# Deep learning for surface scattering data analysis

**Dissertation**

der Mathematisch-Naturwissenschaftlichen Fakultät
der Eberhard Karls Universität Tübingen
zur Erlangung des Grades eines
Doktors der Naturwissenschaften
(Dr. rer. nat.)

vorgelegt von
Vladimir Starostin
aus Moskau, Russland

Tübingen
2023

Gedruckt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät der Eberhard Karls Universität Tübingen.

| | |
|---|---|
| Tag der mündlichen Qualifikation: | 16.05.2024 |
| Dekan: | Prof. Dr. Thilo Stehle |
| 1. Berichterstatter: | Prof. Dr. Frank Schreiber |
| 2. Berichterstatter: | Prof. Dr. Martin Oettel |
| 3. Berichterstatter: | Prof. Dr. Robert Feidenhans'l |

# Abstract

X-ray and neutron surface scattering techniques, encompassing a range of methodologies such as reflectometry and grazing incidence diffraction, are invaluable across a wide spectrum of scientific and technological applications. The capabilities of these techniques are being continually expanded due to ongoing enhancements of modern synchrotron and neutron sources. These developments offer increased resolution and sensitivity, alongside unprecedented volumes of data, facilitating a more comprehensive and detailed understanding of surfaces and interfaces. However, the vast volume and complexity of data generated have begun to outpace the capabilities of traditional data processing tools. This trend underscores that data analysis has become a major bottleneck in experimental science, emphasizing the urgent need for advanced data processing strategies to fully harness the potential of surface scattering techniques. A promising solution is the emerging field of deep learning techniques, which is revolutionizing scientific discovery by providing rapid processing of complex, high-dimensional data. Nonetheless, significant modifications are required to apply these existing techniques effectively to surface scattering data.

This work focuses on two major surface scattering techniques - specular reflectometry and grazing-incidence wide-angle scattering - and introduces innovative solutions for advancing data analysis through deep learning tools specifically designed for surface scattering data.

Firstly, the specular reflectometry technique, essential for studying systems like thin films and layered structures, is examined. Reflectivity data analysis poses significant challenges due to the infamous phase problem, which introduces potential ambiguity into data interpretation. Thus, a comprehensive understanding of the examined sample is crucial for successful analysis, demanding the integration of this experimenter-acquired prior knowledge into data analysis pipelines.

The research in this work demonstrates two deep learning methods for enhancing reflectometry analysis using the most widely employed statistical frameworks - maximum likelihood estimation and Bayesian inference. These advanced methods facilitate real-time reflectometry analysis for complex structures with multiple

estimated parameters. The combination of probabilistic machine learning with conventional tools offers an unparalleled level of precision and reliability of the solution. This is achieved by introducing a new class of prior-aware deep learning methods, which allows the dynamic setting of prior knowledge, a critical factor in handling the phase problem. The introduced methods can be equally applied to other inverse problems.

Secondly, this work discusses the first reported fully automated pipeline for the analysis of grazing-incidence wide-angle scattering, enabling the processing of massive amounts of data in real time. The core of the solution is a modern deep learning object detection technique, tailored to the specifics of the data. In conjunction with traditional data processing tools, the method offers an exhaustive analysis of diffraction data including phase identification, determination of coexisting phases, and lattice parameter refinement.

All methods demonstrated herein pave the way for a fast, fully-automated analysis of surface scattering data, proposing new standards for deep learning-based data processing.

# Deutsche Zusammenfassung

Röntgen- und Neutronenoberflächenstreuungstechniken, die eine Reihe von Methoden wie Reflectometrie und Streuung unter streifendem Einfall umfassen, sind in einem breiten Spektrum von wissenschaftlichen und technologischen Anwendungen von großem Wert. Die Einsatzmöglichkeiten dieser Techniken werden durch die laufenden Verbesserungen an modernen Synchrotron- und Neutronenquellen. Diese Entwicklungen bieten neben einer erhöhte Auflösung und Empfindlichkeit auch sehr große Datenmengen, und ermöglichen ein umfassenderes und detaillierteres Verständnis von Oberflächen und Grenzflächen. Allerdings hat das enorme Volumen und die Komplexität der erzeugten Daten begonnen, das Leistungsvermögen traditioneller Datenverarbeitungswerkzeuge zu übersteigen. Dieser Trend zeigt auf, dass die Datenanalyse zu einem großen Engpass in der experimentellen Wissenschaft geworden ist und signalisiert die dringende Notwendigkeit fortschrittlicher Datenverarbeitungsstrategien, um das Potenzial der Oberflächenstreuungstechniken vollständig ausschöpfen zu können. Eine vielversprechende Lösung ist das aufkommende Gebiet der Deep-Learning-Techniken, die das wissenschaftliche Arbeiten revolutionieren, indem sie eine schnelle Verarbeitung von komplexen, hochdimensionalen Daten ermöglichen. Dennoch sind erhebliche Modifikationen erforderlich, um diese bestehenden Techniken effektiv auf Oberflächenstreuungsdaten anzuwenden.

Diese Arbeit konzentriert sich auf zwei wichtige Oberflächenstreuungstechniken - spekulare Reflektometrie und Weitwinkelstreuung unter streifendem Einfall - und führt innovative Lösungsansätze ein, um die Datenanalyse durch speziell für Oberflächenstreuungsdaten konzipierte Deep-Learning-Werkzeuge zu verbessern.

Zunächst wird die Technik der spekularen Reflektometrie, die für die Untersuchung von Systemen wie Dünnschichten und geschichteten Strukturen unerlässlich ist, untersucht. Die Analyse von Reflexionsdaten stellt aufgrund des bekannten Phasenproblems, das potenzielle Mehrdeutigkeiten in die Dateninterpretation einführt, erhebliche Herausforderungen dar. Daher ist ein umfassendes Verständnis der untersuchten Probe entscheidend für eine erfolgreiche Analyse und erfordert die

Integration dieses vom Experimentator erworbenen Vorwissens in die Datenanalysepipelines.

Diese Arbeit demonstriert zwei Deep-Learning-Methoden zur Verbesserung der Reflektometrieanalyse mit den am weitesten verbreiteten statistischen Modellen - Maximum-Likelihood-Schätzung und Bayes'sche Inferenz. Diese fortgeschrittenen Methoden ermöglichen eine Echtzeitanalyse von Reflektometriemessungen von komplexen Strukturen mit mehreren unbekannten Parametern. Die Kombination probabilistischer Methoden mit konventionellen Werkzeugen bietet ein unübertroffenes Maß an Präzision und Zuverlässigkeit der Lösung. Dies wird erreicht durch die Einführung einer neuen Klasse von Deep-Learning-Methoden, die das dynamische Setzen von Vorwissen ermöglichen, was bei der Lösung des Phasenproblems einen entscheinder Faktor ist. Die vorgestellten Methoden können gleichwertig auf andere inverse Probleme angewendet werden.

Als Zweites demonstriert diese Arbeit die erste publizierte vollautomatisierte Pipeline für die Echtzeitanalyse von Weitwinkelstreuung unter streifendem Einfall. Das Kernkonzept dieser Anwendung ist eine moderne Deep-Learning-Objekterkennungstechnik, die an die spezifischen Eigenschaften der Daten angepasst ist. In Verbindung mit traditionellen Datenverarbeitungswerkzeugen bietet die Methode eine umfassende Analyse von Beugungsdaten einschließlich Phasenidentifikation, der Bestimmung von koexistierenden Phasen und der Verfeinerung von Gitterparametern.

Alle hier gezeigten Methoden ebnen den Weg für eine schnelle, vollautomatisierte Analyse von Oberflächenstreuungsdaten und stellen neue mögliche Standards für die datenbasierte Verarbeitung mit Deep Learning dar.

# Contents

# 1 Introduction

## 1.1 Scientific background and motivation

X-ray and neutron surface scattering techniques are indispensable in the landscape of modern experimental science [1–6]. These methods enable non-destructive examination of surfaces and interfaces, a fundamental aspect critical to a broad spectrum of scientific disciplines, proving to be pivotal in the study of diverse materials, ranging from semiconductors and metals to biological membranes and polymers [7–13].

X-ray and neutron surface scattering techniques are progressively improved via ongoing advancements in contemporary synchrotron and neutron sources [14]. These enhancements yield improved resolution and sensitivity, alongside an unparalleled amount of data. However, the immense quantity and complexity of the resultant data are beginning to challenge the capabilities of conventional data analysis tools. This situation delineates data analysis as the major bottleneck in experimental science and underscores the requirement for novel data processing strategies to fully exploit the potential of surface scattering techniques.

Deep learning, an emerging field presently transforming the domain of scientific discovery, appears to offer a feasible solution. Deep learning techniques based on artificial neural networks (NNs) offer rapid processing of complex, high-dimensional data, thereby promising to mitigate the limitations of traditional data analysis tools. Its scientific applications range from bioinformatics and genomic analysis [15–17] to astronomy [18, 19] and climate science [20]. Physics has been strongly impacted by deep learning techniques [21–26], including data-driven material science applications [27–32] and scattering data analysis [33–49]. Indeed, significant research is being

directed towards the development of deep learning techniques for surface scattering data [50].

This dissertation focuses on two surface scattering experimental techniques: specular reflectometry and grazing-incidence wide-angle scattering (GIWAS). Both techniques stand as well-recognized and frequently employed methods of surface scattering [2, 4–6, 51–53]. X-ray and neutron reflectometry have seen widespread usage in examining a broad range of systems such as liquid and solid thin films [54–60], polymer layers [61, 62], lipids [13, 63], self-assembled monolayers [64, 65], magnetic materials [66], and organic semiconductors (OSCs) [67–69], including both *ex situ* and real-time *in situ* experiments [70–72]. Grazing-incidence wide-angle scattering (GIWAS) is the key method for investigating crystalline structures on surfaces [2, 4, 6]. The fixed surface-sensitive grazing incidence geometry enables real-time, non-destructive probing of crystal structures during their synthesis [1], in particular, making this method indispensable for studying perovskite crystallization processes to design future-generation solar cells [73–80].

The analysis of reflectometry data comprises a complex optimization task in high-dimensional space featuring multiple local extrema, largely constraining the use of conventional numerical methods. The standard statistical framework employed is Maximum Likelihood Estimation (MLE), aiming to find the single sample configuration that best aligns with the observed data. Standard reflectometry packages [81–83] obtain conventional MLE-based "Parratt fit" through traditional global optimization algorithms. These algorithms undertake a random exploration of the parameter space, leading to a potentially lengthy iterative process with no success guarantee. Numerous methods have proposed the use of deep learning to improve and expedite this analysis [84–91], showing promising results on simple experimental scenarios. However, the inherent limitations of the overall framework include the possibility of overlooking potential solutions and the absence of uncertainty estimation. To counter these issues, the Bayesian analysis of reflectometry data is gaining increased traction [92], which aims to estimate the entire distribution of estimated parameters given the observed data. Bayesian inference is currently addressed by

the combination of global optimization algorithms with Markov Chain Monte Carlo methods [83], although a compelling strategy for handling multiple solutions and high-dimensional scenarios is yet to be proposed. The research in this work improves current neural network-based solutions for reflectometry analysis and further extends the capabilities of this approach by proposing two novel deep learning methods, advancing both Maximum Likelihood Estimation and Bayesian analysis of reflectivity data [93–95].

The analysis of diffraction images produced by GIWAS measurements varies significantly based on the application and scientific task; however, a common key task is the accurate determination of Bragg peak positions and their properties. Typical *in situ* experiments at synchrotron facilities generate hundreds of thousands of diffraction images per day, vastly exceeding the capabilities of conventional data processing methods. Several methods have been proposed to expedite semi-automated extraction of peak positions via deep learning techniques [41, 42]. This work delves further into this direction, demonstrating, to our knowledge, the first fully automated pipeline for deep learning-based real-time analysis of *in situ* GIWAS data [96, 97].

## 1.2 Structure of this thesis

The three chapters following the introduction of this thesis introduce the methods fundamental to this work. Chap. 2 discusses principles of surface scattering and the experimental techniques relevant to this thesis: specular reflectometry and grazing-incidence wide-angle scattering. Chap. 3 provides an overview of conventional methods for analysis of reflectometry data. Chap. 4 discusses principles of deep learning and certain techniques employed in this work.

The fundamental methods and principles introduced in the initial chapters provide the groundwork for the original research and discussions that follow. Chap. 5 presents a novel deep learning method for reflectometry analysis, which significantly improves upon previously reported Maximum Likelihood Estimation approaches by integrating prior knowledge into the deep learning pipeline. Chap. 6 further delves into reflectometry analysis, introducing a pioneering approach to deep learning-based Bayesian analysis of reflectivity data. Chap. 7 shifts focus to a different surface scattering technique, discussing the first reported approach to fully automated analysis of grazing-incidence wide-angle scattering data based on a deep learning object detection solution.

Chap. 8 summarizes the results of this work and offers an outlook on potential future research avenues. The results demonstrated in this dissertation are, to a considerable extent, based on Refs. [93–97].

# 2 Experimental methods

## 2.1 Principles of scattering theory

### 2.1.1 X-Ray wave propagation

X-rays are electromagnetic waves with wavelengths in the region of $\lambda$ = 0.1 - 100 Ångströms (1 Å = $10^{-10}$ m). According to the wave-particle duality [98], an electromagnetic wave can be described as a stream of photons - quanta of the electromagnetic field. This description becomes essential for understanding quantum effects in the interaction of the electromagnetic field with matter. For the purposes of this work, it is sufficient to use wave formalism.

A monochromatic electromagnetic wave with wavelength $\lambda$ is a transverse wave [6] propagating in the direction of a wave vector $\mathbf{k}$ ($|\mathbf{k}| = \frac{2\pi}{\lambda}$) with the electric and magnetic fields oscillating perpendicular to the wave vector with the angular frequency $w$. Throughout this work, we do not consider magnetic effects, so we focus on the electric field component $\mathbf{E}$ of the X-ray electromagnetic field. At position $\mathbf{r}$ and time $t$, an electric field of a plane monochromatic wave can be described as

$$\mathbf{E}(\mathbf{r}, t) = \boldsymbol{\epsilon} E_0 e^{i(\mathbf{k}\cdot\mathbf{r} - wt)}, \tag{2.1}$$

where $\boldsymbol{\epsilon}$ is a polarization unit vector. In the following section, we consider how the plane X-ray wave is scattered from the medium.

### 2.1.2 Interaction of X-Rays with Matter

In this work, we focus on *elastic* scattering, where the energy of the scattered photons is not altered by the interaction with matter. In contrast, inelastic X-ray scattering involves the transfer of energy between the incident X-ray and the material being studied. The simplest case of a photon inelastically scattered on a free electron is described by Compton scattering [99]. Furthermore, X-rays can lead to photoionization of the atoms in the studied material depending on the used wavelength and energy levels. For the discussion in this thesis, we only consider non-resonant elastic scattering.

**Scattering from an electron** is the simplest case we shall consider first. The elastic scattering from a single electron is called Thompson scattering and results in a spherical wave:

$$\mathbf{E}_f(\mathbf{R}, t) \propto \boldsymbol{\epsilon}' \frac{e^{-ikR}}{R} E_{in}, \tag{2.2}$$

where $R$ is the distance from the electron, $\boldsymbol{\epsilon}' = \boldsymbol{\epsilon}'(\boldsymbol{R})$ is a polarization unit vector of a scattered wave, and $E_{in}$ is the amplitude of the incident electric field. The exact formula can be derived via classical Maxwell equations and has the following form:

$$\mathbf{E}_f(\mathbf{R}, t) = -\boldsymbol{\epsilon}' |\boldsymbol{\epsilon} \cdot \boldsymbol{\epsilon}'| r_e \frac{e^{-ikR}}{R} E_{in}, \tag{2.3}$$

where $r_e \approx 2.814 \times 10^{-5}$ Å is the classical electron radius. The coefficient $|\boldsymbol{\epsilon} \cdot \boldsymbol{\epsilon}'|$ depends on the wave polarization, and the minus sign corresponds to the phase shift by $\pi$ between the incident and the scattered waves.

Typically, X-ray detectors only measure the intensity of scattered electromagnetic waves:

$$I = \mathbf{E}\mathbf{E}^*, \tag{2.4}$$

where $\mathbf{E}^*$ denotes the complex conjugate of an electric field. The scattered intensity from a single electron is equal to

$$I(\mathbf{R})/I_0 = r_e^2 P/R^2. \tag{2.5}$$

Here $I_0 = E_{in}^2$ is the intensity of the incoming beam, and $P = |\boldsymbol{\epsilon} \cdot \boldsymbol{\epsilon}'|^2$ is a polarization factor that depends on the polarization of the incoming beam. Leaving aside the general coefficient $P/R^2$, we can define a scattering length $b$ of an object that characterizes how strongly it scatters. In this way, for a single electron, the scattering length equals the minus classical electron radius, which is also called the Thomson scattering length:

$$b_e = -r_e \tag{2.6}$$

(as before, minus corresponds to the phase shift). Using Eq. 2.6, it is natural to introduce *scattering length density (SLD)* as a measure of the scattering power of a material $\rho(r)$, which in the case of X-rays depends on electron density $\rho_e(r)$:

$$\rho(r) = -r_e \rho_e(r). \tag{2.7}$$

**Scattering from an atom** can be derived via a classical approach as a superposition of contributions from different volumes of a continuous electron density distribution $\rho_e(\mathbf{r})$ of an atom, as illustrated in Fig. 2.1. An infinitesimally small volume $d\mathbf{r}$ has a scattering length $b_e d\rho(\mathbf{r}) = -r_e \rho(\mathbf{r}) d\mathbf{r}$ which contributes to the scattered field with a phase factor $e^{i\Delta\phi} = e^{i\mathbf{r}\cdot\mathbf{q}}$, where $\mathbf{q} = \mathbf{k_f} - \mathbf{k_i}$ is the momentum transfer vector. Therefore, the total scattering length of the atom can be written as

$$b_{atom}(\mathbf{q}) = -r_e \int \rho_e(\mathbf{r}) e^{i\mathbf{q}\cdot\mathbf{r}} d\mathbf{r} = -r_e f(\mathbf{q}). \tag{2.8}$$

Here $f(\mathbf{q})$ is the atomic form factor. According to Eq. 2.8, it can be calculated as the Fourier transform of the electron density $\rho_e(\mathbf{r})$. Notably, $f(0) = r_e \cdot Z$, where $Z$

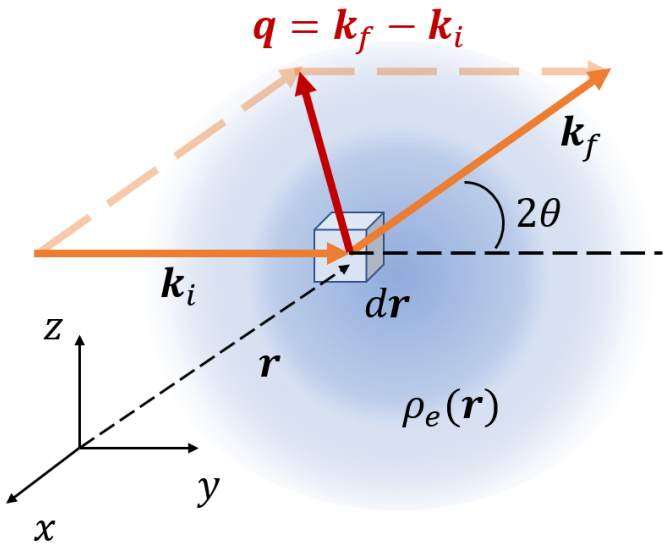


Figure 2.1: Scattering from a cloud of electron density $\rho_e(\mathbf{r})$. The moment transfer vector $\mathbf{q}$ is the difference between the final and incoming wave vectors, and the angle $2\theta$ is the scattering angle.

is the atomic number. By taking into account quantum effects, one can refine Eq. 2.8 by adding the *dispersion corrections* $f'$ and $f''$ such that

$$f(\mathbf{q}) = f^0(\mathbf{q}) + f' + if'', \tag{2.9}$$

where $f^0(\mathbf{q}) = FT(\rho_e(\mathbf{r}))$. The correction term $f''$ depends on the wavelength $\lambda$ and describes the absorption of the medium.

**Scattering from a molecule.** Analogically to the form factor of an atom, we can derive a form factor of a molecule composed of multiple atoms:

$$F^{mol}(\mathbf{q}) = \sum_j f_j(\mathbf{q}) e^{i\mathbf{q}\cdot\mathbf{r_j}}. \tag{2.10}$$

As before, we calculate the superposition of the scattered fields from atoms with the atomic form factors $f_j(\mathbf{q})$ and the corresponding phase shift factors $e^{i\mathbf{q}\cdot\mathbf{r_j}}$, where $j$ is an atom index and $r_j$ is the position of an $j$th atom.

### 2.1.3 Diffraction from a crystal

A crystal is characterized by the periodic *lattice structure* of atoms. The lattice structure is defined by three *primitive translation vectors* $\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3$ that construct an infinite array of a *Bravais lattice*:

$$\mathbf{R}_n = n_1\mathbf{a}_1 + n_2\mathbf{a}_2 + n_3\mathbf{a}_3, \tag{2.11}$$

where $n_i \in \mathbb{Z}$ are integers ($n \in \mathbb{Z}^3$). An ideal crystal is invariant to translations along any of the $R_n$ vectors:

$$\rho_e(\mathbf{r}) = \rho_e(\mathbf{r} + \mathbf{R}_n). \tag{2.12}$$

The volume constructed by the basis vectors $\{\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3\}$ defines a *unit cell* of the crystal. Due to the translational symmetry, the positions of any atom in a crystal can be expressed as $\boldsymbol{R}_n + \boldsymbol{r}_j$, where $\boldsymbol{R}_n$ represents the position of the unit cell (as per Eq. 2.11), and $\boldsymbol{r}_j$ denotes the absolute position of a $j$th atom (e.g., relative to a reference unit cell).

Therefore, the form factor of a crystal can be split into two terms:

$$F^{\text{crystal}}(\mathbf{q}) = \overbrace{\sum_j f_j(\mathbf{q}) e^{i\mathbf{q}\cdot\mathbf{r}_j}}^{\text{Unit cell structure factor}} \overbrace{\sum_n e^{i\mathbf{q}\cdot\mathbf{R_n}}}^{\text{Lattice sum}}. \tag{2.13}$$

Here, the first term called the *unit cell structure factor* is a sum of all the atoms within a single unit cell, while the second term is a sum of all $N$ unit cells within a crystal. One can observe that the second sum is of order unity for a random vector $\mathbf{q}$, but it becomes of order $N$ if the following condition is fulfilled:

$$\mathbf{q} \cdot \mathbf{R}_n = 2\pi \times \text{integer}. \tag{2.14}$$

Therefore, one should expect a huge increase in scattered intensity in the directions corresponding to $\mathbf{q}$ values that satisfy Eq. 2.14 called the Laue condition - the necessary condition for observing a so-called Bragg peak [100] in the direction

corresponding to momentum transfer $\mathbf{q}$. To solve Eq. 2.14 w.r.t. $\mathbf{q}$, we shall first introduce the concept of the reciprocal lattice $\mathbf{G}_m$, which can be defined via Fourier transform series $\rho_m$ of a periodic Bravais lattice:

$$\rho_e(\mathbf{r}) = \sum_{m_1,m_2,m_3} \rho_{m_1,m_2,m_3} e^{i\mathbf{G}_{m_1,m_2,m_3}\cdot\mathbf{r}} = \sum_m \rho_m e^{i\mathbf{G}_m\cdot\mathbf{r}}, \tag{2.15}$$

where $m = \{m_1, m_2, m_3\} \in \mathbb{Z}^3$.

Using the translation invariance property (Eq. 2.12), we find that

$$\sum_m \rho_m e^{i\mathbf{G}_m\cdot\mathbf{r}} = \sum_m \rho_m e^{i\mathbf{G}_m\cdot\mathbf{r}} e^{i\mathbf{G}_m\cdot\mathbf{R}_n} \tag{2.16}$$

.

The equality of two Fourier series implies the equality of the coefficients, therefore

$$e^{i\mathbf{G}_m\cdot\mathbf{R}_n} = 1 \tag{2.17}$$

for any $n, m \in \mathbb{Z}^3$, which can only hold if

$$\mathbf{G}_m \cdot \mathbf{R}_n = 2\pi \times \text{integer}. \tag{2.18}$$

This proves that the reciprocal vectors $\mathbf{G}_m$ satisfy the Laue condition (Eq. 2.14).

The reciprocal vectors can be explicitly defined based on the basis vectors in the real space:

$$\mathbf{a}_1^* = 2\pi\frac{\mathbf{a}_2 \times \mathbf{a}_3}{\mathbf{a}_1 \cdot (\mathbf{a}_2 \times \mathbf{a}_3)}, \quad \mathbf{a}_2^* = 2\pi\frac{\mathbf{a}_3 \times \mathbf{a}_1}{\mathbf{a}_1 \cdot (\mathbf{a}_2 \times \mathbf{a}_3)}, \quad \mathbf{a}_3^* = 2\pi\frac{\mathbf{a}_1 \times \mathbf{a}_2}{\mathbf{a}_1 \cdot (\mathbf{a}_2 \times \mathbf{a}_3)}. \tag{2.19}$$

In this way, Bragg peaks occur on a reciprocal lattice with momentum transfer:

$$\boldsymbol{q}_{hkl} = h\mathbf{a}_1^* + k\mathbf{a}_2^* + l\mathbf{a}_3^*, \tag{2.20}$$

where $h, k, l \in \mathbb{Z}$, $\{h, k, l\} = \{m_1, m_2, m_3\}$ are integers called Miller indices [101]. Each set of three integers defines a family of lattice planes of the given Bravais

lattice orthogonal to the vector $\boldsymbol{q}_{hkl}$. Therefore, the determination of the positions of Bragg peaks is essential for inferring the lattice parameters of the crystal structure under study and the peak intensities carry information about atomic positions within the unit cell as per Eq. 2.13.

## 2.2 Surface scattering techniques

X-ray surface scattering methods are indispensable tools in a variety of fields, encompassing both fundamental research and practical applications [1, 4–6, 102, 103]. They provide powerful ways to explore the structural properties of surfaces, interfaces, and thin films at length scales ranging from atomic dimensions to several micrometers.

In the following, we consider some general properties of X-ray surface scattering and discuss two experimental techniques of interest: X-ray reflectivity (XRR) and grazing-incidence wide-angle X-ray scattering (GIWAXS).

It is worth noting that the analogous neutron surface scattering techniques - neutron reflectivity (NR) and grazing-incidence wide-angle neutron scattering (GIWANS) - are equally relevant for various applications and share similar properties with X-ray scattering. The key difference is that, unlike X-rays that scatter on electrons, neutrons also scatter on atomic nuclei, making the corresponding techniques indispensable for various applications. For clarity, the following derivations focus on X-ray scattering, however, in most cases, the analogous results for neutrons can be obtained by simply substituting the SLD derived for X-rays by the SLD for neutron scattering that depends on atomic density and atomic scattering length [104].

### 2.2.1 Scattering from ideal interface

First, we consider the scattering of a plane monochromatic X-ray wave with wavelength $\lambda$ from an infinite flat interface between ambient and a homogeneous material characterized by SLD $\rho$. SLD of the medium determines the refractive index

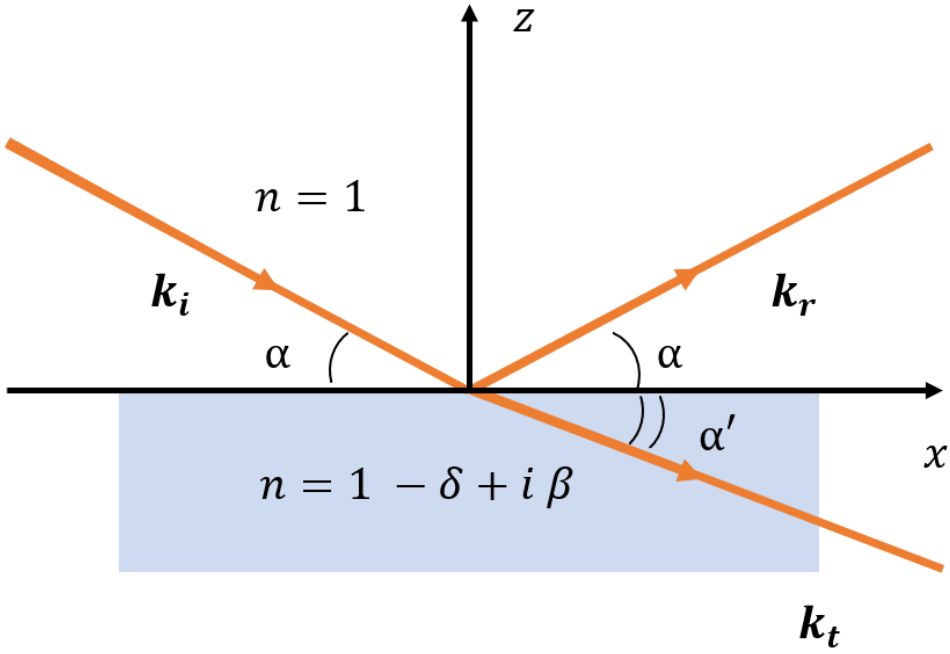


Figure 2.2: Scattering from a flat interface between two homogeneous media used to derive the Snell's law and the Fresnel equations.

$$n = 1 - \frac{\lambda^2}{2\pi}\rho = 1 - \delta + i\beta, \tag{2.21}$$

where $\delta = \frac{\lambda^2}{2\pi}\mathrm{Re}(\rho)$ is typically of order $10^{-5}$, and the imaginary part $\beta = -\frac{\lambda^2}{2\pi}\mathrm{Im}(\rho)$ corresponds to the absorption (see the term $f''$ in Eq. 2.9).

As illustrated in Fig. 2.2, an incoming wave $a_I e^{i\boldsymbol{k}_i\cdot\boldsymbol{r}}$ at an incidence angle $\alpha$ to the surface characterized by an amplitude $a_I$ and a wave vector $\boldsymbol{k}_i$ results in a specular reflected wave $a_R e^{i\boldsymbol{k}_r\cdot\boldsymbol{r}}$ and a transmitted wave $a_T e^{i\boldsymbol{k}_t\cdot\boldsymbol{r}}$ at angle $\alpha'$ to the surface. The boundary conditions requiring the wave and its derivative to be continuous at the interface ($z = 0$) directly lead to Snell's law:

$$\cos(\alpha) = n\cos(\alpha') \tag{2.22}$$

and Fresnel coefficients for small angles:

$$r \equiv \frac{a_R}{a_I} = \frac{\alpha - \alpha'}{\alpha + \alpha'}; t \equiv \frac{a_T}{a_I} = \frac{2\alpha}{\alpha + \alpha'}, \tag{2.23}$$

that are frequently more useful to consider in terms of momentum transfer $q$:

$$r = \frac{q - q'}{q + q'} \quad ; \quad t = \frac{2q}{q + q'}, \tag{2.24}$$

where $q = 2k \sin\alpha \approx 2k\alpha$, $q' = 2k \sin\alpha' \approx 2k\alpha'$.

Importantly, for X-rays, the refractive index $n$ is slightly less than unity, which leads to the effect of total external reflection for incidence angles below a *critical angle* $\alpha_c$ and the corresponding momentum transfer $q_c$:

$$\alpha_c \approx \sqrt{2\delta}, q_c \approx \sqrt{16\pi\rho} \tag{2.25}$$

as follows from the linear approximation of Eq. 2.22 for small angles. In the same way, one can obtain

$$q'^2 \approx q^2 - q_c^2. \tag{2.26}$$

Eq. 2.26 allows us to rewrite the Fresnel coefficients as functions of momentum transfer $q$:

$$r(q) = \frac{q - \sqrt{q^2 - q_c^2}}{q + \sqrt{q^2 - q_c^2}} \quad ; \quad t(q) = \frac{2q}{q + \sqrt{q^2 - q_c^2}} \ . \tag{2.27}$$

Consequently, the measured reflected intensity $|r(q)|^2$ (in $I_0 = |a_I|^2$ units) is a function of momentum transfer, which in the case of specular geometry, equals its $z$-component $q_z$:

$$R_F(q_z) = |r(q_z)|^2 \approx \frac{16\pi^2\rho^2}{q_z^4} \propto \frac{1}{q_z^4}, \tag{2.28}$$

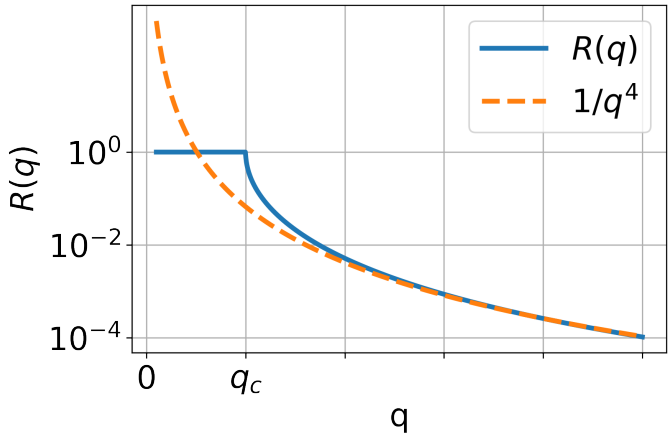


Figure 2.3: Fresnel reflectivity from a homogeneous material as a function of momentum transfer features an asymptotic behavior $\propto 1/q^4$. The critical angle corresponds to the total reflection edge at $q_c$.

where the "Fresnel reflectivity" $R_F(q_z)$ describes reflection from an infinitely thick and flat homogeneous slab with constant SLD $\rho$ (see Fig. 2.3). In the case of a graded interface characterized by average interface roughness $\sigma$, the corresponding decrease of the reflected intensity is well-described by the Névot-Croce factor $e^{-qq'\sigma^2} \approx e^{-q^2\sigma^2}$ (see also Subsec. 2.2.2):

$$R(q_z) = R_F(q_z)e^{-q^2\sigma^2}. \tag{2.29}$$

For more complex structures with SLD changing along the depth axis $z$, the measurement of reflected intensity $R(q)$ in specular geometry allows reconstructing the SLD profile of the studied material. The corresponding experimental technique is called reflectometry and is discussed in Subsec. 2.2.2.

Same as the refraction index, $\alpha' \in \mathbb{C}$:

$$\alpha' = \mathrm{Re}(\alpha') + i\,\mathrm{Im}(\alpha') \tag{2.30}$$

due to the absorption coefficient $\beta$. The absorption of the medium leads to the exponential decay of the amplitude $a_T$ of the transmitted wave:

$$a_T \mathrm{e}^{i(\mathrm{k}\alpha')z} = a_T \mathrm{e}^{ik\,\mathrm{Re}(\alpha')z} \mathrm{e}^{-k\,\mathrm{Im}(\alpha')z}. \tag{2.31}$$

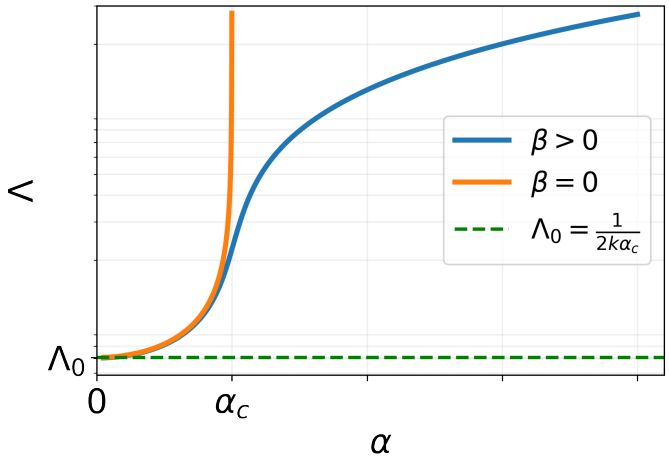


Figure 2.4: Penetration depth as a function of incidence angle approaches an asymptotic value $\Lambda_0 = \frac{1}{2k\alpha_c}$ for small incidence angles.

The corresponding characteristic depth $\Lambda$ at which the intensity of the transmitted wave decreases by a factor of $1/e$ is called *penetration depth*:

$$\Lambda = \frac{1}{2k\,\mathrm{Im}\,(\alpha')} \approx \frac{1}{\sqrt{2}k\sqrt{\alpha_c^2 - \alpha^2 + \sqrt{(\alpha_c^2 - \alpha^2)^2 + 4\beta^2}}}. \tag{2.32}$$

Fig. 2.4 illustrates penetration depth dependency on incidence angle. This dependency indicates that a small incidence angle around the critical angle $\alpha_c$ results in a so-called *evanescent wave* with a small penetration depth, approaching an asymptotic value $\Lambda_0 = \frac{1}{2k\alpha_c}$. This important property is the foundation of surface-sensitive experimental techniques conducted at *grazing incidence geometry* such as GIWAXS discussed in Subsec. 2.2.3.

### 2.2.2 Specular reflectivity measurements

X-ray and neutron reflectometry (XRR and NR) are well-established and indispensable experimental techniques commonly used to investigate the SLD profile along the direction perpendicular to the surface of a sample such as thin films and multilayers [5, 51–53, 105]. Reflectometry has been extensively used in both *in situ* and *ex situ* studies of a large variety of systems, such as liquid and solid thin films [55–60,

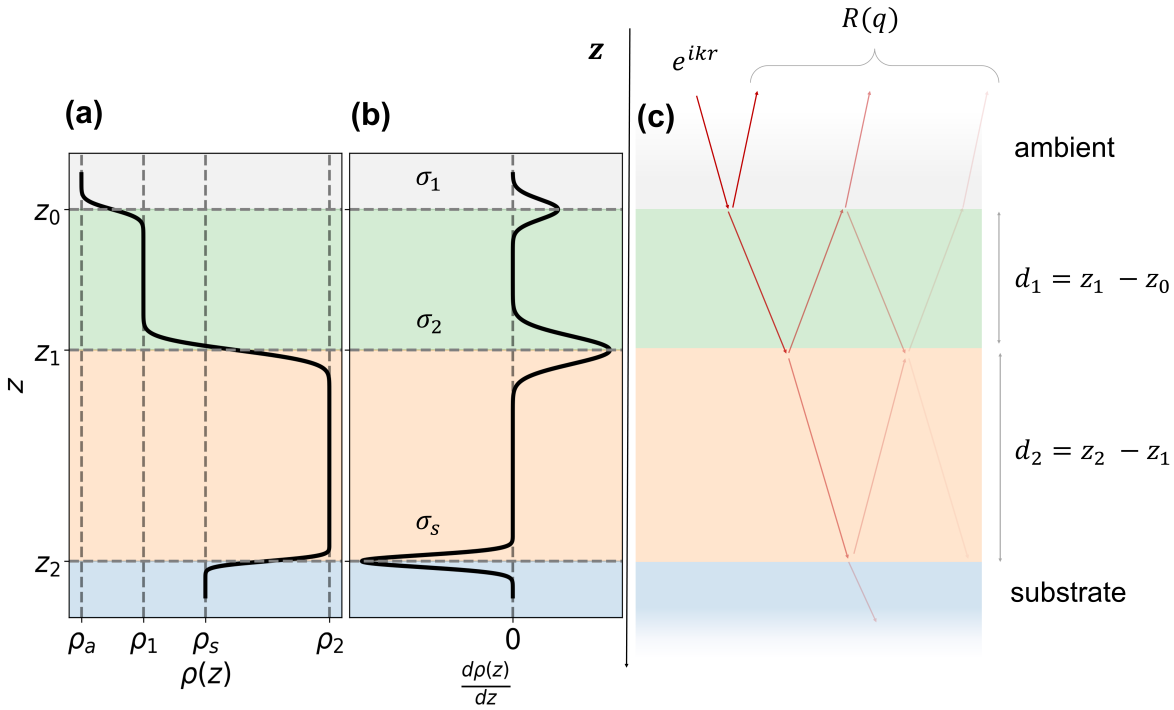


Figure 2.5: An illustration of the standard parameterization of the SLD profile $\rho(z)$ (a) with the corresponding derivative $\frac{d\rho(z)}{dz}$ (b) for a two-layered structure (c). In this manner, the sample is characterized by layer thicknesses $d_i$, interface roughnesses $\sigma_i$, and densities (SLD) $\rho_i$.

70, 72, 106–108], layers of polymers [61, 62], lipids [13, 63, 109, 110], self-assembled monolayers [64, 65] and organic solar cells [67–69]. In addition, polarized NR [66] can be used to study the magnetic properties of thin films.

Reflectometry measurements are conducted in specular geometry discussed in the previous section. The reflected intensity $R(q)$ from a sample as a function of momentum transfer $q$ contains information about the SLD profile $\rho(z)$, $z$ being the axis perpendicular to the surface of the sample. XRR is typically conducted by gradually increasing the incidence angle and the detector and measuring the reflected intensity in specular geometry at corresponding discrete $q$ points.

The most common way of modeling $\rho(z)$ of a studied layered sample is a so-called box model illustrated in Fig. 2.5. In the box model, the $i$th sample layer is characterized by a thickness $d_i$, an interface roughness $\sigma_i$, and a homogeneous SLD

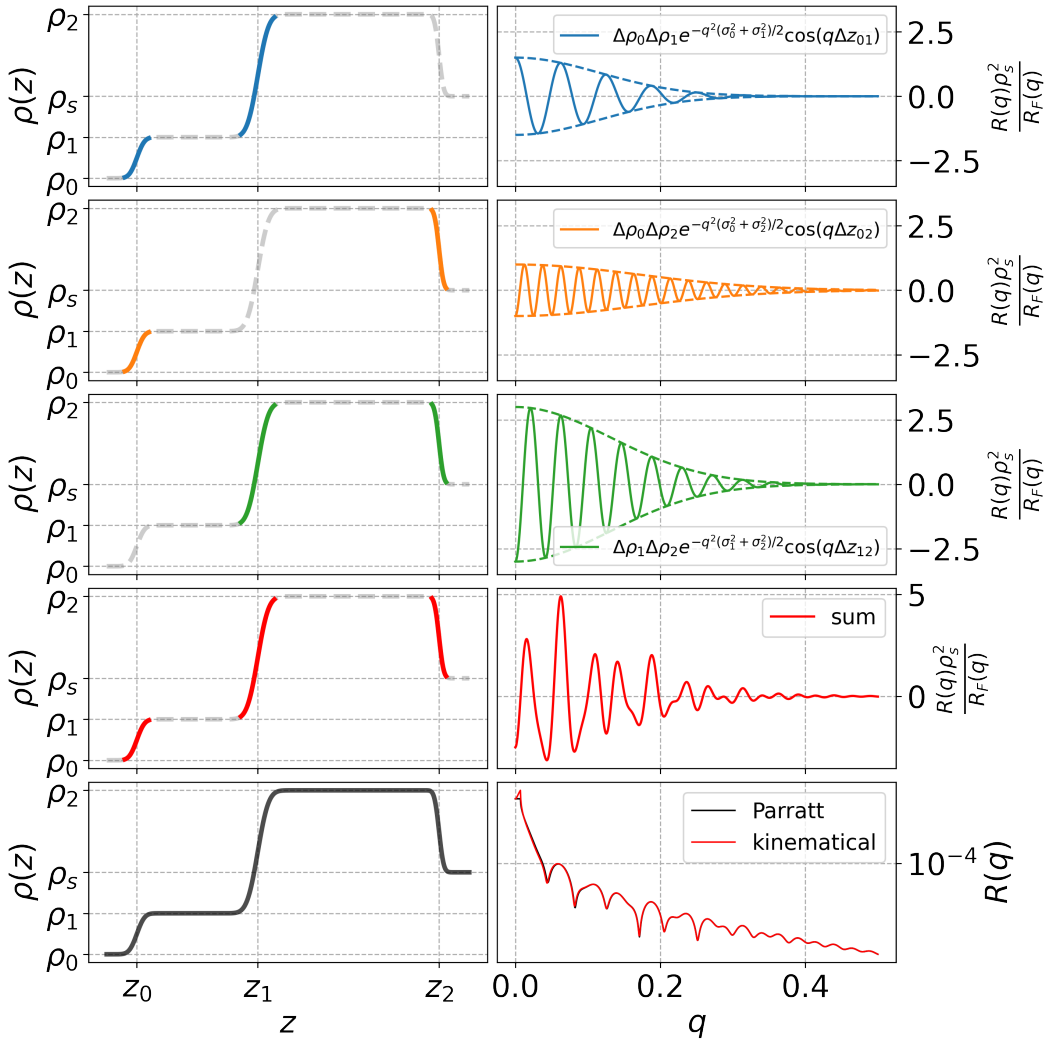


Figure 2.6: The kinematical approximation is decomposed into the interference contributions $\Delta\rho_i\Delta\rho_j e^{-q_z^2(\sigma_i^2+\sigma_j^2)/2}\cos(q_z\Delta z_{ij})$ between all (three) pairs of interfaces $(i,j)$, according to Eq. 2.35. The distinct contributions are illustrated by the top three pairs of plots, each denoting a colored pair of interfaces on the left-hand side and the resulting interference pattern on the right-hand side. Oscillation periods, $\Delta q_{ij} = \frac{2\pi}{\Delta z_{ij}}$, are determined by the distance $\Delta z_{ij}$ between a given pair of interfaces $(i,j)$. Oscillation amplitudes are determined by the corresponding density contrasts $\Delta\rho_i\Delta\rho_j$ and decrease with $q$ due to the interface roughnesses, as per term $e^{-q_z^2(\sigma_i^2+\sigma_j^2)/2}$. The fourth pair of plots illustrates Kiessig fringes as the sum of three interference components. Ultimately, the kinematical approximation (depicted by the red curve on the bottom right plot) is obtained by further adding the term $\sum_{i=1}^{N}\Delta\rho_i^2 e^{-q_z^2\sigma_i^2}$ and multiplying the resulting interference pattern by Fresnel reflectivity. The kinematical approximation deviates from an accurate Parratt computation (represented by the black curve on the bottom right plot) for smaller momentum transfer $q$ close to $q_c$.

$\rho_i$. Graded interfaces between the layers are modeled by a smoothly changing density profile. The commonly employed function has a derivative that follows a Gaussian function $\frac{d\rho(z)}{dz} = \Delta\rho e^{-\frac{(z-z_i)^2}{2\sigma_i^2}}$ with its width being equal to the interface roughness $\sigma_i$ (see the Névot-Croce factor introduced in Subsec. 2.2.1). The resulting SLD profile follows

$$\rho(z) = \rho_{\text{ambient}} + \sum_{i=1}^{N} \Delta\rho_i \cdot \text{erf}\left(\frac{z - z_i}{\sqrt{2}\sigma_i}\right), \tag{2.33}$$

where $\rho_{\text{ambient}} = \rho_0$ is the density of ambient, $\text{erf}(z)$ is the error function, $N$ is the number of interfaces, $z_i$ the position of the $i$th interface, $\sigma_i$ is the roughness of the $i$th interface and $\Delta\rho_i = \rho_i - \rho_{i-1}$. We note that this layered representation of the SLD profile can be straightforwardly extended to a general continuous case by providing a finite discretization of the profile and approximating it with a stack of thin layers.

As illustrated in Fig. 2.5c, an incoming X-ray beam results in multiple scattering events at the interfaces of a layered sample, with each event being described by Eq. 2.27. By combining the sum of resulting reflected waves with the corresponding phases, one can obtain the total theoretical reflectivity $R(q)$ given an SLD profile $\rho(z)$, which is a foundation of the classical Parratt formalism [54]. Alternatively, one can employ a computationally faster Abelès transfer-matrix method [111] that provides equivalent results.

Notably, by ignoring multiple scattering negligible at higher $q$ values, one can obtain a closed-form solution for $R(q)$, leading to the kinematical approximation for reflectometry:

$$R(q_z) = \frac{R_{\text{F}}(q_z)}{\rho_{\text{s}}^2} \left| \int_{-\infty}^{+\infty} \frac{d\rho(z)}{dz} e^{iq_z z} dz \right|^2 = \frac{R_{\text{F}}(q_z)}{\rho_{\text{s}}^2} \left| \text{FT}\left(\frac{d\rho(z)}{dz}\right) \right|^2, \tag{2.34}$$

where $R_{\text{F}}(q_z)$ denotes the Fresnel reflectivity introduced in Subsec. 2.2.1, $\rho_{\text{s}}$ is the SLD of the substrate, and FT stands for the Fourier transform. Eq. 2.34 provides

a good approximation for $q_z \gg q_c$. We employ it in the following to illustrate how different parameters influence the reflectivity curve $R(q)$.

By substituting Eq. 2.33 into Eq. 2.34, we can explicitly calculate the Fourier transform:

$$\left|\mathrm{FT}\left(\frac{d\rho(z)}{dz}\right)\right|^2 = \left|\int_{-\infty}^{+\infty}\sum_{i=1}^{N}\Delta\rho_i\frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{(z-z_i)^2}{2\sigma^2}}e^{iq_z z}dz\right|^2 = \left|\sum_{i=1}^{N}\Delta\rho_i e^{iq_z z_i - q_z^2\sigma_i^2/2}dz\right|^2 .$$

Finally, we obtain a decomposition of the scattered intensity into the sum of a constant term and $\frac{N(N-1)}{2}$ cosine components with amplitudes $\Delta\rho_i\Delta\rho_j e^{-q_z^2(\sigma_i^2+\sigma_j^2)/2}$ and frequencies $\Delta z_{ij}$:

$$\frac{R_{\mathrm{box}}(q_z)}{R_{\mathrm{F}}(q_z)}\rho_{\mathrm{s}}^2 = \sum_{i=1}^{N}\Delta\rho_i^2 e^{-q_z^2\sigma_i^2} + 2\sum_{i=2}^{N}\sum_{j=1}^{i-1}\Delta\rho_i\Delta\rho_j e^{-q_z^2(\sigma_i^2+\sigma_j^2)/2}\cos(q_z\Delta z_{ij}), \qquad (2.35)$$

where $\Delta z_{ij} = z_i - z_j$ is the distance between $i$th and $j$th interfaces. The first term determines the slope of the reflectivity curve relative to $R_F(q_z) \propto q_z^{-4}$, and the second term is the sum of all the $N(N-1)/2$ pairs between $N$ interfaces that produce interference patterns oscillating along $q$ with periods $\Delta q_{ij} = \frac{2\pi}{\Delta z_{ij}}$, as illustrated in Fig. 2.6. The resulting oscillations are called Kiessig fringes. As discussed in Subsec. 2.1.3, diffraction from crystalline materials results in Bragg peaks that can also be measured in specular geometry.

The reflectometry analysis aims at estimating the parameters $\theta = \{d_i, \sigma_i, \rho_i\}$ of the studied sample based on a measured reflectivity curve $R(q)$. In simple scenarios, some of these parameters can be obtained directly based on a shape of a curve. In this way, the thickness of a single-layered structure $d = \frac{2\pi}{\Delta q}$ can be obtained by simply estimating the period of Kiessig fringes. Estimating other parameters, especially in the case of multiple layers, requires numerical methods. Notably, there exist analytical solutions for restoring model-independent SLD profile such as the Gel'fand-Levitan-Marchenko inverse scattering equation [112]. However, they rely

on information about the phase $\phi(q)$ of the reflected wave $a_R(q)e^{i\phi(q)r}$, which cannot be measured in a typical reflectometry setup. The *phase problem* is common for scattering experiments that measure the intensity of scattered radiation. Generally, the phase problem leads to ambiguity, i.e. a multitude of theoretically possible samples resulting in the same scattered patterns. In the case of reflectometry, some of such scenarios are derived theoretically based on the kinematical approximation (Eq. 2.34), most known being the case of mirrored derivatives reported in earlier studies [113–115] and based on the following equality:

$$\left|\int_{-\infty}^{+\infty} \frac{d\rho(z)}{dz} e^{iq_z z} dz\right|^2 = \left|\int_{-\infty}^{+\infty} \frac{d\rho(-z)}{dz} e^{iq_z z} dz\right|^2 . \tag{2.36}$$

The equivalent solutions derived from the kinematical approximation generally diverge at low $q$ where multiple scattering is significant. On the other hand, measured curves provide noisy reflectivity levels for a finite number of $q$ points. Consequently, in practice, the number of ambiguous solutions can be even larger than what follows from the kinematical approximation.

Indeed, the ambiguity problem complexifies the reflectometry analysis. Apart from additional measurements, certain computational methods can be employed to address this issue. The conventional approaches to reflectometry analysis are discussed in Chap. 3.

### 2.2.3 Grazing-incidence wide-angle scattering

GIWAXS measurements are performed in grazing incidence geometry introduced in Subsec. 2.2.1 and illustrated in Fig. 2.7. Unlike reflectometry, GIWAXS is conducted at a constant incidence angle, with the scattered intensity being measured by 2D area detectors with wide scattering angles typically lying in ranges $\alpha_f \in [1°, 45°]$, or $|q| \in [0.1, 5]$ Å$^{-1}$, probing atomic and molecular distances in crystal lattices. Bragg peaks emerging at these scales are discussed in Subsec. 2.1.3. The same geometry can indeed be applied to larger length scales by measuring scattering intensity at small angles, and the corresponding technique is called grazing-incidence small-angle

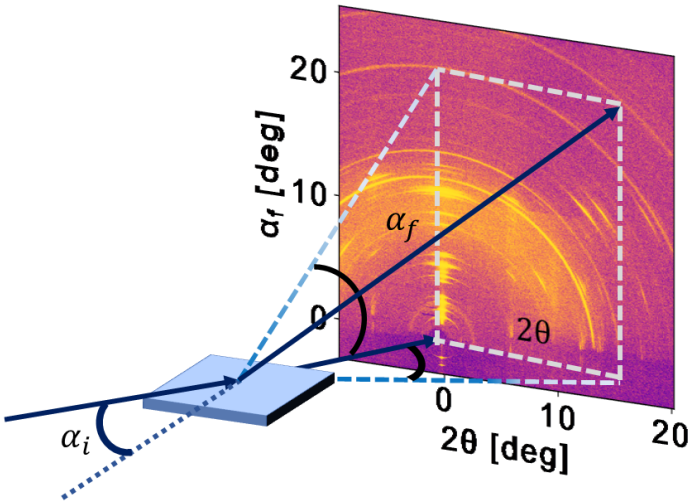


Figure 2.7: Geometry of GIWAXS measurements with grazing incidence angle $\alpha_i$. Measured scattering events are characterized by in-plane angle $2\theta$ and out-of-plane angle $\alpha_f$. The corresponding momentum transfer $q$ can be calculated via Eq. 2.37. The figure is adapted from Ref. [96].

scattering (GISAS) (grazing-incidence small-angle X-ray scattering (GISAXS) and grazing-incidence small-angle neutron scattering (GISANS)), which is widely used for studying nanoscaled surface structures and structures in thin films. In this work, we focus on wide-angle geometry employed for studying polycrystalline materials on a surface. As a model system, we choose organic-inorganic perovskite materials, being the key components of the future-generation solar cells [73, 116]. A detailed understanding of crystallization pathways of perovskites during their synthesis is essential for optimizing their structural and optoelectronic properties [75, 77–79].

As follows from Eq. 2.32, the penetration depth can be controlled by changing the incidence angle relative to the critical angle of the studied sample. To investigate the distribution of the crystalline materials along the $z$ axis, one can perform several experiments at different incidence angles, which is typically done for *ex situ* measurements, while *in situ* measurements are commonly carried out at constant incidence angles to preserve high time resolution. While Fig. 2.4 shows a general penetration depth dependency on incidence angle, Fig. 2.8 demonstrates absolute values for an

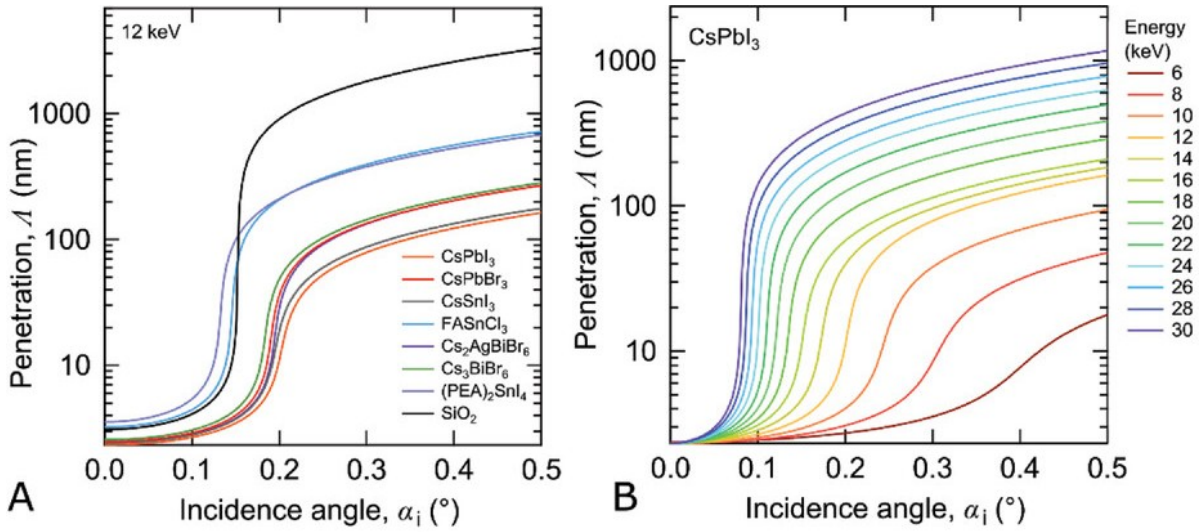


Figure 2.8: (A) X-ray (12 keV) penetration depth Λ for different perovskite compounds as a function of the grazing incidence angle. (B) Penetration depth as a function of energy for CsPbI3 perovskite. Figure is adapted from Ref. [117].

array of commonly studied perovskite compounds [117] with typical critical angles $\alpha_c \approx 0.1°$.

In GIWAXS geometry, each detected scattering event corresponds to a momentum transfer

$$q = \begin{pmatrix} q_x \\ q_y \\ q_z \end{pmatrix} = k \begin{pmatrix} \cos(\alpha_f)\cos(2\theta) - \cos(\alpha_i) \\ \cos(\alpha_f)\cos(2\theta) \\ \sin(\alpha_i) + \sin(\alpha_f) \end{pmatrix}. \tag{2.37}$$

Typically, the measured scattering pattern is converted to $\{q_{xy}, q_z\}$ coordinates via interpolation algorithms, where $q_{xy} = \sqrt{q_x^2 + q_y^2}$. Fig. 2.9a shows an example of measured GIWAXS data in reciprocal space. The dark "blindspot" along $q_z$ axis is called a *missing wedge*. Due to the fixed incidence angle, only two points on a detector measure momentum transfer along $q_z$ axis with $q_{xy} = 0$: the direct beam and the specular reflection. Consequently, Bragg peaks at $q_{xy} = 0$ simulated in Fig. 2.9b do not appear in GIWAXS data and require specular geometry to observe, i.e. reflectivity technique. In this way, the two techniques provide complementary information. Combined, they enable efficient algorithms for crystal structure identification [119].

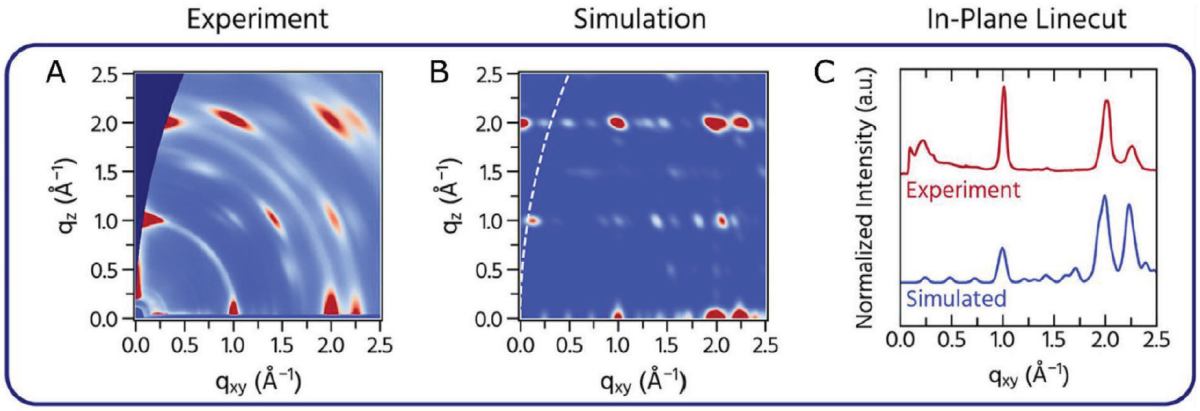


Figure 2.9: A) Measured GIWAXS from a thin film of $(BA)_2(MA)_2Pb_3I_{10}$ Ruddlesden–Popper 2D perovskite. B) Simulated GIWAXS pattern with peak broadening to agree with experimental GIWAXS results. C) Comparison of linecuts of both the experimental and simulated intensities along the $q_{xy}$ axis. Figure is adapted from Ref. [118].

Subsec. 2.1.3 discusses diffraction from a single crystal resulting in Bragg peaks emerging at discrete $q$ values. In the case of polycrystalline materials made of a large number of grains, the diffraction pattern depends on the distributions of such properties as grain orientation, size, etc. Each grain produces Bragg peaks with absolute momentum transfer values $|q_{hkl}|$ being independent of grain orientation and defined by the lattice parameters (see Eq. 2.11 and Eq. 2.19). However, grain orientation determine the azimuthal positions $q_\phi = \text{argtan}(\frac{q_z}{q_{xy}})$. Consequently, the orientation distribution of grains leads to diffraction features being prolonged along the azimuthal direction, as illustrated in Fig. 2.10. Therefore, analyzing the intensity distribution along the azimuthal direction can shed light on the orientation distribution of grains.

The broadening $B_{hkl}$ of a diffraction peak $(hkl)$ in radial direction $|q|$ is related to the average grain length $D_{hkl}$ via the Scherrer equation [120]:

$$D_{hkl} = \frac{K\lambda}{B_{\text{hkl}} \cos\theta_{\text{hkl}}} \ , \tag{2.38}$$

where $\theta_{\text{hkl}}$ is the Bragg angle and $K$ is the shape factor typically set to $K = 0.9$. In GIWAXS geometry, Eq. 2.38 should be corrected for the angular resolution $B_{res}$ of

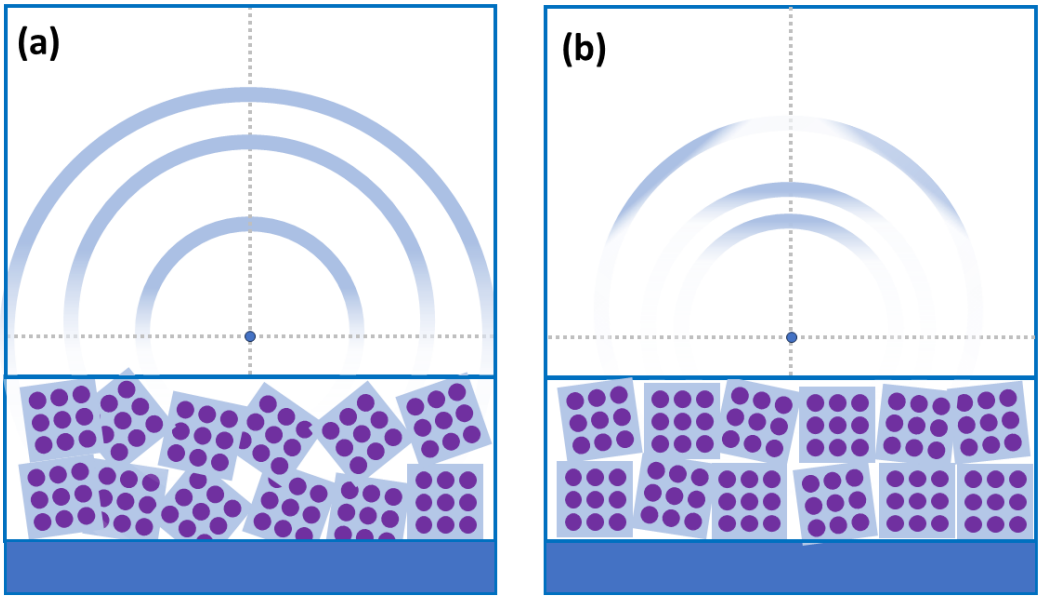


Figure 2.10: Azimuthal length of diffraction features depends on the orientation distribution of crystalline grains. A wide distribution of random orientations leads to Debye-Scherrer rings (a), while more coherently oriented grains result in narrow ring segments (b).

the measurements that depends on the beam divergence, the energy bandwidth, and the experimental geometry [117].

The peak intensities carry essential information about the crystal structures and fractions of coexisting materials. Simulating the intensities to match the experimental values is generally challenging (see Fig. 7.4) and requires applying multiple corrections to the absolute squared form factor Eq. 2.13, such as Lorenz-polarization and absorption corrections [121]. These corrections are necessary for the precise determination of atomic positions in the unit cell. The perovskite experiments typically study the relative intensities between peaks produced by co-existing materials to track crystallization processes in *in situ* experiments [122].

# 3 Conventional numerical methods

This chapter briefly summarizes conventional numerical methods in the context of reflectometry analysis discussed in Subsec. 2.2.2. There are two main approaches used to formulate reflectometry analysis [92].

The first approach is based on Maximum Likelihood Estimation (MLE), or "Parratt fit" implemented in various reflectometry packages [81–83]. Sec. 3.1 discusses traditional approaches to MLE, and Chap. 5 introduces machine learning (ML)-based techniques to improve MLE for reflectometry analysis.

The second approach relies on Bayesian analysis that is becoming increasingly more popular in reflectometry analysis [83, 92]. The traditional Bayesian tools are discussed in Sec. 3.2, while Chap. 6 introduces ML-based methods for accelerating Bayesian reflectometry analysis.

## 3.1 Maximum likelihood estimation

### 3.1.1 Poisson likelihood

The likelihood function is a fundamental concept in statistics, used to quantify the probability of observed data given specific parameters within a statistical model. In the context of reflectometry, it can be defined as a conditional probability $p(\boldsymbol{R} \mid \boldsymbol{\theta})$ of observing the measured curve $\boldsymbol{R}$ given the parameters of the sample (SLD) $\boldsymbol{\theta}$. Consequently, likelihood depends on the noise distribution of the measured data. For a set of measured points $\boldsymbol{R} = \{R_i\}_{i=1}^{n_q}$, $R_i \equiv R(q_i)$ and given parameters $\boldsymbol{\theta}$, the likelihood $p(\boldsymbol{R} \mid \boldsymbol{\theta})$ has a form:

$$p(\boldsymbol{R} \mid \boldsymbol{\theta}) = \prod_{i=1}^{N} p(R_i \mid \boldsymbol{\theta}), \tag{3.1}$$

where $N$ is a number of measured values $R_i$, and $p(R_i \mid \boldsymbol{\theta})$ are corresponding probability densities. A common approach to simplify Eq. 3.1 is to calculate log-likelihood:

$$\log\left(p(\boldsymbol{R} \mid \boldsymbol{\theta})\right) = \sum_{i=1}^{N} \log\left(p(R_i \mid \boldsymbol{\theta})\right) \tag{3.2}$$

that shares all the extrema with the initial likelihood since the logarithm is a strictly monotonic function.

Indeed, the choice of the probability density $p(R_i \mid \boldsymbol{\theta})$ and the underlying statistical model shall be based on the physical understanding of the scattering process. For both X-ray and neutron reflectivity, the measured reflected number of photons $n_{R_i} = n_0(q_i) \cdot R_i$ is well-described by counting statistics, i.e. Poisson distribution:

$$p(n_r(q_i) \mid \theta) = \mathrm{Pois}(\lambda_i(\boldsymbol{\theta}); n_r(q_i)) = \frac{e^{-\lambda_i} \lambda_i^{n_r(q_i)}}{n_r(q_i)!}, \tag{3.3}$$

where $\lambda_i(\boldsymbol{\theta}) = n_0(q_i) \cdot R(q_i; \boldsymbol{\theta})$ is the *expected* number of reflected counts given parameters $\boldsymbol{\theta}$. For any set of parameters, $\lambda_i(\boldsymbol{\theta})$ and, consequently, likelihood $p(n_r(q_i) \mid \boldsymbol{\theta})$ from Eq. 3.1 can be calculated via Parratt or Abelès formalisms (see Subsec. 2.2.2).

The corresponding log-likelihood for a whole curve with $N$ measured points depends on both the reflected intensity and the total number of incoming photons $n_0(q_i)$ and has the following form:

$$\log\left(p(n_0(q_i), \boldsymbol{R} \mid \boldsymbol{\theta})\right) \propto \sum_{i=1}^{n_q} n_0(q_i)\Big(R_i \log(R(q; \boldsymbol{\theta})) - R(q; \boldsymbol{\theta})\Big)\ , \tag{3.4}$$

where the constant normalization term that does not depend on $\theta$ is omitted.

In practice, Poisson distribution is commonly approximated by a normal distribution, and the corresponding likelihood is used by default in the standard packages [83]. This approximation can be justified by the fact that the Poisson distribution quickly converges to the normal distribution when the number of counts increases. However, for a smaller number of counts, the use of normal distribution may lead

to biased estimations. We do not see any practical benefits from switching to the normal distribution in any case, so in the following, we always employ the likelihood derived for the Poisson distribution (Eq. 3.4).

### 3.1.2 Optimization problem

MLE is a standard statistical framework that is widely employed for reflectometry analysis. Given a measured reflectivity curve $R_{\text{exp}}(q)$, MLE aims at obtaining the parameter $\theta^*$ that maximizes likelihood:

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}\in\Theta}{\operatorname{argmax}}\, p(\boldsymbol{R}_{\text{exp}}|\boldsymbol{\theta}) \ , \tag{3.5}$$

where $\Theta$ is the set of all the physically possible parameters of the sample.

The resulting solution corresponds to the "best fit" of the measured data, i.e., it provides a single solution corresponding to the most probable SLD profile of the studied sample. Due to the phase problem, multiple solutions may correspond to a high likelihood. In practice, one can significantly reduce the parameter space of the maximization problem by using prior knowledge about the studied sample. By setting some narrow parameter ranges $\Theta_{\text{sample}}$ based on a physical understanding of the sample parameters one can *assume* that there is a single solution within the defined ranges, which can be obtained by solving the following optimization problem:

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}\in\Theta_{\text{sample}}}{\operatorname{argmax}}\, p(\boldsymbol{R}_{\text{exp}}|\boldsymbol{\theta}) \ . \tag{3.6}$$

Several algorithms used to obtain $\boldsymbol{\theta}^*$ are discussed below.

### 3.1.3 Gradient descent

Gradient descent is a traditional first-order iterative local optimization algorithm already known at least in the 19$^{\text{th}}$ century [123]. It requires the calculation of partial derivatives of the optimization function w.r.t. all the optimization parameters at each iteration.

Gradient descent (or, inversely, gradient ascent) can be employed to find an extremum of a convex differentiable scalar-valued function $\mathcal{L}(\boldsymbol{\theta})$, $\mathcal{L}$: $\mathbb{R}^d \to \mathbb{R}$ w.r.t parameters $\boldsymbol{\theta} \in \mathbb{R}^d$. The necessary (but not sufficient) condition for an extremum is:

$$\nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}) = 0, \tag{3.7}$$

where

$$\nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}) \equiv \begin{bmatrix} \frac{\partial \mathcal{L}}{\partial \theta_1}(\boldsymbol{\theta}) \\ \vdots \\ \frac{\partial \mathcal{L}}{\partial \theta_d}(\boldsymbol{\theta}) \end{bmatrix} \tag{3.8}$$

is the gradient of the function $\mathcal{L}(\boldsymbol{\theta})$.

The parameters initialized by $\boldsymbol{\theta}_0$ are then updated iteratively based on the calculated gradient:

$$\boldsymbol{\theta}_i = \boldsymbol{\theta}_{i-1} - \nu \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}) \tag{3.9}$$

until Eq. 3.7 is satisfied. $\nu$ is the parameter defining the size of each step.

In the case of reflectometry, gradient descent (or, more correctly in this case, gradient ascent) can be employed to find the likelihood maximum ($\mathcal{L}(\boldsymbol{\theta}) = p(\boldsymbol{R}_{\text{exp}}|\boldsymbol{\theta})$). Given that $R(q, \boldsymbol{\theta})$ is differentiable w.r.t. $\boldsymbol{\theta}$, the necessary extremum conditions for MLE lead to the following equations obtained from Eq. 3.4:

$$\frac{\partial p(n_0(q_i), R_i \mid \boldsymbol{\theta})}{\partial \theta_j} = 0 \Rightarrow \sum_{i=1}^{N} \left[ \frac{\partial R(q_i; \boldsymbol{\theta})}{\partial \theta_j} \cdot n_0(q_i) \cdot \frac{(R(q_i; \boldsymbol{\theta}) - R_i)}{R(q_i; \boldsymbol{\theta})} \right] = 0 \ . \tag{3.10}$$

However, in the case of reflectometry, the likelihood can feature multiple extrema, i.e. multiple parameters $\boldsymbol{\theta}$ satisfy Eq. 3.10. Fig. 3.1 illustrates it for a simple simulated scenario in which a scalar parameter $\theta$ represents the thickness of the studied sample. Initialized by $\theta_0$, the algorithm converges to the closest local maximum.

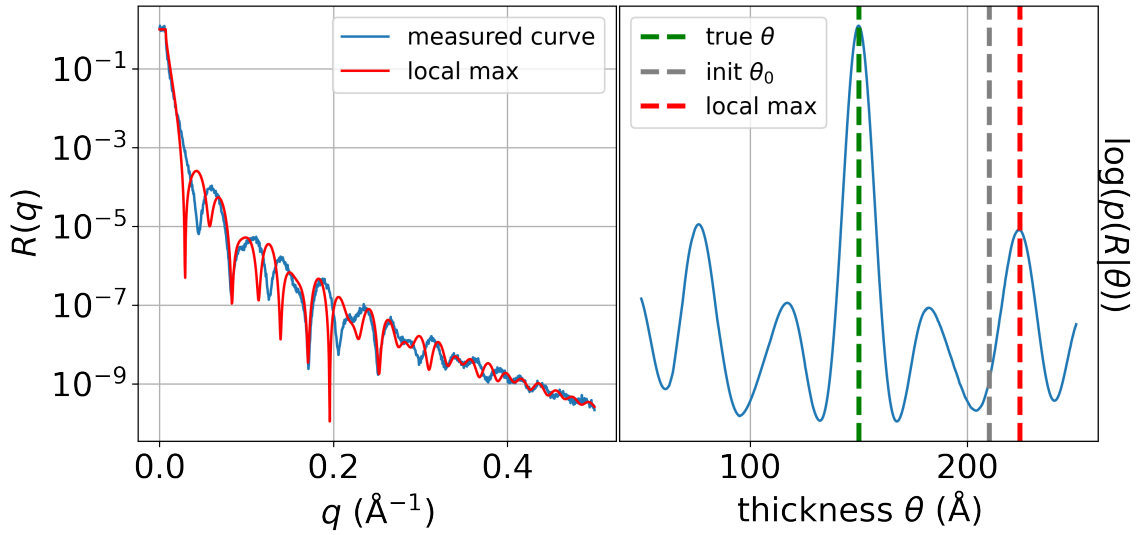


Figure 3.1: An illustration of the simulated reflectivity curve (blue curve on the left-hand side) fitted via gradient descent (red curve on the left-hand side). All parameters are fixed except for the thickness of the top layer in a simulated two-layer structure, resulting in a one-dimensional likelihood (blue curve on the right-hand side). Due to the presence of multiple extrema in the reflectometry likelihood, gradient descent (or, in this case, ascent) may converge to an incorrect solution if the initial parameters are sufficiently distant from the global extremum.

Formally, the algorithm fails since the optimized function is not convex as required for obtaining the global extremum.

Therefore, gradient descent is only used for refining ("polishing") the obtained parameters, while the initial search is performed via random exploration of the parameter space $\Theta_{\text{sample}}$ via corresponding *global* optimization methods. De facto the standard method employed in various reflectometry packages is the differential evolution (DE) algorithm discussed below.

### 3.1.4 Differential evolution

DE is a population-based, stochastic optimization algorithm [124] and is the standard algorithm employed by various reflectometry packages [81–83] to obtain a "Parratt fit". Inspired by the principles of evolution and natural selection, DE applies oper-

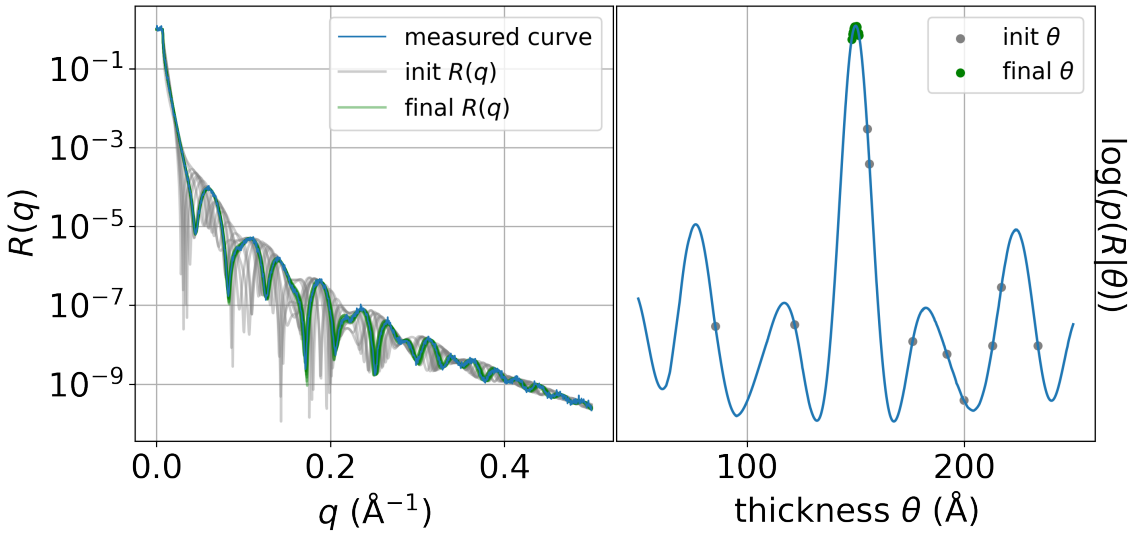


Figure 3.2: Differential evolution applied to the simulated curve from Fig. 3.1. Given the initial randomly sampled population (gray dots and curves), it converges to the global maximum (green dots and curves).

ations of mutation, crossover, and selection to a population of candidate solutions to evolve toward an optimal solution. Fig. 3.2 illustrates the result of the algorithm applied to the simulated curve introduced above (Fig. 3.1).

Indeed, there is no guarantee that the obtained parameter is not a local maximum. If the quality of the obtained fit is not good enough, experimentalists can rerun the algorithm, potentially adjusting the parameter ranges and relying on the physical understanding of the problem and their intuition, unless a good fit is obtained.

In summary, in the case of reflectometry analysis, MLE is obtained by stochastic global optimization algorithms such as DE that conduct random exploration of the parameter space $\Theta_{\text{sample}}$. The obtained solution can be refined via gradient descent optimization. Importantly, even if the correct maximum likelihood is obtained, MLE does not provide the essential information about the uncertainty of the parameters and potential secondary solutions due to the phase problem. These issues are addressed by the Bayesian analysis methods discussed below.

## 3.2 Bayesian analysis

Bayesian analysis is a general statistical framework, where available knowledge about parameters in a statistical model is updated with the information in observed data. According to Bayes' theorem [125], in the case of measured reflectivity $\boldsymbol{R}$, the estimated posterior distribution $p(\boldsymbol{\theta} \mid \boldsymbol{R})$ of the parameters $\boldsymbol{\theta}$ equals

$$p(\boldsymbol{\theta} \mid \boldsymbol{R}) = \frac{p(\boldsymbol{R} \mid \boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\boldsymbol{R})}, \tag{3.11}$$

where $p(\boldsymbol{R} \mid \boldsymbol{\theta})$ is the likelihood (see Section 3.1.1), $p(\boldsymbol{\theta})$ is *the prior distribution*, and $p(\boldsymbol{R}) = \int p(\boldsymbol{R} \mid \boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta}$ is the normalization term called *the evidence*.

Approaching the reflectometry analysis using a probabilistic, Bayesian framework resolves the fundamental problem related to lost information about a multitude of potential solutions when applying MLE-based methods discussed above. In the case of Bayesian analysis, multiple solutions can be revealed as multiple distributional modes in the estimated posterior distribution.

In general, a solution that approximates an unknown distribution should provide two operations: 1. calculating a (log) probability $p(\boldsymbol{\theta} \mid \boldsymbol{R})$ for any parameter $\boldsymbol{\theta}$ within the probability domain, and 2. sampling an arbitrary amount of points $\boldsymbol{\theta}_i \sim p(\boldsymbol{\theta} \mid \boldsymbol{R})$ from an estimated distribution. The second operation enables Monte Carlo estimations of integrals (given an integral is finite), such as parameter moments $\bar{\theta} \approx \frac{1}{N}\sum_i^N \theta_i$, $\bar{\theta^2} \approx \frac{1}{N}\sum_i^N \theta_i^2$, as well as estimation of other statistical properties such as confidence intervals.

In the case of the posterior distribution, the first operation requires an estimation of the unknown evidence $p(\boldsymbol{R})$ that can also be estimated via Monte Carlo using the second operation: $p(\boldsymbol{R}) \approx \sum_i^N p(\boldsymbol{R} \mid \boldsymbol{\theta}_i)p(\boldsymbol{\theta}_i)$. Therefore, the sampling operation $\boldsymbol{\theta}_i \sim p(\boldsymbol{\theta} \mid \boldsymbol{R})$ is the key to posterior estimation. Once an adequate amount of samples is obtained, even straightforward visualization techniques such as the corner plot can be sufficient for investigating the presence of multiple solutions, and Monte Carlo estimations can be obtained for providing uncertainty estimation.

Below we briefly discuss the main approaches to the sampling problem suitable for reflectometry analysis. We note that in the following we focus on the reflectometry case with *tractable likelihood*, i.e. computable likelihoods (see Subsec. 3.1.1), and do not consider *intractable likelihoods*, i.e. likelihoods that are computationally prohibitive or contain unobserved variables. Furthermore, we would like to emphasize once again that, given the properties of the reflectometry data, efficient gradient descent-based methods are not applicable in our case.

### 3.2.1 Importance sampling

Importance Sampling (IS) technique is the classical method for estimating probability distributions [126–128]. The idea behind IS is to construct an additional, *proposal* distribution $u(\boldsymbol{x})$, that is easy to sample from and that *covers* the target distribution $p(\boldsymbol{x})$. Formally, the support of the proposal distribution,

$$\text{supp}(u) = \{\boldsymbol{x} \in X : u(\boldsymbol{x}) \neq 0\} \tag{3.12}$$

i.e. its subdomain with positive probabilities, must include the support of the target distribution:

$$\text{supp}(p) \subseteq \text{supp}(u) \tag{3.13}$$

Typically, a proposal distribution belongs to a standard class of distributions (or a mixture of distributions) such as uniform and normal distributions that allow the fast inverse transform sampling procedure based on an inverse cumulative distribution function.

Below we illustrate the main idea of importance sampling by constructing the Monte Carlo estimation $\mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x})}[f(\boldsymbol{x})]$ of a function of interest $f(\boldsymbol{x})$ given the estimated integral is finite. We do so by sampling from the proposal distribution $\boldsymbol{x}_i \sim u(\boldsymbol{x})$ and computing the corresponding probabilities of the target distribution $p(\boldsymbol{x})$:

$$\mathbb{E}_p[f(\boldsymbol{x})] = \int f(\boldsymbol{x})p(\boldsymbol{x})d\boldsymbol{x} = \int f(\boldsymbol{x})\frac{p(\boldsymbol{x})}{u(\boldsymbol{x})}u(\boldsymbol{x})d\boldsymbol{x} = \mathbb{E}_q[f(\boldsymbol{x})\frac{p(\boldsymbol{x})}{u(\boldsymbol{x})}] \approx \sum_i f(\boldsymbol{x}_i)w_i,$$

where

$$w_i = \frac{p(\boldsymbol{x}_i)}{u(\boldsymbol{x}_i)} \tag{3.14}$$

are so-called importance weights. The estimation in the Eq. 3.2.1 is unbiased [128]. In the case of the posterior distribution, importance weights cannot be calculated since $p(\boldsymbol{\theta} \mid \boldsymbol{R})$ is only known up to a normalization term. Generally, in the case of an unnormalized probability $p(\boldsymbol{x}) = c_p p_0(\boldsymbol{x})$ with the unknown normalization term $c_p$, a so-called self-normalized importance sampling scheme can be employed:

$$\begin{aligned}
\int f(\boldsymbol{x})p(\boldsymbol{x})d\boldsymbol{x} &= \int \frac{f(\boldsymbol{x})p(\boldsymbol{x})}{u(\boldsymbol{x})}u(\boldsymbol{x})d\boldsymbol{x} \\
&= \frac{\int \frac{f(\boldsymbol{x})p(\boldsymbol{x})}{u(\boldsymbol{x})}u(\boldsymbol{x})d\boldsymbol{x}}{\int \frac{p(\boldsymbol{x})}{u(\boldsymbol{x})}u(\boldsymbol{x})d\boldsymbol{x}} \\
&= \frac{\int \frac{f(\boldsymbol{x})p_0(\boldsymbol{x})}{u(\boldsymbol{x})}u(\boldsymbol{x})d\boldsymbol{x}}{\int \frac{p_0(\boldsymbol{x})}{u(\boldsymbol{x})}u(\boldsymbol{x})d\boldsymbol{x}} \\
&= \frac{\int f(\boldsymbol{x})w(\boldsymbol{x})u(\boldsymbol{x})d\boldsymbol{x}}{\int w(\boldsymbol{x})u(\boldsymbol{x})d\boldsymbol{x}} \\
&= \frac{E_q[f(\boldsymbol{x})w(\boldsymbol{x})]}{E_q[w(\boldsymbol{x})]},
\end{aligned}$$

where the unnormalized importance weights

$$w(\boldsymbol{x}) = \frac{p_0(\boldsymbol{x})}{u(\boldsymbol{x})} \tag{3.15}$$

can be computed. We note that if the prior distribution is used as a proposal distribution for estimating the posterior, the importance weights simply equal the likelihoods:

$$w_i = p(\boldsymbol{R} \mid \boldsymbol{\theta}_i) \ . \tag{3.16}$$

Generally, this approach allows estimating a posterior distribution by employing various tools mentioned above including visualization of the weighted samples, estimation of confidence intervals, etc.

The efficiency of the IS technique can range from being extremely efficient to practically inapplicable, depending on the given proposal distribution. A common efficiency measure is the *the number of effective samples* $n_{\text{eff}}$:

$$n_{\text{eff}} = \frac{\left(\sum_{i=1}^{n} w_i\right)^2}{\sum_{i=1}^{n} w_i^2} \tag{3.17}$$

and the corresponding overall *sample efficiency* $\epsilon$:

$$\epsilon = \frac{n_{\text{eff}}}{n_{\text{total}}} = \frac{\left(\overline{w_i}\right)^2}{\overline{\left(w_i^2\right)}} \tag{3.18}$$

Sample efficiency depends on the proposal distribution, making it essential to construct an efficient proposal distribution for this method. In the case of Bayesian analysis, the natural choice for the proposal distribution is the prior distribution, which incorporates prior knowledge about the estimated distribution. Therefore, the availability of more prior knowledge leads to higher efficiency in the method. However, as the number of estimated parameters increases, the efficiency of the method rapidly diminishes.

We note that despite IS being a classical algorithm that reliably resolves all the distributional modes, it is not widely used in reflectometry analysis. In the following, we briefly discuss the more popular method.

### 3.2.2 Markov Chain Monte Carlo

The commonly employed approach to Bayesian reflectometry analysis involves the use of Markov Chain Monte Carlo (MCMC) methods [83, 92]. MCMC methods [129–132] comprise a class of algorithms that enable sampling from complex and high-dimensional probability distributions. As a result, these algorithms have significantly transformed the field of Bayesian inference and affected various scientific disciplines.

The foundation of MCMC methods relies on the theory of Markov chains. A Markov chain is a sequence of random variables $(\boldsymbol{x}_0, \ldots, \boldsymbol{x}_i)$, where the distribution of each variable depends only on the value of the previous variable, a property known as the Markov property:

$$p(\boldsymbol{x}_i \mid \boldsymbol{x}_{i-1}, \ldots, \boldsymbol{x}_0) = T(\boldsymbol{x}_i \mid \boldsymbol{x}_{i-1}), \tag{3.19}$$

where $T(\boldsymbol{x}_i \mid \boldsymbol{x}_{i-1})$ is called a *transition kernel*. In the context of MCMC, the Markov chain is constructed in such a way that its *stationary distribution* is the same as the target distribution (e.g. the posterior distribution in Bayesian analysis). By running the Markov chain for a sufficiently long time, the samples generated from the chain will approximate the target distribution. Essentially, one initializes an ensemble of "walkers" that move stochastically according to a certain rule, and each "move" produces a sample $\boldsymbol{x}$. Once the algorithm has converged, it can continue running in the equilibrium state producing Markov chains of (dependent) samples that by the design of the algorithm correspond to the *stationary distribution*, which is set to be equal to the estimated posterior distribution.

The key to constructing such a Markov chain is the detailed balance equation, which is a sufficient (but not necessary) condition that ensures the stationary distribution of the chain is the target distribution $p(\boldsymbol{x})$. The detailed balance equation describes *reversible* Markov chains with respect to $p(\boldsymbol{x})$:

$$p(\boldsymbol{x}_i)T(\boldsymbol{x}_i \mid \boldsymbol{x}_{i-1}) = p(\boldsymbol{x}_{i-1})T(\boldsymbol{x}_{i-1} \mid \boldsymbol{x}_i) \tag{3.20}$$

This condition ensures that $p(\boldsymbol{x})$ is the stationary distribution, i.e. when the algorithm has converged and the sample distribution follows $p(\boldsymbol{x})$, it remains stationary since the average number of samples moving from $\boldsymbol{x}_{i-1}$ to $\boldsymbol{x}_i$ is the same as the "flow" in the opposite direction.

Running MCMC requires computing probabilities of the target distribution $p(\boldsymbol{x})$ at different stochastically proposed points $\boldsymbol{x}$. In the case of Bayesian analysis, the posterior distribution $p(\boldsymbol{\theta} \mid \boldsymbol{R})$ cannot be computed due to the unknown normaliza-

tion term. Typically, MCMC algorithms are constructed in a way that only requires the computation of the probability ratio between two points $\frac{p(\boldsymbol{\theta}_i|\boldsymbol{R})}{p(\boldsymbol{\theta}_{i-1}|\boldsymbol{R})}$, which does not include the normalization term and only requires computing likelihoods and priors at the corresponding points. Reflectometry analysis relies on Affine Invariant MCMC algorithms [133, 134].

MCMC methods are indeed not without their challenges. The quality of the samples generated by the Markov chain depends on the chain reaching its stationary distribution, a state known as convergence. However, determining when convergence has occurred is not straightforward and is a topic of ongoing research. Furthermore, the convergence may take an exceedingly long time period depending on a given distribution $p(\boldsymbol{x})$, the employed algorithm, and, importantly, the initialization of the ensemble of "walkers". Apart from influencing convergence rate, initialization of MCMC is crucial for another reason - bad initialization can lead to missed distributional modes. In theory, even if MCMC is initialized around one peak, there is a non-zero probability of exploring other isolated distant peaks. However, in practice, such probability can be negligibly low, leading to the situation when the algorithm "converges" to an isolated part of the distribution and fails to reveal other distributional modes. In reflectometry, this issue is especially relevant as the posterior distribution often features multiple narrow peaks.

The standard reflectivity packages such as refnx [83] rely on the DE algorithm to provide good initialization for MCMC. In this manner, DE first provides maximum likelihood estimation $\boldsymbol{\theta}^*$. The population of initial parameters is then initialized by sampling from a "ball" with some predefined size around the found maximum $\boldsymbol{\theta}^*$. This approach helps MCMC to rapidly converge and resolve the peak found by DE. However, as mentioned above, other potential distributional modes can be missed. These issues are further discussed in Chap. 6.

# 4 Deep learning methods

This chapter provides the basic principles of deep learning and introduces the deep learning algorithms employed in this dissertation.

## 4.1 Learning from data

Deep learning is a subfield of machine learning (ML) that employs deep artificial neural networks (NNs) to "learn" from data. These NNs mirror the operational principles of biological NNs found in living organisms, to some extent emulating the organic learning process in a computational context [135]. Typically, a NN is a function $f_{\boldsymbol{w}}(\mathbf{x})$ that is "trained" to approximate some unknown functional dependency $\mathbf{y} = f(\mathbf{x})$ based on the given data samples $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^N$. The training process involves optimizing the parameters $\boldsymbol{w}$ of the model. In contrast to traditional statistical models, which are designed to be well-specified to unveil functional relationships between model parameters, deep learning models are often constructed as over-parameterized "black boxes". The primary practical focus for such models lies in providing accurate predictions $\mathbf{y}$ based on input $\mathbf{x}$. This approach is most suitable when it is practically challenging to model the considered functional dependency analytically, but considerable amount $N$ of the corresponding data samples $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^N$ is available. The fundamental strength of deep learning is its capacity for feature learning, models being capable of automatically identifying relevant features from raw input data, thus removing the necessity for manual feature extraction.

## 4.2 Deep learning applications

Deep learning has demonstrated noteworthy success in various domains due to its ability to process high-dimensional and unstructured data. Key areas of impact include image [136, 137] and speech [138, 139] recognition, natural language processing [140–142], autonomous vehicles [143], and many other domains which handle large volumes of complex data.

Furthermore, deep learning has become an essential tool in various scientific disciplines, influencing the methodology and pace of discovery. Its applications range from the recognition of complex patterns in vast and diverse datasets to the prediction of scientific phenomena. In bioinformatics, deep learning accelerates drug discovery and genomic analysis [15–17]. In the field of astronomy, deep learning techniques are deployed to classify galaxies and detect exoplanets [18, 19]. In climate science, it helps model and predict environmental changes [20].

Physics has been strongly impacted by deep learning techniques. They are employed in the identification of new particles [21, 22], the analysis of complex systems [23–26], the data-driven material science [27–32]. In scattering physics, deep learning methods are employed to determine crystal structures based on X-ray diffraction data [33–40, 42], estimate particle shapes from small-angle scattering [43–46] and grazing-incidence small-angle scattering measurements [47–49], and accelerate reflectometry analysis [84–89, 91].

In the following, we introduce the basic concepts of deep learning with a focus on feedforward neural networks [144].

## 4.3 Feedforward neural networks

### 4.3.1 General concept

In this section, we consider feedforward neural networks $f_{\boldsymbol{w}}(\boldsymbol{x}) = f(\boldsymbol{x}, \boldsymbol{w})$ (both notations are used interchangeably) with parameters $\boldsymbol{w}$ that are used to approximate an unknown target function $\boldsymbol{y} = f(\boldsymbol{x})$ based on given data samples $\{\boldsymbol{x}_i, \boldsymbol{y}_i\}_{i=1}^{N}$.

Feedforward networks are designed as layered structures where the input data $\boldsymbol{x}$ is consecutively transformed by the corresponding *layers* - parameterized functions $f_l(\boldsymbol{x}_l, \boldsymbol{w}_l)$ - such that the network can be represented as a function composition of the layers:

$$f(\boldsymbol{x}, \boldsymbol{w}) \equiv f_l \circ f_{l-1} \circ \cdots \circ f_1(\boldsymbol{x}) = f_l(f_{l-1}(\cdots(f_1(\boldsymbol{x}))\cdots)) \ , \tag{4.1}$$

where the layer parameters $\boldsymbol{w}_l$ are omitted for clear notation. In this notation, $f_1$ is the *first layer*, $f_2$ is the *second layer*, and so on. The number of layers is usually referred to as the *depth* of the network. The choice of layers $f_l$ defines a so-called *network architecture* and is generally determined by the different aspects of the data, such as dimensionality, intrinsic symmetries, statistical properties, etc.

To approximate the target function $\boldsymbol{y} = f(\boldsymbol{x})$ by the NN $\hat{\boldsymbol{y}} = f(\boldsymbol{x}, \boldsymbol{w})$, the parameters $\boldsymbol{w}$ have to be adjusted (see Subsec. 4.3.3). The Universal Approximation Theorem [145–147] establishes that a feedforward network with a single hidden layer containing a finite number of neurons and an appropriate activation function can approximate any continuous function on compact subsets of real numbers. This essentially means that NNs has the theoretical capability to model any input-output relationship to a high degree of precision, no matter how complex. However, the theorem does not provide guidance on how to set the parameters $\boldsymbol{w}$ to achieve this approximation. Formally, it requires solving an optimization problem with some distance metric $\mathcal{L}(\hat{\boldsymbol{y}}, \boldsymbol{y}; \boldsymbol{w}) \equiv \mathcal{L}(f(\boldsymbol{x}, \boldsymbol{w}), \boldsymbol{y})$ between the model output and the target output. Given a dataset $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^N$, one can calculate the average of this metric across the dataset:

$$J(\boldsymbol{w}) = \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(f(\boldsymbol{x}_i, \boldsymbol{w}), \boldsymbol{y}_i) \tag{4.2}$$

and minimize the function $J(\boldsymbol{w})$ with respect to the network parameters $\boldsymbol{w}$:

$$\boldsymbol{w}^* = \arg\min_{\boldsymbol{w}} J(\boldsymbol{w}) = \arg\min_{\boldsymbol{w}} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(f(\boldsymbol{x}_i, \boldsymbol{w}), \boldsymbol{y}_i), \tag{4.3}$$

where $N$ is the number of samples.

### 4.3.2 Loss functions

Usually, the function $J(\boldsymbol{w})$ calculated for a given dataset is called the *cost function*, and the function $\mathcal{L}(\boldsymbol{y}, \hat{\boldsymbol{y}})$ is called the *loss function*, although these terms are sometimes used interchangeably. A large variety of ML problems such as regression, classification, probability estimation, or even unsupervised learning tasks can be expressed in the same terms of optimizing a certain cost function with respect to the parameters of the deep learning model. Indeed, the choice of a loss function depends on the considered task.

In the case of a regression task, where the model is trained to approximate an unknown function $f(\boldsymbol{x})$ by minimizing the distance between the model outputs $\hat{\boldsymbol{y}}_i = f_{\boldsymbol{w}}(\boldsymbol{x})$ and the data samples $\boldsymbol{y}_i$, a typical loss function employed is mean squared error (MSE):

$$\mathcal{L}_{\text{MSE}}(\boldsymbol{y}, \hat{\boldsymbol{y}}) = ||\boldsymbol{y} - \hat{\boldsymbol{y}}||_2^2 = \sum_{j=1}^{d} (y_j - \hat{y}_j)^2 \ , \tag{4.4}$$

$$J_{\text{MSE}} = \frac{1}{N} \sum_{i=1}^{N} ||\boldsymbol{y}_i - \hat{\boldsymbol{y}}_i||_2^2 \ , \tag{4.5}$$

where $\boldsymbol{y}_i, \hat{\boldsymbol{y}}_i \in \mathbb{R}^d$. MSE loss naturally comes from an assumption that the training samples follow a normal distribution with some identical variance $\sigma^2$:

$$\boldsymbol{y}_i \sim p(\boldsymbol{y} \mid \boldsymbol{x}_i) = \mathcal{N}(f(\boldsymbol{x}_i), \mathbb{1} \cdot \sigma^2) \ . \tag{4.6}$$

If this assumption is satisfied, minimizing MSE loss is equivalent to Maximum Likelihood Estimation (MLE) introduced in Sec. 3.1.

Another typical ML task involves the classification of the input data $\boldsymbol{x}$, where the model estimates the probability $p(y = m \mid \boldsymbol{x})$ of $\boldsymbol{x}$ belonging to a class $m$ based on given labeled data $\{\boldsymbol{x}_i, y_i\}$, $y_i \in \{1, \ldots, C\}$. The commonly used cost function

is the cross-entropy $H(p, p_{\boldsymbol{w}})$ between the data distribution $p(y \mid \boldsymbol{x}_i)$ and the model distribution $p_{\boldsymbol{w}}(y \mid \boldsymbol{x}_i)$:

$$H(p, p_{\boldsymbol{w}}) \equiv -\mathbb{E}_p \left[\log\left(p_{\boldsymbol{w}}\right)\right] \approx -\frac{1}{N}\sum_{i=1}^{N}\sum_{m=1}^{C}[y_i = m]\log(p_{\boldsymbol{w}}(y = m \mid \boldsymbol{x}_i)). \tag{4.7}$$

In this work, we use binary classification ($C = 2$) to classify image patches as either containing certain diffraction features ($y_i = 1$) or not ($y_i = 0$) (see Sec. 4.5 and Chap. 7). For binary classification, the cost function can be simplified as follows:

$$H(p, p_{\boldsymbol{w}}) \approx -\frac{1}{N}\sum_{i=1}^{N}\Big(y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)\Big), \tag{4.8}$$

where $y_i \in \{0, 1\}$ is the "true" label of data sample $\boldsymbol{x}_i$, and $\hat{y}_i$ is the estimated probability that $\boldsymbol{x}_i$ has a label $y_i = 1$: $\hat{y}_i = p_{\boldsymbol{w}}(y = 1 \mid \boldsymbol{x}_i) = 1 - p_{\boldsymbol{w}}(y = 0 \mid \boldsymbol{x}_i)$, $\hat{y}_i \in (0, 1)$.

Another cost function relevant to this work is the relative entropy, or Kullback–Leibler (KL) divergence [148]:

$$D_{\mathrm{KL}}(p \| p_{\boldsymbol{w}}) \equiv \mathbb{E}_p \left[\log\left(\frac{p}{p_{\boldsymbol{w}}}\right)\right] = H(p, p_{\boldsymbol{w}}) - H(p)\ , \tag{4.9}$$

where $H(p) \equiv -\mathbb{E}_p \left[\log\left(p\right)\right]$ is Shannon's entropy [149], which quantifies the expected value of the information contained in a sample (or "message", as this fundamental concept is initially derived for a theory of communication) generated from a distribution $p$. While it is similar to thermodynamic entropy in that it measures uncertainty or disorder, it is important to not conflate these concepts. Both the KL divergence $D_{\mathrm{KL}}(p \| p_{\boldsymbol{w}})$ and the cross-entropy $H(p, p_{\boldsymbol{w}})$ indicate discrepancies between distributions $p$ and $p_{\boldsymbol{w}}$. In the context of training ML models, minimizing these losses is equivalent:

$$\boldsymbol{w}^* = \arg\min_{\boldsymbol{w}} D_{\mathrm{KL}}(p \| p_{\boldsymbol{w}}) = \arg\min_{\boldsymbol{w}} \left(H(p, p_{\boldsymbol{w}}) - H(p)\right) = \arg\min_{\boldsymbol{w}} H(p, p_{\boldsymbol{w}}). \tag{4.10}$$

We note that these losses are not symmetric, e.g. $D_{\mathrm{KL}}(p\|p_{\boldsymbol{w}}) \neq D_{\mathrm{KL}}(p_{\boldsymbol{w}}\|p)$. In this work, we use the forward KL divergence $D_{\mathrm{KL}}(p\|p_{\boldsymbol{w}})$ (see Sec. 4.6 and Chap. 6).

### 4.3.3 Training feedforward neural networks

The gradient descent algorithm (see Subsec. 3.1.3) was long considered inapplicable to the training of NNs due to the non-convex nature of the loss function being optimized. Moreover, the number of parameters in modern NNs typically ranges from millions to billions. Therefore, obtaining the optimal model parameters $\boldsymbol{w}^*$ in Eq. 4.3 requires finding a global minimum in an extremely high-dimensional space. The complex, non-convex landscape of loss functions, characterized by multiple local minima and saddle points, was believed to be an insurmountable challenge for the method. However, this perspective shifted substantially following a key discovery that despite the non-convexity of the loss function, gradient descent could indeed be effectively applied to NNs [150, 151].

In this way, gradient descent has become the cornerstone of modern deep learning algorithms. Consequently, certain constraints are imposed on the designed architectures, necessitating the use of differentiable cost functions and layer functions in the NN architectures. Furthermore, a significant research effort is being put to improve different aspects of the method, such as architecture-dependent parameter initialization schemes [152–154] and optimization algorithms with momentum [155–158].

In the most simplified way, the training of a NN is an iterative process of updating its parameters $\boldsymbol{w}$ via

$$\boldsymbol{w} \leftarrow \boldsymbol{w} - \nu \nabla_{\boldsymbol{w}} J(\boldsymbol{w}) \ , \tag{4.11}$$

where the constant $\nu$ is called the *learning rate*. In this case, each iteration requires evaluating the loss function on all the available *training* data $\{\boldsymbol{x}_i, \boldsymbol{y}_i\}_{i=1}^N$, which might consist of millions of samples. This procedure rapidly becomes practically unfeasible as the number of training samples increases. To address this issue, the standard gradient descent is usually replaced by stochastic gradient descent (SGD) [159]. In-

stead of evaluating the cost function on the entire dataset, the dataset is randomly divided into smaller subsets, or *mini-batches*. At each iteration, the cost function is evaluated on a single mini-batch, followed by an update of the model parameters as given by Eq. 4.11. This process repeats until all mini-batches have been processed, after which the dataset is randomly divided again. A complete cycle - randomly dividing the dataset and then sequentially processing all mini-batches - is referred to as an *epoch*. This is the most common way to train NNs on large datasets. In this approach, each sample from the training dataset is used once per epoch to update the model parameters, and the complete training procedure may require hundreds to thousands of epochs until the training loss converges.

In this work, we adopt a different approach. We utilize data-generating processes as the source of our training data and generate new samples to construct a fresh batch for each training iteration. As a result, each sample is used only once throughout the entire training procedure, making the concept of epochs redundant. Consequently, the cost function $J(\boldsymbol{w})$ no longer includes a fixed set of training samples, but instead represents a Monte Carlo estimation of the loss function for each batch:

$$J(\boldsymbol{w}) = \mathbb{E}\left[\mathcal{L}(y, f_{\boldsymbol{w}}(x))\right] \approx \frac{1}{N_{\text{batch}}} \sum_{i=1}^{N_{\text{batch}}} \mathcal{L}(y_i, f_{\boldsymbol{w}}(x_i)) \ , \tag{4.12}$$

where the data samples $\{x_i, y_i\}_{i=1}^{N_{\text{batch}}}$ are produced by the data generation process for every new training iteration. This approach improves the model performance, as discussed in the next section.

The computation of the gradient (Eq. 4.11) is performed using the algorithm called back-propagation [150, 151, 160]. It is based on a nested structure of the NNs and the chain rule of calculus:

$$\frac{d}{dx}\Big(g(f(x))\Big) = \frac{dg}{df}\frac{df}{dx} \tag{4.13}$$

Deep learning models with a large number of layers result in long chains of such derivative multiplications. In practice, it may lead to *gradient explosion* in the case derivatives are larger than 1, or, more frequently, to *gradient vanishing* in the case

of small gradients. Consequently, the training process is disrupted, since the model parameters updated based on Eq. 4.11 either increase to practically infinite numbers or do not update at all. Several architectural adjustments are proposed to address this issue, some of which we discuss in Sec. 4.4.

### 4.3.4 Prediction error and generalization

The error of the model prediction $f_{\boldsymbol{w}}(x)$ depends on several factors. Considering the data generation process $y = f(x) + \epsilon$, where $\epsilon$ is noise with variance $\mathrm{Var}(\epsilon) = \sigma_\epsilon^2$ and zero mean, the prediction error can be decomposed into several terms [135]:

$$\mathbb{E}\left[(y - f_{\boldsymbol{w}}(x))^2\right] = \sigma_\epsilon^2 + \mathrm{Bias}^2 + \mathrm{Variance}\ , \tag{4.14}$$

where $\mathrm{Bias} = \mathbb{E}\left[f_{\boldsymbol{w}}(x)) - f(x))\right]$ and $\mathrm{Variance} = \mathbb{E}\left[(f_{\boldsymbol{w}}(x) - \mathbb{E}\left[f_{\boldsymbol{w}}(x)\right])^2\right]$. The term $\sigma_\epsilon^2$ is called the *irreducible error* and it reflects the influence of data quality, or "noisiness", on model performance. Bias refers to the inability of the model to capture all the complexity in the data, while variance refers to the error arising from the model's sensitivity to fluctuations in the training data. Both bias and variance are dependent on model capacity. Insufficient model capacity often leads to higher bias, resulting in a phenomenon known as *underfitting*, where the model prediction fails to reflect certain features present in the data. Higher model capacity generally reduces the bias but, if not properly managed, can increase variance due to *overfitting*.

Overfitting arises when the model performs well on the training data but fails to effectively generalize to unseen data. In essence, the model replicates the training data very closely, capturing both the true signal and the noise, but it does not interpolate well between the training points or predict accurately on new data points. There are various practical techniques to mitigate overfitting, one of which involves monitoring the model's performance on a *validation set* after each training epoch. This allows the identification of the point during training when the validation loss begins to increase, often signaling overfitting. As discussed in the previous section, in this work, the training samples are freshly generated from the data generation process

for each batch. Consequently, under mild conditions such as a sufficiently large batch size and an appropriate learning rate $\nu$, the employed training scheme effectively protects the model from overfitting, eliminating the need for validation during the training process. Nevertheless, other sources of error, including underfitting and noise inherent in the data generation process, must still be considered.

## 4.4 Building blocks of neural networks

There is a variety of "building blocks" developed to design NNs for different domains and types of data, such as graphs, sequences, audio, volumetric data, etc. Below we briefly describe the basic data transformations used in neural networks that are most relevant for this work.

### 4.4.1 Fully-connected layer

The most basic and well-studied transformation used in NN architectures is the fully-connected (FC) layer. It requires fixed dimensionality of the input $\boldsymbol{x}$ and does not rely on any symmetries in the data. FC layer is defined as an affine transformation:

$$\boldsymbol{z} = f^{FC}(\boldsymbol{x}; \boldsymbol{W}, \boldsymbol{b}) = \boldsymbol{W}^\top \boldsymbol{x} + \boldsymbol{b}$$

with input and output vectors $\boldsymbol{x} \in \mathbb{R}^n$, $\boldsymbol{z} \in \mathbb{R}^m$. The *weight* matrix $\boldsymbol{W} \in \mathbb{R}^{n \times m}$ and the *bias* vector $\boldsymbol{b} \in \mathbb{R}^m$ are the parameters of the FC layer ($\boldsymbol{w} = [\boldsymbol{W}, \boldsymbol{b}]$). It is straightforward to show that the composition of affine transformations is an affine transformation:

$$f_l^{FC} \circ f_{l-1}^{FC} \circ \cdots \circ f_1^{FC}(\boldsymbol{x}) = f_l^{FC}(f_{l-1}^{FC}(\cdots(f_1^{FC}(\boldsymbol{x})))) = \boldsymbol{W}^\top \boldsymbol{x} + \boldsymbol{b}$$

that can only approximate linear functions. To introduce non-linearity, FC layer is typically followed by a non-linear scalar function $\sigma : \mathbb{R} \to \mathbb{R}$ applied element-wise to the FC layer output:

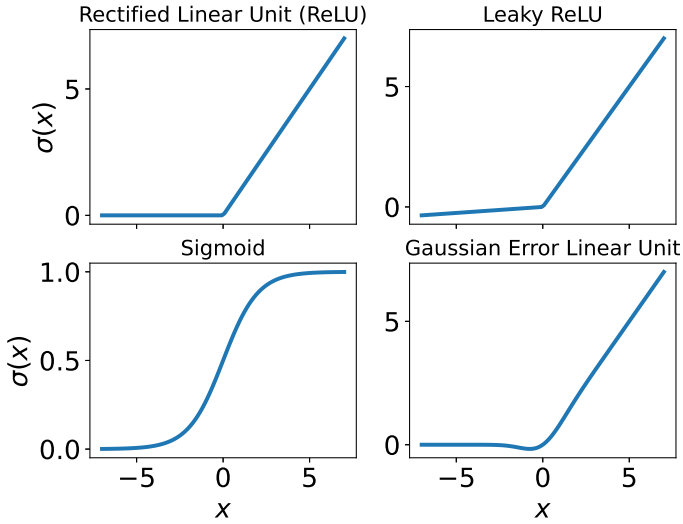


Figure 4.1: An illustration of common activation functions that are used in this thesis.

$$\sigma(\boldsymbol{z}) = \sigma(\boldsymbol{W}^\top \boldsymbol{x} + \boldsymbol{b})$$

These non-linear functions are called *activation functions* inspired by the biological terminology and are used in virtually every NN architecture to enable non-linear transformations. The composition of FC layers followed by non-linear activations forms a famous multi-layer perceptron (MLP) [161]:

$$f_{\boldsymbol{w}}^{\text{MLP}}(\boldsymbol{x}) \equiv \sigma\left(f_l^{\text{FC}}\right) \circ \sigma\left(f_{l-1}^{FC}\right) \circ \cdots \circ \sigma\left(f_{l-1}^{FC}\right)(\boldsymbol{x}) \tag{4.15}$$

Below we discuss some commonly used activation functions.

### 4.4.2 Activation functions

Nowadays, different types of activation functions are being used in NN architectures [153, 162–166]. Fig. 4.1 illustrates common activation functions that are relevant for this work and which are discussed below.

One of the most common activation functions is the rectified linear unit (ReLU) [162]:

$$\mathrm{ReLU}(x) = \max(x, 0) \tag{4.16}$$

It is a piece-wise linear function that returns zero for negative input and behaves as an identity function for non-negative input. Although it is widely employed in various applications, it has a characteristic drawback: during the training process, it may potentially result in a state when it returns zero for essentially all inputs from the training dataset (the "dead neuron" state). Consequently, the corresponding gradients become zero, contributing to the vanishing gradient problem and reducing the efficiency of the training process.

To circumvent this issue, various modifications of ReLU have been proposed, such as leaky rectified linear unit (LReLU) [163]:

$$\mathrm{LReLU}(x) = \begin{cases} x \text{ for } x > 0 \\ 0.01x \text{ for } x \leq 0 \end{cases} \tag{4.17}$$

In general, the effect of using different activation functions may vary between different architectures and datasets, so empirical study is usually required to determine the optimal choice. In this thesis, we use both ReLU and LReLU, as well as Gaussian Error Linear Unit (GELU) activation function [164] which is being widely employed in more recent NN architectures:

$$\mathrm{GELU}(x) = x\Phi(x) = x \cdot \frac{1}{2}[1 + \mathrm{erf}(x/\sqrt{2})] \tag{4.18}$$

where

$$\mathrm{erf}(x) = \frac{2}{\pi} \int_0^x e^{-t^2/2} dt \tag{4.19}$$

is the error function.

The activation function applied to the final layer defines the range of the output. For instance, in a classification task, the output typically corresponds to the estimated probability and should be constrained to the range from 0 to 1. In this

case, the commonly employed activation function is the sigmoid function in the case of binary classification [167]:

$$\text{Sigmoid}(x) = \frac{1}{1 + e^{-x}} \ , \tag{4.20}$$

or the softmax function [150] in the case of multi-class classification with $C$ classes:

$$\text{Softmax}(\boldsymbol{x}) = \frac{e^{x_i}}{\sum_i^C e^{x_i}} \ , \tag{4.21}$$

where $\boldsymbol{x} \in \mathbb{R}^C$.

### 4.4.3 Convolutional layer

Convolutional neural networks (CNNs) are widely used for image and video analysis. A convolutional layer applies a set of kernels to the input data. For 2D image analysis, each kernel is a small matrix that is supposed to detect a specific feature in the input image by performing the discrete convolution operation. The kernel is moved across the input data with a small step size, called the stride, and at each position, the dot product is taken between the kernel and a small region of the input data. Usually, the edges of the input data are extended by adding padding values to process the edges of an image and preserve the size of an image. This results in a new feature map, which is a transformed version of the input data that highlights the presence of the specific feature represented by the kernel. A typical convolutional layer consists of multiple kernels, each of which corresponds to a different feature. The elements of the kernel matrices are trainable parameters that are adjusted during the training process of the neural network.

In general, kernels of convolutional layers are tensors $w \in \mathbb{R}^{N_1 \times \cdots \times N_k}$ with dimensionality $k$ corresponding to the dimensionality of the input data. In this work, we apply 1D convolutional layers to reflectivity curves (see Chap. 5) and 2D convolutional layers to diffraction images (see Chap. 7).

By design, the convolution operation $g(\mathbf{x}) = \mathbf{w} \circledast \mathbf{x}$ has the property of translation equivariance; if an image $\mathbf{x}$ is shifted via a translation operator $T(\mathbf{x})$, the corresponding output $g(\mathbf{x})$ is shifted accordingly: $g(T(\mathbf{x})) = T(g(\mathbf{x}))$. This is the key property that enables efficient pattern recognition regardless of the absolute position of a pattern on an image. Therefore, convolutional layers are highly efficient at processing data that exhibits translational symmetry such as images (for instance, regardless of whether a cat in a photo is shifted from one side to the other, it remains a cat).

### 4.4.4 Batch normalization

There are different types of data normalization techniques that can potentially optimize the training process of NNs. In this thesis, we use the commonly employed batch normalization layer [168] that normalizes $k$-dimensional input across the training samples by calculating the mean $\mathbb{E}[\boldsymbol{x}^{(k)}]$ and the variance $\mathrm{Var}[\boldsymbol{x}^{(k)}]$ along the training set for each input dimension:

$$f^{BN}(\boldsymbol{x}^{(k)}) = \frac{\boldsymbol{x}^{(k)} - \mathbb{E}[\boldsymbol{x}^{(k)}]}{\sqrt{\mathrm{Var}[\boldsymbol{x}^{(k)}] + \epsilon}} \cdot \gamma^{(k)} + \beta^{(k)}$$

where $\gamma^{(k)}$ and $\beta^{(k)}$ are $2k$ trainable parameters and $\epsilon$ is a small positive value introduced for numerical stability. Batch normalization layers are typically placed in-between other layers in the NN to periodically normalize the data flow. In this case, the moments $\mathbb{E}[\boldsymbol{x}^{(k)}]$ and $\mathrm{Var}[\boldsymbol{x}^{(k)}]$ change over training iterations with the training parameters, so it is computationally inefficient to calculate the moments for each intermediate input to the batch normalization layer among all the training set for each training iteration. Instead, an optimized computational scheme is used that is compatible with mini-batch stochastic gradient descent and requires calculating the moments only over a mini-batch during the training iteration [168]. The batch normalization technique is reported to smooth out the landscape of the optimization problem and accelerate the training process [169].

### 4.4.5 Skip connections

Skip connections, also referred to as residual connections, are a crucial component in the architecture of deep neural networks. They are designed to alleviate the problem of the vanishing gradient discussed in Subsec. 4.3.3, a phenomenon in which gradient values progressively diminish during the training to the point where they become excessively small.

Skip connections address this issue by allowing the gradient to be directly back-propagated to earlier layers. They achieve this by creating a shortcut or bypass between two layers, enabling the output of one layer to be directly fed into a subsequent layer, in the simplest case described as follows:

$$f_l(\boldsymbol{x}_l) = f_l(f_{l-1}(\boldsymbol{x}_{l-1}) + \boldsymbol{x}_{l-1}). \tag{4.22}$$

This technique ensures that even if the gradient becomes very small in the deeper layers, the earlier layers can still be updated. By doing so, skip connections can preserve the gradient through the network, allowing for more efficient learning and performance.

## 4.5 Deep learning methods for object detection

### 4.5.1 Object detection task

Object detection is a classic task in computer vision that has been revolutionized by the advent of deep learning methods. Object detection involves identifying and locating objects of interest within an image or a video. It demands the model not only identify and classify the objects present, but also accurately predict their spatial location within the image via bounding boxes. The complexity of the task arises from the variation in object size, shape, appearance, contextual information, and location.

Unlike traditional regression discussed above, where the output is a vector with fixed dimensionality, object detection necessitates the prediction of a variable number

of objects within an image. Each object is typically represented by four coordinates and a class label. Depending on the image, the number of objects can vary, leading to a fluctuating number of bounding boxes. Since traditional neural networks are not designed to manage a varying number of outputs, innovative architectures are being developed to address this challenge.

Currently, there are multiple competing deep learning object detection techniques, many of which share similar ideas and issues. A comprehensive overview of all proposed approaches is beyond the scope of this dissertation. However, in the following sections, we describe the commonly employed metrics as well as the two common classes of anchor-based deep learning object detection architectures employed in this work. We note that nearly every emerging research paper presents modifications to the framework discussed below. Furthermore, there are multiple anchor-free approaches that we also do not cover. In the following, we focus on the approach utilized in this dissertation.

### 4.5.2 Metrics

The performance metrics for the evaluation of object detection models typically rely on the number of correct, *true positive*, predictions, the number of incorrect, *false positive*, predictions, as well as the number of missed, *false negative* objects. These numbers alone as well as their combinations provide insight into the model performance.

Precision and recall are two other fundamental metrics used in object detection. Precision is defined as the ratio of true positives to the total predicted positive observations (the sum of true positives and false positives):

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{4.23}$$

This metric essentially shows how precise the model is out of the positively predicted samples. Recall, on the other hand, is the ratio of true positives to all the present objects (the sum of true positives and false negatives):

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{4.24}$$

Recall indicates how many of the actual positives the model is able to capture.

Accuracy, another commonly used metric, provides an overall measure of how well the model correctly identifies both positives and negatives. It is calculated as the ratio of true positives divided by the sum of the number of observations and the number of missed objects:

$$\text{Accuracy} = \frac{\text{TP}}{\text{TP} + \text{FP} + \text{FN}}. \tag{4.25}$$

We note that classification tasks employ a modified Eq. 4.25 that also includes true negatives.

Indeed, the evaluation of the metrics presented above requires us to preliminarily classify each prediction as either correct or incorrect based on how well it matches a ground truth object. Therefore, we require the ground truth objects, typically obtained either by manual labeling or by simulating the image, and some matching criterion that depends on a measure of overlap between the ground truth box and the predicted box from the detection model.

Intersection over Union (IoU) is a key measure of overlap widely employed in object detection. IoU is calculated as the area of intersection divided by the area of union of the two boxes:

$$\text{IoU}(\text{Box}_A, \text{Box}_B) = \frac{|\text{Box}_A \cap \text{Box}_B|}{|\text{Box}_A \cup \text{Box}_B|}. \tag{4.26}$$

A high IoU score (closer to 1) indicates a good match with the ground truth, while a low score (closer to 0) suggests a poor match. Typically, a certain IoU threshold is set (e.g., IoU= 0.5) and any prediction with IoU above this threshold is considered a true positive.

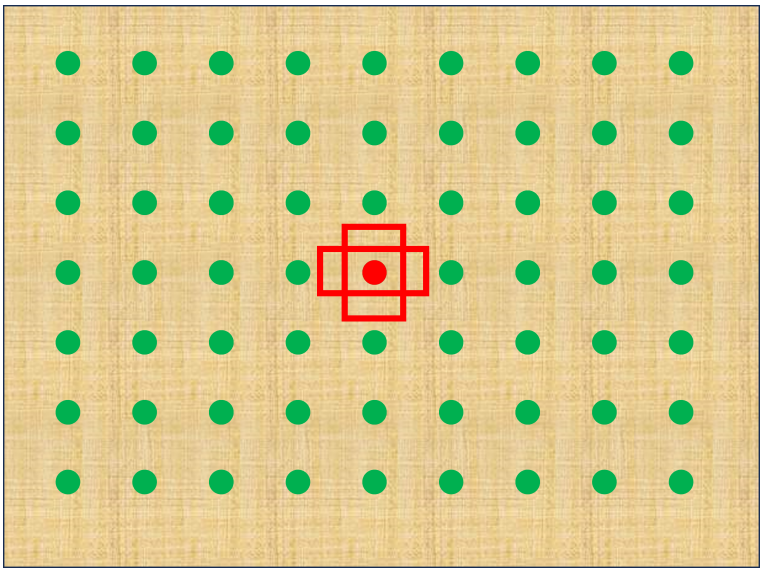

Figure 4.2: An illustration of a grid $H \times W = 7 \times 9$ of anchors (green dots) on an image and a set of $n_{\text{scale}} = 2$ anchors with different aspect ratios for a randomly chosen grid point (red). The total number of anchors in this case equals $\text{H} \times \text{W} \times n_{\text{scale}} = 7 \times 9 \times 2 = 126$.

### 4.5.3 Training an anchor-based model

Given an input image or a video, object detection models provide bounding boxes to characterize both the location and size of detected objects. A bounding box can be defined by four values $\{x, y, w, h\}$, where $x$ and $y$ are the coordinates of the center of the bounding box, while $w$ and $h$ are the width and height of the bounding box.

Both architectures discussed below exploit the idea of anchor boxes. Anchor boxes represent a predefined set of boxes that serve as reference points or initial guesses to guide the model in locating objects within an image. The purpose of anchors is to handle different scales and aspect ratios of objects in an image. As illustrated in Fig. 4.2, anchor boxes are placed on a grid $\text{H} \times \text{W}$ covering the image, and each grid node can share several anchors $n_{\text{scale}}$ with different scales and aspect ratios, resulting in a total number of anchors $n_{\text{anchors}} = \text{H} \times \text{W} \times n_{\text{scale}}$. Each anchor box then is responsible for predicting an object that is close to its size, shape, and location. Along with the coordinates, the model outputs *an objectness score* $\hat{p}_i \in (0, 1)$ that indicates whether the corresponding anchor overlaps with a real object on a given

image. Alternatively, when the classification of a detected object is required, the model outputs a set of $C$ probabilities indicating which class an object belongs to, leading to $n_{\text{anchors}} \times (5 + C)$-dimensional output. In this case, an objectness score can be simply replaced by a "no object" class. In the following, we do not focus on the classification part of the detection process, which is not relevant to this work, i.e., we consider $(n_{\text{anchors}} \times 5)$-dimensional output.

During the training process, each ground truth object is assigned the best matching anchor according to the IoU metric, splitting the anchors into positive ($p_i = 1$) and negative ($p_i = 0$) training samples. The model is trained to distinguish between "object" and "no object" samples by minimizing the loss $\mathcal{L}_{score}(p_i, \hat{p}_i)$, typically in the form of MSE [170] (Eq. 4.4) or binary cross-entropy [171] (Eq. 4.8):

$$\frac{1}{n} \sum_{i}^{n} \mathcal{L}_{score}(p_i, \hat{p}_i),$$

where $n = n_{\text{anchors}}$ for a single image. Indeed, in practice, objects from multiple images form a training mini-batch. Furthermore, different balancing schemes exist that address a typically higher number of negative anchors than positive ones.

Furthermore, the model learns to predict object locations *relative* to anchor boxes $x', y', w', h'$. In this work, we use the commonly employed centroids representation encoding for conversion to relative coordinates:

$$\begin{aligned} x' &= \frac{x_{\text{gt}} - x_{\text{anchor}}}{w_{\text{anchor}}} \\ y' &= \frac{y_{\text{gt}} - y_{\text{anchor}}}{h_{\text{anchor}}} \\ w' &= \ln\left[\frac{w_{\text{gt}}}{w_{\text{anchor}}}\right] \\ h' &= \ln\left[\frac{h_{\text{gt}}}{h_{\text{anchor}}}\right] \end{aligned} \tag{4.27}$$

The corresponding regression loss $\mathcal{L}_{reg}$ is minimized only for the positive samples. In this work, we employ a common smooth $L_1$ loss [172]:

$$\mathcal{L}_{\text{smooth } L_1}(e) = \begin{cases} 0.5e^2 & \text{if } |e| < 1 \\ |e| - 0.5 & \text{otherwise} \end{cases}, \tag{4.28}$$

where $e$ is the prediction error for relative coordinates (e.g., $e_i = x'_i - \hat{x}'_i$). The total loss is given by

$$\mathcal{L} = \mathcal{L}_{reg} + \lambda \mathcal{L}_{score}, \tag{4.29}$$

where $\lambda$ is a hyperparameter.

Below we discuss how NNs can be coupled with the anchor-based representation of the output discussed above.

### 4.5.4 One-stage detectors

CNNs are a common choice for computer vision tasks as explained in Subsec. 4.4.3. As illustrated in Fig. 4.3, an input image with size $H_0 \times W_0$ and a number of channels (colors) $C_0$ is processed by a series of convolutional blocks. Convolutional blocks consist of convolutional layers (Subsec. 4.4.3), activation functions (Subsec. 4.4.2), and normalization layers (Subsec. 4.4.4), and can also feature skip connections (Subsec. 4.4.5) depending on the architecture. The most commonly used architectures include VGG [173], ResNet [174], Inception [175], DarkNet [176], ConvNet [177].

Typically, the resulting intermediate feature maps decrease in spacial size $H_i \times W_i$ and increase in depth, i.e. the number of channels $C_i$. In this manner, an input image is gradually compressed into smaller feature maps, and each feature map pixel with $C_i$ channels corresponds to an area of pixels on the initial image called *the receptive field*. To couple this general scheme with the anchor-based approach discussed above, one should simply set $C_{\text{last}} = n_{\text{scale}} \times 5$. In this way, the last feature map provides relative coordinates and objectness scores for corresponding anchors on an image.

An important practical constraint of this approach is related to the fixed size of the anchor grid being equal to the size of the used feature map. Various scales of object sizes require different grid resolutions. To address this issue, modern detectors

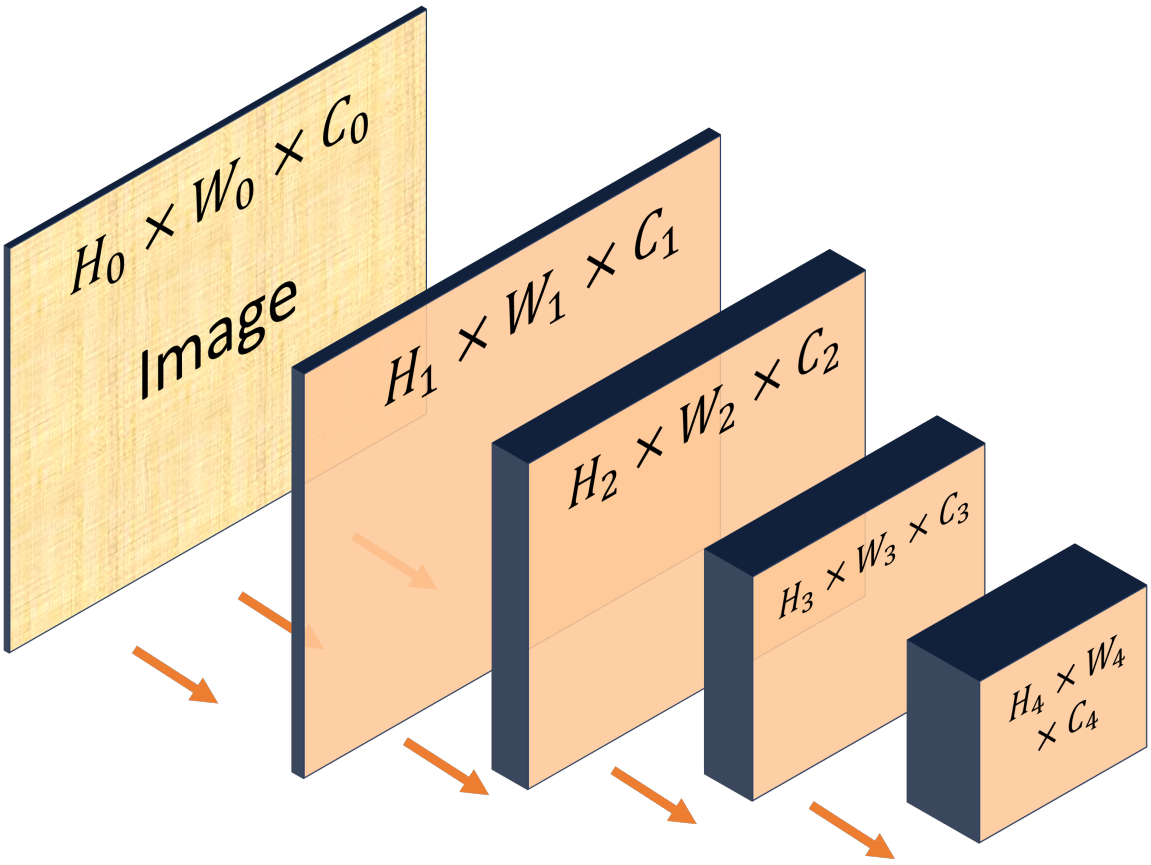


Figure 4.3: An illustration of a CNN feature extractor employed for object detection. An input image with size $H_0 \times W_0$ and a number of channels (colors) $C_0$ is processed by a series of convolutional blocks. The resulting intermediate feature maps decrease in spacial size $H_i \times W_i$ and increase in depth, i.e. the number of channels $C_i$.

typically use multiple feature maps that extract objects at different scales and the corresponding CNN architectures such as Feature Pyramid Network [178].

One-stage detector architectures such as versions of You Only Look Once (YOLO) [170, 176] enable fast inference, but frequently suffer from inaccurate predictions of bounding boxes. Below, we discuss how the additional use of a second detection stage can improve the results at the cost of slower inference.

### 4.5.5 Two-stage detectors

The most known two-stage detector is Faster R-CNN [171]. Despite the fact that it has been developed separately from the one-stage detector architectures discussed above, we explain it conceptually as an addition to the one-stage approach, which is especially valid since we do not discuss the classification task in object detection. In this context, the second detection stage is a model that treats the positive predictions obtained by the first stage as *region proposals* and refines both the objectness score and the bounding boxes based on the same feature maps as the first stage. For each proposal, the network receives a part of the feature map cropped and interpolated based on the bounding box provided by the first detection stage. The corresponding interpolation mechanism is called RoiAlign [179], although other approaches are possible. The choice of architecture for the second detection stage varies depending on the model. In this work, we use FC layers for the second stage. The further details of our model are discussed in Chap. 7.

## 4.6 Normalizing flows

In this section, we discuss normalizing flows - a type of generative neural network model used in Chap. 6.

### 4.6.1 Generative models

Generative deep learning models are a special type of machine learning algorithm that aim to create new data instances that resemble the training data. These models utilize deep neural networks to learn the intricate patterns within the input data and subsequently generate output that shares similar statistical properties. Formally, generative models are trained to provide samples $\boldsymbol{x}_i \sim p_{\boldsymbol{w}}(\boldsymbol{x})$ where the NN-based distribution $p_{\boldsymbol{w}}(\boldsymbol{x})$ approximates the distribution of the training data $p(\boldsymbol{x})$ (or, in the case of conditional distributions, $p(\boldsymbol{y} \mid \boldsymbol{x})$).

The applications of generative models span various domains. In the field of computer vision, they are instrumental in generating and transforming images, contributing to advancements in areas such as virtual reality, video game development, and medical imaging. Natural language processing has also greatly benefitted from these models, where they are utilized to generate coherent and contextually appropriate text.

Generative approaches in machine learning encompass a variety of models designed to generate new data instances that mimic the statistical properties of the training data. Generative Adversarial Networks (GANs) represent one of the most widely used generative approaches. In a GAN, two neural networks, the generator and the discriminator, are trained simultaneously: the generator produces synthetic data while the discriminator tries to differentiate real from generated data. Variational Autoencoders (VAEs) are another key generative models. They employ an encoder-decoder architecture to learn the underlying data distribution and generate new data points from that distribution. Other generative methods include autoregressive models and Transformer-based models like ChatGPT, which have proven successful in generating human-like text. In the following, we discuss another type of generative model called normalizing flows.

### 4.6.2 Principles of flow-based models

This section provides a brief description of the flow-based models following Ref. [180].

Normalizing flows [181] operate by processing a simple density $p_{\mathrm{u}}(\mathbf{u})$ through a series of *invertible transformations* $T(\mathbf{u}) = (T_n \circ \cdots \circ T_1)(\mathbf{u}) = \boldsymbol{x}$ to produce a richer, potentially intractable distribution $p_{\boldsymbol{x}}(\boldsymbol{x})$. $T(\mathbf{u}) = T_{\boldsymbol{w}}(\mathbf{u})$ are "trainable" transformations parameterized by $\boldsymbol{w}$. $p_{\mathrm{u}}(\mathbf{u})$ is referred to as *base distribution*. A typical choice of base distribution is the multivariate normal distribution, which gives the technique its name. Generally, base distribution should allow fast sampling and probability evaluation operations.

Sampling from $p_{\mathrm{x}}(\mathbf{x})$ requires the ability to sample from $p_{\mathrm{u}}(\mathbf{u})$ and to compute the *forward* transformation $T$:

$$\mathbf{x} = T(\mathbf{u}) \quad \text{where} \quad \mathbf{u} \sim p(\mathbf{u}) \tag{4.30}$$

The transformation $T$ must be invertible and both $T$ and $T^{-1}$ must be differentiable. Such transformations are known as diffeomorphisms and require that $\mathbf{u}$ has the same dimensionality as $\mathbf{x}$, i.e. $\mathbf{u}, \mathbf{x} \in \mathbb{R}^d$. Under these conditions, the density of $x$ can be obtained by a change of variables:

$$p_{\mathrm{x}}(\mathbf{x}) = p_{\mathrm{u}}(\mathbf{u}) \left|\det J_T(\mathbf{u})\right|^{-1} \quad \text{where} \quad \mathbf{u} = T^{-1}(\mathbf{x}). \tag{4.31}$$

where the Jacobian $J_T(\mathbf{u})$ is the $d \times d$ matrix of all partial derivatives of $T$:

$$J_T(\mathbf{u}) = \begin{bmatrix} \frac{\partial T_1}{\partial \mathrm{u}_1} & \cdots & \frac{\partial T_1}{\partial \mathrm{u}_D} \\ \vdots & \ddots & \vdots \\ \frac{\partial T_D}{\partial \mathrm{u}_1} & \cdots & \frac{\partial T_D}{\partial \mathrm{u}_D} \end{bmatrix}. \tag{4.32}$$

Normalizing flows provides two operations: sampling from $p_{\mathrm{x}}(\mathbf{x})$ via Eq. 4.30 and evaluating the density $p_{\mathrm{x}}(\mathbf{x})$ via Eq. 4.31. The second operation requires evaluating the determinant of the Jacobian. Importantly, a composition $(T_2 \circ T_1)$ of diffeomorphisms $T_1$ and $T_2$ is also a diffeomorphism. Its inverse and Jacobian determinant are given by:

$$(T_2 \circ T_1)^{-1} = T_1^{-1} \circ T_2^{-1}, \tag{4.33}$$

$$\det J_{T_2 \circ T_1}(\mathbf{u}) = \det J_{T_2}\left(T_1(\mathbf{u})\right) \cdot \det J_{T_1}(\mathbf{u}). \tag{4.34}$$

Therefore, normalizing flows can be constructed via a composition of simple transformations. The main practical challenge is to construct trainable (NN-based) transformations that are capable to approximate complex distributions, but still provide tractable and computationally efficient operations described by Eq. 4.30 and Eq. 4.31. Below, we discuss coupling transformations [182] that address this problem and provide a widely used framework for multiple flow-based models.

### 4.6.3 Coupling transforms

The core idea behind coupling transforms is to split a $d$-dimensional variable $\mathbf{x}$ into two lower-dimensional vectors $\mathbf{x}_1 \in \mathbb{R}^n$ and $\mathbf{x}_2 \in \mathbb{R}^{d-n}$ $(0 < n < d)$:

$$\mathbf{x} = [\mathbf{x}_1, \mathbf{x}_2] = [\mathbf{x}_{1:n-1}, \mathbf{x}_{n:d}]. \tag{4.35}$$

The first vector $\mathbf{x}_1$ is then used to produce parameters $\boldsymbol{w}$ for an element-wise transformation $g_{\boldsymbol{w}}(\mathbf{x}_2)$ of the second vector $\mathbf{x}_2$. For that, an arbitrarily complex function can be used, e.g., a neural network:

$$\mathbf{w} = \mathrm{NN}(\mathbf{x}_{1:n-1}) \tag{4.36}$$

Finally, the transformation $T(\mathbf{x}) = T(\mathbf{x}_1, \mathbf{x}_2) = [\mathbf{y}_1, \mathbf{y}_2]$ is defined as follows:

$$\begin{aligned} \mathbf{y}_1 &= \mathbf{x}_1 \ , \\ \mathbf{y}_2 &= g_{\boldsymbol{w}}(\mathbf{x}_2) \ . \end{aligned} \tag{4.37}$$

The determinant of the Jacobian of this transformation does not involve derivatives w.r.t. NN and simply equals

$$\det J_T(\mathbf{x}) = \prod_{i=n}^{d} \frac{\partial g_{\boldsymbol{w}_i}}{\partial x_i} \ . \tag{4.38}$$

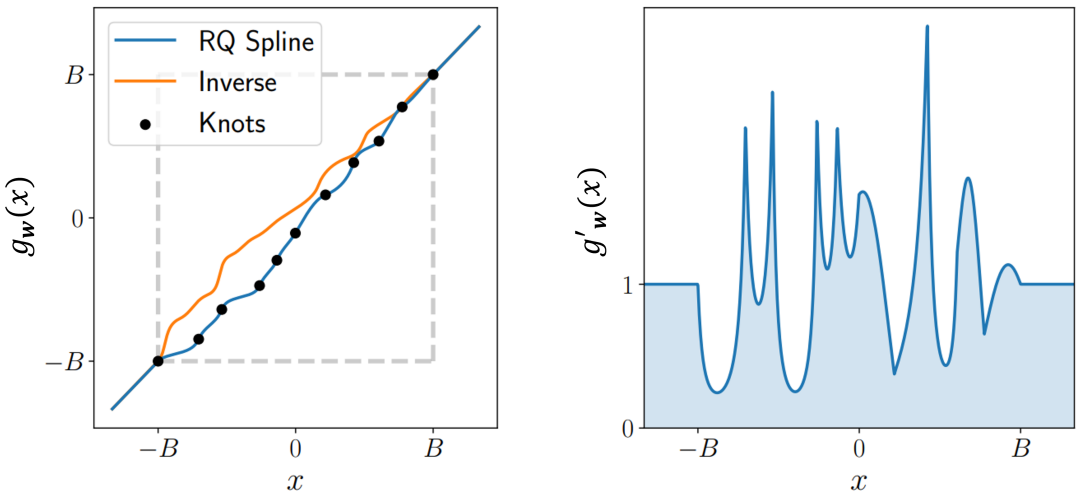


Figure 4.4: A monotonic rational-quadratic transform with $K = 10$ bins (left-hand side) and the derivative of the transform (right-hand size). Figure is adapted from Ref.[183].

This framework enables the use of arbitrarily complex neural networks that define the parameters of simpler, tractable transformations $g_{\mathbf{w}}(\cdot)$. Different $g_{\boldsymbol{w}}(\cdot)$ transformations have been proposed, the simplest one being the affine transformation [182]:

$$g_{\boldsymbol{w}}(x_i) = \alpha_i x_i + \beta_i, \quad [\boldsymbol{\alpha}, \boldsymbol{\beta}] = \boldsymbol{w}. \tag{4.39}$$

which is straightforward to implement in practice. However, the capacity of this transformation might not be sufficient for constructing complex distributions.

In this work, we employ monotonic rational-quadratic splines for the transformation $g_{\boldsymbol{w}}(\cdot)$ proposed in Ref. [183]. As illustrated in Fig. 4.4, the monotonic rational-quadratic spline maps an interval $[-B, B]$ to the same interval $[-B, B]$ via $K$ bins. It is constructed by $K + 1$ knots placed within the interval with two extreme knots being fixed to the edges of the intervals. The parameterization $\boldsymbol{w} \in \mathbb{R}^{3K-1}$ includes $K$ bin widths, $K$ bin heights, and $K - 1$ derivatives $\delta_i = g'_{\boldsymbol{w}}(x_i)$ [184]. The parameters $\boldsymbol{w}$ are provided by a NN according to the general coupling transformations framework discussed above.

### 4.6.4 Training normalizing flows

Similarly to the standard NNs, flow-based models can be trained by minimizing some loss between the target distribution $p^*_{\mathrm{x}}(\mathbf{x})$ and the model $p_{\mathrm{x}}(\mathbf{x}, \boldsymbol{w})$ w.r.t. parameters of NNs $\boldsymbol{w}$. The most commonly employed loss is the KL divergence:

$$D_{\mathrm{KL}}(p\|q) \equiv \mathbb{E}_p \left[\log\left(\frac{p}{q}\right)\right] . \tag{4.40}$$

Since the KL divergence is not symmetric, there are two options for defining the loss: the forward KL divergence $D_{\mathrm{KL}}(p^*_{\mathrm{x}}(\mathbf{x})\|p_{\mathrm{x}}(\mathbf{x}, \boldsymbol{w}))$ and the reverse KL divergence $D_{\mathrm{KL}}(p_{\mathrm{x}}(\mathbf{x}, \boldsymbol{w})\|p^*_{\mathrm{x}}(\mathbf{x}))$. We focus on the first (forward) option that can be efficiently estimated in the case we can sample from the target distribution $p^*_{\mathrm{x}}(\mathbf{x})$, or, equivalently, possess a training dataset of multiple samples $\{\mathbf{x}_i\}_{n=1}^N$ assumed to represent the target distribution:

$$\begin{aligned} J(\boldsymbol{w}) &= D_{\mathrm{KL}}\left[p^*_{\mathrm{x}}(\mathbf{x})\|p_{\mathrm{x}}(\mathbf{x};\boldsymbol{w})\right] \\ &= -\mathbb{E}_{p^*_{\mathrm{x}}(\mathbf{x})}\left[\log p_{\mathrm{x}}(\mathbf{x};\boldsymbol{w})\right] + \text{ const} \\ &= -\mathbb{E}_{p^*_{\mathrm{x}}(\mathbf{x})}\left[\log p_{\mathrm{u}}\left(T^{-1}(\mathbf{x};\boldsymbol{w})\right) + \log\left|\det J_{T^{-1}}(\mathbf{x};\boldsymbol{w})\right|\right] + \text{ const}, \end{aligned}$$

where the constant term does not depend on the model parameters $\boldsymbol{w}$. Given the dataset $\{\mathbf{x}_i\}_{n=1}^N$, we can approximate the KL divergence via Monte Carlo:

$$J(\boldsymbol{w}) \approx -\frac{1}{N}\sum_{n=1}^{N} \log p_{\mathrm{u}}\left(T^{-1}\left(\mathbf{x}_n;\boldsymbol{w}\right)\right) + \log\left|\det J_{T^{-1}}\left(\mathbf{x}_n;\boldsymbol{w}\right)\right| . \tag{4.41}$$

Unlike the forward KL divergence, the reverse KL divergence $D_{\mathrm{KL}}(p_{\mathrm{x}}(\mathbf{x}, \boldsymbol{w})\|p^*_{\mathrm{x}}(\mathbf{x}))$ requires evaluating the likelihood but does not require sampling from it.

# 5 Reflectometry analysis via prior-aware machine learning

The focus of the following two chapters is the analysis of reflectometry data. This chapter introduces a novel deep learning technique for Maximum Likelihood Estimation-based reflectometry analysis discussed in Sec. 3.1. Some of the results presented below are based on Refs. [93–95].

## 5.1 Ambiguity and the standard ML regression

The conventional fitting procedure ("Parratt fit") is based on differential evolution (DE) (see Subsec. 3.1.4). It is generally time-consuming, requires expertise and experience, and does not guarantee a reliable solution. To address these issues, recently multiple solutions have been proposed that exploit modern ML methods. These approaches have the potential to improve both the speed and reliability of the analysis. However, they have been shown to be efficient only for simple scenarios with a small number of estimated parameters. In the following, we briefly discuss the standard ML approach employed for reflectometry analysis in a range of publications [84–89, 91] and illustrate the fundamental limitation of this approach related to ambiguity.

First, one typically defines a slab model with a fixed number of layers. Some of the parameters of the slab model are kept open to be estimated by the neural network based on the measured reflectivity curve, while the other parameters generally can be fixed according to the expected fixed experimental setup. The open parameters are constrained within broad ranges $[\boldsymbol{\theta}^{total\ min}, \boldsymbol{\theta}^{total\ max}]$, and the neural network is

trained to estimate these parameters within the defined ranges. The training is performed using examples obtained by calculating reflectivity curves via Parratt [54] or Abel'es [111] formalisms for a set of training parameters.

The standard training procedure works as follows. First, the training dataset is generated by sampling $N$ parameters $\{\boldsymbol{\theta}_i^{\text{sim}}\}_{i=1}^N$ from a defined range. For each sample, the corresponding reflectivity curve $\boldsymbol{R} = R_i(\boldsymbol{q})$ is calculated for some predetermined set of $q$ points. During this step, various experimental conditions can be addressed by incorporating different types of noise and misalignment into the simulation. During training, the predicted parameters $\text{NN}(\boldsymbol{R}_i) = \boldsymbol{\theta}_i^{\text{ML}}$ are calculated, and the neural network is updated using the backpropagation algorithm (see Subsec. 4.3.3) to minimize the prediction loss $\mathcal{L}_{\text{MSE}} = ||\boldsymbol{\theta}_i^{\text{sim}} - \boldsymbol{\theta}_i^{\text{ML}}||_2^2$ (Eq. 4.4). In this way, the neural network $\text{NN}(\boldsymbol{R}) = \boldsymbol{\theta}^{\text{ML}}$ is trained to approximate the inverse function $F^{-1}(\boldsymbol{R}) = \boldsymbol{\theta}$, where $F(\boldsymbol{\theta}) = \boldsymbol{R}$ is the *forward process* calculated via Parratt algorithm or Abelès transfer-matrix method on a set of fixed $q$ points. If the model prediction is sufficiently close to the likelihood maximum, MLE can be obtained in the post-processing stage using gradient descent [91].

As mentioned above, this approach is meant to replace the slower iterative "Parratt fit". However, unlike the conventional fit, the standard ML models typically do not rely on any *sample-dependent* parameter ranges $\Theta_{\text{sample}}$ that are necessary to avoid ambiguous solutions. Instead, the ranges $[\boldsymbol{\theta}^{total\ min}, \boldsymbol{\theta}^{total\ max}]$ are fixed during the training. Therefore, to address the ambiguity issue, these parameter ranges should be set narrow enough to exclude ambiguous examples. This condition significantly constraints the practical usage of this approach, as every studied sample might have different prior ranges, which in turn might necessitate the training of a new model.

Furthermore, ambiguous solutions occur even within narrow parameter ranges. Fig. 5.1 demonstrates an example with a *single* open parameter - density $\rho$ of a single layer on top of a substrate. There are two SLD profiles with distinct densities $\rho_1$ and $\rho_2$ that, however, correspond to practically the same reflectivity curves, hence the same prediction $\rho_1^{\text{ML}} = \rho_2^{\text{ML}}$. If both SLD profiles appear within the training range,

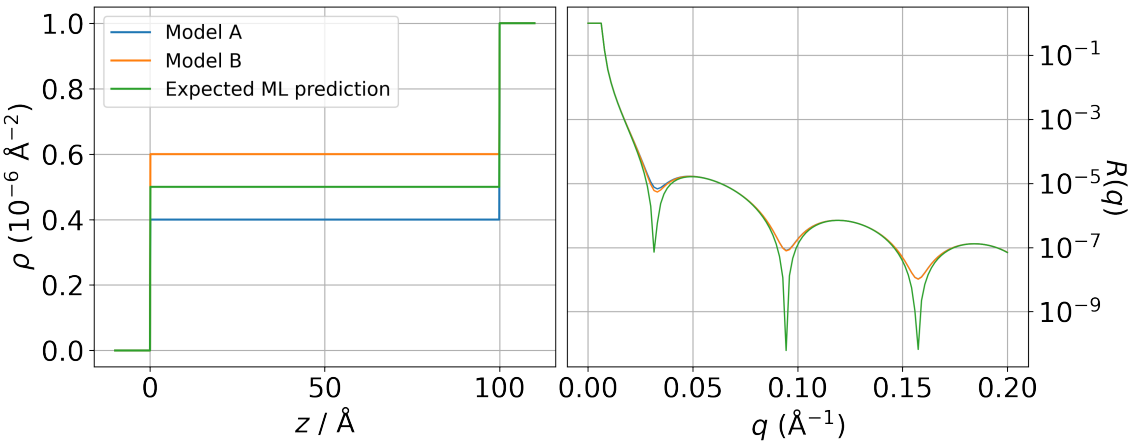


Figure 5.1: An example of ambiguity in reflectometry with a single open parameter of a single layer on a substrate. Model A and B correspond to different SLD profiles (left-hand side) with densities of the single layer being equal to $0.4 \times 10^{-6}$Å$^{-2}$ and $0.6 \times 10^{-6}$Å$^{-2}$. The corresponding reflectivity curves (right-hand side) are practically indistinguishable, revealing only minor changes at lower $q$ points that would typically appear within the ranges of measurement uncertainty due to counting statistics. A neural network trained on such a dataset cannot distinguish between two curves and is expected to provide a single solution for both examples. The "best" ML prediction ($0.5 \times 10^{-6}$Å$^{-2}$) corresponding to the minimal training loss is illustrated by the green SLD profile and the green reflectometry curve.

the result of minimizing the training loss will lead to $\rho_1^{\mathrm{ML}} = \rho_2^{\mathrm{ML}} = (\rho_1 + \rho_2)/2$, i.e. the wrong prediction that generally does not contain information about the correct parameters (in general, $\boldsymbol{\theta}^{\mathrm{ML}} = \sum_{i=1}^{n} w_i \boldsymbol{\theta}_i^{\mathrm{ambiguous}}$). By increasing the number of open parameters and/or parameter ranges, one consequently increases the number of such ambiguous solutions, inevitably leading to incorrect NN predictions on most of the samples.

The natural solution to these issues is to accommodate various (prior) parameter ranges within a single neural network. This approach, developed in this work, is discussed in detail below.

## 5.2 Prior-aware ML regression

### 5.2.1 Incorporating prior knowledge into ML pipeline

We address the ambiguity problem by designing a novel ML method that enables the use of dynamically set prior knowledge. To achieve that, we include the boundaries of the open parameters as an additional input to the neural network. As before, for each open parameter $\theta_j$ we designate broad ranges $[\theta_j^{total\ min}, \theta_j^{total\ max}]$ that determine the general scope of the neural network. Yet, in conjunction with these fixed ranges, we introduce *sample-dependent* ranges $[\theta_j^{\min}, \theta_j^{\max}]$ ($\theta_j^{\min} \leq \theta_j^{\max}; \theta_j^{\min}, \theta_j^{\max} \in [\theta_j^{total\ min}, \theta_j^{total\ max}]$) that impose constraints on the fitted parameters for each sample studied and are supplied as additional input to the neural network alongside the measured curve. This method effectively confines the solution space for a particular sample while maintaining extensive overall parameter ranges within a single neural network. In instances where a single solution exists within the provided sample-dependent boundaries, the inverse operation becomes well-defined, enabling a precise fit. We note that this approach effectively combines the best of two worlds: the flexibility of the parameter ranges of the conventional differential evolution method, and the speed of the neural network.

The new approach requires a modified training procedure since the training data should include both the ground truth parameters and the corresponding sample-dependent prior domain. We generate the training data in the following manner. Firstly, for each parameter $\theta_j$, we obtain the center ($c_j$) and the width ($w_j$) of the local prior domain, the center being uniformly sampled from the global domain $[\theta_j^{total\ min}, \theta_j^{total\ max}]$ and the width being uniformly sampled from a predefined width range for that parameter. The resulting local prior domain is defined by the ranges $[\theta_j^{\min} = c_j - w_j/2, \theta_j^{\max} = c_j + w_j/2]$. Furthermore, we obtain a ground truth training parameter by uniform sampling from the local ranges. Same as in the standard ML approach, the ground truth parameters are used to calculate reflectivity curves, which are updated using the generated noise and the scaling procedure [88]. However, along with the scaled noisy reflectivity curves, we provide an additional input in the form

of local parameter ranges $[\theta_j^{\min}, \theta_j^{\max}]$ to NN, so that $\boldsymbol{\theta}^{\mathrm{ML}} = \mathrm{NN}(\boldsymbol{R}, \boldsymbol{\theta}^{\min}, \boldsymbol{\theta}^{\max})$. In this way, the model has information about the parameter ranges that in principle can be used to eliminate all the ambiguous solutions outside the prior domain. In order to "force" the model to use this information, we explicitly incorporate it into the architecture of the model. We do so by rescaling the ground truth parameters with respect to the corresponding local bounds, making the neural network predict the parameters *relative to the local bounds*. Consequently, the output scaled parameters are expected to be in the range $[-1, 1]$, and the rescaled parameters appear within the local bounds.

### 5.2.2 Distribution of prior ranges

The proposed incorporation of prior information into the neural network relies on the assumption that ambiguous solutions within the prior ranges are infrequent, if not eliminated altogether. Therefore, the distribution of the prior ranges needs to be carefully defined to ensure that the ranges are narrow enough to fulfill this assumption, while still being wide enough to be practically useful.

In practice, to construct such a distribution, we leverage the fact that densities of materials are usually known by experimentalists with a higher level of certainty compared to other parameters. Therefore, we limit the parameter ranges $w_\rho$ of density parameters $\rho$ by some small value while enabling a higher level of uncertainty for other parameters. It means that the model is trained on a large range of density values (the typical range used in this work is [0, 60] $10^{-6}\text{Å}^{-2}$), but for every given sample, the widths ($w_\rho$) of the prior ranges of densities are constrained ($w_\rho \in [0.01, 4]$ $10^{-6}\text{Å}^{-2}$). For other parameters such as thicknesses and roughnesses, we do not set such constraints on parameter ranges.

### 5.2.3 Fast reflectometry calculations

An additional enhancement in the training procedure compared to previous methods is accomplished by incorporating a fast, vectorized, GPU-accelerated reflectometry

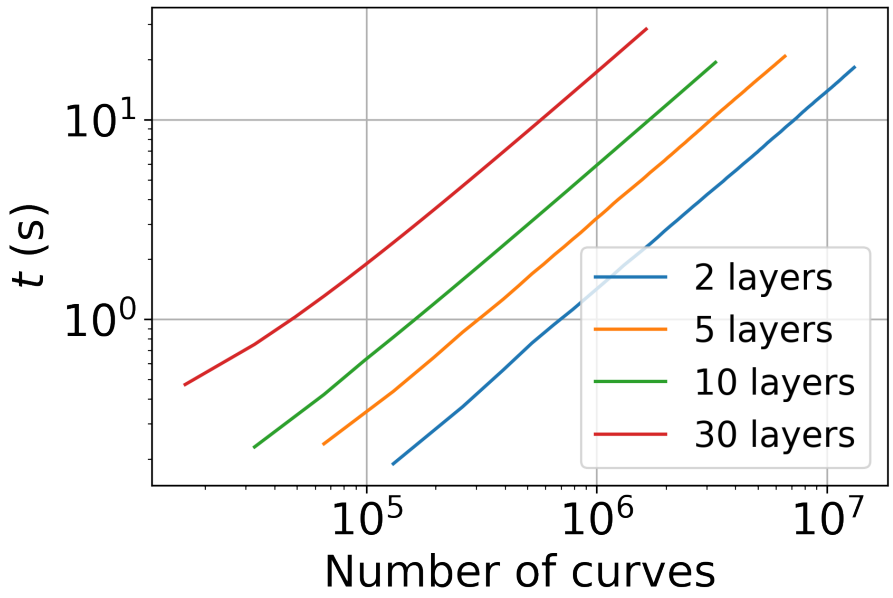


Figure 5.2: Performance tests of the developed PyTorch implementation of reflectometry simulations. The number of calculated curves as a function of time for 2, 5, 10, and 30 layers of a slab model with 128 $q$ points was measured using a single GPU NVIDIA GeForce RTX 2080 Ti on a personal computer (PC). The speed ranges from $10^4$ to $10^6$ curves per second depending on the setup.

calculation using the Abel'es transfer-matrix method in PyTorch [185]. This implementation significantly speeds up reflectometry calculations, producing up to $10^6$ simulated curves per second depending on the used hardware and the setup (see the performance tests in Fig. 5.2). It also allows seamless integration of the simulation process into the ML pipeline. This enables an improved training scheme compared to the standard ML approaches. Instead of generating the training dataset in advance, on each training iteration, we simulate a *new* batch of reflectivity curves. This approach ensures there is no overfitting to the training data, eliminating the need for a separate validation set. With this innovation, multiple models can be trained directly during an experiment at a synchrotron facility. In turn, this allows for the adjustment of training parameters such as $q$ points, parameter ranges, and noise levels to match the experimental setup, thereby enhancing the performance of the model.

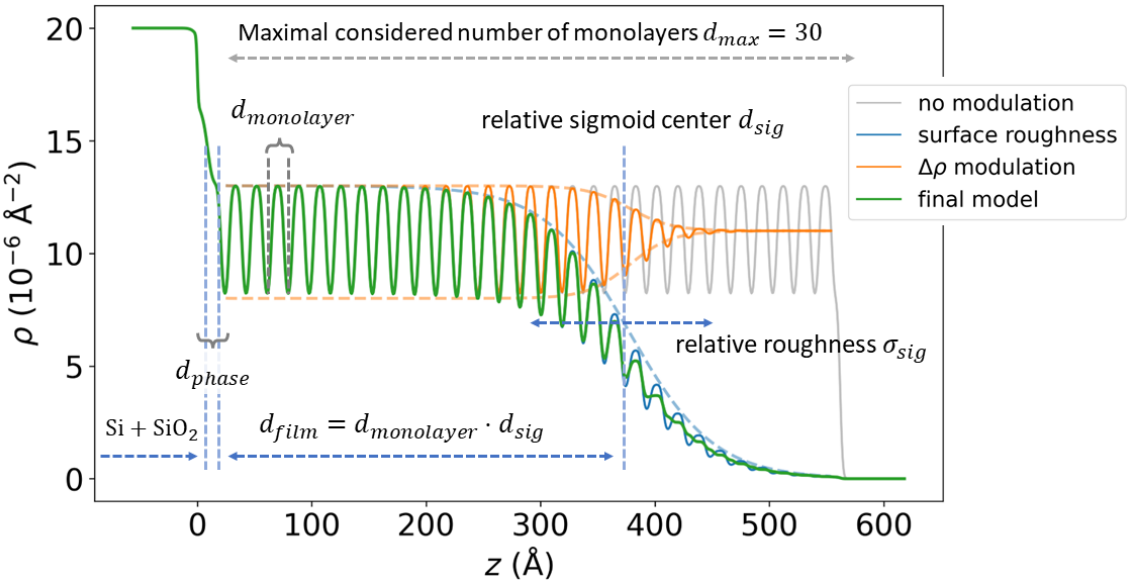


Figure 5.3: Physics-informed parameterization of the SLD profile for a thin film consisting of repeating identical monolayers on top of a substrate. The grey curve shows the base SLD profile of the monolayers, the blue curve corresponds to the SLD profile with surface roughness modulated by the first sigmoid, and the orange curve shows the result of the modulation of the SLD amplitude via the second sigmoid. The green curve represents the final SLD profile.

### 5.2.4 Physics-informed parameterization of SLD profile

The second method for integrating physical knowledge into the ML framework that we employ in this work is the physics-informed parameterization of SLD profiles. In the case of ML-based reflectivity analysis, this approach was first introduced by Ref. [186], where the physics-informed growth model effectively reduced the number of estimated parameters. In this work, we apply this approach to the case of periodic multilayer structures [187, 188]. The standard parameterization of the slab model implies three parameters per single layer - density, thickness, and roughness (or four parameters when including absorption). Given that a single molecular monolayer block is typically modeled by two layers, for 30 monolayers such parameterization would require up to $2 \cdot 30 \cdot 3 = 180$ independently fitted parameters resulting in a $180-$dimensional solution space. However, such parameters are largely correlated, because the monolayers consist of the same material, feature the same thickness, etc.

To provide this information to the neural network, we introduce a set of 17 independently fitted or predicted parameters, that together define all the 180 parameters of the box model. Such parameterization is based on a physical understanding of monolayer structures and significantly decreases the task's complexity.

The physical scenario is the following: on top of a silicon/silicon oxide substrate we consider a thin film composed of repeating identical monolayers (grey curve in Fig. 5.3), each monolayer consisting of two boxes with distinct SLDs. A sigmoid modulating the SLD profile of the monolayers defines the film thickness and the roughness at the top interface (blue curve in Fig. 5.3). A second sigmoid can be used to modulate the amplitude of the monolayer SLDs as a function of the displacement from the position of the first sigmoid (orange curve in Fig. 5.3). These two sigmoids allow one to model a thin film that is coherently ordered up to a certain coherent thickness and gets incoherently ordered or amorphous toward the top of the film. Such a scenario is sometimes encountered when Kiessig and Laue fringes show different periods. In addition, a layer between the substrate and the multilayer is introduced to account for the interface structure, which must not necessarily be identical to the multilayer period. This so-called "phase layer" is important as the relative phase between a strong substrate reflection and a multilayer Bragg reflection can lead to very different shapes of the curve around the Bragg reflection for constructive or destructive interference.

### 5.2.5 Invariant transformations in reflectivity data

Harvesting the symmetries in the data is essential for developing an efficient ML solution. In the case of reflectometry, we highlight two invariant transformations that can be used to improve the ML-based analysis.

The first transformation is related to the unit system used to define SLD parameters $\boldsymbol{\theta}$ and momentum transfer $\boldsymbol{q}$. The units are inverse Angstroms ($Å^{-1}$) for $\boldsymbol{q}$ values, Angstroms (Å) for layer thicknesses $\boldsymbol{d}$ and roughnesses of the interfaces $\boldsymbol{\sigma}$, and inverse squared Angstroms ($Å^{-2}$) for densities $\boldsymbol{\rho}$. It is obvious that changing

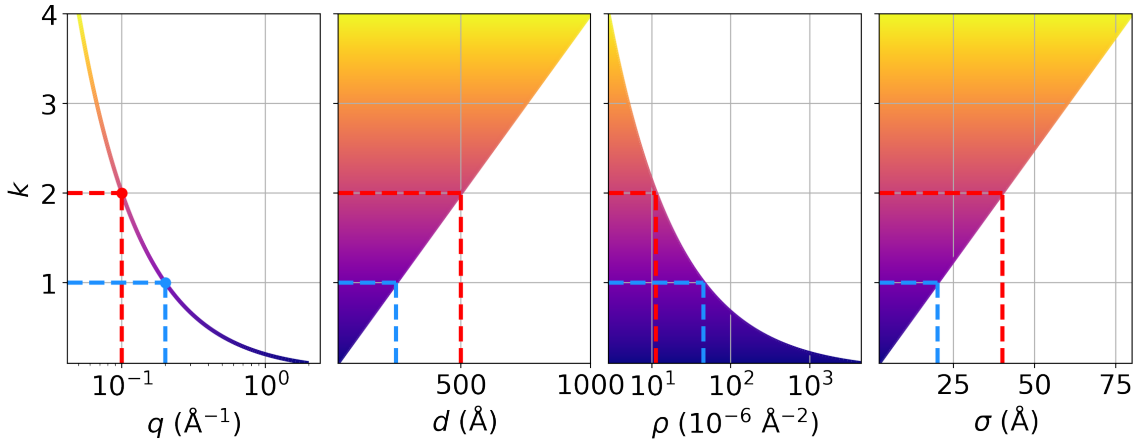


Figure 5.4: Scaling invariance of the SLD parameters and momentum transfer $q$. A model trained with a fixed $q$ range $q_{\text{max}} = 0.2$ Å$^{-1}$ (blue dashed lines) can be then applied to the measured data with a different $q$ range, but the total parameter ranges are scaled accordingly. The red dashed lines demonstrate this scaling effect for $q_{\text{max}} = 0.1$ Å$^{-1}$. The colored areas denote the ranges within which the trained model can be applied. Figure is adopted from Ref. [95].

Angstroms to other units such as microns, meters, or inches does not alter the resulting reflectivity curve. In general, by scaling the $q$ range by a factor of $k$ such that $\hat{\boldsymbol{q}} = \boldsymbol{q} \cdot k$, we can rescale other parameters so that the resulting reflectivity curve stays the same: $\hat{\boldsymbol{d}} = \boldsymbol{d}/k$, $\hat{\boldsymbol{\sigma}} = \boldsymbol{\sigma}/k$, $\hat{\boldsymbol{\rho}} = \boldsymbol{\rho} \cdot k^2$. The resulting invariant transformation

$$T_k(R(\boldsymbol{q}, \boldsymbol{d}, \boldsymbol{\sigma}, \boldsymbol{\rho})) = R(\boldsymbol{q}, \boldsymbol{d}, \boldsymbol{\sigma}, \boldsymbol{\rho}), \tag{5.1}$$

where

$$T_k(R(\boldsymbol{q}, \boldsymbol{d}, \boldsymbol{\sigma}, \boldsymbol{\rho})) \equiv R(\boldsymbol{q} \cdot k, \boldsymbol{d}/k, \boldsymbol{\sigma}/k, \boldsymbol{\rho} \cdot k^2) \tag{5.2}$$

shall be accounted for when designing the ML solution. In this work, we do so by fixing the $q_{\text{max}}$ to some standardized value. Indeed, this approach exactly follows the previous standard ML solutions for reflectivity. However, by exploiting the invariant transformation (Eq. 5.1), we can apply the same neural network on experimental data with different $q_{\text{exp}}$ range and then rescale the resulting parameters using the scaling factor $k = q_{\text{max}}/q_{\text{exp}}$ as illustrated in Fig. 5.4. This property is used in the following for analyzing experimental curves with different $q$ ranges.

The second invariant transformation is based on the fact that the scattering properties of a sample and the resulting reflectivity curve are determined by contrasts - differences between layer densities $\Delta\rho_{ij} = \rho_i - \rho_j$, and not the absolute values $\rho_i$. As a result, the model trained with a fixed (zero) density of an ambient medium, can be then applied for different ambient densities by shifting the resulting densities accordingly.

We note that both approaches discussed above require scaling the predicted parameters. Indeed, this requires that the unscaled predicted parameters stay within the global domain of the model. In this way, the total parameter ranges used to train a NN generally have to be adjusted (increased) based on both the expected parameter ranges of the target experimental scenarios and the expected $q$ ranges.

### 5.2.6 Input data representation

The selection of an appropriate neural network architecture typically hinges on several factors. These include the characteristics of the input and output data representations, the presence of symmetries within the dataset, and other properties of the task that determine the inductive bias. Additionally, empirical studies often guide this decision, with certain architectures demonstrating superior performance in specific applications.

The symmetries in the reflectometry data discussed above do not influence the architecture. The output data depends on the parameterization of SLD profile and is a fixed set of scaled parameters, making a basic fully-connected layer a valid choice for producing the output data. The main challenges in the case of reflectometry analysis are related to the input data. In the following, we discuss the most general way to incorporate the reflectometry data as an input to the neural network and the simplifications employed in this work.

The input reflectometry data typically contains a discrete set of measured reflectivity values corresponding to $q$ points. As discussed in Subsec. 5.2.5, the total $q$ range can be set fixed without loss of generality. However, the discretization of the $q$

axis (i.e., the number and positions of $q$ points) may vary for different measurements. For instance, $q$ points are generally not distributed linearly in an actual experiment. Instead, in a typical XRR experiment, the angles are changed linearly, which leads to a non-linear change in momentum transfer $q$, which is especially relevant in the case of measurements involving larger $q$ ranges with multiple Bragg peaks. In the case of NR and time-of-flight measurements, $q$ points are distributed stochastically and cannot be set fixed ahead of the experiment. To fully address the properties of the data, the model shall receive a set of pairs $\{q_i, R_{\text{exp}}(q_i)\}$. Additional information about the uncertainty of each measurement $\{\delta q_i, \delta R_{\text{exp}}(q_i)\}$ can also be added as an input to the model. We also note that due to the possible errors introduced in the preprocessing procedure (e.g., normalization, footprint correction), one shall consider the input beyond the standard set $\{q_i, R_i, \delta q_i, \delta R_i\}$. In this case, the corresponding training procedure of the neural network should incorporate the corresponding errors in simulated preprocessing operations, as well as a varying number of $q$ points and the corresponding measurements. The corresponding neural network architecture should support sequential input (e.g., Transformers, Graph Neural Networks, Recurrent Neural Networks). Obviously, one should not forget the use of prior information introduced in this work as an additional input with fixed dimensionality.

Despite this general approach seems feasible, it requires substantial efforts in constructing the corresponding simulations and training sequential models. Therefore, we leave these efforts for future research, and focus on the simplified approach introduced in previous studies [84, 91]. The main idea of this approach is to train NN on a fixed set of $q$ points. When applied to experimental data with different $q$ points, the data is interpolated to the fixed $q$ set. In this case, interpolating the uncertainty information $\{\delta q_i, \delta R_i\}$ is questionable at best, therefore, the only input to the model is interpolated reflectivity curve $\{R_{\text{exp}}(q_i^{\text{interp}})\}$. The fixed input dimensionality allows the use of the most basic neural network architectures such as MLP.

We emphasize that despite the interpolated data is used to provide an initial ML prediction, it can be further refined during the postprocessing stage with the use of

original data that includes the uncertainty levels $\{\delta q_i, \delta R_i\}$. Following the previous studies, we employ this simplified approach throughout this work and demonstrate its applicability. However, we note that the problem of efficient data representation and the corresponding choice of NN architectures is a potential way to further improve the overall approach to ML-based reflectometry analysis. Indeed, the introduced prior-aware approach is conceptually orthogonal to the discussed problem and can be combined with different data representations and architectures.

### 5.2.7 Embedding network

Therefore, we use a fixed set of $q$ points and the corresponding reflectivity curve together with prior information as an input to NN. Nevertheless, we improve the previously employed architectures to achieve faster convergence and better accuracy. Unlike some previous works, where a reflectivity curve is directly provided as input to the MLP, we first use a network that produces a latent embedding of the reflectivity curve which is subsequently fed to the MLP.

For the embedding network, we use a one-dimensional (1D) CNN [189], which is parameter-efficient and better leverages the sequential characteristics of the data compared to MLP. While less popular than their 2D counterparts, 1D CNNs have been successfully used in a variety of tasks involving 1D signals such as automatic speech recognition [190] and time-series prediction [191]. The particular architecture and the corresponding hyperparameters might vary depending on the necessary capacity depending on a task-dependent setup (number of parameters, number of $q$ points, parameter ranges, etc.), but the general scheme stays the same throughout this work. The embedding network is a sequence of 5 blocks, each block containing a 1D CNN, followed by a batch-normalization layer and a GELU activation function (see Sec. 4.4). Each convolutional layer features a kernel of size 3, stride= 2, and padding= 1. Consequently, the dimension of the processed reflectometry is (approximately) halved after each layer. The number of channels is doubled in each block,

starting from 32 up to 512. The final embedding with dimensionality $d = 128$ is produced by an adaptive average pooling layer and a fully-connected layer.

The latent representation is then provided to MLP model to transform it into the output parameters. We construct MLP as a composition of 3 residual blocks (see Sec. 4.4) containing a fully-connected layer, batch normalization, and a GELU activation function.

### 5.2.8 Experimental noise and artifacts

To make NNs applicable to the real-world experimental data, the training data should represent experimental noise and possible artifacts such as misalignment and preprocessing errors.

To model counting statistics, we define a noise distribution $p(s \mid \theta)$ as a Poisson distribution with an incoming number of photons $n_0(q)$ sampled uniformly for each $q$ point such that the standard deviation ranges from 5% to 20% of the reflected intensity, which aligns well with typical test experimental data.

Furthermore, we acknowledge two possible sources of experimental artifacts due to misalignment. First is the scaling factor $\Delta I \in [-0.05, 0.05]$ that addresses potential normalization error. The second is $q$ offset $\Delta q \in [-2 \cdot 10^{-3}, 2 \cdot 10^{-3}]$, which provides a linear approximation to a misalignment caused by a systematic error in angle determination $\Delta\alpha$. Finally, we add the constant background $b = 10^{-10}$ for the XRR data and $b \in [10^{-10}, 10^{-4}]$ for the NR data. In the case of neutron measurements, instrumental resolution $\delta q$ plays a significant role and should be also incorporated into the data simulation pipeline. Indeed, all the ranges defined above can be adjusted according to the expected noise levels.

The resulting simulated reflectivity curve before applying Poisson noise follows the equation below:

$$R_{\text{sim}}(\boldsymbol{q}, \boldsymbol{\theta}) = R(\boldsymbol{q} + \Delta q, \boldsymbol{\theta}) \cdot (1 + \Delta I) + b \tag{5.3}$$

We note that for the ML approach discussed in the current chapter, the simulated noise does not have to correspond to a trackable likelihood used in the postprocessing stage, and in principle any type of noise mixtures can be applied if it leads to increased performance on the test set. However, the correspondance between the simulated noise and the likelihood becomes essential in the more advanced approach that we discuss in the next chapter.

## 5.3 ML-controlled *in situ* XRR measurements

*In situ* XRR measurements provide a potent tool, facilitating the real-time exploration of temporal evolutions in the physical properties of the samples under study, potentially under varying experimental conditions. This approach is commonly employed in such scenarios as the real-time examination of film deposition and annealing processes. Typically, the experimental conditions are planned ahead of the measurement or controlled semi-automatically with the help of additional measurements. For instance, a common way to track a film thickness during deposition is via Quartz Crystal Microbalance (QCM) which has certain limitations and potentially non-negligible systematical errors. XRR data provides a way to more accurately estimate the current film thickness, as well as other parameters such as evolving surface roughness (that obviously cannot be measured via QCM). However, the conventional analysis of the obtained data generally cannot be reliably conducted in real time, constraining the potential of the experimental technique.

Our NN provides a solution in milliseconds, enabling real-time analysis of the measured data. This unlocks new possibilities for conducting a new type of data-driven autonomous experiments with the environment parameters being actively controlled based on the real-time ML analysis. The first autonomous *in situ* XRR experiment of this kind with ML-controlled experimental conditions during film deposition was successfully performed at the European Synchrotron Radiation Facility (ESRF) with the use of our prior-aware ML method. We note that the use of the prior-aware approach is essential in this experiment since it enables connecting the analysis of

distinct reflectivity curves via real-time adjustments of the prior ranges based on previously obtained fits. The results obtained during this experiment are discussed below with a focus on the performance of the ML model.

### 5.3.1 Conceptual scheme of the experiments

The described experiment is a proof-of-concept attempt to integrate the developed ML approach into the experimental setup for conducting autonomous *in situ* measurements during the film deposition. To test and verify the results of the experiment, we have defined a set of target thicknesses for the growing film. The autonomous system has been designed to perform fast XRR scans during film deposition at a constant growth rate, obtain the instantaneous parameters of the sample via ML-based fit, and use them together with the results from previous scans to extrapolate the time when the next target thickness will be achieved. When achieving the predicted time, the system temporarily stops the deposition by closing the shutter and performs the long high-resolution XRR scan to accurately estimate the achieved film thickness. After that, the growth is continued until the next target thickness is reached. The correspondence between the target and the measured thicknesses demonstrated in the following confirms both the overall applicability of the method for such autonomous experiments and the accuracy of the ML-based fit.

We note that the use of prior knowledge for the analysis of obtained reflectivity curves is essential for *in situ* measurements, as it allows us to connect the analysis of the subsequent measurements. We do so by defining the sample-dependent constraints on the growing film thickness based on the obtained results from a previous fit. This way, we leverage the physical understanding of the growth process (thickness of the growing layer cannot decrease over time) and the experimental setup (thickness cannot increase too rapidly), without a specific growth model restriction.

### 5.3.2 Fitting the growth rate

One of the challenges of the *in situ* XRR analysis is related to the temporal evolution of the sample parameters *during the measurement of a single reflectivity curve*. As the incidence angle and the corresponding momentum transfer $q$ increase with time during the measurement, so does the thickness of a deposited film. We address this problem during the postprocessing step by introducing an additional fitted parameter that models a linear change of thickness during the measurement of a single curve. Therefore, for a single scan, we take into account the time-dependent layer thickness $d(t)$ and the time-dependent incidence angle corresponding to momentum transfer $q(t)$. We note that while other parameters such as the roughness of the deposited layer can also change in time, the corresponding impact on the quality of the fit is not as prominent as in the case of changing layer thickness. Therefore, we treat other parameters $\boldsymbol{\theta}_{\text{other}}$ as time-independent, leading to the following time-dependence for the reflectivity curve: $\boldsymbol{R} = R(q(\boldsymbol{t}), d(\boldsymbol{t}), \boldsymbol{\theta}_{\text{other}})$. A linear model of a growth process $d(t) = d_0 + t\frac{\Delta d}{\Delta t}$ is a very good approximation for the short duration of a quick real-time scan. Consequently, we fit two additional parameters $d_0$ and $\frac{\Delta d}{\Delta t}$ for each curve via gradient descent-based least mean squares (LMS), using the ML prediction as an initial guess for the parameter $d_0$. The introduction of this time-dependent parameter is required to accurately fit XRR curves measured during the growth in the case of a fast growth process relative to the data acquisition time for a single XRR profile, as Kiessig fringes will be slightly narrower at high $q$ as compared to low $q$.

### 5.3.3 Experimental setup

The experiment was performed on the surface scattering branch of the ESRF ID10 beamline [192]. A beam energy of 17.0 keV and beam size of $30 \times 30$ $\mu m$ was used. Molecular thin film samples are prepared *in situ* using molecular beam deposition [187, 193].

The ML-based closed-loop control system was implemented to achieve the objective of stabilizing and terminating the self-assembly, growth, and crystallization characteristics of molecular thin film studies. In particular, the autonomous system was given control over the operation of two shutters, which involves covering either the substrate or the incoming molecular beam. To limit beam damage, which might occur for longer scan times at lower deposition rates, we reduced the X-ray flux to a level where there was no noticeable impact on Bragg peak intensities over the time of a growth run at our deposition rates.

In this study, molecular thin films of aluminium-tris(8-hydroxychinolin)($Alq_3$, $C_{27}H_{18}AlN_3O_3$), a frequently used component in organic light-emitting diodes, were grown to serve as an exemplary material system for amorphous molecular thin films [194]. To demonstrate the online capabilities regarding the analysis of crystalline multilayer systems the organic semiconductor N,N‘-Dioctyl-3,4,9,10- perylendicarboximid (PTCDI-$C_8$, $C_{40}H_{42}N_2O_4$) [187, 188] was chosen for demonstration purposes. For details on the scientific background of these materials, we refer to other studies [195–197].

### 5.3.4 Single layer fits (Kiessig oscillations)

In Fig. 5.5a we show exemplary reflectometry profiles acquired during thin film growth of $Alq_3$ using fast scans together with the corresponding ML fits of layer parameters. The plot demonstrates a good fit quality using the ML approach with priors. Both the period of Kiessig fringes as well as the roughness-induced damping of Kiessig fringes are correctly reproduced in the ML fits. This is a prerequisite for achieving closed-loop control to terminate growth at the target thickness, but due to the finite duration of the scans (45$s$ per XRR profile in Fig. 5.5a) and an average growth rate of 1 nm/min, the target thickness may be reached during a scan. Therefore, we used a linear extrapolation of the thickness based on the information from previous scans to automatically calculate the best moment to close the shutter. Fig. 5.5b shows the result of the closed-loop deposition control for several target

thicknesses between 80 Å and 640 Å. The target thicknesses are plotted on the horizontal axis, while the true film thicknesses at which the deposition was terminated are given on the vertical axis. As expected for a functioning closed-loop control, the data indeed falls on the diagonal line, except for one outlier. Overall, the chosen target thicknesses could be reached within ±2Å (0.1%) average accuracy, thereby affirming the method's suitability for autonomous experiments.

### 5.3.5 Multilayer fits (Bragg reflections)

Apart from amorphous thin film structures, we studied periodic multilayers of the molecule PTCDI-$C_8$. By incorporating physical knowledge of the sample structure into the parametrization of the used ML model (see Subsec. 5.2.4) we were able to fit Laue fringes and the molecular Bragg peak that arises with increasing film thickness from molecular multilayers (see Fig. 5.6a). To speed up data acquisition, only a relatively short $q$ range centered around the molecular Bragg peak was repeatedly scanned while running in autonomous mode. An initial "full" XRR curve, including the total reflection edge, was used for signal normalization before activating the closed-loop autonomous mode. Using the short $q$ range that does not include the total reflection edge also made it possible to speed up the training of the NN model by employing the kinematical approximation for on-the-fly training data generation. The corresponding curves appeared to be indistinguishable from those calculated using the Abelès transfer-matrix method within the defined $q$ and parameter ranges.

The obtained scans around the Bragg reflection were fitted by ML with high fidelity, reproducing the Bragg peak and the period of the Laue oscillations and their asymmetric intensity to the left and right of a Bragg reflection. The corresponding electron density profiles from the live fits are shown in Fig. 5.6b, from which one can directly infer the number of deposited monolayers, as one oscillation of the SLD corresponds to one PTCDI-$C_8$ monolayer.

Fig. 5.7a) compares the film thicknesses obtained by the model and the total thickness derived from a measurement of the deposition flux with a QCM. For thick-

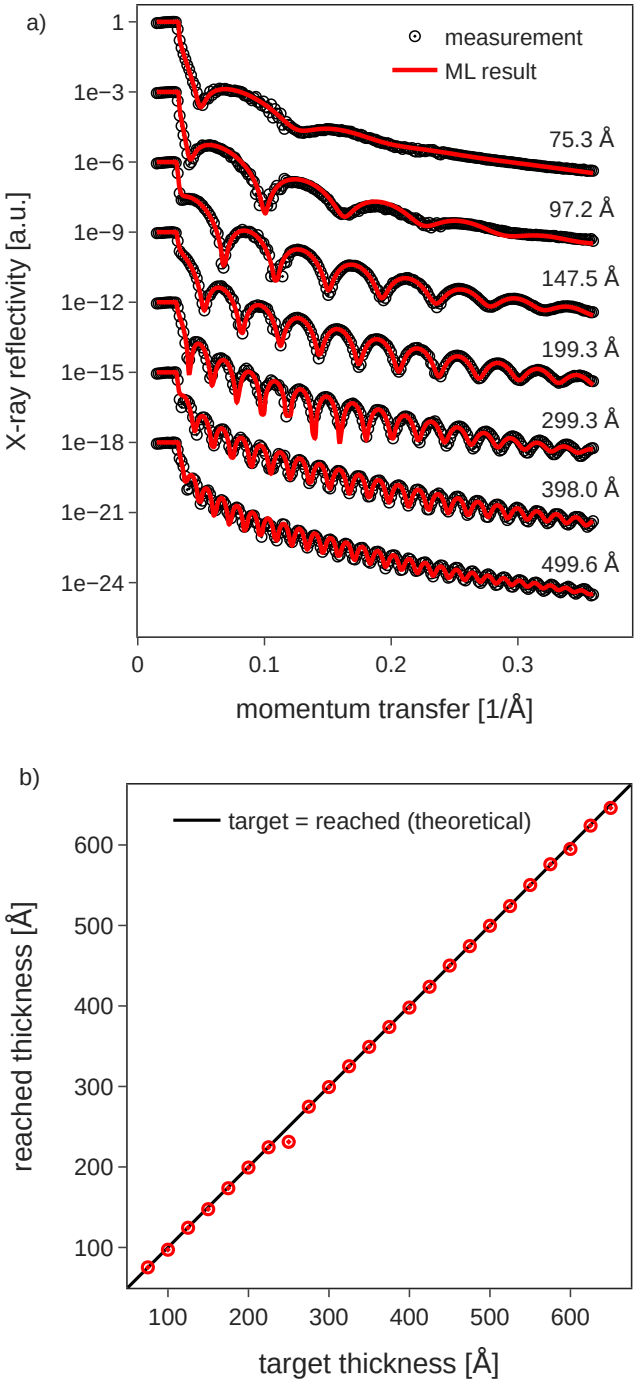


Figure 5.5: ML-controlled thin film deposition: a) Exemplary measurements and fits. The experimental data are acquired using fast, real-time scans, thus the poor counting statistics around $q \simeq 0.11/$ Å is due to the limited number of absorber changes. b) Target thicknesses in closed loop operation vs. actually measured thicknesses after the closed loop feedback terminated the growth. Figure is adopted from Ref. [93].

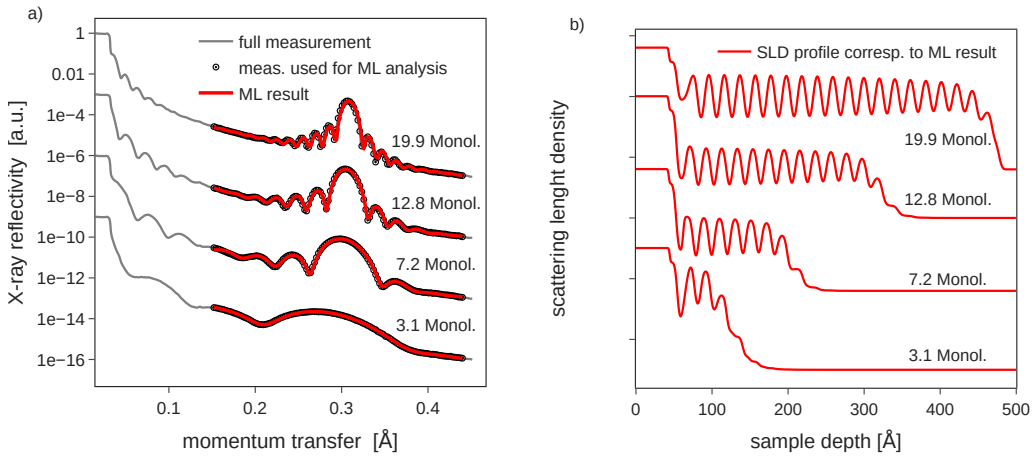


Figure 5.6: a) ML-based fits for "long scans" of PTCDI-$C_8$ multilayer structures. The model is only supplied with the short $q$ range around the PTCDI-$C_8$ Bragg peak to enable faster data acquisition during *in situ* measurements. b) Scattering length density profiles corresponding to a). Figure is adopted from Ref. [93].

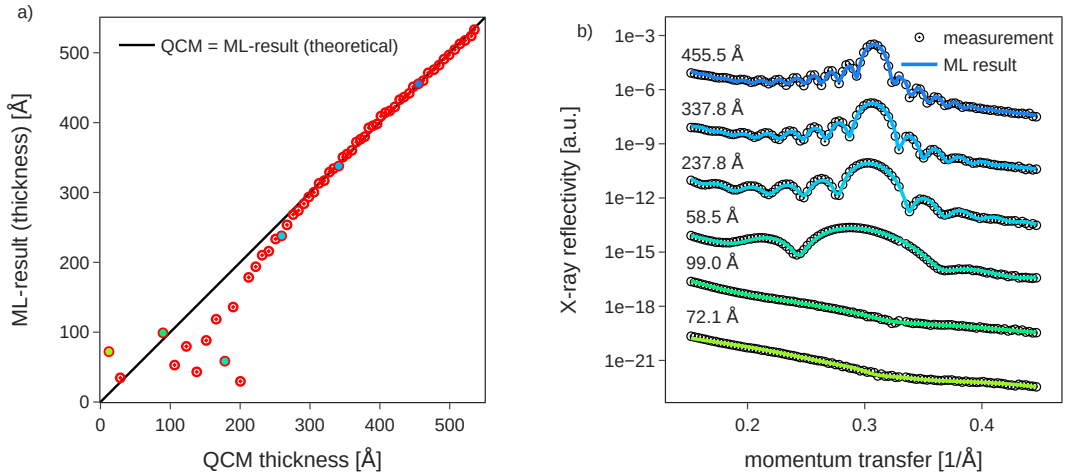


Figure 5.7: a) Thin film thickness extracted through ML model in real-time during the measurement compared to a QCM reference. For films with thickness above $\simeq$ 250 Å, i.e. $\simeq$ 12 molecular monolayers, ML results are reliable. b) Selected fits from the real-time dataset. Ambiguity in the thickness predictions for lower thicknesses mainly results from the absence of significant features in the investigated $q$ range. Figure is adopted from Ref. [93].

nesses above 250 Å, the agreement is good. Some scatter is visible below this thickness, even though the quality ML fit of the individual reflectometry curves is good (Fig. 5.7b). This particular example shows the difficulty of ambiguous XRR fits, where several sample structure models can fit a single X-ray curve. We note that the QCM measures total film thickness only if a constant sticking factor on the substrate and previously deposited molecular material is assumed, however, the obtained low thicknesses do not correspond to the monotonic growth process. Incorrect thicknesses yet corresponding to accurate fits obtained via ML fit for smaller thicknesses highlight the ambiguity issue discussed earlier. This problem is particularly noticeable when dealing with "featureless" curves on rather short $q$ ranges, and the prior information appears insufficient in this case. Further, the Laue oscillations do not correspond to the total film thickness, but the coherently ordered film thickness, so initial disorder in the crystal lattice may to some degree contribute to the observed deviations for low film thicknesses (additionally, of course, the potentially non-integer layer occupancy during growth may interplay as well [198]).

In conclusion, we find that unless ambiguity is a factor, the ML results of the coherent film thickness during data acquisition nicely match our detailed post-growth analysis and yield consistent data for the larger film thickness of our sample system.

## 5.4 Tests on complex simulated data

Finally, we train our prior-aware model on a complex scenario with a high number of parameters. We use a 5-layer slab model with all 17 parameters being open: 5 thicknesses, 6 roughnesses, and 6 densities. We use the fixed $q$ axis with 256 equidistant points in the range $[0.02\text{Å}^{-1}, 0.3\text{Å}^{-1}]$, and the following parameter ranges: [0, 300] Å for all the thicknesses, [0, 60] Å for the roughnesses, and [0, 25] $10^{-6}\text{Å}^{-2}$ for the SLDs. The ranges of the prior bound widths are [0.01, 300] Å for the thicknesses, [0.01, 60] Å for the roughnesses, and [0.01, 4] $10^{-6}\text{Å}^{-2}$ for the SLDs.

Fig. 5.8 shows examples of input simulated curves and predictions for the 5-layer model, together with the corresponding SLD profiles. The performance of the model

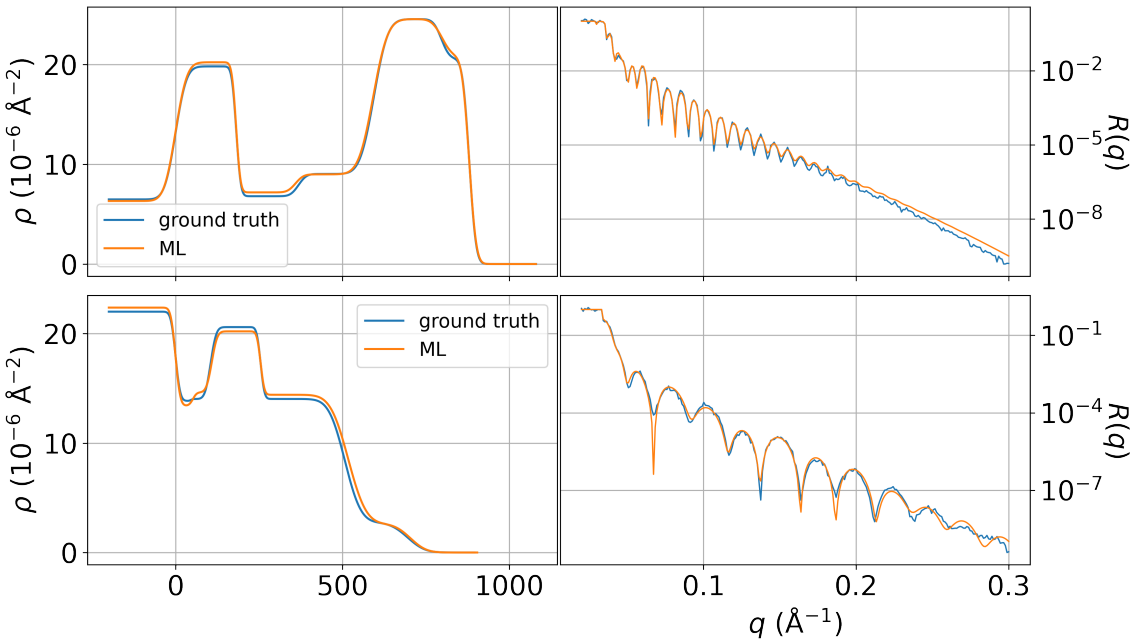


Figure 5.8: Two examples of test simulated samples (blue) and the corresponding neural network predictions (orange) for the 5-layer model. SLD profiles on the left-hand side correspond to the reflectivity curves on the right-hand side of the plot.

can be further enhanced by employing additional postprocessing techniques, such as standard "polishing" via gradient descent.

The absolute errors between the ground truth and the predicted parameters are displayed in Fig. 5.9. Notably, the prediction errors exhibit consistent behavior across the various layers within the model. It is worth mentioning that these absolute errors are either comparable or even lower than the errors previously reported using a standard ML approach with only three open parameters [91]. However, it is important to acknowledge that the total ranges differ between these cases, introducing complexities in making quantitative comparisons. The main conclusion though is that our approach enables employing ML to analyze complex scenarios with many parameters open, significantly increasing the scope and applicability of the method in practice. We note that the computational time required for inference for this model is practically negligible, typically on the order of milliseconds, making our approach a plausible alternative to the conventional iterative fit.

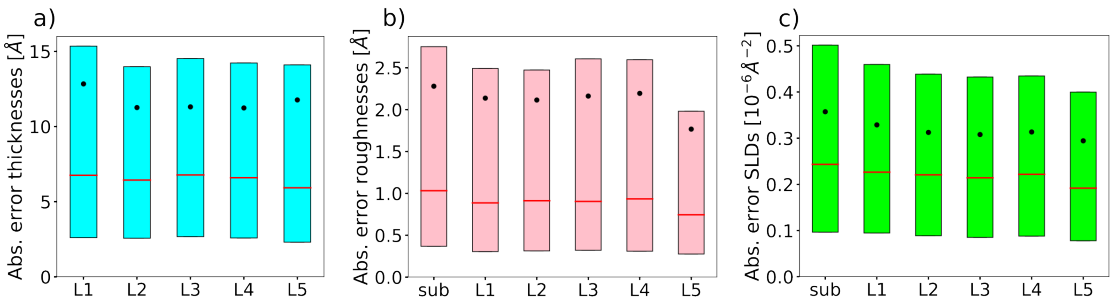


Figure 5.9: Boxplots of the absolute errors of each predicted parameter of the 5-layer model, computed over a batch of 4096 simulated curves. The horizontal red line denotes the median, the black dot denotes the mean and the lower and upper extremities of the box plot denote the $25^{\text{th}}$ percentile and the $75^{\text{th}}$ percentile respectively. Figure is adopted from Ref. [94].

## 5.5 Summary

In this chapter, we address the phase problem as the primary obstacle in machine learning-based approaches for extracting information from X-ray and neutron reflectivity data. The lack of phase information induces non-uniqueness in the space of possible solutions, the inverse problem becoming increasingly underdetermined the larger the considered parameter space. This prevents the successful training of neural networks for complex multilayer structures, with previous solutions being limited to simple scenarios. To address this issue, we propose a procedure that enables training a neural network on a continuous range of smaller subspaces of a large parameter space. Our method allows users to incorporate prior experimental knowledge by specifying upper and lower bounds for each parameter during inference. We note that the proposed ML solution can be adapted to tackle other inverse problems in science affected by the non-uniqueness issue.

Our approach overcomes limitations in existing ML methods, as it allows training networks with a larger number of parameters and expanded parameter ranges. In contrast to previous work, our method scales favorably when increasing the complexity of the inverse problem, giving good predictions for the challenging 5-layer structures. The introduced physics-informed parameterization of the SLD profile enables analyzing even more challenging scenarios with multiple periodic layers. We

demonstrate that in cases when some parameters are not included in the ML pipeline such as growth rate for *in situ* measurements, the solution can be refined during the postprocessing stage.

Our solution is practical and lightweight as it allows training a model from scratch to a working solution within hours. As a result, the training procedure can be performed during the actual experiment, e.g. on a synchrotron facility, to incorporate the current experimental setup into the ML pipeline. We consider our approach a plausible alternative to a conventional iterative fit, being extremely faster, yet allowing experimentalists to incorporate their knowledge into the analysis in the form of parameter ranges. We also note that our approach shares the same means of validation with the conventional fit, since in both cases the quality of the fit ("goodness-of-fit", obtained likelihood, etc.) is the only measure for both methods that indicates the correctness of the obtained physical parameters within the selected physics-informed parameter ranges.

Finally, we emphasize that both the presented solution and conventional fit share the same potential flaws. The general assumption underlying the use of both methods is that, for a large set of cases, the prior information is sufficient to isolate a single solution and resolve intrinsic ambiguity in reflectometry. However, narrow priors do not guarantee a single solution, as multiple solutions might occur close to each other. As shown in Sec. 5.1, close non-unique solutions might arise even in the simplest examples with a single estimated parameter. Furthermore, potential correlations between the estimated parameters may hugely increase the uncertainty levels that are not obtained by the discussed methods. As a result, these methods might provide a "good" single solution, but leave the experimentalist unaware of other potential solutions within the defined parameter ranges. Such a "blind" approach that does not extract the complete information from the measured data is indeed not an intrinsic problem of the measurements themselves. Instead, it is a problem related to the chosen analysis methods. In the next chapter, we focus on other analysis methods that circumvent this general problem and provide a more comprehensive and reliable analysis of reflectometry data.

# 6 Probabilistic machine learning for Bayesian reflectometry analysis

In the previous chapter, we discussed the commonly employed MLE-based approach to reflectometry analysis. However, it can potentially produce unreliable results, constraining the applicability of reflectometry as an experimental technique. To circumvent the flaws of the discussed MLE-based methods, in this chapter, we consider Bayesian analysis, which is becoming increasingly popular in the reflectometry community in light of discussed challenges related to ambiguity. Bayesian inference can provide a reliable analysis of reflectometry data, unlocking the full potential of the experimental technique. The results presented in this chapter are based on Ref. [95].

Below, we first discuss the fundamental limitation of the conventional Bayesian methods in the case of reflectivity data. Second, we introduce a novel ML-based approach that bypasses these limitations providing accelerated Bayesian reflectometry analysis.

## 6.1 Sample efficiency and the curse of dimensionality

In Subsec. 3.2.1 we introduced sample efficiency - a common metric employed by the IS technique that demonstrates the efficiency of Monte Carlo (MC) sampling from a given proposal distribution $u(\boldsymbol{x})$ for resolving a target distribution $p(\boldsymbol{x})$. In the following, we argue that this metric can be applied to other conventional gradient-free Bayesian reflectometry analysis methods.

Sample efficiency depends on the proposal distribution. In the case of Bayesian analysis, the natural choice of the proposal distribution is the prior distribution that reflects all the prior knowledge about the estimated distribution.

As shown in Ref. [199], the sample efficiency computed by importance weights (Eq. 3.18), asymptotically converges to the expected ratio of probability densities $\epsilon^*$:

$$\epsilon = \frac{n_{\text{eff}}}{n_{\text{total}}} \xrightarrow{a.s.} \epsilon^* = \left( \mathbb{E}_{\boldsymbol{x} \sim p(\boldsymbol{x})} \left[ \frac{p(\boldsymbol{x})}{u(\boldsymbol{x})} \right] \right)^{-1} . \tag{6.1}$$

To understand this quantity intuitively, let us first calculate it for the simplest case of two one-dimensional uniform distributions

$$u(x) = \begin{cases} 1/u_0, & \text{if } x \in [0, u_0] \\ 0, & \text{otherwise} \end{cases}$$

and

$$p(x) = \begin{cases} 1/p_0, & \text{if } x \in [0, p_0] \\ 0, & \text{otherwise,} \end{cases}$$

where $p_0 > u_0$ are the widths of the target and proposal uniform distributions correspondingly. In this case, the Eq. 6.1 gives

$$\epsilon* = p_0/u_0$$

Therefore, when considering uniform distributions, sample efficiency is exactly equal to the probability of "hitting" the non-zero peak $p(x)$ by sampling from a proposal distribution $u(x)$. On average, it would require $1/\epsilon^*$ samples to obtain a single sample within the peak of interest in order to locate its position, and it would need orders of magnitude more to reliably reveal all the potential distributional modes of the target distribution.

Typical applications including reflectometry consider multiple estimated parameters, leading to high-dimensional distributions. In our simplistic example with two

uniform distributions, the corresponding sample efficiency decreases exponentially with the increasing number of parameters $d$:

$$\epsilon^* = (p_0/u_0)^d$$

This equation shows a general (intuitively obvious) behavior of the sample efficiency as a ratio of characteristic "volumes" of the estimated and the proposal distributions. Indeed, the estimated posterior distribution is rarely uniform. We note, however, that a uniform distribution is a common choice for a prior (proposal) distribution in reflectometry analysis. For a more general expression that only assumes uniform proposal distribution, we can write

$$\epsilon^* = \left(\mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x})}\left[p(\boldsymbol{x})\right]\right)^{-1} / u_0^d = v(p(\boldsymbol{x}))/v(u(\boldsymbol{x})), \tag{6.2}$$

where

$$v(p(\boldsymbol{x})) \equiv \left(\mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x})}\left[p(\boldsymbol{x})\right]\right)^{-1} \tag{6.3}$$

can be considered an "efficient volume" of the distribution $p(\boldsymbol{x})$, and $v(u(\boldsymbol{x})) = u_0^d$ is the volume of the uniform proposal distribution. To provide some intuition for better understanding this quantity, we provide an analytical form for a normal distribution $\mathcal{N}(\mu,\, \sigma^2)$:

$$\left(\mathbb{E}_{x\sim p(x)}\left[p(x)\right]\right)^{-1} = 2\sqrt{\pi}\sigma \approx 3.55\sigma \ .$$

Correspondingly, in the $d$-dimensional case:

$$\left(\mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x})}\left[p(\boldsymbol{x})\right]\right)^{-1} = (2\sqrt{\pi}\sigma)^d$$

Therefore, in the case of the uniform proposal distribution, sample efficiency is the ratio of the "efficient volumes" between the target and proposal distribution, and it decreases exponentially as the number of parameters increases. The number of samples required to reveal all the distributional modes is inversely proportional

to sample efficiency. While this outcome may appear evident, it causes profound implications for reflectometry analysis.

The problem arises when the corresponding sample efficiency is extremely low. In practice, it leads to the situation that most of the samples result in practically zero probability, and it might take a long time to "hit" a non-zero area. Various techniques propose adaptive importance sampling strategies that aim at increasing the sample efficiency by iteratively updating the proposal distribution based on the previously obtained weights. However, obtaining areas with zero probabilities provides very little information about the location of distributional modes (unless gradient-based methods can be employed), and before such locations are revealed via random exploration techniques, all the refining adaptive procedures are practically useless.

We emphasize that this problem applies equally to other techniques that rely on the random exploration of the parameter space, such as DE (see Subsec. 3.1.4). Importantly, this issue also significantly affects MCMC. As discussed in Subsec. 3.2.2, MCMC depends on the initialization procedure. Consider a scenario where the samples are initialized by drawing an ensemble from a prior distribution. If all corresponding samples fall in an area with (practically) zero probability, they provide no information about the location of the peaks. In this case, the convergence to the target distribution may take *longer* than random exploration via IS depending on the specific MCMC algorithm used (our tests with several MCMC algorithms [133, 200] suggest that it takes significantly longer). Therefore, in cases of low sample efficiency, gradient-free MCMC methods require an additional algorithm to locate good initialization points based on the peak locations, which brings us back to random exploration.

The standard reflectivity packages such as refnx [83] rely on the DE algorithm to provide good initialization for MCMC. The crucial flaw of this approach is that DE provides a single solution, corresponding (in the best case) to the maximum posterior, or MLE in the case of uniform priors. By design, this approach almost guarantees to miss all the other isolated distributional modes that may correspond to correct physical solutions. Indeed, one can find a better initialization algorithm

that saves all the found peaks, or use other techniques such as IS, but the overall problem with potentially low efficiency stays the same.

To provide a numerical estimation of the discussed efficiency, let us assume a typical degree of uncertainty for each estimated parameter. Practically obtained numbers for *a single parameter* (marginal distribution) from the reflectometry analysis of the experimental data discussed in this chapter lie in the range $[10^{-4}, 1]$. In the following, we make an estimation for $p_0/u_0 = 10^{-2}$. The corresponding sample efficiency is $\epsilon^* = 10^{-2d}$, where $d$ is the number of fitted parameters. In practice, one should consider sample efficiency in relation to the computational time required to conduct the analysis, which is determined by the speed at which probability $p(x)$ is estimated. In the case of reflectometry analysis, we rely on the performance metrics from our cutting-edge GPU-based implementation (see Subsec. 5.2.3), which provides an estimate for a maximum speed of $10^6$ curves per second one can obtain on modern hardware without the use of supercomputers. Furthermore, one requires a sufficient number of effective samples $n_{\text{eff}}$ to reliably resolve all the potential distributional modes, which we set here to $n_{\text{eff}} = 100$. Given these exemplary settings, the total required number of samples is $n_{\text{total}} = n_{\text{eff}}/\epsilon^* = 10^{2(d+1)}$, and the corresponding time is $10^{2(d-2)}$ seconds. This rough estimation suggests that already for $d = 3$ open parameters, one requires 100 seconds to obtain a solution. Indeed, in practice, $d$ can be significantly larger. For $d = 10$, our estimation suggests 1B seconds ($\approx 31$ year), which is equivalent to infinity for practitioners. Based on these estimations, we consider sample efficiencies below $10^{-9}$ (corresponding to days of constantly running GPU) as practically unfeasible to analyze via IS. This estimation is done for a simple setup of 2 layers on top of the substrate that corresponds to 1M curves per second. More complex setups lead to longer computations, further constraining the use of the discussed method.

Importantly, these estimations only consider marginal distributions, i.e. they provide an upper bound to the "efficient volume" of the posterior and the corresponding sample efficiency by ignoring correlations between the parameters.

In summary, low sample efficiency is the fundamental limitation of traditional Bayesian reflectometry analysis methods. This limitation restricts the use of conventional methods to simple scenarios with a small number of parameters and narrow prior distributions. To address this issue, we propose the utilization of probabilistic ML solutions discussed below.

## 6.2 Prior Aware Neural Posterior Estimation

The discussed limitation can be circumvented by estimating posterior distribution directly based on a shape of a measured reflectometry curve, however, the analytical approach appears unfeasible, requiring the use of numerical methods. The increasingly popular solution is the Neural Posterior Estimation (NPE) approach - a NN that "learns" to approximate posterior distribution based on the simulated data. It has been successfully applied to multiple scientific tasks [201–204].

The idea behind NPE is to train a flow-based model $p_{\boldsymbol{w}}(\boldsymbol{\theta}|\boldsymbol{R})$ (see Sec. 4.6) to estimate the posterior distribution for a wide range of parameters. It is trained by providing a large dataset of simulated examples sampled from some fixed wide prior distribution $p(\boldsymbol{\theta})$ and adjusting the parameters $\boldsymbol{w}$ of the neural network to minimize the forward KL divergence $D_{KL}(p||p_{\boldsymbol{w}})$. During the inference, the neural network is supplied with the measured data $\boldsymbol{R}$ and provides the approximation of the posterior distribution (Eq. 3.11) for some wide fixed prior distribution $p(\boldsymbol{\theta})$.

Importantly, the prior knowledge about the system under study is fixed and incorporated into the training procedure of this method. This approach is ideally suited for various applications in observational science [202–204]. However, experimental methods actively intervene with the studied systems by controlling various parameters of the experiment. Such active interventions on the studied system effectively change the prior and thereby render the NPE network inapplicable to the updated setup. This prevents an interactive interplay between data analysis and experimental modifications. Reflectometry measurements notably fall into this second category.

To address this, we propose Prior Aware Neural Posterior Estimation (PANPE), which accommodates dynamically set, *sample-dependent* prior distributions – an extension of the NPE method with a static prior distribution and a probabilistic counterpart of the regression approach introduced in the previous chapter. Figure 6.1 illustrates the general concept of the method. Multiple solutions emerging within a broad range of parameters (Fig. 6.1a-b) can be narrowed down by reducing the parameter ranges (Fig. 6.1c-d). In the case of standard NPE, a large number of samples outside the updated narrow prior distribution have to be rejected during the postprocessing stage, significantly reducing sample efficiency. Owing to multiple solutions, the fraction of samples falling within a prior distribution can be as scant as one in a million when generated via NPE, rendering this approach largely inapplicable. Meanwhile, PANPE already generates samples within the given prior domain, achieving high efficiency and enabling real-time analysis. Same as NPE, PANPE enables integration with conventional Bayesian methods such as importance sampling (PANPE-IS) and Markov Chain Monte Carlo (PANPE-MCMC), thereby reliably delivering precise posterior estimation [203]. Ultimately, our method broadens the transformative impact of the NPE approach within experimental science by enabling the dynamic inclusion of prior information.

## 6.3 Training procedure

### 6.3.1 Overall concept

We train the normalizing flows model $p_{\boldsymbol{w}}(\boldsymbol{\theta} \mid \boldsymbol{R})$ to approximate the posterior distribution given the measured data $R$ and *the prior distribution specific to the studied sample*. Therefore, instead of a fixed prior distribution $p(\boldsymbol{\theta})$ used in the standard NPE approach, we define a class of distributions $p_{\boldsymbol{\phi}}(\boldsymbol{\theta})$ parameterized by $\boldsymbol{\phi}$ and provide $\boldsymbol{\phi}$ as an additional input to the neural network along with the curve $\boldsymbol{R}$.

As discussed in Chap. 5, simple box priors are most commonly used in Bayesian reflectometry analysis, i.e. a product of independent uniform distributions $p(\boldsymbol{\theta}) =$

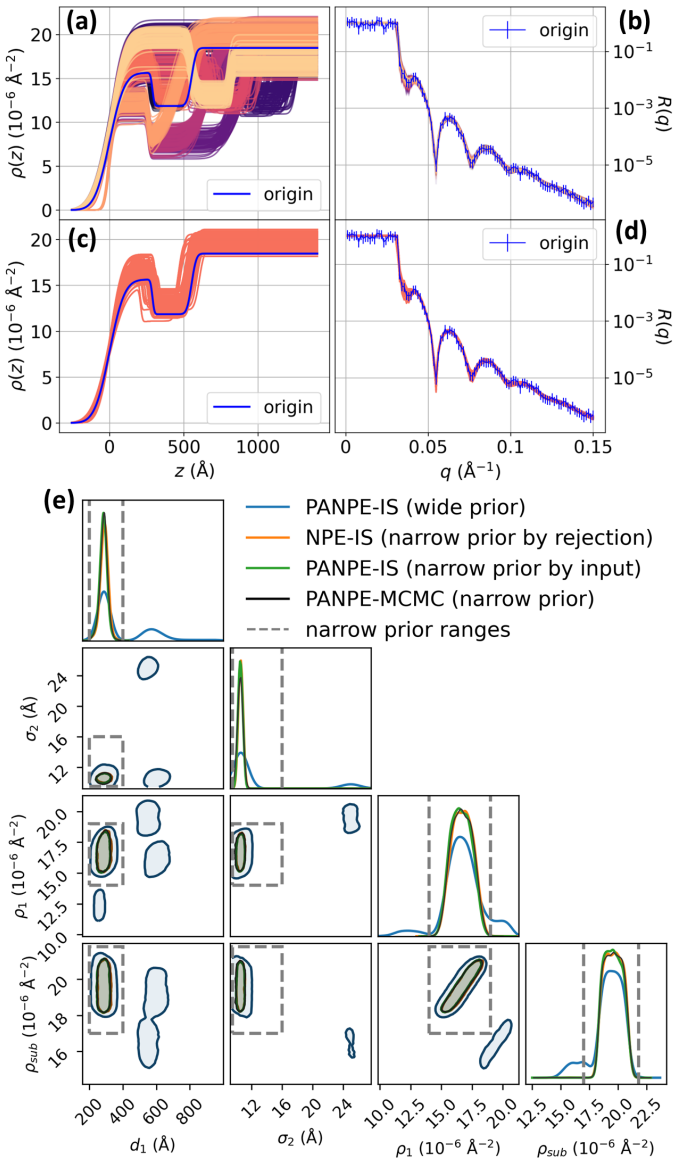


Figure 6.1: Results obtained by different methods on a simulated reflectivity curve with 10 open parameters for a two-layered structure. Our PANPE model produces results in accordance with provided prior information, resolving multiple solutions (a) for a "wide" prior distribution and isolating a single mode (b) for a "narrow" prior (gray dashed lines). The corresponding reflectivity curves (b) and (d) provide a way to improve and validate the results via importance sampling or MCMC in real time. The corner plot (e) shows the resulting marginalized 4d distributions obtained by different methods based on our model and demonstrates the consistency of the distributions.

$\prod_{j=1}^{n} U(\theta_j^{min}, \theta_j^{max})$. Therefore, same as in the previous chapter, we define $\boldsymbol{\phi}$ as a set of corresponding parameter ranges $\boldsymbol{\theta}$: $\boldsymbol{\phi} = \{\theta_j^{min}, \theta_j^{max} \mid j = 1 \dots n\}$ ($\boldsymbol{\theta} \in \mathbb{R}^n$, $\boldsymbol{\phi} \in \mathbb{R}^{2n}$) same as for the previous approach. In this setting, during inference, we can provide wide ranges for the open parameters, or practically fix known parameters by providing narrow ranges. It is worth noting that this approach can be extended to other classes of distributions by providing suitable parametrization.

We note that other parameterizations of priors can be employed if needed. However, the parameterization has to be defined before the costly training procedure, and therefore, it is not as flexible as conventional methods allow. In practice, some of these cases can be addressed in the refinement stage. For instance, the output produced for a uniform distribution can be refined via importance sampling into a truncated Gaussian distribution, at the cost of reduced sample efficiency. Nevertheless, the retraining procedure might be required in specific cases (for instance, in the case of co-dependent parameters).

Furthermore, we define the hyperprior distribution - a distribution of prior parameters $p(\boldsymbol{\phi})$. We only need to sample from this distribution, so it does not have to be tractable and generally can reflect various physical and practical parameter constraints. In our case, the major constraint applied is related to the interface roughness not exceeding (a fraction of) the thicknesses of the adjacent layers [91]. For a larger amount of layers, an additional constraint that can be applied is the maximal total thickness of the sample that defines the minimal oscillation period in the reflectivity curve and which should be well above the measurement resolution $\Delta q_{res}$. We note that despite any of these potentially nonlinear constraints, the resulting prior distributions remain uniform.

During the training, first, the prior parameters are sampled for each trainable sample ($\boldsymbol{\phi}_i \sim p(\boldsymbol{\phi})$) defining the corresponding prior distribution $p_{\boldsymbol{\phi}_i}(\boldsymbol{\theta})$. Then we sample the training parameter from this specific prior distribution $\boldsymbol{\theta}_i \sim p_{\boldsymbol{\phi}_i}(\boldsymbol{\theta})$ along with the noise $\boldsymbol{s}_i \sim p(\boldsymbol{s} \mid \boldsymbol{\theta}_i)$, and finally calculate the corresponding curve $\boldsymbol{R}_i = R(\boldsymbol{q}, \boldsymbol{\theta}_i, \boldsymbol{s}_i)$.

The resulting curve $\boldsymbol{R}_i$ along with the prior information $\boldsymbol{\phi}_i$ (and in general error bars $\boldsymbol{s}_i$) is provided to the neural network $p_{\boldsymbol{w}}(\boldsymbol{\theta}|\boldsymbol{R}, \boldsymbol{s}, \boldsymbol{\phi})$, which is trained by minimizing the forward KL divergence:

$$L = \mathbb{E}_{p(\boldsymbol{\phi})} \mathbb{E}_{p_{\boldsymbol{\phi}}(\boldsymbol{\theta})} \mathbb{E}_{p(\boldsymbol{s}|\boldsymbol{\theta})} \mathbb{E}_{p(\boldsymbol{R}|\boldsymbol{\theta},\boldsymbol{s})} [-\log(p_{\boldsymbol{w}}(\boldsymbol{\theta}|\boldsymbol{R}, \boldsymbol{s}, \boldsymbol{\phi}))] \tag{6.4}$$

In this manner, the model is trained to approximate posterior distribution for a continuous set of uniform prior distributions within a large parameter space.

This method enables multiple new approaches unavailable in the standard NPE setup. Notably, it enables various strategies for incorporating active learning into the training procedure by dynamically adjusting the hyperprior distribution $p(\boldsymbol{\phi})$ based on the model performance. For instance, areas with data that rapidly change with parameters might require more training samples for correct interpolation. We leave the exploration of this direction for future research. It is worth noting that active learning in the NPE setup would necessitate dynamic adjustments of the prior distribution, which can result in posterior distribution estimation with an inherently undefined prior distribution.

Furthermore, PANPE resolves potential cases of parametrization indeterminacy, i.e. multiple parameters representing the same physical system. In the case of reflectometry, the most common occurrence of such indeterminacy is when the studied sample has less number of layers than the maximum number of layers in the training data. In this case, the same density profile corresponds to infinitely many parameters (for instance, zero thicknesses of arbitrary layers having arbitrary densities, etc), which makes it impossible to perform the analysis. By applying constraints on the parameters through the prior distribution, this problem is easily fixed.

Importantly, as recently highlighted [203], the employed training loss, provided it remains finite during training, ensures the probability mass coverage property of the estimated distribution:

$$\operatorname{supp}(p(\boldsymbol{\theta} \mid \boldsymbol{R}, \boldsymbol{s}, \boldsymbol{\phi})) \subseteq \operatorname{supp}(p_{\boldsymbol{w}}(\boldsymbol{\theta} \mid \boldsymbol{R}, \boldsymbol{s}, \boldsymbol{\phi})), \tag{6.5}$$

where $\text{supp}(p(x)) = \{x \in X : p(x) \neq 0\}$ is the support of the probability density $p(x)$. Therefore, unlike MCMC methods, the trained model is guaranteed to resolve all the solutions within the provided prior distribution $p_{\boldsymbol{\phi}}(\boldsymbol{\theta})$. In turn, this property enables further refinement of the obtained posterior estimation $p_w(\boldsymbol{\theta} \mid \boldsymbol{R}, \boldsymbol{s}, \boldsymbol{\phi})$ via the classical self-normalized importance sampling technique (see Subsec. 3.2.1).

### 6.3.2 Training data

The settings used to train PANPE are mostly the same as those discussed in Chap. 5 with some notable differences. We estimate parameters $\boldsymbol{\theta}$ that include thicknesses, roughnesses, and densities of slab layers. However, we also estimate parameters related to experimental conditions - momentum transfer misalignment $\Delta q$, and scale misalignment $\Delta I$. It is required for calculating precise likelihood for accurate refinement of the posterior distribution. We show results for $n_{\text{layer}} = 2$ layers on top of the substrate, resulting in 10 open parameters, as well as examples for $n_{\text{layer}} \in [3, 4]$. We use a fixed set of equidistant $q$ points for each model following Ref. [91]. We note though that the importance sampling weights are calculated with the experimental $q$ points, so the interpolation procedure does not influence the final results.

Same as before, we define a noise distribution $p(\boldsymbol{s} \mid \boldsymbol{\theta})$ as a Poisson distribution with an incoming intensity sampled uniformly for each $q$ point such that the standard deviation ranges from 5% to 20% of the reflected intensity (see Subsec. 3.1.1). The resulting error bars can be provided to the neural network as additional input, but in most cases, we find it sufficient to include it during the importance sampling step same as with the $q$ points.

The training samples are generated during the training procedure without repetition. Each model is trained on $N \approx 1.2$ billion samples, which is a significantly larger amount than the one required for the MLE-based models discussed in the previous chapter.

Tab. 6.1 provides total parameter ranges employed for training the presented models. We note that the main tests are conducted with the first model, including

| Model | Params | Layers | $q_{max}$(Å$^{-1}$) | $d$(Å) | $\sigma$(Å) | $\rho(10^{-6}$Å$^{-2})$ |
|---|---|---|---|---|---|---|
| 1 | 10 | 2 | 0.15 | 0-500 | 0-50 | 0-60 |
| 2 | 10 | 2 | 0.15 | 0-1000 | 0-100 | -20-60 |
| 3 | 13 | 3 | 0.2 | 0-500 | 0-50 | 0-60 |
| 4 | 16 | 4 | 0.2 | 0-500 | 0-50 | 0-60 |

Table 6.1: Parameter ranges for the trained models. All the layers share the same ranges (e.g., thicknesses of the first and the second layers for the first model range in $0 - 500$Å).

the performance on the experimental data. Fig. 6.1 is obtained via model 2, and Fig. 6.6c, d demonstrate results for a higher number of layers via models 3 and 4 correspondingly. The ranges of misalignment parameters are shared among the models: $\Delta q \in [-2 \cdot 10^{-3}, 2 \cdot 10^{-3}]$Å$^{-1}$ and $\Delta I \in [-5\%, 5\%]$.

## 6.4 Bayesian inference

### 6.4.1 Input preprocessing

The input to the network is a measured reflectivity curve $R(\boldsymbol{q})$ along with the set of prior parameters $\boldsymbol{\phi} \in \mathbb{R}^{2n}$. The input is preprocessed before being supplied to the network according to the following procedure.

First, the scaling coefficient $k = q_{\text{max}}/q_{\text{exp}}$ is calculated to rescale the $q$ axis of the measured data to match the training $q$ range (see Subsec. 5.2.5). The input prior parameters $\boldsymbol{\phi}$ are scaled according to Eq. 5.1: $R(\boldsymbol{q}_{\textbf{train}}, \boldsymbol{\phi}_{\textbf{scaled}}) = T_k(R(\boldsymbol{q}_{\textbf{exp}}, \boldsymbol{\phi}))$. After scaling the parameters and the $q$ axis, the intensities $R(\boldsymbol{q})$ are processed as shown below:

$$R_{\text{scaled}}(\boldsymbol{q}) = \log_{10}(R(\boldsymbol{q}) + c) \cdot a + b, \tag{6.6}$$

where the parameters $c = 10^{-10}$, $a = 0.1$, and $b = 0.5$ are fixed. The scaled curve is then interpolated to the fixed $q$ discretization employed for the training. Finally, the prior parameters $\boldsymbol{\phi}_{\textbf{scaled}}$ are normalized with respect to the global parameter ranges.

### 6.4.2 Embedding network

Same as in the approach discussed in the previous chapter, we use a convolutional embedding network before supplying the preprocessed input to the flow-based model.

The embedding network is a sequence of 5 blocks, each block containing a 1D convolutional layer, followed by a batch-normalization layer and a GELU activation function. Each convolutional layer features a kernel of size 3, stride= 2, and padding= 1. Consequently, the dimension of the processed reflectometry is (approximately) halved after each layer. The number of channels is doubled in each block, starting from 32 up to 512. Convolutional layers prove very efficient in processing reflectometry curves, but the absolute positions of the features such as the total reflection edge might be lost due to the translational equivariance of the convolution operation. To enable precise determination of such parameters as misalignment $\Delta q$, the latent representation is concatenated with the preprocessed input curve. The result is further processed by a fully-connected neural network. Finally, the preprocessed parameters $\boldsymbol{\phi}$ are also added to the latent representation by concatenation.

The final latent representation of the input is supplied to the flow-based model. We note that it is possible to pretrain the embedding network using an encoder-decoder architecture. Our preliminary tests suggest that this approach can speed up the training process, which can be relevant for practical applications.

### 6.4.3 Flow-based model

Flow-based models discussed in Sec. 4.6 use a series of reversible and differentiable transformations on a simple, *base distribution* - in our case, the standard normal distribution - with the intent of generating a complex distribution that can be efficiently evaluated and sampled from. In this work, we employ a series of $n$ transformations ($n$ ranging from 30 to 40 depending on a model), each transformation block being a composition of a coupling layer with monotonic rational-quadratic splines [183] and a batch normalization layer [205]. After each transformation block, the parameters are randomly permuted. Furthermore, we introduce a new transformation for incor-

porating the dynamically set priors directly into the flow-based model architecture. We achieve that by applying affine transformation that depends on the prior information $\boldsymbol{\phi}$, essentially compressing the generated parameters into the input-defined range. This invention greatly increases the accuracy of the model and accelerates the training procedure.

### 6.4.4 Generating samples

During inference, we supply our model with a measured reflectivity curve coupled with the corresponding prior parameters $\boldsymbol{\phi}$ preprocessed as discussed above and generate parameters in batches with the corresponding log probabilities. The obtained curves are used for calculating importance sampling weights (PANPE-IS) and streaming estimation of sample efficiency $\epsilon_{\text{eff}}$. We continue this procedure until the effective sample size $n_{\text{eff}} = n_{\text{total}} \cdot \epsilon_{\text{eff}}$ reaches an adequate threshold which we set equal to $n_{\text{eff}} = 500$. The same criterion is employed for the traditional importance sampling from the prior distribution performed for validating the results.

## 6.5 Conventional Bayesian methods

In this work, we compare sample efficiencies between our model and the conventional Bayesian methods. As discussed in Sec. 6.1, in the case of reflectometry, the efficiency of gradient-free methods can be characterized by the sample efficiency of Monte Carlo (MC) sampling from the prior distribution.

### 6.5.1 Estimating low sample efficiencies

There are two possible approaches for estimating sample efficiency for the traditional Monte Carlo method. The first approach is the use of importance sampling weights via direct sampling from the prior distribution $p(\theta)$:

$$\epsilon_{\text{eff}} = \frac{\left(\overline{w_i}\right)^2}{\left(\overline{w_i^2}\right)} \ , \tag{6.7}$$

where $w_i = p(\boldsymbol{R} \mid \boldsymbol{\theta}_i)$, $\boldsymbol{\theta}_i \sim p(\theta)$. An accurate estimation requires $n_{\text{total}} = \text{ESS}/\epsilon_{\text{eff}}$ samples, e.g. sample efficiency $\epsilon_{\text{eff}} = 10^{-12}$ requires more than $10^{12}$ samples, which is computationally unfeasible. An insufficient number of samples $n_{\text{total}}$ only provides an upper bound $\epsilon_{\text{eff}} < 1/n_{\text{total}}$.

The alternative approach employs the analytical form [199]:

$$\epsilon_{\text{eff}} \xrightarrow{a.s.} \epsilon^*_{\text{eff}} = \left( \mathbb{E}_{\boldsymbol{\theta} \sim p(\boldsymbol{\theta}|\boldsymbol{R})} \left[ \frac{p(\boldsymbol{\theta} \mid \boldsymbol{R})}{p(\boldsymbol{\theta})} \right] \right)^{-1} = \frac{p(\boldsymbol{R})}{\mathbb{E}_{\boldsymbol{\theta} \sim p(\boldsymbol{\theta}|\boldsymbol{R})} \left[ p(\boldsymbol{R} \mid \boldsymbol{\theta}) \right]} \ . \tag{6.8}$$

In the case of the uniform prior distribution $p(\boldsymbol{\theta})$, the equation simplifies to

$$\left( \mathbb{E}_{\boldsymbol{\theta} \sim p(\boldsymbol{\theta}|\boldsymbol{R})} \left[ \frac{p(\boldsymbol{\theta} \mid \boldsymbol{R})}{p(\boldsymbol{\theta})} \right] \right)^{-1} = \frac{v(p(\boldsymbol{\theta} \mid \boldsymbol{R}))}{v(p(\boldsymbol{\theta}))} \ , \tag{6.9}$$

where the quantity $v(p(\boldsymbol{x})) \equiv \left( \mathbb{E}_{p(\boldsymbol{x})} \left[ p(\boldsymbol{x}) \right] \right)^{-1}$ can be interpreted as an "efficient volume" of the distribution $p(\boldsymbol{x})$. In this way, $v(p(\boldsymbol{\theta})) = \prod_{j=1}^{n} (\theta_j^{max} - \theta_j^{min}) = \Theta$ is the volume of the prior distribution, and

$$v(p(\theta \mid \boldsymbol{R})) = \left( \int_{\Theta} p(\boldsymbol{\theta} \mid \boldsymbol{R})^2 d\boldsymbol{\theta} \right)^{-1} \tag{6.10}$$

characterizes the efficient volume of the posterior distribution.

We estimate $\epsilon^*_{\text{eff}}$ using samples from our model $\{\boldsymbol{\theta}_i\}_{i=1}^N$ via importance sampling:

$$\epsilon^*_{\text{eff}} = \frac{p(\boldsymbol{R})}{\mathbb{E}_{\boldsymbol{\theta} \sim p(\boldsymbol{\theta}|\boldsymbol{R})} \left[ p(\boldsymbol{R} \mid \boldsymbol{\theta}) \right]} \approx \frac{\left( \sum_{i=1}^N w_i \right)^2}{\sum_{i=1}^N w_i p(\boldsymbol{R} \mid \boldsymbol{\theta}_i)} \ , \tag{6.11}$$

where the importance weights and samples provided by normalizing flows should not be confused with the weights and samples from prior distribution in Eq. 6.7.

In this way, we obtain $\epsilon_{\text{MC}}$ estimations in the case of low sample efficiency of the traditional Monte Carlo method. However, when the prior distribution is narrow enough, we can estimate $\epsilon_{\text{MC}}$ using both methods independently. Fig. 6.2 illustrates the consistency between these two approaches.

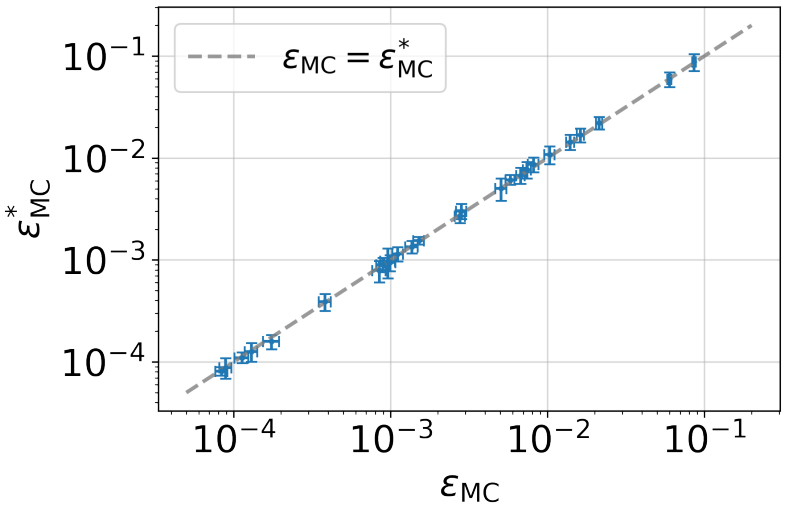


Figure 6.2: The consistency between estimating $\epsilon_{\mathrm{MC}}$ via importance sampling weights (as per Eq. 6.7) and $\epsilon^*_{\mathrm{MC}}$ via our model (as per Eq. 6.11) calculated for samples with sufficiently high efficiency $\epsilon_{\mathrm{MC}}$.

Generally, the volume in Eq. 6.10 can also be employed for estimating the statistical association between the parameters as an alternative to the common mutual information metric:

$$r = \frac{\prod_{j=1}^{n} v(p_j)}{v(p)}, \tag{6.12}$$

where $p_j$ are marginal distributions, and $r$ is the ratio between the volume of the distribution without parameter correlations and the actual volume. We do not calculate the quantity $r$, as normalizing flows do not provide marginal densities directly, necessitating the use of kernel estimations. Nonetheless, we assume that in the case of reflectivity data, high parameter correlations result in high $r$ and low sample efficiency $\epsilon_{MC}$, contributing to the complexity of the posterior estimation task.

### 6.5.2 Streaming calculations

To perform Monte Carlo estimations, we generate samples from the prior distribution in batches with their size being constrained by memory limitations. In our case, $n_{\mathrm{batch}} = 2^{17} \approx 1.3 \cdot 10^5$. In the case of low sample efficiency, a large number of

parameters required is becoming a practical issue. For instance, some of the experimental curves analyzed in this work via Monte Carlo importance sampling required $n_{\text{eff}}/\epsilon_{\text{eff}} = 500/10^{-6} = 5 \cdot 10^8$ samples, which is already problematic to store and process. To this end, we employ streaming calculations of integrals and marginal distributions. It is achieved by updating calculated integrals based on every newly generated batch $\{\boldsymbol{\theta}\}_{j=0}^{n_{\text{batch}}}$. For marginal distribution, we define a binarization of each marginal distribution of interest and calculate Monte Carlo estimations of integrals over each bin. This scheme is implemented in PyTorch and addresses various numerical issues that arise due to low probabilities, requiring the storage of log-probabilities processed through special numerically stable functions provided by the PyTorch library, such as *logsumexp* and *logaddexp*.

### 6.5.3 Employed MCMC methods

As an alternative to importance sampling, we also show results of combining our model with MCMC methods. In this way, the samples generated by the flow-based model are initialization points for the Affine Invariant MCMC algorithm [133]. We employ the differential evolution-based MCMC [200] which is more efficient on multimodal distributions compared to the stretch move-based algorithm used by default in the standard libraries [83, 134]. However, even in the case of this algorithm, our tests suggest that careful usage is necessary. Insufficiently long chains may produce highly biased results, where the proportion of samples on each isolated mode is primarily determined by the proportion of initialized points, rather than the true probability density.

To enable real-time PANPE-MCMC, we implement vectorized GPU-accelerated PyTorch implementations of several Affine Invariant MCMC algorithms. Compared to the basic CPU implementation of MCMC [134] and the reflectivity simulations [83], our code reduces the computational time for running MCMC from 30 minutes [83] to several seconds given the same settings.

The samples generated by PANPE can be used as effective initialization points for MCMC (PANPE-MCMC). The consistency between the solutions obtained by PANPE-IS and PANPE-MCMC attests that the analyzed peaks are fully covered by the model-generated distribution. If the actual peaks have heavier tails, MCMC would reveal that, displaying inconsistency with the importance sampling result. However, in the case of multimodal distributions, PANPE-IS results are generally more reliable than those produced by PANPE-MCMC due to the issues discussed above.

## 6.6 Results

### 6.6.1 Performance on experimental data

We test our model by estimating sample efficiency on both simulated and experimental reflectivity curves. For experimental data, we use 104 curves from two *in situ* XRR measurements of diindenoperylene (DIP) growth on a silicon substrate with a silicon oxide layer analyzed previously via conventional fit [206]. For each experimental curve, we set uniform priors based on a physical understanding of the experiment, in accordance with conventional analysis. The parameters of the known silicon substrate and the oxide layer are essentially fixed by designating narrow ranges. In contrast, the parameters for the thickness $d_1$, roughness $\sigma_1$, and density $\rho_1$ of the growing organic layer have broader prior ranges due to uncertainty. Furthermore, the misalignment parameters $\Delta I$ and $\Delta q$ are set to be open. In this manner, our PANPE model trained on two-layered structures with 10 parameters enables analysis on a practically smaller number of parameters, others being effectively constrained.

The defined prior distributions are sufficiently narrow to carry out conventional Bayesian analysis for validation. Fig. 6.5 shows the time-dependent marginal distributions for three open parameters obtained both by our PANPE-IS model (on the left) and traditional MC method (on the right), exhibiting equivalent solutions. We highlight that both methods rely on the importance sampling technique and only

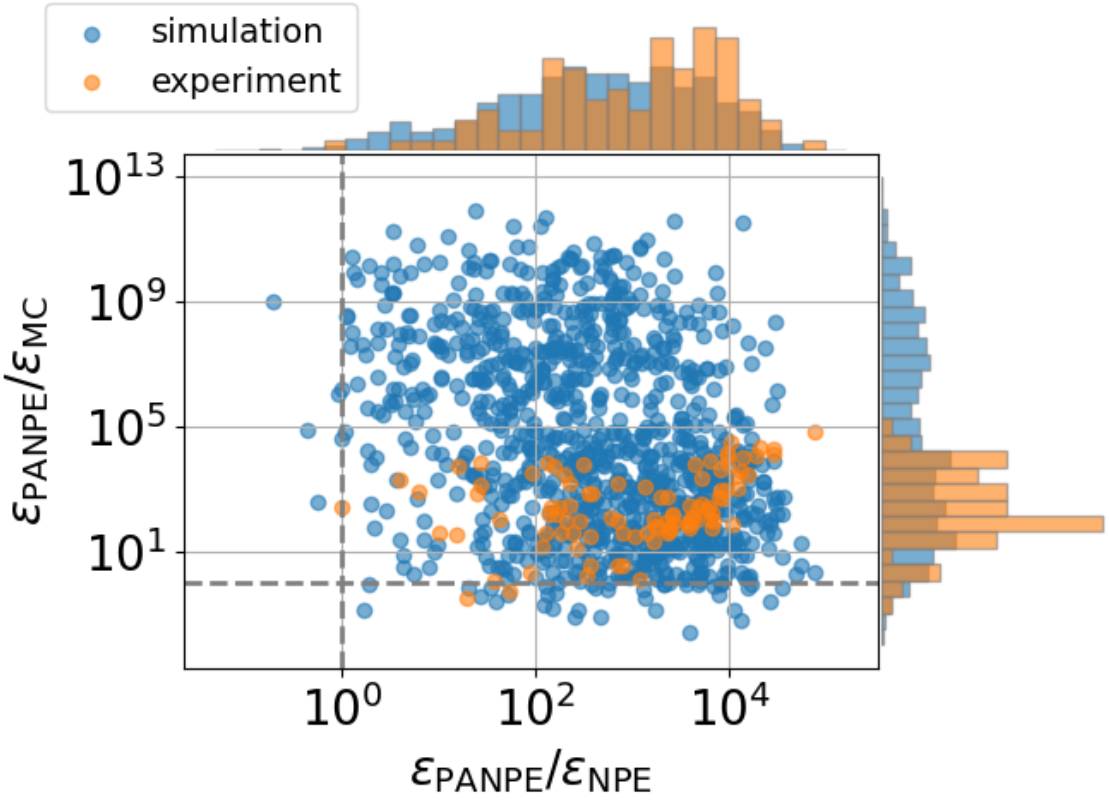


Figure 6.3: Sample efficiency comparison for a test dataset composed of 1000 simulated curves (blue) and 104 experimental XRR curves (orange), contrasting proposal distributions obtained via standard NPE model (denoted as $\epsilon_{\mathrm{NPE}}$), our prior-aware model PANPE ($\epsilon_{\mathrm{PANPE}}$), and via Monte Carlo (MC) sampling from a sample-dependent prior distribution, i.e. by standard importance sampling ($\epsilon_{\mathrm{MC}}$). Small $\epsilon_{\mathrm{MC}}$ values emerge from the relative narrowness of the posterior distributions in comparison to the prior distributions (see also Fig. 6.4). Such narrow multimodal distributions highlight the significant complexity of reflectivity analysis in general and explain the large efficiency gains $\epsilon_{\mathrm{PANPE}}/\epsilon_{\mathrm{MC}}$ up to $10^{12}$ that PANPE exhibits over conventional methods. When compared to NPE, efficiency gains reaching up to $10^{5}$ arise from distributional modes exterior to the sample-dependent prior distributions, consequently resulting in numerous rejected samples. Given these priors, our PANPE model confines its sampling within their corresponding domains.

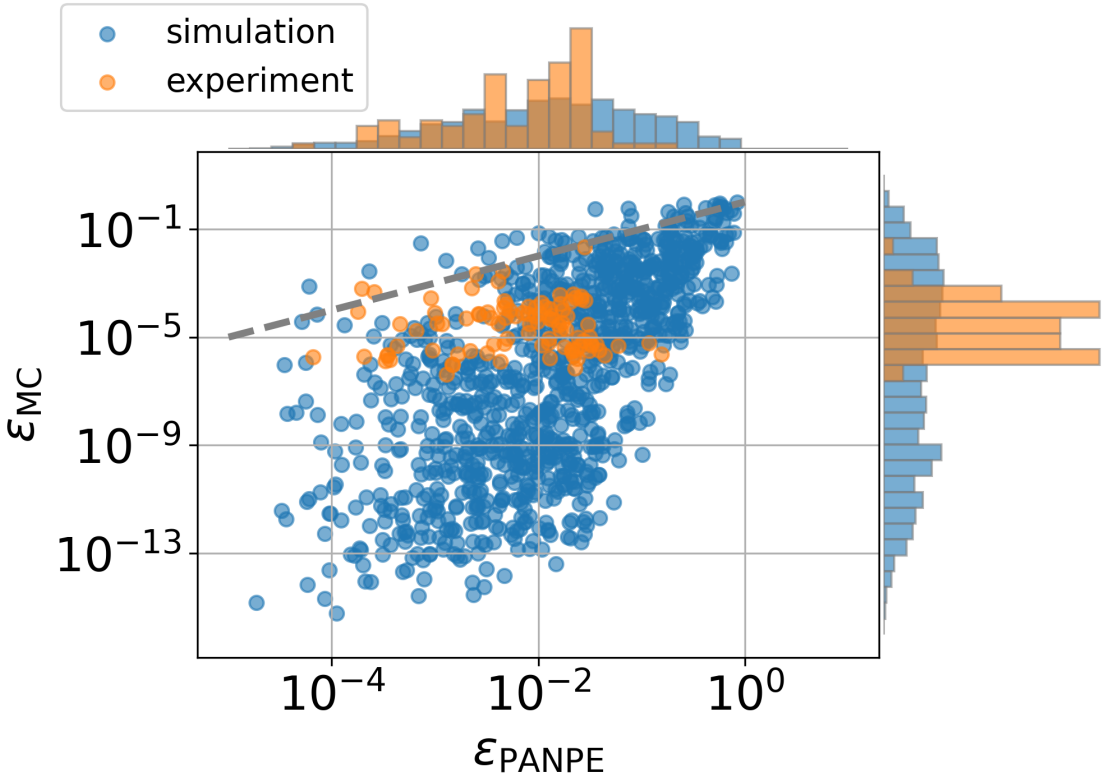


Figure 6.4: Estimated sample efficiencies for the PANPE model (horizontal axis $\epsilon_{\mathrm{PANPE}} \in [10^{-4}, 1]$) and the conventional Monte Carlo (MC) method (vertical axis $\epsilon_{\mathrm{MC}} \in [10^{-14}, 1]$)) for the experimental dataset (orange dots) and simulated dataset (blue dots) from Figure 6.3. Dashed gray line corresponds to $\epsilon_{\mathrm{PANPE}} = \epsilon_{\mathrm{MC}}$. Sample efficiency for MC equals the ratio of density volumes $v(p(\boldsymbol{\theta} \mid \boldsymbol{R}))/v(p(\boldsymbol{\theta}))$ and can be considered as a measure of the complexity of the task. The model efficiency $\epsilon_{\mathrm{PANPE}}$ is affected by this value and correlates with $\epsilon_{\mathrm{MC}}$, but exhibits significantly higher efficiency (see Figure 6.3).

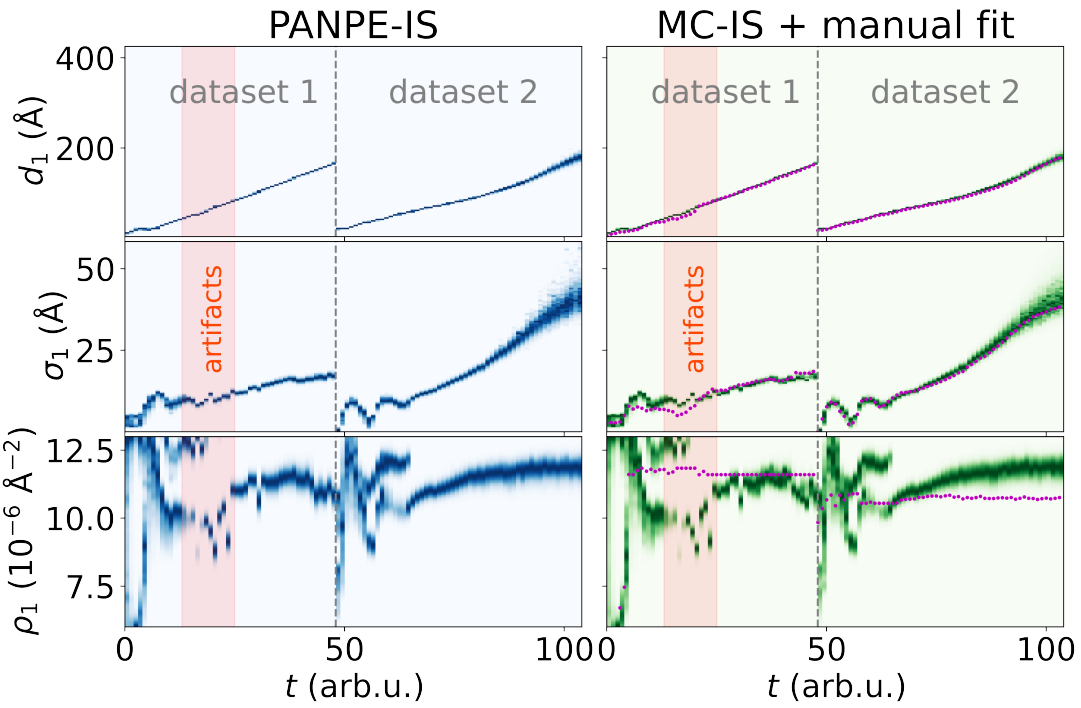


Figure 6.5: Marginal distributions of the thickness $d_1$, roughness $\sigma_1$, and density $\rho_1$ of the diindenoperylene (DIP) growth on a silicon substrate for two *in situ* experimental XRR datasets obtained by our model (on the left) and the traditional Bayesian analysis (on the right). The colors designate normalized probability densities. Additionally, purple dots correspond to conventional manual fits performed via differential evolution. Fig. 6.9 shows time-dependent sample efficiencies and the log evidence estimations for both methods.

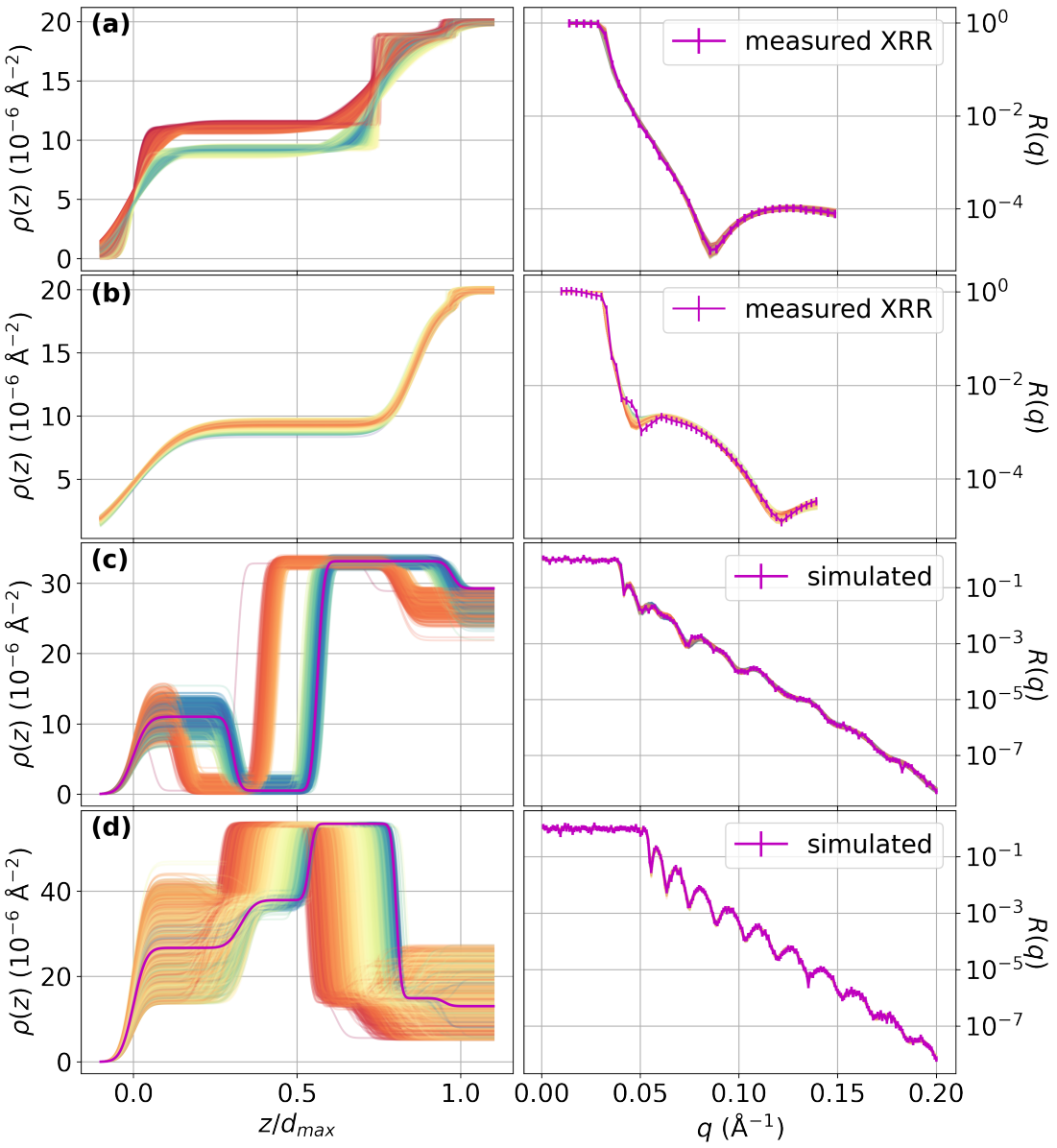


Figure 6.6: Examples of PANPE-IS results for experimental XRR data with two-layered structures on top of a substrate (a, b) and simulated curves with (c) three- and (d) four-layered structures. SLD profiles on the left-hand side provided by PANPE-IS and the corresponding colors produce the reflectivity curves on the right-hand side, that match the input curve (making most of the curves visually indistinguishable). Curve (a) features two solutions for top layer density (see also Fig. 6.8), the lower one being physically correct as concluded based on the time-resolved solution (see Fig. 6.5). Curve (b) features experimental artifact around $q = 0.05$ Å$^{-1}$ not reflected in the training data. Nevertheless, refined by importance sampling, the model provides a reliable result (see also Fig. 6.7).

differ in *proposal distributions*. In the case of MC, it is the prior distribution that is the only source of information for conventional Bayesian analysis. On the other hand, the neural network relies both on information about the prior distribution and the measured data due to the learned connection between the reflectivity curves and the parameters, resulting in significantly more efficient proposal distributions.

We note that despite the relative simplicity of the dataset due to a small number of open parameters, the resulting distributions feature two solution branches for the density parameter (see also Fig. 6.6a). The upper branches disappear beyond a certain time point for each dataset, suggesting that the physical solution corresponds to the lower branches. The conventional fits via differential evolution (indicated by purple dots in Fig. 6.5) are influenced by ambiguity and deviate from the correct solution, underscoring the relevance of probabilistic methods even in such straightforward cases.

### 6.6.2 Out-of-distribution samples

Indeed, the probability mass coverage property of the trained model is not guaranteed to hold in the case of out-of-distribution samples. Unfortunately, sample efficiency generally does not provide any information that might indicate potentially missed distributional modes, limiting the validation of the method. The only certainty is the absence of such cases in the training data ($1.2 \cdot 10^9$ samples) due to finite training loss. Generally, detecting out-of-distribution samples is a relevant emerging direction in ML [207]. As previously reported [203], the low sample efficiency of the model resulting from heavy tails of the proposal distribution potentially indicates an out-of-distribution event. Indeed, the definition of "low efficiency" is not rigorously defined and shall be compared against the distribution of efficiencies on the in-distribution samples. In our case, this distribution ranges from $10^{-4}$ to 1. as demonstrated in Fig. 6.4 (horizontal axis). We note that the correlation between the model efficiency and the "task complexity" quantified as $v(p(\boldsymbol{\theta} \mid \boldsymbol{R}))/v(p(\boldsymbol{\theta}))$ has to be taken into account as well. In other words, low sample efficiency in the case of simple scenarios

with a small number of open parameters is a strong indication of out-of-distribution samples, while the same low efficiency in the case of a higher level of uncertainty might correspond to the in-distribution event.

In the case of the experimental data presented in this work, we validate the results via the conventional MC method that guarantees resolving all the distributional modes. We note that some of the examined experimental curves from the first dataset feature experimental artifacts around $q = 0.05$ Å$^{-1}$ unaccounted for in the simulated training data (see Fig. 6.6b), i.e. are out-of-distribution samples. Nevertheless, refined by either importance sampling or MCMC (see Fig. 6.7), the model provides a reliable and consistent result matching the distribution and the log evidence estimation obtained by MC, as shown in Fig. 6.9.

### 6.6.3 Efficiency gain over conventional methods

Unlike the estimated probabilities, sample efficiencies between PANPE and conventional MC differ significantly in favor of the machine learning solution, median sample efficiencies being equal to $\epsilon_{\text{PANPE}} = 9.4 \cdot 10^{-3}$ and $\epsilon_{\text{MC}} = 3.1 \cdot 10^{-5}$ correspondingly (see Fig. 6.9). GPU implementation of reflectivity calculations results in practically real-time ($\mathcal{O}$(minute)) importance sampling refinement of the PANPE solutions for efficiencies down to $\epsilon_{\text{eff}} = 10^{-4}$. In the case of conventional analysis, sample efficiency on some of the experimental curves go below $\epsilon_{\text{eff}} = 10^{-6}$, requiring hours of GPU computations for analyzing a simple case with three open parameters. This difference in time required for analysis is best reflected in the efficiency ratio $\epsilon_{\text{PANPE}}/\epsilon_{\text{MC}}$, which reaches $10^5$ on the experimental data, as demonstrated in Fig. 6.3 (vertical axis, orange color). Indeed, it is expected to increase with the increasing complexity of the analyzed data with a larger number of open parameters and wider priors.

To demonstrate that, we test the model on 1000 simulated curves with 10 parameters that include samples with priors significantly wider than in the considered experimental data. The resulting efficiency gain of PANPE over MC $\epsilon_{\text{PANPE}}/\epsilon_{\text{MC}}$

reaches $10^{12}$ (blue color in Fig. 6.3), which is the difference between real-time analysis and computationally unfeasible task.

### 6.6.4 Efficiency gain due to prior information

The considered efficiency gain is the difference between the bare use of prior information (MC) and the additional use of information from the curve shape (PANPE). Additionally, we can study the difference between our prior-aware PANPE model and the standard NPE method that does not obtain prior information. To this end, we train a standard NPE model with the same setup and global parameter ranges (fixed uniform prior distribution) as our PANPE model. The resulting difference in sample efficiencies reflects the ambiguity of reflectivity curves within the global parameter ranges that can be compensated by the use of prior information. Therefore, as expected, the distributions of efficiency ratios $\epsilon_{\mathrm{PANPE}}/\epsilon_{\mathrm{NPE}}$ for the experimental and the simulated dataset are similar and reach $10^5$ gain from the use of prior information (horizontal axis in Fig. 6.3). We note that this is a lower estimate since for some test curves NPE produces most of the samples outside the prior distribution, requiring exceedingly long times for accurate estimation of $\epsilon_{\mathrm{NPE}}$.

## 6.7 Summary

Our proposed Prior Aware Neural Posterior Estimation method combines the capacity of the cutting-edge probabilistic NPE method with some of the flexibility of traditional Bayesian methods that utilize sample-dependent prior information. As a result, it surpasses the capabilities of both approaches and enables obtaining all the solutions within wide high-dimensional prior distributions in real time. Other inverse problems across numerous scientific areas could potentially benefit from our prior-aware approach. In general, the static prior distribution is most suitable for passive observations with a fixed setup and corresponding applications. Conversely, active experiments, even in the case of standardized reflectometry measurements, require dynamically adjusted prior distributions. In this sense, our model is potentially suit-

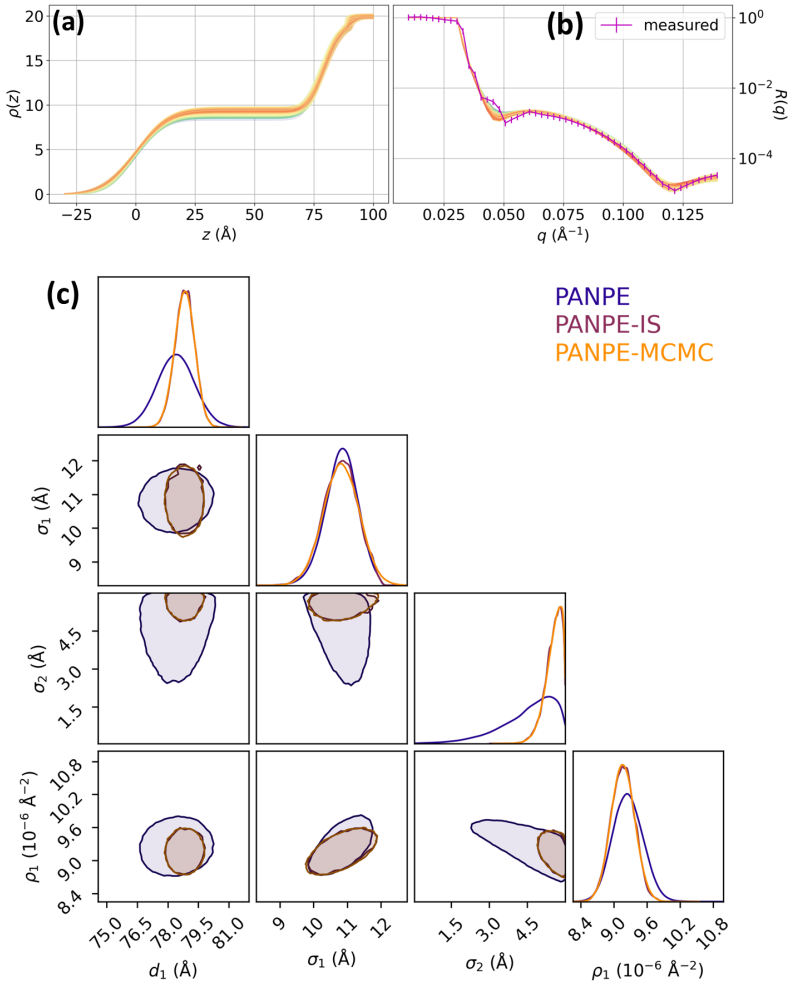


Figure 6.7: PANPE results on an XRR curve with an experimental artifact around $q = 0.05$ Å$^{-1}$. (a) The SLD profiles generated via PANPE-IS and the corresponding (b) reflectivity curves. (c) The marginalized corner plot of 10-dimensional posterior distribution estimated via PANPE model and refined independently via importance sampling and Affine Invariant MCMC [133]. The roughness of the DIP-silicon oxide interface $\sigma_2$ is constrained by providing a narrow prior range $0 - 6$ Å.

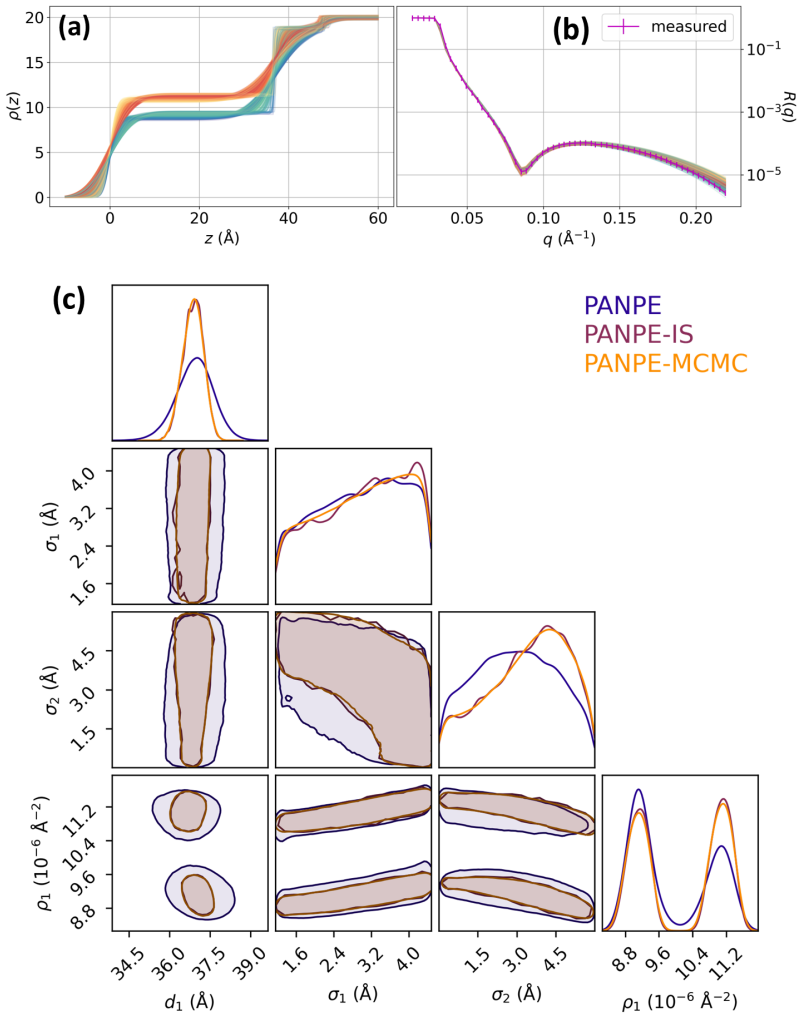


Figure 6.8: PANPE results on an experimental XRR curve featuring two distributional modes. (a) The SLD profiles generated via PANPE-IS and the corresponding (b) reflectivity curves. (c) The marginalized corner plot of 10-dimensional posterior distribution estimated via PANPE model and refined independently via importance sampling and Affine Invariant MCMC [133]. The roughness of the DIP-silicon oxide interface $\sigma_2$ is constrained by providing a narrow prior range $0 - 6$ Å.

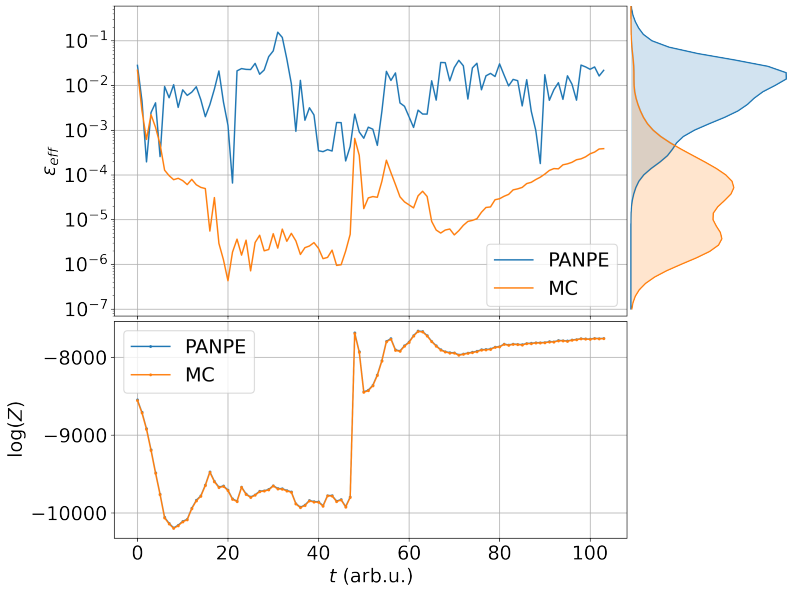


Figure 6.9: Time-resolved evaluation metrics for the studied in situ XRR measurements. The top graph shows the sample efficiencies of proposal distributions acquired by PANPE (represented in blue) and those attained through sampling from the prior distribution (orange). The bottom graph presents the log evidence estimates generated by these respective methods.

able for a wide range of active scientific experiments that permit simulation-based inference.

Applied to reflectometry data, the proposed method allows the acquisition of all possible solutions within a broad prior distribution and for a large set of open parameters - an accomplishment considered to surpass the capabilities of traditional methods. This outcome can guide experimenters by indicating whether the present ambiguity requires additional measurements, thereby directing the experiment until the singular physical solution is determined.

Equipped with our model, we re-examine *in situ* X-ray reflectivity data previously analyzed via a time-consuming conventional fitting procedure [206]. An analysis of more than one hundred curves that previously took days to complete is now accomplished in minutes. Notably, our method not only exposes the consistent inaccuracies of the previous conventional analysis but also brings to light a hitherto unacknowledged ambiguity in the data, unexpected in such straightforward cases

with a small number of open parameters. Our findings suggest a need to reevaluate the anticipated frequency of ambiguous solutions and further underscore the importance of Bayesian analysis for reliable reflectometry experiments, enabled by our method.

The combination of normalizing flows with conventional methods offers a more accurate, refined posterior estimation. The cutting-edge implementation of these employed algorithms enables real-time Bayesian analysis. The main issue that remains, constraining the reliability of the method, as well as virtually all ML-based solutions, is the potential failure of the model on out-of-distribution samples. Nonetheless, the model has shown accurate results on the examined experimental data, including out-of-distribution samples that feature experimental artifacts. We note that currently there are no alternative solutions that offer the same level of reliability for Bayesian analysis in reflectometry data on a large number of parameters.

Overall, our pioneering method sets new standards for reflectometry analysis by reliably and rapidly providing the exact posterior distribution of the physical parameters, even in complex experimental scenarios.

# 7 Deep learning analysis of grazing-incidence wide-angle scattering

## 7.1 Introduction

In the previous chapters, we discussed the reflectometry technique and the methods developed to improve the analysis of reflectivity data. This chapter focuses on GIWAXS and introduces a NN-based pipeline for fast automated analysis of GIWAXS data.

GIWAXS images contain rich and diverse information about the studied sample (see Subsec. 2.2.3). Based on the concrete scientific problem, different types of analysis are required to extract the relevant properties of the studied system. These may include the lattice parameters, the texture, and fractions of co-existing mixtures in powder diffraction, and time-dependent properties of the studied process in case of *in situ* measurements. Due to the large number of related quantities, it can be inefficient to develop separate software solutions for each type of analysis. Since the diffraction peaks contain the most relevant information about the sample in wide-angle geometry, the natural approach is to separate the analysis into an application-agnostic detection of diffraction peaks that determines their positions, sizes, intensities, and other relevant characteristics followed by an application-specific analysis step that relies on the detection results.

Therefore, we identify the peak detection procedure as one of the key bottlenecks on the way to automated GIWAXS analysis. In this way, we separate the analysis into the computer vision task that aims at locating the peak positions, and further processing of the characteristics of detected peaks depending on the specific application.

In the following, we introduce our NN-based peak detector for GIWAXS data. Furthermore, we couple our model with classical algorithms for further processing of the extracted peak characteristics and test it on two applications: 1. the automated phase identification and unit-cell determination of two coexisting phases of Ruddlesden-Popper 2D perovskites, and 2. the fast tracking of $MAPbI_3$ perovskite formation. The results discussed in this chapter are published in Ref. [96, 97].

## 7.2 Deep learning peak detection

### 7.2.1 Data-driven model modifications

We implemented a deep learning object detection model based on a two-stage Faster R-CNN framework [171, 178] to locate diffraction rings and segments in GIWAXS images. Fig. 7.1 illustrates the data pipeline with the model architecture discussed in detail in Subsec. 7.2.3. State-of-the-art object detection algorithms are mainly focused on improving performance on objects in RGB photographs and the corresponding benchmark datasets (PASCAL VOC, MS COCO, Open Images, etc.) [170–172, 179, 208]. X-ray scattering patterns exhibit particular features that require careful adjustments of the existing general methods. To this end, we introduced an array of modifications to the original architecture to conform to the specifics of the scattering data and accelerate the calculations. In the following, we briefly discuss the key features of the grazing-incidence diffraction images that inform the design of the detection algorithm.

**Polar symmetry.** X-ray scattering from crystalline grains results in diffraction patterns showing either full rings or ring segments in reciprocal space (depending on

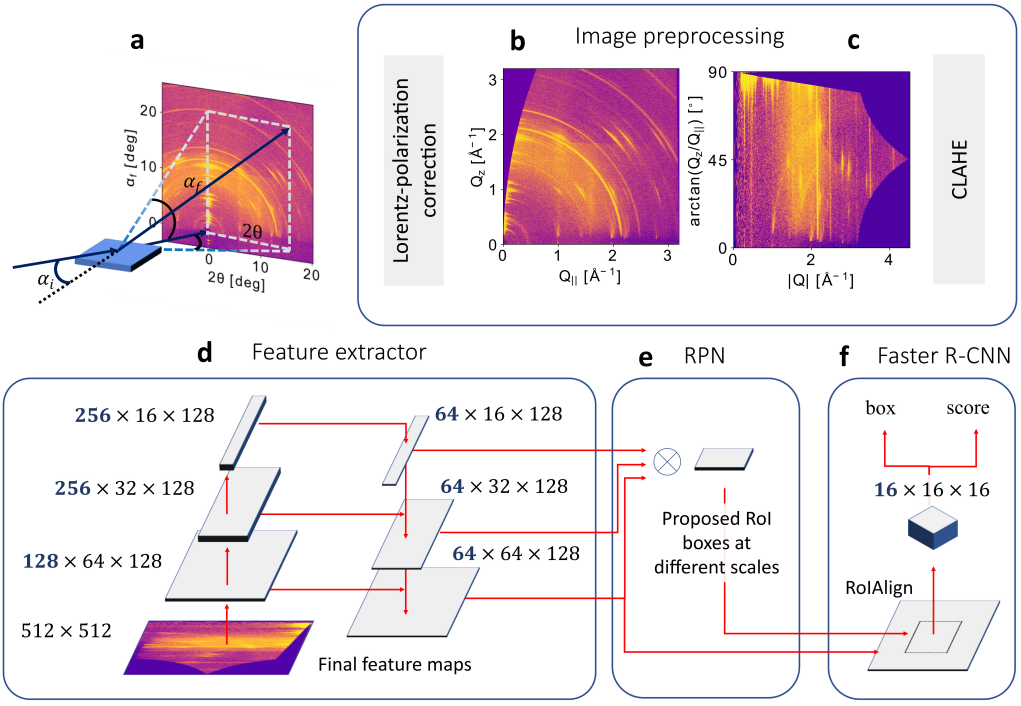


Figure 7.1: Image preprocessing pipeline and the model architecture. (a) The geometry of the measurements. After Lorentz-polarization correction, measured diffraction patterns are sequentially converted from detector coordinates **(a)** to reciprocal space **(b)** and then to polar coordinates **(c)**. For the detection, the contrast is enhanced by CLAHE (all shown images are already contrast-enhanced for visualization). **(d)** Feature extractor with asymmetric feature maps; feature shapes correspond to an input image size of $512 \times 512$ pixels. **(e)** The Region Proposal Network kernels convolve with feature maps to extract RoI at different scales. **(f)** At the second detection stage, a RoIAlign layer extracts features at corresponding positions from the largest feature map, from which the box coordinates and the score of confidence are predicted for each box by a fully-connected network. Figure is adopted from Ref. [96]

their orientation distribution) (see Fig. 7.1b). However, most detection algorithms are designed to detect an object by providing coordinates of a rectangular bounding box around the object. The rectangular shape is most suited for feature extractors that are based on convolution operations with rectangular kernels, while detecting objects with circular shapes would lead to nontrivial complications. To overcome these issues and work with rectangular object shapes, we converted the GIWAXS images to the polar coordinates $|q| = (q_z^2 + q_{||}^2)^{\frac{1}{2}}$ and $q_\phi = \arctan(q_z/q_{||})$, where $q_{||} = (q_x^2 + q_y^2)^{\frac{1}{2}}$. We note that some effects such as refraction may distort the polar symmetry in grazing incidence geometry, but they can be largely neglected here.

**Simple features with complex experimental artifacts.** In general, Bragg peak profiles can be well approximated by a Voigt function or similar profiles (Gaussian, Pearson VII, pseudo-Voigt, etc. [209–211]). At the same time, Bragg peaks are typically influenced by various experimental artifacts: incoherent scattering, scattering from amorphous substrates and gases, counting statistics, the direct beam, detector gaps, etc. In this light, we conclude that a deep learning-based solution is required to filter out artifacts and provide stable detection accuracy. However, in contrast to detecting more complex objects in RGB photographs (e.g. humans, animals, cars), a less deep representation might be sufficient for detecting Bragg reflections. Moreover, we can omit a classifier from the detection algorithm, since there is no task to classify peaks by their shape. These simplifications significantly accelerate the model, which is desirable for on-the-fly analysis of measurements with high acquisition rates (up to tens of kHz on modern detectors [212]).

**Parts of diffraction features can be perceived as complete diffraction features.** In terms of detection, diffraction rings exhibit properties generally unusual for objects in RGB photographs. For example, it is possible that a segment of a ring is detected with a higher confidence score than the whole ring. At the same time, two separate but adjacent segments at the same $|q|$ may be considered a whole ring on a compressed feature map. This behavior causes corresponding detection errors

and requires certain adjustments to the training process that are discussed in Subsec. 7.2.3.

**Asymmetric object shapes at different scales.** Debye-Scherrer rings are typically well-localized in the radial dimension (horizontal in polar coordinates) (see Fig. 7.1c). However, their sizes along the angular (vertical) axis may vary substantially depending on the distribution of orientations of the crystalline grains. Modern deep convolutional feature extractors are designed to decrease the spatial resolution in exchange for richer feature maps that may distinguish between different complex shapes. This strategy would be ill-suited for our task as we would prefer to keep a sufficient resolution along the horizontal $|q|$ axis while compressing the features along the vertical axis. Therefore, we use asymmetric convolutional operations to address the asymmetric shapes of the detected object and to preserve a high resolution along the $|q|$ axis. We also use several feature maps of different shapes to identify peaks at various scales (see Subsec. 7.2.3).

**High dynamic range.** Modern X-ray area detectors can register up to billions of counts per pixel [212] creating exceptionally large dynamic ranges. In practice, such bright spots may correspond to strong reflections from rare large crystalline grains or from the direct beam profile. At the same time, other features may have orders of magnitude less counts, especially at higher $q$ values. This range in contrast makes it challenging for the neural network to equally detect peaks of different brightness. Therefore, we chose to contrast-enhance the experimental images as a preprocessing step.

### 7.2.2 Image preprocessing

Fig. 7.1a-c demonstrate the individual steps of the preprocessing pipeline applied to all the experimental data used in this work. The measured pattern is first corrected by the Lorentz-polarization factor [121], converted to reciprocal space $(q_{||}, q_z)$ using the knowledge about the experimental geometry and mapped to polar coordinates

$(|q|, \arctan(q_z/q_{||}))$ with an image resolution $512 \times 1024$ pixels. The image resolution can be extended for larger area detectors without any adjustments to the method. Finally, the images are contrast-enhanced via the contrast-limited adaptive histogram equalization algorithm (CLAHE) [213]. The contrast-enhanced images are only used for the detection stage, but not for any further analysis.

### 7.2.3 Model architecture

For the detection, we chose a two-stage Faster R-CNN object detection framework [171, 178, 179] and modified it to meet the specifics of the data discussed above. The general pipeline of the detection process is illustrated by the scheme in Fig. 7.1. First, a convolutional neural network (feature extractor) is applied to an image to convert it into several feature maps with reduced spatial resolution that, in general, carry information about the appearance of different objects on an image. During the first detection stage, a Region Proposal Network (RPN) slides over the feature maps via the convolution operation. For each pixel on each feature map, it determines whether there is any object at any of the predefined scales (anchors [171]) by providing objectness score and the box coordinates relative to the anchors. The resulting regions of interest (RoI) with positive objectness score are individually extracted from the feature maps, reshaped and provided to a fully-connected network [172] that classifies an object and refines its box coordinates. For more details on the basic implementation we refer to [171, 178], and below we focus on the key differences determined by the specifics of the data.

**Feature extractor.** We design a feature extractor with asymmetric convolutional layers to preserve resolution along the radial axis $|Q|$. For this purpose, we modify a small ResNet-18 model pre-trained on ImageNet dataset [174] by using asymmetric stride $= (2, 1)$ for the 3 last residual blocks of the model. Fig. 7.2 demonstrates that the use of the pre-trained model considerably speeds up the training process.

Thus, for an input image of $512 \times 512$ pixels, the output feature maps from these 3 blocks would have shapes of $64 \times 128$, $32 \times 128$, and $16 \times 128$, respectively. In this way,

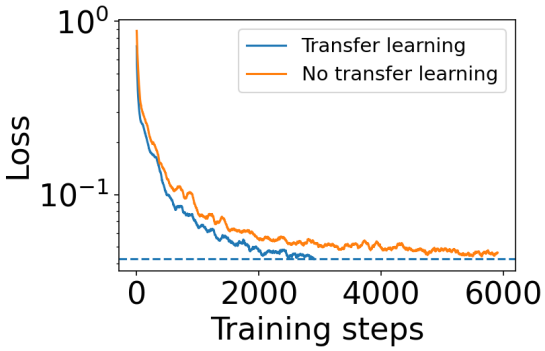


Figure 7.2: The training process for different parameter initializations. The blue curve (transfer learning) corresponds to our model initialized with the publicly available weights of the first layers of the ResNet-18 model pretrained on the classification task on the ImageNet dataset. The orange curve (no transfer learning) corresponds to the standard Kaiming initialization. The transfer learning approach allows to roughly halve the training time (3000 training steps) to achieve the same performance as for the same model with the random initialization (6000 training steps). Figure is adapted from Ref. [96].

image size is mostly reduced in the vertical direction to allow the detection of long narrow rings. At the same time, vertical size reduction from 512 to 16 pixels makes it impossible to resolve segments with small angular (vertical) size. To circumvent this problem, we provide all the 3 feature maps to RPN; each feature map is assigned to extracting objects within a certain size range. As the semantic information is accumulated at the deeper layers of a convolutional network, feature maps from the first blocks provide a shallow representation of an image that in our case might not be sufficient to filter out background and other complex artifacts. To enrich these first feature maps with more complex representation while preserving their resolution, we implement an architecture similar to the Feature Pyramid Network [178]. First, 3 feature maps with sequentially decreasing angular resolution are obtained from the residual blocks as discussed above. After that, smaller and semantically stronger feature maps are summed via upsampling and lateral connections with larger feature maps. To ensure the same number of channels $C = 64$ among the feature maps, the lateral connections are preceded by convolutional layers with $1 \times 1$ kernels.

It is worth noting that the model can be applied on images of arbitrary size; the shapes discussed above correspond to the simulated images with fixed size $512 \times 512$ (see Subsec. 7.2.4), but for the experimental images the resolution might be increased and it generally depends on the initial resolution of the area detector used in the experiment.

**Region Proposal Network.** Our RPN architecture follows the one from [172] with a reduced number of channels $C = 64$. It is applied to each of the 3 feature maps with the corresponding anchors (see Tab. 7.1).

| | Feature map 1 | Feature map 2 | Feature map 3 |
|---|---|---|---|
| Feature map shapes (pixel) | $64 \times 128$ | $32 \times 128$ | $16 \times 128$ |
| Anchor heights (pixel) | 50, 100 | 200, 300 | 400, 500 |

Table 7.1: Feature map shapes and the corresponding anchor shapes for an input image of size $512 \times 512$.

Unlike for usual objects on RGB photographs, there are no well-defined edges of a Gaussian peak; the radial size of a box is arbitrarily defined as a Gaussian RMS width of a peak. However, the surrounding background and overall profile is essential for correct identification of a peak on an image. To address this issue, we extend the target object boxes for RPN by applying padding $w_{RPN} = 1.1 w_{sim} + 1.5$ pixels, so that the box (therefore, the extracted region on the second detection stage) contains the surrounding background.

**Second detection stage.** To extract a proposed RoI from a feature map, we use the RoIAlign layer introduced in [179] that reshapes a RoI from a feature map into a fixed-shape tensor via bilinear interpolation. Compared to the basic implementation, we increase this resulting shape from $7 \times 7$ to $16 \times 16$ to provide a better resolution and, therefore, a better box regression accuracy. At the same time, we decrease the representation size (number of channels) of a RoI from 1024 [172] to 32.

Unlike usual objects in photographs with complex sharp shapes, several distinct segments at the same $|q|$ position may be confused with one pronounced segment,

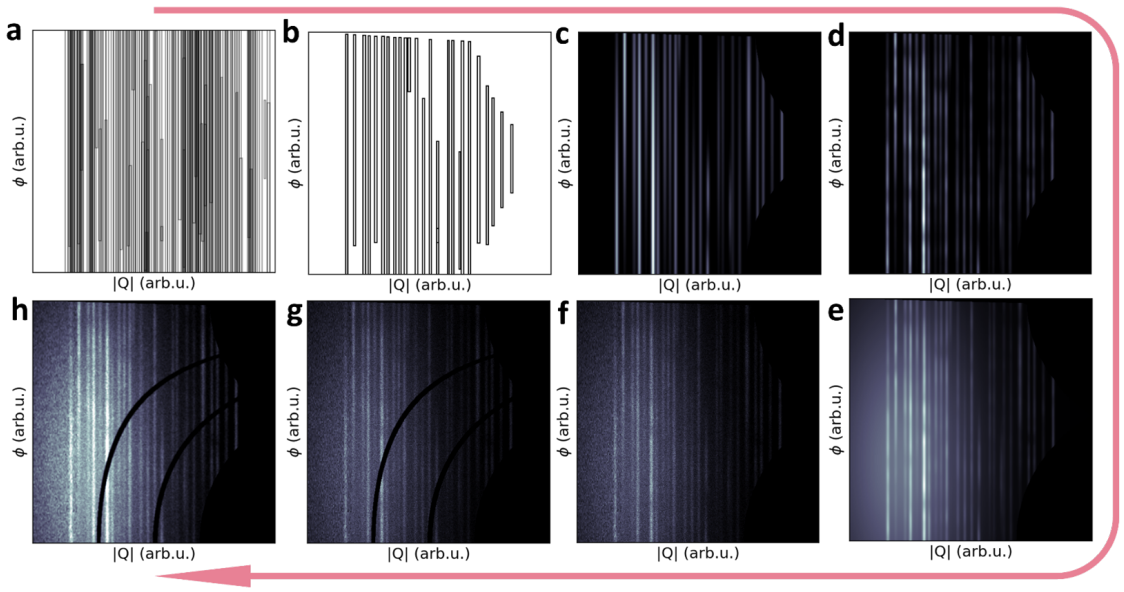


Figure 7.3: Simulation steps for the training data. (a) Generate peak characteristics (positions, sizes, intensities). (b) Filter out peaks in the dark areas and remove strong overlaps. (c) Generate Gaussian profiles corresponding to the peaks. (d) Modulate angular intensity distributions with Perlin noise. (e) Add scattering backgrounds. (f) Apply noise (Poisson, speckle noise, image digitalization). (g) Simulate detector gaps (curved in polar space) and geometry-determined dark areas. (h) Final simulation result (smoothing, contrast correction). Figure is adopted from Ref. [96].

especially if a feature map size is drastically reduced along the angular axis. To reduce the probability of this type of detection errors, we only use the first largest feature map for the second detection stage, i.e. RoiAlign layer extracts all the RPN proposals from the largest feature map (see Fig. 7.1). In this way, the use of the Feature Pyramid Network is essential to provide maximum information for the second detection stage.

We reduce the classifier to 2 classes (object / background) and use the sigmoid loss function instead of cross-entropy loss same as in the RPN classifier. The resulting score of confidence is assigned to refine an objectness score from RPN.

### 7.2.4 Simulated data for the training

Training the model on the experimental data would require enormous efforts in annotating large amounts of the measured data manually. Even then there would be a high risk of overfitting on some particular types of data. We find that simulating the data via simple heuristics is sufficient to achieve a very good performance on various types of experimental data.

Fig. 7.3 illustrates the developed data simulation pipeline. When designing the simulation process, we aim at training the model to identify vertical lines with 2D Gaussian profiles on the images superimposed by different experimental artifacts, levels of noise, and background signals. An image size of a simulation is $512 \times 512$ pixels. Most of the simulation stages described below are optional and are invoked with some probability to cover various combinations of the artifacts and features. Moreover, we define probability distributions for most of the simulation parameters so that they differ for every new simulation. The peak positions are stochastic and model-free; the neural network is trained to detect each of the peaks regardless of the position of the other peaks. In this way, the simulations are not limited to a subset of predefined crystal systems; this approach allows us to analyze various complex mixed systems with the same neural network for peak detection.

Each peak profile is modeled by a two-dimensional Gaussian function. To simulate angular profiles of the peaks, which depend on the orientation distribution of the crystalline grains and may exhibit extremely variable patterns, we multiply a simulated map of 2D-Gaussian peaks by Perlin noise [214]. The simulated background consists of several optional components such as linear background, Perlin noise, and broad Gaussian profiles. A weighted sum of a background and peak profiles map is further modified by applying Poisson noise to simulate counting statistics, adding detector gaps and geometry-dependent dark areas, etc. The simulation process is organized as a series of sequential image processing steps, and each step adds a certain artifact with a defined probability, so in general each image features multiple artifacts and types of background. Finally, simulated images are processed in a similar way as

the experimental ones by applying histogram equalization followed by normalization to the range $I \in [0..1]$. Fig. 7.4a,b show two examples of the simulated images.

### 7.2.5 Training

The model was implemented in the Pytorch deep learning framework [215]. We trained both RPN and Fast R-CNN together for 3000 iterations; on each iteration, we simulated a batch of 16 grayscale images with size 512×512, in total 48000 images. We used the AdamW optimizer (Adam with decoupled weight decay regularization [216]) and halved the learning rate every 500th iteration starting from $lr = 0.002$. The loss function contains 4 components:

$$L = \lambda_1 L_{reg}^{RPN} + \lambda_2 L_{score}^{RPN} + \lambda_3 L_{reg}^{ROI} + \lambda_4 L_{score}^{ROI}, \tag{7.1}$$

where *reg* terms are regression losses for box coordinates calculated as smooth $L1$ loss [172], *score* terms are objectness losses calculated as a sigmoid function, $RPN$ and $ROI$ denote the first and the second detection stages, respectively, and $\lambda_i$ are the weights used to balance these terms; we use $\lambda_1 = \lambda_3 = 10$ and $\lambda_2 = \lambda_4 = 1$.

The model was trained on a training set of 48000 simulated images. The training data is simulated on the fly during the training the same way as for the reflectometry models (see Subsec. 5.2.3). In this way, we did not train a model on the same image twice to eliminate possible overfitting, increase the variability of the data, and omit a validation set. The whole training takes less than 25 minutes on a single NVIDIA 2080Ti graphics card. It is worth noting that, since our model is much lighter (i.e. contains much less parameters) than the original Faster R-CNN implementation, it allows to increase batch size from 2 to 16 images per batch and, therefore, unfreeze and train Batch Normalization layers [172, 217].

In addition to our model, we trained the other three neural networks and evaluated their performance on the simulated data (see Fig. 7.5).

Faster R-CNN model with ResNet-50 and Feature Pyramid Network [178] is trained with 4 images per batch due to memory limitations. Therefore, the training is extended to 12000 iterations.

U-Net [218] is modified by adding Batch Normalization layers and trained to solve a segmentation task with a cross-entropy loss; the separate peak positions are obtained from the segmentation map via the standard algorithm [219]. The confidence score of each peak prediction is estimated by the mean value of the segmentation map within the predicted box. Similar to Faster R-CNN, we use 4 images per batch due to memory limitations and 12000 training iterations in total.

Our Region Proposal Network (a one-stage detector) is trained separately; the training process is equivalent to our two-stage detector.

### 7.2.6 Postprocessing

One disadvantage of the Faster R-CNN architecture is that a single object can be detected several times with overlapping boxes. Thus, after obtaining the box coordinates and scores of confidence, some of the predictions have to be filtered in order to eliminate possible duplicates. In this work, we use a standard non-maximum suppression operation [220] which removes all overlapping boxes except the one with the highest score of confidence if the degree of overlap exceeds an intersection over union metric [172] $IoU = 0.1$. The value of $IoU$ is set quite low to ensure minimal appearance of duplicates, but it can be adjusted if many overlapping peaks are expected on a diffraction pattern.

The other necessary filtering stage that was implemented is the removal of predictions with low confidence score. In this work, we use the threshold value 0.8, though most of the predicted peaks exhibit a confidence score above 0.95. When detecting the peaks in *in situ* data, it is desirable to identify and connect Bragg reflections that appear in sequential time frames. This is achieved by calculating $IoU$ between all the detected peaks from adjacent time frames and connecting those pairs of peaks with largest $IoU$ if it is sufficient (we use an arbitrarily chosen value $IoU = 0.3$;

the results of the algorithm are not sensitive to this parameter). As a result, we obtain a set of reflections with detected positions over time and total duration of the presence of each peak. Based on this duration, we are able to perform an additional filtering procedure by removing those peaks with small duration as they are likely to be related to experimental artifacts or noise.

## 7.3 Performance on the simulated data

The performance of the trained model was first evaluated on a test set of 10000 simulated images. Fig. 7.4a-b show two examples of the simulated patterns. The green lines correspond to the peaks detected by the model. There were 175344 peaks simulated for 10000 images in total, of which 99.22% (173990) were detected correctly, 0.78% (1354) were missed and 0.25% (439) of the detected peaks were false positives. As a performance metric (Fig. 7.4c-d) we chose to simply use the absolute peak center distance $\Delta|q| = ||q_{\text{detected}}| - |q_{\text{truth}}||$ over the more commonly used metric Intersection over Union (IoU) [170–172, 179]. This was done because we deemed the accuracy of the peak center to be more relevant to this task than the peak overlap. Fig. 7.4c shows the distribution of $\Delta|q|$ for 173990 peaks along with the cumulative curve; 95% of the peaks were detected with a $\Delta|q|$ below 0.366 pixels. A stochastic nature of the simulation process (see Subsec. 7.2.4) unavoidably creates a small fraction of peaks that are unrecognizable due to noise, background, and low intensity. Fig. 7.4d illustrates the distributions of falsely detected and missed peaks per image; only 4% of images contained at least one falsely detected peak and 8.5% of images had at least one missed peak. The number of false and missed peaks per image determines the correctness of structure determination. In our case, the vanishingly low numbers of false positive detections per image (0.044 peaks on average) and missed peaks per image (0.14 peaks on average) minimize the corresponding errors in the further analysis of GIWAXS images.

We note that our neural network is lightweight (5.9 M parameters) and fast (122 images per second), which is essential for real-time GIWAXS data analysis.

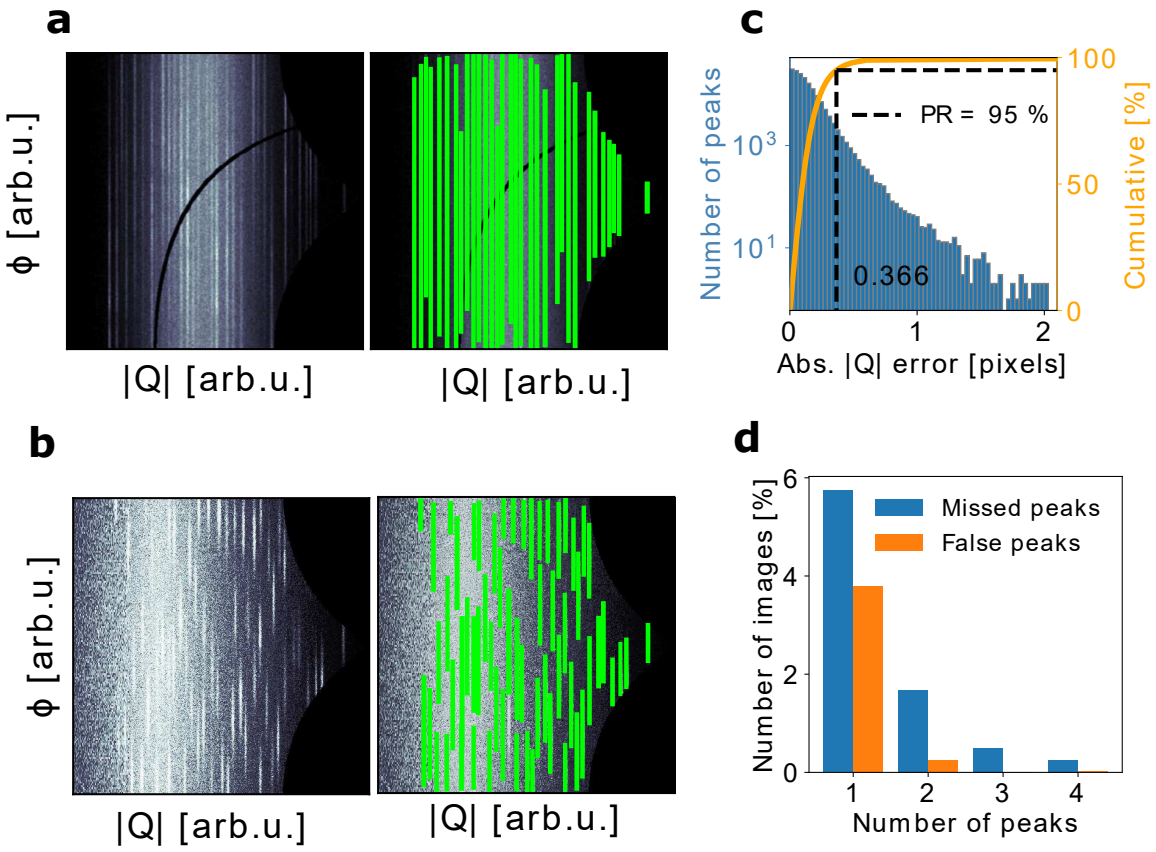


Figure 7.4: Detection results on the simulated test set of 10000 images. **(a)** and **(b)** show two simulated patterns (left) and the corresponding detection predictions (right) in the polar coordinates $|Q| = (Q_z^2 + Q_{||}^2)^{\frac{1}{2}}$ and $\phi = \arctan(Q_z/Q_{||})$. **(c)** The absolute $|Q_{\text{detected}}|$ error distribution with cumulative curve; the error is below 0.366 pixels for 95% of the peaks. **(d)** Distribution of missed and falsely detected peaks per image. Figure is adopted from Ref. [96]

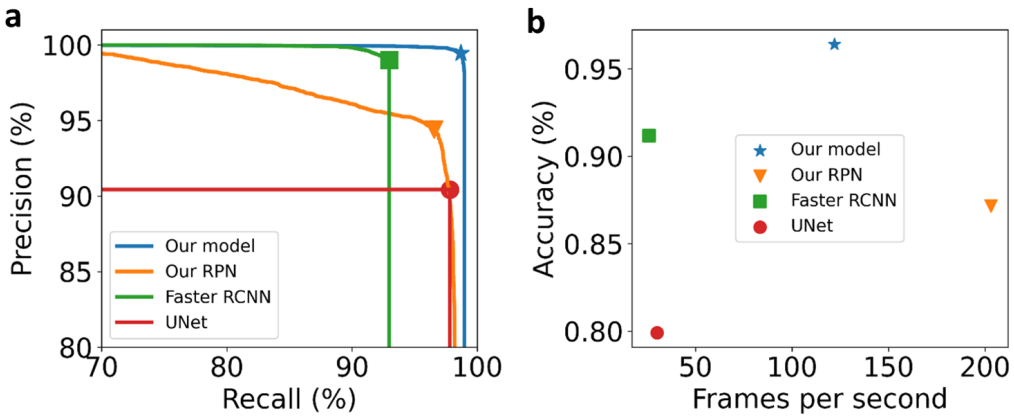


Figure 7.5: Detection performance metrics on the simulated data for our two-stage model, our Region Proposal Network (one-stage detector), the standard Faster R-CNN with ResNet-50 feature extractor and Feature Pyramid Network, and the U-Net model. (a) Precision-recall curves for different models obtained by varying levels of confidence scores and calculating the standard object detection metrics. Markers denote the best detection accuracies on the test set for the studied models. (b) Detection accuracies and speed (frames per second) for the compared models. The presented two-stage detector is the most accurate among the considered solutions and is sufficiently fast for real-time GIWAXS analysis. Figure is adopted from Ref. [96].

We note that our neural network is lightweight (5.9 M parameters) and fast (122 images per second), which is essential for real-time GIWAXS data analysis. For comparison, the analogous unmodified Faster R-CNN model [178] is substantially larger (41 M parameters) and slower (26 images per second), yet it misses 7% of the peaks on the test set. This result illustrates that our model is highly optimized for the task and outperforms much larger standard architectures. Furthermore, to justify the choice of the object detection algorithm and the importance of the second detection stage, we provide performance comparisons with two other types of object detection models: the one-stage detector and the segmentation model U-Net [218]. The results are summarized in Fig. 7.5. Both models exhibit lower detection accuracy than our two-stage detector. Compared to the one-stage detector, the second detection stage is particularly relevant for filtering out false positive predictions (the share of false positives for the one-stage detector is 5.6%). The segmentation model is indeed simpler in terms of implementation; however, in addition to a higher number of false positives (9.6%), it suffers from other systematic problems illustrated in Fig. 7.6: the overlapping peaks cannot be separated, and single peaks can be detected as multiple peaks; the accuracy of peak $|q|$ determination is limited by pixel size; the process of extracting peak positions from segmentation maps increases the inference time to 30 images per second.

In light of these results, it is important to note that, although GIWAXS data is much harder for humans to interpret than the RGB photographs from typical benchmark object detection datasets, the presented model reached near-optimal performance on the test set. The average precision exceeded 99%, a value which has not yet been reached on datasets with RGB photographs with the current state-of-the-art results being around 60% [221]. These results provide evidence that the developed model architecture is well-suited for the current task. This success inspired us to apply our strategy to specific scientific problems and automatize experimental data processing, as explained below.

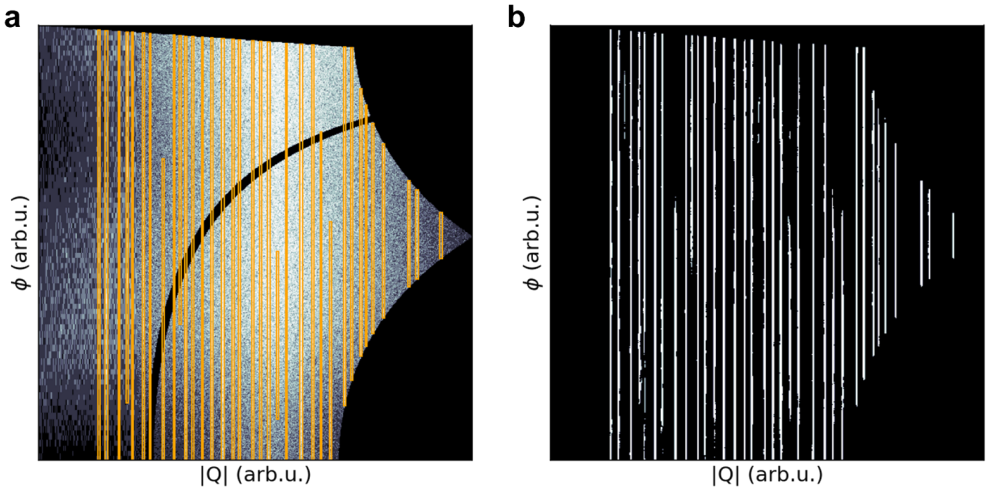


Figure 7.6: An example of the U-Net binary segmentation map (b) for a simulated GIWAXS image (a) with the ground truth peak positions illustrated by orange boxes. The black pixels on the segmentation map (b) correspond to the background and the white pixels correspond to the detected peaks. Apart from obvious false positive predictions, converting the segmentation map to the separated peak positions introduces additional errors. Such errors occur if a single peak results in multiple nonconnected segmentation areas, or if segmentation areas for several peaks are connected. Figure is adopted from Ref. [96].

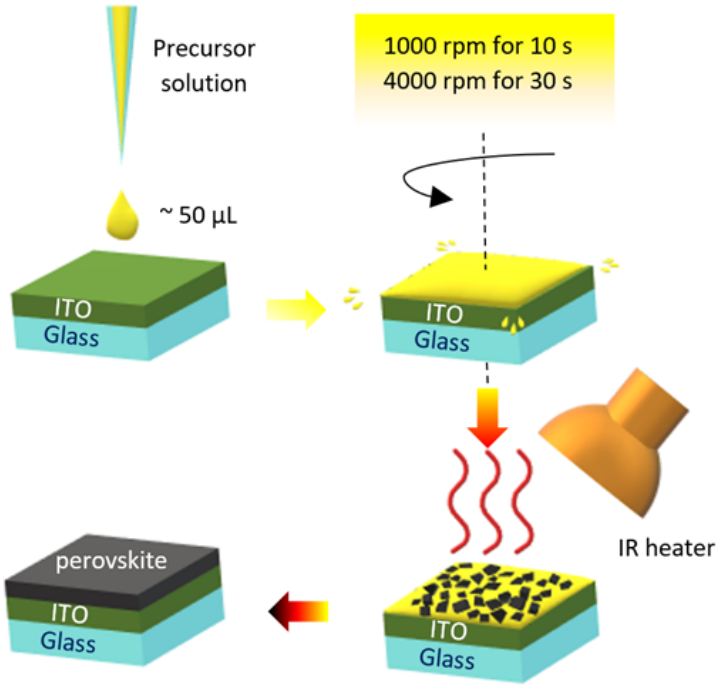


Figure 7.7: Thin perovskite film deposition procedure. 50 µL of precursor solution was spin-coated in 2 stages, 1000 rpm for 10 s and 4000 rpm for 30 s, and subsequently annealed by the IR heater, resulting in a dense perovskite film. Figure is adopted from Ref. [96].

## 7.4 Performance on the experimental data

### 7.4.1 Use case 1: 2D perovskite lattice parameter refinement

In the following, we apply the developed model to analyze data from the *in situ* GIWAXS measurements of the formation of a Ruddlesden–Popper 2D perovskite (butylammonium methylammonium lead iodide $(BA)_2(MA)_{n-1}Pb_nI_{3n+1}$ [222–224]) during annealing.

The experiment was performed at the synchrotron radiation source PETRA III, at beamline P08 [225]. The X-ray beam energy was $E = 18\,\text{keV}$ and the angle of incidence was $\alpha_i = 0.5°$, which is above the critical angle for the substrate $\alpha_c = 0.16°$. The film deposition process is illustrated in Fig. 7.7. The investigated thin film sample was obtained by spin-coating a solution of butylammonium iodide (BAI), methylammonium iodide (MAI) and lead iodide ($PbI_2$) dissolved in a dimethyl formamide

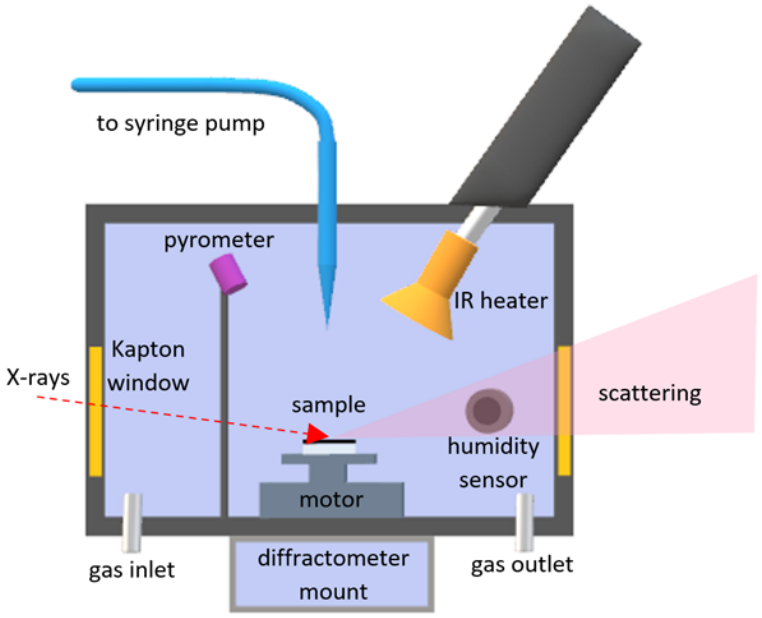


Figure 7.8: Schematic representation of the spin-coating chamber used for in situ GIWAXS measurements with $N_2$-rich atmosphere and measured humidity level. Solution deposition is done by an automated syringe pump, the sample is annealed by an IR heater, and its temperature is measured by a pyrometer. Figure is adopted from Ref. [96].

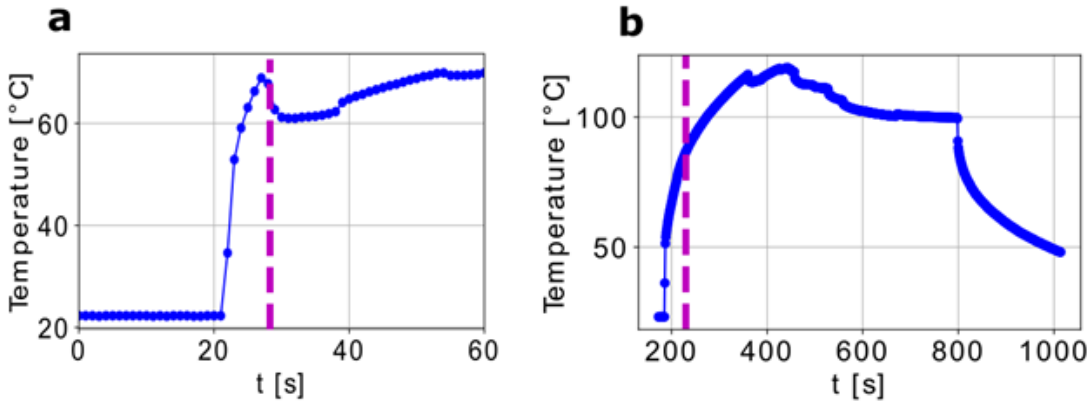


Figure 7.9: The temperature curves of the annealing processes for $(BA)_2(MA)_{n-1}Pb_nI_{3n+1}$ (a) and $MAPbI_3$ (b) measured via a pyrometer. The dashed lines indicate the time of perovskite formation. The changing emissivity of the samples during the measurements can result in increased error bars for the temperature values obtained by the pyrometer. Figure is adopted from Ref. [96].

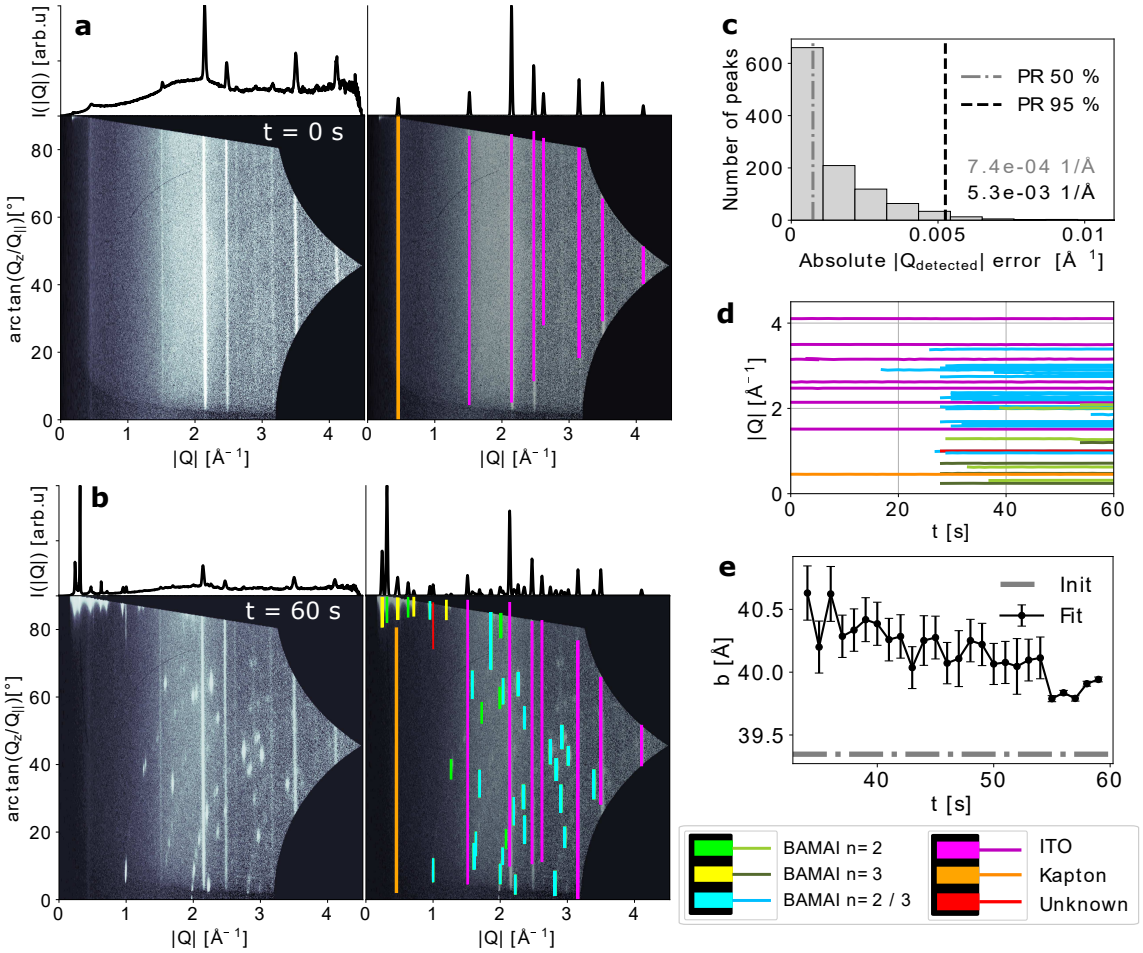


Figure 7.10: Diffraction patterns in **(a)** and **(b)** are annotated with Bragg reflections detected by the machine learning algorithm at $t = 0$ s and $t = 60$ s, respectively. Radial profiles above the images show the initial radial profiles calculated from the diffraction patterns (left images) and the profiles calculated based on the detected peaks (right images). **(c)** The distribution of the absolute error of radial peak positions location by the model with 50% and 95% percentiles. **(d)** The radial positions of the detected peaks over time. **(e)** The evolution of a lattice parameter $b$ over time for $n = 2$ phase of $(BA)_2(MA)_{n-1}Pb_nI_{3n+1}$ (denoted as BAMAI on the legend) perovskite calculated via the positions of the detected peaks. The error bars indicate the standard errors (s.e.m.) of the fit calculated using Hessian matrices. Figure is adopted from Ref. [96].

and dimethyl sulfoxide mixture (DMF:DMSO = 1:4), on a glass substrate coated with indium tin oxide (ITO). An IR lamp mounted at the top of the spin coating chamber was used to anneal the sample (see Fig. 7.8). Fig. 7.9a shows the corresponding time-dependent temperature. Sixty diffraction patterns with 1 s exposure time each were analyzed, corresponding to 60 s of annealing.

Fig. 7.10a-b demonstrate the detection results of the neural network on two diffraction patterns for $t = 0$ s and $t = 60$ s. The lines on the right-hand images correspond to the locations of the detected peaks. Our algorithm detected most of the visible reflections - 1156 diffraction rings and segments from 60 time frames remained after the filtering stage (see Subsec. 7.2.6). The missed reflections typically have low intensities. Some detection inaccuracies in determining peak angular sizes (in particular, lower edges of detected boxes do not cover the whole peaks) may be related to angular profile dissimilarities between the experimental and simulated peaks.

To measure the accuracy of the obtained peak positions, we extracted one-dimensional radial profiles of each peak based on their predicted locations and fitted them with a Gaussian function with linear background via the Levenberg–Marquardt algorithm [226]:

$$I(|q|) = I_0 \exp\left(-\frac{(|q| - q_{\text{fit}})^2}{2w^2}\right) + B|q| + C \tag{7.2}$$

Fig. 7.10c shows the distribution of the absolute error of the detected positions $|q_{\text{detected}}|$ with respect to $q_{\text{fit}}$ obtained by the fit; 95% of the errors are below $5.3 \times 10^{-3}$ Å$^{-1}$. The colors of the peaks in Fig. 7.10a-b denote the result of the indexing algorithm used for the identification of the Bragg reflections (see the legend in Fig. 7.10). We note that our detection algorithm can be coupled with any standard indexing algorithm for diffraction patterns, most (if not all) of which require careful peak position extraction. In this case, it is sufficient to use a simple algorithm for phase identification based on the share of simulated intensities (structure factors) that appear within any of the detected boxes (see Subsec. 7.5.1 for more details).

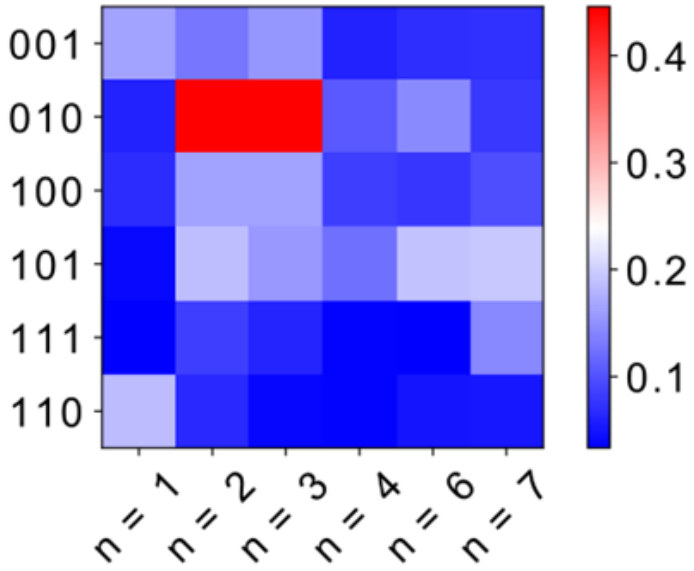


Figure 7.11: Phase matching results. The vertical axis corresponds to different unit cell orientations in the form of miller indices; the horizontal axis denotes different phases of $(BA)_2(MA)_{n-1}Pb_nI_{3n+1}$ 2D perovskite. The colormap shows the matching results, from which two best matches n = 2 and n = 3 with (010) orientation are chosen. Figure is adopted from Ref. [96].

We expect some of the possible phases of $(BA)_2(MA)_{n-1}Pb_nI_{3n+1}$ to emerge during annealing, where $n \in [1, 7]$ defines the perovskite layer thickness [224]. Fig. 7.11 shows that two phases $n = 2$ and $n = 3$ with (010) orientation are identified by our algorithm. The algorithm also identified 7 peaks from the ITO substrate and one peak from the kapton window.

Once the phases are identified, each peak is automatically indexed (one peak remains unidentified and likely belongs to a side product of the annealing). Fig. 7.10d shows radial peak positions over time, from which we can obtain the moment when both perovskite phases emerge ($t \approx 28$ s). One peak was assigned as belonging to both $n = 2$ and $n = 3$ phases and emerging earlier at $t = 17$ s, however, it is in fact a weak reflection from the ITO substrate at $Q = 2.91\,\text{Å}^{-1}$.

Fig. 7.12 demonstrates a systematic change of the relative radial position of a perovskite peak over time that might indicate changes in the lattice structure. To study these changes, we performed the unit cell refinement procedure (see Subsec. 7.5.2) for

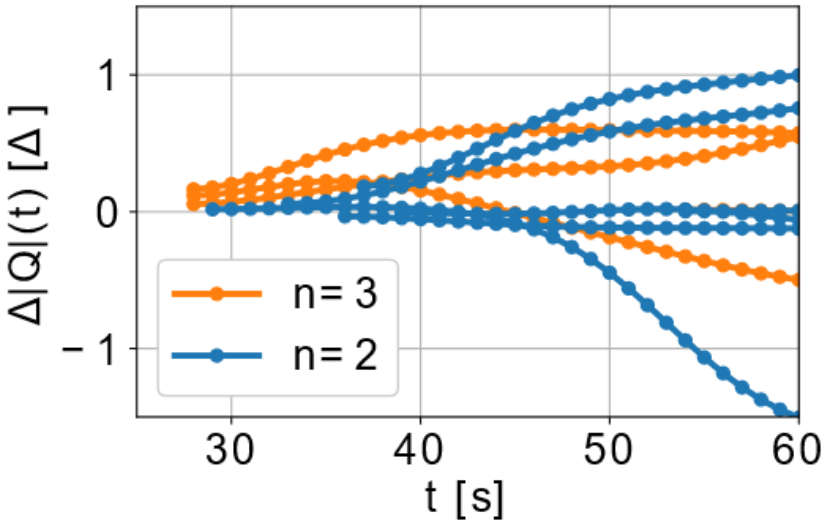


Figure 7.12: The relative peak positions for n=2 and n=3 phases of $(BA)_2(MA)_{n-1}Pb_nI_{3n+1}$ perovskite over time for the use case 1 calculated as $(\frac{|q|(t)}{|q|(t_0)} - 1$. Figure is adopted from Ref. [96].

both phases for each time frame. Fig. 7.10e demonstrates the behavior of the lattice parameter $b$ of $n = 2$ phase that slowly decreases over time from 40.5 Å, which is about 1 Å above the previously reported value [222] that was used as an initial value for the fitting. The lattice length $b$ of the $n = 3$ phase is constant ($b \approx 52.5$ Å) and slightly above the initial value 51.96 Å. This discrepancy is most likely related to the increased temperature during annealing resulting in asymmetric thermal expansion in the direction corresponding to the spacer molecule. As shown in Fig. 7.13, the other parameters remain unchanged and are in agreement with the initial values.

Note that while the demonstrated analysis can be easily reviewed and adjusted by an expert at each step if necessary, in general there is no need for a manual input. This way, our detection model allows to automate the essential steps of GIWAXS data analysis from phase identification and peak indexing to refining the unit cell parameters. Performing the same analysis with conventional tools would require a manual selection and fitting of the peaks for each time frame, which is often so time-consuming that in practice just a small fraction of the obtained reflections is fitted for

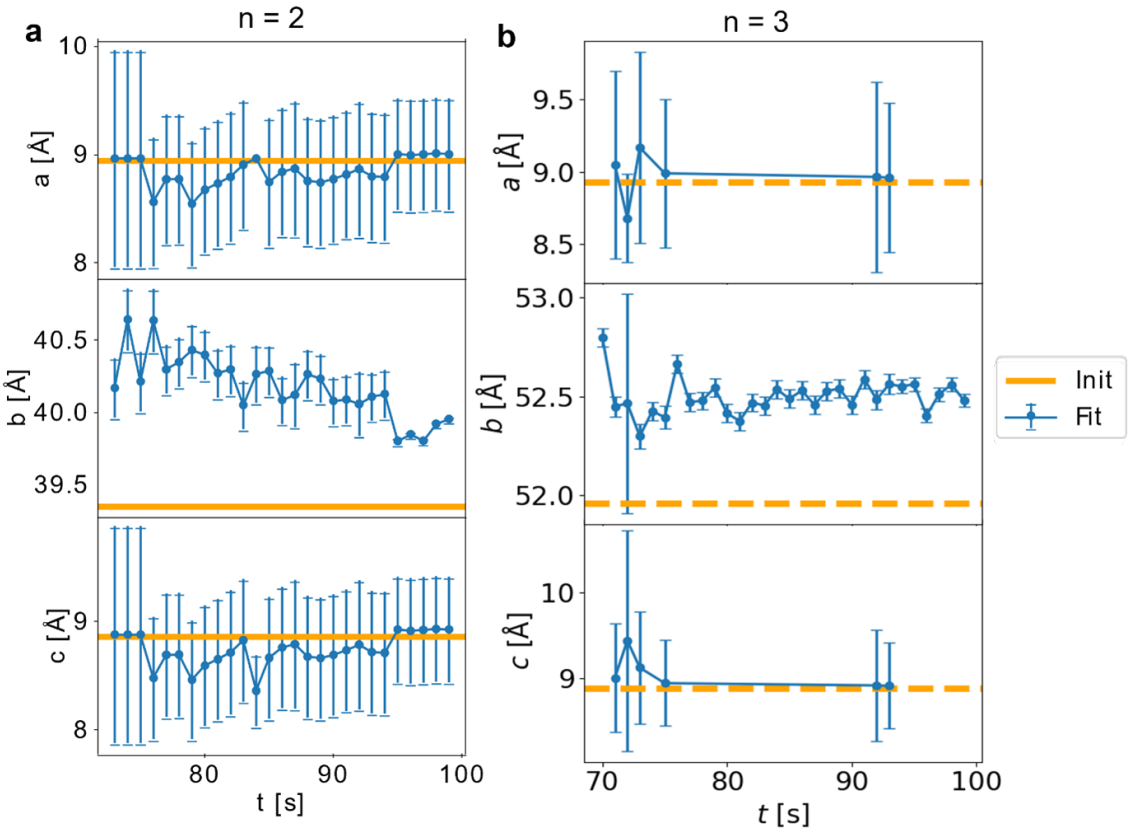


Figure 7.13: Refined lattice parameters for $n = 2$ (a) and $n = 3$ (b) phases of $(BA)_2(MA)_{n-1}Pb_nI_{3n+1}$ perovskite structures based on the positions of the detected diffraction reflections. Horizontal dash lines correspond to the initial fitting values. The missed values for $n = 3$ phase correspond to the time frames without non-overlapping in-plane detected peaks from the n= 3 phase with $q_{||} \neq 0$. The error bars indicate the standard errors (s.e.m.) of the fit calculated using Hessian matrices. Figure is adopted from Ref. [96].

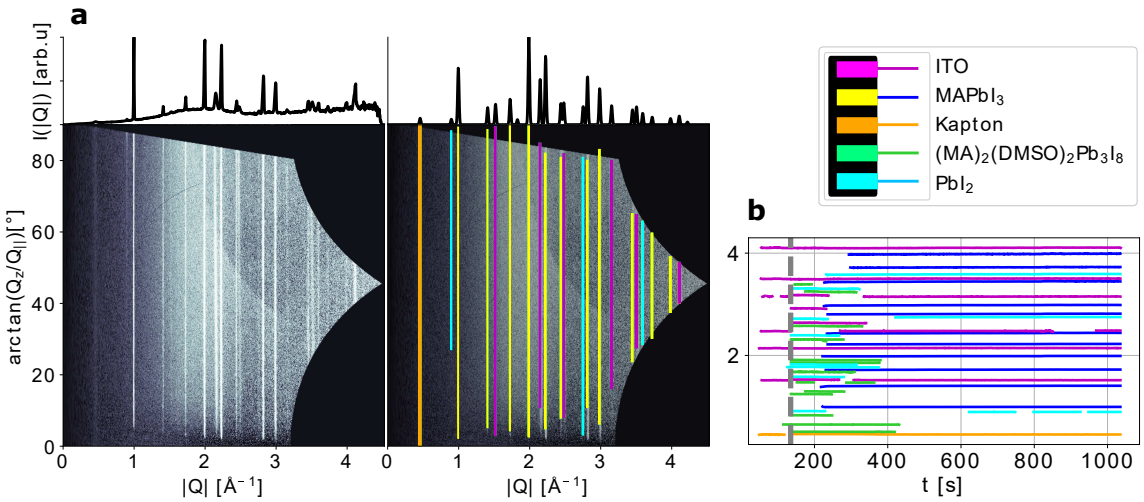


Figure 7.14: Detection results of *in situ* measurements of $MAPbI_3$ formation during annealing. **(a)** The diffraction pattern with the detected peaks at $t = 700$ s. **(b)** The radial positions of the detected peaks over time. The dashed gray line separates the spin-coating and the annealing phases. Figure is adopted from Ref. [96].

further analysis. In contrast, our approach potentially allows a more comprehensive analysis already during the experiment.

### 7.4.2 Use case 2: *in situ* tracking of $MAPbI_3$ perovskite formation

As a second use case for our technique, we analyze 1015 diffraction images from an *in situ* spin-coating and annealing experiment of the formation of 3D $MAPbI_3$ perovskite. The experimental setup was identical to the one described for use case 1. The thin film sample was obtained by spin-coating a solution of MAI and $PbI_2$ dissolved in a mixture of dimethylformamide and dimethyl sulfoxide (DMF:DMSO = 4:1), on a glass substrate coated with ITO.

Fig. 7.14a illustrates the diffraction pattern at $t = 700$ s during annealing (left image) and the corresponding detection results (right image). The model detected all the visible peaks in the pattern, which correspond to the reflections from $MAPbI_3$, the ITO substrate, $PbI_2$, the intermediate structure $(MA)_2(DMSO)_2Pb_3I_8$, and the kapton window. The indexing step was performed in the same way as in the previous case (Subsec. 7.4.1). Fig. 7.14b shows the $|q|$ positions of the detected peaks

over time. In total, there were 20756 peaks left after the filtering stage (see Subsec. 7.2.6). This fast preliminary analysis revealed conversion from the precursor phase $(MA)_2(DMSO)_2Pb_3I_8$ to $MAPbI_3$ perovskite that happens from $t \approx 220$ s to $t \approx 300$ s.

## 7.5 Integration with algorithms for processing detected peaks

### 7.5.1 Phase identification and indexing

For a given crystal structure and orientation, we calculate reflection positions and structure factors for all the combinations of miller indices in a wide range $[-20, 20]$ and determine which reflections appear close enough to the detected peaks based on the following conditions: $||q_{\text{detected}}|-|q_{\text{sim}}||/w_{\text{detected}} \leq 1$ and $|\phi_{\text{detected}}-\phi_{\text{sim}}|/a_{\text{detected}} \leq 1$ where $w_{\text{detected}}, a_{\text{detected}}$ are radial and angular sizes of the detected peak, respectively, and $\phi = \arctan(q_z/q_{||})$. The fraction of the corresponding structure factors gives a metric that appears sufficient for finding the best match among a set of phases and orientations. Since some of the reflections appear at $q_{||} = 0$ ($\phi = 90°$) and their centers are hidden behind a missing wedge, we automatically prolong those peaks that reach the edge of a missing wedge to $\phi = 90°$ to be matched correctly.

The minimal number of detected peaks required to identify the structure correctly depends on the choice of an indexing algorithm. In general, the employed matching algorithm might require substantially more detected peaks to robustly identify the correct structure compared to standard indexing algorithms that may only require three peaks per structure. Nevertheless, the number of peaks detected by the model on the experimental datasets is sufficient to determine every presented structure with the matching algorithm correctly.

When there are both reflections from crystalline grains with preferred orientations (short segments in angular dimension) and long rings from random orientations, it is beneficial to separate the predictions to these two groups and index

separately. We do so for the analysis of 2D perovskite data by clustering predicted peaks into 2 groups based on the relative angular size of peaks $a_{\text{detected}}/H(q_{\text{detected}})$, where $H(q_{\text{detected}})$ is the maximum possible angular size at a given $|q|$ position. In this case, the reflections from ITO and kapton are separated from the reflections from 2D perovskites and they are indexed separately.

The indexing stage is rather trivial and it is performed by assigning the closest simulated reflection to each detected peak given the reflection is close enough (see the conditions above) for each of the phases. Thus, one peak may correspond to several overlapping reflections from different phases (see Fig. 7.10).

### 7.5.2 Unit cell refinement

Lattice parameters of the identified $n = 2$ and $n = 3$ phases of the $(\text{BA})_2(\text{MA})_{n-1}\text{Pb}_n\text{I}_{3n+1}$ 2D perovskite are refined based on the positions of the detected and indexed Bragg reflections. We achieve this by minimizing the mean squared distance between the detected and simulated peak positions with respect to the lattice parameters. Some of the reflections from $n = 2$ and $n = 3$ phases overlap and are detected as single peaks. Indeed, the corresponding detected peak positions are, in general, biased. To achieve better accuracy of the lattice parameters determination, we only use those $N$ peaks that do not overlap with peaks from another phase. The refined lattice parameters are obtained by minimizing

$$\chi^2_{N-3}(a,b,c) = \sum_{i=1}^{N} \left( \frac{Q^i_{\text{detected}} - Q^i_{\text{sim}}(a,b,c,h_i,k_i,l_i)}{\sigma^i_{\text{detected}}} \right)^2 \tag{7.3}$$

with respect to the lattice parameters $a, b, c$ for an orthorhombic unit cell (angles $\alpha = \beta = \gamma = \pi/2$) for each time frame via L-BFGS-B algorithm [227]; the Miller indices $\{h_i, k_i, l_i\}$ of an $i^{\text{th}}$ peak are known from the indexing results. The initial values $a = 8.947\,\text{Å}, b = 39.347\,\text{Å}, c = 8.8589\,\text{Å}$ for $n = 2$ and $a = 8.928\,\text{Å}, b = 51.959\,\text{Å}, c = 8.878\,\text{Å}$ for $n = 3$ are taken from the previously reported structures [223]. The standard errors are calculated using inverse Hessian matrices; detection errors are majorized by $\sigma^i_{\text{detected}} = 0.01\,\text{Å}^{-1}$.

## 7.6 *gixi* package

The introduced model is published as an open-source package called *gixi* [228]. *gixi* is described in Ref. [97] and implements the deep learning model discussed in this chapter. The package consists of a server-side and a client-side application that can be used separately. The client is a lightweight PyQt5-based graphical user interface (GUI) designed for visualizing both the raw images and the processed results, as well as for starting the server jobs on a cluster for real-time image processing. Fig. 7.15 shows a screenshot of the client program.

The overall scheme of the server-side pipeline is illustrated in Fig. 7.16. The server provides all the image processing and data analysis operations from preprocessing the raw data to peak detection and saving the results. The server operations require a fast implementation with multiprocessing and CUDA support to enable real-time analysis. Most synchrotron facilities provide users with powerful computer clusters and the corresponding IT infrastructure to accelerate data processing and simplify data storage. Clusters allow massive parallelization of demanding processing procedures via multiple CPU cores, GPUs, and multiple cluster nodes. *gixi* supports CUDA through PyTorch and optional multiprocessing to accelerate the data pipeline.

Our implementation allows straightforward integration into any cluster infrastructure based on the commonly-used Slurm Workload Manager system. So far, the package has been successfully employed for processing large amounts of GIWAXS data at two different synchrotron facilities: 1) the PETRA III X-ray radiation source at the Deutsches Elektronen-Synchrotron (DESY) and 2) the ESRF. We emphasize, however, that the package can be effortlessly adapted for use at other facilities. The modular structure of the software allows further facility-specific integration, especially direct connections to detector data streams that do not involve disk storage.

We note that the package is intended to be updated and does not yet represent a comprehensive tool for GIWAXS data analysis. Different types of analysis based on the extracted peak characteristics are yet to be implemented in the package (both on the client and on the server side), which is a well-defined but tedious programming

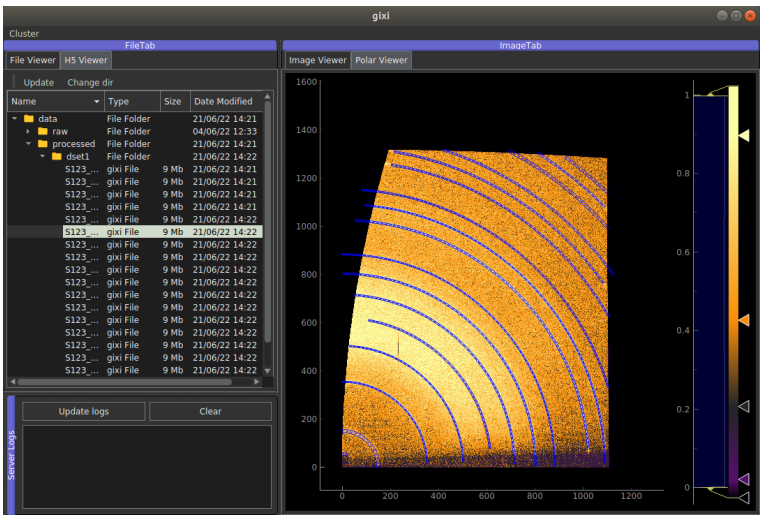


Figure 7.15: A screenshot of the main window of the client-side application that includes the file viewer, the image viewer, and the cluster logs. Figure is adopted from Ref. [97].

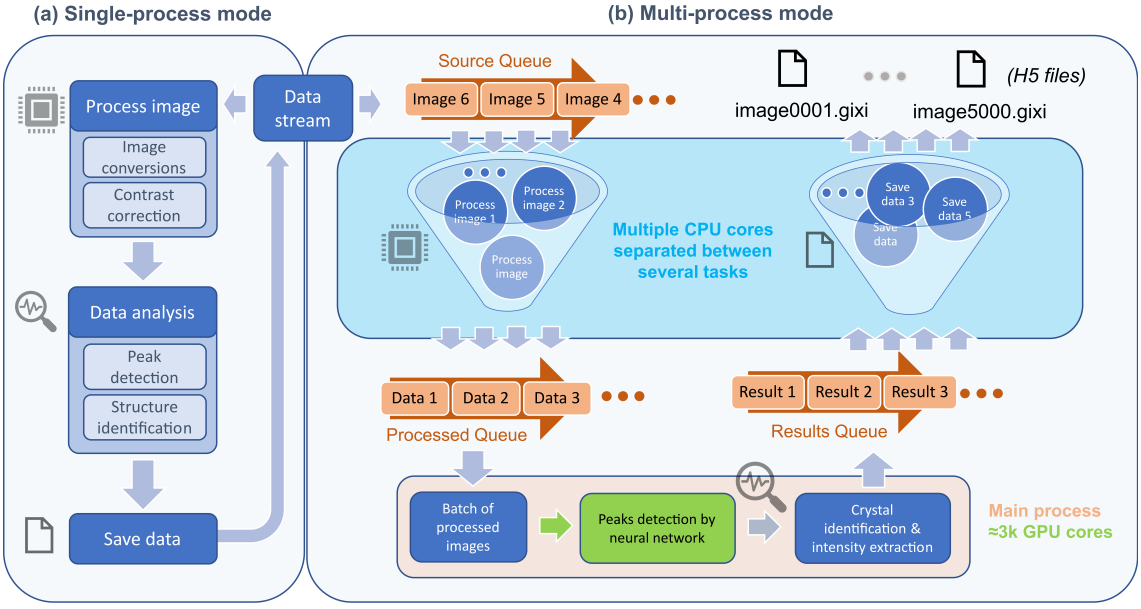


Figure 7.16: A schematic illustration of (a) the single-process and (b) the multi-process pipeline modes of the server-side application. Figure is adopted from Ref. [97].

task. However, the current implementation and tests [97] demonstrate the viability of the designed solution, including the seamless integration with IT ecosystems of synchrotron facilities.

## 7.7 Summary

We demonstrated an automated deep learning approach for the analysis of GIWAXS images. We showed how to optimize modern object detection algorithms to address various properties of the experimental data, such as the specific geometry, strong background, asymmetry of the diffraction features, etc. These improvements enabled high performance on diverse experimental data with accurate detection of most of the reflections. The small absolute error of the radial peak position determination and high detection accuracy made it possible to perform an in-depth analysis of *in situ* GIWAXS data. By processing the extracted feature positions, we identified the formation of two coexisting phases of 2D organic-inorganic perovskite structure and traced the evolution of the lattice parameters in time for these phases. The refinement procedure revealed a slight decrease in the unit cell size for one of the identified phases. We note that such subtle processes may often be overlooked during a manual analysis of the data.

Currently, there is an increasing interest in automated analysis of diffraction data [34, 39, 229–233]. An often proposed solution is a single analysis step where the measured raw data is provided to the machine learning algorithm which is supposed to learn which parts of the data carry information, how to perform the desired analysis and obtain material-related results. This approach needs to be tailored as a whole to variations in materials or experimental setups and does not allow to understand or control the process of such an analysis. In contrast, the more flexible analysis approach presented here consists of modular and transparent parts that are simple to control and adjust for a concrete task.

In general, our solution allows to substantially accelerate the analysis process of GIWAXS images, potentially boosting the speed of scientific discoveries in material

science and organic photovoltaics. The other possible applications of the method include the real-time adjustment of the experimental conditions based on the obtained GIWAXS data and the high-throughput screening of possible perovskite compositions for organic solar cells.

# 8 Conclusion and outlook

This work introduces deep learning solutions for the analysis of two types of surface scattering data: reflectivity curves measured in specular geometry and diffraction images from grazing-incidence wide-angle scattering measurements.

## 8.1 Specular reflectometry

Reflectometry analysis is a challenging task that can greatly benefit from modern deep learning techniques. Due to the phase problem, multiple potential solutions lead to ambiguity in data interpretation, necessitating the use of prior knowledge in the analysis pipeline. Chap. 5 and Chap. 6 introduce two novel prior-aware deep learning methods for the two major statistical frameworks commonly employed in reflectometry data analysis: Maximum Likelihood Estimation and Bayesian inference.

### 8.1.1 Maximum Likelihood Estimation

Chap. 5 introduces a deep learning analog of the conventional "Parratt fit". The method extends the existing machine learning solutions in the Maximum Likelihood Estimation framework by incorporating prior knowledge in the form of dynamic parameter ranges into the deep learning model, as opposed to the previous approaches with the predefined static ranges. By addressing the issue of ambiguity, this approach significantly broadens the applicability of the method from single-layer structures with a few open parameters to more complex multilayer structures. The model is tested on scenarios involving up to 17 open parameters and provides instantaneous Maximum Likelihood Estimation within the used-defined parameter ranges with high

accuracy given sufficient prior information. The introduced physics-informed parameterization further expands the applicability of the method, accommodating crystal structures that produce Bragg peaks and time-evolving structures with parameters that change with $q$ along a single reflectivity curve. The solution is successfully employed for a series of unprecedented closed-loop autonomous *in situ* XRR experiments at a synchrotron facility, where the use of prior information allows connecting the results from sequential measurements. The PyTorch implementation of GPU-accelerated reflectivity simulations via Abelès formalism enables seamless integration into the deep learning pipeline. This facilitates rapid training through direct use of the data generation process and safeguards the model against overfitting. Due to the scaling invariance in reflectivity, the model can be applied to data with various $q$ ranges.

The limitations of this approach, and more broadly the Maximum Likelihood Estimation framework, are discussed. The major fundamental flaw of estimating a single parameter is the risk of overlooking other potential physical solutions within the defined prior ranges, thereby reducing the reliability of both the analysis and the experimental technique as a whole. By design, poor prediction quality from the introduced model may signal the presence of ambiguous solutions, but accurate prediction does not guarantee the uniqueness of the found solution. Therefore, the model, like all other methods for Maximum Likelihood Estimation, should be applied with this caveat in mind. Extremely fast inference helps address this issue through various possible strategies relying on exploring subspaces of the provided parameter ranges. Generally, the proposed method, with its low training and inference time combined with high accuracy given sufficient prior information, makes for a particularly relevant and practical solution for automated reflectometry analysis.

### 8.1.2 Bayesian analysis

Chap. 6 further explores the reflectometry analysis and focuses on the Bayesian framework. Bayesian analysis enables reliable reflectometry data analysis by es-

timating the entire distribution of the studied parameters. This work underpins sample efficiency as a general metric reflecting the feasibility of reflectometry analysis via conventional methods and its connection to the characteristics of the posterior distribution. It is demonstrated that the analysis of any complex scenarios (e.g. structures with several layers) necessitates machine learning methods due to exponentially decreasing sample efficiency. However, similar to the machine learning methods for MLE, the modern probabilistic machine learning methods rely on static prior knowledge, making them practically unsuitable for reflectivity analysis.

This study addresses this issue by introducing the Prior Aware Neural Posterior Estimation method as an extension of the standard Neural Posterior Estimation. Dynamically set priors offer up to a $10^5$ increase in sample efficiency compared to the use of static priors, reflecting the ambiguity in the data resolved by the use of prior information. In comparison to conventional Bayesian methods, the model provides sample efficiency that is up to 12 orders of magnitude higher. This substantiates both the utility of the machine learning model and the inevitable inadequacy of conventional methods when applied to complex scenarios, such as structures with several layers or considerable uncertainty in the sample parameters. However, conventional methods can be effectively applied to refine the neural network-based solution to provide an accurate estimation of the posterior distribution in real time. The properties of the employed flow-based model guarantee resolving all the solutions within the user-defined prior ranges unless an out-of-distribution sample is provided. This outcome can guide experimenters by indicating whether the present ambiguity requires additional measurements, thereby directing the experiment until the singular physical solution is determined. Furthermore, the model naturally provides uncertainty estimation and possible correlations of the parameters in the form of a precise posterior distribution.

Our method presents an unparalleled tool for both the rapid analysis of measured data and for investigating the general properties of reflectivity data. In this way, we demonstrated that non-unique solutions can even emerge in a simple scenario of the benchmark experimental dataset with a few open parameters. The incorrect solutions

given by a conventional fit further emphasize the importance of probabilistic methods for reflectometry analysis. In general, the studied data suggests that the posterior distribution for reflectometry features a potentially high number of narrow distributional modes with largely correlated parameters, resulting in a low "efficient volume" of the distribution. Combined with a large number of local maxima that preclude the use of gradient-based techniques, this property makes reflectometry analysis a highly challenging task for both conventional and machine learning methods.

In general, the static prior distribution is most suitable for passive observations with a fixed setup and corresponding applications. Conversely, active experiments, even in the case of standardized reflectometry measurements, require dynamically adjusted prior distributions. In this sense, our model is potentially suitable for a wide range of scientific experiments that permit simulation-based inference.

### 8.1.3 Outlook

Below, we discuss potential avenues for further improving the introduced prior-aware approach:

- The dissertation discusses possible ways to detect out-of-distribution events (see Subsec. 6.6.2). However, this direction should be explored more rigorously, as it is crucial for the reliability of the method.
- The introduced distribution over prior parameters opens many options for improving the training scheme by adjusting it according to the potential use cases and excluding non-physical combinations of parameters.
- Furthermore, the distribution over prior parameters can be *actively* tuned during training by focusing on areas with complex cases (e.g., where data changes more rapidly with the parameters) [234]. Similar to the previous option, this approach is only feasible with the use of our prior-aware method.
- Other classes of prior distributions (e.g., a Gaussian distribution) are not yet explored and can be relevant for certain applications.

Furthermore, we consider other potential directions that are orthogonal to the prior-aware methodology:

- Co-refinement procedure of multiple-contrast experiments is a relevant direction of the reflectometry analysis [81, 83]. Importantly, this can be addressed using the developed model *without retraining*. In this manner, multiple curves can be processed separately by the model and then refined into the final solution via importance sampling or Markov Chain Monte Carlo methods in the postprocessing stage.
- One of the most significant practical limitations of the current design is the requirement for fixed discretization along the $q$ axis. Allowing for varying numbers of potentially non-equidistant $q$ points and the inclusion of error bars into the input necessitates considerable adjustments to the embedding network. Discretization-invariant learning appears to be a promising approach [235].
- The integration of the developed methods into user-friendly packages and GUI applications is being planned.

## 8.2 Grazing-incidence wide-angle scattering

### 8.2.1 General approach

Chap. 7 demonstrates an automated deep learning approach for the analysis of grazing-incidence wide-angle scattering data. Unlike the previously discussed integral solutions for reflectometry analysis, the analysis of diffraction images allows for a modular approach. This separates the computer vision task of identifying diffraction features from the subsequent processing of the detected features. In this manner, a computer vision task is a common necessary part of the analysis, which is followed by application-specific algorithms. The prior information being the key to reflectometry analysis is equally relevant in this case. However, the inclusion of prior information is shifted to the postprocessing algorithms, leaving the peak detection model agnostic

to studied materials. This approach is crucial for detecting unexpected crystal structures. In the use cases presented in this work, the prior information comes in the form of a list of known expected crystal structures used in the phase identification process. Identifying an unknown structure would necessitate the use of general peak indexing algorithms.

This work demonstrates that the computer vision task of detecting diffraction peaks is the major bottleneck of the automated analysis of grazing-incidence wide-angle scattering data. Once the peaks are identified, other types of analysis can be performed via simple, "classical" algorithms. In this way, we identified the formation of two coexisting phases of 2D organic-inorganic perovskites and traced the evolution of the lattice parameters in time for these phases. The refinement procedure revealed a slight decrease in the unit cell size for one of the identified phases. Such subtle processes may often be overlooked during a manual analysis of the data. Therefore, the presented solution allows us to fully harness the potential of the experimental technique.

### 8.2.2 Deep learning peak detection

Our developed deep learning algorithm enables fast detection of diffraction features, demonstrating high performance on diverse experimental data with accurate detection of most of the reflections.

There are several key ingredients for our solution. First, the image preprocessing step appears crucial. Non-linear adaptive contrast correction addresses the issue of high dynamical range and helps reveal diffraction features with lower intensity, and the conversion to polar coordinates is required to harvest the translational equivariance property of convolutional neural networks. Second, the work demonstrates the way to optimize modern object detection architecture to address various properties of the experimental data, such as the absence of sharp object edges and asymmetry of the diffraction features. The resulting architecture outperforms standard models despite having a substantially less number of parameters. Finally, the variability

of the data generation process of the synthetic training data is the key to robust performance on diverse experimental data.

### 8.2.3 Outlook

Yet, the performance of the detection algorithm is not ideal. Low-intense, overlapping, or narrow peaks are frequently missed, necessitating further adjustments to the deep learning pipeline. The following directions seem promising:

- Further adjustments to the contrast correction algorithm and input data representation. Different types of binarization of the input intensity might be beneficial for processing low-intense peaks.
- Adjustments to the second detection stage with the use of bipartite matching and Transformer architectures [140, 208] might be a solution to detecting strongly overlapping peaks.
- Wider and shallower feature maps can be employed to improve the detection of narrow peaks.
- To address the discrepancies between the simulated and the experimental data, semi-supervised learning techniques can be employed.

## 8.3 Summary

This work underscores that the high acquisition rates and intricate, high-dimensional data provided by experimental science exceed the capabilities of traditional tools and necessitate the use of deep learning methods. Yet, the integration of these methods with traditional analysis remains key to refining and validating the results. Moreover, the research illuminates that an understanding of the specifics of the data being analyzed, the conventional analysis procedures, and the foundational physics is vital for architecting deep learning systems for data analysis. Accordingly, recognizing the ill-posed nature of the reflectometry analysis, as well as the asymmetry and unique

geometry of the examined diffraction data, was critical in the design of the developed deep learning solutions.

## Acknowledgments

My deepest thanks go out to the myriad individuals, both within and beyond our research circle, whose assistance and support were instrumental in shaping this project. I find myself indebted, first and foremost, to my esteemed doctoral mentor, Prof. Frank Schreiber. I am profoundly grateful to him for offering me an extraordinary and enriching opportunity to explore the intricate field of deep learning for scattering data analysis. He not only offered me an avenue of research but also graced me with an incredible degree of autonomy to carve my own path. This level of independence, intertwined with his steadfast support when it was most needed, crafted a truly unique journey for me. The privilege of conducting experiments at world-class experimental facilities was a seminal part of this journey. In numerous instances, the confidence expressed by Prof. Schreiber acted as a catalyst, prompting me to transcend my perceived boundaries and experience significant growth on both professional and personal levels. His ability to identify and harness potential truly underscores the achievements realized and signifies the limitless possibilities within our domain.

My profound appreciation goes out to Dr. Alexander Hinderhofer. His active engagement in daily dialogues, spanning overarching concepts to intricate details, was integral to my work. His readiness to engage in deep technical exchanges and his academic prowess truly formed the bedrock of this research. His influence is evident in key portions of this study, underscoring his indelible contributions. My heartfelt thanks also extend to my co-supervisor, Prof. Martin Oettel, whose presence and support added a valuable dimension to the research journey. Furthermore, I would like to express my appreciation to Dr. Alexander Gerlach for diligently supplying the required computational environment and offering his experienced scientific viewpoint on my research.

I also wish to express my sincere appreciation for my colleague, Dr. Alessandro Greco. Our shared journey was one of both professional collaboration and personal camaraderie, and I am indeed fortunate for the chance to work alongside him. His

insights and our extended dialogues, ranging far beyond the confines of our research topic, have been a source of inspiration and cheer. His friendship enriched this journey significantly, shaping my understanding of our work and fostering a warm spirit of fellowship for which I am profoundly thankful.

My heartfelt thanks also extend to Valentin Munteanu. His fervor for machine learning and intellectual curiosity are both compelling and invigorating. Valentin has generously spent numerous hours providing meticulous and thoughtful reviews of my work. His profound insights and his keen ability to unearth relevant literature, that might have otherwise slipped under my radar, have been a significant asset to my research. His enthusiasm and commitment have unequivocally amplified the scope and depth of my work.

My sincere gratitude goes to Maximilian Dax, who not only introduced me to the intricate world of probabilistic machine learning but also shared invaluable insights that directly influenced the trajectory of this project. His intellectual contributions have indelibly shaped the work presented here, leaving a lasting imprint on my research.

I must also express my profound appreciation for Dr. Linus Pithan, whose enduring enthusiasm, readiness to lend a hand, exceptional professionalism, and deep insights have served as a guiding light. I have been fortunate enough to tap into his extensive expertise, spanning from experimental methods and scattering physics to software engineering, particularly during our experiments at synchrotron facilities. His guidance has left an indelible mark on my professional growth.

In the same vein, I extend my gratitude to the other colleagues with whom I had the privilege of conducting experiments. I particularly want to acknowledge Prof. Stefan Kowarik, whose enlightening presence significantly influenced my research journey. His insightful inputs and far-reaching perspectives have been crucial in shaping my work. I also extend heartfelt thanks to Dr. Ivan Zaluzhnyy, Ekaterina Kneschaurek, Niels Scheffczyk, Constantin Völter, David Mareček, Lukas Petersdorf, and Ingrid Dax. Sharing the journey of scientific exploration with them was truly enriching. I also want to express my appreciation for the staff at DESY, particularly

Dr. Florian Bertram and Dr. Chen Shen, and for Dr. André Rothkirch's assistance in utilizing the Maxwell cluster's resources fundamental to the realization of this project.

I am deeply thankful to the students I worked with, particularly Alina Pleli, Felix Hagen, Divin Gavran, and Verena Herbst, among others. Their patience and dedication to this project were commendable. Explaining ideas to them deepened my own comprehension of the topic. Sharing this journey with them has been a true delight, and I sincerely wish them all the very best in their future endeavors.

In the end, my most profound gratitude is reserved for Dr. Anastasia Ragulskaya and her ceaseless support, bestowed generously and unconditionally across every aspect of this endeavor. Her steadfast faith, motivation, and compassion have provided a constant guiding light throughout this journey.

# Bibliography


1. Sinha, S. K., Sirota, E. B., Garoff, S. & Stanley, H. B. X-ray and neutron scattering from rough surfaces. *Phys. Rev. B* **38,** 2297 (4 1988).
2. Feidenhans'l, R. Surface structure determination by X-ray diffraction. *Surf. Sci. Rep.* **10,** 105 (1989).
3. Lovesey, S. W. *Theory of neutron scattering from condensed matter* (Clarendon Press, 1984).
4. Dosch, H. *Critical Phenomena at Surfaces and Interfaces* (Springer, Berlin, 1992).
5. Tolan, M. *X-ray Scattering from Soft-Matter Thin Films: Materials Science and Basic Research* (Springer, Berlin, 1999).
6. Als-Nielsen, J. & McMorrow, D. *Elements of Modern X-ray Physics* (John Wiley & Sons, West Sussex, United Kingdom, 2011).
7. Kowarik, S. Thin film growth studies using time-resolved x-ray scattering. *J. Phys.: Condens. Matter* **29,** 043003 (2017).
8. Joress, H., Brock, J. D. & Woll, A. R. Quick X-ray reflectivity using monochromatic synchrotron radiation for time-resolved applications. *J. Synchrotron Rad.* **25,** 706 (2018).
9. Lippmann, M., Buffet, A., Pflaum, K., Ehnes, A., Ciobanu, A. & Seeck, O. H. A new setup for high resolution fast X-ray reflectivity data acquisition. *Rev. Sci. Instrum.* **87,** 113904 (2016).

10. Hinderhofer, A., Gerlach, A., Kowarik, S., Zontone, F., Krug, J. & Schreiber, F. Smoothing and coherent structure formation in organic-organic heterostructure growth. *Europhys. Lett.* **91,** 56002 (1 (2010).

11. Miyadera, T., Shibata, Y., Koganezawa, T., Murakami, T. N., Sugita, T., Tanigaki, N. & Chikamatsu, M. Crystallization Dynamics of Organolead Halide Perovskite by Real-Time X-ray Diffraction. *Nano Lett.* **15,** 5630 (2015).

12. Gisriel, C., Coe, J., Letrun, R., Yefanov, O. M., Luna-Chavez, C., Stander, N. E., Lisova, S., Mariani, V., Kuhn, M., Aplin, S., Grant, T. D., Dörner, K., Sato, T., Echelmeier, A., Villarreal, J. C., Hunter, M. S., Wiedorn, M. O., Knoska, J., Mazalova, V., Roy-Chowdhury, S., Yang, J.-H., Jones, A., Bean, R., Bielecki, J., Kim, Y., Mills, G., Weinhausen, B., Meza, J. D., Al-Qudami, N., Bajt, S., Brehm, G., Botha, S., Boukhelef, D., Brockhauser, S., Bruce, B. D., Coleman, M. A., Danilevski, C., Discianno, E., Dobson, Z., Fangohr, H., Martin-Garcia, J. M., Gevorkov, Y., Hauf, S., Hosseinizadeh, A., Januschek, F., Ketawala, G. K., Kupitz, C., Maia, L., Manetti, M., Messerschmidt, M., Michelat, T., Mondal, J., Ourmazd, A., Previtali, G., Sarrou, I., Schön, S., Schwander, P., Shelby, M. L., Silenzi, A., Sztuk-Dambietz, J., Szuba, J., Turcato, M., White, T. A., Wrona, K., Xu, C., Abdellatif, M. H., Zook, J. D., Spence, J. C. H., Chapman, H. N., Barty, A., Kirian, R. A., Frank, M., Ros, A., Schmidt, M., Fromme, R., Mancuso, A. P., Fromme, P. & Zatsepin, N. A. Membrane protein megahertz crystallography at the European XFEL. *Nat. Commun.* **10,** 5021 (2019).

13. Neville, F., Cahuzac, M., Konovalov, O., Ishitsuka, Y., Lee, K. Y. C., Kuzmenko, I., Kale, G. M. & Gidalevitz, D. Lipid Headgroup Discrimination by Antimicrobial Peptide LL-37: Insight into Mechanism of Action. *Biophys. J.* **90,** 1275 (2006).

14. Yabashi, M. & Tanaka, H. The next ten years of X-ray science. *Nat. Photonics* **11,** 12 (2017).

15. Alipanahi, B., Delong, A., Weirauch, M. T. & Frey, B. J. Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning. *Nat. Biotechnol.* **33,** 831 (2015).

16. Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žıdek, A., Potapenko, A., Bridgland, A., Meyer, C., Kohl, S. A. A., Ballard, A. J., Cowie, A., Romera-Paredes, B., Nikolov, S., Jain, R., Adler, J., Back, T., Petersen, S., Reiman, D., Clancy, E., Zielinski, M., Steinegger, M., Pacholska, M., Berghammer, T., Bodenstein, S., Silver, D., Vinyals, O., Senior, A. W., Kavukcuoglu, K., Kohli, P. & Hassabis, D. Highly accurate protein structure prediction with AlphaFold. *Nature* **596,** 583 (2021).

17. Askr, H., Elgeldawi, E., Ella, H. A., Elshaier, Y. A. M. M., Gomaa, M. M. & Hassanien, A. E. Deep learning in drug discovery: an integrative review and future challenges. *Artif. Intell. Rev.* **56,** 5975 (2022).

18. Schawinski, K., Zhang, C., Zhang, H., Fowler, L. & Santhanam, G. K. Generative Adversarial Networks recover features in astrophysical images of galaxies beyond the deconvolution limit. *Mon. Notices Royal Astron. Soc.: Let.,* slx008 (2017).

19. Shallue, C. J. & Vanderburg, A. Identifying Exoplanets with Deep Learning: A Five-planet Resonant Chain around Kepler-80 and an Eighth Planet around Kepler-90. *Astron. J.* **155,** 94 (2018).

20. Reichstein, M., Camps-Valls, G., Stevens, B., Jung, M., Denzler, J., Carvalhais, N. & Prabhat. Deep learning and process understanding for data-driven Earth system science. *Nature* **566,** 195 (2019).

21. Baldi, P., Sadowski, P. & Whiteson, D. Searching for exotic particles in high-energy physics with deep learning. *Nat. Commun.* **5** (2014).

22. Karagiorgi, G., Kasieczka, G., Kravitz, S., Nachman, B. & Shih, D. Machine learning in the search for new fundamental physics. *Nat. Rev. Phys.* **4,** 399 (2022).

23. Albertsson, K. *et al.* Machine Learning in High Energy Physics Community White Paper. *J. Phys. Conf. Ser.* **1085,** 022008 (2018).

24. Karniadakis, G. E., Kevrekidis, I. G., Lu, L., Perdikaris, P., Wang, S. & Yang, L. Physics-informed machine learning. *Nat. Rev. Phys.* **3,** 422 (2021).

25. Wright, L. G., Onodera, T., Stein, M. M., Wang, T., Schachter, D. T., Hu, Z. & McMahon, P. L. Deep physical neural networks trained with backpropagation. *Nature* **601,** 549 (2022).

26. Pederson, R., Kalita, B. & Burke, K. Machine learning and density functional theory. *Nat. Rev. Phys.* **4,** 357 (2022).

27. Erdmann, M., Glombitza, J., Kasieczka, G. & Klemradt, U. *Deep Learning for Physics Research* (World Scientific, 2021).

28. Ludwig, A. Discovery of new materials using combinatorial synthesis and high-throughput characterization of thin-film materials libraries combined with computational methods. *npj Comput. Mater.* **5,** 70 (2019).

29. Egger, A. T., Hörmann, L., Jeindl, A., Scherbela, M., Obersteiner, V., Todorović, M., Rinke, P. & Hofmann, O. T. Charge Transfer into Organic Thin Films: A Deeper Insight through Machine-Learning-Assisted Structure Search. *Adv. Sci.* **7,** 2000992 (2020).

30. Kusne, A. G., Gao, T., Mehta, A., Ke, L., Nguyen, M. C., Ho, K.-M., Antropov, V., Wang, C.-Z., Kramer, M. J., Long, C. & Takeuchi, I. On-the-fly machine-learning for high-throughput experiments: search for rare-earth-free permanent magnets. *Sci. Rep.* **4,** 6367 (2014).

31. Martynec, T., Karapanagiotis, C., Klapp, S. H. L. & Kowarik, S. Machine learning predictions of surface migration barriers in nucleation and non-equilibrium growth. *Commun. Mater.* **2,** 90 (2021).

32. Kern, S., Liehr, S., Wander, L., Bornemann-Pfeiffer, M., Müller, S., Maiwald, M. & Kowarik, S. Artificial neural networks for quantitative online NMR spectroscopy. *Anal. Bioanal. Chem.* **412,** 4447 (2020).

33. Park, W. B., Chung, J., Jung, J., Sohn, K., Singh, S. P., Pyo, M., Shin, N. & Sohn, K.-S. Classification of crystal structure using a convolutional neural network. *IUCrJ* **4,** 486 (2017).

34. Oviedo, F., Ren, Z., Sun, S., Settens, C., Liu, Z., Hartono, N. T. P., Ramasamy, S., DeCost, B. L., Tian, S. I., Romano, G., Kusne, A. G. & Buonassisi, T. Fast and interpretable classification of small X-ray diffraction datasets using data augmentation and deep neural networks. *npj Comput. Mater.* **5,** 60 (2019).

35. Aguiar, J., Gong, M. L., Unocic, R., Tasdizen, T. & Miller, B. Decoding crystallography from high-resolution electron imaging and diffraction datasets with deep learning. *Sci. Adv.* **5,** eaaw1949 (2019).

36. Souza, A., Oliveira, L. B., Hollatz, S., Feldman, M., Olukotun, K., Holton, J. M., Cohen, A. E. & Nardi, L. DeepFreak: Learning Crystallography Diffraction Patterns with Automated Machine Learning. *arXiv,* 1904.11834 (2019).

37. Sullivan, B., Archibald, R., Azadmanesh, J., Vandavasi, V. G., Langan, P. S., Coates, L., Lynch, V. & Langan, P. BraggNet: integrating Bragg peaks using neural networks. *J. Appl. Crystallogr.* **52,** 854 (2019).

38. Vecsei, P. M., Choo, K., Chang, J. & Neupert, T. Neural network based classification of crystal symmetries from x-ray diffraction patterns. *Phys. Rev. B* **99,** 245120 (2019).

39. Lee, J.-W., Park, W. B., Lee, J. H., Singh, S. P. & Sohn, K.-S. A deep-learning technique for phase identification in multiphase inorganic compounds using synthetic XRD powder patterns. *Nat. Commun.* **11,** 86 (2020).

40. Dong, H., Butler, K. T., Matras, D., Price, S. W., Odarchenko, Y., Khatry, R., Thompson, A., Middelkoop, V., Jacques, S. D., Beale, A. M. & Vamvakeros, A. A deep convolutional neural network for real-time full profile analysis of big powder diffraction data. *npj Comput. Mater.* **7,** 74 (2021).

41. Sullivan, B., Archibald, R., Azadmanesh, J., Vandavasi, V. G., Langan, P. S., Coates, L., Lynch, V. & Langan, P. BraggNet: integrating Bragg peaks using neural networks. *J. Appl. Crystallogr.* **52,** 854 (2019).

42. Liu, Z., Sharma, H., Park, J.-S., Kenesei, P., Miceli, A., Almer, J., Kettimuthu, R. & Foster, I. BraggNN: Fast X-ray Bragg Peak Analysis Using Deep Learning. *arXiv,* 2008.08198 (2021).

43. Franke, D., Jeffries, C. M. & Svergun, D. I. Machine Learning Methods for X-Ray Scattering Data Analysis from Biomacromolecular Solutions. *Biophys. J.* **114,** 2485 (2018).

44. Archibald, R. K., Doucet, M., Johnston, T., Young, S. R., Yang, E. & Heller, W. T. Classifying and analyzing small-angle scattering data using weighted $k$ nearest neighbors machine learning techniques. *J. Appl. Crystallogr.* **53,** 326 (2020).

45. Chang, M.-C., Wei, Y., Chen, W.-R. & Do, C. Deep learning-based super-resolution for small-angle neutron scattering data: attempt to accelerate experimental workflow. *MRS Commun.* **10,** 11 (2020).

46. Song, G., Porcar, L., Boehm, M., Cecillon, F., Dewhurst, C., Goc, Y. L., Locatelli, J., Mutti, P. & Weber, T. Deep Learning Methods On Neutron Scattering Data. *EPJ Web Conf.* **225** (eds Lyoussi, A., Giot, M., Carette, M., Jenčič, I., Reynard-Carette, C., Vermeeren, L., Snoj, L. & Dû, P. L.) 01004 (2020).

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

48. Ikemoto, H., Yamamoto, K., Touyama, H., Yamashita, D., Nakamura, M. & Okuda, H. Classification of grazing-incidence small-angle X-ray scattering patterns by convolutional neural network. *J. Synchrotron Rad.* **27,** 1069 (2020).

49. Van Herck, W., Fisher, J. & Ganeva, M. Deep learning for x-ray or neutron scattering under grazing-incidence: extraction of distributions. *Mater. Res. Express* **8,** 045015 (2021).

50. Hinderhofer, A., Greco, A., Starostin V., Munteanu, V., Pithan, L., Gerlach, A. & Schreiber, F. Machine Learning for Scattering Data: Strategies, Perspectives, and Applications to Surface Scattering. *J. Appl. Crystallogr.* **56,** 3 (2023).

51. Zhou, X.-L. & Chen, S.-H. Theoretical foundation of X-ray and neutron reflectometry. *Phys. Rep.* **257,** 223 (1995).

52. Holý, V., Pietsch, U. & Baumbach, T. *High-Resolution X-Ray Scattering from Thin Films and Multilayers* (Springer Berlin, 1999).

53. Daillant, J. & Gibaud, A. *X-Ray and Neutron Reflectivity* (Springer, 2009).

54. Parratt, L. G. Surface Studies of Solids by Total Reflection of X-Rays. *Phys. Rev.* **95,** 359 (2 1954).

55. Braslau, A., Pershan, P. S., Swislow, G., Ocko, B. M. & Als-Nielsen, J. Capillary waves on the surface of simple liquids measured by x-ray reflectivity. *Phys. Rev. A* **38,** 2457 (1988).

56. Metzger, T., Luidl, C., Pietsch, U. & Vierl, U. Novel versatile X-ray reflectometer for angle and energy dispersive characterization of liquid and solid surfaces and interfaces. *Nucl. Instrum. Methods Phys. Res* **350,** 398 (1994).

57. Michely, T. & Krug, J. *Islands, Mounds, and Atoms. Patterns and Processes in Crystal Growth Far from Equilibrium* (Springer, Berlin, 2004).

58. Lehmkühler, F., Paulus, M., Sternemann, C., Lietz, D., Venturini, F., Gutt, C. & Tolan, M. The Carbon Dioxide-Water Interface at Conditions of Gas Hydrate Formation. *J. Am. Chem. Soc.* **131,** 585 (2008).

59. Seeck, O. H., Kim, H., Lee, D. R., Shu, D., Kaendler, I. D., Basu, J. K. & Sinha, S. K. Observation of thickness quantization in liquid films confined to molecular dimension. *EPL* **60,** 376 (2002).

60. Fragneto-Cusani, G. Neutron reflectivity at the solid/liquid interface: examples of applications in biophysics. *J. Phys.: Condens. Matter* **13,** 4973 (2001).

61. Ankner, J., Majkrzak, C. & Satija, S. Neutron reflectivity and grazing angle diffraction. *J. Res. Natl. Inst. Stand. Technol.* **98,** 47 (1993).

62. Mukherjee, M., Bhattacharya, M., Sanyal, M. K., Geue, T., Grenzer, J. & Pietsch, U. Reversible negative thermal expansion of polymer films. *Phys. Rev. E* **66,** 061801 (2002).

63. Skoda, M. W. A., Thomas, B., Hagreen, M., Sebastiani, F. & Pfrang, C. Simultaneous neutron reflectometry and infrared reflection absorption spectroscopy (IRRAS) study of mixed monolayer reactions at the air–water interface. *RSC Adv.* **7,** 34208 (54 2017).

64. Wasserman, S. R., Whitesides, G. M., Tidswell, I. M., Ocko, B. M., Pershan, P. S. & Axe, J. D. The structure of self-assembled monolayers of alkylsiloxanes on silicon: a comparison of results from ellipsometry and low-angle x-ray reflectivity. *J. Am. Chem. Soc.* **111,** 5852 (1989).

65. Skoda, M. W., Conzelmann, N. F., Fries, M. R., Reichart, L. F., Jacobs, R. M., Zhang, F. & Schreiber, F. Switchable $\beta$-lactoglobulin (BLG) adsorption on protein resistant oligo (ethylene glycol) (OEG) self-assembled monolayers (SAMs). *J. Colloid Interface Sci.* **606,** 1673 (2022).

66. Majkrzak, C. Polarized neutron reflectometry. *Phys. B Condens. Matter* **173,** 75 (1991).

67. Tidswell, I. M., Ocko, B. M., Pershan, P. S., Wasserman, S. R., Whitesides, G. M. & Axe, J. D. X-ray specular reflection studies of silicon coated by organic monolayers (alkylsiloxanes). *Phys. Rev. B.* **41,** 1111 (1990).

68. Fenter, P., Schreiber, F., Bulović, V. & Forrest, S. Thermally induced failure mechanisms of organic light emitting device structures probed by X-ray specular reflectivity. *Chem. Phys. Lett.* **277,** 521 (1997).

69. Lorch, C., Banerjee, R., Frank, C., Dieterle, J., Hinderhofer, A., Gerlach, A. & Schreiber, F. Growth of competing crystal phases of $\alpha$-sexithiophene studied by real-time X-ray scattering. *J. Phys. Chem. C* **119,** 819 (2015).

70. Kowarik, S., Gerlach, A., Sellner, S., Schreiber, F., Cavalcanti, L. & Konovalov, O. Real-time observation of structural and orientational transitions during growth of organic thin films. *Phys. Rev. Lett.* **96,** 125504 (2006).

71. Kowarik, S., Gerlach, A., Leitenberger, W., Hu, J., Witte, G., Wöll, C., Pietsch, U. & Schreiber, F. Energy-dispersive X-ray reflectivity and GID for real-time growth studies of pentacene thin films. *Thin Solid Films* **515,** 5606 (2007).

72. Woll, A. R., Desai, T. V. & Engstrom, J. R. Quantitative modeling of in situ X-ray reflectivity during organic molecule thin film growth. *Phys. Rev. B* **84,** 075479 (2011).

73. Kojima, A., Teshima, K., Shirai, Y. & Miyasaka, T. Organometal Halide Perovskites as Visible-Light Sensitizers for Photovoltaic Cells. *J. Am. Chem. Soc.* **131,** 6050 (2009).

74. Park, N.-G. Organometal perovskite light absorbers toward a 20% efficiency low-cost solid-state mesoscopic solar cell. *J. Phys. Chem. Lett.* **4,** 2423 (2013).

75. Hu, Q., Zhao, L., Wu, J., Gao, K., Luo, D., Jiang, Y., Zhang, Z., Zhu, C., Schaible, E., Hexemer, A., Wang, C., Liu, Y., Zhang, W., Grätzel, M., Liu, F., Russell, T. P., Zhu, R. & Gong, Q. In situ dynamic observations of perovskite crystallisation and microstructure evolution intermediated from [PbI6]4- cage nanoparticles. *Nat. Commun.* **8,** 15688 (2017).

76. Schlipf, J. & Müller-Buschbaum, P. Structure of Organometal Halide Perovskite Films as Determined with Grazing-Incidence X-Ray Scattering Methods. *Adv. Energy Mater.* **7,** 1700131 (2017).

77. Chen, A. Z., Shiu, M., Ma, J. H., Alpert, M. R., Zhang, D., Foley, B. J., Smilgies, D.-M., Lee, S.-H. & Choi, J. J. Origin of vertical orientation in two-

dimensional metal halide perovskites and its effect on photovoltaic performance. *Nat. Commun.* **9,** 1336 (2018).

78. Liu, Y., Akin, S., Hinderhofer, A., Eickemeyer, F. T., Zhu, H., Seo, J.-Y., Zhang, J., Schreiber, F., Zhang, H., Zakeeruddin, S. M., *et al.* Stabilization of highly efficient and stable phase-pure FAPbI3 perovskite solar cells by molecularly tailored 2D-overlayers. *Angew. Chem. Int. Ed.* **59,** 15688 (2020).

79. Zhang, H., Eickemeyer, F. T., Zhou, Z., Mladenović, M., Jahanbakhshi, F., Merten, L., Hinderhofer, A., Hope, M. A., Ouellette, O., Mishra, A., Ahlawat, P., Ren, D., Su, T.-S., Krishna, A., Wang, Z., Dong, Z., Guo, J., Zakeeruddin, S. M., Schreiber, F., Hagfeldt, A., Emsley, L., Rothlisberger, U., Milić, J. V. & Grätzel, M. Multimodal host–guest complexation for efficient and stable perovskite photovoltaics. *Nat. Commun.* **12,** 3383 (2021).

80. Qin, M., Chan, P. F. & Lu, X. A systematic review of metal halide perovskite crystallization and film formation mechanism unveiled by in situ GIWAXS. *Adv. Mater.* **33,** 2105290 (2021).

81. Nelson, A. Co-refinement of multiple-contrast neutron/X-ray reflectivity data using *MOTOFIT*. *J. Appl. Crystallogr.* **39,** 273 (2006).

82. Björck, M. & Andersson, G. GenX: An extensible X-ray reflectivity refinement program utilizing differential evolution. *J. Appl. Crystallogr.* **40,** 1174 (2007).

83. Nelson, A. R. J. & Prescott, S. W. *refnx*: neutron and X-ray reflectometry analysis in Python. *J. Appl. Crystallogr.* **52,** 193 (2019).

84. Greco, A., Starostin V., Karapanagiotis, C., Hinderhofer, A., Gerlach, A., Pithan, L., Liehr, S., Schreiber, F. & Kowarik, S. Fast fitting of reflectivity data of growing thin films using neural networks. *J. Appl. Crystallogr.* **52,** 1342 (2019).

85. Mironov, D., Durant, J. H., Mackenzie, R. & Cooper, J. F. K. Towards automated analysis for neutron reflectivity. *Mach. Learn.: Sci. Technol.* **2,** 035006 (2021).

86. Doucet, M., Archibald, R. K. & Heller, W. T. Machine learning for neutron reflectometry data analysis of two-layer thin films. *Mach. Learn.: Sci. Technol.* **2,** 035001 (2021).

87. Loaiza, J. M. C. & Raza, Z. Towards reflectivity profile inversion through artificial neural networks. *Mach. Learn.: Sci. Technol.* **2,** 025034 (2021).

88. Greco, A., Starostin V., Hinderhofer, A., Gerlach, A., Skoda, M. W. A., Kowarik, S. & Schreiber, F. Neural network analysis of neutron and x-ray reflectivity data: pathological cases, performance and perspectives. *Mach. Learn.: Sci. Technol.* **2,** 045003 (2021).

89. Andrejevic, N., Chen, Z., Nguyen, T., Fan, L., Heiberger, H., Zhou, L.-J., Zhao, Y.-F., Chang, C.-Z., Grutter, A. & Li, M. Elucidating proximity magnetism through polarized neutron reflectometry and machine learning. *Appl. Phys. Rev.* **9,** 011421 (2022).

90. Lee, J. & Jin, J. A novel method to design and evaluate artificial neural network for thin film thickness measurement traceable to the length standard. *Sci. Rep.* **12,** 2212 (2022).

91. Greco, A., Starostin V., Edel, E., Munteanu, V., Russegger, N., Dax, I., Shen, C., Bertram, F., Hinderhofer, A., Gerlach, A. & Schreiber, F. Neural network analysis of neutron and X-ray reflectivity data: automated analysis using ml-reflect, experimental errors and feature engineering. *J. Appl. Crystallogr.* **55,** 362 (2022).

92. McCluskey, A. R., Caruana, A. J., Kinane, C. J., Armstrong, A. J., Arnold, T., Cooper, J. F. K., Cortie, D. L., Hughes, A. V., Moulin, J.-F., Nelson, A. R. J., Potrzebowski, W. & Starostin, V.. Advice on describing Bayesian analysis of neutron and X-ray reflectometry. *J. Appl. Crystallogr.* **56,** 12 (2023).

93. Pithan, L., Starostin, V., Mareček, D., Petersdorf, L., Völter, C., Munteanu, V., Jankowski, M., Konovalov, O., Gerlach, A., Hinderhofer, A., Murphy, B., Kowarik, S. & Schreiber, F. Closing the loop: Autonomous experiments en-

abled by machine-learning-based online data analysis in synchrotron beamline environments. *arXiv.* preprint, 2306.11899 (2023).

94. Munteanu, V., Starostin, V., Greco, A., Pithan, L., Gerlach, A., Hinderhofer, A., Kowarik, S. & Schreiber, F. Neural network analysis of neutron and X-ray reflectivity data: Incorporating prior knowledge for tackling the phase problem. *arXiv.* preprint, 2307.05364 (2023).

95. Starostin, V. *et al.* Resolving the Phase Problem in Reflectometry with Prior Aware Neural Posterior Estimation. in preparation (2023).

96. Starostin V., Munteanu, V., Greco, A., Kneschaurek, E., Pleli, A., Bertram, F., Gerlach, A., Hinderhofer, A. & Schreiber, F. Tracking perovskite crystallization via deep learning-based feature detection on 2D X-ray scattering data. *npj Comput. Mater.* **8,** 101 (2022).

97. Starostin, V., Pithan, L., Greco, A., Munteanu, V., Gerlach, A., Hinderhofer, A. & Schreiber, F. End-to-End Deep Learning Pipeline for Real-Time Processing of Surface Scattering Data at Synchrotron Facilities. *Synchrotron Radiat. News* **35,** 21 (2022).

98. De Broglie, L. Waves and Quanta. *Nature* **112,** 540 (1923).

99. Compton, A. H. A Quantum Theory of the Scattering of X-rays by Light Elements. *Phys. Rev.* **21,** 483 (5 1923).

100. Bragg, W. L. The Diffraction of Short Electromagnetic Waves by a Crystal. *Scientia* **23,** 153 (1929).

101. Miller, W. H. *A treatise on crystallography* (Pitt Press, J. & JJ Deighton, 1839).

102. Stoev, K. N. & Sakurai, K. Review on grazing incidence X-ray spectrometry and reflectometry. *Spectrochim. Acta B: At. Spectrosc.* **54,** 41 (1999).

103. Cappuccio, G., Terranova, M., nazionale di fisica nuclear, I., nazionale di fisica nucleare. Laboratori nazionali di Frascati, I. & italiana di cristallografia, A. *Thin Film Characterisation by Advanced X-ray Diffraction Techniques* (SIS Pubblicazioni, Laboratori Nationali di Frascati, 1996).

104. Sivia, D. S. *Elementary scattering theory: for X-ray and neutron users* (Oxford University Press, 2011).

105. Sinha, S. K. & Pynn, R. in *Local Structure from Diffraction* (eds Billinge, S. J. L. & Thorpe, M. F.) 351 (Springer US, Boston, MA, 2002).

106. Festersen, S., Hrkac, S. B., Koops, C. T., Runge, B., Dane, T., Murphy, B. M. & Magnussen, O. M. X-ray reflectivity from curved liquid interfaces. *J. Synchrotron Radiat.* **25,** 432 (2018).

107. Schlomka, J.-P., Fitzsimmons, M., Pynn, R., Stettner, J., Seeck, O., Tolan, M. & Press, W. Characterization of Si/Ge interfaces by diffuse X-ray scattering in the region of total external reflection. *Phys. B: Condens. Matter* **221,** 44 (1996).

108. Treece, B. W., Kienzle, P. A., Hoogerheide, D. P., Majkrzak, C. F., Lösche, M. & Heinrich, F. Optimization of reflectometry experiments using information theory. *J. Appl. Crystallogr.* **52,** 47 (2019).

109. Salditt, T. & Aeffner, S. X-ray structural investigations of fusion intermediates: Lipid model systems and beyond. *Semin. Cell Dev. Biol.* **60,** 65 (2016).

110. Sironi, B., Snow, T., Redeker, C., Slastanova, A., Bikondoa, O., Arnold, T., Klein, J. & Briscoe, W. H. Structure of lipid multilayers via drop casting of aqueous liposome dispersions. *Soft Matter* **12,** 3877 (2016).

111. Abelès, F. La théorie générale des couches minces. *Journal de Physique et le Radium* **11,** 307 (1950).

112. Newton, R. G. in *Scattering Theory in Mathematical Physics* 193 (Springer, Netherlands, 1974).

113. Pershan, P. S. X-ray or neutron reflectivity: Limitations in the determination of interfacial profiles. *Phys. Rev. E* **50,** 2369 (1994).

114. Sivia, D. S., Hamilton, W. A., Smith, G. S., Rieker, T. P. & Pynn, R. A novel experimental procedure for removing ambiguity from the interpretation of neutron and x-ray reflectivity measurements: "Speckle holography". *J. Appl. Phys.* **70,** 732 (1991).

115. Pynn, R. Neutron scattering by rough surfaces at grazing incidence. *Phys. Rev. B* **45,** 602 (1992).

116. Green, M. A., Ho-Baillie, A. & Snaith, H. J. The emergence of perovskite solar cells. *Nat. Photon.* **8,** 506 (2014).

117. Steele, J. A., Solano, E., Hardy, D., Dayton, D., Ladd, D., White, K., Chen, P., Hou, J., Huang, H., Saha, R. A., Wang, L., Gao, F., Hofkens, J., Roeffaers, M. B. J., Chernyshov, D. & Toney, M. F. How to GIWAXS: Grazing Incidence Wide Angle X-Ray Scattering Applied to Metal Halide Perovskite Thin Films. *Adv. Energy Mater.* (2023).

118. Venkatesan, N. R., Kennard, R. M., DeCrescent, R. A., Nakayama, H., Dahlman, C. J., Perry, E. E., Schuller, J. A. & Chabinyc, M. L. Phase Intergrowth and Structural Defects in Organic Metal Halide Ruddlesden–Popper Thin Films. *Chem. Mater.* **30,** 8615 (2018).

119. Simbrunner, J., Simbrunner, C., Schrode, B., Röthel, C., Bedoya-Martinez, N., Salzmann, I. & Resel, R. Indexing of grazing-incidence X-ray diffraction patterns: the case of fibre-textured thin films. *Acta Crystallogr. A* **74,** 373 (2018).

120. Patterson, A. L. The Scherrer Formula for X-Ray Particle Size Determination. *Phys. Rev.* **56,** 978 (1939).

121. Baker, J. L., Jimison, L. H., Mannsfeld, S., Volkman, S., Yin, S., Subramanian, V., Salleo, A., Alivisatos, A. P. & Toney, M. F. Quantification of Thin Film

Crystallographic Orientation Using X-ray Diffraction with an Area Detector. *Langmuir* **26,** 9146 (2010).

122. Chang, C.-Y., Huang, Y.-C., Tsao, C.-S. & Su, W.-F. Formation Mechanism and Control of Perovskite Films from Solution to Crystalline Phase Studied by in Situ Synchrotron Scattering. *ACS Appl. Mater. Interfaces* **8,** 26712 (2016).

123. Lemaréchal, C. Cauchy and the gradient method. *Doc. Math. Extra* **251,** 10 (2012).

124. Storn, R. & Price, K. *J. Glob. Optim.* **11,** 341 (1997).

125. Bayes, T. LII. An essay towards solving a problem in the doctrine of chances. By the late Rev. Mr. Bayes, FRS communicated by Mr. Price, in a letter to John Canton, AMFR S. *Philos. Trans. R. Soc.,* 370 (1763).

126. Kahn, H. & Harris, T. E. Estimation of particle transmission by random sampling. *National Bureau of Standards applied mathematics series* **12,** 27 (1951).

127. Kahn, H. *Use of different Monte Carlo sampling techniques* (Rand Corporation, 1955).

128. Tokdar, S. T. & Kass, R. E. Importance sampling: a review. *Wiley Interdiscip. Rev.: Comput. Stat.* **2,** 54 (2009).

129. Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H. & Teller, E. Equation of State Calculations by Fast Computing Machines. *J. Chem. Phys.* **21,** 1087 (1953).

130. Hastings, W. K. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika* **57,** 97 (1970).

131. Gelman, A., Carlin, J. B., Stern, H. S. & Rubin, D. B. *Bayesian Data Analysis* (Chapman and Hall/CRC, 1995).

132. Neal, R. *MCMC Using Hamiltonian Dynamics* (Chapman and Hall/CRC, 2011).

133. Goodman, J. & Weare, J. Ensemble samplers with affine invariance. *Comm. App. Math. Comp. Sci.* **5,** 65 (2010).

134. Foreman-Mackey, D., Hogg, D. W., Lang, D. & Goodman, J. emcee: The MCMC Hammer. *PASP* **125,** 306 (2013).

135. Goodfellow, I., Bengio, Y. & Courville, A. *Deep Learning* (MIT Press, 2016).

136. Pak, M. & Kim, S. *A review of deep learning in image recognition* in *2017 4th International Conference on Computer Applications and Information Processing Technology (CAIPT)* (2017), 1.

137. Wu, D., Sharma, N. & Blumenstein, M. *Recent advances in video-based human action recognition using deep learning: A review* in *2017 International Joint Conference on Neural Networks (IJCNN)* (2017), 2865.

138. Deng, L., Hinton, G. & Kingsbury, B. *New types of deep neural network learning for speech recognition and related applications: an overview* in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing* (2013), 8599.

139. Nassif, A. B., Shahin, I., Attili, I., Azzeh, M. & Shaalan, K. Speech Recognition Using Deep Neural Networks: A Systematic Review. *IEEE Access* **7,** 19143 (2019).

140. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł. & Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **30** (2017).

141. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., *et al.* Language models are few-shot learners. *Adv. Neural Inf. Process. Syst.* **33,** 1877 (2020).

142. Bubeck, S., Chandrasekaran, V., Eldan, R., Gehrke, J., Horvitz, E., Kamar, E., Lee, P., Lee, Y. T., Li, Y., Lundberg, S., *et al.* Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv,* 2303.12712 (2023).

143. Grigorescu, S., Trasnea, B., Cocias, T. & Macesanu, G. A survey of deep learning techniques for autonomous driving. *J. Field Robot.* **37,** 362 (2020).

144. Bebis, G. & Georgiopoulos, M. Feed-forward neural networks. *IEEE Potentials* **13,** 27 (1994).

145. Kolmogorov, A. K. On the Representation of Continuous Functions of Several Variables by Superposition of Continuous Functions of One Variable and Addition. *Soviet Math. Dokl. SSSR* **114,** 369 (1957).

146. Cybenko, G. Approximation by superpositions of a sigmoidal function. *Math. Control Signal. Syst.* **2,** 303 (1989).

147. Hornik, K. Approximation capabilities of multilayer feedforward networks. *Neural Netw.* **4,** 251 (1991).

148. Kullback, S. & Leibler, R. A. On information and sufficiency. *Ann. Math. Stat.* **22,** 79 (1951).

149. Shannon, C. E. A mathematical theory of communication. *Bell Syst. tech. j.* **27,** 379 (1948).

150. Rumelhart, D. E., Hinton, G. E. & Williams, R. J. Learning representations by back-propagating errors. *Nature* **323,** 533 (1986).

151. LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W. & Jackel, L. D. Backpropagation applied to handwritten zip code recognition. *Neural Comput.* **1,** 541 (1989).

152. Glorot, X. & Bengio, Y. *Understanding the difficulty of training deep feedforward neural networks* in *AISTATS* (2010), 249.

153. He, K., Zhang, X., Ren, S. & Sun, J. *Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification* in *ICCV* (2015), 1026.

154. Sutskever, I., Martens, J., Dahl, G. & Hinton, G. *On the importance of initialization and momentum in deep learning* in *Proceedings of the 30th International Conference on Machine Learning* (eds Dasgupta, S. & McAllester, D.) **28** (PMLR, Atlanta, Georgia, USA, 2013), 1139.

155. Nesterov, Y. E. *A method of solving a convex programming problem with convergence rate* $O\left(1/k^2\right)$ in *Soviet Math. Dokl.* **269** (1983), 543.

156. Duchi, J., Hazan, E. & Singer, Y. Adaptive subgradient methods for online learning and stochastic optimization. *JMLR* **12** (2011).

157. Kingma, D. P. & Ba, J. Adam: A Method for Stochastic Optimization. *arXiv,* 1412.6980 (2017).

158. Loshchilov, I. & Hutter, F. Decoupled weight decay regularization. *arXiv,* 1711.05101 (2017).

159. Bottou, L. *Stochastic gradient learning in neural networks* in *Proceedings of Neuro-Nımes* **91** (Nimes, 1991), 12.

160. LeCun, Y. A., Bottou, L., Orr, G. B. & Müller, K.-R. in *Neural Networks: Tricks of the Trade: Second Edition* (eds Montavon, G., Orr, G. B. & Müller, K.-R.) 9 (Springer Berlin Heidelberg, Berlin, Heidelberg, 2012).

161. Rosenblatt, F. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychol. Rev.* **65,** 386 (1958).

162. Glorot, X., Bordes, A. & Bengio, Y. *Deep sparse rectifier neural networks* in *Proceedings of the fourteenth international conference on artificial intelligence and statistics* (2011), 315.

163. Maas, A. L., Hannun, A. Y. & Ng, A. Y. *Rectifier Nonlinearities Improve Neural Network Acoustic Models* in *ICML* **30** (2013), 3.

164. Hendrycks, D. & Gimpel, K. Gaussian error linear units (gelus). *arXiv,* 1606.08415 (2016).

165. Klambauer, G., Unterthiner, T., Mayr, A. & Hochreiter, S. Self-normalizing neural networks. *Adv. Neural. Inf. Process. Syst.* **30** (2017).

166. Elfwing, S., Uchibe, E. & Doya, K. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural Netw.* **107.** Special issue on deep reinforcement learning, 3 (2018).

167. Cox, D. R. The regression analysis of binary sequences. *J. R. Stat. Soc., B: Stat. Methodol.* **20,** 215 (1958).

168. Ioffe, S. & Szegedy, C. *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift* in *ICML* (eds Bach, F. & Blei, D.) **37** (PMLR, Lille, France, 2015), 448.

169. Santurkar, S., Tsipras, D., Ilyas, A. & Madry, A. How does batch normalization help optimization? *Adv. Neural. Inf. Process. Syst.* **31** (2018).

170. Redmon, J., Divvala, S., Girshick, R. & Farhadi, A. *You only look once: Unified, real-time object detection* in *2016 IEEE Conf. Comput. Vis. Pattern Recogn.* (IEEE, 2016), 779.

171. Ren, S., He, K., Girshick, R. & Sun, J. *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks* in *Adv. Neural Inf. Process. Syst.* **28** (Curran Associates, Inc., 2015), 91.

172. Girshick, R. *Fast R-CNN* in *2015 IEEE Int. Conf. Comput. Vis.* (IEEE, 2015), 1440.

173. Simonyan, K. & Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv,* 1409.1556 (2014).

174. He, K., Zhang, X., Ren, S. & Sun, J. *Deep residual learning for image recognition* in *IEEE Conf. Comput. Vis. Pattern Recogn.* (IEEE, 2016), 770.

175. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. & Wojna, Z. Rethinking the Inception Architecture for Computer Vision. *arXiv,* 1512.00567 (2015).

176. Redmon, J. & Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv,* 1804.02767 (2018).

177. Liu, Z., Mao, H., Wu, C.-Y., Feichtenhofer, C., Darrell, T. & Xie, S. *A convnet for the 2020s* in *Proceedings of the IEEE/CVF Conf. Comput. Vis. Pattern Recogn.* (2022), 11976.

178. Lin, T.-Y., Dollar, P., Girshick, R., He, K., Hariharan, B. & Belongie, S. *Feature Pyramid Networks for Object Detection* in *2017 IEEE Conf. Comput. Vis. Pattern Recogn.* (IEEE, 2017), 936.

179. He, K., Gkioxari, G., Dollár, P. & Girshick, R. *Mask R-CNN* in *2017 IEEE Int. Conf. Comput. Vis.* (IEEE, 2017), 2961.

180. Papamakarios, G., Nalisnick, E., Rezende, D. J., Mohamed, S. & Lakshminarayanan, B. Normalizing Flows for Probabilistic Modeling and Inference. *J. Mach. Learn. Res.* **22** (2021).

181. Rezende, D. & Mohamed, S. *Variational inference with normalizing flows* in *International conference on machine learning* (2015), 1530.

182. Dinh, L., Krueger, D. & Bengio, Y. *NICE: Non-linear Independent Components Estimation* in *ICLR* (eds Bengio, Y. & LeCun, Y.) (2015).

183. Durkan, C., Bekasov, A., Murray, I. & Papamakarios, G. *Neural Spline Flows* in *Adv. Neural Inf. Process. Syst.* (eds Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E. & Garnett, R.) **32** (Curran Associates, Inc., 2019).

184. Gregory, J. A. & Delbourgo, R. Piecewise Rational Quadratic Interpolation to Monotonic Data. *IMA Journal of Numerical Analysis* **2,** 123 (1982).

185. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J. & Chintala, S. *PyTorch: An Imperative Style, High-Performance Deep Learning Library* in *NeurIPS* (2019).

186. Mareček, D., Oberreiter, J., Nelson, A. & Kowarik, S. Faster and lower-dose X-ray reflectivity measurements enabled by physics-informed modeling and artificial intelligence co-refinement. *J. Appl. Crystallogr.* **55,** 1305 (2022).

187. Zykov, A., Bommel, S., Wolf, C., Pithan, L., Weber, C., Beyer, P., Santoro, G., Rabe, J. P. & Kowarik, S. Diffusion and nucleation in multilayer growth of PTCDI-C $_8$ studied with *in situ* X-ray growth oscillations and real-time small angle X-ray scattering. *J. Chem. Phys.* **146,** 052803 (2017).

188. Krauss, T. N., Barrena, E., Zhang, X. N., De Oteyza, D. G., Major, J., Dehm, V., Würthner, F., Cavalcanti, L. P. & Dosch, H. Three-Dimensional Molecular Packing of Thin Organic Films of PTCDI-C $_8$ Determined by Surface X-ray Diffraction. *Langmuir* **24,** 12742 (2008).

189. Kiranyaz, S., Avci, O., Abdeljaber, O., Ince, T., Gabbouj, M. & Inman, D. J. 1D convolutional neural networks and applications: A survey. *MSSP* **151,** 107398 (2021).

190. Collobert, R., Puhrsch, C. & Synnaeve, G. Wav2Letter: an End-to-End ConvNet-based Speech Recognition System. *arXiv,* 1609.03193 (2016).

191. Guessoum, S., Belda, S., Ferrandiz, J. M., Modiri, S., Raut, S., Dhar, S., Heinkelmann, R. & Schuh, H. The Short-Term Prediction of Length of Day Using 1D Convolutional Neural Networks (1D CNN). *Sensors* **22,** 9517 (2022).

192. Jankowski, M., Belova, V., Chushkin, Y., Zontone, F., Levantino, M., Narayanan, T., Konovalov, O. & Pastore, A. The complex systems and biomedical sciences group at the ESRF: Current status and new opportunities after extremely brilliant source upgrade. *Nucl. Instrum. Methods Phys. Res. Sec. B* **538,** 164 (2023).

193. Ritley, K. A., Krause, B., Schreiber, F. & Dosch, H. A portable ultrahigh vacuum organic molecular beam deposition system for in situ x-ray diffraction measurements. *Rev. Sci. Instrum.* **72,** 1453 (2001).

194. Mondal, K. P., Bera, S., Gupta, A., Kumar, D., Gome, A., Reddy, V. R., Ito, N., Yamada-Takamura, Y., Pandit, P. & Roth, S. V. Effect of growth rate on quality of Alq $_3$ films and Co diffusion. *J. Phys. D* **54,** 155304 (2021).

195. Kowarik, S., Gerlach, A. & Schreiber, F. Organic molecular beam deposition: fundamentals, growth dynamics, and *in situ* studies. *J. Condens. Matter Phys.* **20,** 184005 (2008).

196. Schreiber, F. Organic molecular beam deposition: Growth studies beyond the first monolayer. *Phys. Stat. Sol.* **201,** 1037 (2004).

197. Witte, G. & Wöll, C. Growth of aromatic molecules on solid substrates for applications in organic electronics. *J. Mater. Res.* **19,** 1889 (2004).

198. Rieutord, F., Benattar, J. J., Rivoira, R., Lepêtre, Y., Blot, C. & Luzet, D. X-ray phase determination in multilayers. *Acta Crystallogr. A* **45,** 445 (1989).

199. Polo, F. M. & Vicente, R. Effective sample size, dimensionality, and generalization in covariate shift adaptation. *Neural. Comput. Appl.* (2022).

200. Braak, C. J. T. A Markov Chain Monte Carlo version of the genetic algorithm Differential Evolution: easy Bayesian computing for real parameter spaces. *Stat. Comput.* **16,** 239 (2006).

201. Cranmer, K., Brehmer, J. & Louppe, G. The frontier of simulation-based inference. *PNAS* **117,** 30055 (2020).

202. Dax, M., Green, S. R., Gair, J., Macke, J. H., Buonanno, A. & Schölkopf, B. Real-Time Gravitational Wave Science with Neural Posterior Estimation. *Phys. Rev. Lett.* **127** (2021).

203. Dax, M., Green, S. R., Gair, J., Puerrer, M., Wildberger, J., Macke, J. H., Buonanno, A. & Schölkopf, B. Neural Importance Sampling for Rapid and Reliable Gravitational-Wave Inference. *Phys. Rev. Lett.* **130** (2023).

204. Vasist, M., Rozet, F., Absil, O., Mollière, P., Nasedkin, E. & Louppe, G. Neural posterior estimation for exoplanetary atmospheric retrieval. *Astron. Astrophys.* **672,** A147 (2023).

205. Dinh, L., Sohl-Dickstein, J. & Bengio, S. Density estimation using real NVP. *arXiv,* 1605.08803 (2016).

206. Pithan, L., Greco, A., Hinderhofer, A., Gerlach, A., Kowarik, S., Rußegger, N., Dax, I. & Schreiber, F. Reflectometry curves (XRR and NR) and corresponding fits for machine learning. Version 1.0.0. Dataset. Publisher Zenodo. DOI: 10.5281/zenodo.6497437 (2022).

207. Yang, J., Zhou, K., Li, Y. & Liu, Z. Generalized out-of-distribution detection: A survey. *arXiv,* 2110.11334 (2021).

208. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A. & Zagoruyko, S. *End-to-end object detection with transformers* in *Comput. Vis. ECCV* (Springer, 2020), 213.

209. Rietveld, H. M. Line profiles of neutron powder-diffraction peaks for structure refinement. *Acta Cryst.* **22,** 151 (1967).

210. Rietveld, H. M. A profile refinement method for nuclear and magnetic structures. *J. Appl. Crystallogr.* **2,** 65 (1969).

211. David, W. I. F. Powder diffraction peak shapes. Parameterization of the pseudo-Voigt as a Voigt function. *J. Appl. Crystallogr.* **19,** 63 (1986).

212. Dinapoli, R., Bergamaschi, A., Henrich, B., Horisberger, R., Johnson, I., Mozzanica, A., Schmid, E., Schmitt, B., Schreiber, A., Shi, X. & Theidel, G. EIGER: Next generation single photon counting detector for X-ray applications. *Nucl. Instrum. Methods Phys. Res. A* **650,** 79 (2011).

213. Stark, J. A. *Adaptive image contrast enhancement using generalizations of histogram equalization* in. **9** (IEEE, 2000), 889.

214. Perlin, K. An Image Synthesizer. *SIGGRAPH Comput. Graph.* **19,** 287 (1985).

215. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J. & Chintala, S. *PyTorch: An Imperative Style, High-Performance Deep Learning Library* in *Adv. Neural Inf. Process. Syst.* **32** (Curran Associates, Inc., 2019).

216. Loshchilov, I. & Hutter, F. *Decoupled Weight Decay Regularization* in *ICLR* (2019).

217. Ioffe, S. & Szegedy, C. *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift* in *Proc. 32nd Int. Conf. Mach. Learn. ICML 2015* **37** (PMLR, Lille, France, 2015), 448.

218. Ronneberger, O., Fischer, P. & Brox, T. in *Lecture Notes in Computer Science* 234 (Springer International Publishing, 2015).

219. Wu, K., Otoo, E. & Shoshani, A. *Optimizing connected component labeling algorithms* in *SPIE Proc.* (SPIE, 2005).

220. Neubeck, A. & Van Gool, L. *Efficient non-maximum suppression* in *Proc. Int. Conf. Pattern Recognit.* **3** (IEEE, 2006), 850.

221. Wang, C.-Y., Yeh, I.-H. & Liao, H.-Y. M. You Only Learn One Representation: Unified Network for Multiple Tasks. *arXiv,* 2105.04206 (2021).

222. Stoumpos, C. C., Malliakas, C. D. & Kanatzidis, M. G. Semiconducting Tin and Lead Iodide Perovskites with Organic Cations: Phase Transitions, High Mobilities, and Near-Infrared Photoluminescent Properties. *Inorg. Chem.* **52,** 9019 (2013).

223. Stoumpos, C. C., Cao, D. H., Clark, D. J., Young, J., Rondinelli, J. M., Jang, J. I., Hupp, J. T. & Kanatzidis, M. G. Ruddlesden–Popper Hybrid Lead Iodide Perovskite 2D Homologous Semiconductors. *Chem. Mater.* **28,** 2852 (2016).

224. Soe, C. M. M., Nagabhushana, G. P., Shivaramaiah, R., Tsai, H., Nie, W., Blancon, J.-C., Melkonyan, F., Cao, D. H., Traoré, B., Pedesseau, L., Kepenekian, M., Katan, C., Even, J., Marks, T. J., Navrotsky, A., Mohite, A. D., Stoumpos, C. C. & Kanatzidis, M. G. Structural and thermodynamic limits of layer thickness in 2D halide perovskites. *PNAS* **116,** 58 (2018).

225. Seeck, O. H., Deiter, C., Pflaum, K., Bertram, F., Beerlink, A., Franz, H., Horbach, J., Schulte-Schrepping, H., Murphy, B. M., Greve, M. & Magnussen, O. The high-resolution diffraction beamline P08 at PETRA III. *J. Synchrotron Rad.* **19,** 30 (2012).

226. Newville, M., Stensitzki, T., Allen, D. B. & Ingargiola, A. *LMFIT: Non-Linear Least-Square Minimization and Curve-Fitting for Python,* https://zenodo.org/record/11813 2014.

227. Byrd, R. H., Lu, P., Nocedal, J. & Zhu, C. A Limited Memory Algorithm for Bound Constrained Optimization. *SISC* **16,** 1190 (1995).

228. Starostin, V. *gixi package for GIWAXS data analysis:* https://github.com/schreiber-lab/gixi 2022.

229. Huang, X., Jamonnak, S., Zhao, Y., Wang, B., Hoai, M., Yager, K. G. & Xu, W. Interactive Visual Study of Multiple Attributes Learning Model of X-Ray Scattering Images. *IEEE Trans. Vis. Comput. Graph.* **27,** 1312 (2021).

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

231. Jha, D., Kusne, A. G., Al-Bahrani, R., Nguyen, N., Liao, W., Choudhary, A. & Agrawal, A. *Peak Area Detection Network for Directly Learning Phase Regions from Raw X-ray Diffraction Patterns* in *2019 Proc. Int. Jt. Conf. Neural Netw.* (IEEE, 2019).

232. Ziletti, A., Kumar, D., Scheffler, M. & Ghiringhelli, L. M. Insightful classification of crystal structures using deep learning. *Nat. Commun.* **9,** 2775 (2018).

233. Ryan, K., Lengyel, J. & Shatruk, M. Crystal Structure Prediction via Deep Learning. *J. Am. Chem. Soc.* **140,** 10158 (2018).

234. Ren, P., Xiao, Y., Chang, X., Huang, P.-Y., Li, Z., Gupta, B. B., Chen, X. & Wang, X. A survey of deep active learning. *ACM Comput. Surv. (CSUR)* **54,** 1 (2021).

235. Wang, C. & Golland, P. Discretization Invariant Learning on Neural Fields. *arXiv,* 2206.01178 (2022).

# List of own publications

1. Starostin, V. *et al.* Resolving the Phase Problem in Reflectometry with Prior Aware Neural Posterior Estimation. in preparation (2023).
2. Ragulskaya, A., Starostin, V., *et al.* On the analysis of two-time correlation function. in preparation (2023).
3. Dax, I., Zaluzhnyy, I. A., Pylypenko, A., Russegger, N., Starostin, V., Rysov, R., Westermeier, F., Sprung, M., Hinderhofer, A., Pithan, L. & Schreiber, F. Insight into Heterogeneous Dynamics of Growing Islands using Coherent X-ray Scattering. submitted (2023).
4. Munteanu, V., Starostin, V., Greco, A., Pithan, L., Gerlach, A., Hinderhofer, A., Kowarik, S. & Schreiber, F. Neural network analysis of neutron and X-ray reflectivity data: Incorporating prior knowledge for tackling the phase problem. *arXiv.* preprint, 2307.05364 (2023).
5. Pithan, L., Starostin, V., Mareček, D., Petersdorf, L., Völter, C., Munteanu, V., Jankowski, M., Konovalov, O., Gerlach, A., Hinderhofer, A., Murphy, B., Kowarik, S. & Schreiber, F. Closing the loop: Autonomous experiments enabled by machine-learning-based online data analysis in synchrotron beamline environments. *arXiv.* preprint, 2306.11899 (2023).
6. McCluskey, A. R., Caruana, A. J., Kinane, C. J., Armstrong, A. J., Arnold, T., Cooper, J. F. K., Cortie, D. L., Hughes, A. V., Moulin, J.-F., Nelson, A. R. J., Potrzebowski, W. & Starostin, V.. Advice on describing Bayesian analysis of neutron and X-ray reflectometry. *J. Appl. Crystallogr.* **56,** 12 (2023).

7. Hinderhofer, A., Greco, A., Starostin V., Munteanu, V., Pithan, L., Gerlach, A. & Schreiber, F. Machine Learning for Scattering Data: Strategies, Perspectives, and Applications to Surface Scattering. *J. Appl. Crystallogr.* **56,** 3 (2023).

8. Starostin, V., Pithan, L., Greco, A., Munteanu, V., Gerlach, A., Hinderhofer, A. & Schreiber, F. End-to-End Deep Learning Pipeline for Real-Time Processing of Surface Scattering Data at Synchrotron Facilities. *Synchrotron Radiat. News* **35,** 21 (2022).

9. Starostin V., Munteanu, V., Greco, A., Kneschaurek, E., Pleli, A., Bertram, F., Gerlach, A., Hinderhofer, A. & Schreiber, F. Tracking perovskite crystallization via deep learning-based feature detection on 2D X-ray scattering data. *npj Comput. Mater.* **8,** 101 (2022).

10. Greco, A., Starostin V., Edel, E., Munteanu, V., Russegger, N., Dax, I., Shen, C., Bertram, F., Hinderhofer, A., Gerlach, A. & Schreiber, F. Neural network analysis of neutron and X-ray reflectivity data: automated analysis using mlreflect, experimental errors and feature engineering. *J. Appl. Crystallogr.* **55,** 362 (2022).

11. Timmermann, S., Starostin, V., Girelli, A., Ragulskaya, A., Rahmann, H., Reiser, M., Begam, N., Randolph, L., Sprung, M., Westermeier, F., Zhang, F., Schreiber, F. & Gutt, C. Automated matching of two-time X-ray photon correlation maps from phase-separating proteins with Cahn–Hilliard-type simulations using auto-encoder networks. *J. Appl. Crystallogr.* **55,** 751 (2022).

12. Ragulskaya, A., Starostin, V., Begam, N., Girelli, A., Rahmann, H., Reiser, M., Westermeier, F., Sprung, M., Zhang, F., Gutt, C. & Schreiber, F. Reverse-engineering method for XPCS studies of non-equilibrium dynamics. *IUCrJ* **9,** 439 (2022).

13. Greco, A., Starostin V., Hinderhofer, A., Gerlach, A., Skoda, M. W. A., Kowarik, S. & Schreiber, F. Neural network analysis of neutron and x-ray reflectivity

data: pathological cases, performance and perspectives. *Mach. Learn.: Sci. Technol.* **2,** 045003 (2021).

14. Greco, A., Starostin V., Karapanagiotis, C., Hinderhofer, A., Gerlach, A., Pithan, L., Liehr, S., Schreiber, F. & Kowarik, S. Fast fitting of reflectivity data of growing thin films using neural networks. *J. Appl. Crystallogr.* **52,** 1342 (2019).

15. Potemkin, F. V., Bravy, B. G., Bezsudnova, Y. I., Mareev, E. I., Starostin, V., Platonenko, V. T. & Gordienko, V. M. Overcritical plasma ignition and diagnostics from oncoming interaction of two color low energy tightly focused femtosecond laser pulses inside fused silica. *Laser Phys. Lett.* **13,** 045402 (2016).

# List of acronyms

**CNN** convolutional neural network 48, 55–57, 74

**DE** differential evolution 29, 30, 36, 63, 90

**DESY** Deutsches Elektronen-Synchrotron 144

**ESRF** European Synchrotron Radiation Facility 76, 78, 144

**FC** fully-connected 45, 46, 57

**GAN** Generative Adversarial Network 58

**GELU** Gaussian Error Linear Unit 47, 74, 75

**GISANS** grazing-incidence small-angle neutron scattering 21

**GISAS** grazing-incidence small-angle scattering 20, 38

**GISAXS** grazing-incidence small-angle X-ray scattering 21

**GIWANS** grazing-incidence wide-angle neutron scattering 11

**GIWAS** grazing-incidence wide-angle scattering 2–4, 149, 153, 154

**GIWAXS** grazing-incidence wide-angle X-ray scattering 11, 15, 20–23, 117, 118, 120, 129, 131–135, 139, 144, 146, 147

**IoU** Intersection over Union 52, 54

*List of acronyms*

**IS** Importance Sampling 32, 34, 87, 90, 91, 93, 100, 104

**KL** Kullback–Leibler 41, 42, 62, 92

**LMS** least mean squares 78

**LReLU** leaky rectified linear unit 47

**MC** Monte Carlo 100, 104, 106, 109, 110

**MCMC** Markov Chain Monte Carlo 3, 34–36, 90, 93, 94, 97, 103, 104, 110, 153

**ML** machine learning 25, 37, 40, 41, 63, 64, 66, 73, 74, 76–87, 92, 109, 115, 149, 151, 152

**MLE** Maximum Likelihood Estimation 2–4, 25, 27, 28, 30, 31, 40, 63, 64, 87, 90, 97, 149–151

**MLP** multi-layer perceptron 46, 73–75

**MSE** mean squared error 40, 54

**NN** neural network 1, 3, 37, 39, 42, 43, 45–47, 49, 55, 58, 60–62, 65, 72–76, 80, 92, 117, 118, 151

**NPE** Neural Posterior Estimation 92, 93, 96, 105, 151

**NR** neutron reflectivity 11, 16, 73, 75

**OSC** organic semiconductor 2

**PANPE** Prior Aware Neural Posterior Estimation 93, 94, 96, 97, 100, 103–106, 108, 110, 111, 151

**QCM** Quartz Crystal Microbalance 76, 80, 82, 83

**ReLU** rectified linear unit 46, 47

**SAS** small-angle scattering 38

**SGD** stochastic gradient descent 42

**SLD** scattering length density 7, 11, 14–16, 18, 19, 25, 27, 64, 71, 72, 80, 85

**VAE** Variational Autoencoder 58

**XRR** X-ray reflectivity 11, 16, 73, 75–78, 150

**YOLO** You Only Look Once 57