Proof of Concept Demonstration and Evaluation – Use Case Electron Microscopy

[WP1; T1.2; Deliverable: D1.2, version M36]

public

AI training using Simulated Instruments for Machine Optimization and Verification

Document Information

Project	ASIMOV
Grant Agreement No.	20216 ASIMOV - ITEA
Deliverable No.	D1.2
Deliverable No. in WP	WP1; T1.2
Deliverable Title	Proof of Concept Demonstration and Evaluation – Use Case Electron Microscopy
Dissemination Level	public
Document Version	M36
Date	2024.05.07
Contact	Holger Kohr
Organization	Thermo Fisher Scientific
E-Mail	holger.kohr@thermofisher.com

The ASIMOV-project was submitted in the Eureka Cluster AI Call 2021 https://eureka-clusters-ai.eu/

Task Team (Contributors to this deliverable)

Name	Partner	E-Mail
Hans Vanrompay	TFS	Hans.Vanrompay@thermofisher.com
Narges Javaheri	TFS	Narges.Javaheri@thermofisher.com
Maurits Diephuis	TFS	Maurits.Diephuis@thermofisher.com
Ilona Armengol	TNO	Ilona.ArmengolThijs@tno.nl
Richard Doornbos	TNO	Richard.Doornbos@tno.nl
Bram Van der Sanden	TNO	bram.vandersanden@tno.nl
Jilles van Hulst	TU/e	J.S.V.Hulst@tue.nl
Roy van Zuijlen	TU/e	R.A.C.Zuijlen@tue.nl
Holger Kohr	TFS	Holger.kohr@thermofisher.com

Formal Reviewers

Version	Date	Reviewer
M36	2024.05.07	Niklas Braun (AVL)
M36	2024.05.07	Jan Willem Bikker (CQM)

Change History

Version	Date	Reason for Change
M36	2024-04-24	Contact changed to Holger Kohr; contributor added

Abstract

This work presents the latest advancements in leveraging artificial intelligence (AI) and digital twin (DT) technologies to automate the operation of electron microscopes, with a particular focus on exploring the feasibility of automated electron microscope alignment.

The alignment of electron microscopes is a laborious process, demanding significant time and expert knowledge. This time-intensive operation not only affects the actual experimental time available for microscope operators to investigate nanomaterials.

In our approach, we employ DT and AI methodologies to automate and expedite these alignment procedures, aiming to streamline the operation of electron microscopes, reduce manual effort, and enhance the efficiency of both routine operations and the calibration of new microscope installations.

Table of Contents

Table of Figures

1. Electron Microscopy

1.1 Transmission Electron Microscope

The process of forming images in Transmission Electron Microscopy (TEM) closely resembles that of an optical light microscope, with the primary distinction being the utilization of electrons as the light source and the consequent substitution of glass lenses with electromagnetic coils.

In TEM, high voltages are employed to accelerate electrons toward the specimen, allowing for higher resolution images compared to optical light microscopes. These electrons are emitted from either a thermionic gun or a field emission gun (FEG) and subsequently pass through a series of condenser lenses to generate a beam with specified size, intensity, and convergence. In TEM mode, a parallel coherent beam is created, ensuring uniform illumination of the sample. Conversely, in Scanning Transmission Electron Microscopy (STEM) mode, the beam is focused into a fine probe that scans across the specimen [1].

Following the formation of the desired electron beam, it interacts with the specimen positioned in a dedicated holder, located between the two pole pieces of the objective lens. The objective lens then focuses the transmitted electrons into a diffraction pattern in the back focal plane, where they recombine to produce an enlarged image of the specimen in the objective lens's image plane. An ideal lens system is anticipated to image a single point source as a point.

Scherzer [2], however, demonstrated that for round symmetric electromagnetic lenses, aberrations are unavoidable. These aberrations contribute to blurring of the image and therefore a loss in resolution. For both the condenser- and objective lenses, stigmators are therefore present which apply an asymmetric, correcting field to minimize the effect of the aberrations.

Beneath the objective lens, a series of intermediate and projector lenses work in tandem to generate a magnified image, either of the sample in real space or the corresponding diffraction pattern in reciprocal space. Visualization of this image can be accomplished using devices such as a Charged Coupled Device (CCD) or a direct electron detector. The comprehensive assembly of a transmission electron microscope is depicted in Figure 1.

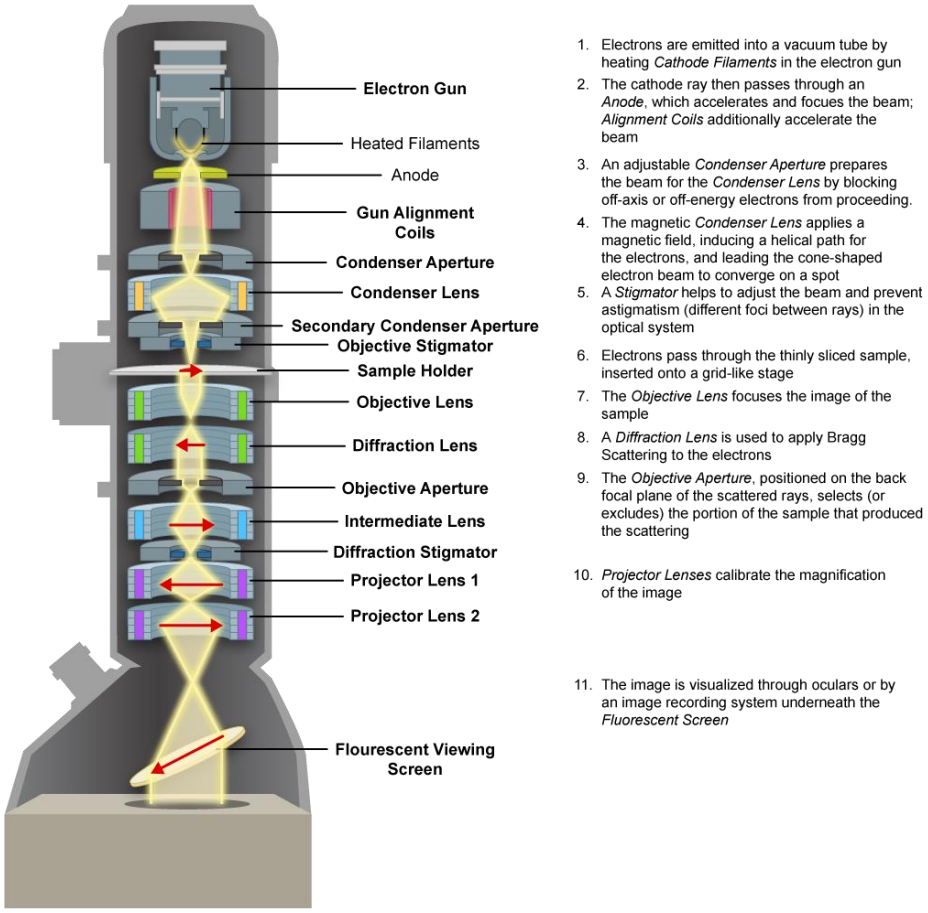

Figure 1: Illustration of the components of an electron microscope [3]

1.2 Ronchigram

In the context of STEM, the use of Ronchigram images plays a crucial role in determining optimal parameters to minimize aberrations and enhance spatial resolution. The term "Ronchigram" draws inspiration from the "Ronchi test," a standardized method for shaping optical lenses devoid of aberrations. This involves introducing a diffraction grating into the optical lens's focus, enabling the identification of lens imperfections through the resulting interference pattern.

However, the application of Ronchi's grating in TEM proves impractical. The accelerated electrons' high frequency necessitates grating spacings in the order of a few picometers for interference to occur, posing a significant challenge in construction.

Instead, TEM relies on the atomic arrangement in amorphous materials, offering a nearly random distribution of atomic potentials. This arrangement approximates a noisy grating, simulating the Ronchi test and generating interference patterns that unveil aberrations in electromagnetic lenses.

Crucial elements within the Ronchigram, exploitable for correcting lower-order aberrations, include its symmetry and magnification. In a focused state, the Ronchigram's central region exhibits high local magnification, representing the aberration-free segment of the electron beam. Moving away from the center along the optical axis introduces aberrations, resulting in reduced local magnification (Figure 2, center row). A maximally magnified central region serves as an initial indicator of optimal defocus.

Asymmetric aberrations disrupt the rotational symmetry of the Ronchigram, with two-fold astigmatism elongating the high magnification region unidirectionally, producing distinct streaks (refer to Figure 2, bottom row). Axial coma shifts the Ronchigram's center, and higher-order aberrations further compromise its symmetry (refer to Figure 2, top row).

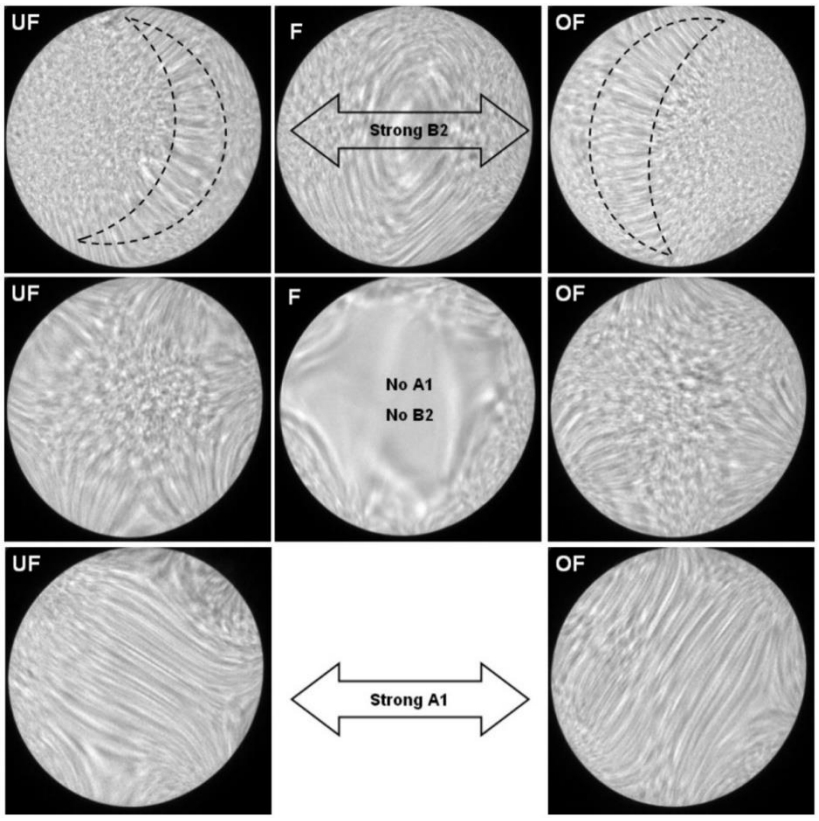

Figure 2: Visualizations of a Ronchigram obtained on amorphous carbon in underfocus (UF) and overfocus (OF) as a function of astigmatism (A1) and axial coma (B2). [4]

If we want to train a reinforcement learning agent to correct the lower order aberrations, it is essential that we have large amounts of labeled Ronchigram images. Obtaining such images would require substantial microscope time which is expensive. To alleviate the need for physical system time we will therefore explore Digital Twinning (DT) as an alternative to create labeled Ronchigram data.

2. Advancements in the Digital Twin

2.1 Dependency on sample's material

The Ronchigram is a diffraction pattern which conveys information about both, properties of the electron beam and, the sample in the microscope. This means that although we are interested in electron beam aberrations for the current application, the sample affects the appearance of the Ronchigram. Therefore, it is essential to develop and test our aberration correction solution on relevant samples/materials.
the initial use case, the interacting of the electron wave with amorphous materials was simulated to generate the dataset of Ronchigram images for training the neural network (NN) model. We have extended our simulation to support crystalline materials in the desired crystal orientation. Consequently, new datasets for training AI were generated and compared to the measured data of crystalline samples. Figure 3 and Figure 4 show the response of a Ronchigram (and its Fourier transform; FT) to the change in aberrations for amorphous and crystalline materials. Although there are clear differences between simulated and real data the trend in the shape of FT images as a function of aberrations (defocus and astigmatism) is similar. As in the case of the amorphous materials, the challenging task for the AI model is to learn the trend and ignore the dissimilarities. However, in the case of crystalline materials the pattern of frequencies observed in the FT image are more complicated due to the structured diffraction rather than the random diffraction of sample. In such a case augmenting the training data, e.g., by random rotations, could potentially help the generalization to the real-world data.

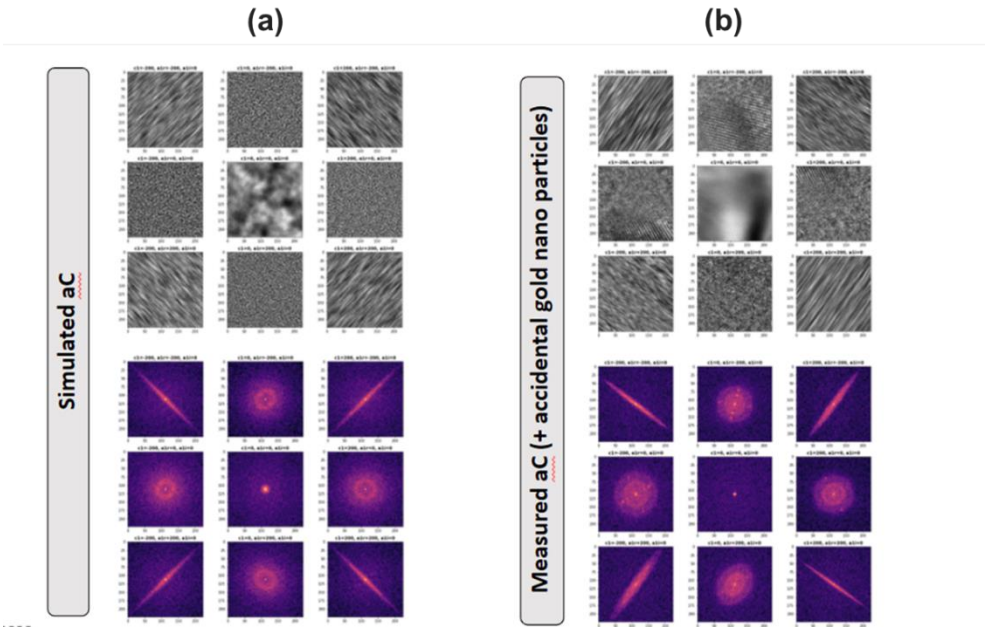

Figure 3: Ronchigram and the amplitude of its Fourier transform as a function of defocus (x-axis) and 2-fold astigmatism the real part (y-axis). (a) Simulated data with amorphous Carbon (aC) materials. (b) Measured data on a sample with amorphous Carbon and some Gold nanoparticles

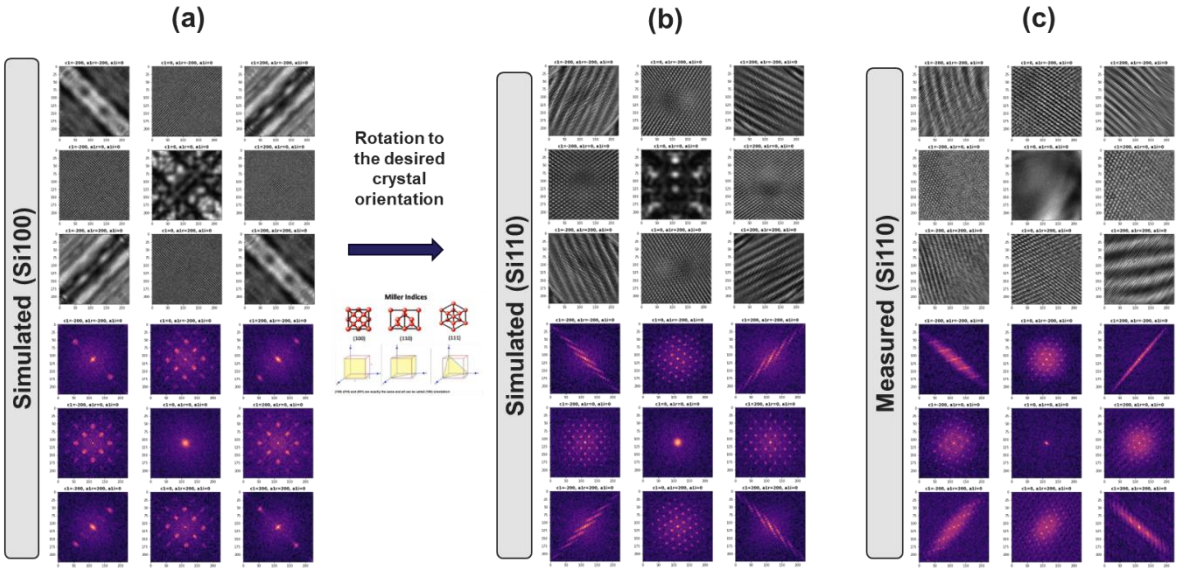

Figure 4: Ronchigram and the amplitude of its Fourier transform as a function of defocus (x-axis) and 2-fold astigmatism the real part (y-axis). (a) Simulated data with Si100 crystals. (b) Simulated data with Si110 crystals, (c) Measured data on a sample with Si110 region.

2.2 Defining a distance-to-goal metric

One of the methods for tunning aberrations in TEM that we have explored is to use a supervised learning network to predict all aberrations simultaneously. However, this is a complicated task, and it requires multiple Ronchigram images as input to the NNs. Alternatively, we can change the task of the NNs to predict the distance to the best aberration state, i.e., the distance-to-goal, which is a scalar and requires only a single image. To tune the aberrations, an optimization algorithm should work on top of SV minimizing the inferred distance-to-goal. This method is discussed in detail in chapter 3. Here, the different flavours of the distance-to-goal metric which are generated by the DT are discussed.

We have employed two types of metrics for distance-to-goal, based on:

- **Explicit use of aberration coefficients**:
  Here the known individual aberration coefficients from simulated data are used, e.g., in L2 form, to calculate the distance. When higher order aberrations are present, we use a scaled aberration distance, which follows the same scaling as in aberration function itself. In this metric, effectively, the higher order aberrations are scaled with a smaller number.

- **Implicit use of aberration coefficients**:
  The effect of combined aberrations is calculated during the simulation by a metric which is then used for training. Specifically, the radius of the largest circle with phase error of less than pi/4 is

calculated (see Figure 6), which subsequently, is translated to a more convenient metric; the Rayleigh resolution (in Angstrom) of the wave with aperture size equal to the above circle. From this type, more metrics which measure the beam quality can also be used. The main difference between explicit and implicit metrics arises from the combined effect of aberrations. In reality the effect of some aberrations can cancel out each other, which is not reflected in the explicit metric, while the implicit metrics represent the observed effect of combined aberrations, and thus, the final resolution of the STEM image more truthfully.

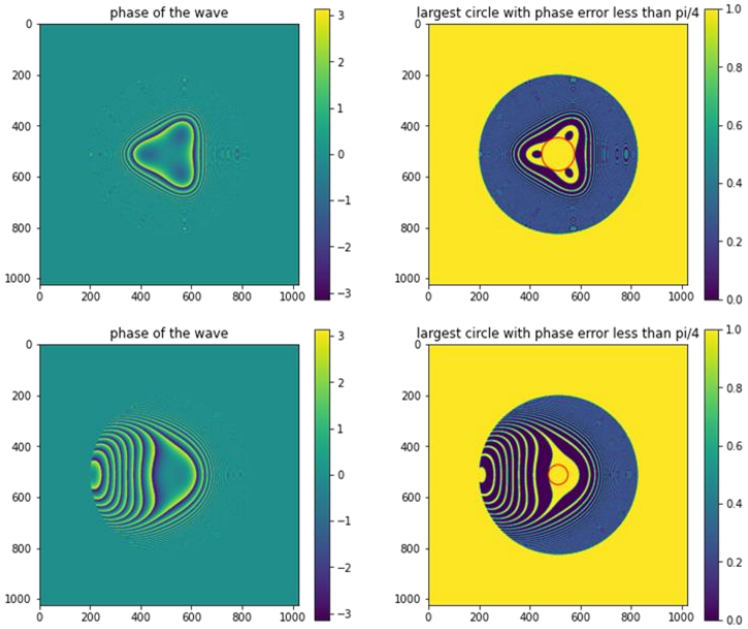

Figure 5: The radius of the largest circle with small (less than π/4) phase error. Each row shows an example of aberration configurations.

2.3 Flexible STEM imaging

Additionally, we have enabled producing STEM images with the simulator. In STEM imaging multiple Ronchigram patterns, each from one spot on the sample, are generated. For instance, in HAADF (high-angle annular dark-field) a ring-shape detector will collect the intensity in part of the Ronchigram and the resulting reported number produces one pixel in the STEM image. There are, however, other modes of STEM imaging depending on the detector shape and the post processing of the data it collects. To enable all these modes for the future, a flexible detector definition was added. Figure 6 shows an example of a STEM detector. Moreover, the simulation of STEM is far more computationally expensive than a single Ronchigram. Therefore, we will further develop methods for faster calculations.

Currently, we work with the Ronchigram data for aberration estimation, however, enabling STEM imaging enables more applications, e.g., the use of STEM data for aberration estimation. Moreover, the final

quality of the electron beam in STEM mode is measured by the resolution of the STEM image. By enabling STEM simulation, we can predict the resolution given the estimated aberrations and compare this to the real data.

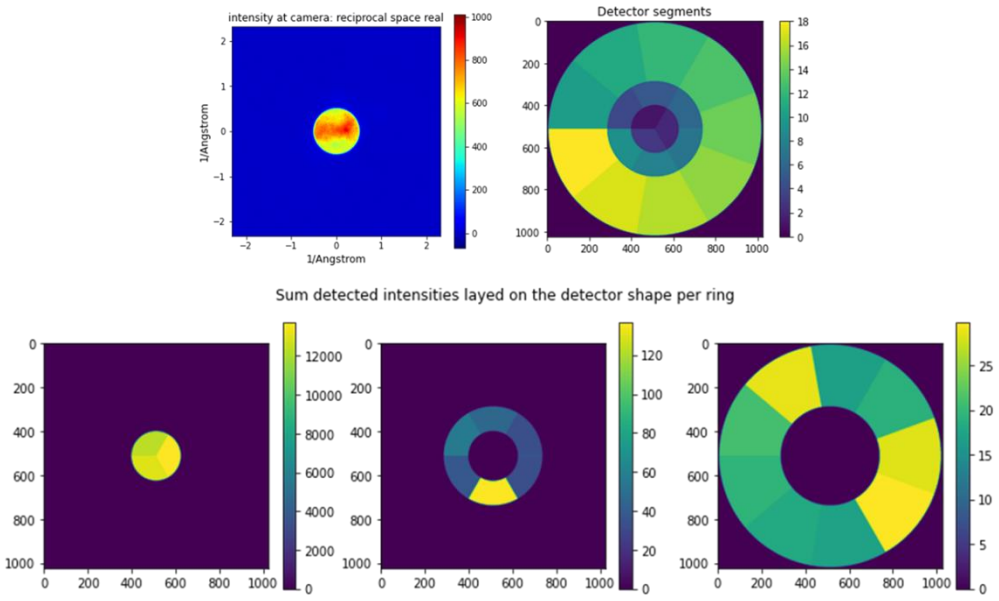

Figure 6: An example of a STEM detector with multiple rings and multiple segments on each ring. The total intensity on each segment is reported as a single number which then is used for a specific STEM imaging mode

3. Exploration of Bayesian optimisation for microscope alignment

We have applied Bayesian optimization (BO) with a Gaussian process (GP) estimator and expected improvement acquisition function to the aberration calibration problem. The to-be-optimized function is a cost function which is constructed based on manual features constructed from the Fourier transform of the Ronchigram images. Ronchigrams of amorphous carbon under the presence of defocus, two-fold astigmatism, and low higher-order aberrations have Fourier transforms which are generally elliptical in shape. When both defocus and two-fold astigmatism are low, we obtain a small circle in Fourier transform. This motivates a cost function which penalizes the two axes that describe an elliptical fit of the Fourier transform shape. Specifically, we interpret the Fourier shape as a 2-dimensional Gaussian and find the ellipse axes through the eigenvalues of the Guassian covariance matrix. The feature extraction process is summarized in Figure 7.

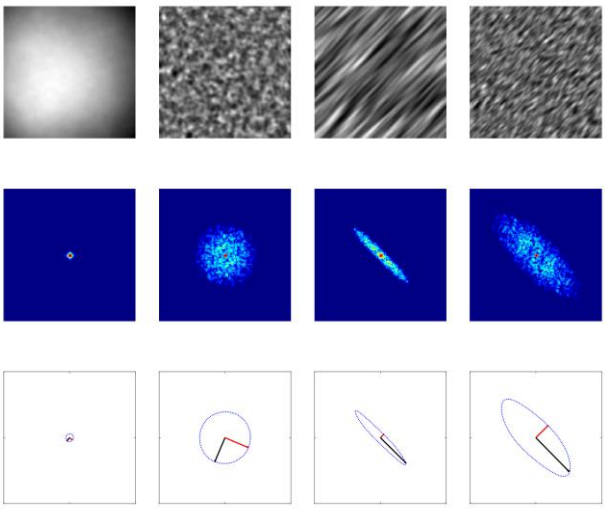

Figure 7: The feature extraction process of the Ronchigram. The first row displays four synthetic Ronchigrams which contain, respectively, no aberrations (fully calibrated), underfocus, astigmatism with slight underfocus, and astigmatism with severe underfocus. The second row displays the same four images after taking the Fourier transform in dB scale. The third row displays the eigenvectors of a Gaussian covariance matrix fitted to the second row of images. The sizes of the eigenvectors in the figure (given by the eigenvalues) are used to construct the cost function used for GP/BO.

This combined GP/BO approach effectively deals with two important properties of the EM calibration problem: the smoothness of the ellipse features, and the fact that individual observations are non-Markov.

The Gaussian process estimator exploits the smoothness of the ellipse parameters and creates an extremely sample-efficient estimate. Specifically, we use a standard squared-exponential kernel function which assumes that the cost function has some smoothness. The degree of smoothness is a hyperparameter in the GP, which can be tuned based on the data using marginal likelihood.

The fact that the individual observations are non-Markov implies that from a single image, we can generally not uniquely determine the state of the microscope. Bayesian optimization uses the full history of observations to restore the Markov property. Specifically, we use the GP estimate which characterizes our belief of the cost landscape in terms of an expected value and an expected covariance, given the observations. From this estimate, we can determine which areas of the cost landscape are most promising depending on how we expect that area to perform. This expectation is captured in a so-called

acquisition function. We use the standard expected improvement acquisition function, multiplied by a factor that prioritizes the neighbourhood of the best setting so far.

Since we cannot directly measure the aberrations on a real TEM, a stopping condition based on the measurements is required. In this approach, we use the fact that we know the ellipse features that correspond to the calibrated image. For this reason, we stop the automated calibration process once the measured cost is sufficiently low.

Calibration performance (of C1 and A1x) on simulated data is extremely fast and consistent. Empirically, in 493 out of 500 attempts, we achieved calibration settings that are within a Euclidean distance of 20 nm. In all attempts, we stop the process within a Euclidean distance of 30 nm. Additionally, the average number of samples needed before we reach the stopping condition is 6.5, with a standard deviation of 3.8. In Figure 8 we show an example run. Notice that initially, the extremes are explored. Afterwards, the most promising regions are exploited. After 7 observations, the stopping condition is reached.

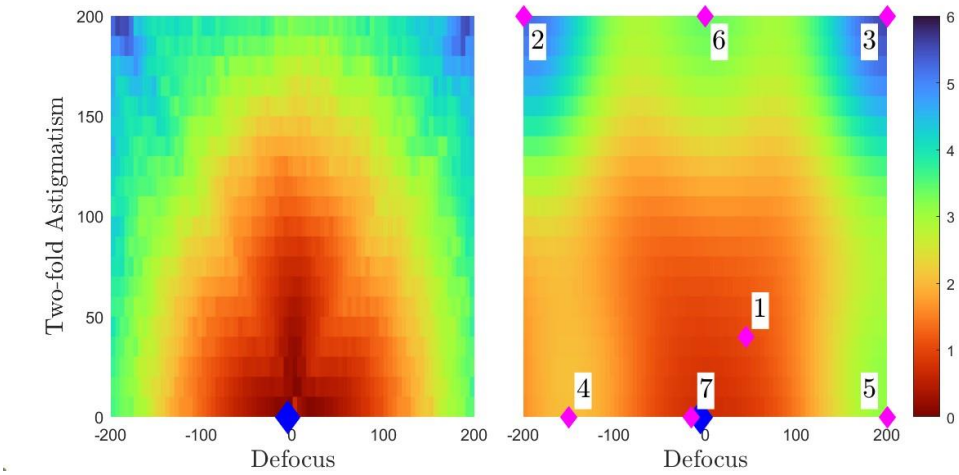

Figure 8: Left: Cost landscape generated using elliptical features based on Fourier transform of synthetic Ronchigrams. The optimum is highlighted using the blue diamond. Right: Gaussian process estimate of the cost landscape using 7 noisy measurements. The numbers indicate the sequential order of the next measurement locations that BO proposes.

4. Strategies to improve the explainability of Reinforcement Learning

Understanding the behaviour of a black-box AI software is desirable. In the development phase researchers can learn to take the right decision towards improving the solution by making the right development choices. In value-iteration approaches, like DQN, visualising the q-values and the actions taken allows to understand and characterize the learned policy; the learned policy can then be analysed, and this information can in turn be used to set new requirements.

In the first generation (correction of C1 and positive A1x values), the models, generated through training on the provided data, were successfully applied to the electron microscope for the first time. Visualization of the sum and maximum of the Q-values, and the corresponding actions was performed to comprehend the policy. While not flawless, this approach served as a tangible proof of concept, indicating that the agent was indeed learning. In Figure 9 and Figure 10, we showcase simulated and experimental Q-maps, the brightest yellow spots indicate the goal, while darker areas indicate further position from goal. Such maps could be used for example to assess the impact of different pre-processing methods on the learned policy.

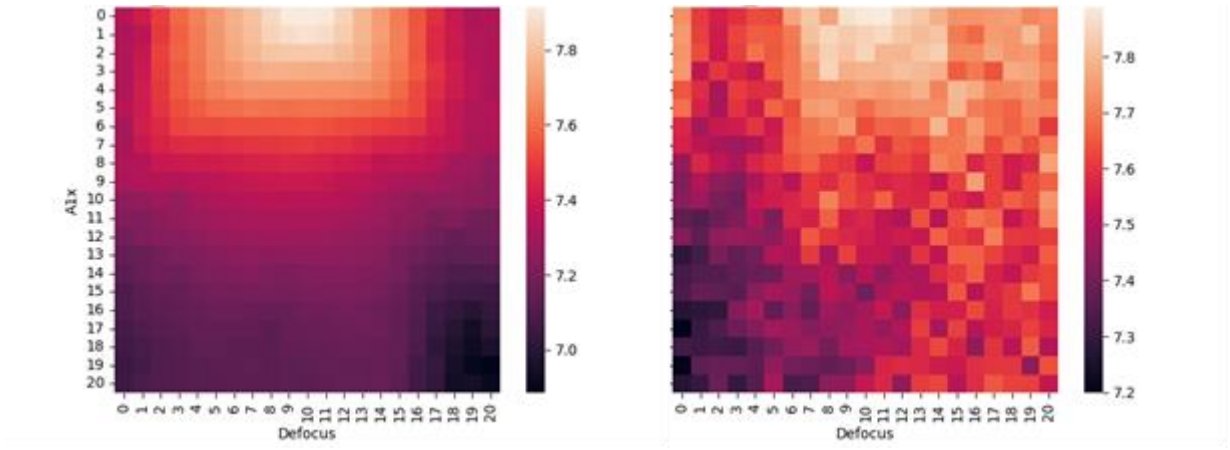

Figure 9: Consistent Q-values map in simulated data (left image) and on microscope data (right).

.

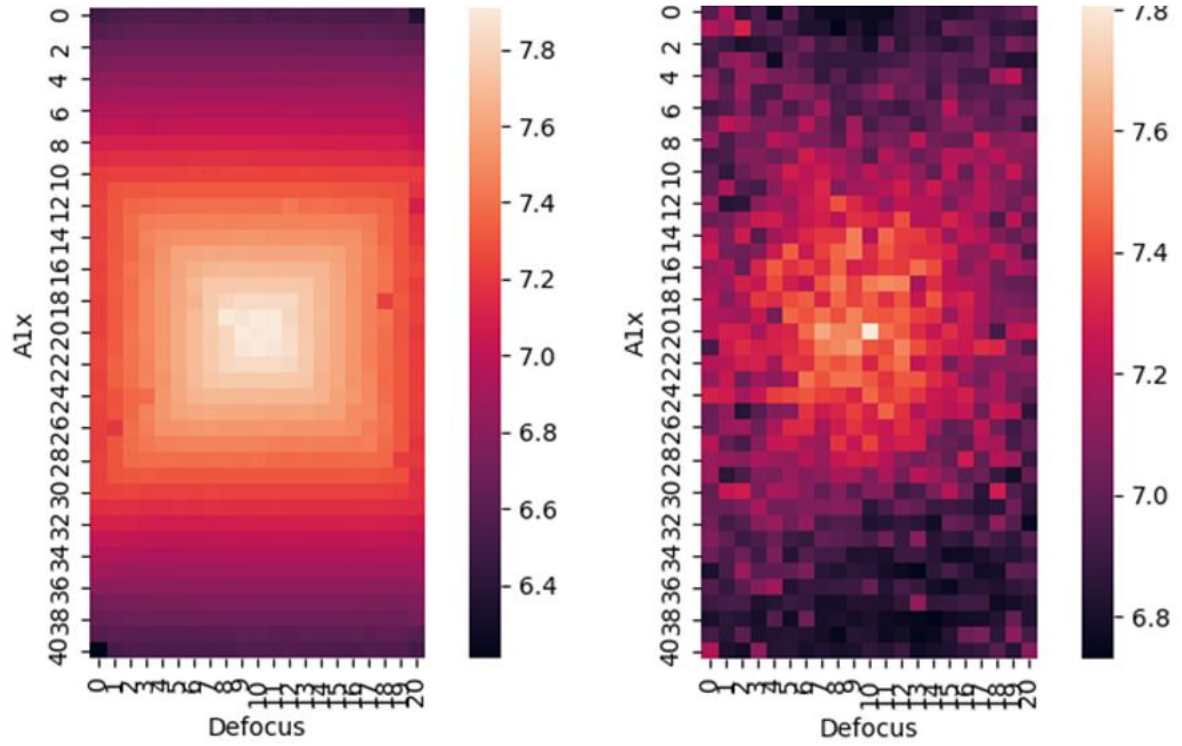

Figure 10: Consistent Q-values map in simulated data (left image) and on microscope data (right).

For the second generation (correction of C1 and A1x), of models, Q-values were in agreement between simulation and experimental data. However, the actions were showing a wrong policy. In this case we could for instance recognize diverging actions (Figure 11). From there we learnt that a state can have a certain value while the decision taken upon it can still be wrong. This can be due to many reasons. In this specific case, the actions seemed contradictory, sometimes diverging, and sometimes converging from very similar positions. We learned that the visualizations showed a symmetry problem in the state-space, thereby illustrating the the individual images are non-Markov. A solution needs to be introduced to reestablish the Markov property by including additional information; e.g. introducing multiple images per state (Figure 12).

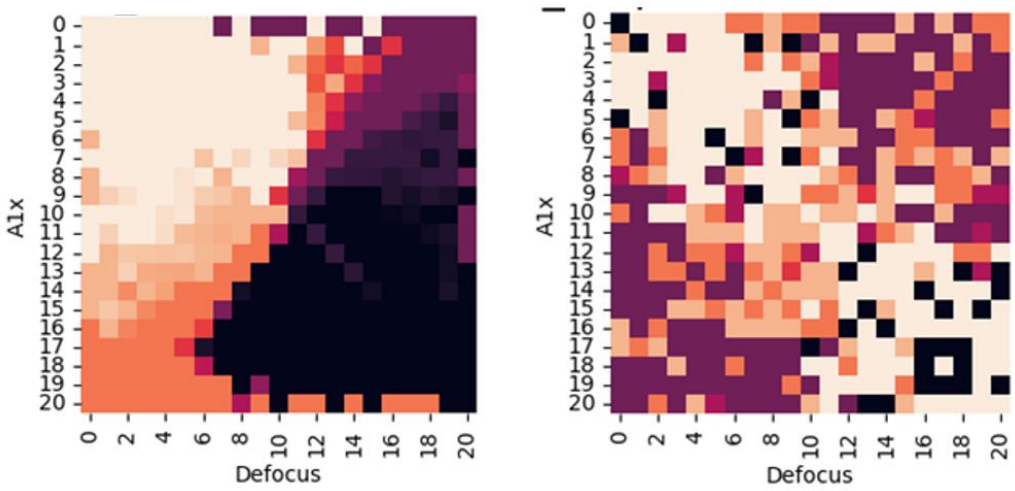

Figure 11: Inconsistent action map in simulated data (left image) and on microscope data (right).

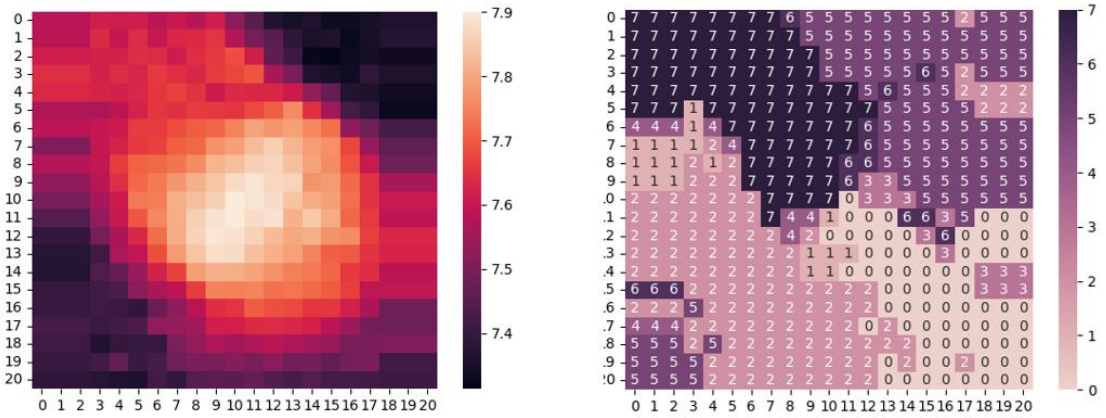

Figure 12: A consistent model having a correct Q-map (left) and a matching policy (right) in 2D

In the third and current iteration (correction of C1, A1x and A1y), a different set of problems is again encountered and analysed through the Q-values and the resulting policies. Provided that the size of the problem space increases, the RL has a much harder time to learn converging policies. Here we have seen that employing dense rewards has proven beneficial for facilitating AI training convergence.

However, real-world testing reveals that while the distance to the goal is learned, the actions tend to deviate from the optimal solution (Figure 13). Although the sum of Q-values appears accurate, the behaviour in terms of actions requires more in-depth analysis. Additionally, the lack of correlation between max(Q-values) and sum(Q-values), a behaviour observed in correct models, suggests the need for further investigation to extract accurate information (Figure 14). Below several examples illustrating Q-values

and policies are described to provide more context on the analysis that were carried out based on this information source.

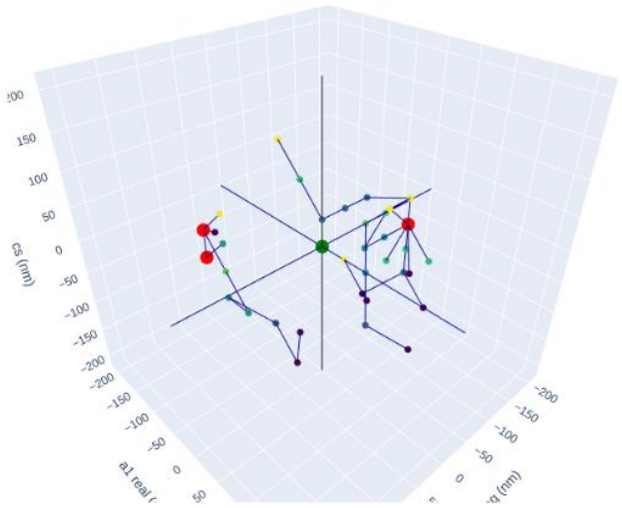

Figure 13: Illustration of diverging policies ((0,0,0) = goal, red dot = end state).

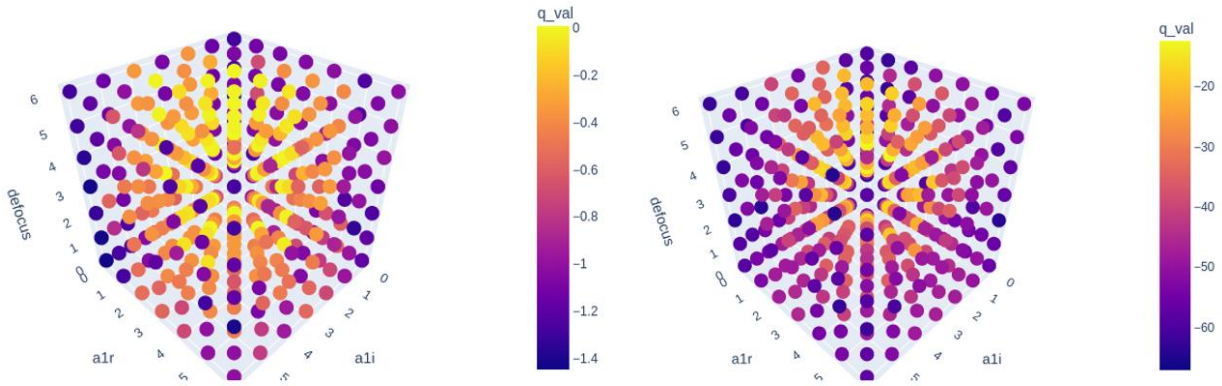

Figure 14: Maximum and Sum of Q-values not being consistent in a 3D example.

The Q-values produced by the trained models and the actions taken in the resulting policy are of high value for detecting working models for deployments and help the researcher and other profiles understand the model being used. They are as well invaluable tools for detecting problems. Q-values and policies that do not look as expected point to problems in the design, in the method and/or in the processing pipelines. The hard part is understanding what kind of problem is causing the wrong Q-maps and policies.

5. Accelerating Investigation through the use of Supervised Learning

Steps are being made to generalise the solution for full 3D, trying a variety of methods, which include supervised learning, continuous learning and in general methods oriented to splitting the problem into more manageable pieces. Extending the problem into multiple dimensions is one of the difficulties especially in RL where the curse of dimensionality is a known factor. In the current situation, dependency of the input variables is assumed, making the splitting of the problem less trivial. In supervised learning, labelling allows for more information to be present for the AI per taken step. Given the discrete and limited nature of the dataset this approach could work.

5.1 On Reinforcement and Supervised Learning

One of the problems in Reinforcement learning is that the process is both very data inefficient next to being unstable. This means that training sessions are not only long, but also plentiful in order to find good hyperparameters. This hinders rapid prototyping of ideas. To speed up this process, we have transformed our use-case in such a way that we can use classical super-vised learning. Although not perfect, this does allow us to quickly iterate over ideas and implementations and sift out future directions efficiently.

The primary starting point for this approach is the observations in the work “Reinforcement Learning Upside Down: Don’t Predict Rewards – Just Map Them to Actions”. [5] Our approach is simpler. Informally you can think of the difference between Supervised Learning (SV) and Reinforcement Learning (RL) as the following:

- In SV you task an algorithm with learning a direct association between a state and a label. The label is an instruction independent of the specific learning algorithm.
- In RL the reward is a consequence. This is the value of all the steps you just took and might take from here on. Data is not provided, that the algorithm must collect actively.

This begs the question why we would do RL in the first place. To quickly recap, RL exists for situations where:

- It is near impossible to iterate over all states. Think of all the possible states in the game of Go.
- It is very hard, if not impossible, to predefine an optimal action for a state at that time, without considering its relation to a future goal.

In basic RL settings there are too many states and an optimal policy is unknown. To reiterate:

- There is a (sparse) reward for a taken sequence of states. The reward is a consequence tied in to what the agent did as action in those states.
- Data is not available in the traditional SV sense: (image, label) or (chess position, best move). Both the data and this link must be ascertained.
- Data must be learned to be collected. Be it on or off policy.
- Agents simultaneously learn to collect what data that will lead to learned optimal actions all leading to a maximized discounted return.

Imagine an environment that provides pairs $\{x_i, y_i\}$ where $\boldsymbol{x_i}$ is the i-th state and $\boldsymbol{y_i}$ is the action which will maximize the expected return. This link resembles supervised learning and in RL you wouldn't know this association, but would have to learn a policy:

$$\pi(\boldsymbol{s}): \boldsymbol{S} \rightarrow \boldsymbol{A}$$

If we further examine the relation between $\{\boldsymbol{x_i}, \boldsymbol{y_i}\}$ in SV and $\{\boldsymbol{s}_t, \boldsymbol{a}_t\}$ in RL in terms of the discounted expected return, then for this example:

$$G_t = \sum_{k=1}^{\infty} \gamma^k R_{t+k+1},$$

and:

$$\boldsymbol{y} = \boldsymbol{\pi}^*(s_t) = argmax\, \mathbb{E}_{A\sim\boldsymbol{\pi}^*}\left[\sum \boldsymbol{G_t} \mid \boldsymbol{S}_t = \boldsymbol{s}, \boldsymbol{A}_t = \boldsymbol{a}\right],$$

where $\pi^*(s)$ is the optimal policy. But as long as you don't know this optimal policy function, you can't establish the mapping between states and actions (or labels) and thus can't reduce an RL problem to supervised learning.

However, when using the digital twin for training, there are two more conditions that are set:

- The datasets generated by both the digital twin and our real microscopes are finite in practice and easy iterable.
- We have a known ersatz optimal policy function. In this use-case we take the shortest path to goal as (fixed) policy. This is also the policy corresponding to the least amount of knob changes

on the control panel, virtual or otherwise. This policy is easily derived from the labels the digital twin provides us, or estimated from the current software stack on the microscope.

This turns our RL problem into SV. Obviously, this comes at a cost of having a fixed policy. An RL algorithm has much more freedom to discover hidden and novel solutions, to 'beat the system' in away. In return we obtain a very fast way to test (encoder) architectures and sift out methods to help bridging the domain between real and synthetic data.

For completeness, can you turn an SV problem into RL? This is possible by definition.
Let there be a set of pairs $\{x_i, y_i\}$ where $\boldsymbol{x_i}$ is the i-th state and $\boldsymbol{y_i}$ is some label and $f(\,)$ a model. A run-of-the-mill loss function would then be $\mathcal{L}(\boldsymbol{y}, \hat{\boldsymbol{y}})$, where $\hat{y} = f(\boldsymbol{x})$.

Minimizing this loss is equivalent to maximizing the expected reward, which means that it is possible to construct trajectories in the following fashion:

$$\boldsymbol{T} = \{(\boldsymbol{x}, f(\boldsymbol{x})|f(\boldsymbol{x}) - \boldsymbol{y})^0, \dots, (\boldsymbol{x}, f(\boldsymbol{x})|f(\boldsymbol{x}) - \boldsymbol{y})^n\},$$

where $\boldsymbol{x}$ is the state, $f(\boldsymbol{x})$ is the action provided by the model and $\hat{\boldsymbol{y}} - \boldsymbol{y}$ is the inverted reward. Over these trajectories you could deploy q-learning [6] [7].

5.2 Supervised learning for ASIMOV

As architecture for supervised learning for ASIMOV prototyping, a fixed dataset is collected from the digital twin, spooled through the environment to do all needed image (pre) processing and then trained supervised using either a shaping reward (distance to origin) and regression or using the optimal action in the shortest-path sense with binary cross entropy, i.e. classical classification. Shortcomings have been extensively covered in the previous section. An annotated result may be seen in Figure 15 and Figure 16. However, problems arising from the differences between real and synthetic images continue to hurt performance, as visually shown in Figure 17.

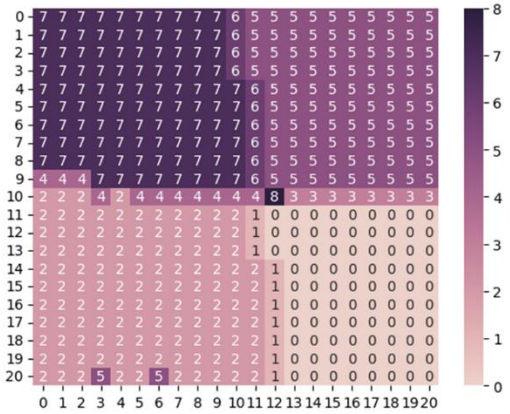

Figure 15: Annotated pseudo V map for real microscope data, where number 8 indicates that the model chose to stop.

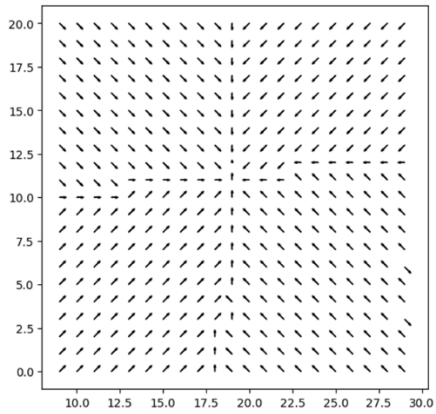

Figure 16: Quivers indicating the direction of the action taken by the model on real data.

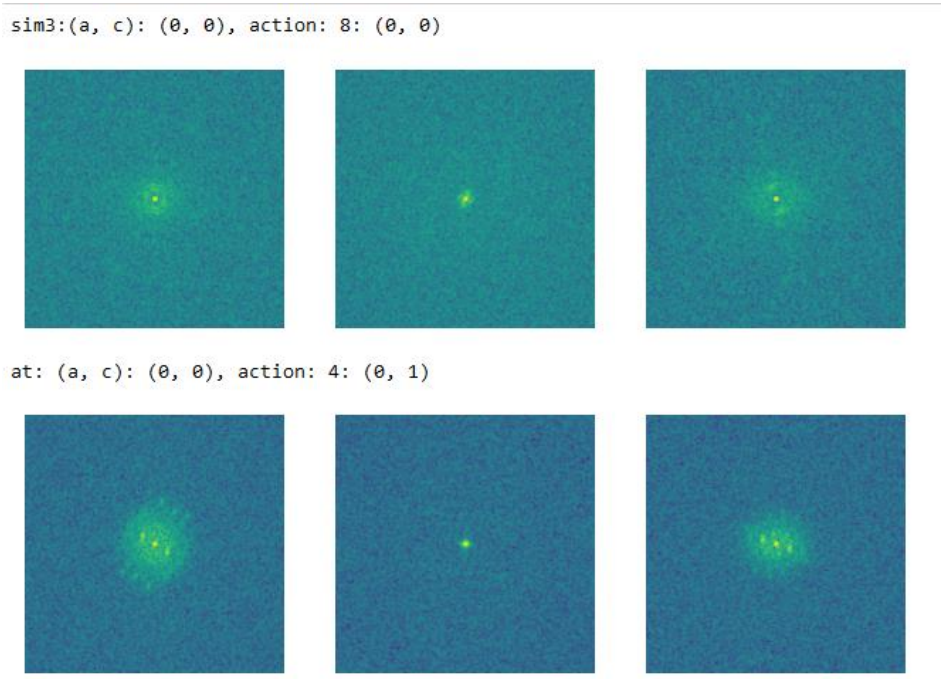

Figure 17: Above are the FFT of 3 synthetic images for the (0,0) optimal position, below is a similar (0,0) set from real data. These are not the same, and the model behaves differently for both.

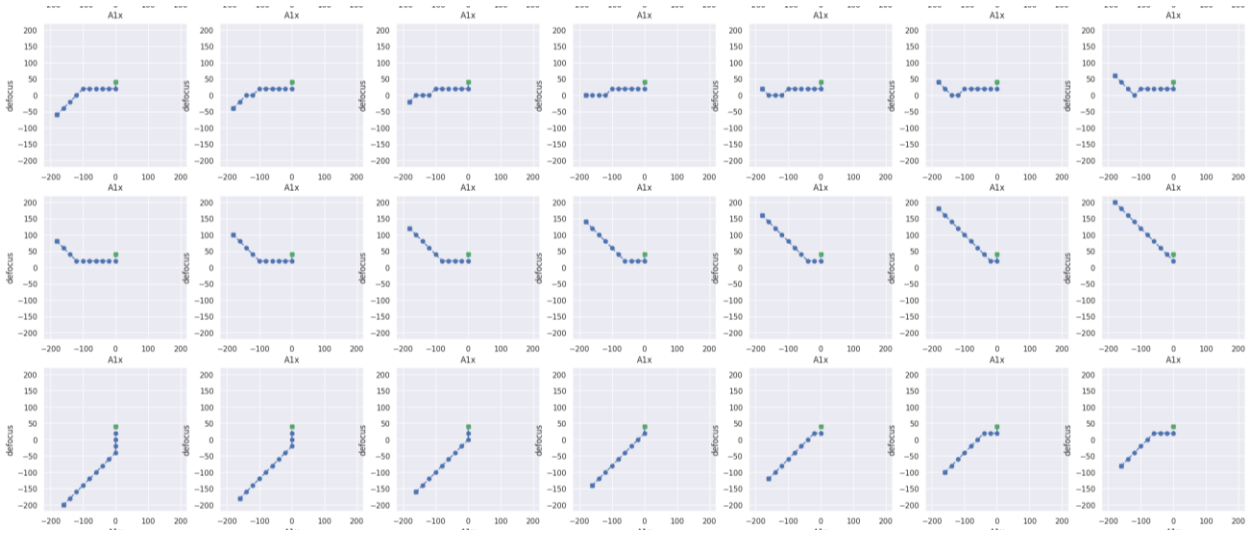

Figure 18: Example paths taking by model trained on synthetic data and deployed on real data. A green point indicated the model stopped by choice.

With training regimes running under 20 minutes, this setup does allow to do a sweep over architectures and hyper parameters. Examples of the training and validation curves can be seen in **Error! Reference source not found.** and Figure 20.

All models are trained on synthetic data and tested against real data. As training and validation sets come from the digital twin they have an identical distribution and therefor the validation curves are not informative. As is evident, there are no clear winners. By far most models end up with roughly 65% performance. This is measured as taking the optimal labelled action for that image state. Also clear is that the model overtrains extremely fast. Best performance on the test set is reached in under 10 epochs for most cases. Similar behavior is seen in the training curves on synthetic data. However, this method allows to very quickly test and iterate over ideas and prototypes. Performance is roughly identical to RL based methods. The downside is that there will never be more advanced sequence based models in this setting. Also, there is no free lunch in the sense that no single architecture or layer outperforms all.

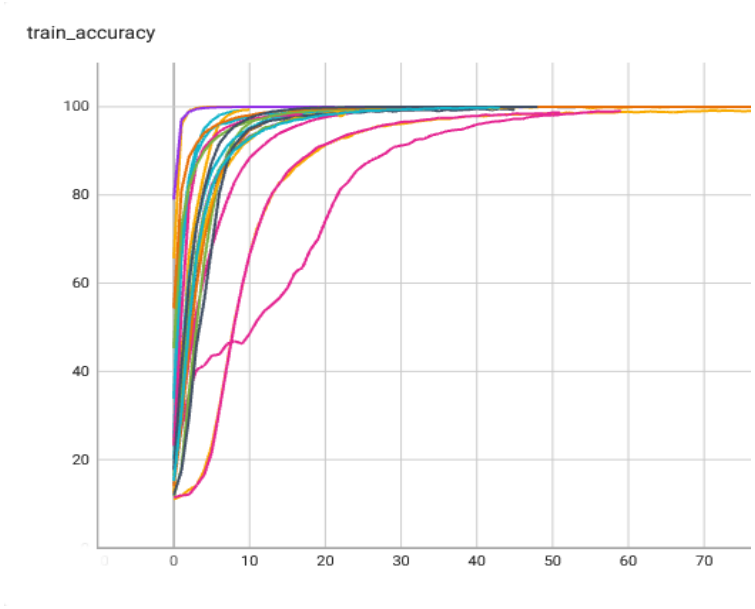

Figure 19: Training curves show all models manage to overtrain, most of them within 10-12 epochs. Only the no-FFT models take more iterations.

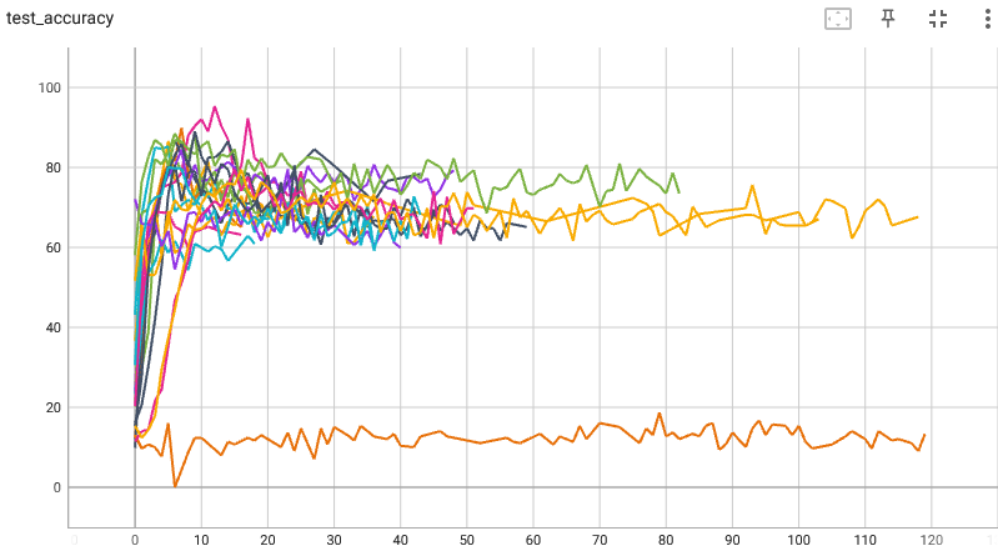

Figure 20: Real data test performance against the number of training epochs on synthetic data

6. Advancements in productization

There are system engineering challenges that must be addressed when productizing a proof-of-concept solution. We consider different phases in the development of a product: *Proof-of-Concept, Prototype, Initial Product, Mature Product*. In Proof-of-Concept development typical questions drive the development: how feasible is the basic solution? Can we achieve the required qualities (e.g., speed and accuracy)? In Prototype development the focus is on the integration of the basic solution in the product context. The challenges are in finding out how to control/manage the qualities (e.g., reproducibility, speed, response time), and in flexibility (e.g., changing the purpose of the solution or use case, ease of evolution of the future product). Product development is aiming to productize the prototype and has different type of challenges, rooted in the final goal of creating business (i.e. generating money). Topics such as manufacturing, maintainability, deployment, and footprint come to the foreground. The Scaling-to-series deals with scaling up in manufacturing and roll-out of products. The challenges are e.g., in large scale deployment of (probably) fast changing AI solutions in the field, and in business-related challenges such as customer acceptance. Note that each phase requires a significant increase in company effort (e.g., a factor of 3 to 5).

The phase of the ASIMOV work at Thermo Fisher Scientific, may currently be characterized as *Prototype.* This phase involves various analysis steps e.g., stakeholder analysis, feature specification, and mapping to the TEM reference architecture; and secondly, synthesis steps e.g., systems architecture creation, initial system design, and business case creation.

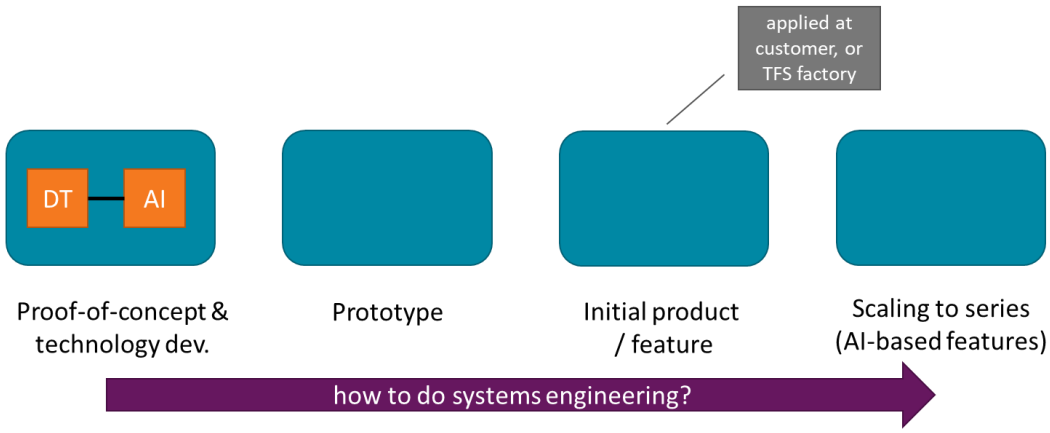

Figure 21: Simplified process of developing products containing innovative components.

A generic research question to be answered in ASIMOV is: “how is systems architecting / systems engineering changed by the involvement of digital twins and AI components?” A second research question is: “is the DT-AI combination posing extra challenges?” The output of the research could be new approaches or methods to address the specific challenges.

The TFS productization efforts will be a use case to investigate the two questions. This entails a close look at the systems architecting and systems engineering activities and identifying new or different actions, compared to regular development. The above-mentioned analytic and synthetic steps provide a framework to guide the investigation.

6.1 Systems architecture creation

The verification and validation of the ASIMOV reference architecture will provide feedback to the TFS architecture, and vice versa. This entails the mapping of the TFS solution and design decisions to the reference architecture, and comparisons with the UUV case solutions, which will generate new insights. An overlay of the TFS proof-of-concept architecture on the ASIMOV reference architecture is shown in Figure 22. It is clear that the current architecture adheres to the generic reference, although some parts are not yet covered. In later stages of development these will likely be added.

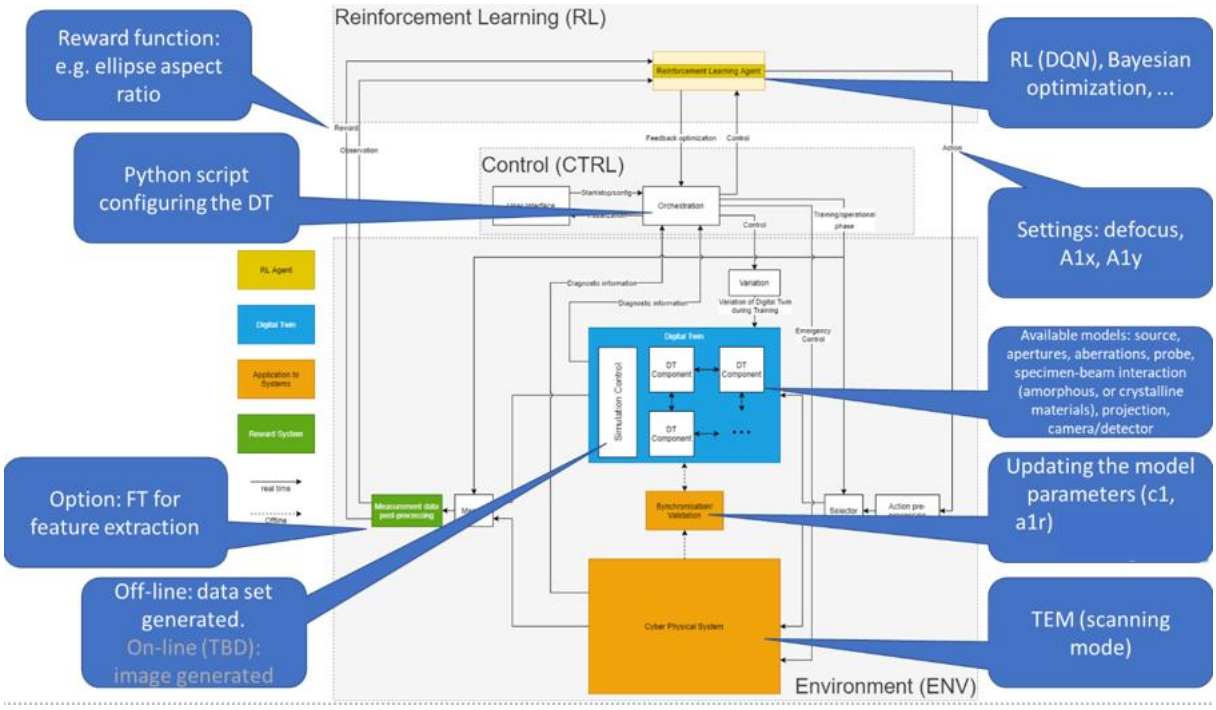

Figure 22: The ASIMOV reference architecture [8] overlayed with information about the current TFS architecture.

6.2 Existing standards

External frameworks, such as the 'Platform Stack Architectural Framework' recently published by the Digital Twin Consortium may serve as guideline for organizing and prioritizing the systems architecting activities. The structure shown in Figure 23 is a basic setup for generic digital twins, combining various viewpoints in one diagram. The framework provides further decomposition of the required capabilities and best practices and guidelines towards implementation.

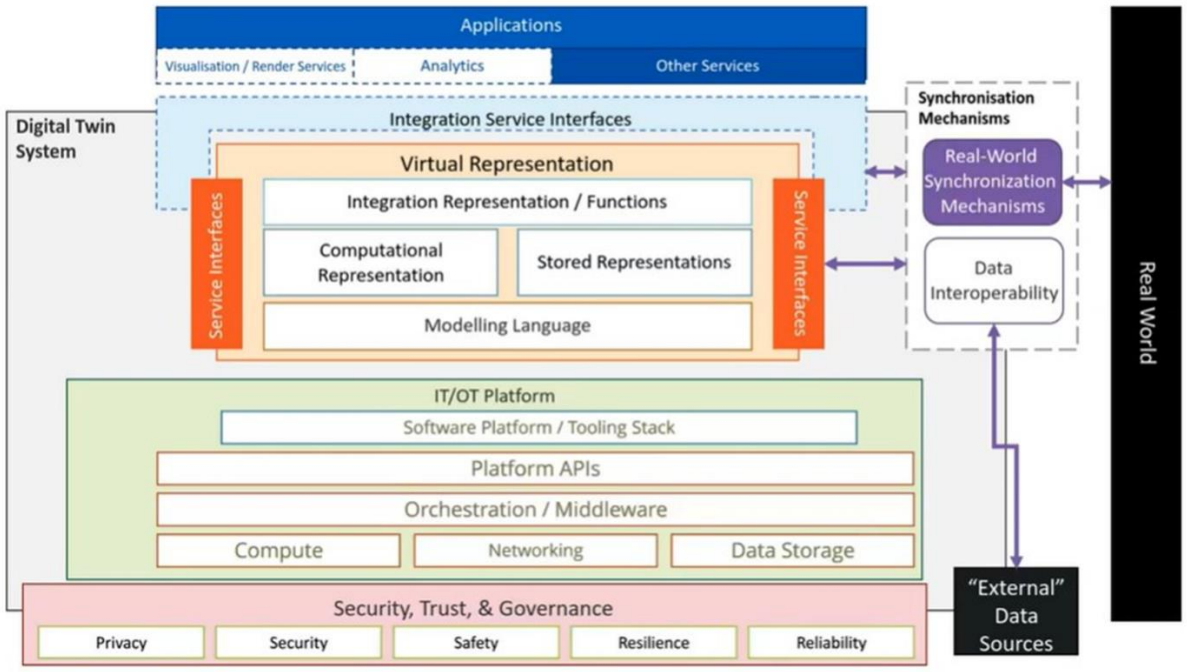

Figure 23: The Platform Stack Architectural Framework created by the Digital Twin Consortium [9].

Another framework that may serve as a guideline is the AI/ML workflow published by the AI Infrastructure Alliance, see Figure 24. This workflow shows the key steps involved in a typical AI/ML workflow, and how the required capabilities link together [10].

We can map the (S)TEM use case to this framework in the following way, where the relevant framework blocks are mentioned between square brackets in the text. In the (S)TEM use case, input data are either images generated by the digital twin [Synthetic Data Generation] or experimental data from the microscope stored on the server [Network FS]. There is no cleaning or validation of data needed, but a transformation step to get experimental data to the right format. The [transformation] ensures normalization of the sizes and resolution of the images. [Labels] are created by the applications, and not afterwards in the data pipeline. In the (S)TEM use case, features are not used. During the [training stage], [experiment] and [train] are applicable. There is no [tuning], as there is no network used where a head needs to be tuned. During training, [metadata storage] relates to storing hyperparameters, training parameters, and versioning of evaluation data. [Logging] is used for training and evaluation, based for example of figures like visualizing Q-map values, but also on metrics like the accuracy. [Deployment] is first on offline acquired experimental data and if the results there are good, also on the real microscope.

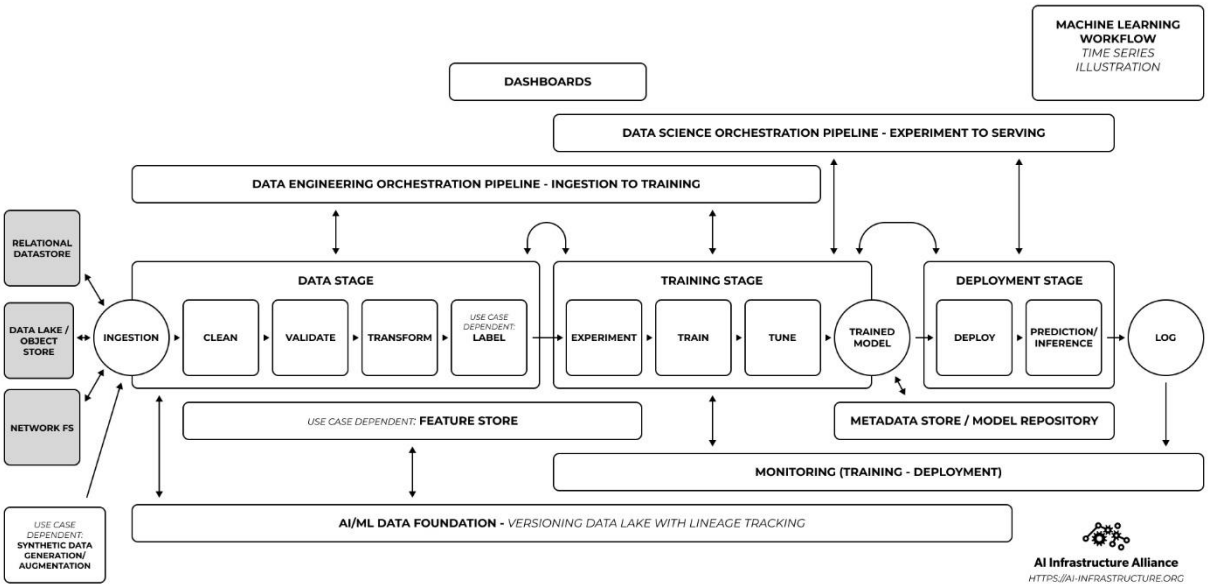

Figure 24: Blueprint of an AI/ML workflow created by the AI Infrastructure Alliance [10].

7. Terms, Abbreviations and Definitions

Table 1 - Terms, Abbreviations and Definitions

ABBREVIATION	EXPLANATION
AI	Artificial Intelligence
DT	Digital Twin
EM	Electron Microscopy
RL	Reinforcement Learning
SV	Supervised Learning
STEM	Scanning Transmission Electron Microscopy
TEM	Transmission Electron Microscopy
HAADF	High Angle Annular Dark Field
C1	Defocus
A1	Two-fold Astigmatism
A1x	X component of two-fold astigmatism
aC	Amorphous carbon
FEG	Field Emission Gun
CCD	Charge-Coupled Device
NN	Neural Network
BO	Bayesian Optimization
GP	Gaussian Processes
UUV	Unmanned Utility Vehicle

8. Bibliography

[1] H. Vanrompay, Towards Fast and Dose Efficient Electron Tomography, Antwerp: University of Antwerp, 2020.

[2] O. Scherzer, *Zeitschrift für Phys,* vol. 593, p. 101, 1936.

[3] I. Lazić and E. G. T. Bosch, "Analytical review of direct STEM imaging techniques for thin samples," *Advances in Imaging and Electron Physics,* vol. 199, pp. 75-184, 2017.

[4] E. J. Kirkland, "Advanced computing in electron microscopy," *Springer Nature,* 2010.

[5] J. Schmidhuber., "Reinforcement Learning Upside Down: Don't Predict Rewards – Just Map Them to Actions.," *arXiv: 1912.02875 ,* 2020.

[6] A. G. B. a. T. G. Dietterich., "Reinforcement learning and its relationship to supervised learning". In: Handbook of learning and approximate dynamic programming 10, 2004.

[7] "Learning to predict by the methods of temporal differences.," *Machine learning 3 ,* pp. 9-14, 1988.

[8] ASIMOV-consortium, "ASIMOV Reference Architecture," 2022.

[9] D. McKee, "Platform Stack Architectural Framework:," 2023.

[10] AI Infrastructure Alliance, "Why We Started the AIIA and What It Means for the Rapid Evolution of the Canonical Stack of Machine Learning," [Online]. Available: https://ai-infrastructure.org/why-we-started-the-aiia-and-what-it-means-for-the-rapid-evolution-of-the-canonical-stack-of-machine-learning/. [Accessed August 2023].

[11] T. Haarnoja, A. Zhou, P. Abbeel and S. Levine, "Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor," in *Proceedings of the 35th International Conference on Machine Learning*, Stockholm, 2018.

[12] M. G. Kapteyn, J. V. Pretorius and K. E. Willcox, "A probabilistic graphical model foundation for enabling predictive digital twins at scale," *Nature Computational Science,* vol. 1, pp. 337-347, 2021.

[13] D. Jones, M. Schonlau and W. Welch, "Efficient Global Optimization of Expensive Black-Box Functions," *Journal of Global Optimization,* vol. 13, no. 4, p. 455–492, 1998.

[14] E. Brochu, M. Hoffman and N. de Freitas, "Portfolio Allocation for Bayesian Optimization," in *Proceedings of the 27th Conference on Uncertainty in Artificial Intelligence (UAI)*, 2011.

[15] T. McDermott, D. DeLaurentis, P. Beling, M. Blackburn and M. Bone, "AI4SE and SE4AI: A Research Roadmap," *INCOSE Insight,* vol. 23, no. 1, pp. 8-14, March 2020.

[16] "https://open.oregonstate.education/generalmicrobiology/chapter/microscopes/," [Online]. [Accessed 23 11 2022].

[17] FEI, "FEI Application Instructions: How to tune STEM probe on Cs-corrected Titan.".

[18] H. N. a. H. La, “Review of Deep Reinforcement Learning for Robot Manipulation,” in *Third IEEE International Conference on Robotic Computing (IRC)*, Naples, Italy, 2019.

[19] “https://lilianweng.github.io/posts/2018-02-19-rl-overview/,” [Online]. [Accessed 11 23 2022].

[20] ASIMOV consortium, “itea.org,” 23 07 2022. [Online]. Available: https://itea4.org/community/project/workpackage/documents/5119.html. [Accessed 23 11 2024].