ETH zürich

PAUL SCHERRER INSTITUT
PSI

Differential phase contrast mammogram denoising and integration

Master Thesis
Master of Science in Biomedical Engineering

Alexander Pereira

18 August 2021

Supervisor: Prof. Dr. Marco Stampanoni
Advisor: Stefano van Gogh

Abstract

Breast cancer is the most common type of cancer in women, making early detection particularly important. Unfortunately, conventional screening techniques are still restricted as they either provide insufficient resolution and/or soft-tissue contrast. For this reason, X-ray phase contrast has been investigated in recent years, potentially leading to higher soft-tissue contrast while still retaining the ability to image at high spatial resolution. X-ray grating interferometry (GI) is a phase contrast technique that has been extensively studied as it meets the requirements for clinical compatibility. Several ex-vivo mastectomies studies have been performed which demonstrated the advantages that GI can bring to mammography. To make the system ready for clinical use, a grating interferometer has been installed in a Philips Microdose mammography system at the University Hospital Zurich and is ready for the first in-vivo studies. In 2D the GI system is only capable of outputting the differential phase contrast (DPC) image. To get the clinically relevant phase contrast information, the DPC signal has to be integrated in the direction of the phase stepping. This integration step is challenging due to intrinsic noise in the DPC image which leads to severe blurring and stripe artefacts. In order to obtain high-quality phase contrast images, the DPC signal has to be nearly perfectly denoised. In this work we propose a data-driven denoising algorithm, which tries to cope with the high intrinsic noise amplitudes arising in a clinical GI system for the DPC channel. In particular, we trained an in-house developed deep learning algorithm – developed for Grating Interferometry Breast Computed Tomography (GI-BCT) called Interpretable NonexpanSIve Data-Efficient network (INSIDEnet) –, and trained it on a simulated DPC breast projections dataset, while also adding a Bayesian perspective on to it to model the uncertainty. Furthermore, we combined the method with ideas from state-of-the-art deep learning architectures, namely the U-Net, which we call Collaborative-Pyramid Bayesian Neural Network (CP-BNN). We trained the models in a supervised and unsupervised fashion by exploiting known statistical noise properties of the GI-setup, and showed that the proposed algorithms outperform traditional state-of-the-art filters and are competitive with sophisticated deep-learning algorithms, while simultaneously providing information of the uncertainty in the predictions. As a final step, we evaluated our models on real-world experimental data acquired by the Philips Microdose system. We found that the deep-learning models can denoise simulated as well as real images efficiently and enable the retrieval of phase contrast images with less stripe artefacts and more details than without denoising.

Contents

1 Introduction

Receiving a breast cancer diagnosis is the reality for nearly 2.2 million women each year. It is the most prevalent malignancy in women, which is why early detection of the disease is imperative [1] [2]. This has led to the introduction of many screening programs – the most widespread screening and evaluation technique nowadays being mammography. It is able to significantly reduce the number of deaths caused by breast cancer due to early detection [3]. Unfortunately, mammography is restricted by its limited sensitivity and specificity, resulting in both overlooked tumors and psychological distress [3]. The causes for these limitations range from low soft tissue contrast to the inability to generate fully three-dimensional data [4]. Alternative methods such as digital tomosynthesis (DBT) have shown improvements to the screening performance, but, as with traditional mammography, are limited in extremely dense breasts [4].

As a consequence, an ever increasing effort is being made by the research community to exploit X-ray phase contrast imaging, which can potentially lead to orders of magnitude higher soft-tissue contrast compared to conventional absorption-based imaging, but still retains the ability to generated images at high spatial resolutions [5]. X-ray grating interferometry (GI) not only measures the beam attenuation but is also capable of analyzing the beam refraction induced by the sample as well as the sample scattering leading to the phase contrast and dark-field image, with the latter having potential benefits for non-invasive classification and detection of microcalcifications and breast lesions [6] [7]. Several mastectomy studies have been conducted in the past to see the contribution that GI can make to mammography [7] [8]. They demonstrated that the differential phase contrast and dark-field signal did provide complementary information to the conventional attenuation signal. It was shown that the phase contrast mammography could improve image quality, sharpness, lesion delineation and microcalcification visibility, as well as clearer representation of the breast anatomy. To translate the idea of a GI-setup to clinical practice, a 2D grating interferometry setup has been integrated into an existing Philips mammography scanner [9]. The scanner is installed at the University Hospital Zurich and will be tested in first in-vivo trials soon.

To obtain the clinically interesting phase contrast (PC) signal, the collected differential phase contrast (DPC) signal from the GI has to be integrated along the direction of phase stepping. Unfortunately, this integration step has demonstrated to be very challenging to perform. In fact, due to the instrinsic noise in the DPC-channel, which occurs from detector quantum noise and phase jittering [10], even low amounts of noise can lead to severe artefacts in the retrieved phase. These manifest themselves as blurring or – more prominently – stripe artefacts along the integration direction in the phase contrast image. Thus, an almost perfect denoising must take place in order for the integration to be successful.

Traditional state-of-the-art denoising algorithms, such as non-local means (NLM) [11] or the block matching and 3D (BM3D) filter [12], form reliable and stable pipelines when applied to images with little noise. However, their performance deteriorates drastically when being confronted with images with lower quality. These algorithms were usually built to deal with noise arising from the same distribution and are thus not fit to deal with heteroscedastic noise as in the DPC-channel. As a result of the increasing use of deep learning in image analysis and computer vision tasks, denoising deep neural networks (DNN) have emerged in the field of biomedical imaging [13–17]. These models have shown impressive denoising results by implicitly learning a prior from the data provided during training. Consequently, these models are heavily dependent on the data given. Furthermore, due to the concatenation of non-linear calculations, they lose interpretability. Additionally, traditional deep learning models are prone to overfitting, leading to lower generalization capabilities. They also tend to be overconfident about their prediction, which can become especially problematic in areas such as medical diagnostics or autonomous driving [18]. As a consequence, there have been several approaches to mitigate this problem, especially via the introduction of stochastic neural networks – namely Bayesian neural networks – which provide both the prediction as well as the uncertainty in the prediction. On the basis of this development, a subgroup of our TOMCAT team has implemented its own deterministic denoising network called "Interpretable NonexpanSIve Data-Efficient network" (INSIDEnet) for the in-house built grating-interferometer breast computer tomography (GI-BCT) with promising results on simulated data [19]. Their method deviates from classical deep learning methods in that they try to combine the interpretability of classical filters and the flexibility of data-driven models by using transform learning and collaborative filters and adapting them to take advantage of supervised learning and multiscale processing. Following their work, we propose to translate their method to our system and implement a probabilistic perspective over the parameters of the model, allowing us to model the uncertainty and generalize better to the data at hand. We call this method "probabilistic-INSIDEnet" (P-INSIDEnet). Being inspired by the success of the state-of-the-art deep neural network architecture – namely the U-Net [20] – we implemented a combination of both algorithms, leading to a second model called "Collaborative-Pyramid Bayesian Neural Network" (CP-BNN). Due to the lack of available real data on the Philips system, we simulated medio lateral oblique (MLO) breast projections and used them to train our own denoising pipeline in both a supervised and unsupervised way. The reason for unsupervised training lies in matching the real-world scenario of missing clean ground truth data.

The goal of this thesis is to generate artefact free phase contrast (PC) images from noisy DPC images. We first demonstrate the demonising performance of both our models – P-INSIDEnet and CP-BNN – compared to the deterministic INSIDEnet model, the U-Net model, and the BM3D algorithm on our simulated DPC breast projections. We show that the data-driven model accomplishes better results than the traditional, non-learning based model. An even superior performance is achieved by the CP-BNN when compared to the state-of-the-art U-Net model, when evaluated on our simulated data. Although these results already show high potential, neither one is able to completely eliminate all noise. Thus, artefacts such as stripes were still visible in the integrated PC images, which required the use of dedicated destriping algorithms to remove residual artefacts. We found that the „wavelet fourier filter“ [21] is a great match for this purpose. Eventually, we run our trained models over a set of real collected DPC-images on the Philips system.

2 Background

In comparison to conventional X-ray techniques, phase-contrast X-ray imaging allows for the measurement of two additional physical properties: refraction and scattering. The refraction induced by the sample provides the so-called phase contrast. The phase contrast has the potential to better distinguish between different soft tissues. Meanwhile, the scattering information leads to the so-called dark-field (DF) signal, which allows for the detection of strong scattering structures such as microcalcification crystals [22]. The interaction of a X-ray beam with matter can be described via the refractive index, which is defined as:

$$n = 1 - \delta - i\beta \tag{1}$$

where δ is the real part of the index of refraction which causes changes in the wavefront's phase and β is responsible for the absorption of the X-rays in the medium [23]. The phase shift and absorption can be explained using simple wave equations. The general form of a wave propagating in z-direction (i.e. $\boldsymbol{k} = (0, 0, k)$) in vacuum can be written as:

$$\psi_v = A \exp\left(i(kz - \omega t)\right) \tag{2}$$

A refers to the wave's amplitude, ω the anglular frequency, and t to the time. If this wave now propagates in a medium, the resulting wave becomes as follows:

$$\psi = A \exp\left(i(nkz - \omega t)\right) = \psi_v \exp\left(-i\delta kz\right) \exp\left(-\beta kz\right) \tag{3}$$

We can see that the medium induces a phase shift as well as attenuates the wave, with the effects being dependent on δ and β, respectively. In conventional X-ray imaging, the phase shift information cannot be detected since only the intensity of the attenuated X-ray beam is measured. At energies higher than 10keV and for soft-tissues which are made up of light elements, δ is typically three orders of magnitude larger than β [23]. With the goal of measuring the phase shift, multiple methods have been developed – the most common ones being crystal interferometry [24], diffraction-enhanced imaging [25], propagation based [26], and grating interferometry [27]. While all methods can be used with a synchrotron light source, only the latter holds the prerequisites for clinical applications, such as mechanical robustness, a large field-of-view (FOV), limited exposure time, and clinically acceptable dosages while keeping a sufficiently high image quality. Importantly, it only requires moderate spatial coherence and monochromaticity, allowing for it to be used with conventional X-ray tubes [27].

2.1 Grating Interferometry

Grating Interferometry (GI) is a method which encodes phase shifts induced by a sample into intensity modulated signals. In the Talbot-Lau configuration, three gratings are placed between the source and the detector, where one grating is placed directly after the source (G0), the next directly after the sample (G1) and the last one in front of the detector (G2)[1] [29] (Fig. 1 left). In the case of a conventional X-ray source, the source itself does not provide a sufficiently high spatial coherence for the GI-rationale. Therefore, by using an absorbing grating (G0) placed in front of the source, we can create an array of individually coherent sources, which are, however, mutually incoherent to each other. Yet, if the period of G0 fulfills the condition

$$p_0 = \frac{L}{d} p_2 \tag{4}$$

where L is the source-to-G1 distance, d the G1-G2 distance and p_2 the G2 period length, then the interference patterns of the neighbouring line sources shift by one period exactly. As a consequence –

[1] Variations exist where G1 is placed directly before the sample [28].

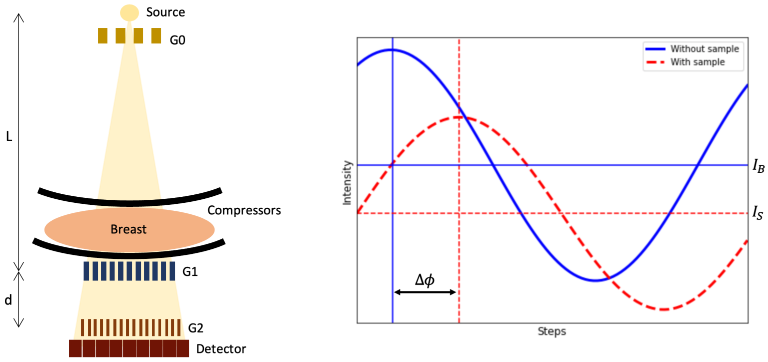

Figure 1: This figure shows the schematic of GI-setup in a mammography scanner (left) and the phase stepping curve with (red) and without (blue) a sample (right). The differential phase signal can be obtained by using the relative phase as a reference. The absorption signal is retrieved from the average of each curve.

since all line sources produce the same interference pattern – spatial coherence is preserved [22]. The grating after the sample (G1) splits the beam, which results in the incoming beam being divided into several diffraction orders. The coherence of the beam then leads to constructive interference of the diffraction orders at distance d – usually where G2 is. Placing a sample in the beam then results in a distortion of the interference pattern. This distortion is explained by the refraction of the medium, which shifts the interference pattern laterally by Δx. From this shift, the refraction angle α can be deduced since the lateral shift is proportional to the refraction angle (i.e. $\Delta x \approx \alpha$)[22]. However, these interference fringes are usually too small to be resolved by a conventional X-ray detector. Thus, a second line grating (G2) of purely absorbing structure is used to analyze the interference pattern, which arises as a consequence of the fractional Talbot effect. Usually, this is achieved by laterally stepping one of the gratings, which translates the lateral offset of the interference pattern into a change of intensity at the detector [30]. This collected pixel-wise intensity signal is called the phase stepping curve (PSC). By combining the PSC with and without sample, it is possible to retrieve the absorption, scattering, and differential phase contrast (DPC) signal (see Fig. 1 right). From this the refraction angle can be deduced which is proportional to the DPC signal $\frac{\partial\phi}{\partial x}$ multiplied by a geometric conversion factor, which depends on the G1-G2 distance d and the G2 period length p_2 [31]:

$$\alpha = \frac{p_2}{2\pi d}\frac{\partial\phi}{\partial x} \tag{5}$$

2.2 Noise Propagation in Grating Interferometer

The noise behaviour in grating-based X-ray imaging in the attenuation, DPC, and dark-field channel have been analyzed in [10] and [32]. The former found the two main noise sources to be detector quantum noise and phase-stepping jitter noise. In this work, we neglect the phase-stepping jitter noise, since we do not simulate mechanical vibrations of the gratings and the piezo-motor (see chapter 3.1). The focus lies on the detector quantum noise. [10] assume that the detector quantum noise variance is directly proportional to the mean intensity I:

$$\sigma_I^2 = f_1 I \tag{6}$$

Here, the slope f_1 is linked to the signal and noise transfer of incoming X-ray photons to the output in arbitrary digital units. It has to be considered that due to beam hardening of the X-ray spectrum when passing through an object, f_1 is generally different for the reference measurement f_1^r and for the sample measurement f_1^s. The uncertainty σ_I then translates into the errors of the individual images –

transmission (T), differential phase (DPC), and dark-field (V) by using the error propagation formula:

$$\left(\frac{\sigma_T}{T}\right)^2 = \frac{f_1^r}{Na_0^r}\left(1 + \frac{f_1^s}{Tf_1^r}\right) \tag{7}$$

$$\sigma_{DPC}^2 = \frac{f_1^r}{2\pi^2 \nu^{r2} N a_0^r}\left(1 + \frac{f_1^s}{f_1^r T V^2}\right) \tag{8}$$

$$\left(\frac{\sigma_V}{V}\right)^2 = \frac{f_1^r}{\nu^{r2} N a_0^r}\left[\nu^{r2}\left(1 + \frac{f_1^s}{f_1^r T}\right) + 2\left(1 + \frac{f_1^s}{f_1^r T V^2}\right)\right] \tag{9}$$

where we define:

T: Mean transmission signal
V: Mean dark-field signal
a_0^r: Mean intensity of the reference measurement
N: Number of phase steps acquired over one period
ν^r: Visibility of reference measurement
f_1^r / f_1^s: Slope of Eq. (6) for reference/sample measurement

The slopes f_1^r and f_1^s have to be investigated on the individual settings and detectors. However, for a photon-counting detector, the Poisson noise model applies and we can assume f_1 to be 1.

To a similar conclusion also came [32]. They formulated the problem as a least-squares fitting in matrix notation and calculated the covariance matrix between the three different contrast channels (assuming $f_1 = 1$). The only difference in the variance of the individual images is in the DPC variance, where they calculated it to be:

$$\sigma_{DPC}^2 = \frac{2}{N a_0^r \nu^{r2}}\left(1 + \frac{1}{TV^2}\right) \tag{10}$$

Therefore, to get from Eq. (8) to Eq. (10), we have to multiply it by $4\pi^2$, which leads to a higher variance in their model compared to [10]. Unfortunately, we have not found a detailed derivation for this extra term. Additionally, they found the noise in DPC to be better modelled as a Von-Mises distribution instead of a normal Gaussian distribution [33]. Following these statistics, it becomes evident that these images are corrupted by noise that does not allow simple filtering as it would with additive white Gaussian noise (AWGN). This applies even more so when clinical standards have to be met due to lower mean intensity and lower number of phase steps. Thus, more sophisticated algorithms are needed to cope with the heteroscedastic noise intrinsic in the DPC images.

2.3 Deep Neural Networks

Deep Neural Networks (DNN) have been successfully used in a variety of fields, such as speech recognition, natural language processing, and computer vision, providing solutions to many complex problems [34]. The goal of DNNs is to map an input $\boldsymbol{x}$ to a desired output $\boldsymbol{y}$ by a learned function $f(\boldsymbol{x}) = NN(\boldsymbol{x}) = \boldsymbol{y}$. Generally, DNNs are composed of an input layer, several hidden layers, and an output layer. The hidden layers are usually constructed by applying a linear transformation to the previous layer, followed by a non-linear function φ, also called *activation function*. In short, the idea of DNNs is to parametrize the feature maps and optimize over the parameters.

$$\boldsymbol{l}_0 = \boldsymbol{x} \tag{11}$$

$$\boldsymbol{l}_i = \varphi(\boldsymbol{W}_i \boldsymbol{l}_{i-1} + \boldsymbol{b}_i) \tag{12}$$

$$\boldsymbol{l}_n = \boldsymbol{y} \tag{13}$$

where i is the current layer, and $\boldsymbol{W}_i \in \mathbb{R}^{N_i \times N_{i-1}}$ and $\boldsymbol{b}_i \in \mathbb{R}^N$ are the linear transformation matrix and bias respectively, which depend on the dimension of the current and previous layer. Optimization of the parameters $\boldsymbol{\theta} = \{\boldsymbol{W_1}, \boldsymbol{b_1}, \dots, \boldsymbol{W_n}, \boldsymbol{b_n}\}$ is performed by minimizing a cost-function L over some training data D using the back-propagation algorithm [35].

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \sum_{\boldsymbol{x},\boldsymbol{y} \in D} L(\boldsymbol{\theta}; \boldsymbol{y}, \boldsymbol{x}) \tag{14}$$

From a statistical point of view, this optimization can be described as a Maximum Likelihood Estimation (MLE). By adding a regularization term to Eq. (14), the optimization becomes a Maximum A-Posteriori (MAP) estimation.

2.4 Convolutional Neural Networks

Convolutional neural networks (CNN) are deep learning networks for specialized applications such as image classification, image segmentation, object recognition, and many more in the Computer Vision domain [36][37]. They are specialized for data with grid-like structures such as images. The main advantage of CNNs is the relatively small number of parameters compared to fully-connected multilayer networks. Instead of each node being dependent on the entire input, the nodes are only dependent on the inputs "close-by" (e.g. pixels in the neighborhood). Furthermore, this weight matrix is shared across the entire image and is identical for all nodes in the layer. Thus, the number of parameters gets reduced – thereby promoting robustness against translations in the image. Formally, (12) is modified to:

$$\boldsymbol{l}_{i,j} = \varphi(\boldsymbol{W}_{i,j} * \boldsymbol{l}_{i-1} + \boldsymbol{b}_{i,j}) \tag{15}$$

where $\boldsymbol{W}_{i,j} \in \mathbb{R}^{K \times K \times C_{i-1}}$ is now a tensor dependent on the kernel size K and input channels C_{i-1}, $\boldsymbol{b}_{i,j} \in \mathbb{R}^{C_{i-1}}$ is the bias term depending on the output-channel size, and $j \in \{1, \dots C_i\}$ represents the corresponding output channel. Another advantage for image analysis is based on the fact that images are composed of hierarchical structures, which CNNs are well suited to leverage. In the initial layers, the CNN can discern only a small portion of the image and encodes local features such as edges or corners. With increasing network depth, more global information is encoded since the receptive field[2] increases as well.

2.5 Bayesian Neural Networks/Bayesian Deep Learning

While DNNs perform fairly well in complex tasks, they may be overconfident when confronted with out-of-training data. Being overconfident in its prediction becomes troublesome in situations where uncertainty has critical consequences, such as autonomous-driving or medical diagnostics [18]. This behaviour is due to the *frequentist* approach of the optimization, i.e. its goal is to optimize a loss function with the optimal set of parameters as in Eq. (14) [38]. Usually, Stochastic Gradient Descent (SGD) [39] or variations of it are performed to iteratively find the parameters. Bayesian Neural Networks (BNN) are a special type of DNN where instead of calculating a point estimate for the parameters, we find a suitable probability distribution over the model parameters:

$$\theta \sim P(\theta \mid D) \tag{16}$$

[2] The receptive field is the region in the input image that a pixel in the output feature map is affected by.

We call (16) the *posterior distribution.* With this, we are now able to model uncertainty in our predictions. To compute the posterior, we use Bayes' Theorem:

$$P(\theta \mid D) = \frac{P(D \mid \theta)P(\theta)}{P(D)} \tag{17}$$

$$\text{where } P(D) = \int P(\theta) \prod_{i=1}^{n} P(y_i \mid x_i, \theta)\, d\theta \tag{18}$$

where $P(D \mid \theta)$ is called the likelihood, $P(\theta)$ the prior and $P(D)$ the marginal likelihood or evidence of our data, where each data point is assumed to be independent and identically distributed (i.i.d.). Usually, calculating $P(D)$ is impossible or intractable since we would need to evaluate the integral in Eq. (18) over all possible model parameters, which can be prohibitively large. Therefore, different methods have to be performed to obtain the desired parameters of the distribution. One option is to use *Markov-Chain Monte Carlo methods* [38, 40–42], where the goal is to seek an approximation of $P(\theta \mid D)$ by sampling from a simulated Markov Chain. Another option is to perform *Variational Inference* [43–45], which is the method used in this work. The goal of variational inference is to approximate the intractable distribution $P(\theta \mid D)$ by a parameterized simple one $Q(\theta \mid \lambda)$ that is "as close as possible" that we call *variational distribution.* λ in Q correspond to the variational parameters which describe the new distribution. The approximation is performed by minimizing the Kullback-Leiber (KL) Divergence between the desired distribution and the simple one:

$$Q* \in \arg\min_{Q \in \mathcal{Q}} KL(Q||P) \tag{19}$$

where $\mathcal{Q}$ specifies the variational family and $KL(Q||P)$ defines the KL-Divergence between two distributions as:

$$\begin{aligned} KL(Q||P) &= \int q(\theta) \log \frac{q(\theta)}{p(\theta)}\, d\theta & (20) \\ &= \mathrm{E}_{\theta \sim Q}\left[\log \frac{q(\theta)}{p(\theta)}\right] & (21) \end{aligned}$$

where q and p are the probability density functions of the two distributions. Although the KL-Divergence is seen as a distance metric between two distributions, it is not symmetric – i.e. $KL(Q||P) \neq KL(P||Q)$. Since we do not have access to the posterior distribution, we have to rewrite the KL-Divergence to gain access to available distributions:

$$\begin{aligned} \arg\min_{Q} KL(Q||P) &= \arg\min_{q} \int q(\theta) \frac{q(\theta)}{\frac{1}{Z} p(\theta, D)}\, d\theta & (22) \\ &= \arg\max_{q} \int q(\theta) \left[\log p(\theta, D) - \log Z - \log q(\theta)\right]\, d\theta & (23) \\ &= \arg\max_{q} \int q(\theta) \log p(\theta, D)\, d\theta + H(q) & (24) \\ &= \arg\max_{q} \mathrm{E}_{\theta \sim q(\theta)} \left[\log p(\theta, D)\right] + H(q) & (25) \\ &= \arg\max_{q} \mathrm{E}_{\theta \sim q(\theta)} \left[\log p(D \mid \theta)\right] - KL(q||p(\cdot)) & (26) \end{aligned}$$

Here, we are writing the posterior distribution as in Eq. (17), where the evidence is defined as Z. Since Z is only a constant in Eq. (23), we can neglect it. In Eq. (24), we introduce H as the entropy of a distribution:

$$H(q) = -\int q(\theta) \log q(\theta)\, d\theta \tag{27}$$

Equations (25) and (26) are also called the "Evidence Lower Bound" (ELBO) i.e. $\log P(D) \geq$ ELBO. This inequality can easily be shown using *Jensen's inequality* and is derived in [46]. Therefore, maximizing the ELBO leads to a higher evidence, meaning that the marginal probability of our observed data becomes high as well, indicating that training is on the right track. Intuitively, the optimization can be understood in two ways: It either prefers distributions Q that maximize the expected joint data likelihood but are also uncertain, or it prefers distributions Q that maximize the conditional data likelihood but are also close to the prior. Therefore, our cost function over which we want to optimize is as follows:

$$L(\lambda) = \mathrm{E}_{\theta \sim q(\cdot \mid \lambda)} \left[\log p(D \mid \theta)\right] - KL(q_\lambda || p(\cdot)) \tag{28}$$

Here, we introduce the variational parameters λ into the equation again, since these are our parameters we optimize and calculate our gradients over. Unfortunately, this would mean we would need to differentiate an expectation with respect to a distribution q, which depends on the variational parameters. To avoid this, we can use the so called "Reparameterization Trick" [47]: Suppose we have a random variable $\epsilon \sim \phi$ sampled from a base distribution and consider $\theta = g(\epsilon, \lambda)$ for some invertible function g, then it holds that $q(\theta \mid \lambda) = \phi(\epsilon)|\nabla_\epsilon g(\epsilon, \lambda)|^{-1}$ and $\mathrm{E}_{\theta \sim q_\lambda}[f(\theta)] = \mathrm{E}_{\epsilon \sim \phi}[f(g(\epsilon, \lambda))]$. Thus, using this trick allows us to calculate the expectation with respect to a distribution ϕ that does not depend on λ. Suppose we use a Gaussian variational approximation $q(\theta \mid \lambda) = \mathcal{N}(\theta; \mu, \Sigma)$, with our variational parameters $\lambda = [\mu, \Sigma]$ we can reparametrize $\theta = C\epsilon + \mu = g(\epsilon, \lambda)$ such that $\Sigma = CC^T$ and $\phi(\epsilon) = \mathcal{N}(\epsilon; 0, I)$. This allows us to differentiate (28)

$$\begin{aligned} \nabla_\lambda L(\lambda) &= \nabla_\lambda \mathrm{E}_{\theta \sim q(\cdot \mid \lambda)} \left[\log p(D \mid \theta)\right] - \nabla_\lambda KL(q_\lambda || p(\cdot)) && (29) \\ &= \nabla_{C,\mu} \mathrm{E}_{\epsilon \sim \mathcal{N}(0,I)} \left[\log p(D \mid C\epsilon + \mu)\right] - \nabla_{C,\mu} KL(q_{C,\mu} || p(\cdot)) && (30) \end{aligned}$$

where $\nabla_{C,\mu} KL(q_{C,\mu} || p(\cdot))$ can be calculated exactly via automatic differentiation and the gradient of the expectation can be obtained via an unbiased stochastic gradient estimate. Finally, after training we can make predictions by calculating the approximate predictive distribution by sampling from the variational posterior $Q(\cdot \mid \lambda)$

$$\begin{aligned} P(y^* \mid x^*, D) &= \int P(y^*, \theta \mid x^*, D) \, d\theta && (31) \\ &= \int P(y^* \mid \theta, x^*) P(\theta \mid D) \, d\theta && (32) \\ &= \mathrm{E}_{\theta \sim P(\cdot \mid D)} \left[P(y^* \mid x^*, \theta)\right] && (33) \\ &\approx \mathrm{E}_{\theta \sim Q(\cdot \mid \lambda)} \left[P(y^* \mid x^*, \theta)\right] && (34) \\ &\approx \frac{1}{m} \sum_{j=i}^{m} P(y^* \mid x^*, \theta^{(j)}) && (35) \\ &\text{s.t. } \theta^{(j)} \sim Q(\cdot \mid \lambda) && (36) \end{aligned}$$

In short, we draw m sets of weights from the posterior and average the predictions of the network. When we assume Gaussian likelihoods, this approximate predictive distribution becomes a mixture of Gaussians. Using a Bayesian approach on the weights, we are now able to model the uncertainty of the model, also called *Epistemic Uncertainty*. By using the individual predictions from Eq. (35) as $\mu(x^*, \theta^{(j})$ and the calculated mean prediction $\bar{\mu}(x^*)$, we can compute the epistemic uncertainty:

$$\mathrm{Var}\left[\mathrm{E}\left[y \mid x, \theta\right]\right] = \frac{1}{m} \sum_{j=1}^{m} \left(\mu(x^*, \theta^{(j)}) - \bar{\mu}(x^*)\right)^2 \tag{37}$$

2.6 Recent Work in Image Denoising

Obtaining high-quality images in real-life settings is challenging due to physical limitations of the recording device or mechanical vibrations during acquisition, which eventually manifest themselves as random noise in the image. Several techniques have been proposed to remove the noise and maintain as many features as possible. Technically, image denoising is an inverse-problem where the goal is to recover a clean image $\boldsymbol{y}$ from a noisy observation $\boldsymbol{x} = \boldsymbol{y} + \boldsymbol{n}$. There are several categories of denoising algorithms used nowadays – both internal statistics methods as well as deep learning methods. Internal statistics methods do not need any training data and are based on hand-crafted priors. One well-known denoising algorithm is the *non-local means* (NLM) [11], which predicts pixel values based on an average of selected pixel values in the image. Block-Matching and 3D Filtering (BM3D) [12], which is also a widely used algorithm today, tries to find repeated and similar patches in an image, group them together, and filter them jointly after applying a suitable transformation. Unfortunately, the computational cost is quite high. These filters provide reliable frameworks when dealing with data with relatively litte noise but fail when confronted with low image quality. One reason is that these algorithms were mainly developed to deal with AGWN and can thus not cope with heteroscedastic noise. Due to the progress in machine learning, especially deep learning, many different approaches have been investigated in the area of image denoising with impressive results. In 2008, [48] first applied CNNs in the context of image denoising, where denoising is seen as a regression. The advantage of deep learning methods is that they implicitly learn the prior with the data at hand. The feed-forward deep convolutional neural network (DnCNN) [13] and the U-Net [20] are considered as state-of-the-art denoising networks. The idea of DnCNN is based on residual learning. Instead of learning the clean image, it attempts to predict the noise in every pixel. This allows the network to be trained for a variety of noise levels. The U-net is a fully-convolutional neural network with a very deep encoder-decoder architecture, with symmetric skip-connections between the two parts, making use of residual learning as well. The latter is used as comparison for our proposed method. Although deep CNNs have shown to provide great denoising capacities, they are strongly dependent on the training data and are therefore prone to overfitting. Furthermore, it is well-known that denoising networks can add or remove structures, which would or would not be present in the ground-truth image, respectively [16]. Such behaviour is undesirable in medical imaging and, in the worst case, could lead to misdiagnosis.

Recent research has turned the focus of denoising onto unsupervised training. Usually, ground truth images are not available – especially in medical imaging – and training the aforementioned networks thus becomes infeasible. [14] presented a method called Noise2Noise (N2N), which uses two noisy image pairs of the same subject to train its network. Their results are competitive with networks trained with clean images. It is assumed that the added noise has zero-mean and therefore the mean of multiple corrupted images of the same signal will result in the desired true signal, which is then expected to be predicted by the network. Obtaining two noise realisations of an image is difficult – it would need a static setup and no subject movement. Furthermore, in clinical X-ray investigations, dose requirements have to be fulfilled, making it difficult to collect two noisy images. Therefore, [15] uses only one noisy image for training by masking single pixel images in the input and predicting the noisy pixel value, calling their method Noise2Void (N2V). While the idea is similar to the N2N method, they assume that the true pixel signals are not conditionally independent from each other, while the noise is. Another unsupervised method is the Deep Image Prior (DIP) [17]. A network is optimised for a single image by using a random image as input and optimizing the network's weights (and the random image). With this method, the network successfully learns to retrieve the denoised image, while having a hard time recovering the noise. However, training a network for each image is computationally expensive and the user has to interrupt network training at the right time.

3 Methods

3.1 Projection Simulations

Due to the lack of available real data, we created *in-silico* breast phantoms from which single projections were obtained to simulate the standard mediolateral oblique view-projection (MLO-projections) in mammography screenings. The phantoms were generated using a 3D mask of a real breast, acquired with a breast CT scanner at the University Hospital in Zurich. With this, two binary masks were created to simulate the skin and the whole breast. To simulate the compression of the breast, a scaling was applied to the dimension in projection direction as well as a morphological erosion operation to deal with thickening skin generated by the scaling operation (see Appendix A Fig. 23). Eventually, the final binary masks consisted of $400 \times 1646 \times 1233$ voxels with a voxel size of 100 µm. To facilitate upcoming calculations for the models, the masks were zero padded to obtain squared image dimensions in the projections (i.e. $400 \times 1646 \times 1646$). The interior breast tissue was simulated using 50 randomly generated ellipsoids with different sizes, shapes, orientation, and position, followed by a threshold to differentiate between adipose and glandular tissue. The edges of the ellipsoids were used to create duct-like structures coming out the glandular tissue. Additionally, to simulate high scattering areas such as microcalcifications, a maximum of 10 randomly created balls with radii between 2 and 5 voxels were added to the preliminary phantom created by the ellipsoids. Then we multiplied the generated phantom with the previously created binary mask. The complete phantom is then immersed into water, because when using air as background material, the created projections showed very poor contrast. Mostly responsible for this were numerical problems in the dynamic range. Due to the high phase difference from background to breast, the largest part of the dynamic range is used for the differentiation of general breast and background, i.e. a very high pixel value difference between the two. This means that small differences in intensity in the projected breast, which consequently only take up a small part of the dynamic range, can be poorly represented and thus allow for hardly any contrast to be visible.

For the generation of absorption and phase projections, the phantom's distinct areas representing adipose, glandular, microcalcifications, and skin, were assigned to realistic attenuation coefficients μ [cm^{-1}] and phase shift coefficients ϕ [cm^{-1}]. The attenuation coefficients were obtained using the NIST X-ray Mass attenuation database [49] and [50] for water, skin, and adipose and glandular tissue, respectively. Similarly, for the phase shift coefficients, the decrement of the real part of the index of refraction δ was first collected using [50] and [51]. Using the known relation $\phi = 2\pi\delta/\lambda$ [31], where λ corresponds to the X-ray wavelength matching the design energy of the used mammogram (i.e. 26keV), the phase shift coefficients were calculated. The dark field image's phantom was created purely empirically and no physically meaningful values were provided. The dark field signal measures the small-angle scatter caused by the object. This signal depends not only on the object of interest, but also on the relative orientation, the direction of the X-rays, and the gratings. To get a physically correct projection, more complex algorithms would be required that take the distribution of the scattering into account. In this work, we have concentrated on conventional projection algorithms, which use forward operations that do not consider the scattering distribution. From Eq. (10), however, we know that the DF signal has an influence on the noise in the DPC image, thus resulting in our DPC noise modeling being less realistic if not simulated correctly. Therefore, having more sophisticated projection algorithms would allow for better noise simulation. Our phantom itself constituted of the edges of the ellipsoids and the microcalcifications, where a value of 0.6 and 0.9 was assigned. This was mainly performed to highlight scattering areas in a breast, where special focus was put on the microcalcifications.

Each of the generated phantoms (absorption, phase, dark-field) was further used to generate clean and noisy projections. For this, we used the ASTRA Toolbox projections [52] to generate the differential

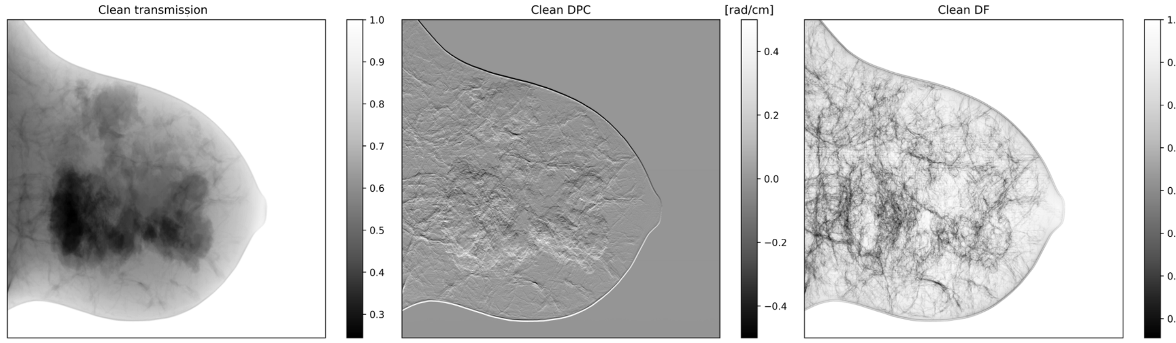

Figure 2: Transmission, differential phase, and dark-field projections without noise. Here, only the pixel entries of the transmission and differential phase channel correspond to physically meaningful values, while the dark-field projection is built purely empirically. We display the transmission signal instead of absorption as the former is the direct output of the Philips Microdose System.

phase, transmission, and dark-field image. The projections were generated as follows [31]

$$\varphi_s = \frac{\lambda d}{p_2} \frac{\partial}{\partial x} \int \phi(x, y, z)\, dz \tag{38}$$

$$T = \exp\left[- \int \mu(x, y, z)\, dz \right] \tag{39}$$

$$D = \exp\left[- \int \tau\sigma(x, y, z)\, dz \right] \tag{40}$$

where d is the G1-G2 distance and p_2 is the G2 period length chosen to match our constructed design. Furthermore, σ corresponds to the aforementioned empirical scattering areas with either 0, 0.6, or 0.9 for background, scattering edges, and microcalcifications respectively. τ is a scaling factor used to achieve a dynamic range similar to images obtained in experimental settings and was empirically set to 2.6. These obtained projections served as ground truth images.

For the generation of the noisy images, we simulated a PSC using flat-field data (intensity map I_0, visibility map V_0, and phase map φ_r) from previous experiments in 2012 (see Appendix A Fig. 22), since obtaining flat-field information from the Philips Mammogram is only possible via correct decoding keys, which were not provided for this work. With the collected projections from the phantoms and the flat-field data, we simulated the PSC as follows:

$$I_{s,k} = I_0 T \cdot [1 + V_0 D \cdot \cos(k + \varphi_r - \varphi_s)] \tag{41}$$

where s and r stand for signal with and without sample, and k is the k-th phase step uniformly sampled between 0 and 2π. Thus, $I_{s,k}$ represents the intensity value measured at the k-th phase step. Similarly, for the background PSC:

$$I_{r,k} = I_0 \cdot [1 + V_0 \cdot \cos(k + \varphi_r)] \tag{42}$$

Next, we scaled the intensity values to achieve an average photon count of 1000, 3000 and 5000, to simulate multiple noise levels. To simulate the detector quantum noise, we sampled from a Poisson distribution with mean $I_{s,k}$. With this, we performed simple Fourier analysis to retrieve the noisy projections corresponding to our ground truth images. An example of a generated clean projection set is depicted in Fig. 2.

3.2 Developed denoising algorithms

The proposed method picks up on the work of GI-Breast CT (GI-BCT) denoising [19] and was modified to better process the data at hand. It is a combination of multiple methods, namely collaborative filtering, transform learning, multi scale processing, explicit orthogonality, Bayesian learning, and standard convolution neural networks. The idea of collaborative denoising filtering consists of three steps: block matching, filtering, and aggregation as explained in the background section on the BM3D algorithm. In the block matching step, the algorithm searches for similar patches in the image and stacks them on top of each other to create a 3D block. Searching for similar patches comes with high computational costs and thus long denoising time. Next, the 3D blocks are transformed into another domain such as the wavelet domain or the discrete cosine transform (DCT) domain [12], assuming that the 3D blocks can be expressed as a linear combination of a few basis elements. Due to the similarity within and between the 3D blocks, the transformed blocks will be highly sparse. Thus, the noise can be well separated by performing thresholding or shrinkage of the obtained coefficients. Lastly, the blocks are transformed back to the original domain and aggregated together to generate the final denoised image.

Collaborative filters, such as the BM3D, have a good performance when dealing with images with low noise. However, they fail when applied to images corrupted with high noise amplitudes. In addition, aggregation of similar patches involves long computational times, especially for large images such as those found in mammography – they can have over 4000 pixels in each dimension. From Eq. (10) we know that the noise in the DPC channel depends on the flats as well as the transmission and DF image, which means that each pixel follows a different standard deviation. The noise is thus heteroscedastic, which makes it hard to be handled by these filters, since they were designed to deal with AGWN. This applies even more so when the uncertainties deviate strongly, where a falsely chosen parameter for the BM3D either leads to signal loss or remaining noise.

Instead of using fixed, hand-designed operators such as the Wavelet transform or DCT, it can be more beneficial to learn this operator with the available data. This method is called transform learning and aims to learn a transformation matrix where the transformed data is highly sparse [53], allowing for shrinking or thresholding operators in the transformed space – equivalent to the collaborative filtering step. [19] uses a combination of these methods in a supervised setting combined with multiscale image processing. For the transformation matrix an explicit orthogonality constraint has been adapted to provide high model robustness and interpretability. Instead of the block-matching step from the collaborative filtering, overlapping patches from the whole image are stacked on top of each other, independent of their similarity. This dramatically increases computational time. Next, the transformation is carried out with a learnable transformation matrix and thresholded with learnable thresholding coefficients. Furthermore, the images are processed across four different scales to ensure noise removal across a wide frequency band.

Using the ground-work from [19], we propose a modified version of the model, adding a Bayesian view and a convolutional network component onto it. The Bayesian view allows to model uncertainty in the prediction, making it possible to question the prediction and see whether noise might still be in the image. This is crucial in medical imaging where artefacts can lead to misdiagnosis.

The upcoming sections are organized as follows: We will first introduce the theory of the model in [19] in a supervised way, while simultaneously introducing the developed Bayesian view. It is then combined with a state-of-the-art CNN architecture. Finally, the developed method is modified to be compatible with unsupervised training.

3.3 Probabilistic Interpretable NonexpanSIve Data-Efficient Network

Let $X_n \in \mathbb{R}^{N\times N}$ be the noisy input image and $X_c \in \mathbb{R}^{N\times N}$ the corresponding clean image, where for simplicity we assume a quadratic image with length N. First, X_n is divided into overlapping patches to facilitate the computational costs by using smaller transformation matrices. Here, stripe $S = 2$

was empirically determined to be optimal. We denote the stacked flattened overlapping patches as $P \in \mathbb{R}^{M \times M \times N_p^2}$, where M depends on the input image size N, patch size N_p, and stride S in horizontal and vertical direction as $M = \frac{N-N_p}{S} + 1$. This tensor is then multiplied via an Einstum sum with the orthogonal transform matrix $Q \in \mathbb{R}^{N_p^2 \times N_p^2}$ to generate the transformed patches $\hat{P} \in \mathbb{R}^{M \times M \times N_p^2}$

$$\hat{P} = QP \tag{43}$$

Here, Q is created via the Cayley transform [54] to ensure orthogonality. In detail, let A be any skew-symmetric matrix (i.e. $A^T = -A$) and I the identity matrix, then the orthogonal matrix Q can be computed as $Q = (A - I)(A + I)^{-1}$. Furthermore, a skew-symmetric matrix can in turn be constructed by an arbitrary matrix B via $A = B - B^T$. Thus, B is our trainable matrix, which will be transformed into a orthogonal matrix Q.

This is where our Bayesian perspective comes into play: Let us assume our flattened matrix $\hat{B}$ is sampled from a multivariate Gaussian distribution $\hat{B} \sim \mathcal{N}(\mu, \Sigma)$ where $\mu \in \mathbb{R}^{N_p^4}$ and $\Sigma = \operatorname{diag}(\sigma_B)$ are the mean and covariance matrix and σ_B is a vector in $\mathbb{R}^{N_p^4}$. We used a diagonal covariance matrix, since simulating a complete covariance matrix would dramatically increase the computational burden of our system. It is important to mention here that this leads to underestimation of the uncertainty in the model, which has to be taken into account when evaluating the images. Thus, our trainable parameters were the mean μ and covariance vector σ_B of the distribution. We approximate the real intractable posterior distribution $P(\hat{B} \mid D)$ with our variational posterior $Q(\hat{B} \mid D, \lambda)$ constructed as a Gaussian multivariate. To allow for efficient gradient calculation and sampling, we rely on the reparametrization trick:

$$\hat{B} = \mu + \epsilon \odot \sigma_B \tag{44}$$

where $\odot$ denotes the element-wise multiplication and $\epsilon \sim \mathcal{N}(0, I)$. Finally, the vector $\hat{B}$ has to be reshaped to a matrix B with size $N_p^2 \times N_p^2$ to be used as our input matrix for the Cayley transform.

After transformation, the individual entries are filtered using an approximation of the hard thresholding on the coefficient magnitudes. Here, a steep sigmoid function is used to allow the threshold parameter to be trainable.

$$T = \frac{1}{1 + \exp\left(-\left(|\hat{P}| - \Gamma\right) \cdot \rho\right)} \tag{45}$$

ρ is the steepness of the sigmoid function and was set to 100 to approximate the box function sufficiently well. Furthermore, the threshold parameter Γ was initially set to 10^{-6} to ensure that all coefficients are kept at the start of the training. The transformed patches are then multiplied with T element-by-element to represent the filtered coefficients of our image and finally transformed back to the original image domain via an Einstein sum to generate the filtered image.

$$P_D = Q^T [\hat{P} \odot T] \tag{46}$$

The described steps are then repeated n times, representing an iterative denoising algorithm where each transform domain thresholding step improves the quality of the image. It can also be seen as a layer in a CNN. Importantly, however, this operation follows a clear mathematical reasoning whilst in CNNs, the layer represents an unconstrained forward operator followed by a non-linear operation. Eventually, the filtered patches are put back to their original position to generate the denoised image X_D. To avoid border artefacts, we used Tukey windows [55].

3.3.1 Image decomposition and reconstruction

To capture a wider noise frequency range, we denoise the images at multiple scales. After going through the transform learning layer, we downsample the output image $X_{TL_1} \in \mathbb{R}^{N \times N \times 2}$ by first convolving it with a Gaussian kernel of radius $r = 3$ and $\sigma = 0.667$ to blur the image and next apply an average pooling to downsample it by a factor of 2. σ is purposefully chosen so that 99% of the downsampled area is covered by the kernel with $r = 3$ representing a sufficiently large coverage. The downsampled image is then again fed to a collaborative filtering layer and the process is repeated m times, generating m sequentially denoised images $X_{TL_i} \in \mathbb{R}^{\frac{N}{2^{i-1}} \times \frac{N}{2^{i-1}} \times 2}$ for $i \in \{1, \dots, m\}$. It is important to note that the complete pipeline happens sequentially. Denoising it in parallel would be possible as well, however, this would lead to redundancies since we would denoise the same frequency at multiple scales.

After denoising at multiple scales m, the denoised images X_{TL_i} are used to reconstruct the predicted image. For this [19] lends its ideas from [56]. At each scale m, the low- L_m and high-frequency components H_m are separated. The low-frequency component L_m is obtained in a similar fashion as the downsampling operation from before – smoothing and downsampling, which will remove the high-frequency components – followed by upsampling and smoothing. The high frequencies, on the other hand, are obtained by subtracting the low frequencies from the image. By combining the low frequencies from the lower scale and the higher frequencies from the upper scale, we can assemble our denoised image $X_{m,denoised}$. Starting at the lowest two scales and continuing it iteratively until the top generates our final denoised image $X_{denoised}$.

3.3.2 Loss function and optimization

Due to the Bayesian perspective on the weights of our model, we resorted to our variational inference loss in Eq. (28) and used the reparameterization trick to be able to calculate the gradient. We modelled the likelihood of our data $p(D \mid \theta)$ as a Gaussian distribution, where $\theta = \{\mu_{L_1}, \sigma_{L_1}, \Gamma_{L_1} \dots \Gamma_{L_N}\}$ represents the trainable parameters in the model. Modifying the cost function in Eq. (28) from a maximum to a minimization problem and inserting the Gaussian distribution for the likelihood results in:

$$\begin{aligned} L(\lambda) &= -E_{\theta \sim q(\cdot|\lambda)} \left[\log p(y \mid x, \theta)\right] + KL(q_\lambda || p(\cdot)) && (47) \\ &= -E_{\epsilon \sim \mathcal{N}(0,I)} \left[\log p(y \mid x, C\epsilon + \mu)\right] + KL(q_\lambda || p(\cdot)) && (48) \\ &\approx -\frac{1}{m} \sum_{i=1}^{m} \log p(y \mid x, C\epsilon_i + \mu) + KL(q_\lambda || p(\cdot) && (49) \\ &= \frac{1}{m} \sum_{i=1}^{m} ||f(x, \theta_i) - y||_2^2 + KL(q_\lambda || p(\cdot) && (50) \end{aligned}$$

where again we used Monte Carlo sampling to approximate the expected value. Here, $f(x, \theta_i)$ represents the neural network with the input being the noisy image x and the sampled parameteres θ_i. Although in the equations the reparameterization trick seems to only be applied for one layer, it counts for every single layer where a variational distribution is used. Only the B matrices are modelled as distributions, while the threshold parameters Γ are modelled as point masses, – i.e. deterministic. In general, the Γ could also be modelled as a distribution, yet we were not able to generate converging and satisfactory results with it. Intuitively, every single forward pass through the model constitutes of different parameters and only the means and standard deviations – next to the thresholds Γ – are optimized. For the KL-Divergence term we assumed a Gaussian distribution on the prior of the weights $p(\cdot) = \mathcal{N}(\mu_P, \sigma_P^2 I)$. Usually, it is not possible to know the distribution of the weights a priori, which has become a challenging problem and active area of research [57] [58]. We draw on the work from [59], where the idea is to use a pretrained deterministic model as prior knowledge for the prior distribution. For this, we trained the INSIDEnet model from [19] until convergence on our data and used the optimized weights as the mean in our prior distribution μ_P. Still, the uncertainty σ_P in our prior remains a hyperparameter, which has to be tuned by the user. Since we initially modelled our variational family as a multivariate Gaussian distribution with diagonal covariance matrix, we could calculate the KL-loss per transformation layer in a closed form [60]:

$$KL(q_\lambda||p(\cdot)) = \sum_{i=1}^{N_p^2} \log \frac{\sigma_P}{\sigma_i} + \frac{1}{2\sigma_P^2}\left[(\mu_i - \mu_{P,i})^2 + \sigma_i^2 - \sigma_P\right] \tag{51}$$

μ_i and σ_i are the entries of the vectors containing the mean and standard deviation of the corresponding variational family in a layer, μ_P the weights from the pretrained model, and σ_P is a single scalar value representing the standard deviation from the prior weight distribution. Intuitively, putting a prior on the weights is equivalent to a regularization, ensuring that the weights do not deviate too much from the prior.

For the optimization, we relied on the Adam optimization algorithm [61] and used a batch size of 1 to generate the unbiased stochastic gradient estimates. Finally, inserting Eq. (51) into Eq. (50) and multiplying a scalar factor to the KL-loss, we got our final loss for a single training step as:

$$L(\alpha, \lambda, x, y) = \frac{1}{N}||f(x,\lambda) - y||_2^2 + \frac{\tau}{B}KL(q_\lambda||p(\alpha)) \tag{52}$$

With $\alpha = \{\mu_P, \sigma_P\}$ representing the prior parameters. Here, we changed the L_2 loss into the MSE and divide the KL-loss by the number of batches/training samples, to ensure that the KL-Divergence is only calculated over the complete dataset once per epoch. τ is a regularization parameter as it is known from Ridge regression [35]. Our loss is optimized over the variational parameters λ and threshold parameters Γ. We trained the network until convergence, i.e. until the loss on the validation set did not improve anymore.

3.4 Collaborative Pyramid Bayesian Neural Network CP-BNN

Continuing on the work from the previous chapter, we implemented a combination of collaborative transform filtering and standard convolution layers. While the INSIDEnet model allows to simultaneously denoise absorption and phase, it is not possible to fuse both informations into one image. Therefore, using convolutional layers after the collaborative filtering, we could fuse both channels and give the DPC channel more weight. The final architecture of the model is depicted in Fig. (3). The model takes two images as input – DPC image and absorption image $X_n \in \mathbb{R}^{N\times N\times 2}$. These images are then fed into the collaborative filtering layers. To capture noise at lower frequencies, we applied the same rationale for the downsampling as in the INSIDEnet described in chapter 3.3.1. As seen in Fig. (3), the described pipeline resembles the encoder in the U-Net architecture, while providing interpretability in each single step of the encoder.

At each scale, the denoised outputs X_{TL_i} that the collaborative filters produce are additionally fed to a convolution block. These blocks consist of two 2D deterministic convolution layers followed by a 2D variational layer – where the convolution kernel is sampled from a trained distribution following the rationale from chapter 2.5 – and a 2D-convolution layer at the end. The idea, on the one hand, was to fuse the informations from both channels and on the other to resemble the skip-connections from the U-Net at lower scales. Next, the downscaled information has to be brought up to the original dimensions. This happens via an upsampling layer, where first the images are upsampled to twice the size in dimension using a bilinear interpolation and afterwards blurred again as in the downsampling layer. This information is then fed to a concatenation layer, where the information from the higher scale is concatenated with the upsampled information from the lower scale. This resembles the composition of low and high frequencies components from chapter 3.3.1. However, here, the operation does not have any constraints and we allow the network to learn the fusion operation from lower to upper scales. The images are then again fed

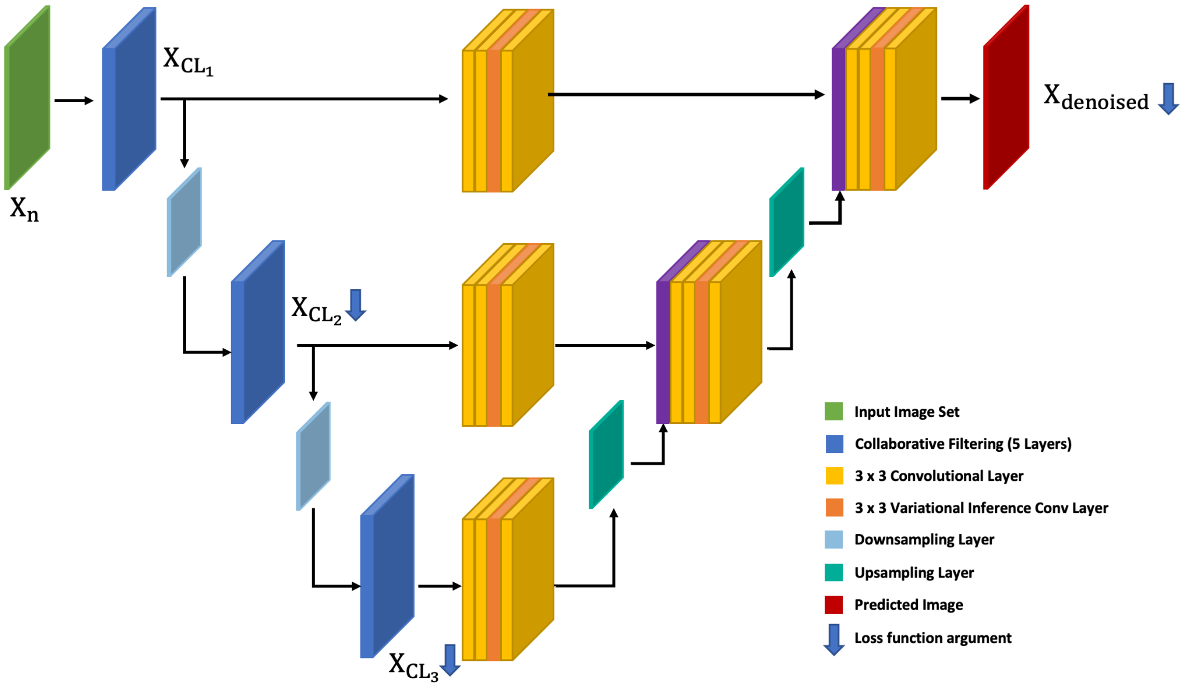

Figure 3: Overview of the collaborative pyramid Bayesian neural network denoising algorithm. The input image is first denoised sequentially at every scale using the transform learning filter. Afterwards, the images at lower scales are upsampled and concatenated to fuse information from high frequencies and low frequencies implicitly learned by the convolutional layers itself. With the variational inference layer, the epistemic uncertainty of the model can be analysed, which allows to see whether more careful consideration has to be given in the output image.

to a convolution block. This procedure is repeated until reaching the original image dimensions where eventually the output of the last convolutional layer is the predicted image.

The model was trained on a loss function consisting of three terms splitted across different sections of the model. We used the mean squared error (MSE) on the full-resolution image as well as on the lower scale layers after the collaborative filtering. The latter ensured that the lower scaled collaborative filtering layers learn the transformation matrix properly and put weight on early denoising effects inside the model. It acted as a regularization on the transformation matrix and prevented from learning the identity matrix, which may happen due to vanishing gradients or imbalances in the role distribution of the network, which could give more value to the expressiveness of the convolutional layers. Training the model without deep supervision resulted in a similar outcome. However, when analyzing the encoder, we found that the filtering was more adapted to the kernels of the convolutional layer, while our goal was to keep the transformation matrices similar to those in the INSIDEnet model. Next, due to the Bayesian view on some convolutional layers, the KL-Divergence had to be considered between the trained posterior and the prior as described in Eq. (28). Finally, our loss function was as follows:

$$\mathcal{L} = \frac{1}{N^2}||X_{\text{denoised}} - X_{\text{clean}}||_2^2 + \rho \sum_{i=2}^{3} \frac{2^{(i+1)}}{N^2}||X_{\text{TL}_i} - X_{\text{clean}_i}||_2^2 + \frac{\tau}{B} KL(q_\lambda||p(\cdot)) \tag{53}$$

We used the Adam optimization algorithm with an exponentially decaying learning rate and trained the model with a batch size of 1 until convergence.

3.5 Unsupervised denoising

Most of the work presented in the literature for deep learning denoising focuses on supervised learning, where clean and noisy images are available. Yet, obtaining those image pairs in is generally difficult and becomes even harder in medical imaging, where dose requirements have to be met. Additionally, most networks are trained with AWGN where a single standard deviation is used for the noise generation. However, this simple aspect is not valid in medical images, especially in DPC images. Thus, we focused on trying to train the network with only noisy images. For this, we drew our inspiration from [62], where the idea is to add simulated noise to the already noisy image, which is statistically close to the inherent noise of the image itself. They called this method *Noise as Clean (NAC)*.

Training a supervised network f_θ is equivalent to minimizing an empirical loss function $\mathcal{L}$ over the parameters θ describing the network. If we now state that the probability that the clean and noisy image pairs (y_i, x_i) occur with probability $p(y_i, x_i) = p(x_i)p(y_i \mid x_i)$, the optimization can be written as:

$$\theta^* = \arg\min_\theta \sum_i p(y_i, x_i)\mathcal{L}(f_\theta(y_i), x_i) \tag{54}$$

$$= \arg\min_\theta \sum_i p(x_i)p(y_i \mid x_i)\mathcal{L}(f_\theta(y_i), x_i) \tag{55}$$

$$= \arg\min_\theta \mathbb{E}_x \left[\mathbb{E}_{y|x}\left[\mathcal{L}(f_\theta(y), x)\right]\right] \tag{56}$$

Next, we assume that the mean and the variance of the image intensity is much higher than that of the noise. i.e. $\mathbb{E}[x] \gg \mathbb{E}[n_o]$ and $\mathrm{Var}[x] \gg \mathrm{Var}[n_o]$. By assuming additive noise ($y = x + n_o$), the expectation of the corrupted image should have a similar value as the clean image.

$$\mathbb{E}[y] = \mathbb{E}[x + n_o] = \mathbb{E}[x] + \mathbb{E}[n_o] \approx \mathbb{E}[x] \tag{57}$$

Now, we add simulated noise n_s, which has similar statistics as the observed noise n_o – i.e. $\mathbb{E}[n_s] \approx \mathbb{E}[n_o]$ and $\mathrm{Var}[n_s] \approx \mathrm{Var}[n_o]$ – to the observed image y, which generates a new image $z = y + n_s$. From Eq. (57) it holds that $\mathbb{E}[z] \approx \mathbb{E}[y]$. By the *Law of Total Expectation* [63], it follows:

$$\mathbb{E}_y\left[\mathbb{E}_z\left[z \mid y\right]\right] = \mathbb{E}[z] \approx \mathbb{E}[y] = \mathbb{E}_x\left[\mathbb{E}_y\left[y \mid x\right]\right] \tag{58}$$

With this as well as Eq. (56), we see that the optimal parameters change little when adding similar noise to the corrupted image, making it possible to train a network with only noisy observations, if the noise statistics are known.

$$\theta^* = \arg\min_\theta \mathbb{E}_x \left[\mathbb{E}_{y|x}\left[\mathcal{L}(f_\theta(y), x)\right]\right] \tag{59}$$

$$\approx \arg\min_\theta \mathbb{E}_y \left[\mathbb{E}_{z|y}\left[\mathcal{L}(f_\theta(z), y)\right]\right] \tag{60}$$

Fortunately, the noise propagation in grating interferometers have been studied in previous years, which allowed us to adapt the NAC rationale from before to our data and methods. With Eq. (7) - (10), we were able to generate noise realisations similar to the ones inherent in the simulated image. Since we only used the absorption and differential phase channel, we only simulated noise for these channels. To check whether our simulations cope with the theory described in [10] and [32], we calculated the standard deviations according to Eq. (7) - (10), and sampled a new noisy image z from it, and compared it numerically with the original noisy image y. We performed this by comparing the MSE from the clean to the original noisy image and the MSE from the simulated noisy image to the original corrupted image. The simulated noise for the absorption channel was sampled from a Gaussian distribution with mean 0 and pixel wise variance according to Eq. (7) and added to the corrupted image y_T to generate

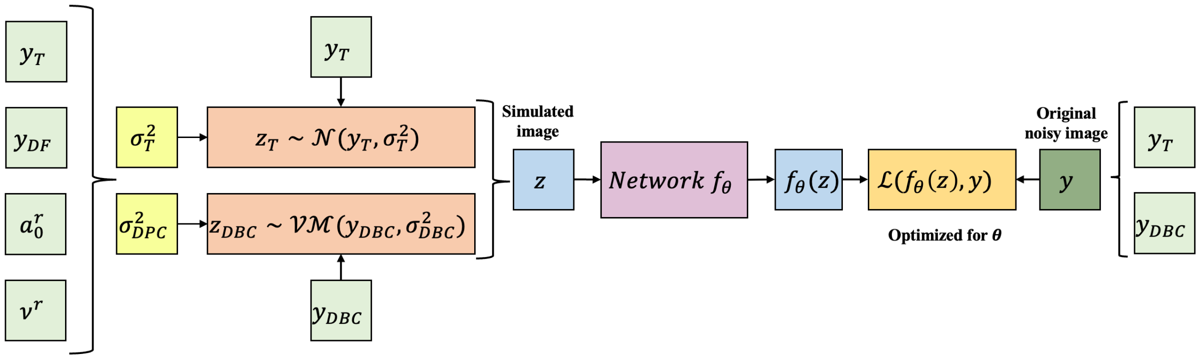

Figure 4: The NAC framework takes as inputs the noisy images from all three channels $\{y_T, y_{DPC}, y_{DF}\}$ as well as the flat images from the mean intensity and visibility $\{a_0^r, v^r\}$. With these, uncertainties in the images can be approximatively calculated and a new noisy image can be sampled. Eventually, the network is trained in a supervised fashion by using the original noisy images as target and the simulated noisy images as input

$z_T = y_T + n_T$ with $n_T \sim \mathcal{N}(0, \sigma_T^2)$. Noise distribution in X-ray absorption images can be modelled with a Poisson-Gaussian distribution [64]. To facilitate our calculations, we sampled our synthetic noise from a Gaussian distribution. For the differential phase, we performed the same procedure but sampled the new image from a Von-Mises distribution with mean the original corrupted image y_{DPC} and a measure of concentration $\kappa = 1/\sigma_{DPC}^2$, leading to $z_{DPC} \sim \mathcal{VM}(y_{DPC}, \kappa)$. Sampling from a Von-Mises distribution leads to an image in $[-\pi, \pi]$, so no phase wrapping can occur. We found that the sampled noise using Eq. (10) was more in agreement – in terms of MSE – with our generated simulations than Eq. (8).

The stated uncertainties are all calculated using the clean signal of every channel. However, in practice it is not possible to gather noise-free images. Therefore, we resorted to the noisy images for the calculations of the uncertainties. Yet, due to the disagreement in statistics coming from using the noisy signals instead of the clean, the user may or may not tune the strength of the uncertainties with a constant factor, such that the sampled uncertainties are more in agreement with the inherent noise of our simulations. This becomes even more important when dealing with relatively higher noise amplitudes, occuring when using a lower average photon count. With the newly generated noisy images, we could then train the model in the standard supervised fashion as stated in Eq. (53). The general pipeline is depicted in Fig. 4.

3.6 DPC integration and stripe noise removal

To collect the phase contrast image, we needed to integrate Eq. (38) in the direction perpendicular to the grating lines. An integration can be seen as a low-pass filter since the operation in the image domain is a division by the spatial frequency in the Fourier domain [65]. Hence, the image gets smoothened in the integration direction, whereas perpendicular to it, high frequencies remain unfiltered and lead to amplified noise, which results in strong stripe artefacts along the direction of integration. The intuition for their existence can also be understood from a purely statistical view, which we demonstrate here on AWGN. Let $X \in \mathbb{R}^{N \times N}$ be a purely noisy image, where each pixel is sampled independently and identically distributed (i.i.d.) from a Gaussian distribution $\mathcal{N}(0, \sigma^2)$. For our analysis, we create, a purely noisy image with dimensions 2000 × 2000 sampled from a Gaussian distribution with $\mu = 0$ and $\sigma = 0.3$ as depicted in Fig. 5 (left). Without loss of generality we integrate our image in y-direction using a Riemann-sum with constant step-size – a different step-size would only result in different scaling.

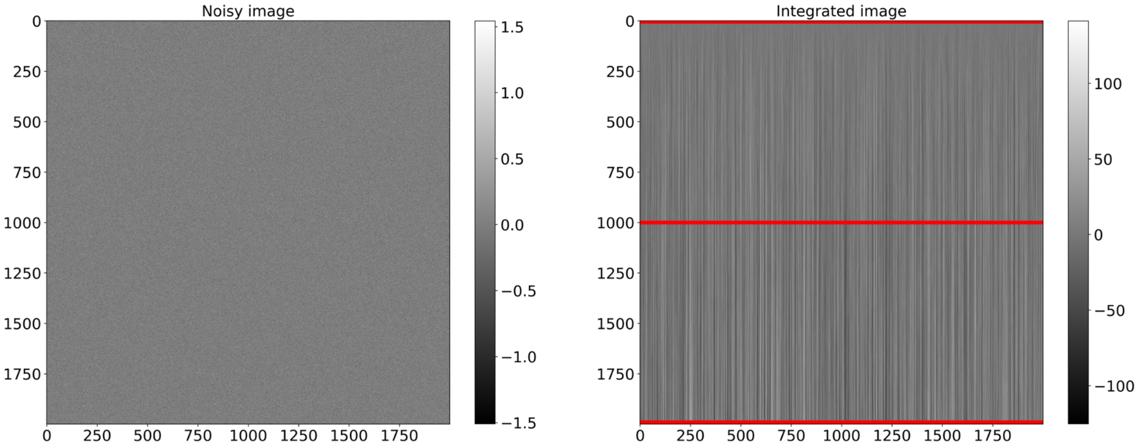

Figure 5: The left image depicts pure noise sampled from a Gaussian distribution with $\mu = 0$ and $\sigma = 0.3$. On the right is the integrated image with the vertical stripe artefacts. In red are the analysed rows corresponding to the histograms in Fig. 6

Therefore, the integrated image I at position (x_i, y_j) is computed as follows:

$$I(x_i, y_j) = \sum_{k=1}^{j} X(x_i, y_k) \tag{61}$$

Since every pixel is i.i.d, we fix a column i in the noisy image and only do our analysis of the stripes over this column. For simplicity, we will omit the second dimension and write our noisy image as $X(y_j)$. When looking at the integrated image in Fig. 5 (right), it can be seen that the stripes become stronger along the Riemann-sum direction. This can be explained via simple statistical calculations. As our pixels are i.i.d., we can calculate the expectation value and variance at a depth j as follows:

$$\mathbb{E}\left[I(y_j)\right] = \mathbb{E}\left[\sum_{k=1}^{j} X(y_k)\right] = \sum_{k=1}^{j} \mathbb{E}\left[X(y_k)\right] = 0 \tag{62}$$

$$\operatorname{Var}\left[I(y_j)\right] = \operatorname{Var}\left[\sum_{k=1}^{j} X(y_k)\right] = \sum_{k=1}^{j} \operatorname{Var}\left[X(y_k)\right] = \sum_{k=1}^{j} \sigma^2 = j\sigma^2 \tag{63}$$

In Eq. (63), we use the fact that our pixels are i.i.d. such that the covariance terms $\operatorname{Cov}(X(y_i), X(y_l)) = 0$ for $\forall i \neq l$. Therefore, we can sum up the individual variances. We expect the intensities of the stripes to be zero, however, the intensity range becomes larger, due to the increasing variance. Thus, we see stripes with larger absolute intensity values. After integration we choose three rows and analyse their intensity distribution. If we now plot the histogram of the intensity values of the three rows of Fig 5 (right) and fit a Gaussian curve on the data provided, the values match our calculation – with small errors (see Fig. 6). However, when dealing with heteroscedastic noise the rows will not follow a Gaussian distribution anymore, since every single entry in the row has a different variance. This complicates both the analysis as well as the stripe removal.

3.6.1 Combined Wavelet Fourier Filtering

Knowing the origin of the stripes demonstrates how important denoising in the DPC channel is. Little noise already results in a striped integrated image. To compute stripe-free images would require a completely noise free DPC image, which, depending on the noise level, can be difficult to achieve. Thus, we proposed to use our denoising networks as a first preprocessing step to collect integrated noisy images

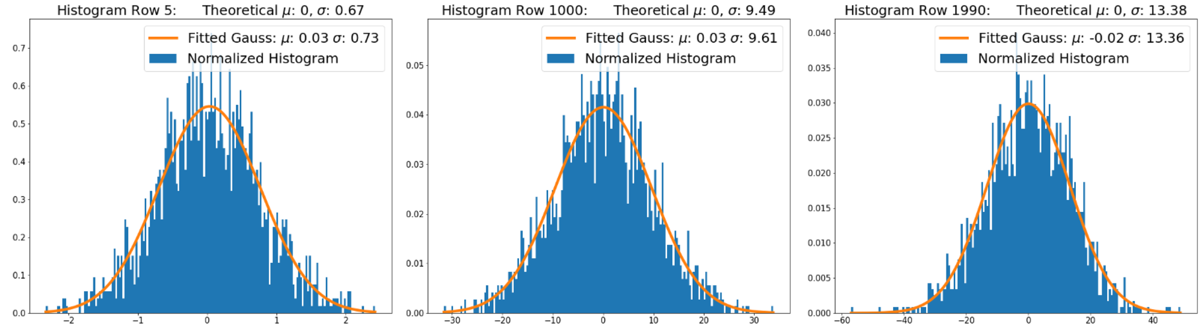

Figure 6: Normalized intensity histogram of each row and the fitted Gaussian probability density function with the fitted and theoretical parameters. Note, that the scales are different in each histogram, which dependent on the calculated and fitted standard deviation depicted in every image.

with significantly less stripes and – even more crucial – more remaining image features compared to integrating our noisy images directly. To tackle the problem of stripe-artefacts, we used a dedicated algorithm called combined wavelet Fourier filter (WFF) [21]. The WFF was mainly designed to deal with ideal stripes, i.e. vertical stripes that have a constant offset of arbitrary width over the whole image. Yet, the occurrence of ideal stripes in integrated DPC images is unlikely. Due to the randomness and the heteroscedasticity of the noise, the generated stripes resemble more fluctuating stripes, monotonic increasing / decreasing stripes, or even partial stripes, where they are only visible at some parts of the image. We could nonetheless show that the WFF is also able to remove imperfect stripes, if the right parameters are chosen.

In Fourier and numerical wavelet analysis, a discrete signal $f(t)$ can be approximated by a set of basis functions $\Gamma_n(t), n \in \{1, \ldots, N\}$, so that $f(t) \approx \sum_n a_n \Gamma_n(t)$, where these basis functions are orthogonal to each other. Decomposing a signal into orthogonal basis functions allows to group and modify specific structural properties of the image. A single step of 2D-discrete wavelet transformation decomposes an image $f(x, y)$ into a set of four different coefficient bands by letting the image pass successively through a series of filters in horizontal and vertical direction. It splits the signal using high-pass (H) and low-pass filter (L) into high frequency and low frequency components [21]. The high frequency components are called the detail coefficients and are represented as c_h, c_v, c_d, the horizontal, vertical, and diagonal detail bands, respectively, created as HL, LH, and HH. The low frequency components are the approximation coefficients a and generated by passing them through both low-pass filters in either direction, i.e. LL. In dyadic, decimated wavelet transforms the split into low and high frequency parts leads to a signal with half the coefficients. To increase the frequency resolution, the decomposition is repeated N times over the approximation coefficients. Thus, the 2D wavelet representation of a signal is a set of coefficients:

$$\mathcal{W}_\Psi(x, y) = \{a_{L,m,n}, c_{h,l,m,n}, c_{v,l,m,n}, c_{d,l,m,n}\}, l \in \{1, \ldots, N\} \tag{64}$$

$\mathcal{W}_\Psi$ represents the wavelet decomposition with wavelet set Ψ and m, n are the entries/coordinates in the coefficients at level l.

Due to the coefficient partitioning, when the wavelet transformation is applied to the striped images, the information from the stripes is exclusively contained in the vertical coefficients $c_{v,l,m,n}$ and the final approximation coefficient from the low frequency band $a_{L,m,n}$. Since we fractionate the signal dyadically, the frequency band of each successive vertical detail band consists of dyadically decreasing focal frequencies. Hence, the stripe information at each level is dependent on the spatial frequency spectrum of the stripes in horizontal direction, which is directly linked with its width. Therefore, the highest decomposition N is directly correlated with the maximum expected stripe width.

Having the information from the stripes now stored exclusively in the vertical detail band, we can now use the FFT to filter them. When performing the FFT $F(\hat{x}, \hat{y})$ on an ideal striped image $f(x, y)$ the complete frequency information of them is stored on the $\hat{x}$-axis – i.e. $\delta(\hat{y})$ Dirac delta functions at all $\hat{x}$ – and no frequency components in $\hat{y} \neq 0$ stem from the stripes. Consequently, by eliminating the information stored on the $\hat{x}$-axis, the stripes would be erased in the back-transformed image. However, ideal stripes are unlikely. Thus, a simple approach would be to apply a bandpass filter in the Fourier domain around $\hat{y} \approx 0$. This can be done with a Gaussian function:

$$g(\hat{x}, \hat{y}) = 1 - \exp(-\frac{\hat{y}^2}{2\sigma^2}) \tag{65}$$

The width of the filter in $\hat{y}$-direction is determined by the standard deviation σ. It takes the expected deviation from the vertical of the stripes in x-direction into account and is selected accordingly. Thus, multiplying the Gaussian filter with the vertical detail coefficients will eliminate the stripes stored in it. Filtering the coefficients on every level N will generate our filtered wavelet coefficients $\tilde{c}_{v,l,m,n}$, which can then be used to reconstruct the final destriped image (see. Alg. 1).

$$\tilde{f}(x, y) = \mathcal{W}_\Psi^{-1}(C) \qquad \text{where} \qquad C = \{a_{L,m,n}, c_{h,l,m,n}, \tilde{c}_{v,l,m,n}, c_{d,l,m,n}\}, l \in \{1, \dots, N\} \tag{66}$$

Algorithm 1 Wavelet-FFT Filter

```
Input: Noisy Image $X$, $\sigma$, wavelet basis $\Psi$, Number of levels $N$
Output: Destriped image $\tilde{X}$
  for $n \leftarrow \{1, \dots, N\}$ do
      $\{X, c_{h,n}, c_{v,n}, c_{d_n}\} = \mathcal{W}_\Psi(X)$
  for $n \leftarrow \{1, \dots, N\}$ do
      $\hat{c}_{v,n} = \mathcal{FT}(c_{v,n})$
      $g(\hat{x}, \hat{y}) = 1 - \exp(-\hat{y}^2/2\sigma^2)$
      $\hat{c}_{v,n} = \hat{c}_{v,n} \odot g(\hat{x}, \hat{y})$          ▷ Element-wise multiplication
      $c_{v,n} = \mathcal{FT}^{-1}(\hat{c}_{v,n})$
  $\tilde{X} = X$
  for $n \leftarrow \{N \dots, 1\}$ do
      $\tilde{X} = \mathcal{W}_\Psi^{-1}(\{\tilde{X}, c_{h,n}, c_{v,n}, c_{d_n}\})$
  return $\tilde{X}$
```

In short, the algorithm consists of three distinct parts. First, the wavelet composition is calculated by recursively splitting the high-frequencies components and using the low-frequencies components as a new input for the wavelet transformation. Next, the vertical detail coefficients are Fourier transformed and bandpassed on all decomposition levels to generate the filtered coefficients. Lastly, the destriped image is reconstructed using the filtered coefficients and the inverse wavelet transformation.

4 Results

To evaluate the effectiveness of the described algorithms, we performed a comparative study on the simulated mammographic projections. We compared the algorithms to a deep CNN, namely the U-Net, as well as to the classical state-of-the-art BM3D filter. Furthermore, we analysed the performance of the deterministic INSIDEnet in comparison to the probabilistic INSIDEnet (P-INSIDEnet) and compared the epistemic uncertainty of the CP-BNN and P-INSIDEnet. The study was performed on supervised trained models as well as on unsupervised trained models using the NAC method. Lastly, we performed destriping of the integrated predicted denoised DPC images using the wavelet-FFT filter. All deep learning models were implemented in Tensorflow 2.1 [66] and trained on a NVIDIA Titan RTX GPU with 24GB of memory.

4.1 Supervised Denoising: Comparative Study

Before entering the denoising algorithms, the clean projections – absorption and differential phase – were scaled to be within $[0, 1]$. The noisy counterparts were scaled identical to the clean projections. Each two channels were then stacked together to create the 2D multichannel images. To prevent biased training, these images were randomly shuffled in each epoch. We trained all models on 440 image pairs, validated them on 110 and tested them on 64. The images were simulated with an average photon count of 1000 to simulate a high noise level. The INSIDEnet models and the U-Net model processed the differential phase and absorption images jointly, while the CP-BNN, although taking both channels as input, only predicted a denoised DPC image, thus presenting with more features per data point. The deterministic INSIDEnet predicted on three different scales (i.e. $m = 3$) with patch size $N_p = 8$ and $n = 5$. We trained the model using the MSE loss until the validation loss did not improve anymore. We then used this model as a prior for the probabilistic INSIDEnet with the trained B-matrices as our prior mean and a prior standard deviation of 10^{-3}. The standard deviation was chosen by performing various tests, which compared the performances of the training, and selecting the one with the best results. The P-INSIDEnet was then trained using the loss function from Eq. (52), where B was set to 440 to meet the number of training images, since we were training with a batch size of 1 and $\tau = 0.01$. Our CP-BNN model was constructed as seen in Fig. 3 and trained on the loss function from Eq. (53) with a multivariate normal distribution as prior over the weights. The U-Net model evaluated features on five different scales, leading to a total model with 510338 parameteres and trained with a MSE loss. The parameter number was chosen to be similar to the parameter numbers of our other models. We then used the Adam optimization algorithm ($\beta_1 = 0.9, \beta_2 = 0.999$) with an initial learning rate of 0.0001 with exponential decay on all models for training. Lastly, we compared the obtained images from our machine learning models to the state-of-the-art BM3D model. An overview of all models and their parameters is provided in Tab. 1. It can be immediately determined that the deep learning models are at least 300 times faster than the BM3D.

We only show the denoising results of the DPC channel, as this is the channel this work focused on.

	Trainable Parameters	Prediction Time/Image
Probabilistic INSIDEnet	493455	0.28s (× samples)
INSIDEnet	247695	0.26s
CP-BNN	300875	0.26s (× samples)
U-Net	510338	0.04s
BM3D	0	61.26s

Table 1: Overview of the trained and compared models. Listed are the number of trainable parameters and the prediction time for a single image, respectively. Note that in the Bayesian models we predict multiple images for the same input image – it should therefore be considered that the prediction time is increased depending on the number of samples used. While the BM3D model has no trainable parameters, the standard deviation of the noise still has to be computed and provided prior to the denoising.

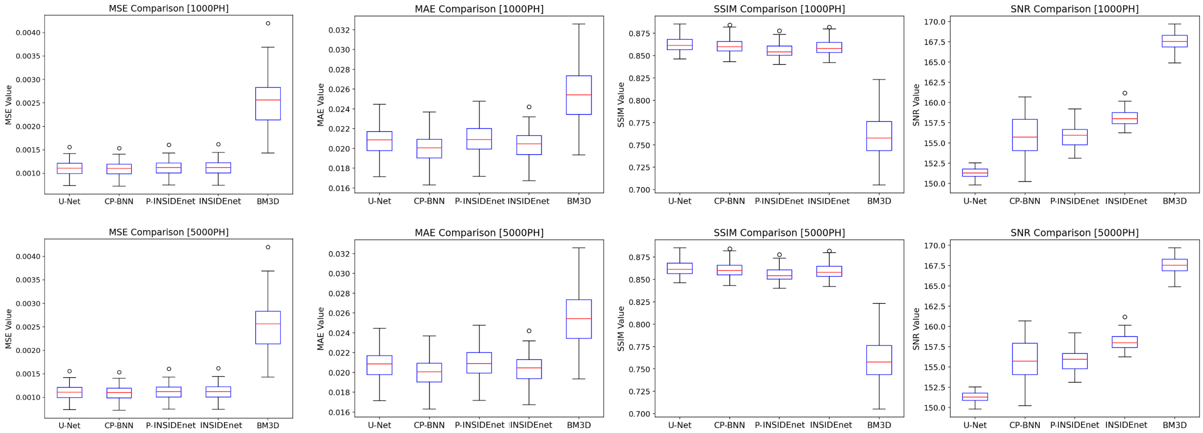

Figure 7: Supervised results: MSE, MAE, SSIM and SNR over the whole test set evaluated on all five models trained in a supervised fashion. The SNR was calculated over the whole background, where a mask was obtained using the transmission image. On the top: results using an average photon count of 1000. On the bottom: results with an average photon count of 5000.

	MSE	MAE	SSIM	SNR
1000 Photons				
Input	3.54e-2 (4.46e-3)	0.135 (5.96e-3)	0.196 (1.52e-3)	25.81 (0.2)
P-INSIDEnet	1.12e-3 (1.62e-4)	2.09e-2 (1.42e-3)	0.856 (7.43e-3)	155.76 (1.29)
INSIDEnet	1.12e-3 (1.66e-4)	2.04e-2 (1.48e-3)	0.859 (8.27e-3)	158.14 (0.96)
CP-BNN	**1.1e-3** (1.58e-4)	**2.00e-2** (1.47e-3)	0.861 (8.38e-3)	155.7 (2.64)
U-Net	1.11e-3 (1.6e-4)	2.07e-2 (1.46e-3)	**0.863** (8.08e-3)	151.33 (0.57)
BM3D	2.54e-3 (5.55e-4)	2.53e-2 (2.62e-3)	0.76 (2.4e-2)	**167.57** (1.09)
5000 Photons				
Input	6.56e-3 (7.16e-4)	5.95e-2 (2.47e-3)	0.561 (2.45e-3)	55.38 (0.38)
P-INSIDEnet	8.97e-4 (1.38e-4)	1.85e-2 (1.51e-3)	0.888 (8.73e-3)	161.2 (1.35)
INSIDEnet	9.02e-4 (1.4e-4)	1.79e-2 (1.46e-3)	**0.891** (9.1e-3)	163.3 (0.51)
CP-BNN	**8.73e-4** (1.3e-4)	**1.78e-2** (1.41e-3)	0.890 (8.89e-3)	156.12 (2.21)
U-Net	8.86e-4 (1.31e-4)	1.85e-2 (1.39e-3)	0.884 (7.22e-3)	154.34 (0.33)
BM3D	1.16e-3 (2.2e-4)	1.99e-2 (1.97e-3)	0.842 (1.7e-2)	**166.45** (0.68)

Table 2: Supervised results: Denoising results summarised from Fig. 7 in mean and standard deviation (in parentheses) from all metrics across all 64 test images. Outlined next to the denoising results are the original metrics values between clean image and noisy image (here referred to as Input). The best performing value of each model is highlighted.

The results from the absorption channel of the INSIDEnet and U-Net models can be seen in Appendix A.2. We evaluated the models using 64 test images with an average photon count of 1000 on one hand – same as in the training –, and an average photon count of 5000 on the other hand to see whether the models are capable of denoising lower noise levels. In the Bayesian models (P-INSIDEnet and CP-BNN) we predicted 15 images per given input image to approximate our predictive posterior distribution (see Eq. (35)), since every time we predict, the weights are newly sampled from the trained variational distribution. Our predicted image was then equivalent to the mean from all 15 images. The quality of the denoisig was evaluated using the MSE, structural similarity index (SSIM), mean absolute error (MAE), and signal-to-noise (SNR). For all metrics except the SNR, a ground truth image is required. For the calculation of the SNR we added π to every pixel so that the image would be between $[0, 2\pi]$, and the mean remain positive. Since each simulated breast has a different structural anatomy, it is challenging to find a constant signal region inside the breast. Therefore, we calculated the SNR over the whole background (i.e. ratio of mean over standard deviation) using a mask obtained from the absorption image. Quantitative results of our evaluation can be seen in Fig. 7 and summarised as mean and standard deviation in Tab. 2.

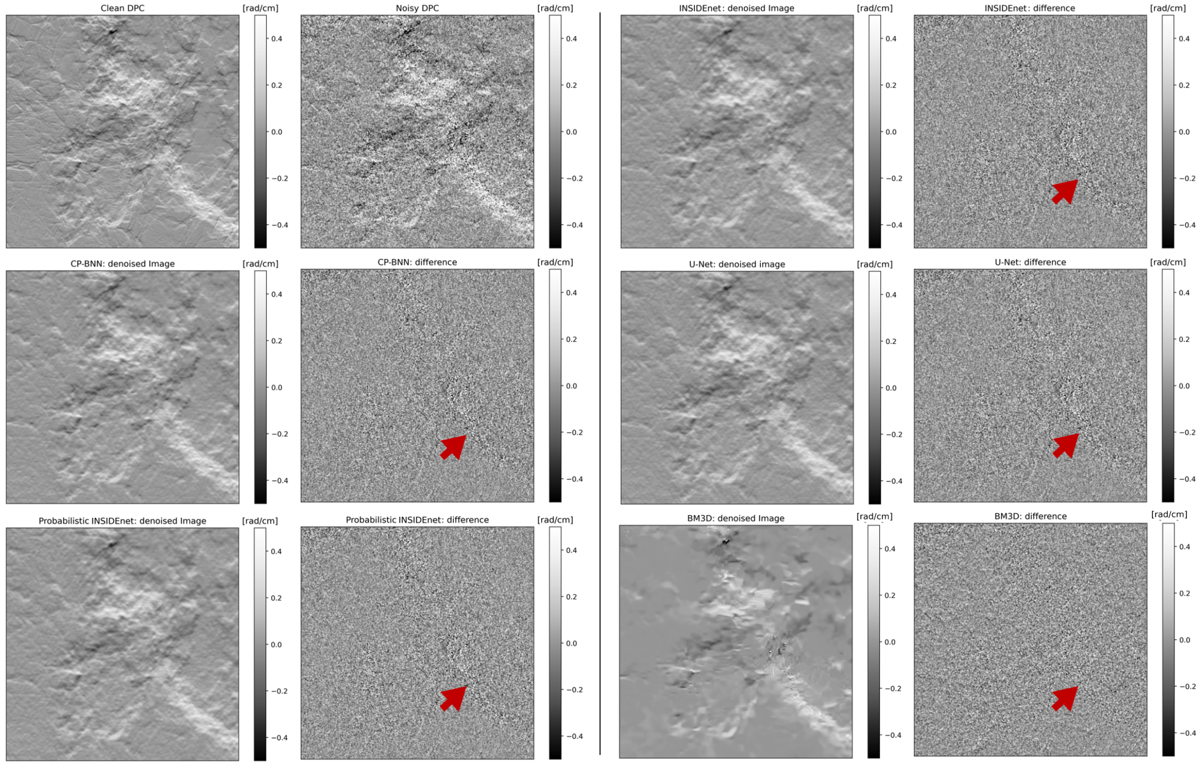

Figure 8: Supervised results: Denoising results on a zoomed-in region from a DPC projection over all five models and the difference between the original noisy and predicted image. Top row left: clean and noisy image. In subsequent rows (left and right) are the predicted images and differences from CP-BNN, P-INSIDEnet, INSIDEnet, U-Net, and BM3D. The gray value units are angles per centimetre [rad/cm].

Overall, the CP-BNN outperforms all other proposed models in MSE and MAE and is even slightly better than the U-Net, while the U-Net and INSIDEnet show better results in the SSMI at 1000 and 5000 photons, respectively. All data-driven denoising models demonstrate similarly satisfying results in denoising capacities on the test images, while the BM3D filter's performance is clearly inferior. It is not constant but fluctuates strongly depending on the input image. However, the BM3D model scored in the SNR values, which is better than the data-driven models. It has to be emphasized that the SNR values have only been calculated in the background of the breast and no conclusion could therefore be made over the SNR values inside the breast projection. Interestingly, all deterministic models – including the BM3D filter – have an almost constant predicted SNR value, while the probabilistic models show strong fluctuations (especially the CP-BNN). Another interesting observation can be made between the P-INSIDEnet and the INSIDEnet, where the latter is used as prior for the former. Providing a probabilistic view on the weights of the INSIDEnet not only did not improve the overall performance of the denoising, but rather worsened it. While still having better values in the MSE at 5000 photons, they underperformed on the MAE and especially SSMI. This indicates that gaining insight into the uncertainty of the model can come at the cost of less accuracy in the results. This can be statistically explained: The P-INSIDEnet weights are taken from a learned variational distribution, whereas the INSIDEnet weights are purely deterministic. However, the value of these weights is consistent with the mode of the MAP estimation – that is, the weights are chosen to always match the local optima on the training set. The P-INSIDEnet weights, on the other hand, are chosen randomly, so it is unlikely that all weights will match the mode of this variational distribution, resulting in lower accuracy. Comparing the values from the test images with 1000 and 5000 photons, we can see that the models are able to adapt to less noise levels with even better overall results. While all models significantly improve the metrics on less noisy images, the values of the BM3D have only been improved to the extent that they can now be compared with the values of

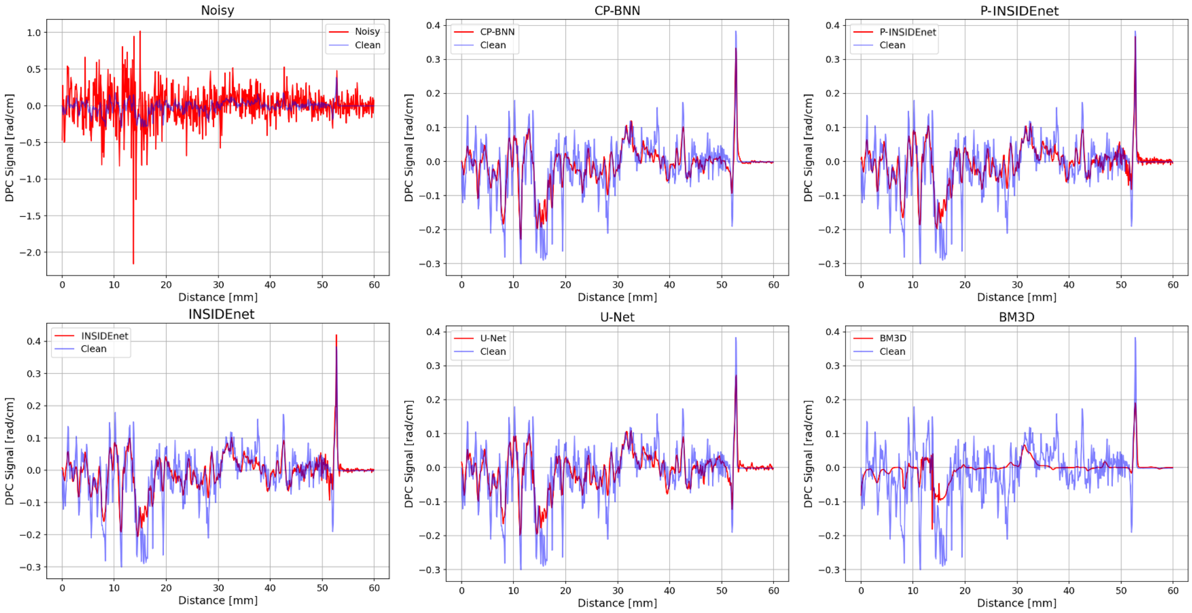

Figure 9: Detail profiles over a horizontal line in the DPC image for each algorithm. The clean detail profile is superimposed over the predicted ones in blue color. The profiles are taken over a large part of the image, i.e. 600 pixel long (original image dimension 1536 × 1536). Note that the scale in the first figure is larger in order to depict the complete noise profile.

the data-driven models over images with 1000 photons.

The denoising results of the five algorithms on DPC projections are depicted in Fig. 25. The results are displayed on a zoomed-in patch of the image for better visualization of the features and noise. The top left row depicts the clean image with an average photon count of 1000 and its noisy counterpart. The other rows (left and right) show the performance of the five algorithms along with the corresponding difference image, displaying the discrepancy between the denoised and the noisy input image. The data-driven models effectively remove a large part of the noise, while keeping most of the features in the image. Nevertheless, some details that are visible on the clean image are not retained by any of the data-driven algorithms. The predicted images are also less crisp than the clean image. The BM3D filter (bottom right), meanwhile, is much less effective in comparison. While it is able to remove the high frequencies noise uniformly well, it fails to retain small features and over-smooths the complete image. The difference image in the BM3D shows that it is unable to deal with heteroscedastic noise – the data-driven methods, meanwhile, show areas with higher noise amplitudes (see red arrows), indicating the presence of heteroscedastic noise.

To investigate the detail preservation of each model, Fig. 9 shows intensity profiles taken from a test image depicted in Fig. 10, overlayed over each intensity profile from the predicted DPC images of the models. It reveals that all deep learning models are able to retain most of the information, whereas the traditional BM3D model over-smooths the image and loses substantial information. For the most part, the deep learning models underestimate the absolute signal, leading to images with lower phase shifts. This may indicate that the models are too sensitive and therefore prefer to attenuate the signal more. One reason could stem from the training loss, where high deviations lead to a higher penalty. DPC images, or gradient images in general, can be approximated by a Laplacian distribution and are centred around 0. Overpredicting the signal therefore leads to a higher global penalty in the overall image than cautiously predicting the signal closer to 0, which we believe is the reason for the underprediction as seen in the line profiles. Nevertheless, the line profiles demonstrate the astonishing detail preservation from deep learning models.

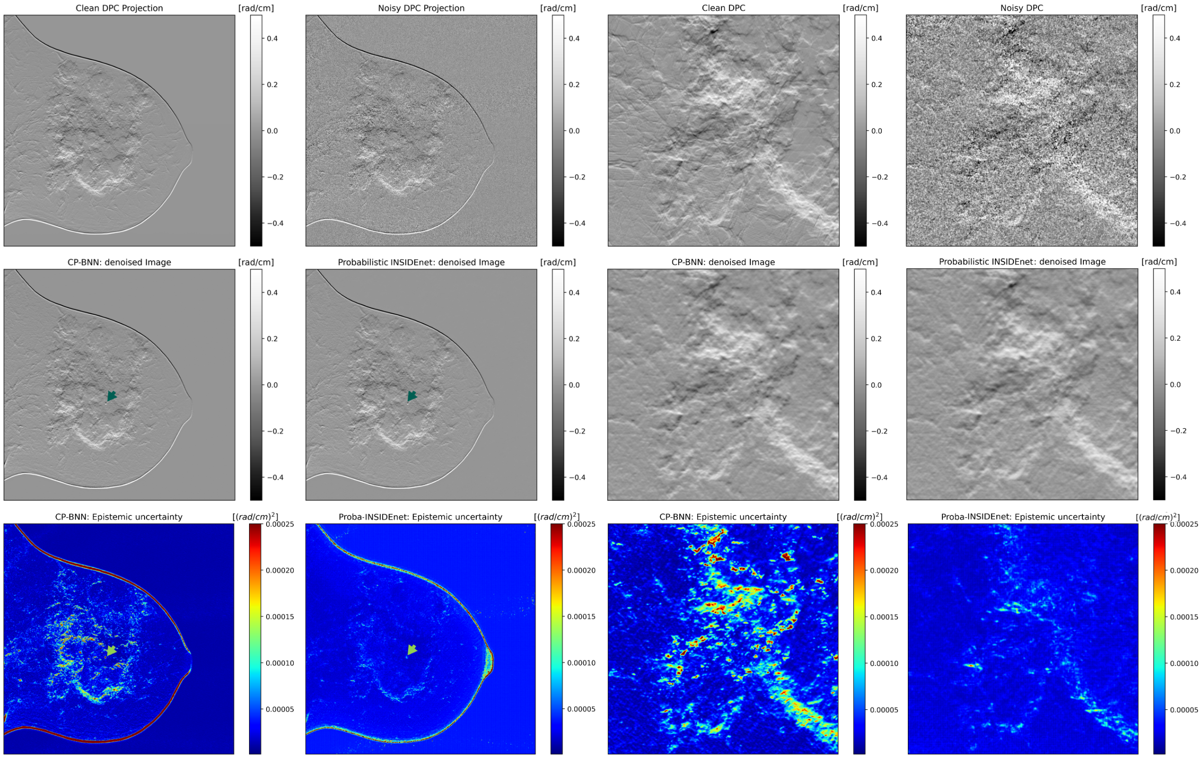

Figure 10: Supervised results: Denoising results and uncertainty from the CP-BNN and P-INSIDEnet model. Top row: clean and noisy images – whole projection and zoomed-in region. Second row: predictions from the CP-BNN and P-INSIDEnet. Last row: epistemic uncertainty of the models on the whole projection and zoomed-in region.

4.1.1 Uncertainty in Bayesian Models

We demonstrated the results on purely deterministic predictions as well as the approximative means of the predictive posterior. We will now show the advantage of having a Bayesian view on the weights. For this, we resort to the prior comparative study and demonstrate the results from the Bayesian model in hand with their epistemic uncertainty in Fig. 10.

Here, we focus on the uncertainty of the individual models: The epistemic uncertainty was calculated as the mean squared difference from the approximative mean predictive posterior and the individual images (see Eq. (37)). By analysing the uncertainties from each mode, we immediately see that the uncertainty in the P-INSIDEnet is generally higher in the background but the CP-BNN shows more uncertainty in regions with high signal amplitdues, such as edges. Interestingly, both models are highly uncertain around the border of the breast, indicating that high signal amplitudes or edges lead to high uncertainty. Especially in the vicinity of the nipple, the P-INSIDEnet displays high deviations over all predicted sample images. Zooming in on the patches, we can see that the CP-BNN is more uncertain in areas with high absolute signal amplitude – even more so with high positive signal, which also manifests itself on the whole breast. On the other hand, the P-INSIDEnet is more certain in its prediction but is highly uncertain in areas with high positive signal. Also visible in both images – although stronger in the P-INSIDEnet than in the CP-BNN – is the profile of the flat image of the visibility (see Appendix A.1). Especially in the same regions where the signal of the visibility flat reaches a local minimum, the uncertainty becomes slightly higher. Another source of high uncertainty are microcalcifications, indicated by the green arrows. Microcalcifications are highly scattering and absorbent and are thus well displayed in absorption and dark-field images. Following Eq. (10), these microcalcifications result in high noise amplitudes in the DPC channel, due to the low transmission and dark-field signal.

4.2 Unsupervised Analysis

In clinical settings, it is difficult to collect noise-free radiographic images without compromising the patients' safety. It is therefore impossible to train a deep neural network in a supervised fashion. To overcome the problem of insufficient training data, an area of research is to move to unsupervised settings, where clean images are not necessary. In this section we will analyse the method from chapter 3.5. In this case, we only need noisy projections from all the three contrast channels to generate synthetic noise and overlay it on the original noisy images. We generated our synthetic noise using Eq. (10) for DPC and Eq. (7) for absorption. It should be emphasized again that the clean signal is necessary to calculate the correct uncertainty intrinsic in the noisy images. To see whether the calculations would also hold for noisy images, we calculated the MSE from the original clean image to the noisy image and the MSE from the newly calculated syntethic noisy image to the original image. We found that using the provided equations led to higher MSE than between the original pairs, meaning that our synthesized noise consisted of higher noise amplitudes. We thus tuned the equations by chancing the number of phase steps N and found that $N = 5$ approximated the original MSE well. With this we were able to train all our models again in a supervised fashion by using the newly synthetic noisy image as input and the noisy original image as target. However, we found that matching the MSE value did not provide good training results, so we performed various tests by again changing the number of phase steps. The best results in terms of SSMI were achieved with N = 4 – anything lower yielded worse outcomes, indicating that the noise level added to the images was too different compared to the intrinsic noise. The calculated uncertainty along with the absolute value of the added noise can be seen in Fig. 11. It shows that both the intrinsic noise as well as our added noise follow the calculated uncertainty distribution.

One of the disadvantages of the NAC method is that the performance is highly dependent on the inherent noise level [62]. Consequently, we tried the NAC method on various noise levels by changing the average photon count from 5000 downwards. We found that satisfactory results could only be reached if the images had an average photon count of at least 3000. Therefore, we only present the results originating from models trained on such images.

Similar to the supervised case, we tested our algorithms on 64 images, whereas training and validation

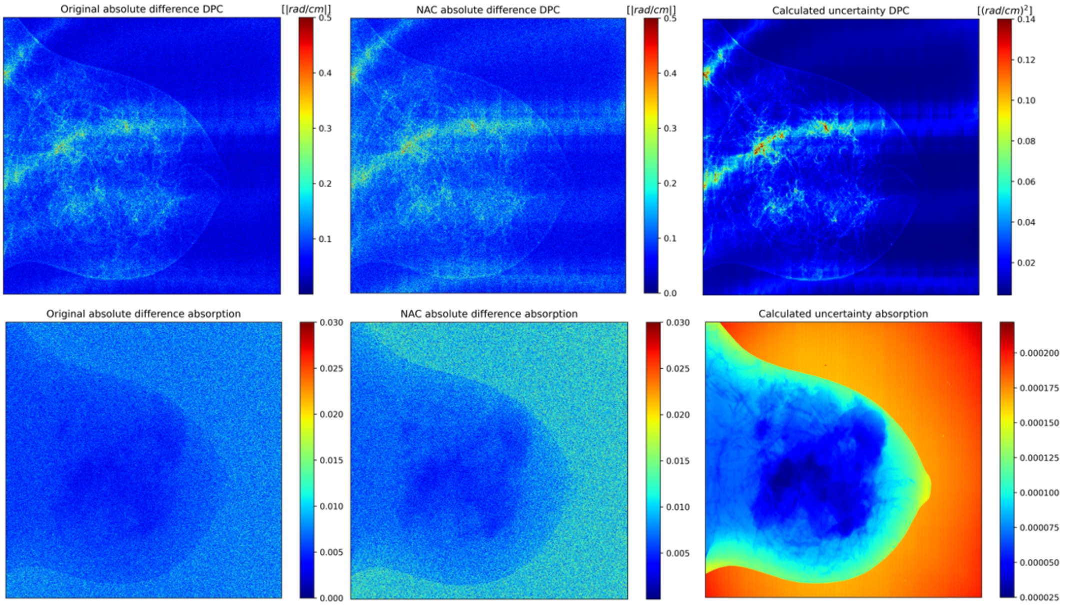

Figure 11: Calculated uncertainties based on Eq. (10) and (7) along with the synthetic noise and original noise. For better visualization we depict the absolute values of the difference between clean and noisy image and between the synthetic image and original noisy image.

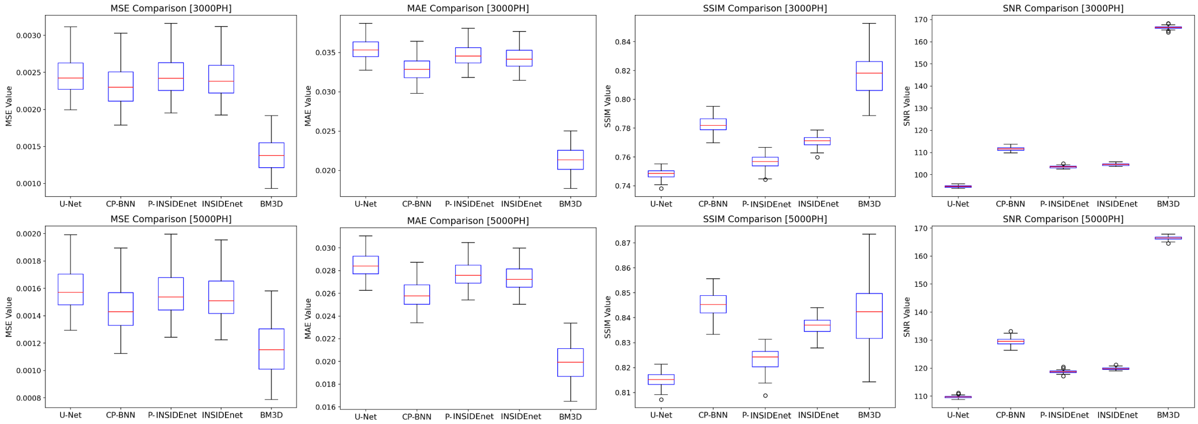

Figure 12: Unsupervised results: MSE, MAE, SSIM and SNR over the whole test set evaluated on all five models. On the top: results using an average photon count of 3000. On the bottom: results with an average photon count of 5000.

	MSE	MAE	SSIM	SNR
3000 Photons				
Input	1.11e-2 (1.07e-3)	7.72e-2 (2.77e-3)	0.435 (1.8e-3)	43.93 (0.305)
CP-BNN	2.31e-3 (2.8e-4)	3.29e-2 (1.58e-3)	0.783 (5.24e-3)	111.64 (0.879)
U-Net	2.44e-3 (2.51e-4)	3.54e-2 (1.4e-3)	0.748 (3.28e-3)	94.52 (0.458)
INSIDEnet	2.4e-3 (2.66e-4)	3.42e-2 (1.44e-3)	0.771 (3.7e-3)	104.5 (0.465)
P-INSIDEnet	2.43e-3 (2.7e-4)	3.46e-2 (1.46e-3)	0.756 (4.86e-3)	103.48 (0.498)
BM3D	**1.38e-3** (2.38e-4)	**2.14e-2** (1.82e-3)	**0.818** (1.56e-2)	**166.43** (0.685)
5000 Photons				
Input	6.59e-3 (6.22e-4)	5.96e-2 (2.12e-3)	0.561 (1.88e-3)	55.47 (0.358)
CP-BNN	1.44e-3 (1.75e-4)	2.58e-2 (1.26e-3)	**0.845** (4.55e-3)	129.51 (1.32)
U-Net	1.59e-3 (1.6e-4)	2.85e-2 (1.14e-3)	0.815 (2.86e-3)	109.67 (0.43)
INSIDEnet	1.52e-3 (1.67e-4)	2.73e-2 (1.17e-3)	0.837 (3.3e-3)	119.8 (0.417)
P-INSIDEnet	1.55e-3 (1.7e-4)	2.77e-2 (1.18e-3)	0.823 (4.58e-3)	118.71 (0.518)
BM3D	**1.16e-3** (1.93e-4)	**2e-2** (1.69e-3)	0.842 (1.41e-2)	**166.41** (0.626)

Table 3: Unsupervised results: Denoising results summarised from Fig. 12 in mean and standard deviation (in parentheses) from all metrics across all 64 test images. Outlined next to the denoising results are the original metrics values between clean image and noisy image (here referred to as Input). The best performing value of each model is highlighted.

was performed on 320 and 84 images, respectively. The hyperparameters for training were kept the same. Prior to entering the denoising pipeline, the original noisy images were scaled to be within $[0, 1]$. This scaling is similar to the one used in the supervised setting, with the difference that we applied the information from the noisy images instead of the clean images to ensure completely unsupervised training. We applied the same scaling on the synthesized noisy images and trained the models until convergence. The results on the test set are depicted in Fig. 12 and summarized in Table 3, where again the MSE, MAE, SSIM, and the background SNR have been evaluated using the clean images.

Overall, the traditional BM3D outperforms all unsupervised trained deep learning models significantly. The deep learning models, in turn, have similar values amongst themselves, with CP-BNN providing the best performance. Interestingly, all our developed models outperform the U-Net model, indicating that the combination of a data-driven and model-based approach performs better when being trained in an unsupervised fashion. Fig. 13 depicts the zoomed-in denoising results of the single algorithms. Although the BM3D model shows the best numerical results on all test images, we can clearly see that it fails to retrieve small features and the image appears washed-out and blurry. The deep learning models on the other hand are able to mostly keep these small high-frequency features and not lose resemblance to the clean image. Compared to the original noisy image, the predicted images are less affected by

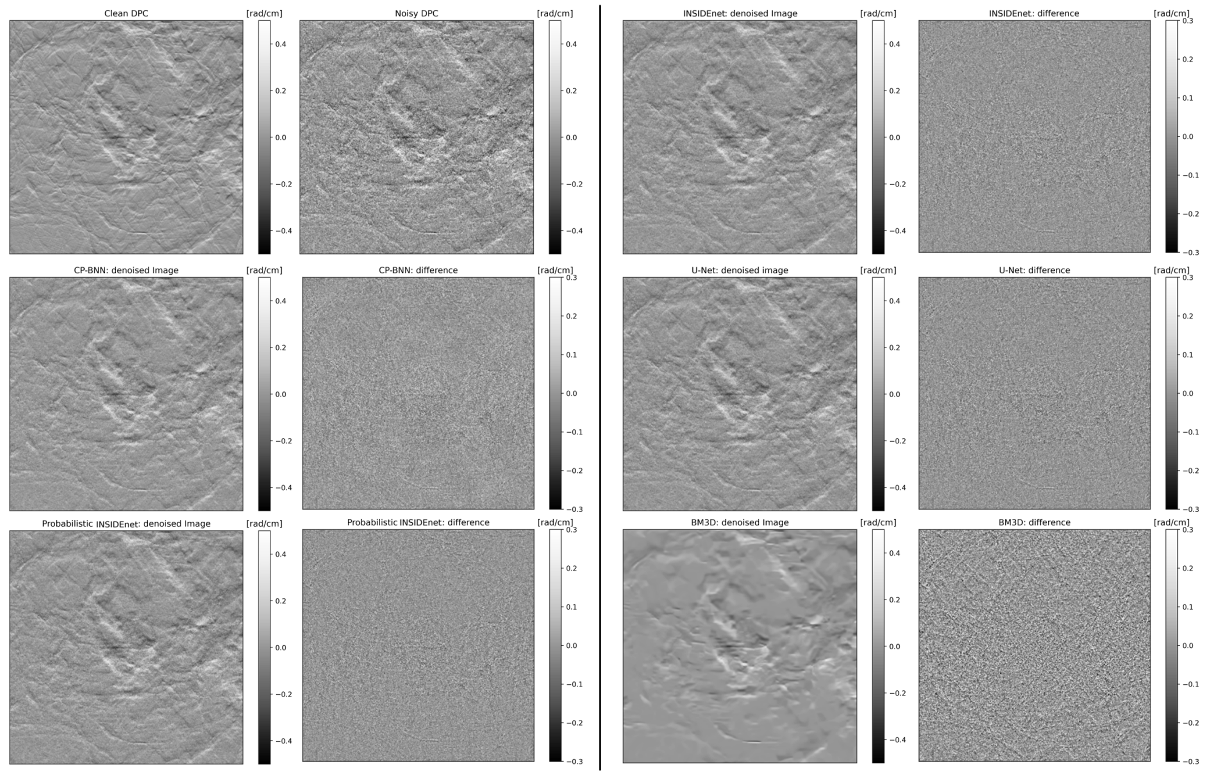

Figure 13: Unsupervised results: Denoising results on a zoomed-in region from a DPC projection over all five models and the difference between the original noisy and predicted image. Top row left: Clean and noisy image. In subsequent rows (left and right) are the predicted images and differences from CP-BNN, P-INSIDEnet, INSIDEnet, U-Net, and BM3D. The gray value units are angles per centimetre [rad/cm]. Note, the dynamic range in the difference image is smaller to depict it more clearly.

noise, but they still have a sandy texture, which explains the poor numerical results compared to the BM3D. Furthermore, both traditional as well as data-driven algorithms fail to attenuate the high noise amplitudes, which are mostly apparent in regions where the visibility flat has its lowest values (see Appendix A.3). When inputting images with a higher average photon count – and consequently lower noise level – the deep learning models approach the numerical values of the BM3D, where even superior mean results can be achieved with the CP-BNN on the SSMI metric. Similar to the supervised case, the models are able to adapt to lower noise levels than they were trained on. This can be explained by Eq. 59. Training the models in a unsupervised fashion and following the rationale from chapter 3.5, it can be concluded that the trained parameters approximate the optimal parameters when trained in a supervised fashion with clean and noisy images – concluding that our unsupervised models have the same characteristics as our supervised models.

4.3 Destriping of integrated images

The generation of completely noise free images is challenging and has not been achieved with either our proposed or traditional models. Thus, following the reasoning from chapter 3.6, we expected our integrated denoised images to be corrupted by stripe artefacts. However, according to our statistical analysis, the stripes should be less pronounced. We therefore continued our denoising analysis with known destriping methods and investigated whether the WFF was appropriate for our purpose.

To find suitable wavelet basis and hyperparameters for the WFF, we performed a grid search analysis over a single striped image and compared the resulted destriped image with the original clean phase image using the MSE and SSIM. We found that a Daubechies wavelet (DB) with a size of 16, a decomposition level of 6, and a normalized standard deviation of 0.125 performed well on our data. We investigated the

	P-MSE	MSE	P-MAE	MAE	P-SSIM	SSIM	P-SNR	SNR	P-CNR	CNR
Lowest Quality										
Input	0.108	2.79e-3	0.232	3.8e-2	4.48e-2	0.654	2.23	8.72	0.73	4.16
P-INSIDEnet	1.69e-2	1.48e-2	0.108	0.108	0.429	0.739	11.14	11.89	4.63	6.04
INSIDEnet	4.74e-3	4.32e-3	5.92e-2	5.79e-2	0.668	0.814	11.25	11.52	7.41	7.77
CP-BNN	2.50e-3	2.29e-3	3.89e-2	3.78e-2	0.591	0.625	11.22	11.33	8.44	8.82
U-Net	3.00e-3	2.8e-3	4.43e-2	4.41e-2	0.763	0.845	10.93	10.94	6.84	7.06
BM3D	8.99e-3	3.24e-3	6.73e-2	4.65e-2	0.53	0.726	8.68	10.18	3.01	4.86
Highest Quality										
Input	6.78e-2	1.2e-3	0.188	2.68e-2	0.05	0.766	3.47	15.54	1.76	11.42
P-INSIDEnet	3.33e-2	2.98e-2	0.1454	0.1453	0.365	0.711	15.46	24.51	5.10	13.83
INSIDEnet	3.30e-3	2.98e-3	4.72e-2	4.57e-2	0.701	0.826	17.63	19.5	13.02	14.91
CP-BNN	8.06e-3	7.88e-3	7.83e-2	7.81e-2	0.748	0.795	17.85	20.09	11.87	13.46
U-Net	3.32e-3	3.19e-3	4.28e-2	4.82e-2	0.749	0.818	17.59	19.72	12.35	14.14
BM3D	5.12e-3	1.18e-3	5.17e-2	3.48e-2	0.595	0.769	11.29	17.04	6.43	12.36

Table 4: Destriping quantitative analysis: Measurements were performed prior and after the destriping algorithm. Prior measurements are indicated by the letter P in the table.

algorithm on both the best and worst denoised images of our previous supervised analysis. Afterwards we compared the appearance of the integrated output from our DPC denoising models before and after the destriping algorithm. The integration of the images was performed with a Riemann-sum along the vertical direction as in Eq. (61) with a step-size of 0.01cm, which matched the simulated pixel-size. For the comparison, we computed the MSE, SSIM, MAE, SNR, and the contrast-to-noise ratio (CNR). Unlike the evaluation of the denoised DPC images in the previous chapters, it was now possible to calculate the SNR and CNR in the interior of the breast. To be able to compute the SNR and CNR, we selected an area in the clean PC image with high pixel values – representing highly refractive material (HR) (i.e. adipose tissue) –, and an area with lower pixel values inside the breast, representing the projected glandular tissue (GT). We calculated the SNR using the mean and standard deviation of the area, which is highly refractive, and the CNR using the mean and standard deviations of both areas:

$$SNR = \frac{\mu_{HR}}{\sigma_{HR}} \tag{67}$$

$$CNR = \frac{\mu_{HR} - \mu_{GT}}{\sqrt{\sigma_{HR}^2 + \sigma_{GT}^2}} \tag{68}$$

The results of our analysis can be seen in Table 4 and in Fig. 14 and 15. Note that we only depicted the results from our own developed models (CP-BNN and P-INSIDEnet) and refer for the complete list to Appendix A.

Fig. 14 depicts the results on the images where the model performed the worst. The comparison to the clean image was performed by first scaling the clean image to $[0, 1]$ and then applying the same scaling to the rest of the images. This was done to mitigate the bias terms added by the models, which could lead to higher SNR values. Overall, the performance prior to the destriping is dominated by the results from the U-Net, CP-BNN and INSIDEnet. The values from the images corrupted by large stripe artefacts are remarkably improved after running the destriping filter. These improvements are less prominent in images with already little stripe artefacts, such as the images from the CP-BNN, U-Net, and INSIDEnet. Although the values improved significantly in the original noisy input and the prediction from the BM3D after destriping, the images themselves are still corrupted by wavy stripe artefacts, remaining noise, and blurriness. This is also supported by the SSIM, SNR and CNR values, where the performance from the deep-learning models show superior results. Surprisingly, although the P-INSIDEnet showed better results in the DPC channel than the BM3D, its integrated image depicts strong and bright stripe artefacts in the area of the breast, which did not improve much even after destriping, indicating remaining high noise amplitudes in the DPC channel.

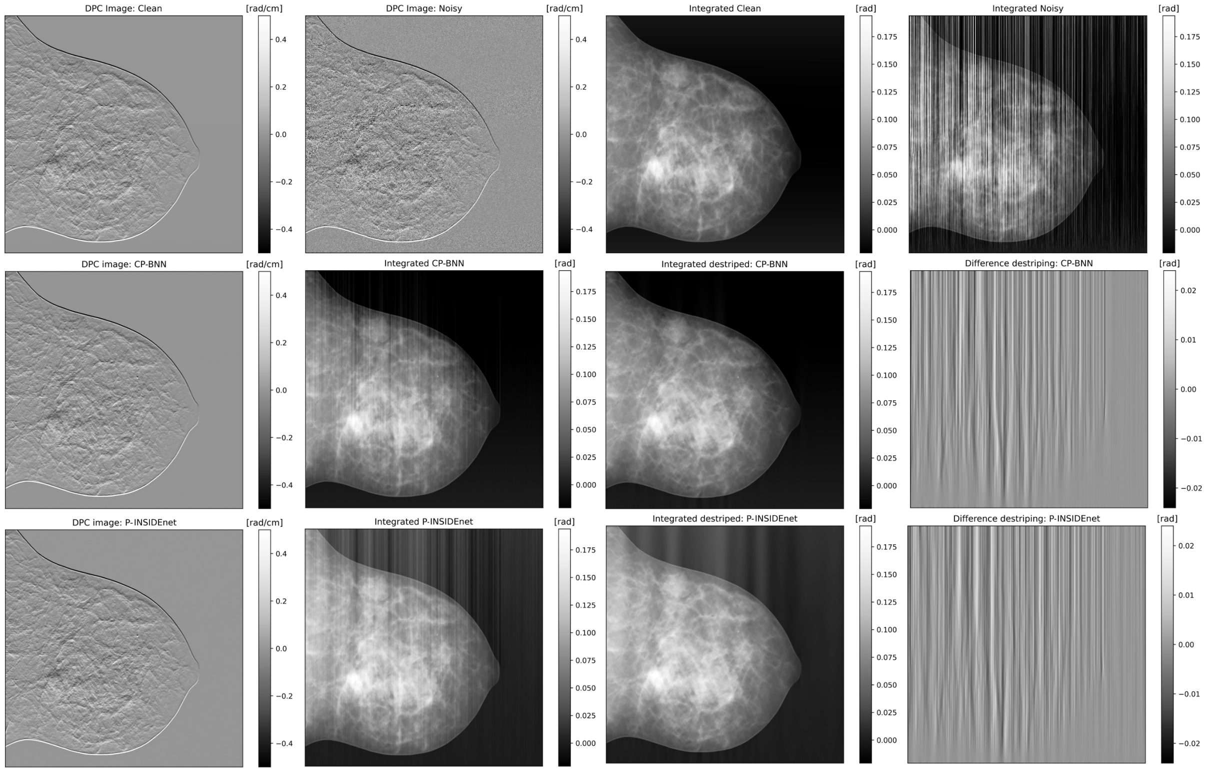

Figure 14: Depiction of the destriping performance of the WFF over the worst performing image of the supervised analysis. First row depicts the DPC and integrated image pairs. Second row depicts the denoised DPC image from the CP-BNN model along with the integrated image, the destriped image, and their difference. Last row shows the equivalent order but for the P-INSIDEnet model.

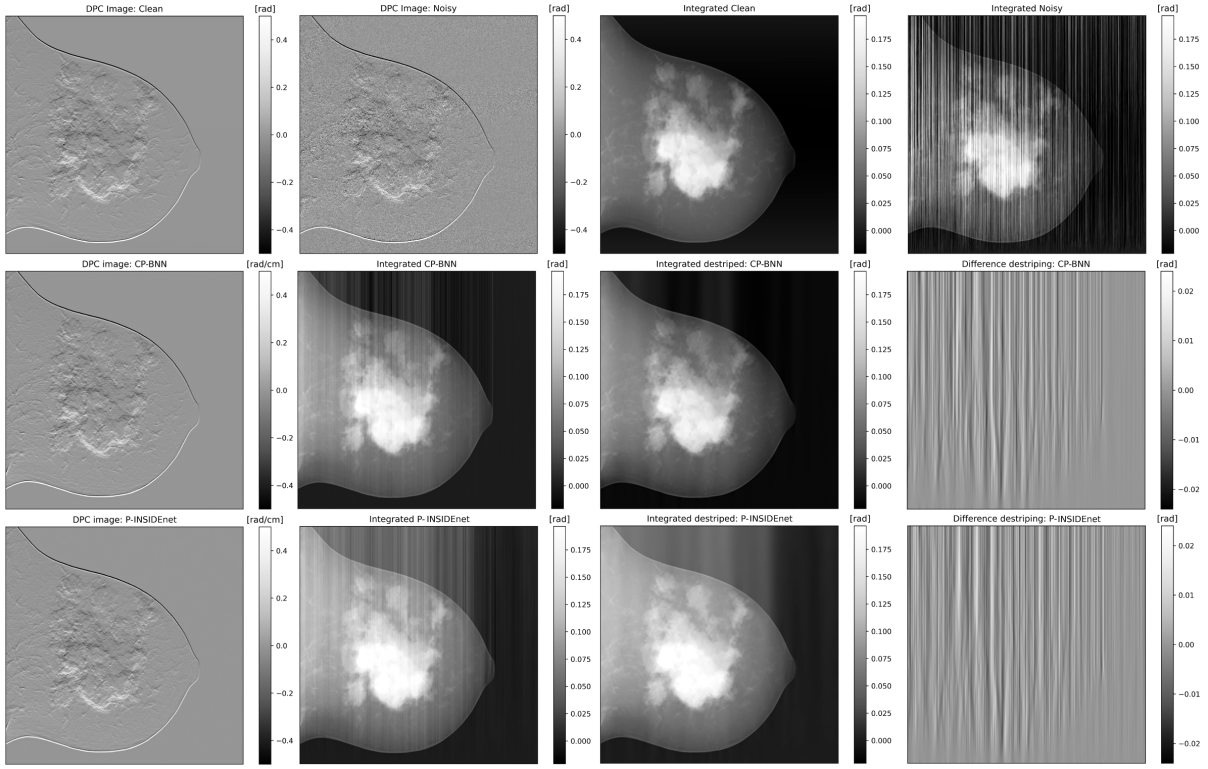

Figure 15: Depiction of the destriping performance of the WFF over the best performing image of the supervised analysis. Equivalent order as in Fig. 14.

Continuing on the higher quality image, we see the same trend as in the lower quality. However, the CP-BNN has approximately two times higher MSE and MAE values compared to the U-Net and INSIDEnet, although visually, the CP-BNN prediction seems equally affected by stripe artefacts. This leads to the conclusion that the pixel values of the integrated CP-BNN are elevated compared to the clean image. Looking at Fig 14, this can also be confirmed visually. The projected glandular tissue is brighter than in the clean image. This increased brightness is also visible in all other deep learning models, where the INSIDEnet even shows the least increase. Compared to the low quality image, the P-INSIDEnet model is worse at restoring the integrated image with wider and stronger stripe artefacts – although having better prior values in the DPC image. This indicates that the P-INSIDEnet remains corrupted by high noise amplitudes. In the higher quality image, too, the destriping in the original noisy input leads to a significant improvement of the image quality. However, if we check the SNR and CNR values, we see that although the stripes are attenuated, the image is still noisy and distorted by artefacts, which could not be improved much even after applying traditional denoising filters before integration, thus, demonstrating the superiority of deep learning models over traditional filters. It has to be noted that although the results lead to higher image quality, the images are still corrupted by wide and blurred stripes, with the width of the stripes depending on the stripe density before destriping.

4.4 Real world denoising results

So far we evaluated our models on simulated data. We now analyse the denoising capabilities of the models, which have been trained on simulated data, on real DPC images. These images depicting a breast specimen with and without compression we were acquired on the Philips Microdose GI-Mammogram system see Fig. 16. The acquired images from the system were not calibrated to the correct physical output values. Their dynamic range was between 10^8 to 10^9 for absorption and -10^9 to 10^9 in phase. In order to match the dynamic range of our training data, we scaled the images to be approximately in the same range as our training images by comparing the histograms of the single images. Nevertheless, to prevent the drawing of false conclusions, we omit numerical values in our further analysis.

In theory, the histogram of a gradient image (such as the DPC image) can be approximately explained by a Laplace distribution [67]. We can see that this holds true for the simulated DPC images (see Fig. 17 first row). However, the histograms of the acquired images are bimodal (see. Fig 17 first image in second row). Since there is a mismatch between training and testing data, naively applying the trained models to real data, does not yield satisfying result. To address this problem we tried to shift the two peaks to a center of mass at zero, such that it resembles the histograms of the training images. We believe that one reason for this distribution could be grating misalignments, which would lead to phase shifts that are structured over the whole image. This reasoning is plausible, since strong stripe artefacts are visible on the images, leading to shifts in the positive and negative DPC signal. To centre the histogram, we took

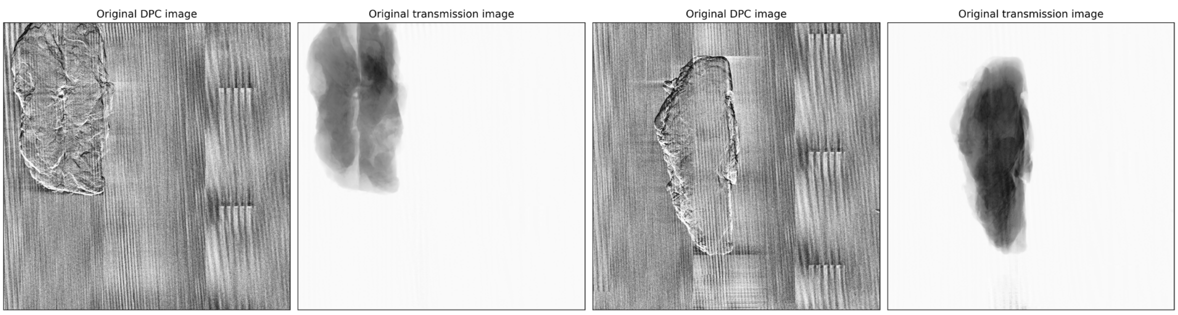

Figure 16: Images acquired on the Philips Microdose system with built in GI. Left set constitutes of the breast without compression and right with compression.

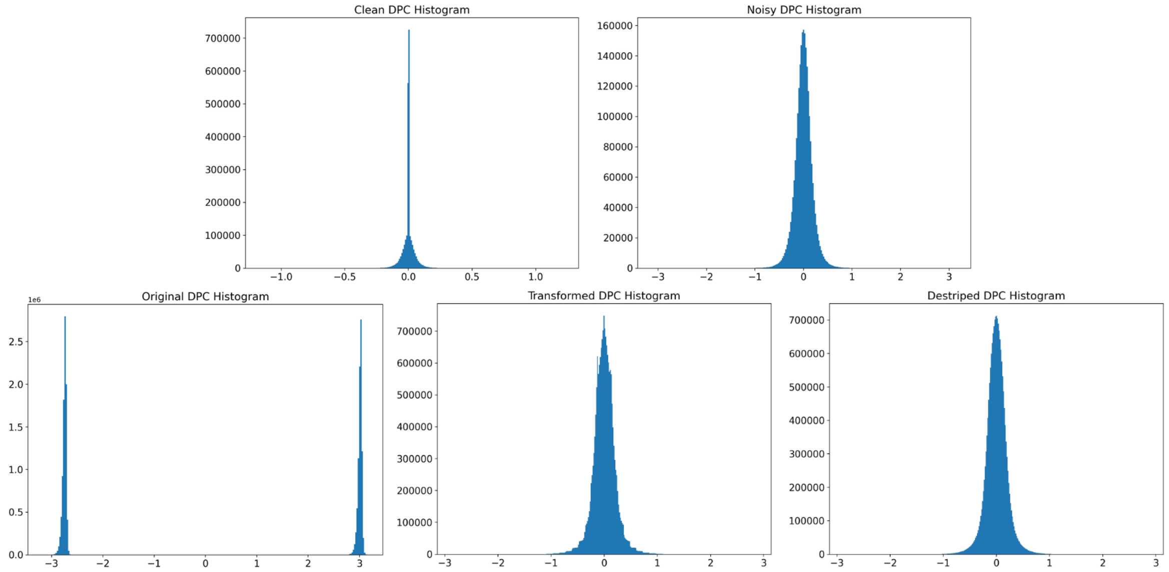

Figure 17: Histograms of the training images and the acquired images of the Philips Microdose system. The first row depicts the histogram of a training image pair, where the distribution resembles a Laplace distribution. Second row depicts the distribution of the compressed specimen acquired with the Philips system. The three histograms correspond to the acquired, the transformed, and the destriped images, respectively. Note, that the histogram of the uncompressed specimen is similar to the compressed one.

an image without a sample from the system and calculated the mean value $\hat{m}$ from it. We used the mean value without the sample because we only take into account values from the background that are not influenced by the sample. Since the histogram distribution was almost symmetrical in the background, the image was set to a new center of mass by subtracting $\hat{m}$ and then using its absolute value. It then only had to be shifted to 0 to form the desired transformed image T. The whole process is described as follows:

$$z = |x - \hat{m}| \tag{69}$$

$$T = z - \hat{z} \tag{70}$$

where $\hat{z}$ is the mean value of z, which shifts the whole image to be centred around zero. The histograms of the transformed images are depicted in Fig. 17 along with the transformed images in Fig. 18. The transformed images show significantly less stripes than the original image, thus suggesting truth in our assumption that the stripes are responsible for the bimodal distribution. To remove remaining stripes we applied the WFF with a *DB16* wavelet basis, a normalized standard deviation of 0.2, and a decomposition level of 6. The destriped images and their histograms can be seen in Fig. 18 and Fig. 17, respectively. After applying the WFF, the histograms were smoother and visually similar to the histogram of the noisy training image. We then used the destriped image as input for the algorithms. To facilitate reading we will call the transformed destriped image as the *noisy image* and the original acquired image as the *original image*.

The images are 4965 × 4413 in dimensions. In our models we used Tukey windows to avoid border artefacts, which were hard-coded to work with images of dimension 1536 × 1536. Therefore, we first cut the edges of the image where only background was depicted to create images of the size 4176 × 4176. We then divided this image into overlapping patches of 1536 × 1536 with a stride of 660 to facilitate numerical computations – more patches would result in more computational time but not in better prediction. After denoising we used these patches to reconstruct the final image. Results from the individual algorithms are depicted in Fig. 18 with zoomed-in regions in Fig 19.

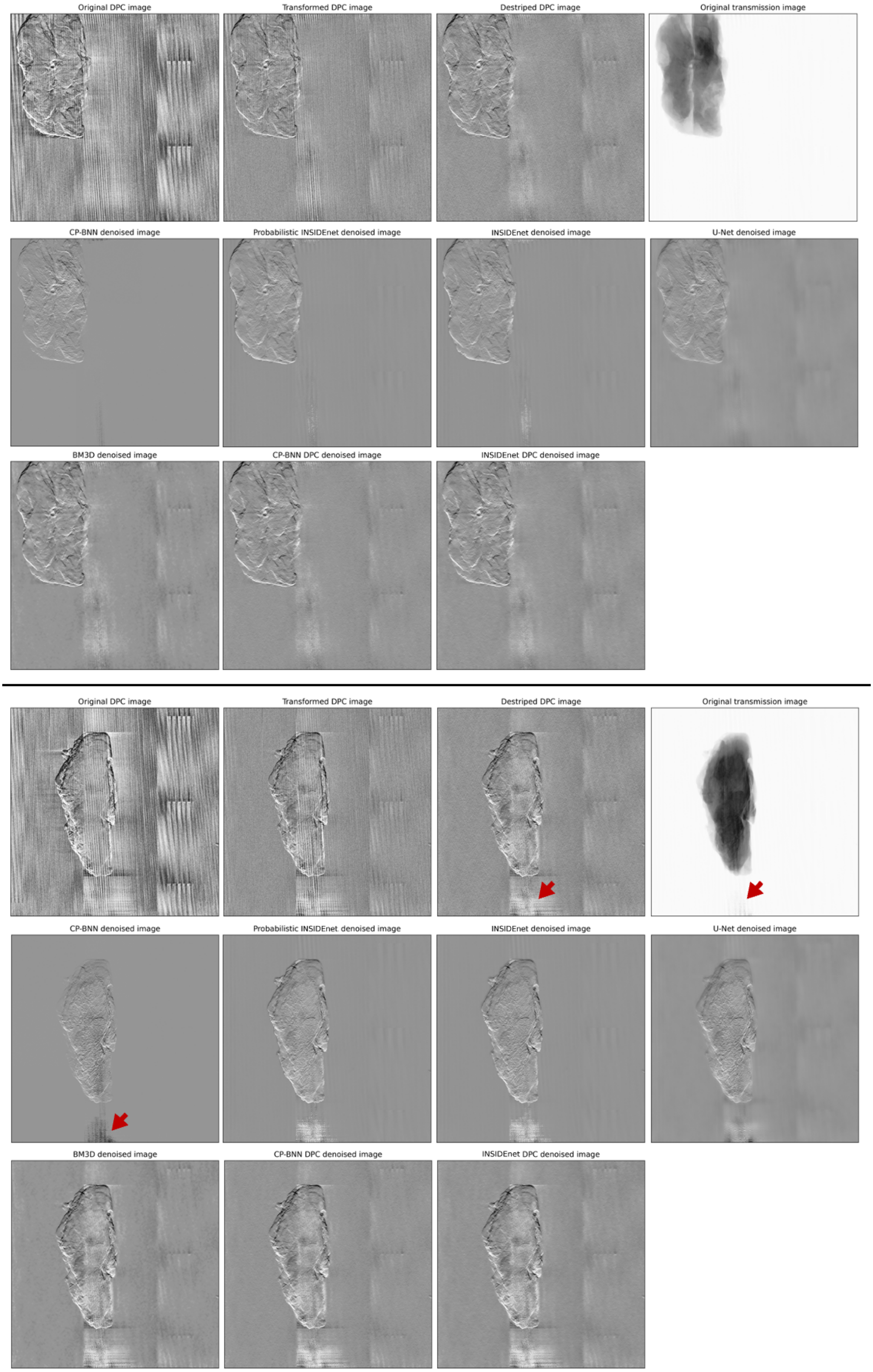

Figure 18: The two acquired images along with the denoised results from both the trained and traditional algorithms. First row from each half depicts the original acquired image along with its transformed and destriped image, and the original transmission image. The next two rows depict the results from the algorithms.

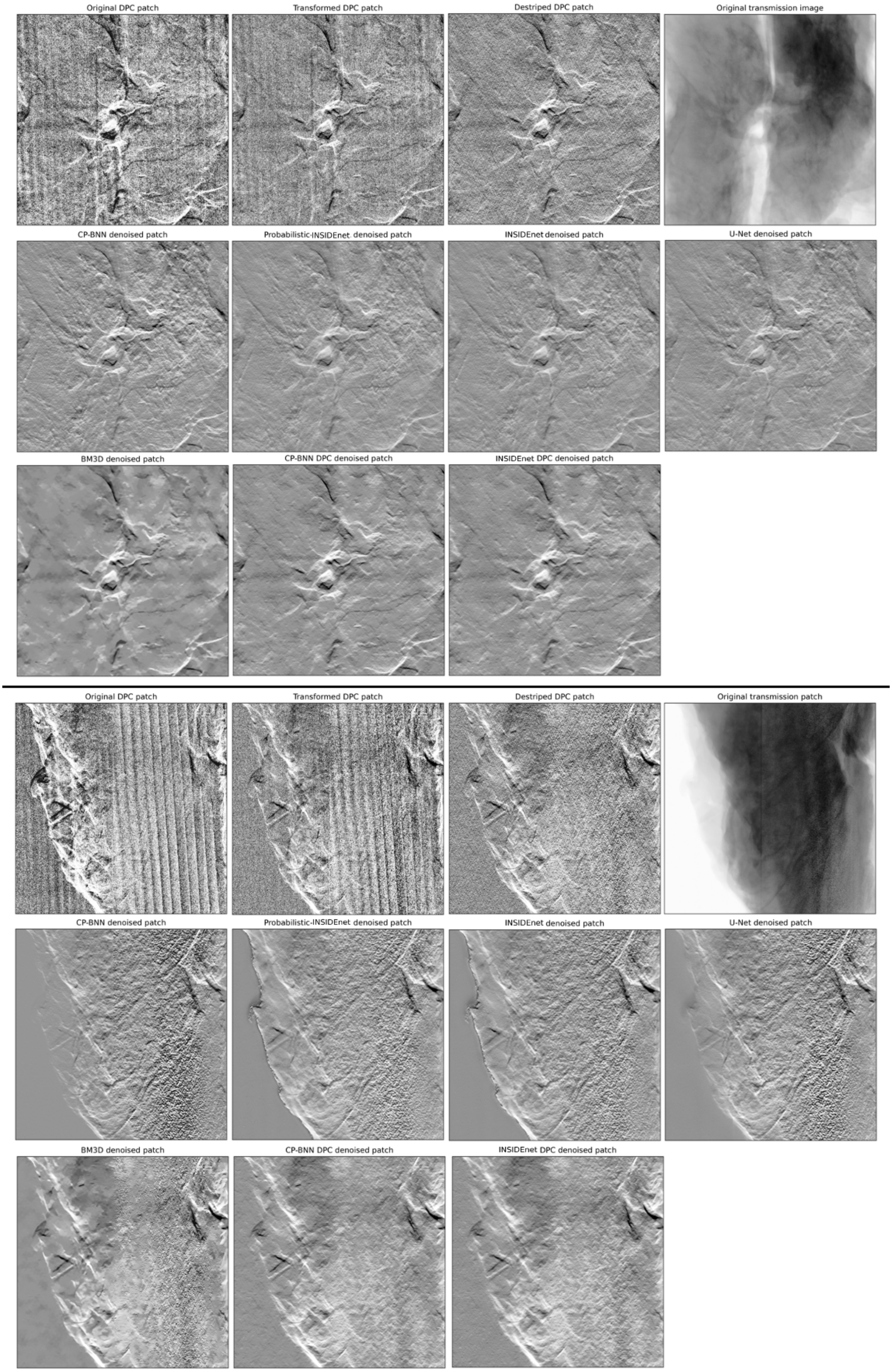

Figure 19: Zoomed-in regions from the acquired images along with the denoised results from both the trained and traditional algorithms. First row from each half depicts the original acquired image along with its transformed and destriped image, and the original transmission image. The next two rows depict the results from the algorithms.

On first glance, all methods have managed to denoise the image to a certain degree. Starting at the top, the CP-BNN model showed the best results in denoising the background. However, the signal intensities of the DPC image were lower compared to the other models. Therefore, we adjusted the contrast so that signals were visible in the image. Nevertheless, by looking at the patched images, we can clearly see that it contains features that previously were not visible in the original noisy patches. These additional details are also apparent in all other models except for the BM3D. From then on, the question arose whether the models that take two channels as input learn to translate essential information from the absorption channel to the DPC channel. To investigate this, we trained two additional models which only take the DPC channel as input (referred by DPC after the model name in the figures). Interestingly, the predicted images from these models do not show these added details, indicating that they have been translated from the absorption channel to the DPC channel in the previous models. Another indication for this is the predicted background. All models trained on two channels are able to significantly reduce the noise in the background, while both the single channel models and the BM3D have a harder time in mitigating the noisy signal. A model that probably depends very strongly on the absorption channel is the CP-BNN. This is especially evident in the prediction of the compressed breast where the stripe artefacts in the lower field change sign and correspond more to the artefacts on the absorption channel (see red arrows). On all other models, these stripes remain the same as in the original noisy image.

We see that the P-INSIDEnet and INSIDEnet model show similar results with no major differences in neither the whole image nor the patches. Surprisingly, when comparing the sample and the background, there is a shift in the values that makes the sample look like it is floating above the background. Again, we believe this is due to the translation of the absorption channel into the DPC channel. By looking at the predictions of the U-Net, it can be seen that the images are more blurred at the edges of the specimen compared to the (P)-INSIDEnet. Inside the specimen, however, little difference can be observed. The traditional BM3D model is able to satisfactorily retrieve the high signals arising from the edges of the sample, but adds patchy and blurry artefacts in areas with lower signal intensities. The trained one-channel models, on the other hand, are able to improve the image quality while simultaneously keeping the high-frequency details. Comparing the two one-channel models, we can see no visual differences between their predictions.

4.4.1 Integrated DPC

To assess the effectiveness of the denoising performed by both the proposed algorithms and the BM3D, we investigated the image quality of the integrated denoised images. The denoised images show that while the predictions in the background may differ, they show promising signal retrieval inside the specimen. When integrating the images directly, the images become blurred and are affected by heavy stripe artefacts, which do not allow any analysis. In order to compare the predictions of the individual models, we manually segmented the specimens and only integrated over them. As described before, the predicted images from the two-channel (P)-INSIDEnet had an added bias, which separated them from the background. To ensure that the integration is not affected by a constant bias term – which in the integrated image leads to a linear increase in pixel values – we subtracted the mean value of the masked specimen from the DPC image to shift the distribution inside the specimen to zero mean. Afterwards, we integrated the images, the results of which can be seen in Fig. 20. Due to remaining acquisition artefacts after denoising in the images of the compressed specimen, we only show the results from the specimen without compression. After integration we applied the WFF to remove stripe artefacts caused by the remaining noise in the DPC image. As a wavelet basis we used again the *DB16* with a normalized standard deviation of 0.3 and a decomposition level of 8. The destriped integrated images can be seen in Fig. 20 next to their original counterpart. Fig. 21 depicts zoomed-in regions of the integrated images.

The image with the most stripes shows the original integrated image. This was to be expected, because

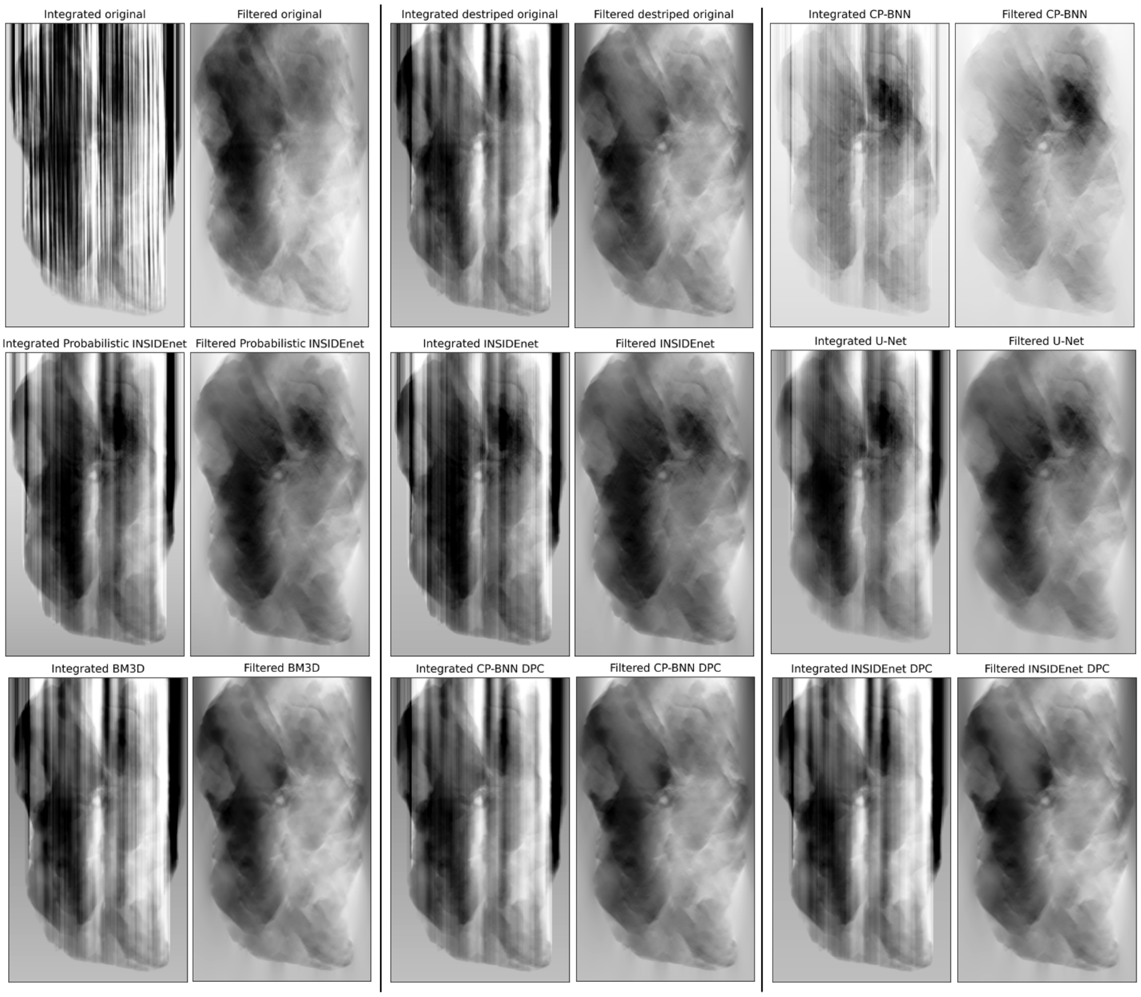

Figure 20: Integrated images of the specimen without compression. The images come in pairs of original integrated images and images after applying the WFF for stripe removal.

the DPC picture as seen in Fig. 18 has the poorest image quality. Surprisingly, the integrated image of the noisy DPC image already shows significant improvement, with the number of occurring stripes being comparable to the integrated images of the individual algorithms. What stands out is the integrated CP-BNN prediction, which looks completely different from the rest of the images. In particular, the image resembles the original absorption image, proving our previous claim about the CP-BNN. The other two-channels are less affected by the absorption channel. However, while the dark area at the top right of the sample is not visible in the filtered original image, it is visible in the absorption channel, indicating that some information is transferred from the absorption channel to the predictions as well. This can also be deduced from both the one-channel models and the BM3D, which do not have this dark region in the upper right corner. Another example of transferred information from the absorption channel is clearly depicted by the green arrows. These vessel-like structures are barely visible in the original image, nor in the denoised one-channel images, however, they become apparent in the predictions of the two-channel algorithms. It has to be further investigated to which extent a translation is benefical for the diagnosis. Clearly, a complete information leakage as it has happened in the CP-BNN is not desirable. However, small features from absorption could help in pinpointing high-frequency details and help the denoising of the image. When comparing the results of the one-channel algorithms, almost no difference can be observed – especially not much improvement over the noisy image. This is surprising, since in the predicted DPC channels all algorithms were able to significantly reduce the noise. When then comparing the results of the one-channel models with the original and noisy images in the zoomed-in images, we

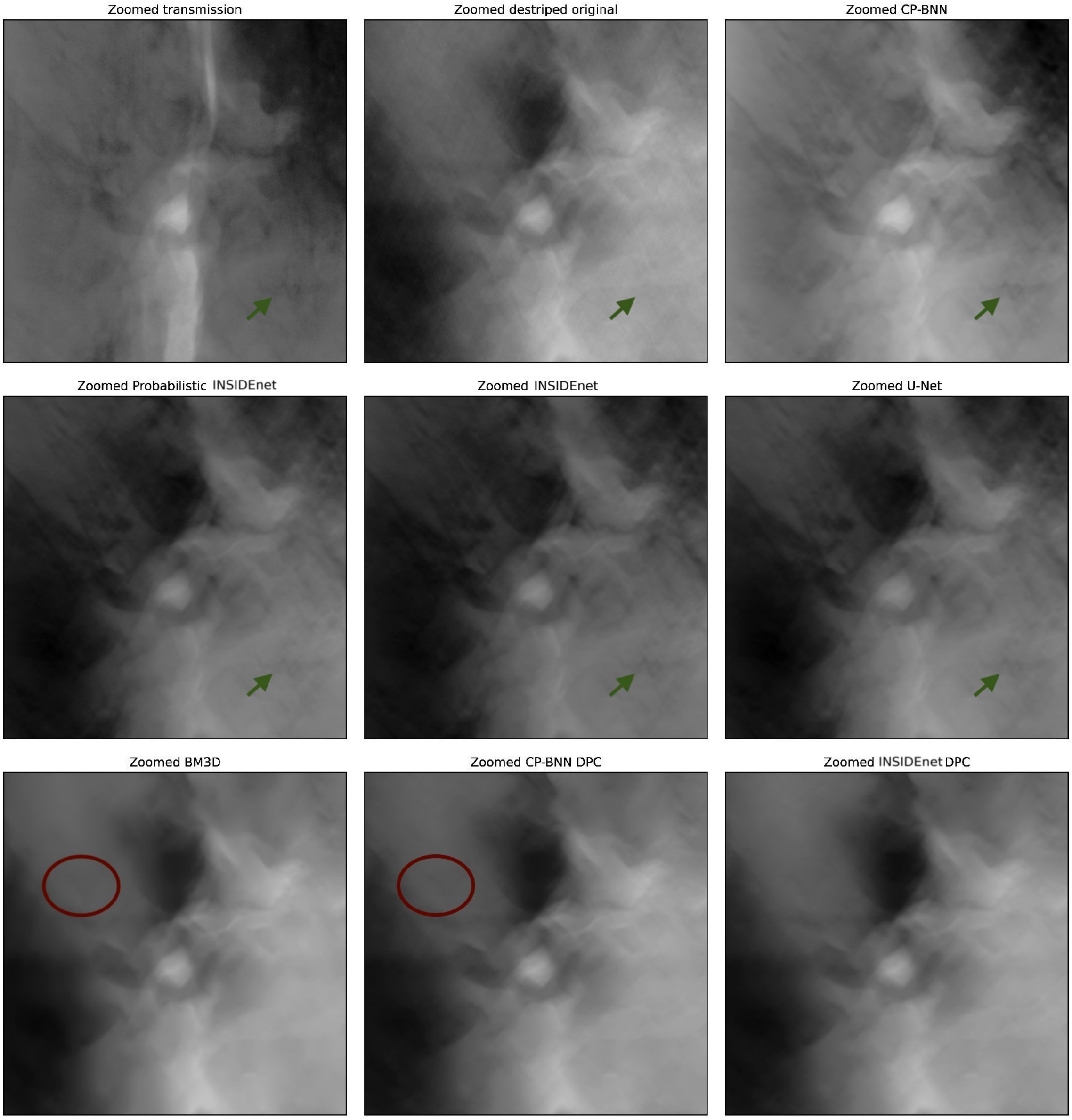

Figure 21: Zoomed-in images of the integrated DPC images from the Philips Microdose System. The original image can be seen in Fig. 20. The green arrows point to added translated information from the absorption channel. The red circles show the detail preservation from deep learning models compared to BM3D.

can see differences: The noisy image has a fibrous, washed-out texture that is superimposed on the whole image, while the denoised images are smoother and show the details more clearly. By looking very carefully, minor differences between the deep learning models and the BM3D can be perceived – indicated by the red circles in the image. This is the consequence of the missing high frequency detail conservation of the BM3D model in the DPC denoising. Although the image quality of the DPC images improved after denoising, this did not result in the same effect in the integrated image. Consequently, the DPC images have to be perfectly denoised to be able to compute high quality PC images.

5 Discussion

The aim of this thesis was to investigate image denoising in DPC images using deep learning algorithms with the goal of retrieving a high-quality phase contrast image. This image is especially relevant in clinical settings, as it can result in higher soft tissue contrast compared to absorption, but without sacrificing spatial resolution. To provide clinics with this advantage, a GI system was built into a Philips Microdose Mammography system. However, images attained with in-vivo patients are required to comply with clinical regulations, which result in significantly higher noise levels in the images. Thus, strong denoising tools are crucial in obtaining the desired high-quality phase contrast image from the collected DPC image. To this end, we have developed a set of deep learning algorithms and compared them to already established methods such as the BM3D and U-Net.

In chapter 4.1, we analyzed the denoising results from models trained in a supervised fashion. Immediately apparent was the inability of the BM3D filter to denoise DPC images while keeping relevant high frequency features. This becomes clear both visually (see Fig. 25) as well as when checking the numerical values (see Tab. 2), and even more so when regarding the intensity profile on a single image (see Fig. 9). This demonstrates the need for more sophisticated algorithms that can cope with more complex noise. We were able to show that data-driven models significantly improved image quality, which is especially impressive as they were able to keep up with the U-Net when evaluated on our simulated data. Unfortunately, all models failed to retrieve very detailed information from the noisy image, as seen in the patch images from Fig. 25, and again more clearly in the respective intensity profiles. In fact, the algorithms were capable of following the intensity profiles, but underestimated the amplitudes by up to a factor of 2, which in turn led to information loss. This problem may arise from the MSE loss and the Laplacian distribution of the images. In a clean image, relatively few pixel values deviate from zero. Particularly the background takes up a significant proportion of the entire image where pixel values ideally should be zero. When the MSE is calculated, it averages the error in the pixels over the whole image. The training then tries to tune the weights to optimize the image globally. Since most of the values are close to zero, it might tend to pull the prediction close to zero as well.

The idea behind the INSIDEnet was finding a good trade-off between performance and interpretability/robustness. All steps follow a clear mathematical reasoning, which is missing in the U-Net. By moving back and forth from the image domain to a learned transformation domain, it is possible to visualize the denoising after each filtering step (see Appendix A.5), leading to an explainable deep learning model. Visualizing the single filtering steps in a U-Net architecture is not possible since the output of single layers consists of multiple channels and do not lie in an interpretable image space. By comparing the two architectures, however, similarities can be observed. The INSIDEnet has a similar "U"-structure as the U-Net. In addition, the image processing consists of linear matrix multiplication followed by a non-linear activation, where the latter is motivated by the proximal operator of the l_0 norm [19]. While the INSIDEnet is already quite similar to the U-Net, we went one step further and integrated the decoder structure missing in the INSIDEnet architecture to create the CP-BNN model. The main idea was to combine the pre-filtered channels into one single channel by applying multiple convolutional layers in every scale and eventually predicting the filtered denoised DPC image.

Continuing on the path of explainable deep learning models, we added a Bayesian perspective on the transformation matrices of the INSIDEnet and in the convolutional layers of the decoder in the CP-BNN. This allowed us to inspect not only the denoised image but also the uncertainty of the model as depicted in Fig. 10. While all transformation matrices in the INSIDEnet follow a variational distribution, only single convolutional layers in the CP-BNN are modelled from a variational family. This means that only the uncertainty of the fusion and the one after the pre-filtering of the encoder was modelled in the CP-BNN.

Both models were highly uncertain at the boarder of the breast and in regions with high absolute intensity amplitude such as edges. This can be deduced from the intrinsic noise statistics. The edges inside the breast were simulated as highly scattering with a low DF signal and thus, following Eq. (8), led to high uncertainty and high noise amplitudes. Therefore, we believe that the models were most uncertain inside the breast where the DF signal was low. Following this rationale, we can also explain the high uncertainty of pixels containing microcalcifications. Microcalcification are highly absorbing and scattering which lead to a low transmission and DF signal, and consequently to an increased uncertainty. Another reason may be the loss landscape and the susceptibility of the model to small changes in the weights. In a Bayesian neural network the weights are modelled as a distribution. During prediction, we try to approximate the predictive posterior using Monte Carlo sampling over the trained distribution of the weights (see Eq. (35)). This means that each prediction is performed with different weights. The uncertainty is then equivalent to the variance of the predicted images. Having a smooth convex loss landscape with a clear minimum makes the model less prone to small changes in the weights and thus results in similar predictions. However, if the learned distribution falls in a peaked, non-smooth local optimum, even slight deviations lead to a significant increase of the loss and stronger deviations in the predictions. We thus believe that some areas in the image are more susceptible to small deviations than others, leading to the uncertainty seen in the images.

While the CP-BNN had low uncertainty in the background, the P-INSIDEnet model had a constant uncertainty bias. To move from the image space to the sparse transformation domain we performed an Einstein sum of the matrix Q with all stacked patches. In the transformation domain, the entries of the transformed tensor are thresholded by the deterministic threshold constant Γ. Due to the probabilistic modelling of B – which leads to the matrix Q – and Γ, the entries in the tensor are differently thresholded, which eventually affects the whole image after back-transformation. Interestingly, the uncertainty in the P-INSIDEnet showed similarities with the uncertainty image calculated using Eq. (8). The high uncertainty is most prominent at the minima of the flat-field visibility map.

Training our networks in a supervised fashion with clean and noisy image pairs has yielded promising results. Yet, in clinical settings it is not possible to obtain clean images, nor is it possible to extract the same projection of a breast twice in order to train a network in a N2N fashion without jeopardizing the patient's safety or violating clinical standards. Even if it was possible to obtain two noisy projections, the images would have to be perfectly registered, which in real-world physical settings is unlikely due to patient movement or machine vibrations. Therefore, a pre-processing step would be needed prior to inputting the data into our algorithms. Following the theory from N2N and NAC, and using the well-known noise statistical relations in the individual contrast channels, we were able to train our networks in a supervised fashion using synthetic generated noise overlapped on the existing noisy image and using this noisy image as target. Yet, this method comes with a disadvantage: it only performs well when dealing with images with low noise level – in our case requiring at least an average photon count of 3000.

Numerically, our models performed worse than the BM3D but were better in retaining high-frequency features with small amplitude, where the BM3D fails. The BM3D is dependent on a single parameter provided by the user – namely the standard deviation of the noise. Especially in data with heteroscedastic noise, where the noise amplitudes vary spatially in the image, this can lead to information loss. Deep learning techniques overcome this problem and learn to distinguish the noise levels in the image. Unlike the supervised setting, where we used 440 images for training, we only used 320 in the unsupervised setting. This was done to be more memory efficient, since for the calculations of the uncertainty we would load an extra image channel (DF image) into the pipeline. Another reason for less training images was to see if our models can be trained with less data, since not much real data is available yet. Surprisingly, our models led to superior results compared to the widely used U-Net model. As stated in [19], the INSIDEnet imposes a strong inductive bias on the denoising problem which allows it to be trained with very limited data. From this, it may be concluded that the U-Net requires more data to perform at equal level as the INSIDEnet and CP-BNN. Another reason is the architecture: As soon as the images propagate through the U-Net, they are divided into different image channels, which most likely are in different spaces and only reach the image space again in the last layer. The INSIDEnet, on the other hand, goes back into the image space after each filter step. We therefore believe that it can better deal with little data as well as the NAC method. In contrast to the U-Net, the CP-BNN was trained with an intermediate-loss term, to ensure that after each layer in the encoder the images were less perturbed by noise. The convolutional layers in the skip connections and in the encoder therefore needed to deal with less noise and could thus use their expressive power on "simpler" data.

Perfect denoising is a challenging task and almost impossible to achieve if the noise level is high. We have tried various architectures to see whether they are capable to denoise our images so that a clean integration can be performed. While the models showed promising results in retrieving the clean signal information, the predicted images were still corrupted by noise, which became most apparent after integration. Still, due to the prior denoising, the integrated images showed significantly less blurring and stripe artefacts, which enabled us to draw conclusions about the interior structure of the breast. To further increase image quality, we used the WFF to get rid of the remaining stripes in the images.

With our proposed algorithms and the WFF, we were able to retrieve images without any major disturbing stripe artefacts inside the breast, but some wide stripes with low amplitude remained on the overall image. While the original integrated image and the BM3D denoised image showed significantly less stripes after applying the WFF, blurred structures, strong undulating stripes, and noise remained. Regions with high noise amplitudes were especially affected. Our deep learning models in combination with the WFF in turn provided high quality images, which demonstrates the potential of deep learning techniques compared to traditional denoising algorithms. However, tuning the WFF to perform well can be a time-consuming task. It depends on multiple factors: the type of the stripes, the width, the bias, and the image size. While we adapted the parameters of the WFF to fit one predicted image from the deep learning models, we believe that finding new parameters for every single prediction would result in better destriped images. However, the idea of this work was to bring fast and reliable denoising algorithms to clinics, where no adjustment of parameters is necessary. Consequently, we only used one set of parameters to destripe all predicted images. One way to overcome parameter tuning is to implement a fully end-to-end deep learning technique from noisy DPC images to clean phase contrast images using the ideas from WFF. For example, it could be possible to train the wavelet coefficients instead of preselecting the wavelet prior for the transformation. Furthermore, one could train the standard deviation in the Gaussian window used in the FFT, or even make it adaptive to the data, which would allow for adjustments to the specific type of stripes in the image.

Finally, we have evaluated the algorithms on real data acquired on the Philips Microdose System at the University Hospital Zurich. One of the issues with our deep learning models was that the image values had to be scaled to a range equivalent to the one from the training set. The obtained images however, had a dynamic range in the order of 109, which made it hard to correctly calibrate them. We thus found that the best denoising results were obtained when scaling the absorption images to the same range as in the training and scaling the DPC images so that the histogram of the new images becomes similar to the one used in training. Yet, due to the bimodal distribution of the DPC histogram in the acquired images, further pre-processing had to be performed to ensure correct denoising performance. Interestingly, transforming the histogram to a unimodal distribution improved the image quality notably where stripes from the acquisition have been removed or diminished. The nature of these stripes has yet to be investigated; however, we assume that they might result from grating imperfections and misalignments. After destriping the remaining stripes in the transformed image, the calculated histogram resembled the one from training. Nevertheless, correctly calibrated data in $[-\pi, \pi]$ as well as $[0, 1]$ for absorption (and DF – in case of NAC), is crucial for the algorithms to function properly.

Although our models were trained on simulated data, the denoising results on real data were surprisingly good. Each model was able to mitigate the noise and improve image quality. The results from the two-channel models indicated that information from the absorption channel leaked to the DPC image – the CP-BNN was most affected by this. Our reason for training the models with both channels was that they gave better results overall on all training, validation, and test sets. Here we already suspected that there would be a flow of information between absorption and DPC. However, we did not expect this information to be as high as can be seen in the CP-BNN. Consequently, we believe that our simulated projections in absorption and DPC are physically too similar. In fact, we assume that the gradient of phase and absorption are similar up to scalar factors. This means that the network, especially the CP-BNN in the fusion channels, learns to calculate the gradient of the absorption channel and scale it to the correct range of the DPC image. We found that the denoising capacities in the background is strongly dependent of the scaling of the absorption channel. We thus believe that choosing the right scaling could improve image quality and even make it less prone to information leakage from the absorption channel into the DPC channel. Yet, finding the right scale is challenging. This difficulty has already resonated in the scaling of the DPC channel, where, however, the challenge is less pronounced due to the approximation to a Laplacian distribution. It is now to be determined to which degree information leakage from absorption to DPC is justifiable and/or desirable.

To avoid information being leaked to the DPC channel, additional models have been trained to only take the DPC image as input. These models, however, showed worse results compared to the two-channel models when evaluated on the test set (see Appendix A.6). However, we believe that their denoising capacities are more effective since they have to perform without any information from the absorption channel. On the real images, they were able to significantly reduce the noise inside the specimen. This could be observed especially in the zoomed-in regions. Particularly noteworthy is the comparison to the BM3D. The deep learning models are able to extract the information from the noisy image without adding patchy structures or blurring. This shows the superiority of the deep learning models over conventional denoising tools, even though they were trained on purely simulated data. We believe that, with more real data, it should be possible to fine-tune our trained models in either a NAC-fashion or N2V-fashion, as we suspect that the weight distribution only requires slight adjustment to denoise the current images satisfactorily.

Lastly, we compared the results of the integrated denoised DPC images. Segmentation of the region of interest, as we did in our experiment, is generally not possible in clinical settings. On the one hand, the boundaries of the regions are sometimes difficult to identify due to noise and acquisition artefacts. On the other hand, it is time-consuming to perform a manual segmentation. Simple automatic segmentation such as thresholding would be a possibility, but due to the zero centering of the data it is challenging to find an exact threshold, especially if each object is different. It should be emphasized that by integrating the predicted images with the background, we would have gotten worse results and heavier artefacts than seen on the masked and integrated images. This is mainly due to the fact that no algorithm was able to denoise the complete background and thus remaining noise would result in heavier stripe artefacts and blurring in the direction of integration. Surprisingly, what was supposed to be a pre-processing step with histogram transformation and destriping, turned out to be a big advantage not only in the DPC channel, but even more so in the integrated image. The original integrated image showed strong stripe artefacts, which could be removed with the WFF. However, it would not have been possible if we had integrated over the background as well. The noisy integrated image, on the other hand, has significantly less stripe artefacts, from which a wide amount of information can be retrieved. Remarkably, the results of the algorithms did not provide any outstanding improvements for the integrated image, which is counter-intuitive as the DPC images themselves showed less noise. Even more astonishing are the similarities between the integrated noisy image and the images coming from the one-channel model, where basically no difference can be perceived. By looking at the zoomed-in images, it is possible to see that the noisy PC image has some smearing inside the specimen, where the filtered ones do not. However, apart from that, the images look the same and would most likely not lead to a misinterpretation of the image. Our assumption about the information leakage with the CP-BNN model were proven after we integrated the predicted DPC image, which resulted in an integrated image that looked very similar to the original absorption image. This concludes that although the model has shown a better performance on our simulated data, it is not trustworthy on real data. However, with better simulations and/or better network architecture, the one stream information flow from absorption to DPC could be mitigated.

6 Conclusion

Deep learning denoising algorithms have become increasingly useful in various imaging areas. However, until recently only one paper has investigated denoising and enhancement of the image quality from DPC images [68]. Their approach differed from ours in that they directly tackled the problem during the signal retrieval from the phase-stepping curve with subsequent image denoising using known and widely used deep learning architectures. Our approach, on the other hand, tackles the problem in the projection domain, thus after the signal retrieval, using newly developed architectures with a clear and mathematical reasoning. Additionally, we added a Bayesian perspective to model the uncertainty in the prediction. While we were able to train our models in a supervised and even unsupervised way with satisfactory results, the predicted images from the models themselves did not deviate much from each other. This suggests that with both the newly developed methods and the state-of-the-art U-Net architecture, no perfect denoising can be achieved, while still keeping all the details from the original image. Yet, a perfect denoising is crucial to be able to generate high-quality phase contrast images, which has not only proven itself in the simulations but also on real data. We thus propose to go into a different direction when continuing this work. One possibility would be to train a variational network, which has become popular in image reconstruction [69–71]: Instead of first predicting the denoised DPC image and performing the integration afterwards, it would directly learn to reconstruct the phase contrast image from the noisy DPC image by exploiting the knowledge of the differentiation as a forward operator. The denoised DPC image, if desired, could then be directly retrieved by taking the gradient of the predicted phase contrast image. We believe that with this approach, it would be possible to retrieve the phase contrast image, even if it was trained on simulated data. This would pave the way to model-based deep learning algorithms and more reliable and interpretable models.

While we successfully implemented a Bayesian view on our models, we believe that this extra feature adds only little value. Knowing the uncertainty in the prediction can be crucial in areas where decisions have to be made, such as medical diagnostics. Being able to assess uncertainty in our denoised predictions, however, did not improve the image quality in any sense. Yet, it gave us the ability to better interpret the decision made by the single algorithms and help understand where the stripe artefacts may originate from when integrating the image. This Bayesian aspect may be more useful when combined with a variational network. The uncertainty could then be helpful to classify how well the model approximation, such as the forward operator, and the regularizations work, and which uncertainties result intrinsically out of the reconstruction of the variational model.

Applying the models on real data has shown impressive results, even though they were trained on simulated data. Yet, after integration and segmentation, the added value from the previous denoising has greatly diminished. The integrated predicted images only slightly improved the visual quality of the images compared to the noisy image. Therefore, we believe that the background noise in DPC has a significant negative effect on the quality of the images. Using only the region-of-interest shrinks the amount of blurriness and stripe artefacts in the integrated images. Hence, we believe that with an automatic segmentation model, the quality of the phase contrast images could be greatly improved. The problem of noise in the background would be completely eliminated and the noise inside the sample could be reduced with our proposed models.

In conclusion, our models have not only shown promising results in denoising DPC images in both the simulated case and on real samples but were also fast in their predictions. Moreover, the results on real images – especially the ones from the CP-BNN – have shown the importance of accurate simulations for our task. These should be simulated even better in the future. Nevertheless, it should be emphasized that these models have only been trained on simulated images and can lead to better results if trained or fine-tuned on real data. Finally, to build an automatic pipeline from DPC to phase contrast, better models and algorithms are required.

References

[1] Nadia Harbeck and Michael Gnant. “Breast cancer”. In: *The Lancet* 389.10074 (2017), pp. 1134–1150. ISSN: 0140-6736.

[2] WHO. *Cancer*. URL: `https://www.who.int/news-room/fact-sheets/detail/cancer`. (accessed: 19-Juli-2021).

[3] Magnus Løberg et al. “Benefits and harms of mammography screening”. In: *Breast Cancer Research* 17.1 (2015), pp. 1–12.

[4] Carl D’Orsi, L Bassett, S Feig, et al. “Breast imaging reporting and data system (BI-RADS)”. In: *Breast imaging atlas* (2018).

[5] Shu-Ang Zhou and Anders Brahme. “Development of phase-contrast X-ray imaging techniques and potential medical applications”. In: *Physica Medica* 24.3 (2008), pp. 129–148.

[6] Zhentian Wang et al. “Non-invasive classification of microcalcifications with phase-contrast X-ray mammography”. In: *Nature communications* 5.1 (2014), pp. 1–9.

[7] Marco Stampanoni et al. “The first analysis and clinical evaluation of native breast tissue using differential phase-contrast mammography”. In: *Investigative radiology* 46.12 (2011), pp. 801–806.

[8] Nik Hauser et al. “A study on mastectomy samples to evaluate breast imaging quality and potential clinical relevance of differential phase contrast mammography”. In: *Investigative radiology* 49.3 (2014), pp. 131–137.

[9] Carolina Arboleda et al. “Towards clinical grating-interferometry mammography”. In: *European radiology* 30.3 (2020), pp. 1419–1425.

[10] Vincent Revol et al. “Noise analysis of grating-based x-ray differential phase contrast imaging”. In: *Review of Scientific Instruments* 81.7 (2010), p. 073709.

[11] Antoni Buades, Bartomeu Coll, and J-M Morel. “A non-local algorithm for image denoising”. In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*. Vol. 2. IEEE. 2005, pp. 60–65.

[12] Kostadin Dabov et al. “Image denoising by sparse 3-D transform-domain collaborative filtering”. In: *IEEE Transactions on image processing* 16.8 (2007), pp. 2080–2095.

[13] Kai Zhang et al. “Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising”. In: *IEEE transactions on image processing* 26.7 (2017), pp. 3142–3155.

[14] Jaakko Lehtinen et al. “Noise2noise: Learning image restoration without clean data”. In: *arXiv preprint arXiv:1803.04189* (2018).

[15] Alexander Krull, Tim-Oliver Buchholz, and Florian Jug. “Noise2void-learning denoising from single noisy images”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 2129–2137.

[16] Abdelrahman Abdelhamed, Radu Timofte, and Michael S Brown. “Ntire 2019 challenge on real image denoising: Methods and results”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 2019, pp. 0–0.

[17] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. “Deep image prior”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 9446–9454.

[18] Laurent Valentin Jospin et al. “Hands-on Bayesian Neural Networks–a Tutorial for Deep Learning Users”. In: *arXiv preprint arXiv:2007.06823* (2020).

[19] Stefano van Gogh, Zhentian Wang, and Michal Rawlik. “Multiscale multichannel stacked orthogonal transform learning for image denoising in Grating Interferometry Breast CT”. In: *under Review in Medical Physics* ().

[20] Xiao-Jiao Mao, Chunhua Shen, and Yu-Bin Yang. “Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections”. In: *arXiv preprint arXiv:1603.09056* (2016).

[21] Beat Münch et al. “Stripe and ring artifact removal with combined wavelet—Fourier filtering”. In: *Optics express* 17.10 (2009), pp. 8567–8591.

[22] Peter Modregger et al. “-Grating-Based X-Ray Phase-Contrast Imaging”. In: *Emerging Imaging Technologies in Medicine.* CRC Press, 2012, pp. 62–75.

[23] Shu-Ang Zhou and Anders Brahme. “Development of phase-contrast X-ray imaging techniques and potential medical applications”. In: *Physica Medica* 24.3 (2008), pp. 129–148.

[24] UetMHART Bonse and M Hart. “An X-ray interferometer”. In: *Applied Physics Letters* 6.8 (1965), pp. 155–156.

[25] TJ Davis and AW Stevenson. “Direct measure of the phase shift of an x-ray beam”. In: *JOSA A* 13.6 (1996), pp. 1193–1198.

[26] A Snigirev et al. “On the possibilities of x-ray phase contrast microimaging by coherent high-energy synchrotron radiation”. In: *Review of scientific instruments* 66.12 (1995), pp. 5486–5492.

[27] Timm Weitkamp et al. “X-ray phase imaging with a grating interferometer”. In: *Optics express* 13.16 (2005), pp. 6296–6304.

[28] Maria Büchner. “Towards the development of an X-Ray phase contrast breast CT scanner”. PhD thesis. ETH Zurich, 2019.

[29] Franz Pfeiffer et al. “Phase retrieval and differential phase-contrast imaging with low-brilliance X-ray sources”. In: *Nature physics* 2.4 (2006), pp. 258–261.

[30] Atsushi Momose et al. “Demonstration of X-ray Talbot interferometry”. In: *Japanese journal of applied physics* 42.7B (2003), p. L866.

[31] Rainer Raupach and Thomas Flohr. “Performance evaluation of x-ray differential phase contrast computed tomography (PCT) with respect to medical imaging”. In: *Medical physics* 39.8 (2012), pp. 4761–4774.

[32] Thomas Weber et al. “Noise in x-ray grating-based phase-contrast imaging”. In: *Medical physics* 38.7 (2011), pp. 4133–4140.

[33] T Weber et al. “Measurements and simulations analysing the noise behaviour of grating-based X-ray phase-contrast imaging”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 648 (2011), S273–S275.

[34] Saptarshi Sengupta et al. “A review of deep learning with special emphasis on architectures, applications and recent trends”. In: *Knowledge-Based Systems* 194 (2020), p. 105596.

[35] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning.* MIT press, 2016.

[36] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. “Imagenet classification with deep convolutional neural networks”. In: *Communications of the ACM* 60.6 (2017), pp. 84–90.

[37] Yann LeCun, Koray Kavukcuoglu, and Clément Farabet. “Convolutional networks and applications in vision”. In: *Proceedings of 2010 IEEE international symposium on circuits and systems.* IEEE. 2010, pp. 253–256.

[38] Christopher M Bishop. *Pattern recognition and machine learning.* springer, 2006.

[39] Sebastian Ruder. "An overview of gradient descent optimization algorithms". In: *arXiv preprint arXiv:1609.04747* (2016).

[40] Max Welling and Yee W Teh. "Bayesian learning via stochastic gradient Langevin dynamics". In: *Proceedings of the 28th international conference on machine learning (ICML-11)*. Citeseer. 2011, pp. 681–688.

[41] Yi-An Ma et al. "Sampling can be faster than optimization". In: *Proceedings of the National Academy of Sciences* 116.42 (2019), pp. 20881–20885.

[42] Chain Monte Carlo. "Markov chain monte carlo and gibbs sampling". In: *Lecture notes for EEB* 581 (2004).

[43] Carl Edward Rasmussen. "Gaussian processes in machine learning". In: *Summer school on machine learning*. Springer. 2003, pp. 63–71.

[44] Rajesh Ranganath, Sean Gerrish, and David Blei. "Black box variational inference". In: *Artificial intelligence and statistics*. PMLR. 2014, pp. 814–822.

[45] Michalis Titsias and Miguel Lázaro-Gredilla. "Doubly stochastic variational Bayes for non-conjugate inference". In: *International conference on machine learning*. PMLR. 2014, pp. 1971–1979.

[46] Michael I Jordan et al. "An introduction to variational methods for graphical models". In: *Machine learning* 37.2 (1999), pp. 183–233.

[47] Diederik P Kingma and Max Welling. "Auto-encoding variational bayes". In: *arXiv preprint arXiv:1312.6114* (2013).

[48] Viren Jain and Sebastian Seung. "Natural image denoising with convolutional networks". In: *Advances in neural information processing systems* 21 (2008), pp. 769–776.

[49] John H Hubbell and Stephen M Seltzer. *Tables of X-ray mass attenuation coefficients and mass energy-absorption coefficients 1 keV to 20 MeV for elements Z= 1 to 92 and 48 additional substances of dosimetric interest.* Tech. rep. National Inst. of Standards and Technology-PL, Gaithersburg, MD (United ..., 1995.

[50] Srinivasan Vedantham and Andrew Karellas. "X-ray phase contrast imaging of the breast: Analysis of tissue simulating materials a". In: *Medical physics* 40.4 (2013), p. 041906.

[51] Burton L Henke, Eric M Gullikson, and John C Davis. "X-ray interactions: photoabsorption, scattering, transmission, and reflection at E= 50-30,000 eV, Z= 1-92". In: *Atomic data and nuclear data tables* 54.2 (1993), pp. 181–342.

[52] Wim Van Aarle et al. "The ASTRA Toolbox: A platform for advanced algorithm development in electron tomography". In: *Ultramicroscopy* 157 (2015), pp. 35–47.

[53] Saiprasad Ravishankar and Yoram Bresler. "Learning doubly sparse transforms for images". In: *IEEE Transactions on Image Processing* 22.12 (2013), pp. 4598–4612.

[54] Asher Trockman and J Zico Kolter. "Orthogonalizing Convolutional Layers with the Cayley Transform". In: *arXiv preprint arXiv:2104.07167* (2021).

[55] Nicolas Pielawski and Carolina Wählby. "Introducing Hann windows for reducing edge-effects in patch-based image segmentation". In: *PloS one* 15.3 (2020), e0229839.

[56] Harold Christopher Burger and Stefan Harmeling. "Improving denoising algorithms via a multi-scale meta-procedure". In: *Joint Pattern Recognition Symposium*. Springer. 2011, pp. 206–215.

[57] Anqi Wu et al. "Deterministic variational inference for robust bayesian neural networks". In: *arXiv preprint arXiv:1810.03958* (2018).

[58] Andrei Atanov et al. "The deep weight prior". In: *arXiv preprint arXiv:1810.06943* (2018).

[59] Ranganath Krishnan, Mahesh Subedar, and Omesh Tickoo. “Efficient priors for scalable variational inference in Bayesian deep neural networks”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*. 2019, pp. 0–0.

[60] Alex Graves. “Practical variational inference for neural networks”. In: *Advances in neural information processing systems*. Citeseer. 2011, pp. 2348–2356.

[61] Diederik P Kingma and Jimmy Ba. “Adam: A method for stochastic optimization”. In: *arXiv preprint arXiv:1412.6980* (2014).

[62] Jun Xu et al. “Noisy-as-clean: learning self-supervised denoising from corrupted image”. In: *IEEE Transactions on Image Processing* 29 (2020), pp. 9316–9329.

[63] Patrick Billingsley. *Probability and measure*. John Wiley & Sons, 2008.

[64] Sangyoon Lee, Min Seok Lee, and Moon Gi Kang. “Poisson–Gaussian noise analysis and estimation for low-dose X-ray images in the NSCT domain”. In: *Sensors* 18.4 (2018), p. 1019.

[65] S. E. Tavares. “A Comparison of Integration and Low-Pass Filtering”. In: *IEEE Transactions on Instrumentation and Measurement* 15.1/2 (1966), pp. 33–38. DOI: `10.1109/TIM.1966.4313498`.

[66] Martın Abadi et al. “TensorFlow: Large-scale machine learning on heterogeneous systems, software available from tensorflow. org (2015)”. In: *URL https://www.tensorflow.org* (2015).

[67] Yuanhao Gong and Ivo F Sbalzarini. “Image enhancement by gradient distribution specification”. In: *Asian Conference on Computer Vision*. Springer. 2014, pp. 47–62.

[68] Yongshuai Ge et al. “Enhancing the X-ray differential phase contrast image quality with deep learning technique”. In: *IEEE Transactions on Biomedical Engineering* 68.6 (2020), pp. 1751–1758.

[69] Erich Kobler et al. “Variational networks: connecting variational methods and deep learning”. In: *German conference on pattern recognition*. Springer. 2017, pp. 281–293.

[70] Kerstin Hammernik et al. “Learning a variational network for reconstruction of accelerated MRI data”. In: *Magnetic resonance in medicine* 79.6 (2018), pp. 3055–3071.

[71] Melanie Bernhardt et al. “Training variational networks with multidomain simulations: Speed-of-sound image reconstruction”. In: *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control* 67.12 (2020), pp. 2584–2594.

A Appendix

A.1 Masks and flats

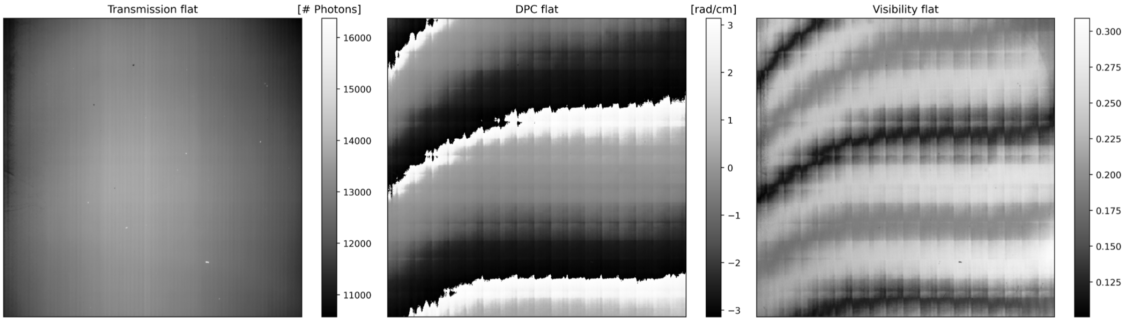

Figure 22: Used flats for the simulation of the PSC curve and the generation of the individual projections.

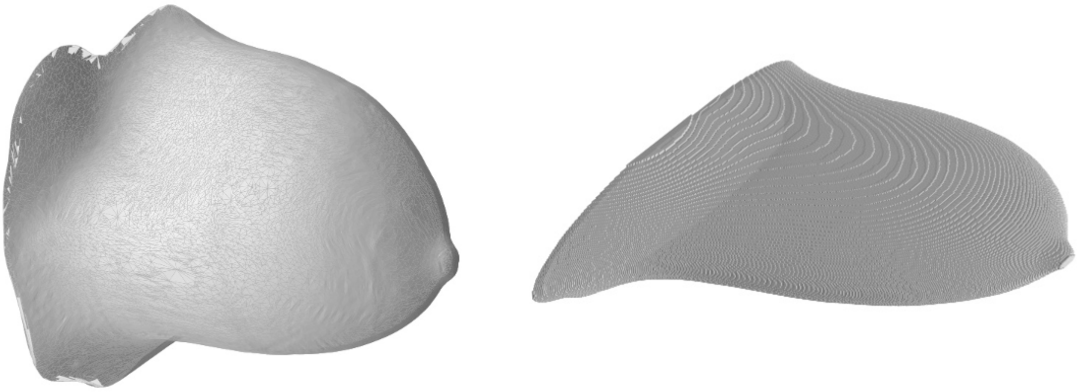

Figure 23: 3D mask of the used breast. Left is the original 3D mesh acquired on the Breast-CT system of the University Hospital Zurich. On the right is the simulated squeezed breast created with an affine transformation and morphological operations.

A.2 Absorption Results

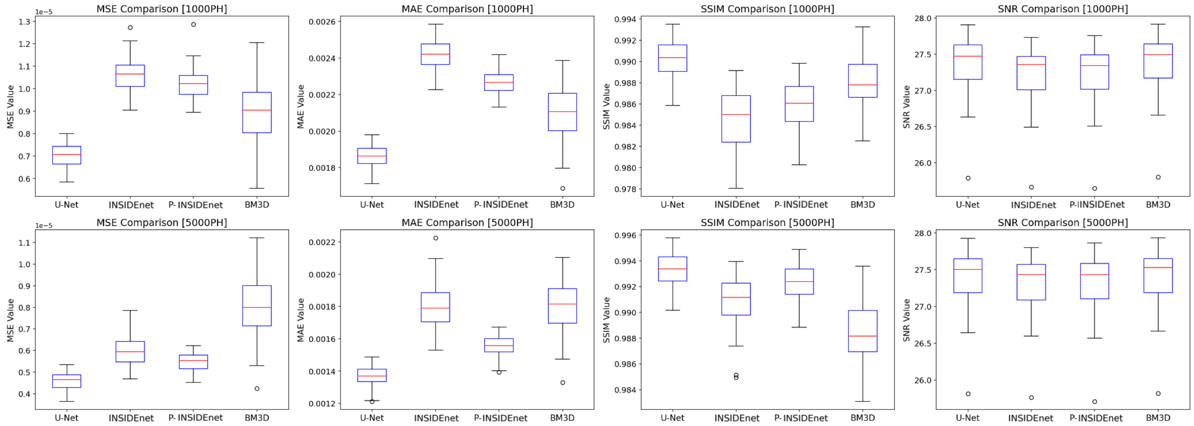

Figure 24: Absorption results: MSE, MAE, SSIM and SNR over the whole test set evaluated on all two-channel prediction models. On the top: results using an average photon count of 1000. On the bottom: results with an average photon count of 5000.

	MSE	MAE	SSIM	SNR
1000 Photons				
Input	2.32e-3 (1.77e-3)	2.25e-2 (9.38e-3)	0.958 (7.51e-3)	24.9 (0.282)
U-Net	**7.05e-6** (5.12e-7)	**1.86e-3** (6.2e-5)	**0.990** (1.66e-3)	27.35 (0.376)
INSIDEnet	1.03e-5 (6.48e-7)	2.27e-3 (6.03e-5)	0.986 (2.29e-3)	27.21 (0.375)
P-INSIDEnet	1.06e-5 (7.53e-7)	2.43e-3 (9.38e-5)	0.984 (2.74e-3)	27.21 (0.369)
BM3D	9.06e-6 (1.42e-6)	2.1e-3 (6.2e-5)	0.988 (2.13e-3)	**27.361** (0.374)
5000 Photons				
Input	4.179e-5 (1.187e-6)	5.08e-3 (8.321e-5)	0.92 (1.31e-2)	26.8 (0.351)
U-Net	**4.602e-6** (3.971e-7)	**1.371e-3** (6.107e-5)	**0.994** (1.188e-3)	27.37 (0.374)
INSIDEnet	5.494e-6 (4.06e-7)	1.557e-3 (6.249e-5)	0.992 (1.303e-3)	27.30 (0.382)
P-INSIDEnet	6.017e-6 (6.674e-7)	1.802e-3 (1.41e-4)	0.991 (1.906e-3)	27.294 (0.378)
BM3D	8.101e-6 (4.178e-5)	1.803 (1.6e-4)	0.989 (2.127e-3)	**27.373** (0.374)

Table 5: Absorption results: Denoising results summarised from Fig. 24 in mean and standard deviation (in parentheses) from all metrics across all 64 test images. Outlined next to the denoising results are the original metrics values between clean image and noisy image (here referred to as Input). The best performing value of each model is highlighted.

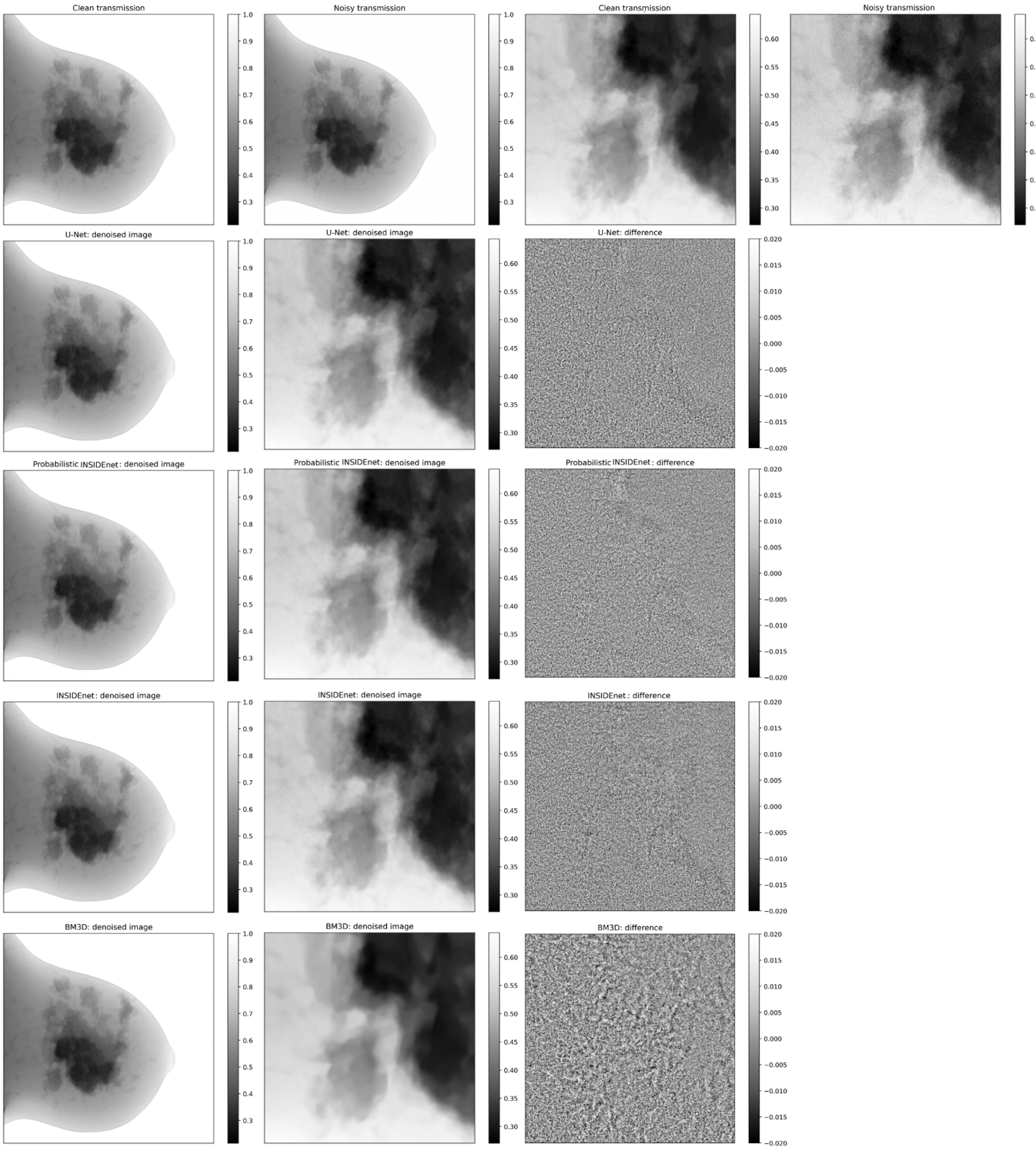

Figure 25: Denoising results on the whole breast and zoomed-in region from a absorption projection over all two-channel models and the difference between the original noisy and predicted image. Top row left: Clean and noisy image and patches, respectively. In subsequent rows are the predicted images and zoomed-in regions along with the difference from P-INSIDEnet, INSIDEnet, U-Net, and BM3D.

A.3 Unsupervised Results

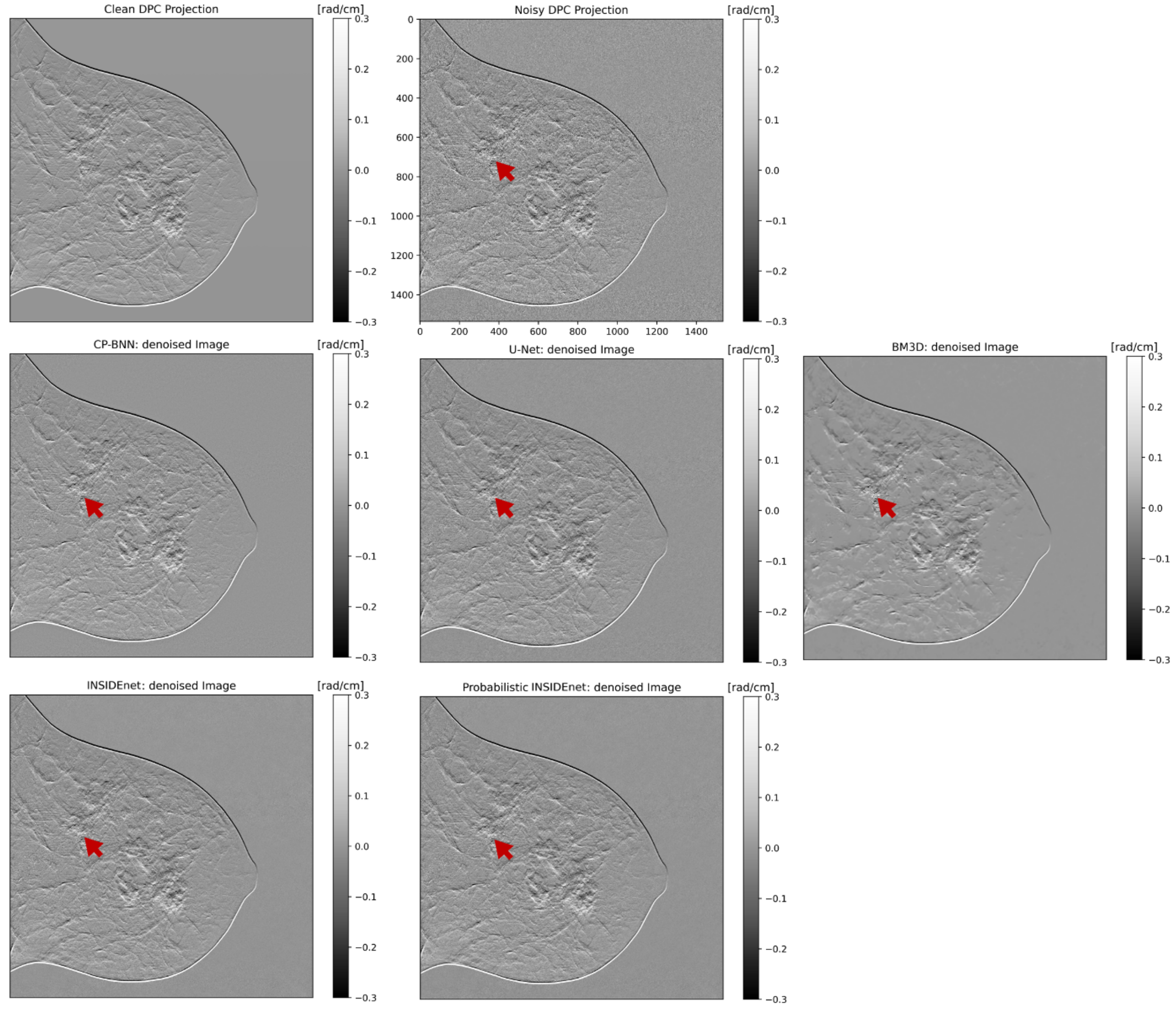

Figure 26: Denoising results visualized on the whole projection, from which zoomed-in regions have been taken in Fig. 13. Note that the dynamic range is smaller to better visualize high noise amplitudes. Arrows indicate regions where the visibility flat has a minimum and therefore the noise levels are high. Each algorithm fails to retrieve the clean signal hidden behind these high corrupted regions.

A.4 Destriping Results

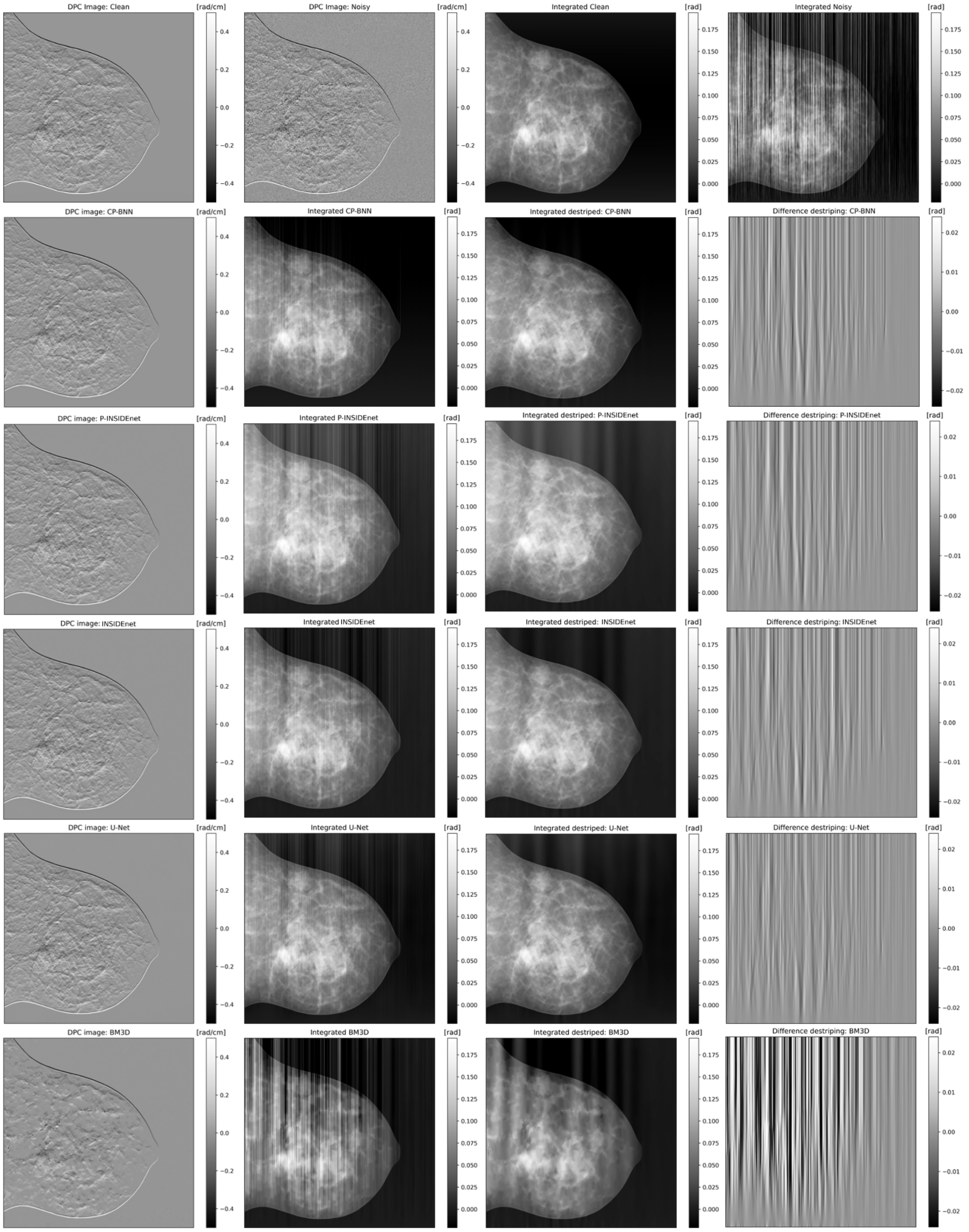

Figure 27: Depiction of the destriping performance of the WFF over the worst performing image of the supervised analysis. First row depicts the original DPC and integrated image pairs. The next rows depicts the denoised DPC image from the individual models along with the integrated image, the destriped image, and their difference.

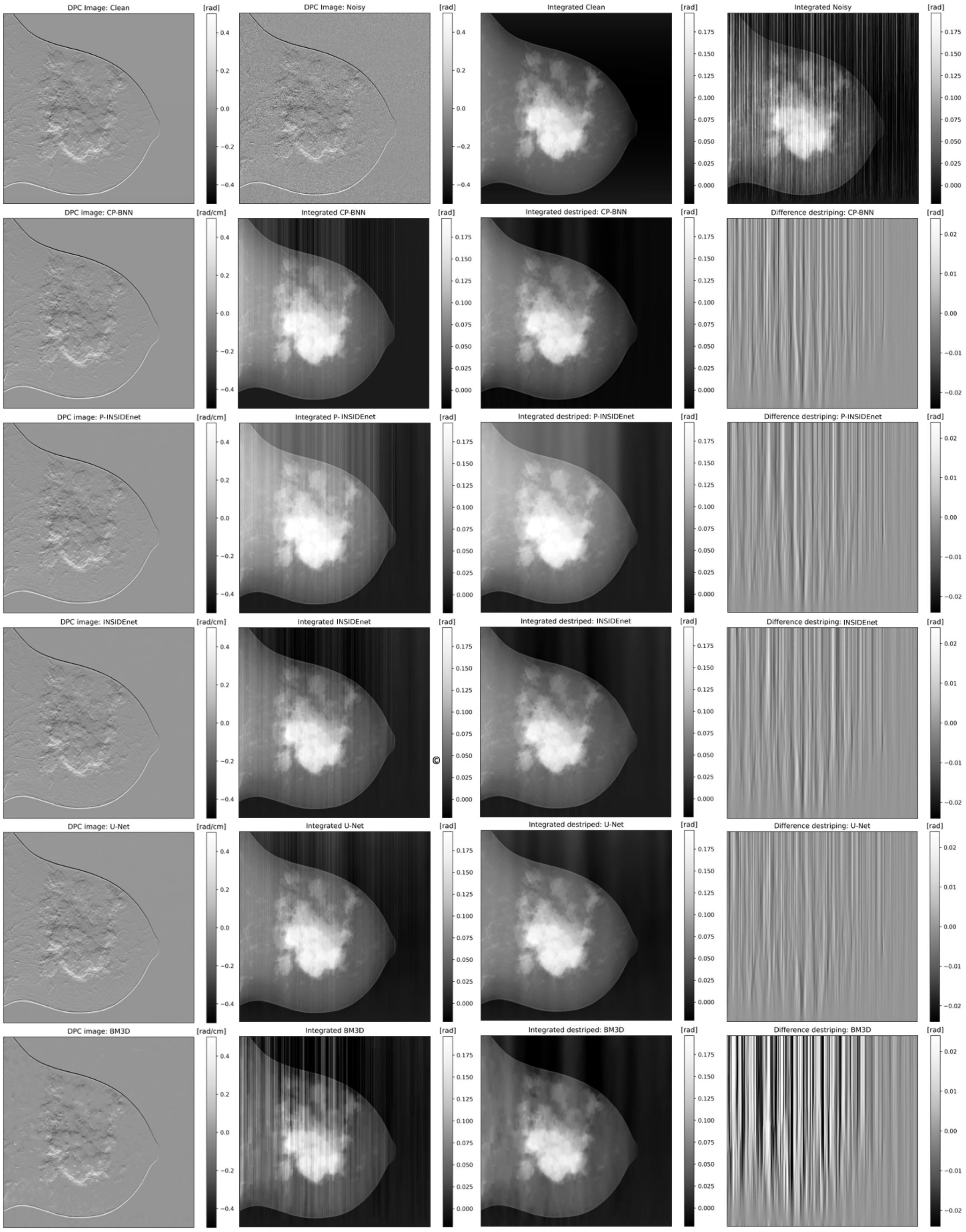

Figure 28: Depiction of the destriping performance of the WFF over the best performing image of the supervised analysis. First row depicts the original DPC and integrated image pairs. The next rows depicts the denoised DPC image from the individual models along with the integrated image, the destriped image, and their difference.

A.5 Filters

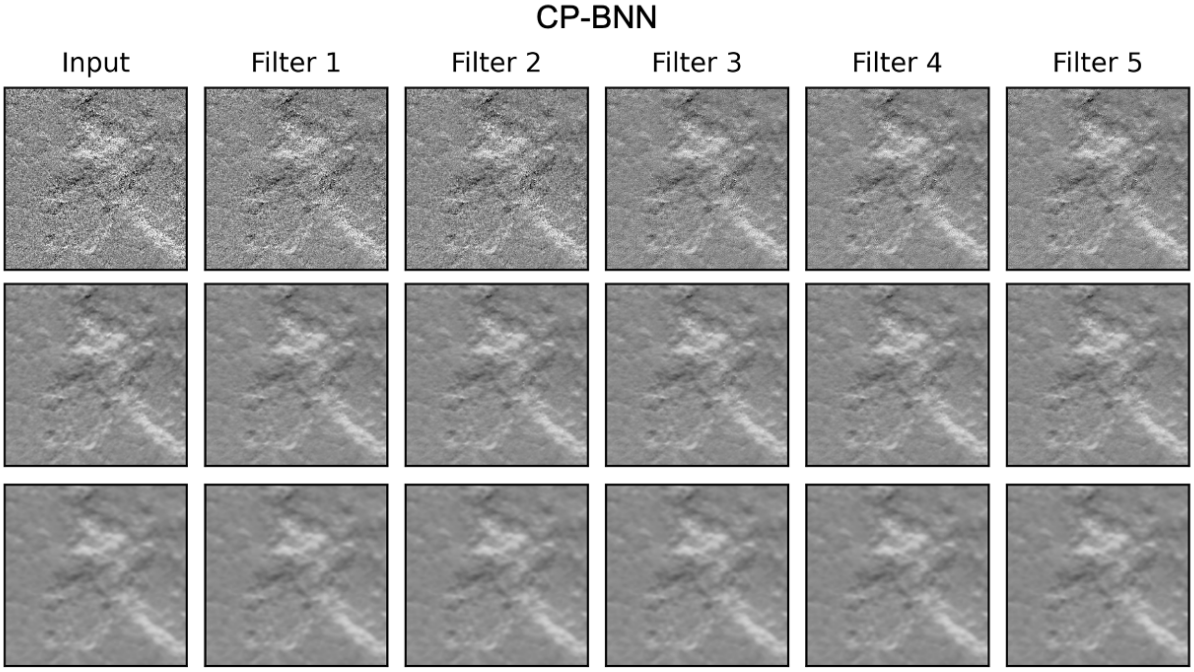

Figure 29: Denoising capabilities of the CP-BNN visualized after each filter layer at every scale. The rows indicate the scale, starting from the highest scale – i.e. full resolution – to the lowest scale

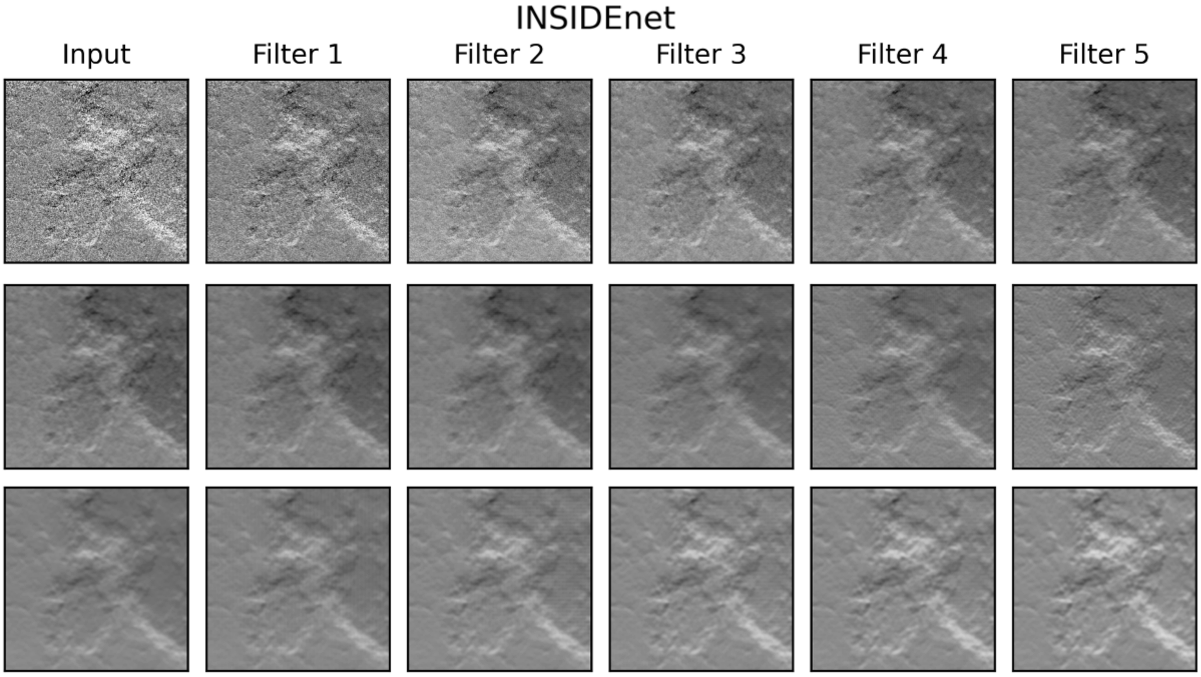

Figure 30: Denoising capabilities of the INSIDEnet visualized after each filter layer at every scale. The rows indicate the scale, starting from the highest scale – i.e. full resolution – to the lowest scale

A.6 DPC models evaluation

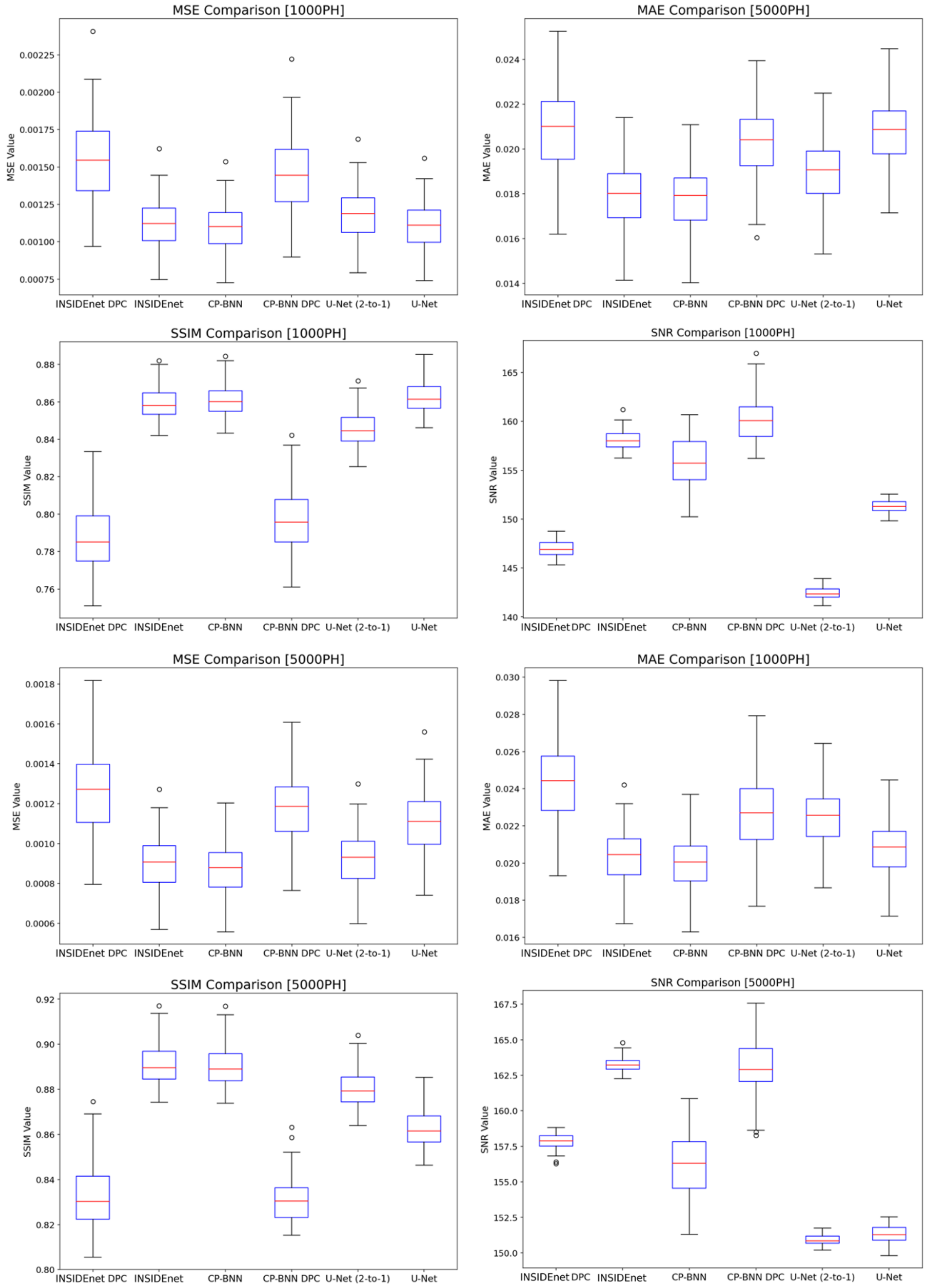

Figure 31: MSE, MAE, SSIM and SNR over the whole test set evaluated on all six models trained in a supervised fashion. The SNR was calculated over the whole background, where a mask was obtained using the transmission image. Models trained only on DPC images are marked with "DPC" after the name. On the top: results using an average photon count of 1000. On the bottom: results with an average photon count of 5000.