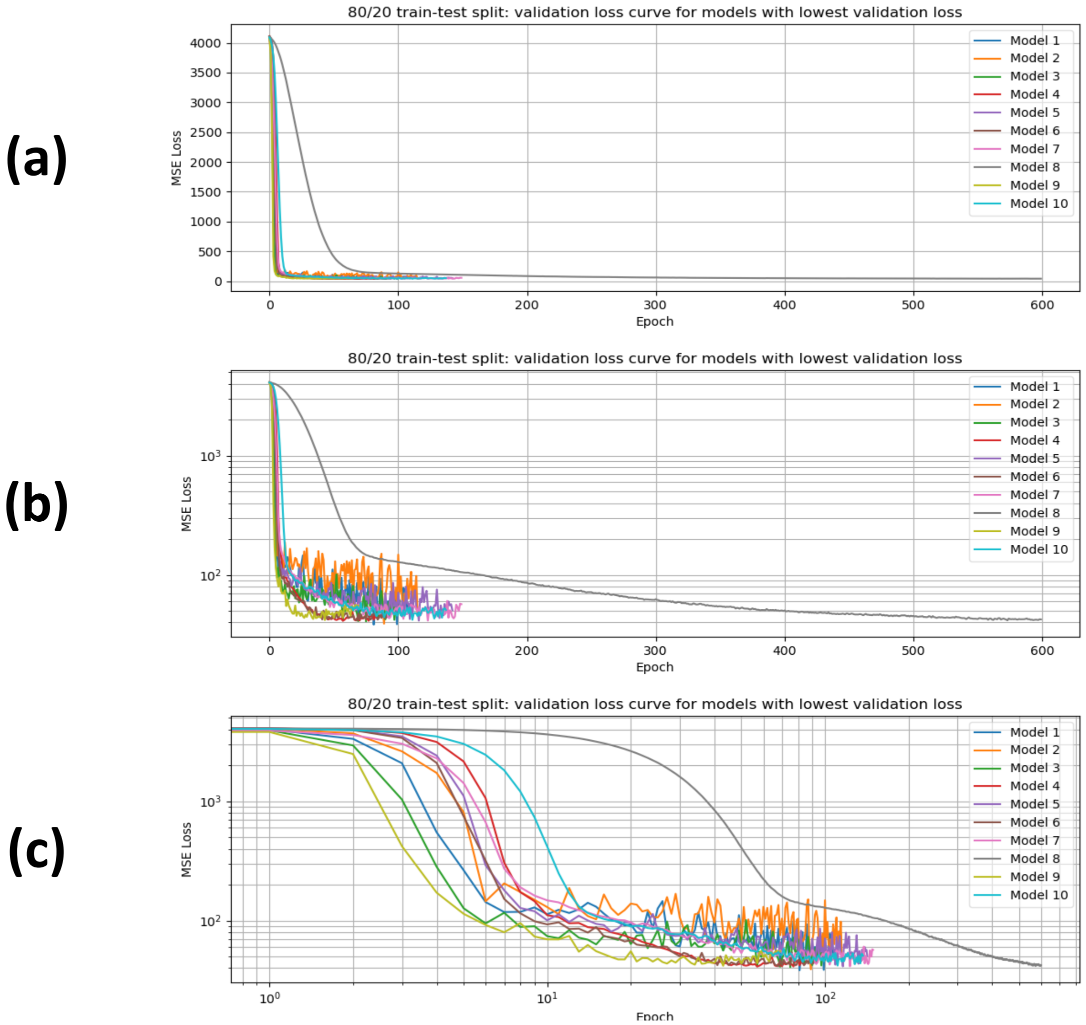


Figure 20: MSE validation loss of the 10 best models using pad granularity inputs on the 80%/20% datasplit. (a) Epochs and loss on linear scale. (b) Epochs on linear scale and loss on logarithmic scale. (c) Epochs and loss on logarithmic scale.

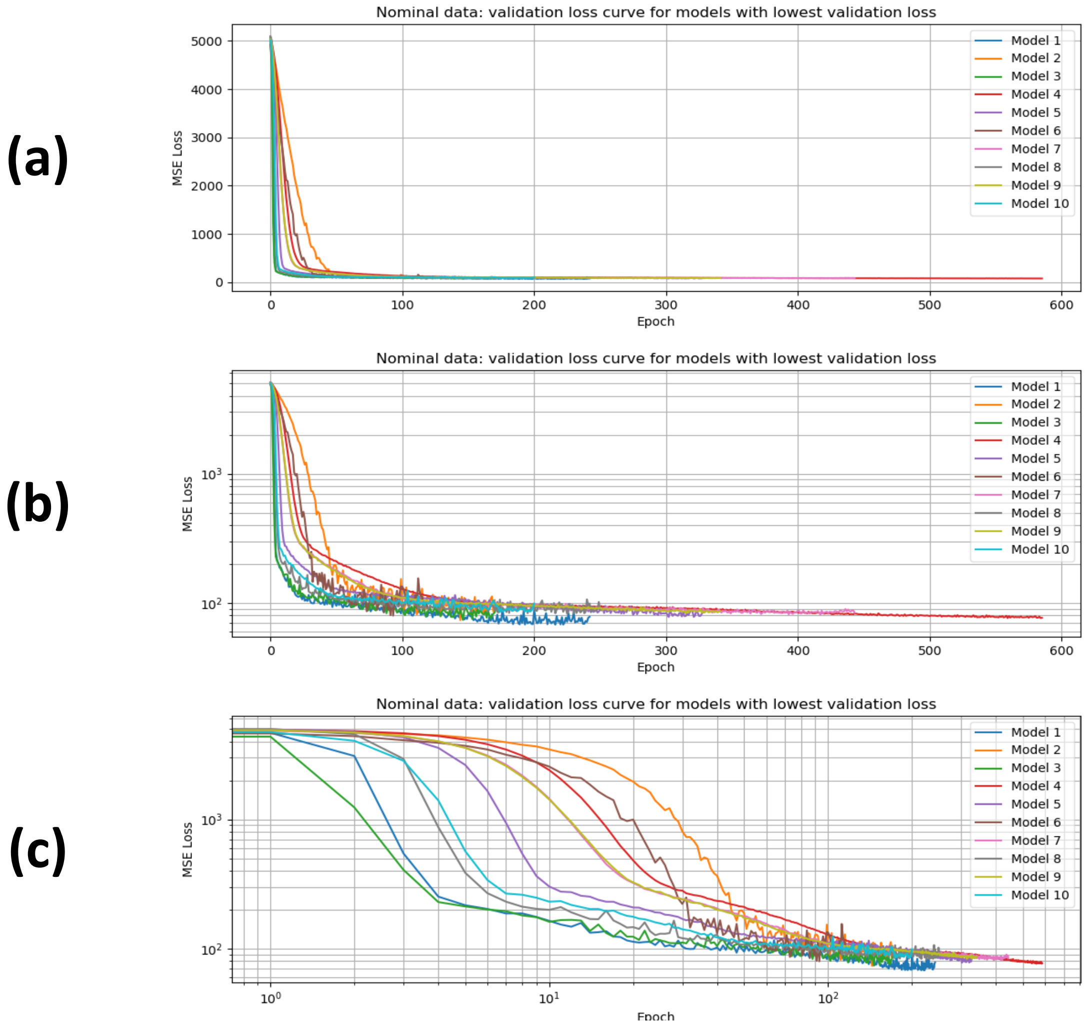


Figure 21: MSE validation loss of the 10 best models using pad granularity inputs on the nominal datasplit. (a) Epochs and loss on linear scale. (b) Epochs on linear scale and loss on logarithmic scale. (c) Epochs and loss on logarithmic scale.

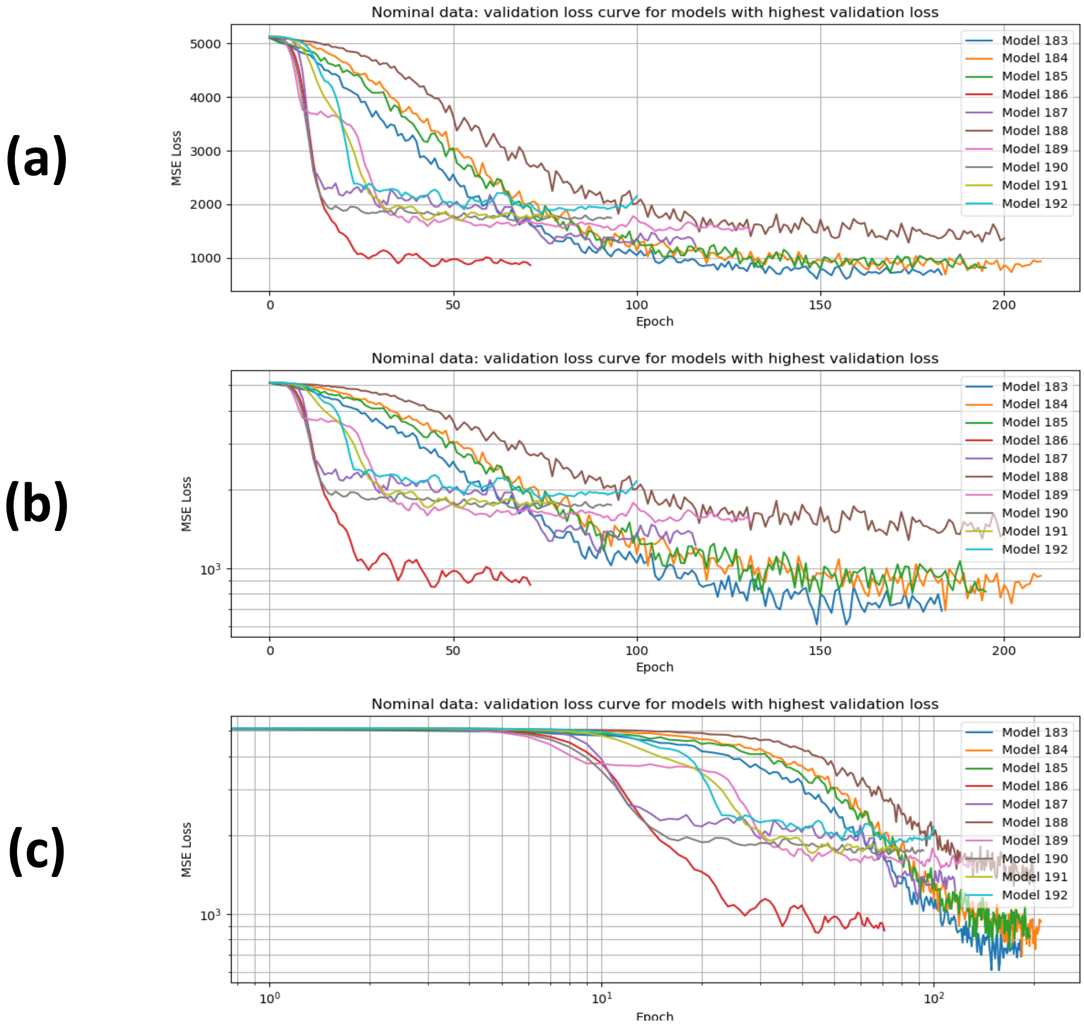


Figure 22: MSE validation loss of the 10 worst models using pad granularity inputs on the nominal datasplit. (a) Epochs and loss on linear scale. (b) Epochs on linear scale and loss on logarithmic scale. (c) Epochs and loss on logarithmic scale.

III. On all data

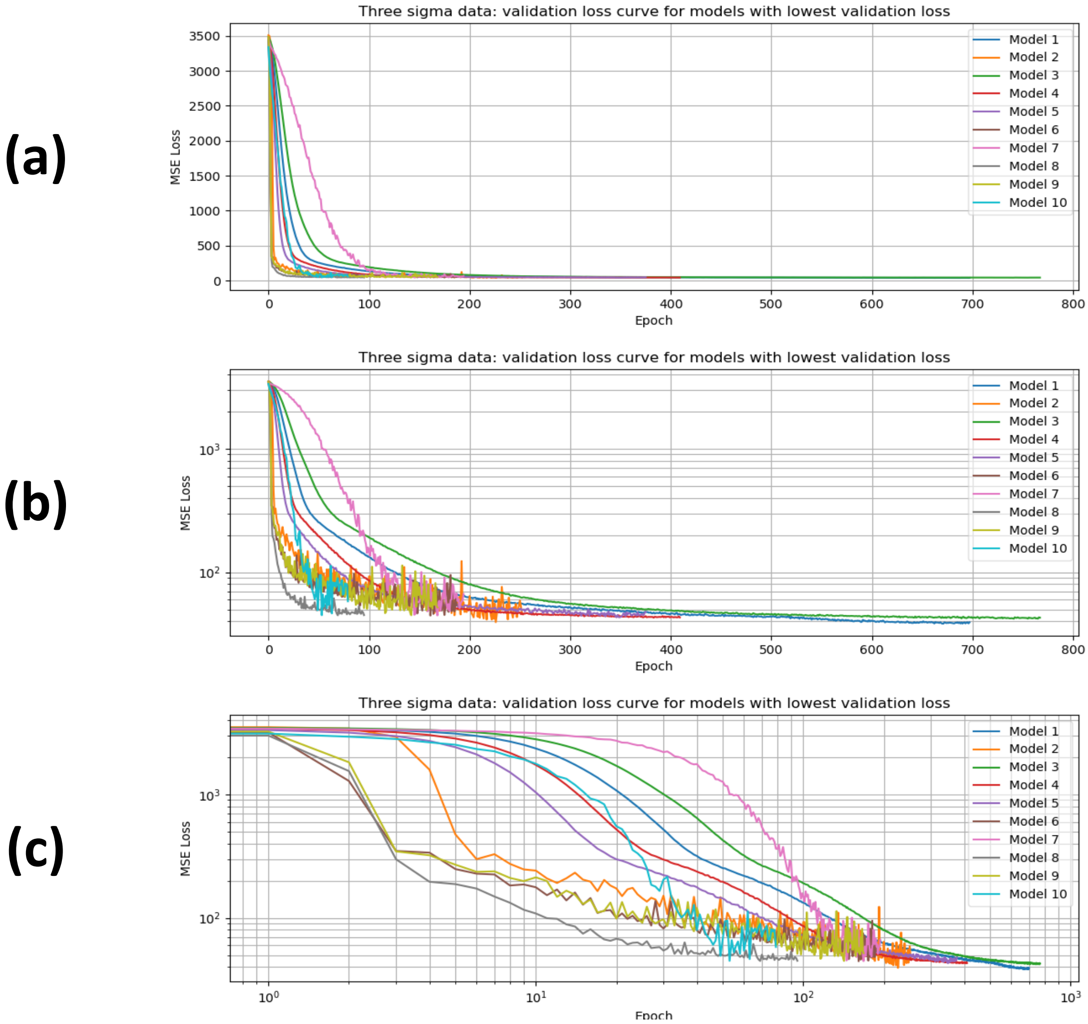


Figure 23: MSE validation loss of the 10 best models using pad granularity inputs on the 3σ datasplit. (a) Epochs and loss on linear scale. (b) Epochs on linear scale and loss on logarithmic scale. (c) Epochs and loss on logarithmic scale.

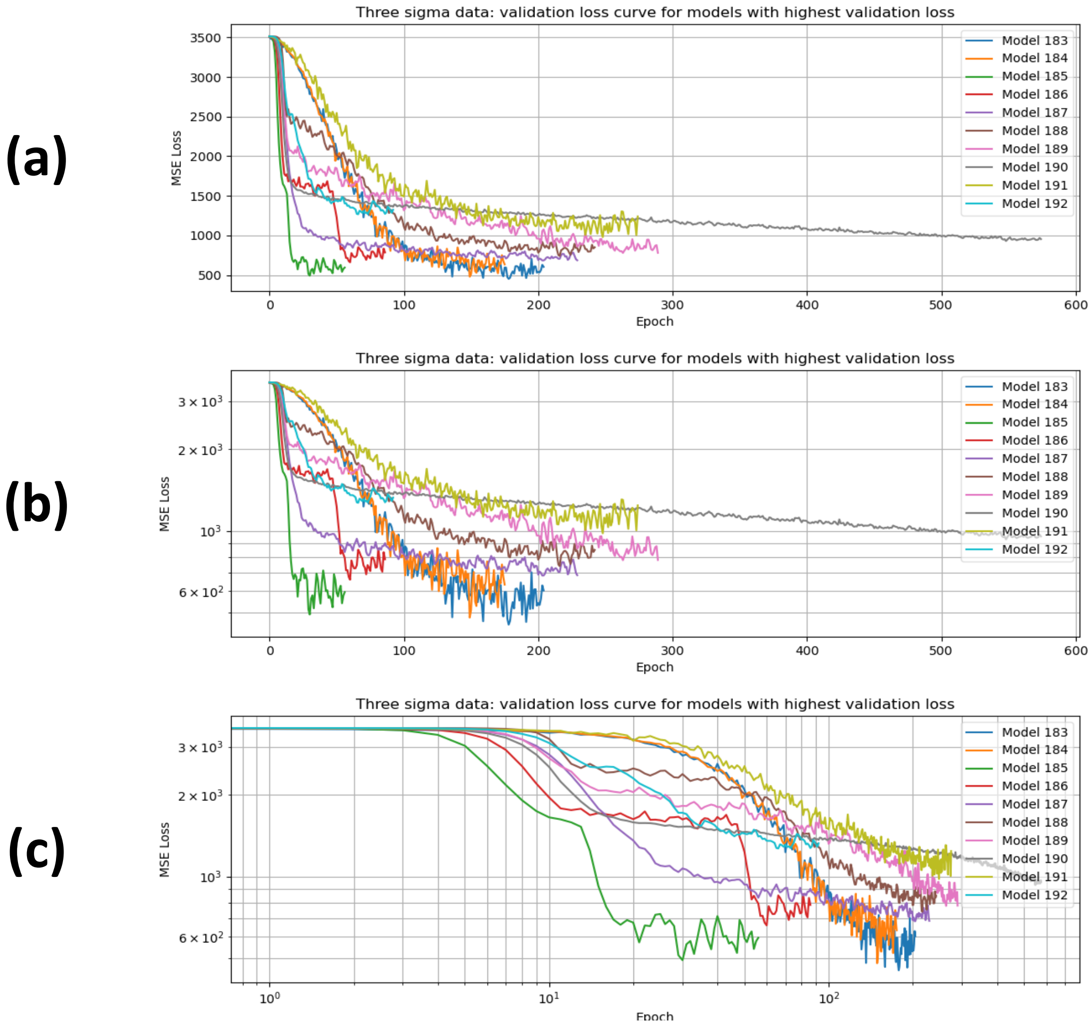


Figure 24: MSE validation loss of the 10 worst models using pad granularity inputs on the 3σ datasplit. (a) Epochs and loss on linear scale. (b) Epochs on linear scale and loss on logarithmic scale. (c) Epochs and loss on logarithmic scale.

SEM granularity

I. 80%/20% random data split

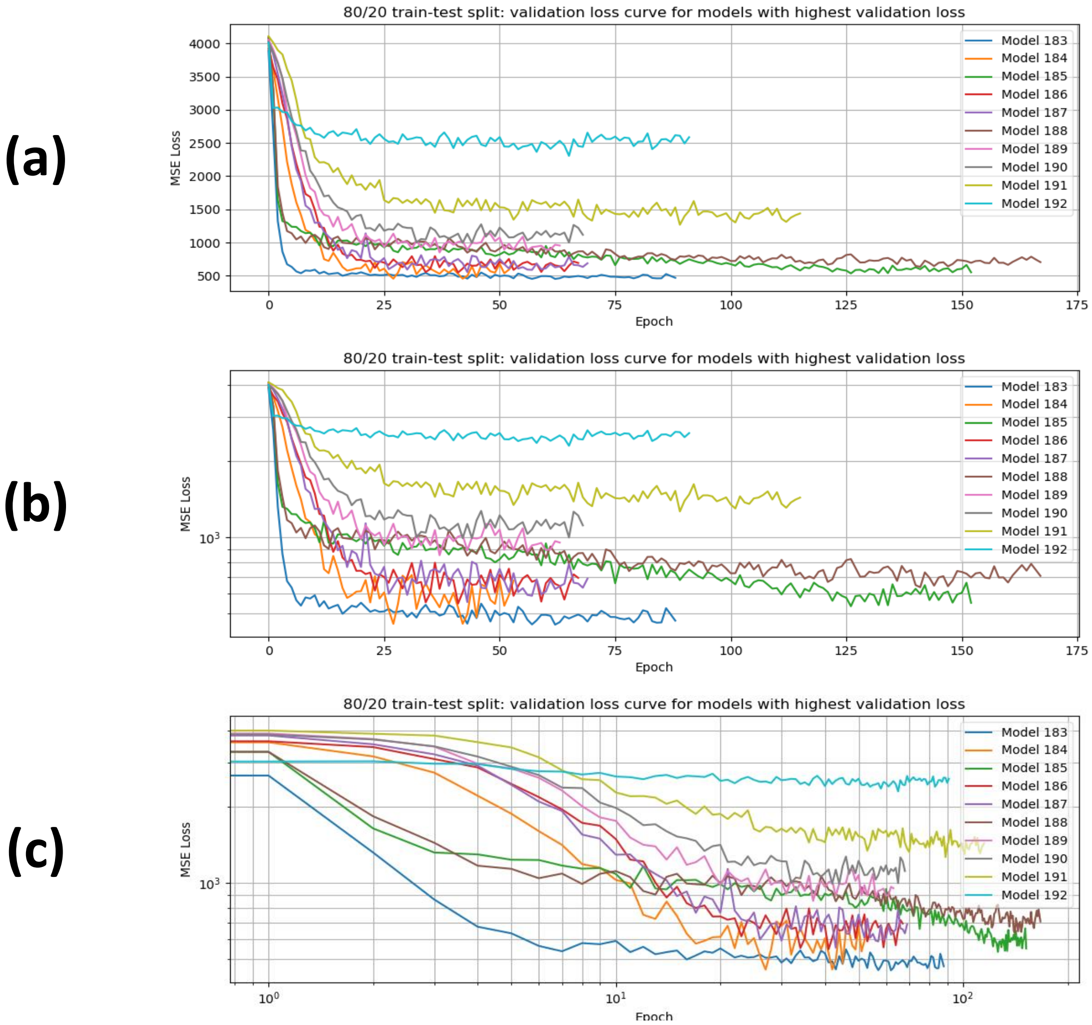


Figure 25: MSE validation loss of the 10 worst models using SEM granularity inputs on the 80%/20% datasplit. (a) Epochs and loss on linear scale. (b) Epochs on linear scale and loss on logarithmic scale. (c) Epochs and loss on logarithmic scale.

II. Nominal range

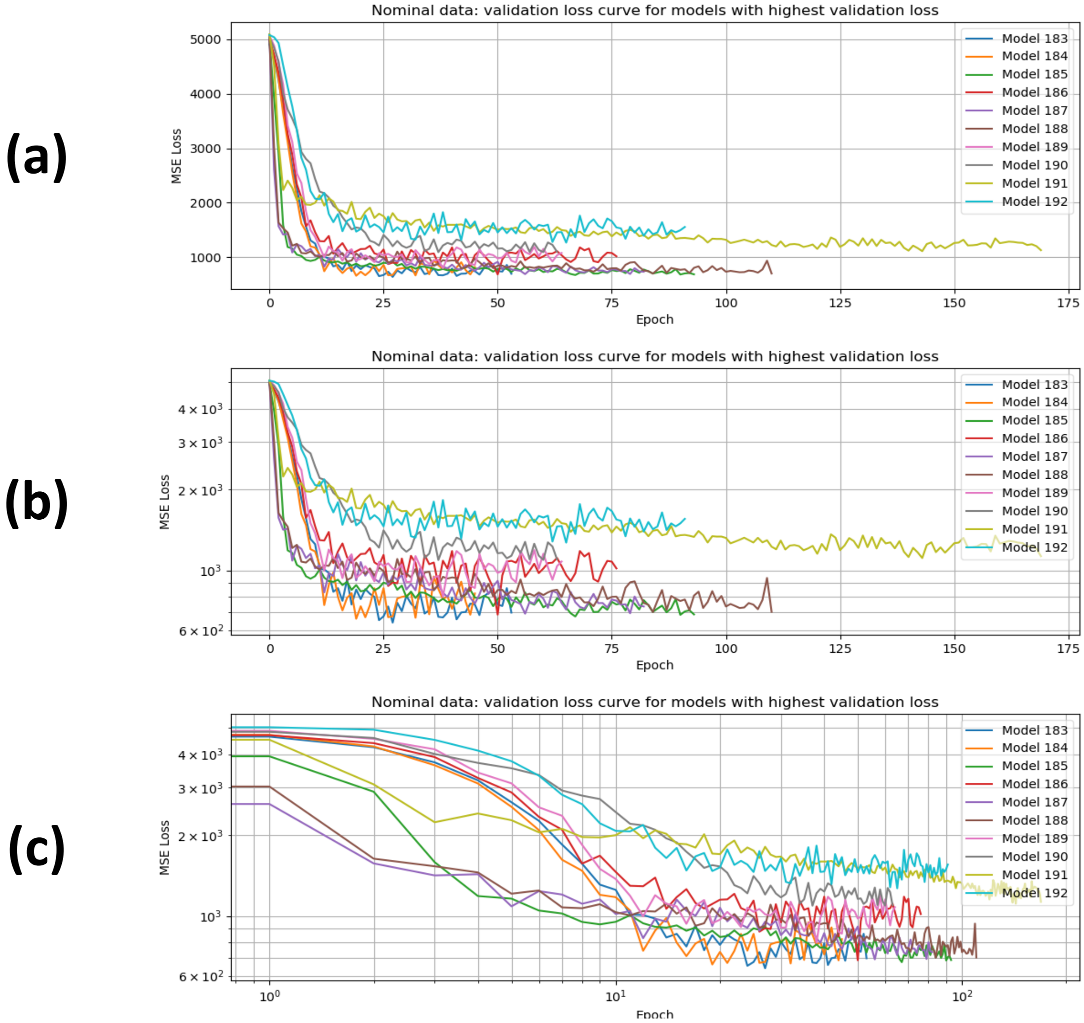


Figure 26: MSE validation loss of the 10 worst models using SEM granularity inputs on the nominal datasplit. (a) Epochs and loss on linear scale. (b) Epochs on linear scale and loss on logarithmic scale. (c) Epochs and loss on logarithmic scale.

III. On all data

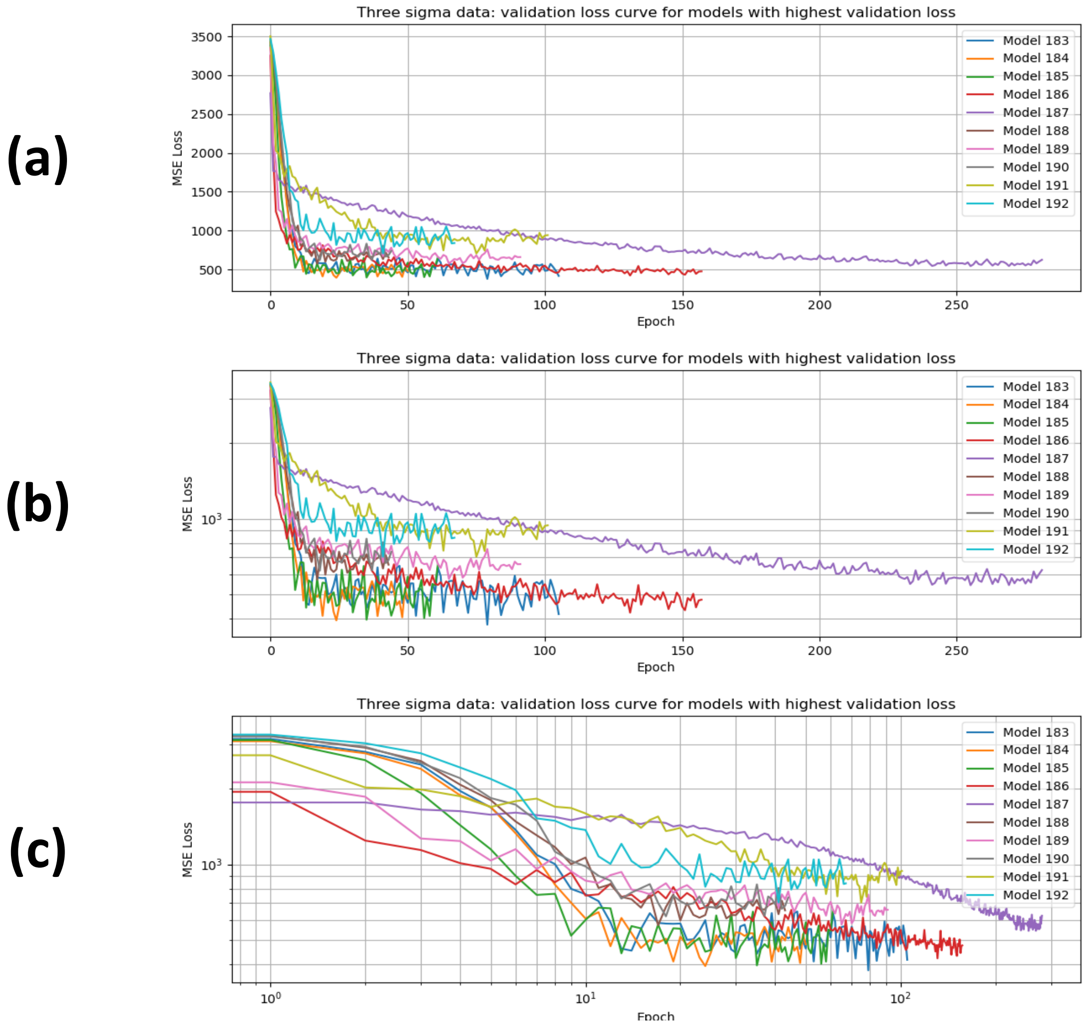


Figure 27: MSE validation loss of the 10 worst models using SEM granularity inputs on the 3σ datasplit. (a) Epochs and loss on linear scale. (b) Epochs on linear scale and loss on logarithmic scale. (c) Epochs and loss on logarithmic scale.

Appendix C - Experiment D additional loss curves

Field granularity

I. 80%/20% random data split

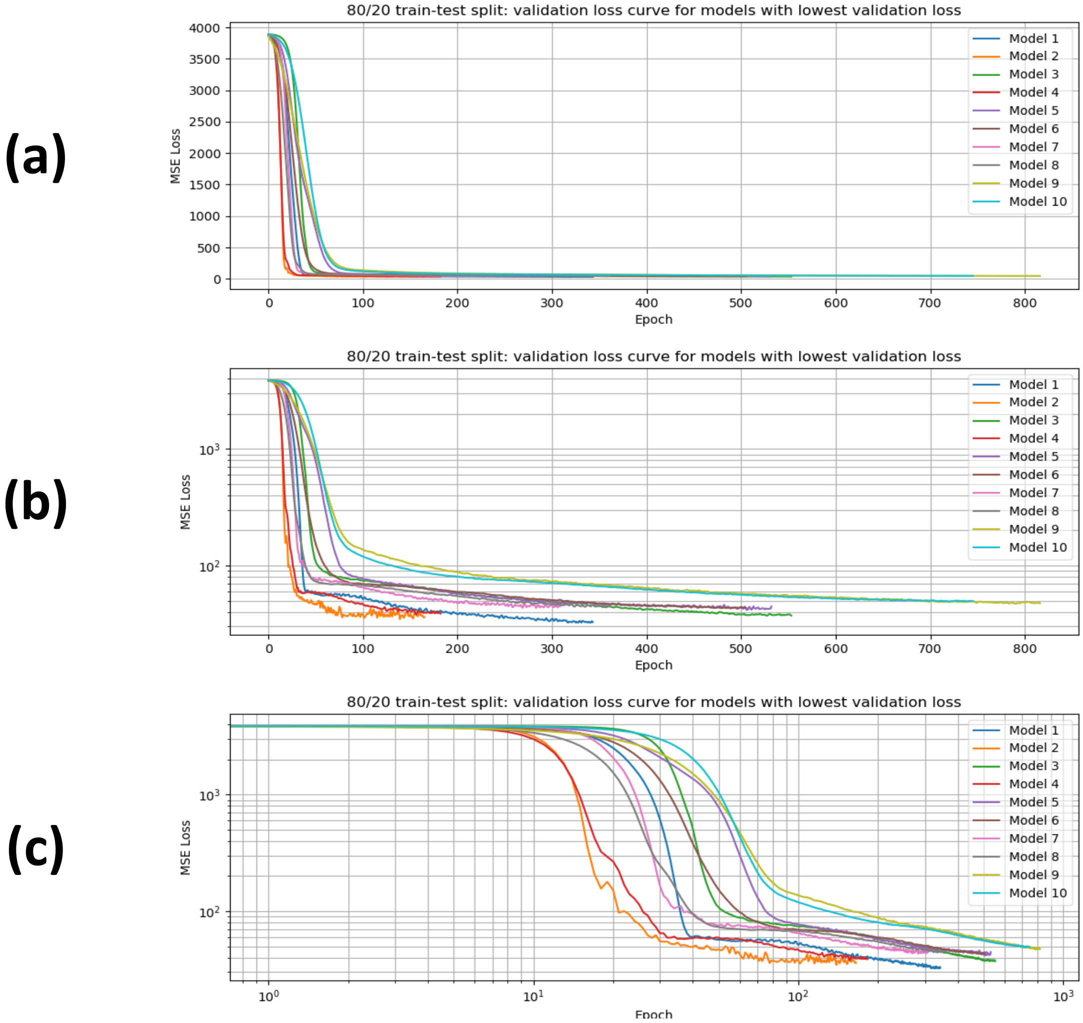


Figure 28: MSE validation loss of the 10 best models using field granularity inputs on the 80%/20% datasplit. (a) Epochs and loss on linear scale. (b) Epochs on linear scale and loss on logarithmic scale. (c) Epochs and loss on logarithmic scale.

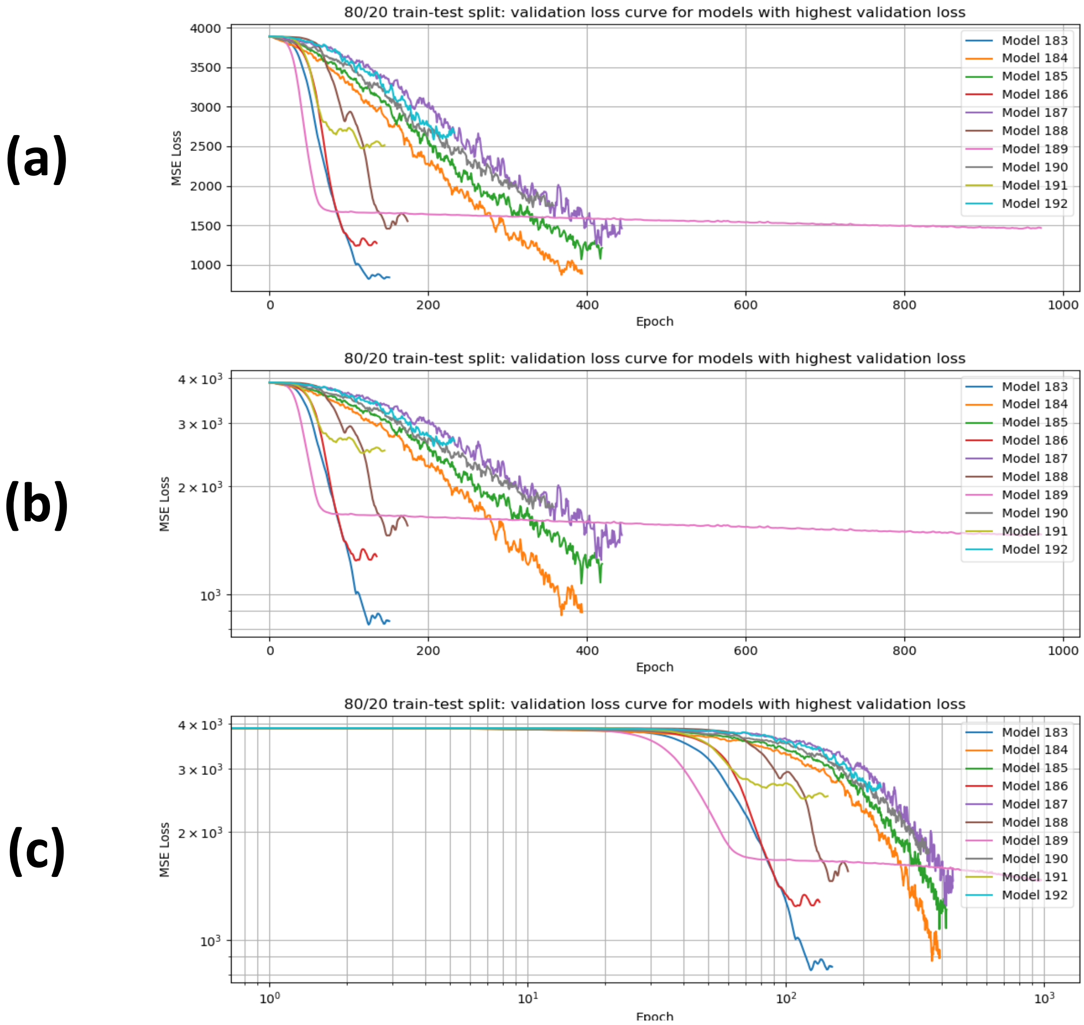


Figure 29: MSE validation loss of the 10 worst models using field granularity inputs on the 80%/20% datasplit. (a) Epochs and loss on linear scale. (b) Epochs on linear scale and loss on logarithmic scale. (c) Epochs and loss on logarithmic scale.

II. Nominal range

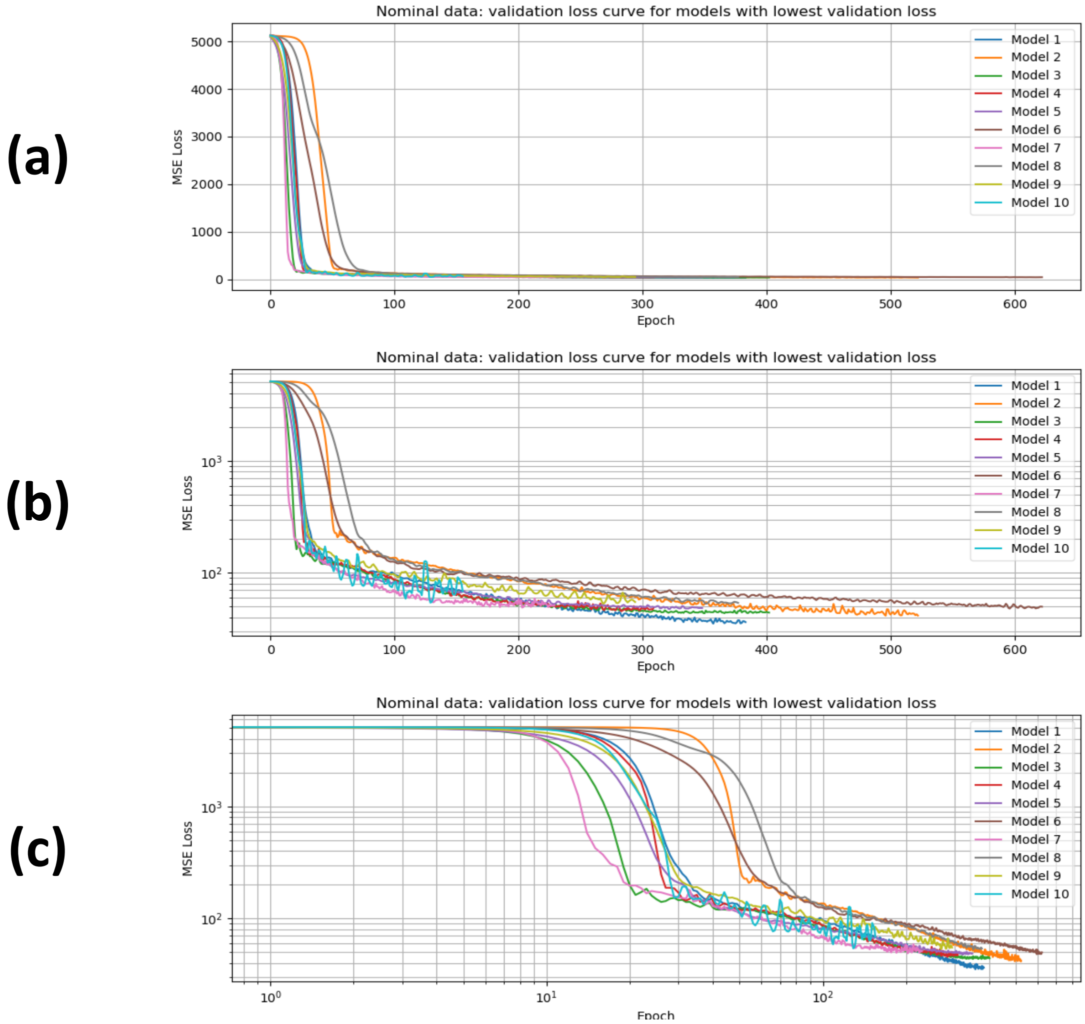


Figure 30: MSE validation loss of the 10 best models using field granularity inputs on the nominal datasplit. (a) Epochs and loss on linear scale. (b) Epochs on linear scale and loss on logarithmic scale. (c) Epochs and loss on logarithmic scale.

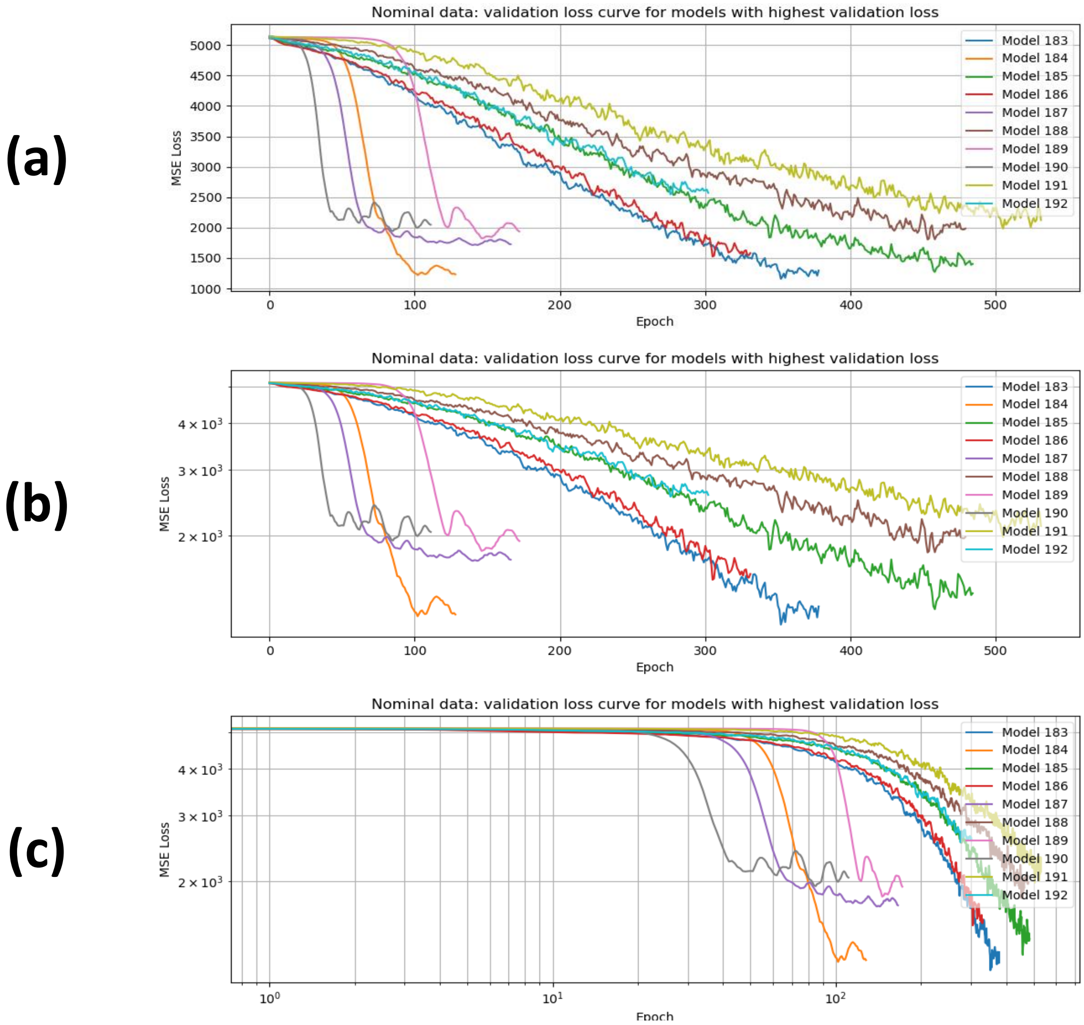


Figure 31: MSE validation loss of the 10 worst models using field granularity inputs on the nominal datasplit. (a) Epochs and loss on linear scale. (b) Epochs on linear scale and loss on logarithmic scale. (c) Epochs and loss on logarithmic scale.

III. On all data

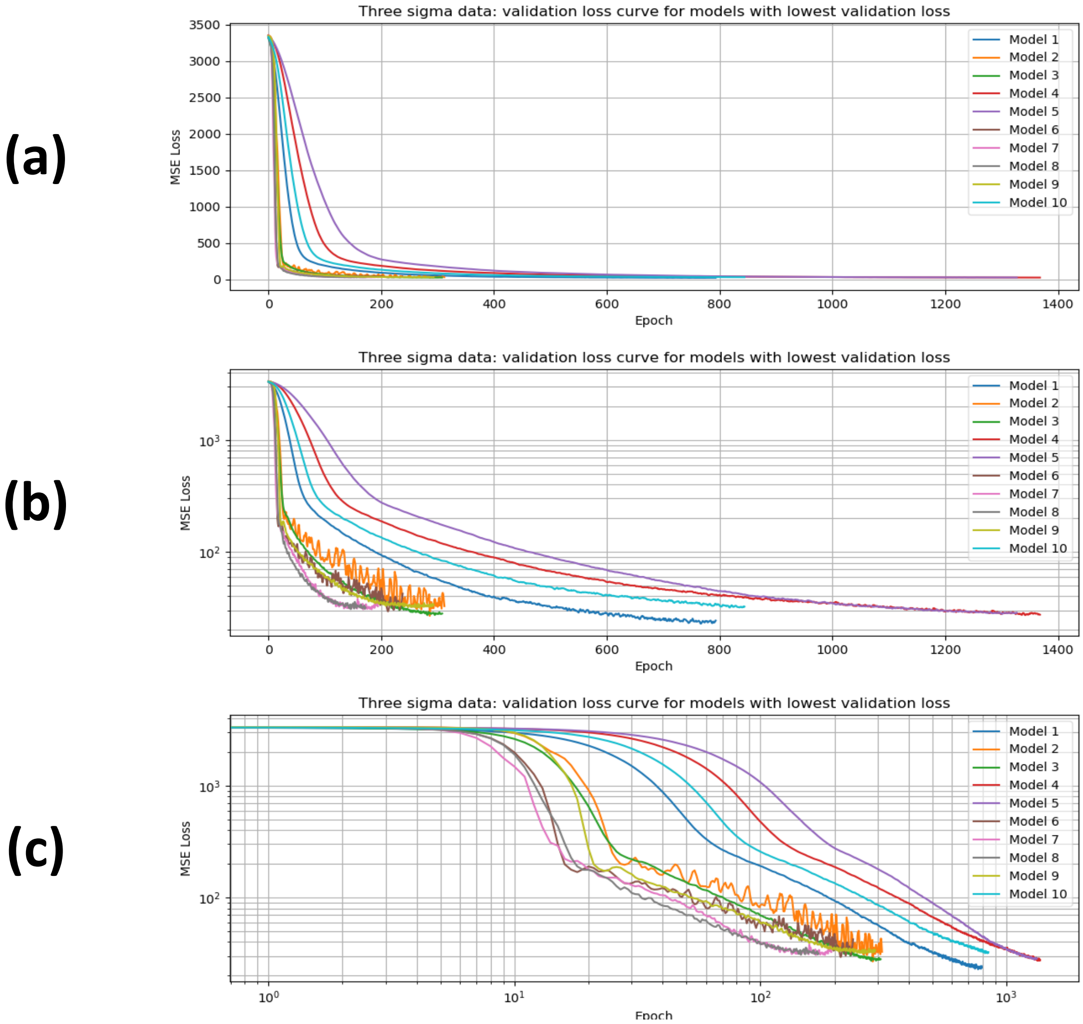


Figure 32: MSE validation loss of the 10 best models using field granularity inputs on the 3σ datasplit. (a) Epochs and loss on linear scale. (b) Epochs on linear scale and loss on logarithmic scale. (c) Epochs and loss on logarithmic scale.

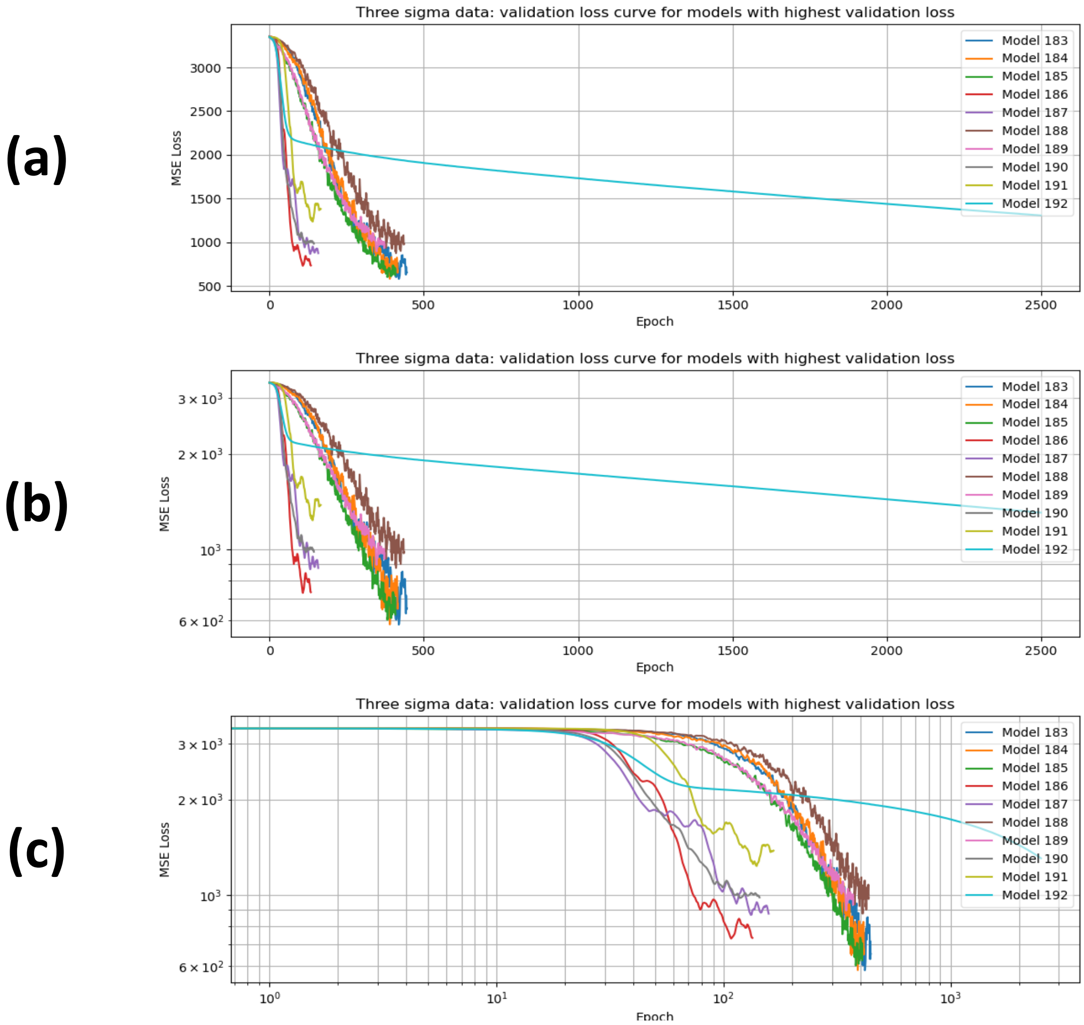


Figure 33: MSE validation loss of the 10 worst models using field granularity inputs on the 3σ datasplit. (a) Epochs and loss on linear scale. (b) Epochs on linear scale and loss on logarithmic scale. (c) Epochs and loss on logarithmic scale.

Pad granularity

I. 80%/20% random data split

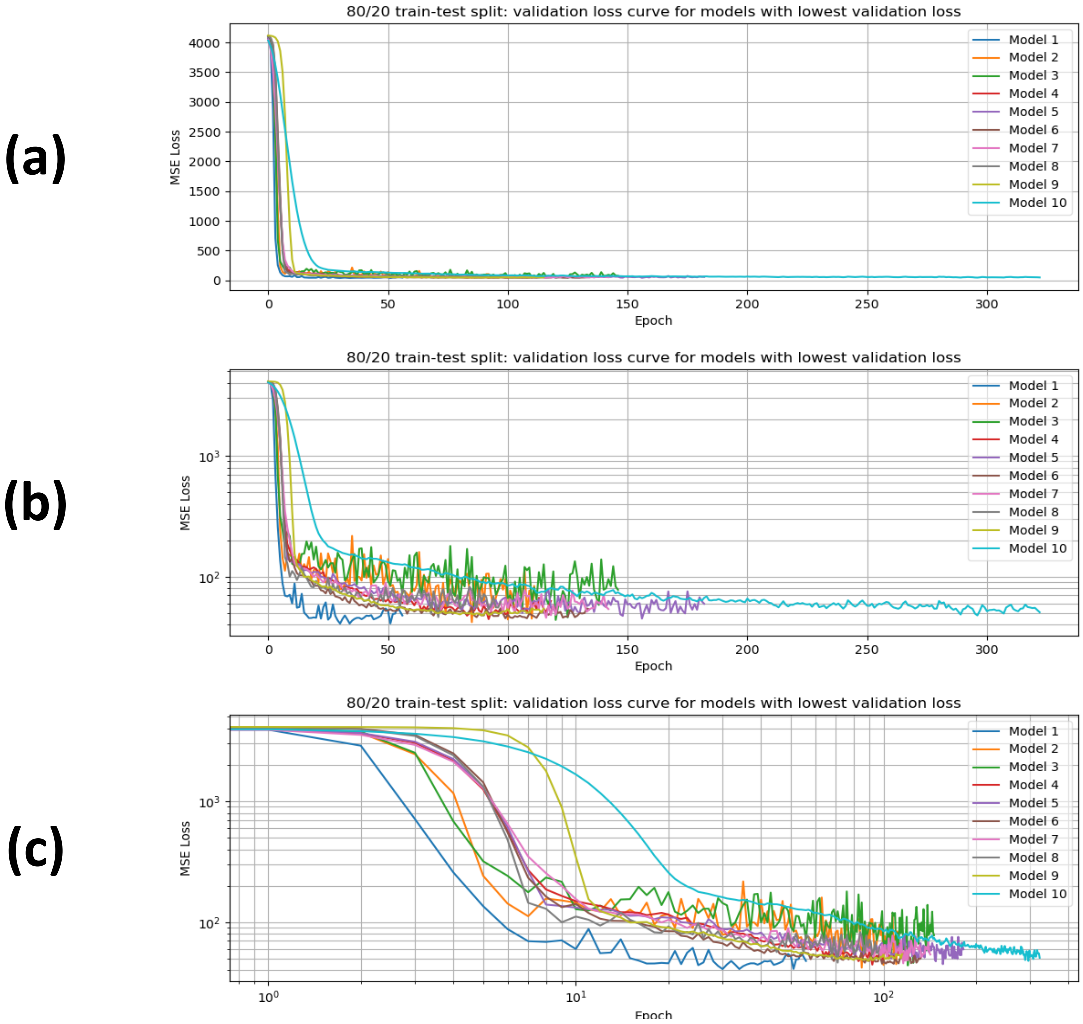


Figure 34: MSE validation loss of the 10 best models using pad granularity inputs on the 80%/20% datasplit. (a) Epochs and loss on linear scale. (b) Epochs on linear scale and loss on logarithmic scale. (c) Epochs and loss on logarithmic scale.

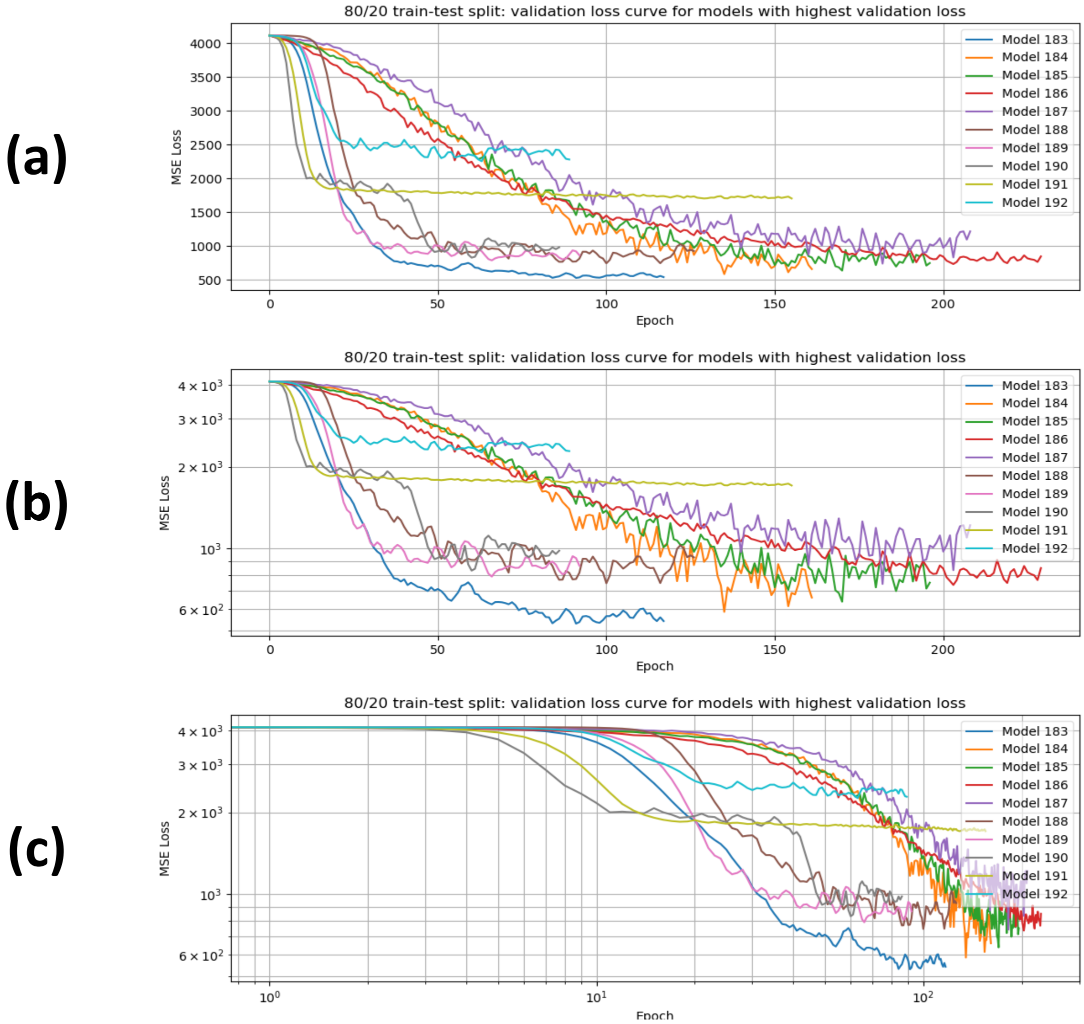


Figure 35: MSE validation loss of the 10 worst models using pad granularity inputs on the 80%/20% datasplit. (a) Epochs and loss on linear scale. (b) Epochs on linear scale and loss on logarithmic scale. (c) Epochs and loss on logarithmic scale.

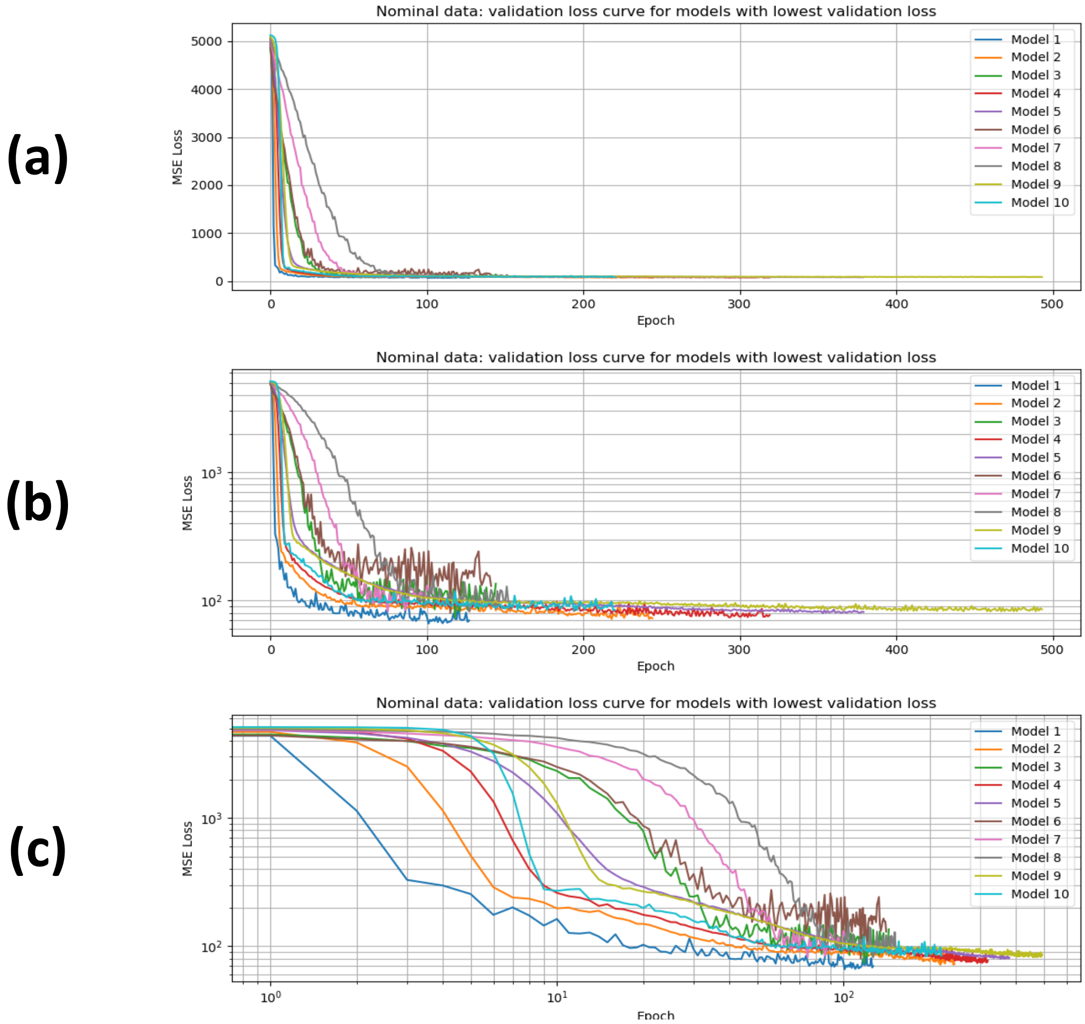


Figure 36: MSE validation loss of the 10 best models using pad granularity inputs on the nominal datasplit. (a) Epochs and loss on linear scale. (b) Epochs on linear scale and loss on logarithmic scale. (c) Epochs and loss on logarithmic scale.

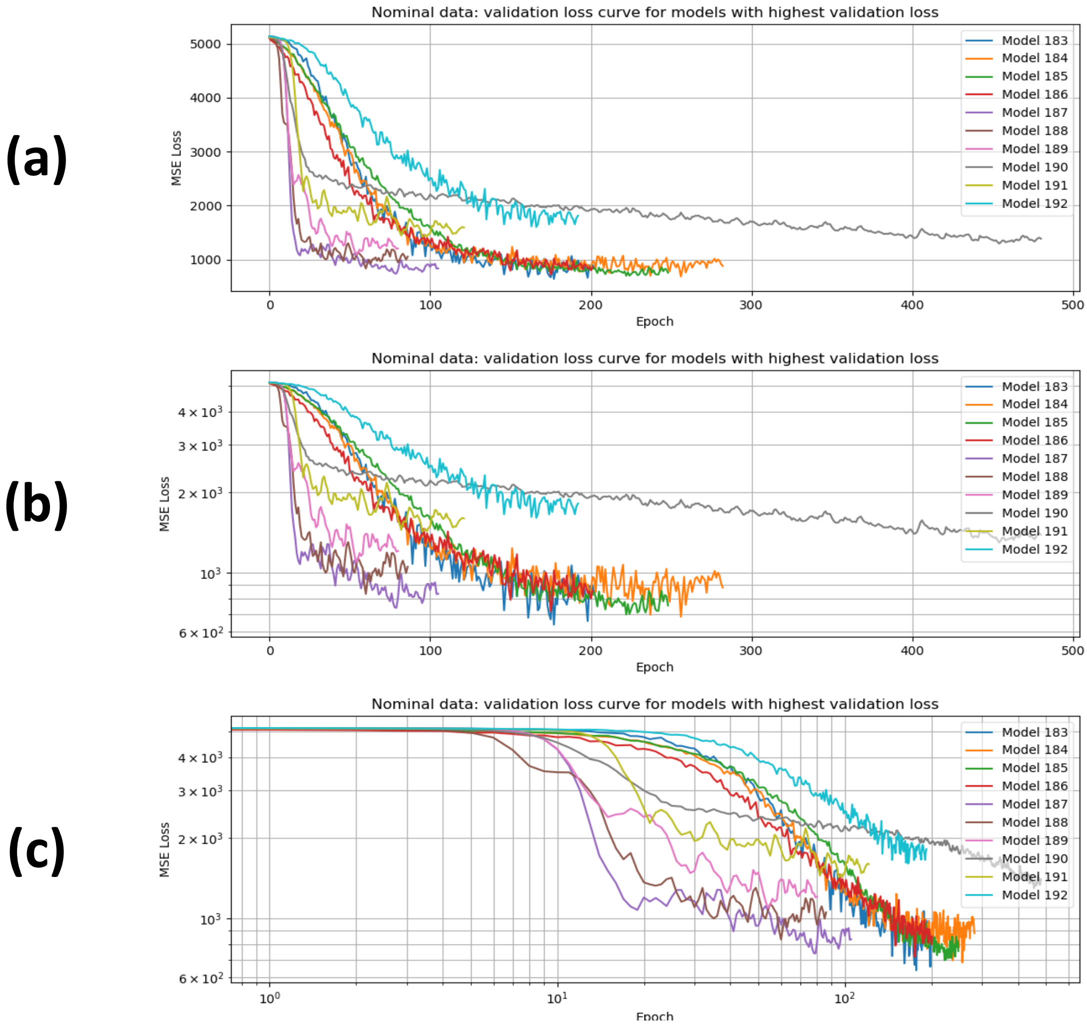


Figure 37: MSE validation loss of the 10 worst models using pad granularity inputs on the nominal datasplit. (a) Epochs and loss on linear scale. (b) Epochs on linear scale and loss on logarithmic scale. (c) Epochs and loss on logarithmic scale.

III. On all data

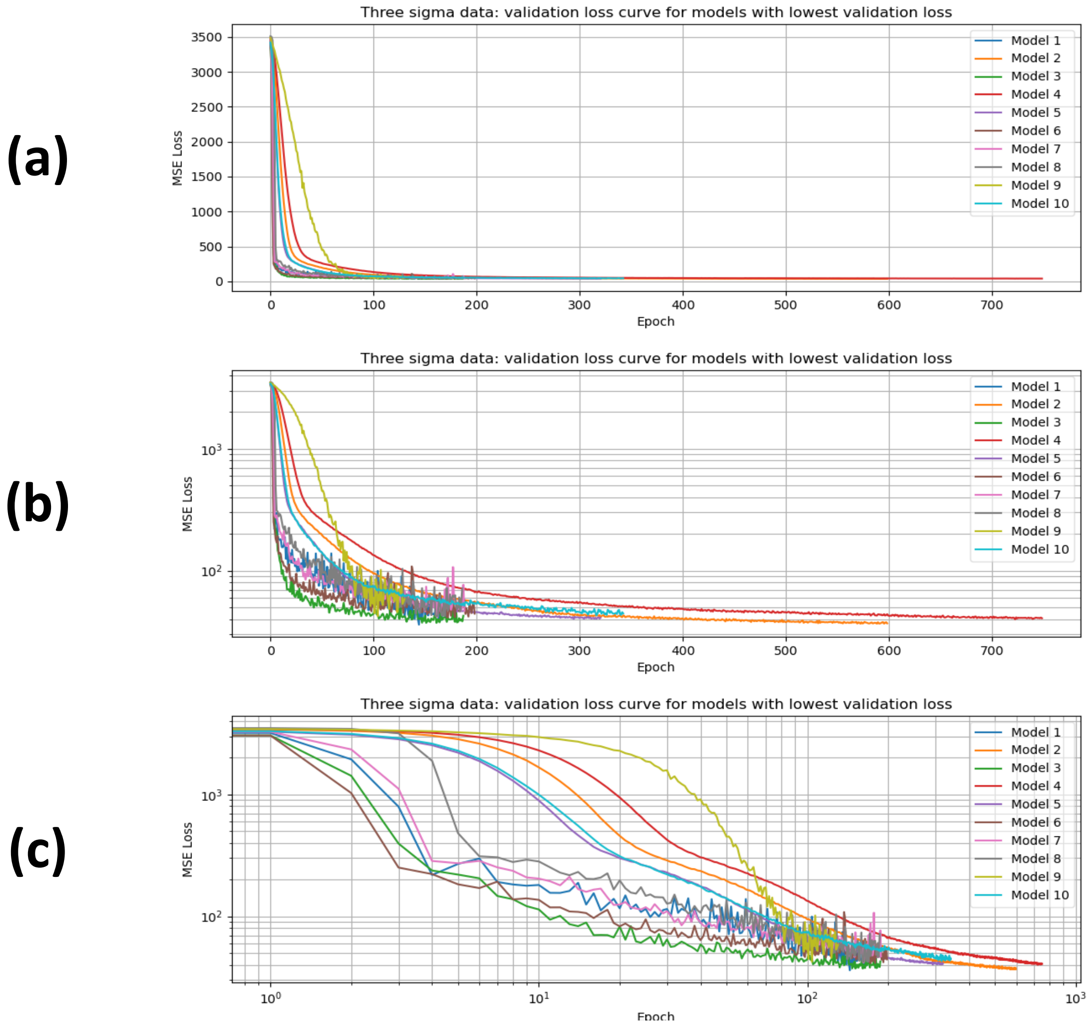


Figure 38: MSE validation loss of the 10 best models using pad granularity inputs on the 3σ datasplit. (a) Epochs and loss on linear scale. (b) Epochs on linear scale and loss on logarithmic scale. (c) Epochs and loss on logarithmic scale.

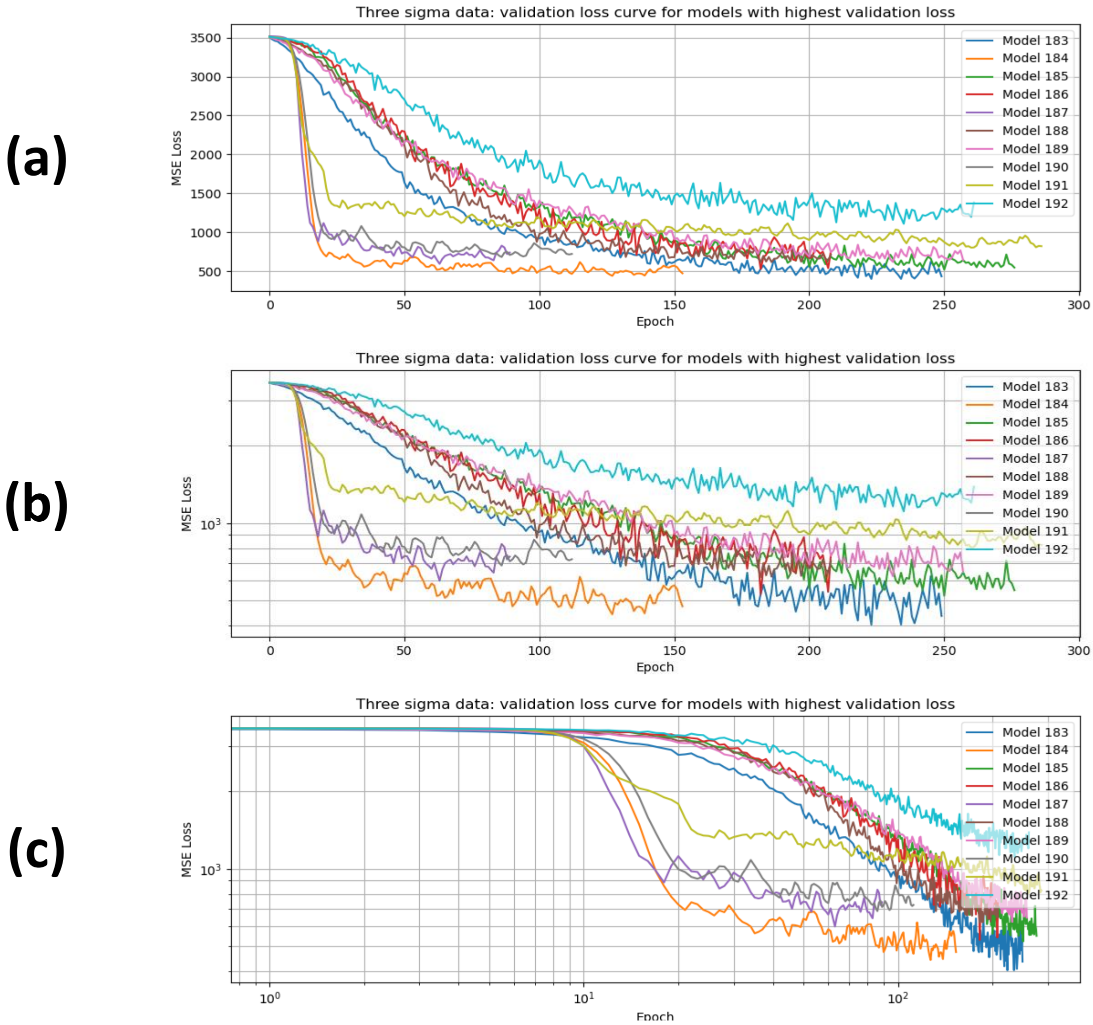


Figure 39: MSE validation loss of the 10 worst models using pad granularity inputs on the 3σ datasplit. (a) Epochs and loss on linear scale. (b) Epochs on linear scale and loss on logarithmic scale. (c) Epochs and loss on logarithmic scale.

SEM granularity

I. 80%/20% random data split

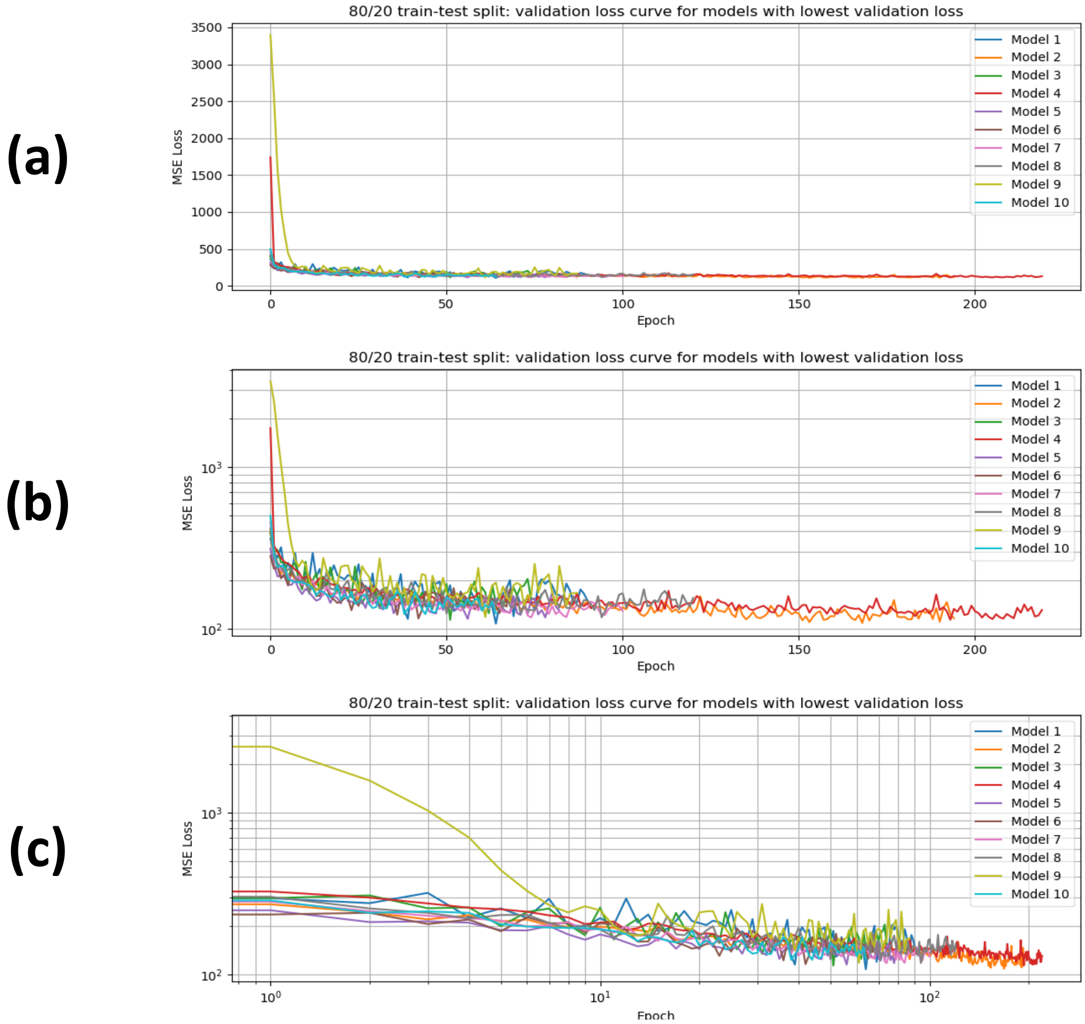


Figure 40: MSE validation loss of the 10 best models using SEM granularity inputs on the 80%/20% datasplit. (a) Epochs and loss on linear scale. (b) Epochs on linear scale and loss on logarithmic scale. (c) Epochs and loss on logarithmic scale.

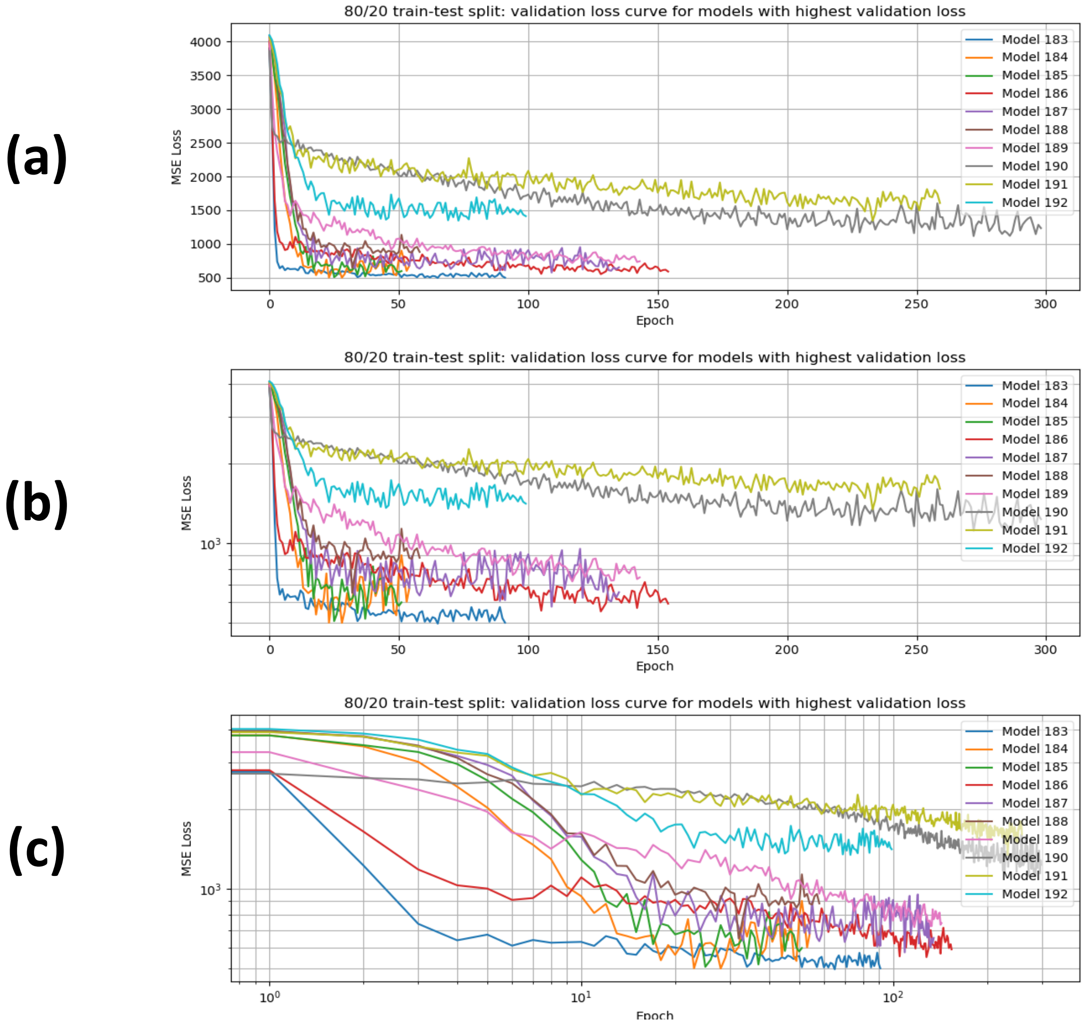


Figure 41: MSE validation loss of the 10 worst models using SEM granularity inputs on the 80%/20% datasplit. (a) Epochs and loss on linear scale. (b) Epochs on linear scale and loss on logarithmic scale. (c) Epochs and loss on logarithmic scale.

II. Nominal range

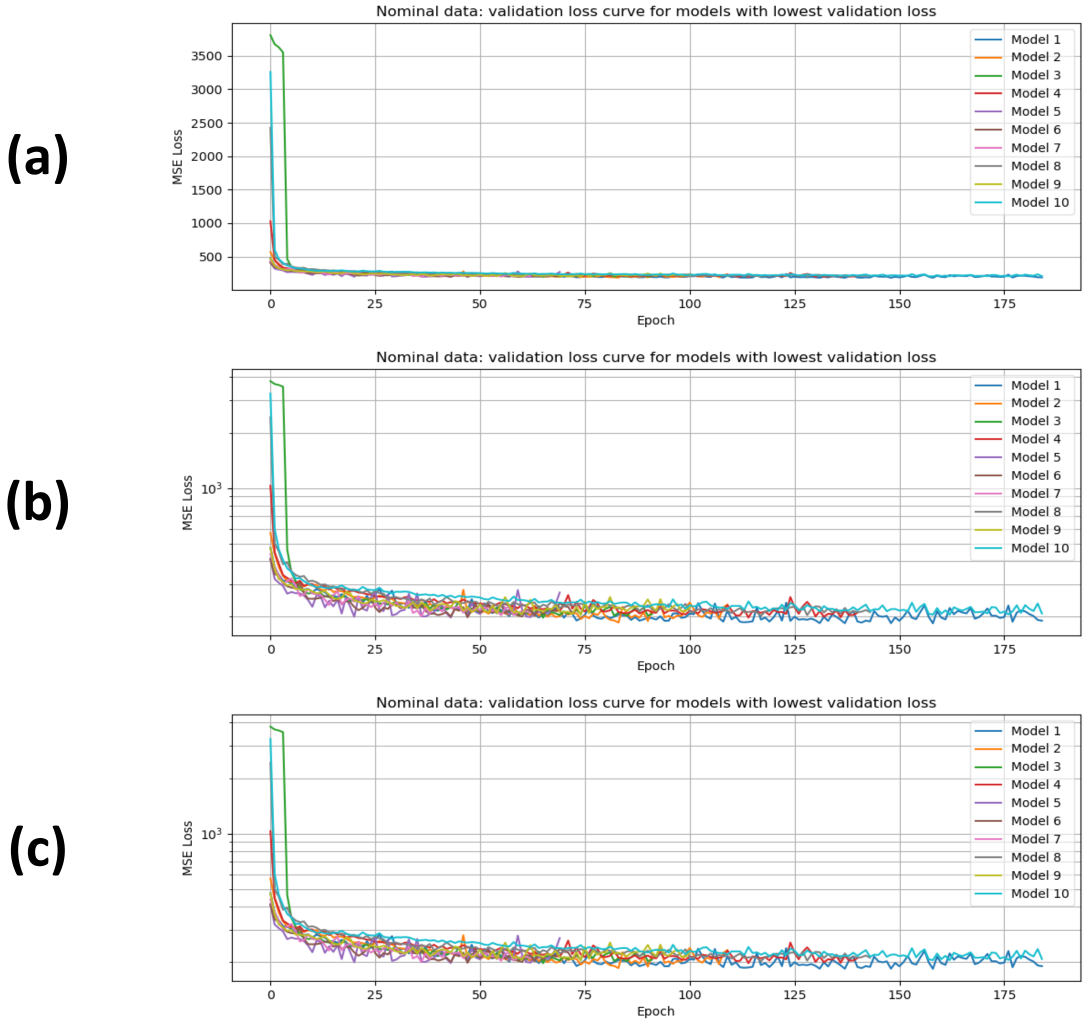


Figure 42: MSE validation loss of the 10 best models using SEM granularity inputs on the nominal datasplit. (a) Epochs and loss on linear scale. (b) Epochs on linear scale and loss on logarithmic scale. (c) Epochs and loss on logarithmic scale.

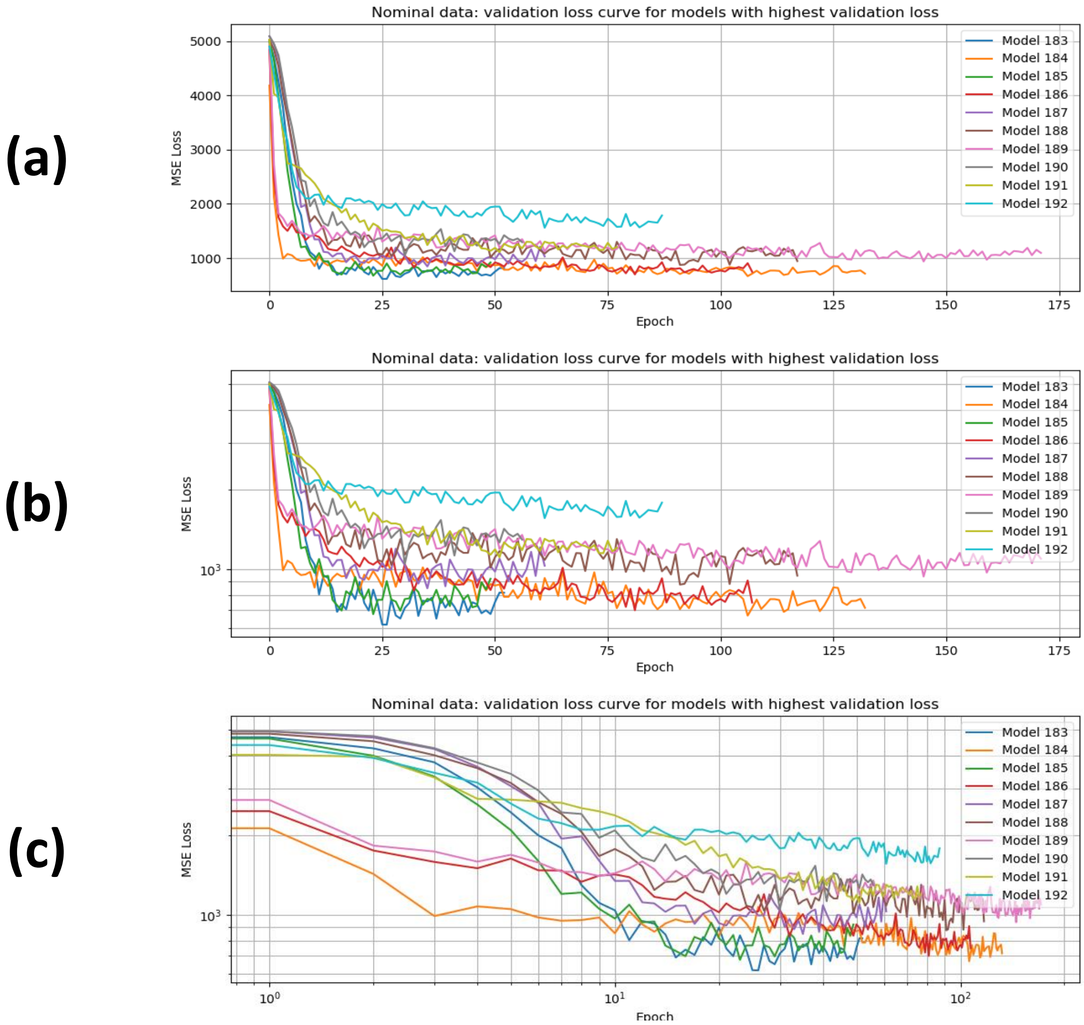


Figure 43: MSE validation loss of the 10 worst models using SEM granularity inputs on the nominal datasplit. (a) Epochs and loss on linear scale. (b) Epochs on linear scale and loss on logarithmic scale. (c) Epochs and loss on logarithmic scale.

III. On all data

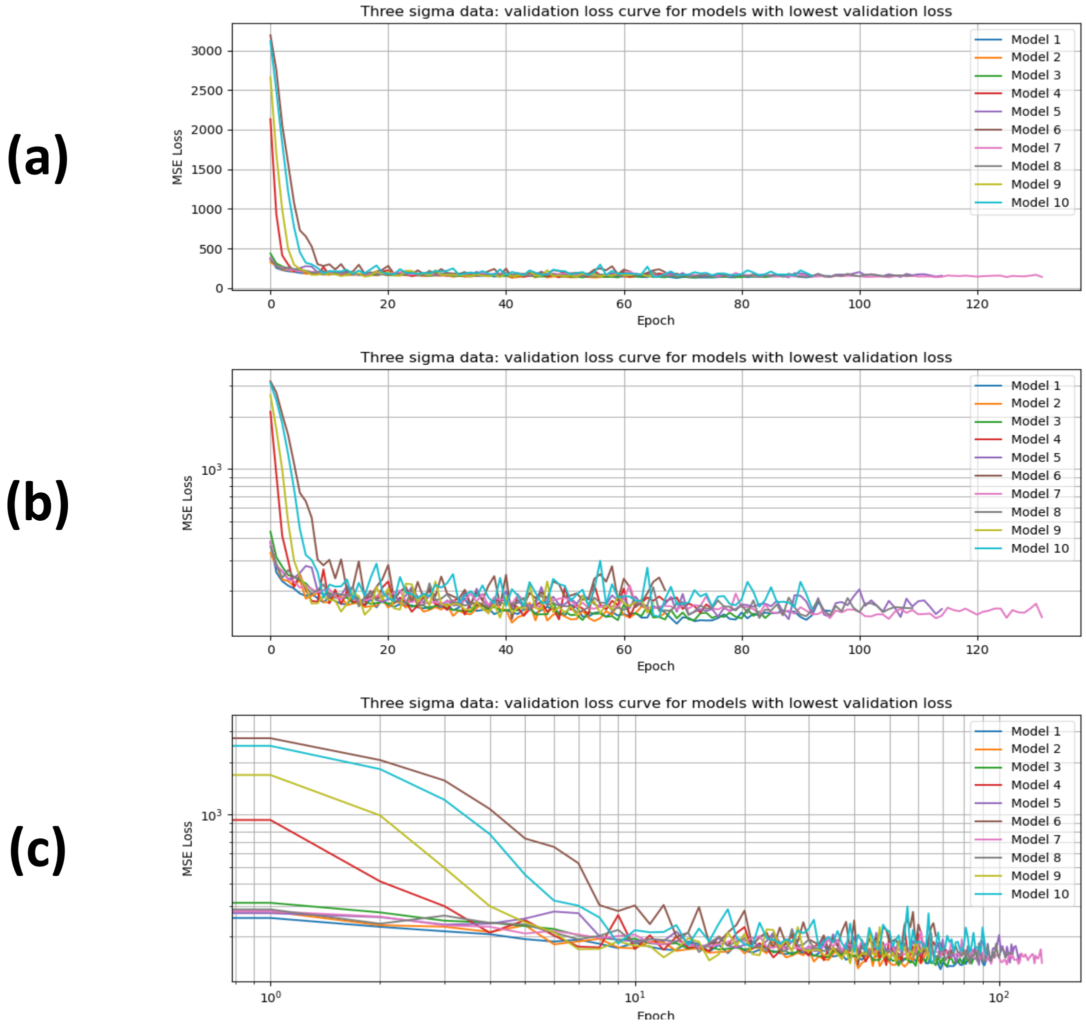


Figure 44: MSE validation loss of the 10 best models using SEM granularity inputs on the 3σ datasplit. (a) Epochs and loss on linear scale. (b) Epochs on linear scale and loss on logarithmic scale. (c) Epochs and loss on logarithmic scale.

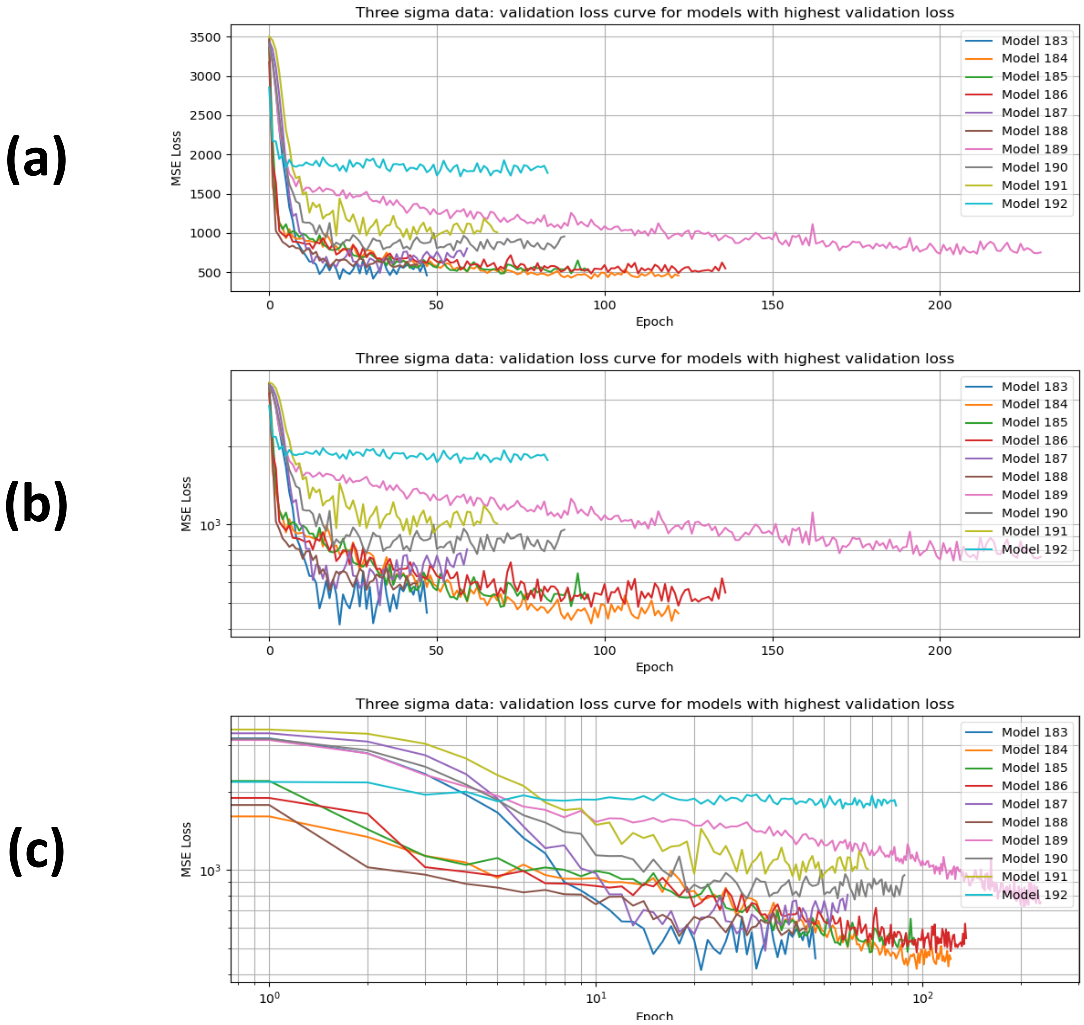


Figure 45: MSE validation loss of the 10 worst models using SEM granularity inputs on the 3σ datasplit. (a) Epochs and loss on linear scale. (b) Epochs on linear scale and loss on logarithmic scale. (c) Epochs and loss on logarithmic scale.

Appendix D - Coefficients of features in linear models

Pad granularity

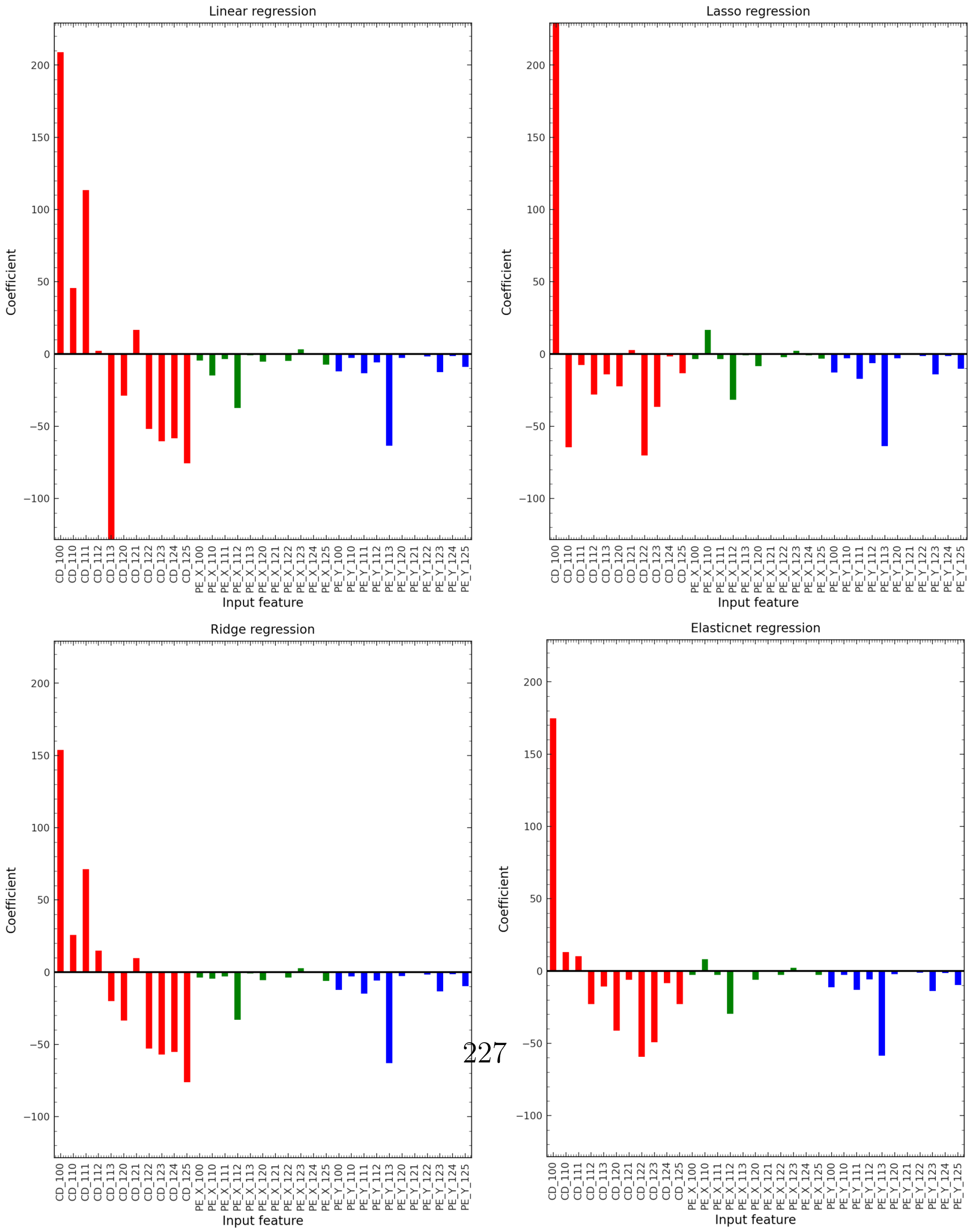
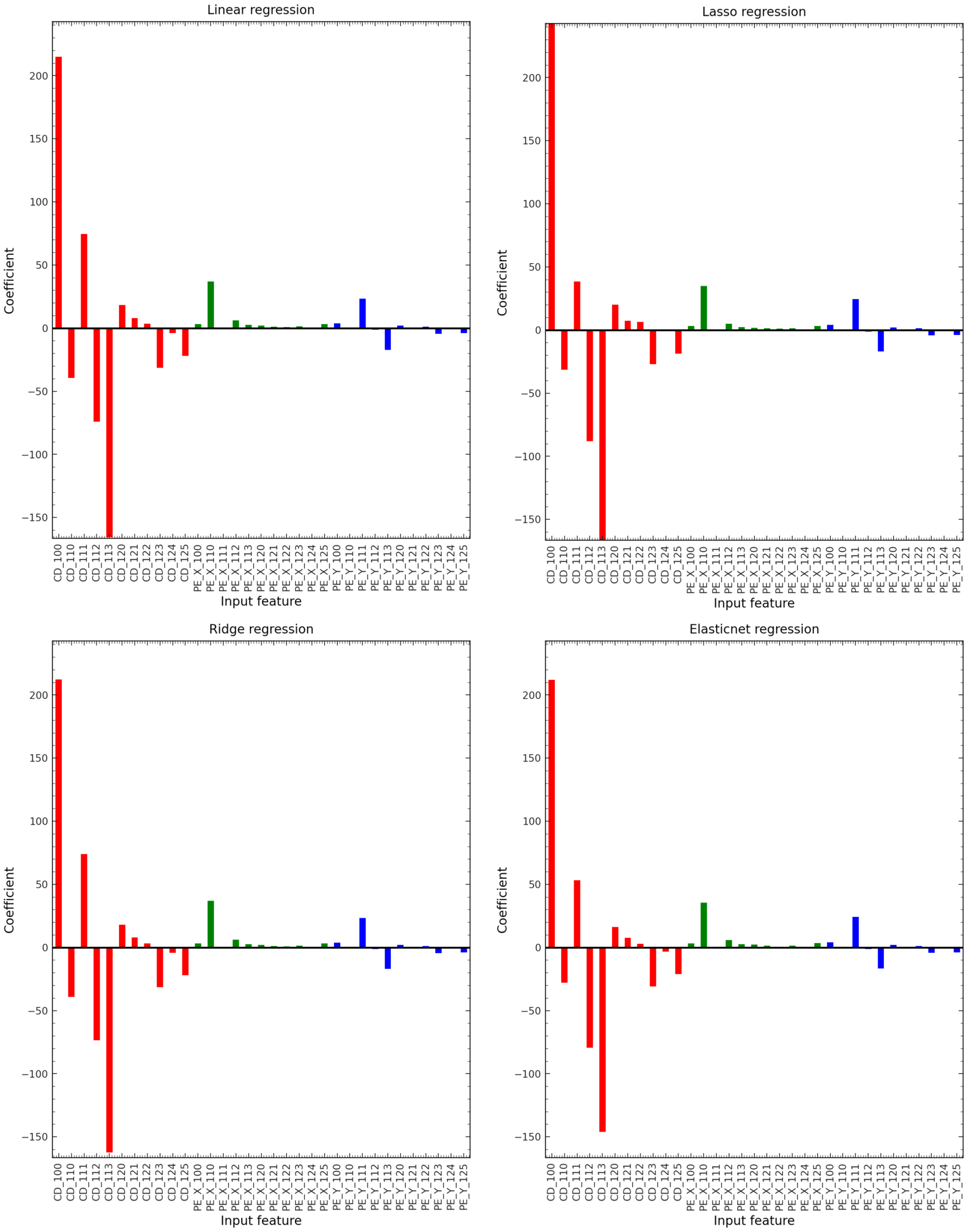


Figure 46: Coefficients of each input feature per model using pad granularity

SEM granularity


Linear regression

Lasso regression

Ridge regression

Elasticnet regression

Figure 47: Coefficients of each input feature per model using SEM granularity input instances.