Master Thesis

Vision Transformers for Accurate Overlay Metrology

by

Lars de Wolf
(lwo103)

First Supervisor: Asst. Prof. Lyuba Amitonova
Daily Supervisor: Maximilian Lipp
Second Reader: Asst. Prof. Michael Cochez

July 18, 2025

Submitted in partial fulfillment of the requirements for
the VU degree of Master of Science in Artificial Intelligence

Table of Contents

1 Introduction 3
- 1.1 Contribution 5

2 Background 6
- 2.1 Diffraction-Based Overlay 6
- 2.2 Dark-Field Digital Holographic Microscopy 6
- 2.3 Wavefront Aberrations and Coherent Imaging Effects . . 7

3 Related Work 8
- 3.1 DHM Aberration Correction and Noise Reduction 8
- 3.2 Deep Learning for DHM 9

4 Methodology 10
- 4.1 Multi-Layer Perceptron 10
- 4.2 Convolutional Neural Network 11
- 4.3 Vision Transformer 12

5 Experiments & Results 13
- 5.1 Setup 13
- 5.2 Full Dataset Evaluation 16
- 5.3 Limited Dataset Evaluation 19

6 Conclusion 20

7 Discussion 20

A Hyperparameters 24

Vision Transformers for Accurate Overlay Metrology

Lars de Wolf

Vrije Universiteit Amsterdam, Amsterdam
l.de.wolf@student.vu.nl

Abstract. Performing fast and accurate metrology parameter extraction is critical for semiconductor manufacturing and has a direct impact on the yield of the final product. Dark-field Digital Holographic Microscopy (df-DHM) offers a promising method that allows for the extraction of such parameters, but it's effectiveness is often hindered by optical aberrations and coherent imaging effects. This thesis explores data-driven approaches which directly infer a metrology parameter of interest from df-DHM measurements affected by aberrations, without the need for any phase measurements. In particular, we investigate the application of Vision Transformers (ViTs) and compare the its performance to other well established architectures such as Convolutional Neural Networks (CNNs) and Multilayer Perceptrons (MLPs). We utilize simulated df-DHM datasets that incorporate a variety of aberrations and coherent imaging effects, and perform extensive experiments to effectively compare the performance between the models in terms of accuracy, robustness to aberrations, and data efficiency. We report that our models are capable of making fast and accurate metrology measurements from df-DHM images with fixed aberrations, offering similar performance df-DHM images that feature no aberrations. Furthermore, we find that ViTs consistently achieve low prediction error, and excel under limited data regimes compared to our baselines. These findings highlight the potential ViTs for robust and scalable optical metrology, especially in real-world semiconductor pipelines where obtaining large, high-quality labeled datasets can be time consuming and expensive.

Keywords: Vision Transformer · Diffraction-Based Overlay · Digital Holographic Microscopy

1 Introduction

Semiconductor devices are well-established in our society and are the driving force behind almost all technology, ranging from consumer electronics to complex systems. Modern micro-chips offer superior computational performance and are able to perform trillions of calculations per second, enabling real-time processing in applications such as self-driving cars and medical imaging. The computational performance gains of semiconductor devices have been driven in large part by Moore's law [22], which predicts that the number of transistors in one

chip approximately doubles every two years. This projection has continued to guide semiconductor manufacturers for decades to produce micro-devices with increasingly smaller dimensions, enabling greater functionality, higher processing speeds, and improved energy efficiency, while maintaining a minimal increase in production cost. As a result, modern semiconductor manufacturers are able to cram billions of transistors on a single silicon wafer, which are all distributed over dozens of layers.

Due to this decrease in feature size, the required precision to align each of these layers has increased significantly. Measuring this alignment between layers, also known as overlay (OV), is essential to reduce fabrication errors, which impact the yield of the final product. Traditionally, OV has been determined using Image-Based Overlay (IBO). In IBO, an optical microscope produces a high-resolution image of IBO metrology targets that contain top and bottom edges. Then OV is measured by comparing the relative distance between edges of both layers. However, edges on these targets have to be resolvable, which becomes harder with the continued down scaling of microchips. In fact, visible light sources are already unable to resolve structures smaller than 190 nanometer, resulting in blurred images which do not allow any individual edge detection. Light sources with shorter wavelengths allows for the imaging of smaller structures. However, this also introduces challenges, as materials such as glass and air increasingly absorb light at shorter wavelengths, and in turn require more complex optical setups. Therefore, new methods to measure the overlay error are essential to meet the sub-nanometer accuracy and precision requirements of modern chips.

Diffraction-based overlay (DBO) is one of these methods, and has become the dominant method in the industry to measure overlay error. Unlike IBO, DBO does not rely on imaging individual lines but instead leverages overlay information encoded in the intensity differences between the $+1^{st}$ and -1^{st} diffraction orders of DBO metrology marks, which contain a pair of overlapping gratings. This allows for accurate overlay measurements when using visible light without the need to individually resolve structures, making DBO an appropriate candidate for measuring overlay on targets with features well below the diffraction-limit. Additionally, the transition to intensity based overlay measurements allows the use of smaller grating pitches and reduces sensitivity to optical aberrations [5,18], further improving the overlay accuracy. Measuring the $+1^{st}$ and -1^{st} diffraction orders can be done using a dark-field Digital Holographic Microscope (df-DHM)[30], which captures the full complex field of a sample. By leveraging both amplitude and phase of a sample, it becomes possible to computationally correct for aberrations and coherent imaging effects, which occur due to imperfections in the optical setup and the coherent nature of the light source used in df-DHM. An essential step, because aberrations and coherent imaging effects can significantly distort intensity measurements, thereby leading to imprecise overlay measurements derived from these intensities. Several methods address this issue by first correcting the aberrated phase maps and then reconstructing the intensity maps using the corrected phase maps. Despite the success of these efforts, most methods only address certain wavefront aberrations and/or coherent

noise effects and may introduce additional imperfections, especially when combined. Moreover, these methods increase the computational complexity during the metrology process, which can be an expensive bottleneck due to the numerous measurements that need to be performed for each waver during the process. To address the need for an end-to-end method capable of performing accurate overlay measurements, we explore data-driven solutions which leverage deep neural networks. More specifically, we propose to leave out any computationally expensive phase aberration corrections and instead directly model the relationship between aberrated df-DHM images and overlay measurements. By relying exclusively on intensity measurements during training and inference, we reduce complexity and eliminate the need for phase measurements, which enables the use of less complex optical systems to obtain intensity measurements. We focus on the application of Vision Transformers (ViTs) and compare their performance to traditional computer vision architectures such as Convolutional Neural Networks (CNNs) and Multi-Layer Perceptrons (MLPs). Unlike these conventional architectures, ViT treats images as sequences of patches and applies self-attention to capture long-range dependencies between them. As a result, ViTs incorporate fewer inductive biases and are able to model context more effectively. We hypothesize that this gives may give them a performance advantage over CNNs and MLPs. In our experiments, we utilize several simulated datasets and assess the influence of the DBO target size, wavefront errors, model scale, and data hungriness. Due to the scarcity of experimentally obtained data, we perform experiments on datasets originating from a df-DHM simulator. Furthermore, we aim to answer the following research questions:

- RQ1: How do model architecture and size affect performance?
- RQ2: How does the magnitude of wavefront aberration impact performance?
- RQ3: How does the DBO target size influence the model's predictive capabilities.
- RQ4: How does the size of the training dataset affect model performance?

1.1 Contribution

In this thesis, we address the task of inferring overlay errors from (aberrated) df-DHM intensity measurements by training and evaluating several deep learning architectures, focusing on the performance of the Vision Transformer. We summarize our contributions as follows:

- We propose a novel method to directly infer overlay errors without using phase measurements for aberration calibration/correction.
- We assess the performance when varying the model architecture, model size, wavefront aberration types, and DBO target sizes.
- We analyze how performance changes by limiting the dataset size.

2 Background

2.1 Diffraction-Based Overlay

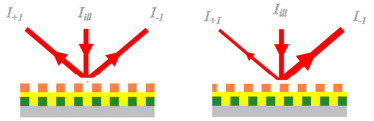


Fig. 1: The symmetric (left) and asymmetric (right) scattering of light on a DBO mark. Source:[1]

In Diffraction-Based-Overlay, layer displacement is derived by observing how light scatters from a sample when it is illuminated. The sample, which is a DBO mark that consists of stacked overlapping gratings, forms a symmetric diffraction pattern around the 0^{th} order when the gratings perfectly align, i.e. have zero overlay. However, when the gratings do not form a symmetric composite grating, i.e. have non-zero overlay, the symmetric scattering properties of the sample breaks, resulting in measurable intensity differences between the $+1^{\text{st}}$ (I_{+1}) and -1^{st} (I_{-1}) diffraction orders, shown in Fig. 1. For small overlay ranges, the intensity differences between the diffraction orders scales linearly with OV and is approximately equal to the product of the overlay error with K(λ) [1]:

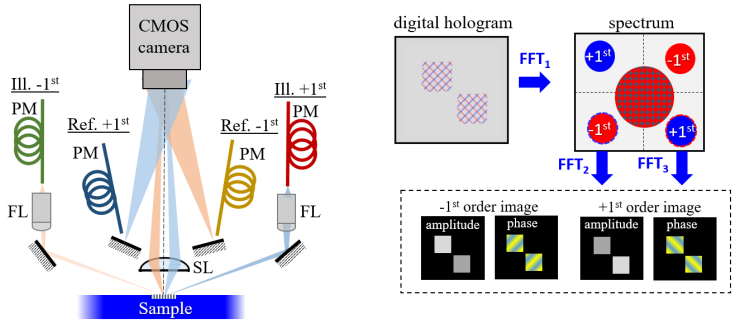
$$\Delta I = I_{+1} - I_{-1} \approx K(\lambda) \times \text{OV} \tag{1}$$

where K(λ) is a unknown grating parameter which depends on the geometry, materials, and selected wavelength used for the DBO mark. Removing this grating parameter from Eq. 1 can be achieved by a careful selection of wavelengths w.r.t. thickness of the waver, which cancel out K(λ) but may result in unstable overlay measurements. Alternatively, two pairs of DBO marks with known overlay shifts +d and -d can be utilized to extract the overlay parameter [1,18]. By measuring the intensity differences $\Delta I_{+\text{d}}$ and $\Delta I_{-\text{d}}$ for the +d and -d DBO marks respectively, the grating parameter cancels out and overlay can be determined as follows:

$$\text{OV} = \text{d}\frac{\Delta I_{+\text{d}} + \Delta I_{-\text{d}}}{\Delta I_{+\text{d}} - \Delta I_{-\text{d}}} \tag{2}$$

2.2 Dark-Field Digital Holographic Microscopy

The intensity measurements for both diffraction orders can be obtained by using a dark-field Digital Holographic Microscope (df-DHM) as demonstrated in [18]. This off-axis df-DHM setup creates a single multiplexed hologram of both the $+1^{\text{st}}$ and -1^{st} diffraction orders in parallel and introduces several other improvements over conventional imaging techniques. Df-DHM constructs the hologram by leveraging two off-axis illumination beams (one for each diffraction order) originating from the same coherent source, which constructively/destructively interfere with the light of their corresponding reference beams and result in two interference patterns on the image plane. By carefully choosing the angle of the reference beams, the complex object waves for both diffraction orders become separated in k-space. This allows to retrieve both complex fields by applying the



(a) Sensor head (b) Holographic Reconstruction process

Fig. 2: Df-DHM setup on the left and hologram post-processing steps on the right. Source: [17]

Fourier transformation to the hologram, selecting the separated sidebands and shifting them to the center, and finally applying the inverse Fourier Transformation to both sidebands to retrieve the complex object fields. The full df-DHM setup and hologram processing steps are shown in Fig. 2. Since both holograms are acquired in parallel, df-DHM benefits from reduced intensity noise and faster acquisition times compared to sequential hologram acquisition. Furthermore, the full numerical aperture (NA) of the image sensor can be used to image both diffraction orders, enhancing the spatial resolution. Finally, the retrieved complex image allows for post-processing steps to reduce aberrations and coherent imaging effects, as demonstrated in [17].

2.3 Wavefront Aberrations and Coherent Imaging Effects

Optical aberrations can severely distort the wavefront measured by df-DHM, which can result in highly inaccurate OV measurements [19]. These aberrations originate from various sources, such as imperfections in the optical setup, unstable measurement conditions, and post-processing steps. For instance, imperfections in the curvature of the lens can cause incoming light rays to focus at different spots after passing through the lens, reducing clarity and resolution. This is known as a spherical aberration, and can also occur when a different wavelength is used than the one the lens was designed for [19]. Moreover, lens imperfections can also result in astigmatism, which happens when perpendicular light rays have different foci. This can cause the image to become blurred or stretched along a particular axis. Coma is another aberration which arises when light rays enter the lens at a certain angle and result in a comet-like blur on the image plane. This often happens when lenses are misaligned or when off-axis components are integrated into the optical setup. Other commonly encountered aberrations include defocus, distortion, and field curvature. Fortunately, wavefront aberrations can be effectively described using a sequence of polynomials

known as the Zernike polynomials[1]. Each mode of the Zernike polynomials describes a specific type of optical aberration, and the coefficient determines the magnitude of the aberration.

In addition to wavefront aberrations, df-DHM also suffers from imaging effects caused by the coherent light source, such as speckle noise, Gibbs ringing, and optical crosstalk. Speckle noise occurs due to non-reflecting objects present in the optical setup (e.g. dust), causing visible spots on the image plane and reducing the contrast. Gibbs ringing can be observed during measurements as oscillations that appear near the border of metrology targets. These ringing effects are a result of the sharp cut-off made by the Fourier Transform, which blocks frequencies at the sampling border of the angular spectrum. Moreover, Gibbs-ringing propagates in every direction and interacts with other nearby structures, resulting in additional optical crosstalk.

3 Related Work

To our knowledge, directly inferring the overlay from aberrated DHM intensity measurements is novel, and therefore, limited prior work addressing this approach is available. In the following section, we first cover conventional methods to improve overlay measurements. These methods enhance the imaging quality by correcting aberrations caused by the optical setup, as well as imaging effects caused by the coherent illumination source used in DHM. We will then discuss research that incorporates deep learning to correct for optical aberrations and coherent noise.

3.1 DHM Aberration Correction and Noise Reduction

Various methods have been proposed to compensate for aberrations present in DHM measurements and mainly rely on optical setup optimizations and/or numerical algorithms. For instance, spherical aberrations has been compensated in the past with the use of optical modifications such as telecentric arrangements [29,31], the addition of a second MO in the reference arm [15], and a second adjustable lens as a condenser lens [26]. Despite their effectiveness, these methods often introduce additional complexity to the optical setup, require precise calibration, or limit the flexibility of the imaging system. Alternatively, numerical methods leverage the complex field of the holograms to computationally correct for aberrations without the need for hardware modifications. Most numerical methods achieve this using reference conjugated holograms [4], surface fitting methods [20,21,24], double exposure methods [8], and phase masks [3,8]. Despite the ability of these methods to compensate for a wider range of aberrations, they impose extra computation and unreliable behavior when applied to highly aberrated holograms.

Noise caused by the coherent illumination source can effectively be reduced using a variety of methods. Optical modifications often reduce the coherence

[1] https://en.wikipedia.org/wiki/Zernike_polynomials

of the illumination source by using low spatial coherent sources [2,9,17]. On the downside, lowering coherence may decrease focal depth and lead to incremental difficulty in adjusting the light configuration. On the other hand, averaging multiple holograms obtained through various optical methods (also known as hologram multiplexing) has also proven to be beneficial [7,28,33] and is well-researched. Unfortunately, taking multiple measurements under different conditions is time-consuming and requires intricate setups. Additionally, several image processing techniques have been proposed to numerically compensate for coherent noise effects such as speckle, which can be compensated by using mean/median filtering [10], filtering the Fourier plane by shifting a band-pass filter [16], or utilizing window functions [17,23]. However, these methods result in a reduction of spatial resolution due to information loss.

3.2 Deep Learning for DHM

Several works leverage deep neural networks to correct wavefront aberrations in DHM measurements [11,13,14,25,27]. Most methods achieve this by estimating the coefficients of the Zernike polynomials, which are then used to subtract the aberrations from the measured wavefront. For example, [25] used a U-Net CNN architecture to predict background regions in aberrated phase maps, from which the Zernike coefficients are derived and used to compensate the aberrated wavefront error in the frequency domain. Similarly, [11] also uses a CNN to predict Zernike coefficients, but performs the coefficient extraction and aberration compensation before phase unwrapping. Even though both methods report impressive results, their reliance on Zernike polynomial fitting and intermediate processing steps increases computational complexity. A different method is presented in [27], where an end-to-end deep learning framework called HRNet is used to reconstruct amplitude, phase, and two-sectional objects. They do not rely on background separation or Zernike polynomial fitting, but use a ResNet-like architecture to reconstruct noise-free images from raw inputs. Moreover, [13] also corrects for optical aberrations, but uses a ResNet50 module to first identify the different types of aberrations present in the phase map, along with their aberration coefficients. These are then fed into a U-Net module that constructs the unaberrated phase image. Coherent noise effects, for example speckle noise, has also been successfully suppressed in several works using deep learning, where most methods use a CNN architecture to reconstruct clean phase images from noisy DHM phase measurements. In [12], a U-Net architecture with residual connections is trained to reduce speckle noise, using noisy and noise-free DHM phase image pairs. They evaluate their method on several datasets by using gaussian noise to obtain the noisy and noise-free phase image pairs. The work of [35] also uses a CNN to suppress speckle noise from noisy and noise-free image pairs, but applies their model on the wrapped phase and evaluate on both simulations and experimental data. Self-supervised learning has also been applied to coherent noise reduction in DHM, and has the advantage that no noise-free image has to be provided during training. The work of [36] outlines an algorithm to train a denoising model by maximizing the maximum likelihood estimate of pairs of

images with random noise distributions. They also use a U-Net backbone and report good performance on both simulated and experimental datasets.

4 Methodology

This section outlines the methodology used to conduct our experiments. We start by introducing the Multi-Layer perceptron and the Convolutional Neural Network, which serve as baselines for our experiments. We will use their performance to evaluate the strengths and weaknesses of the Vision Transformer, which we cover next. Models are implemented in Python using the PyTorch library.

4.1 Multi-Layer Perceptron

The Multi-Layer Perceptron is a neural network composed of multiple layers of perceptrons. Each perceptron applies a linear transformation to the values from the previous layer, followed by a non-linear activation function. Each layer is fully connected to the preceding layer through a weight matrix $W^{(l)}$, which determines the weighted combination of neuron values from the previous layer. The neuron activations at layer l is shown in Eq. 3:

$$\mathbf{z}^{(l)} = f(\mathbf{A}^{(l-1)}W^{(l)} + \mathbf{b}^{(l)}) \tag{3}$$

where:

- $\mathbf{z}^{(l)} \in \mathcal{R}^{N \times D}$ is the activation matrix at layer l
- $W^{(l)} \in \mathcal{R}^{P \times D}$ is the weight matrix at layer l
- $\mathbf{A}^{(l-1)} \in \mathcal{R}^{N \times P}$ is the activations from layer $l-1$
- $\mathbf{b}^{(l)} \in \mathcal{R}^{D}$ is the bias vector at layer l
- $f(\cdot)$ is a non-linear activation function

At the first hidden layer, the activations of the previous layer are initialized as the input features, i.e. $\mathbf{A}^{(0)} = X$. The activation function at the output layer is task-dependent, and is in most cases the identity function or the Softmax function for regression tasks and classification tasks respectively.

Implementation. We construct a simple MLP that consists of 3 hidden layers, with ReLU activations applied after each hidden layer in the network. Before our first linear layer, we convert our input to a column vector, by flattening along width, height, and channel dimensions. This column vector is then projected to the specified hidden dimension by the first hidden layer, continued by a projection that maintains the dimensionality of the input, and then followed by a final projection to a single scalar value.

4.2 Convolutional Neural Network

Convolutional Neural Networks extract features by applying convolutions over (multi-dimensional) data using learnable kernels. These kernels perform local linear transformations over sliding windows of the input data. Each kernel is a set of weights that is shared across all spatial locations of the previous layer, making CNNs more parameter-efficient compared to MLPS. In addition to activation functions, other components such as pooling layers, fully connected layers, and normalization layers are often part of the CNN architecture. Padding is also frequently used, which ensures that information near the boundaries is retained after the convolution. Stride controls the step size of the sliding window, and can be set equal to the kernel size to obtain non-overlapping receptive fields. If we assume a simple convolution that features no padding and a stride of one, we can define the convolution of a two-dimensional kernel K over a two-dimensional input I by using the cross-correlation operator $\star$, as shown in Eq. 4:

$$(\mathrm{I} \star \mathrm{K})[i,j] = \sum_{\mathrm{m=0}}^{\mathrm{M\text{-}1}} \sum_{\mathrm{n=0}}^{\mathrm{N\text{-}1}} \mathrm{I}[i+m, j+n]\mathrm{K}[m,n] \tag{4}$$

where:

- $I \in \mathcal{R}^{H \times W}$ is a two-dimensional input.
- $K \in \mathcal{R}^{M \times N}$ is a two-dimensional kernel.
- $(I \star K) \in \mathcal{R}^{(H-M+1)\times(W-N+1)}$ is the output feature map.

Implementation. Our CNN features a stack of convolutional blocks, which halve the spatial resolutions of the input and produce $c_i \in [32, 64, 128, 256, 512]$ feature maps after each block i. Blocks apply a two-dimensional convolution, a ReLU activation function, and a two-dimensional max-pooling operation. Convolutions use a kernel size of 3, stride of 1, and add 1 padding token near the borders of the inputs to ensure that spatial dimensionalities are retained after

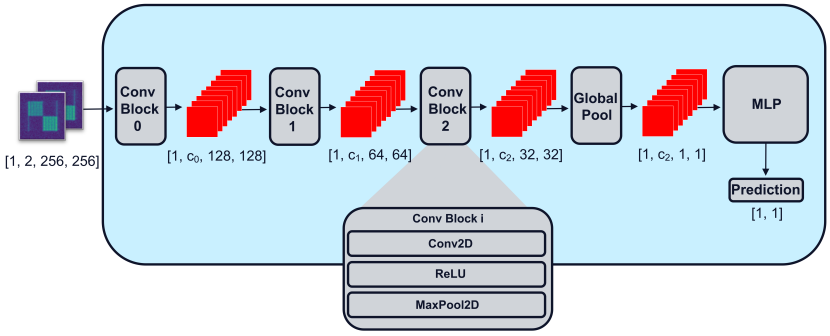


Fig. 3: Architecture of a CNN with 3 convolutional blocks.

the operation. The max pooling operation uses a kernel size of 2 which halve the spatial resolutions. After the convolutional blocks, we apply average pooling over the spatial resolutions and use a MLP regression head which first doubles the amount of feature maps before projection to a single value, using a ReLU activation in between the layers. The CNN architecture is shown in Fig. 3.

4.3 Vision Transformer

Vision Transformer is an adaptation specifically designed for Computer Vision (CV) tasks build upon the Transformer [32] architecture introduced in 2017. While the original Transformer was developed for textual data and functions by handling sequences of tokens, Vision Transformers extends this concept by treating images as a sequence of flattened patches. These patches are linearly embedded to D dimensional vectors and incorporated with learned position embeddings, to serve as input to a conventional Transformer encoder. By treating images as sequences of patches, the number of required deviations from the original work was kept at a minimum.

The Transformer encoder consists of multiple layers, which contain a multi-head self-attention block and a MLP block. To stabilize training, layer normalization is applied before each component, and residual connections are added after each component. The encoder uses a scaled dot-product attention mechanism, which computes attention scores over all the elements in the input sequence. To compute these, inputs are projected to Query, Key, and Value matrices of dimensions d_k and d_v for Q/K and V matrices respectively (Eq. 5). Then, a weighted sum is taken over de values, where weights are determined by a similarity function (dot-product) between the queries and their corresponding keys. The weights are computed by applying the Softmax over the outputs scaled by $\sqrt{d_k}$ (Eq. 6). This scaling factor controls the magnitude of the dot product as d increases, reducing gradients vanishing as a result of the Softmax. In multi-head attention, h self-attention operations are applied by projecting the inputs to h d_k and d_v matrices. These operations are applied in parallel, concatenated over d_v, and finally projected back to D dimensions (Eq. 7).

$$Q_i = XW_i^Q; \quad K_i = XW_i^K; \quad V_i = XW_i^V; \quad \text{for 1, ..., h} \tag{5}$$

$$\text{ATT}_i(Q_i, K_i, V_i) = \text{Softmax}\left(\frac{Q_i K_i^T}{\sqrt{d_k}}\right) V_i \tag{6}$$

$$\text{MH-ATT} = \text{Concat}\left(\text{ATT}_1, ..., \text{ATT}_h\right) W^O \tag{7}$$

where:

- $X \in \mathcal{R}^{N \times D}$ are the N embedded patches
- $W_i^Q \in \mathcal{R}^{D \times d_k}; W_i^K \in \mathcal{R}^{D \times d_k}; W_i^V \in \mathcal{R}^{D \times d_v}$ are the Q, K, and V weights
- $W^O \in \mathcal{R}^{hd_v \times D}$ are the output weights

Implementation Our implementation of the ViT closely follows the methodology described in [6] with some slight modifications. We first divide the image

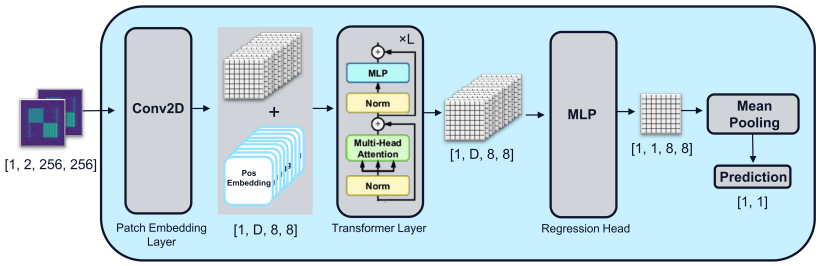


Fig. 4: Architecture of the Vision Transformer.

into non-overlapping patches by applying a two-dimensional convolution with a kernel size and stride equal to 32. We then add learned positional embeddings to the patch embeddings before feeding them to the Transformer encoder layers. These encoder layers utilize pre-layer normalization as described in [34] to remove the need for a warm-up stage during training and use GeLU activations in their MLP modules. We then project to a single feature map by using a MLP equivalent to the MLP as described in Sec. 4.2 with a GeLU activation instead of ReLU. Finally, we apply average pooling over the feature map to obtain a single value. An overview of the ViT architecture is shown in Fig. 4.

5 Experiments & Results

The following section outlines the experimental setups and evaluation results of the experiments carried out during this research. We first elaborate on the datasets and model variants, followed up by our hyper parameter optimization strategy which determines the hyper parameter configurations we use throughout all our experiments. We then continue to discuss the results of our experiments, which includes the evaluation of model performance when a vast amount of data is available during training, as well as evaluation of performance in limited-data regimes.

5.1 Setup

We train all our models from scratch for 250 epochs (unless stated otherwise) and minimize the mean squared error (MSE) between predicted and ground truth overlay. In addition to MSE, we keep track of the mean absolute error (MAE) during evaluation, which is measured in the same unit as the target variable (nanometer). We update our weights using the standard configuration for the AdamW optimizer, which we combine with a cosine annealing learning rate scheduler. Additionally, we clip gradients for weights with a magnitude higher than one and use mixed-precision training to prevent exploding gradients and reduce memory overhead respectively. All model where implemented in Python using the PyTorch (Lightning) library and trained on NVIDIA RTX 2070 Super and/or NVIDIA A100 GPUs.

Datasets We thoroughly asses the performance of our method for overlay metrology using several datasets. Each of these datasets originate from df-DHM simulations implemented in MatLab and contain read noise and shot noise by default, which emulate the effects of real-world sensor imperfections and photon behavior. Simulated DBO targets are also surrounded by neighboring structures, which add CrossTalk imaging effects to the intensity maps. Moreover, we experiment with different types of wavefront errors, which aim to mimic wavefront aberrations commonly encountered in df-DHM systems. Since each optical setup is different and introduces distinct wavefront errors, which remain roughly constant throughout measurements, we distinguish between the following types of wavefront errors, which are characterized as different scalings of predefined coefficients of the first 83 Zernike polynomials.

- Wavefront 0: Images contain no wavefront error
- Wavefront 1: Images contain randomly scaled Zernike coefficients
- Wavefront 2: Images contain identically scaled Zernike coefficients

This approach enables us to evaluate performance on the ideal case (wavefront 0), as well as a more realistic scenario with constant aberrations (wavefront 2). Additionally, the randomly scaled wavefront error (wavefront 1) serves as a more challenging and unrealistic setting, which can provide valuable insights in the limitations of our method. To assess the influence of DBO target size, we apply each type of wavefront error on both the C10 (5 x 5 μm^2) and C16 (8 x 8 μm^2) DBO targets, resulting in a total of 6 datasets. Each of these datasets contain a total of 2048 simulated image-label pairs, where the image is a two-channeled 256 by 256 dimensional tensor representing $\pm 1^{st}$ diffraction orders, and the label is one continues value uniformly distributed in the open interval (-10, 10), serving as the ground-truth overlay value measured in nanometers. During our experiments, we use 1024 image-label pairs for training and 1024 pairs for evaluation.

Model Variants Since each dataset is relatively small and contains similar images, choosing the appropriate model size for optimal performance is non-trivial. We therefore train each architecture at varying scales to observe the trade-offs between model size and performance. For the baselines, we use two types of MLPs with 140 and 240 hidden neurons per layer, and use three types of CNNs with varying amount of convolutional blocks ranging from 3 to 5. These configurations result in 15M and 30M for the MLPs, and 0.1M, 0.5M, and 2M for

Table 1: ViT Model Configurations

Model	Hidden Dim	Heads	Layers	MLP Factor
vit_500K	96	2	4	2
vit_1M	120	4	6	2
vit_5M	240	6	8	3
vit_15M	360	8	10	4

the CNNs. All vision transformers use fixed patch size of 32, resulting in a total sequence length of 64 for a 256 by 256 image. Furthermore, ViTs are trained across 4 different scales by varying the hidden dimension, number of attention heads, number of transformer layers, and MLP dimensions (calculated as MLP Factor $\times$ hidden Dim). Configurations are shown in Table 1.

Hyper Parameter Tuning To ensure equal evaluation conditions, we conduct extensive hyper parameter search for all model variants, optimizing both the learning rate and weight decay. We refrain from optimizing for all datasets and instead lay our focus on the wavefront 0 error, since optimizing on the other wavefront errors may result in over-specific hyper parameter configurations. Additionally, by only focusing on the non-aberrated case, we allow ourselves to assess whether the hyper parameters generalize well to different types of wavefront errors. However, we do differentiate between target sizes and search for hyper parameters for both the C10 and C16 targets separately, since early testing indicated significant differences between target sizes. Optimization was performed using Bayesian optimization, more specifically we used Tree-structured Parzen Estimaton (TPE) algorithm as implemented in Optuna. For each trial, we train each model for 75 epochs on random subsets (75%) of the training data and use the rest for validation. Final parameter configurations were determined after 25 trials and can be found in Appendix A.

Computational Time Evaluation The following section evaluates the computational cost of each model during training and inference. To ensure fair comparisons, we eliminate the overhead of logging, data loading, and other non-essential operations by only recording the timing of forward and backward passes on randomly generated data. Table 2 summarizes average runtime performance for sequential and batched inference, as well as batched training. All measurements were performed on a single NVIDIA RTX 2070 Super GPU, were repeated 100 times, and underwent 10 warm-up rounds to account for initialization. Inference

Table 2: Computational costs for each model

Model	Sequential Inference (ms/image)	Batched Inference (ms/image)	Batched Training (s/1024 images)
conv_100K	0.505	0.394	1.196
conv_500K	0.540	0.441	1.335
conv_2M	0.505	0.472	1.441
mlp_15M	0.200	0.012	0.155
mlp_30M	0.330	0.020	0.219
vit_500K	2.904	0.059	0.260
vit_1M	3.223	0.069	0.333
vit_5M	4.492	0.177	0.657
vit_15M	5.197	0.435	1.490

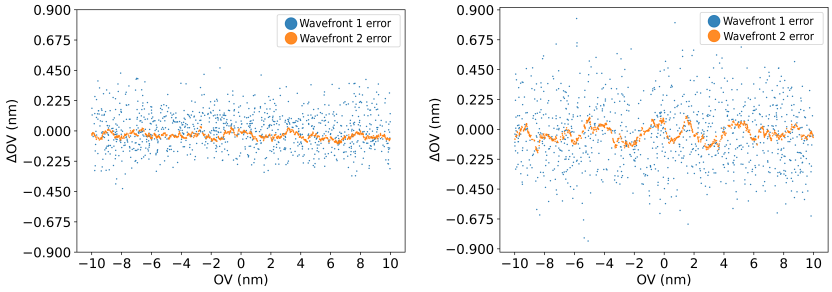
time is reported in milliseconds per image, and is measured for both sequential (batch=1) and batched (batch=64) inputs. Considering the sequential case, we observe that all models process inputs in less than a few milliseconds. ViTs show significantly slower inference compared to CNNs and MLPs in the sequential setting, with processing times for the smallest model being almost 9 times higher than the largest MLP model. This decrease in inference speed will mainly be due to complexity of the attention mechanism of ViT, but other factors (e.g. lack of optimization for single inputs) could also have had an impact. When shifting to batched inference, we noticed that the processing time of ViTs improved dramatically. In contrast to sequential inference, ViTs offer per image processing speeds well below a single millisecond on batched inputs across all model scales. This suggests that despite the high cost of self-attention, ViTs scale well under parallel inputs, making them more efficient than most CNNs. MLPs have the fastest image throughput, and perform well on both sequential and batched inputs due to its simple structure. Since the models are relatively small, training progresses quickly across all architectures, with batched (batch=64) training times remaining under 1.5 seconds per 1024 images. We find that MLPs are again the most efficient, but they are closely followed by the smaller ViTs, which can traverse a full epoch (without validation) in less than a second.

5.2 Full Dataset Evaluation

We first consider the performance of the models on the bigger C16 target. Table 3 shows the MSE (nm^2) and MAE (nm) over the different wavefront errors and model types, we achieve on average sub-nanometer precision across all wavefront errors and model types. Initially, we observe similar performance between wavefront 0 and wavefront 2 errors, which suggests that models can still achieve competitive performance when df-DHM measurements are severely aberrated, as long as the aberrations remain roughly constant. Results for the wavefront 1 error confirm this behavior, as models exhibit significantly lower performance across all scales compared to aberration types. This difference in performance

Table 3: Test-loss (mean ±std) for each model on on the C16 Targets. Each model is trained five times on the full dataset.

Model	Wavefront 0 Error		Wavefront 1 Error		Wavefront 2 Error	
	MSE (nm^2)	MAE (nm)	MSE (nm^2)	MAE (nm)	MSE (nm^2)	MAE(nm)
conv_100K	0.003 ±0.001	0.042 ±0.007	0.110 ±0.045	0.256 ±0.051	0.016 ±0.004	0.106 ±0.018
conv_500K	0.002 ±0.001	0.033 ±0.006	0.056 ±0.026	0.180 ±0.041	0.008 ±0.002	0.065 ±0.008
conv_2M	0.005 ±0.001	0.053 ±0.007	0.052 ±0.007	0.181 ±0.014	0.003 ±0.001	0.039 ±0.008
mlp_15M	0.002 ±0.001	0.034 ±0.006	**0.016 ±0.001**	**0.101 ±0.002**	**0.001 ±0.000**	0.024 ±0.006
mlp_30M	**0.001 ±0.000**	0.022 ±0.001	**0.016 ±0.001**	0.103 ±0.002	**0.001 ±0.000**	**0.020 ±0.003**
vit_500K	**0.001 ±0.000**	0.029 ±0.001	0.023 ±0.002	0.119 ±0.005	0.003 ±0.000	0.046 ±0.003
vit_1M	0.002 ±0.001	0.037 ±0.007	0.031 ±0.004	0.140 ±0.008	0.003 ±0.001	0.043 ±0.007
vit_5M	**0.001 ±0.000**	**0.019 ±0.001**	0.021 ±0.000	0.114 ±0.001	0.002 ±0.000	0.038 ±0.001
vit_15M	**0.001 ±0.000**	0.020 ±0.001	0.022 ±0.002	0.118 ±0.006	**0.001 ±0.000**	0.030 ±0.003



(a) Test results on C16 Target (b) Test results on C10 Target

Fig. 5: Test results for wavefront 1 and 2 errors on the C16 (a) and C10 (b) targets. Y-axis shows the difference between ground truth and predicted overlay. Results are achieved by the vit_5M model.

between wavefront errors is further visible in Fig. 5a, where we plot the difference between predicted and ground truth overlay values (ΔOV) for each sample in the test dataset. The wavefront 1 error clearly shows higher variance and more scattered errors compared to the wavefront 2 error, which forms a tighter distribution around 0. This indicates that for randomly scaled wavefront errors, our models struggle to make consistent overlay predictions for targets with similar overlay errors. Our results further show that in general, MLPs consistently achieve lower losses across wavefront errors. However, ViTs achieve competitive performance, while being more scalable and containing less parameters compared to the MLPs. CNNs have the worst performance, but to not greatly deviate from ViTs or MLPs.

Table 4 shows the results on the C10 targets. Similar to our previous results, we achieve sub-nanometer precision on average across models and wavefront errors and again observe the similarity in performance between the wavefront 0 and wavefront 2 errors. However, we perform in almost all cases marginally

Table 4: Test-loss (mean ±std) for each model on on the C10 Targets. Each model is trained five times on the full dataset

Model	Wavefront 0 Error		Wavefront 1 Error		Wavefront 2 Error	
	MSE (nm^2)	MAE (nm)	MSE (nm^2)	MAE (nm)	MSE (nm^2)	MAE (nm)
conv_100K	0.006 ±0.001	0.060 ±0.008	0.089 ±0.008	0.236 ±0.010	0.006 ±0.002	0.060 ±0.009
conv_500K	0.006 ±0.001	0.065 ±0.006	0.109 ±0.014	0.264 ±0.018	0.014 ±0.008	0.087 ±0.028
conv_2M	0.011 ±0.008	0.075 ±0.025	0.124 ±0.042	0.278 ±0.043	0.011 ±0.004	0.082 ±0.016
mlp_15M	0.004 ±0.000	0.050 ±0.003	**0.054 ±0.000**	**0.187 ±0.001**	**0.004 ±0.001**	0.053 ±0.006
mlp_30M	0.006 ±0.002	0.059 ±0.010	0.057 ±0.004	0.193 ±0.006	**0.004 ±0.001**	**0.048 ±0.006**
vit_500K	0.003 ±0.001	0.047 ±0.007	0.061 ±0.001	0.197 ±0.002	0.005 ±0.000	0.057 ±0.001
vit_1M	**0.001 ±0.000**	**0.030 ±0.001**	0.062 ±0.000	0.201 ±0.001	0.005 ±0.000	0.056 ±0.002
vit_5M	**0.001 ±0.000**	**0.030 ±0.001**	0.063 ±0.001	0.200 ±0.002	**0.004 ±0.000**	0.053 ±0.002
vit_15M	0.004 ±0.001	0.052 ±0.006	0.070 ±0.007	0.209 ±0.009	0.007 ±0.002	0.070 ±0.009

worse compared to the bigger target. This is especially visible for the Wavefront 1 error, as shown in Fig. 6, where a large gap between the C10 and C16 error can be observed. However, it seems that on average this decrease in performance behaves as a constant factor that is invariant to the different wavefront errors, as the MAE error for the C16 targets is approximately a third smaller than the error achieved on the C10 targets. Fig. 5b further demonstrates the difference in performance between targets, we observe that both wavefront errors exhibit more variance in their predictions compared to the C16 targets. Moreover, we notice that scaling the model size has a more negative effect for the C10 targets, in particular the CNNs seem to perform worse when trained at a larger scale. MLPs do exhibit better performance at larger scale, but the difference is slight and only present on the wavefront 2 error. For the ViTs, scaling model size increases performance on both the wavefront 0 and wavefront 2 errors, but starts decreasing when scaling over 5 million parameters.

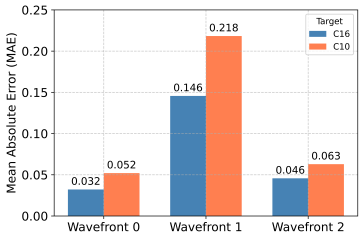


Fig. 6: MAE of each target over all wavefront errors. Errors are averaged over model types.

Table 5: Performance of subsets of Wavefront 1 and 2 errors for C16 Target (MAE)

Error Type	Model	Train Dataset Size				
		32	64	128	256	512
Wavefront 1	conv_100K	0.784 ±0.216	0.687 ±0.128	0.569 ±0.108	0.391 ±0.086	0.266 ±0.055
	conv_500K	0.813 ±0.183	0.654 ±0.211	0.425 ±0.073	0.368 ±0.053	0.245 ±0.052
	conv_2M	2.664 ±0.823	3.046 ±0.480	0.628 ±0.101	0.340 ±0.053	0.248 ±0.024
	mlp_15M	0.591 ±0.153	0.518 ±0.160	0.352 ±0.070	0.192 ±0.023	**0.123 ±0.006**
	mlp_30M	1.482 ±0.233	2.017 ±0.430	0.448 ±0.045	0.226 ±0.021	0.129 ±0.005
	vit_500K	**0.545 ±0.080**	0.482 ±0.018	0.283 ±0.041	0.178 ±0.026	0.157 ±0.006
	vit_1M	0.718 ±0.203	0.577 ±0.127	0.400 ±0.051	0.252 ±0.034	0.193 ±0.021
	vit_5M	0.546 ±0.083	0.424 ±0.061	0.258 ±0.025	**0.176 ±0.009**	0.134 ±0.004
	vit_15M	0.549 ±0.108	**0.378 ±0.046**	**0.254 ±0.047**	0.183 ±0.016	0.139 ±0.006
Wavefront 2	conv_100K	0.485 ±0.266	0.305 ±0.114	0.249 ±0.082	0.132 ±0.046	0.099 ±0.015
	conv_500K	0.332 ±0.067	0.323 ±0.060	0.158 ±0.035	0.095 ±0.017	0.106 ±0.039
	conv_2M	4.939 ±0.082	2.209 ±0.670	0.332 ±0.117	0.104 ±0.054	0.082 ±0.017
	mlp_15M	1.084 ±0.183	0.382 ±0.166	0.123 ±0.028	0.057 ±0.027	0.033 ±0.007
	mlp_30M	3.520 ±0.691	2.396 ±0.628	0.127 ±0.039	0.064 ±0.021	**0.032 ±0.009**
	vit_500K	0.172 ±0.073	0.116 ±0.037	0.144 ±0.022	0.102 ±0.021	0.063 ±0.004
	vit_1M	0.334 ±0.267	0.276 ±0.084	0.142 ±0.072	0.075 ±0.007	0.067 ±0.024
	vit_5M	**0.122 ±0.021**	**0.091 ±0.018**	**0.076 ±0.004**	0.060 ±0.009	0.042 ±0.002
	vit_15M	0.180 ±0.056	0.174 ±0.049	0.078 ±0.040	**0.041 ±0.013**	**0.032 ±0.006**

5.3 Limited Dataset Evaluation

We now consider the amount of data as limiting factor and perform repeated experiments to evaluate the models performance when scaling the amount of training data down to just 32 points. We focus on the wavefront 1 and wavefront 2 datasets, as we already established that similar performance can be achieved between the wavefront 0 and wavefront 2 errors. Table 5 shows the MAE error achieved by each model on this target for both wavefront errors on the C16 target. We observe that all models benefit from more training data, which not only reduces the average error achieved, but also decreases the variance of the results between models that share the same architecture and model size. In most cases, we still obtain nanometer precision on average across wavefront errors, but this does not hold for the larger CNNs and MLPs at the higher levels of data reduction. For these models, we observe that some random initializations can result in plateauing loss curves, indicating non-meaningful learning behavior. ViTs on the other hand do not show this behavior for the C16 targets, and outperform the other architectures consistently when little data is available during training, with improved performance for larger models. This is most noticeable when limiting the amount of training points to 256 or less, which makes ViTs a robust solution when data is scarce. MLPs and CNNs can also achieve good performance, but need more training data compared to ViTs, especially CNNs. Table 6 shows the results for the C10 target, which follow the same trend observed previously. Vision Transformers, especially the smaller sized models, offer superior performance compared to CNNs and MLPs when limited amount of data is presented during training. In general, smaller sized models perform better compared to

Table 6: Performance of subsets of Wavefront 1 and 2 errors for C10 Target (MAE)

Error Type	Model	Train Dataset Size				
		32	64	128	256	512
Wavefront 1	conv_100K	1.000 ±0.176	0.875 ±0.152	0.408 ±0.030	0.336 ±0.040	0.274 ±0.008
	conv_500K	1.011 ±0.230	1.124 ±0.380	0.416 ±0.061	0.358 ±0.032	0.307 ±0.027
	conv_2M	1.078 ±0.352	1.157 ±0.702	0.506 ±0.097	0.378 ±0.030	0.292 ±0.023
	mlp_15M	2.994 ±0.091	1.458 ±0.279	0.367 ±0.022	**0.219 ±0.008**	**0.199 ±0.006**
	mlp_30M	4.680 ±0.253	4.270 ±0.110	0.967 ±0.392	0.261 ±0.023	1.179 ±2.186
	vit_500K	**0.365** ±0.039	**0.374 ±0.028**	**0.251 ±0.022**	0.223 ±0.006	0.206 ±0.002
	vit_1M	0.381 ±0.036	0.400 ±0.021	0.276 ±0.005	0.239 ±0.004	0.215 ±0.004
	vit_5M	0.419 ±0.017	0.395 ±0.025	0.273 ±0.006	0.232 ±0.009	0.214 ±0.002
	vit_15M	4.551 ±0.565	1.677 ±0.780	0.387 ±0.060	0.300 ±0.022	0.228 ±0.007
Wavefront 2	conv_100K	0.191 ±0.054	0.264 ±0.080	0.111 ±0.051	0.115 ±0.025	0.095 ±0.023
	conv_500K	0.327 ±0.224	0.262 ±0.107	0.131 ±0.056	0.168 ±0.046	0.093 ±0.031
	conv_2M	0.237 ±0.108	0.222 ±0.060	0.162 ±0.047	0.158 ±0.068	0.104 ±0.026
	mlp_15M	0.285 ±0.096	0.185 ±0.043	0.135 ±0.061	**0.068 ±0.018**	**0.059 ±0.005**
	mlp_30M	3.288 ±0.638	3.436 ±0.514	0.254 ±0.124	0.099 ±0.032	0.062 ±0.008
	vit_500K	**0.103 ±0.018**	**0.122 ±0.027**	**0.084 ±0.010**	0.070 ±0.000	0.062 ±0.005
	vit_1M	0.179 ±0.020	0.197 ±0.016	0.125 ±0.022	0.084 ±0.010	0.066 ±0.004
	vit_5M	0.185 ±0.028	0.157 ±0.031	0.108 ±0.014	0.091 ±0.011	0.074 ±0.006
	vit_15M	0.621 ±0.391	0.708 ±0.595	0.135 ±0.044	2.062 ±2.636	0.087 ±0.021

their bigger counterparts, which is less pronounced in the C16 results. This suggests that smaller model sizes may be more effective when target size decreased, whereas for a bigger target size, performance can benefit from increased model scale.

6 Conclusion

In this thesis we demonstrated the application of data-driven, end-to-end methods which directly infer overlay errors from aberrated df-DHM simulations. We accomplished this by modeling the relation between intensity images and overlay measurements using various deep learning architectures, bypassing any computationally expensive aberration correction and coherent imaging effect reduction. Our main architecture of interest was the Vision Transformer, which we compared to other popular architectures such as Convolutional Neural Networks and Multilayer Perceptrons. We conducted a comprehensive analysis by training and evaluating each architecture while varying the DBO target size, wavefront aberration type, model scale, and dataset size.
Our results show that all of the models are capable of making sub-nanometer accurate overlay predictions on df-DHM measurements when (1) aberrations are constant, and (2) sufficient data is available during training, achieving performance nearly identical to non-aberrated images. This highlights the potential of real-world deployment of these models in optical systems where aberrations remain roughly throughout measurements. However, performance significantly decreased when aberrations fluctuated over measurements, which resulted in predictions with high variance between DBO targets with similar overlay errors. This clearly shows the limitation of our model's ability to perform steady measurements under non-static optical environments that feature fluctuating aberrations over time. Furthermore, we find that ViTs consistently outperform CNNs and MLPs in limited-data regimes. In particular, ViTs offer superior performance compared to the other models when the amount of training data was limited to 256 samples or fewer, which makes them combined with their high image throughput and fast training speeds an attractive solution for practical applications. We further quantified the influence of the DBO target size on performance and report that all models exhibit reduced predictive overlay precision when trained on a smaller target size, even in the ideal unaberrated case. This observation suggests that reducing the target sizes has a fixed negative effect on the overlay precision, regardless of present wavefront errors. Additionally, we notice that larger targets benefit from increased model size, while for the smaller targets the contrary holds, which is observed in the full and limited-data regime.

7 Discussion

Even though our methods have proven to be effective across a variety of conditions, it is important to acknowledge the limitations of our methods and lay

out potential improvements for future work. We start by addressing characteristics our datasets, which differ in many aspects from conventional computer vision datasets. For instance, the size of our datasets is relatively small and contains similar images of the same target. Moreover, both the training and test set originate from the same simulation. These attributes limit the possibility to effectively evaluate generalization and most likely will result in models overfitting on the training dataset. While this is not necessarily a problem for our use case, it would be interesting to see how the models perform when more variance between images in the datasets is present, as this would add to the robustness of our methods. Furthermore, our simulations contain image-label pairs with uniformly distributed ground-truth overlay values. This is not representative of real-world df-DHM measurements, where overlay errors close to 0 are more common. Therefore our results might give an idealistic estimate of performance, especially for our full data experiments. We argue that making the simulations more representative of experimental data, in terms of content and label distribution, could provide more realistic insights to the performance differences when transitioning from simulations to experimental data. Additionally, this would contribute to narrowing the domain gap and may open the doors to obtaining robust results on experimental data while using simulated data during training. Other potential improvements include making changes to the model's architecture, using domain adaptation techniques to further bridge the domain gap, and incorporating characteristics of the optical setup during the training process.

References

1. Boef, A.: Optical wafer metrology sensors for process-robust cd and overlay control in semiconductor device manufacturing. Surface Topography: Metrology and Properties **4**, 023001 (02 2016). https://doi.org/10.1088/2051-672X/4/2/023001
2. Choi, Y., Yang, T., Lee, K., Choi, W.: Full-field and single-shot quantitative phase microscopy using dynamic speckle illumination. Optics Letters **36**, 2465–2467 (06 2011). https://doi.org/10.1364/OL.36.002465
3. Colomb, T., Cuche, E., Charrière, F., Kühn, J., Aspert, N., Montfort, F., Marquet, P., Depeursinge, C.: Automatic procedure for aberration compensation in digital holographic microscopy and applications to specimen shape compensation. Appl. Opt. **45**(5), 851–863 (Feb 2006). https://doi.org/10.1364/AO.45.000851
4. Colomb, T., Kühn, J., Charrière, F., Depeursinge, C., Marquet, P., Aspert, N.: Total aberrations compensation in digital holographic microscopy with a reference conjugated hologram. Opt. Express **14**(10), 4300–4306 (May 2006). https://doi.org/10.1364/OE.14.004300
5. Dasari, P., Smith, N., Goelzer, G., Liu, Z., Li, J., Tan, A., Koh, C.H.: A comparison of advanced overlay technologies. In: Raymond, C.J. (ed.) Metrology, Inspection, and Process Control for Microlithography XXIV. vol. 7638, p. 76381P. International Society for Optics and Photonics, SPIE (2010), https://doi.org/10.1117/12.848189
6. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.:

An image is worth 16x16 words: Transformers for image recognition at scale (10 2020). https://doi.org/10.48550/arXiv.2010.11929
7. Feng, P., Xiao, W., Liu, S., Wang, F., Rong, L., Li, R.: Coherent noise reduction in digital holographic phase contrast microscopy by slightly shifting object. Optics Express **19**, 3862–3869 (02 2011). https://doi.org/10.1364/OE.19.003862
8. Ferraro, P., De Nicola, S., Finizio, A., Coppola, G., Grilli, S., Magro, C., Pierattini, G.: Compensation of the inherent wave front curvature in digital holographic coherent microscopy for quantitative phase-contrast imaging. Applied Optics **42**, 1938–1946 (04 2003). https://doi.org/10.1364/AO.42.001938
9. Garcia-Sucerquia, J.: Noise reduction in digital lensless holographic microscopy by engineering the light from a light-emitting diode. Appl. Opt. **52**(1), A232–A239 (Jan 2013). https://doi.org/10.1364/AO.52.00A232
10. Garcia-Sucerquia, J., Ramírez, J.A.H., Prieto, D.V.: Reduction of speckle noise in digital holography by using digital image processing. Optik **116**(1), 44–48 (2005). https://doi.org/10.1016/j.ijleo.2004.12.004
11. Huang, L., Tang, J., Yan, L., Chen, J., Chen, B.: Wrapped phase aberration compensation using deep learning in digital holographic microscopy. Applied Physics Letters **123**(14), 141109 (10 2023). https://doi.org/10.1063/5.0166210
12. Jeon, W., Jeong, W., Son, K., Yang, H.: Speckle noise reduction for digital holographic images using multi-scale convolutional neural networks. Opt. Lett. **43**(17), 4240–4243 (Sep 2018). https://doi.org/10.1364/OL.43.004240
13. Lin, Z., Jia, S., Xu, Y., Wen, B., Zhang, H., Wang, L., Han, M.: Fast phase distortion identification and automatic distortion compensated reconstruction for digital holographic microscopy using deep learning. Optics and Lasers in Engineering **185**, 108718 (2025). https://doi.org/https://doi.org/10.1016/j.optlaseng.2024.108718
14. Ma, S., Fang, R., Luo, Y., Liu, Q., Wang, S., Zhou, X.: Phase-aberration compensation via deep learning in digital holographic microscopy. Measurement Science and Technology **32**(10), 105203 (jun 2021). https://doi.org/10.1088/1361-6501/ac0216
15. Mann, C., Yu, L., Lo, C., Kim, M.: High-resolution quantitative phase-contrast microscopy by digital holography. Optics Express **13**(22), 8693–8698 (Oct 2005). https://doi.org/10.1364/OPEX.13.008693
16. Maycock, J., Hennelly, B., McDonald, J., Frauel, Y., Castro, A., Javidi, B., Naughton, T.: Reduction of speckle in digital holography by discrete fourier filtering. Journal of the Optical Society of America A **24**, 1617–1622 (05 2007). https://doi.org/10.1364/JOSAA.24.001617
17. Messinis, C., Adhikary, M., Cromwijk, T., van Schaijk, T., Witte, S., Boer, J., Boef, A.: Pupil apodization in digital holographic microscopy for reduction of coherent imaging effects. Optics Continuum **1**, 1202–1217 (05 2022). https://doi.org/10.1364/OPTCON.460029
18. Messinis, C., van Schaijk, T., Tenner, V., Witte, S., Boer, J., den Boef, A.: Diffraction-based overlay metrology using angular-multiplexed acquisition of dark-field digital holograms. Optics Express **28**, 37419–37435 (11 2020). https://doi.org/10.1364/OE.413020
19. Messinis, C., van Schaijk, T.T.M., Pandey, N., Koolen, A., Shlesinger, I., Liu, X., Witte, S., de Boer, J.F., den Boef, A.: Aberration calibration and correction with nano-scatterers in digital holographic microscopy for semiconductor metrology. Opt. Express **29**(23), 38237–38256 (Nov 2021). https://doi.org/10.1364/OE.438026
20. Miccio, L., Alfieri, D., Grilli, S., Ferraro, P., Finizio, A., De petrocellis, L., De Nicola, S.: Direct full compensation of the aberrations in quantitative phase

microscopy of thin objects by a single digital hologram. Applied Physics Letters **90**, 041104 – 041104 (02 2007). https://doi.org/10.1063/1.2432287

21. Min, J., Yao, B., Gao, P., Ma, B., Yan, S., Peng, F., Zheng, J., Ye, T., Rupp, R.: Wave-front curvature compensation of polarization phase-shifting digital holography. Optik **123**(17), 1525–1529 (2012). https://doi.org/https://doi.org/10.1016/j.ijleo.2011.09.018
22. Moore, G.E.: Cramming more components onto integrated circuits, reprinted from electronics, volume 38, number 8, april 19, 1965, pp.114 ff. vol. 11, pp. 33–35 (2006). https://doi.org/10.1109/N-SSC.2006.4785860
23. Morimoto, Y., Matui, T., Fujigaki, M., Kawagishi, N.: Subnanometer displacement measurement by averaging of phase difference in windowed digital holographic interferometry. Optical Engineering **46**(2), 025603–025603 (2007). https://doi.org/10.1117/1.2538709
24. Nehmetallah, G., Nguyen, T.: Optical and digital aberration compensation in dhm. In: Imaging and Applied Optics 2016. p. JW4A.5. Optica Publishing Group (2016), https://opg.optica.org/abstract.cfm?URI=AIO-2016-JW4A.5
25. Nguyen, T., Bui, V., Lam, V., Raub, C.B., Chang, L.C., Nehmetallah, G.: Automatic phase aberration compensation for digital holographic microscopy based on deep learning background detection. Opt. Express **25**(13), 15043–15057 (Jun 2017). https://doi.org/10.1364/OE.25.015043
26. Qu, W., Choo, C., Singh, V., Yingjie, Y., Asundi, A.: Quasi-physical phase compensation in digital holographic microscopy. Journal of the Optical Society of America A **26**, 2005–2011 (08 2009). https://doi.org/10.1364/JOSAA.26.002005
27. Ren, Z., Xu, Z., Lam, E.Y.M.: End-to-end deep learning framework for digital holographic reconstruction. Advanced Photonics **1**(1), 016004 (2019). https://doi.org/10.1117/1.AP.1.1.016004
28. Rong, L., Xiao, W., Feng, P., Liu, S., Li, R.: Speckle noise reduction in digital holography by use of multiple polarization holograms. Chinese Optics Letters **8**, 653–655 (07 2010). https://doi.org/10.3788/COL20100807.0653
29. Sánchez-Ortiga, E., Ferraro, P., Martínez-Corral, M., Saavedra, G., Doblas, A.: Digital holographic microscopy with pure-optical spherical phase compensation. J. Opt. Soc. Am. A **28**(7), 1410–1417 (Jul 2011). https://doi.org/10.1364/JOSAA.28.001410
30. van Schaijk, T.T.M., Messinis, C., Pandey, N., Koolen, A., Witte, S., de Boer, J.F., Boef, A.D.: Diffraction-based overlay metrology from visible to infrared wavelengths using a single sensor. Journal of Micro/Nanopatterning, Materials, and Metrology **21**(1), 014001 (2022). https://doi.org/10.1117/1.JMM.21.1.014001
31. Stadelmaier, A., Massig, J.: Compensation of lens aberrations in digital holography. Optics Letters **25**, 1630–1632 (11 2000). https://doi.org/10.1364/OL.25.001630
32. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A., Kaiser, L., Polosukhin, I.: Attention is all you need (06 2017). https://doi.org/10.48550/arXiv.1706.03762
33. Xiao, W., Zhang, J., Rong, L., Feng, P., Liu, S., Wang, F., He, A.: Improvement of speckle noise suppression in digital holography by rotating linear polarization state. Chinese Optics Letters **9**, 060901 (06 2011). https://doi.org/10.3788/COL201109.060901
34. Xiong, R., Yang, Y., He, D., Zheng, K., Shuxin, Z., Xing, C., Zhang, H., Lan, Y., Wang, L., Liu, T.Y.: On layer normalization in the transformer architecture (02 2020). https://doi.org/10.48550/arXiv.2002.04745

35. Yan, K., Chang, L., Andrianakis, M., Tornari, V., Yu, Y.: Deep learning-based wrapped phase denoising method for application in digital holographic speckle pattern interferometry. Applied Sciences **10**(11) (2020). https://doi.org/10.3390/app10114044
36. Yin, D., Gu, Z., Zhang, Y., Gu, F., Nie, S., Feng, S., Ma, J., Yuan, C.: Speckle noise reduction in coherent imaging based on deep learning without clean data. Optics and Lasers in Engineering **133**, 106151 (2020). https://doi.org/10.1016/j.optlaseng.2020.106151

Appendix A Hyperparameters

Model	C10		C16	
	lr	weight_decay	lr	weight_decay
conv_100K	7e-4	5e-2	3e-3	7e-3
conv_500K	9e-5	9e-3	1e-3	4e-3
conv_2M	1e-4	3e-3	4e-5	3e-3
mlp_15M	5e-3	9e-3	4e-3	4e-2
mlp_30M	3e-3	7e-2	9e-4	2e-3
ViT_500K	5e-3	4e-3	1e-3	6e-2
ViT_1M	4e-4	3e-3	1e-3	2e-3
ViT_5M	2e-4	9e-3	1e-5	8e-3
ViT_15M	5e-3	9e-3	1e-3	9e-3

Table 7: Model types and their corresponding parameters for C10 and C16 targets.