Universität Rostock
Traditio et Innovatio

Analyzing and Controlling Large Nanosystems with Physics-Trained Neural Networks

Dissertation

zur Erlangung des akademischen Grades

doctor rerum naturalium (Dr. rer. nat.)

am Institut für Physik

der Mathematisch-Naturwissenschaftlichen Fakultät

der Universität Rostock

vorgelegt von

Thomas Stielow, geb. am 17.07.1992 in Stade

aus Rostock

Rostock, den 01.02.2022

https://doi.org/10.18453/rosdok_id00003781

Gutachter

Prof. Dr. Stefan Scheel

Institut für Physik, Universität Rostock

Prof. Dr. Jörg Main

Intitut für theoretische Physik I, Universität Stuttgart

Einreichung: 23.02.2022

Verteidigung: 27.05.2022

Abstract

Simulating real-world processes on a macroscopic scale with microscopic physical theories usually requires extensive numerical calculations. This time-consuming procedure is a limiting factor in the evaluation of modern large-scale experiments. In recent years, neural networks have emerged as powerful artificial intelligence algorithms capable of learning complex relations from just limited sets of training data. This thesis investigates the possible application of neural networks in accelerating the evaluation of physical experiments while minimizing the required simulation effort. One focus is the reconstruction of silver nanoclusters from single-shot wide-angle scattering patterns with only limited information content. It turns out that neural networks are capable of inferring universal reconstruction rules from only a small set of simulated scattering data and offer a significant speed-up compared to classical reconstruction algorithms. When trained directly on scattering theory, the reconstructions reach an unprecedented level of detail, even for objects outside the training data space. Further, a dynamic excitation scheme for giant dipole states of Rydberg excitons in cuprous oxide is derived through deep reinforcement learning interacting with an atomic simulation environment.

Zusammenfassung

Das Simulieren von makroskopischen physikalischen Prozessen mit mikroskopischen Modellen der theoretischen Physik erfordert immense numerische Berechnungen. Der dazugehörige Zeitaufwand schränkt die Auswertungsmöglichkeiten für datenreiche moderne Experimente stark ein. In den letzten Jahren haben neuronale Netze immer mehr an Bekanntheit erlangt als künstliche Intelligenzen, die in der Lage sind aus einem sehr eingeschränkten Datensatz allgemeine Regeln abzuleiten. In dieser Arbeit wird untersucht, wie Neuronale Netze genutzt werden können um die Auswertung von Experimenten durch Minimierung des Simulationsaufwandes beschleunigt werden kann. Ein Schwerpunkt ist dabei die Rekonstruktion von Silber Nanoclustern aus Einzelschuss-Weitwinkel Streubildern, die nur einen eingeschränkten Informationsgehalt haben. Es zeigt sich, dass Neuronalen Netzen bereits aus kleinen Datenätzen allgemeine Rekonstruktionsregeln ableiten können und dabei signifikant schneller sind als klassische Rekonstruktionsalgorithmen. Durch Training durch die Streuphysik selbst erreicht diese Rekonstruktion in einer bisher unerreichten Detailschärfe, sogar für Objekte, die außerhalb des Raumes der Trainingsdaten liegen. Weiter wird ein dynamisches Anregungsschema für Giant Dipole Zustände von Rydbergexzitonen in Kupferoxydul mittels Deep Reinforcement Learning hergeleitet, das anhand einer atomaren Simulationsumgebung trainiert wird.

List of own publications

[1] T. Stielow, and S. Scheel, *Reconstruction of nanoscale particles from single-shot wide-angle free-electron-laser diffraction patterns with physics-informed neural networks*, Physical Review E **103**, 053312 (2021).

[2] T. Stielow, R. Schmidt, C. Peltz, T. Fennel, and S. Scheel, *Fast reconstruction of single-shot wide-angle diffraction images through deep learning*, Machine Learning: Science and Technology **1**, 045007 (2020).

[3] T. Stielow, S. Scheel, and M. Kurz, *Angular-momentum couplings in ultra-long-range giant dipole molecules*, Physical Review A **97**, 022501 (2018).

[4] T. Stielow, S. Scheel, and M. Kurz, *A Green's function approach to giant-dipole systems*, Journal of Physics B: Atomic, Molecular and Optical Physics **51**, 022501 (2017).

[5] S. Ribeiro, S. Buhmann, T. Stielow, and S. Scheel, *Casimir-Polder interaction from exact diagonalization and surface-induced state mixing*, Europhysics Letters **110**, 51003 (2015).

Notation

Mathematical notation

a	scalar variable
$\boldsymbol{x}$	real- or complex valued vector
$\mathbf{A}$	matrix or higher dimensional tensor
x	quaternion

e	Euler's number
i	the imaginary unit
j, k	additonal quaternion imaginary units

Acronyms

AI	artificial intelligence
ML	machine learning
DL	deep learning
NN	neural network
CNN	convolutional neural network
RL	reinforcement learning

Contents

1. Introduction

Recent years have seen a rapid advances in artificial intelligence (AI) to a state where AI systems have integrated in our everyday life [6–8]. This ranges from automatic translation programs capable of translating the meaning of a text (e.g. Google translate or DeepL) [7, 9], over driving assistant systems in cars at the brink of automated self-driving (e.g. Tesla's autopilot), to AIs even beating humans in even the most complex of games (e.g. chess or Go [10, 11]). The common ground of all these advanved systems is their core technique: deep neural networks.

The fundamental idea of a neural network is to mimic the workings of biological nervous systems, like the human brain, with a network of mathematical functions [7]. These basic building blocks are the artificial neurons and their concept dates back all the way to the 1940s [12–14]. Although this model oversimplifies the structure of biological nervous system neural networks were reckoned as powerful mathematical functions, capable of approximating the solution to any given problem [15–17]. Deep neural networks are data driven machine learning (ML) architectures, meaning that the entire relation function is extracted from a given set of data, in contrast to just a few parameters of a given model-function. This, however, comes at the price of requiring an enormous set of function parameters. The extensive computational effort of optimizing these parameters hampered the development of neural networks until the early 2000s [7]. These computational resources became widely available not only over time following Moore's law [18] but also by advances in graphics cards (GPUs). The introduction of Nvidias CUDA library enabled parallelization of the optimization of neural network parameters by calculation on GPUs, that was quickly adopted [19, 20].

The major breakthrough for neural networks, now under the name of deep learning (DL), came in 2012 with the yearly ImageNet image-classification competition, where AI systems are tasked with identifying images from 1000 different classes. For the first time, it was not only won by a neural network [6, 8], but it also outperformed any classical AI method by far. This brought neural networks back to the general attention and they soon overtook the field of computer vision [8]. The development of deep learning solutions was then simplified through specialized programming libraries, like THEANO [21], TORCH [22] and TENSORFLOW [23]. The final factor in popularizing deep learning techniques came with the GeForce 10 series of graphics cards in 2016, that, through their price and energy efficiency, made available the required computation power even to home users.

As of today, the field of computer vision is dominated by deep learning. Since 2012, the aforementioned ImageNet competition is only decided between different neural network architectures and is generally assumed a solved problem [7]. With their capability of image recognition and processing, they find practical applications, e.g. in the interpretation of satellite imagery [24, 25], medical imaging with the identification of tissue types [26] or cancer recognition [27–29]. While the recognition of hand-written text is one of the oldest computer vision tasks solved with neural networks [30], modern architectures are capable of finding text-baselines [31] and even transcribe historical hand-writings [32, 33], like the DL based software TRANSKRIBUS. Aside from recognizing the glyphs of words and texts, specialized neural networks are also used in natural language processing [7] for finding representations for the meaning of a text [34], that in turn is utilized in precise translations of texts [9]. Moreover, neural networks were even developed to create images on their own. Advanced architectures like styleGAN can create photo-realistic images of faces [35] based on comprehensive features and GauGan generates landscapes from simple drawings [36]. Similar, the capability of manipulating existing images has advanced to a point, where entire photo-realistic video sequences can be created [37–39], like an artificial speech of former US president Barack Obama in Ref. [38]. In this, AI has advanced thus far that present society with whole new challenges.

Another field deeply impacted by neural networks is that of problem solving AIs. For a long time, the game of Chess had been a a benchmark system for problem-solving AI systems, until in 1996 the IBM's supercomputer Deep Blue bested the Chess world champion Garry Kasparov [40]. However, more complex games, like Go, were deemed to complex to be ever solved by AI systems. Yet, this assumption was proven wrong in 2016 when the Google DeepMind's AlphGo, an architecture of several neural networks, bested Lee Sedol, one of the world's Go champions [10], in 4 out of 5 games. The follow-up architecture of AlphaGo Zero was able to learn the game even without seeing human strategies but from the rules alone [11], which was then generalized to the games of Chess and Shogi in AlphaZero [41, 42]. With modern architectures, even human level cooperation between independent AIs is in reach, as demonstrated on the team-based computer game DOTA 2 [43]. Neural networks can also be found at the heart of car driving-assistant systems and self-driving agents, like Tesla's *autopilot*.

With the broad range of applications described above neural networks also become interesting as tools for fundamental sciences where they could aid scientists in understanding complex systems. This also includes problems in theoretical and experimental physics. In particle physics, neural networks have had their place already for quite some time: Small neural networks are used in tandem with classical ML methods in the classification of particle signals [44–49]. In other fields of physics, the number of neural network applications has exploded over the past five years, to a degree, that it is almost impossible to keep a complete overview. Thus, the following list can only cover some topics and may not be complete. One of the more popular examples is the control of a magneto-optical trap (MOT) trap [50], which achieved better confinement of a Bose–Einstein condensate (BEC) than any human scientist. On a larger scale, neural networks help in the simulation of complex physical processes in the Wendelstein 7-X nuclear fusion experiment [51–53] and are planned to aid in the control of the confined plasma [54, 55]. In quantum physics, a special form of neural networks, the restricted Boltzmann-machines (RBMs), are used in spin-system simulations [56–60]. Further, neural networks appear both as representations and methods in quantum state reconstructions [58–65] and through their structure enable the construction of generalized entanglement witnesses [57, 66, 67]. Other fields of applications include material sciences [44, 68–72], identifications of phase diagrams [73–76], and the numerical solving of differential equations [77, 78]. The image processing capabilities of neural networks are of direct use in the inversion of diffractive imaging, which appears both in sub-wavelength microscopy and holography [79–83] and single-shot diffractive imaging [84–92].

The goal of this thesis is to further explore possible neural network applications in physics, especially ones that are practically unsolvable with classical method. This thesis project started in 2017, at a time, when many of the works cited above had not yet been published or even developed. We aim to demonstrate the utility of neural networks on two previously unsolved tasks and we introduce the required fundamentals of deep learning in Chap. 2. The first system of interest is an inversion problem in the form of reconstructing nanoparticles from single-shot wide-angle scattering patterns , like in Ref. [93], which usually requires human intervention. With the repetition rate of modern free electron lasers (FELs) human aided reconstruction is futile and automated reconstruction is required. Therefore, implementing the particle identification task with neural networks would provide exactly this key capability and may enable even real-time data-evaluation during experiments. Solving this problem forms the main part of this thesis with Chap. 3. Another focal point of this thesis is utilizing neural networks in the control of a physical system. This is done on the example of an exciton, a quasi-particle similar to an atom, that is to be excited into a giant dipole state. Although predicted by theoretical works since the 1990s, giant dipole states have never been observed to this day, mainly because of lacking simple excitation schemes. This is to be solved in Chap. 4 by steering the system with neural networks. The background of both main topics will be explored in more detail in the beginning of the corresponding chapters. We give an outlook on the consequences of our findings and the future development of neural network applications in physics in Chap. 5.

2. Fundamentals of Deep Learning

Deep learning as a research field has been popularized over the past decade with the wide-spread availability of suitable computation hardware in the form of modern graphics cards. With the advances made, especially in revolutionizing computer vision, deep learning gained the interest of other scientific fields. In physics in particular, the number of publications including deep learning techniques has exploded within the past five years. Nevertheless, it is still an emerging topic and many readers may not be familiar with the specific terminology or internal structure of neural networks. This chapter gives a brief introduction to the topics of deep learning most relevant to and necessary in understanding the later parts of this thesis. An excellent introduction to the field can be found in the book *Deep Learning*, Ref. [7], by Ian Goodfellow, Yoshua Bengio, and Aaron Courville, which also covers general design principles and possible fields of applications. For a more hands-on introduction on implementing deep learning algorithms we recommend Ref. [8], written by Francois Chollet.

Deep learning (DL) is a special form of machine learning (ML), that itself belongs to the overall concept of artificial intelligence (AI), see Fig. 2.1. The field of AI is not precisely defined and can be best described with algorithms and machines, aiding human decision making and solving tasks [7]. The knowledge on how to solve the given problem may be given in imperative statements to the algorithm. In ML, on the other hand, the algorithm has to extract at least a part of the knowledge (usually parameters) from a limited set of data (training data) by itself (the learning), to later apply this knowledge. ML encompasses many different techniques and most physicists may actually be familiar with some ML techniques [44], albeit unknowingly. A fundamental example is finding a generalizing dependency in form of a best fit of a linear regression to a set of measurement data. This requires a quantity to optimize, i.e. in case of a linear regression the mean-squared error between input data and fit-function. Depending on the parameter space and fit-function, the iterative optimization can either be performed directly by gradient methods, e.g. gradient descent optimization, or stochastically, by random Monte-Carlo sampling of parameters. An example of a gradient based ML algorithm is Newton's method for numerical root finding. An example for a stochastic method are random-forest algorithm of randomly modified decision-trees [44, 49].

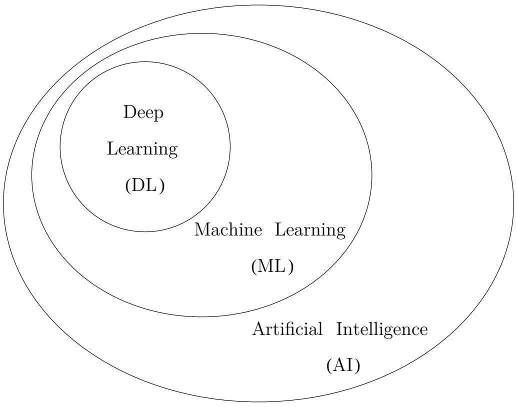


Figure 2.1: Deep learning is a specialized field of machine learning, that itself is part of the larger concept of artificial intelligence. Simplified, after Ref. [7].

An important factor for the success of a ML method is matching the capacity of the algorithm to the complexity of a task. For example, a linear fit function (linear capacity) would be insufficient for fitting a set of quadratic data (quadratic complexity), while higher-order polynomials (polynomial capacity) may deviate strongly for interpolations between the given data-points [7]. The former case of too low capacity is usually called underfitting, while the latter case of too high capacity is called overfitting. When fitting a ML algorithm, the fitting regime can be identified by reserving a part of the training data as a validation set, that is checked against the optimization measure regularly. Usually, underfitting is encountered when the optimization measure stagnates on the validation set, whilst still improving on the training set. Overfitting is observed when the validation measure further diverges from the training measure, by getting worse with further training [7]. Yet, in case of noisy training data and an exceedingly high capacity, overfitting can, to a certain extend, be beneficial to the generalization capability in a process called *benign overfitting* [94, 95].

The distinction of deep learning from classical machine learning lies in building a representation function from chaining of simple basis functions for solving a given task. Although being only a sub-class of machine learning, the possible range of applications for DL stretches that of any other ML algorithm and beyond. However, the capacity of DL algorithms exceeds the complexity of many classical ML tasks. By far the most common architecture for deep learning algorithms are deep neural networks (NNs), to an extend, that the terms of DL and NN are commonly used synonymous.

2.1. Deep Neural Networks

The concept of neural networks dates back to the 1940s. The history of neural network research can be divided into three eras, each coming with different names. The idea of a mathematical function modeling the signal transmission of biological neurons [12, 13] coined the name of artificial neural networks (ANNs) [14] and was part of the greater research field of cybernetics in the 1940s–1960s [7]. The next epoch came under the name of concetionism, in the 1980s and 1990s and saw the emerging of many theoretical work on, from modern view, shallow neural networks, at the times called multi-layer perceptrons (MLPs). This era saw the formulation of the universal approximation theorem, stating that any function is representable by a neural networks [15–17]. Further, back-propagation was firstly used successfully in training a neural network [96, 97]. However, it was believed too unwieldy for any practical application due to the high demand in computation power [7]. This issue led to the decline of neural network research and only few groups continued research. Most prominently, the Canadian Institute for Advanced Research (CIFAR) which united the research groups of Geoffrey Hinton, Yochua Bengio and Yann LeCun [7], possible the three most famous deep learning scientists, today [6].

The third and current era of neural network started in 2006 with the discovery of the first efficient training strategies for deep neural networks [7, 30], rapidly improved their generalization [98, 99]. With this development and ever increasing depths of trained networks also came the notation of deep learning [7]. Outside the existing community, the improvement to neural networks gained attention in 2012, when the yearly image-classification challenge ImageNet was won by a neural network entry from the group of Geoffrey Hinton. In prior years, the competition was tackled with classical algorithm and the 2012 neural network outclassed even the best by a large margin [6, 8]. Today, since then, the challenge is dominated entirely by neural networks and deemed solved with accuracies well above 95%.

The most common building block of modern neural networks is that of a fully connected neuron,

$$y = f\left(\sum_i w_i\, x_i + b\right) = f\left(\boldsymbol{w} \cdot \boldsymbol{x} + b\right)\ . \tag{2.1}$$

It implements an affine transformation of the vector $\boldsymbol{x}$ of input values x_i through the scalar product with the vector $\boldsymbol{w}$ of weights w_i and adds a bias parameter b. The transformation can also be understood as calculating the scaled distance

$$d = |\boldsymbol{w}| \left(\boldsymbol{x} \cdot \frac{\boldsymbol{w}}{|\boldsymbol{w}|} + \frac{b}{|\boldsymbol{w}|}\right) \tag{2.2}$$

to a hyperplane in the space of $\boldsymbol{x} \in \mathbb{R}^n$, defined in Hesse's normal form by normal vector $\boldsymbol{w}/|\boldsymbol{w}|$ and offset $b/|\boldsymbol{w}|$. It is the argument of the, typically non-linear, activation function $f(d)$. In essence, a fully connected neuron evaluates the scaled distance to a hyperplane in a non-linear function. Early neuron designs used either linear [13, 14] or binary activation functions, like the sign(d)-functions [12], that branched into the development of support vector machines (SVMs) [7]. Later, especially during the second era, differentiable but saturating function became popular, like the tanh-function or the logistic sigmoid[1] function

$$\sigma(t) = \frac{1}{1 + \mathrm{e}^{-t}}\ , \tag{2.3}$$

exemplary plotted in Fig. 2.2. Beneficial for gradient-based training, both functions possess iterative expressions of their derivatives [100]. Yet, they are comparably expensive in computation relative to a linear function and in modern architectures are mostly encountered just in the final layer, where their saturation property is used in enforcing parameter bounds. Instead, the rectified linear-unit (ReLU) function

$$\mathrm{ReLU}(x) = \max(0, x)\,, \tag{2.4}$$

was developed [101–103], that to offer an optimal balance between non-linearity and simplicity for a majority of tasks. Yet, some specialized task may benefit from using specific, other activation functions, like in Sec. 3.2 of this thesis.

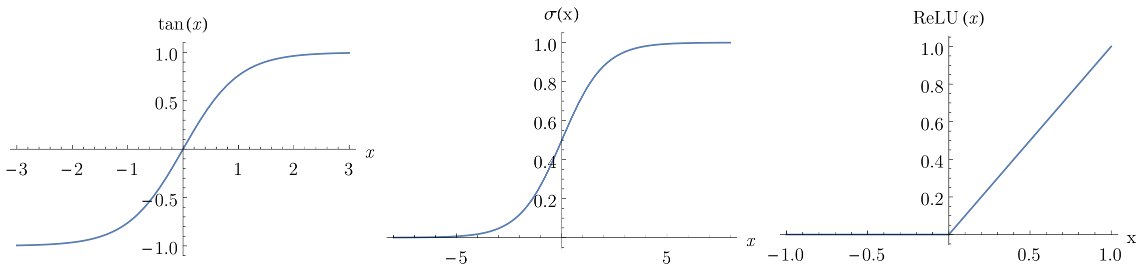


Figure 2.2: The activation functions of neurons most commonly encountered in modern neural network architectures are the tanh-function (left), the logistic sigmoid (center) and ReLU function (right).

The fully connected neuron in Eq. (2.1) itself is a rather simple mathematical function, that is the building block of a deep neural network. Several independent neurons are arranged in parallel, accepting the same input vector $\boldsymbol{x}$ and each return an output y_i, that are arranged in the output vector $\boldsymbol{y} \in \mathbb{R}^n$ for n neurons. By both accepting and returning

[1] In the context of neural networks usually called just *sigmoid*.

vectors, fully connected layers can be stacked after each other, with the output-vector of the previous layer forming the input of the next layer, like sketched in Fig. 2.3. The final layer of the neural network is usually called the output layer and has a number of neurons equal to the dimension of the output vector, e.g. 2 in Fig. 2.3. All previous layers, whose output vectors are processed internally but never returned as outputs of the complete neural network are called hidden-layers. The number of neurons in each layer, also called dimension or size of the layer, can vary and is usually a hyperparameter[2] in designing a neural network.

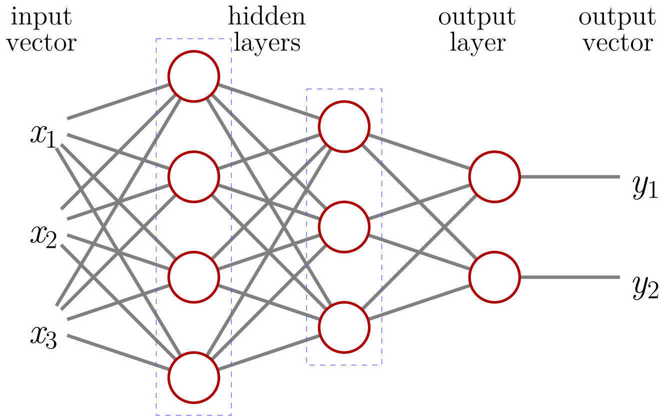


Figure 2.3: A neural network is constructed from single neurons (red circles). Neurons arranged in parallel, accepting the same input vectors form a layer, like the ones inside the blue, dashed boxes. The output values of the neurons from the final, output layer compose the output vector. Internal layers that again feed their own outputs into other layers are called hidden layers, as they are not visible to an outside observer.

Knowing that each neuron implements a distance measure to a hyperplane, the stacking of fully connected layers also bears a geometric interpretation. A single layer implements several hyperplanes into the input space $\mathcal{X}$ and returns a vector of distance measures.

Especially with ReLU or tanh-activation, these hyperplanes are decision boundaries, where the activation function changes significantly, e.g. in slope or sign, respectively. Each neuron of the next layer then implements new decision boundaries on the previous decisions, combining them into regions. Without activation functions, these decision regions on $\mathcal{X}$ would be faceted polytopes. Here, the non-linearity of the activation function introduced previously comes into play. It allows for an effective curving of the region boundary in the vicinity of intersections of hyperplanes from the previous layer. The effect of this boundary shaping is visible in Fig. 2.4, where a circle in $\mathbb{R}^2$ is approximated with a two-layer neural network with just three hidden and one output neuron. While the three initial decision boundaries (white lines) form only a triangle, the combined decision boundary (white area of the red-white-green colormap) is close to circular (black curve).[3] In fact, it was proven with the universal approximation theorem, that a neural network with just one, sigmoid-activated, hidden layer and a single output neuron can approximate any

[2] A hyperparameter is a parameter of either the ML algorithm or training process, that is not optimized by training but pre-definined. Hyperparameters also require optimization that usually requires realization through systematic exploration rather than gradient methods.

[3] Of course, this very fundamental problem is trivially solved in polar coordinates by a single decision plane. From this view, the hidden layer implements a coordinate transformation, projecting the circular region into a linear, three-dimensional set, while the second layer places the decision plane. However, finding the right coordinate transformation (or kernel for a SVM) for a given problem usually involves human intervention while the neural network is capable of inferring it from data.

continuous function up to any precision desired, with a sufficient number of neurons in the hidden layer [15–17]. However, it does not make a statement on how may neurons are actually required, which may as well be infinitely many for some cases. While such elementary neural network architectures may seem a pure academic example, they actually have precedence in nature. The feeding behavior of the pond snail *Lymnaea stagnalis* was found to be controlled by just two neurons,[4] one activating if the snail is hungry and the other if there is food nearby [104].

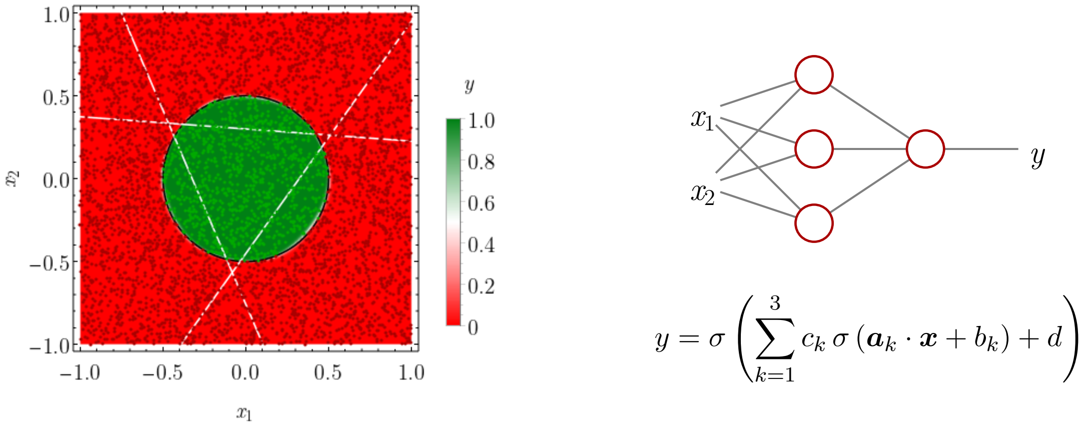


Figure 2.4: The classification problem of a circular region (black contour) into a region inside (green, label 1) and outside (red, label 0) is well approximated by a neural network with a single hidden layer (right) of just three sigmoid-activated neurons (see formula below). The whole network is constituted from just 13 parameters. The decision planes of the three hidden neurons form an equilateral triangle (white lines). The red-white-green colormap in the background shows the resulting decision map of the complete neural network, that achieves a near circular decision boundary (white region, nearly obfuscated entirely) through combining the non-linear activation results. The network is trained on a set 5000 random points (red/ green dots), but can also be optimized analytically using a series approximation.

The near perfect approximation of an explicitly known function is, however, not the task neural networks are usually required for. Instead, neural networks excel in data driven application, where the relations underlying to a set of data is too complex to be represented with any classical machine learning algorithm. The most common type of datasets is that of projections $\mathbf{x}_i \rightarrow \mathbf{y}_i$ between input tensors $\mathbf{x}_i$ of arbitrary shape and corresponding output or target tensors $\mathbf{y}_i$. On the example of Fig. 2.4, these would be two-dimensional vectors $\boldsymbol{x}_i \in \mathbb{R}^2$ and boolean values $y_i \in [\text{False}, \text{True}]$. The optimization quantity of the neural network $\mathcal{N}(\mathbf{x})$ on such a dataset is the loss-function $L(\mathbf{y}_i, \mathbf{p}_i)$ between the neural network predictions $\mathbf{p}_i = \mathcal{N}(\mathbf{x}_i)$ and the targets $\mathbf{y}_i$. The optimal choice of the loss-function is crucial for a good training result and is usually determined by the nature of the target data. Regression problems, where the output values are ordinary float-point numbers, are handled well by the mean-squared error

$$L_{\text{MSE}}(\boldsymbol{y}, \boldsymbol{p}) = \frac{1}{N} \sum_{i=1}^{N} (y_i - p_i)^2 \,, \tag{2.5}$$

between linear vectors[5] $\boldsymbol{y}, \boldsymbol{p} \in \mathbb{R}^N$. Other loss-function are derived from information theory, such as the Kullback-Leibler-divergence between two probability distribution functions or

[4] To be precise, not two singular neurons but two neuron types, responsible for the snail to switch between two distinct feeding behaviors: appetite bites and consummatory feeding [104].

[5] Note, that any tensor of arbitrary shape can be re-arranged into a linear vector.

cross-entropy functions for classification tasks [7, 8]. In the context of this thesis, the binary cross-entropy

$$H(\boldsymbol{y}, \boldsymbol{p}) = \frac{1}{N} \sum_{i=1}^{N} y_i \, \log(p_i) + (1 - y_i) \, \log(1 - p_i) \tag{2.6}$$

between two Bernoulli-distributions, i.e. binary values equivalent to labels True/False or object/void, and it's extension to systems with a number of C classes in the categorical cross-entropy

$$H_C(\boldsymbol{y}, \boldsymbol{p}) = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{C} y_{i,j} \, \log(p_{i,j}) \tag{2.7}$$

are of importance.

The high number of free parameters in the weights of any single neuron inside a neural network typically results in a much higher representation capacity as supported of the present training data. In many cases, the number of free-parameters may even exceed the number of available data samples.[6] Consequently, neural networks are prone to overfitting, which to a certain degree can be beneficial as *benign overfitting* [94, 95], but also may decrease the generalization capability on data outside the training set [7]. In DL, overfitting can be countered[7] through regularization, which summarizes different techniques on adding constraints to the training task [7]. One example are additional loss terms, called regularization functions, that penalize certain configurations, like the L_1-regularization that adds the sum of all neural networks weights to the loss, thus penalizing large weights for singular inputs and enforcing a more distributed attention. Other regularizations act on the structure of the neural network like dropout-layers, that randomly delete entries of incoming vectors during training, thus muting sections of the neural network and enforcing a more distributed training. Another regularization approach is to artificially increase the amount of available training data, called data-augmentation. It is commonplace in computer vision, where images are either rotated or shifted to represent additional angles of view of noise and defects are added, to focus the recognition of the most important features [30]. Some design strategies rely on first designing a neural network architecture capable of overfitting the available data, thus ensuring sufficient capacity, and then introducing regularization to counteract overfitting and increase generalization [8].

The optimization of the neural network parameters $\boldsymbol{\theta}$ in accordance with the loss is realized though gradient-descend. It involves calculating the gradient of the loss function L with respect to the set of parameters $\boldsymbol{\theta}$ and the subsequent adaption of the parameters,

$$\boldsymbol{\theta} \to \boldsymbol{\theta} - \alpha \boldsymbol{\nabla}_{\boldsymbol{\theta}} L \,, \tag{2.8}$$

by a step α in descending direction of the gradient. The gradient calculation of deep neural networks is computationally expensive and is what held back neural network development until recently. While it seems desirable to calculate the gradient over the complete dataset and find the global minimum, it still is too resource-demanding in many cases. Instead, the gradient is calculated for smaller batches of the dataset individually, holding the additional benefit of converging much faster due to more frequent parameter updates [7]. By, over

[6] Note, that in some cases not all of these parameters can be chosen freely but may depend on each other.

[7] Another common method present in many tutorials is aborting the training at the optimal validation-set performance, called early-stopping. However, it is often inferior to regularization approaches that can lead to even better generalization capabilities.

time, averaging to a noise version of the complete gradient, this approach is called stochastic gradient descent [6]. Each iteration of the complete dataset is called an epoch, episode or round. The step size α by which the parameters are modified in Eq. (2.8) is determined by the optimizer function. It may not only depend on the current value of the gradient but also past values, in form of a momentum [105]. The optimizer can prevent the training from converging to local minima or around saddle points, where the gradient vanishes, and is of such importance, that it is dedicated an entire chapter in Ref. [7].

2.2. Convolutional Neural Networks

The fully connected neuron in Eq. 2.1 defines the basic principle of operation, from which many other neuron types are derived. An important development, especially in image procession architectures, is that of the convolutional neuron. When working with two dimensional data. like images, fully connected layers have three main drawbacks [106]:

1. The number of weights increases rapidly with the number of pixels in an image. A 64 neuron layer processing a 128 × 128 pixel image already needs more than one million weights.
2. The fully connected weights are position sensitive. The same object inside an image, shifted by a small margin, is not detected.
3. The local structure and correlation of an image may be unrecognized in favor of larger, global features. However, small scale correlations, like edges, are important features in image processing.

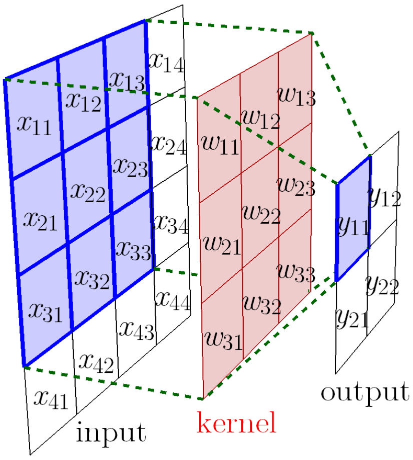


Figure 2.5: A convolutional neuron acts on an input image by local convolution with a kernel matrix, producing a single output value. Scanning the kernel the input image produces a feature map in form of an output image.

The solution comes in the form a convolutional neuron, that acts only locally on an image by convolving it with an input matrix, the kernel. As shown in Fig. 2.5 for a 3 × 3 matrix, the kernel usually is smaller than the input image. It therefore acts only on a part of the image with the same size as the kernel, but in this window is evaluated just like a fully connected neuron in Eq. 2.1. By scanning the convolution kernel over the input image, a new output image is created, which contains the activation value of the convolutional neuron over the input, and therefore is called a feature map of the kernel. The convolutional neuron is capable of detecting only local structures, like gradient or edges independent of their position inside the image an map them into the feature map. Scanning along the edges of the image reduces the lateral (or image) dimension of the feature map by half the kernel size, which, if desired, can be prevented by scanning beyond the image borders and filling missing pixels either by constants, like zero, or symmetric, in a process called padding. The size of the output image is further influenced by the step-size with which the kernel is transported, the stride. Usually, a stride of 1 is used, where the convolution kernels of consecutive steps overlap partially. However, a stride of 2 can be used to half the lateral dimensions of the input image, which is exploited in some architectures.

Similar to fully connected layers, a convolutional layer is formed by a parallel evaluation f convolutional neurons on the same input, producing f individual feature maps or filters. When feeding into the next convolution layer, each kernel matrix now requires a shape of $k \times k \times f$, where k is the kernel-size, to be sensitive to all previous feature maps. In stacking convolution layers, like in Fig. 2.6, each consecutive convolution filter has access to higher-order features, that cover increasing areas of the input. For example, an image of a cat could be recognized by first detecting edges and gradient, combining those to surfaces, which then combine to more global structures like eyes, legs and ears and finally combine to a cat. The real recognition process, however, is much more complex and locally driven than usually assumed [107].

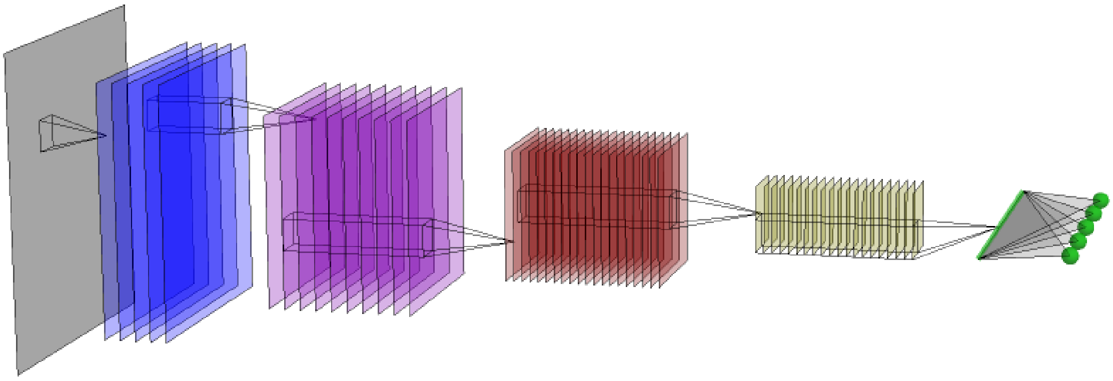


Figure 2.6: Convolutional neural networks are bulid from stacking convolution layers. Here visualized are 4 consecutive convolution operations, each with stride 2 (halving the lateral dimensions), with 4, 8, 16 and 16 filters, each. The feature maps of each convolutions are grouped in equal colors. The last stack of feature maps is flattened into a one-dimensional vector and fed into five fully-connected neurons.

The earliest architecture build around convolutional layers was LeNet [106]. In addition to convolutional layers with stride 1 and kernel sizes of 3 × 3, it uses pooling layers for dimensional reduction. A pooling layer acts similar to a convolution layer with kernel size 2 and stride 2, in that is scans over the input image with a limited filter window, but then applies fixed operations like taking the max- or mean-value within the window. A general design principle for convolutional neural networks is to double the number of filters after each lateral dimension reduction [7]. After a certain number of convolutions, the entries of the resulting feature maps are re-arranged into a linear vector through a flattening-layer. It is the fed into 2–3 fully connected layers, that return the final output tensor. Based upon this design, many modern convolutional neural network architectures were developed, like ResNet [108, 109], ResNeXt [110], or fully-convolutional dense-nets (FCdense) [111]. The concept of convolutions layers also allows generalization to three-dimensional input tensors and kernels or even arbitrary dimensions.

2.3. Deep Reinforcement Learning

Reinforcement learning (RL) is a field of ML, that revolves around learning to perform actions that maximize a returned numerical reward. Deep reinforcement learning then focuses on solving any such tasks with the help of deep learning methods, or more specifically: neural networks. Solving RL tasks requires additional fundamental equations and terminology to the previously discussed feed-forward DL problems. A detailed introduction into the topic of RL can be found in the book *Reinforcement Learning: An Introduction*, Ref. [112], and a brief introduction focused specifically on deep RL applications is given on the website of the OpenAI project, Ref. [113].

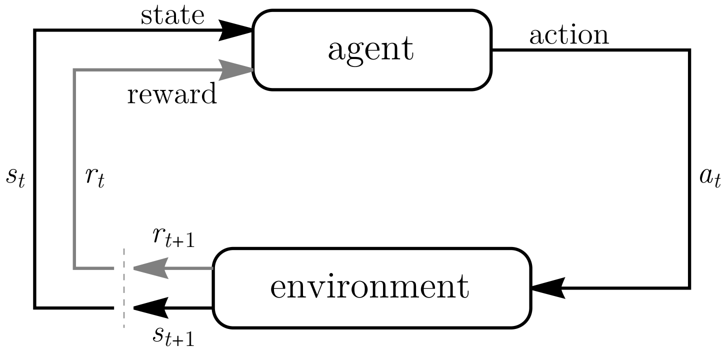


Figure 2.7: In the agent-environment loop, the agent receives an observation of the current state s_t of the environment, nd decides on an action a_t to take on the environment. This action alters the state of the environment, resulting in the next observable state s_{t+1}. At each step, the agent receives the reward r_t as a performance measure from the environment. After Ref. [112].

The fundamental concept of RL is that of an agent interacting with an environment. The agent can observe the state s of the environment and decides on an action a to perform, which may alter the state s. Further, the agent receives feedback on how good the state s is, in relation to a given goal, in form of the reward r from the environment. The majority of RL theories demand finite Markovian environments, meaning that the future evolution of the system is fully determined by the current state s and independent of the past states. The interaction is usually simulated in cycles, as sketched in Fig. 2.7. An important restriction of some environments is only partial observation, where the agent can not directly access s, but only an observation o with partial information of s. However, this has only limited effect as long as the reward also depends only on o, the information available to the agent [112]. Thus, s is commonly used instead of o in RL literature [113] and we will conform with this notation. The reward r_t received at any step is determined by the (environment-specific) reward function $\mathcal{R}$

$$r_t = \mathcal{R}(s_t, a_t, a_{t+1}) \tag{2.9}$$

from at most the current state s_t, current action a_t and next state s_{t+1}. In many cases, however, it depends just on the current state $r_t = \mathcal{R}(s_t)$ [113]. In its actions, the agent aims to maximize the total reward accumulated over (a usually finite) interaction time with the environment, the return R. Thereby, the agent is reinforced in taking the right actions to achieve the goal of the environment.

The set of rules by which the agent decides which action to take based upon a state s_t (or observation o_t) is the policy[8] [112]. Fundamentally, one has to differentiate between deterministic and stochastic policies. Deterministic policies return a specific action, given a state, while a stochastic policy returns a probability distribution over the available action

[8] The terms agent and policy are often used interchangeable.

space, from which a decision on the action is made. This can be done by either choosing the action of maximal likelyhood (greedy decision) or randomly sampled from the distribution. The convention is, to denote deterministic policies by μ

$$a_t = \mu(s_t)\,, \tag{2.10}$$

and stochastic policies by π

$$a_t \sim \pi(s_t)\,. \tag{2.11}$$

Depending on the environment, the action space, and thereby the return value of the policy, can be either discrete or continuous. Usually, discrete action are easier to handle and are found in many classical arcade or computer games (like moving left or right). Thus, they are commonly used in RL examples, while real-world systems usually allow continuous action (like moving through velocity in either direction). This distinction is so fundamental, that many RL architectures function exclusively with either one or the other [113].

The evolution of an agent-environment interaction in the alternating sequence of states s_i and actions a_i is a trajectory

$$\tau = \{s_0, a_0, s_1, a_1, \dots\}\,. \tag{2.12}$$

Given a trajectory τ, and recalling the reward calculation in Eq. (2.9), the infinite-horizon discounted return [112]

$$R(\tau) = \sum_{t=0}^{\infty} \gamma^t\, r_t\,, \tag{2.13}$$

with the discount factor $\gamma < 1$ is calculated. It is usually preferred over the flat sum of all r_i, because it is guaranteed to converge for (close to) infinite trajectories and gives higher weight to earlier rewards, thereby effectively favoring sooner rewards from immediate actions over later rewards.

Problematically the rewards from Eq. 2.13 can only be calculated after the fact from a given trajectory. Yet, the task of the agent is choosing the right action a_t in the current state s_t, that will result in the maximal return. This is formalized in the value function $V^\pi(s)$, giving the value of a state s in terms of expected return when acting to the policy π (or μ), and the action-value-function $Q^\pi(s,a)$, telling the expected return when in the state s taking the action a and then continuing accordingly.[9] Ultimately, one is interested in the optimal value functions $V^*(s)$ and $Q^*(s,a)$, telling the values of state and action if we act according to the optimal policy [112]. Logically, if we know $Q^*(s,a)$, we can act according to the optimal policy by, each time, choosing the optimal action

$$a^*(s) = \operatorname*{argmax}_a Q^*(s,a) \tag{2.14}$$

for the given state s.

At this point, is seems like we have come full circle, in that we determine the optimal policy from the optimal state-value function of a state and action when acting according to the optimal policy. The value functions, however, obey a recursive relation, following the basic idea, that the next value is equal to the current value plus the current reward.

[9] Note, that $V^\pi(s)$ and $Q^\pi(s,a)$ are connected by the chance of taking the action a

These are the Bellman equations, that for the optimal value functions read

$$V^*(s) = \max_a \mathop{\mathbb{E}}_{s'\sim P}\left[r(s,a) + \gamma\, V^*(s')\right] \tag{2.15}$$

$$Q^*(s,a) = \mathop{\mathbb{E}}_{s'\sim P}\left[r(s,a) + \gamma \max_a Q^*(s',a')\right] \tag{2.16}$$

where $\mathop{\mathbb{E}}_{s'\sim P}[\dots]$ is the expectation value regarding s' sampled from the transition rules of the environment [113], that becomes just the evaluation of s' if the environment is deterministic. These equations can be solved iteratively by using approximations of the optimal value functions until convergence to the actual, optimal functions is achieved [112]. To some degree, value function approximation is used in all RL algorithms [113]. The value function is either used in learning the optimal policy for judging the current performance or by directly learning the action-value policy function (also called Q-learning) or even a combination of both [113].

2.3.1. The DDPG algorithm

For the control of a physical system in Chap. 4, we use the deep deterministic policy gradient (DDPG) algorithm from Ref. [114]. As the name gives away, DDPG is a deterministic policy algorithm implemented with neural networks. At the time, it represented the first implementation of a deterministic actor-critic model, in contrast to the classical, stochatic and discrete actor-critic [115] models. Actor-critic methods simultaneously learn both the optimal policy μ, in form of the actor, and the action-value function Q, in form of the critic. In case of DDPG, both are implemented as neural networks. An implementation of DDPG in Keras and TensorFlow can be found in the Keras github-repository, Ref. [116].

DDPG is trained in an off-policy way, meaning that training data is not required to be sampled with the most-recent policy. Instead, the training data of any episode is collected in a buffer in pairs,

$$\{s_t, a_t, r_{t+1}, s_{t+1}\}\,, \tag{2.17}$$

of previous state s_t, taken action a_t, resulting reward r_{t+1} and new state s_{t+1}. From this buffer, batches $\mathcal{D}$ of random samples are used to learn the optimal action-value function $Q^*(s,a)$ through the Bellman-equation. The self-consistent form in Eq. (2.16) comes with the problem of instability, which is resolved by introducing additional target networks to the training. For any element $\boldsymbol{d} \in \mathcal{D}$, the right hand side of Eq. (2.16) is then calculated as

$$y(\boldsymbol{d}) = r_{t+1} + \gamma\, Q_{\text{targ}}(s_{t+1}, \mu_{\text{targ}})\,, \tag{2.18}$$

using the target versions of both the actor μ_{targ} and critic Q_{targ}. The actual critic is then optimized for minimizing the mean-squared error between it's predictions and the corresponding result of Eq. (2.18) in a normal supervised way, with the mean-squared-Bellman-error loss

$$L(\mathcal{D}) = \mathop{\mathbb{E}}_{\{s_t,a_t,r_{t+1},s_{t+1}\}\in\mathcal{D}}\left[\Big(Q(s_t,a_t) - \big(r_{t+1} + \gamma\, Q_{\text{targ}}(s_{t+1},\mu_{\text{targ}})\big)\Big)^2\right]. \tag{2.19}$$

Then, according to Eq. (2.14), the current best policy μ is the one maximizing Q on the current set $\mathcal{D}$

$$\max_\mu \mathop{\mathbb{E}}_{s\in\mathcal{D}}\Big[Q(s,\mu(s))\Big]\,. \tag{2.20}$$

In practice, this is achieved by training μ on the set of all states $s_t \in \mathcal{D}$ with the critic Q itself acting as the loss function by gradient ascend.

The two target networks, μ_{targ} and Q_{targ} are created from the actor and critic initially as direct copies. However, upon training, their parameters are being adapted only slowly by fractional copying the parameters θ and ϕ of the actor and critic, respectively. This is done by setting

$$\theta_{\text{targ}} = (1-\tau)\,\theta_{\text{targ}} + \tau\,\theta\,, \tag{2.21}$$

$$\phi_{\text{targ}} = (1-\tau)\,\phi_{\text{targ}} + \tau\,\phi\,, \tag{2.22}$$

for the parameter θ_{targ} for the target actor μ_{targ} and ϕ_{targ} for the target critic Q_{targ}, respectively, with a small weigth $\tau = 0.005$. This cycle is repeated, parallel to acquiring ever more data from agent-environments loops, until convergence is achieved.

The DDPG algorithm is known to be highly sensitive to training hyperparameters [117–119]. Aside from the obvious parameters like the weight-copying parameter τ, the learning rate, or the batch size [118], this also includes the relative scaling of the loss functions [117]. Compared to ordinary feed-forward learning, where hyperparameters mostly optimize the generalization quality or training time, for DDPG they easily make the difference in if a training is successful or fails entirely [118].

3. Recovery of Structure Information from Scattering Patterns

The imaging of systems on a nanometer scale is of paramount importance for many branches in medical, biological, chemical, and physical sciences. The laws of wave optics demand the usage of few nanometer or even smaller wavelenths for the imaging of such structures [120], i.e. x-ray radiation. The development of x-ray diffraction imaging has enabled many discoveries funcamental to modern society and our everyday life. It led to the discovery of the structure of cholesterin, penicillin, vitamin B12, and later insulin [121], resulting in the awarding of the Nobel prize in chemistry of 1964 to Dorothy Crowfoot Hodgkin [122]. It also facilitated the postulation of the double-helix structure of DNA by James Watson and Francis Crick [123] (Nobel prize in medicine, 1962 [124]) from measurements by Rosalind Franklin [125]. A more recent example, although using cryoelectron tomography, is the imaging of the spike-protein of the SARS-CoV-2 virus [126, 127] responsible for binding to human-cell receptors [128]. A common challenge of all nanoscale imaging procedures is the small cross-section of the target systems. The discoveries of Refs. [121–124] were only possible by crystallizing the target molecules into a lattice structure, thus increasing the overall cross-section to yield sufficient, measurable scattering signals. At the same time, the high dose energies carried by x-ray photons also damage such delicate molecules, deteriorating the sample over time [129, 130]. Consequently, crystals that are too small or do noit crystallize well to begin with, cannot be investigates with continuous x-ray sources. Examples would be nanoclusters [93] or membrane proteins [130], respectively.

Yet, The deterioration of the sample impacting the scattering signal can be avoided entirely if the scattering occurs on a much shorter timescale than the effects of the damaging unfold [131]. The use of short-pulse sources, outrunning microscopical processes, also allows the imaging of unsupported particles in free space, but also require much higher peak-intensities. Under these conditions, the damaging of the particle is much more severe in the form of a total ionization and subsequent destruction in a Coulomb explosion [132]. Such short, femtosecond-length high-intensity x-ray pulses are produced by free electron lasers (FELs) [133, 134]. Because the object features and the probing wavelength are still of comparable magnitude, the resulting image is dominated by scattering features. As a result, the underlying real-space structure can needs to be recovered by further processing [133]. To date, improvements in object reconstruction allowed the investigation of ever smaller unsupported nanosystems such as viruses [135–137], small protein crystals [129, 130], helium droplets [84, 138, 139], rare-gas clusters [140], or metallic nanoparticles [93, 141].

The scattering patterns captured in x-ray scattering experiments encode information about both geometric shape [133] and internal structure of an object [136, 142]. In first Born approximation the scattering into the far-field is calculated by the integral transformation

$$\boldsymbol{E}(\boldsymbol{q}) \sim \boldsymbol{E}_0 \int \rho(\boldsymbol{r})\, \mathrm{e}^{\mathrm{i}\,\boldsymbol{q}\cdot\boldsymbol{r}}\, \mathrm{d}\boldsymbol{r} \tag{3.1}$$

of the object's electrical density $\rho(\boldsymbol{r})$, with the transfer momentum $\boldsymbol{q} = \boldsymbol{k}_{\text{out}} - \boldsymbol{k}_{\text{in}}$ between the incoming $\boldsymbol{k}_{\text{in}}$ and outgoing $\boldsymbol{k}_{\text{out}}$ wave-vectors [142] (see Fig. 3.1) and assuming elastic scattering, i.e. $|\boldsymbol{k}_{\text{in}}| = |\boldsymbol{k}_{\text{out}}| \equiv k$. In the asymptotic case of large q, the scattered intensity by Eq. (3.1) is described by Porod's law $I(q) \sim q^{-4}$ [142, 143]. Thus, the scattered signal decays rapidly at higher scattering angles until vanishing under natural noise at a value of $q_{\max}$. From the definition of the wavenumber $k = 2\pi/\lambda$ follows, that $q_{\max} \sim \lambda^{-1}$. Therefore, the wide-angle scattering decreases rapidly when using smaller wavelengths. For sufficiently small wavelengths, i.e. large k, the component $\boldsymbol{q}_{\parallel}$ parallel to $\boldsymbol{k}_{\text{in}}$ of the detectable transfer momentum $\boldsymbol{q} = \boldsymbol{q}_{\perp} + \boldsymbol{q}_{\parallel}$ becomes negligible small, thereby simplifying Eq. (3.1) to the Fourier transform

$$\boldsymbol{E}(\boldsymbol{q}) \sim \boldsymbol{E}_0 \int \rho(\boldsymbol{r})\, \mathrm{e}^{\mathrm{i}\, \boldsymbol{q}_{\perp} \cdot \boldsymbol{r}}\, \mathrm{d}\boldsymbol{r} \tag{3.2}$$

$$= \boldsymbol{E}_0 \int \rho_p(x, y)\, \mathrm{e}^{\mathrm{i}\,(q_x\, x + q_y\, y)}\, \mathrm{d}x\, \mathrm{d}y = \boldsymbol{E}_0\, \mathcal{FT}\left[\rho_p(x, y)\right] \tag{3.3}$$

of the projected density $\rho_p(x, y) = \int \rho(\boldsymbol{r})\, \mathrm{d}z$ [142], also known from the Fraunhofer limit of diffraction [144]. Hard x-ray radiation with wavelengths in the order of 1 nm usually falls into this regime. The small-angle scattering formula Eq. (3.3) is invertible by simply applying the inverse Fourier transform $\mathcal{FT}^{-1}$. Although the phase-information of the scattering field is lost upon measurement there exist a range of reliable iterative phase retrieval algorithms capable of recovering the phase information from a single scattering pattern, given only few assumptions about the object [135, 145]. Despite a single scattering pattern containing only information of the projected density, the full three dimensional density can be retrieved using tomographic methods. By recording scattering patterns from various directions, a full three dimensional scattering volume is assembled by matching common lines of intersection in the scattering pattern. It is inverted via a three-dimensional phase retrieval and inverse Fourier transform [136, 137, 141].

While yielding excellent levels of details and allowing for error correction by combining multiple similar scattering patterns [141], the tomographic small-angle scattering is only applicable for reproducible objects. If the objects of interest are too dissimilar or inherently non-reproducible, all information has to be extracted from a singular scattering pattern. In this situation additional three-dimensional information can be extracted from wide-angle reflexes of Eq. 3.1 [146], which requires the use of longer wavelengths, i.e. soft x-ray radiation with $\lambda \sim 10$ nm. Under the assumption of a uniform density $\rho(\boldsymbol{r})$, the full scattering integral Eq. (3.1) of a polyhedral object is solved analytically by decomposition into simplex objects[1] γ. With the set $V(\gamma)$ of vertices P, the scattering integral becomes

$$\boldsymbol{E}(\boldsymbol{q}) \sim \boldsymbol{E}_0 \int_{\gamma} \mathrm{e}^{\mathrm{i}\, \boldsymbol{q} \cdot \boldsymbol{x}}\, \mathrm{d}\boldsymbol{x} = \mathrm{i} \sum_{P \in V(\gamma)} \frac{|\det(\boldsymbol{P}\boldsymbol{P'} : P' \in N(P))|}{\prod_{P' \in N(P)} \boldsymbol{P}\boldsymbol{P'} \cdot \boldsymbol{q}}\, \mathrm{e}^{\mathrm{i}\, \boldsymbol{P} \cdot \boldsymbol{q}} \tag{3.4}$$

wherein $N(P) = \{P' : P' \in V(\gamma) \wedge P' \neq P\}$ is the set of vertices $V(\gamma)$ without P itself, derived in Ref. [147] from Ref. [148]. Using Eq. (3.4) the wide-angle scattering from a tetrahedron is plotted in Fig. 3.1. It highlights the additional three dimensional information carried by wide-angle contributions through the strong reflexes originating from the tetrahedron faces. In fact, the scattering profile in the vicinity of high tranfer momenta is equivalent to small-angle scattering along a tilted object plane, as shown in Ref. [93]. Utilizing Eq. (3.4), recent theoretical works indicate in principle the completeness

[1] A simplex is a solid of $n + 1$ vertices in an n-dimensional real vector space.

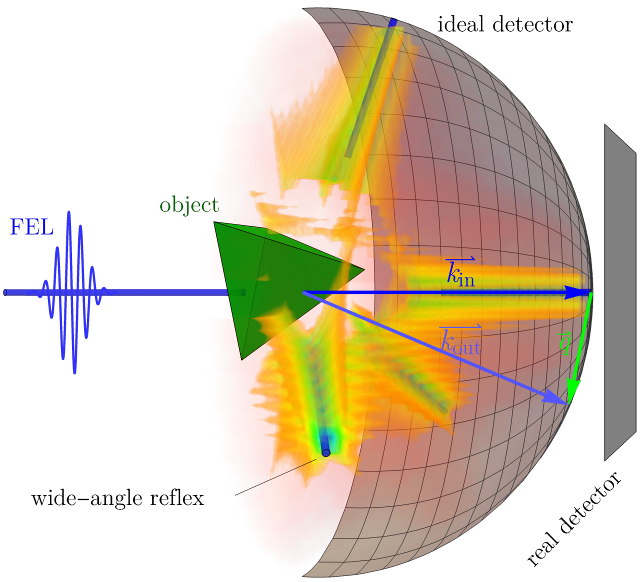

ideal detector
object
FEL
$\overrightarrow{k_{\text{in}}}$
$\overrightarrow{k_{\text{out}}}$
$\vec{q}$
wide−angle reflex
real detector

Figure 3.1: The wide-angle scattering reflexes at large transfer momenta $\boldsymbol{q} = \boldsymbol{k}_{\text{out}} - \boldsymbol{k}_{\text{in}}$ contain additional three-dimensional information on the object shape. The scattering around a momentum $\boldsymbol{q}$ contains information of the object projected to a tilted plane parallel to $\boldsymbol{q}$, as demonstrated in Ref. [93]. This fact is highlights by the strongest wide-angle reflexes appearing perpendicular to the triangular faces of the tetrahedron.

of such three-dimensional information encoded in wide-angle scattering signals [147, 149, 150] for solid convex objects in the first Born approximation and infinitely many exact measurements.

Despite transporting much more information, the reconstruction problem of finding $\rho(\boldsymbol{r})$ from the full scattering integral Eq. (3.1) lacks a direct inversion formula [93, 139, 146]. Further, the optical properties of the material have a much stronger impact in wide-angle scattering through absorption and repeated scattering, rendering the first Born approximation insufficient. Moreover, a real detector is always of finite size and can never fulfill the requirement of perfect measurement and the aforementioned non-reproducibility further hinders the independent acquisition of additional shape information using alternative experimental techniques. Further, under certain experimental conditions, especially in the case of objects very short along the optical axis,[2] the solution to this problem may not be unique [146, 151, 152]. Thus far, the reconstructions of wide-angle scattering patterns mostly rely on iterative forward fitting methods, which are based on repeated simulations of the scattering process of a suitably parametrized object model [84, 93, 139]. While highly successful and reliable, the repeated scattering simulations are computationally expensive. Further, the requirement of a parametrized object model in part pre-determines the outcome of a reconstruction and may restrict the available parameter space too. Moreover, the high variety in object shapes of some experiments even require a coarse shape identification to use different parameter models.

The idea for this part of this thesis was inspired by a talk on the reconstructions of silver nanoclusters from wide-angle scattering pattern in Ref. [93], given in 2017. Due to highly diverse shapes of clusters observed in the experiment, each scattering pattern had

[2] In the limit of a completely two-dimensional object, the full scattering formula Eq. (3.1) simplifies into the small-angle case of Eq. (3.3), thus containing no further information even at large scattering angles.

to be associated manually by a human scientist to a corresponding shape and only size and orientation could be reconstructed using automated algorithms. On the other hand, the identification of image contents, i.e. classification, is a prime task for neural networks. At the time, neural networks applications were just emerging in physics, which then changed rapidly over the course of the next five years, recall Sec. 1. In diffractive imaging, neural networks architectures haven been explored for applications in both small-angle and wide-angle scattering. A first step was automating the pre-sorting of the raw wide-angle scattering data into blank pulses and actual scattering events and further subdivinding the latter into different categories of patterns in Ref. [84]. In a follow-up publication, a generative neural network was trained on creating artificial experimental scattering patterns, allowing conclusions on the most prevalent features inside the dataset and characteristics of the detector [85]. On the data of small angle experiments the transitions between different object geometries could be observed in the latent space of an autoencoder, trained without any knowledge on the exact object shape [92]. Furthermore, direct inversion of the small angle scattering Eq. (3.3) has been tackled with neural networks. Although reliable, classical phase retrieval algorithms are a bottleneck in the reconstruction of small-angle scattering patterns not only by computation time but often by the requirement of very detailed input information on the object shape. Both phase retrieval and direct density reconstruction of purely binary two-dimensional object densities with convolutional neural networks was successfully demonstrated in Ref. [86]. This has further been extended to the reconstruction of discretized three-dimensional objects models from three-dimensional Fourier volumes to [87–90]. Almost all of these works are trained an tested solely on perfect, simulated data. Just recently, benefiting from the inherent error correction of three-dimensional Fourier volume construction from large sets of small-angle diffraction patterns, such a neural network was used to successfully to reconstruct an object from experimental data [90]. Also, neural networks have been tried for denoising and error corrections on experimental scattering patterns [91].

The goal of this work is to improve and speed up the reconstruction of wide-angle scattering patterns with the help of neural networks. Due to the lack of readily reconstructed experimental data and the inherent uncertainty, the neural network needs to be trained on simulated scattering data. Simultaneously, the objective for it is to perform experimental data that are subject to a wide range of defects. Because it was inspired by the shape identification problem in Ref. [93], that also provides a set of experimental data reconstructed by classical algorithms, it is developed along the specific case of silver nanoclusters illuminated by the FLASH FEL. The specifications of the experiments are introduced further in Sec. 3.1. The first step in improving the reconstruction of wide-angle scattering patterns with neural networks is matching the capabilities of existing, classical algorithms and is presented in Sec. 3.2. The feasibility of the initial idea of shape identification through a neural network classification is analyzed in Sec. 3.3. It ultimately fails due to the oversimplification of the object-model. Instead, the object density is reconstructed as a full, three-dimensional model in Sec. 3.4, achieving unprecedented levels of details. This is achieved by training on directly on the laws of scattering physics. It leads to the discovery of an entirely new possible shape for silver nanoclusters and the further deployment in experiments and further development is discussed in Sec. 3.5. Yet, while developed along the example of Ref. [93], the techniques presented in this chapter are not limited to this specific problem but rather can be transferred to any wide-angle scattering reconstruction problem, given a sufficient scattering simulation algorithm for data generation and eventually training, if so desired.

3.1. Single-Shot Wide-Angle X-ray Scattering with Silver Nanoclusters

The scattering setup investigated throughout this chapter is that of silver nanoclusters illuminated with soft x-rays from the FLASH free electron laser, as discussed in Ref. [93]. The silver metal clusters for the experiment are produced in a modified magnetron sputtering source (for details see Ref. [153]), that ejects a beam of condensated metal clusters. The freshly grown clusters are in transient states and have not relaxed into an equilibrium. Hence, metastable states are expected that allow insights on the dynamics involved in the cluster formation process. Further, no two particles may be identical due to the inherent randomness of the condensation process. Consequently, a single-shot imaging method is required.

For imaging, the cluster beam is injected into the beamline of the FLASH free electron laser at DESY and illuminated with soft x-ray[3] pulses of wavelength $\lambda = 13.5\,\mathrm{nm}$ and durations of $\approx 100\,\mathrm{fs}$ [93]. With the observed cluster sizes between 50 and 400 nm the scattering can be regarded as in the wide-angle limit. The scattering signal was recorded using a composite detector. The x-ray photons were detected using multichannel plate (MCP) and converted to optical photons on a phosphorous screen, that in turn are reflected by a mirror towards a CCD camera outside the vacuum chamber [154]. For protection from the harmful intensity of the direct x-ray beam, the MCP detector has a central hole (generally called beam-stop). The detector microchannels are tilted, resulting in a decreased sensitivity at a scattering angle of 8° (sometimes called the 8°-hole) [142]. Lacking a direct inversion algorithm, the objects corresponding to the obtained scattering patterns were reconstructed by forward fitting (see Sec. 3.1.2 below) using a numerical scattering simulation algorithm (see Sec. 3.1.1 below). From reconstructing 100 out of a total of 25 000 scattering patterns recorded, the authors in Ref. [93] report the finding of truncated octahedra, decahedra, icosahedra and truncated twinned tetrahedra. Aside from the truncated octahedra, neither of these structures are deemed stable for the observed particle shapes. The reconstructed scattering patterns published in Ref. [93] can be assumed as accepted object candidates to the experimental scattering patterns. Therefore, they are utilized as a test bed of real experimental data for any new reconstruction method. For use in this thesis, the images from Ref. [93] are extracted and re-scaled. The images are zero-padded to include the full 2π half-sphere into which wide-angle scattering occurs and dimensionally rescaled to an image size of 128×128 pixel, in accordance to the data format described later in Sec. 3.1.2. The resulting images and reconstructions from Ref. [93] are listed in Fig. 3.2.

The intensity profiles recorded in experiments like Ref. [93] or obtained by the simulation algorithm are monochromatic, single channel images, i.e. either sensitive to or assuming a single wavelength. For better visibility, scattering patterns depicted in the figures throughout this thesis are scaled logarithmically and converted to false color images. Still, internally, all data is stored and processed as single-channel grayscale images. A comparison between a grayscale and false-color version of the same scattering pattern is given in Fig. 3.3 with a bar-legend for the SunsetColors-colormap used throughout this chapter.

[3] Sometimes also classified as XUV [142] or EUV.

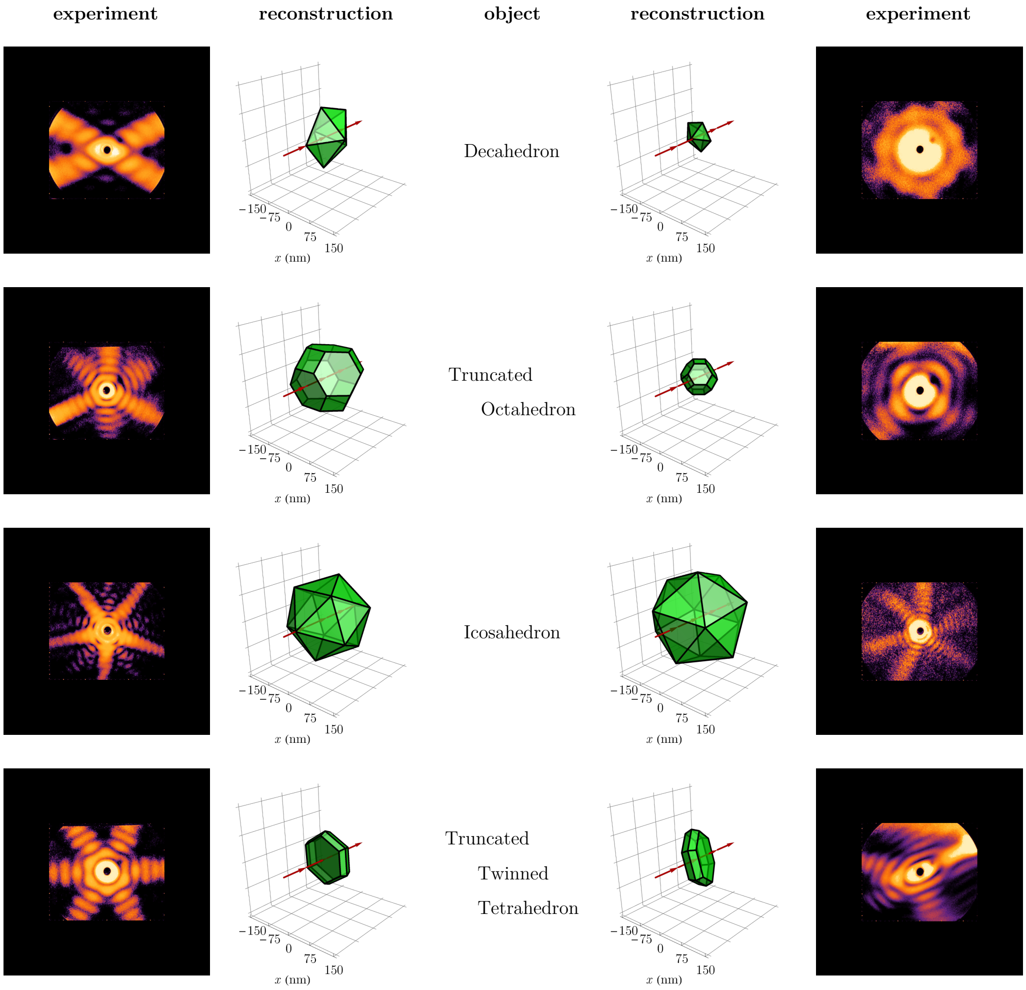


Figure 3.2: The experimental scattering patterns obtained by Ref. [93] (permitted by Creative Commons CC-BY 4.0 license (http://creativecommons.org/licenses/by/4.0/)) are re-scaled to conform with the convention and used in testing the neural networks developed in this thesis. The corresponding object reconstruction from Ref. [93] are assumed as established object candidates, that any newly developed reconstruction algorithm nedds to either match or achieve better reproduction of the scattering pattern.

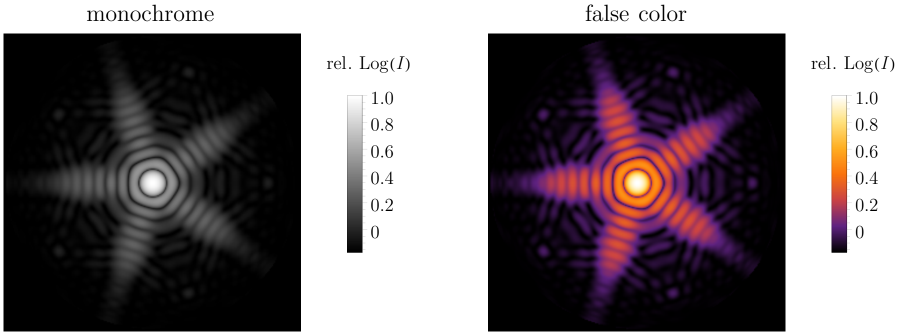


Figure 3.3: Comparison between grayscale (left) and false-color (right) version of the same simulated scattering pattern of an icosahedron (see Fig. 3.4). The colormap used are the SunsetColors from Mathematica.

3.1.1. Numerical Scattering Simulation

Analyzing the large datasets obtained from scattering experiments requires a scattering simulation algorithm that does not only yield precise result but is also fast to compute. This usually means a trade-off between neglecting higher order physical effects and evaluation time. The most basic theory for elastic scattering theory with plane incident wave is the first Born approximation, neglecting all higher order effects such as absorption, the finite speed of light, and repeated scattering. For silver nanoclusters, the first Born approximation with further including an effective absorption model already yields sufficient results. This is shown in Refs. [93, 142, 155, 156] though extensive comparisons with more elaborate methods.

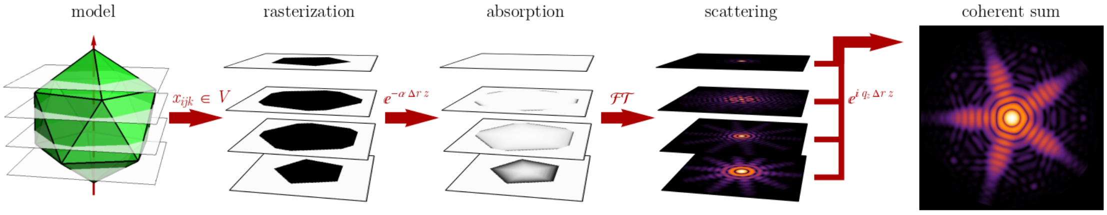


Figure 3.4: Visualization of the numerical multi-slice Fourier transform (MSFT) simulation scheme with absorption for wide-angle scattering after the definition in Ref. [93]. Analytical object models need to be rasterized on a three-dimensional grid into a discretized object density. Absorption is treated in an effective theory through an exponential decay with absorption length a_{abs} of the bulk material to obtain effective visible densities. The elastic scattering into the far field is decomposed into a Fourier transform and subsequent wide-angle phase modulation depending on the q_z-component of the transfer momentum $\boldsymbol{q}$. The complete scattering pattern is obtained through the coherent sum of all scattering slices.

In the following, the optical axis is fixed to the z-axis. Due to the large number of atoms (typically on the order of 10^9), the silver nanoparticles can be assumed to be macroscopic dielectric bodies, that are well described by a binary permittivity function $\epsilon_r(\mathbf{r}) = \epsilon_{\text{silver}}$ for $\mathbf{r} \in V_{\text{object}}$, and $\epsilon_r = 1$ otherwise [93]. Absorption is treated in an effective theory through an exponential decay

$$\rho_{\text{eff}}(\boldsymbol{r}) = \rho(\boldsymbol{r})\,\mathrm{e}^{-a_{\text{abs}}\,\Delta z} \tag{3.5}$$

with absorption length $a_{\text{abs}} = 12.5\,\text{nm}$ of bulk silver to simulate the effective illuminated object density ρ_{eff} at any depth Δz from the surface. At the present cluster sizes of $40\ldots320\,\text{nm}$, the short absorption length of silver means that a considerable region of the object does not receive enough incident radiation (see Fig. 3.5) to add a mearurable contribution to the scattering pattern. Consequently, some objects are indistinguishable from their scattering patterns. The numerical scattering simulation is performed in the multi-slice Fourier transform (MSFT) scheme (see Fig. 3.4) with a discretized optical density. The optical density is rasterized on a cubic grid,

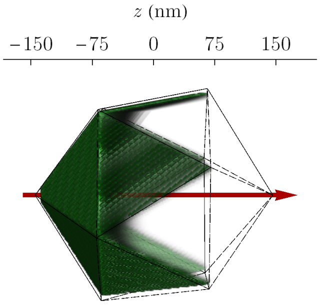


Figure 3.5: The absorption length of bulk silver of 12.5 nm is much smaller than the present particle diameters. Consequently, a relevant intensity penetrates only a few slices deep in the discretized particle model. In this case, only the front-faces of the icosahedron are illuminated enough to contribute a measured signal while the far-side lies within a shadow-region.

the simulation area, with length l and grid points N along each coordinate axis. The contribution to the far-field scattering pattern of each effective density slice along the z-axis is calculated individually. The calculation,

$$\boldsymbol{E}_j(q_x, q_y) = \mathcal{FT}[\rho_{\text{eff},j}]_{q_x,q_y}\, \mathrm{e}^{\mathrm{i}\, q_z(q_x,q_y)\, \frac{l}{N}\, j}\,, \tag{3.6}$$

decomposes into a Fourier transform $\mathcal{FT}$ and wide-angle phase factor depending on the q_z-component of the transfer momentum $\boldsymbol{q}$. In order to increase the interpolation density, the lateral dimensions of the object tensor (and thereby the simulation region) is usually expanded by zero padding (calculation resolution) before Fourier transform. The region of $\boldsymbol{E}_j(q_x, q_y)$ relevant for elastic forward fitting is that where the condition $q_x^2 + q_y^2 + q_z^2 = k^2$ is fulfilled, with wave number $k = 2\pi/\lambda$ of the incident beam.

All simulations within this thesis are performed with object resolutions of $192\times192\times192$, simulation area length $l = 318.75\,\text{nm}$ and calculation resolutions of 512×512, resulting in a physically relevant region of 128×128 pixels for elastic forward scattering. If not stated otherwise, all calculations are performed with these resolutions. Figures throughout this thesis are simulated with twice the calculation resolution, increasing the interpolation density. In general, we demand all objects to be convex, as this a necessary condition for the existence of a unique solution in the ideal scenario of informationally complete measurements [149].

3.1.2. Scattering Reconstruction by Forward Fitting

Due to the lack of a direct inversion algorithm the reconstruction of the real-space object creating a certain scattering pattern classically has to be achieved by comparative methods, like in Refs. [84, 93, 155]. This usually means an iterative forward fit with a parametrized object model and some initial assumptions. From there, the object is manipulated slightly and the corresponding scattering pattern compared to the input. The goal of each iteration step is to find a set of parameters that achieves a closer fit to the input scattering pattern than the previous and the cycle is repeated until a convergence is achieved. However, the dependency of the scattering function not only on the discretized object model used in MSFT but any set of model parameters is too complex for using gradient based methods. Instead, a Monte-Carlo simplex algorithm is utilized in Refs. [84, 93, 155]. This stochastic method involves generating random points inside a simplex in parameter space around the current values, simulating the corresponding scattering patterns, and advancing the parameter set as soon as a new-best fit is found until convergence is achieved. Due to the inherent randomness reconstructing a single scattering pattern requires performing several Monte-Carlo runs,from different starting parameters. In total, in the order of 10^5 scattering simulations are needed for one such reconstruction, which in itself is computationally expensive. While the MSFT algorithm is considerably faster than e.g. FDTD simulations, it still is too slow to match the repetition rate of FEL experiments, allowing only reconstruction of a fraction of experimental dataset. Also, creating the object model and especially in the case of Ref. [93] the initial assumption requires intervention of a human scientist. Yet, it is an extremely reliable reconstruction method and the inherent model restrictions guarantee finding the object, under the given assumptions, that best reproduces the scattering patterns [84, 93, 155]. Therefore, the predictions obtained by the MCS MSFT are used as references in testing the neural networks developed in this thesis. Also, in Sec. 3.2.5, the reconstruction algorithm from Ref. [84] is matched up directly against a neural network, solving the same parameter reconstruction task.

3.2. Parameter Reconstruction for Icosahedral Nanoclusters

The first step towards the goal of improving the reconstruction of nanoclusters from single-shot wide-angle scattering patterns with neural networks is to replicate the capabilities of existing reconstruction techniques. For the silver nanoclusters introduced in Sec. 3.1 this means determining the size and orientation of the particle, after the shape has already been identified manually by a scientist. The main results documented in this section are published in Ref. [2].

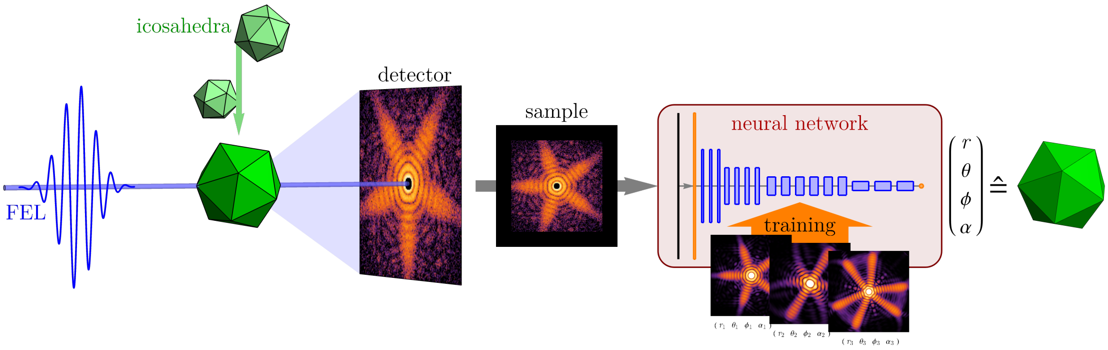

icosahedra
detector
sample
neural network
training
FEL
$\begin{pmatrix} r \\ \theta \\ \phi \\ \alpha \end{pmatrix} \widehat{=}$
$(r_1\ \theta_1\ \phi_1\ \alpha_1)$
$(r_2\ \theta_2\ \phi_2\ \alpha_2)$
$(r_3\ \theta_3\ \phi_3\ \alpha_3)$

Figure 3.6: The icosahedral nanoclusters of varying size and orientation are illuminated with short-length soft X-ray pulses from a FEL. The resulting scattering patterns are of varying quality due to noise, missing regions and further defects. The neural network is trained to reconstruct the size r and orientation parameters $\{\theta, \phi, \alpha\}$ corresponding to the original object. Training is performed with pairs of parameters and simulated scattering data.

The neural network has to reconstruct size and orientation of a silver nanoparticle of known shape from the corresponding wide-angle scattering pattern, as sketched out in Fig. 3.6. The geometry of choice is the icosahedron (see Fig. 3.6, which is one of the shapes observed in Ref. [93], to which Sec. 3.2.1 is dedicated. The symmetry of the particle has implications for the representation of rotations, which are elaborated in Sec. 3.2.2. For real-world application, the neural networks needs to be capable of processing experimental data, like the one obtained in Ref. [93]. However, the existing set of reconstructed scattering patterns currently available is too small for the training of a neural network. Instead, simulated scattering patterns have to be utilized in training, which can be generated using the MSFT algorithm also used in classical reconstructions [84, 93, 142, 155] (see Sec. 3.1 for details). Simulated data holds the additional advantage of the parameters associated to a scattering pattern being precise. In contrast, the ground truth parameter set (or even the exact shape) for an experimental pattern can never be known for sure, as the particle is destroyed by the FEL pulse. Yet, simulated scattering patterns are clearly distinguishable from experimental ones, as they are free of any defects. To ensure reliable predictions even in the presence of heavy image artifacts, data augmentation techniques are used, which are introduced in Sec. 3.2.3.

The adaptions to the design of the neural network and training process leading to reliable and reproducible results are documented in Sec. 3.2.4. Neural networks may offer a significant time advantage through their nearly instantaneous evaluation time compared to scattering simulations. This is further investigated in Sec. 3.2.5. Finally, the prediction capabilities on experimental scattering patterns are demonstrated in Sec. 3.2.6.

3.2.1. The Icosahedron

The icosahedron (see Fig. 3.7) is a regular polyhedron composed of 20 equilateral triangles. It is one of the Platonic solids, which are the five regular solids assembled from regular polygons, namely the tetrahedron (4 triangles), cube (6 squares), octahedron (8 triangles), dodecahedron (12 pentagons) and icosahedron (20 triangles). They were already known in ancient Greece, and are named after Plato who assigned them to the four elements[4] in his *Timaeus* [157]. Due to the universal symmetry, the Platonic solids are commonly used as playing dice [158, 159], most prominently the cube [160].

The icosahedron is chosen due to its widespread appearance in nanometer-sized objects. Many viruses have an icosahedral capsid, such as the herpes simplex virus [161], the macroscopic mimivirus [135, 136, 162], some bacteriophage [137, 163], and all Adenoviruses[5] [165]. These viruses build their shells from the repetition of a single symmetric protein, thus simplifying their genome. With this constraint, the icosahedron is optimal in offering the most volume with the simplest building blocks (triangles) for the shell [161]. On a smaller scale, also organelles of some bacteria are of icosahedral shape [166]. Clusters of rare-gas molecules, interacting only by van-der-Waals forces, are also known to form into icosahedra [167] as they again minimize the surface to volume ratio for close-sphere packing. Most importantly, silver clusters are also observed for silver nanoclusters of varying sizes [93, 168, 169] and they are part of the set of shapes recorded in Ref. [93], allowing for testing the neural network against classical reconstruction methods.

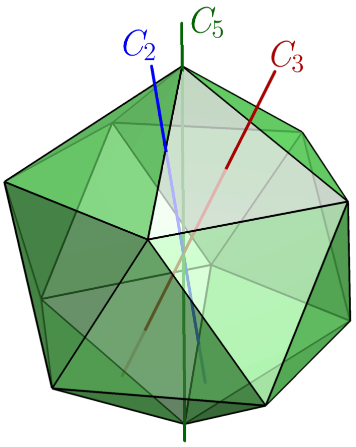


Figure 3.7: The icosahedron is formed by 20 equilateral triangles, with 5 triangles meeting at every vertex. It has five- (green), three- (red) and two-fold (blue) rotation symmetries around the vertices, faces and edges, respectively. One example of each is shown with the corresponding cyclic group symbol C_n for n-fold rotation symmetries.

As a regular solid, the icosahedron is highly symmetric. It is name-giving to the corresponding symmetry group I_h, the icosahedron group, containing 120 elements. This group can be decomposed into the group-product $I_h = i \otimes I$ of the point inversion group i and the icosahedral rotation group I with 60 elements. These 60 symmetry rotations are formed by

- 6 five-fold rotation axes C_5 through each pair of opposing vertices,
- 10 three-fold rotation axes C_3 through the centers of opposing triangle faces
- 15 two-fold rotation axes C_2 through the centers of opposing edges,

of which one is shown in Fig. 3.7, each.

The symmetry properties of the icosahedron pose an additional challenge to the recon-

[4] Plato assigned the tetrahedron to fire, the cube to earth, the octahedron to air, and the icosahedron to water. The remaining dodecahedron is assigned to the universe as a whole [157].

[5] The family of Adenoviruses recently became familiar to the broader public due to the application as delivery vectors in Covid-19 vaccines such as Vaxzevria (Astra Zeneca), Janssen COVID-19 vaccine (Johnson & Johnson) and Sputnik V [164]

struction task. On a playing dice each side is unique due to an individual number carved in or printed on. In contrast, a perfectly isotropic nanocluster does not have such identifying features. As a result, for any orientation of a blank icosahedron there exist 59 symmetry rotations[6] resulting in the same shape, although internally the individual (indistinguishable) atoms may change places. This inherent symmetry needs to be accounted for in the representation of icosahedral rotations.

3.2.2. Rotation Representations for Polyhedral Clusters

The orientation of any rigid body in space can universally be defined by a rotation from a reference orientation. In the case of an icosahedron, the reference is usually a point-up orientation with one five-fold symmetry axis parallel to the z-axis. Due to the symmetry of the icosahedron, any orientation can be reached through 60 different rotations from the reference. This high degree of symmetry is advantageous for the reconstruction task, as it reduces the overall complexity of possible orientations. Yet, it also introduces a challenge to any parameter representation. The mapping from scattering patterns in non-injective, because any object orientation (and in turn scattering pattern) can be assigned to 60 different sets of parameters.[7] However, the degeneracy can be lifted by restricting rotations into a subset, where every orientation is associated with just one set of rotation parameters, the so-called fundamental domain. Any set of rotation parameters may be projected onto one element of the fundamental domain by executing the nearest inverse symmetry operation from I.

The Lie-algebra so(3) of the three dimensional rotation group SO(3) has three basis elements and consequently any rotation in the R^3 may be represented with just three parameters. One convention for the representation of rotations are the Euler-angles. They are a set of three rotation angles about fixed axes, which in combination can represent any rotation of an object. Depending on the specific convention, these axes can be fixed to either the rest-frame (extrinsic), the object itself (intrinsic) or less commonly a combination of both. One example are the three rotation angles pitch, yaw and roll of an airplane, which are directly accessible through the movable aerodynamic surfaces. Euler angles are also used in Ref. [93] in reconstructing rotation parameters of nanoclusters, with the third rotation angle aligned to the beam direction. This choice simplifies the optimization task, as the third rotation can be accessed through rotation of the scattering pattern without the need of re-simulating the scattering process. However, Euler-angles have two major drawbacks: The appearance of singularities (gimbal-lock) and non-trivial combination of successive rotations into one single rotation.

Another representation of rotations tracing back to Leonard Euler is the axis-angle representation. It arises from Euler's rotation theorem, stating that any rotation of a rigid body may be executed by a single rotation about one axis. Expressed by a unit vector $\boldsymbol{n}$ and angle φ, the rotation of any vector can be calculated using the Rodrigues' rotation formula [170] (see App. B.1 for details) or transformation into a rotation matrix. Again, the combination of two successive rotations into one single rotation is not trivial in the axis angle representation. It is, however, directly related to another representation offering exactly this feature: the rotation quaternions.

[6] Not counting the identity rotation.

[7] A similar problem arises when inverting the equation $\sin(x) = y$, where an infinte number of solutions exist for any $|y| \leq 1$. Per convention, the arcsin-function as the inverse function of the sin is defined for the period of the smallest x around the origin, which is defined as the fundamental domain.

Quaternions are a four-dimensional extension of the imaginary numbers by additional elements j, k introduced by Sir William Rowan Hamilton[8] with the constituting relation

$$\mathrm{i}^2 = \mathrm{j}^2 = \mathrm{k}^2 = \mathrm{i}\,\mathrm{j}\,\mathrm{k} = -1\,. \tag{3.7}$$

A short introduction to quaternions can be found in App. B.1. Any quaternion $\mathtt{x}$ may be expressed by four real-valued parameters $\{x_0, x_1, x_2, x_3\}$ through

$$\mathtt{x} = x_0 + x_1\,\mathrm{i} + x_2\,\mathrm{j} + x_3\,\mathrm{k}\,, \tag{3.8}$$

and the imaginary quaternions (with real part $x_0 = 0$) are isomorphic to the $\mathbb{R}^3$ in the same way as the complex numbers $\mathbb{C}$ are isomorphic to $\mathbb{R}^2$. This is the very property that led to their discovery and motivates the vector notion of quaternions

$$\mathtt{x} = (x_0, \boldsymbol{x}) \quad \text{with} \quad \boldsymbol{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}\,. \tag{3.9}$$

Further, the rotation of a vector, represented by an imaginary quaternion, is facilitated through the quaternion product with a unit quaternion (see appendix B.1 for details). The corresponding rotation quaternion is calculated from the axis-angle representation through

$$\mathtt{q}_{\mathrm{rot}} = \left(\cos\left(\frac{\varphi}{2}\right),\ \sin\left(\frac{\varphi}{2}\right)\boldsymbol{n}\right)\,. \tag{3.10}$$

In quaternion space, two consecutive rotations are consolidated into a single rotation quaternion through the quaternion product of both. The sin-function in the imaginary part of eq. (3.10) highlights the fact, that any rotation angle φ larger than π (sin changes sign) is equivalent to a rotation of $\varphi - \pi$ in reverse direction or about the negative axis $-\boldsymbol{n}$. However, the rotation is still uniquely identified through the real part of $\mathtt{q}_{\mathrm{rot}}$. Even if restricting rotation angles to $0 \leq \varphi < \pi$ all possible rotations of a rigid body still can be realized and the rotation quaternion is fully determined by the imaginary part alone. This allows to map rotations onto points inside a three dimensional unit-sphere in imaginary quaternion space.

The fundamental domain of the rotations of an icosahedron in quaternion space is described in Ref. [171]. Through calculating the shortest rotation angle to the nearest symmetry operation (or primordial point in Ref. [171]) from the icosahedral rotation group I, each rotation inside the unit-quaternion space can be assigned to a proximity domain, tiling the unit-quaternion space into 60 cells. The fundamental domain is the cell centered around the identity-rotation. When visualized in imaginary quaternion space, it takes on the form of a dodecahedron, as depicted in Fig. 3.8. The dodecahedron faces intersect the axis to the nearest five-fold symmetry rotations (green dots in Fig. 3.8(a)) half-way at a distance of $\sin(\pi/10)$ [171].

Inside the unit-sphere in imaginary quaternion space, similar rotations are mapped close to each other, allowing for a metric of similarity between rotations by the quaternion norm (see App. B.1). Rotations about symmetry axes form straight lines (as in Fig. 3.8(b)), intersecting several cells. When restricting to the fundamental domain, lines reaching the fringe of the domain wrap to the opposite end of the domain and again continue straight.

Due to the favorable conditions of the quaternion space, any rotation $\mathtt{q}_{\mathrm{rot}}$ is easily

[8] Most prominently known for introducing the mechanics formalism named after him.

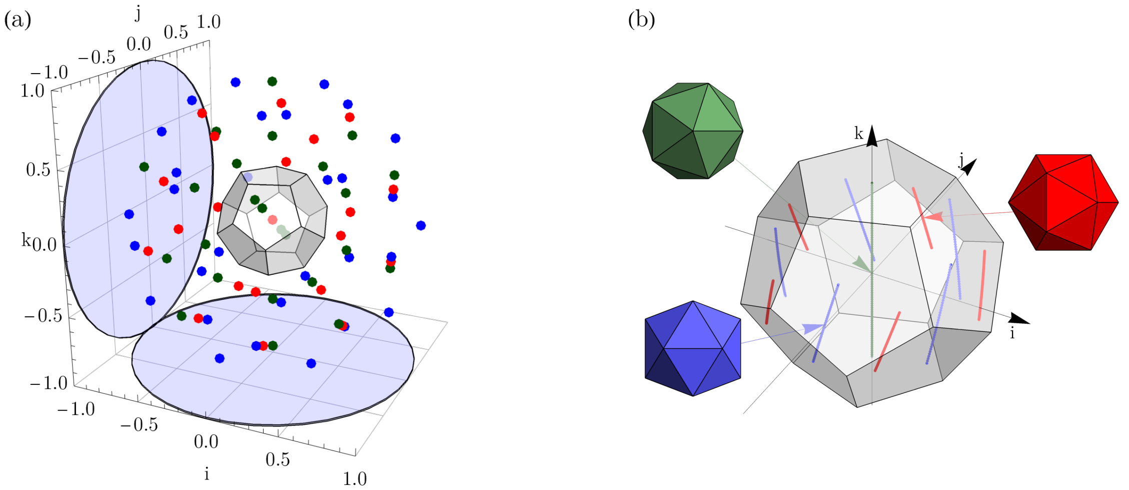


Figure 3.8: In imaginary quaternion space (a) each point inside the unit sphere translates to a rotation. The dots indicate all symmetry rotations of the icosahedron in green (five-fold), red (three-fold) and blue (two-fold). The fundamental domain is of dodecahedron shape with the faces intersecting the axis to the nearest five-fold symmetry rotations at half-angle $\sin(\pi/10)$. Inside the fundamental domain (b) similar rotations are mapped to points close to each other, indicated by lines for rotations around the five-fold axis (green), three-fold axis (red) and two-fold axis (blue). At the fringe of the fundamental domain, the lines wrap to the opposite end of the domain and continue straight until reaching the next fringe.

projected into the fundamental domain, following two steps:

1. determine the closest symmetry rotation $\mathfrak{q}_{\text{symm}}$,
2. apply the quaternion product with the inverse symmetry rotation $\mathfrak{q}'_{\text{rot}} = \mathfrak{q}_{\text{rot}}\,\mathfrak{q}_{\text{symm}}^{-1}$.

For practical application, is is usually not required to know which of the symmetry operation is the closest. Hence, the procedure is simplified by calculating the quaternion product with all 60 symmetry rotations and then selecting the one with the smallest rotations angle through either the maximal real part or shortest imaginary part.

The fundamental domain projection generates a unique mapping between rotation parameters and icosahedron orientations. It allows for generation of a training dataset with fully randomized rotations, that are mapped into the fundamental domain in a preprocessing step before training the neural network. Yet, the very same projection could also be utilized inside a loss function, that calculates the loss for all symmetry rotations and selects the best. This approach, however, is more suited for systems with a smaller number of symmetry operations, like the one in App. B.3.

With the rotation quaternions, icosahedra are represented by five parameters: a size parameters and the four components of the rotation quaternion. Trough using all four components of the quaternion, the rotation inside the fundamental domain is over-determined. This, however, is beneficial for checking the confidence of predictions made by the neural network. If a predicted quaternion is not of unit length it is most likely erroneous.

3.2.3. Simulating Artifacts in Data Augmentation

The scattering patterns obtained in experiments show a number of deviations when compared to simulated patterns [93]. these deviation can mostly be divided between quality defects and restrictions of the experiment. The quality of the image recorded is reduced

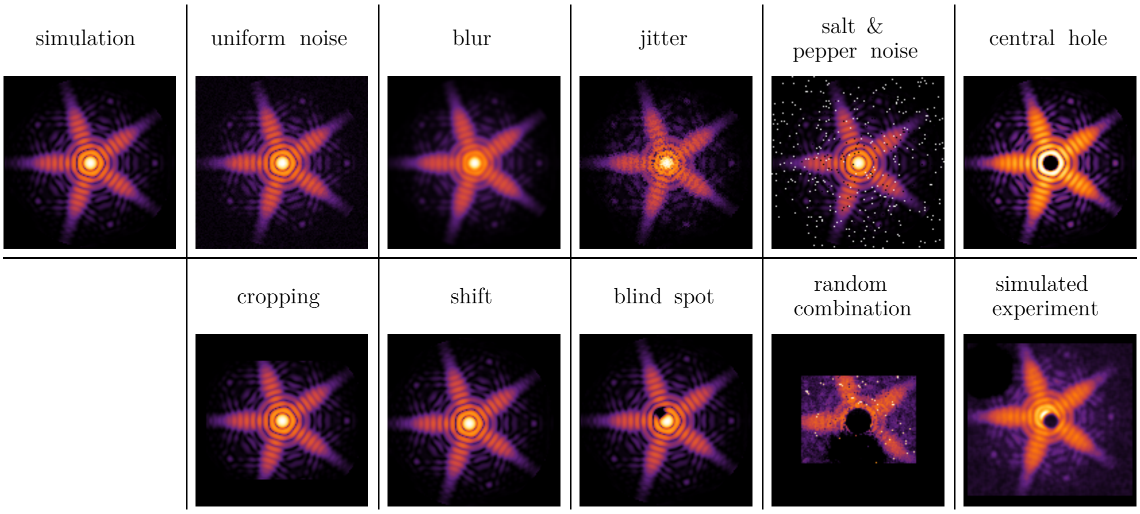


Figure 3.9: The simulated scattering patterns (top left) are modified with a variety of defects in an image augmentation step in order to both increase the robustness of the neural network against defects appearing in real experiments and increase the effective size of the dataset. The augmentation filters can be divided into quality defects such as noise or blur and sensor effects like deleting a central hole or restricting the angular range by cropping. Also random combinations of the previous effects and a special combination inspired by experiment data is used. The defect strength is randomly selected from a range largely exceeding the magnitude in real experimental data.

by noise and scattering patterns appear blurred compared to simulations. As it will become apparent in Sec. 3.4, the blurring can be attributed to deviations from the perfect icosahedral shape. Further, the shape of the detector limits the information content of the scattering patterns. The multichannel detector used in most wide-angle scattering experiments is flat, limiting the angular range available for recording. Further, it sports a central hole to let the unscattered direct beam pass, as otherwise it would damage the detector through intense radiation.

The appearance of such artifacts hold the possibility to result in patterns outside the interpolation-range of the neural network. Testing shows a relative robustness for a network trained on clean simulation data. The introduction of cropping or central hole, however, exceeds the capabilities of the trained network and results in erroneous or even outright nonsensical predictions such as the rotation quaternions not having unit-length. Hence, the regularization technique of image augmentation [7] is utilized to alter the input images upon training. In computer vision tasks, image augmentation is used to increase the generalization capability of a neural networks by shifting, rotating and partially deleting existing images or applying noise and other image defects [7, 8]. It also prevents relying on single image-features that identify whole attributes. For example, size of the icosahedra is encoded in the spacing of the fringe pattern and it could be reconstructed by focusing on a small windows of the image, which is prevented through obfuscation or outright deleting by augmentation filters.

In our case, a number of different image defects are introduced, listed in Tab. 3.1 and an example for each shown in Fig. 3.9. Most of these filters are either inspired by defects appearing in experiments (such as the detector hole) or general image distortion (like jitter). The upper limit for the magnitude of the defects are aimed to exceed the experiments by a good margin order to guarantee good prediction results on experimental data. The augmentation filters are designed to generalize types of defects. They are not tailored specifically for a specific target dataset, as it would easily be possible e.g. with the size of the hole and detector range. This is done to achieve a robustness against changes in the setting, as e.g. view angles or detector sensitivity can vary even over the course of a single measurement sequence.

Before training, an augmented dataset is created by applying each filter from the list in Tab. 3.1 to each of the scattering patterns in the training-dataset, thereby increasing the effective size 11-fold. This augmentation is performed once before training and the resulting set is kept for a complete training run.

Table 3.1: Full list and description of the augmentation filters used.

Filter Name	**Decription**
identity	Returns the input image.
uniform noise	Add uniformly sampled values between 0 and 0.1 to any pixel to simulate noise in general.
blur	Convolve the image with a Gaussian kernel of size $r \in [0, 3]$. As will later be shown, the blur can be attributed to deviations from the perfect icosahedral shape in the object.
jitter	Each pixel is replaced randomly with another pixel in a rectangle of size r around the pixel. The sampling window r is randomly chosen between 0 and 2.
salt & pepper	A fraction f of the image pixels is set to either 0 (black) or 1 (white). The fraction f is randomly chosen between 0 and 0.05. It simulates defect detector units.
central hole	The pixels inside a circle of radius r around the center of the image are set to 0, with $r \in [6, 20]$. It simulates the central hole in the detector to let the direct x-ray beam pass without damaging the detector.
cropping	Crop the image in both height and with to a length of down to 64 pixels, individually, and fill the missing regions with zeros. It recreates the limited angular range of experimental setups.
shift	Shift every pixel by an random value between -5 and 5 in both horizontal and vertical direction. Pixels mapped to regions outside the image are discarded and missing regions filled with zeros.
blind spot	Multiply every pixel of the image with a Gaussian defect function $1 - \exp\{-((x - x_0)^2 + (y - y_0)^2)/(2\sigma^2)\}$ around a random point $(x_0, y_0)^T$ and Variance between 0 and 0.2. It simulates an uneven sensitivity of the detector pixels and also the detector blindspot, appearing in MCP detectors as in Ref. [93] due to the tilting angle of the channels.
random combination	Randomly choose between 2 and 5 different filters from the above list (without identity) and apply them consecutively.
simulated experiment	In order, apply the filters **shift**, **central hole**, **uniform noise**, **blur**, **blind spot** and **cropping**. It combines the individual defects observed in experimental scattering patterns from Ref. [93].

3.2.4. Network Design for the Reconstruction of Icosahedra

We have established the data format for the reconstruction of icosahedral nanoclusters from scattering patterns. The icosahedra are represented by a set of five parameters: the four components of the rotation quaternion q of the orientation relative to the reference orientation and one size parameter. Because all vertices of the icosahedron lie on the surface of a sphere around the center of the icosahedron, the radius R of this circumsphere is used as the size parameter. Hence, any icosahedron is uniquely represented by a parameter vector

$$\boldsymbol{v} = (q_0, q_1, q_2, q_3, R)^T \,. \tag{3.11}$$

These five components are of different character, each. The imaginary parts q_1, q_2, q_3 are limited to $\sin(\pi/10) \approx 0.31$ and the real part q_0 lies between $\cos(\pi/10) \approx 0.95$ and 1.0 due to the restriction to the fundamental domain. The radius R, on the other hand, is a dimensional parameters and clusters sizes in a range of $30\,\text{nm} \leq R \leq 160\,\text{nm}$ are considered. To avoid unbalanced learning due to the uneven scaling between the parameters, both the real part of the quaternion and the size are re-scaled by

$$\boldsymbol{v} = \begin{pmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \\ R \end{pmatrix} \rightarrow \boldsymbol{v}' = \begin{pmatrix} 10\,(1 - q_0) \\ q_1 \\ q_2 \\ q_3 \\ (R - 30\,\text{nm})/(130\,\text{nm}) \end{pmatrix} \tag{3.12}$$

to a dimensionless unit in the range of 0 to 1.

For the purpose of generating a dataset, the parameter vectors $\boldsymbol{v}$ are randomly sampled. The radius R is uniformly sampled from the size range $30\,\text{nm} \leq R \leq 160\,\text{nm}$. The rotation is generated in the axis-angle representation. The rotation axis $\boldsymbol{n}$ is sampled from a uniform distribution over the surface of the unit sphere and the rotation angle from the linear interval $\varphi \in [0, 2\pi[$. For simulation of the corresponding scattering pattern a discretized density matrix of 192 × 192 voxels is created from a cubic volume of 320 nm in length. The entries of the three dimensional tensor are set to either 1 (inside the icosahedron) or 0 (outside). This object tensor is zero padded to a size of 512 × 192 in the lateral dimension and the scattered electric field is calculated through the MSFT algorithm described in Sec. 3.1. The resulting field is cropped to the central 128 × 128 pixels containing a physical signal, which corresponds to a field of view of 2π in beam direction, recall the half sphere in Fig. 3.1. To simulate a measurement, the scattering field is converted into an intensity profile and a random dark-noise factor is added. The intensity is further scaled logarithmically to account for the range of relevant intensities, spanning over several orders of magnitude. Finally, the logarithmic intensity is normalized to a range of 0 to 1 and the scattering pattern is saved as a .png-image file and the corresponding parameters are written to a table. Through the normalization process, the information of the signals magnitude is lost. This information, however, is also unavailable in experiments as the FELs intensity can vary between individual shots. The data generation is performed in Mathematica and each generation takes roughly 30 s. Hence, the dataset is pre-generated and stored on a hard drive and can later be loaded in a relatively short time. Upon import, the axis-angle parameters are converted into rotation quaternions and projected into the fundamental domain, as described in Sec. 3.2.2. The dataset is split into a larger training set and a validation set at a ratio of 0.2 [7]. The scattering patterns of the training set are augmented according to Sec. 3.2.3 and randomly shuffled. The size of the dataset was increased in parallel to the development of the neural network until a convergence of the prediction performance on the validation portion relative to the set-size was observed. The final dataset contains 25 361 scattering pattern-parameter pairs.

With the structure of both input- and output-data determined, the neural network needs to predict a five-dimensional continuous parameter vector from a two dimensional, 128×128 sized, single color-channel image. Convolutional neural networks are particularly well suited for solving this image-regression task, recall Sec. 2.2. Yet, tests with a simple LeNet-style architecture (recall Sec. 2.2) did not yield adequate results, likely due to vanishing gradients. Instead, a ResNet (from **Res**idual **Net**work) architecture is used [108]. It implements convolution operations f as residual operations $\mathbf{x}_{i+1} = \mathbf{x}_i + f(\mathbf{x}_i)$ through the use of identity shortcuts to add up the input tensor $\mathbf{x}_i$ with the output of the convolution operation.

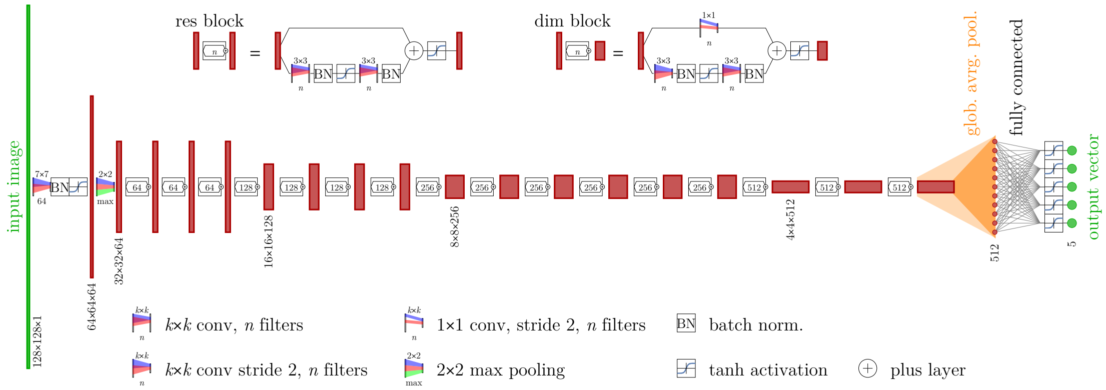


Figure 3.10: The neural network design is adapted from the ResNet34 architecture [108], constructed from res-blocks (top left section). It realized the residual operation, by adding up the input tensor with the ouput of a double convolution- and batch normalization (BN) stack with central activation in a plus layer. The sum is then post-activated in another elementwise activation layer (act.). Dimensionality reduction is achieved through a modified version, the dim-block (top right section), using a 3×3 convolution layer with stride 2 in the conv-stack and a 1×1 convolution with stride 2 as the skip-connection. The input stage of the network is formed by a 7×7 convolution layer with stride 2 (also followed by BN and act.) and a max-pooling layer. The main body consists of 14 residual blocks, arranged in four stages of doubling filter dimensions and halving lateral dimensions. The final 512 convolution filter maps of lateral dimensions 4×4 are flattened through a global average pooling layer, converting each convolution map into its mean value. It feeds into the final five fully connected neurons, forming the output stage of the network. All activation functions are set to tanh.

We use a modified version of the 34-layer deep ResNet34 from [108], sketched out in Fig. 3.10. The basic building block is that of a so-called res-block, see top of Fig. 3.10, realizing the residual operation. The input tensor feeds into a stack of 3×3 convolution layer, batch normalization, element-wise activation, another convolution layer and, again, batch normalization. It is then added to the original input tensor through a plus layer, followed by the the second activation layer in a post-activation design. Different to the original ResNet design, all activation functions are set to tanh, resulting in more consistent results over multiple training runs with random initialization. ResNet utilizes convolution operation with a stride of 2 for lateral dimension reduction, skipping every other input pixel, instead of pooling layers. The dimension operations are also embedded into modified res-blocks, called dim-blocks (see top-right of Fig. 3.10). Therein, the identity shortcuts are implemented by 1×1 convolution operations with stride 2, to match the dimensionality of the convolution path. In total, ResNet34 contains 16 residual blocks.

In the input stage of the neural network, a quick reduction in the lateral dimension is achieved through two operation: one 7 × 7 wide convolution operation with stride 2 and a single max-pooling operation. Another feature of ResNet is the usage of Mean-pooling before transition to the terminal fully-connected layer instead of a flattening operation. It effectively converts the 512 4×4 convolution feature maps into a vector of the 512 mean-values of each of the feature maps. They feed into five fully-connected neurons, giving the final output of a five-dimensional parameter vector. In total, the network has 21 302 917 trainable parameters.

The loss function to be optimized upon training is the mean-squared error between the predicted parameter vector $\boldsymbol{p}$ and target vector $\boldsymbol{t}$ of each sample. The complete data-handling, neural network and training is performed in MATHEMATICA. We use the *ADAM* (adaptive moments) optimizer [7, 172] and a batch size of 128. Training is performed over 40 full iteration of the training dataset, containing 223 168 augmented samples. At the end of each full iteration of the dataset (epoch), the average loss on the validation set is calculated for monitoring. An example learning curve with both training loss (smoothed, for each batch) and validation loss (for each epoch) is shown in Fig. 3.11. A convergence on the validation set is observed at 40 epoch, with a clear overfitting to the training set.[9] The calculations are executed on the computer *qoms* with a single Nvidia GTX1060 consumer graphics card and takes approximately 4h 30min. For simplicity, this trained neural network is later referred to as ICONET.

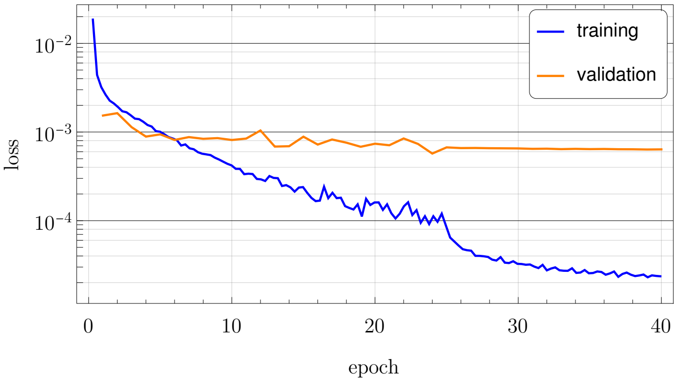


Figure 3.11: The learning curve of the neural network training converges within 40 epoch for the (epoch-wise) value of the validation loss (blue curve). The (batch-wise, smoothed) training loss (orange curve) indicates significant overfitting to the training dataset.

The prediction consistency and accuracy of the trained neural network is tested on an independent training set, containing 5000 images, generated with the same procedure as the main dataset. The re-normalization of the parameter vector $\boldsymbol{v}$ in Eq. (3.12) still applies slightly different weights to the single parameters. So, an evenly weighted parameter-vector difference is defined

$$\boldsymbol{d}_{\text{par}}(\boldsymbol{p},\boldsymbol{t}) = \begin{pmatrix} |p_0 - t_0|/\sqrt{1-q_{\max}^2} \\ |p_1 - t_1|/(2\,q_{\max}) \\ |p_2 - t_2|/(2\,q_{\max}) \\ |p_3 - t_3|/(2\,q_{\max}) \\ |p_R - t_R|/(130\,\text{nm}) \end{pmatrix}, \tag{3.13}$$

by dividing the absolute difference between the components of each vector by the allowed

[9] In this case, we see *benign overfitting* [94, 95], due to the limited information content of the scattering patterns. This introduces an effective noise onto the training data.

parameter range. The range of the imaginary quaternions is limited by the diameter of the circumsphere of the fundamental domain with radius $q_{\max} = \sqrt{3}\,(2\,\cos(\pi/5))^{-3}$. From this evenly weighted parameter difference, error measures can be derived. Most importantly, the mean-error and max-error with the corresponding function applied to the difference-vector, respectively. The distribution of both error metrics is shown in Fig. 3.12(a).

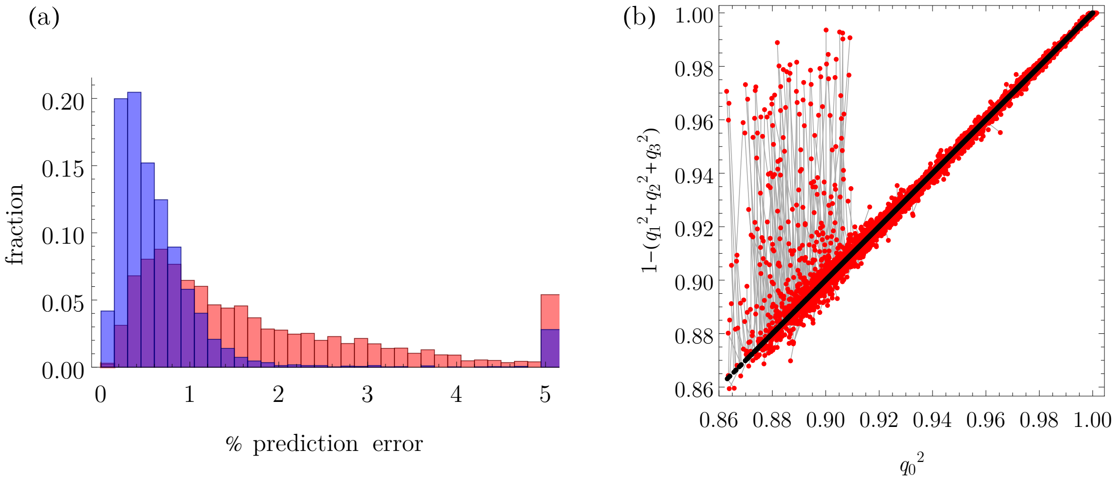


Figure 3.12: The parameter prediction errors (a) of the IcoNet on the test set are evaluated in both the mean-error (blue) or maximum-error (red). For the majority of test-samples, the mean-prediction error is below 2%. The right-most bin contains all samples with errors above 5%. These can mainly be attributed to self-inconsistent prediction. They can be identified (b) through inconsistencies in between the four quaternion parameters, by exploiting the unit-length relation in eq. (3.14). While the target parameters (black dots) form a straight diagonal line and most predictions (red dots, connected to targets by gray lines) are scattered closely around the diagonal, some major outliers appear at large rotation angles. These appear close to the border of the fundamental domain.

The performance is slightly worse than on the validation set, mainly due to some quadrants of the fundamental domain being underrepresented in validation set due to an error in the sampling distribution. Still, for a large majority of the samples the mean-prediction error is below 2%. In the last bin in Fig. 3.12(a) all predictions with an error above 5% are collected. These mainly correspond to self-inconsistent predictions, with conflicting quaternion parameters. To form a unit quaternion, the relation

$$q_0{}^2 = 1 - \left(q_1{}^2 + q_2{}^2 + q_3{}^2\right) \tag{3.14}$$

between the quaternion parameters q_i needs to be fulfilled. When plotting the right-hand side over the left-hand side of Eq. (3.14), as done in Fig. 3.12(b), the target parameters (black dots) form a straight line. Most predictions follow this line (red dots, connected to corresponding target by gray line), with some major outliers. The outliers appear mostly at large rotation angles, close to the border of the fundamental domain where continuing rotations are mapped to the opposite end end of the domain. As already discussed in Sec. 3.2.2, the use of all four quaternion parameters enables us to easily detect such prediction errors to improve the overall prediction result.

3.2.5. Comparison to Classical Reconstruction Methods

We have demonstrated, that a trained neural network is capable of reconstructing size and orientation parameters for known particle shapes from scattering patterns with high accuracy. However, in this task it has to compete with the classical Monte-Carlo-simplex (MCS) algorithm utilized in Refs. [84, 93, 155]. For comparison with this established forward fitting method, we also reconstruct 30 random scattering patterns of the test set with the state-of-the-art Monte Carlo simplex procedure, as used in [84].

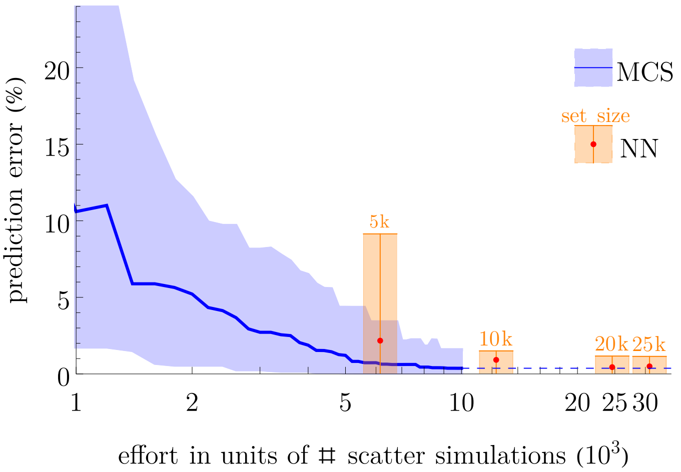


Figure 3.13: From the test set, 30 random samples are reconstructed with the Monte-Carlo-Simplex (MCS) algorithm, with 50 random starting parameters, each. The median accuracy (blue line) of the parameter errors as defined in Sec. 3.2.4 reaches a value of 0.37% within 50 iterations, of which each require 4 scattering simulations, on average. The blue shaded area outlines the region between the minimal and the 90% quantile of the best-fit runs for each image, visualizing the error margin of the MCS method. In comparison, the performance of neural networks trained only on subsets of the dataset of different sizes are marked by red dots and orange bars, representing the median error and 90% quantile, respectively. Each neural network marker is positioned horizontally at the sum of the training set size and scattering simulation equivalent of the training time. On the full dataset (25k) a comparable accuracy is achieved with a computation effort of only a few MCS reconstructions.

For each sample, the MCS reconstruction is started from 50 random initial points in parameter space and iterated over 50 simplex optimization steps. On average, each iteration step required the simulation of four scattering patterns. Depending on the starting conditions and trajectories, some iterations do not reach the optimal set of parameters. The convergence of the reconstruction error as a function of scattering pattern images simulated is drawn in Fig. 3.13. The solid line marks the median best approximation, and the shaded area outlines the 90% quantile. By using the median and 90% quantile, the impact of stark outliers is limited.

Each MCS reconstruction required the simulation of roughly 10 000 scattering patterns, which is on a similar magnitude with the dataset of 30 000 scattering patterns used in training the neural network. The evaluation time of the neural network for any scattering pattern is 5 ms, which is nearly instantaneous compared to a single scattering simulation. However, the neural network also requires time for training beforehand. To allow a direct comparison, the training time is converted into a scattering simulation equivalent. On a hexa-core Intel Xeon E5, the optimized scattering code from Ref. [84] implemented in Matlab requires ~ 2.5 s per image. The training of the neural network takes 4.5 h for 40 epoch on the full dataset of 25 361 scattering patterns, resulting in a time equivalent of 31k scattering pattern calculations to yield the ready-to-use neural network.

As a direct comparison, the prediction accuracy and total time equivalent for neural networks trained only on portions of the full dataset are marked in Fig. 3.13, again as median error (red dots) and 90% quantile (orange bars) for the complete test set. Already with a dataset of 10k scattering patterns the neural network accuracy is close to the MCS algorithm. Training with the full dataset matches the accuracy of the MCS method, requiring a time equivalent of three MCS reconstruction. Consequently, already for a small number of reconstructed images, in this case three, the computational overhead of generating a training dataset and network training are compensated by the near instantaneous reconstruction speed whilst achieving results of comparable accuracy. Therefore, the time required for running reconstructions on larger datasets is reduced significantly and would even allow to reconstruct entire experimental datasets instead of small sample portions. With the megahertz repetition rate of modern FEL experiments [173] fast reconstruction methods become especially important. Given the right hardware neural networks further unlock the real-time reconstruction of experimental data.

3.2.6. Evaluation on Experimental Data

The neural network trained on augmented simulated data has shown excellent parameter reconstruction capabilities matching existing MCS methods on simulated data. The network's ability in recognizing the size and orientation parameters from imperfect experimental images can be tested on the data from Ref. [93], wherein the parameters for two scattering patterns attributed to icosahedra (left column in Fig. 3.14) haven been reconstructed using the MCS method. The size and spatial orientation reconstructed by the neural network are shown in Fig. 3.14 (central column) in the same view-direction as in Ref. [93]. As the true parameters of the objects are impossible to know (hence, the effort of reconstruction), the only way of validating the prediction is to again simulate the corresponding scattering pattern (right column in Fig. 3.14). The neural network predictions match the MCS results published in Ref. [93], with the exception of the radius of particle (b) with the triangle face in beam-direction. We attribute this deviation to the reduced visibility of the radial fringes due to noise, which complicates an accurate radius determination with any method.

Our results demonstrate, that neural networks are capable of matching the reconstruction accuracy for spatial parameters of established, classical algorithms. They are capable of extracting plausible parameters even on experimental data, while being trained solely on simulated data with the help of data augmentation techniques. They offer a significant computation speedup with their near instantaneous execution time over iterative methods, as the time overhead of dataset generation and training is offset by only a small number of MCS reconstructions.

The approach presented here is transferable to any object geometry without any change to the neural network, as long as the rotational symmetry group is known. However, similar to the MCS approach with a parametrized object model, it requires that the object geometry is already identified. This shape identification is crucial, as the neural network, trained solely on icosahedra, will always predict an icosahedron that matches the input pattern as closely as possible from its experience, even if the input particle is not an icosahedron. We will later see in Sec. 3.4.9 that in fact the scattering pattern with five-fold symmetry in Fig. 3.14(a) may not belong to an icosahedron, but a very similar particle with the same symmetry along the observed orientation. Consequently, the correct shape identification is paramount.

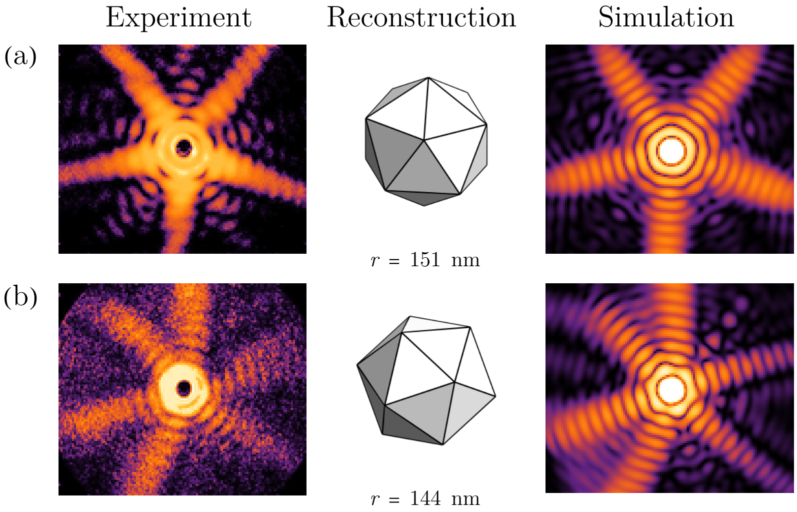


Figure 3.14: The interpretation capacity of the neural network on experimental data is tested with scattering patterns taken from Ref. [93] (left column, permitted by Creative Commons CC-BY 4.0 license (`http://creativecommons.org/licenses/by/4.0/`)). The central column shows the reconstructed radii and orientation as observed in beam direction and is very close to the MCS results in Ref. [93]. Scattering patterns simulated from the reconstructed parameters (right column) reproduce the features of the experimental very well. The simulated intensities are clipped at a maximum intensity to account for saturation effects in the MCP detector.

3.3. Identification of Object Shapes from Scattering Patterns

In the reconstruction of silver nanoclusters from wide-angle scattering patterns, the correct identification of the particle shape is a crucial step. Currently, this task still needs to be performed by a human scientist, as the shape of silver nanoclusters observed in Ref. [93] offer no simple general parametrization, in contrast to e.g. Helium nanodroplets in Ref. [84]. Conveniently, the identification of an object shape from an image is an excellent example of an image classification task, in that a scattering pattern (image) is to be associated with a label (name of the object shape) from a limited set. Image classification tasks are of the core applications for neural network, where even simple architectures have outclassed any classical algorithm for the last decade [6–8]. Consequently, the particle shape identification of a nanoparticle from the corresponding scattering pattern should be a task well suited for a neural network. In fact, this very assumption sparked the idea, that led to the discoveries documented in this whole chapter.

The training set is generated in Mathematica, just as for the icosahedra in Sec. 3.2. The choice of base shapes is introduced in Sec. 3.3.1. The neural network construction and training is done in python with the TensorFlow framework and Keras frontend, due to the better performance and customizability. Switching to python also offers more dynamic data augmentation, described in Sec. 3.3.2. The neural network architecture and training performance on simulated data are reported in Sec. 3.3.3. In Sec.3.3.4 the incapability of the neural network in classifying experimental data is analyzed.

3.3.1. Base Shapes for Classification

For a successful identification of base shapes from scattering patterns, all geometries appearing in the experiment also have to be present in the training dataset. The experiment in Ref. [93] surprised in this regard, as it indicates the existence of previously unexpected shapes. A good example are the decahedra or icosahedra with five-fold symmetries, which are deemed unstable for atom counts above 10^6 [93]. Additionally, the high degree of symmetry in the observed scattering patterns is linked to a highly symmetric shape of the solids. To include a large variety of highly symmetric but still distinguishable convex object shapes, we assemble our basis set from all Platonic solids, all Archimedean solids (except the snub dodecahedron), the decahedron (or pentagonal dipyramid) and truncated twinned tetrahedron as well as spheres and convex polyhedra with fully random vertices, named potatos. This set is depicted in Fig. 3.15 and sample scattering patterns can be found in the App. B.2. The objects from the random class are formed from the enclosure of 50 random vertices, and it is meant as a fallback case if non-symmetric objects appear. It is worth noting, that some of the included geometries, especially the tetrahedron, are highly unlikely to ever be formed by silver atoms in an experiment. They are included nonetheless, to allow identification of other currently unexpected geometries.

For data generation, one object class is randomly chosen (with even weights) and scaled and rotated randomly. The radius, as a size parameter, is defined by the distance between the center of mass and the farthest vertex, and ranges from 31.875 nm to 159.375 nm (or 0.2 to 1.0).

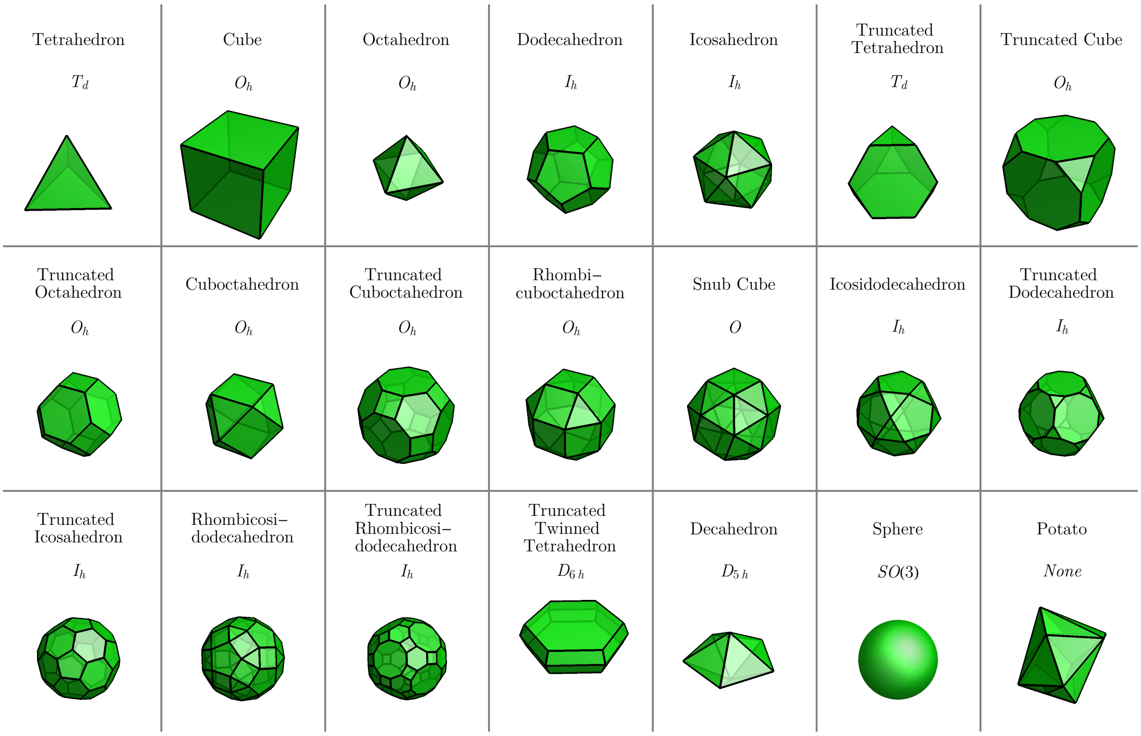


Figure 3.15: The basis set of 21 shapes is assembled from all Platonic and Archimedean solids (except for the snub dodecahedron) and, additionally, the decahedron, the truncated twinned tetrahedron, spheres and polyhedras with random vertices. An example of each is given with it's name and space symmetry group symbol.

3.3.2. Augmentations for Classification Tasks

The successful extraction of size and orientation parameters from experimental data in Sec. 3.2 has proven the feasibility of a training on simulated data when utilizing image augmentation filters for introducing image defects. Switching from Mathematica to python requires re-implementation of the augmentation filters listed in Sec. 3.2.3. On the upside, the data generator class of Keras allows for live-augmentation during training. Instead of loading the complete dataset into the RAM and creating augmented versions of the scattering patterns before training the image files are read, loaded and augmented parallel to the training process in batches when needed and discarded afterwards. This procedure does not only save RAM space, it also even more limits overfitting. The repeated augmentation ensure that the neural network is never presented with the exact same image twice.

The full list of augmentation filters is given in Tab. 3.2 and examples are shown in Fig. 3.16. Wherever the definition of an augmentation filter is unchanged, Tab. 3.1 is referenced. In accordance with common practice in computer vision tasks and the identity filter is dropped. Therefore,the neural network is always challenged with an augmented image [7]. Additionally, scattering patterns are allowed to be rotated for the classification task in order to expand the space of rotations experienced. There are two different ways of approaching such rotations. The first being randomly rotating scattering patterns during training and later feeding input patterns in their original orientation, which allows for checking the consistency of a prediction through feeding the neural networks a set of random rotations of the same pattern. An alternative approach is to perform a principal rotation.

Table 3.2: Full list and description of the augmentation filters used.

Filter Name	**Decription**
uniform noise	Add a uniformly sampled value between 0 and 0.2 to any pixel to simulate noise in general.
Poissonian noise	Add a a value of $0.1\,p$ with p sampled from a Poissonian distribution with $\lambda = 1.0$ to any pixel. Although not correctly simulating shot noise, it helps in mimicing the graininess appearing in some experimental patterns.
blur	Same as in Tab. 3.1.
salt & pepper	Same as in Tab. 3.1.
saturation	The intensity values of the scattering pattern are multiplied with a scalar between 1 and 2 and clipped at 1 to represent saturation effects of the detector.
central hole	Same as in Tab. 3.1 but with radius $r \in [0, 10]$.
cropping	Same as in Tab. 3.1 but down to lengths of 43 pixels.
shift	Same as in Tab. 3.1.
simulated experiment	In order, apply the filters **central hole**, **uniform noise**, **blur**, **Poissonian noise**, **saturation**, and **cropping**.

By dividing the scattering pattern into angular sectors and rotating the sector with the highest total intensity upwards, the network effectively has to learn fever rotations of the same particle and lifts the burden of identifying the main reflex from the neural network. Both approaches are tested during training with equal results.

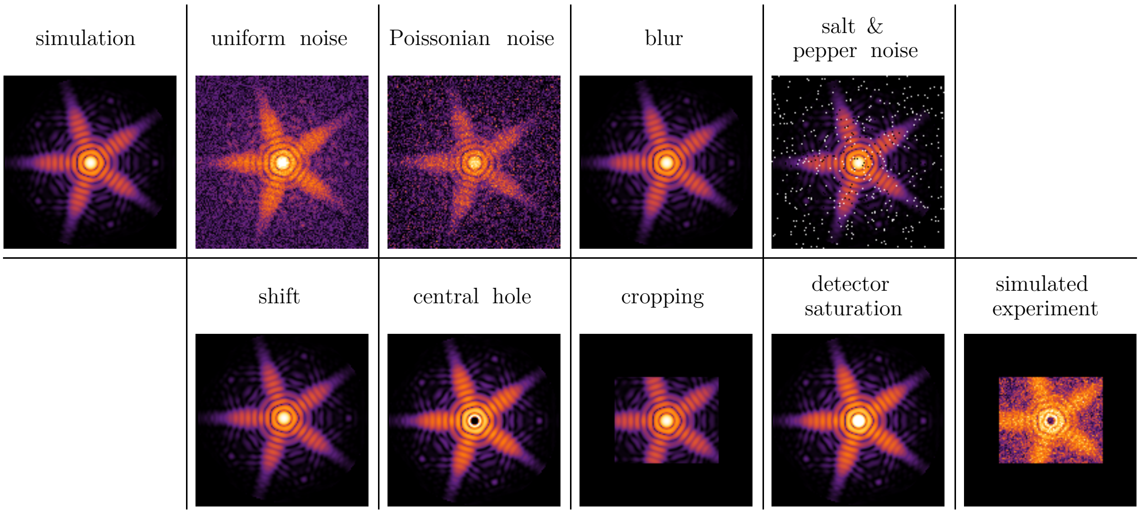


Figure 3.16: The simulated scattering patterns (top left) are modified with a variety of defects in an image augmentation step, in order to both increase the robustness of the neural network against defects appearing in real experiments and increase the effective size of the dataset.

3.3.3. Neural Network Identification of Cluster Shapes

The dataset for training a neural network in the classification task of determining the object shape from the corresponding wide-angle scattering pattern is composed of 58 241 individual samples with an approximately even distribution, as listed in Tab. 3.3.

Table 3.3: Counts of samples per class in the classification dataset in descending order.

Shape	Count
Sphere	2870
Octahedron	2854
Truncated Cuboctahedron	2838
Truncated Twinned Tetrahedron	2833
Icosidodecahedron	2832
Truncated Octahedron	2811
Dodecahedron	2808
Truncated Tetrahedron	2795
Cube	2784
Tetrahedron	2784
Rhombicuboctahedron	2778
Cuboctahedron	2758
Truncated Rhombicosidodecahedron	2754
Truncated Dodecahedron	2754
Truncated Icosahedron	2745
Truncated Cube	2735
Potato	2721
Snub Cube	2720
Decahedron	2714
Icosahedron	2692
Rhombicosidodecahedron	2661

Again, the full dataset is split at a ratio of 0.2 into a training and a validation set. The dataset is loaded in form of a table containing the path to the image-file of the scattering pattern and corresponding shape label. As described in Sec. 3.3.2, batches of data-pairs are loaded parallel to the training process. The images files of the afflicted selected entries are loaded from their data-paths and augmented with a random augmentation filter. Afterwards they are rotated either randomly or to their principal rotation. The class labels are converted into 21-entry long vectors using one-hot-encoding, where each position of the vector corresponds to one object shape and the corresponding entry gives the probability for that class. Logically, the dataset labels are binary values of either 1 or 0. The indices of the training dataset are shuffled randomly at the start of each training epoch, giving a random order for each iteration of the dataset. The validation set is neither augmented nor shuffled, but the principal rotation is applied if active for the training run.

Motivated by the success in the parameter reconstruction in Sec. 3.2, the neural network is constructed as a stock ResNet34 with ReLu activation functions, as in Ref. [108], and a terminal layer of 21 neurons with softmax-activation. The ResNet34 incorporates a total number of 21 293 397 trainable parameters. The deeper ResNet50 and ResNet100 [109] structures are also tested, but offer no significant performance boost while requiring longer training times. The softmax function of the terminal layer normalizes the entire output vector. Accordingly, the neural network output is interpreted as a probability distribution over the set of basis classes, and ideally should converge to a near-binary distribution. The loss function best suited for a multi-label classification is the categorical crossentropy

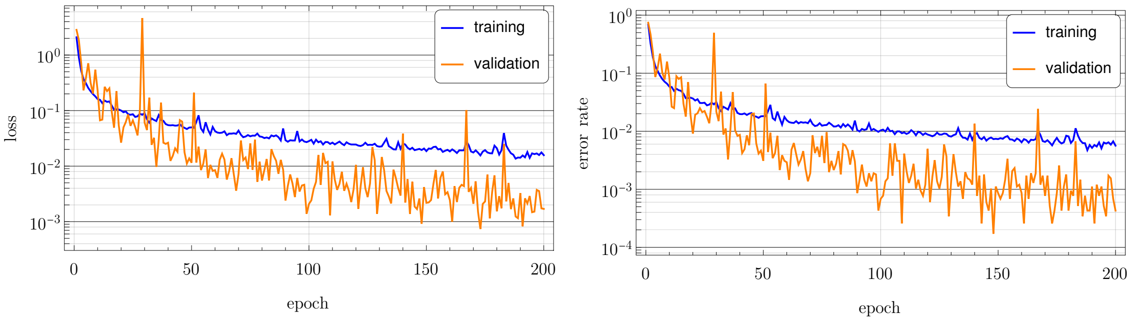
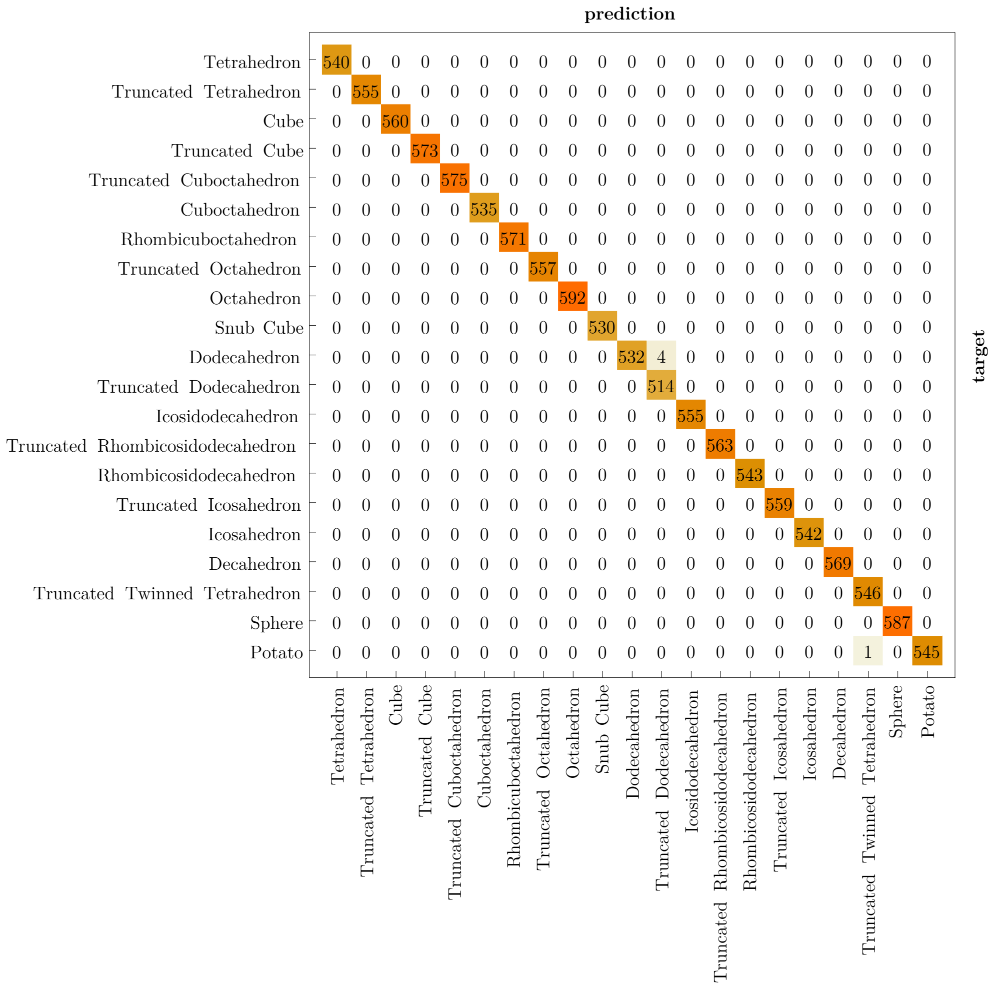


Figure 3.17: The metrics for the training of the neural network identifying cluster shapes through classification are the loss (left) and error rate of the predictions (right). The validation set metrics (orange) are consistently lower than the training set metrics (metrics), due to the absence of augmentations on the validation set. The lack of significant overfitting highlights the improved regularization of utilizing live-augmentation.

function [7]

$$H(\boldsymbol{p}, \boldsymbol{t}) = \sum_{i=1}^{21} t_i * \log(p_i)\,, \qquad (3.15)$$

between the predicted probability vector $\boldsymbol{p}$ and the target vector $\boldsymbol{t}$. Further, as a metric the prediction accuracy is recorded. It counts the fraction of predictions, where the index of the highest probability of the prediction vector $\boldsymbol{p}$ conforms with the target vector $\boldsymbol{t}$. While the accuracy counts only if the largest probability in the prediction vector matches the correct class, the loss function also takes into account the confidence of the prediction through the amplitude of the vector entries. The training is performed on the server *rechenkencht*, on a Nvidia GTX1080ti consumer graphics card, over 200 epochs, taking 4h 20min in total.



	Tetrahedron	Truncated Tetrahedron	Cube	Truncated Cube	Truncated Cuboctahedron	Cuboctahedron	Rhombicuboctahedron	Truncated Octahedron	Octahedron	Snub Cube	Dodecahedron	Truncated Dodecahedron	Icosidodecahedron	Truncated Rhombicosidodecahedron	Rhombicosidodecahedron	Truncated Icosahedron	Icosahedron	Decahedron	Truncated Twinned Tetrahedron	Sphere	Potato
Tetrahedron	540	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Truncated Tetrahedron	0	555	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Cube	0	0	560	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Truncated Cube	0	0	0	573	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Truncated Cuboctahedron	0	0	0	0	575	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Cuboctahedron	0	0	0	0	0	535	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Rhombicuboctahedron	0	0	0	0	0	0	571	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Truncated Octahedron	0	0	0	0	0	0	0	557	0	0	0	0	0	0	0	0	0	0	0	0	0
Octahedron	0	0	0	0	0	0	0	0	592	0	0	0	0	0	0	0	0	0	0	0	0
Snub Cube	0	0	0	0	0	0	0	0	0	530	0	0	0	0	0	0	0	0	0	0	0
Dodecahedron	0	0	0	0	0	0	0	0	0	0	532	4	0	0	0	0	0	0	0	0	0
Truncated Dodecahedron	0	0	0	0	0	0	0	0	0	0	0	514	0	0	0	0	0	0	0	0	0
Icosidodecahedron	0	0	0	0	0	0	0	0	0	0	0	0	555	0	0	0	0	0	0	0	0
Truncated Rhombicosidodecahedron	0	0	0	0	0	0	0	0	0	0	0	0	0	563	0	0	0	0	0	0	0
Rhombicosidodecahedron	0	0	0	0	0	0	0	0	0	0	0	0	0	0	543	0	0	0	0	0	0
Truncated Icosahedron	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	559	0	0	0	0	0
Icosahedron	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	542	0	0	0	0
Decahedron	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	569	0	0	0
Truncated Twinned Tetrahedron	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	546	0	0
Sphere	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	587	0
Potato	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	545

Figure 3.18: Visualizing the number of predicted classes (horizontal axis) in relation to the true target classes (vertical axis) on the validation set in a confusion matrix reveals a near complete diagonal matrix with only few false classification. The highest number of confusions appear for objects form the potato class, which due to their random nature may sport similar structures to other classes or may lie outside of the training space of the neural network.

The learning curves for loss and error ratio as the fraction of false predictions (1 - accuracy) is shown in Fig. 3.17. The validation set metrics are consistently better than the training values. In comparison to the training of the icosahedron reconstructor in Fig. 3.11, it highlights the improved prevention of overfitting through live-augmentation. In this case, the validation set metrics are expected to be consistently lower than the training set metrics due to the lack of augmentations. The loss curve does not fully converge during the 200 epochs of training, but still reaches near perfect accuracy (zero error rate). The neural network trained while recording the metrics in Fig. 3.17 is used in the following as the example network under the name ShapeNet.

The prediction accuracy on the validation set is visualized through the confusion matrix in Fig. 3.18, plotting a heatmap of the predicted shapes (horizontal axis) over the target shapes (vertical axis). It forms a nearly perfect diagonal matrix. The most false classification appear for the randomly shaped potato class. This is an expected outcome, because some of the randomized shapes may actually come close to objects of other classes. The results are very consistent between individual training runs started from random initializations.

3.3.4. Shape Identification from Experimental Data

The reliability of the shape identification neural network on experimental data is tested with the full set of 8 scattering patterns from Ref. [93]. Although the number of patterns is very limited, it is still sufficient for our purposes. The example network achieves a good success rate and the predictions are listed in Fig. 3.19. However, between individual

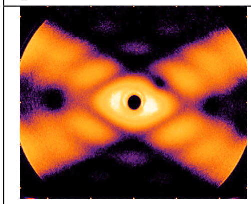
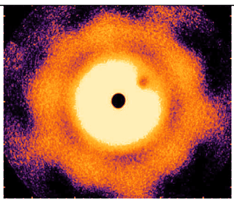
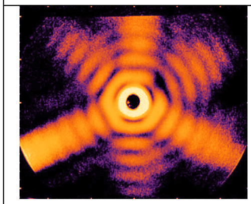
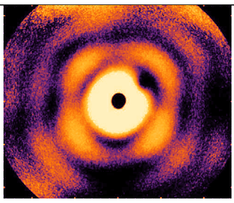
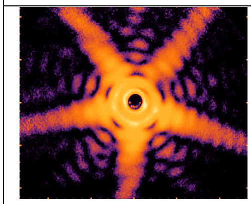
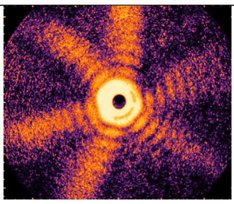
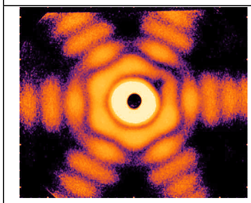
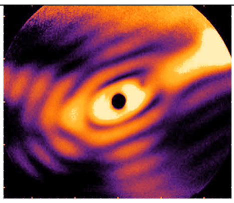
pattern	shape	pattern	shape
	Decahedron 0.9999349		Decahedron 0.9998956
	Truncated Octahedron 0.6077950		Truncated Octahedron 0.0212663 Potato 0.607795
	Icosahedron 2.1×10^{-7} Tetrahedron 0.999969		Icosahedron 0.900110
	Truncated Twinned Tetrahedron 0.995048		Truncated Twinned Tetrahedron 0.999417

Figure 3.19: The shape predictions of the neural network are tested on the experimental patterns from Ref. [93] (permitted by Creative Commons CC-BY 4.0 license (`http://creativecommons.org/licenses/by/4.0/`)). If the neural network prediction agrees with the classical identification from Ref. [93], the label is colored green and the prediction amplitude is given underneath the label. If a disagreement occurs, the neural network prediction is colored red and the shape from Ref. [93] is given in black.

training runs the results vary substantially. Most trained neural networks agree with Ref. [93] in identifying all the pattern associated to decahedra and truncated twinned tetrahedra in the first and last row of Tab. 3.19, respectively. The predictions on the second row (truncated octahedra by Ref. [93]) and the three-fold icosahedron pattern vary substantially with low confidence by the neural network. The low reproducibility indicates that the experimental scattering patterns are too different from the samples seen during training.

A special case is the scattering pattern with five-fold symmetry in the third row of Table 3.19, previously assumed as icosahedral in Sec. 3.2.6. Here, the neural networks of any training run consistently identify it as a tetrahedron instead of an icosahedron. Yet, this association is highly unlikely, both from the view of cluster formation and scattering theory. The scattering transformation translates symmetries of the object into the scattering pattern, and the tetrahedron with symmetry group T_d features only two- and three-fold symmetries. A comparison between a tetrahedral scattering pattern and the experimental pattern is shown in Fig. 3.20. To a human observer the difference becomes immediately obvious in the number of reflexes in the scattering pattern: The tetrahedron (left) produces six, while the experimental pattern in the center of Fig. 3.20 has only five. Neural networks, however, are known to base their decisions also on other features, less obvious to a human observer. This was prominently featured in Ref. [107] through the fact that convolutional neural networks trained on the ImageNet dataset base their decision nearly entirely on the microstructure of an image. As an example, a simple overlay of an image of a cat with the texture of elephant hide, still readable as a cat to a human observer, is enough to make the neural network predict an elephant. A similar effect may be present in the experimental scattering pattern in Fig. 3.20. The fringe pattern of the reflexes is much smoother than for the icosahedron (right panel) and the smoothest fringes from the whole dataset are produced by the tetrahedron. Hence, the neural network may be tricked into predicting a tetrahedron with almost complete certainty just from the microstructure.

The confusion of the neural network opens the question of what was exactly the object producing the five-fold pattern. The steep angles between the triangle faces are the unique feature of the tetrahedron and are by far the smallest over the whole dataset. At the same time, the object producing the five-fold pattern is required to possess a five-fold symmetry along the beam axis. Combining both features is possible in stretching a pentagonal bipyramid along the five-fold symmetry axis, as shown in the bottom panel of Fig. 3.20. The decahedron is related to the icosahedron, in that the latter can be sectioned along any diagonal line into a front-facing pentagonal pyramid, a mantle of ten alternating triangles and a second rotated pentagonal pyramid. In the present orientation, the scattering pattern of the icosahedron is dominated by the mantle faces, aligned nearly parallel to the optical axis. Their contribution to the scattering signal is a five-fold star with broad arms, that dominates at small scattering angles. The front face contribution is only noticeable at wide scattering angles. In contrast, a regular decahedron produces a scattering pattern with ten-fold symmetry. However, the elongation of a decahedron breaks that symmetry in the wide-angle scattering pattern, resulting in two distinct sets of five reflexes, each, with different intensities. This can be seen in the bottom central panel of Fig. 3.20. Due to the shared symmetry, the orientation parameters predicted by IcoNet in Sec. 3.2 are directly transferable. The rotation of the elongated decahedron about the C_5 axis in Fig. 3.20 is the very same as for the icosahedron. Only the predicted size has to be converted by matching the pentagonal bases of both the icosahedrons and decahedron front-facing pyramid. The resulting circumradius of the pentagon from IcoNets prediction is

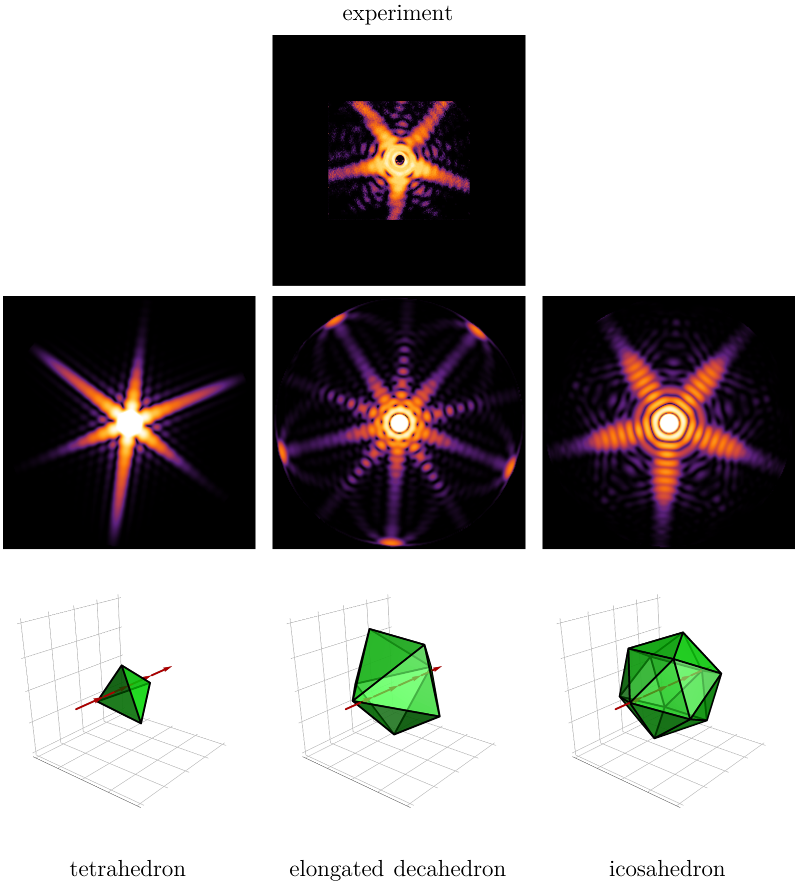


Figure 3.20: The five-fold experimental pattern (top panel, taken from Ref. [93], permitted by Creative Commons CC-BY 4.0 license (`http://creativecommons.org/licenses/by/4.0/`)), is associated to a tetrahedron (bottom left panel) by the neural network, even though the global six-fold symmetry of the tetrahedral scattering pattern contradicts this association. The five-fold star pattern is much closer reproduced by an icosahedron, but with a more prominent fringe pattern. Combining both the steep face angles of the tetrahedron and the front-face of the icosahedron is possible in the shape of an elongated decahedron (bottom, center panel), with a scattering pattern replicating the experimental pattern much closer, but still not perfectly.

134 nm which is confirmed by matching the fringe spacing in the radial profile. This result indirectly confirms that IcoNet indeed finds the best fitting icosahedron even for non-icosahedral particles. The elongation factor s, by which the decahedron is stretched, is determined from a grid search with increments of 0.1 and the best-fit value is $s = 1.6$. Therein, it combines the pentagonal geometry of the scattering pattern with steep face-angles, like appearing in the tetrahedron.

Summarizing, the incapability of the neural network to determine viable structure candidates from experimental scattering patterns may result from an insufficient training set space. Adding elongations as deformations to the training set could help in circumventing the problem. It would require an additional label for elongations, but still may be insufficient to cover all major defects appearing in experimental scattering patterns. One example are varying truncation rates, as reported in Ref. [93]. Further, the experimental scattering patterns in Fig. 3.19 feature small asymmetries linked to deviations from a completely symmetric shape. Hence, the classification approach is discontinued in favor of a more flexible object model capable of directly representing any defects or asymmetries.

3.4. Model Independent Three-Dimensional Object Reconstruction

Identifying the shape of silver nanoclusters in Sec. 3.3 through a classification approach fails due to the the complex geometries appearing in experiments. Nonetheless, the parameter reconstruction network IcoNet has proven its robustness even in the presence of unknown shapes. Consequently, solving the reconstruction task with neural networks may still be possible when giving the neural network enough freedom in the shape representation. The scattering simulation through the MSFT scheme, recall Sec. 3.1, already involves such a representation: the discretized density. For simulation, the optical density of the object is discretized on a cubic coordinate grid and processed numerically. The grid elements can be understood as voxels[10], the volumetric equivalent of a pixel. Such a voxel model is able to represent any shape resolvable with the given grid spacing. Hence, designing a neural network that predicts a discretized three-dimensional object density from an input scattering pattern (called VoxelNet) would allow universal reconstruction of any cluster shape. The main results documented in this section are published in Ref. [1]. The source code is available with a sample portion of each dataset within the repository [174].

Being able to create a three-dimensional model of an object from photographs, also called 3D scanning is beneficial for many practical application, e.g. in re-manufacturing existing parts of machinery. As many problems in computer vision, it already has been efficiently tackled with neural networks [175, 176]. Of special interest for this thesis is Ref. [176]. Therein, a voxel model is utilized in reconstructing objects from a set of images taken at different view-points. The neural network is designed in an encoder-decoder architecture with recurrent elements in the latent space to process sequences of input images. Still, the network is also capable of reproducing all visible features from just one view angle. Ref. [176] bears a strong similarity to the cluster reconstruction task described above. The main difference between the 3D scan and the scattering reconstruction is that only a single wide-angle scattering pattern is available for each reconstruction. Yet, this single pattern includes object information from many view angles simultaneously. For this reason, it appears promising to employ an encoder-decoder network, similar to that of Ref. [176], to reconstruct full voxel models of silver nanoclusters from single wide-angle scattering patterns.

A simplified version of the discretized density reconstruction can be achieved through eliminating one dimension from the task. By projetion into the xy-plane the problem is effectively reduced to a two-dimensional small-angle reconstruction problem.[11] Due to the lower dimensionality it is much more favorable in terms of runtime for testing out methods before implementing them in the three dimensional problem. The solution of this simplified reconstruction is documented in App. B.3 in close relation to the methods described throughout this section.

The exact representation of silver nanoclusters in a discrete voxel model and the method of object generation are given in Sec. 3.4.1. Sec. 3.4.2 revisits the topic of image augmentation in the context of detailed reconstructions which are not confined to distinctive shapes. The neural network architecture itself is described in Sec. 3.4.3. The training of the discrete reconstruction network differs from previous tasks in the special requirements

[10] A voxel (short for: volumetric pixel) is the volumetric extension of the 2D pixel. In turn, the word pixel stands for picture (short: pix) element.

[11] For two dimensional objects, even wide-angle reflexes cannot pick up additional phases. Consequently, the Fraunhofer approximation is always applicable, which is equivalent to the small-angle regime.

to the loss function, which is introduced in Sec. 3.4.4 with the concept of physics informed learning. The training performance is documented in Sec. 3.4.5, followed by an analysis of the test metrics and interpretation in Sec. 3.4.6. A further comparison between trainings with different loss functions is drawn in Sec. 3.4.7 and demonstrates the necessity of the chosen approach. The neural network is applied to experimental scattering patterns in Sec. 3.4.8, where it yields object tensors with currently unmatched levels of details. Finally, we report the prediction of novel shaped candidates in Sec. 3.4.9.

3.4.1. Generation of Discretized Object Densities

As described in Sec. 3.1 the silver nanoclusters can be assumed as macroscopic objects of homogeneous optical density with permittivity $\epsilon_{\mathrm{silver}}$ of bulk silver. By including the value of $\epsilon_{\mathrm{silver}}$ in the scattering algorithm it suffices to define the object density as a binary function

$$\rho(\boldsymbol{r}) = \begin{cases} 1 & \text{for } \boldsymbol{r} \in V_{\mathrm{object}}\,, \\ 0 & \text{otherwise,} \end{cases} \tag{3.16}$$

which takes on the value of 1 within and 0 outside the object, respectively. As in previous sections, a resolution of 192×192×192 is used for the generation of training data. However, this resolution is unwieldy for a neural network output for two major reasons. Firstly, it is not a power of two, thus complicating the design of the neural network while following the general design rule of of changing lateral dimensions and kernel sized by the same factor [7]. Secondly, the deconvolution operations used for unfolding the three-dimensional object are costly operations, both in computation time and memory consumption under back-propagation. In the context of the already considerable demands for computation resources it is sensible to restrict the size of the predicted density. A tradeoff between retaining enough detail and a small volume size is the a resolution of 64 × 64 × 64 voxels. An example of an icosahedron rasterized at this resolution is shown in Fig. 3.21(a).

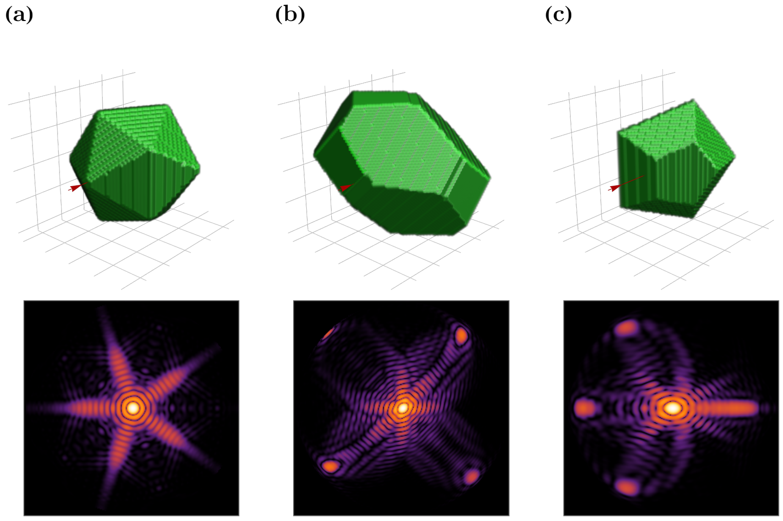


Figure 3.21: In the voxel representation (top row), (a) objects are discretized on a three dimensional grid (here with a resolution of 64 × 64 × 64), as for the already familiar icosahderon. It allows for the introduction of arbitrary deformations like (b) stretching of the base solids, shown here on a truncated octahedron. For testing the neural networks, (c) chopped objects are used, which are truncated along one random symmetry axis, like the elongated decahedron. The red arrowed line in each real-space volume marks the optical axis and the corresponding scattering patterns are found in the bottom row.

While the voxel model allows for creating any possible geometry, we still demand all objects to be convex, recall Sec. 3.1. We do so in order to meet the conditions for unique inversion under ideal conditions [149]. The set of shapes used for classification in Sec. 3.3.1 still covers a wide range of convex geometries, despite the limitations encountered in Sec. 3.3. These elements are used as base solids for object generation. Aside from scaling and rotation, the base solids are further deformed by stretching or squashing the whole object along a randomly chosen symmetry axis, as demonstrated in Fig. 3.21(b). Henceforth, each base solid produces a wider family of generalized objects, like the sphere generalizing to both prolate and oblate ellipsoids. With the freedom of the voxel representation the neural network may be able to interpolate between known geometries even for objects outside of this basis space. To test this hypothesis a separate test set is created. The test set objects are truncated along a single random symmetry axis (see Fig. 3.21(c)). The terminology of truncation is already occupied in the context of regular polyhedra. Therefore, this operation is called *chopping* of the solids. The name is illustrative of the operation being similar to chopping off a slice from each side of the object, like the tips of the elongated decahedron in Fig. 3.21(c).

The object rasterization as well as the MSFT scattering calculations require considerable computation times. Therefore, a data set of 140 000 objects is pre-generated and stored. As in previous sections, scattering patterns are stored as single-channel PNG-images. Further, the original discretized object densities used in simulating the scattering patterns are stored as PNG-images on the full resolution of $192 \times 192 \times 192$ voxels. This is achieved by converting each object slice in z-direction (beam propagation direction) into a black-and-white image and assembling the images in a square grid. In Fig. 3.22 this technique is demonstrated on a smaller resolution with the icosahedron from Fig. 3.21(a). The data generation is executed in Mathematica, distributed over the computers *qoms*, *QuantumChaos* and *computationPi*[12]. The neural network training is executed on the GPU server *eve* (see App. A.1 for hardware details). Storing the object densities as PNG-images ensure compatibility and further minimizes the file size. Upon read-in, the composite images are decomposed into the constituent layer-images and re-stacked into a three dimensional tensor. This $192 \times 192 \times 192$ tensor is then scaled down to dimensions $64 \times 64 \times 64$. To prevent the computational overhead of this pre-processing from thwarting the entire training it is executed in advance and the object tensors are stored as fast-loading binary files on the training server.

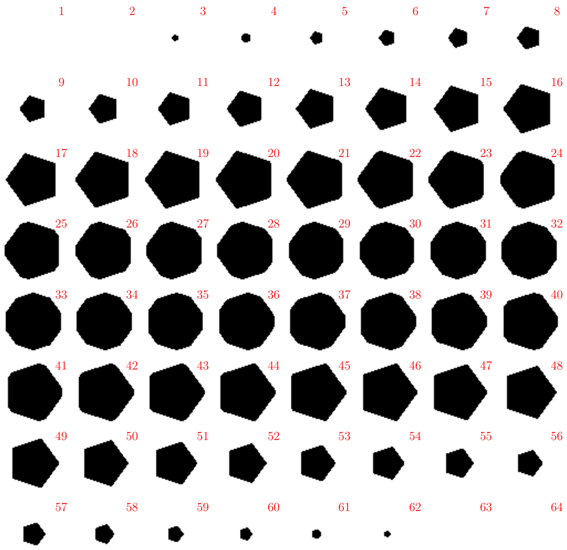


Figure 3.22: Representation of the icosahedron from Fig. 3.21(a) as a two dimensional image, that can be stored as a PNG-file. Each slice of the object density in z direction (red arrowed line in Fig. 3.21(a)) is converted into a black-and-white image (black = object). The images are then assembled in order to a square image. The red number in the top right is the index of the slice in beam-propagation direction.

[12] Over the course of a several months even a Raspberry Pi can contribute over 1000 samples (see App. A.1 for hardware details).

The randomized object generation is performed exactly as in Sec. 3.3.1 with evenly distributed parameters, but with the addition of defects. For elongation one of the symmetry axes of the corresponding symmetry group, recall Fig. 3.15, is selected by random and a scaling parameter between 0.5 and 2.0 is applied. It is clipped at the inradius of the real space volume, to prevent the object from protruding from the sampling area. For the test set, the chopping is executed along the same axis as the elongation at a random fraction between 0.2 and 1.0 of the size parameter. In total, a training dataset of 140 000 objects is generated. The set is organized into portions of 10 000 samples and generation was distributed over the aforementioned computers. The test set is created with 1000 objects.

3.4.2. Simulating Experimental Artifacts by Image Augmentation

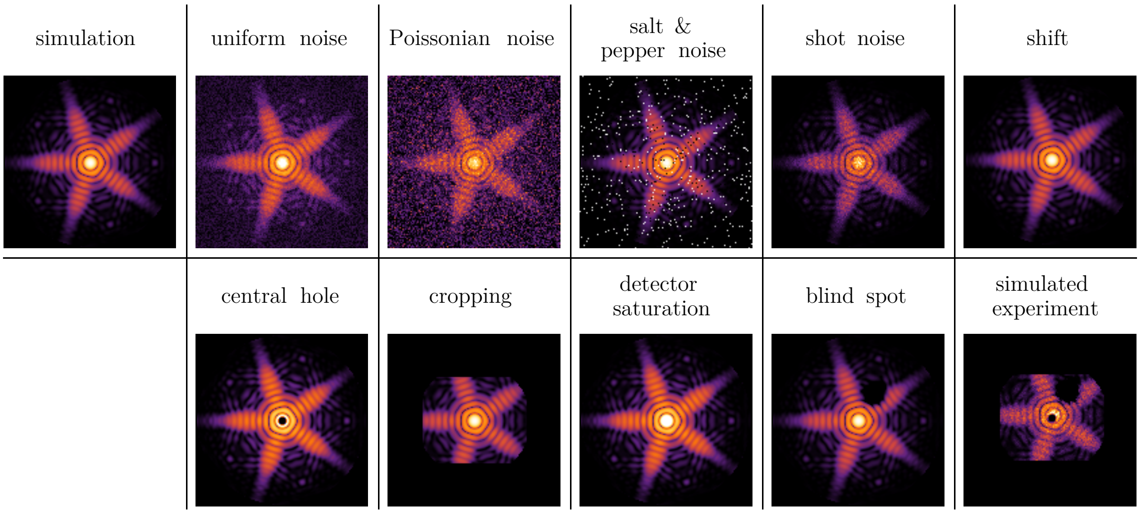


Figure 3.23: The simulated scattering patterns (top left) are modified with a variety of defects in an image augmentation step. An example for each filter listed in Tab. 3.4 is shown in each panel. Upon training, the neural network is always presented with a different augmented image in each epoch and never sees the original simulated scattering pattern.

Like in previous section, the discrete reconstruction neural network is trained with data augmentation to achieve robustness against experimental artifacts. Like in Sec. 3.3.2 training is performed in PYTHON using KERAS with on-the-fly augmentation and using the existing augmentation filters, with slight modifications. For a full list, see Tab. 3.4. The **blur** augmentation is not included, because it is related to the exact type of shape deviations the VOXELNET is meant to reconstruct. The **blindspot** filter form Sec. 3.2.3 is re-introduced and the **cropping** filter is improved to include both rectangular and circular masks. Finally, a new augmentation filter is introduced in the **shot noise** function. It aims at generating a similar effect to the shot noise originating from the discrete nature of photons visible at low intensities. In an exact implementation the intensity profile itself would serve as the argument λ of the Poissonian distribution. Yet, this would require knowledge of the absolute intensity value, which is unavailable from the existing scattering pattern. It would also significantly increase the computation time of this specific filter beyond a critical limit, thus slowing down the overall training process. An example of each augmentation filter is shown in Fig. 3.23.

Table 3.4: Full list and description of the augmentation filters used.

Filter Name	Decription
uniform noise	Same as in Tab. 3.2.
Poissonian noise	Same as in Tab. 3.2.
shot noise	Multiply the scattering pattern with a random Poissonian matrix with variance $\lambda = 10^{r+1}$ where r is an uniform random number from the interval $[0, 1]$. Although not physically accurate, this filter approximates the shot noise originating in the discrete nature of photons in the low-intensity limit with a minimal computational effort.
salt & pepper	Same as in Tab. 3.1.
saturation	Same as in Tab. 3.2, but with a multiplier between 1.0 and 1.5.
blindspot	Similar to Tab. 3.1. Multiply every pixel of the image with a Gaussian defect function $1 - \exp\{-((x - x_0)^2 + (y - y_0)^2)/(2\sigma^2)\}$ around a random point $(x_0, y_0)^T$ from the central half of the image in both width and half, respectively, and variance σ between 0 and 0.25.
central hole	Same as in Tab. 3.2.
cropping	Simulate a limited angular range by multiplying the intensity with both a rectangular and circular binary mask, i.e. setting the affected regions to 0. The size of the central rectangular mask is between 1 and 0.5 times the image dimension in both height and width, deleting up to half of the pixels in both dimensions. The radius of the circular mask is randomly selected between 0.5 and 1 times the larger image dimension.
shift	Same as in Tab. 3.1.
simulated experiment	In order, apply the filters **shot noise**, **shift**, **blindspot**, **saturation**, **central hole**, **cropping**, and **shift**.

3.4.3. Design of the Discrete Scattering Reconstruction Network

The reconstruction of a three-dimensional object density from a two-dimensional single-channel image is very similar to the creating of a three-dimensional model from a series of photographs. The latter is a well known task in classical image processing, which has successfully been solved using neural networks [175, 176]. The dimensional conversion usually requires the use of encoder-decoder architectures, where the input information is encoded into a latent space and then expanded into the target space. Multi-view reconstructions further require the use of recurrent elements in the latent space. Without the recurrent elements, the 3D-R2N2 architecture from Ref. [176] implements a conversion from a two dimensional image to a discrete three dimensional object. It inspired the general architecture of VoxelNet, but was strongly modified to conform with the given task.

The general design idea is to use convolution layers within the encoder and pooling operations to compress the two-dimensional input image into an one-dimensional latent space. The decoder then uses three-dimensional convolution operations and upsampling operations to create three-dimensional feature maps and combines them into one output

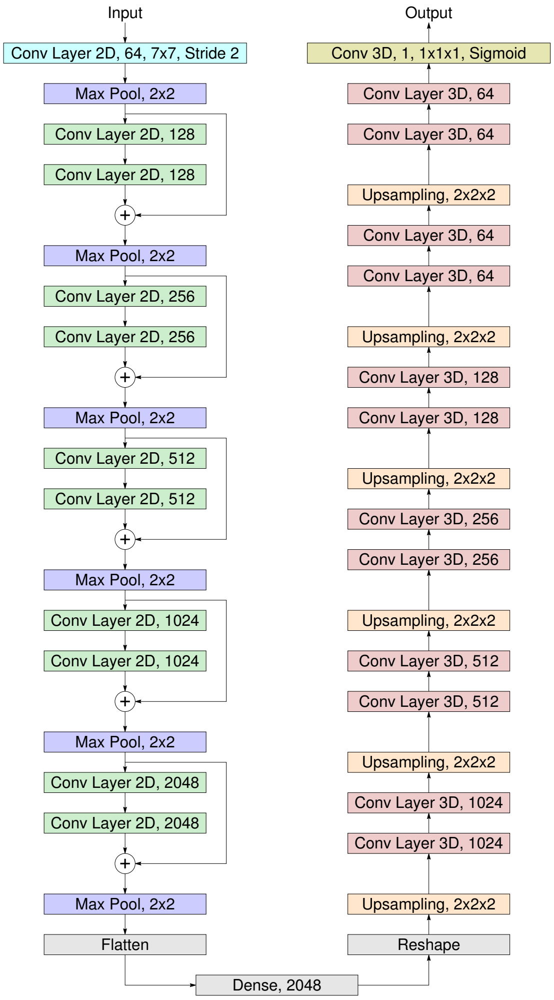


Figure 3.24: The VoxelNet is build in an encoder-decoder structure. The encoder (left column) consists of five residual blocks, each containing two consecutive 2D convolution layers with 3×3 kernels. The filter size is doubled with each residual block, while the lateral dimensions are reduced by pooling layers. The latent space (bottom) is one-dimensional and is cross-linked by a dense layer. After reshaping, the decoder (right column) applies $2 \times 2 \times 2$ upsampling operations followed by two 3D convolution layers each. All convolution layers are regularized with a dropout ratio of 0.2 and batch normalization is applied before the leaky ReLU activation.

density tensor. The upsampling operation (also referred to as deconvolution) is the reverse operation to the pooling operation. It increases the output dimensionality by multiplying each input with a higher dimensional kernel. In our case, we multiply one voxel from the feature maps with a $2 \times 2 \times 2$ tensor. The large number of free parameters of any upsampling neuron makes this operation comparably expensive for backpropagation.

The complete design of VoxelNet is sketched in Fig. 3.24. The input stage of the encoder is a single convolution layer with 7×7 convolution kernels and stride 2, followed by Max pooling operations. It is used to rapidly convert the input tensor size from $128 \times 128 \times 1$ to $32 \times 32 \times 64$ elements. The main body of the encoder is constructed from five residual blocks. Each is formed by a sequence of two 3×3 convolution layers and an identity shortcut, implemented by 1×1 convolution operations for dimensionality matching, recall Sec. 3.2.4. Both paths are added up in a summation layer. Each residual block is immediately followed by a max pooling operation. While the lateral dimension halves between each block, the filter size is doubled, following the general design principles formulated in Ref. [7]. After the fifth residual block, the tensor shape is reduced to $1 \times 1 \times 2048$ and the trivial dimensions are removed by a flatten operation. Interconnections within the latent space are realized by a fully connected layer with 2048 neurons. The decoder stage of the neural network starts with a reshape layer, adding three trivial dimensions, resulting in $1 \times 1 \times 1 \times 2048$ tensors. Similar to the encoder, the decoder is constructed from blocks, but in reverse order. Each starts with a $2 \times 2 \times 2$ 3D upsampling layer (the inverse of a pooling layer) and is followed by a sequence of two $3 \times 3 \times 3$ 3D convolution layers. Different to the encoder, the structure is entirely linear and no residual connections are included. Empirically, we find that residual connections in the decoder stage do not have the same benefits as in the encoder stage but increase the training time significantly. The decoder contains six blocks, resulting in a final tensor with dimensions $64 \times 64 \times 64 \times 64$, i.e. 64 feature maps of three dimensional cubic volumes with length 64. The output stage is formed by a single $1 \times 1 \times 1$ 3D convolution filter, combining all 64 feature maps into a single density volume. It is sigmoid-activated, which suits the binary nature of the object densities.

Aside from the terminal layer and inspired by Ref. [176] all convolution and fully connected layers are activated by the leaky ReLU function

$$\mathrm{lReLU}(x) = \begin{cases} x & \text{if } x > 0\,, \\ 0.01x & \text{otherwise.} \end{cases} \tag{3.17}$$

Further, any convolution layer, both 2D and 3D, includes regularization functions. All convolutions layers are implemented as a stack of

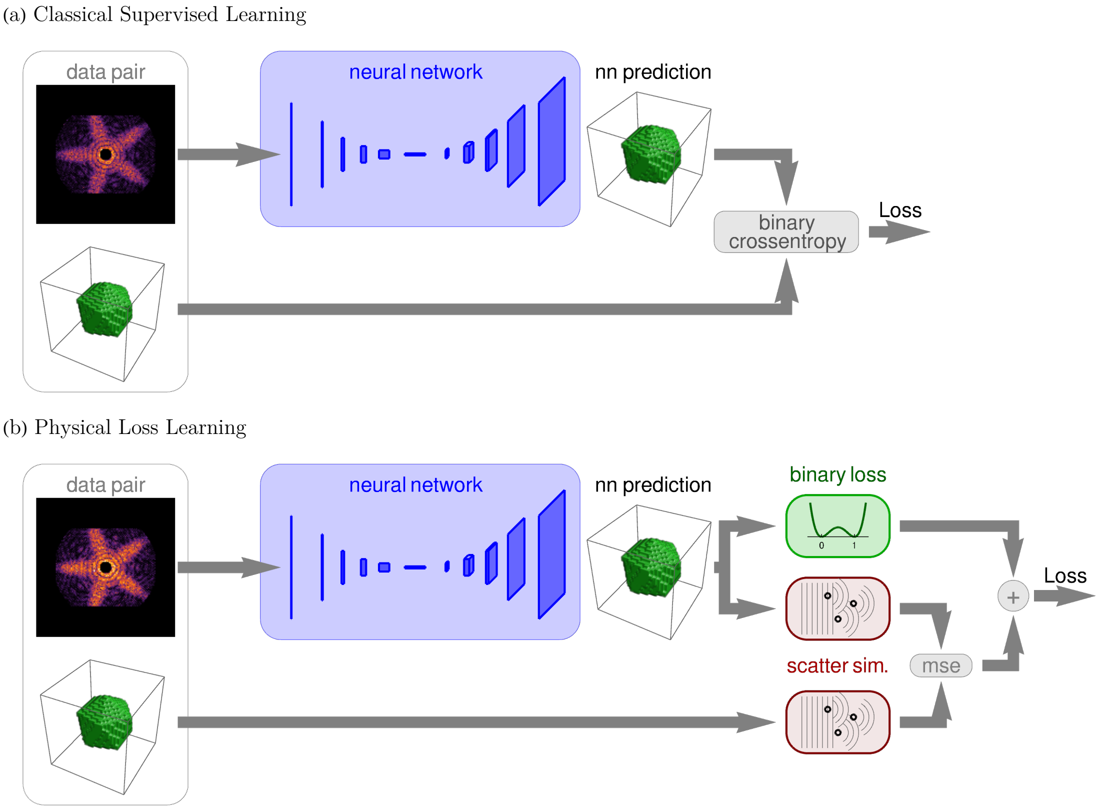
1. convolution operation with padding,
2. dropout regularization with ratio 0.2,
3. batch normalization,
4. activation.

The heavy use of regularization in every layer is a necessary adaption. It ensures consistent convergence results independently of the randomized initialization. Batch normalization counteracts the convergence to no-objects predictions (all voxels are zero), which represent a trivial plateau of the loss-function. Simultaneously, dropout regularization prevents learning non-physical predictions. These are objects like non-binary point clouds that form no solid convex (or at least star-shaped) bodies but produce similar scattering patterns. The full network has 195 813 825 free parameters.

3.4.4. Physics Informed Training

Untouched by the dimensionality, training a neural network in predicting a three dimensional tensor from an image is a straightforward process. In a classical supervised training the neural network is presented with batches of image-tensor-pairs and it is optimized to minimize a loss function between the prediction $\mathbf{p}$ and target-tensor $\mathbf{t}$, as illustrated in Fig. 3.25(a). The binary nature of the object tensors (*object* vs. *no object*) is effectively a binary classification task. It is served best by the use of the binary cross-entropy function [7]

$$H(\mathbf{t}, \mathbf{p}) = \frac{1}{N^3} \sum_{i,j,k=1}^{N} \Big[t_{i,j,k} \, \log(p_{i,j,k}) + (1 - t_{i,j,k}) \, \log(1 - p_{i,j,k}) \Big] \,. \tag{3.18}$$


(a) Classical Supervised Learning

(b) Physical Loss Learning

Figure 3.25: In classical supervised learning (a), the neural network is optimized to match the input object through the loss score, which is determined by the binary cross-entropy between the network prediction and the target entry of each data pair. In the physical learning scheme (b), the loss score is calculated within the scattering space rather than the object space. This is done by simulating the scattering pattern of both the network prediction as well as the target object, and calculating their mean squared difference (scatter loss). To enforce the binary nature of the object model, an additional regularization function (binary loss) is applied to the prediction.

However, it turns out that this approach is not ideal. The reason lies in the optical properties influencing the light scattering. To recall from Sec. 3.1, the absorption length of bulk silver for the present wavelength is 12.5 nm. This is a relatively short length compared to the considered cluster diameters ranging from 63 to 320 nm. Consequently, the incoming radiation intensity decays rapidly along the penetration depth and a substantial part of the particle contributes no significant signal to the final scattering pattern. This is especially true for the far side[13] of the particle, which is shadowed from the incoming radiation. As a result, by using the binary cross-entropy from Eq. (3.18) the neural network is urged to correctly reconstruct portions of the particle that have no representation in the scattering pattern. It either has to reconstruct these sections through symmetric completion from visible sections (which is indeed observed to some degree) or is driven into severe overfitting. The consequence is a poor generalization capability, that will be analyzed in the next section.

At this point it is sensible to recall what the exact goal of the scattering reconstruction is: When working with experimental data the ground truth shape of the object is always unknown, hence there is no target object to match. Instead, we aim to create an object that produces the exact same scattering pattern (under simulation) as the input, while

[13] Similar to the far side of the moon being not visible from earth.

simultaneously fulfilling the restrictions we impose on its geometry, like done with classical algorithms in Refs. [84, 93, 142, 155]. This very approach can be transferred to the training of a neural network through modifying the loss function. Instead of minimizing the difference between predicted $\mathbf{p}$ and target $\mathbf{t}$ object tensor in Eq. (3.18) we train the neural network to minimize the difference between the scattering patterns calculated from $\mathbf{p}$ and $\mathbf{t}$ via

$$L_{\mathrm{s}}(\mathbf{t},\mathbf{p}) = \frac{1}{M^2}\sum_{i,j=1}^{M}\left[\log\left(\left|\boldsymbol{E}_{\mathrm{MSFT}}(\mathbf{t})_{i,j}\right|^2+\epsilon\right)-\log\left(\left|\boldsymbol{E}_{\mathrm{MSFT}}(\mathbf{p})_{i,j}\right|^2+\epsilon\right)\right]^2. \tag{3.19}$$

The scattering patterns of both $\mathbf{p}$ and $\mathbf{t}$ are simulated using the MSFT algorithm, as indicated by $|\boldsymbol{E}_{\mathrm{MSFT}}|^2$, adapted to the reduced resolution. Further, we need to enforce the assumptions of our object model, i.e. the binary nature of our density function. While the output neurons are sigmoid-activated non-binary float-point predictions are still possible, but conflict with the assumed object model. Therefore, the binary condition is enforced by the introduction of a regularization function: the binary loss

$$L_{\mathrm{b}}(\mathbf{t},\mathbf{p}) = \frac{1}{N^3}\sum_{i,j,k=1}^{N}(p_{i,j,k})^2\,(1-p_{i,j,k})^2\,. \tag{3.20}$$

It is added to the scatter loss from eq. (3.19) and weighted by a parameter w to form the total physical loss function

$$L_{\mathrm{phys}}(\mathbf{t},\mathbf{p}) = L_{\mathrm{s}}(\mathbf{t},\mathbf{p}) + w\,L_{\mathrm{b}}(\mathbf{t},\mathbf{p})\,. \tag{3.21}$$

The weight parameter w is a hyperparameter of the training and a good balance is achieved with $w = 0.1$. For larger w, the binary regularization overrules the learning from differences in the scattering pattern, while for smaller w the neural network is not penalized enough for non-binary predictions that approximate the target scattering pattern. A comparison between the full physical training scheme and the classical supervised approach is depicted in Fig. 3.25(a). The physical loss function Eq. (3.21) is implemented in TensorFlow to enable the backpropagation algorithm to trace gradients through. The source code can be found in the code repository Ref. [174].

The method of enforcing physical constraints and implementing generative equation into loss functions is known from the field of PDE solving with neural networks by the term *physics informed learning* (PIL) [77, 78]. Accordingly, our approach of a physical loss function in Eq. (3.21) also qualifies as a case of PIL. Physics informed neural networks (PINNs) usually combine supervised loss functions, in our case Eq. (3.18), with pure physical loss functions. In a similar fashion, the physics loss Eq. (3.21) originally was envisioned as an auxiliary loss function to implement the hybrid loss function

$$L_{\mathrm{hybrid}}(\mathbf{t},\mathbf{p}) = L_{\mathrm{phys}}(\mathbf{t},\mathbf{p}) + w_H\,H(\mathbf{t},\mathbf{p})\,, \tag{3.22}$$

combining Eqs. (3.18) and (3.21) with weight w_H. However, the training with the hybrid loss proves unstable with non-convex predictions and stabilizes only in the limit $w_H \to 0$, equal to the pure physics loss Eq. (3.21). As a result, the training with the physical loss function in Eq. 3.21 is preferred over other approaches. However, a deeper comparison between all three loss functions introduced in this section is made in Sec. 3.4.7, using insights gained from the physical loss training. Further, we refrain from the term of physics informed learning and use the respective names of the loss functions instead, to avoid confusion.

3.4.5. Training the Discrete Cluster Network

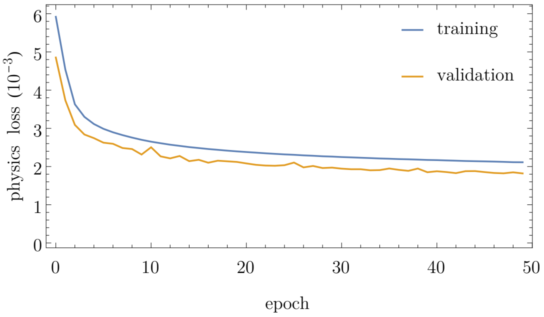


Figure 3.26: The training loss of the ScatterNet converges within 50 full cycles of the training set (epochs) to a near halt. The loss on the validation set follows a similar trajectory, but is consistently smaller than the training loss, due to the absence of augmentations and regularization on the data and the successful prevention of overfitting.

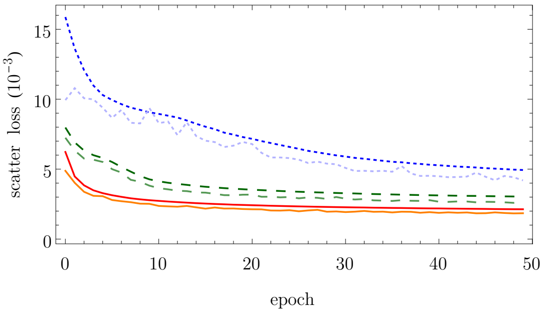


Figure 3.27: Scatter loss recorded over the training of a reconstruction neural network by binary cross-entropy (blue, dotted), hybrid loss (green, dashed) and physics informed training (red, solid). The regular curves mark the training set performance over each iteration of the training set, while the desaturated curves correspond to the validation set.

The VoxelNet neural network and loss functions from Sec. 3.4.4 are implemented within the TensorFlow 2.3.1 Keras framework and Python 3.6.6, with the code published in Ref. [174]. The complete dataset, containing 140 000 samples, is split with a ratio of 0.2 into a training set and validation set. The pre-generated and pre-processed object tensors and scattering patterns are read-in from the hard drive parallel to training. Equal to the procedure described in Sec. 3.3.3, the training set is shuffled randomly every epoch and the training scattering patterns are augmented on-the-fly. The validation dataset is neither shuffled nor augmented and the performance metrics are recorded at the end of every epoch. Again, the network parameters are optimized using ADAM [7, 172]. Training was executed on a the dedicated GPU sever *eve*, distributed over four Nvidia RTX2080ti GPUs. The distributed training enabled batch sizes up to 32 and training with the physical loss converges within 50 epochs. The heavy use of regularization ensures reliable convergence on any individual training run. The corresponding learning curve of the network used throughout the rest of this section is shown in Fig. 3.26(b). For the sake of simplicity, it will be referenced as the ScatterNet. In total, the training over 50 epochs on four GPUs takes 63h. The scattering loss adds a major share to the computation time. For comparison, training with the binary cross-entropy loss (allowing a batch size on 64 on four GPUs) takes only 22h. It is used as a benchmark to the ScatterNet further down this section and is named SupervisedNet.

During training, the scatter loss is recorded as a separate metric at each round. It represents the capability of each neural network to reproduce the input scattering patterns from both the training and validation set. In Fig. 3.27 the metrics for the ScatterNet (red / orange solid), SupervisedNet (blue dotted) and a HybridNet (green, dashed), trained with the hybrid loss from Eq. (3.22), are traced over 50 training epoch, each. Unsurprisingly, both ScatterNet and HybridNet perform much better than SupervisedNet, because they are directly optimized on this metric.

3.4.6. Testing the Discrete Cluster Reconstruction

The generalization capability of the ScatterNet is monitored on the validation set during training. Therein, the learning curve in Fig. 3.26 promises good interpolation results for objects outside the training set. Still, the validation set is based on the same generation rules as the elements of the training set. Consequently, we challenge the neural network by testing the predictions on the test set, described in Sec. 3.4.1, consisting of chopped down versions of the base solids.

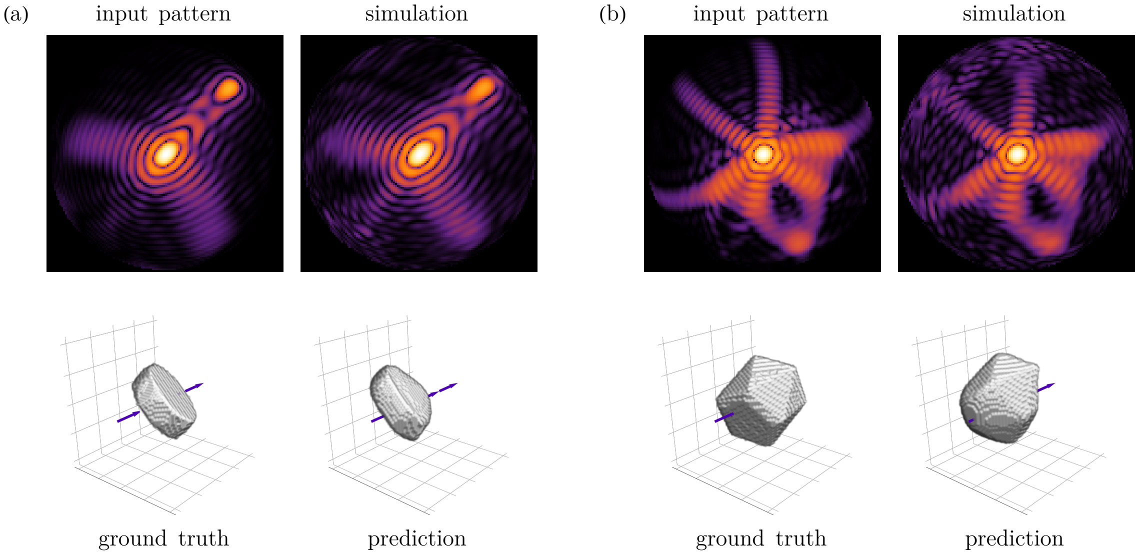


Figure 3.28: For most of the scattering patterns from the test set, the ScatterNet is capable of reconstructing the modified shapes of the real-space objects, like for the heavily shopped rhombicosidodecahedron in panel (a). For some examples, the predicted object is reconstructed without the far side or sports a shallow dome in the beam direction (b), both of which have no significant impact on the scattering pattern.

In the first stage, the network is tested on the unaugmented pristine versions of the test set images. For most of the test samples ScatterNet is capable of detecting the new deformations. An example is shown in Fig. 3.28(a) with a heavily truncated rhombicosidodecahedron. The original object (bottom left) appears almost disc-like and the neural network prediction closely resembles this shape, with slightly smoothed edges. Yet, the corresponding scattering patterns are almost indistinguishable and the additional speckles appearing in the simulation of the prediction can be attributed to artifacts of the decreased object resolution.

The second example in Fig. 3.28(b), a chopped elongated icosahedron, helps in highlighting object defects that are intrinsically connected to the physical loss training. At first sight, the ground truth object (bottom left) and neural network prediction (bottom right) seem to deviate severely. At the same time, the corresponding scattering patterns (top row) are extremely similar and deviate only by speckles. This effect is also observed on the validation set and even the training set and originates in the imbalance in the contributions to the scattering pattern from different parts of the object. Most prominently, the far-side of the prediction in Fig. 3.28(b) is missing entirely. Recalling Secs. 3.1 and 3.4.4 this behavior is to be expected. The incoming intensity reaching the far side is too weak to contribute to the scattering pattern due to the large absorption of silver. Consequently, the ScatterNet has no information on what lies behind the object. Mainly, one of three different behaviors is observed. ScatterNet either performs a symmetric completion (see Fig. 3.28(a)), reconstructs nothing (see Fig. 3.28(b)), or adds a narrowing smooth droplet-profile to the far-side. Another difference between predictions and targets is the relative position of the object inside the real-space volume. The Fourier transform does not conserve the absolute position and the predicted objects are only roughly placed in the vicinity of the volume center. This is different to the ground truth data, which are fixed with their center of mass to the center of the volume and the exact positioning of any prediction varies even between different training runs.

The last important defect visible in Fig. 3.28(b) is the so-called *domeing* on the front-face of the object prediction. Instead of a flat surface of the ground truth object the prediction has a shallow rounded dome. A similar effect was already encountered in Sec. 3.3.4 in the discussion of the icosahedral scattering pattern. Steep angled surfaces, like the mantle sides of an icosahedron, contribute much stronger reflexes to the scattering pattern than shallow angled surfaces. The same effect is demonstrated in Fig. 3.29 for a two dimensional slice of a trapezoid (black). If added a shallow triangle face (orange), the change in the one-dimensional scattering signal profile over the full forward-scattered range is barely visible and easily obfuscated by image defects such as noise or lost in limited detector resolutions. Stronger reflexes of these surfaces could be observed in a full 4π detector configuration in the back-scattered signal. However, this is usually inaccessible at FEL experiments. The effects discussed here are important to keep in mind when further analyzing predictions made by the ScatterNet and should always be considered before making assumptions about predicted objects.

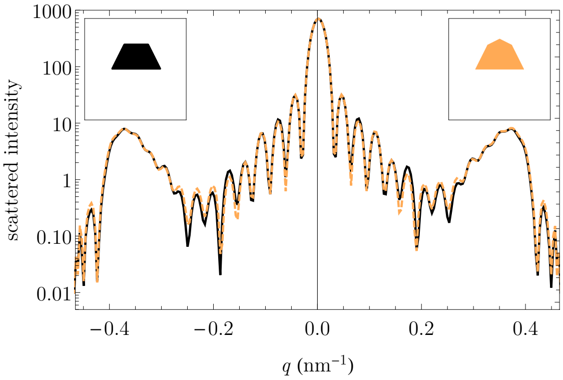


Figure 3.29: The scattered intensity signals of a truncated triangle with a footprint of 212.5 nm and of the same object equipped with a shallow tip of 25% of its height are almost identical.

The difficulty of the reconstruction test is increased by applying augmentations to the test set. The scattering transformation projects small sized and delicate features of the objects to high frequency regions, i.e. large scattering angles. Hence, the first test is to limit the simulated detector range. This is done incrementally through a circular cropping mask. The predictions for a series of decreasing detector ranges are shown in Fig. 3.30. Between full-sized image (top row) and cropping to half detector diameter (center row) the predicted object does not change significantly, indicating that even at half range the image contains enough information about the object. Again, slight domeing is observed on the originally flat surfaces of the polyhedron. Still, it is more pronounced at half detector range. With less than half the detector range, the reconstruction quality deteriorates rapidly and at a quarter angular range (bottom panel) just the general dimensions of the object are recognizable. However, at such extreme view-angle limitations nearly to no wide-angle information is preserved and the scattering transitions to the small-angle regime. Yet, inside the visible region (gray mask) the prediction (right panel) still matches the input pattern to a considerable degree.

In summary, when trained in a classical supervised manner, the generalization capability of SupervisedNet on the test set is still comparable to ScatterNet. The Supervised-

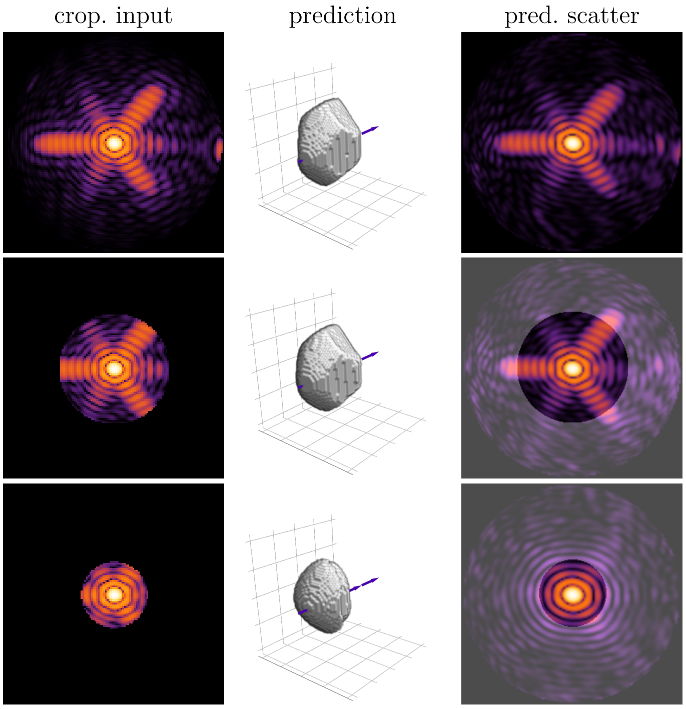


Figure 3.30: Shrinking the angular span of the detection range (left column) leads to the loss of high-frequency information in the scattering pattern. Thus, the neural network predictions (central column) appear less crisp, and corners and edges are rounded, while the corresponding scatter simulation (right column) still matches the input pattern within the input region (framed by gray mask).

NET produces much shaper edges and, enforced by training, also constructs far-sides, see Fig. 3.31(a) and (b). Yet, for some objects, like the chopped, elongated icosahedron in Fig. 3.31 it does not reconstruct the defects but rather recalls the closest unperturbed object from the training set space.

3.4.7. Comparing supervised and physics informed learning

So far, the SCATTERNET did only show a slight superiority over the SUPERVISEDNET in reproducing the input scattering patterns, while returning object predictions with distinct uncertainty elements. However, the superiority of the physics informed training becomes apparent once challenging both networks with conditions similar to that of experimental data. The neural networks are evaluated on the scattering patterns of the test set, augmented with the **simulated experiment** filter from Tab. 3.4. The result for one sample is shown in Fig. 3.32. It is a chopped elongated decahedron, chopped perpendicular to the five-fold symmetry axis at half height. The comparison also includes a HYBRIDNET, trained with the hybrid loss function from Eq. (3.22). Darker regions of the object predictions indicate uncertainty regions, where the prediction is not binary but an intermediate value.

The object prediction made by the SUPERVISEDNET is only superficially similar to the ground truth object and includes large regions with intermediate prediction. Although sporting a pentagonal base, it is rotated perpendicular (by $\pi/5$ rad) to the ground-truth base-pentagon. Moreover, the corresponding scattering pattern bears little resemblance to the input scattering pattern. The given condition clearly supersede the generalization capabilities of the SUPERVISEDNET and it is questionable if it could return usable predictions

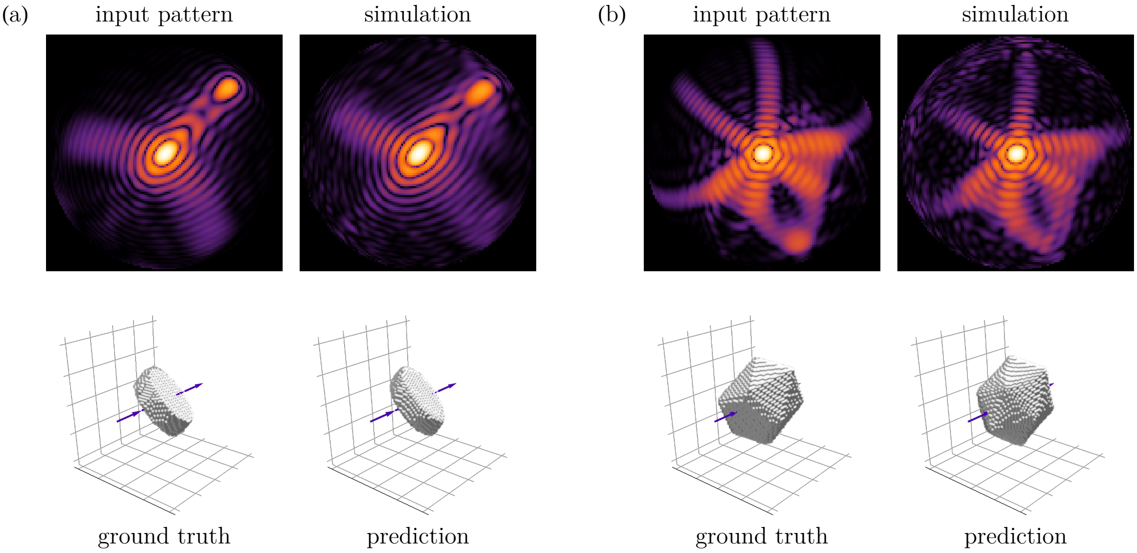


Figure 3.31: Challenged with unperturbed samples of the test set, the SupervisedNet produces much sharper object shapes. The far-sides are usually completed from symmetrically, as visible in both panel (a) and (b). In some cases, like the chopped elongated icosahedron in panel (b), SupervisedNet does not recognize the deformation but rather recalls the closest uperturbed object from the training set space. In this case, a perfect elongated icosahedron.

on real experimental data.

The prediction of the HybridNet in the third column of Fig. 3.32 highlights the instability of the hybrid training approach. The predicted object is that of an un-chopped version of the ground-truth elongated decahedron, surrounded by a funneled halo of non-binary voxels. The resulting scattering pattern closely resembles the input pattern, but the object prediction does not conform with the restrictions to convex, binary objects. Instead, the HybridNet seems to have learned general rules how to crate artifacts that are not penalized enough by both binary regularization and cross-entropy loss while optimizing the scatter loss. To recall, the impact of both scatter loss and cross-entropy is balanced by the weight parameter w_H with the limiting cases of supervised training for $w_H \to \infty$ and purely physics loss training for $w_H \to 0$. Trials showed the object artifacts only disappear if the contribution of the supervised loss is reduced to an effective $w_H \to 0$ case.

Finally, the right column of Fig. 3.32 shows the prediction of the physics loss trained ScatterNet. The predicted objects is subject to the known artifacts. In this sample, the shadow region behind the object is clearly visible as a gray-colored, i.e. non-binary valued, veil. Also, a smooth dome is added to the flat front-face of the object. More importantly, the side-face angles and edge-position match the ground-truth object and the scattering pattern reproduces the input pattern inside the available view-range quite well. The result is not as crisp as for the hybrid approach, but the predicted object fully conforms with the assumed object model. Therefore, the ScatterNet is the only viable candidate for application to real experimental data.

3.4.8. Discrete Density Reconstructions from Experimental Data

After extensive testing on simulated data and understanding the predictions made by ScatterNet, the final challenge is a test on real experimental data. Again, we use the scattering patterns published in Ref. [93] and treat the object candidates obtained by

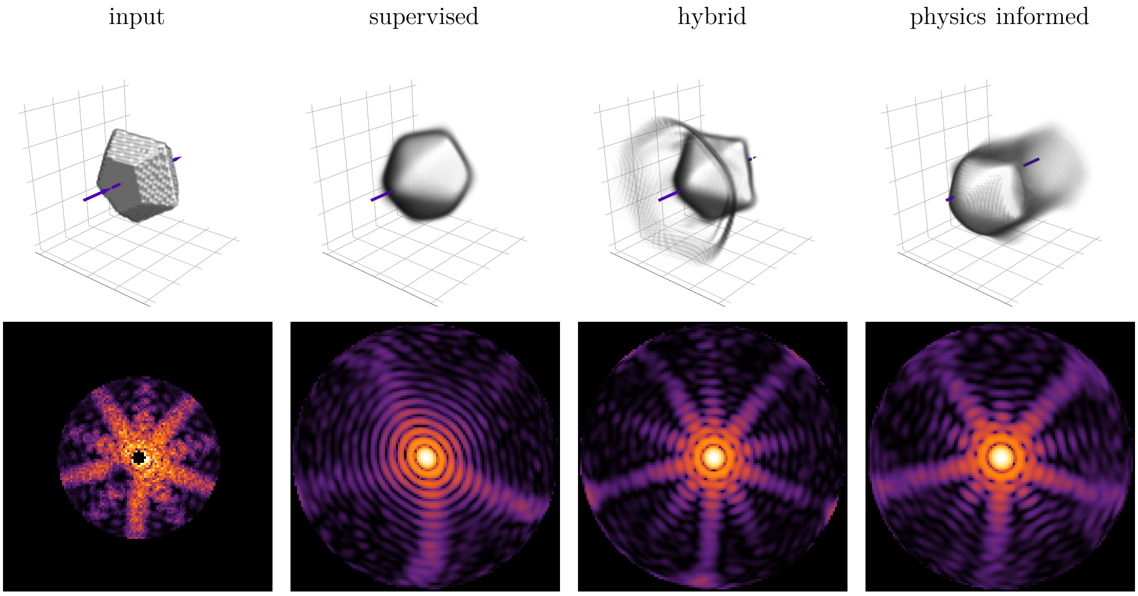


Figure 3.32: The augmented scattering pattern (bottom left) of an elongated decahedron, cut at half distance along the five-fold symmetry axis (top left) is used to test the generalization capabilities of neural networks tested in different training schemes. The upper row shows the object predictions made from the test pattern, while the bottom row illustrates the corresponding scattering patterns.

classical methods as references.

The prediction results are shown in Figs. 3.33 and 3.34. The experimental input patterns (top rows) are re-scaled and zero-padded to conform with the dimensions required by ScatterNet. The reference object candidates from Ref. [93] are placed in the second row in the form of vectorized polyhedra and are scaled to the correct dimensions. The third rows contain the predictions made by ScatterNet. The bottom rows contain simulated scattering patterns, calculated from the ScatterNet predictions. Therein, the available detector region is traced by a gray mask to highlight the comparable regions.

Again, we see the uncertainty defects in the predictions made by ScatterNet. The far-sides of the objects are either missing (e.g. first column in Fig. 3.33 or second column in Fig. 3.34) or droplet shaped (e.g. 2nd column in Fig. 3.33 or third column in Fig. 3.34). Also the previously discussed doming appears, most prominently on the twinned-truncated tetrahedron in the third row of Fig. 3.34. Overall, the neural network predictions appear smoothed with less sharp edges than the reference solids. However, the scattering patterns also contain no information related to such small structures due to the limited angular range of the detector.

Aside from the deviations originating from the different object models, both methods agree quite well on most solids. The main facets are reconstructed reliably, resulting in a similar global structure. However, the neural network predicts some elongated objects, most prominently the decahedron in the first column of Fig. 3.33 and the truncated octahedron in the third column of Fig. 3.34. Moreover, the reconstructed objects are no longer perfectly symmetric, contrary to the assumption of the parametrized model. Through this local defects, also the corresponding simulated scattering patterns are no longer perfectly symmetric. In this, they close to perfectly reproduce the non-symmetric features of the input scattering patterns within the available region. A good example are the slightly curved and narrowed main-reflexes of the icosahedron in the second column of Fig. 3.34.

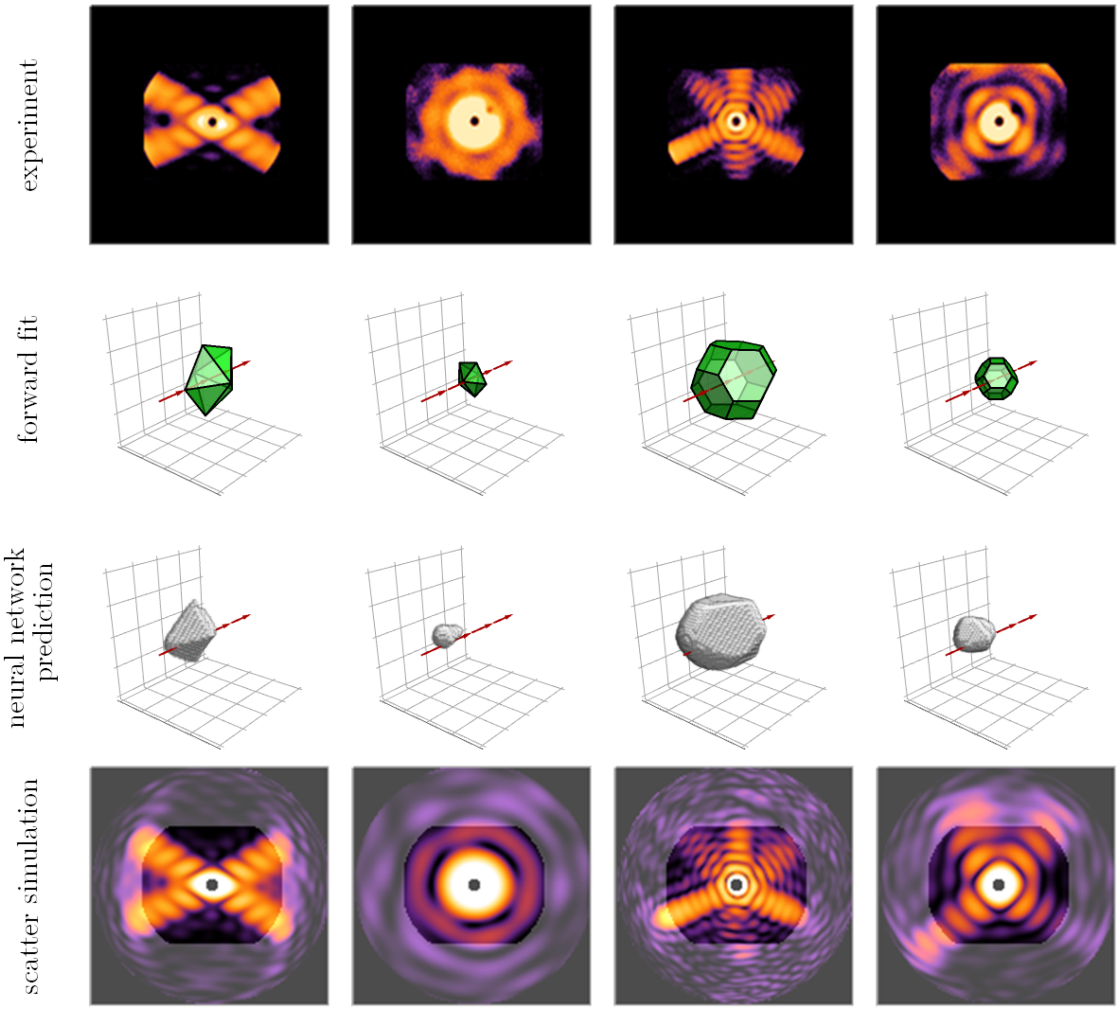


Figure 3.33: The neural network is tested with the first half of the experimental scattering patterns from Ref. [93] (left top row, permitted by Creative Commons CC-BY 4.0 license (`http://creativecommons.org/licenses/by/4.0/`)) and the corresponding shape candidates obtained by forward fitting (second row, green solids). The neural network predictions are shown in gray in the third row. The simulated scattering patterns (bottom row) show excellent agreement with the input pattern inside the available region (confined by the gray masks).

3.4.9. Uncovering Novel Structures of Silver Nanoclusters

The one exception from the overall agreement between classical forward fit and ScatterNet prediction is the star-shaped pattern with five-fold symmetry (first column in Fig. 3.34), which was attributed to an icosahedron in Ref. [93]. It is the very same pattern that had already been discussed in Sec. 3.3.4 and had been consistently identified as tetrahedral by ClassNet. This disagreement was one of the main reasons that led to the development of the discrete reconstruction approach in the first place.

A closer comparison for this specific scattering pattern is given in Fig. 3.35. The icosahedral candidate from Ref. [93] is placed in the left column and the elongated decahedron proposed in Sec. 3.3.4 in the right-most column, with their respective scattering pattern in the center row, each. The prediction from ScatterNet is placed in the second column, just below the experimental pattern. The proposed object clearly has a pentagonal base with approximately triangular front faces that ends in a shallow tip. Different to an icosa-

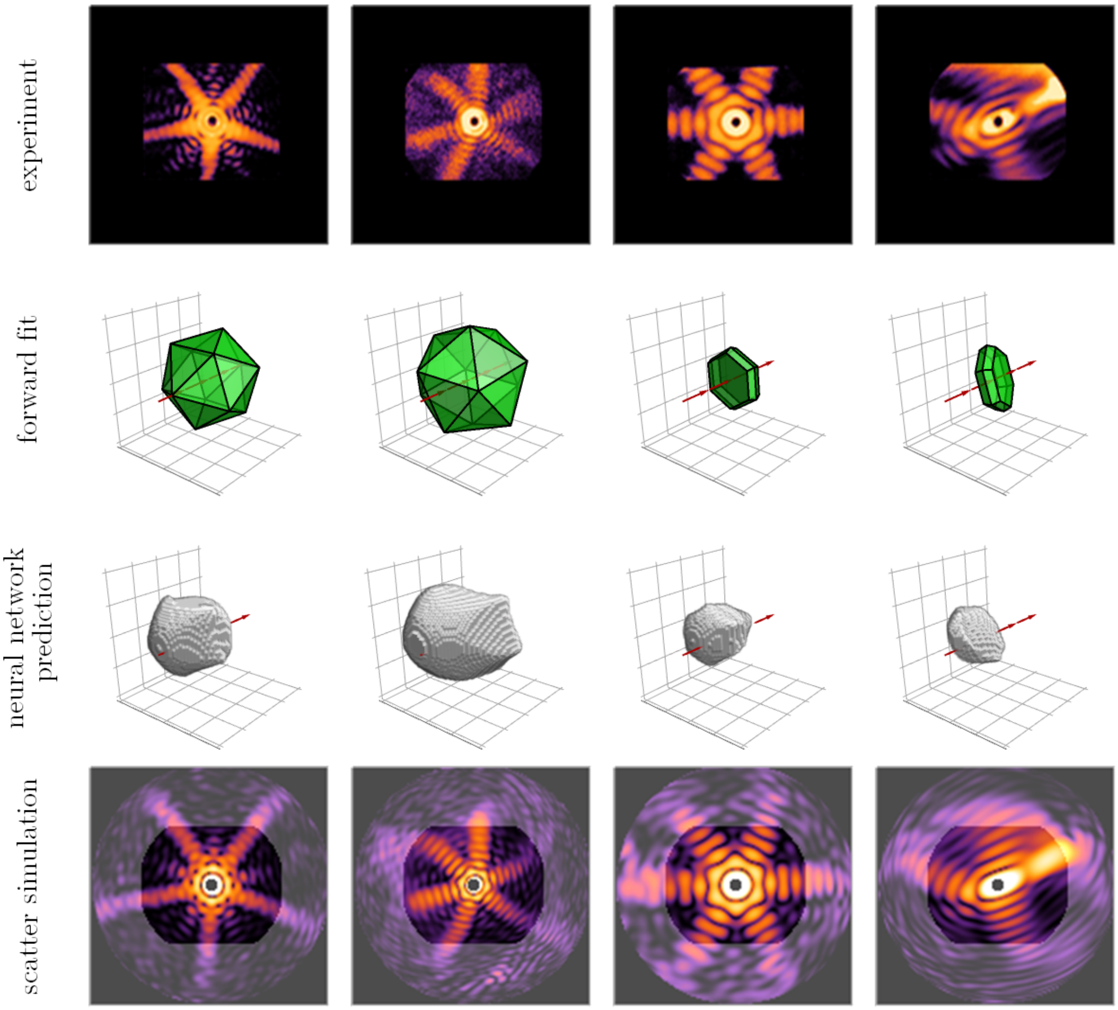


Figure 3.34: The neural network is tested with the second half of the experimental scattering patterns from Ref. [93] (left top row, permitted by Creative Commons CC-BY 4.0 license (`http://creativecommons.org/licenses/by/4.0/`)) and the corresponding shape candidates obtained by forward fitting (second row, green solids). The neural network predictions are shown in gray in the third row. The simulated scattering patterns (bottom row) show excellent agreement with the input pattern inside the available region (confined by the gray masks).

hedron it has no prominent mantle surface but abruptly ends in an amorphous far-side. Consequently, the only surface relevant in the scattering process is the front-facing pentagonal pyramid. The surface angles precisely match those of the elongated decahedron proposed in Sec. 3.3.4, but with a slightly smaller base area and a dull tip. Considering the doming effect, the rounded tip could indicate that the decahedron is chopped, like shown in the third column of Fig. 3.35. Yet, when comparing the scattering patterns of both elongated and chopped elongated decahedron they appear indistinguishable within the available detector region. Hence, they are treated as equal candidates.

To obtain an objective measure of similarity, the mean-squared difference between the simulated scattering patterns of the prediction candidates and the input scattering pattern within the detector region is calculated, i.e. the scatter loss within the detector region. The reference value is provided by the icosahedron with a benchmark value of 7.10×10^{-3}. The elongated decahedra achieve a difference of 4.74×10^{-3} and the ScatterNet prediction reaches an error of 4.63×10^{-3}. Both decahedra and ScatterNet yield a much closer

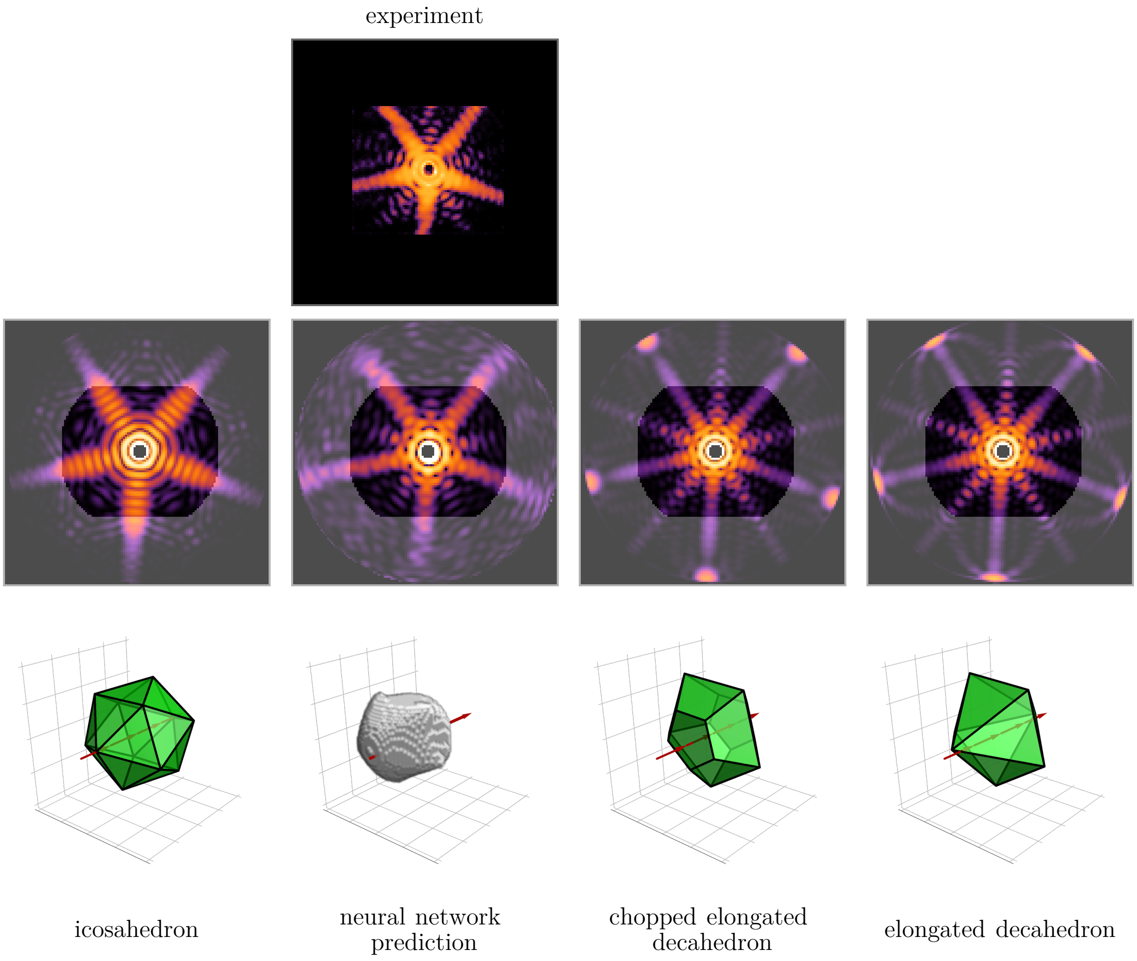


Figure 3.35: The quality of different prediction candidates (bottom row) can be judged by comparing the corresponding scattering patterns (center row) to the experimental input scattering pattern (top panel, taken from Ref. [93], permitted by Creative Commons CC-BY 4.0 license (http://creativecommons.org/licenses/by/4.0/)).

fit than the icosahedron, with ScatterNet slightly ahead. The scattering patterns of all four candidates differ significantly outside the available detector region, traced by the gray mask. This implies that the reconstruction quality is mostly limited by the angular range and could only be improved through using larger detectors.

In summary, ScatterNet achieves a much closer fit than the parametrized forward fits through successful interpolation between the object classes learned during training. The predicted structure on the five-fold scattering pattern resembles an elongated, possibly chopped, decahedron and is a novel observation for silver nanoclusters of the given size, to the best of our knowledge. This result showcases how neural network reconstructions can advance diffractive imaging techniques by uncovering novel and more detailed structures, which currently lie outside the scope of classical reconstruction algorithms. The ScatterNet in its current state is ready for deployment onto larger datasets of scattering patterns currently unidentified. Investigating a broader dataset could potentially reveal even more novel structures and also support the prediction made for the five-fold pattern in Fig. 3.35 if more similar shaped objects could be found in different orientations.

3.5. Deployment and Future Development

We have progressively tackled increasingly harder reconstruction problems for wide-angle scattering patterns with neural networks. The reconstruction of structure parameters with neural networks, presented in Sec. 3.2, matches the accuracy and capability of existing methods, while offering an exceptional speedup. The failed identification of particle shapes in the form of a simple classification highlights how a biased object model can influence the interpretation of scattering patterns. The obvious misclassification of experimental scattering patterns at the end of Sec. 3.3 further demonstrates how features hidden to the human observer can be uncovered by deep learning systems.

The result of this development is the physics informed training of VoxelNet in Sec. 3.4. Given the available reference data, it predicts novel shapes for silver nanoclusters beyond the interpretation scope of classical reconstruction algorithms. The neural network, as is, is ready for deployment to a larger dataset of currently unidentified data. An investigation of a broad dataset could support and refine the new object candidates through patterns from similar structures in varying orientations or even reveal more unexpected and remarkable shapes. Despite the complexity of the neural network, the evaluation time on sufficient hardware is still in the µs range. In this it paves the way to a fully automated reconstruction of the complete structure of nanoparticles from single-shot wide-angle scattering images in real time, even with the increased repetition rate of modern FEL experiments [173]. Still, there are some additions and changes to different aspects of the architecture and training of ScatterNet that could improve the prediction quality even further.

Dataset

The set of base solids described in Sec. 3.4.1 already covers a broad range of convex solids and the results on the test set have proven the capability of the neural network to handle unknown shapes. However, directly including the chopping modifications of the test set will always improve the generalization capability through covering an even larger object space during training. Aside from the Platonic and Archimedean Solids, the basis set includes some singular shapes, i.e. the twinned truncated tetrahedron and decahedron (or pentagonal dipyrimid), that currently do not fall into a general class of objects. Yet, both can be generalized into a single object class, that of the chopped, elongated n-gonal bipyramids (see top row first in Fig. 3.36). Similar to the pentagonal bipyramids, these solids are formed from a regular polygon with n vertices as a basis, connecting to vertices above and below the center to form n-faced pyramids on both sides. The truncated twinned tetrahedron is then created by chopping a hexagonal bipyramid along the pyramidal axis. Also, further object classes can be derived from chopped versions of the existing base solids. The mantle section of a rhombicuboctahedron is an eight-fold prism (top row second in Fig. 3.36), while the mantle-section of an icosahedron is a ten-fold antiprism (top row third in Fig. 3.36). Aside from bipyramids, prisms and antiprisms, another large class of regular polyhedra are the Catalan solids. These are the dual solids to the Archimedean solids and depicted in the bottom two rows of Fig. 3.36. Both highly symmetric regular convex polyhedra, the Archimedean solids are constructed from different regular n-gonal polyhedra and have identical vertices, while each Catalan solid is constructed from repetitions of only one (not regular) polygon and in return sport different classes of vertices. Extending the basis set in such a way would greatly extend the object space cover during training and expose the neural network to even more structures and facet-reflexes. Diversifying the training set also simplifies the transfer of the existing neural network to other materials.

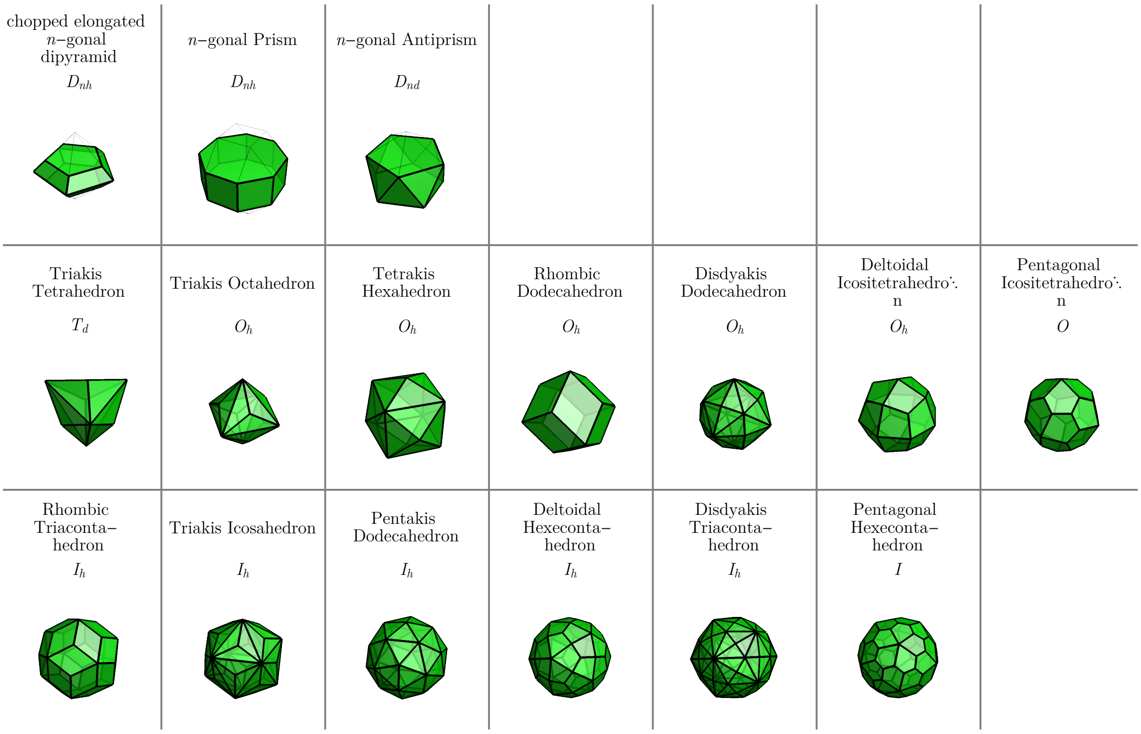


Figure 3.36: Some of the existing objects in the train- and test-set can be generalized into larger classes of objects. Both decahedron and truncated twinned tetrahedron are special cases of the chopped elongated n-gonal dipyramid (top row, first). The mantle faces of a rhombicubocahedron, isolated by chopping, form a octagonal prism (top row, second), that generalizes to the class of n-gonal prisms. Similar, the mantle faces of an icosahedron (top row, third) belongs to the n-gonal antiprisms. Another class of regular convex polyhedra are the Catalan solids (center and bottom rows), that are the dual solids to the Archimedean solids.

Scattering Algorithm

The MSFT algorithm with an effective absorption model approximates the scattering off silver nanoparticles to a high level of detail. This was shown in Ref. [93] through comparisons with finite-difference time-domain (FDTD) simulations respecting the full light-matter interaction. Still, the higher accuracy comes at the cost of extensive computation times. Since time is a crucial factor when analyzing (or generating) large datasets, the small errors made by MSFT might be acceptable. For physics informed learning, MSFT also is favorable as it translates into a single tensor operation on a three-dimensional object tensor projected into a three dimensional Fourier-space and then compressed into a two-dimensional image. While keeping the simple calculation structure, the MSFT algorithm can be improved to also include multiple scatterings of the signal through the propagation MSFT (pMSFT) algorithm [177]. The existing TensorFlow code could easily be modified to pMSFT if required.

Prediction Resolution

The resolution of 64 × 64 × 64 voxel of the predicted object tensor is a trade-off between resource demand for training and sufficient detail of the object. The reconstruction quality on view-range limited scattering patterns demonstrates that the existing resolution is sufficient given the information content of the scattering patterns. For simulation, however, the limited resolution is responsible for artifact. This can be indentified from Fig. 3.37

where the same object is simulated on different object resolutions. With the general concept proven in this thesis, it would be beneficial to increase the output resolution to at least 128 or even 256 voxels in each dimensions. The demands in computational power and especially VRAM space are easily met with the current generation of graphics card or specialized tensor cards. In the current implementation, the artifacts in the scattering patterns due to limited resolutions do not lead to false reconstructions. This is because both prediction and target tensor are scatter-simulated on the same resolution and the resolution artifacts can give away additional orientation information on the particle. They will, however, become problematic when dropping the technically redundant scattering simulation of the target tensor.

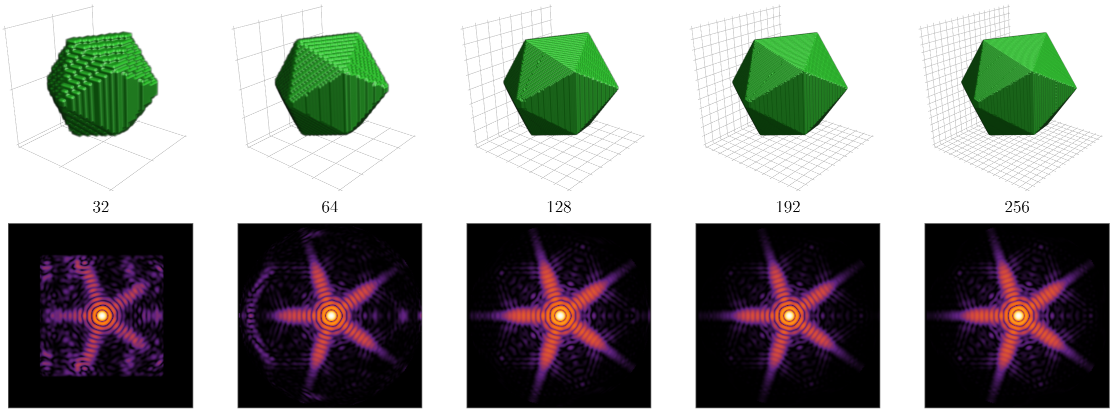


Figure 3.37: At lower object resolutions, like 32 or 64 voxels in length, the scattering simulates produces significant artifacts, that break the symmetry inherited from the object shape, in this case an icosahedron. At a resolution of 128 voyels, the result is already comparable to the quality of the dataset generated with 192 voxels in length. An increase to 256 voxel does not appear to yield any more improvements.

Unsupervised Learning

The physics informed training as described in Sec. 3.4 still is a supervised training in that it requires both input scattering patterns as well as the corresponding target object. Interestingly, including the scattering simulation into the loss function also allows the implementation of an unsupervised training, as sketched in Fig. 3.38, without the need for ground-truth object tensors. In this configuration, the loss functions needs further modifications. First, it need to be handed a mask-function, indication which portion of the image are to be reconstructions as it would otherwise be forced to create scattering patterns that reproduce the beamstop-artifact. Second, the loss function also needs to be indifferent to noise appearing in the input pattern and other defects introduced by augmentation. Without the need for ground-truth data and learning solely from the scattering algorithm, the unsupervised training could even be performed directly on experimental data. It would require exact knowledge of the detector characteristics and artifacts appearing in the experiment. Theoretically, this identification of common features could also be implemented into a neural network, as done with the generative network in Ref. [85] for wide-angle scattering patterns of Helium nanodroplets.

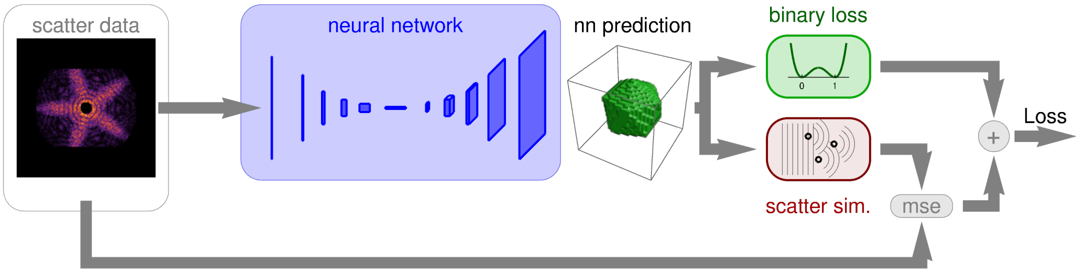


Figure 3.38: The physics informed training of the discrete cluster reconstruction neural network also enables unsupervised training, where the need for a ground-truth object is eliminated. It requires further modification to the mean-squared-error (mse) comparison to make it robust against artifacts on the input scattering pattern.

4. Excitonic Giant Dipole States in Cuprous Oxide

The boundaries between the microscopic and macroscopic world were always of special interest for all sciences. It is here were some effects of one world have to wane off and other effects appear. The silver nanoclusters analyzed in the previous chapter fall in this regime. Over their growth, they cross the boundary between a small molecule and a bulk solid, both favoring very different crystal structures and geometries. At an even smaller scale, Rydberg states of atoms reach onto the border between quantum and classical world,[1] where the wave function of the highly excited atom behaves almost like a classical Bohr radius [181]. Molecules formed from just one Rydberg atoms with one ground state atoms reach such sizes, that they appear from the outside as classical particles, e.g. in having a quantum-mechanically forbidden permanent electric dipole moment [182, 183]. Another kind of atomic states where electron and core are still bound (but separated over macroscopic distances) are the so-called giant dipole (GD) states [184–187]. GD states are predicted to form in the presence of extreme external electromagnetic states and thus could not be observed directly up to this day. This could change with the recent advances in the excitations of Ryberg states of excitons in cuprous oxide (Cu_2O) [188]. These excitons experience a relative amplification of external fields, putting GD states into realistic range of existing experiments [189–192]. The macroscopic distance of a GD states prevents a direct optical excitation. Instead, a dynamic excitation scheme is required [193]. Meanwhile, the higher field-sensitivity of excitons forbids the usage of existing proposals, developed for atomic beam experiments. Thus, a new excitation scheme adapted to the special conditions of the excitonic system needs to be developed [192].

Motivated by the recent advances in the control of physical systems with neural networks and the success in inverting scattering patterns in Chap. 3, we aim to develop this excitation scheme with deep reinforcement learning. Related to previous works [193], this is done in a semi-classical approach by the simulation of classical trajectories. Throughout the past, the existence of GD states has been debated controversially. The history of giant dipole states is reflected in Sec. 4.1, together with the current state of Rydberg exciton research in cuprous oxide. Further, the derivation of the giant dipole Hamiltonian for arbitrary hydrogenic systems is documented in a closed-form in Sec. 4.2 and transferred to the excitonic system. It is the foundation for the simulation environment, described in Sec. 4.3, required for the training of a reinforcement learning agent in Sec. 4.4. The results and insights gained from this approach are discussed in Sec. 4.5.

[1] Some quantum effects also manifest over macroscopic ranges, like entanglement [178] or the Casimir effect [179, 180].

4.1. Permanent Atomic-Level Dipole Moments

A Rydberg state of an atoms is characterized by one valence electron being excited to very large principal quantum numbers n. They earn their name from approximately conforming with the Rydberg formula of the transition wavelength between atomic states [194]

$$\frac{1}{\lambda} = R\left(\frac{1}{n_1^2} - \frac{1}{n_2^2}\right) , \tag{4.1}$$

originally derived to describe the spectrum of hydrogen. The spectrum of any other element with more than one electron usually is much more complex due to interactions between the electrons. However, if only one electron is excited to very large principal quantum numbers n this interaction becomes less important as the spatial distribution overlaps less with the inner electrons. With the limited interaction, the system becomes almost hydrogen-like. When further increasing the angular quantum number l the wave function can be completely expelled from the inner shell region and becomes circular, centered around the classical Bohr orbits.

Nowadays, Rydberg atoms are not only subject of research on their own but also a tool for high precision measurements of matter-light interaction. Maybe the most prominent example are the direct measurement of the quantized nature of the light field in a cavity through Rabi oscillations between neighboring circular Rydberg states [195], and observing the life cycle of a single photon inside a cavity by the group of Serge Haroche [196]. These groud-breaking experiments led to the awarding of the 2012's Nobel prize in physics to Serge Haroche and David Wineland [197]. Other applications of Rydberg atoms as tools of measurements include but are not limited to trace gas detection [198], cooling of molecular gases [199], orientation control of polar molecules [200], and the study of macroscopic quantum effects like the Casimir-Polder effect [5].

Polar Rydberg Molecules

Experiments of such precision are only possible due to the fine control of Rydberg states. Currently, the highest excitable states in rubidium are just above a principal quantum number of $n = 200$, where the expectation value of the orbits radius even exceeds the dimensions of the cloud of ultra-cold atomic vapor from which it is excited [201]. The corresponding spectra show, aside from the main excitation peak, also features of the interaction of the Rydberg electron with every other ground state atom of the cloud [202]. The first signs of interactions of Rydberg states with ground states atoms were discovered simultaneously in the 1930 by Christian Füchtbauer at the University of Rostock [203–205] and Eduardo Amaldi and Emilio Segrè in Rome [206, 207]. They reported line shifts in both the red- and blue-regime depending on the type of surrounding gas and density, which, at the time, came as a surprise. The corresponding theory was developed by Enrico Fermi, based upon the scattering of the Rydberg electron off a ground state perturber atom [208].

Further advances in theory and experiments, revealed the possibility of the formation of so-called Rydberg molecules from the scattering-interaction as a bound state between a Rydberg atom and a ground atom [182]. While ordinary binuclear molecules have binding lengths in the order of 0.1 nm, like e.g. water [209], the inter-nuclear distance of Rydberg molecules is several hundred times larger, in the order of 50 nm. In the simplest approximation, Rydberg molecules fall into two classes. For states with small angular momentum quantum numbers l, i.e. S, P, and D-states, molecules with binding lengths of several hundred Bohr-radii are predicted and measured [210]. Although intrinsically neutral, they also possess a small permanent dipole moment due to admixtures of the second type of molecular states [211].

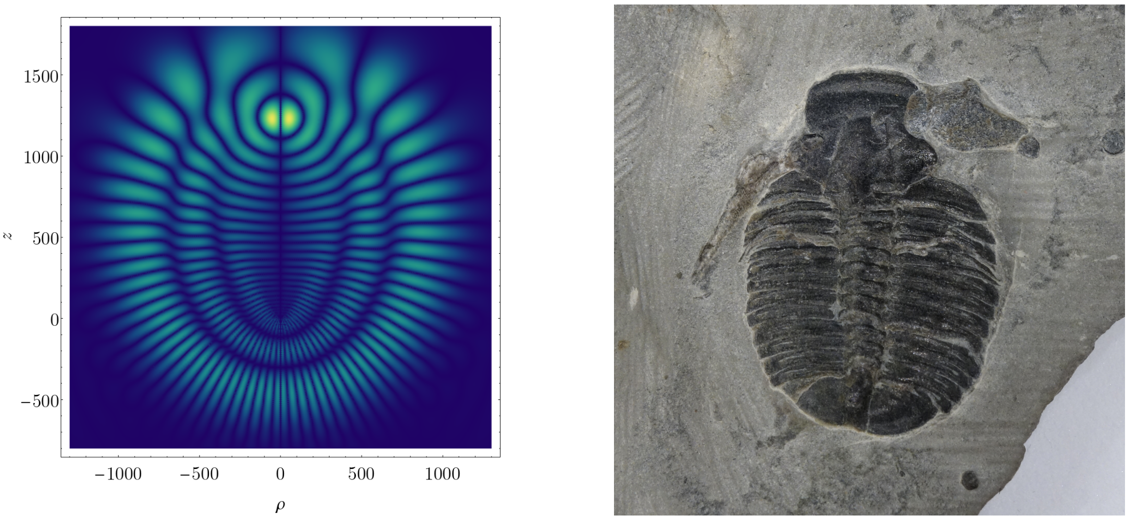


Figure 4.1: The electron density $\rho|\psi(\rho, z, 0)|$ in cylindrical coordinates of a trilobite state (left) is highly concentrated around the perturber atom, here located at the potential minimum of the $n = 30$ high-l manifold at $z = 1232\,a_0$. Within the ρ-z-plane, the density function appears oval shaped with regular ripples, similar to the appearance of a trilobite fossil (right), thus earning the name of trilobite-states. The trilobite specimen shown here is of the species *elrathia kingii* with an age of approximately 520 million years (private collection).

This second type of Rydberg molecules arises from a superposition of high-l Rydberg states, that are all energetically degenerate due to vanishing quantum defects. The result is a concentration of the wave-function of the Rydberg electron in a fringe pattern, plotted in Fig. 4.1, with a strong concentration around the perturber atom at a distance of 1000s Bohr radii. The pattern of the wave function strongly reminds of a trilobite fossil, henceforth, they were named trilobites states. The high angular momentum makes excitation considerably harder, which could only be achieved just recently [212]. For trilobite molecules, the intra-nuclear distance is so large, that the period time of any rotation mode is longer than the state-lifetime [183]. As a result, the role of Rydberg atom and perturber can not be switched through exchange interaction and the polarity is conserved. Combined with the concentration of the electronic wave function around the wave-function, an excessively large permanent dipole moment, several orders of magnitude stronger than in any other known molecule. Ordinary ultracold heteronuclear polar molecules [213, 214] (such as K-Rb) already find application not only as microscopic antennas but also for the control of chemical reactions [215] and the considerably larger dipole moments of trilobite states may open routes to even more discoveries [211, 216–218].

Giant Dipole States

The separation distance of Rydberg-electron and ionic-core of a trilobite molecule stretches over macroscopic lengths. There exists, at least theoretically, an even larger kind of separated polar atomic state: the giant dipole (GD) states. Instead of a perturber, the separation in a GD state is realized through interaction with strong external crossed electric and magnetic field. In this configuration, a free electron would travel in a spiral trajectory, like the one in Fig. 4.2. The Hamiltonians of the electric and magnetic field interactions, on the other hand, are linear and parabolic, respectively. Thus, for sufficiently large field

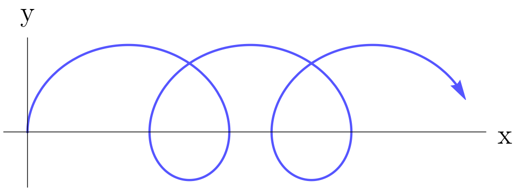


Figure 4.2: Example trajectory for a positively charged particle inside crossed homogeneous electric and magnetic fields with the magnetic field pointing in z-direction and electric field in y-direction.

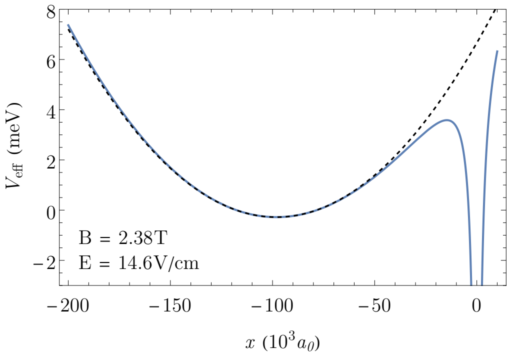


Figure 4.3: The effective single particle potential of a hydrogen atom in crossed electric and magnetic fields ($\boldsymbol{E}$-field in x-direction, $\boldsymbol{B}$-field in z-direction) is a superposition of the Coulomb-potential around $\boldsymbol{r} = 0$, a linear Stark-term and an effective parabolic potential from the magnetic field coupling. For suitable field strengths, a local parabolic minimum can form at distances of $100\,000\,a_0$. After Ref. [187].

strengths, it would be possible to create a parabolic potential well outside the Coulomb-potential of an atom, like plotted in Fig. 4.3. The exact physics is, however, not as simple, as one would hope for.

The theoretical foundations of giant dipole states have been discovered at least two times, independently. The earliest publication by scientists from the Leningrad State University dates back to 1976 [184]. They derived an atomic potential with an outer harmonic minimum for a hydrogenic system in crossed homogeneous electric and magnetic fields under symmetric gauge and for a static proton of infinite mass. Aside from atomic hydrogen, which at the time was not yet available for experiments in crossed fields, they also considered excitons in semiconductors as a possible route for the realization of giant dipole states. The same derivation was also published independently in 1979 [219]. However, in these early works the position of the potential minimum depends on the choice of gauge, and thus directly violates the gauge freedom of electromagnetism. Accordingly, the magnetic field can not impose a localized potential on the electron and the giant dipole minimum should not exist. This was argued in Ref. [220] and in their place the quasi-Penning resonances or Stark-saddle states were proposed. They are localized above the local maximum in Fig. 4.3 at the point where the Coulomb attraction and external electric force cancel each other and are stabilized by the magnetic field interaction, similar to ions in a Penning-trap [220–222].

Despite the fundamental theoretical differences between giant dipole states and Stark-saddle states they both share the fundamental property of being field-separated states of ion-electron-pairs with a large directed separation and hence a permanent dipole moment. A first attempt of measuring such states was made in 1987 with a cold-atom beam experiment with rubidium [223]. It was set up with an inhomogeneous electric field, that should deflect bound states with a permanent electric dipole moment in an angle and allow detection and calculation of the dipole moment. Although a deflected beam was detected, the signal was weak and no statement about the nature of the underlying states could be made. Shortly after, the first experiment with an atomic hydrogen beam in crossed electric and magnetic fields with a similar setup was realized [224]. Here, a manifold of states in the right energy regime appeared which coincide with closed-loop, polar classical trajectories. Yet still, no direct evidence for atomic states with a permanent dipole moment was found. Improvements on the 1987 experiment with rubidium atoms led to the, up to this date, most precise measurements of polar atomic states, which were submitted in 1991 and published in 1993 [225]. By measuring the signal of the deflected beam via an MCP detector, recall Sec. 3.1, with angular resolution the appearance and position of a beam deflections could be detected for excitations around the ionization threshold. Also, the dependency of the deflection and conversely the induced dipole moment could be shown. On the downside, the experimental results were interpreted with the theory of a static proton of infinite mass and fixed gauge, just like in Ref. [184]. This led to an intense debate with alternative interpretations of the measurement results [226].

Parallel to this controversy, a new theory of giant dipole states was developed, dropping all approximations in also including the finite mass of the proton. This had previously been neglected, in order to eliminate the center-of-mass dependency from the system Hamiltonian and yielding a solvable single-particle problem. An exact solution is, however, possible by using the pseudomomentum as a constant of motion of atoms in homogeneous electromagnetic fields [227]. Including the coupling of the proton to the magnetic field results in the appearance of a small effective gauge-free harmonic potential, which was first demonstrated in Ref. [185] through introducing gauge-independent canonic variables. A more descriptive derivation was published shortly after in 1993, identifying the kinetic terms of the pseudo-separated Hamiltonian and thereby extracting the harmonic potential contribution to the effective potential [186]. A follow-up publication demonstrated the complete gauge-independence of this effective single-particle Hamiltonian and derived the quantum-mechanic spectrum of states bound in the local minimum [187]. This effective harmonic potential arrives as a residual from the coupling difference between electron and proton to the magnetic field. Thus, the effective potential is much weaker, resulting in only loosely bound giant dipole states are localized at much greater distances. Ref. [187] reports binding lengths of around 100 000 Bohr radii (plotted in Fig. 4.3), which in turn may explain the lack of polar states in previous states such as Ref. [224], as the overlap of such states with a ground-state wave-function and tunneling probabilities for excited states are effectively non-existing. An alternative route of excitation was described in Ref. [193] for atomic beam experiments. By exciting an initial state in a magnetic-field region without an electric field, the latter can be increased along the propagation axis of the beam, resulting in an effective time-profile of the electric field in the frame of the atom. While ramping up the electric field, the electron can propagate outwards over the forming saddle point and may get trapped once the full internal barrier between giant-dipole minimum and Coulomb-well is build up. By simulating classical trajectories, an optimal profile for the electric field could be found, that, however, was never implemented in an experiment.

The electron in a giant dipole states experiences only a very shallow potential with even weaker kinematics than in a Rydberg states. Hence, if the giant dipole electron meets a perturber atom, the interaction meets the same conditions as in Rydberg molecules [228]. This is a non-negligible scenario as the the giant dipole wave function spans over a considerable volume, especially in the unrestricted magnetic field direction. In case of an attractive interaction, as the triplet S-wave interaction for rubidium, the formation of Rydberg-like molecules could help stabilizing giant dipole states [4, 228]. From Rydberg molecules it is known, that the interaction between electron and perturber depends strongly on the relative orientation of the electronic spins in singlet or triplet configuration. In the presence of magnetic fields, singlet and triplet combinations are no pure eigenstates anymore, as both spins couple to the magnetic field independently [229]. As giant dipole states can only exist in very strong magnetic field, the effect is even stronger, separating the molecular potentials into four distinct branches of the distinct spin configurations ($|\downarrow\downarrow\rangle$,$|\downarrow\uparrow\rangle$,$|\uparrow\downarrow\rangle$,$|\uparrow\uparrow\rangle$) [3]. From these, the parallel eigenstates observe only triplet character, while the anti-parallel eigenstates couple equally to the singlet- and triplet-scattering channel. Still, all of these branches support bound molecular states, which could indeed aid in stabilizing possible giant dipole states.

As of today, no clear and direct detection of giant dipole states could be achieved. A major problem stays that of the excitation, as the giant dipole ground state predicted in Ref. [187] is located at an energy equivalent to a principal quantum number of $n \approx 200$, which even nowadays is the highest state excited in a controlled way in a field-free environment [201]. Nevertheless, to this problem could be overcome by the use of excitons, like initially proposed in Ref. [184].

Rydberg Excitons in Cupruos Oxide

An exciton inside a semiconductor material is a quasi-particle, which is formed as a bound state of an electron and a hole. The hole itself is a quasi particle, too, as it represents a missing electron inside the valence band. Yet, the collective motion of the residual electron enables the description as a positively charged quasi-particle. Excitons in bulk semiconductors can be approximated as a hydrogen-like system with the effective masses of electron and hole replacing the masses of free electron and proton and the Coulomb potential attenuated by the relative permittivity, respectively. In consequence, the exciton is described with the same quantum theory as the hydrogen atom, but the associated Rydberg energy is much lower (for example 4.2 meV in gallium arsenide GaAs), while the coupling to external fields is nearly unchanged or even amplified through the lower effective mass. Unfortunately, the smaller energy spacing together with the intrinsic linewidth usually limits the number of resolvable exciton states to low quantum numbers, like $n = 3$ for GaAs [230]. Since the hole represents a missing electron, the effective hole mass usually is of similar magnitude to the electron mass, which is at a stark contrast to atoms. Hence, when discussing exciton dynamics, the approximation of the internal motion as mainly the electron's motion is no longer valid and always has to be processed as true two-particle motion.

Excitons in bulk semiconductor materials were first observed in 1956 in cuprous oxide Cu_2O [233][2]. Cuprous oxide, also called *cuprite* in mineralogy, is a rare occurring oxide of copper. It forms red crystals (see Fig. 4.4), that are also used as gemstones. The lattice structure of Cu_2O (see Fig. 4.5) is cubic, with a bcc lattice of O-atoms embedded into an fcc lattice of Cu-atoms, shifted by $1/4$ along the cell diagonal. So far, synthesizing Cu_2O proves challenging and the purest and best samples are still obtained from a few naturally formed crystals [234, 235]. The band structure of Cu_2O supports four series of excitons between the closest valence- and conduction bands, labeled after their corresponding wavelenths as yellow, green, blue and violet series [236]. In most cases, the yellow exciton series of Cu_2O with a gap energy of $E_g = 2.17\,\text{eV}$ is investigated. The first high Rydberg states of excitons were discovered in the very same material, when the original experiment was repeated with modern equipment as part of a Bachelor's thesis in Dortmund. Subsequent measurements resolved the exciton series up to a quantum number of $n = 25$, which laid the foundations

[2] The publication reports observations of exciton states up to $n = 8$ in a liquid helium cooled cryostate at $T = 1.3\,\text{K}$. Modern experiment are conducted at similar temperatures [188]. Nevertheless, states up to $n = 6$ are already visible with a liquid nitrogen cooling at $T = 77.3\,\text{K}$ [233].

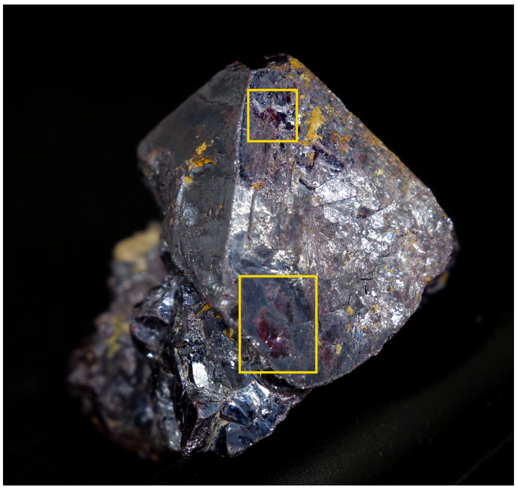


Figure 4.4: A natural cuprite crystal from the Altai region, Russia (private collection). The large crystal has octahedral shape, which is a shape from the cubic symmetry group O_h. Due to impurities, a large portion of the crystal has a silvery metallic shine, which is a common feature of cuprites from the Rubtsovsk mine [231, 232]. The yellow rectangles mark relatively pure cuprous oxide crystallites with the characteristic translucent ruby color.

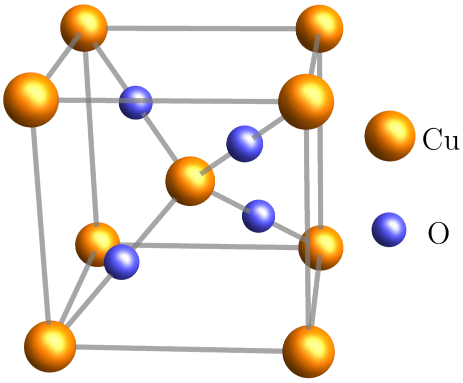


Figure 4.5: The lattice structure of cuprous oxide Cu_2O is cubic (space group O_h), composed from a fcc Cu-sublattice with an embedded bcc O-sublattice shifted a quarter length along the cell diagonal.

of Rydberg exciton science [188]. This observation exceeded older experiment with $n = 8$ [233] or $n = 12$ [237] by far and was only superseded just recently with resonances up to $n = 28$ with improved cooling [238] and $n = 30$ inside a different natural crystal [239]. The fact that excitons are quasi-particles implies some differences to atoms. Firstly, the $1S$ state, which is the ground state of a hydrogen atom, is just the lowest exciton state but not the ground state. Instead, the ground state of the semiconductor is the excitonic vacuum, where the valence band is completely filled, and from which an exciton is created by excitation. Secondly, defining the effective mass for the electron and hole depends on the parabolicity of the conduction and valence band, respectively. This however, is only true to a certain degree of approximation around the Γ-point, as the band structure of Cu_2O is much more complex [240]. This in turn leads to deviations from the n^{-2} series of the energy levels. Still, the deviations are only minor and can be treated by introducing a quantum defect $\delta_{n,l}$ to the principal quantum number $n^* = n + \delta_{n,l}$ [241]. This is similar to the treatment of atoms, where the quantum defect arises from the interactions with the lower closed electron shells [181]. Thirdly, the semiconductor environment allows for more decay channels than in atoms, since electron and hole are an anti-particle pair. For Cu_2O, the decay is dominated by scattering with optical phonons which mediate transitions into low lying states such as the the$1s$ state [242], limiting the lifetimes to few nanoseconds [188].

Because of the smaller binding energies and lower effective masses, the relative coupling of excitonic states to external electric and magnetic fields is amplified, compared to atoms [243]. Already with magnetic fields up to 7 T, the Landau-regime can be reached, where the diamagnetic coupling dominates the Hamiltonian and all anti-unitary symmetries are broken [244]. The quasi-Landau levels that appear in this regime even above the band-gap (which is equivalent to the classical ionization threshold) can be observed up to a principal quantum number of $n = 50$ [189]. Also the polarizing effects of external electric fields with strengths up to 2.6 kV/cm have been studied with the observation of strong Stark-splitting bridging the gap between neighboring n-levels and leading to state crossings [190, 245]. The strong couplings even allow for a direct observation of the influence of the pseudomomentum on the internal exciton dynamics through the magneto-Stark effect [246]. In this case, if the pseudomomentum is non-vanishing, the motion through the magnetic field induces an effective electric field, acting on the internal dynamics [191]. The availability of both Rydberg states, relative field amplifications and strong fields already implemented offer a perfect environment for a realization of giant dipole potentials. In fact, theoretical calculations indicate the possibility of the formation of excitonic giant dipole potentials at energies equivalent to $n = 6$ with reasonable field strengths [192]. Yet, through the coupling of the band structure to the external fields, these potentials are highly dependent on the field orientations relative to the crystal axes. While the separation length of giant dipole states at low energy levels is much smaller, the energy space is also less dense populated as for high quantum numbers. Therefore, relevant resonances are much easier to identify, compared with the dense spectra observed in atom experiments in Refs. [223–225].

Despite these favorable conditions for the creation of giant dipole potentials, the question how to excite a giant dipole state with the wave function located in the outer potential well remains unsolved. As the exciton is created from the electron being excited from the hole position, there is no spatial overlap with the giant dipole wave function, just like with atomic ground states. Further, the lifetimes of excitons, shortened by the rapid phonon-scattering decay [242], may be too short for tunneling of an excited state into the giant dipole well to occur. Therefore, a similar solution as proposed in Ref. [193] may be required, where an initial state is transferred into a giant dipole state by temporal variation of the external electric field. However, whereas Ref. [193] aimed for a highly controlled, slow transition, the transitions needs to be performed as swiftly as possible. Recently, deep reinforcement learning algorithms have outperformed human control and imagination not only in games such as *Go* [10, 11] but also in the optimization of experimental setups like the control of a MOT trap [50]. In the previous chapter it was shown how neural networks can master even complex non-unique inversion problems. Thus, we implement a reinforcement learning agent for finding an optimal transition scheme for giant dipole states of excitons. Reinforcement learning algorithms are notorious for their unstable learning behavior. Therefore, it is advised to start with the simplified task of the hydrogenic approximation of the exciton.

4.2. The Giant Dipole Hamiltonian

The controversy around the existence of giant dipole states has continued, in parts, up to this date. It is further not helped by the lack of any direct experimental evidence. Moreover, the complete derivation of the giant dipole potential in its full form is distributed over several publications. Therefore, in the following, we will repeat the complete derivation of the GD Hamiltonian, including all crucial steps which currently can only be found distributed over Refs. [186, 187, 246]. Further, the derivation is executed in full SI units rather than in atomic units. Although it is possible to define system-specific atomic units, such as excitonic atomic units as derived in App. C.1, carrying over all constants aids in highlighting the small but important differences arising between hydrogen atoms and excitons. We then transfer this Hamiltonian to the hydrogenic model of excitons in cuprous oxide and identify possible start- and target-state for giant dipole transitions.

4.2.1. Derivation of the Giant Dipole Hamiltonian

In the following, we assume a system of two oppositely charged particles, e and p, from which are the negatively and positively charged constituents of the system, respectively. Identified by the sign of their charge, e is the electron-like particle with mass m_e and charge $q_\mathrm{e} = -q_0$ and p is the proton-like particle with mass m_p and $q_\mathrm{p} = +q_0$. We will further assume a surrounding medium with arbitrary permittivity $\varepsilon = \varepsilon_\mathrm{r}\,\varepsilon_0$. In accordance with existing literature, we refrain from explicitly marking quantum mechanical operators unless it is necessary to avoid confusion with their classical counterparts. Starting from the minimal coupling Hamiltonian [246, 247],

$$\mathcal{H} = \frac{1}{2m}\left(\boldsymbol{p} - q\,\boldsymbol{A}(\boldsymbol{r})\right)^2 - q\,\Phi(\boldsymbol{r}) \tag{4.2}$$

of a spin-free charged particle, the full system Hamiltonian reads

$$\mathcal{H} = \frac{1}{2m_\mathrm{p}}\left(\boldsymbol{p}_\mathrm{p} - q_0\,\boldsymbol{A}(\boldsymbol{r}_p)\right)^2 + \frac{1}{2m_\mathrm{e}}\left(\boldsymbol{p}_\mathrm{e} + q_0\,\boldsymbol{A}(\boldsymbol{r}_e)\right)^2 + q_0\,\phi\left(\boldsymbol{r}_\mathrm{e} - \boldsymbol{r}_\mathrm{p}\right) - \frac{{q_0}^2}{4\pi\varepsilon}\frac{1}{|\boldsymbol{r}_\mathrm{e} - \boldsymbol{r}_\mathrm{p}|}. \tag{4.3}$$

We further assume constant external fields $\boldsymbol{E}$, $\boldsymbol{B}$. The choice of gauge is left open, by including an arbitrary gauge field $\Lambda(\boldsymbol{r})$ into the symmetric gauge. Hence, the electromagnetic potentials are defined as

$$\boldsymbol{A}(\boldsymbol{r}) = \frac{1}{2}\boldsymbol{B} \times \boldsymbol{r} + \boldsymbol{\nabla}\Lambda(\boldsymbol{r})\,, \tag{4.4}$$

$$\phi(\boldsymbol{r}) = \boldsymbol{E} \cdot \boldsymbol{r}\,. \tag{4.5}$$

Inserting these potentials into the Hamiltonian (4.3) results in a Hamiltonian

$$\begin{aligned}\mathcal{H} =& \frac{1}{2m_\mathrm{p}}\left(\boldsymbol{p}_\mathrm{p} - \frac{q_0}{2}\boldsymbol{B} \times \boldsymbol{r}_\mathrm{p} - q_0\boldsymbol{\nabla}_{\boldsymbol{r}_\mathrm{p}}\Lambda(\boldsymbol{r}_p)\right)^2 \\ &+ \frac{1}{2m_\mathrm{e}}\left(\boldsymbol{p}_\mathrm{e} + \frac{q_0}{2}\boldsymbol{B} \times \boldsymbol{r}_\mathrm{e} + q_0\boldsymbol{\nabla}_{\boldsymbol{r}_e}\Lambda(\boldsymbol{r}_\mathrm{e})\right)^2 + q_0\,\boldsymbol{E}\left(\boldsymbol{r}_\mathrm{e} - \boldsymbol{r}_\mathrm{p}\right) - \frac{{q_0}^2}{4\pi\varepsilon}\frac{1}{|\boldsymbol{r}_\mathrm{e} - \boldsymbol{r}_\mathrm{p}|}\,,\end{aligned} \tag{4.6}$$

of two particles coupled to the external fields $\boldsymbol{E}$, $\boldsymbol{B}$. The first step for the transformation into a single-particle problem of the internal dynamics is the introduction of the center-of-mass coordinates

$$\boldsymbol{R} = \frac{m_\mathrm{p}\,\boldsymbol{r}_\mathrm{p} + m_\mathrm{e}\,\boldsymbol{r}_\mathrm{e}}{M}\,, \tag{4.7}$$

$$\boldsymbol{P} = M\,\boldsymbol{R}\,, \tag{4.8}$$

with the total mass $M = m_\mathrm{e} + m_\mathrm{p}$, and relative coordinates

$$\boldsymbol{r} = \boldsymbol{r}_\mathrm{e} - \boldsymbol{r}_\mathrm{p}\,, \tag{4.9}$$

$$\boldsymbol{p} = \mu \boldsymbol{r}\,, \tag{4.10}$$

with the reduced mass $\mu = (m_\mathrm{p}\,m_\mathrm{e})/M$. The coordinate transformation $\{\boldsymbol{r}_\mathrm{p}, \boldsymbol{r}_\mathrm{e}\} \to \{\boldsymbol{R}, \boldsymbol{r}\}$ transforms the gradient operators

$$\boldsymbol{\nabla}_{\boldsymbol{r}_\mathrm{p}} = \frac{\partial \boldsymbol{R}}{\partial \boldsymbol{r}_\mathrm{p}} \frac{\partial}{\partial \boldsymbol{R}} + \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{r}_\mathrm{p}} \frac{\partial}{\partial \boldsymbol{r}} = \frac{m_\mathrm{p}}{M} \boldsymbol{\nabla}_{\boldsymbol{R}} - \boldsymbol{\nabla}_{\boldsymbol{r}} \tag{4.11}$$

$$\boldsymbol{\nabla}_{\boldsymbol{r}_\mathrm{e}} = \frac{m_\mathrm{e}}{M} \boldsymbol{\nabla}_{\boldsymbol{R}} + \boldsymbol{\nabla}_{\boldsymbol{r}} \tag{4.12}$$

By further defining $\Lambda_\mathrm{p} = \Lambda(\boldsymbol{R} - \boldsymbol{r}\,m_\mathrm{e}/M)$ and $\Lambda_\mathrm{e} = \Lambda(\boldsymbol{R} + \boldsymbol{r}\,m_\mathrm{p}/M)$, the Hamiltonian (4.6) is transformed into

$$\begin{aligned}\mathcal{H} \;=&\; \frac{1}{2m_\mathrm{p}} \left(\frac{m_\mathrm{p}}{M} \boldsymbol{P} - \boldsymbol{p} - \frac{q_0}{2} \boldsymbol{B} \times \left(\boldsymbol{R} - \frac{m_\mathrm{e}}{M} \boldsymbol{r} \right) - q_0 \left(\frac{m_\mathrm{p}}{M} \boldsymbol{\nabla}_{\boldsymbol{R}} - \boldsymbol{\nabla}_{\boldsymbol{r}} \right) \Lambda_\mathrm{p} \right)^2 \\ &+ \frac{1}{2m_\mathrm{e}} \left(\frac{m_\mathrm{e}}{M} \boldsymbol{P} + \boldsymbol{p} + \frac{q_0}{2} \boldsymbol{B} \times \left(\boldsymbol{R} + \frac{m_\mathrm{p}}{M} \boldsymbol{r} \right) + q_0 \left(\frac{m_\mathrm{e}}{M} \boldsymbol{\nabla}_{\boldsymbol{R}} + \boldsymbol{\nabla}_{\boldsymbol{r}} \right) \Lambda_\mathrm{p} \right)^2 \\ &+ q_0\,\boldsymbol{E}\,(\boldsymbol{r}) - \frac{{q_0}^2}{4\pi\varepsilon} \frac{1}{|\boldsymbol{r}|}\end{aligned} \tag{4.13}$$

Due to the coupling terms between the canonical momentum operators and the magnetic field inside the quadratic brackets, the center-of-mass momentum $\boldsymbol{P}$ is not a preserved quantity and hence can not be traced out of the Hamiltonian. Instead, for a given particle i of charge q_i the pseudomomentum

$$\boldsymbol{k}_i = m_i \dot{\boldsymbol{r}}_i + q_i\,\boldsymbol{B} \times \boldsymbol{r}_i = \boldsymbol{p}_i - q_i\,\boldsymbol{A}(\boldsymbol{r}_i) + q_i\,\boldsymbol{B} \times \boldsymbol{r}_i \tag{4.14}$$

is the preserved quantity of the motion inside a constant magnetic field [227, 246] and commutes with the mechanical momentum $(\boldsymbol{p}_i - q_i\,\boldsymbol{A}(\boldsymbol{r}_i))$. As shown in App. C.2, the pseudomomentum can be interpreted as a quantity similar to the mean value of the total momentum $\boldsymbol{P}$, although this interpretation is not exactly true (see Ref. [246] and references therein), since the value of the pseudomomentum is gauge dependent. For systems with vanishing net charge $Q = \sum_i q_i = 0$ the total pseudomomentum $\boldsymbol{K} = \sum_i \boldsymbol{k}_i$ is a preserved quantity of the system Hamiltonian [227]. In our case, the total pseudomomentum reads

$$\boldsymbol{K} = \boldsymbol{P} - \frac{q_0}{2} \boldsymbol{B} \times \boldsymbol{r} - q_0 \left(\left(\frac{m_\mathrm{p}}{M} \boldsymbol{\nabla}_{\boldsymbol{R}} - \boldsymbol{\nabla}_{\boldsymbol{r}} \right) \Lambda_\mathrm{p} - \left(\frac{m_\mathrm{e}}{M} \boldsymbol{\nabla}_{\boldsymbol{R}} + \boldsymbol{\nabla}_{\boldsymbol{r}} \right) \Lambda_\mathrm{e} \right)\,. \tag{4.15}$$

This allows an ansatz that separates the eigenfunctions of $\mathcal{H}$ into a unitary part depending on the center-of-mass coordinates $\boldsymbol{R}$ and a purely internal part:

$$\psi_{\boldsymbol{r},\boldsymbol{R}} = U(\boldsymbol{r}, \boldsymbol{R})\,\psi_0(\boldsymbol{r})\,. \tag{4.16}$$

Since $\boldsymbol{K}$ is a preserved quantity of $\mathcal{H}$, they need to share a common spectrum. Hence the eigenequation of the operator $\hat{\boldsymbol{K}}\,\psi(\boldsymbol{r},\boldsymbol{R}) = \boldsymbol{K}\,\psi(\boldsymbol{r},\boldsymbol{R})$ must hold true. As $\boldsymbol{K}$ only includes $\boldsymbol{P}$, the same equation $\hat{\boldsymbol{K}}\,U(\boldsymbol{r},\boldsymbol{R}) = \boldsymbol{K}\,U(\boldsymbol{r},\boldsymbol{R})$ applies to the unitary part. This can be fulfilled by the ansatz [187]

$$U(\boldsymbol{r},\boldsymbol{R}) = \exp\left\{\frac{\mathrm{i}}{\hbar}\left(\boldsymbol{K}\cdot\boldsymbol{R} + \frac{q_0}{2}\left(\boldsymbol{B}\times\boldsymbol{r}\right)\cdot\boldsymbol{R} + \chi(\boldsymbol{r},\boldsymbol{R})\right)\right\}\,, \tag{4.17}$$

which leads to

$$\hat{\boldsymbol{P}}U(\boldsymbol{r},\boldsymbol{R}) = -\mathrm{i}\hbar\boldsymbol{\nabla}_{\boldsymbol{R}}U(\boldsymbol{r},\boldsymbol{R}) = \boldsymbol{K} + \frac{q_0}{2}\boldsymbol{B}\times\boldsymbol{r} + \boldsymbol{\nabla}_{\boldsymbol{r}}\chi(\boldsymbol{r},\boldsymbol{R}) \tag{4.18}$$

when applied to the momentum operator $\hat{\boldsymbol{P}}$. Consequently, the function $\chi(\boldsymbol{r},\boldsymbol{R})$ is determined by the gauge terms in Eq. (4.15) through the differential equation

$$\boldsymbol{\nabla}_{\boldsymbol{r}}\chi(\boldsymbol{r},\boldsymbol{R}) = q_0\left(\left(\frac{m_\mathrm{p}}{M}\boldsymbol{\nabla}_{\boldsymbol{R}} - \boldsymbol{\nabla}_{\boldsymbol{r}}\right)\Lambda_\mathrm{p} - \left(\frac{m_\mathrm{e}}{M}\boldsymbol{\nabla}_{\boldsymbol{R}} + \boldsymbol{\nabla}_{\boldsymbol{r}}\right)\Lambda_\mathrm{e}\right)\,. \tag{4.19}$$

Noting the relation of $\boldsymbol{\nabla}_{\boldsymbol{r}}\Lambda_\mathrm{p} = -\frac{m_\mathrm{e}}{M}\boldsymbol{\nabla}_{\boldsymbol{R}}\Lambda_\mathrm{p}$ and $\boldsymbol{\nabla}_{\boldsymbol{r}}\Lambda_\mathrm{e} = \frac{m_\mathrm{p}}{M}\boldsymbol{\nabla}_{\boldsymbol{R}}\Lambda_\mathrm{p}$, the differential equation (4.19) is reduced to contain only derivatives in $\boldsymbol{R}$ as

$$\boldsymbol{\nabla}_{\boldsymbol{r}}\chi(\boldsymbol{r},\boldsymbol{R}) = q_0\boldsymbol{\nabla}_{\boldsymbol{R}}\left(\lambda_\mathrm{p} - \Lambda_\mathrm{e}\right)\,. \tag{4.20}$$

It is solved by

$$\chi(\boldsymbol{r},\boldsymbol{R}) = q_0\left(\lambda_\mathrm{p} - \Lambda_\mathrm{e}\right) + f(\boldsymbol{r})\,, \tag{4.21}$$

which is fully determined up to an internal gauge function $f(\boldsymbol{r})$ [187]. Tracing out $U^\dagger(\boldsymbol{r},\boldsymbol{R})\,\mathcal{H}\,U(\boldsymbol{r},\boldsymbol{R})$ leads to the cancellation of all gauge dependent terms Λ_p and Λ_e in Eq. (4.13), resulting in the effective Hamiltonian of the internal motion

$$\begin{aligned}\mathcal{H}_\mathrm{eff} =& \frac{1}{2m_\mathrm{p}}\left(\frac{m_\mathrm{p}}{M}\boldsymbol{K} + \frac{q_0}{2}\boldsymbol{B}\times\boldsymbol{r} - \boldsymbol{p} - \boldsymbol{\nabla}_{\boldsymbol{r}}f(\boldsymbol{r})\right)^2 \\ &+ \frac{1}{2m_\mathrm{e}}\left(\frac{m_\mathrm{e}}{M}\boldsymbol{K} + \frac{q_0}{2}\boldsymbol{B}\times\boldsymbol{r} + \boldsymbol{p} + \boldsymbol{\nabla}_{\boldsymbol{r}}f(\boldsymbol{r})\right)^2 + q_0\,\boldsymbol{E}\left(\boldsymbol{r}\right) - \frac{{q_0}^2}{4\pi\varepsilon}\frac{1}{|\boldsymbol{r}|}\,.\end{aligned} \tag{4.22}$$

By expanding the quadratic brackets and using the reduced masses $\mu = (m_\mathrm{p}\,m_\mathrm{e})/(m_\mathrm{p}+m_\mathrm{e})$ and $\tilde{\mu} = (m_\mathrm{p}\,m_\mathrm{e})/(m_\mathrm{p} - m_\mathrm{e})$ the effective Hamiltonian can be rewritten as

$$\begin{aligned}\mathcal{H}_\mathrm{eff} =& \frac{1}{2\mu}\Bigg(p^2 + \frac{\mu}{\tilde{\mu}}\frac{q_0}{2}\left(\left(\boldsymbol{B}\times\boldsymbol{r}\right)\cdot\boldsymbol{p} + \boldsymbol{p}\cdot\left(\boldsymbol{B}\times\boldsymbol{r}\right)\right) + \left(\frac{q_0}{2}\boldsymbol{B}\times\boldsymbol{r}\right)^2 \\ &+ q_0\left(\boldsymbol{B}\times\boldsymbol{r}\right)\cdot\boldsymbol{\nabla}_{\boldsymbol{r}}f(\boldsymbol{r}) + \left(\boldsymbol{p}\cdot\boldsymbol{\nabla}_{\boldsymbol{r}}f(\boldsymbol{r}) + \boldsymbol{\nabla}_{\boldsymbol{r}}f(\boldsymbol{r})\cdot\boldsymbol{p}\right) + \left(\boldsymbol{\nabla}_{\boldsymbol{r}}f(\boldsymbol{r})\right)^2\Bigg) \\ &+ \frac{K^2}{2M} + q_0\left(\frac{\boldsymbol{K}\times\boldsymbol{B}}{M} + \boldsymbol{E}\right)\cdot r - \frac{{q_0}^2}{4\pi\varepsilon}\frac{1}{|\boldsymbol{r}|}\,,\end{aligned} \tag{4.23}$$

while respecting the order of non-commuting pairs of operators like $[\hat{\boldsymbol{r}},\hat{\boldsymbol{p}}] \neq 0$.

It is worth noting, that by fixing the gauge by assuming $f(\boldsymbol{r}) \equiv 0$ and using the definition of the projection angular momentum operator $\hat{L}_{\boldsymbol{n}} = \frac{1}{2}\frac{\boldsymbol{n}}{|\boldsymbol{n}|}\cdot\left(\boldsymbol{r}\times\boldsymbol{p} + \boldsymbol{p}\times\boldsymbol{r}\right)$ the

Hamiltonian (4.23) reduces to

$$\mathcal{H}_{\text{eff}} = \frac{1}{2\mu}\left(p^2 + \frac{\mu}{\tilde{\mu}} q_0\, B\, L_{\boldsymbol{B}} + \left(\frac{q_0}{2}\boldsymbol{B}\times\boldsymbol{r}\right)^2\right) + \frac{|\boldsymbol{K}|^2}{2M} + q_0\left(\frac{\boldsymbol{K}\times\boldsymbol{B}}{M} + \boldsymbol{E}\right)\cdot\boldsymbol{r} - \frac{{q_0}^2}{4\pi\varepsilon}\frac{1}{|\boldsymbol{r}|}\,, \tag{4.24}$$

which is exactly the form described in the older Refs. [184, 219, 225].

4.2.2. Identifying the Giant Dipole Potential

We are interested in finding a closed form respecting any gauge freedom, which is not fulfilled with Eq. (4.24). Therefore, we need to find a simplification of Hamiltonian (4.22) with all contributions of the internal gauge function $f(\boldsymbol{r})$. This can be done by exploiting $1/\mu = \mu/\tilde{\mu}^2 + 4/M$ to split the term proportional to $(\boldsymbol{B}\times\boldsymbol{r})^2$ in Eq. (4.23) in order to form two new quadratic brackets

$$\mathcal{H}_{\text{eff}} = \frac{1}{2\mu}\left(\boldsymbol{p} + \frac{\mu}{\tilde{\mu}}\frac{q_0}{2}\boldsymbol{B}\times\boldsymbol{r} + \boldsymbol{\nabla}_{\boldsymbol{r}} f(\boldsymbol{r})\right)^2 + \frac{1}{2M}\left(\boldsymbol{K} + q_0\boldsymbol{B}\times\boldsymbol{r}\right)^2 + q_0\boldsymbol{E}\cdot\boldsymbol{r} - \frac{{q_0}^2}{4\pi\varepsilon}\frac{1}{|\boldsymbol{r}|}\,, \tag{4.25}$$

which is the very same form as derived in Ref. [187]. The first term of this Hamiltonian bears a striking resemblance to the kinetic energy terms in Eq. (4.6). That this term is in fact the kinetic energy term of the internal motion can be deduced in the classical equivalent of the Ehrenfest theorem [248], using the Hamiltonian equations of motion

$$\frac{\partial}{\partial t}x_i = \frac{\partial\mathcal{H}}{\partial p_i}\,, \tag{4.26}$$

$$\frac{\partial}{\partial t}p_i = -\frac{\partial\mathcal{H}}{\partial x_i}\,. \tag{4.27}$$

For the expectation values of $\mathcal{H}_{\text{eff}}$ we arrive at [186]

$$\frac{\partial}{\partial t}\boldsymbol{R} = \frac{\partial\mathcal{H}_{\text{eff}}}{\partial\boldsymbol{K}} = \frac{1}{M}\left(\boldsymbol{K} + q_0\boldsymbol{B}\times\right)\,, \tag{4.28}$$

$$\frac{\partial}{\partial t}\boldsymbol{K} = -\frac{\partial\mathcal{H}_{\text{eff}}}{\partial\boldsymbol{R}} = 0\,, \tag{4.29}$$

$$\frac{\partial}{\partial t}\boldsymbol{r} = \frac{\partial\mathcal{H}_{\text{eff}}}{\partial\boldsymbol{p}} = \frac{1}{\mu}\left(\boldsymbol{p} + \frac{\mu}{\tilde{\mu}}\frac{q_0}{2}\boldsymbol{B}\times\boldsymbol{r} + \boldsymbol{\nabla}_{\boldsymbol{r}} f(\boldsymbol{r})\right)\,, \tag{4.30}$$

$$\begin{aligned}\frac{\partial}{\partial t}\boldsymbol{p} = -\frac{\partial\mathcal{H}_{\text{eff}}}{\partial\boldsymbol{r}} &= -\frac{1}{2\mu}\left(\boldsymbol{p} + \frac{\mu}{\tilde{\mu}}\frac{q_0}{2}\boldsymbol{B}\times\boldsymbol{r} + \boldsymbol{\nabla}_{\boldsymbol{r}} f(\boldsymbol{r})\right)\Delta_{\boldsymbol{r}} f(\boldsymbol{r}) \\ &\quad + \frac{q_0^2}{M}\boldsymbol{B}\times\boldsymbol{B}\times\boldsymbol{r} + q_0\left(\frac{\boldsymbol{K}\times\boldsymbol{B}}{M} + \boldsymbol{E}\right) - \frac{q_0^2}{4\pi\varepsilon}\frac{\boldsymbol{r}}{|\boldsymbol{r}|^3}\,.\end{aligned} \tag{4.31}$$

In turn, the kinetic energy is defined through the canonical velocities $\dot{x}_i$ as $\mathcal{T}_i = \frac{1}{2}m_i{\dot{x}_i}^2$. Combined with the result of the Hamiltonian equation of motion in Eq. (4.30) it allows to specify the kinetic energy of the internal motion of $\mathcal{H}_{\text{eff}}$ as

$$\mathcal{T}_{\text{int}} = \frac{1}{2}\mu\left(\frac{\partial\boldsymbol{r}}{\partial t}\right)^2 = \frac{1}{2\mu}\left(\boldsymbol{p} + \frac{\mu}{\tilde{\mu}}\frac{q_0}{2}\boldsymbol{B}\times\boldsymbol{r} + \boldsymbol{\nabla}_{\boldsymbol{r}} f(\boldsymbol{r})\right)^2\,, \tag{4.32}$$

which exactly is the first term of the Hamiltonian (4.25) [186]. It has the same form as the kinematic term of the minimal coupling Hamiltonian in Eq. (4.2) for a single particle of effective charge $q_{\text{eff}} = -q_0\,\mu/\tilde{\mu}$ but with the vector potential $\boldsymbol{A}(\boldsymbol{r}) = \frac{1}{2}\boldsymbol{B}\times r - 1/q\boldsymbol{\nabla}_{\boldsymbol{r}} f(\boldsymbol{r})$.

The full Hamiltonian $\mathcal{H}_{\text{eff}}$ can then be understood as describing such a particle with effective charge q_{eff} inside a constant magnetic field that is further subject to an effective potential $\mathcal{V}_{\text{eff}}$ [187]. This effective internal potential $\mathcal{V}_{\text{eff}}$ can be determined as

$$\begin{aligned}\mathcal{V}_{\text{eff}} = \mathcal{H}_{\text{eff}} - \mathcal{T}_{\text{int}} &= \frac{1}{2M}\left(\boldsymbol{K} + q_0 \boldsymbol{B} \times \boldsymbol{r}\right)^2 + q_0 \boldsymbol{E}\cdot\boldsymbol{r} - \frac{{q_0}^2}{4\pi\varepsilon}\frac{1}{|\boldsymbol{r}|} \qquad (4.33)\\ &= \frac{K^2}{2M} + \frac{q_0^2}{2M}\left(\boldsymbol{B}\times\boldsymbol{r}\right)^2 + q_0\left(\frac{\boldsymbol{K}\times\boldsymbol{B}}{M} + \boldsymbol{E}\right)\cdot\boldsymbol{r} - \frac{{q_0}^2}{4\pi\varepsilon}\frac{1}{|\boldsymbol{r}|}\,. \qquad (4.34)\end{aligned}$$

The effective potential contains both terms in the order of $\boldsymbol{r}^2$ and $\boldsymbol{r}$. It is explicitly dependent on $\boldsymbol{B}$ but also completely independent of the gauge of the vector potential. Most importantly, it may possess a local parabolic minimum for specific combinations of the fields $\boldsymbol{B}$, $\boldsymbol{E}$ and the pseudomomentum $\boldsymbol{K}$. In Eq. (4.34) the bracket $((\boldsymbol{K}\times\boldsymbol{B})/M + \boldsymbol{E})$ also illustrates the action of the magnetic field as an effective electric field for non-vanishing pseudomomentum. This is the so-called motional Stark-effect [246]. Due to his equivalence, some authors, such as in Ref. [187], introduce a modified pseudomomentum

$$\boldsymbol{K}' = \boldsymbol{K} - M\,\boldsymbol{v}_{\text{D}}\,, \qquad (4.35)$$

with the drift velocity $\boldsymbol{v}_{\text{D}} = (\boldsymbol{E}\times\boldsymbol{B})/B^2$, to eliminate the $\boldsymbol{E}$-dependency from Eq. (4.34). Yet, in this work, we keep both $\boldsymbol{K}$ and $\boldsymbol{E}$ as separate parameters and we will later set $\boldsymbol{K} \equiv 0$.

4.2.3. Excitonic Giant Dipole States

As discussed previously, excitons in cuprous oxide can be approximated as a hydrogenic system [188]. Excitons in bulk semiconductors are of the Wannier-type, where the distance between the constituents spans over several elementary cells. Hence, the Coulomb interaction can be modeled as traversing the bulk semiconductor as a background dielectric medium with relative permittivity $\varepsilon_r = 7.5$ in Cu_2O. The second main difference is the mass of the positively charged component, which for an exciton is the hole whose effective mass is determined by the curvature of the valence band [241]. For the yellow series of Cu_2O, it assumes the value of $m_p = 0.58\,m_0$, while the electron mass of $m_e = 0.99\,m_0$ remains nearly unchanged. The Rydberg energy is

$$Ry^* = \frac{\mu\, e^4}{2\,(4\,\pi\,\varepsilon_0\,\varepsilon_r)^2\hbar^2} = 88.5\,\text{meV}\,. \qquad (4.36)$$

A fit to the raw experimental data yields a Rydberg energy of 92 meV [188], whereas respecting deviations originating from the anisotropic band structure in the form of a quantum defect gives 86 meV [241]. The value in Eq. (4.36) lies in between these fit results and is used throughout this chapter for self-consistency.

A measure for the relative strength of the magnetic field interaction is the quotient $\gamma = (\hbar\omega_c)/(2\,Ry^*)$ between the Rydberg energy Ry^* and the Landau-spacing of the non-interacting system inside the present magnetic field with cyclotron frequency $\omega_c = e\,B/\mu$. The value of γ is an indicator for the coupling regime with $\gamma \ll 1$ indicating a perturbative coupling for weak $\boldsymbol{B}$-fields while $\gamma \gg 1$ indicates the quasi-Landau regime [249]. This quotient between hydrogenic and Landau energy spacing holds for both atoms and excitons. In turn, γ may also be used to calculate the relative field amplification that the excitonic system experiences compared to the atom. With

$$\gamma = \frac{2\,(4\,\pi\,\varepsilon_0\,\varepsilon_r)^2\hbar^2}{\mu\,e^4}\frac{\hbar\,e\,B}{\mu} = \frac{2(4\,\pi\,\varepsilon_0)^2\hbar^3}{e^4 m_0^2}B\,\frac{m_0^2\,\varepsilon_r^2}{\mu^2} = \gamma_0\,\underbrace{\frac{m_0^2\,\varepsilon_r^2}{\mu^2}}_{s_B} \tag{4.37}$$

and the reduced mass $\mu = 0.366\,m_0$ of the excitonic system, the amplification factor is $s_B = 420$. In essence, for a given magnetic field strength B, the dynamics of the excitonic system are equivalent to the hydrogen atom exposed to a field of $420 \times B$.

When considering the electric field scaling, one has to distinguish between the external (vacuum) field E_{vac} and internal field $E_{\text{int}} = \varepsilon_r\,E_{\text{vac}}$ due to the dielectric polarization. Unless otherwise stated, the internal electric field $\boldsymbol{E}_{\text{int}}$ is used throughout this chapter and labeled as $\boldsymbol{E}$. The scaling factor for the internal electric field $s_E = m_0\varepsilon_r^2/\mu = 154$ is defined in the same manner as Eq. (4.37). On the experimental side, realizations of magnetic fields up to 7 T are published [189, 244], as well as external electric fields up to 2.7 kV/cm [190], that translate into internal fields of 356 V/cm.

Aside from the magnetic field $\boldsymbol{B}$ and electric field $\boldsymbol{E}$, the existence and shape of a giant dipole potential in Eq. (4.34) further depends on and the pseudomomentum $\boldsymbol{K}$. Through conversation of momentum, an optically excited exciton is equipped with a pseudomomentum equal to the photon's momentum [191]. The magneto-Stark and Stark term in Eq. (4.34)

$$\mathcal{H}_{\text{S}} = q_0\left(\frac{\boldsymbol{K}\times\boldsymbol{B}}{M} + \boldsymbol{E}\right)\cdot\boldsymbol{r} \tag{4.38}$$

depends through the cross product solely on the projection $\boldsymbol{K}_\perp$ of the pseudomomentum perpendicular to the magnetic field. Hence, if the magnetic field and optical axis are aligned parallel (also called Faraday configuration [191]), we can assume $K \equiv 0$, similar to the case in atomic beam experiments [193]. The opposite case is the Voigt configuration, where the exciting photon travels perpendicular to the magnetic field. With an intermediate wave number of $2.79 \times 10^7\,\text{m}^{-1}$ for the yellow series of Cu_2O [191], the effective magneto-Stark field strength is

$$E_{\text{mS}} = \frac{K_\perp\,B}{M} = 7.52\,\frac{\text{V/cm}}{\text{T}} \times B\,. \tag{4.39}$$

This electric field contribution, scaling linearly with B, is too small to create a local minimum in the effective potential in Eq. (4.34), as the minimal required field for creating at least a saddle point scales with $B^{4/3}$ (see App. C.3 for derivation). As a result, we can neglect the influence of $\boldsymbol{K}$ entirely and assume $\boldsymbol{K} \equiv 0$.

With further fixing the field orientations with $\boldsymbol{B}$ in z-direction and $\boldsymbol{E}$ in x-direction the effective potential in Eq. (4.34) simplifies to

$$\mathcal{V}_{\text{eff}} = \frac{e^2B^2}{2M}\left(x^2+y^2\right) + e\,E\,x - \frac{e^2}{4\pi\varepsilon_0\varepsilon_r}\frac{1}{\sqrt{x^2+y^2+z^2}}\,. \tag{4.40}$$

From the residual degrees of freedom, the magnetic field dictates how close to the Coulomb center the giant dipole well can be formed. The distance at which a saddle point for sufficient electric fields forms scales with $B^{-2/3}$ (see App. C.3). In return, increasing the electric field deepens the giant dipole well, while also shifting the position of the minimum outwards, away from the Coulomb center. While in atoms the choice of fields is mainly restricted by the magnetic field strengths available to experiments, the field

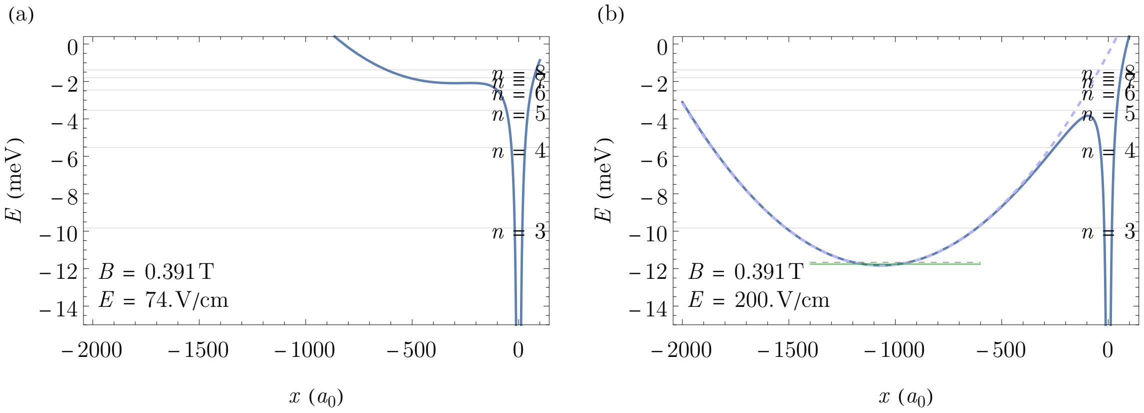


Figure 4.6: Under a magnetic field of $B = 0.391\,\mathrm{T}$, a saddle point (a) is formed at an electric field strength of $74\,\mathrm{V/cm}$. It is located at a distance of $300\,a_0$ and energetically between the $n = 6$ and $n = 7$ state energy levels of the field-free exciton. (b) At an electric field up of $E = 200\,\mathrm{V/cm}$, a giant dipole well is formed at a distance of $x_0 = -1067\,a_0$. Below the dissociation threshold (dashed gray line) the shape of the potential around the local minimum can be approximated harmonically (dashed blue curve). Above the giant dipole ground state (state energy level marked by green line) several excited giant dipole states are supported.

amplifications in Cu_2O would even allow locating the giant dipole minimum energetically below the excitonic ground state with a field of $B = 7\,\mathrm{T}$. As a result, the choice of the magnetic field is more of a balance between desired depth, saddle point height, and distance to the Coulomb center. A good balance can be achieved with a magnetic field of $B = 70 \times 10^{-5}$ a.u., equal to $B = 0.391\,\mathrm{T}$. A saddle point then appears at an electric field strength of $74\,\mathrm{V/cm}$ (see Fig. 4.6(a)) at an energy just below the field-free $n = 7$ state energy level. Increasing the electric field up to $E = 200\,\mathrm{V/cm}$ creates a giant dipole well at a distance[3] of $x_0 = -1067\,a_0$ (see Fig. 4.6(b)) or $1.17\,\mu\mathrm{m}$. Following the definition in Ref. [3], the harmonic approximation (dashed blue curve in Fig. 4.6(b)) is viable up to the dissociation threshold (dashed gray line) and above the ground state (green line) the potential supports several m- and n_z-excited states as well as one n-excitation (see Refs. [3, 250] for details on the giant dipole quantum numbers in cylindrical approximation).

The giant dipole transition scheme proposed in Ref. [193] starts with an initial state in the form of a chaotic trajectory of the electron[4] with only the magnetic field present. The energy of the chosen trajectory corresponds to the energy of the $n = 60$ state of the field-free hydrogen atom but seems unrelated to an exact quantum state of the atom-field-system.[5] In our case, however, with the $n = 7$ energy level of the exciton being located just above the saddle point, it is very much possible to calculate exact eigenstates. This is done in the basis of hydrogen wave functions $\psi_{nlm}(\boldsymbol{r})$ in a series

$$\Psi_i(\boldsymbol{r}) = \sum_{n=1}^{n_{\max}} \sum_{l=0}^{n-1} \sum_{m=-l}^{l} c_{nlm}\,\psi_{nlm}(\boldsymbol{r}) \tag{4.41}$$

up to a maximum principal quantum number of $n_{\max}$. Although this basis is not complete

[3] Admittedly, working with a distance of exactly $300\,a_0$ under a field of nearly $75\,\mathrm{V/cm}$ and a distance of nearly $1000\,a_0$ at field of exactly $200\,\mathrm{v/cm}$ is quite satisfying.

[4] In case of the hydrogen atom, the single particle system can still be considered as the action of the electron alone to a good degree of approximation

[5] Which even today is very challenging to diagonalize due to the rapid increase in sub-states of l and m to consider with increasing n.

and calculating the full quasi-Landau spectrum of resonances above the band-gap energy [181] would require either inclusion of the Landau wave functions [181, 249] or use of Coulomb-Sturmian functions [191, 251–253], it is sufficient for the low lying n states. With $n_{\mathrm{max}} = 20$ the basis contains 2870 elements and eigenstates and energies are calculated in the range of $0 \dots 7\,\mathrm{T}$ (for details on the numerical diagonalization see App. C.4) with the result shown in Fig. 4.7. The linear Zeeman splitting commonly seen in atoms for weak magnetic fields is barely visible for fields of $B < 0.1\,\mathrm{T}$ and the spectrum is dominated by the diamagnetic term with its quadratic B-dependency. Increasing with n, the coupling is strong enough to result in not only l-mixing but also the mixing of n-states. The only good quantum number preserved by the magnetic field is the magnetic quantum number m. The magnetic field of $B = 0.391\,\mathrm{T}$ required for giant dipole states is traced with a dashed vertical line. At this field strength, the fan originating from the $n = 7$ state is mostly affected by l-mixing. Due to the even parity of the valence and conduction band, only excitonic states with odd parity (like P and F states) are dipole-allowed for photonic excitation from the ground state [254]. Consequently, a possible initial state for the giant dipole transition excited from the excitonic vacuum needs sufficient admixtures of the 7P state. Possible states are the 120th, 121st, and 129th state,[6] which can be traced back to the field free 7P $m = -1$, $m = 0$, and $m = 1$ hydrogen wave functions, respectively. The most relevant mixture amplitudes c_{nlm} as defined in Eq. (4.41) for these states are

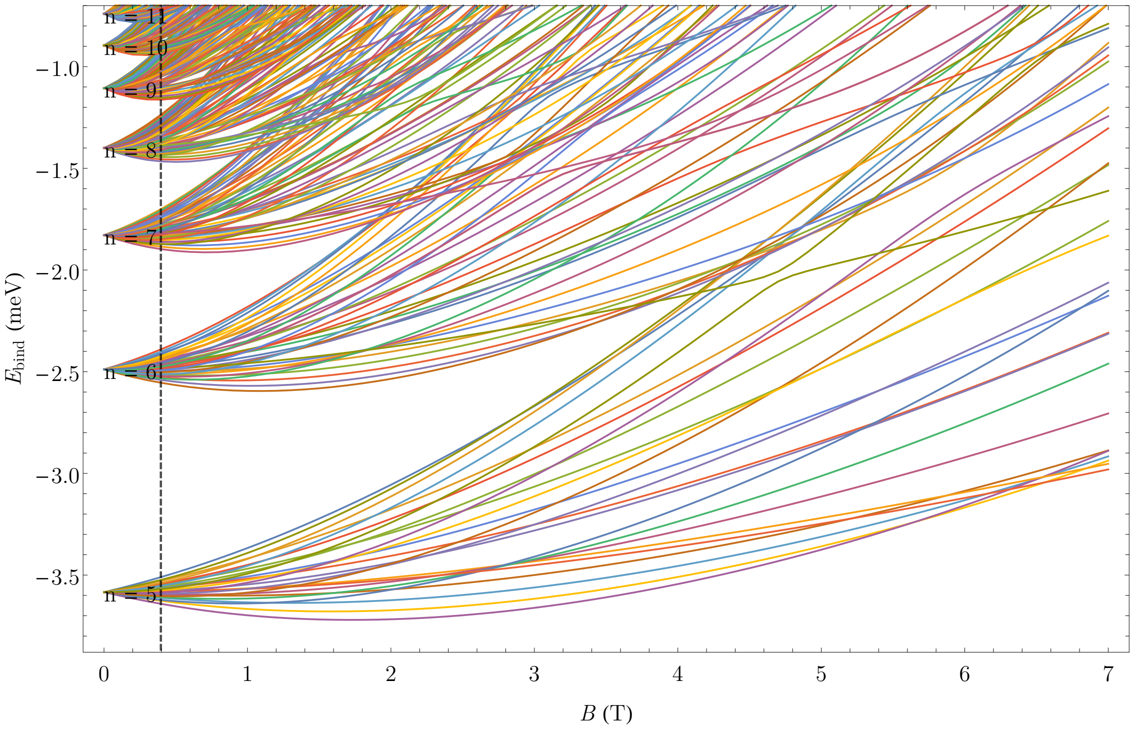


Figure 4.7: The energy levels of magnetoexcitons in the hydrogen model can be obtained by diagonalization in a hydrogen basis for low states below the band gap. The energy splitting is dominated from the diamagnetic coupling, resulting in parabolic curves with linear splitting appearing only below $B < 1\,\mathrm{T}$. The splitting is strong enough for intersections between different n-fans to appear. Due to the tight coupling restrictions for hydrogenic wave functions only a few avoided crossings appear. The dashed vertical line marks the magnetic field of $B = 0.391\,\mathrm{T}$, required for the formation of giant dipole states. Energies are given as binding energies below the band gap.

[6] Here, the states are counted by the index i in energetic order at $B = 0.391\,\mathrm{T}$, starting from the lowest exciton state, which is the nearly pure 1S state.

$i = 120$	
$\{n, l, m\}$	c_{nlm}
$\{7, 1, -1\}$	0.894
$\{7, 3, -1\}$	-0.431
$\{7, 5, -1\}$	0.108
$\{8, 1, -1\}$	0.050
$\{8, 3, -1\}$	-0.033
$\{6, 1, -1\}$	-0.022
$\{8, 5, -1\}$	0.015
$\{9, 1, -1\}$	0.010
$\{6, 3, -1\}$	0.009
$\{9, 3, -1\}$	-0.006

$i = 121$	
$\{n, l, m\}$	c_{nlm}
$\{7, 3, 0\}$	0.733
$\{7, 1, 0\}$	-0.607
$\{7, 5, 0\}$	-0.302
$\{8, 3, 0\}$	0.039
$\{8, 5, 0\}$	-0.029
$\{8, 1, 0\}$	-0.022
$\{6, 1, 0\}$	0.014
$\{6, 3, 0\}$	-0.013
$\{8, 7, 0\}$	0.008
$\{9, 3, 0\}$	0.007

$i = 129$	
$\{n, l, m\}$	c_{nlm}
$\{7, 1, 1\}$	-0.894
$\{7, 3, 1\}$	0.431
$\{7, 5, 1\}$	-0.108
$\{8, 1, 1\}$	-0.050
$\{8, 3, 1\}$	0.033
$\{6, 1, 1\}$	0.022
$\{8, 5, 1\}$	-0.015
$\{9, 1, 1\}$	-0.010
$\{6, 3, 1\}$	-0.009
$\{9, 3, 1\}$	0.006

Table 4.1: Weight parameters c_{nlm} for the three states with the highest 7P admixture at $B = 0.391\,\mathrm{T}$ arising from the $m = -1$, $m = 0$ and $m = 1$ states respectively. For each state the largest 10 amplitudes are given. The states are mixed in both l and n, with only the magnetic quantum number m preserved. Both the $m = 1$ and $m = -1$ states are dominated by the 7P state with only one more relevant admixture of the 7F state. The $m = 0$ state has much larger contributions from higher l states, even exceeding the 7P contribution.

listed in Tab. 4.1. It is worth noting that the state with the highest admixture of the 7P $m = 0$ wave is not dominated by this state but superimposed with $l = 3$ and $l = 5$ wave. The 120th and 129th state, however, are dominated by the 7P wave with a smaller 7F admixture. Consequently, these two states are the most accessible states of the $n = 7$ fan and are chosen as initial states for the giant dipole transition. Further, due to the dominance of the 7P weights, they are in the following reduced to pure 7P states in order to simplify the sampling of the initial state.

4.3. Excitonic Simulation Environment

The GD Hamiltonian in Eq. (4.25) breaks all spatial symmetries of the Coulomb potential through the magnetic- and electric-field interactions. In its general form, it does not possess an obvious eigenbasis and the double-well structure in combination with the residual gauge term complicates an exact diagonalization. Deep learning approaches, on the other hand, are data-driven and require extensive amounts of training data. In the case of deep reinforcement learning, they are obtained through interaction between agent and environment and thus require a sufficiently fast simulation. Here, the correspondence principle allows for the approximate solution of quantum systems through classical trajectories [193]. Consequently, we aim to demonstrate the feasibility of optimizing a GD excitation scheme with deep reinforcement learning on a semi-classical simulation.

The only outside control of the GD system in Eq. (4.25) comes through the fields $\boldsymbol{E}, \boldsymbol{B}$. From these, only the electric field $\boldsymbol{E}$ can possibly be manipulated on a sufficient timescale by voltage regulation, while changing $\boldsymbol{B}$ would require regulation of superconducting currents [243]. Continuing with the fields $\boldsymbol{E}(t) = E(t)\,\boldsymbol{e}_x$ and $\boldsymbol{B} \equiv B\,\boldsymbol{e}_z$ from Sec. 4.2.3, and the corresponding effective potential in Eq. (4.40), the general GD Hamiltonian from Eq. (4.25) simplifies to

$$\begin{aligned}\mathcal{H}_{\text{eff}} =& \frac{1}{2\mu}(p_x^2 + p_y^2 + p_z^2) + \frac{e\,B}{\tilde{\mu}}(x\,p_y - y\,p_x) + \frac{e^2B^2}{8\mu}(x^2+y^2) \\ &+ e\,E(t)\,x - \frac{e^2}{4\pi\varepsilon_0\varepsilon_r}\frac{1}{\sqrt{x^2+y^2+z^2}} + e\,E_s\,z\,. \end{aligned} \tag{4.42}$$

Therein, the single-particle gauge-term $\boldsymbol{\nabla}_{\boldsymbol{r}} f(\boldsymbol{r})$ has been neglected, as it has, by definition, no effect on the following classical equations of motion. Further, a stray electric field E_s in z direction bas been introduced. It simulates either misalignments of the external electric fields or other perturbations that act in the z-direction where the magnetic field does not confine the particle motion [193]. Thus, it may ionize trajectories that are only metastable. The equation of motion, corresponding to Eq. (4.42), are obtained through the Hamilton equation of motion, recall Eqs. (4.26), (4.27), through derivation:

$$\dot{x} = \frac{\partial\mathcal{H}_{\text{eff}}}{\partial p_x} = \frac{p_x}{\mu} - \frac{e\,B}{\tilde{\mu}}y\,, \tag{4.43}$$

$$\dot{y} = \frac{\partial\mathcal{H}_{\text{eff}}}{\partial p_y} = \frac{p_y}{\mu} + \frac{e\,B}{\tilde{\mu}}x\,, \tag{4.44}$$

$$\dot{z} = \frac{\partial\mathcal{H}_{\text{eff}}}{\partial p_z} = \frac{p_z}{\mu}\,, \tag{4.45}$$

$$\dot{p_x} = -\frac{\partial\mathcal{H}_{\text{eff}}}{\partial x} = -\frac{e\,B}{2\tilde{\mu}}p_y - \frac{e^2B^2}{4\mu}\,x - e\,E(t) - \frac{e^2}{4\pi\varepsilon_0\varepsilon_r}\frac{x}{\sqrt{x^2+y^2+z^2}^3}\,, \tag{4.46}$$

$$\dot{p_y} = -\frac{\partial\mathcal{H}_{\text{eff}}}{\partial y} = \frac{e\,B}{2\tilde{\mu}}p_x - \frac{e^2B^2}{4\mu}\,y - \frac{e^2}{4\pi\varepsilon_0\varepsilon_r}\frac{y}{\sqrt{x^2+y^2+z^2}^3}\,, \tag{4.47}$$

$$\dot{p_z} = -\frac{\partial\mathcal{H}_{\text{eff}}}{\partial z} = -e\,E_s - \frac{e^2}{4\pi\varepsilon_0\varepsilon_r}\frac{z}{\sqrt{x^2+y^2+z^2}^3}\,. \tag{4.48}$$

From a starting phase space point $\boldsymbol{\gamma}_0 = (\boldsymbol{r}_0, \boldsymbol{p}_0)^T$, the complete phase space trajectory is obtained by numerical integration of the equations of motion. All calculations are executed in atomic units (see App. C.1). The integration is perfomed with a Runge–Kutta method [255, 256]. It solves the differential equation $y'(t) = f(t, y(t))$ iteratively with a t-step size of Δt from a starting point (t_0, y_0) through the relation

$$k_1 = f\left(t_i, y_i\right)\,, \tag{4.49}$$

$$k_2 = f\left(t_i + \frac{\Delta t}{2}, y_i + \frac{\Delta t}{2}\,k_1\right)\,, \tag{4.50}$$

$$k_3 = f\left(t_i + \frac{\Delta t}{2}, y_i + \frac{\Delta t}{2}\,k_2\right)\,, \tag{4.51}$$

$$k_4 = f\left(t_i + \Delta t, y_i + \Delta t\,k_3\right)\,, \tag{4.52}$$

$$y_{i+1} = y_i + \frac{\Delta t}{6}\left(k_1 + 2\,k_2 + 2\,k_2 + k_4\right)\,. \tag{4.53}$$

The divergence of the Coulomb potential at small distances destabilizes the numerical integration when performed with constant step sizes. This is either resolved by projection into a four-dimensional space [222], or, as in our case, an adaptive time-scaling. The time steps

$$\Delta t = \Delta\tau\ \min(\sqrt{x^2+y^2+z^2}, 100) \tag{4.54}$$

are scaled from a base value of $\Delta\tau = 0.2 \times 10^{-3}$ a.u. with the distance r from the Coulomb center in atomic units. The scaling is further clipped at a value of 100 since the curvature is sufficiently flat beyond this radius. The step size was determined by testing the stability of known closed orbits over several orbit periods. Yet, the level of detail for observation purposes is still sufficient if only recording every 5×10^4th value of the iteration.

The starting phase space point of the trajectory is sampled semi-classically from the approximate quantum eigenstate $\{n, l, m\} = \{7, 1, -1\}$, recall Sec. 4.2.3. The position is sampled from the hydrogenic wavefunction $\psi_{nlm}(r, \theta, \phi) = R_{nl}(r)\, Y_l^m(\theta, \phi)$ by fixing $\theta = 0$, $\phi = 0$ and randomly sampling r from the cumulative probability distribution

$$c(r) = \int_0^r R_{nl}(\rho)^2\, \rho^2\, \mathrm{d}\rho\,. \tag{4.55}$$

The momentum components are then determined by solving the semi-classical angular momentum projection, yielding p_y,

$$m \overset{!}{=} L_z = x\, p_y - y\, p_x\,, \tag{4.56}$$

the semi-classical equation of the angular-momentum quantum number, for p_z,

$$l(l+1) = \boldsymbol{r} \times \boldsymbol{p}\,, \tag{4.57}$$

and, finally, the classical energy conservation in first order perturbation theory,

$$E_{n,m} = -\frac{1}{2n^2} + \frac{B}{\tilde{\mu}}\, m \overset{!}{=} \frac{|\boldsymbol{p}|^2}{2\mu} + \frac{1}{|\boldsymbol{r}|}\,, \tag{4.58}$$

$$|\boldsymbol{p}|^2 = p_x^2 + p_y^2 + p_z^2\,, \tag{4.59}$$

giving the last, p_x-component. This semi-classical, self-consistent phase point is integrated for a random number of $4\times10^6 \ldots 6\times10^6$ iteration steps with $E(t) \equiv 0$ to further randomize especially the staring position in the y-z-plane.

4.3.1. Environment Interaction and Intuitive Strategies

The simulated excitonic system is only partially observable under real-world conditions, with just the current time t and field strength $E(t)$ known at any moment. The training of an RL algorithm further requires the system to provide the same trajectory information as used for the reward calculation [112]. In our case, this is the trajectory center $\overline{\boldsymbol{r}}(t)$ defined by the mean-position of the particle integrated over a certain period. Therefore, the state observation $\boldsymbol{s}_i$ at any step i comprises of the elements time t_i, field strength $E_i = E(t_i)$, and trajectory center $\overline{\boldsymbol{r}}_i$. Provided this information, the agent can interact with the simulation environment by stipulating the slope of the electric field $\partial E = \Delta E / \Delta t$ by which it is propagated over the next iteration interval.

The simulation environment is implemented in Mathematica with a compiled iteration function. Still, simulation times are considerable, with an average of 12 min on *Quantum-Chaos* and 24 min on *qoms* for a single episode. The small data-structures even allow for running the training on the Raspberry Pi *ComputationPi*, taking 1 h per episode.

Environment Constraints

The simulation environment is restricted in its parameters. The propagation time is limited to $t_{\mathrm{max}} = 5.0\,\mathrm{ns} = 1.34 \times 10^6\,\mathrm{a.u.}$. The position-space is normalized to a range of $r_{\mathrm{max}} = 2000\,a_0$ and constrained to a cubic volume of $2\,r_{\mathrm{max}}$. Any trajectory surpassing this boundary is assumed to ionize, which leads to an instant termination of the simulation. It is the only hard failure condition. The absolute value of the electric field is capped at a value of $E_{\mathrm{max}} = 300\,\mathrm{V\,cm^{-1}}$, well below the limit of existing experiment setups [190, 245], and the slope is limited to $\partial E_{\mathrm{max}} = 150\,\mathrm{V\,cm^{-1}\,ns^{-1}}$. It still allows reaching E_{max} within 2 ns. Between each agent-environment iteration, the environment integrates the equations of motion for $N_{\mathrm{it}} = 5 \times 10^5$ steps and calculates $\overline{\boldsymbol{r}}_i$ as the mean of the position-space trajectory. The number of integration steps for every environment iteration is a trade-off between decision frequency and precision of the trajectory-center calculation. For a sensible result, it is required to cover at least one complete orbital period of the trajectory but it also has to be short enough to allow frequent interactions. Because the oscillation period in z-direction is considerably longer than in the xy-plane, the observation of the trajectory-center is oversampled by a factor of 2, meaning that the returned observation of $\overline{\boldsymbol{r}}_i$ is calculated as the mean of both $\overline{\boldsymbol{r}}_i$ and $\overline{\boldsymbol{r}}_{i-1}$. With these parameters an average of 280 agent-environment loops are executed until the simulated time reaches t_{max}.

Intuitive Strategies

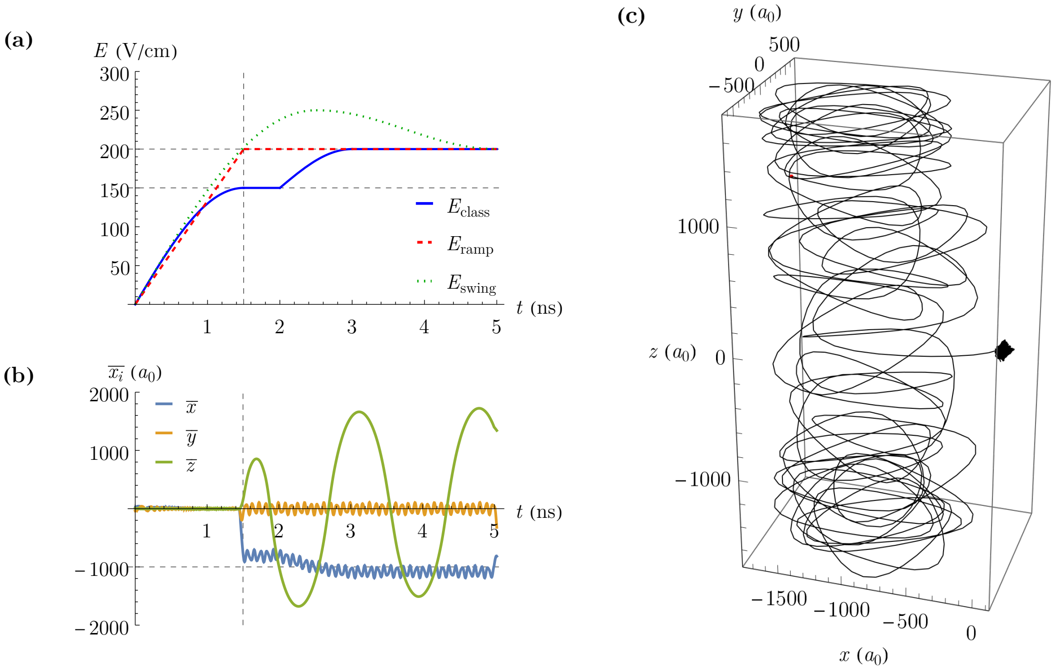


Figure 4.8: The field evolution from Eqs. (4.60), (4.61), and (4.62) are plotted in panel (a). The particle trajectory of a sample simulation with the classical strategy (solid blue line) are shown in panels (b) and (c), respectively. The time evolution of the trajectory center in (b) reveals that the particle moves to the outer well at 1.5 ns, indicated by a dashed line. The extent and period of the orbit in z direction is much larger than in the xy-plane due to the weaker confinement along the magnetic field axis.

When searching for the optimal field evolution it is rational to benchmark it against simple and intuitive excitation strategies. A strategy for highly excited Rydberg states of hydrogen atoms was developed in Ref. [193] in the form of a composite function

$$E_{\text{class}}(t) = \begin{cases} E_c \sin\left(\frac{\pi\, t}{2\, t_1}\right) & t \leq t_1\,, \\ E_c & t_1 < t \leq t_2\,, \\ E_c + (E_f - E_c) \sin\left(\frac{\pi\,(t - t_2)}{2\,(t_f - t_2)}\right) & t_2 < t \leq t_f\,, \\ E_f & t > t_f\,. \end{cases} \tag{4.60}$$

The first step of this evolution is to increase the field strength up to the formation of saddle point E_c and then wait for a certain timespan $t_2 - t_1$ to let all trajectories pass over the saddle point before increasing the field to the final value E_f. Letting all trajectories pass at the same field value guarantees a narrow energy distribution of the final states [193]. This strategy can be transferred to the excitonic system by modifying the constants $t_1 = 1.5\,\text{ns}$, $t_2 = t_1 + 0.5\,\text{ns}$, $t_f = t_2 + 1.0\,\text{ns}$, $E_f = 200\,\text{V}\,\text{cm}^{-1}$, and $E_c = 150\,\text{V}\,\text{cm}^{-1}$. It is plotted in Fig. 4.8(a) and the corresponding state evolution of a sample simulation is shown in the panels (c) and (b) with the trajectory and trajectory-center, respectively. It is worth noting that any trajectories only traverse from the Coulomb well to the outer potential well at a field strength of $E_c = 150\,\text{V}\,\text{cm}^{-1}$, although the saddle point already forms at a value of $E = 74\,\text{V}\,\text{cm}^{-1}$. Another noteworthy observation is the much longer orbit period and amplitude of the trajectory center in the z-direction compared to the x- or y-coordinate. It is another side-effect of the weak confinement along the magnetic field axis and needs to be considered in shaping the reward function.

In contrast to the atomic case we observe the excitation trajectories to travel into the outer potential well almost simultaneously, varying only within one orbit period. With the field plateau having no notable effect the field evolution can be simplified to a linear ramp the field strength until E_f is reached and then keep the field constant

$$E_{\text{ramp}}(t) = \begin{cases} E_f\,\frac{t}{t_1} & t \leq t_1\,, \\ E_f & t > t_1\,. \end{cases} \tag{4.61}$$

Another possible field evolution could also feature an over-swing of the field above the value of E_f and a subsequent return, e.g. in the form of

$$E_{\text{swing}}(t) = \begin{cases} E_{\text{sw}} \sin\left(\frac{\pi t}{2 t_{\text{sw}}}\right) & t \leq t_{\text{sw}}\,, \\ E_{\text{sw}} + (E_f - E_{\text{sw}})\left(1 - \cos\left(\frac{\pi\,(t - t_{\text{sw}})}{2\, t_{\text{sw}} - t_{\text{sw}}}\right)\right)/2 & t_{\text{sw}} < t <= 2t_{\text{sw}} \\ E_f & t > 2t_{\text{sw}}\,. \end{cases} \tag{4.62}$$

with $t_{\text{sw}} = 2.5\,\text{ns}$ and $E_{\text{sw}} = 250\,\text{V}\,\text{cm}^{-1}$. A comparison of all three field evolutions presented here is plotted in Fig. 4.8(a). We observe an ionization rate of approximately 40% over a test of 100 trajectories for all three strategies, each. This rate is considerably higher than for the atomic case in Ref. [193] and is likely caused by the weaker confinement of the excitonic system along the magnetic field direction. It also appears to be independent of the specific field evolution.

4.3.2. Simplified Environment

The computation time of the excitonic simulation environment of 12 min in the best case is considerably larger than that of typical example systems. This hampers the ability to perform large scale tests like hyperparameter optimizations within a reasonable timeframe.

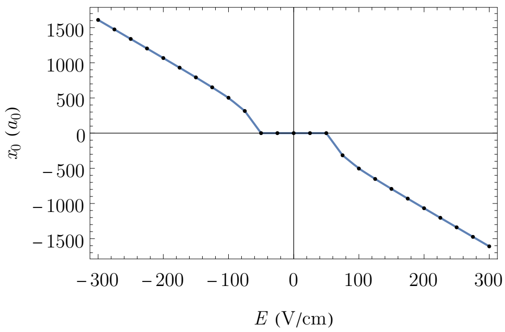


Figure 4.9: The center of the GD trajectory ideally should coincide with the position of the GD minimum. Where existent, its position x_0 depends approximately linearly on the electric field strength E.

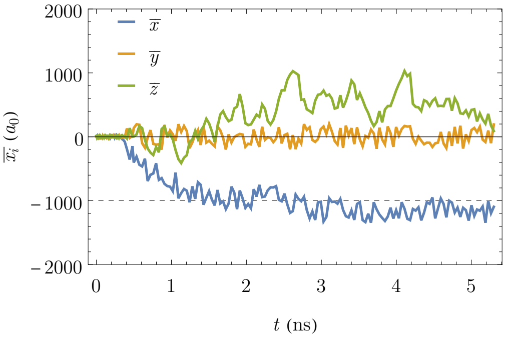


Figure 4.10: Under the interaction with the classical field evolution from Eq. (4.60) the simplified simulation environment calculates a similar trajectory as in Fig. 4.8(b), but is lacking the temporal correlation.

Yet, the effective propagation of the excitonic systems can be approximated through a set of simple relations. If neglecting the delayed transfer to the GD well seen in Fig. 4.8 the trajectory center should always fall in the vicinity of either an existing GD minimum or the origin. The position of the GD minimum $x_0(E)$ depends approximately linearly on the electric field strength E [186, 187]. The approximation is refined by numerically determining the minimum of the effective GD potential from Eq. (4.40) and interpolating linearly between these values, like shown in Fig. 4.9. The oscillations visible in Fig. 4.8 are replaced with uniform noise. Further, the overall timestep length Δt over N iterations is approximated by multiplying the value of Eq. (4.54) for the initial trajectory center by N. Thus, the evolution of the exact simulation environment over N_{it} iterations is simulated by a set of three equations. Given a field slope ∂E, any system state with time t_i, field E_i, and trajectory center $\overline{r}_i$ evolves by:

$$t_{i+1} = t_i + \Delta t = t_i + \Delta\tau \, \min(|\overline{\boldsymbol{r}}_i|, 100)\, N_{\text{it}} \,, \tag{4.63}$$

$$E_{i+1} = E_i + \Delta t \, \partial E \,, \tag{4.64}$$

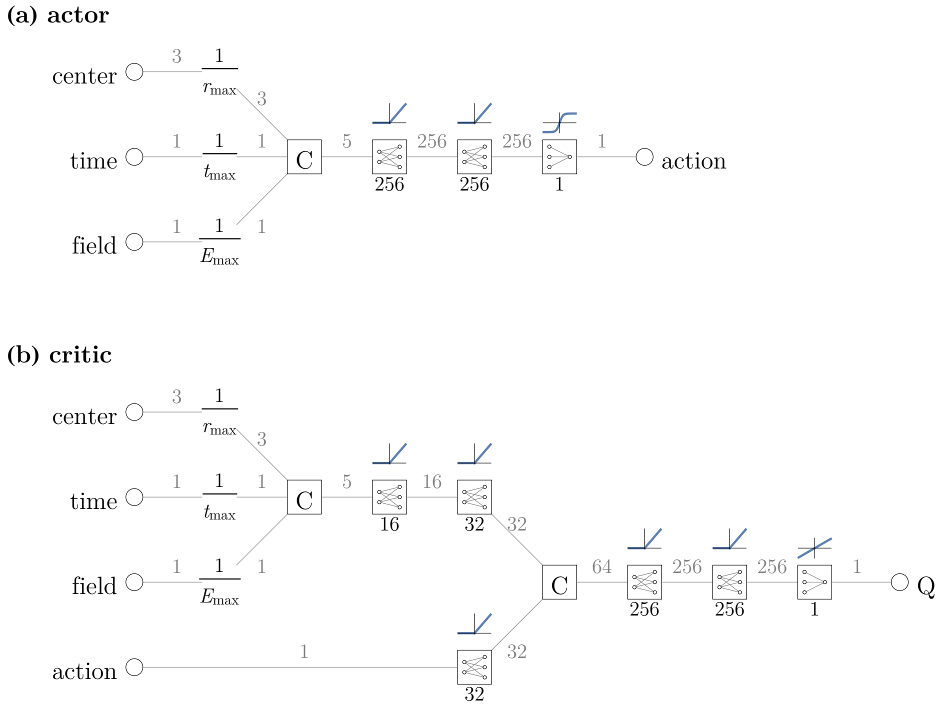
$$\overline{\boldsymbol{r}}_{i+1} = (x_0(E_{i+1}), 0, \overline{z}_i)^T + 20\, \boldsymbol{v}_{\mathrm{n}} \begin{cases} 1 & |\overline{\boldsymbol{r}}_i| < 100 \\ 10 & |\overline{\boldsymbol{r}}_i| \geq 100 \end{cases} \,, \tag{4.65}$$

with a uniform noise vector $\boldsymbol{v} \in \mathbb{R}^3$ sampled from the interval $[-1, 1]$. The larger oscillation in z-direction and occasional ionization is simulated by a random process of accumulating the noise in the z-component. An example evolution of the simplified simulation is plotted in Fig. 4.10. The mean of the trajectory center follows a similar evolution like Fig. 4.8(b) but with much less temporal correlations. With evaluation times in the sub-second regime and equal variable scaling as the exact simulation it enables us to perform large scale tests within a reasonable timeframe.

4.4. Dynamic Field Optimization through Deep Reinforcement Learning

The goal of the GD excitation is to bring the effective single particle[7], described by the Hamiltonian in Eq. (4.42), into a stable orbit around the targeted GD minimum $\boldsymbol{r}_{\text{targ}} = (x_{\text{targ}}, 0, 0)^T$ with $x_{\text{targ}} = -1067\,a_0$, recall Sec. 4.2.3. Consequently, the RL agent needs to be rewarded for minimizing the distance between the orbit-center $\overline{\boldsymbol{r}}(t)$, as a mean value of the trajectory $\boldsymbol{r}(t)$ over some time interval, and the target point $\boldsymbol{r}_{\text{targ}}$. The simulation environment is implemented in Mathematica, due to requiring specific analytic functions . Consequently, the DDPG RL algorithm is implemented in Mathematica, too.

Agent Design


(a) actor

(b) critic

Figure 4.11: The design of the actor and critic networks is adopted from Ref. [116]. The actor (a) normalizes the three state-input variables before concatenation [C] into a single vector with five entries (gray numbers mark the vector sizes). It is then processed by a stack of two fully connected layers with 256 neurons each with are ReLU-activation (see small plots above). The normalized action output value is obtained from a single, tanh-activated neuron. The critic uses a similar structure for processing the state input but with fewer neurons. The action input is further processes by a single fully connected layer and the two input-branches are consolidates into a single vector of size 64. It is then processed by a stack of two fully connected layers. The terminal layer is also a single linear-activated neuron.

The deep reinforcement learning agent chosen for controlling the GD excitation is the DDPG algorithm [114] introduced in Sec. 2.3.1. It is an actor-critic model composed of two neural networks. The actor learns the policy $a = \mu(s)$, and the critic learns the action-value function $Q(s, a)$. The design of both actor and critic is adopted from the implementation in

[7] In the context of an hydrogen atom, this would be the electron. With the similar mass of the electron and hole of an exciton, the single-particle trajectory describes the relative distance of both the electron- and hole-orbits.

Ref. [116] and depicted in Fig. 4.11. Both actor and critic are designed to accept the state components time t, field $E(t)$, and trajectory center $\overline{\boldsymbol{r}}$ in atomic units. In the first layer of the network, these three state inputs are normalized by the corresponding environment constraints and then concatenated into a single vector. The actor uses two fully connected, ReLU-activated layers with 256 neurons each for processing the state input and a final single neuron with *tanh*-activation to predict the single output action. Different to the state-variables the action is normalized to the interval $[-1, 1]$ for all learning purposes. Only when given to the environment, the action is scaled to atomic units by multiplication with $\partial E_{\max}$. The critic uses a similar structure of two fully connected layers for processing the state input, but with just 16 and 32 neurons, respectively. It further processes the action input in a single fully connected layer with 32 neurons and combines bot input branches by a concatenate layer. From this, the Q-function is calculated by a stack of two fully connected layers with 256 neurons and a single linear activated neuron.

Reward Shaping and Local Attractors

With the goal of bringing the system into a GD state the reward should be given depending on the distance of the trajectory center $\overline{\boldsymbol{r}}$ to the target point $\boldsymbol{r}_{\text{targ}}$. Still, the question remains how to calculate this reward. The design of the reward signal (also called reward shaping) is a critical part of an RL application [112]. While some rewards appear intuitive for the human mind, they may get exploited by an RL agent to accumulate rewards in an unintended way. An additional challenge of the GD environment lies in its dynamics. Because the particle can only leave the Coulomb well beyond a certain electric field value, there exists a region where the particle position the reward does not change significantly. Conversely, once leaving the Coulomb well the propagation is comparably fast and it is quite likely to overshoot the target position. These challenges result in the existence of two local attractors of learned strategies which are near impossible to escape even with strong exploration. The first is to keep the particle inside the Coulomb well by regulating the electric field around zero-value. We call this the *do-nothing* strategy. The second attractor is to try to constantly increase the field with maximum slope, even when reaching $E_{\max}$, dubbed *max-field*.

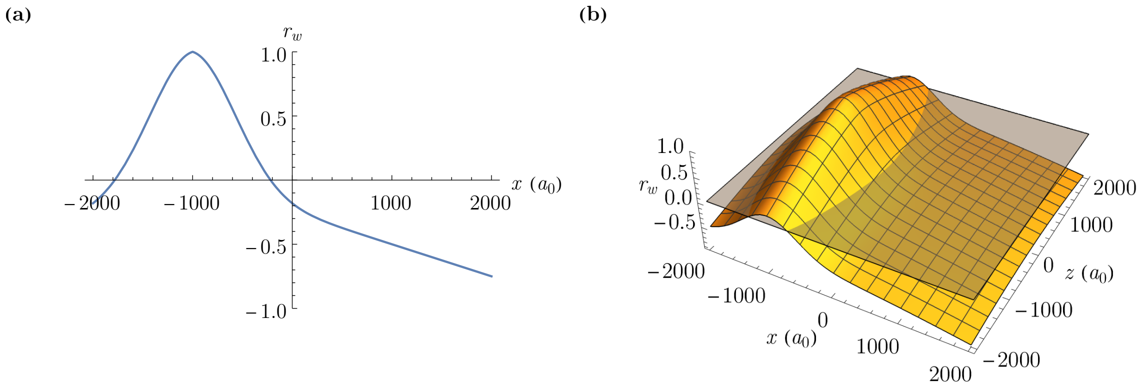


Figure 4.12: The reward function, plotted along the x-coordinate in (a), is a combination of a Gaussian-shaped positive reward around the target point of $x_{\text{targ}} \approx 1000$ and a negative, linear penalty. As a result, an increasing reward can only be accumulated in the direct vicinity of the target position. Otherwise the reward is even decreasing over time. In the xz-plane (b), the increased width of the Gaussian z-direction can be seen, which accounts for the greater oscillation lengths in this direction due to weaker confinement.

We found the best performance by implementing a composite reward function that implements a linear global slope added to a Gaussian reward in the direct vicinity of the target position. The reward function

$$r_w(\overline{\boldsymbol{r}}) = s_r \left(\mathrm{e}^{-\left(\frac{(\overline{x}-x_{\mathrm{targ}})^2}{2\sigma_r^2}+\frac{\overline{y}^2}{2\sigma_r^2}+\frac{\overline{z}^2}{2(5\sigma_r)^2}\right)} - 0.5\frac{|\overline{\boldsymbol{r}}-\boldsymbol{r}_{\mathrm{targ}}|}{r_{\mathrm{max}}} \right) , \tag{4.66}$$

gives a positive reward in the direct vicinity of the target position $\boldsymbol{r}_{\mathrm{targ}}$ and returns a penalty at larger distances (see Fig. 4.12). The variance $\sigma_r = 125\, a_0$ of the Gaussian is multiplied by a factor of 5 in z-direction to compensate for the sweeping oscillation in this direction, recall Fig. 4.8. The entire reward function is scaled by a factor s_r which is known as a hyperparameter for the training of a DDPG algorithm [117]. The dynamic scaling of the simulation timesteps in Eq. (4.54) also requires a time normalization of the returned reward in order to obtain consistent returns. The time normalization between two iteration steps i and $i+1$ is performed through multiplying r_w by $\Delta t/\Delta t_{\mathrm{max}}$, with $\Delta t = t_{i+1} - t_i$ and $\Delta t_{\mathrm{max}} = 100\, \Delta\tau\, N_{\mathrm{it}}$.

Exploration Noise

The DDPG algorithm is a deterministic agent and thus always acts according to the most recent policy. In order to learn new strategies, an algorithm needs to explore new actions outside the current policy. Therefore, a deterministic agent needs outside help for exploration in form of an exploration noise function $\mathcal{N}_e$. In the original publication, Ref. [114], of the DDPG the use of an Ornstein–Uhlenbeck process [257] is proposed. It simulates the time-correlated fluctuations of a Brownian motion, an example of which is shown in Fig. 4.13. Knowing that the excitation of a GD state requires a long, time-correlated action, this choice of exploration noise is more appropriate than uncorrelated noises [258].

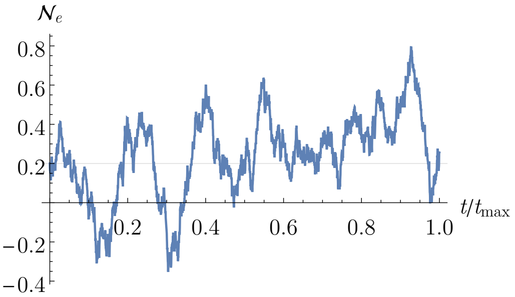


Figure 4.13: The exploration noise is sampled from an Ornstein–Uhlenbeck process [257], here shown for long-term mean $\mu = 0$, volatility $\sigma = 1.0$, mean reversion speed $\theta = 0.15$ and initial value $n_0 = 0.2$ also marked by the gray line.

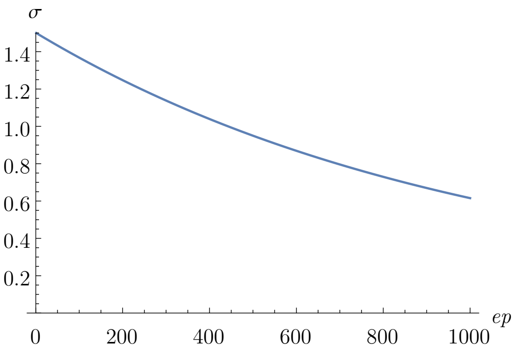


Figure 4.14: The volatility σ is decreased over progressing episodes ep to reduce the variance of the exploration noise.

For noise generation, the Ornstein–Uhlenbeck process is simulated with long-term mean $\mu = 0$, volatility σ, mean reversion speed $\theta = 0.15$ and a random initial value $n_0 \in [-1, 1]$. The volatility σ is a hyperparameter and also determines the variance of the noise distribution. We aim for a wider exploration in the earlier stages that narrows down during training and subsequently allows finer tuning in the later training episodes ep. This is achieved by an exponential decay

$$\sigma(ep) = 0.1 + 1.4\,\mathrm{e}^{-\frac{ep-1}{1000}}\,, \tag{4.67}$$

plotted in Fig. 4.14. It is also used for scaling the initial value $n_0' = \sigma(ep)\,n_0$. During training, the actions are determined by the noise-modified policy

$$\mu'(s) = \mu(s) + \mathcal{N}_e(ep)\,, \tag{4.68}$$

and are further clipped to the interval $[-1, 1]$ to conform with the environment constraints on legal actions. Note that the noise itself is not clipped and thus may override the actor in some cases. This helps further with exploration especially in the earlier stages of training.

Hyperparameter Optimization

Table 4.2: Optimized hyperparameters

quantity	value
loss scaling s_r	0.1
actor learning rate	1.0×10^{-3}
critic learning rate	2.0×10^{-3}
critic L_2-regularization	2.0×10^{-5}
batch size	1024
buffer length	∞

The learning performance of the DDPG algorithm is known to be highly sensitive to hyperparameters [118, 119]. For the present GD excitation task none of the sets of hyperparameters reported in Refs. [114, 116, 118] achieve a stable convergence and requires optimization. However, the slow evaluation speed of the simulation environment prevents a wide parameter scan within a reasonable time. Instead, a broad parameter search is performed using the simplified simulation environment described in Sec. 4.3.2 to find a set of parameters that is then fine-tuned on the exact simulation. The final set of hyperparameters is listed in Tab. 4.2. In agreement with Ref. [117] we find that the training stability improves by scaling the reward by $s_r = 0.1$. The stability if further improved by increasing the batch size to 1024 whilst even larger batch sizes hamper the convergence. An important hyperparameter for the long-term stability of the training is the L_2-regularization of the critic, which needs to be balanced against the learning rates of both actor and critic. Without regularization the critic frequently runs into a positive feedback loop with the target-critic in Eq. (2.19) that leads to exploding values and a subsequent breakdown of the training. Yet, too heavy regularization denies any learning progress at all. The most stable solution was found by combining the learning rates from Ref. [116] with an L_2 regularization of 2.0×10^{-5}. The long term stability benefits further from observation buffer lengths capable of preserving all observations of the training. Thus, it is effectively set of infinity by never deleting any old observations from the buffer.

With the given set of hyperparameters the DDPG algorithm converges on the simplified environment within $500 - 1000$ episodes. An example learning curve is shown in Fig. 4.15. The learned strategy is almost time-invariant and instead depends only on the x-coordinate of the trajectory-center and the value of the electric field. It implements a decision boundary through the target point in the xE-plane, with slope $+\partial E_{\max}$ below and $-\partial E_{\max}$ above the target point. Effectively, it implements the same temporal field evolution as the ramp-strategy in Eq. (4.61). With the fast evaluation speed of the simplified environment it is further possible to cross-check this result with a Monte–Carlo approach. It is implemented as an on-policy learning by starting out with a random field function $E(t)$ and then adding the same decaying exploration noise as for the DDPG algorithm. If the total reward after any episode is larger than the current best result it is adopted as the new strategy.

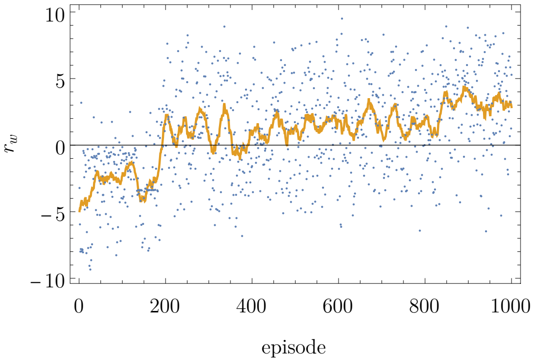


Figure 4.15: The total reward received (blue dots) varies widely between the training episodes due to the strong exploration noise. Averaged over 10 episodes each, the global trend and convergence within 1000 episodes becomes apparent.

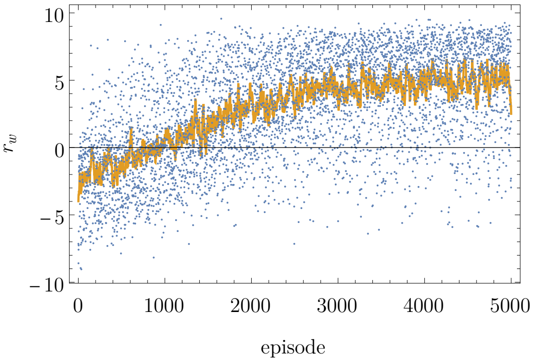


Figure 4.16: On the simplified simulation environment a Monte–Carlo learning algorithm with decaying exploration noise converges within 5000 episodes.

The learning curve in Fig. 4.16 shows convergence after 5000 episodes. The Monte–Carlo approach agrees with the DDPG algorithm in also learning the ramp-strategy.

Optimized Field Results

Executed on the exact simulation environment the training with the DDPG algorithm proves less stable than on the simplified environment. A large portion of the overall training process is taken up with finding the rather narrow region of a valid excitation strategy that involves bringing the particle into the outer well and then stabilizing it. However, the existence of the previously described local attractors frequently results in the RL agent never finding a suitable strategy. The tests with the simplified simulation environment has shown that the ramp-strategy from Eq. (4.61) is a valid solution for the simplified simulation environment. Therefore, we aim to accelerate the training and help with exploration by pre-filling the observation buffer with observation data from valid strategies. These are obtained from simulations with the intuitive strategies from Sec. 4.3.1. We perform 20 simulations with each of the three strategies.

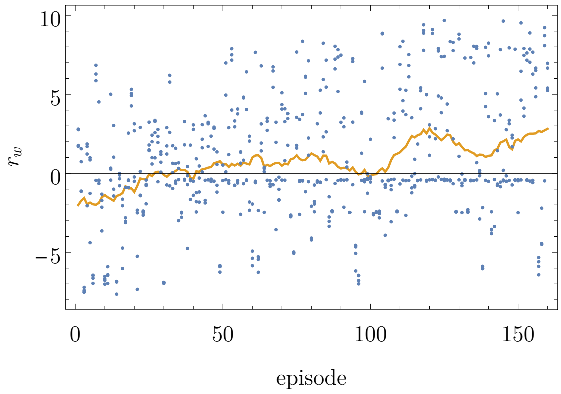


Figure 4.17: The learning curve shows the total reward gained from all simulation runs during each episode as blue dots, with four dots per episode representing the parallelized simulations. The distribution of rewards is very broad due to the strong exploration noise employed. When taking the rolling average over ten episodes (orange curve) the improvement in the total rewards are clearly visible.

The reason for the training instability appears to lie in the much larger ionization rates in z-direction of the exact environment. By controlling only the electric field in x-direction the possible influence on this dynamic is limited to the levels of being nearly nonexistent. For comparison, about half of the simulations performed for pre-filling the observation buffer do ionize independently of the present strategy. Thus, the virtually random process of state ionization can lead to confusion during the training when frequently encountering ionizing trajectories along an otherwise successful strategy. We overcome this statistical challenge by acquiring more data during training. The trajectory simulation is a time consuming, consecutive task that only can be parallelized by executing several independent simulations simultaneously. This also allows to a much more efficient use of modern multi-CPU computers. For each training episode we initialize as many independent trajectories as physical CPUs available which are propagated simultaneously. The actions for each simulation are computed individually with the same actor-net but all parallel simulations are modified by the same exploration noise. It leads to a much faster exploration of the reward statistic starting from a certain state. The learning curve corresponding to this training setup is shown in Fig. 4.17. The total rewards gained from the individual simulation runs are still widely distributed due to the strong exploration noise. However, it converges more steadily and faster than in Fig. 4.15, benefiting from the pre-generated observation data and improved statistics. In total, the training requires only 160 episodes with a parallelization of four.

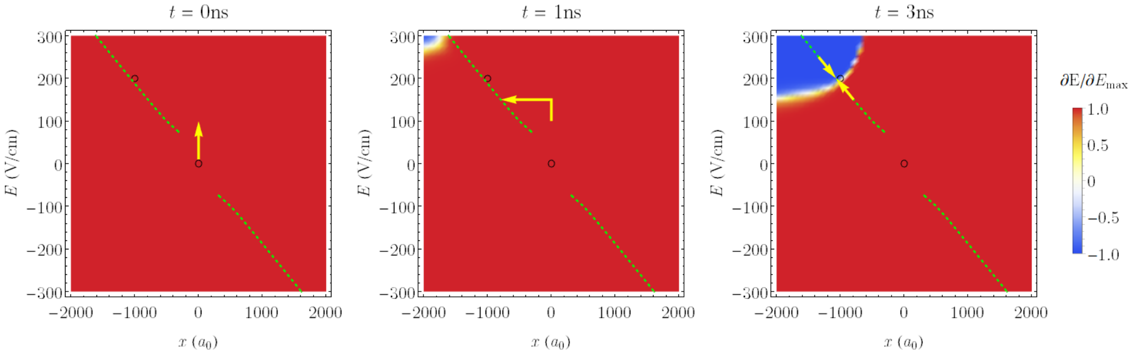


Figure 4.18: The state activation map of the actor is plotted over the observation space of the trajectory center component $\overline{x}$ and field strength E. The dashed lines mark the positions of the GD minimums for both positive and negative field values. From the starting point (central circle), the action is positive propagating the system state in positive field direction (yellow arrow in left panel). The position component of the state can not leave the confines of the Coulomb well until a critical field strength is surpassed (yellow arrow in central panel) at which the center of the trajectory moves to the outer minimum along the dashed green line. With increasing field strength the system state travels along this line until the action changes sign when crossing the target position $x_{\text{targ}} = 1000\,a_0$, thus balancing the system around the target state.

The strategy learned by the DDPG algorithm is an improvement on the ramp-strategy in Eq. (4.61). The global strategy can be read from the activation map of the actor plotted in the xE-plane of the state space in Fig. 4.18. The system is initialized at $t = 0$ (left panel) inside the central black circle and the activation map is entirely positive with action $a = +\partial E_{\text{max}}$. The action results in a rapid increase of the field strength, while the trajectory is still trapped inside the Coulomb well. At $t = 1\,\text{ns}$ a region of negative action form in the top right corner. At this point in time, the field could not yet have reached the threshold value of $E = 150\,\text{v}\,\text{cm}^{-1}$. Once surpassing this value, the trajectory travels into the outside potential well (yellow arrow) and centers around the position of the outer minimum, traced by the dashed green line. At $t = 3\,\text{ns}$ we see that the activation value changes sign at the target position, meaning that the system is balanced by the actor around the target point.

The actions as visible from the activation map in Fig. 4.18 equal the ramp-strategy from Eq. (4.61). Finer details of the learned strategy become visible when testing it on sample trajectories. Fig. 4.19 shows the interaction with 10 randomly initialized trajectories. Overall, all field profiles generally follow a ramp-strategy with at first increasing the field

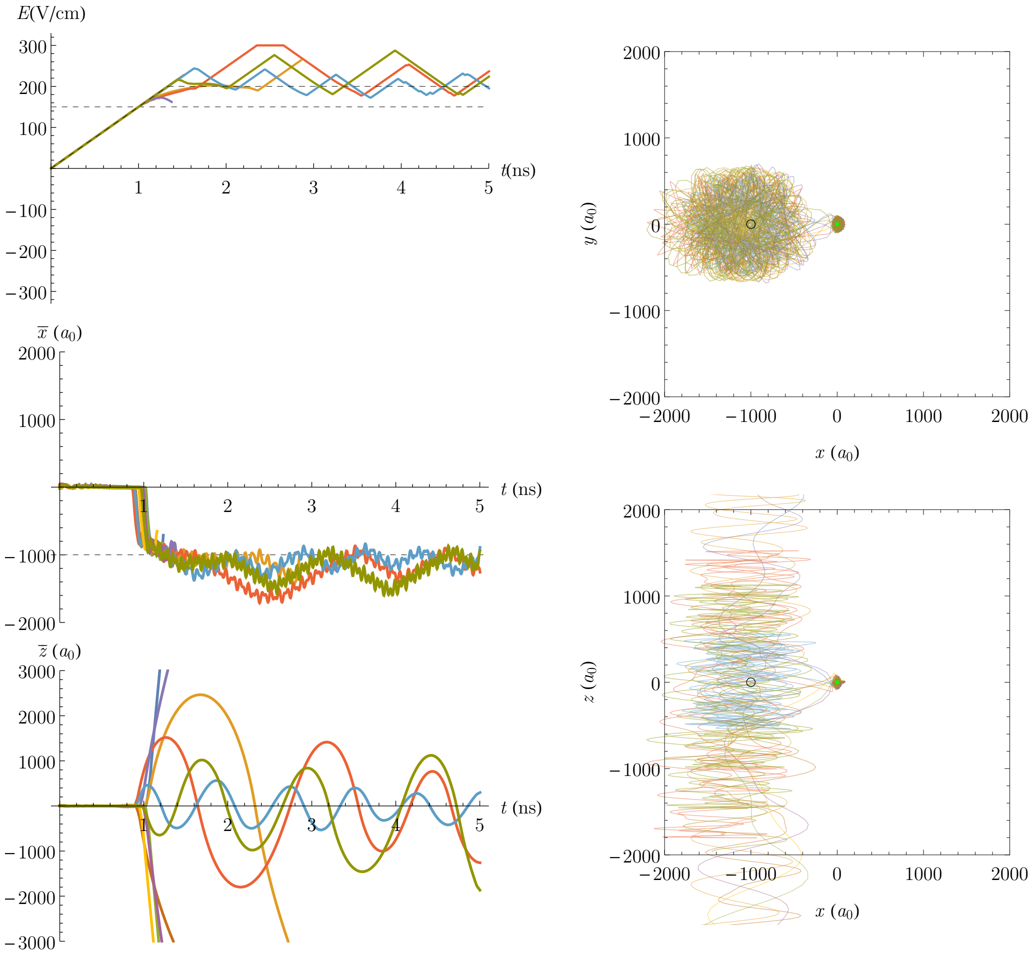


Figure 4.19: The RL agent is tested on 10 randomly initialized trajectories. The top left plot shows the evolution of the corresponding electric fields. Effectively, the RL agent learns an improved version of the ramp-strategy from Eq. (4.61). The left central panel shows the x-component of the trajectory centers, where it can be seen that all trajectories move into the outer well almost simultaneously. In the bottom plot of the corresponding z-component, ionization loss of some trajectories becomes visible. The remaining trajectories form stable orbits around the target point, which can be seen in the right panels with projections into the xy-plane and xz-plane. The electric field curves show oscillations of the same period as the z-component of the trajectory center (compare left column top and bottom) but with a phase difference. This leads to a long-term damping of the oscillations in z-direction.

with maximum slope (dashed diagonal) and then regulating the field along the target value. However, the oscillations in the balancing region show oscillations with regular periods and appear larger than required. These oscillations translate directly into the x-position of the trajectory center (left central panel). When comparing the field profiles in the top panel to the z-coordinate of the trajectory center in the bottom panel it appears that the field oscillations share the period length of the harmonic oscillations in z-direction but are shifted out of phase. It further can be noticed that the z-oscillations decrease in amplitude over time. This is a behavior that could not be observed when testing the intuitive strategies. The z-direction is the only coordinate in which the state can be ionized, like visible with e.g. the violet and yellow trajectories. The RL algorithm seems to have learned how to damp these oscillations which in turn leads to closer and more stable trajectories that ultimately return higher rewards. However, this strategy may not be relevant for the experimental realization of GD states. The transition into the GD state is already finished for all nonionizing trajectories at $t = 1.3\,\mathrm{ns}$ when the electric field reaches $E = E_f$. Further, it is not certain that the stabilizing actions on the classical trajectory would have the same effect on the real quantum states. Therefore, it can be concluded that the RL algorithm effectively proposes an excitation via the ramp-strategy.

4.5. Implications for the Realization of Giant Dipole States

The reinforcement learning agent succeeds in transferring a Rydberg exciton in cuprous oxide into a giant dipole state in a hydrogenic model. The optimal solution from the value function of the RL agent is to increase the electric field strength as fast as possible to the target value. This is a much simpler field evolution as proposed for atoms in Ref. [193]. However, in contrast to the atomic case, the crossing of the particle trajectories from the Coulomb well into the outer minimum happens almost simultaneously for all test trajectories. Therefore, no long-lasting field plateau is required. Still, the ionization rate is rather high in the excitonic case, due to the much weaker binding strength compared to the atomic case. This could be counteracted by giving either a strong penalty to ionizing trajectories (above the already high penalty collected at large z-distances) or by giving a high reward for states surviving until $t_{\max}$ is reached. Yet, this again needs to be balanced against other rewards to not create new, unintended local attractors.

The simplicity of the present solutions makes it easy to implement and test in existing experimental setups. The excitation of the giant dipole state is finished at the end of the linear field-ramp and all further interaction revolves around stabilizing the state around the target point. Therefore, it it could be implemented by a sawtooth or triangle wave. The ramp-time of $1.3\,\mathrm{ns}$ is equivalent to a sawtooth wave with a radio frequency of $780\,\mathrm{MHz}$ and voltage amplitude of $11.25\,\mathrm{V}$ in the experimental experimental setup of Ref. [190]. It is currently unknown how the polarization effects of $\mathrm{Cu_2O}$ would influence the temporal evolution of the internal electric field on such short time-scales. Still, the excitation process has to unfold at a shorter time than the decay of the excitons, which is known from experiments like Ref. [188] to be in the range of $1\,\mathrm{ns}$. Under this constraint, the excitation scheme proposed here may be barely fast enough. Yet, most likely, only the transition into the outer potential minimum may be required to be achieved within this time window. The fast decay of the observed Rydberg excitons happens through a phonon-scattering mediated transition into low lying states like the $1s$ exciton [242]. Yet, in a giant dipole state, the spatial separation leads to a vanishing overlap with the $1s$ or any other core-centric wavefunction, as demonstrated for positronium in Ref. [259].

Based on our results, the excitation scheme can further be improved, mainly by reducing the number of approximations in the simulation environment. The first improvement would be a transfer to actual quantum calculations of the hydrogenic system but propagated adiabatically. It could be followed by implementing the full band-Hamiltonian. The last modification would be to also implement full quantum time-propagation through a master equation. This, however, has not yet been done for excitons in $\mathrm{Cu_2O}$. Still, it would allow for testing coherent-control excitations that could offer a direct transition route from an ordinary Rydberg exciton into a giant dipole state without the need to propagate the external fields.

5. Summary & Outlook

5.1. Summary

In this thesis we have explored several applications of neural networks in physics. They are grouped in the data-driven solution of inversion problems, on the example of scattering reconstructions, and optimizing the control of a physical system through deep reinforcement learning. Our approaches offer a significant speed-up over existing methods and some are capable of solving problems that lie beyond the reach of classical algorithms.

In Chap. 3 we have investigated how neural networks can aid in the reconstruction of the shapes of silver nanoclusters from single-shot wide-angle scattering patterns. Even with classical methods, the inversion of single-particle scattering patterns is an inherently data-driven problem. Classical algorithms, both forward fitting and iterative phase retrieval, rely on repeated forward calculations and convergence from random initial conditions. In this, their reconstruction tasks are not too dissimilar from classical computer vision tasks and prime candidates for the application of neural networks. We have shown in Sec. 3.2 how neural networks can solve the task of extracting the size and orientation of a particle with known shape and found a significant speed-up over classical methods. The scattering simulation effort required for reconstructing just a few scattering patterns is already sufficient for generating a randomized dataset and training a neural network that evaluates in milliseconds. Neural networks offer fast evaluation times, required for fully analyzing datasets obtained from modern FEL experiments, that can contain several 10 – 100 thousand images [93, 173].

Equipped with the knowledge learned from reproducing the capability of existing reconstruction techniques we have advanced the reconstruction capability of neural network beyond the scope of classical algorithms. With the discrete density reconstruction described in Sec. 3.4 we perform reconstructions at the limit of object-information contained in scattering patterns. The generalization capability was considerably improved through training the neural network in a physics-informed approach, on the scattering formula itself. Although trained only on simulated data the neural network is capable of interpreting experimental data and predicts a novel shape for silver nanoclusters from publicly available scattering data. In its present state, it is ready for deployment on larger experimental dataset and our method is easily transferable to different single-shot wide-angle reconstruction tasks.

The field optimization for the excitation of a giant dipole state through deep reinforcement learning in Chap. 4 has yielded mixed results. We succeeded in training a neural network in reliably controlling the transition of a hydrogenic exciton into a giant dipole state. The solution is of rather simple character, which however may be attributed to the simplified character of the simulation environment. However, trough its simplicity, the excitation scheme could easily be implemented in in experimental setups and tested. In a next step, our approach should be transferred to a full quantum mechanical simulation of the exciton-environment.

5.2. Outlook

We expect neural networks to become a key tool in the processing of FEL scattering patterns in the near future. With ever increasing repetition rates of FEL experiments [173] and thereby dataset sizes, full analyses are only possible with ultra-fast reconstruction algorithms. The generalization capability of neural networks in interpolating between the known basis shapes further adds to their benefits. The prediction is reinforced by the increasing number of manuscripts published on the neural network aided reconstruction of FEL scattering patterns like Refs. [84–86], with an emphasis on three-dimensional reconstruction from sets of small-angle scattering patterns in the year 2021 with Refs. [87–92].

For the future of reinforcement learning in physics we see potential especially in the control of large scale experiments around a stable operating state. This has already been demonstrated with Refs. [50, 54] on both small-scale experiments and control of large-scale machines.

During the past five years, we did not only observe the developing field of neural network applications in physics but actively contributed to it with Refs. [1, 2] and through peer-reviewing of manuscript. We see a recurring theme in the repeated re-invention of the same idea from different sub-communities, that aim to solve similar problems but are oblivious of each other due to using different terminology. A good example of this are the different branches of optical diffraction, that all aim to reconstruct some kind of real-space density from an image that is, to a varying degree, subject to diffraction signals. Here, we see a strong need for better coordination between the different fields of physics, that could be achieved through interdisciplinary conferences focused on DL applications in physics or natural sciences in general or joint journals.

Overall, we expect deep learning to become a staple tool for solving data-driven problems in physics. The universal representation capabilities of neural networks allows the adaptions to a wide range of tasks. Still, we do not expect DL to fully replace traditional ML methods in physics. Neural networks allow the fast solution of highly complex tasks in environments with an abundance of experimental data. Examples are modern large-scale research facility like FELs or particle accelerators. Therein, neural networks could aid in solving open questions at the frontiers of modern physics.

A. Appendix: Supporting Material

A.1. Computing Hardware List

In obtaining the findings of this thesis, several different computers have been used. because the computation times, or batch sized in case of neural network trainings, scale with the available resources, all computers, identified by their hostname, are listed below.

name	description	OS	CPU	GPU	RAM	disc space
qoms	main workstation	Ubuntu 20 LTS	Intel Xeon E5-1650 $6 \times 3.2\,\text{GHz}$	Nvidia GTX1060	32GB	256GB SSD 500GB HDD
QuantumChaos	personal computer	Windows 10	Intel i7-6700K $4 \times 4.0\,\text{GHz}$	Nvidia GTX1080	32GB	1TB SSD 3TB HDD
rechenknecht	server	CentOS 7	2× Intel Xeon E5-2643 $6 \times 3.4\,\text{GHz}$	Nvidia GTX1080ti	64GB	500GB SSD 2TB HDD
eve	deep learning server	Ubuntu 20 LTS	2× Intel Xeon Silver 4216 $16 \times 2.1\,\text{GHz}$	4× Nvidia RTX2080ti	192GB	2TB SSD 2TB HDD
computationPi	Raspberry Pi 4	Raspberry Pi OS 5.10	Cortex-A72 $4 \times 1.5\,\text{GHz}$	n.a.	4GB	64GB SD card

B. Appendix: Scattering Inversion

B.1. Quaternion Fundamentals

The Quaternions $\mathbb{Q}$ (also sometimes $\mathbb{H}$) are a four-dimensional extension of the complex numbers $\mathbb{C}$. They were discovered by Sir William Rowan Hamilton in 1843 as a solution on how to extend the isomorphism between the $\mathbb{R}^2$ and $\mathbb{C}$ to the $\mathbb{R}^3$ [260, 261]. The final idea for the full concept of quaternions came to Hamilton on a walk along the Royal Canal in Dublin with his wife on the 16th of October 1843 and he carved the formula

$$\mathrm{i}^2 = \mathrm{j}^2 = \mathrm{k}^2 = \mathrm{i}\,\mathrm{j}\,\mathrm{k} = -1 \tag{B.1}$$

into the stone of the Brougham Bride [261, 262]. While the carving is long gone, today, a plaque reminds of Hamilton's idea and yearly the *Hamilton Walk* is celebrated with many prominent mathematicicans and scientists attending [262]. Although the same concept was independently discovered by Olinde Rodrigues in 1840 and Carl Friedrich Gauss in 1819, they missed to publish their results.

The defining properties of the quaternions expressed in Eq. (B.1) are the introduction of two additional imaginary units j and k which do not commute with i or each other. Instead, the sign changes under commutation as

$$\mathrm{i}\,\mathrm{j} = -\mathrm{j}\,\mathrm{i}\,, \tag{B.2}$$

$$\mathrm{j}\,\mathrm{k} = -\mathrm{k}\,\mathrm{j}\,, \tag{B.3}$$

$$\mathrm{k}\,\mathrm{i} = -\mathrm{i}\,\mathrm{k}\,. \tag{B.4}$$

Each quaternion $\mathtt{x}_q$ can be represented by a set of four real numbers $\{x_0, x_1, x_2, x_3\}$ through

$$\mathtt{x}_q = x_0 + x_1\,\mathrm{i} + x_2\,\mathrm{j} + x_3\,\mathrm{k}\,. \tag{B.5}$$

Like complex numbers, quaternions can be separated into a real part x_0 and an imaginary part $x_1\,\mathrm{i} + x_2\,\mathrm{j} + x_3\,\mathrm{k}$. Further, complex conjugation is defined by

$$\overline{\mathtt{x}_q} = x_0 - x_1\,\mathrm{i} - x_2\,\mathrm{j} - x_3\,\mathrm{k}\,, \tag{B.6}$$

which allows for the definition of a scalar norm

$$|\mathtt{x}_q|^2 = \mathtt{x}_q\,\overline{\mathtt{x}_q} = x_0^2 + x_1^2 + x_2^2 + x_3^2\,. \tag{B.7}$$

Another representation for the imaginary part of the quaternion is that of a vector $\boldsymbol{x} = (x_1, x_2, x_3)^T$, allowing to express a full quaternion through the form

$$\mathtt{x}_q = (x_0,\ \boldsymbol{x})\ . \tag{B.8}$$

This vector notation comes in handy when writing down the quaternion product of two quaternions a_q and b_q

$$\begin{aligned}\mathtt{a}_q\,\mathtt{b}_q =& a_0\,b_0 - a_1\,b_1 - a_2\,b_2 - a_3\,b_3\\ &+ (a_0\,b_1 + a_1\,b_0 + a_2\,b_3 - a_3\,b_2)\ \mathrm{i}\\ &+ (a_0\,b_1 - a_1\,b_3 + a_2\,b_0 + a_3\,b_1)\ \mathrm{j}\\ &+ (a_0\,b_2 + a_1\,b_2 - a_2\,b_1 + a_3\,b_0)\,\mathrm{k}\end{aligned} \tag{B.9}$$

as

$$\mathtt{a}_q\,\mathtt{b}_q = (a_0,\ \boldsymbol{a})\ (b_0,\ \boldsymbol{b}) = (a_0\,b_0 - \boldsymbol{a}\cdot\boldsymbol{b},\ a_0\,\boldsymbol{b} + b_0\,\boldsymbol{a} + \boldsymbol{a}\times\boldsymbol{b}) \tag{B.10}$$

through utilizing the vectorial scalar product $\cdot$ and cross product $\times$.

The vector notation of quaternions already reveals the relation between quaternions and the $\mathbb{R}^3$, in that the imaginary (or pure) quaternions are isomorphic to the $\mathbb{R}^3$. If we take two vectors $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^3$, then the corresponding imaginary quaternions are $\mathtt{x}_q = (0, \boldsymbol{x})$ and $\mathtt{y}_q = (0, \boldsymbol{y})$. In this formulation, addition and multiplication with a real valued scalar are expressed in the same way in quaternion space and $\mathbb{R}^3$. However, both scalar product and crossed product are yielded by the quaternion product from the real and imaginary part, respectively, through

$$\boldsymbol{x}\cdot\boldsymbol{y} = \mathrm{Re}\left\{\mathsf{x}_q\,\overline{\mathsf{y}_q}\right\}\ , \tag{B.11}$$

$$\boldsymbol{x}\times\boldsymbol{y} = \mathrm{Im}\left\{\mathsf{x}_q\,\mathsf{y}_q\right\}\ . \tag{B.12}$$

The rotation of pure quaternions can be achieved by utilizing unit quaternions, similar to the rotation of complex numbers by multiplication with a factor of $\mathrm{e}^{\mathrm{i}\phi}$. Unit quaternions u_q are of unit length $|\mathsf{u}_q| = 1$ and conversely can be inverted by conjugation $\mathsf{u}_q^{-1} = \overline{\mathsf{u}_q}$. Through Euler's rotation theorem, it is known that any rotation in $\mathbb{R}$ is fully defined by a single rotation of an angle α about a single axis $\boldsymbol{n}$. Such a rotation translates to the rotation quaternion

$$\mathtt{q}_{\mathrm{rot}} = \left(\cos\left(\frac{\alpha}{2}\right)\,,\ \sin\left(\frac{\alpha}{2}\right)\,\boldsymbol{n}\right)\ , \tag{B.13}$$

which rotates the pure quaternion $\mathtt{x}_q = (0,\ \boldsymbol{x})$ by

$$\mathsf{x}_q' = \mathtt{q}_{\mathrm{rot}}\,\mathsf{x}_q\,\mathtt{q}_{\mathrm{rot}}^{-1} = \big(0, (\boldsymbol{x} - \boldsymbol{n}\,(\boldsymbol{n}\cdot\boldsymbol{x}))\cos\alpha + (\boldsymbol{n}\times\boldsymbol{x})\sin\alpha + \boldsymbol{n}\,(\boldsymbol{n}\cdot\boldsymbol{x})\big)\ . \tag{B.14}$$

The imaginary quaternion expression in Eq. (B.14) is equivalent to Rodrigues' rotation formula [170]. By successively rotating $\mathtt{x}$ by two rotation quaternions $\mathtt{q}$ and $\mathtt{p}$

$$\mathsf{x}_q' = \mathsf{p}\,\mathsf{q}\,\mathsf{x}_q\,\bar{\mathsf{q}}\,\bar{\mathsf{p}} = (\mathsf{p}\,\mathsf{q})\,\mathsf{x}_q\,\overline{(\mathsf{p}\,\mathsf{q})} \tag{B.15}$$

it becomes apparent, that subsequent rotations can be contracted into a single rotation quaternion by simple multiplication.

Historically, the quaternion algebra with imaginary quaternions paved the way for defining the first algebra on the real space $\mathbb{R}^3$. Subsequently, the modern vector algebra was developed by Willard Gibbs and Oliver Heaviside in the late 19th century on the foundation of the quaternion algebra. The modern notation of vector algebra was formalized in 1901 with Ref. [263] and as a relic from the quaternion origin some textbooks still introduce the cross product with basis elements named $\mathrm{i}, \mathrm{j}, \mathrm{k}$. However, there still remain some fields of applications for quaternions, especially for unit quaternions in the representation and handling of rotations.

B.2. Scattering Patterns of Platonic and Archimedean Solids

The Platonic and Archimedean solids all share either the tetrahedral, cubic or icosahedral symmetry (aside from the snub solids). When grouped in their respective symmetry group, a transition between the scattering patterns along equal axis can be observed, as shown in Fig. B.1. In each case, the Platonic solids form the edge cases with the cube and octahedron for cubic symmetry O_h and icosahedron and dodecahedron for the icosahedral symmetry I_h, respectively, while the Archimedean Solids combine features of both edge cases.

B. Appendix: Scattering Inversion

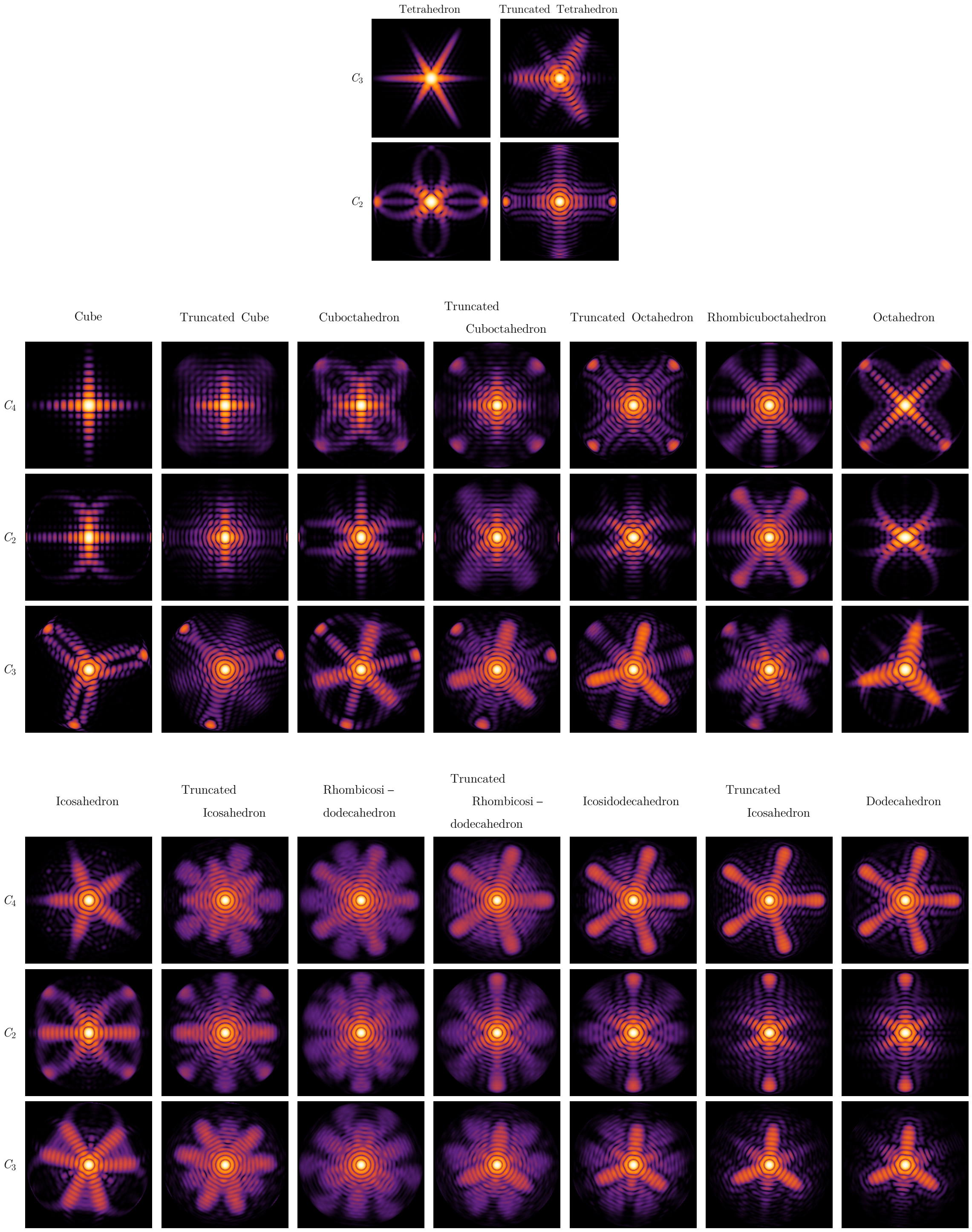


Figure B.1: All Platonic and most Archimedean solids share either the tetrahedral T_d, cubic O_h or icosahedral I_h symmetry group. Grouped accordingly, the scattering patterns of the Archimedian solids (along shared axes) combine features from the related Platonic solids (edge cases) to varying degrees.

B.3. Scattering Inversion for Two Dimensional Shutters with Neural Networks

The reconstruction of discretized object densities from scattering patterns in Sec. 3.4 is simplified by removing one dimension from the real-space volume. This is either done by removing a lateral dimension reducing to a two-dimensional slice parallel to the optical axis, or by removing the propagation dimension, resulting in an effective small-angle problem. The effective density projection observed in small-angle scattering usually is an inhomogeneous density function, recall Eq. (3.3). The case of a binary two-dimensional object density, just as assumed in Ref. [86], is more similar to the setup of a shutter, cut from a solid plane, as depicted in Fig. B.2. This exact experiment is realizable on a macroscopic level using a low-intensity optical laser. Due to the smaller dimensionality, training times are much smaller compared to the full three-dimensional problem in Sec. 3.4. Throughout the development of this thesis the shutter-system served as a testbed for architectures and loss functions, especially the much simpler scatter loss function. Yet, lacking a wide-angle interference information the inversion problem becomes degenerate introducing a new challenge unique to the small-angle regime. This section follows the general structure of the sections of Chap. 3 with even demonstrating the evaluation on experimental data, obtained through an improvised setup.

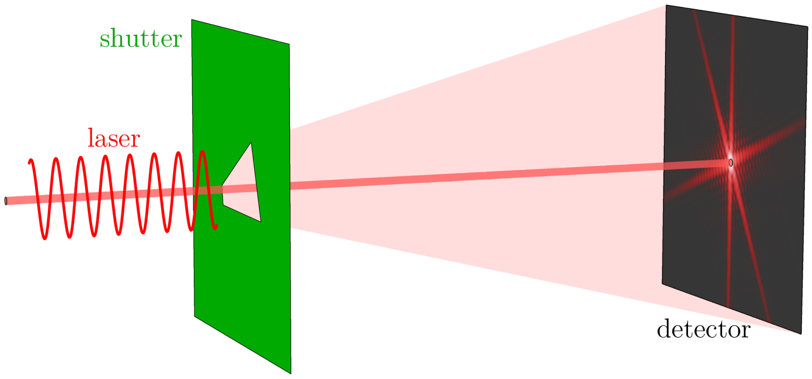


Figure B.2: The scattering on two-dimensional object densities is equivalent to diffraction of optical light (like a laser) on an opening in a two-dimensional plane.

The far-field scattering intensity for a two-dimensional object density $\rho(x,y)$ is that of the Fraunhofer approximation [144]

$$I_{\text{scatt}}(q_x,q_y) \sim \left| \int \rho(x,y)\, \mathrm{e}^{\mathrm{i}\,(q_x\,x+q_y\,y)}\,\mathrm{d}x\,\mathrm{d}y \right|^2 \tag{B.16}$$

$$= \left| \mathcal{FT}\left[\rho(x,y)\right](q_x,q_y) \right|^2 . \tag{B.17}$$

This projection from the object density $\rho(x,y)$ to the scattered intensity $I(q_x,q_y)$ is inherently degenerated, due to the loss of the phase information. The two degeneracies are most easily demonstrate, when assuming a one-dimensional Fourier transform

$$f(q) = \int_{-\infty}^{\infty} \rho(x)\, \mathrm{e}^{2\pi\mathrm{i}\,x\,q}\,\mathrm{d}x\,. \tag{B.18}$$

The first degeneracy, also present in wide-angle scattering, is that of translational invariance. For a finite object density, that is constant for $|x| \to \infty$, a finite shift d results in

$$\tilde{f}(q) = \int_{-\infty}^{\infty} \rho(x+d)\, \mathrm{e}^{2\pi\mathrm{i}\, x\, q}\, \mathrm{d}x \tag{B.19}$$

$$\overset{x\to x-d}{=} \int_{-\infty}^{\infty} \rho(x)\, \mathrm{e}^{2\pi\mathrm{i}\,(x-d)\, q}\, \mathrm{d}x = \mathrm{e}^{2\pi\mathrm{i}\, d\, q}\, f(q)\,, \tag{B.20}$$

which differs from Eq. (B.18) by nothing but a phase factor, that vanishes upon intensity measurement. The second degeneracy, not present in wide-angle scattering, is that of inversion symmetry. For the mirrored density $\rho(-x)$, the Fourier transform,

$$\tilde{f}(q) = \int_{-\infty}^{\infty} \rho(-x)\, \mathrm{e}^{2\pi\mathrm{i}\, x\, q}\, \mathrm{d}x \tag{B.21}$$

$$\overset{x\to x-d}{=} \int_{-\infty}^{\infty} \rho(x)\, \mathrm{e}^{-2\pi\mathrm{i}\,(x)\, q}\, \mathrm{d}x = \overline{f(q)}\,, \tag{B.22}$$

is the complex conjugate of Eq. (B.18), that again becomes identical under intensity measurement. Both degeneracies are problematic for the training of a neural network in solving the inverse problem of $I(q_x, q_y) \to \rho(x, y)$ and have to be accounted for.

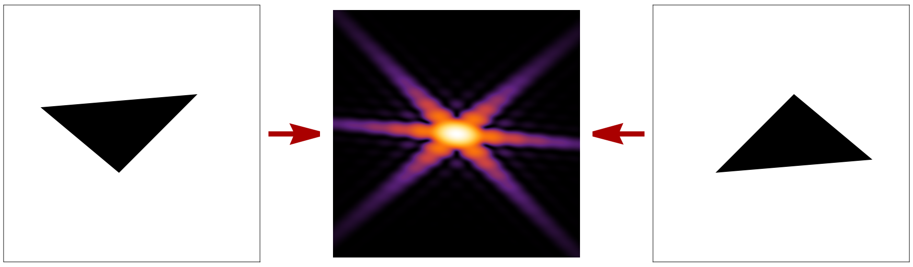


Figure B.3: The two triangles in the left and right panel are inversion symmetric, in that they were produced by point mirroring at the panel-center, which does not coincide with their center-of-mass. Still, they both produce the exact same scattering intensity profile in the central panel.

The first degeneracy of translational invariance can already be eliminated in the dataset generation stage by defining a common point of origin. This is achieved by calculating the center-of-mass of the binary density functions and shifting it to the origin. For the rotation symmetry two approaches were explored.

1. The degeneracy is removed in the data generation stage by always rotating object densities "point-up". For polynomial densities the mean-position of all vertices above and below the x-axis is calculated. The density is then mirrored around the origin to place the larger vertex-mean-norm above the x-axis resulting in a "pointy-end-up" orientation. Recalling Sec. 3.2.2, this can be understood as a fundamental-domain projection. Still, it struggles with close-to axis-symmetric objects.

2. The loss function is tailored to both rate the original target density and its mirror image as correct densities. It is implemented by comparing the prediction to both original and mirrored target and returning the better loss of the two.

The latter is implemented in Sec. B.3.2. Anyhow, all of these degeneracies are handled intrinsically, when training the neural network in a physical loss scheme, similar to the scatter- and binary-loss of the wide-angle case from Sec. 3.4.4. This training regime was first tested on the two-dimensional system and is further introduced in Sec. B.3.2. As a side-note, none of the above effects seems to be accounted for in Ref. [86], which appears puzzling to the author.

B.3.1. Shutter Data Generation

Similar to the three dimensional case in Sec. 3.4 the object densities are rasterized as binary tensors, but with just two-dimensions. The densities are sampled from random n-gons with $n \in [3, 8]$, and circles (see Fig. B.4). The n-gons are created by choosing n points from the unit circle via uniformly sampling their angles and multiplying them with a size parameter relative to the size of the object space. The vertices are further translationally normalized by shifting them for the center-of-mass of the resulting figure to coincide with the origin in the center of the real-space window. Any density is rasterized on a grid of 128 × 128 pixels. The main dataset is created from 30 000 random samples. Further, a test-set of 100 random n-gons with 9–12 vertices is created. For scattering calculation the object densities are zero-padded to 512 pixels and the Fourier-transformed. From the resulting scattering field the central 256 × 256 pixels are taken as the intensity profile and plotted on a logarithmic scale, like in Fig. B.4.

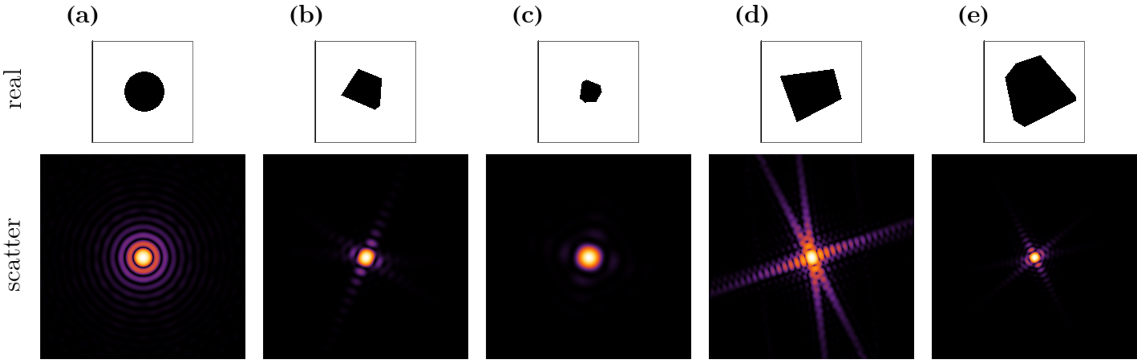


Figure B.4: A set of samples from the main dataset consisting of circles (a) and random n-gons with 3–8 vertices (b)-(d).

Meant as a toy model, all scattering simulations were performed scale free and due to the lack of a wide-angle phase it is in fact scale independent. If we assume an arbitrary side-length l_{real} of the real-space window the grid spacing is $\Delta r = l_{\text{real}}/128$. It translates into the transfer-momentum range $q_{\text{range}} = \pi/\Delta r$ covered by the Fourier-transform that in turn determines the Fourier grid-spacing $\Delta q = q_{\text{range}}/512$. Finally, in the small-angle approximation, the transfer momentum is connected to the radial scattering angle θ by $q = k \sin(\theta)$ with wave number $k = 2\pi/\lambda$. In this relation, assuming a different q_{range}, for example by a false scaling of an input image by a factor s, is equivalent to scaling the corresponding object by a factor of $1/s$ without any further changes to the scattering pattern.

B.3.2. Symmetric and Physics Informed Loss

The inherent inversion symmetry poses an additional challenge for training a neural network in the reconstruction of small-angle scattering patterns. Nevertheless, it is an ideal system for demonstrating the application of a symmetric loss function, like proposed but not implemented in Sec. 3.2.2. The binary nature of the object densities favors the binary cross-entropy (BCE) loss from Eq. (2.6). The symmetric BCE loss is defined as

$$H_{\mathrm{symm}}(\mathbf{y}, \mathbf{p}) = \min\big[H(\mathbf{y}, \mathbf{p}), H(\mathbf{J}\,\mathbf{y}\,\mathbf{J}, \mathbf{p})\big]\,, \tag{B.23}$$

with the exchange matrix $\mathbf{J}$ generating the point-inverted object density $\mathbf{J}\,\mathbf{y}\,\mathbf{J}$. It returns the best-fit BCE loss to either the original or point-inverted version of the target tensor $\mathbf{y}$.

The second approach to handle the underlying symmetries inside the loss function is by not comparing object tensors but the corresponding scattering patterns via the scatter loss introduced in Sec. 3.4.4. In the small-angle regime the scatter loss

$$L_{\mathrm{scatter}}(\mathbf{y}, \mathbf{p}) = \Big\{\log\Big(\big|\mathcal{FT}\,[\mathbf{y}]\big|^2 + n_d\Big) - \log\Big(\big|\mathcal{FT}\,[\mathbf{p} + \boldsymbol{\epsilon}]\big|^2 + n_d\Big)\Big\}^2 \tag{B.24}$$

is the mean-squared difference between the logarithmized intensity patterns (with constant dark-noise n_d) of the Fourier transforms of both prediction and target object tensors. The prediction object tensor is modified by a small noise tensor $\boldsymbol{\epsilon}$ of uniform noise with magnitude 10^{-3} to zero-predictions, that causes value exceptions in the Fourier transform. Again, an additional binary loss

$$L_{\mathrm{b}}(\mathbf{t}, \mathbf{p}) = \frac{1}{N^2} \sum_{i,j=1}^{N} (p_{i,j})^2\,(1 - p_{i,j})^2\,, \tag{B.25}$$

is needed to enforce the binary nature of the object densities. Together, they form the (small-angle) version of the physical loss function

$$L_{\mathrm{phys}}(\mathbf{t}, \mathbf{p}) = L_{\mathrm{s}}(\mathbf{t}, \mathbf{p}) + w\,L_{\mathrm{b}}(\mathbf{t}, \mathbf{p})\,, \tag{B.26}$$

with a balancing hyperparameter $w = 0.001$.

B.3.3. Network Design & Training

The neural network for the reconstruction of two dimensional shapes from their corresponding scattering pattern, ShutterNet, is designed in a u-Net pattern, following Ref. [26]. The architecture is sketched in Fig. B.5. Derived from an autoencoder architecture with encoder- and decoder stage it includes additional skip connections spanning from encoder- to decoder-sections with equal tensor sizes. These aid in providing later stages of the network with better access to original positional features of the image, that do not have to be transported through the latent space. As in Sec. 3.4.3, each convolution layer marked in Fig. B.5 includes heavy regularization. Inspired by Ref. [109], they are further designed in a pre-activation scheme. Aside from the initial 5 × 5 conv-layer with stride 2, each 3 × 3 conv-layer in Fig. B.5 is a stack of

1. batch normalization,
2. ReLU activation,
3. 3 × 3 convolution,
4. dropout regularization with rate 0.2.

The pre-activation scheme was found to not offer any benefit over the classical post-activation. Therefore, it was not implemented in the VoxelNet architecture in Sec.3.4.3. The ShutterNet has a total of approximately 34.5 million trainable parameters.

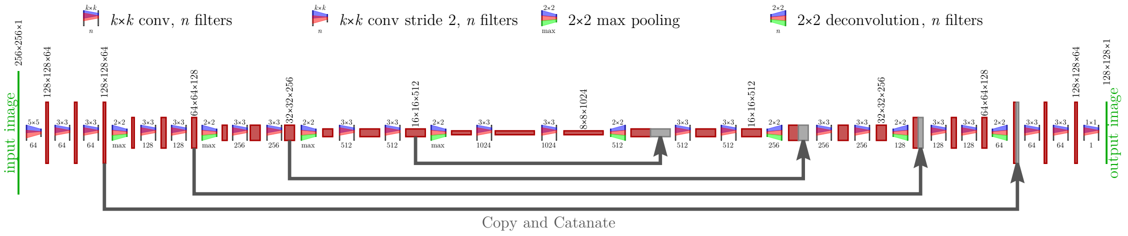


Figure B.5: The ShutterNet is designed after the original u-net scheme from Ref. [26]. In principle an autoencoder-architecture, it employs additional skip-connection from the encoder- to the decoder-stage of the network, that transport tensors with equal lateral dimensions. By concatenating these earlier features maps, the later stages of the decoder are provided with further spatial structure information.

The training of the ShutterNet is performed under the same conditions as the VoxelNet in Sec.3.4.5 using the TensorFlow 2.3.1 Keras framework and Python 3.6.6. The main dataset is split at a ratio of 0.2 into training and validation set, and the images are read from the hard-drive parallel to training. The training set is shuffled at each epoch and the scattering patterns are augmented using the exact same augmentations as for the VoxelNet in Sec. 3.4.2, regardless whether they are sensible for the given problem. The network is trained for 200 epoch with a batch size of 32 using the ADAM optimizer. Unlike the wide-angle scattering, the calculation of the scatter loss adds no significant overhead to the training time. The training takes 24 h in QuantumChaos and 20 h on eve (see App. A.1 for hardware configurations). During training, both the scatter loss and the symmetric mean-squared-error defined equally to Eq. (B.23) are recorded as metrics. Both the learning curves for training with symmetric BCE loss and physical loss are shown in Fig. B.6(a) and (b), respectively.

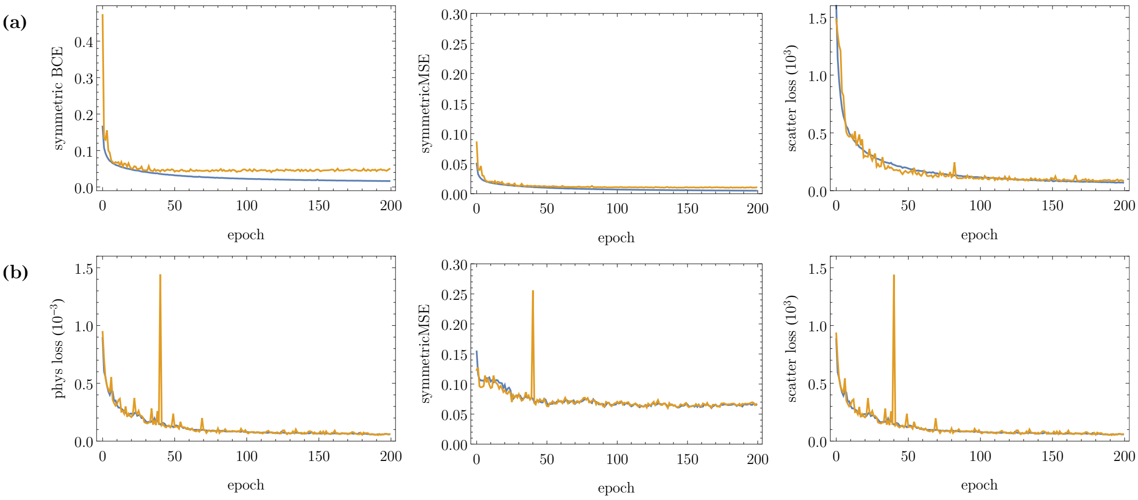


Figure B.6: The training of the ShutterNet with (a) symmetric BCE loss shows minimal overfitting (left panel) between the training set (blue curve) and validation set (orange curve). The same effect is also visible in the symmetric-means-squared error metric, but with a smaller magnitude. The scatter loss recorded over the training run (right panel) shows no clear sign of overfitting. The learning curve (b) of the training with physical loss shows absolutely no overfitting with the validation curve closely following the training curve. The symmetric mean-squared-error is much worse than for the BCE training, due to the unaccounted translation invariance. The pure scatter loss metric is slightly better than for the BCE training.

The training with symmetric BCE loss in Fig. B.6(a) shows minimal overfitting (left panel) between the training set (blue curve) and validation set (orange curve). The same effect is also visible in the symmetric-means-squared error metric, but with a smaller magnitude. The scatter loss recorded over the training run (right panel) shows no clear sign of overfitting. This indicates that the reconstruction accuracy reaches close to the limit of information contained within the input scattering patterns. The training with the physical loss from Eq. (B.26) in Fig. B.6(b) shows no overfitting. Instead, the validation curve closely follows the training curve with just a few outliers. The symmetric mean-squared error in the center panel is much worse than for the BCE training, because the physical loss is invariant to translations of the object density, and thus performs no centering as present in the dataset. The scatter loss in the right panel converges to even better values than the BCE training, which comes at little surprise as it is the direct optimization goal of the training.

B.3.4. Evaluation on Test Data

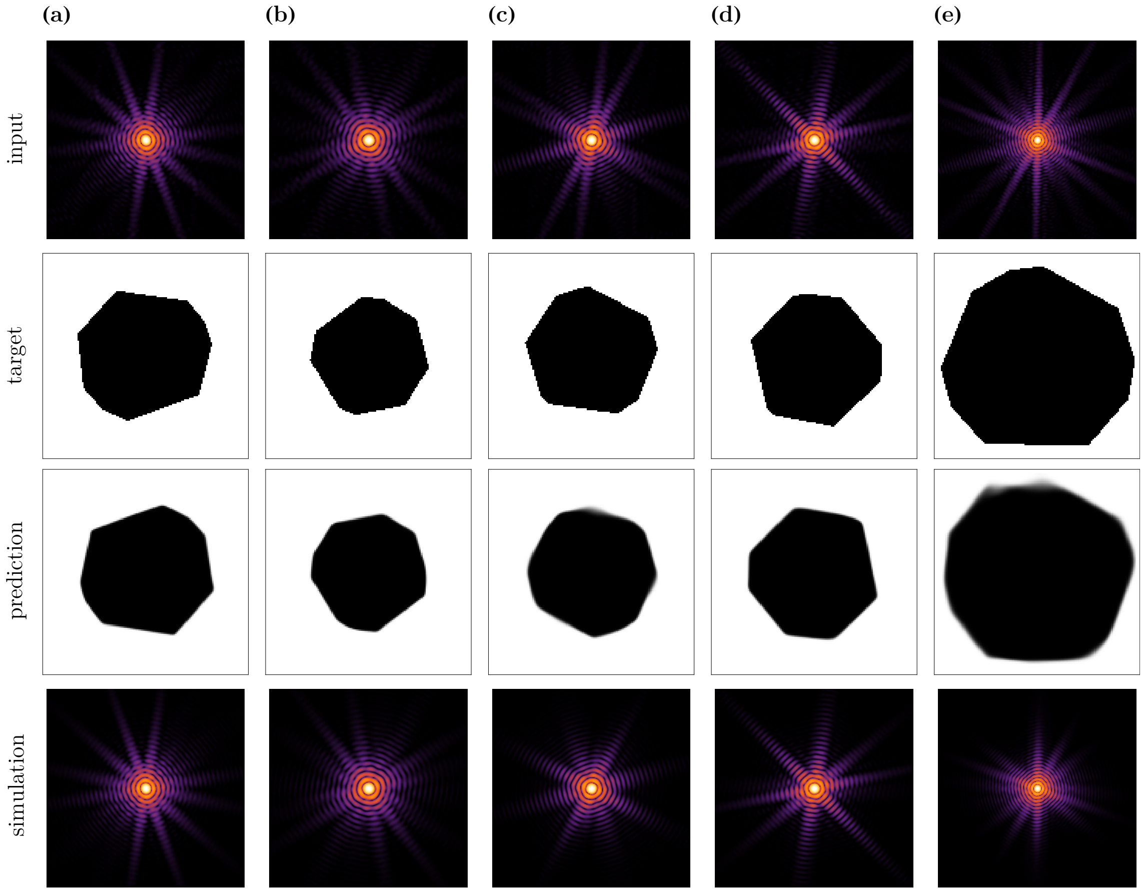


Figure B.7: Test set prediction results of the ShutterNet trained with the symmetric BCE loss. The predictions on samples (a), (b), and (d) show good agreement with the targets. On samples (c) and (e), however, the predictions appear close to point-symmetric and include non-binary (gray colored) regions of low prediction confidence. The corresponding scattering patterns (bottom row) deviate significantly from the inputs (top row) at large scattering angles (outer regions).

Both symmetric and physical trained ShutterNets are tested on the pre-generated test set composed of n-gons with 9–12 edges. With the training set containing 3–8-gons and circles (∞-gons) it is hypothesized, that the trained neural networks should be capable of interpolating such shapes. The predictions on sample of 5 training set elements are shown in Figs. B.7, B.8.

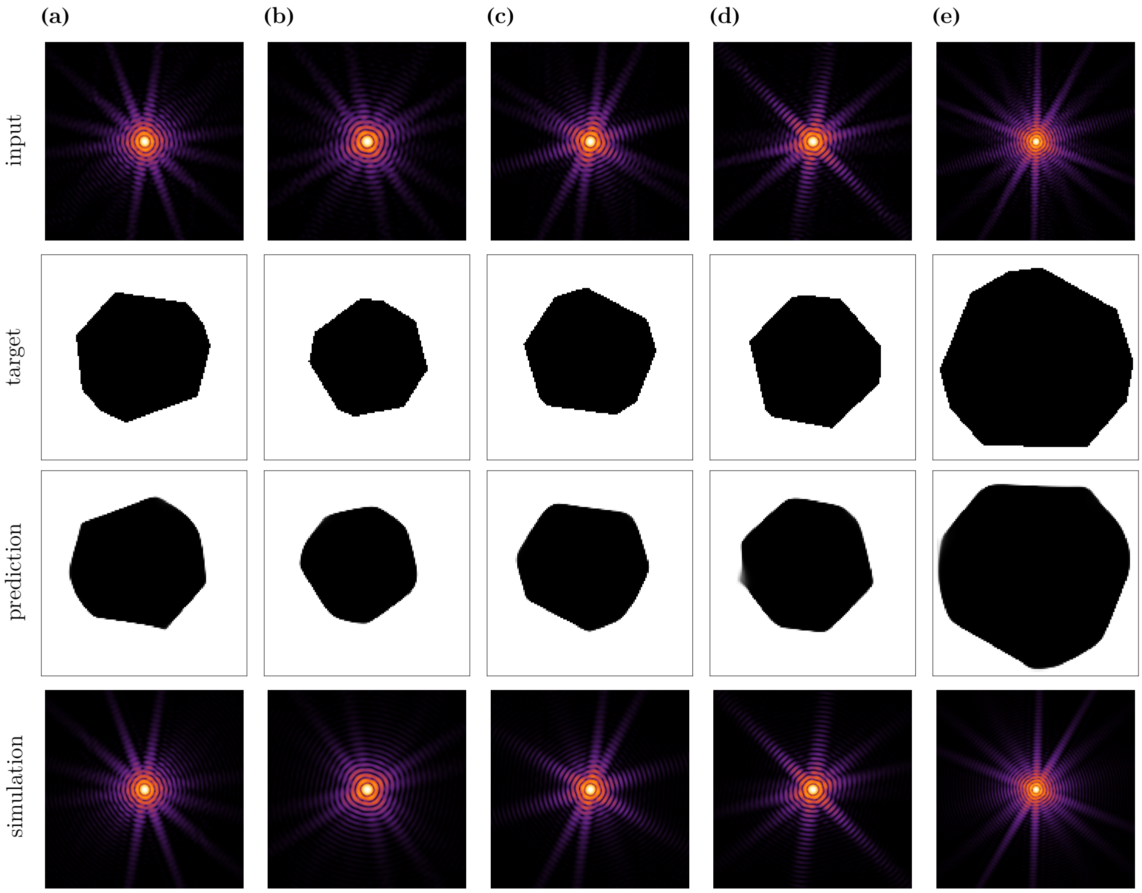


Figure B.8: Test set prediction results of the ShutterNet trained with the physics loss. The predictions are near perfect reconstruction of the target object densities and only (d) shows a minor defect in form of a small extrusion. The simulated scattering patterns are in perfect agreement with the input patterns.

The symmetric BCE trained ShutterNet succeeds in reconstructing most shapes, like the samples (a), (b), and (d) in Fig. B.7. On samples (c) and (e), however, the predictions include non-binary regions (gray colored), resulting in a overall rounded appearance. Especially in Fig. B.7(e), the predicted tensor is nearly a superposition of both the target and point-inverted target. The corresponding scattering patterns (bottom row) deviate significantly from the inputs (top row) at large scattering angles (outer regions) and lack entire reflexes.

On the other side, the physics loss trained ShutterNet successfully handles all samples of the test set, shown in Fig. B.8. Only a very minor artifact is visible on (d), with a small non-binary extrusion, that has no impact on the scattering pattern. Overall, the simulated scattering patterns achieve an excellent agreement with the input images.

B.3.5. Application to Experimental Data

The scattering setup sketched in Fig. B.2 can be realized with all optical (meaning the visible spectrum) components and only commonplace materials. As a light source, a cheap laser-diode with $\lambda = 650\,\text{nm}$ is used. Shutter shapes are cut from a sheet of paper using a

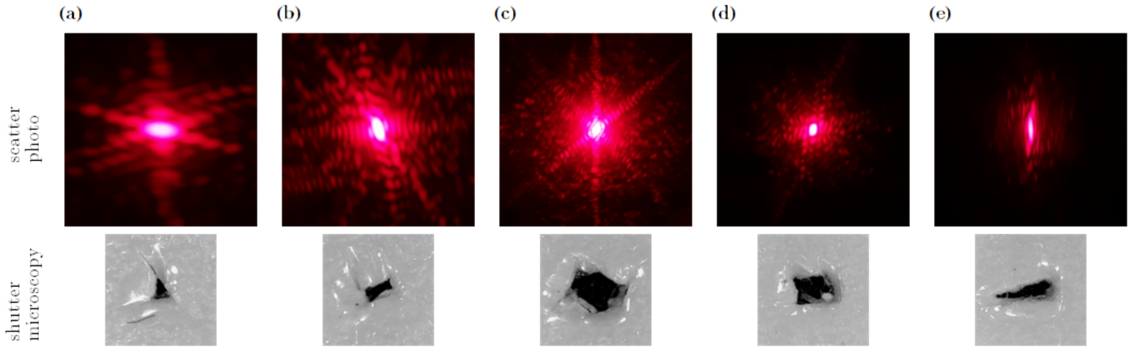


Figure B.9: The diffraction patterns (top row) created with a red laser diode and paper shutters are photographed using a normal camera. The corresponding shutter shapes (bottom row) are obtained using an optical USB microscope. The images are rectangular cropped and resized to conform with the shapes required by ShutterNet.

scalpel. The shutter is fixed in the laser beam using clothespins and the scattering pattern is projected down a 5 m-long hallway onto a white sheet of A4 paper, attached to a wall. The scattering patterns of five different shutters are photographed using an ordinary camera. The corresponding shutter shapes are obtained using an USB microscope. The acquired dataset is shown in Fig. B.9. The view ranges are matched using the relation between λ, l_{real} and q_{range} described in Sec. B.3.1. Fur further processing, the shutter images are binarized by manually determining a threshold value and the scattering patterns are converted to gray-levels, ignoring the non-linear response of the multi-color CCD sensor. The monochromatic response is assumed to be logarithmic.

Again, both versions of ShutterNet are tested on this experimental dataset. The symmetric BCE trained version is incapable of handling the experimental data and fails entirely in its predictions, as apparent from Fig. B.10. Just the triangle in (a) is a valid object density that also bears a resemblance to the target shape. All other predictions have large uncertainty regions. The binarized versions of the shutter photographs in the second row reveal that the shapes are not cut clean from the paper. The triangle in (a) is surrounded by additional line-cuts and the edges of the other patterns are not straight but have paper-fibers pertruding. They have a significant impact on the experimental scattering patterns in the top row. They add interferences resulting in much more irregular fringe patterns and point clouds, that obscure the main pattern produced by the global structure. Such shapes lie clearly outside the training set space of convex shapes.

In contrast, the physical trained version of ShutterNet is capable of processing the experimental data to some extend. It reconstructs both the triangles in (a) and (e) to an acceptable accuracy ignoring the small perturbations to the object shape by paper fibers or stray cuts. On the other samples, (b), (c), and (d), it returns valid, binary object densities, in stark contrast to the BCE version in Fig. B.10. However, they deviate strongly from the target shapes obtained by microscopy, which are the most perturbed samples. Still, the simulated scattering patterns succeed in re-creating the most prominent reflexes from the input scattering patterns.

B.3.6. Further Development

The reconstruction of two-dimensional binary object densities from single (small-angle) scattering patterns mainly served as a toy model and testbed in the development of the

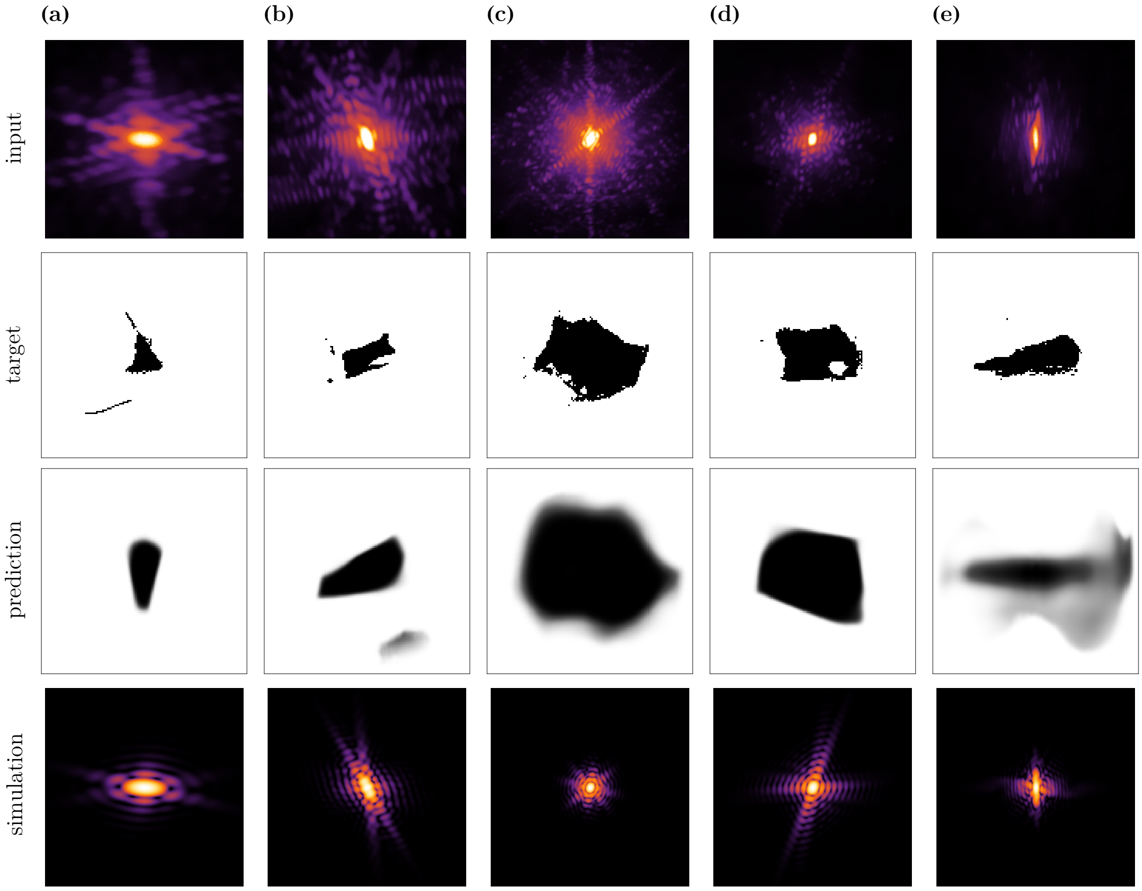


Figure B.10: The symmetric BCE trained version of ShutterNet fails entirely on the experimental dataset. Just the triangle in (a) bears a little resemblance of the input pattern. All other predictions show large regions of intermediate, uncertain predictions.

three-dimensional object reconstruction from single-shot wide-angle scattering patterns in Sec. 3.4. A similar two-dimensional reconstruction approach has been published in Ref. [86], but lacking the symmetry considerations of Sec. B.3.2. Small-angle scattering experiments in FELs are usually performed with reproducible targets allowing the collection of three-dimensional Fourier-densities [136]. Just recently, neural networks capable of reconstructing three-dimensional object densities from such Fourier-volumes have been developed based upon three-dimensional versions of u-net [87, 90].

Still, the simplicity of the system assumed in this appendix make it a perfect environment for demonstrating different techniques. With the easy-to-understand all optical setup and relatively fast training it could be implemented as an advanced student lab course. It would cover the topics of optical diffraction in the Fraunhofer limit, numerical phase retrieval and fundamentals of neural networks. In order to improve reconstruction results the manufacturing of shutters and recording of scattering patterns need to be improved. Further, the training dataset could be extended to also cover ellipses. A more advanced extension would be to modify the input shutter shapes with random non-convex defects and simulate scattering patterns on-the-fly to obtain robustness against the defects observed in Fig. B.11. Overall, developing such a lab course could help in introducing physics students to the rapidly advancing field of deep learning and possible applications in physics.

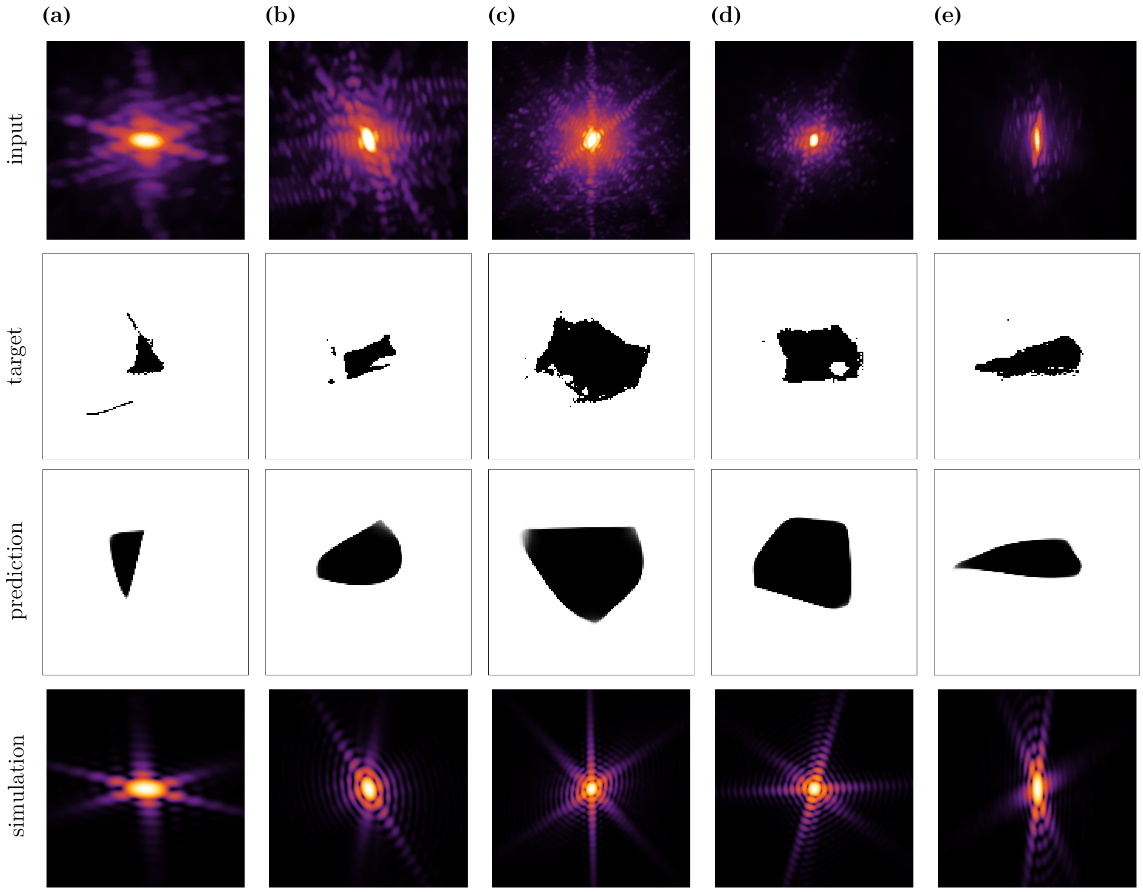


Figure B.11: The physical trained version of ShutterNet is to some extend capable of handling the experimental training data. The reconstruction of the triangular shapes in (a) and (e) is acceptable. On samples (b), (c), and (d), the shapes clearly deviate. These shapes are, however, the ones with the strongest perturbations from paper fibers pertruding into the shutter. Nevertheless, the simulations for samples (b) and (c) succeed in re-creating the position of the main-reflexes of the input scattering patterns.

C. Appendix: Giant Dipole States of Excitons

C.1. Generalized Atomic Units

Many fields of physics make use of special, non-SI unit systems. A popular example are the natural units used in high-energy physics, directly illustrating the energy-mass equivalence. Theoretical works in atomic physics often use the so-called atomic units, which are abbreviated with a.u. or au. From a theoretical standpoint, the introduction of atomic units can be motivated by the desire to eliminate all unnecessary constants from the eigenequation

$$\left(-\frac{\hbar^2}{2\mu}\Delta - \frac{e^2}{4\pi\varepsilon}\frac{1}{|\boldsymbol{r}|} - E\right)\psi(\boldsymbol{r}) = 0\,, \tag{C.1}$$

of a hydrogenic one-particle system with reduced mass μ within a surrounding medium of permittivity ε. In atoms, the reduced mass μ is often set equal to the electron mass m_0 due to the huge mass difference between electron and proton without introducing much of an error. For different systems, however, where the masses of positively and negatively charged particles are of similar magnitude, the reduced mass has to be used. The reduction of Eq. (C.1) can be accomplished with just two units. The first atomic unit to be defined is usually the Bohr radius

$$a_0 = \frac{4\pi\varepsilon\hbar}{\mu e^2} \tag{C.2}$$

as the unit of length. It is followed by the Hartree energy

$$E_\mathrm{H} = \frac{\hbar^2}{\mu a_0^2} \tag{C.3}$$

as the unit of energies. By replacing $\boldsymbol{\rho} = \boldsymbol{r}/a_0$ and $\mathcal{E} = E/E_\mathrm{H}$, the hydrogenic eigenequation (C.1) is reduced to

$$\left(-\Delta_{\boldsymbol{\rho}} - \frac{1}{|\boldsymbol{\rho}|} - \mathcal{E}\right)\psi(\boldsymbol{\rho}) = 0\,, \tag{C.4}$$

where all constants are absorbed into the units of the variables. In the same way, any other atomic unit can be deduced from their interaction with the Hamiltonian of the hydrogenic system. A list of atomic units relevant to this work can be found in Tab. C.1 together with their value for excitons in Cu_2O. In reverse, the introduction of atomic units can also be motivated by setting the four base units reduced mass μ, elementary charge e, reduced Planck's constant $\hbar$, and Coulomb's constant $k_e = 1/(4\pi\varepsilon)$ to unity, each. The use of atomic units offers several advantages. For once, all unnecessary prefactors vanish from equations, which simplifies writing and helps in identifying the most important features. It also simplifies the linking to standard solutions from mathematical literature. Further, quantities are normalized to the ground state dimensions of the system and so may be represented as float-point numbers without exponents for numerical calculations. This in

Table C.1: Atomic units of various physical constants appearing throughout this thesis with their corresponding conversions factors to SI units for hydrogenic excitons in Cu_2O with masses $m_\mathrm{e} = 0.99\,m_0$, $m_\mathrm{h} = 0.65\,m_0$ and relative permittivity $\varepsilon_\mathrm{r} = 7.5$ ($\varepsilon = \varepsilon_\mathrm{r}\,\varepsilon_0$).

Quantity	Atomic Unit	SI value
mass	$\mu = m_e\,m_h/(m_e+m_h)$	$3.331\,597\,478 \cdot 10^{-31}\,\mathrm{kg}$
charge	e	$1.602\,176\,634 \cdot 10^{-19}\,\mathrm{C}$
length	$a_0 = (4\pi\varepsilon\hbar)/(\mu e^2)$	$1.075\,988\,196 \cdot 10^{-9}\,\mathrm{m}$
energy	$E_\mathrm{H} = \hbar^2/(\mu a_0^2)$	$2.883\,266\,352 \cdot 10^{-20}\,\mathrm{J}$
time	$\hbar/E_\mathrm{H}$	$3.657\,559\,397 \cdot 10^{-15}\,\mathrm{s}$
velocity	$v_0 = a_0\,E_\mathrm{H}/\hbar$	$2.941\,820\,158 \cdot 10^{5}\,\mathrm{m/s}$
momentum	$k_0 = \hbar/a_0$	$9.800\,960\,619 \cdot 10^{-26}\,\mathrm{kg\,m/s}$
angular momentum	$\hbar$	$1.054\,571\,726 \cdot 10^{-34}\,\mathrm{J\,s}$
magnetic field strength	$B_0 = \hbar/(ea_0^2)$	$5.661\,155\,614 \cdot 10^{2}\,\mathrm{T}$
electric field strength	$E_0 = E_\mathrm{H}/(ea_0)$	$1.665\,410\,170 \cdot 10^{8}\,\mathrm{V/m}$

turn prevents numerical instabilities due to limited precision in the presence of extreme magnitudes. When using their specific atomic units, different hydrogenic systems are easier to compare than in SI units and may be converted by simply replacing the corresponding units.

On the other hand, the negligence of all prefactors can hinder the transfer of results and equations to systems with slightly different parameters and may require full re-derivations in order to trace the correct prefactors and constants. Hence, in this thesis, all equations are written in SI units, while numerical calculations are performed in atomic units. Results are usually given in the most descriptive unit. For example, the relative distance within the exciton is given in Bohr radii a_0, while energies are given in eV.

C.2. Interpreting the Pseudomomentum

The pseudomomentum

$$\boldsymbol{k}_i = m_i \dot{\boldsymbol{r}}_i + q_i\, \boldsymbol{B} \times \boldsymbol{r}_i = \boldsymbol{p}_i - q_i\, \boldsymbol{A}(\boldsymbol{r}_i) + q_i\, \boldsymbol{B} \times \boldsymbol{r}_i \tag{C.5}$$

of a charged particle i is, at first glance, a rather unintuitive quantity, as it interconnects position r and momentum p to form a momentum-like quantity. A better understanding can be achieved by investigating the pseudomomentum in a basic problems in classical physics.

A charged particle inside a homogeneous magnetic field $\boldsymbol{B}$ experiences the Lorentz force

$$F = q\, \boldsymbol{v} \times \boldsymbol{B}\,. \tag{C.6}$$

For a magnetic field in z-direction, the corresponding equations of motion form a differential equation of first order in $\boldsymbol{p}$ with the solution

$$\boldsymbol{p}(t) = \begin{pmatrix} \hat{p}_x \cos(\omega_c t) + \hat{p}_y \sin(\omega_c t) \\ -\hat{p}_x \sin(\omega_c t) + \hat{p}_y \cos(\omega_c t) \\ 0 \end{pmatrix}, \tag{C.7}$$

which is integrated into the trajectory

$$r(t) = \frac{1}{m\,\omega_c} \begin{pmatrix} \hat{p}_x \sin(\omega_c t) - \hat{p}_y \cos(\omega_c t) \\ \hat{p}_x \cos(\omega_c t) + \hat{p}_y \sin(\omega_c t) \\ 0 \end{pmatrix} + \begin{pmatrix} \hat{r}_x \\ \hat{r}_y \\ 0 \end{pmatrix} + \frac{1}{m\,\omega_c} \begin{pmatrix} \hat{p}_y \\ -\hat{p}_x \\ 0 \end{pmatrix} \tag{C.8}$$

with the cyclotron frequency $\omega_c = q\,B/m$ and the initial conditions $\boldsymbol{r}(0) = (\hat{r}_x, \hat{r}_y, 0)^T$, $\boldsymbol{p}(0) = (\hat{p}_x, \hat{p}_y, 0)^T$.

It is apparent from Eq. (C.7) that the momentum vector $\boldsymbol{p}$ is not a conserved quantity for the motion inside a magnetic field. Although the magnitude $|\boldsymbol{p}| = \sqrt{\hat{p}_x^2 + \hat{p}_y^2}$ is constant, the direction of $\boldsymbol{p}$ rotates with the cyclotron frequency of the particle. Now, in calculating the pseudomomentum from Eq. (C.5), the rotation of $\boldsymbol{p}(t)$ is canceled by the rotating component of $\boldsymbol{r}(t)$, leaving only the constants

$$\boldsymbol{k} = \begin{pmatrix} \hat{p}_x \\ \hat{p}_y \\ 0 \end{pmatrix} + q\,B \begin{pmatrix} -\hat{r}_y \\ \hat{r}_x \\ 0 \end{pmatrix} = \boldsymbol{p}(0) + q\,\boldsymbol{B} \times \boldsymbol{r}(0)\,. \tag{C.9}$$

Consequently, $\boldsymbol{k}$ is a constant of motion for a charged particle inside a magnetic field. The solution of Eq. (C.9) allows for a more intuitive interpretation. If we assume $\boldsymbol{r}(0) = 0$, then the pseudomomentum $\boldsymbol{k} = \boldsymbol{p}(0)$ is equal to the initial momentum with which the particle has entered the magnetic field. However, the pseudomomentum additionally takes into account the initial position of the particle with $\boldsymbol{B} \times \boldsymbol{r}(0)$, making the pseudomomentum more than the initial momentum.

C.3. Field Scalings for the Existence of a Giant Dipole Well

The conditions under which a giant dipole potential well is formed can be derived from the simplified effective potential in Eq. (4.40) for a homogeneous magnetic field $\boldsymbol{B}$ in z-direction and electric field $\boldsymbol{E}$ in x-direction. To recall, the potential reads

$$\mathcal{V}_{\text{eff}} = \frac{e^2 B^2}{2M}\left(x^2 + y^2\right) + e\,E\,x - \frac{e^2}{4\pi\varepsilon_0\varepsilon_r}\frac{1}{\sqrt{x^2+y^2+z^2}}\,. \tag{C.10}$$

A extreme point along the x-axis exists where the condition

$$\frac{\partial}{\partial x}\mathcal{V}_{\text{eff}} = \frac{e^2 B^2}{M}x + e\,E + \frac{e^2}{4\pi\varepsilon_0\varepsilon_r}\frac{x}{\sqrt{x^2}^3} \overset{!}{=} 0 \tag{C.11}$$

is fulfilled. The solution of Eq. (C.11) is that of a polynomial of third order, which can be expressed analytically but is quite lengthy and not intuitive to understand. Instead, it is more helpful to recall the physical assumptions made in Sec. 4.2. In the absence of an electric field, the magnetic term in Eq. (C.10) imposes a parabolic potential over the Coulomb well. With increasing field strength of E, this parabola is shifted towards the negative x-direction until a fringe field strength E_s is reached, where the curvatures of magnetic field, electric field, and Coulomb potential cancel out at a point x_0 and a saddle point appears, as in Fig. 4.6(a). By further increasing E, the saddle point turns into a local minimum and shifts outwards to ever greater distances in negative x-direction. Consequently, the field configuration where a saddle point appears marks the minimal electric field for which the potential Eq. (C.10) can be called a giant dipole potential.

A saddle point is an extreme point, that simultaneously is a turning point. The turning point condition is the vanishing of the second derivative, in our case

$$\frac{\partial^2}{\partial x^2}\mathcal{V}_{\text{eff}} = \frac{e^2 B^2}{M} - \frac{e^2}{4\pi\varepsilon_0\varepsilon_r}\frac{2}{|x|^3} \overset{!}{=} 0\,. \tag{C.12}$$

Recalling that $x < 0$, the solution x_s to Eq. (C.12) is trivial with

$$x_s = -\left(2\frac{M}{4\pi\varepsilon_0\varepsilon_r\,B^2}\right)^{1/3}\,. \tag{C.13}$$

For representing a saddle point, x_s additionally has to fulfill the extreme point condition in Eq. (C.11). Inserting x_s into Eq. (C.11) returns the field relation

$$E_s = \frac{3}{2^{2/3}}\left(\frac{e^3}{4\pi\varepsilon_0\varepsilon_r\,M^2}\right)^{1/3} B^{4/3} \tag{C.14}$$

for the electric field E_s in dependency of the magnetic field B. This field relation is plotted in Fig. C.1 (line) together with the magneto-Stark field in Voigt configuration from Eq. (4.39) (orange), which scales only linearly in B. It is clearly visible that E_s exceed the magneto-Stark field by large for any relevant magnetic fields B. Hence, even in Voigt configuration, the contribution of K through the magneto-Stark effect is safe to neglect and we can assume $\boldsymbol{K} \equiv 0$.

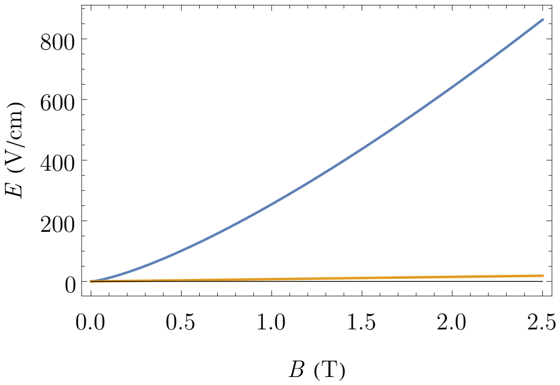


Figure C.1: The fringe electric field strength E_s (blue) for which the effective potential in Eq. (C.10) possesses a saddle point scales with $B^{4/3}$, whereas the effective magneto-Stark field (orange) is linear in B and is considerably smaller. Hence, the magneto-Stark effect does not contribution significantly to the formation of a giant dipole potential.

C.4. Magnetoexciton Diagonalization in the Hydrogen Basis

The name magnetoexcitons refers to exciton states in the presence of a strong homogeneous magnetic field $\boldsymbol{B}$. When reduced to a hydrogen model, they are described by the Hamiltonian in Eq. (4.25). Without loss of generality, we can assume the magnetic field $\boldsymbol{B} = B\,\boldsymbol{e}_z$ to be aligned in z-direction. Further, we set $\boldsymbol{E} = 0$ in the absence of an electric field. Following the discussion in Appendix. C.3 and observations in Ref. [191], we further neglect the magneto-Stark effect and set $\boldsymbol{K} = 0$. Further, with the dynamics inside the xy-plane being symmetric around the center of the Coulomb potential we can fix the symmetric gauge and set $f(\boldsymbol{r}) \equiv 0$ for the internal gauge function. With these assumptions, the Hamiltonian in Eq. (4.25) from Sec. 4.2 reads

$$\mathcal{H}_{\text{eff}} = \underbrace{\frac{p^2}{2\mu} - \frac{e^2}{4\pi\varepsilon_0\varepsilon_r}\frac{1}{\sqrt{x^2+y^2+z^2}}}_{\mathcal{H}_0} + \underbrace{\frac{e\,B}{2\tilde{\mu}}L_z}_{\mathcal{H}_{\text{Z}}} + \underbrace{\frac{e^2B^2}{8\mu}\left(x^2+y^2\right)}_{\mathcal{H}_{\text{dia}}}\,, \tag{C.15}$$

where the quadratic terms of both the kinetic energy and potentials are joined back together and introducing the angular momentum operator $L_z = \boldsymbol{e}_z\cdot(\boldsymbol{p}\times\boldsymbol{r}+\boldsymbol{r}\times\boldsymbol{p})/2$ is introduced [181]. As highlighted in Eq. (C.15) the Hamiltonian can be subdivided into three parts: the hydrogen Hamiltonian $\mathcal{H}_0$, the Zeeman term $\mathcal{H}_{\text{Z}}$ and the diamagnetic term $\mathcal{H}_{\text{dia}}$.

The hydrogenic Hamiltonian $\mathcal{H}_0$ possesses the well known orthonormal basis of eigenfunctions

$$|nlm\rangle = \psi_{nlm}(\boldsymbol{r}) = Y_l^m(\theta,\phi)\,R_{nl}(r) \tag{C.16}$$

in spherical coordinates (r,θ,ϕ) with the spherical harmonics $Y_l^m(\theta,\phi)$ and the radial wave functions

$$R_{nl}(r) = \sqrt{\left(\frac{2}{na_0}\right)^3\frac{(n-l-1)!}{2n\,(n+l)!}}\,\mathrm{e}^{-\rho/2}\rho^l\,L_{n-l-1}^{2l+1}(\rho)\,, \tag{C.17}$$

with $\rho = (2r)/(n\,a_0)$ and the associated Laguerre polynomials $L_n^k(x)$ [181]. The Zeeman shares the same eigenbasis while acting only on the radial part with the eigenequation

$$\mathcal{H}_{\text{Z}}\,|nlm\rangle = \frac{e\,B}{2\tilde{\mu}}\,\hbar m\,|nlm\rangle\;. \tag{C.18}$$

The only non-diagonal component of Eq. (C.15) is the diamagnetic term $\mathcal{H}_{\text{dia}}$ which in spherical coordinates reads

$$\mathcal{H}_{\text{dia}} = \frac{e^2B^2}{8\mu}r^2\sin^2(\theta)\,. \tag{C.19}$$

Hence, we require the calculation of matrix elements

$$\langle n'l'm'|\,r^2\sin^2(\theta)\,|nlm\rangle = \langle R_{n'l'}|\,r^2\,|R_{nl}\rangle\;\langle Y_{l'}^{m'}|\sin^2(\theta)\,|Y_l^m\rangle\;, \tag{C.20}$$

that can be separated in purely radial and angular integrals. The solution for the radial matrix elements of arbitrary ranks of r can be found in Ref. [264] for the generalized case of effective quantum numbers $n^* = n - \delta_l$ affected by quantum defects δ_l. Reduction to the discrete case and adapting the normalization gives the analytic expression

$$\begin{aligned}\langle R_{n_1 l_1}| \, r^d \, |R_{n_2 l_2}\rangle = & \frac{2^{l_1+l_2+2}}{n_1^{l1+2}\, n_2^{l2+2}} \sqrt{\frac{(n_1-l_1-1)!\,(n_2-l_2-1)!}{(n_1+l_1)!\,(n_2+l_2)!}} \\ & \times \sum_{m_1=0}^{n_1-l_1-1} \sum_{m_2=0}^{n_2-l_2-1} (-2)^{m_1+m_2} \frac{(l_1+l_2+m_1+m_2+d+2)}{m_1!\,m_2!\,n_1^{m_1}\,n_2^{m_2}} \left(\frac{n_1\,n_2}{n_1+n_2}\right)^{l_1+l_2+m_1+m_2+d+3} \\ & \times \begin{pmatrix} n_1+l_1 \\ n_1-l_1-m_1-1 \end{pmatrix} \begin{pmatrix} n_2+l_2 \\ n_2-l_2-m_2-1 \end{pmatrix} . \end{aligned} \tag{C.21}$$

The angular part can be solved by exploiting the integral relation between three spherical harmonics

$$\begin{aligned}\int & Y_{l_1}^{m_1}(\theta,\phi)\, Y_{l_2}^{m_2}(\theta,\phi)\, Y_{l_3}^{m_3}(\theta,\phi) \mathrm{d}\Omega \\ & = \sqrt{\frac{(2l_1+1)(2l_2+1)(2l_3+1)}{4\pi}} \begin{pmatrix} l_1 & l_2 & l_3 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} l_1 & l_2 & l_3 \\ m_1 & m_2 & m_3 \end{pmatrix} \end{aligned} \tag{C.22}$$

and the Wigner-3J symbols [265]. This requires the transformation

$$\sin^2(\theta) = 1 - \cos^2(\theta) = \frac{2}{3} - \frac{1}{3}\sqrt{\frac{16\pi}{5}} Y_2^0(\theta,\phi) \tag{C.23}$$

of the sine into a spherical harmonic function. With further utilizing $(Y_l^m(\theta,\phi))^* = (-1)^m Y_l^{-m}(\theta,\phi)$ the angular matrix elements can be calculated from

$$\begin{aligned}\langle Y_{l_1}^{m_1}| \sin^2(\theta) \, |Y_{l_2}^{m_2}\rangle = & \frac{2}{3}\delta_{l_1,l_2}\delta_{m_1,m_2} \\ & + (-1)^{m_1+1}\frac{2}{3}\sqrt{(2l_1+1)(2l_2+1)} \begin{pmatrix} l_1 & l_2 & 2 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} l_1 & l_2 & 2 \\ -m_1 & m_2 & 0 \end{pmatrix} . \end{aligned} \tag{C.24}$$

The angular matrix elements are non-zero only if either both $l_1 = l_2$ and $m_1 = m_2$ are fulfilled at the same time, so both the Kronecker deltas and the Wigner-3J Symbols are non-zero, or if the condition $|l_1 - l_2| = 2$ is fulfilled.

For diagonalization, the magnetoexciton wave functions Ψ_i are expanded into the basis of hydrogenic wave functions

$$\Psi_i(\boldsymbol{r}) = \sum_{n=1}^{n_{\max}} \sum_{l=0}^{n-1} \sum_{m=-l}^{l} c_{nlm}\, \psi_{nlm}(\boldsymbol{r}) \tag{C.25}$$

up to a terminal quantum number $n_{\max}$. With this representation, the eigenequation

$$\mathcal{H}\,|\Psi_i\rangle = E_i\,|\psi_i\rangle \tag{C.26}$$

is solved by diagonalizing the matrix

$$\mathbf{H} = \{\langle\Psi_i|\,\mathcal{H}\,|\Psi_j\rangle\}_{i,j} \, . \tag{C.27}$$

The eigenvalues of $\mathbf{H}$ are the states eigenenergies E_i, while the corresponding eigenvectors

$\boldsymbol{v}_i$ are composed of the weights c_{nlm} from Eq. (C.25) in the exact ordering chosen in the flattening into a one-dimensional vector. The diagonalization returns exactly $n_{\max}$ eigenstates, which usually can only be sorted energetically. When calculating the spectrum over several values of a variable, like different magnetic field strengths, it can become problematic to trace certain states if intersections of energy levels occur. In this case, it is important to analyze if two levels cross each other or if an avoided crossing appears. This can be done by tracking the eigenvectors along the magnetic field. For this, starting from $B = 0$, each weight vector is assigned to the most common weight vector from the next higher B-field diagonalization. In this way, states can be tracked over several true crossings and anti-crossings, just like in Fig. 4.7.

Bibliography

[1] T. Stielow and S. Scheel, *Reconstruction of nanoscale particles from single-shot wide-angle free-electron-laser diffraction patterns with physics-informed neural networks*, Physical Review E **103**, 053312 (2021).

[2] T. Stielow, R. Schmidt, C. Peltz, T. Fennel, and S. Scheel, *Fast reconstruction of single-shot wide-angle diffraction images through deep learning*, Machine Learning: Science and Technology **1**, 045007 (2020).

[3] T. Stielow, S. Scheel, and M. Kurz, *Angular-momentum couplings in ultra-long-range giant dipole molecules*, Physical Review A **97**, 022501 (2018).

[4] T. Stielow, S. Scheel, and M. Kurz, *A Green's function approach to giant-dipole systems*, Journal of Physics B: Atomic, Molecular and Optical Physics **51**, 024004 (2017).

[5] S. Ribeiro, S. Y. Buhmann, T. Stielow, and S. Scheel, *Casimir-Polder interaction from exact diagonalization and surface-induced state mixing*, EPL (Europhysics Letters) **110**, 51003 (2015).

[6] Y. LeCun, Y. Bengio, and G. Hinton, *Deep learning*, Nature **521**, 436 (2015).

[7] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning* (MIT Press, 2017) `http://www.deeplearningbook.org`.

[8] F. Chollet, *Deep learning with Python* (Simon and Schuster, 2017).

[9] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, *Attention is all you need*, in Advances in neural information processing systems (2017) pp. 5998–6008.

[10] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, *Mastering the game of Go with deep neural networks and tree search*, nature **529**, 484–489 (2016).

[11] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, Y. Chen, T. Lillicrap, F. Hui, L. Sifre, G. van den Driessche, T. Graepel, and D. Hassabis, *Mastering the game of go without human knowledge*, nature **550**, 354–359 (2017).

[12] W. S. McCulloch and W. Pitts, *A logical calculus of the ideas immanent in nervous activity*, The bulletin of mathematical biophysics **5**, 115–133 (1943).

[13] F. Rosenblatt, *The perceptron: a probabilistic model for information storage and organization in the brain.*, Psychological review **65**, 386 (1958).

[14] B. Widrow and M. E. Hoff, *Adaptive switching circuits*, Tech. Rep. (Stanford Univ Ca Stanford Electronics Labs, 1960).

[15] G. Cybenko, *Approximation by superpositions of a sigmoidal function*, Mathematics of Control, Signals and Systems **2**, 303–314 (1989).

[16] K. Hornik, M. Stinchcombe, and H. White, *Multilayer feedforward networks are universal approximators*, Neural networks **2**, 359–366 (1989).

[17] K. Hornik, *Some new results on neural network approximation*, Neural networks **6**, 1069–1072 (1993).

[18] G. E. Moore, *Cramming more components onto integrated circuits*, Proceedings of the IEEE **86**, 82–85 (1998).

[19] R. Raina, A. Madhavan, and A. Y. Ng, *Large-scale deep unsupervised learning using graphics processors*, in Proceedings of the 26th annual international conference on machine learning (2009) pp. 873–880.

[20] D. C. Cireşan, U. Meier, L. M. Gambardella, and J. Schmidhuber, *Deep, big, simple neural nets for handwritten digit recognition*, Neural computation **22**, 3207–3220 (2010).

[21] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio, *Theano: a CPU and GPU math expression compiler*, in Proceedings of the Python for scientific computing conference (SciPy), Vol. 4 (Austin, TX, 2010) pp. 1–7.

[22] R. Collobert, K. Kavukcuoglu, and C. Farabet, *Torch7: A matlab-like environment for machine learning*, in BigLearn, NIPS workshop, CONF (2011).

[23] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mane, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viegas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, *TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems* (2016), software available from `http://www.tensorflow.org`, arXiv:1603.04467 [cs.DC] .

[24] R. Li, W. Liu, L. Yang, S. Sun, W. Hu, F. Zhang, and W. Li, *DeepUNet: A Deep Fully Convolutional Network for Pixel-level Sea-Land Segmentation* (2017), arXiv:1709.00201 [cs.CV] .

[25] G. M. Foody, F. Ling, D. S. Boyd, X. Li, and J. Wardlaw, *Earth observation and machine learning to meet sustainable development goal 8.7: Mapping sites associated with slavery from space*, Remote Sensing **11**, 266 (2019).

[26] O. Ronneberger, P. Fischer, and T. Brox, *U-net: Convolutional networks for biomedical image segmentation*, in International Conference on Medical image computing and computer-assisted intervention (Springer, 2015) pp. 234–241.

[27] D. Ardila, A. P. Kiraly, S. Bharadwaj, B. Choi, J. J. Reicher, L. Peng, D. Tse, M. Etemadi, W. Ye, G. Corrado, D. P. Naidich, and S. Shetty, *End-to-end lung cancer screening with three-dimensional deep learning on low-dose chest computed tomography*, Nature medicine **25**, 954–961 (2019).

[28] A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, and S. Thrun, *Dermatologist-level classification of skin cancer with deep neural networks*, Nature **542**, 115 (2017).

[29] L. Shen, L. R. Margolies, J. H. Rothstein, E. Fluder, R. McBride, and W. Sieh, *Deep learning to improve breast cancer detection on screening mammography*, Scientific reports **9**, 1–12 (2019).

[30] G. E. Hinton and R. R. Salakhutdinov, *Reducing the dimensionality of data with neural networks*, Science **313**, 504–507 (2006).

[31] J. Michael, R. Labahn, T. Grüning, and J. Zöllner, *Evaluating Sequence-to-Sequence Models for Handwritten Text Recognition* (2019), arXiv:1903.07377 [cs.CV] .

[32] T. Strauß, M. Weidemann, J. Michael, G. Leifert, T. Grüning, and R. Labahn, *System Description of CITlab's Recognition & Retrieval Engine for ICDAR2017 Competition on Information Extraction in Historical Handwritten Records* (2018), arXiv:1804.09943 [cs.IR] .

[33] T. Grüning, G. Leifert, T. Strauß, J. Michael, and R. Labahn, *A two-stage method for text line detection in historical documents*, International Journal on Document Analysis and Recognition (IJDAR) **22**, 285–302 (2019).

[34] J. Zöllner, K. Sperfeld, C. Wick, and R. Labahn, *Optimizing Small BERTs Trained for German NER*, Information **12**, 443 (2021).

[35] T. Karras, S. Laine, and T. Aila, *A Style-Based Generator Architecture for Generative Adversarial Networks* (2019), arXiv:1812.04948 [cs.NE] .

[36] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu, *Semantic image synthesis with spatially-adaptive normalization*, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2019) pp. 2337–2346.

[37] J. Thies, M. Zollhofer, M. Stamminger, C. Theobalt, and M. Nießner, *Face2face: Real-time face capture and reenactment of rgb videos*, in Proceedings of the IEEE conference on computer vision and pattern recognition (2016) pp. 2387–2395.

[38] S. Suwajanakorn, S. M. Seitz, and I. Kemelmacher-Shlizerman, *Synthesizing obama: learning lip sync from audio*, ACM Transactions on Graphics (ToG) **36**, 1–13 (2017).

[39] T.-H. Oh, T. Dekel, C. Kim, I. Mosseri, W. T. Freeman, M. Rubinstein, and W. Matusik, *Speech2face: Learning the face behind a voice*, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2019) pp. 7539–7548.

[40] G. Kasparov, *Chess, a Drosophila of reasoning* (2018).

[41] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, T. Lillicrap, K. Simonyan, and D. Hassabis, *Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm* (2017), arXiv:1712.01815 [cs.AI] .

[42] T. McGrath, A. Kapishnikov, N. Tomašev, A. Pearce, D. Hassabis, B. Kim, U. Paquet, and V. Kramnik, *Acquisition of Chess Knowledge in AlphaZero* (2021), arXiv:2111.09259 [cs.AI] .

[43] C. Berner, G. Brockman, B. Chan, V. Cheung, P. Dębiak, C. Dennison, D. Farhi, Q. Fischer, S. Hashme, C. Hesse, R. Józefowicz, S. Gray, C. Olsson, J. Pachocki, M. Petrov, H. P. d. O. Pinto, J. Raiman, T. Salimans, J. Schlatter, J. Schneider, S. Sidor, I. Sutskever, J. Tang, F. Wolski, and S. Zhang, *Dota 2 with large scale deep reinforcement learning* (2019), arXiv:1912.06680 [cs.LG] .

[44] G. Carleo, I. Cirac, K. Cranmer, L. Daudet, M. Schuld, N. Tishby, L. Vogt-Maranto, and L. Zdeborová, *Machine learning and the physical sciences*, Reviews of Modern Physics **91**, 045002 (2019).

[45] L. Lönnblad, C. Peterson, and T. Rögnvaldsson, *Using neural networks to identify jets*, Nuclear Physics B **349**, 675–702 (1991).

[46] P. Baldi, P. Sadowski, and D. Whiteson, *Searching for exotic particles in high-energy physics with deep learning*, Nature communications **5**, 1–9 (2014).

[47] G. Kasieczka, T. Plehn, M. Russell, and T. Schell, *Deep-learning top taggers or the end of QCD?*, Journal of High Energy Physics **2017**, 6 (2017).

[48] G. Kasieczka, T. Plehn, A. Butter, K. Cranmer, D. Debnath, B. M. Dillon, M. Fairbairn, D. A. Faroughy, W. Fedorko, C. Gay, L. Gouskos, J. F. Kamenik, P. T. Komiske, S. Leiss, A. Lister, S. Macaluso, E. M. Metodiev, L. Moore, B. Nachman, K. Nordstrom, J. Pearkes, H. Qu, Y. Rath, M. Rieger, D. Shih, J. M. Thompson, and S. Varma, *The Machine Learning Landscape of Top Taggers*, SciPost Phys. **7**, 14 (2019).

[49] N. Meinert, *Search for Rare b to Open-Charm Two-Body Decays of Baryons at LHCb*, Ph.D. thesis, Universität Rostock (2020).

[50] A. D. Tranter, H. J. Slatyer, M. R. Hush, A. C. Leung, J. L. Everett, K. V. Paul, P. Vernaz-Gris, P. K. Lam, B. C. Buchler, and G. T. Campbell, *Multiparameter optimisation of a magneto-optical trap using deep learning*, Nature communications **9**, 1–8 (2018).

[51] D. Böckenhoff, M. Blatzheim, H. Hölbe, H. Niemann, F. Pisano, R. Labahn, T. S. Pedersen, and The W7-X Team, *Reconstruction of magnetic configurations in W7-X using artificial neural networks*, Nuclear Fusion **58**, 056009 (2018).

[52] D. Böckenhoff, M. Blatzheim, and The W7-X Team, *Application of improved analysis of convective heat loads on plasma facing components to Wendelstein 7-X*, Nuclear Fusion **59**, 086031 (2019).

[53] M. Blatzheim, D. Böckenhoff, and The W7-X Team, *Neural network regression approaches to reconstruct properties of magnetic configuration from Wendelstein 7-X modeled heat load patterns*, Nuclear Fusion **59**, 126029 (2019).

[54] T. Thum, *Reinforcement Learning for Wendelstein 7-X Divertor heat load control*, Master's thesis, Rheinisch-Westfälische Technische Hochschule Aachen (2021).

[55] D. Böckenhoff, T. Thum, and The W7-X Team, (2022), to be published.

[56] G. Carleo and M. Troyer, *Solving the quantum many-body problem with artificial neural networks*, Science **355**, 602–606 (2017).

[57] D.-L. Deng, X. Li, and S. D. Sarma, *Quantum entanglement in neural network states*, Physical Review X **7**, 021021 (2017).

[58] G. Torlai, B. Timar, E. P. van Nieuwenburg, H. Levine, A. Omran, A. Keesling, H. Bernien, M. Greiner, V. Vuletić, M. D. Lukin, R. G. Melko, and M. Endres, *Integrating neural networks with a quantum simulator for state reconstruction*, Physical review letters **123**, 230504 (2019).

[59] G. Torlai, G. Mazzola, G. Carleo, and A. Mezzacapo, *Precise measurement of quantum observables with neural-network estimators*, Physical Review Research **2**, 022060 (2020).

[60] M. Neugebauer, L. Fischer, A. Jäger, S. Czischek, S. Jochim, M. Weidemüller, and M. Gärttner, *Neural-network quantum state tomography in a two-qubit experiment*, Physical Review A **102**, 042604 (2020).

[61] G. Torlai, G. Mazzola, J. Carrasquilla, M. Troyer, R. Melko, and G. Carleo, *Neural-network quantum state tomography*, Nature Physics **14**, 447 (2018).

[62] T. Xin, S. Lu, N. Cao, G. Anikeeva, D. Lu, J. Li, G. Long, and B. Zeng, *Local-measurement-based quantum state tomography via neural networks*, npj Quantum Information **5**, 1–8 (2019).

[63] J. Gao, L.-F. Qiao, Z.-Q. Jiao, Y.-C. Ma, C.-Q. Hu, R.-J. Ren, A.-L. Yang, H. Tang, M.-H. Yung, and X.-M. Jin, *Experimental machine learning of quantum states*, Physical review letters **120**, 240501 (2018).

[64] J. Carrasquilla, G. Torlai, R. G. Melko, and L. Aolita, *Reconstructing quantum states with generative models*, Nature Machine Intelligence **1**, 155–161 (2019).

[65] T. Schmale, M. Reh, and M. Ǵarttner, *Scalable quantum state tomography with artificial neural networks* (2021), arXiv:2109.13776 [quant-ph] .

[66] S. Lu, S. Huang, K. Li, J. Li, J. Chen, D. Lu, Z. Ji, Y. Shen, D. Zhou, and B. Zeng, *Separability-entanglement classifier via machine learning*, Physical Review A **98**, 012315 (2018).

[67] Y. Levine, O. Sharir, N. Cohen, and A. Shashua, *Quantum entanglement in deep learning architectures*, Physical review letters **122**, 065301 (2019).

[68] N. Laanait, Q. He, and A. Y. Borisevich, *Reconstruction of 3-D Atomic Distortions from Electron Microscopy with Deep Learning* (2019), arXiv:1902.06876 [cond-mat.mtrl-sci] .

[69] N. Laanait, J. Romero, J. Yin, M. T. Young, S. Treichler, V. Starchenko, A. Borisevich, A. Sergeev, and M. Matheson, *Exascale Deep Learning for Scientific Inverse Problems* (2019), arXiv:1909.11150 [cs.LG] .

[70] C. Chen, Y. Zuo, and W. Ye, *Learning properties of ordered and disordered materials from multi-fidelity data*, Nat Comput Sci **1**, 46–53 (2021).

[71] S. Liu, C. N. Melton, S. Venkatakrishnan, R. J. Pandolfi, G. Freychet, D. Kumar, H. Tang, A. Hexemer, and D. M. Ushizima, *Convolutional neural networks for grazing incidence x-ray scattering patterns: thin film structure identification*, MRS Communications **9**, 586–592 (2019).

[72] W. Van Herck, J. Fisher, and M. Ganeva, *Deep learning for x-ray or neutron scattering under grazing-incidence: extraction of distributions*, Materials Research Express **8**, 045015 (2021).

[73] L. Wang, *Discovering phase transitions with unsupervised learning*, Physical Review B **94**, 195105 (2016).

[74] J. Carrasquilla and R. G. Melko, *Machine learning phases of matter*, Nature Physics **13**, 431 (2017).

[75] E. P. Van Nieuwenburg, Y.-H. Liu, and S. D. Huber, *Learning phase transitions by confusion*, Nature Physics **13**, 435 (2017).

[76] J. Venderley, V. Khemani, and E.-A. Kim, *Machine learning out-of-equilibrium phases of matter*, Physical Review Letters **120**, 257204 (2018).

[77] M. Raissi, P. Perdikaris, and G. E. Karniadakis, *Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations*, Journal of Computational Physics **378**, 686–707 (2019).

[78] M. Raissi, A. Yazdani, and G. E. Karniadakis, *Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations*, Science **367**, 1026–1030 (2020).

[79] D. Ushizima, C. Yang, S. Venkatakrishnan, F. Araujo, R. Silva, H. Tang, J. V. Mascarenhas, A. Hexemer, D. Parkinson, and J. Sethian, *Convolutional neural networks at the interface of physical and digital data*, in 2016 IEEE Applied Imagery Pattern Recognition Workshop (AIPR) (IEEE, 2016) pp. 1–12.

[80] C. L. Chen, A. Mahjoubfar, L.-C. Tai, I. K. Blaby, A. Huang, K. R. Niazi, and B. Jalali, *Deep learning in label-free cell classification*, Scientific reports **6**, 1–16 (2016).

[81] W. Ouyang, A. Aristov, M. Lelek, X. Hao, and C. Zimmer, *Deep learning massively accelerates super-resolution localization microscopy*, Nature biotechnology **36**, 460–468 (2018).

[82] A. Durand, T. Wiesner, M.-A. Gardner, L.-É. Robitaille, A. Bilodeau, C. Gagné, P. De Koninck, and F. Lavoie-Cardinal, *A machine learning approach for online automated optimization of super-resolution optical microscopy*, Nature communications **9**, 1–16 (2018).

[83] Y. Rivenson, Y. Wu, and A. Ozcan, *Deep learning in holography and coherent imaging*, Light: Science & Applications **8**, 1–8 (2019).

[84] B. Langbehn, K. Sander, Y. Ovcharenko, C. Peltz, A. Clark, M. Coreno, R. Cucini, M. Drabbels, P. Finetti, M. Di Fraia, L. Giannessi, C. Grazioli, D. Iablonskyi, A. C. LaForge, T. Nishiyama, V. Oliver Álvarez de Lara, P. Piseri, O. Plekan, K. Ueda, J. Zimmermann, K. C. Prince, F. Stienkemeier, C. Callegari, T. Fennel, D. Rupp, and T. Möller, *Three-dimensional shapes of spinning helium nanodroplets*, Physical review letters **121**, 255301 (2018).

[85] J. Zimmermann, B. Langbehn, R. Cucini, M. Di Fraia, P. Finetti, A. C. LaForge, T. Nishiyama, Y. Ovcharenko, P. Piseri, O. Plekan, , K. C. Prince, F. Stienkemeier, K. Ueda, C. Callegari, T. Möller, and D. Rupp, *Deep neural networks for classifying complex features in diffraction images*, Physical Review E **99**, 063309 (2019).

[86] M. J. Cherukara, Y. S. Nashed, and R. J. Harder, *Real-time coherent diffraction inversion using deep generative networks*, Scientific reports **8**, 1–8 (2018).

[87] H. Chan, Y. S. Nashed, S. Kandel, S. O. Hruszkewycz, S. K. Sankaranarayanan, R. J. Harder, and M. J. Cherukara, *Rapid 3D nanoscale coherent imaging via physics-aware deep learning*, Applied Physics Reviews **8**, 021407 (2021).

[88] D. Ratner, F. Christie, J. Cryan, A. Edelen, A. Lutman, and X. Zhang, *Recovering the phase and amplitude of X-ray FEL pulses using neural networks and differentiable models*, Optics express **29**, 20336–20352 (2021).

[89] Y. Zhang, M. A. Noack, P. Vagovic, K. Fezzaa, F. Garcia-Moreno, T. Ritschel, and P. Villanueva-Perez, *PhaseGAN: a deep-learning phase-retrieval approach for unpaired datasets*, Optics Express **29**, 19593–19604 (2021).

[90] Y. Yao, H. Chan, S. Sankaranarayanan, P. Balaprakash, R. J. Harder, and M. J. Cherukara, *AutoPhaseNN: Unsupervised Physics-aware Deep Learning of 3D Nanoscale Coherent Imaging* (2021), arXiv:2109.14053 [physics.app-ph] .

[91] S. Y. Lee, D. H. Cho, C. Jung, D. Sung, D. Nam, S. Kim, and C. Song, *Denoising low-intensity diffraction signals using k-space deep learning: Applications to phase recovery*, Physical Review Research **3**, 043066 (2021).

[92] Y. Zhuang, S. Awel, A. Barty, R. Bean, J. Bielecki, M. Bergemann, B. J. Daurer, T. Ekeberg, A. D. Estillore, H. Fangohr, K. Giewekemeyer, M. S. Hunter, M. Karnevskiy, R. A. Kirian, H. Kirkwood, Y. Kim, J. Koliyadu, H. Lange, R. Letrun, J. Lübke, A. Mall, T. Michelat, A. J. Morgan, N. Roth, A. K. Samanta, T. Sato, Z. Shen, M. Sikorski, F. Schulz, J. C. H. Spence, P. Vagovic, T. Wollweber, L. Worbs, P. L. Xavier, O. Yefanov, F. R. N. C. Maia, D. A. Horke, J. Küpper, N. D. Loh, A. P. Mancuso, H. N. Chapman, and K. Ayyer, *Unsupervised learning approaches to characterize heterogeneous samples using X-ray single particle imaging* (2021), arXiv:2109.06179 [eess.IV] .

[93] I. Barke, H. Hartmut, D. Rupp, L. Flückiger, M. Sauppe, M. Adolph, S. Schorb, C. Bostedt, R. Treusch, C. Peltz, S. Bartling, T. Fennel, K.-H. Meiwes-Broes, and T. Möller, *The 3D-architecture of individual free silver nanoparticles captured by X-ray scattering*, Nature communications **6**, 6187 (2015).

[94] P. L. Bartlett, P. M. Long, G. Lugosi, and A. Tsigler, *Benign overfitting in linear regression*, Proceedings of the National Academy of Sciences **117**, 30063–30070 (2020).

[95] Z. Li, W. Su, and D. Sejdinovic, *Benign Overfitting and Noisy Features* (2021), arXiv:2008.02901 [stat.ML] .

[96] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, *Learning representations by back-propagating errors*, nature **323**, 533–536 (1986).

[97] Y. LeCun, *Modèles connexionistes de l'apprentissage*, Ph.D. thesis, Université de Paris VI (1987).

[98] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, *Greedy layer-wise training of deep networks*, in Advances in neural information processing systems (2007) pp. 153–160.

[99] M. Ranzato, C. Poultney, S. Chopra, and Y. LeCun, *Efficient learning of sparse representations with an energy-based model*, Advances in neural information processing systems **19**, 1137 (2007).

[100] A. A. Minai and R. D. Williams, *On the derivatives of the sigmoid*, Neural Networks **6**, 845–853 (1993).

[101] V. Nair and G. E. Hinton, *Rectified linear units improve restricted boltzmann machines*, in ICML (2010) pp. 807–814.

[102] X. Glorot, A. Bordes, and Y. Bengio, *Deep sparse rectifier neural networks*, in Proceedings of the fourteenth international conference on artificial intelligence and statistics, Proceedings of Machine Learning Research, Vol. 15 (JMLR Workshop and Conference Proceedings, 2011) pp. 315–323.

[103] K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun, *What is the best multi-stage architecture for object recognition?*, in 2009 IEEE 12th international conference on computer vision (IEEE, 2009) pp. 2146–2153.

[104] M. Crossley, K. Staras, and G. Kemenes, *A two-neuron system for adaptive goal-directed decision-making in Lymnaea*, Nature communications **7**, 1–13 (2016).

[105] B. T. Polyak and A. B. Juditsky, *Acceleration of stochastic approximation by averaging*, SIAM journal on control and optimization **30**, 838–855 (1992).

[106] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, *Gradient-based learning applied to document recognition*, Proceedings of the IEEE **86**, 2278–2324 (1998).

[107] R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F. A. Wichmann, and W. Brendel, *ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness* (2019), arXiv:1811.12231 [cs.CV] .

[108] K. He, X. Zhang, S. Ren, and J. Sun, *Deep residual learning for image recognition*, in Proceedings of the IEEE conference on computer vision and pattern recognition (2016) pp. 770–778.

[109] K. He, X. Zhang, S. Ren, and J. Sun, *Identity mappings in deep residual networks*, in European conference on computer vision (Springer, 2016) pp. 630–645.

[110] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, *Aggregated residual transformations for deep neural networks*, in Proceedings of the IEEE conference on computer vision and pattern recognition (2017) pp. 1492–1500.

[111] S. Jégou, M. Drozdzal, D. Vazquez, A. Romero, and Y. Bengio, *The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation*, in Proceedings of the IEEE conference on computer vision and pattern recognition workshops (2017) pp. 11–19.

[112] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. (MIT press, 2018).

[113] OpenAI, *Introduction to RL*, `https://spinningup.openai.com/en/latest/spinningup/rl_intro.html`.

[114] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, *Continuous control with deep reinforcement learning* (2015), arXiv:1509.02971 [cs.LG] .

[115] J. Peters and S. Schaal, *Reinforcement learning of motor skills with policy gradients*, Neural networks **21**, 682–697 (2008).

[116] Keras-team, *ddpg_pendulum*, `https://github.com/keras-team/keras-io/blob/master/examples/rl/ddpg_pendulum.py` (2020).

[117] Y. Duan, X. Chen, R. Houthooft, J. Schulman, and P. Abbeel, *Benchmarking deep reinforcement learning for continuous control*, in International conference on machine learning (PMLR, 2016) pp. 1329–1338.

[118] R. Liessner, J. Schmitt, A. Dietermann, and B. Bäker, *Hyperparameter Optimization for Deep Reinforcement Learning in Vehicle Energy Management.*, in ICAART (2) (2019) pp. 134–144.

[119] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, *Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor* (2018), arXiv:1801.01290 [cs.LG] .

[120] E. Abbe, *Beiträge zur Theorie des Mikroskops und der mikroskopischen Wahrnehmung*, Archiv für mikroskopische Anatomie **9**, 413–468 (1873).

[121] M. J. Adams, T. L. Blundell, E. J. Dodson, G. G. Dodson, M. Vijayan, E. N. Baker, M. M. Harding, D. C. Hodgkin, B. Rimmer, and S. Sheat, *Structure of rhombohedral 2 zinc insulin crystals*, Nature **224**, 491–495 (1969).

[122] Royal Swedish Academy of Sciences, *Award Ceremony Speech* (1964).

[123] J. D. Watson and F. H. Crick, *Molecular structure of nucleic acids: a structure for deoxyribose nucleic acid*, Nature **171**, 737–738 (1953).

[124] Royal Swedish Academy of Sciences, *James Watson – Biographical* (1962).

[125] B. Maddox, *The double helix and the 'wronged heroine'*, Nature **421**, 407–408 (2003).

[126] B. Turoňová, M. Sikora, C. Schürmann, W. J. Hagen, S. Welsch, F. E. Blanc, S. von Bülow, M. Gecht, K. Bagola, C. Hörner, G. van Zandbergen, J. Landry, N. T. D. de Azevedo, S. Mosalaganti, A. Schwarz, R. Covino, M. D. Mühlebach, G. Hummer, J. Krijnse Locker, and M. Beck, *In situ structural analysis of SARS-CoV-2 spike reveals flexibility mediated by three hinges*, Science **370**, 203–208 (2020).

[127] H. Yao, Y. Song, Y. Chen, N. Wu, J. Xu, C. Sun, J. Zhang, T. Weng, Z. Zhang, Z. Wu, L. Cheng, D. Shi, X. Lu, J. Lei, M. Crispin, Y. Shi, L. Li, and S. Li, *Molecular architecture of the SARS-CoV-2 virus*, Cell **183**, 730–738 (2020).

[128] M. Hoffmann, H. Kleine-Weber, S. Schroeder, N. Krüger, T. Herrler, S. Erichsen, T. S. Schiergens, G. Herrler, N.-H. Wu, A. Nitsche, M. A. Müller, C. Drosten, and S. Pöhlmann, *SARS-CoV-2 cell entry depends on ACE2 and TMPRSS2 and is blocked by a clinically proven protease inhibitor*, cell **181**, 271–280 (2020).

[129] S. Boutet, L. Lomb, G. J. Williams, T. R. Barends, A. Aquila, R. B. Doak, U. Weierstall, D. P. DePonte, J. Steinbrener, R. L. Shoeman, M. Messerschmidt, A. Barty, T. A. White, S. Kassemeyer, R. A. Kirian, M. M. Seibert, P. A. Montanez, C. Kenney, R. Herbst, P. Hart, J. Pines, G. Haller, S. M. Gruner, H. T. Philipp, M. W. Tate, M. Hromalik, L. J. Koerner, N. van Bakel, J. Morse, W. Ghonsalves, D. Arnlund, M. J. Bogan, C. Caleman, R. Fromme, C. Y. Hampton, M. S. Hunter, L. C. Johansson, G. Katona, C. Kupitz, M. Liang, A. V. Martin, K. Nass, L. Redecke, F. Stellato, N. Timneanu, D. Wang, N. A. Zatsepin, D. Schafer, J. Defever, R. Neutze, P. Fromme, J. C. H. Spence, H. N. Chapman, and I. Schlichting, *High-resolution protein structure determination by serial femtosecond crystallography*, Science **337**, 362–364 (2012).

[130] H. N. Chapman, P. Fromme, A. Barty, T. A. White, R. A. Kirian, A. Aquila, M. S. Hunter, J. Schulz, D. P. DePonte, U. Weierstall, R. B. Doak, F. R. N. C. Maia, A. V. Martin, I. Schlichting, L. Lomb, N. Coppola, R. L. Shoeman, S. W. Epp, R. Hartmann, D. Rolles, A. Rudenko, L. Foucar, N. Kimmel, G. Weidenspointner, P. Holl, M. Liang, M. Barthelmess, C. Caleman, S. Boutet, M. J. Bogan, J. Krzywinski, C. Bostedt, S. Bajt, L. Gumprecht, B. Rudek, B. Erk, C. Schmidt, A. Hömke, C. Reich, D. Pietschner, L. Strüder, G. Hauser, H. Gorke, J. Ullrich, S. Herrmann, G. Schaller, F. Schopper, H. Soltau, K.-U. Kühnel, M. Messerschmidt, J. D. Bozek, S. P. Hau-Riege, M. Frank, C. Y. Hampton, R. G. Sierra, D. Starodub, G. J. Williams, J. Hajdu, N. Timneanu, M. M. Seibert, J. Andreasson, A. Rocker, O. Jönsson, M. Svenda, S. Stern, K. Nass, R. Andritschke, C.-D. Schröter, F. Krasniqi, M. Bott, K. E. Schmidt, X. Wang, I. Grotjohann, J. M. Holton, T. R. M. Barends, R. Neutze, S. Marchesini, R. Fromme, S. Schorb, D. Rupp, M. Adolph, T. Gorkhover, I. Andersson, H. Hirsemann, G. Potdevin, H. Graafsma, B. Nilsson, and J. C. H. Spence, *Femtosecond X-ray protein nanocrystallography*, Nature **470**, 73–77 (2011).

[131] R. Neutze, R. Wouts, D. Van der Spoel, E. Weckert, and J. Hajdu, *Potential for biomolecular imaging with femtosecond X-ray pulses*, Nature **406**, 752–757 (2000).

[132] Z. Jurek, G. Faigel, and M. Tegze, *Dynamics in a cluster under the influence of intense femtosecond hard X-ray pulses*, The European Physical Journal D-Atomic, Molecular, Optical and Plasma Physics **29**, 217–229 (2004).

[133] H. N. Chapman, A. Barty, M. J. Bogan, S. Boutet, M. Frank, S. P. Hau-Riege, S. Marchesini, B. W. Woods, S. Bajt, W. H. Benner, R. A. London, E. Plönjes, M. Kuhlmann, R. Treusch, S. Düsterer, T. Tschentscher, J. R. Schneider, E. Spiller, T. Möller, C. Bostedt, M. Hoener, D. A. Shapiro, K. O. Hodgson, D. van der Spoel, F. Burmeister, M. Bergh, C. Caleman, G. Huldt, M. M. Seibert, F. R. N. C. Maia, R. W. Lee, A. Szöke, N. Timneanu, and J. Hajdu, *Femtosecond diffractive imaging with a soft-X-ray free-electron laser*, Nature Physics **2**, 839 (2006).

[134] K. Gaffney and H. Chapman, *Imaging atomic structure and dynamics with ultrafast X-ray scattering*, Science **316**, 1444–1448 (2007).

[135] M. M. Seibert, T. Ekeberg, F. R. Maia, M. Svenda, J. Andreasson, O. Jönsson, D. Odić, B. Iwan, A. Rocker, D. Westphal, M. Hantke, D. P. DePonte, A. Barty, J. Schulz, L. Gumprecht, N. Coppola, A. Aquila, M. Liang, T. A. White, A. Martin, C. Caleman, S. Stern, C. Abergel, V. Seltzer, J.-M. Claverie, C. Bostedt, J. D. Bozek, S. Boutet, A. A. Miahnahri, M. Messerschmidt, J. Krzywinski, G. Williams, K. O. Hodgson, M. J. Bogan, C. Y. Hampton, R. G. Sierra, D. Starodub, I. Andersson, S. Bajt, M. Barthelmess, J. C. H. Spence, P. Fromme, U. Weierstall, R. Kirian, M. Hunter, R. B. Doak, S. Marchesini, S. P. Hau-Riege, M. Frank, R. L. Shoeman, L. Lomb, S. W. Epp, R. Hartmann, D. Rolles, A. Rudenko, C. Schmidt, L. Foucar, N. Kimmel, P. Holl, B. Rudek, B. Erk, A. Hömke, C. Reich, D. Pietschner, G. Weidenspointner, L. Strüder, G. Hauser, H. Gorke, J. Ullrich, I. Schlichting, S. Herrmann, G. Schaller, F. Schopper, H. Soltau, K.-U. Kühnel, R. Andritschke, C.-D. Schröter, F. Krasniqi, M. Bott, S. Schorb, D. Rupp, M. Adolph, T. Gorkhover, H. Hirsemann, G. Potdevin, H. Graafsma, B. Nilsson, H. N. Chapman, and J. Hajdu, *Single mimivirus particles intercepted and imaged with an X-ray laser*, Nature **470**, 78 (2011).

[136] T. Ekeberg, M. Svenda, C. Abergel, F. R. Maia, V. Seltzer, J.-M. Claverie, M. Hantke, O. Jönsson, C. Nettelblad, G. Van Der Schot, M. Liang, D. P. DePonte, A. Barty, M. M. Seibert, B. Iwan, I. Andersson, N. D. Loh, A. V. Martin, H. Chapman, C. Bostedt, J. D. Bozek, K. R. Ferguson, J. Krzywinski, S. W. , Epp, D. Rolles, A. Rudenko, R. Hartmann, N. Kimmel, and J. Hajdu, *Three-dimensional reconstruction of the giant mimivirus particle with an x-ray free-electron laser*, Physical review letters **114**, 098102 (2015).

[137] K. Ayyer, A. J. Morgan, A. Aquila, H. DeMirci, B. G. Hogue, R. A. Kirian, P. L. Xavier, C. H. Yoon, H. N. Chapman, and A. Barty, *Low-signal limit of X-ray single particle diffractive imaging*, Optics Express **27**, 37816–37833 (2019).

[138] L. F. Gomez, K. R. Ferguson, J. P. Cryan, C. Bacellar, R. M. P. Tanyag, C. Jones, S. Schorb, D. Anielski, A. Belkacem, C. Bernando, R. Boll, J. Bozek, S. Carron, G. Chen, T. Delmas, L. Englert, S. W. Epp, B. Erk, L. Foucar, R. Hartmann, A. Hexemer, M. Huth, J. Kwok, S. R. Leone, J. H. S. Ma, F. R. N. C. Maia, E. Malmerberg, S. Marchesini, D. M. Neumark, B. Poon, J. Prell, D. Rolles, B. Rudek, A. Rudenko, M. Seifrid, K. R. Siefermann, F. P. Sturm, M. Swiggers, J. Ullrich, F. Weise, P. Zwart, C. Bostedt, O. Gessner, and A. F. Vilesov, *Shapes and vorticities of superfluid helium nanodroplets*, Science **345**, 906–909 (2014).

[139] D. Rupp, N. Monserud, B. Langbehn, M. Sauppe, J. Zimmermann, Y. Ovcharenko, T. Möller, F. Frassetto, L. Poletto, A. Trabattoni, F. Calegari, M. Nisoli, K. Sander, C. Peltz, M. J. Vrakking, T. Fennel, and A. Rouzée, *Coherent diffractive imaging of single helium nanodroplets with a high harmonic generation source*, Nature communications **8**, 493 (2017).

[140] D. Rupp, M. Adolph, T. Gorkhover, S. Schorb, D. Wolter, R. Hartmann, N. Kimmel, C. Reich, T. Feigl, A. De Castro, R. Teusch, L. Strüder, T. Möller, and C. Bostedt, *Identification of twinned gas phase clusters by single-shot scattering with intense soft x-ray pulses*, New Journal of Physics **14**, 055016 (2012).

[141] K. Ayyer, P. L. Xavier, J. Bielecki, Z. Shen, B. J. Daurer, A. K. Samanta, S. Awel, R. Bean, A. Barty, M. Bergemann, T. Ekeberg, A. D. Estillore, H. Fangohr, K. Giewekemeyer, M. S. Hunter, M. Karnevskiy, R. A. Kirian, H. Kirkwood, Y. Kim, J. Koliyadu, H. Lange, R. Letrun, J. Lübke, T. Michelat, A. J. Morgan, N. Roth, T. Sato, M. Sikorski, F. Schulz, J. C. H. Spence, P. Vagovic, T. Wollweber, L. Worbs, O. Yefanov, Y. Zhuang, F. R. N. C. Maia, D. A. Horke, J. Küpper, N. D. Loh, A. P. Mancuso, and H. N. Chapman, *3D diffractive imaging of nanoparticle ensembles using an x-ray laser*, Optica **8**, 15–23 (2021).

[142] K. Sander, *Reconstruction Methods for Single-shot Diffractive Imaging of Free Nanostructures with Ultrashort X-ray and XUV Laser Pulses*, Ph.D. thesis, Universität Rostock (2018).

[143] S. Sinha, E. Sirota, S. Garoff, and H. Stanley, *X-ray and neutron scattering from rough surfaces*, Physical Review B **38**, 2297 (1988).

[144] B. E. Saleh and M. C. Teich, *Fundamentals of photonics* (john Wiley & sons, 2019).

[145] S. Marchesini, H. He, H. N. Chapman, S. P. Hau-Riege, A. Noy, M. R. Howells, U. Weierstall, and J. C. Spence, *X-ray image reconstruction from a diffraction pattern alone*, Physical Review B **68**, 140101 (2003).

[146] K. S. Raines, S. Salha, R. L. Sandberg, H. Jiang, J. A. Rodríguez, B. P. Fahimian, H. C. Kapteyn, J. Du, and J. Miao, *Three-dimensional structure determination from a single view*, Nature **463**, 214 (2010).

[147] K. Engel, *An identity theorem for the Fourier transform of polytopes on rationally parameterisable hypersurfaces* (2020), arXiv:2008.00935 [math.CA] .

[148] M. Brion and M. Vergne, *Lattice points in simple polytopes*, Journal of the American Mathematical Society , 371–392 (1997).

[149] K. Engel and B. Laasch, *The modulus of the Fourier transform on a sphere determines 3-dimensional convex polytopes* (2020), arXiv:2009.10414 [math.MG] .

[150] K. Engel and B. Laasch, *Reconstruction of polytopes from the modulus of the Fourier transform with small wave length* (2020), arXiv:2011.06971 [math-ph] .

[151] G. Wang, H. Yu, W. Cong, and A. Katsevich, *Non-uniqueness and instability of 'ankylography'*, Nature **480**, E2–E3 (2011).

[152] H. Wei, *Fundamental limits of 'ankylography'due to dimensional deficiency*, Nature **480**, E1–E1 (2011).

[153] H. Hartmann, V. Popok, I. Barke, V. von Oeynhausen, and K.-H. Meiwes-Broer, *Design and capabilities of an experimental setup based on magnetron sputtering for formation and deposition of size-selected metal clusters on ultra-clean surfaces*, Review of Scientific Instruments **83**, 073304 (2012).

[154] C. Bostedt, E. Eremina, D. Rupp, M. Adolph, H. Thomas, M. Hoener, A. R. de Castro, J. Tiggesbäumker, K.-H. Meiwes-Broer, T. Laarmann, , H. Wabnitz, E. Plönjes, R. Treusch, J. R. Schneider, and T. Möller, *Ultrafast x-ray scattering of xenon nanoparticles: imaging transient states of matter*, Physical review letters **108**, 093401 (2012).

[155] K. Sander, C. Peltz, C. Varin, S. Scheel, T. Brabec, and T. Fennel, *Influence of wavelength and pulse duration on single-shot x-ray diffraction patterns from nonspherical nanoparticles*, Journal of Physics B: Atomic, Molecular and Optical Physics **48**, 204004 (2015).

[156] B. Kruse, B. Liewehr, C. Peltz, and T. Fennel, *Quantum coherent diffractive imaging*, Journal of Physics: Photonics **2**, 024007 (2020).

[157] D. Zeyl and B. Sattler, *Plato's Timaeus*, in The Stanford Encyclopedia of Philosophy, edited by E. N. Zalta (Metaphysics Research Lab, Stanford University, 2019) summer 2019 ed.

[158] G. Gygax and D. Arneson, *Dungeons & Dragons*, Vol. 19 (Tactical Studies Rules Lake Geneva, WI, 1974).

[159] Wizards RPG Team, *Dungeons & Dragons Core Rulebook*, 5th ed. (Wizards of the Coast, 2014).

[160] Warhammer Studio, *Warhammer 40,000 rulebook*, 9th ed. (Games Workshop Ltd., 2020).

[161] S. Li, P. Roy, A. Travesset, and R. Zandi, *Why large icosahedral viruses need scaffolding proteins*, Proceedings of the National Academy of Sciences **115**, 10971–10976 (2018).

[162] T. Gorkhover, A. Ulmer, K. Ferguson, M. Bucher, F. R. Maia, J. Bielecki, T. Ekeberg, M. F. Hantke, B. J. Daurer, C. Nettelblad, J. Andreasson, A. Barty, P. Bruza, S. Carron, D. Hasse, J. Krzywinski, D. S. D. Larsson, A. Morgan, K. Mühlig, M. Müller, K. Okamoto, A. Pietrini, D. Rupp, M. Sauppe, G. van der Schot, M. Seibert, J. A. Sellberg, M. Svenda, M. Swiggers, N. Timneanu, D. Westphal, G. Williams, H. N. Zani, Alessandro amd Chapman, G. Faigel, T. Möller, J. Hajdu, and C. Bostedt, *Femtosecond X-ray Fourier holography imaging of free-flying nanoparticles*, Nature Photonics **12**, 150 (2018).

[163] M. K. Parvez, *Geometric architecture of viruses*, World journal of virology **9**, 5 (2020).

[164] Z. Andreadakis, A. Kumar, R. G. Román, S. Tollefsen, M. Saville, and S. Mayhew, *The COVID-19 vaccine development landscape* (Oct 5, 2021).

[165] B. Harrach and M. Benkő, *Adenoviruses (Adenoviridae)*, in Encyclopedia of Virology (Fourth Edition), edited by D. H. Bamford and M. Zuckerman (Academic Press, Oxford, 2021) fourth edition ed., pp. 3–16.

[166] M. F. Hantke, D. Hasse, F. R. Maia, T. Ekeberg, K. John, M. Svenda, N. D. Loh, A. V. Martin, N. Timneanu, D. S. Larsson, G. van der Schot, G. H. Carlsson, M. Ingelman, J. Andreasson, D. Westphal, M. Liang, F. Stellato, D. P. DePonte, R. Hartmann, N. Kimmel, R. A. Kirian, M. M. Seibert, K. Mühlig, S. Schorb, K. Ferguson, C. Bostedt, S. Carron, J. D. Bozek, D. Rolles, A. Rudenko, S. Epp, H. N. Chapman, A. Barty, J. Hajdu, and I. Andersson, *High-throughput imaging of heterogeneous cell organelles with an X-ray laser*, Nature Photonics **8**, 943–949 (2014).

[167] W. Miehle, O. Kandler, T. Leisner, and O. Echt, *Mass spectrometric evidence for icosahedral structure in large rare gas clusters: Ar, Kr, Xe*, The Journal of chemical physics **91**, 5940–5952 (1989).

[168] L. Marks, *Experimental studies of small particle structures*, Reports on Progress in Physics **57**, 603 (1994).

[169] H. Hofmeister, *Fivefold twinned nanoparticles*, Encyclopedia of nanoscience and nanotechnology **3**, 431 (2004).

[170] O. Rodrigues, *Des lois géométriques qui régissent les déplacements d'un système solide dans l'espace, et de la variation des coordonnées provenant de ces déplacements considérés indépendamment des causes qui peuvent les produire*, Journal de mathématiques pures et appliquées **5**, 380–440 (1840).

[171] F. C. Frank, *The icosahedral orientation manifold as a model for icosahedral close-packing*, Le Journal de Physique Colloques **47**, C3–165 (1986).

[172] D. P. Kingma and J. Ba, *Adam: A Method for Stochastic Optimization* (2017), arXiv:1412.6980 [cs.LG] .

[173] E. Sobolev, S. Zolotarev, K. Giewekemeyer, J. Bielecki, K. Okamoto, H. K. Reddy, J. Andreasson, K. Ayyer, I. Barak, S. Bari, A. Barty, R. Bean, S. Bobkov, H. N. Chapman, G. Chojnowski, B. J. Daurer, K. Dörner, T. Ekeberg, L. Flückiger, O. Galzitskaya, L. Gelisio, S. Hauf, B. G. Hogue, D. A. Horke, A. Hosseinizadeh, V. Ilyin, C. Jung, C. Kim, Y. Kim, R. A. Kirian, H. Kirkwood, O. Kulyk, J. Küpper, R. Letrun, N. D. Loh, K. Lorenzen, M. Messerschmidt, K. Mühlig, A. Ourmazd, N. Raab, A. V. Rode, M. Rose, A. Round, T. Sato, R. Schubert, P. Schwander, J. A. Sellberg, M. Sikorski, A. Silenzi, C. Song, J. C. H. Spence, S. Stern, J. Sztuk-Dambietz, A. Teslyuk, N. Timneanu, M. Trebbin, C. Uetrecht, B. Weinhausen, G. J. Williams, P. L. Xavier, C. Xu, I. A. Vartanyants, V. S. Lamzin, A. Mancuso, and F. R. N. C. Maia, *Megahertz single-particle imaging at the European XFEL*, Communications Physics **3**, 1–11 (2020).

[174] T. Stielow, *scatter-2D3D-nn*, `https://github.com/thstielow/scatter-2D3D-nn` (2021).

[175] C. Niu, J. Li, and K. Xu, *Im2struct: Recovering 3d shape structure from a single rgb image*, in Proceedings of the IEEE conference on computer vision and pattern recognition (2018) pp. 4521–4529.

[176] C. B. Choy, D. Xu, J. Gwak, K. Chen, and S. Savarese, *3d-r2n2: A unified approach for single and multi-view 3d object reconstruction*, in European conference on computer vision (Springer, 2016) pp. 628–644.

[177] A. Hare and G. Morrison, *Near-field soft X-ray diffraction modelled by the multislice method*, Journal of Modern Optics **41**, 31–48 (1994).

[178] A. Einstein, B. Podolsky, and N. Rosen, *Can quantum-mechanical description of physical reality be considered complete?*, Physical review **47**, 777 (1935).

[179] H. B. Casimir, *On the attraction between two perfectly conducting plates*, in Proc. Kon. Ned. Akad. Wet., Vol. 51 (1948) p. 793.

[180] H. B. Casimir and D. Polder, *The influence of retardation on the London-van der Waals forces*, Physical Review **73**, 360 (1948).

[181] T. F. Gallagher, *Rydberg atoms*, 3rd ed. (Cambridge University Press, 2005).

[182] C. H. Greene, A. Dickinson, and H. Sadeghpour, *Creation of polar and nonpolar ultra-long-range Rydberg molecules*, Physical review letters **85**, 2458 (2000).

[183] T. Pfau, *Neue Beziehungskisten für Riesenatome*, Physik Journal **14**, 47–50 (2015).

[184] L. A. Burkova, I. E. Dzyaloshinskii, S. F. Drukarev, and B. S. Monozon, *Hydrogen-like system in crossed electric and magnetic fields*, Sov. Phys.–JETP **44**, 276 (1976).

[185] D. Baye, N. Clerbaux, and M. Vincke, *Delocalized states of atomic hydrogen in crossed electric and magnetic fields*, Physics Letters A **166**, 135–139 (1992).

[186] P. Schmelcher and L. S. Cederbaum, *Two-body effects of the hydrogen atom in crossed electric and magnetic fields*, Chemical physics letters **208**, 548–554 (1993).

[187] O. Dippel, P. Schmelcher, and L. Cederbaum, *Charged anisotropic harmonic oscillator and the hydrogen atom in crossed fields*, Physical Review A **49**, 4415 (1994).

[188] T. Kazimierczuk, D. Fröhlich, S. Scheel, H. Stolz, and M. Bayer, *Giant Rydberg excitons in the copper oxide Cu_2O*, Nature **514**, 343–347 (2014).

[189] J. Heckötter, M. Freitag, D. Fröhlich, M. Aßmann, M. Bayer, M. Semina, and M. Glazov, *Scaling laws of Rydberg excitons*, Physical Review B **96**, 125142 (2017).

[190] J. Heckötter, M. Freitag, D. Fröhlich, M. Aßmann, M. Bayer, M. Semina, and M. Glazov, *High-resolution study of the yellow excitons in Cu_2O subject to an electric field*, Physical Review B **95**, 035210 (2017).

[191] P. Rommel, F. Schweiner, J. Main, J. Heckötter, M. Freitag, D. Fröhlich, K. Lehninger, M. Aßmann, and M. Bayer, *Magneto-Stark effect of yellow excitons in cuprous oxide*, Physical Review B **98**, 085206 (2018).

[192] M. Kurz, P. Grünwald, and S. Scheel, *Excitonic giant-dipole potentials in cuprous oxide*, Physical Review B **95**, 245205 (2017).

[193] V. Averbukh, N. Moiseyev, P. Schmelcher, and L. S. Cederbaum, *Transition from Rydberg to giant-dipole-moment states of hydrogen atoms in crossed fields: A suggestion for an experiment*, Physical Review A **59**, 3695 (1999).

[194] J. R. Rydberg, *XXXIV. On the structure of the line-spectra of the chemical elements*, The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science **29**, 331–337 (1890).

[195] M. Brune, F. Schmidt-Kaler, A. Maali, J. Dreyer, E. Hagley, J. Raimond, and S. Haroche, *Quantum Rabi oscillation: A direct test of field quantization in a cavity*, Physical review letters **76**, 1800 (1996).

[196] S. Gleyzes, S. Kuhr, C. Guerlin, J. Bernu, S. Deleglise, U. B. Hoff, M. Brune, J.-M. Raimond, and S. Haroche, *Quantum jumps of light recording the birth and death of a photon in a cavity*, Nature **446**, 297–300 (2007).

[197] Royal Swedish Academy of Sciences, *Measuring and Manipulating Individual Quantum Systems* (2012).

[198] J. Schmidt, Y. Münzenmaier, P. Kaspar, P. Schalberger, H. Baur, R. Löw, N. Fruehauf, T. Pfau, and H. Kübler, *An optogalvanic gas sensor based on Rydberg excitations*, Journal of Physics B: Atomic, Molecular and Optical Physics **53**, 094001 (2020).

[199] S. D. Huber and H. P. Büchler, *Dipole-interaction-mediated laser cooling of polar molecules to ultracold temperatures*, Physical review letters **108**, 193006 (2012).

[200] F. Jarisch and M. Zeppenfeld, *State resolved investigation of Förster resonant energy transfer in collisions between polar molecules and Rydberg atoms*, New Journal of Physics **20**, 113044 (2018).

[201] J. B. Balewski, A. T. Krupp, A. Gaj, D. Peter, H. P. Büchler, R. Löw, S. Hofferberth, and T. Pfau, *Coupling a single electron to a Bose–Einstein condensate*, Nature **502**, 664–667 (2013).

[202] T. C. Liebisch, M. Schlagmüller, F. Engel, H. Nguyen, J. Balewski, G. Lochead, F. Böttcher, K. M. Westphal, K. S. Kleinbach, T. Schmid, A. Gaj, R. Löw, S. Hofferberth, T. Pfau, J. Pérez-Ríos, , and C. H. Greene, *Controlling Rydberg atom excitations in dense background gases*, Journal of Physics B: Atomic, Molecular and Optical Physics **49**, 182001 (2016).

[203] C. Füchtbauer and F. Gössler, *Entgegengesetzte Unsymmetrie der Verbreiterung bei verschiedenen Linien einer Serie*, Naturwissenschaften **21**, 315–315 (1933).

[204] C. Füchtbauer and F. Gössler, *Verschiebung und unsymmetrische Verbreiterung von Absorptionslinien durch Fremdgase*, Zeitschrift für Physik **87**, 89–104 (1934).

[205] C. Füchtbauer, P. Schulz, and A. Brandt, *Verschiebung von hohen Serienlinien des Natriums und Kaliums durch Fremdgase, Berechnung der Wirkungsquerschnitte von Edelgasen gegen sehr langsame Elektronen*, Zeitschrift für Physik **90**, 403–415 (1934).

[206] E. Amaldi and E. Segre, *Effetto della pressione sui termini elevati degli alcalini*, Il Nuovo Cimento (1924-1942) **11**, 145–156 (1934).

[207] E. Amaldi and E. Segrè, *Effect of pressure on high terms of alkaline spectra*, Nature **133**, 141–141 (1934).

[208] E. Fermi, *Sopra lo spostamento per pressione delle righe elevate delle serie spettrali*, Il Nuovo Cimento (1924-1942) **11**, 157–166 (1934).

[209] M. R. Milovanović, J. M. Živković, D. B. Ninković, I. M. Stanković, and S. D. Zarić, *How flexible is the water molecule structure? Analysis of crystal structures and the potential energy surface*, Physical Chemistry Chemical Physics **22**, 4138–4143 (2020).

[210] V. Bendkowsky, B. Butscher, J. Nipper, J. P. Shaffer, R. Löw, and T. Pfau, *Observation of ultralong-range Rydberg molecules*, Nature **458**, 1005–1008 (2009).

[211] W. Li, T. Pohl, J. Rost, S. T. Rittenhouse, H. Sadeghpour, J. Nipper, B. Butscher, J. Balewski, V. Bendkowsky, R. Löw, and T. Pfau, *A homonuclear molecule with a permanent electric dipole moment*, Science **334**, 1110–1114 (2011).

[212] K. S. Kleinbach, F. Meinert, F. Engel, W. J. Kwon, R. Löw, T. Pfau, and G. Raithel, *Photoassociation of trilobite Rydberg molecules via resonant spin-orbit coupling*, Physical review letters **118**, 223001 (2017).

[213] K.-K. Ni, S. Ospelkaus, M. De Miranda, A. Pe'Er, B. Neyenhuis, J. Zirbel, S. Kotochigova, P. Julienne, D. Jin, and J. Ye, *A high phase-space-density gas of polar molecules*, science **322**, 231–235 (2008).

[214] J. Deiglmayr, A. Grochola, M. Repp, K. Mörtlbauer, C. Glück, J. Lange, O. Dulieu, R. Wester, and M. Weidemüller, *Formation of ultracold polar molecules in the rovibrational ground state*, Physical review letters **101**, 133004 (2008).

[215] S. Ospelkaus, K.-K. Ni, D. Wang, M. De Miranda, B. Neyenhuis, . G. Quéméner, P. Julienne, J. Bohn, D. Jin, and J. Ye, *Quantum-state controlled chemical reactions of ultracold potassium-rubidium molecules*, Science **327**, 853–857 (2010).

[216] L. D. Carr, D. DeMille, R. V. Krems, and J. Ye, *Cold and ultracold molecules: science, technology and applications*, New Journal of Physics **11**, 055049 (2009).

[217] S. T. Rittenhouse and H. Sadeghpour, *Ultracold giant polyatomic Rydberg molecules: coherent control of molecular orientation*, Physical review letters **104**, 243002 (2010).

[218] B. Zhao, A. Glaetzle, G. Pupillo, and P. Zoller, *Atomic Rydberg reservoirs for polar molecules*, Physical review letters **108**, 193007 (2012).

[219] J. Gay, L. Pendrill, and B. Cagnac, *New resonances in the atomic spectra in crossed electric and magnetic fields*, Physics Letters A **72**, 315–318 (1979).

[220] C. W. Clark, E. Korevaar, and M. G. Littman, *Quasi-Penning resonances of a Rydberg electron in crossed electric and magnetic fields*, Physical review letters **54**, 320 (1985).

[221] H. Cartarius, J. Main, and G. Wunner, *Signatures of the classical transition state in atomic quantum spectra*, Physical Review A **79**, 033412 (2009).

[222] F. Schweiner, J. Main, H. Cartarius, and G. Wunner, *Classical dynamics and localization of resonances in the high-energy region of the hydrogen atom in crossed fields*, Physical Review E **91**, 012915 (2015).

[223] M. Fauth, H. Walther, and E. Werner, *Rydberg atoms in crossed external magnetic and electric fields: Experimental observation of an outer potential minimum*, Zeitschrift für Physik D Atoms, Molecules and Clusters **7**, 293–296 (1987).

[224] G. Wiebusch, J. Main, K. Krüger, H. Rottke, A. Holle, and K. Welge, *Hydrogen atom in crossed magnetic and electric fields*, Physical review letters **62**, 2821 (1989).

[225] G. Raithel, M. Fauth, and H. Walther, *Atoms in strong crossed electric and magnetic fields: Evidence for states with large electric-dipole moments*, Physical Review A **47**, 419 (1993).

[226] I. Dzyaloshinskii, *Effects of the finite proton mass in a hydrogen atom in crossed magnetic and electric fields: a state with a giant electric dipole moment*, Physics Letters A **165**, 69–71 (1992).

[227] J. Avron, I. Herbst, and B. Simon, *Separation of center of mass in homogeneous magnetic fields*, Annals of Physics **114**, 431–451 (1978).

[228] M. Kurz, M. Mayle, and P. Schmelcher, *Ultra-long-range giant dipole molecules in crossed electric and magnetic fields*, EPL (Europhysics Letters) **97**, 43001 (2012).

[229] M. Kurz and P. Schmelcher, *Ultralong-range Rydberg molecules in combined electric and magnetic fields*, Journal of Physics B: Atomic, Molecular and Optical Physics **47**, 165101 (2014).

[230] S. Nam, D. Reynolds, C. Litton, R. Almassy, T. Collins, and C. Wolfe, *Free-exciton energy spectrum in GaAs*, Physical Review B **13**, 761 (1976).

[231] V. Levitskiy, *A New Find of Copper and Cuprite Crystals in Russia*, Rocks & Minerals **84**, 324–325 (2009).

[232] I. V. Pekov and I. S. Lykova, *Rubtsovskoe Deposit*, Mineralogical Almanac **16** (2011).

[233] E. Gross, *Optical spectrum of excitons in the crystal lattice*, Il Nuovo Cimento (1955-1965) **3**, 672–701 (1956).

[234] S. A. Lynch, C. Hodges, S. Mandal, W. Langbein, R. P. Singh, L. A. Gallagher, J. D. Pritchett, D. Pizzey, J. P. Rogers, C. S. Adams, and M. P. A. Jones, *Rydberg excitons in synthetic cuprous oxide Cu_2O*, Physical Review Materials **5**, 084602 (2021).

[235] S. Steinhauer, M. A. Versteegh, S. Gyger, A. W. Elshaari, B. Kunert, A. Mysyrowicz, and V. Zwiller, *Rydberg excitons in Cu_2O microcrystals grown on a silicon platform*, Communications Materials **1**, 1–7 (2020).

[236] T. Kazimierczuk, D. Fröhlich, S. Scheel, H. Stolz, and M. Bayer, *Giant Rydberg excitons in the copper oxide Cu_2O*, Nature **514**, suppl. (2014).

[237] H. Matsumoto, K. Saito, M. Hasuo, S. Kono, and N. Nagasawa, *Revived interest on yellow-exciton series in Cu2C: An experimental aspect*, Solid state communications **97**, 125–129 (1996).

[238] J. Heckötter, D. Janas, R. Schwartz, M. Aßmann, and M. Bayer, *Experimental limitation in extending the exciton series in Cu_2O towards higher principal quantum numbers*, Physical Review B **101**, 235207 (2020).

[239] M. A. M. Versteegh, S. Steinhauer, J. Bajo, T. Lettner, A. Soro, A. Romanova, S. Gyger, L. Schweickert, A. Mysyrowicz, and V. Zwiller, *Giant Rydberg excitons in Cu_2O probed by photoluminescence excitation spectroscopy* (2021), arXiv:2105.07942 [cond-mat.mes-hall] .

[240] M. French, R. Schwartz, H. Stolz, and R. Redmer, *Electronic band structure of Cu_2O by spin density functional theory*, Journal of Physics: Condensed Matter **21**, 015502 (2008).

[241] F. Schöne, S.-O. Krüger, P. Grünwald, H. Stolz, S. Scheel, M. Aßmann, J. Heckötter, J. Thewes, D. Fröhlich, and M. Bayer, *Deviations of the exciton level spectrum in Cu_2O from the hydrogen series*, Physical Review B **93**, 075203 (2016).

[242] H. Stolz, F. Schöne, and D. Semkat, *Interaction of Rydberg excitons in cuprous oxide with phonons and photons: optical linewidth and polariton effect*, New Journal of Physics **20**, 023019 (2018).

[243] F. Schweiner, J. Main, G. Wunner, M. Freitag, J. Heckötter, C. Uihlein, M. Aßmann, D. Fröhlich, and M. Bayer, *Magnetoexcitons in cuprous oxide*, Physical Review B **95**, 035202 (2017).

[244] M. Aßmann, J. Thewes, D. Fröhlich, and M. Bayer, *Quantum chaos and breaking of all anti-unitary symmetries in Rydberg excitons*, Nature materials **15**, 741–745 (2016).

[245] S. Zielińska-Raczyńska, D. Ziemkiewicz, and G. Czajkowski, *Electro-optical properties of Rydberg excitons*, Physical Review B **94**, 045205 (2016).

[246] B. R. Johnson, J. O. Hirschfelder, and K.-H. Yang, *Interaction of atoms, molecules, and ions with constant electric and magnetic fields*, Reviews of Modern Physics **55**, 109–153 (1983).

[247] L. D. Landau and E. M. Lifšic, *Quantenmechanik*, 9th ed., Lehrbuch der theoretischen Physik Bd. 3 (Akademie-Verlag, Berlin, 1979).

[248] P. Ehrenfest, *Bemerkung über die angenäherte Gültigkeit der klassischen Mechanik innerhalb der Quantenmechanik*, Zeitschrift für Physik **45**, 455–457 (1927).

[249] Y. Yafet, R. Keyes, and E. Adams, *Hydrogen atom in a strong magnetic field*, Journal of Physics and Chemistry of Solids **1**, 137–142 (1956).

[250] T. Stielow, *Angular-Momentum Couplings in Ultra-Long-Range Giant Dipole Molecules*, Master's thesis, Universität Rostock (2017).

[251] F. Schweiner, J. Main, M. Feldmaier, G. Wunner, and C. Uihlein, *Impact of the valence band structure of Cu_2O on excitonic spectra*, Physical Review B **93**, 195203 (2016).

[252] F. Schweiner, *Theory of excitons in cuprous oxide*, Ph.D. thesis, Universität Stuttgart (2017).

[253] S. O. Krüger, *Rydberg excitons in external fields*, Ph.D. thesis, Universität Rostock (2021).

[254] L. A. P. Gallagher, J. P. Rogers, J. D. Pritchett, R. A. Mistry, D. Pizzey, C. S. Adams, M. P. A. Jones, P. Grünwald, V. Walther, C. Hodges, W. Langbein, and S. A. Lynch, *Microwave-optical coupling via Rydberg excitons in cuprous oxide* (2021), arXiv:2109.09614 [cond-mat.mes-hall] .

[255] I. N. Bronstein, J. Hromkovic, B. Luderer, H.-R. Schwarz, J. Blath, A. Schied, S. Dempe, G. Wanka, and S. Gottwald, *Taschenbuch der Mathematik*, Vol. 1 (Springer-Verlag, 2012).

[256] I. N. Bronshtein and K. A. Semendyayev, *Handbook of mathematics* (Springer Science & Business Media, 2013).

[257] G. E. Uhlenbeck and L. S. Ornstein, *On the theory of the Brownian motion*, Physical review **36**, 823 (1930).

[258] OpenAI, *Deep Deterministic Policy Gradient*, `https://spinningup.openai.com/en/latest/algorithms/ddpg.html` (2018).

[259] J. Shertzer, J. Ackermann, and P. Schmelcher, *Positronium in crossed electric and magnetic fields: The existence of a long-lived ground state*, Physical Review A **58**, 1129 (1998).

[260] W. R. Hamilton, *On a new species of imaginary quantities, connected with the theory of quaternions*, Proceedings of the Royal Irish Academy (1836-1869) **2**, 424–434 (1840 – 1844).

[261] J. Pujol, *Hamilton, Rodrigues, Gauss, quaternions, and rotations: a historical reassessment*, Communications in Mathematical Analysis **13**, 1–14 (2012).

[262] F. Ó Cairbre, *Twenty years of the Hamilton walk*, Irish Mathematical Society Bulletin **65**, 33–49 (2010).

[263] J. W. Gibbs, *Vector analysis: A text-book for the use of students of mathematics and physics, founded upon the lectures of J. Willard Gibbs* (Yale University Press, 1901).

[264] V. A. Kosteleckỳ and M. M. Nieto, *Analytical wave functions for atomic quantum-defect theory*, Physical Review A **32**, 3243 (1985).

[265] I. I. Sobelman, *Atomic spectra and radiative transitions*, Vol. 12 (Springer Science & Business Media, 2012).

Danksagung

An dieser Stelle möchte ich allen beteiligten Personen danken, die mich während meiner Promotion begleitet und auf viele Weisen unterstützt haben:

Zuallererst möchte ich mich bei meinem Doktorvater Stefan Scheel bedanken. Diese Arbeit wurde nur durch seien fachliche Betreuung und seien Offenheit für neue Themen ermöglicht. Ich danke ihm für die Chance und die Unterstützung das Hauptthema meiner Arbeit zu wechseln. Auch möchte ich ihm für das Verständnis für meine familiäre Situation danken, insbesondere während der Corona Zeit.

Ein allergrößter großer Dank gilt auch meiner Partnerin Imke für die Unterstützung und Geduld, auch während der vielen kleinen Katastrophen in dieser Zeit. Erwähnt sei auch unser gemeinsamer Sohn Björn, der seit Anfang des Promotionsprojekts dabei ist und Papa immer nur widerwillig zur Arbeit gehen lässt. Ich liebe euch beide!

Danken möchte ich auch meinen Eltern und meiner Schwester für die moralische Unterstützung, immer ehrlichen Rat und manchmal sehr spontane Kinderbetreuung.

Ein großer Dank gilt auch meinen Freunden und Kommilitionen aus Rostock die meinen Weg zum Teil schon seit zehn Jahre begleiten und durch die diese Zeit einzigartig geworden ist.

Darüber hinaus danke ich auch meinen Kollegen aus der Arbeitsgruppe und dem NEISS Projekt, mit denen ich mich regelmäßig fachlich austauschen konnten und auch bei Lösungen für die vielen verschiedenen Probleme helfen konnten. Darunter ganz besonders Robin, der mir bei den ersten Schritten in der Welt des Deep Learnings geholfen hat.

Nicht zuletzt danke ich dem *Evangelischen Studienwerks Villigst* für die finanzielle und ideelle Förderung durch ein Promotionsstipendium. Desweiteren wurde dieser Arbeit in Teilen finanziert vom Europäischen Sozialfonds (ESF) und dem Ministerium für Bildung, Wissenschaft und Kultur des Landes Mecklenburg-Vorpommern innerhalb des Projekts *NEISS - Neural Extraction of Information, Structure and Symmetry in Images* (ESF/14-BM-A55-0007/19), sowie von der Deutschen Forschungsgemeinschaft (DFG) im Rahmen des Schwerpunktprogramms SPP 1929: *Giant Interactions in Rydberg Systems* (GiRyd).