

Politecnico di Torino



Politecnico di Torino

Master of Science in Biomedical Engineering

Accademic Year 2025/2026

Graduation Session September 2026

Development and integration of a neural network to detect image misalignment in cryo-electron microscopy

Supervisors:

Massimo Salvi

Carlos Oscar Sánchez Sorzano

David Herreros Calero

Candidate:

Francesca Cavallo

Abstract

In cryo-EM, particle misalignment —- arising from factors such as preferred orientation, low signal-to-noise ratio, or poor initial models —- affects the quality of 3D reconstruction and refinement. While several deep learning tools have improved map resolution, map anisotropy, pose estimation accuracy, or have incorporated quality control modules as part of the pipeline, no dedicated validation tool has been proposed to detect particle misalignment independently of its cause.

The present study proposes *cryoCheck*, a supervised deep learning approach designed to be used as a validation tool to detect particle misalignment in the cryo-EM pipeline. *CryoCheck* employs an MLP architecture trained on Fourier Ring Correlation (FRC) curves computed between simulated projections and their corresponding experimental images.

Once the pose parameters of each cryo-EM image in the dataset are extracted, they are used to generate an aligned projection of a reference 3D density map, as well as a misaligned projection obtained by perturbing these parameters. For each of the two resulting projections, an FRC curve is computed against the corresponding experimental image, and each curve is independently provided as input to the neural network, which classifies it as either aligned or misaligned based on how rapidly it decays to zero, a direct measure of the mismatch between the projection and the experimental image. Three datasets of increasing complexity were used to train the model. It achieved 99% accuracy in misalignment detection on both the simulated and ribosome datasets, and 98% accuracy on the SARS-CoV-2 Spike dataset, by applying random misalignment to the validation subset in terms of both Euler angles, with a minimum detectable rotation of about 2–3 degrees, depending on the dataset, and shifts, with a minimum detectable value of 2 pixels.

The use of FRC curves as input allows the extraction of frequency-domain information directly from the projections, avoiding the need to feed raw images into the model, which considerably reduces computational costs. In conclusion, *cryoCheck* can be introduced in cryo-EM workflows to automate the detection of misaligned particles, supporting higher quality 3D reconstructions and speeding up the refinement process. Future work could focus on training the model on larger datasets and evaluating its applicability to real-world cryo-EM systems.

Table of Contents

List of Tables 4

List of Figures 5

1 Introduction 7

2 Cryo-EM 8
- 2.1 Cryo-Electron Microscopy for Structural and Dynamic Macromolecular Analysis . . . 8
- 2.2 Sample Preparation . . . 9
- 2.3 Single Particle Analysis (SPA) . . . 10
  - 2.3.1 Movie Alignment . . . 11
  - 2.3.2 CTF estimation . . . 11
  - 2.3.3 Particle Picking . . . 12
  - 2.3.4 2D Classification and 3D Reconstruction . . . 13
- 2.4 Source of Misalignment in Cryo-EM . . . 14
- 2.5 3D Map Validation . . . 14

3 State of The Art 16

4 Materials and Methods 18
- 4.1 The JAX Framework . . . 18
- 4.2 Dataset Variants . . . 19
- 4.3 Data Preprocessing . . . 20
  - 4.3.1 Input Data and Metadata Extraction . . . 20
  - 4.3.2 Volume Intensity Adjustment via Gaussian Splatting . . . 21
  - 4.3.3 Volume Projection . . . 22
  - 4.3.4 Controlled Misalignment as Data Augmentation . . . 23
  - 4.3.5 Fourier Ring Correlation . . . 25
- 4.4 Neural Network Architecture . . . 26
- 4.5 Training and Validation . . . 27
- 4.6 Scipion Implementation . . . 28
- 4.7 Evaluation Metrics . . . 30

5 Results 33
5.1 Simulated Data 33
5.2 Ribosome Data 37
5.3 Spike Data 40
5.3.1 Testing Pose Parameters Perturbations 43
5.3.2 Generalization Across Datasets 45

6 Conclusions 47

7 Future Outlooks 48

Bibliography 50

List of Tables

4.1 Misalignment perturbation parameters for each dataset. 24

5.1 Training hyperparameters used for the simulated dataset. 33
5.2 Classification metrics for the simulated dataset. 34
5.3 Training hyperparameters used for the ribosome dataset. 37
5.4 Classification metrics for the ribosome dataset. 38
5.5 Training hyperparameters used for the spike dataset. 41
5.6 Classification metrics for the spike dataset. 41
5.7 Euler angle perturbation ranges used to test the model. 43
5.8 Shift perturbation ranges used to test the model. 43
5.9 Classification metrics in prediction mode 45

List of Figures

2.1 Workflow of SPA in cryo-EM. Adapted from [6]. 10
2.2 Schematic illustration of the image formation model described by Equation 2.1. The Fourier representations of the PSF and the object are multiplied and the inverse Fourier transform is then applied to recover the final observed image [9]. 12
2.3 Particle picking on a cryo-EM micrograph [13]. 12
2.4 Individual cryo-EM images are averaged into 2D class averages (left), which are then combined across different viewing angles to reconstruct the 3D structure (right) [14]. 13

4.1 Cryo-EM images belonging to the simulated dataset (left), the ribosome dataset (center) and the SARS-CoV Spike Protein dataset (right). . . . 20
4.2 Comparison between an experimental image from the SARS-CoV-2 Spike protein dataset (left) and a projection of the reference 3D volume, generated with the `volumeProjection` function (right). 23
4.3 Comparison between an experimental image from the SARS-CoV-2 Spike protein dataset (left) and a projection of the reference 3D volume, generated with the `volumeProjection` function using perturbed pose parameters (right). 24
4.4 FRC curves computed between the experimental image and the corresponding aligned and misaligned projections shown, respectively, in Figure 4.2 and 4.3 26
4.5 cryoCheck MLP architecture. 27
4.6 The CryoCheck protocol form within Scipion's GUI, showing the Input and the Network sections required to run the misalignment detection tool. 29
4.7 Scipion configuration used to run the cryoCheck model. 30
4.8 Example of a confusion matrix belonging to the validation step of the simulated dataset: 96 particles were correctly classified as aligned, 95 particles were correctly classified as misaligned, and 1 particle was misclassified as aligned. 31
4.9 Youden's index selection [53]. 32

5.1 Cryo-EM image (ID 497) belonging to the simulated dataset (CTF free) (left), its aligned projection (center) and its misaligned projection (right). 33
5.2 Comparison between the aligned and misaligned pose parameters for particle 497, together with the corresponding FRC curves. 34

5.3 ROC curve (up) and confusion matrix (bottom) to evaluate cryoCheck performance. 35
5.4 Training and validation loss curves. 36
5.5 Cryo-EM image (ID 47917) belonging to the ribosome dataset (left), its aligned projection (center) and its misaligned projection (right). 37
5.6 Comparison between the aligned and misaligned pose parameters for particle 47917, together with the corresponding FRC curves. 38
5.7 ROC curve (up) and confusion matrix (bottom) to evaluate cryoCheck performance. 39
5.8 Training and validation loss curves. 40
5.9 Cryo-EM image (ID 173749) belonging to the spike dataset (left), its aligned projection (center) and its misaligned projection (right). 40
5.10 Comparison between the aligned and misaligned pose parameters for particle 173749, together with the corresponding FRC curves. 41
5.11 ROC curve (up) and confusion matrix (bottom) to evaluate cryoCheck performance. 42
5.12 Training and validation loss curves. 43
5.13 Overlapping ROC curves belonging the to model trained at different Euler angles perturbations ranges (left) and shifts ranges (right). 44
5.14 FRC curves at angular perturbation Range 1 44
5.15 FRC curves at angular perturbation Range 2 45
5.16 FRC curves at shift perturbation Range 1 45
5.17 Confusion matrix belonging to cryoCheck in predict mode 46

1. Introduction

Proteins are macromolecules made of long chains of amino acids, involved in many biological processes whose different functionalities can be derived from their three-dimensional (3D) atomic structure. Proteins fold into their so-called native state, which corresponds to the most stable conformation under physiological conditions; four different structural levels can be distinguished: primary, secondary, tertiary, and quaternary. They also form larger structures with other proteins or macromolecules, referred to as complexes.

Nowadays, protein structure prediction has become a pivotal challenge in structural biology, drug design, treatment development, numerous biomedical domains, antibody design, and vaccine research. Cryo-EM is the leading technique for resolving high-resolution protein density maps used to derive macromolecular atomic models; it requires only a small amount of sample, does not require crystallization, and preserves the complex's native state thanks to vitrification.

As in most scientific fields, structural biology, and cryo-EM in particular, has largely embraced deep learning to complement its objectives and reduce computational time and cost. Focusing on cryo-EM, deep learning tools have been introduced at every step of the pipeline, ranging from sample preparation — where automated tools assist technicians in defining optimal acquisition modalities and parameters — to 3D density map generation and refinement. 3D reconstruction is a critical step in cryo-EM, where each particle image needs to be correctly aligned to obtain a reliable density map. A particle image is considered correctly aligned when its estimated pose parameters accurately reflect the true orientation and position of the particle at the moment it was projected by the microscope to form the image. If the estimated pose parameters are incorrect, a misaligned projection of the particle is obtained. This can occur due to several factors related to the acquisition process and image processing pipeline: the preferred orientation phenomenon, which limits the range of orientations captured during acquisition and can favor the so-called "attraction problem"; inaccurate estimation of acquisition parameters, such as the CTF; and the intrinsically low signal-to-noise ratio (SNR) that characterizes cryo-EM images. It is intuitive to conclude that an unreliable 3D reconstruction leads to a series of consequences that ultimately compromise the accuracy of the derived atomic models, with potential downstream effects on the structural interpretation of the macromolecule and, more broadly, on applications such as drug design and vaccine development that rely on this information. This concern is not only theoretical: refinement runs performed with different alignment algorithms on the same set of particle images have been reported to yield disagreement in the estimated pose for 20 to 50% of particles [1], highlighting the need for dedicated tools capable of assessing alignment quality.

2. Cryo-EM

2.1 Cryo-Electron Microscopy for Structural and Dynamic Macromolecular Analysis

Cryo-electron microscopy (cryo-EM) is a high-resolution imaging technique that enables three-dimensional visualization of biological macromolecules in their *near-native*, *hydrated environment*. Structural biology techniques such as X-ray crystallography have traditionally required the formation of high-quality crystals, which can be difficult or impossible to obtain for many biological samples. Cryo-EM overcomes this limitation, allowing for direct observation of isolated protein molecules and large complexes with a significantly smaller amount of sample. This approach is particularly important because biological entities typically function as intact structures in fully or partially hydrated environments, with large surface areas exposed to water [2].
The fundamental innovation of the method is *vitrification*, a process developed in the 1980s by Jacques Dubochet and colleagues. In this technique, a thin film of biological specimen is flash-frozen so rapidly that water molecules do not have time to form ice crystals. Instead, the water enters a glass-like amorphous state, which preserves the specimen's native conformation. Because different molecules are captured in different conformational states at the moment of vitrification, transmission electron microscopy (TEM) imaging of the vitrified sample can reveal multiple functional states within a single dataset, offering rich insight into the dynamic behavior of biological nanomachines.

The two main Cryo-EM techniques are [3]:

1. **Single Particle Analysis (SPA)**: This is the most common approach for purified samples. It involves averaging thousands of unique 2D projections of identical particles that are frozen in random orientations to generate a high-resolution 3D reconstruction.

2. **Cryo-electron Tomography (cryo-ET)**: This method is used to study unique objects, such as organelles or viruses, in their original cellular context (in situ), by tilting the specimen stage and taking a series of images.

In the last decade, the field has undergone a **"resolution revolution"** that earned three of its key developers the 2017 Nobel Prize in Chemistry. This breakthrough was driven by two major advancements [2]:

1. **Direct Electron Detectors (DEDs)**: These sensors can detect electrons directly

with high frame rates, allowing for "movie mode" acquisition that compensates for beam-induced movement and significantly improves the signal-to-noise ratio.

2. **Sophisticated Image-Processing Algorithms**: Advanced software can now align, classify, and average hundreds of thousands of noisy 2D projections to reconstruct 3D maps at near-atomic resolution.

2.2 Sample Preparation

Sample preparation is widely considered the primary bottleneck in cryo-EM. Molecular structural biology (*in vitro*) and cellular structural biology (*in situ*) have somewhat different sample preparation requirements, but share some common key points [4]:

- The sample must be *thin enough* for the electron beam to penetrate. The ideal molecular cryo-EM sample has an ice thickness of <50-100 nm, while, for cellular specimens this value can reach up to several hundred nanometers. The thickness limitation, and consequently the maximum observable object size, are set by the penetration limit of the electron beam, which is approximately 500 nm for a 300kV TEM.

- *Vitrification* is required for both types of samples to preserve their near-native, hydrated conformation. It also allows the hydrated sample to withstand the high-vacuum environment inside the electron microscope column, needed to avoid strong scattering events, and slows down radiation damage.

The practical steps of this workflow begin with the choice of the support: electron microscopy (EM) grids, typically 3 mm in diameter, are made of conductive materials such as copper, gold, or molybdenum. These are covered with a perforated support film, most commonly a thin layer of holey carbon or gold, featuring regular hole arrays. Once the target specimen is extracted, it is suspended in water and dropped across the grid. A robotic arm plunges the grid into a cryogen—usually liquid ethane or an ethane-propane mix—precooled by liquid nitrogen and the water, along with the suspended specimen, vitrifies almost instantaneously. The cooling rate is so fast that water molecules are "trapped" in a glass-like amorphous state before they can form ice crystals: an ideal vitrified specimen is easily identifiable and randomly oriented across a wide range of angles with molecules don't touching each others [5] . The vitrified grids are preserved in liquid nitrogen to prevent recrystallization until they are loaded into the microscope. A cryo-transfer holder, equipped with a liquid nitrogen reservoir, is used to load the ice-embedded specimen into the microscope column while maintaining the frozen state and avoiding contamination. Finally, high-resolution TEM images are acquired in **low-dose mode** to minimize radiation damage to delicate biological structures [2].

2.3 Single Particle Analysis (SPA)

Single Particle Analysis (SPA) is a crucial method for resolving the near-atomic structures of purified macromolecular complexes, with the ultimate goal of decoding their structure-function relationships. The principle of SPA is based on two hypotheses [3]:

1. **Identical copies assumption**: it assumes that all imaged particles are multiple copies of an ideal canonical complex in the *same conformational state*.

2. **Projection assumption**: because the specimens are vitrified in random orientations, the acquired images represent unique 2D projections of the 3D complex from different directions, eliminating the need for physical stage tilting.

These two assumptions guarantee that 2D projection images of the same object can be computationally combined to generate 3D structures of biological macromolecules.
The major challenge related to SPA is *noisy image datasets* (SNR = 0.1 – 0.01); the amorphous ice layer around the specimen is the main source of noise and because it is independent in each particle, this noise can be effectively suppressed by statistical power of averaging. This fundamental strategy relies on the fact that while random noise tends to cancel out when thousands of images are combined, the signal from identical protein particles adds up coherently [2]. However, before the averaging process can effectively recover structural information, several pre-processing steps must be implemented to correct experimental and optical artifacts that would otherwise compromise the data accuracy. The following sections describe these steps in sequence, as shown in Figure 2.1.

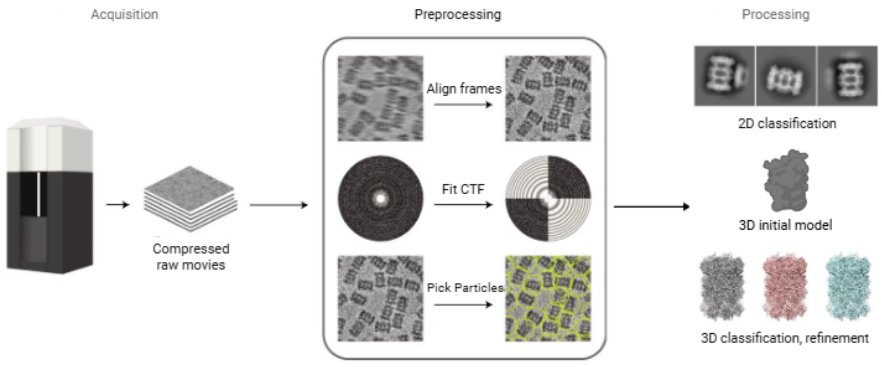


Figure 2.1: *Workflow of SPA in cryo-EM. Adapted from [6].*

2.3.1 Movie Alignment

The SPA image pre-processing workflow starts with *movie alignment* to compensate for the **beam-induced motion (BIM)** phenomenon, which causes the specimen to move during electron exposure. To mitigate it, images are acquired as movies. Movie alignment algorithms align these individual frames to correct for local or global displacements, thereby eliminating motion blur. Dose weighting is also applied; this process assigns higher weights to the initial frames—which have suffered the least radiation damage—to preserve high-frequency atomic details. The motion corrected movies are called *micrographs* [3].

2.3.2 CTF estimation

The interaction between the electron beam and the specimen produces two types of electron waves: an unscattered wave, whose electrons have not interacted with the sample, and a scattered wave, whose phase is altered by this interaction. Because biological samples are composed of light atoms, they scatter electrons weakly and absorb almost none of them; as a result, they are referred to as "weak phase objects," meaning that structural information is encoded almost entirely in the phase of the scattered wave rather than in its amplitude. This is the reason why Cryo-EM images are intentionally acquired under a *high level of defocus* to generate the necessary phase contrast. However, this defocus, together with unavoidable imperfections in the microscope's electron optics — known as *aberrations* —distorts the image as it is formed.
In real space, this distortion is described by the Point Spread Function (PSF): the image of a single point, which would ideally be a perfect projection of that point, appears in practice as a pattern of concentric rings surrounding it. The same distortion, expressed instead in terms of spatial frequency, is described by the **Contrast Transfer Function (CTF)**—the Fourier transform of the PSF [7]. Working in Fourier space is preferred because it simplifies the mathematical description of the microscope's effect on the image: rather than requiring a convolution between the object and the PSF, the operation becomes a simple multiplication between the Fourier transform of the object *(O)* and the CTF:

$$I(f) = O(f) \cdot \mathrm{CTF}(f) \tag{2.1}$$

The CTF is modeled as:

$$CTF(k, \lambda, \Delta f, C_s) = -sin\left(\pi\lambda|k|^2\Delta f - \frac{\pi\lambda^3|k|^4 C_s}{2} + \alpha\right) \cdot E(k)$$

where k is the frequency index, α is a small phase shift term, λ is the electron wavelength, Δf is the objective defocus, C_s is the spherical aberration and $(E|k|)$ is an exponentially decaying envelope function [8]. CTF estimation is a mandatory step in SPA because it

makes it possible to correct the degradation introduced by the microscope and to restore the delocalized information back to its correct location. Algorithms estimate the CTF of micrographs by fitting the experimental Power Spectral Density (PSD)—also known as the power spectrum, $|I(f)|^2$ to the theoretical model, $|CTF|^2$ [3].

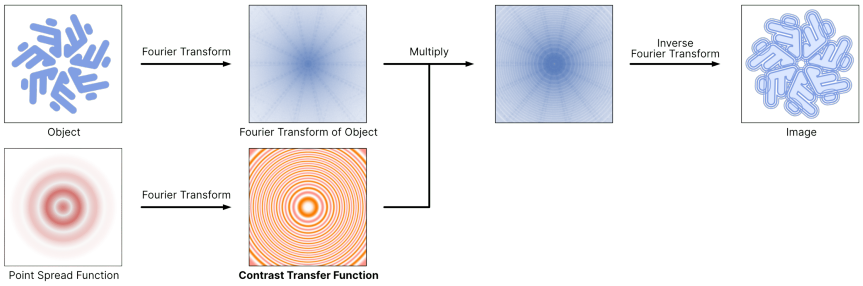


Figure 2.2: *Schematic illustration of the image formation model described by Equation 2.1. The Fourier representations of the PSF and the object are multiplied and the inverse Fourier transform is then applied to recover the final observed image [9].*

2.3.3 Particle Picking

Once the micrographs are corrected, several different views of the same object under study, called *particles*, can be cropped from the image in a manual, semi-automated or fully automated way. Picking methods have also been affected by recent advances in machine learning to identify particles with minimal human intervention (e.g. DeepPicker [10], Topaz [11], crYOLO [12]). Even with advanced algorithms, the initial set of picked particles often contains "junk," such as carbon edges, ice crystals, or protein aggregates. Once the coordinates of the particles are detected, the corresponding regions are extracted from the micrograph as individual boxed images.

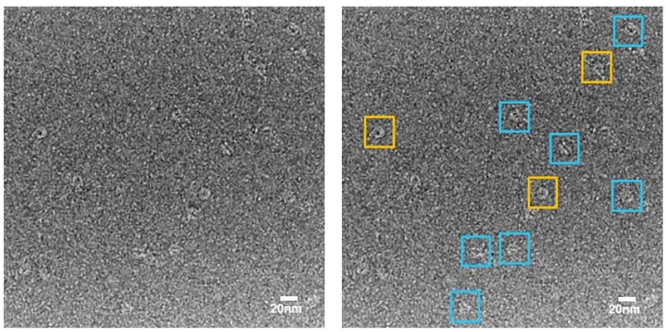


Figure 2.3: *Particle picking on a cryo-EM micrograph [13].*

2.3.4 2D Classification and 3D Reconstruction

2D classification groups the picked particles into classes according to their viewing angle. Class representatives are obtained by computing the average of all particles in a class. 2D classes can be used to:

- Sort "junk" items into dedicated classes. Usually, this step is re-iterated to clean the dataset as much as possible.
- Obtain class representatives that have a higher SNR than single particle images.
- Identify good-quality classes, recognizable by visible secondary structures, and detect symmetry within the particle.

The primary challenge in 3D reconstruction lies in determining the pose parameters (three Euler angles and two in-plane shifts) of the 2D projection images to later invert the projection process [3] using back-projection algorithms. 3D reconstruction algorithms are based on the **Central Section Theorem**, which states that 2D Fourier transforms of a projection of a 3D object correspond to a plane centered at the origin of the 3D transform of the object, oriented according to the projection's viewing angle [2].
Once the pose parameters of enough particles have been estimated, their corresponding Fourier sections can be combined to progressively fill the 3D Fourier space, from which the object's structure is recovered via an inverse Fourier transform.

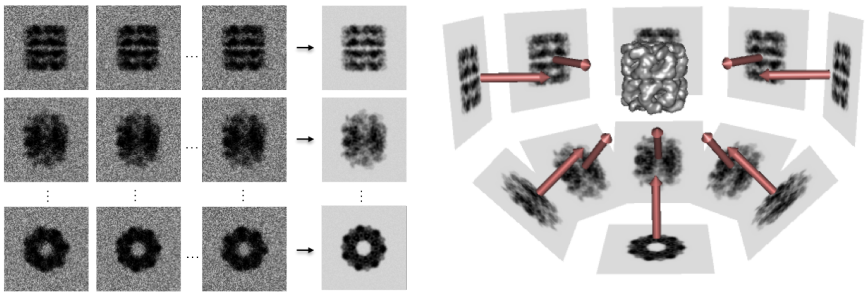


Figure 2.4: *Individual cryo-EM images are averaged into 2D class averages (left), which are then combined across different viewing angles to reconstruct the 3D structure (right) [14].*

However, because the initial pose estimates are typically imprecise, an iterative refinement process is required: the **map refinement** step is an iterative process that determines the precise relative orientation of 2D particles with respect to a 3D reference volume to reconstruct the macromolecular structure at high resolution. Modern software, such as cryoSPARC [15], uses algorithms like stochastic gradient descent to generate accurate *ab initio* 3D models from 2D classes without requiring a pre-existing reference. Also, RELION [16] has become the dominant choice thanks to the introduction of

Bayesian/maximum likelihood methods, which have made it possible to handle the high noise levels in the images [4].

2.4 Source of Misalignment in Cryo-EM

Although the iterative refinement process is designed to improve both the pose estimates and the resolution of the reconstructed 3D model, this convergence is not always reliable: if the assignment of incorrect pose parameters occurs, it contributes to creating erroneous Fourier sections for the 3D reconstruction. **Particle misalignment** can introduce artifacts or degrade the resolution of specific regions of the final map. The main sources of misalignment include the following:

- **Attraction problem**: this is a source of systematic error where images are incorrectly assigned to certain references (2D classes or 3D projections). Because references with more assigned images have less noise due to averaging (higher SNR), they mathematically "attract" more experimental images, even if those images do not actually match the signal, creating a biased angular distribution [17]. The preferred orientation phenomenon [18] can worsen the attraction problem: if biological macromolecules adopt a limited set of orientations in the cryo-EM sample, due to their interaction with the air-water interface (AWI), the resulting over-represented views accumulate a disproportionately higher SNR, further strengthening their tendency to attract incorrectly assigned images.

- **Low SNR**: it makes it hard to distinguish the signal from the background noise, leading pose parameters estimation to be intrinsically difficult.

- **CTF and acquisition parameter errors**: systematic errors in estimating CTF — such as incorrect defocus values, beam tilt, or inaccurate pixel size—cause miscorrections of the phase flip [17].

- **Poor Initial Model**: the quality of the initial reference model can substantially bias the convergence of the refinement process, potentially producing distorted or entirely incorrect reconstructions [19].

2.5 3D Map Validation

The final step of this workflow is the **validation** of the map. While the field still lacks a single metric to evaluate the consistency of a map and its resolution, the standard metric is the **Fourier Shell Correlation (FSC)** curve. It consists in splitting the cryo-EM dataset into two subsets, generating a 3D model from each of them, and then computing, for each spatial frequency shell, their correlation. At low frequencies the correlation is

typically high, since the true signal dominates the independent noise in both half-maps; at high frequencies it progressively decays as noise increasingly outweighs the signal. The resolution of the map is conventionally defined as the spatial frequency at which the FSC curve drops below a threshold value (commonly 0.143). The shape of the curve can reveal issues affecting the reconstruction, such as heterogeneity, an uneven Euler angle distribution, or overfitting. It is important to state that FSC is not a measure of the reliability of the reconstructed model since any bias or artifact can be shared between the two independent subsets of particles [20].

3. State of The Art

This section examines the integration of Deep Learning (DL) tools in cryo-EM pipelines, emphasizing how their adoption in specific stages of the analysis can automate the workflow, improve the resulting reconstructions and reduce human bias.
Particle picking is a critical step in the cryo-EM workflow because, as already mentioned in section 2.3.3, it is particularly prone to false positives; as a result, it benefits from a large number of DL-based methods. Supervised CNNs, such as crYOLO [12] and DeepPicker [10], are trained on labeled datasets, while semi-supervised CNNs, such as Topaz [11], identify regions of the micrograph most likely to contain particles without requiring fully labeled training data. AutoCryoPicker [21], on the contrary, is an unsupervised method that clusters particles based on their visual features. The current state of the art is represented by CryoSegNet [22], which implements a hybrid strategy, coupling a supervised U-Net with a semi-supervised foundation model, the Segment Anything Model (SAM) [23]. Auxiliary DL tools such as micrographCleaner [24] (U-Net segmentation) and deepConsensus [25] (CNN classifier) further reduce false positives by automatically masking carbon edges, ice contamination, and aggregates [3].

2D classification works by iteratively assigning particles to a fixed number of class representatives. Since this is inherently an unsupervised problem, and unsupervised DL models remain more complex to design and less widely adopted, this step is still dominated by traditional analytical methods such as RELION [16], CryoSPARC, and ISAC [3]. Deep learning currently plays only a minor role here, mainly limited to classifying and discarding low-quality ("junk") 2D class averages.

Several recent works have proposed deep learning-based approaches for joint pose estimation and 3D reconstruction, presenting varying assumptions regarding macromolecular homogeneity. Based on the overview proposed by [26], early autoencoder-based frameworks such as cryoPoseNet [27], cryoAI [28], and CryoGAN [29], based on a GAN architecture, operate under the assumption that the macromolecule adopts a single, rigid conformation. Moving beyond rigid-volume reconstructions, state-of-the-art tools model continuous, non-rigid protein dynamics and transitions, such as cryoDRGN2 [30] and cryoFIRE [31], which estimate poses and conformational states; 3DFlex [32], which maps flexible proteins; CryoTRANS [33], which predicts high-resolution maps of rare conformations; CryoSTAR [34], which tackles heterogeneous reconstruction; and HetSIREN [35], which reconstructs heterogeneous conformational landscapes directly in real space, decoupling structural variations from pose and CTF estimations.

The atomic model building process represents the final goal of the entire cryo-EM pipeline.

However, Cryo-EM maps usually suffer loss of contrast, particularly at high resolution. To make 3D maps more reliable to use as reference models, DeepEMhancer [36], based on U-net architecture, can be used to enhance their quality and interpretability without requiring an atomic model, a mask, or a local resolution estimate as input, unlike earlier sharpening approaches. EMReady2 [37], based on a bidirectional Mamba-convolutional architecture, has also been proposed as a state-of-the-art alternative for cryo-EM volume post-processing. It accounts for local resolution heterogeneity for both cryo-EM SPA and cryo-ET modality.

Moreover, AlphaFold [38], whose development was awarded the 2024 Nobel Prize in Chemistry, is a DL approach capable of solving the 3D structure of a protein, directly from its amino acid sequence. It can be integrated into cryo-EM workflows to further refine the 3D structures obtained from experimental density maps.

Some tools incorporate mechanisms for assessing prediction reliability; cryoPARES [1] is a pose estimation model that embeds a pruning criterion by computing a z-score from the network's predicted pose distribution to discard unreliable particles, while spIsoNet [39] addresses the preferred-orientation problem by including a Misalignment Correction module that improves particle alignment accuracy from the information recovered from well-sampled orientations.

4. Materials and Methods

Building on the challenges discussed in the previous chapter, this thesis proposes a validation network designed to detect particle misalignment in cryo-EM SPA. This chapter describes the materials and methods developed to this end.
Section 4.1 introduces the computational framework and software tools underlying the proposed implementation.
Section 4.2 presents the dataset variants used for training and evaluating the proposed approach.
Section 4.3 details the data preprocessing pipeline, including metadata extraction, Gaussian splatting to adjust the gray levels of the input volume, volume projection, the introduction of controlled misalignment in terms of both Euler angles and shifts and the computation of Fourier Ring Correlation (FRC) profiles, used as input to the network.
Section 4.4 describes the architecture of *CryoCheck*, the supervised deep learning model employed to detect misalignment.
Section 4.5 outlines the training and validation procedures adopted to ensure the generalization capability of the model.
Section 4.6 describes the integration of the proposed method into Scipion, the software framework used to implement and run all the experiments presented in this thesis.
Finally, Section 4.7 introduces the metrics used to evaluate the performance of the proposed method.

4.1 The JAX Framework

The entire data processing and modeling pipeline presented in this thesis is implemented using **JAX** [40], a library for accelerator-oriented array computation that provides a NumPy-like interface, `jax.numpy`, combined with automatic differentiation and just-in-time compilation for high-performance execution on CPU, GPU, or TPU.
Unlike NumPy arrays, **JAX arrays are immutable**: once created, their contents cannot be modified in place. Additionally, JAX arrays can be **sharded**, which means that data can be split and spread across multiple devices, allowing computations to be distributed and executed in parallel.
A central feature of JAX is **just-in-time (JIT) compilation**, enabled through the `jax.jit` function or a `@jit decorator`. The entire sequence of operations is traced on the first run by the XLA compiler before execution, substantially reducing runtime compared to pure Python code. This comes at the cost of requiring static, known array shapes at compile time.

JAX also provides **automatic differentiation** through the `jax.grad` transformation, which computes the gradient of a function with respect to its inputs and can be differentiated as many times as needed. Another function is `jax.value_and_grad`, which is exceptionally appealing for deep learning applications because it computes both the value of the function and its gradient.
To efficiently apply operations across batches of data, JAX offers **automatic vectorization** via the `jax.vmap` transformation. Rather than explicitly looping over the batch dimension (i.e. with a for loop), `jax.vmap` transforms a function written for a single example into a vectorized version.
Finally, JAX adopts a different approach to **pseudorandom number generation** compared to NumPy. Since JAX's functional programming model avoids global mutable state, an explicit random `key` needs to be passed to every random function; this key is never modified, ensuring reproducibility. To generate multiple independent random values, a key must be explicitly split into new subkeys using `jax.random.split`, since reusing the same key always produces identical output.

4.2 Dataset Variants

In order to train and evaluate the proposed misalignment detection network, three datasets of different nature and size were used. The evaluation progresses from controlled, idealized data to large-scale, real-world cryo-EM systems.

Simulated Data

The first dataset consists of a set of 500 CTF-corrupted and noise-free synthetic images, generated from the *adenylate kinase* structure (PDB entry 4AKE) with uniformly distributed poses and CTF information. It can be downloaded directly from the Scipion environment with the following command: `Scipion3 testdata - download FlexHub_Tutorials` [41]. Being free of experimental noise, this dataset provides a controlled setting to calibrate the proposed model.

Ribosome Data

The second dataset is an acquisition of the *P. falciparum* 80S ribosome bound to emetine, available with the EMPIAR-10028 entry. It was pre-processed in Scipion, obtaining a set of around 50,000 images, and it can be downloaded with the same Scipion command used for the simulated dataset. Ribosomes represent a relatively favorable structure to be analyzed in cryo-EM, due to their globular shape, large molecular weight, and high RNA content, which increase their contrast [42], resulting in acquisitions with a relatively high signal-to-noise ratio.

SARS-CoV Spike Protein Data

The last dataset consists of cryo-EM images of the SARS-CoV Spike protein. The cryo-EM datasets were acquired from the same sample at two different vitrification temperatures, 4°C and 37°C and are stored in the PDB database, respectively, with codes 9GDX and 9GDY and in the EM Data Bank as EMD-51279 and EMD-51280 [41]

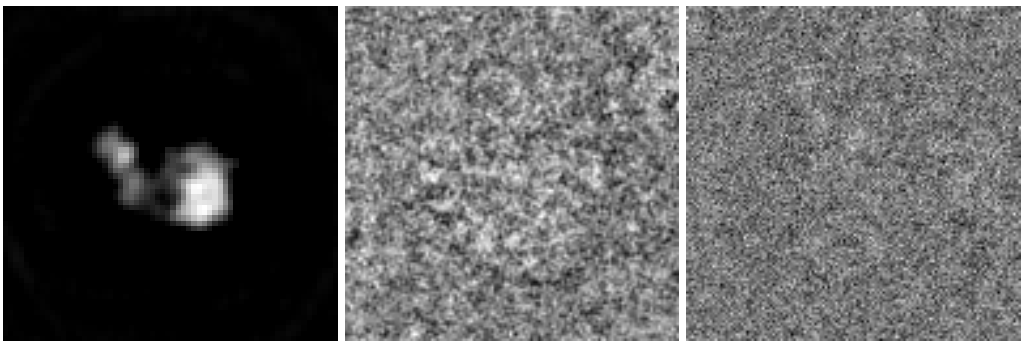


Figure 4.1: *Cryo-EM images belonging to the simulated dataset (left), the ribosome dataset (center) and the SARS-CoV Spike Protein dataset (right).*

4.3 Data Preprocessing

Before being used to train and evaluate the proposed model, all three datasets are preprocessed to ensure data quality and consistency.

4.3.1 Input Data and Metadata Extraction

Each dataset used in this work consists of a reconstructed 3D volume, stored as a `.mrc` file, and its associated set of experimental 2D images, together with a metadata file in the Xmipp `.xmd` format. Depending on the specific experiment, dataset sizes range from approximately 500 to 200,000 particle images with a spatial resolution of 128x128 pixels. Metadata is parsed using the `md_extraction` function, which extracts relevant fields for the preprocessing pipeline—including pose parameters, CTF parameters, identifiers—and organizes them into JAX arrays.
Particle images can be loaded following different strategies, depending on dataset size and available computational resources: entirely into RAM, as memory-mapped shards on disk, or as sharded `ArrayRecord` files—a format that allows efficient parallel reading during training. Regardless of the chosen data storage strategy, data are handled through **Grain** [43], a data-loading library developed by Google specifically for training and evaluating JAX models.

The dataset is split into training and validation subsets according to the `--dataset_split_fraction` argument, which defaults to 0.8/0.2 but can be configured to any desired proportion. Particles are then loaded in batches, with a configurable `--batch_size` argument. For each particle in a batch, both an aligned and a misaligned projection are generated and jointly used for training and validation, resulting in an effective batch size of $2 \times$ `batch_size`, with a balanced 50/50 split between the two classes.

4.3.2 Volume Intensity Adjustment via Gaussian Splatting

Given the reconstructed volume, its intensity values are adjusted to better match the information contained in the experimental images. Volumes in cryo-EM are usually represented as a regular voxel grid (e.g. RELION[16], cryoSPARC[15]) of intensity values but this kind of representation has high requirements in terms of memory and limits its usage in deep learning pipelines for structural analysis [44]. To address this, the volume is represented as a collection of 3D Gaussians and rendered through a differentiable technique known as **Gaussian splatting** [45], using the `fit_volume` function.
A fixed number of Gaussians is initialized inside a volume, followed by an optimization of their positions and amplitudes to reproduce the input volume as closely as possible, with over 20,000 iterations and a learning rate of 0.001. The standard deviation of all the Gaussians is represented by a single global learnable parameter. Once the Gaussian representation of the volume has been obtained, its weights are further refined using the `adjust_weights_to_images` function by comparing the projections of the model against the experimental particle images, accounting for the corresponding CTF of each image. The contrast adjustment follows a linear approximation, such as:

$$V_f = a \cdot V_i + b$$

where, V_i and V_f are the intensity values before and after the adjustment and a, b are the two parameters to be optimized. This adjustment step allows image comparison metrics based on direct subtraction, such as Mean Squared Error (MSE) or Mean Absolute Error (MAE) because they are particularly sensitive to contrast mismatches between the two images being compared. If the intensity scales are not aligned, the global minimum of the residual no longer corresponds to zero, even when the underlying structures match perfectly in shape. This introduces a local minimum that can degrade the optimization process. By correcting for this contrast mismatch, the adjustment step ensures that comparisons are more accurate.
It is worth mentioning that the number of epochs used to adjust the weights was set to 500 for the simulated dataset and to 5 for the other two datasets because, since the experimental datasets usually contain a larger number of images, more updates can be computed per epoch. Therefore, to obtain a comparable number of total optimization steps, the simulated dataset requires a higher number of epochs.

4.3.3 Volume Projection

To compare a candidate 3D volume against the experimental particle images, a projection operator is implemented. The projection is computed by the `volumeProjection` function which, given a volume, a binary mask, and the extracted metadata (Euler angles, in-plane shifts, CTF), generates a batch of simulated 2D projections that can be directly compared against the corresponding experimental images. As mentioned in Section 4.3.1, the `md_extraction` function extracts the metadata associated with each cryo-EM image, relying on an `index` that uniquely identifies every image within the dataset. This guarantees that each simulated projection is generated using the parameters specific to its matching experimental image.
Once the coordinates and intensity values of the voxels inside the mask are extracted, for each image in the batch, they are rotated and shifted according to the corresponding pose parameters. To ensure that the rotation of the volume is with respect to its center and not one of the corners of the mask grid, the coordinates $\mathbf{p} = (x, y, z)$ are first centered by subtracting a factor equal to half of the volume's box size.

$$\mathbf{p}_{\text{centered}} = \mathbf{p} - \frac{N}{2}$$

The rotation of a 3D volume is defined by a series of rotations around three axes of the coordinate system of reference [46]; the corresponding angles are called Euler angles: φ, θ, ψ. A common standard in cryo-EM (e.g. Xmipp) is the ZYZ convention, according to which the rigid object is rotated about Z, then Y, then Z again with respect to an internal coordinate system: the first rotation is around the Z-axis, defined by φ; the second rotation is around the Y-axis, defined by θ; and the third rotation, again around the Z-axis, is defined by ψ.
To apply such a rotation to the volume coordinates, these are multiplied by a rotation matrix $\mathbf{R}$, obtained as the product of three elementary rotation matrices [47], defined as:

$$R_z(\varphi) = \begin{pmatrix} \cos\varphi & \sin\varphi & 0 \\ -\sin\varphi & \cos\varphi & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$R_y(\theta) = \begin{pmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{pmatrix}$$

$$R_z(\psi) = \begin{pmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$R = R_z(\psi) \cdot R_y(\theta) \cdot R_z(\varphi)$$

Once the voxel coordinates have been rotated, the projection is obtained by only retaining the x and y components. These coordinates are then corrected by subtracting the estimated in-plane shift $\mathbf{\Delta} = (\Delta_x, \Delta_y)$, expressed in pixels, compensating for the translational offset between the particle and the center of the box, and then shifted back to align with the origin of the image pixel grid.

$$\mathbf{p}_{\text{proj}} = (R \cdot \mathbf{p}_{\text{centered}})_{xy} - \mathbf{\Delta} + \frac{N}{2}$$

Since the rotated and shifted coordinates do not align exactly with the discrete pixel grid, a scatter-based forward mapping is employed: the intensity of each voxel is distributed across the four nearest pixels using bilinear interpolation weights, and accumulated onto the output image. A Gaussian filter is applied to compensate for the discretization artifacts introduced by this splatting-based projection. Finally, the corresponding CTF of each image is applied in Fourier space, as mentioned in 2.3.2, yielding a batch of simulated projections consistent with the image formation model of the microscope.

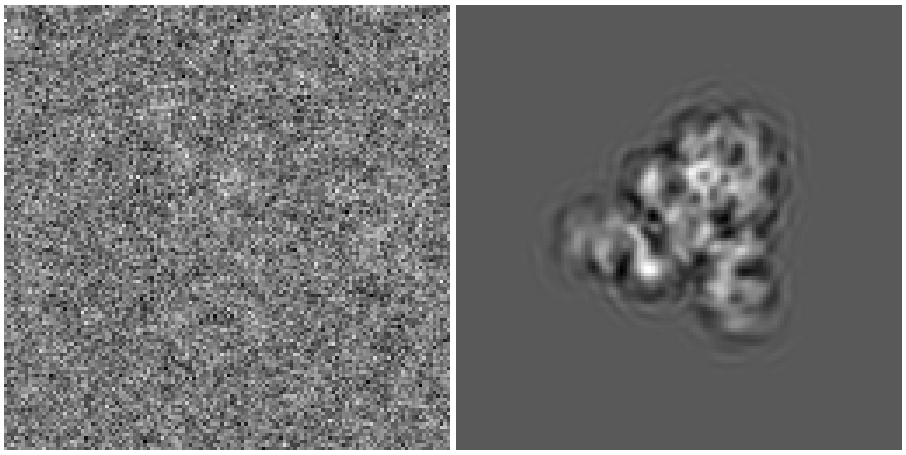


Figure 4.2: *Comparison between an experimental image from the SARS-CoV-2 Spike protein dataset (left) and a projection of the reference 3D volume, generated with the* `volumeProjection` *function (right).*

4.3.4 Controlled Misalignment as Data Augmentation

In order for the model to learn to recognize particles affected by alignment errors, their **misaligned versions** are generated. Since misalignment can manifest either as an incorrect orientation, an incorrect in-plane position, or a combination of both, Euler angles and in-plane shifts are perturbed to generate a misaligned projection of the volume using the `generate_misalignment` function.
The perturbation magnitude of the Euler angles is sampled from a uniform distribution over the interval $[\alpha_{\min}, \alpha_{\max}]$, with a random sign applied to each angle to allow rotations

in either direction. The minimum value, reported in Table 4.1, is computed once per dataset through the `compute_min_rotation_angle` function and corresponds to the smallest rotation such that the mask voxel farthest from its center of mass travels at least `pixel_threshold = 2.0` pixels of arc. The maximum angle is set to 180°, covering the full range of possible misorientations.
For the translational component, the shift magnitude is instead sampled from a uniform distribution over the interval `[shift_min_px, shift_max_frac]` assuming a minimum shift value of 2 pixels and a maximum shift value that is 10% of the box size, as reported in Table 4.1, while the direction is drawn uniformly over the full circle, so that every direction of displacement is equally likely.

Dataset	$\alpha_{\min}$ (°)	$\alpha_{\max}$ (°)	**Shift range (px)**
Simulated	3.06	180	2 – 12.8
Ribosome	2.12	180	2 – 12.8
SARS-CoV-2 Spike	2.24	180	2 – 12.8

Table 4.1: *Misalignment perturbation parameters for each dataset.*

The resulting *noisy Euler angles* and *noisy Shifts* are then used together with the original volume to generate misaligned projections through the same `volumeProjection` function described in Section 4.3.3, which are compared against the corresponding aligned ones as part of the model's training procedure.
Figure 4.3 shows a representative example (particle ID 173749) of an image belonging to the spike dataset compared against a projection generated using perturbed pose parameters, with the corresponding numerical values reported in Table 5.10.

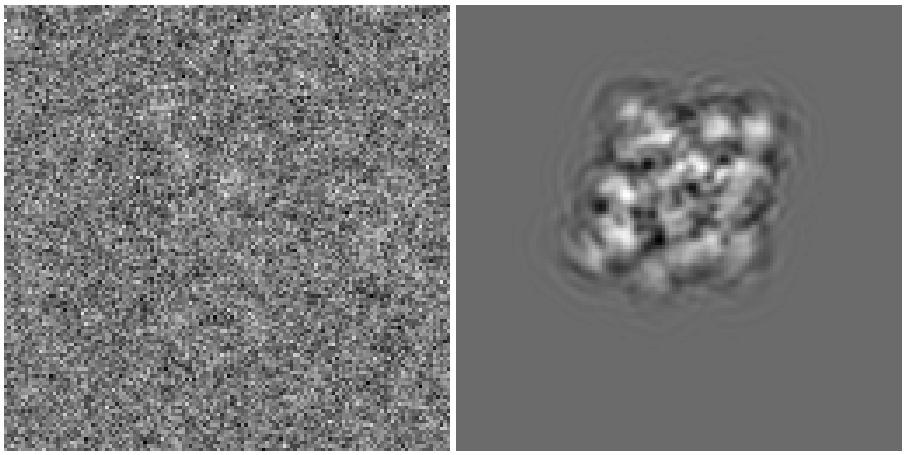


Figure 4.3: *Comparison between an experimental image from the SARS-CoV-2 Spike protein dataset (left) and a projection of the reference 3D volume, generated with the* `volumeProjection` *function using perturbed pose parameters (right).*

4.3.5 Fourier Ring Correlation

To quantify the discrepancy between the projection and its corresponding experimental image, they are compared in Fourier space through the `compute_fourier_residual` function. Both images are transformed via a 2D Fast Fourier Transform, and the resulting spectrum is partitioned into a fixed number of concentric rings (referred to as `n_shells` in the implementation, following a naming convention shared with the 3D case), each corresponding to a discrete range of spatial frequencies. For both the aligned and the misaligned pair, the **Fourier Ring Correlation** (FRC) is computed as the normalized cross-correlation between the two spectra:

$$\text{FRC} = \frac{\sum \left(F_{\text{proj}} \cdot F_{\text{exp}}^{*} \right)}{\sqrt{\sum |F_{\text{proj}}|^2 \cdot \sum |F_{\text{exp}}|^2}}$$

where F_{proj} is the Fourier transforms of the projection, F_{exp} is the Fourier transforms of the experimental image, '*' denotes the complex conjugate of F_{exp} and all the summations are computed all over the Fourier space frequencies within a ring [48].

The resulting FRC curve quantifies the degree of agreement between the two images, ranging from -1 (perfect anti-correlation) to 1 (perfect correlation); a well-aligned pair is expected to remain close to 1 over a wide range of frequencies and a misaligned pair typically shows a faster decay toward zero. As a general rule, in Fourier space, low spatial frequencies encode the overall smooth areas of the object, while high spatial frequencies represent its finer details. Looking at the FRC curves, the one corresponding to the correlation between the experimental image and the aligned projection, shows higher correlation at low spatial frequencies, because the two images are expected to remain coherent; by contrast, the misaligned pair loses this match at low frequencies and at higher frequencies, both curves become affected by noise, and the distinction is less pronounced.

Finally, the curves are standardized through the `standardize_frc_curve` function. Since, datasets with different box sizes have frequency axes with different length and scale, each curve is resampled onto a reference frequency grid, according to a reference pixel size of $T_s = 1.0$ Å/px. Resampling is performed via zero-order-hold (step) interpolation, which assigns each new frequency bin the value of the nearest preceding native sample. The FRC curves are given as input to the model, described in Section 4.4, to recognize the overall decay pattern.

The misaligned FRC curve in Figure 4.4 exhibits a faster decay towards 0 at lower spatial frequencies compared to the aligned FRC curve, reflecting the loss of coherence introduced by the perturbed pose parameters.

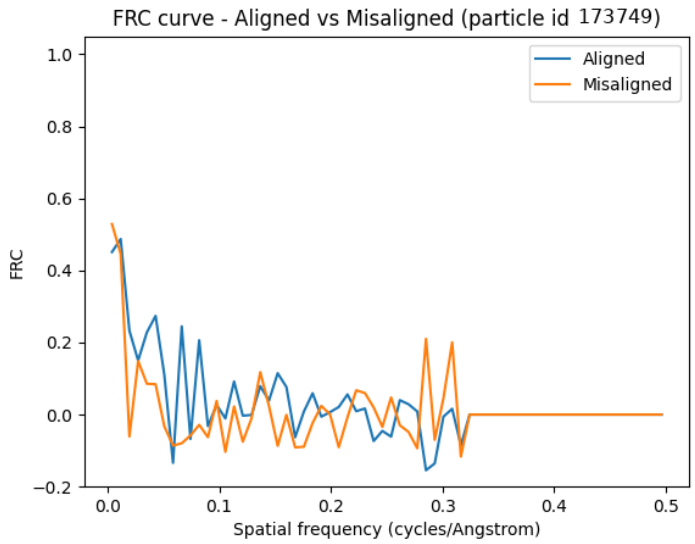


Figure 4.4: *FRC curves computed between the experimental image and the corresponding aligned and misaligned projections shown, respectively, in Figure 4.2 and 4.3*

4.4 Neural Network Architecture

The so-called model, *cryoCheck*, was implemented using a **multilayer perceptron** (MLP) architecture that takes the standardized FRC curves as input and produces a single scalar output representing the predicted probability that the image pair is misaligned. The network consists of one input layer, whose size matches the number of frequency shells used to compute the FRC curves, two hidden fully connected layers, with 64 and 32 neurons, respectively, and a final output layer with a single neuron. Taking into account that the number of shells is $n_{\text{shells}} = x_{\text{size}}/2$ and all the dataset that were used have images of `128 x 128` pixels, the number of neurons of the input layer is equal to 64, as shown in Figure 4.5

Each hidden layer is followed by **batch normalization** to keep the input to each layer stable. Then, the **ReLU** activation function is applied. It is defined as $f(x) = max(0, x)$, where the output is x only if it is a positive value, otherwise it is 0. It introduces non-linearity in the neural network to make it recognize complex patterns, it is also easy and fast to compute and it helps mitigate the vanishing gradient problem. Given $n_{\text{shells}} = 64$, the resulting architecture comprises a total of 6,465 trainable parameters.

The model was built using **Flax NNX** [49], JAX's neural network library. It automatically manages the model's internal state (e.g. layer weights and batch normalization statistics), allowing the network to be used with an object-oriented syntax, while internally handling

the representation required by JAX's compilation and automatic differentiation mechanisms. In Flax NNX, everything is explicit: the model directly stores its own parameters as regular object attributes, random keys must be passed explicitly by the user, and the exact size of every layer must be specified when the model is created.

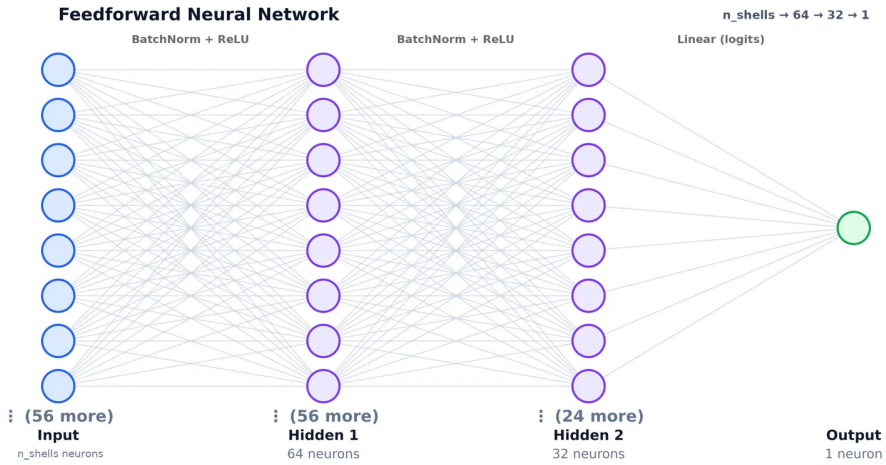


Figure 4.5: *cryoCheck MLP architecture.*

4.5 Training and Validation

The model has been trained and validated using the `cryoCheck_step` function. The disagreement between the predicted output and the labels is computed with a **binary cross-entropy** (BCE) loss. Once the loss is computed, the backpropagation algorithm uses the chain rule to calculate the gradients of the loss function with respect to the weights and biases of each neuron; the **AdamW** optimizer then updates these parameters to minimize the loss function. Both the BCE loss and the AdamW optimizer are provided by the Optax library [50].
BCE is a loss function commonly used in binary classification problems: the lower its value, the better the classification capabilities of the model; due to its logarithmic nature, incorrect predictions are penalized more heavily than correct ones. The AdamW optimizer decouples weight decay from the gradient-based update: weights are adjusted based on the gradient direction, while an additional decay term prevents them from growing excessively large, helping to reduce overfitting.
Each dataset was split into training and validation subsets according to the `-dataset_split_fraction` argument, which defaults to $[0.8, 0.2]$, meaning that 80% of the data is used to train the model and the remaining 20% to validate it.
The `-mode` argument determined whether the model would be trained or used for inference,

with two available options: train and predict. At the implementation level, this distinction corresponded to switching the model between two internal states. The train/eval mode of the model is set, respectively, via `cryoCheck.train()` and `cryoCheck.eval()`, which determine the behavior of the Batch Normalization layers: in training mode, the normalization statistics (mean and variance) are computed on the current batch, while in eval mode the statistics accumulated during training are used, making the model's output deterministic and reproducible across evaluations. During training, the output layer produces raw logits, which are used directly to compute the BCE loss. This procedure is repeated within the training loop for every batch of images, whose size is controlled by the `-batch_size` argument. The model is validated at the end of each epoch using the `cryoCheck_step` function but with the flag `train` set to `False`: the loss is still computed but without any gradient update. During validation, the model additionally operates in evaluation mode (`eval=True`), applying a **sigmoid activation** to the output logit and thus producing explicit probabilities in the $[0, 1]$ range — where 0 corresponds to the aligned label and 1 to the misaligned label — rather than raw logits. This is necessary to compute the evaluation metrics defined in Section 4.7.

4.6 Scipion Implementation

Scipion [51] is an open source cryo-EM processing framework that integrates several software packages, developed both by the facility and by third-party developers, within a single platform. Scipion is mainly written in Python and comprises three core pip packages — `scipion-pyworkflow`, `scipion-em`, and `scipion-app` — along with a launcher, `scipion3`, used to run Scipion within a conda environment or a virtualenv. Documentation and installation instructions can be found at `https://scipion-em.github.io/docs/release-3.0.0/index.html` and `https://github.com/I2PC/scipion/wiki/How-to-Install`, respectively.

The main advantage of Scipion is its manageability: users have all the necessary tools centralized within a single platform, enabling them to work in streaming mode, with every step annotated and always accessible, while data conversions and format compatibility between different software packages are handled automatically. This also makes it possible to directly compare results obtained from different algorithms that address the same purpose. Inside the Scipion environment, two concepts can be distinguished: *plugins* and *protocols*. A plugin is a software package that integrates an external tool into Scipion, handling the installation, compatibility and functionalities. A protocol is a single step, corresponding to a single operation, within a Scipion workflow, with its own inputs and outputs. A plugin can contain multiple protocols. *CryoCheck* was implemented within *Hax* [52], a Python environment for heterogeneity analysis in cryo-EM, then it was made available as a Scipion protocol within the `Hax` plugin, as `hax-train-misalignment detection`, allowing it to be used as part of a cryo-EM workflow. In the Scipion GUI,

the protocol window shows an `Input` section that takes as inputs a set of experimental cryo-EM images and a reference 3D volume, and a `Network` section that allows the user to set the hyperparameters of the model, such as the number of epochs, batch size and learning rate and if loading images to RAM is preferred over the memory mapping; the user can also add the GPU device to use.

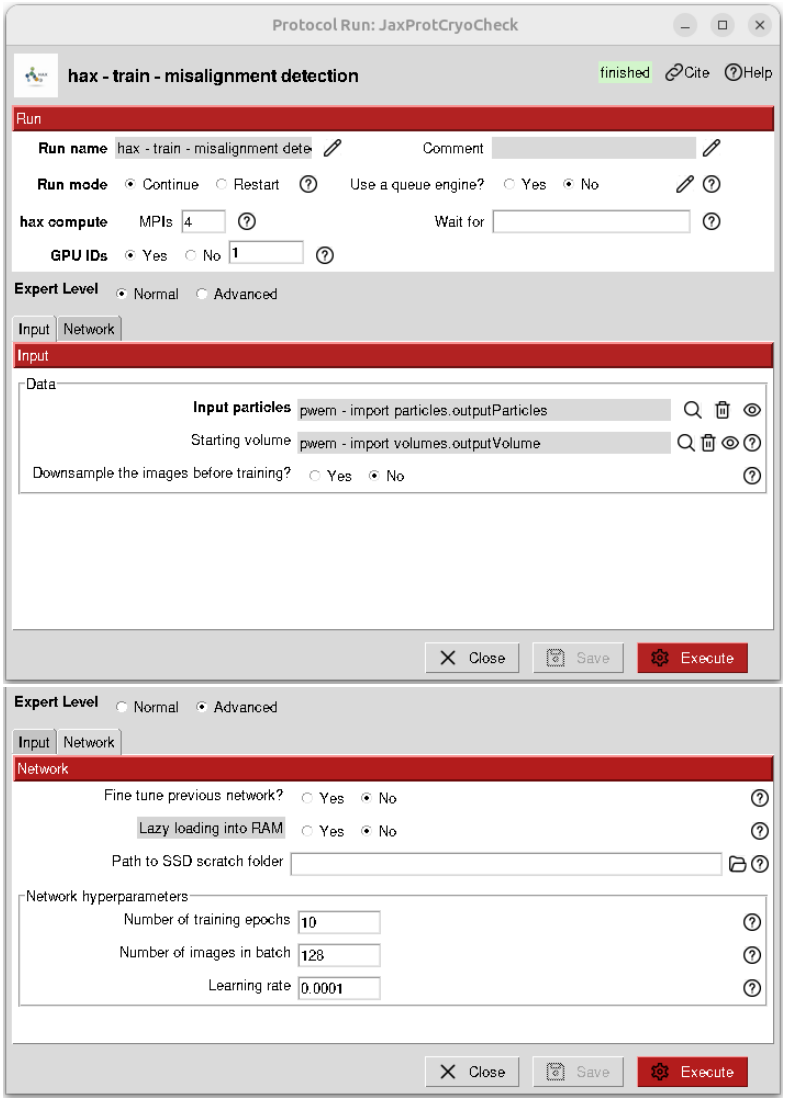


Figure 4.6: *The CryoCheck protocol form within Scipion's GUI, showing the Input and the Network sections required to run the misalignment detection tool.*

The easiest configuration that could be run in Scipion and, also, the one that was used to run the model inside the Scipion GUI is shown in Figure 4.7 and consists of three protocols:

- pwem–import particles protocol, to import a set of particles to the project
- pwem–import volumes protocol, to import a volume to the project
- hax–train–misalignment detection, to detect misalignment in cryo-EM maps, using as inputs the output of the two previous protocols.

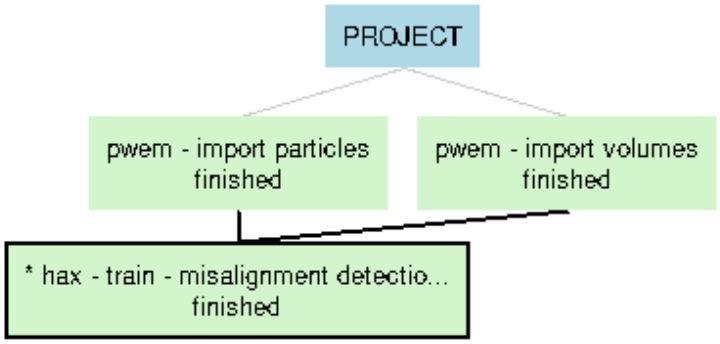


Figure 4.7: *Scipion configuration used to run the cryoCheck model.*

4.7 Evaluation Metrics

To assess the performance of *CryoCheck*, the following classification metrics are used throughout training and validation.

Since misalignment detection is formulated as a binary classification problem, the model's predictions can be characterized through a **confusion matrix**, consisting of four values: True Positives (TP), False Positives (FP), True Negatives (TN) and False Negatives (FN). In this context, the *positive* class corresponds to correctly aligned particles (label 0), while the *negative* class corresponds to misaligned particles (label 1). A TP is a correctly aligned particle identified as such by the model, a TN is a misaligned particle correctly identified as misaligned, an FP is a misaligned particle incorrectly classified as aligned, and an FN is an aligned particle incorrectly classified as misaligned. Figure 4.8 shows an example.

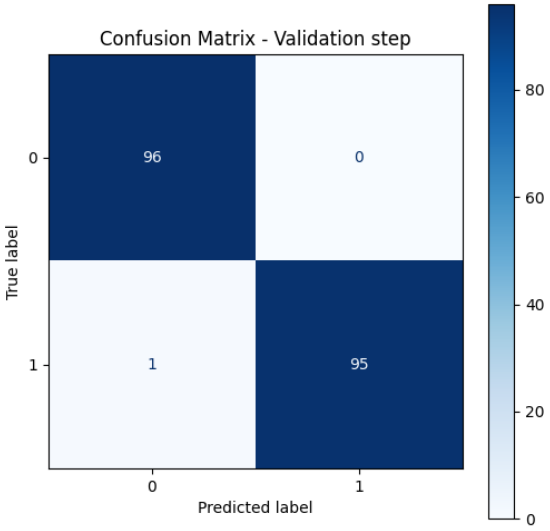


Figure 4.8: *Example of a confusion matrix belonging to the validation step of the simulated dataset: 96 particles were correctly classified as aligned, 95 particles were correctly classified as misaligned, and 1 particle was misclassified as aligned.*

After each epoch, the **Receiver Operating Characteristics** (ROC) curve and its associated **Area Under the Curve** (AUC) are computed from the predicted probabilities and the corresponding labels. The ROC curve plots the true positive rate ($\text{TPR} = \text{TP}/(\text{TP}+\text{FN})$) against the false positive rate ($\text{FPR} = \text{FP}/(\text{FP}+\text{TN})$) as the classification threshold is varied across its entire range, providing a comprehensive view of the trade-off between correctly identifying aligned particles and incorrectly misclassifying misaligned ones. The AUC summarizes this trade-off into a single scalar value, ranging from 0.5, corresponding to random guessing, to 1, corresponding to perfect separation between the two classes.

From this curve, an optimal classification threshold is then derived using **Youden's index**, allowing the continuous probability values produced by the model to be converted into binary predictions. Epoch after epoch during training, the Youden's index is computed as follows:

$$J = TPR - FPR = Sensitivity + Specificity - 1$$

By choosing its max value, an optimal cut-off can be set to distinguish between the aligned and misaligned class because it balances both specificity and sensitivity.

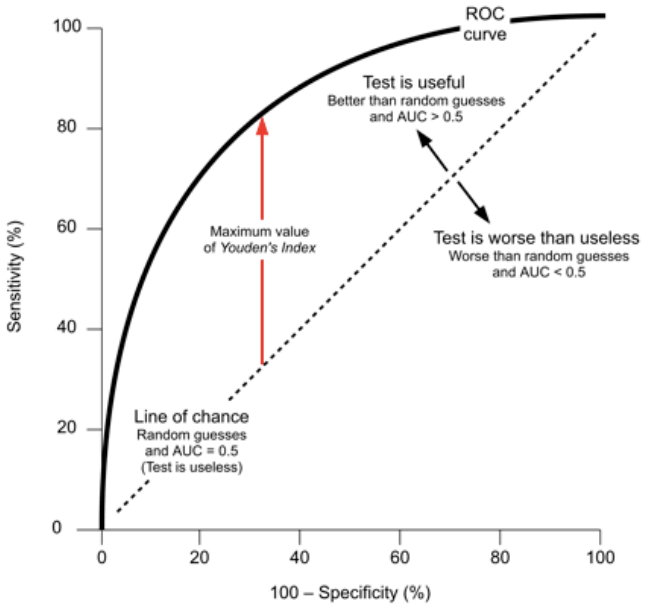


Figure 4.9: *Youden's index selection [53].*

The binary predictions are used to compute accuracy, precision, recall, and F1-score. **Accuracy** is the percentage of correct predictions out of the total number of cases — how often the model correctly classifies a particle, whether it is aligned or misaligned:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

Precision measures the percentage of correct positive predictions out of all the cases the model claimed were positive — how reliable the model's "aligned" predictions are:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

Recall quantifies the percentage of actual positive cases correctly identified by the model — how many truly aligned particles are correctly identified as such by the model:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

F1-score is the harmonic mean of Precision and Recall, introduced to show a single balanced measure:

$$\text{F1} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

The **loss curves** are then plotted at the end of the training step to evaluate how the model behaves during training and validation, in order to detect malfunctioning, avoid overfitting, and adjust the model's hyperparameters.

5. Results

The coding work was implemented using Python in Visual Studio Code (VS Code) and all the experiments were run on NVIDIA Tesla T4 GPU inside the Scipion environment. For each of the three datasets — simulated, ribosome, and SARS-CoV-2 Spike protein — results are reported in the following sections: the hyperparameters and dataset-specific parameters used for training are first summarized, together with a representative visual example comparing an aligned and a misaligned projection against the corresponding experimental image, along with the associated FRC curves. This is followed by the confusion matrix and the related classification metrics described in Section 4.7, the ROC curve, and the training and validation loss curves.

5.1 Simulated Data

The simulated dataset, described in Section 4.2, consists of 500 noise-free images belonging to the adenylate kinase structure. Since the simulated dataset is relatively small, all images were loaded into RAM prior to training and its hyperparameters are summarized in Table 5.1. In Figure 5.1, a representation of a simulated cryo-EM image (ID 497) with its projections.

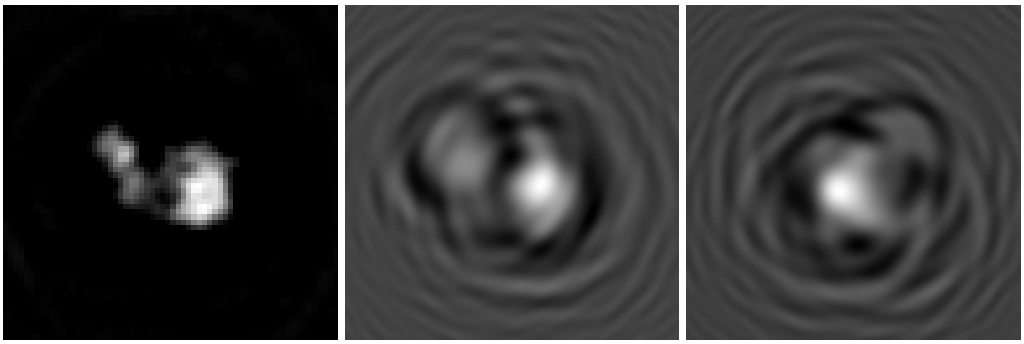
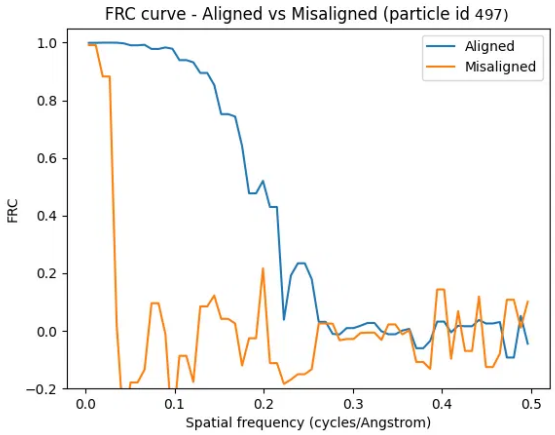


Figure 5.1: *Cryo-EM image (ID 497) belonging to the simulated dataset (CTF free) (left), its aligned projection (center) and its misaligned projection (right).*

Epochs	Batch size	Sampling rate	Learning rate	Train/Val split
50	32	1	1e-4	0.8 / 0.2

Table 5.1: *Training hyperparameters used for the simulated dataset.*

By correlating each simulated image with its aligned and misaligned projections, two FRC curves are computed; Figure 5.2 shows the FRC curves belonging to particle 497, together with the corresponding pose parameters and perturbations. Looking at the curves, the correlation between the image and its aligned projection remains higher at low frequencies, consistent with the fact that they match in terms of their overall shape; while the misaligned curve shows lower correlation values already at low frequencies, reflecting the poor consistency between the image and its misaligned version.



Parameter	**Values**
Aligned	
Euler angles (°)	19.5, 126.7, 170.2
Shifts (px)	-1.6, -2.6
Misaligned	
Euler angles (°)	44.4, 229.4, 190.5
Shifts (px)	-1.4, -5.7
Perturbation	
Δ Euler angles (°)	24.9, 102.7, 20.3
Δ Shifts (px)	0.2, -3.1

Figure 5.2: *Comparison between the aligned and misaligned pose parameters for particle 497, together with the corresponding FRC curves.*

CryoCheck is fed with those FRC, computed on the fly for each image pair in the dataset, that is split into training and validation subsets with a 80/20 proportion. The ROC curve and confusion matrix associated with the trained model are shown in Figure 5.3 , whose results correspond to the final training epoch. Focusing on the ROC curve, the model is capable of learning the underlying patterns of the FRC curves, achieving near-perfect discrimination between the aligned and misaligned class. The same conclusion can be obtained by analyzing the confusion matrix: 96 particles were correctly classified as aligned, 95 were correctly classified as misaligned and only 1 particle was misclassified as aligned. Its associated metrics are reported in Table 5.2. The model achieves a Recall of 100%, correctly identifying every aligned particle in the validation set, while the misclassified particles slightly affect Precision.

Accuracy	**Precision**	**Recall**	**F1-score**
99.48%	98.97%	100%	99.48%

Table 5.2: *Classification metrics for the simulated dataset.*

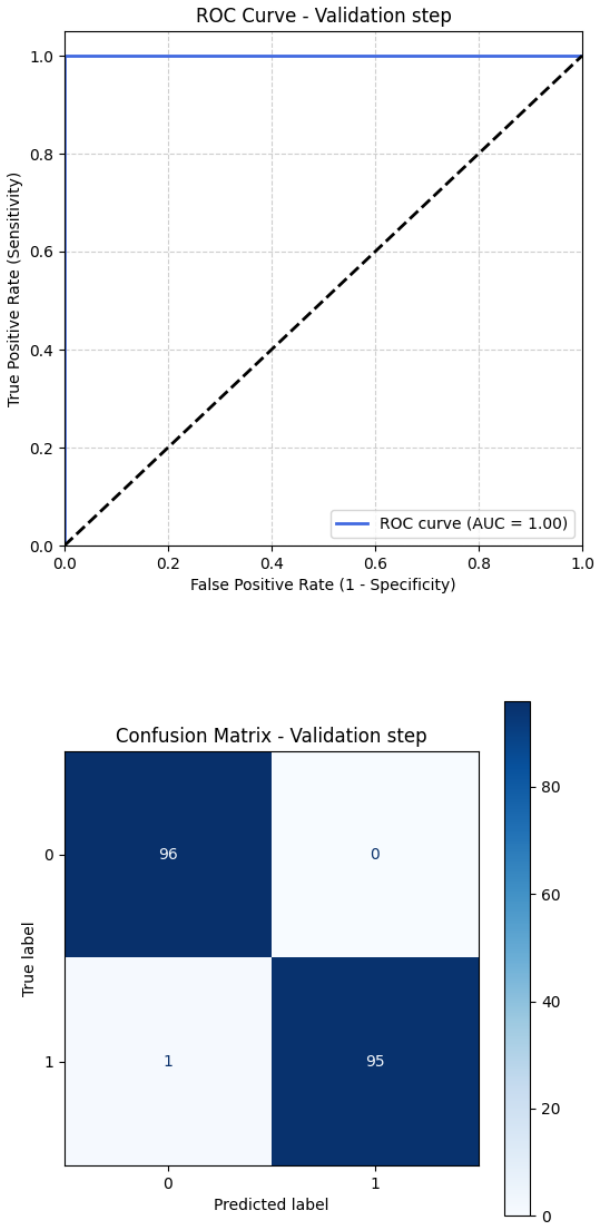


Figure 5.3: *ROC curve (up) and confusion matrix (bottom) to evaluate cryoCheck performance.*

The learning curves for both training and validation steps are shown in Figure 5.4. They converge after approximately 200 steps, with the training loss approaching zero under supervised conditions. This indicates that, beyond learning to distinguish aligned from

misaligned FRC curves within the training set, the model becomes capable of correctly discriminating curves from previously unseen images. The training time for this datasets is approximately of 2.51 minutes.

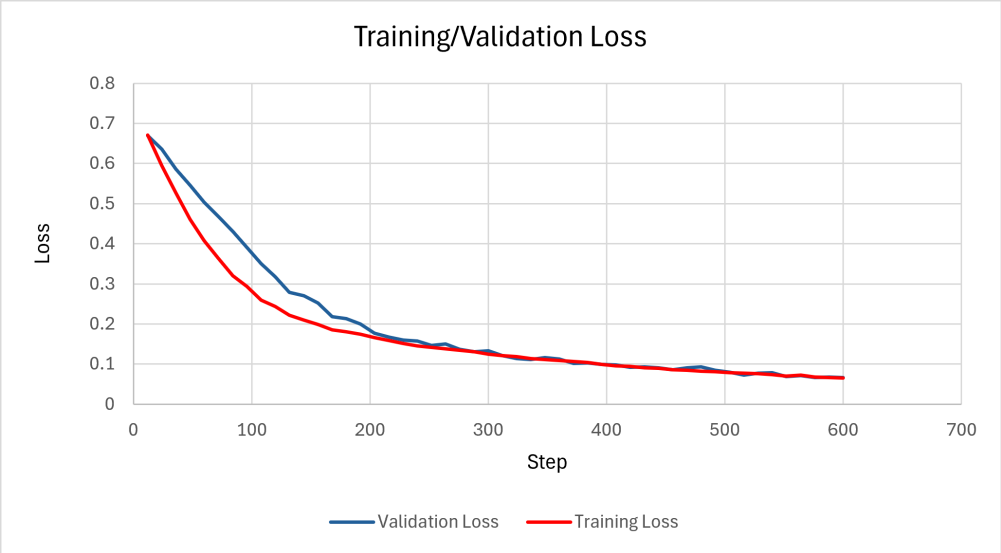


Figure 5.4: *Training and validation loss curves.*

5.2 Ribosome Data

The ribosome dataset, described in Section 4.2, consists of about 50,000 images of the structure of the *P. falciparum* 80S ribosome bound to emetine. Due to the larger size of the dataset, images were memory–mapped rather than loaded entirely into RAM; this results in slightly lower execution performance, as reading from disk introduces additional latency compared to accessing data directly from memory. The hyperparameters used to train the model are reported in Table 5.3. Since GPU memory usage remained low with the default batch size of 32, it was possible to increase it, while the number of epochs was reduced compared to the simulated dataset, thanks to the larger size of the dataset.

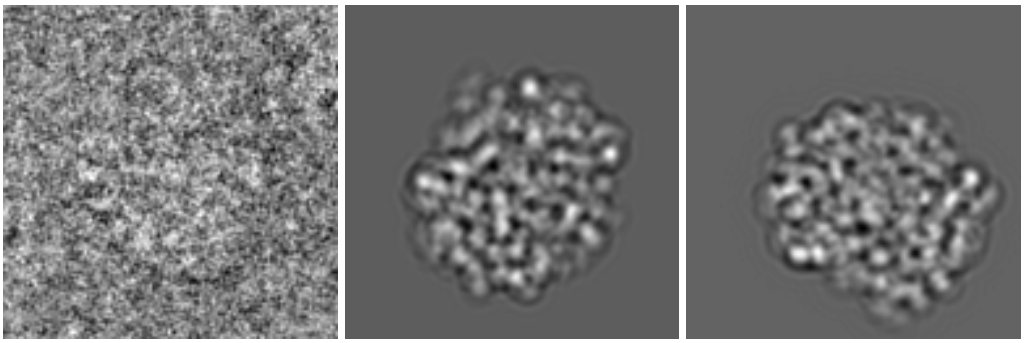
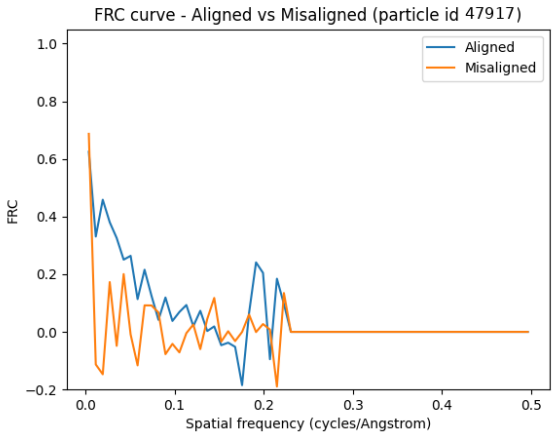


Figure 5.5: *Cryo-EM image (ID 47917) belonging to the ribosome dataset (left), its aligned projection (center) and its misaligned projection (right).*

Epochs	Batch size	Sampling rate	Learning rate	Train/Val split
10	128	3.14	1e-4	0.8 / 0.2

Table 5.3: *Training hyperparameters used for the ribosome dataset.*

The corresponding FRC curves and pose parameters/perturbations of particle sample 47917 are taken as an example and they are shown in Figure 5.6. As already mentioned in Section 5.1, the aligned curve shows higher correlation values at lower frequencies compared to the misaligned one; however, due to higher noise levels in this dataset, the distinction between the two curves' trends is less pronounced.



Parameter	Values
Aligned	
Euler angles (°)	103.7, 64.6, -134.3
Shifts (px)	-4.0, -7.4
Misaligned	
Euler angles (°)	97.1, -104.6, -16.5
Shifts (px)	-9.2, -7.4
Perturbation	
Δ Euler angles (°)	-6.6, -169.2, 117.8
Δ Shifts (px)	-5.2, 0.0

Figure 5.6: *Comparison between the aligned and misaligned pose parameters for particle 47917, together with the corresponding FRC curves.*

The ROC shape and its corresponding AUC value in Figure 5.7, show the model's ability to distinguish between aligned and misaligned curves achieving good results on unseen images. Regarding the confusion matrix, 9853 particles were correctly classified as aligned, 9850 were correctly classified as misaligned, 3 particle were misclassified as misaligned and 6 particles were misclassified as aligned. The overall performance of the model is confirmed by the metrics in 5.4: both Precision and Recall exceed 99.9%, indicating that the model reliably distinguishes aligned from misaligned and the resulting F1-score of 99.95% reflects a well-balanced classification performance across both classes.

Accuracy	Precision	Recall	F1-score
99.95%	99.94%	99.97%	99.95%

Table 5.4: *Classification metrics for the ribosome dataset.*

The training and validation loss curves are shown in Figure 5.8. The initial divergence and higher validation loss values can be attributed to the random selection of images and perturbations at the start of training; as training progresses, the two curves converge. The training time for this datasets is approximately of 15.35 minutes.

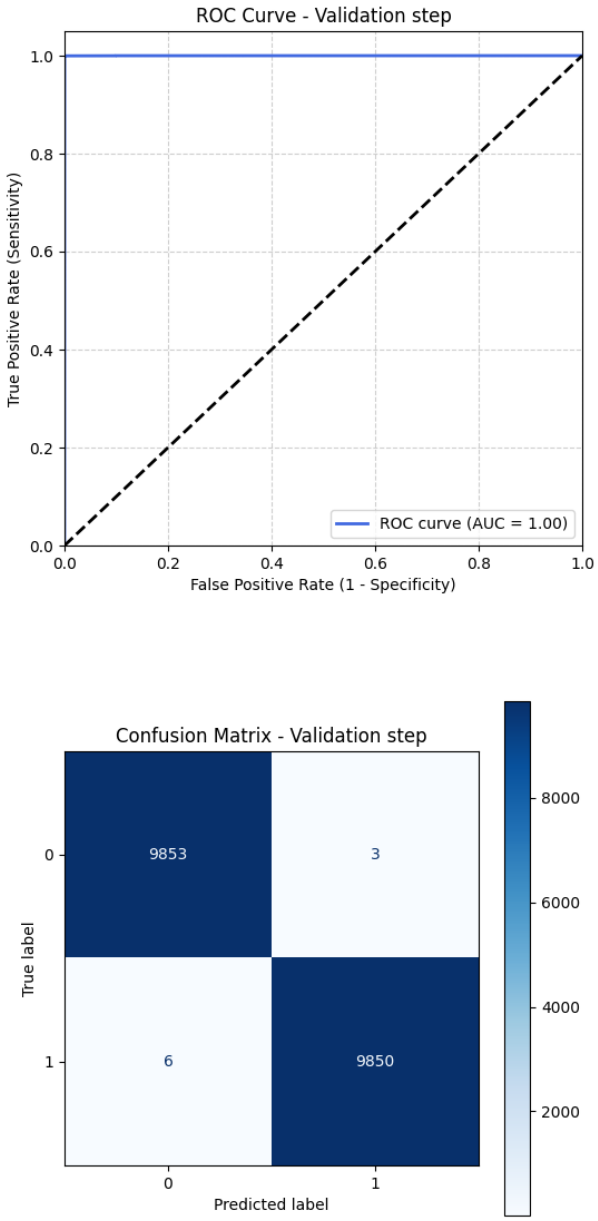


Figure 5.7: *ROC curve (up) and confusion matrix (bottom) to evaluate cryoCheck performance.*

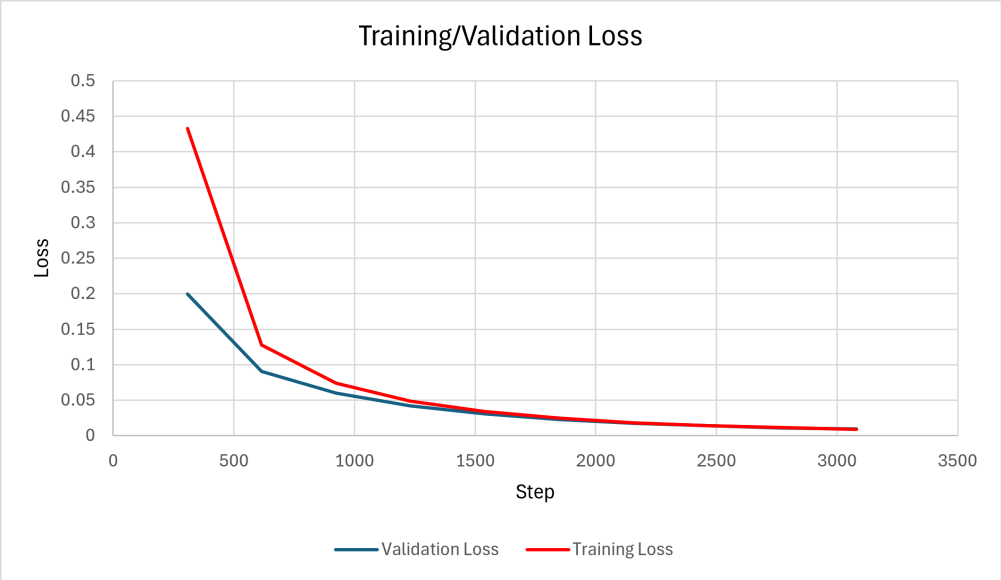


Figure 5.8: *Training and validation loss curves.*

5.3 Spike Data

The SARS-CoV Spike protein dataset, described in Section 4.2, consists of about 200,000 images, it is the largest in size and the one that is more heavily affected by noise. The images were memory–mapped and the hyperparameters used to train the model are summarized in Table 5.5: here, the same considerations regarding the hyperparameters choice of the ribosome dataset can be repeated.

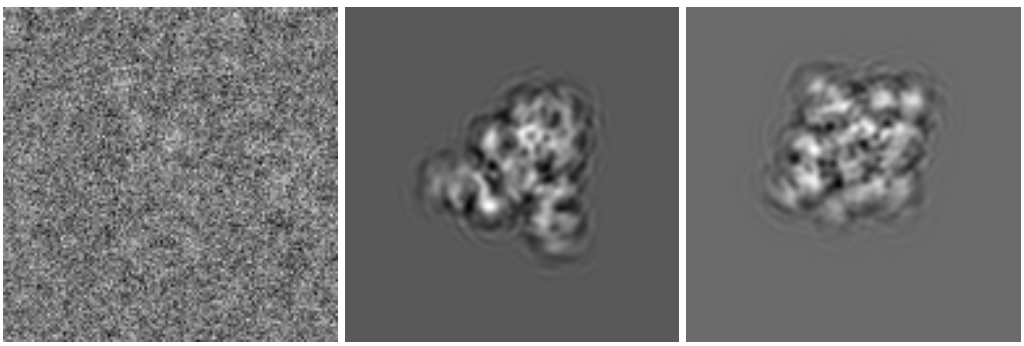


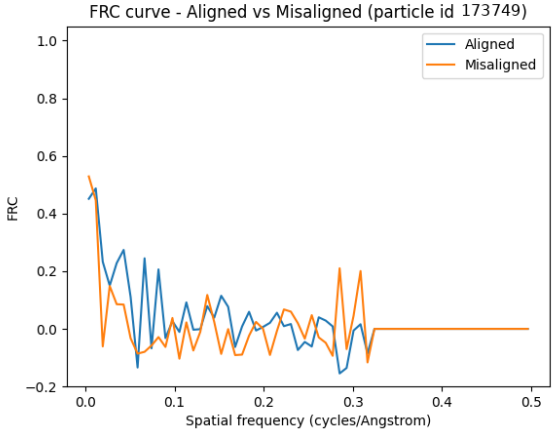
Figure 5.9: *Cryo-EM image (ID 173749) belonging to the spike dataset (left), its aligned projection (center) and its misaligned projection (right).*

Figure 5.10 shows the FRC curves and pose parameters/perturbations belonging to particle 173749: as the images that are fed to the model become more affected by noise, the curves

Epochs	Batch size	Sampling rate	Learning rate	Train/Val split
10	128	2.19	1e-4	0.8 / 0.2

Table 5.5: *Training hyperparameters used for the spike dataset.*

become less discernible with respect to the other, still showing high correlation values at low frequencies for the aligned one, according to the shape match of the cryo-EM image and its aligned projection.



Parameter	Values
Aligned	
Euler angles (°)	-145.2, 35.2, 70.5
Shifts (px)	-5.0, 5.4
Misaligned	
Euler angles (°)	-134.0, 134.8, 46.5
Shifts (px)	-1.3, 12.9
Perturbation	
Δ Euler angles (°)	11.2, 99.6, -24
Δ Shifts (px)	3.7, 7.5

Figure 5.10: *Comparison between the aligned and misaligned pose parameters for particle 173749, together with the corresponding FRC curves.*

The ROC curve and the confusion matrix in Figure 5.11, shows the overall behaviour of the model after the training session. 38705 particles were correctly classified as aligned, 38873 were correctly classified as misaligned, 975 particles were misclassified as misaligned and 807 particles were misclassified as aligned. The associated metrics, reported in Table 5.6, confirm the model's performance even on this dataset: Precision and Recall are closely balanced (97.96% and 97.54%, respectively), indicating that the model does not exhibit a pronounced bias toward either false positives or false negatives.

Accuracy	Precision	Recall	F1-score
97.75%	97.96%	97.54%	97.75%

Table 5.6: *Classification metrics for the spike dataset.*

The difference between the training and validation loss curves indicates slightly higher loss values during validation, but they follow the same trend and converge after a small

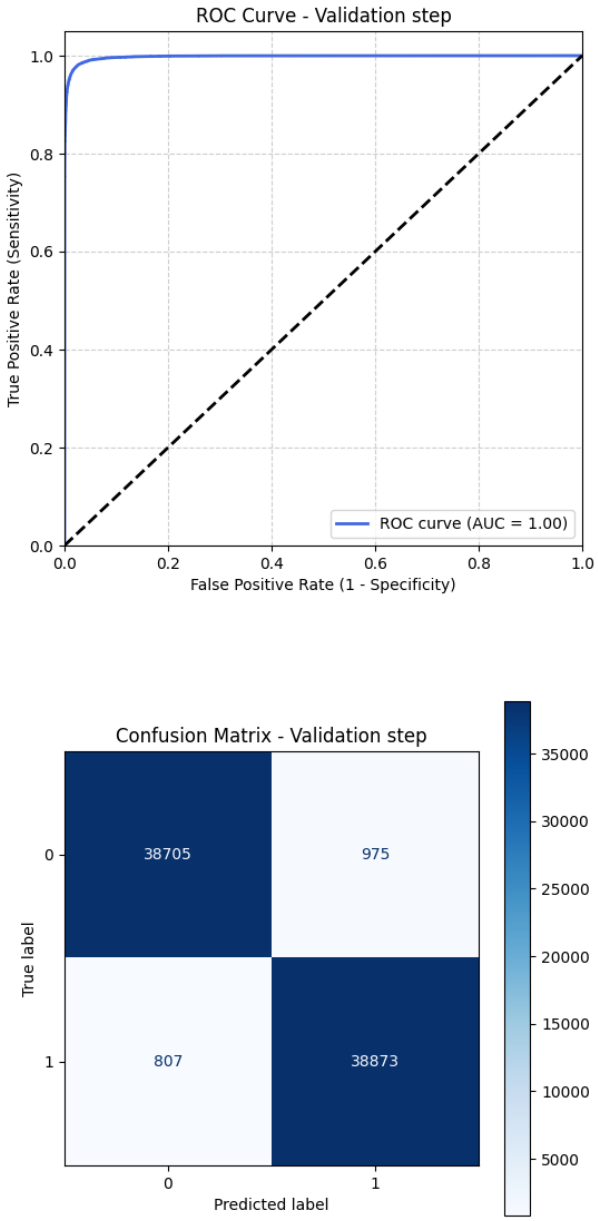


Figure 5.11: *ROC curve (up) and confusion matrix (bottom) to evaluate cryoCheck performance.*

number of steps, indicating that the model learns quite fast how to distinguish one type of curve from the other. The training time for this datasets is approximately of 35.8 minutes.

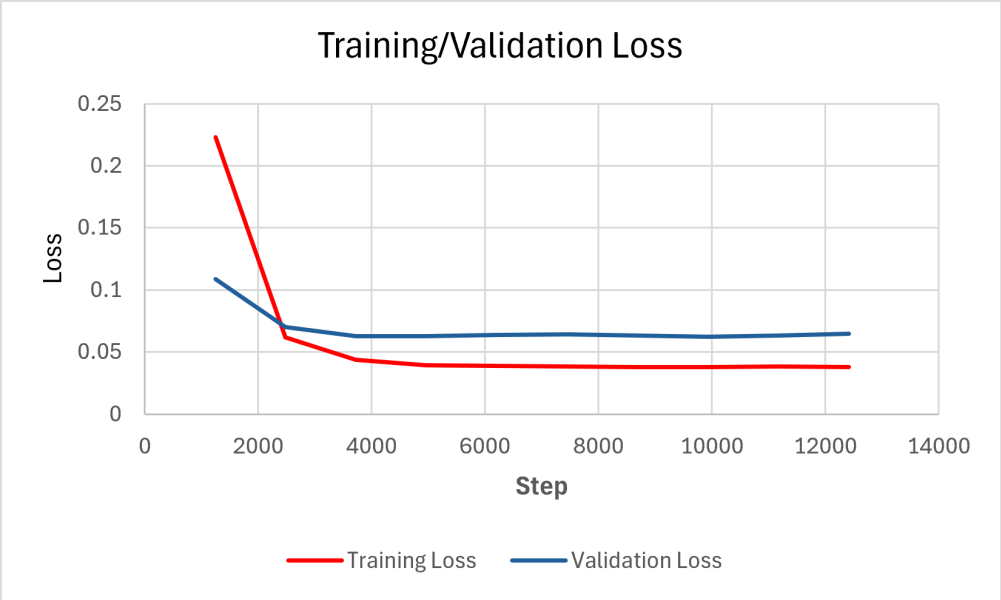


Figure 5.12: *Training and validation loss curves.*

5.3.1 Testing Pose Parameters Perturbations

To further demonstrate that the model is enough responsive to small perturbation values, the ribosome dataset was tested on different pose parameters perturbation ranges. The Euler angles intervals, shown in Table 5.7, were chosen with different granularity: they were much more narrow at lower perturbations values, to understand if the model was also reliable when it comes to recognize really small deviations from the real image, and larger at higher perturbations values, where the rotations become clearly visible. While the shifts intervals, summarized in Table 5.8, were approximately regular.

Range 1	**Range 2**	**Range 3**	**Range 4**	**Range 5**	**Range 6**
2.12–10°	10–30°	30–60°	60–90°	90–120°	120–180°

Table 5.7: *Euler angle perturbation ranges used to test the model.*

Range 1	**Range 2**	**Range 3**
2–6 px	6–9 px	9–12.8 px

Table 5.8: *Shift perturbation ranges used to test the model.*

When testing the model performance to distinguish various Euler angles perturbation

ranges, the shift perturbations were set to 0 and vice versa. The following ROC curves were extracted to visually inspect the model performance.

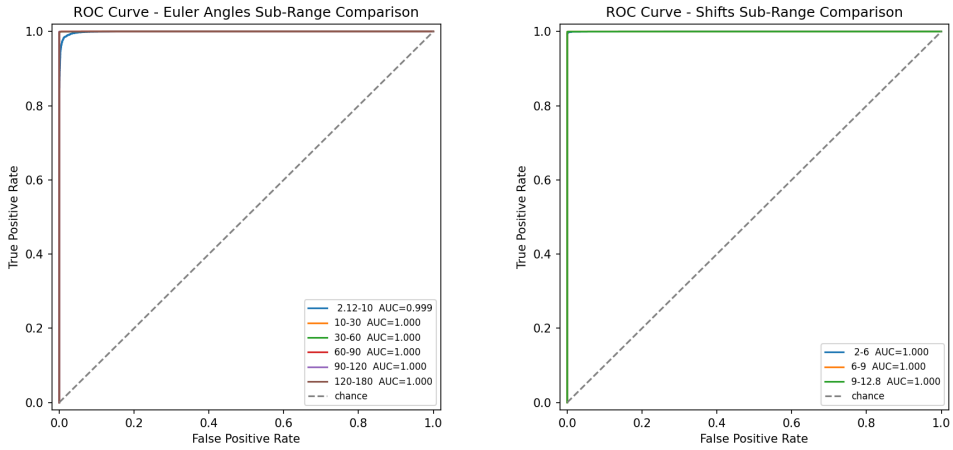


Figure 5.13: *Overlapping ROC curves belonging the to model trained at different Euler angles perturbations ranges (left) and shifts ranges (right).*

The FRC curves of a small set of particles were plotted to visualize their trends.

In Figures 5.14 and 5.15, referring to the training session related to the Euler angle perturbation range of [2.12–10]° and [10–30]°, the curves were clearly different, supporting the ability of the model to classify the aligned and misaligned version of a particle even at small angular fluctuations. The same experiment was conducted for the smallest shift range [2–6] px, shown in Figure 5.16 and the same conclusion applies.

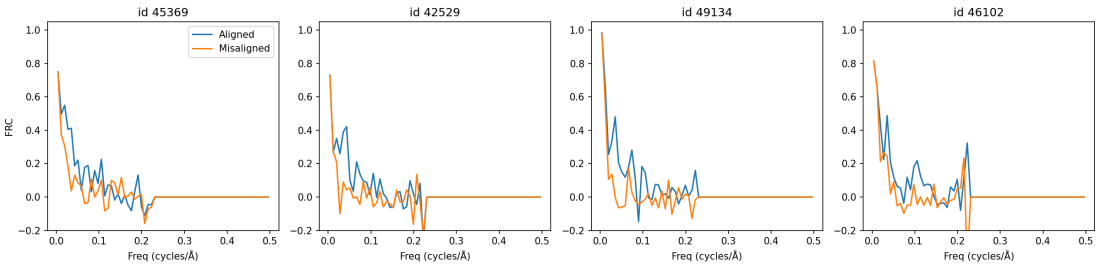


Figure 5.14: *FRC curves at angular perturbation Range 1*

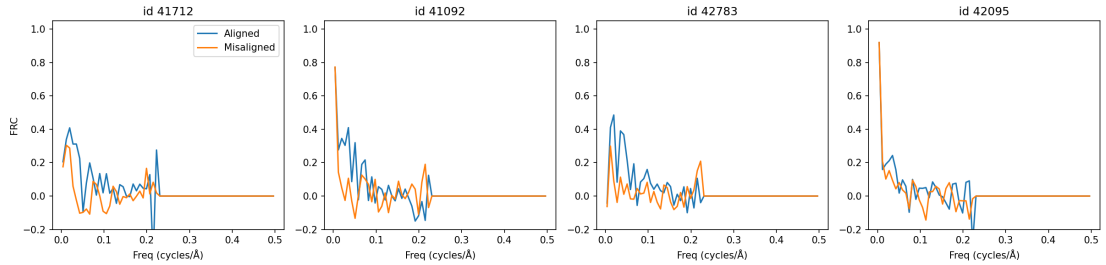


Figure 5.15: *FRC curves at angular perturbation Range 2*

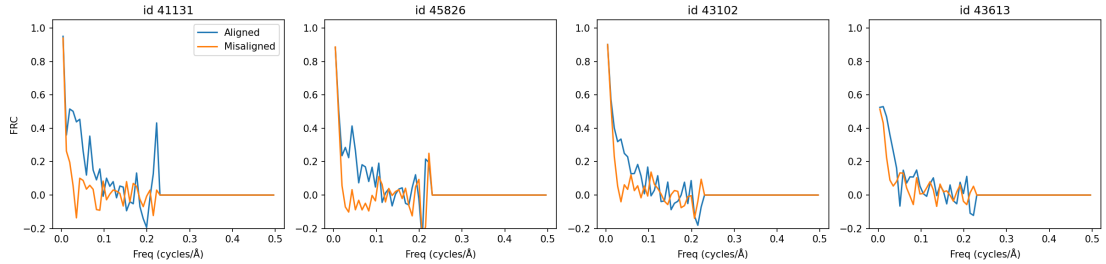


Figure 5.16: *FRC curves at shift perturbation Range 1*

5.3.2 Generalization Across Datasets

The model was finally tested in `-mode predict` to identify whether the trained model was capable of classifying images it had never seen. The test was set up as follows: the ribosome dataset was used as input to the model trained on the SARS-CoV-2 Spike protein dataset and misaligned versions of the projections were artificially generated, expecting to obtain 50% of true positives (aligned particles correctly classified as aligned) and 50% of true negatives (misaligned particles correctly classified as misaligned). It is worth mentioning that the images used to train *cryoCheck* had a sampling rate of 2.19, whereas the ribosome dataset used for testing had a different pixel size, corresponding to a sampling rate of 3.14 Å/px. Thanks to the `standardize_frc_curve` function, however, the frequency range over which the FRC curves are plotted remains consistent across both datasets. The results in terms of confusion matrix are shown in Figure 5.17 with its correlated metrics in Table 5.9.

Accuracy	Precision	Recall	F1-score
99.84%	99.98%	99.69%	99.85%

Table 5.9: *Classification metrics in prediction mode*

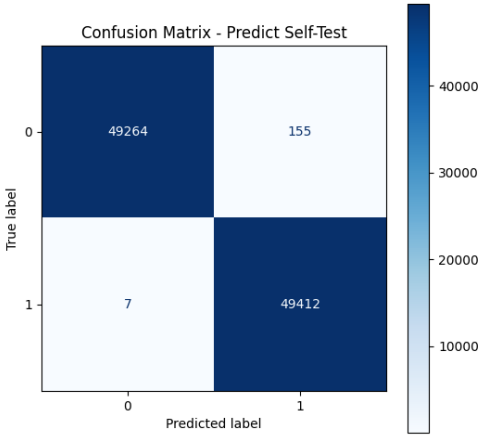


Figure 5.17: *Confusion matrix belonging to cryoCheck in predict mode*

This test had two objectives: not only to evaluate the model's performance but also to demonstrate that it can be applied to datasets containing particles with a completely different structure. Since *cryoCheck* analyzes FRC curves rather than image patterns, it is able to generalize well across structurally diverse datasets without requiring retraining. This suggests that, in practice, it may not be necessary to train the model separately on a wide range of datasets to achieve reliable performance on previously unseen structures. In conclusion, the experimental results are highly encouraging: *cryoCheck* showed strong performance across all tested scenarios, supporting its application as a generalized model.

6. Conclusions

The aim of this project was to introduce a deep learning tool to automatically detect alignment errors in 3D cryo-EM maps. After assessing the importance that a good 3D map has, as it is strongly linked to the macromolecules' function, it is crystal clear that particle misalignment can introduce artifacts during reconstruction and can affect how accurate the atomic modeling is.
Cryo-EM images are intrinsically noisy and it is the major challenge that the DL model had to handle. This is the reason why particles images were not directly employed as inputs of the model, instead, FRC curves, a frequency-domain feature associated to the correlation between the real experimental image and the corresponding volume projection in Fourier space, were used. Upon consultation of the results of Chapter 5, the model achieved a good classification performance both for the simulated and the experimental data. Different metrics and characteristics of the model were extracted and all of them seem to agree on the ability of the model to detect misalignment, even if it is slight. FRC curves tendencies clearly distinguish when the experimental image is consistent with its projection. The aligned one always tends to zero in a slower manner compared to the misaligned one, even when images are strongly affected by noise, like in the SARS-CoV Spike protein dataset. A part from this visual analysis, also the metrics extracted from the confusion matrices assess the robustness of the model achieving accuracy values up to 0.99 for the experimental data, taking into account the reliability of this metric in this study case, where the two classes are balanced.
Choosing FRC curves over cryo-EM images as inputs to the model had several benefits: no intensive preprocessing (e.g. filtering) of the images, an easier neural network architecture can be employed and computational training times heavily decrease. It was not a straightforward process: the initial idea of the project was to build a model based on a CNN architecture, a ResNet-50-style backbone with approximately 2.79 million trainable parameters. Residual images were given as input to train the model to discriminate correctly aligned images from misaligned ones. Two residuals were computed for each experimental image, obtained by subtracting the image itself from its aligned and misaligned projections, respectively. When employing those inputs, the model was not able to achieve good classification results due to noise: cryo-EM images are so affected by noise that even when subtracting the images or correcting the CTF, the real overall shape of the particles was difficult to recognize.

7. Future Outlooks

While working on this project, various ideas for future development have emerged. These can be divided into two categories: adjustments to the model's design and further exploration of the model's potential applications.
One possible improvement concerns the `standardize_frc_curve` function. At present, this function standardizes the frequency axis on which the FRC curves are plotted, using an interpolation strategy. FRC curves are computed over $n_{\text{shells}} = x_{\text{size}}/2$ shells, so n_{shells} varies depending on the box size and/or pixel size, leading to curves of different lengths. Two scenarios can occur:

- The model receives as inputs images with different pixel sizes: the issue that has already been addressed by the `standardize_frc_curve` function.
- The model receives as inputs images with different box sizes: every curve cover the same frequency interval but but the sampling density differs, which in real space corresponds to a different field of view of the object.

Regarding this second case, if the box size used to compute the curves is too small, the target object is clipped in real space, it means that it is physically cut off at the image boarders, leading to a loss of information in Fourier space. On the contrary, if the box size is too large, it can introduce artifacts that can affect the SNR of the image. A solution could be to standardize the n_{shells} using a large reference box size (e.g. 1000 px). Smaller images could then be padded, resulting in smoother FRC curves; larger images could instead be cropped: if no information is lost in the process, a smoother FRC curves is obtained, whereas if the target object is affected by the cropping, the FRC curve becomes more discontinuous, and an interpolation strategy may need to be applied.
Further testing could be carried out to assess *cryoCheck*'s generalization capabilities. One possible approach would be to test the model on additional datasets, extract the particles it classifies as misaligned, and reconstruct a volume using only those particles: if the model performs correctly, the resulting reconstruction should differ noticeably from the reference volume. If the model generalizes well, the associated `hax-train-misalignment detection` protocol can be integrated in Scipion, in order to be used as part of the processing pipeline.
Then, a foundation model can be developed. A foundation model is a deep-learning tool trained of a massive amount of data and capable of addressing multiple tasks, rather than a single one. A promising direction in this sense would be to extend *cryoCheck* beyond binary classification, introducing a regression-based loss function capable of estimating the magnitude of misalignment — rather than simply detecting its presence — directly

from the FRC curve. This would represent a step toward a more general-purpose tool for pose quality assessment in cryo-EM.

Bibliography

[1] Ruben Sanchez-Garcia, Alex Berndt, Amir Apelbaum, Judith Reeks, Pamela A Williams, Carl Poelking, Charlotte Deane, and Michael Saur. «Supervised Deep Learning for Efficient Cryo-EM Image Alignment in Drug Discovery with cryoPARES». In: *bioRxiv* (2026) (cit. on pp. 7, 17).

[2] Kazuyoshi Murata and Matthias Wolf. «Cryo-electron microscopy for structural analysis of dynamic biological macromolecules». In: *Biochimica et Biophysica Acta* 1862.2 (2018), pp. 324–334. DOI: 10.1016/j.bbagen.2017.07.020 (cit. on pp. 8–10, 13).

[3] Jose Luis Vilas, Jose Maria Carazo, and Carlos Oscar S. Sorzano. «Emerging Themes in CryoEM Single Particle Analysis Image Processing». In: *Chemical Reviews* 122.17 (2022), pp. 13915–13951. DOI: 10.1021/acs.chemrev.1c00850 (cit. on pp. 8, 10–13, 16).

[4] Helen R. Saibil. «Cryo-EM in molecular and cellular biology». In: *Molecular Cell* 82.2 (2022), pp. 274–284. DOI: 10.1016/j.molcel.2021.12.016 (cit. on pp. 9, 14).

[5] Cheng Y, Grigorieff N, Penczek PA, and Walz T. «A primer to single-particle cryo-electron microscopy». In: *Cell* (2015) (cit. on p. 9).

[6] Ashwin Dhakal, Rajan Gyawali, Liguo Wang, and Jianlin Cheng. «CryoPPP: A Large Expert-Labelled Cryo-EM Image Dataset for Machine Learning Protein Particle Picking». In: *bioRxiv* (2023) (cit. on p. 10).

[7] Structural Biotechnology Inc. *CTF Estimation*. https://guide.cryosparc.com/processing-data/all-job-types-in-cryosparc/ctf-estimation. Accessed: 2026-07-06. 2026 (cit. on p. 11).

[8] Tamir Bendory, Alberto Bartesaghi, and Amit Singer. «Single-Particle Cryo-Electron Microscopy: Mathematical Theory, Computational Challenges, and Opportunities». In: *IEEE Signal Processing Magazine* (2020) (cit. on p. 11).

[9] Structura Biotechnology Inc. *CTF Estimation*. CryoSPARC Guide. Accessed 22 agosto 2026. URL: https://guide.cryosparc.com/processing-data/all-job-types-in-cryosparc/ctf-estimation (cit. on p. 12).

[10] Feng et al. Wang. «DeepPicker: A deep learning approach for fully automated particle picking in cryo-EM». In: *Journal of structural biology* (2016) (cit. on pp. 12, 16).

[11] Tristan et al. Bepler. «Positive-unlabeled convolutional neural networks for particle picking in cryo-electron micrographs». In: *Nat Methods.* (2019) (cit. on pp. 12, 16).

[12] Thorsten et al. Wagner. «SPHIRE-crYOLO is a fast and accurate fully automated particle picker for cryo-EM». In: *Communications Biology* (2019) (cit. on pp. 12, 16).

[13] Tiago R. D. Costa, Athanasios Ignatiou, and Elena V. Orlova. «Structural Analysis of Protein Complexes by Cryo Electron Microscopy». In: Springer New York, 2017, pp. 377–413 (cit. on p. 12).

[14] MIT CSAIL. *Cryo-Electron Microscopy*. `https://people.csail.mit.edu/gdp/cryoem.html`. Accessed 22 August 2026 (cit. on p. 13).

[15] A. Punjani, J. Rubinstein, and D. et al. Fleet. «cryoSPARC: algorithms for rapid unsupervised cryo-EM structure determination». In: *Nat Methods* (2017) (cit. on pp. 13, 21).

[16] Sjors H.W. Scheres. «A Bayesian View on Cryo-EM Structure Determination». In: *Journal of Molecular Biology* (2012) (cit. on pp. 13, 16, 21).

[17] C. O. S. et al. Sorzano. «On bias, variance, overfitting, gold standard and consensus in single-particle analysis by cryo-electron microscopy.» In: *Acta crystallographica Section D: Structural biology* (2022) (cit. on p. 14).

[18] Gabriel C. Lander. «Single particle cryo-EM map and model validation: It's not crystal clear». In: *Current Opinion in Structural Biology* (2024) (cit. on p. 14).

[19] Yonggang Lu et al. «3D reconstruction from cryo-EM projection images using two spherical embeddings». In: *Communications Biology* (2022) (cit. on p. 14).

[20] Dmitry Lyumkis. «Challenges and opportunities in cryo-EM single-particle analysis». In: *Journal of Biological Chemistry* (2019) (cit. on p. 15).

[21] Adil et al. Al-Azzawi. «AutoCryoPicker: an unsupervised learning approach for fully automated single particle picking in Cryo-EM images». In: *BMC bioinformatics* (2019) (cit. on p. 16).

[22] Rajan Gyawali, Ashwin Dhakal, Liguo Wang, and Jianlin Cheng. «CryoSegNet: accurate cryo-EM protein particle picking by integrating the foundational AI image segmentation model and attention-gated U-Net». In: *Briefings in Bioinformatics* (2024) (cit. on p. 16).

[23] S. Iqbal, E.T. Eng, and M.A. et al. Kamal. «Artificial intelligence in cryo-EM: emerging deep neural network methods from sample preparation, particle picking, map reconstruction, modelling to enhanced resolution». In: *BMC Artificial Intelligence* (2026) (cit. on p. 16).

[24] Ruben Sanchez-Garcia, Joan Segura, David Maluenda, C.O.S. Sorzano, and J.M. Carazo. «MicrographCleaner: A python package for cryo-EM micrograph cleaning using deep learning». In: *Journal of Structural Biology* (2020) (cit. on p. 16).

[25] Ruben et al. Sanchez-Garcia. «Deep Consensus, a deep learning-based approach for particle pruning in cryo-electron microscopy». In: *IUCrJ* (2018) (cit. on p. 16).

[26] A.W. DiIorio M.C.; Kulczyk. «Intelligence-Based Approaches for Ab Initio Structure Determination and Atomic Model Building for Cryo-Electron Microscopy». In: *Micromachines* (2023) (cit. on p. 16).

[27] Youssef S. G. Nashed, Frédéric Poitevin, Harshit Gupta, Geoffrey Woollard, Michael Kagan, Chun Hong Yoon, and Daniel Ratner. «CryoPoseNet: End-to-End Simultaneous Learning of Single-particle Orientation and 3D Map Reconstruction from Cryo-electron Microscopy Data». In: *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*. 2021 (cit. on p. 16).

[28] Axel Levy, Frédéric Poitevin, Julien Martel, Youssef Nashed, Ariana Peck, Nina Miolane, Daniel Ratner, Michael Dunne, and Gordon Wetzstein. «CryoAI: Amortized Inference of Poses for Ab Initio Reconstruction of 3D Molecular Volumes from Real Cryo-EM Images». In: *Computer Vision – ECCV 2022*. 2022 (cit. on p. 16).

[29] Harshit Gupta, Michael T. McCann, Laurène Donati, and Michael Unser. «CryoGAN: A New Reconstruction Paradigm for Single-Particle Cryo-EM Via Deep Adversarial Learning». In: *IEEE Transactions on Computational Imaging* (2021) (cit. on p. 16).

[30] Ellen Zhong, Adam Lerer, Joseph Davis, and Bonnie Berger. «CryoDRGN2: Ab initio neural reconstruction of 3D protein structures from real cryo-EM images». In: *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021 (cit. on p. 16).

[31] Axel Levy, Gordon Wetzstein, Julien Martel, Frederic Poitevin, and Ellen D. Zhong. *Amortized Inference for Heterogeneous Reconstruction in Cryo-EM*. 2022. arXiv: 2210.07387 [cs.CV]. URL: https://arxiv.org/abs/2210.07387 (cit. on p. 16).

[32] Ali Punjani and David J Fleet. «3DFlex: determining structure and motion of flexible proteins from cryo-EM». In: *Nat Methods* (2023) (cit. on p. 16).

[33] Xiao et al. Fan. «CryoTRANS: predicting high-resolution maps of rare conformations from self-supervised trajectories in cryo-EM.» In: *Commun Biology* (2024) (cit. on p. 16).

[34] Yilai Li, Yi Zhou, Jing Yuan, Fei Ye, and Quanquan Gu. «CryoSTAR: leveraging structural priors and constraints for cryo-EM heterogeneous reconstruction». In: *Nat Methods* (2024) (cit. on p. 16).

[35] Noddings C. et al. Herreros D. Mata C.P. «Real-space heterogeneous reconstruction, refinement, and disentanglement of CryoEM conformational states with HetSIREN». In: *Nature Communications* (2025) (cit. on p. 16).

[36] R. Sanchez-Garcia, J. Gomez-Blanco, A. Cuervo, and et al. «DeepEMhancer: a deep learning solution for cryo-EM volume post-processing». In: *Commun Biol.* (2021) (cit. on p. 17).

[37] H. Cao, Y. Zhu, and T. Li. «EMReady2: improvement of cryo-EM and cryo-ET maps by local quality-aware deep learning with Mamba». In: *Nat Commun* (2026) (cit. on p. 17).

[38] J. Abramson, J. Adler, J. Dunger, et al. «Accurate structure prediction of biomolecular interactions with AlphaFold 3». In: *Nature* (2024) (cit. on p. 17).

[39] Liu YT, Fan H, Hu JJ, and Zhou ZH. «Overcoming the preferred orientation problem in cryoEM with self-supervised deep-learning». In: *bioRxiv* (2024) (cit. on p. 17).

[40] James Bradbury et al. *JAX: composable transformations of Python+NumPy programs*. Version 0.3.13. 2018. URL: `http://github.com/jax-ml/jax` (cit. on p. 18).

[41] D. Herreros, C.P. Mata, and C. Noddings. «Real-space heterogeneous reconstruction, refinement, and disentanglement of CryoEM conformational states with HetSIREN». In: *Nat Commun* (2025) (cit. on pp. 19, 20).

[42] Xiao-chen Bai, Israel S Fernandez, Greg McMullan, and Sjors HW Scheres. «Ribosome structures to near-atomic resolution from thirty thousand cryo-EM particles». In: *eLife* (2013) (cit. on p. 19).

[43] Marvin Ritter et al. *Grain - Feeding JAX Models*. Version 0.2.12. 2023. URL: `http://github.com/google/grain` (cit. on p. 20).

[44] Suyi Chen and Haibin Ling. «CryoSplat: Gaussian Splatting for Cryo-EM Homogeneous Reconstruction». In: *eLife* (2026) (cit. on p. 21).

[45] Guikun Chen and Wenguan Wang. *A Survey on 3D Gaussian Splatting*. 2026. arXiv: `2401.03890 [cs.CV]`. URL: `https://arxiv.org/abs/2401.03890` (cit. on p. 21).

[46] CCP-EM. *Euler angle conventions*. `https://www.ccpem.ac.uk/docs/euler-angle-conventions/` (cit. on p. 22).

[47] Lingbo Yu, Robert R. Snapp, Teresa Ruiz, and Michael Radermacher. «Projection-based volume alignment». In: *Journal of Structural Biology* (2013) (cit. on p. 22).

[48] Marin Van Heel. «Similarity measures between images». In: *Ultramicroscopy* (1987) (cit. on p. 25).

[49] Jonathan Heek, Anselm Levskaya, Avital Oliver, Marvin Ritter, Bertrand Rondepierre, Andreas Steiner, and Marc van Zee. *Flax: A neural network library and ecosystem for JAX*. Version 0.12.8. 2024. URL: `http://github.com/google/flax` (cit. on p. 26).

[50] DeepMind et al. *The DeepMind JAX Ecosystem*. 2020. URL: `http://github.com/google-deepmind` (cit. on p. 27).

[51] J.M. de la Rosa-Trevín et al. «Scipion: A software framework toward integration, reproducibility and validation in 3D electron microscopy». In: *Journal of Structural Biology* (2016) (cit. on p. 28).

[52] David Herreros et al. *Hax: A suite for Heterogeneity analysis for CryoEM data*. `https://github.com/DavidHerreros/hax-em`. 2026 (cit. on p. 28).

[53] Odei Obeng et al. «Using MUAC to predict and avoid negative outcomes in CMAM programmes: Work inspired by en-net». In: *Field Exchange 60 (en)* (2019) (cit. on p. 32).