Unsupervised deep learning for removing structured noise in microscopy and flow cytometry

by

Benjamin Raymond Salmon

A thesis submitted to the University of Birmingham for the degree of

DOCTOR OF PHILOSOPHY

School of Computer Science
College of Engineering and Physical Sciences
University of Birmingham
July 2025

Abstract

Microscopy and flow cytometry are pillars of life science research. They use agents such as light or electrons to analyse the structure and composition of biological samples at micrometre and nanometre scales. However, the samples can be damaged by this very process, resulting in observations that do not capture them in their most natural state. Illumination intensity must therefore be minimised, but this reduces the ratio of signal to noise. When signal intensity is limited by practical constraints, and noise cannot be mitigated at the time of observation, denoisers must be employed to estimate the signal underlying a noisy observation post hoc. Currently, the most accurate estimates are made by deep learning-based denoisers. Their strength comes from utilising the information contained in training data, but this is also one of their greatest limitations.

To reliably remove all forms of noise, existing deep learning-based denoisers require paired training data, typically consisting of noisy observations and corresponding clean signals. In the life sciences, paired data and clean signals may be unobtainable. Techniques exist to train denoisers with unpaired noisy observations, *i.e.*, the very data that is to be denoised, but these face another challenge: structured noise. Structured noise is prevalent in both microscopy and flow cytometry, and it is defined as noise that is correlated over pixels or time points. Currently, no deep learning-based denoiser can reliably remove it without either paired training data or by making sacrifices in the quality of the output.

In this thesis, we develop unsupervised deep learning-based denoisers for structured noise as it commonly occurs in microscopy and flow cytometry. These methods are trained without paired observations or clean signals, and have quality approaching and sometimes exceeding that of denoisers trained with paired data. We also identify another limitation of unsupervised deep learning-based denoisers – their slow inference time – and present a method to reduce their inference time by three orders of magnitude. We believe that the methods presented here will remove some of the biggest barriers to applying deep learning denoisers to life science data.

Acknowledgements

Firstly, I would like to express my sincerest gratitude to my supervisor, Dr. Alexander Krull. Alex's enthusiasm and uncompromising support has been essential to my development as a researcher. From brainstorming projects down to debugging code, working with Alex has been both enlightening and a pleasure. I am extremely fortunate to have completed a PhD under his supervision, and I am sure this is only the beginning of our work together.

I would also like to thank my thesis group, Prof. Iain Styles, Prof. Aleš Leonardis and Dr. Jinming Duan, for their invaluable feedback and advice. Our discussions have been instrumental in shaping the research presented here. Furthermore, I am grateful to Sadao Ota and Yuichiro Iwamoto from the University of Tokyo for a productive and enjoyable collaboration. Their hard work and expertise have not only enriched this thesis, but have been a true source of inspiration.

Finally, I would like to thank my parents. My mum, for her endless patience and unwavering support. And my dad, for being a steady source of optimism, confidence and karting.

CONTENTS

ACRONYMS

AP-BSN Asymmetric PD BSN.

AR autoregressive.

AWG Additive White Gaussian.

CARE Content Aware Image Restoration.

CNN Convolutional Neural Network.

COSDD Correlated and Signal-Dependent Denoising.

CT Computed Tomography.

DL Deep Learning.

DNM Deep Nanometry.

ELBO Evidence Lower Bound.

EMCCD Electron-Multiplying Charge-Coupled Device.

EV extracellular vesicle.

GAN Generative Adversarial Network.

GMM Gaussian Mixture Model.

HDN Hierarchical DivNoising.

IR infrared.

KL Kullback-Leibler.

LSCM Laser Scanning Confocal Microscopy.

MAE mean absolute error.

mAP mean Average Precision.

MMAE minimum mean absolute error.

MMSE minimum mean square error.

MSE mean square error.

N2N Noise2Noise.

N2V Noise2Void.

N2V2 Noise2Void2.

PA photoacoustic.

PBS phosphate-buffered saline.

PMT Photomultiplier Tube.

PR Precision-Recall.

PSNR Peak Signal-to-Noise Ratio.

RI-PSNR Range-Invariant Peak Signal-to-Noise Ratio.

sCMOS scientific Complementary Metal–Oxide–Semiconductor.

SN2V Structured Noise2Void.

SNR Signal-to-Noise Ratio.

STEM Scanning Transmission Electron Microscopy.

VAE Variational Autoencoder.

1 INTRODUCTION

1.1 Motivation

Microscopy confirmed the existence of cells and microbes [7]. Flow cytometry later made it possible to quickly and quantitatively phenotype them in heterogeneous mixtures [8, 9]. Today, both techniques remain indispensable for life scientists. Indeed, to study the dynamics of living cellular systems, researchers arguably have no choice but to use light microscopy [10], and no other technique can analyse or sort a population of cells with the throughput of flow cytometry [11]. However, despite transforming the life sciences, these tools are prevented from reaching their full potential by a practical constraint: the need to protect the sample.

In light microscopy and flow cytometry, high light intensity can harm living samples or bleach fluorescent dyes [12, 13, 14]. In electron microscopy, high electron beam intensity can completely destroy a sample [15]. Low signal intensity is therefore essential for preserving sample health, but it comes at the cost of increasing the relative power of noise.

Noise consists of all the random errors that are made as a device measures a signal. For light and electron signals, the first source of such errors is shot noise, which results from the inherent randomness in the arrival of photons or electrons at a detector [16]. Another source is readout noise, which arises from the random movement of electrons during signal amplification [17]. Because it is possible for noise to degrade different signals into the same observation, working backwards to recover an exact original signal is impossible. Nonetheless, we can make an estimate that we believe to be close to the original signal. Methods for doing so are called denoisers.

Traditional, non-deep learning-based denoisers separate noise from signal by utilising prior knowledge of the characteristics that distinguish the two [18, 19, 20]. These methods are highly refined and widely used [21]. However, in terms of output quality, deep learning-based approaches predominate [22]. These methods leverage the information contained in training data to teach a deep neural network to transform noisy inputs into clean signals [23, 24, 5, 25, 26, 27]. This avoids the need for handcrafted priors and produces impressive results, but such power comes at a cost. For example, deep learning requires a significant quantity of training data and substantial computing resources, both of which can be limited in scientific applications. In this thesis, we identify some of the key challenges faced by deep learning-based denoisers in the life sciences and develop methods to overcome them. Our aim is to minimise the disruption imposed by deep learning-based denoisers on a practitioner's workflow, without compromising on restoration quality. Moreover, thanks to technical similarities between microscopes, flow cytometers and the observation technologies used in other fields, these denoisers could have broad impact.

1.2 Challenges

The following are four challenges that are faced by deep learning-based denoisers when they are applied to life science data.

1. Limited access to training data

Deep learning-based denoisers require training data. The type of training data they require varies with their learning method. Supervised learning denoisers are trained using a collection of input-target pairs. The input is always a noisy observation and the target is typically the corresponding noise-free signal [5]. The quality of results from supervised denoisers trained with noisy-clean pairs is consistently high [28]. However, with enough training data, the same quality can be achieved using noisy-noisy pairs. This is data where both the input and the target contain the same underlying signal but different and independent noise realisations [6].

For either case, the signals and noise in training pairs must be similar to that of the data that the denoiser will ultimately be applied to [29]. For noisy-clean pairs, it can be impossible to collect suitable clean examples if the sample is too delicate [30]. This is no problem for noisy-noisy pairs, but they too can be difficult to collect if the sample moves between acquisitions or if it is so delicate that it is destroyed by one [30]. And for historic data that was not collected with the intention of training a denoiser, any form of external training data can be impossible to obtain.

Another learning method is self-supervised learning [25, 31]. This only requires unpaired noisy observations. Such a requirement is ideal because the training data can be the very data that the user wants denoised, assuming that it is of a sufficient quantity. However, a price is paid for this convenience. In self-supervised denoisers, the denoised value of an element, such as a pixel or time point, is a function of all elements in the original observation except for that element itself. The ignored element is known as a blind spot, and excluding it from predictions fundamentally limits how accurately a self-supervised denoiser can a recover signal.

The most recent learning method for deep learning-based denoisers is unsupervised learning [26, 4]. This does not require paired data or blind spots. Instead, users must provide a noise model – an approximation of the statistics of the noise. The noise model can be fitted to specially collected calibration data [32] or, by assuming the noise is unstructured, bootstrapped from noisy observations [26].

2. Structured and signal-dependent noise

Noise in microscopy and flow cytometry is often not as simple as some denoisers assume. One assumption that is commonly broken is the belief that the noise in each pixel of an image, or time point of a time series, is statistically independent of the noise in other pixels or time points. In reality, noise is often what we refer to as structured, meaning that it exhibits correlations across space or time [2]. For example, in scanning-based microscopy techniques such as confocal microscopy [33, 34], noise can be correlated in the direction of the scan [35].

If paired training data is available, structured noise can be removed effectively by a supervised denoiser. However, to apply a self-supervised denoiser, its blind spot must be extended from just excluding the element that is being denoised to excluding every element that contains correlated noise [2]. Ignoring so much information sets an even lower limit on performance.

To use an unsupervised denoiser, a model of the structured noise is required, but all current applications of unsupervised denoisers use noise models that can only represent unstructured noise. A workaround was proposed by Prakash *et al*. [4], but we show that this is not successful for all cases of structured noise in Sec. 6.5.

Another characteristic of noise that is important to consider is signal dependence. This is where the variance of the noise is a function of the intensity of the underlying signal, which is the case for shot noise [36]. Although all current deep learning approaches to denoising can handle signal-dependent noise, we must ensure that new denoisers retain this ability if they are to be applied to data where shot noise is a limit on sensitivity.

3. Uncertainty

Denoising is an ill-posed problem with no unique solution. But while there are infinitely many signals that could underlie a given noisy observation, we will believe some signals to be more probable than others. The uncertainty in our beliefs is a combination of epistemic and aleatoric uncertainty. Epistemic uncertainty can be reduced by learning more about the nature of the noise and the signals. Aleatoric uncertainty is the uncertainty that remains after we have learnt all the information we can [37]. In scientific applications, it is important to have a level of confidence in data's support for conclusions. Denoisers should therefore return an estimate of uncertainty for a denoised signal.

The supervised denoiser Content Aware Image Restoration (CARE) [5] estimates epistemic uncertainty using a method called deep ensembles [38], and it estimates aleatoric uncertainty by predicting a per-pixel variance for denoised images. The unsupervised denoisers DivNoising [26] and Hierarchical DivNoising (HDN) [4] estimate aleatoric uncertainty by allowing users to randomly sample from an estimated probability distribution of clean signals. Randomly sampled signals can be compared for semantic differences or used to compute a per-pixel variance.

4. Limited access to computing resources

While the ability to sample from the distribution of clean signals makes DivNoising and HDN useful for uncertainty estimation, it makes them much slower than supervised and self-supervised denoisers during inference. This is because users are often interested in seeing one consensus solution that summarises all possible signals. Typically, this will be the expected value of the solutions, equivalent to averaging infinite samples from the distribution of possible signals. Supervised and self-supervised denoisers estimate this value directly, but unsupervised denoisers require users to estimate it by sampling and averaging solutions manually, usually with 100 or 1,000 samples [26, 4]. Scientific image datasets can consist of thousands of images and access to computing resources is often time-limited. Randomly sampling at least 100 signals per image can incur an infeasible time cost.

1.3 Thesis outline

We believe that unsupervised denoisers have the most potential to overcome the above challenges and become a standard tool for life scientists. They do not require paired training data, are not limited by blind spots, and provide meaningful aleatoric uncertainty estimates. However, the range of modalities that they can be applied to is restricted by their inability to reliably remove structured noise, and their slow inference can be expensive. The following list describes how each chapter of this thesis tackles these challenges.

Chapter 2 formalises the denoising task and all necessary concepts. It then introduces the unsupervised deep learning-based denoisers DivNoising [26] and HDN [4], which we build upon.

Chapter 3 accelerates inference of unsupervised denoisers by co-training them alongside a supervised denoiser. The inputs for this supervised network are real noisy observations, and its targets are corresponding denoised signals produced by the unsupervised denoiser. We refer to this new network as the Direct Denoiser, and it learns to predict the expected value of the denoised signal in a single forward pass, reducing inference time by three orders of magnitude. As a trade-off, training time is increased by only 26%.

Chapter 4 shows how deep autoregressive noise models can represent structured noise. However, while they can model any type of structure, their reliance on calibration data limits them to signal-independent noise in practice. We combine this noise model with an existing unsupervised denoiser and, using both simulated data and real data from photoacoustic imaging, show that we can remove structured, signal-independent noise.

Chapter 5 demonstrates that despite being limited to signal-independent noise, the method from Chapter 4 can be used to denoise light scattering observations in flow cytometry. Specifically, we enhance the diagnostic potential of a nanoparticle analyser by removing structured background noise.

Chapter 6 presents an unsupervised learning method for removing row-correlated and signal-dependent noise in microscopy images. Row correlation is a common form of structure for noise in many microscopy modalities. The denoiser is trained using only noisy images, meaning no paired observations or calibration data. This is made possible by co-training a deep autoregressive noise model alongside the main denoiser.

Chapter 7 applies the noise model co-training principle of the previous chapter to develop a denoiser for structured and signal-dependent flow cytometry noise. It is enabled by a proposed data acquisition setup where a beam splitter is used to collect paired observations with identical underlying signal but independent noise realisations. We show how this can be used to denoise the fluorescence signal in flow cytometry data without any clean data. Although it still requires noisy-noisy pairs to train, the method achieves better results than a typical supervised denoiser trained with noisy-noisy pairs.

2 BACKGROUND

2.1 Signals, noise and denoising

A signal is a physical quantity that varies with one or more independent variables and carries information [39]. The signals we are interested in are those in microscopy and flow cytometry, two of the most popular approaches for studying life at the micrometre and nanometre scale. In these modalities, the physical quantity is typically the light intensity at a photodetector, but electron beam intensity is also commonly used [40]. Other quantities used in microscopy and flow cytometry include sound and ion intensity [41, 42]. The independent variables are usually space or time, and the information these signals carry describes the physical structure of biological samples.

Detectors sample discretely. Therefore, for images, a signal, $\mathbf{s} \in \mathbb{R}^{N \times M}$, is an array with N rows and M columns of pixels. The signal in the pixel in the ith row and the jth column is denoted as $s_{i,j} \in \mathbb{R}$. For time series, a signal, $\mathbf{s} \in \mathbb{R}^{T}$, is an array with T time points, where the signal at time point t is $s_t \in \mathbb{R}$.

If we used a microscope or flow cytometer to repeatedly observe a fixed signal, $\mathbf{s}$, each observation, $\mathbf{x} \in \mathbb{R}^{N \times M}$ or $\mathbf{x} \in \mathbb{R}^{T}$, would be different. The differences would be random, so we can think of observations as samples from a probability distribution,

$$\mathbf{x} \sim p(\mathbf{x}|\mathbf{s}). \tag{2.1}$$

We refer to this as the observation likelihood [43]. It describes the random fluctuations due to shot noise, thermal motion of electrons, or any other random errors that may occur.

The noise-free signal is defined to be the average of infinite observations, or equivalently, the expected value of the observation given the signal [44],

$$\mathbf{s} = \mathbb{E}_{p(\mathbf{x}|\mathbf{s})}[\mathbf{x}]. \tag{2.2}$$

Under this definition, the noise-free signal includes deterministic errors such as the detector's point-spread function, pixel response non-uniformities or the dark current rate [45, 36]. For brevity, we will often refer to this simply as the signal.

Noise, $\mathbf{n}$, is defined to be the difference between an observation and its signal,

$$\mathbf{n} = \mathbf{x} - \mathbf{s}. \tag{2.3}$$

As a consequence of Eq. (2.2), the noise has an expected value of zero,

$$\mathbb{E}_{p(\mathbf{n}|\mathbf{s})}[\mathbf{n}] = 0. \tag{2.4}$$

Frequently, we find ourselves in a situation where we have an observation, $\mathbf{x}$, that we know to be corrupted by noise. We would therefore like to know what the signal, $\mathbf{s}$, that underlies our observation is. Naively, we could find the signal estimate that maximises the observation likelihood in Eq. (2.1). For a symmetric unimodal distribution, such as a Gaussian, we discover that the signal with the highest likelihood is perfectly equal to the observation. We are almost certain that this solution is not correct because we know that the signal is unlikely to have features that resemble noise.

Knowledge about what the signal is likely or unlikely to be, which we held before looking at the observation, comes from our prior. A prior is often defined by a set of beliefs such as smoothness [46] or self-similarity [18], or by a probability distribution, $p(\mathbf{s})$ [47, 26]. For images and time series of natural phenomena, the prior distribution will have complex correlations between pixels and time points. Nonetheless, a prior allows us to make more confident estimates of the signal underlying our observation. In fact, the probability distribution of signals for a given observation is proportional to the observation likelihood weighted by a prior,

$$p(\mathbf{s}|\mathbf{x}) \propto p(\mathbf{x}|\mathbf{s})p(\mathbf{s}). \tag{2.5}$$

The process of estimating a signal or signals that we believe to be close to the original is denoising. Supervised and self-supervised deep learning-based denoisers typically estimate the signal with the lowest expected mean squared error under Eq. (2.5) [5, 6, 25], learning an implicit prior from the data. Unsupervised denoisers, on the other hand, randomly sample from an approximation of Eq. (2.5) [26, 4].

The accuracy of a signal estimate has two sources of uncertainty. The first is epistemic, which is uncertainty in how accurately we have modelled Eq. (2.5). The second is aleatoric uncertainty, which stems from how diffusely the probability is distributed in our model of Eq. (2.5) [37].

Estimation of epistemic uncertainty for deep learning models is challenging and an area of active research [48]. While various approaches have been developed, including deep ensembles [38] as applied by the supervised denoiser CARE [5], we do not attempt to estimate it in this work. However, by basing our methods on unsupervised deep learning, we inherit the aleatoric uncertainty estimation method of DivNoising [26] and HDN [4]. The random samples that these methods provide can be examined for semantic differences, allowing users to interpret aleatoric uncertainty for themselves.

2.2 Noise models

Unsupervised denoisers avoid the need for paired training data or blind spots by utilising a noise model. A noise model is an estimation of the observation likelihood in Eq. (2.1), representing our understanding of the statistics of the noise produced by an observation device. While noise models enable an unsupervised denoiser to produce results of the same quality as a supervised denoiser [4], they restrict the denoiser to removing noise that an explicit model can be collected for. In this section, we will describe the characteristics that a noise model should be able to represent to enable the denoising of a wide range of microscopy and flow cytometry data.

2.2.1 Signal-dependent noise

Arguably, the most simple noise model is the additive white Gaussian noise model. This assumes that the noise in each pixel or time point of an observation is a random sample from a Gaussian distribution with a fixed variance σ^2. For images, we can write an additive white Gaussian noise model as

$$p(\mathbf{x}|\mathbf{s}) = \prod_i^N \prod_j^M p(x_{i,j}|s_{i,j}), \quad \text{where } p(x_{i,j}|s_{i,j}) = \mathcal{N}(x_{i,j}; s_{i,j}, \sigma^2). \tag{2.6}$$

For time series, the distribution factorises similarly over time points. This is an example of a signal-independent noise model because the statistics of the noise, $\mathbf{n} = \mathbf{x} - \mathbf{s}$, are independent of the signal,

$$p(\mathbf{n}|\mathbf{s}) = p(\mathbf{n}) = \prod_i^N \prod_j^M p(n_{i,j}), \quad \text{where } p(n_{i,j}) = \mathcal{N}(n_{i,j}; 0, \sigma^2). \tag{2.7}$$

While this model may be accurate for some of the electronic noise sources within a detector, it ignores a noise source that affects all signals carried by light or electrons: shot noise.

Light from a sample travels to a detector in discrete packets called photons, and photon emissions are independent random events. The total number of photons, $\mathbf{x}_{\text{photon}}$, emitted towards a detector during an exposure is therefore a random variable with a Poisson distribution [16],

$$\mathbf{x}_{\text{photon}} \sim \mathcal{P}(\mathbf{x}_{\text{photon}}; \lambda), \tag{2.8}$$

where the rate of emissions towards the detector, λ, is a proportional to the brightness of the light source. The difference between the rate and the number of emitted photons is shot noise,

$$\mathbf{n}_{\text{shot}} = \mathbf{x}_{\text{photon}} - \lambda. \tag{2.9}$$

A Poisson distribution has a variance equal to its rate, so the variance of shot noise must also be proportional to the brightness of the light source,

$$\mathrm{Var}[\mathbf{n}_{\mathrm{shot}}] = \mathrm{Var}[\mathbf{x}_{\mathrm{photon}} - \lambda] = \mathrm{Var}[\mathbf{x}_{\mathrm{photon}}] = \lambda. \tag{2.10}$$

In a perfect photodetector, shot noise is the only noise source. It occurs independently over space and time, so we can write a model of shot noise for images as,

$$p(\mathbf{x}|\mathbf{s}) = \prod_{i}^{N}\prod_{j}^{M} p(x_{i,j}|s_{i,j}), \quad \text{where } p(x_{i,j}|s_{i,j}) = \mathcal{P}(x_{i,j}; s_{i,j}). \tag{2.11}$$

The shot noise model for time series factorises equivalently over time points.

In reality, the number of photons actually detected is affected by imperfect quantum efficiency and the presence of dark current [36], and additional noise sources will compound with shot noise. Nonetheless, the variance of the final observation will still increase with its expected value, *i.e.*, its underlying signal. We refer to noise that is statistically dependent on the underlying signal as signal-dependent. Formally, noise, $\mathbf{n}$, is signal-dependent when the following is true,

$$p(\mathbf{n}|\mathbf{s}) \neq p(\mathbf{n}). \tag{2.12}$$

A parametric model of the sum of additive white Gaussian and Poisson noise was proposed by Foi *et al*. [49]. This models imaging noise by assuming a fixed variance for the electronic noise component and a linear function of the underlying signal as the variance for the shot noise component. While this is an accurate representation of combined electronic and shot noise, a more general model of signal-dependent noise was used to enable denoising by Krull *et al*. [43]. They collected paired noisy-clean images, binned the signal intensities and plotted a histogram of observed noisy values for each signal intensity. By normalising the histograms, they produced a separate probability distribution of noise for every signal intensity. Another flexible approach was applied by Prakash *et al*. [50]. They mapped signal intensities to the parameters of a Gaussian mixture model using a polynomial function. Using a dataset of paired noisy-clean images, they fit the parameters of the polynomial function with iterative maximum likelihood optimisation.

Although the signal-dependent noise models described here are fit to paired noisy-clean data, they are still relatively straightforward to implement. This is because their data only needs to represent the range of signal intensities that will be encountered. The patterns and structures in the data can be arbitrary as they are ignored by the noise models. Suitable data can therefore be obtained by, for example, taking around 100 noisy images of an out-of-focus light source and averaging them to estimate their underlying signal [32].

2.2.2 Structured noise

The noise models that we saw in the previous section (Sec. 2.2.1) were pixel-independent, *i.e.*, the probability distribution of noise in one pixel is unaffected by the noise in any other pixel. However, noise generated by the electronics of a detector often exhibits spatial or temporal correlation. For instance, a confocal microscope produces an image one pixel after the other by point-scanning across a sample in a raster scan fashion. Often, the noise in each pixel is correlated with the noise in previously produced pixels [35]. For such images, the observation likelihood in Eq. (2.1) can be factorised as

$$p(\mathbf{x}|\mathbf{s}) = \prod_{i}^{N}\prod_{j}^{M} p(x_{i,j}|\mathbf{s}, x_{i,<j}), \tag{2.13}$$

where $x_{i,<j}$ are all the pixels in row i and columns preceding column j. A more general factorisation that would capture any structure is

$$p(\mathbf{x}|\mathbf{s}) = \prod_{i^*}^{N\times M} p(x_{i^*}|\mathbf{s}, x_{<i^*}), \tag{2.14}$$

where i^* indexes pixels in a raster scan order [51].

For time series, a general representation of structured noise is given by

$$p(\mathbf{x}|\mathbf{s}) = \prod_{t}^{T} p(x_t|\mathbf{s}, x_{<t}). \tag{2.15}$$

More examples of structured noise can be found in Chapters 4, 5 and 6. Currently no models of structured noise have been applied to unsupervised denoisers, prohibiting their application to these modalities. Developing suitable models of structured and signal-dependent noise is a key theme of this thesis.

2.3 Unsupervised deep learning-based denoisers

The goal of this section is to explain the unsupervised deep learning-based denoiser DivNoising [26] and its follow-up HDN [4], as they are the starting point of the methods developed in this thesis. But before doing so, we must first introduce their backbone, the Variational Autoencoder (VAE) [52].

2.3.1 Variational Autoencoders

The following is based on the tutorial in [53].

The VAE framework [52] trains a latent variable model, $p_\theta(\mathbf{x}, \mathbf{z})$, with observed variables $\mathbf{x}$, latent variables $\mathbf{z}$ and parameters θ. The latent variable model can take the form

$$p_\theta(\mathbf{x}, \mathbf{z}) = p_\theta(\mathbf{x}|\mathbf{z}) p_\theta(\mathbf{z}), \tag{2.16}$$

or, to make it more expressive, it can use a hierarchy of latent variables $\mathbf{z} = (\mathbf{z}_1, \ldots, \mathbf{z}_L)$ and take the form

$$p_\theta(\mathbf{x}, \mathbf{z}) = p_\theta(\mathbf{x}|\mathbf{z}_1) \left(\prod_{l=1}^{L-1} p_\theta(\mathbf{z}_l|\mathbf{z}_{l+1}) \right) p_\theta(\mathbf{z}_L). \tag{2.17}$$

Each p_θ is parametrised with a deep neural network with parameters θ.

To optimise the model, we have access to random samples of $\mathbf{x}$ but not $\mathbf{z}$. These could be used to maximise the marginal log-likelihood objective,

$$\log p_\theta(\mathbf{x}) = \log \frac{p_\theta(\mathbf{x}, \mathbf{z})}{p_\theta(\mathbf{z}|\mathbf{x})}, \tag{2.18}$$

if it were possible to cancel out the latent variables with the posterior, $p_\theta(\mathbf{z}|\mathbf{x})$. Unfortunately, the posterior is often intractable.

The VAE overcomes this problem by introducing an approximate posterior, $q_\phi(\mathbf{z}|\mathbf{x})$, which is parametrised by a deep neural network with parameters ϕ. When using the hierarchical latent variable model in Eq. (2.17), the approximate posterior factorises as

$$q_\phi(\mathbf{z}|\mathbf{x}) = \left(\prod_{l=1}^{L-1} q_\phi(\mathbf{z}_l|\mathbf{x}, \mathbf{z}_{l+1})\right) q_\phi(\mathbf{z}_L|\mathbf{x}). \tag{2.19}$$

Expressing the marginal log-likelihood in terms of the approximate posterior, we have

$$\log p_\theta(\mathbf{x}) = \underbrace{\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log \frac{p_\theta(\mathbf{x}, \mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})}\right]}_{\text{ELBO}} + \mathrm{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z}|\mathbf{x})), \tag{2.20}$$

where KL is the Kullback-Leibler (KL) divergence [54]. Of course, this form of the objective still contains the true posterior, so it cannot be computed. However, notice that KL divergence is always non-negative. Therefore, the first term in Eq. (2.20) will always be less than or equal to the true marginal log-likelihood. For that reason, it is known as the Evidence Lower Bound (ELBO).

By rearranging Eq. (2.20),

$$\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log \frac{p_\theta(\mathbf{x}, \mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})}\right] = \log p_\theta(\mathbf{x}) - \mathrm{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z}|\mathbf{x})), \tag{2.21}$$

we see that maximising the ELBO is equivalent to simultaneously maximising the marginal log-likelihood and minimising the KL divergence from the true to the approximate posterior. This will give us a latent variable model of the observed data and an approximate way to randomly sample latent variables that could underlie an observation.

The ELBO can be optimised with gradient descent, which can be seen by expressing it in terms of the distributions parametrised with neural networks,

$$\mathcal{L}(\theta, \phi; \mathbf{x}) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[-\log \frac{p_\theta(\mathbf{x}, \mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})}\right] \tag{2.22}$$

$$= \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[-\log p_\theta(\mathbf{x}|\mathbf{z})] + \mathrm{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z})). \tag{2.23}$$

The expectation in the first term of Eq. (2.23) can be approximated using the reparametrisation trick [52] and, when the latent variable distributions are Gaussians, the second term can be computed analytically. For the hierarchical latent variable model in Eq. (2.17), the second term becomes

$$\mathrm{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z})) = \sum_{l=1}^{L-1} \mathrm{KL}(q_\phi(\mathbf{z}_l|\mathbf{x}, \mathbf{z}_{l+1})||p_\theta(\mathbf{z}_l|\mathbf{z}_{l+1})) + \mathrm{KL}(q_\phi(\mathbf{z}_L|\mathbf{x})||p_\theta(\mathbf{z}_L)), \tag{2.24}$$

which can also be computed analytically for Gaussian distributions [55].

Equation (2.23) reveals that VAEs are a form of regularised autoencoder: An observation, $\mathbf{x}$, is first encoded as a random sample, $\mathbf{z}$, from the approximate posterior, $q_\phi(\mathbf{z}|\mathbf{x})$. The approximate posterior is regularised by rewarding it for having a low KL divergence from the prior, $p_\theta(\mathbf{z})$. The latent variable is then decoded as a random sample from $p_\theta(\mathbf{x}|\mathbf{z})$. For this reason, $q_\phi(\mathbf{z}|\mathbf{x})$ is referred to as the encoder and $p_\theta(\mathbf{x}|\mathbf{z})$ is referred to as the decoder.

2.3.2 DivNoising and HDN

The unsupervised denoisers DivNoising [26] and HDN [4] combine the VAE framework with a pre-trained noise model. Current implementations of unsupervised denoisers pre-train noise models using the methods described in Sec. 2.2.1. This means that they represent signal-dependent but unstructured noise. DivNoising and HDN have only been applied to images, so their noise models are denoted as

$$p_\eta(\mathbf{x}|\mathbf{s}) = \prod_{i=1}^{N}\prod_{j=1}^{M} p_\eta(x_{i,j}|s_{i,j}), \tag{2.25}$$

with parameters η.

The noise model, $p_\eta(\mathbf{x}|\mathbf{s})$, is used as part of the following latent variable model,

$$p_\theta(\mathbf{x}, \mathbf{z}) = p_\eta(\mathbf{x}|\mathbf{s} = f_\theta(\mathbf{z}))p_\theta(\mathbf{z}), \tag{2.26}$$

where $\mathbf{x}$ and $\mathbf{z}$ are conditionally independent given $\mathbf{s}$. Here, f_θ is a deterministic deep neural network with parameters θ.

Both DivNoising and HDN use the VAE framework to optimise this model,

$$\mathcal{L}(\theta, \phi; \mathbf{x}) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[-\log p_\eta(\mathbf{x}|\mathbf{s} = f_\theta(\mathbf{z}))] + \mathrm{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z})) \tag{2.27}$$

$$= -\log p_\theta(\mathbf{x}) + \mathrm{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z}|\mathbf{x})). \tag{2.28}$$

When the variables $\mathbf{x}$ are noisy images, minimising this loss trains a latent variable model of those noisy images. The trained model can be used to sample entirely new noisy images by first sampling a latent variable from the prior, $p_\theta(\mathbf{z})$, then sampling a noisy image from the decoder, $p_\eta(\mathbf{x}|f_\theta(\mathbf{z}))$. The decoder is fixed as the noise model, so the only variability it can model is variability due to noise. Variability in the underlying signal must therefore be modelled by the prior. Thus, to generate realistic noisy images, the latent variables must represent clean signals, and $f_\theta(\mathbf{z}) = \mathrm{s}$ must transform them into actual clean signals.

The VAE also estimates a posterior distribution of latent variables, meaning that we can randomly sample latent variables that might underlie an observed variable. Since latent variables represent clean signals, we now have everything needed to approximately sample from the denoising distribution, $p_\theta(\mathbf{s}|\mathbf{x})$,

$$f_\theta(\mathbf{z}) \overset{\text{approx.}}{\sim} p_\theta(\mathbf{s}|\mathbf{x}), \quad \text{where } \mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x}). \tag{2.29}$$

The key difference between DivNoising and HDN is the fact that DivNoising uses the standard latent variable model in Eq. (2.16) whereas HDN uses the hierarchical latent variable model in Eq. (2.17), specifically with $L = 6$ layers. The hierarchical model makes the approximate posterior more flexible and able to more closely match the true posterior, therefore giving more realistic denoised signal; but the authors discovered an additional benefit.

When modelling images containing structured noise, but using a pixel-independent noise model, the authors found that the high-frequency structured noise content was represented by the two lowest latent variables in the hierarchy, $\mathbf{z}_1$ and $\mathbf{z}_2$. They then showed that modifying their approximate posterior to neglect the observation at levels 1 and 2,

$$\begin{aligned} q_\phi(\mathbf{z}_1|\mathbf{z}_2, \mathbf{x}) &= q_\phi(\mathbf{z}_1|\mathbf{z}_2) \\ q_\phi(\mathbf{z}_2|\mathbf{z}_3, \mathbf{x}) &= q_\phi(\mathbf{z}_2|\mathbf{z}_3), \end{aligned} \tag{2.30}$$

removed structured noise content from denoised images. This variant is called HDN$_{3-6}$. Unfortunately, as we show in Sec. 6.5, this strategy does not always work.

3 DIRECT UNSUPERVISED DENOISING

The first of the challenges in Sec. 1.2 that this thesis tackles is **Limited access to computing resources**. In this chapter, we present a method to reduce the inference time of unsupervised denoisers by predicting the expected value of a denoised signal in a single evaluation step, overcoming the current requirement of 100 or 1,000 steps.

This chapter is based on: [56] **B. Salmon** and A. Krull, "Direct unsupervised denoising," in 2023 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), pp. 3840–3847, 2023.

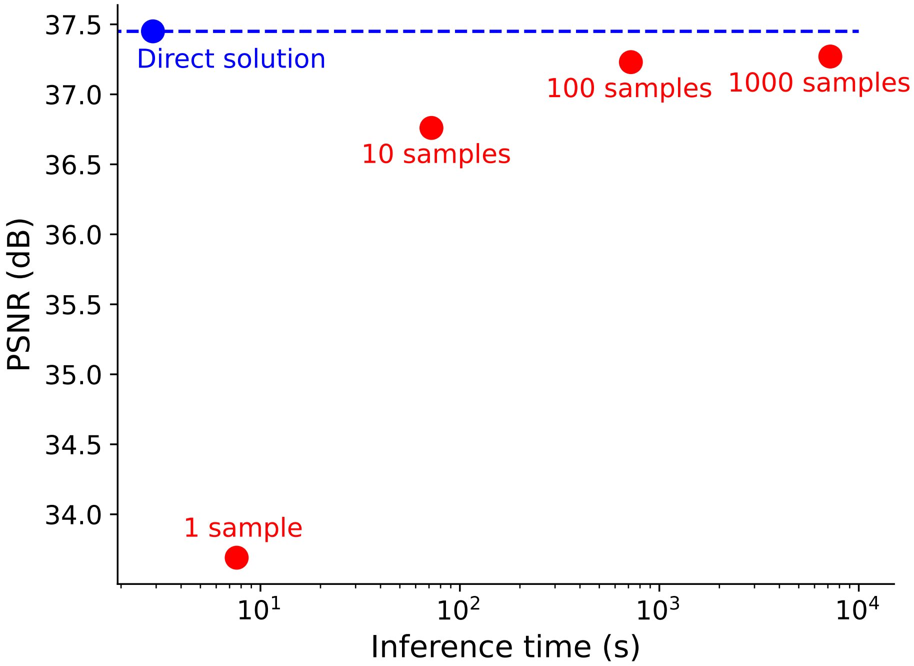

Figure 3.1: **Our Direct Denoiser outperforms unsupervised VAE-based denoising (HDN) [4] while requiring only a fraction of the computational cost.** In red, the time to draw 1, 10, 100 and 1,000 samples from HDN's learned denoising distribution plotted against the PSNR (higher is better) of the per-pixel mean of these samples. Additionally, in blue, the time to take a single solution from our Direct Denoiser is plotted against its PSNR. These results are from denoising the *Convallaria* dataset.

3.1 Introduction

While the quality of restorations from unsupervised deep learning-based denoisers is now approaching and even sometimes matching that of their supervised counterparts [26, 4], the way these two methods produce denoised images is fundamentally different. By training a VAE [52], unsupervised methods approximate a posterior distribution of the clean images that could underlie a noisy input image. This distribution will be referred to as the *denoising distribution*. Random samples from the denoising distribution then constitute the infinite possible solutions to a denoising problem. Supervised and self-supervised learning methods, on the other hand, offer a single prediction that compromises between all possible solutions. This is usually a central tendency of the denoising distribution, and the specific central tendency that is predicted depends on the loss function used. For example, a supervised method trained with the mean square error (MSE) loss function will predict the mean, which is also known as the minimum mean square error (MMSE) estimate. A model trained with the mean absolute error (MAE) loss function will predict the pixel-wise median, which is known as the minimum mean absolute error (MMAE) estimate.

While the ability of unsupervised methods to produce diverse solutions can in some circumstances be beneficial for downstream processing [26], users often require only a single solution such as the MMSE estimate. If they are to obtain this from an unsupervised learning-based denoiser, they must process their image many times and average many possible sampled solutions, leading to a significant computational overhead. For example, for a single image, the authors of [26, 4] average 100 or 1,000 samples from the denoising distribution to obtain their MMSE estimate. Such an approach requires substantial computational effort and is not likely to be economically or ecologically reasonable for labs regularly analysing terabytes of data.

This chapter presents an alternative route to estimating the central tendencies from an unsupervised denoiser, one that requires noisy images to be processed only once. We do so by training an additional deterministic Convolutional Neural Network (CNN), called the *Direct Denoiser*, that directly predicts MMSE or MMAE solutions. It is trained alongside the main VAE by using noisy training images as input and the sampled predictions from the VAE as training targets. Being deterministic, this network will minimise its MSE or MAE loss function by predicting the mean or pixel-wise median of the denoising distribution. The result is a denoising network with the evaluation times of a supervised approach and the training data requirements of an unsupervised approach.

In summary, we propose an extension to unsupervised deep learning-based denoisers that reduces inference time by estimating a central tendency of the learned denoising distribution in a single evaluation step. Moreover, we show these estimates to be more accurate than those obtained by averaging even 1,000 samples from the denoising distribution. Figure 3.1 shows how much shorter inference time is with our proposed approach, and how much higher the quality of results are.

The remainder of the chapter is structured as follows. In Sec. 3.2, we give a brief overview of related work, concentrating on different approaches to inference methods in deep learning-based denoising. In Sec. 3.3, we provide background on inference in unsupervised VAE-based denoising, which is the foundation of our method. In Sec. 3.4, we describe the training of the Direct Denoiser. We evaluate our approach in Sec. 3.5, showing that we consistently outperform our baseline at a fraction of the computational cost. Finally, in Sec. 3.6 and Sec. 3.7, we discuss our results and give an outlook on the expected impact of our work and future perspectives.

Code for the work presented here can be found at `github.com/krulllab/DirectDenoiser`.

3.2 Related work

3.2.1 Supervised denoising

Traditional supervised deep learning-based methods (*e.g.* [57, 5]) rely on paired training data consisting of corresponding noisy and clean images. These methods view denoising as a regression problem, and usually train a UNet [58] or variants of the architecture to learn a mapping from noisy to clean. The most commonly used loss function for this purpose is the sum of pixel-wise quadratic errors (MSE), which directs the network to predict the MMSE estimate for the noisy input.

The approach's requirement for clean training images greatly limits its applicability, particularly for scientific imaging applications, where often no clean data can be obtained. In 2018, Lehtinen *et al*. [6] had the insight that training of equivalent quality can be achieved by replacing the clean training image with a second noisy image of the same content; a training method termed Noise2Noise (N2N). In practice, such image pairs can often be acquired by recording two images in quick succession. By using the MSE loss and assuming that the imaging noise is zero-centred (Eq. (2.4)), the network is expected to minimise the loss to its noisy training target by converging to the same MMSE estimate as in noisy-clean training.

While Noise2Noise and traditional supervised methods are state-of-the-art with respect to the quality of their results, their requirement for paired training data makes them inapplicable in many situations. In contrast, our method requires only unpaired noisy data, which is available for any denoising task, making it directly applicable in situations where supervised methods are not.

3.2.2 Self-supervised denoising

Self-supervised methods were introduced to enable denoising with unpaired noisy data. Here we focus on *blind-spot* approaches (*e.g*. [25, 31, 59, 60]), which mask individual pixels in the input image and use them as training targets. These methods rely on the assumption that imaging noise unstructured, *i.e.*, pixel-wise independent given an underlying signal. By training the network to predict each pixel value from its surroundings, blind-spot approaches can learn to denoise images without the need for paired noisy-clean or noisy-noisy data. Like most supervised methods, self-supervised denoisers are trained with the MSE loss, so predict an MMSE estimate for each pixel. However, they do so with less information, since the corresponding input pixel cannot be used during prediction. As a result, the quality of the output can be worse than supervised methods. The blind-spot approach has been improved to reintroduce the lost pixel information during inference [32, 28], achieving improved quality in some situations. In [61], Broaddus *et al*. extended the method to allow for the removal of structured noise.

Our method also does not require paired data, but we do not follow the self-supervised blind-spot paradigm. As a consequence, we do not have to address the loss of pixel information.

3.2.3 Knowledge distillation

Knowledge distillation [62] is the process of training a smaller *student* network using a large *teacher* network or an ensemble [63] of teachers. The goal of this approach is to reduce the computational effort required during inference and enable more efficient employment of a powerful model. Surprisingly, the student model can achieve better results compared to being trained on the data directly. A survey of the topic can be found in [64].

The approach of training our Direct Denoiser with the output of another network can be seen as knowledge distillation. However, in our case the Direct Denoiser is not intended as a smaller replacement of the VAE, but as a model with a faster inference procedure.

3.3 Background

Here we will describe the inference procedure of the unsupervised deep learning-based denoisers DivNoising [26] and HDN [4]. For a formal introduction to the training of these denoisers, see Sec. 2.3.

The authors of DivNoising [26] and HDN [4] adapt the VAE for denoising by incorporating a known explicit noise model, $p_\eta(\mathbf{x}|\mathbf{s})$, into its training objective Eq. (2.27). After minimising this objective, the signal, $\mathbf{s}$, underlying a given $\mathbf{x}$ can be estimated by first encoding $\mathbf{x}$ with the learned approximate posterior, $q_\phi(\mathbf{z}|\mathbf{x})$, sampling a $\mathbf{z}$ and mapping that sample to an estimate of the signal with the learned decoder $f_\theta(\mathbf{z})$. These solutions are samples from an approximation of the true posterior distribution of signals,

$$q_{\phi,\theta}(\mathbf{s}|\mathbf{x}) = \int_{\mathbf{z}} q_\phi(\mathbf{z}|\mathbf{x})\delta(\mathbf{s} - f_\theta(\mathbf{z}))d\mathbf{z}, \tag{3.1}$$

where δ is the Dirac delta function [65]. We refer to this as the *denoising distribution*.

Each sample from the denoising distribution is unique, allowing users to examine the aleatoric uncertainty involved in their denoising problem. However, a single consensus solution is often desired. The authors of [26, 4] chose to calculate the per-pixel mean of 100 or 1,000 samples, deriving the MMSE estimate of the denoising distribution, *e.g.*,

$$\begin{gathered}\frac{1}{100}\sum_{i=1}^{100} f_\theta(\mathbf{z}_i) \approx \arg\min_{\hat{\mathbf{s}}} \mathbb{E}_{q_{\phi,\theta}(\mathbf{s}|\mathbf{x})}\left[||\mathbf{s}-\hat{\mathbf{s}}||_2^2\right], \\ \text{where } \mathbf{z}_i \sim q_\phi(\mathbf{z}|\mathbf{x}),\end{gathered} \tag{3.2}$$

where $||\cdot||_2^2$ is the squared L_2 norm. This consensus solution was then used for measuring denoising performance. Taking so many samples requires many forward passes of the denoiser and incurs a potentially prohibitive computational overhead for large datasets.

Our method extends the high quality denoising performance and minimal training requirements of VAE-based denoisers by allowing them to directly and efficiently produce MMAE and MMSE results without repeated sampling.

3.4 Method

When given samples from a probability distribution, we are often interested in what a representative value of those samples is. In the case of unsupervised denoising, we are interested in a representative signal from the denoising distribution. A common value to choose for this is the central tendency of the distribution [66], a point which minimises some measure of deviation from all of the samples.

For samples from a denoising distribution, $q_{\phi,\theta}(\mathbf{s}|\mathbf{x})$, this would be

$$\hat{\mathbf{s}}^* = \arg\min_{\hat{\mathbf{s}}} \mathbb{E}_{q_{\phi,\theta}(\mathbf{s}|\mathbf{x})}[L(\mathbf{s},\hat{\mathbf{s}})], \tag{3.3}$$

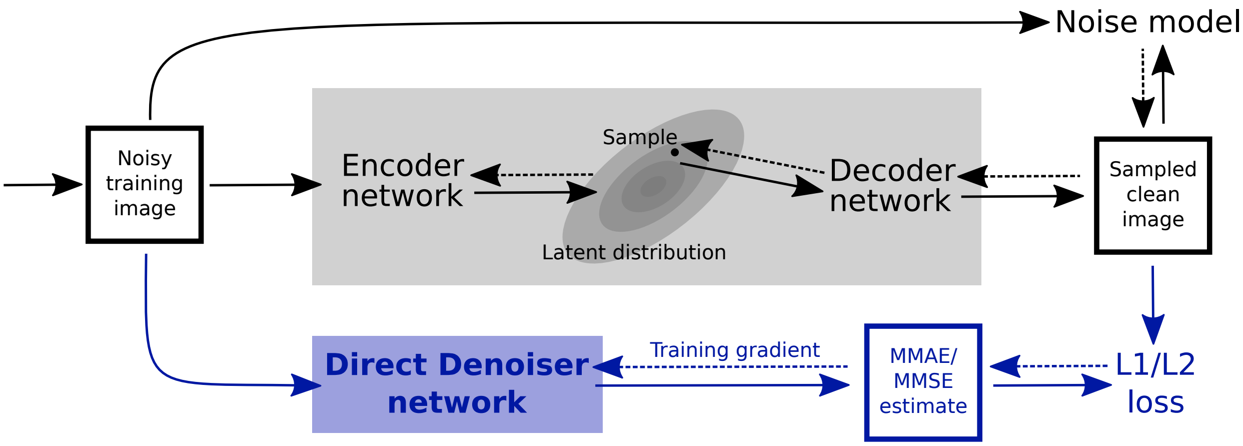

Figure 3.2: **Training scheme:** We train our novel *Direct Denoiser* (blue) alongside a Variational AutoEncoder (VAE) [26, 4]. The processing of data is shown with solid arrows and the backward propagation of gradients required for training is shown with dashed arrows. The VAE encoder takes a noisy observation as input and predicts the parameters of a distribution in latent space. A sample is drawn from here and mapped to a possible clean signal by the decoder network. The reconstruction loss is computed using a pre-trained noise model. Our Direct Denoiser is trained using noisy observations as input and the clean signal samples from the VAE as target. Since individual samples differ for the same input, there is no unique correct solution for this task. As a consequence, by using a squared L_2 loss, the Direct Denoiser will learn to predict the expected value, *i.e.*, the MMSE solution. Using an L_1 loss leads to predicting the pixel-wise median. We block gradients from passing through the sampled clean image to prevent the VAE changing its outputs.

where L is some per-pixel loss function. If L is the L_1 norm of errors,

$$L(\mathbf{s}, \hat{\mathbf{s}}) = \|\mathbf{s} - \hat{\mathbf{s}}\|_1, \tag{3.4}$$

then $\hat{\mathbf{s}}^*$ corresponds to the pixel-wise median of the distribution, *i.e.*, the MMAE estimate. For the squared L_2 norm of errors,

$$L(\mathbf{s}, \hat{\mathbf{s}}) = \|\mathbf{s} - \hat{\mathbf{s}}\|_2^2, \tag{3.5}$$

$\hat{\mathbf{s}}^*$ will be the arithmetic mean, *i.e.*, the MMSE.

The authors of [26, 4] estimated $\hat{\mathbf{s}}^*$ using a large number of samples from their denoising distribution. We propose instead training a CNN to directly predict a central tendency.

Let g_ψ be our Direct Denoiser with parameters ψ. The following objective,

$$\mathcal{L}(\psi; \mathbf{x}) = \mathbb{E}_{q_{\phi,\theta}(\mathbf{s}|\mathbf{x})}[L(\mathbf{s}, g_\psi(\mathbf{x}))], \tag{3.6}$$

where L is either the L_1 or squared L_2 norm of errors, would train g_ψ to predict either the pixel-wise median or mean of $q_{\phi,\theta}(\mathbf{s}|\mathbf{x})$, respectively.

In practice, we can approximate the expectation in Eq. (3.6) with a single sample from a denoising distribution pre-trained by DivNoising [26] or HDN [4]. Furthermore, we find that it is possible to train both the Direct Denoiser and the denoising distribution simultaneously. A single training step is as follows:

1. Pass a noisy training observation $\mathbf{x}$ to the unsupervised denoiser and sample a possible solution $\mathbf{s} \sim q_{\phi,\theta}(\mathbf{s}|\mathbf{x})$.
2. Update the parameters (θ, ϕ) towards minimizing the loss function in Eq. (2.27).
3. Pass the same $\mathbf{x}$ to the Direct Denoiser, calculating $g_\psi(\mathbf{x})$.
4. Update the parameters ψ towards minimising Eq. (3.6).
5. Repeat until convergence.

A visual representation of this training scheme can be found in Figure 3.2.

3.5 Experiments

Our Direct Denoiser was trained alongside HDN [4] using six datasets of intrinsically noisy microscopy images that come with known ground truth signal. Each dataset can be found in [4], as can details of their size, spatial resolution and train, validation and test splits. Note that for the *Struct. Convallaria* dataset, we adapted HDN into HDN$_{3-6}$, making it capable of handling structured noise.

Table 3.1: **Average PSNR of consensus solutions from HDN [4] and direct solutions from our novel *Direct Denoiser*.** HDN's consensus solutions were obtained by taking samples of varying sizes from its denoising distribution and calculating both their per-pixel median and their per-pixel mean. The Direct Denoiser's solutions were obtained from a single pass of a network trained under an L_1 loss and a single pass of a network trained under a squared L_2 loss. PSNRs in the upper table are for median consensus for HDN and the solution from the L_1 network for the Direct Denoiser. PSNRs in the lower table are for mean consensus for HDN and the solution from the squared L_2 network for the Direct Denoiser. Best results for a loss function are printed in **bold** and best results overall are underlined.

		Number of samples (HDN)				
	Dataset	**1**	**10**	**100**	**1,000**	**Direct**
L_1	Convallaria	33.69	36.59	37.17	37.19	**37.50**
	Confocal Mice	35.43	37.30	37.58	37.62	**37.77**
	2 Photon Mice	31.21	32.63	32.86	32.89	**33.55**
	Mouse Actin	31.62	33.52	33.87	33.91	**34.22**
	Mouse Nuclei	33.48	36.24	36.79	36.81	**36.87**
	Struct. Convallaria	29.02	30.88	31.22	31.27	**31.58**
Squared L_2	Convallaria	33.69	36.76	37.23	37.27	**37.45**
	Confocal Mice	35.43	37.42	37.68	37.69	**37.75**
	2 Photon Mice	31.21	32.68	32.87	32.89	**33.54**
	Mouse Actin	31.62	33.66	33.92	33.95	**34.28**
	Mouse Nuclei	33.48	36.44	36.89	36.90	**36.93**
	Struct. Convallaria	29.02	31.00	31.27	31.29	**31.64**

3.5.1 Denoising performance

To evaluate denoising performance, we compare the Peak Signal-to-Noise Ratio (PSNR) of our Direct Denoiser's direct solutions to the PSNR of HDN's consensus solutions. The consensus solutions were produced by averaging samples of size 1, 10, 100 and 1,000, reporting both their per-pixel median and mean. The Direct Denoiser's solutions were reported from a network trained with an L_1 loss and a network trained with a squared L_2 loss. Results are in Table 3.1. Visual results from the same experiment can be seen in Figure 3.3.

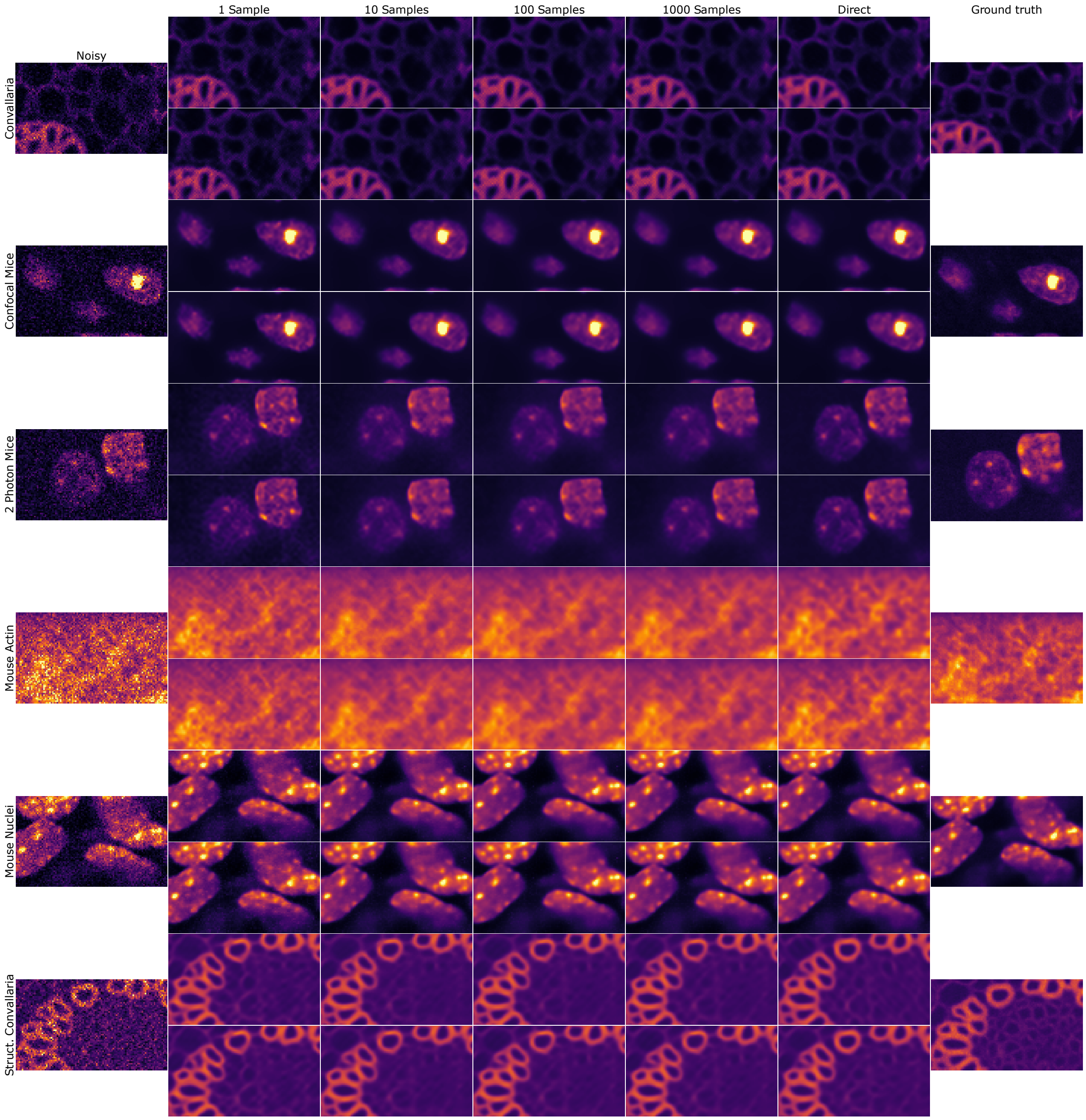

Figure 3.3: **Visual results:** Cropped images from each dataset showing consensus solutions of varying sample sizes from HDN's denoising distribution with direct solutions from our Direct Denoiser. For each dataset, the top row shows the median of HDN samples and a solution from our L_1 trained Direct Denoiser, while the bottom row shows the mean of HDN samples and a solution from our squared L_2 trained Direct Denoiser.

3.5.2 Inference times

We also compared inference time to denoising performance. In Figure 3.1, the total time for HDN to generate 1, 10, 100 and 1,000 samples for all 100 images in the *Convallaria* test set was measured, then plotted against the PSNR of the mean of those samples, averaged over all 100 images. On the same plot, the total time for our Direct Denoiser to produce single solutions for each image is plotted against their average PSNR. Each test image consisted of 512 × 512 pixels.

Using our GPU (an NVIDIA GeForce RTX 3090 Ti), generating a single 512 × 512 solution from HDN's denoising distribution takes 0.076 seconds, using 2207MB of the GPU's memory. Our Direct Denoiser takes 0.029 seconds at 1909MB to do the same. Processing one image with either model uses the full capacity of the GPU's parallelism, so we saw no speed improvements by processing more than one image at a time.

If a consensus solution from HDN with PSNR approaching that of the the Direct Denoiser requires sampling 1,000 solutions, inference with the proposed method is 2,621× faster.

3.5.3 Training times and memory usage

Finally, the additional training time incurred by co-training HDN with the Direct Denoiser was examined. The authors of HDN [4] train their network for 200,000 steps for all datasets, using a batch size of 64 and image patch size of 64 × 64. Using our GPU, training HDN alone takes 0.27 seconds per step for 15 hours total, using 13GB of GPU memory. Training both HDN and the Direct Denoiser takes 0.34 seconds per step for 18.9 hours total, using 15GB of GPU memory. Note that smaller virtual batches can be used as in [4] to reduce memory consumption. For the proposed method to be a net time saving, inference would have to take 3.9 hours less. Using our hardware and inference image resolution, time is saved when the inference test set consists of 185 images with 512 × 512 resolution.

3.5.4 Network architecture and training

The Direct Denoiser used in these experiments was a UNet [58] with approximately 12 million parameters, while the unsupervised denoiser was the same Hierarchical VAE [55] used in [4] with approximately 7 million parameters. We chose to give our UNet more parameters than the

Hierarchical VAE to ensure the former had the capacity to learn the full relationship between noisy images and solutions generated by the latter. This may not have been necessary, and training a Direct Denoiser with a lower computational demand would be an interesting topic for future research.

Our UNet had a depth of four, with a residual block [67] consisting of two convolutions followed by a ReLU activation function [68] at each level. Downsampling was performed by convolutions with a stride of two, and upsampling by nearest neighbour interpolation [69] followed by a single convolution with stride one. All convolutions had a kernel size of 3. The number of filters was 32 at the first level and that number doubled at each subsequent level. Skip connections were merged by concatenating the skipped features with the features from the previous level and passing the two through a residual block.

Training followed the same procedure described in [4], with the only difference being that our Direct Denoiser had its own Adamax optimiser [70] with an initial learning rate of 3×10^{-4} that reduced by a factor of 0.5 when validation loss had plateaued for 10 epochs.

3.6 Discussion

Solutions from our Direct Denoiser consistently scored a higher PSNR than consensus solutions of 1,000 samples from HDN. Table 3.1 shows HDN's PSNRs converging towards our direct prediction result with increased sample size. It seems that solutions from our Direct Denoiser are sometimes equivalent to averaging sample sizes orders of magnitude larger than the largest samples size we used in our experiment. Moreover, by looking at the inference times reported in Figure 3.1, the time required to take such a sample size would be impractical for large datasets.

3.7 Conclusions

We have demonstrated that an extension of the unsupervised denoising approach – the Direct Denoiser – can be used to reduce inference time while improving performance when compared the standard inference procedure with up to 1,000 sampled images. We believe our approach will become the default way of producing central tendencies from unsupervised denoising models with the increase in speed potentially allowing an easy adaptation by the community.

While we have evaluated our method only for MSE and MAE loss functions, we believe the approach could also be used with other loss functions such as *Tukey's biweight loss* [71], which might allow us to find regions of high probability density or even the *maximum a posteriori* estimate.

Recent work in image restoration has also suggested the use of more sophisticated perceptual loss functions (see *e.g*. [72]). These types of loss functions would likely only be usable in a supervised setting with clean training data and would be unlikely to work with Noise2Noise or self-supervised methods. However, since the training targets sampled by our VAE are essentially clean images, they should be compatible with different types of complex loss functions, opening the door to using perceptual loss with noisy unpaired data.

4 TOWARDS STRUCTURED NOISE MODELS FOR UNSUPERVISED DENOISING

In this chapter, we address **structured noise**. Specifically, we train deep autoregressive models of structured imaging noise to enable its removal. At this stage, the process for training these models restricts us to signal-independent noise, but we will show that this does not prevent us from enhancing the diagnostic potential of a nanoparticle analyser in Chapter 5.

The work in this chapter is based on: [73] **B. Salmon** and A. Krull, "Towards structured noise models for unsupervised denoising," in Computer Vision – ECCV 2022 Workshops (L. Karlinsky, T. Michaeli, and K. Nishino, eds.), (Cham), pp. 379–394, Springer Nature Switzerland, 2023.

The photoacoustic imaging data was provided by Paul Beard and Nam Huynh from University College London.

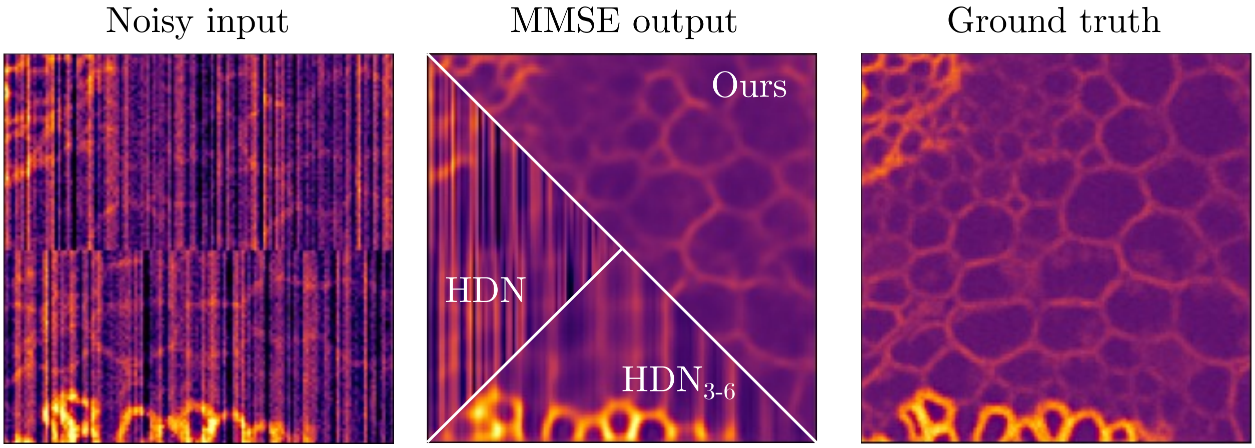

Figure 4.1: **Comparing HDN with our novel autoregressive noise model to HDN and HDN$_{3-6}$ with the standard pixel-independent noise model.** Structured noise can be observed in many imaging modalities. Here, the simulated striped pattern in the noise is designed to mimic real noise as it frequently in some sCMOS cameras. HDN with our novel autoregressive noise model is able to remove structured noise, while HDN with the established pixel-independent noise model only removes the pixel-independent component. HDN$_{3-6}$ performs slightly better, but struggles with long range correlation. Just as in the standard HDN method, we can sample possible solutions from the VAE and compute the minimum mean square error (MMSE) estimate by averaging them.

4.1 Introduction

Since the introduction of digital image processing, a plethora of denoising methods have been proposed; [18, 19, 74] to name a few. The last decade however, has seen a revolution of the field, with Deep Learning (DL) emerging as the technology capable of producing the most accurate results [75, 22]. Traditional supervised DL-based methods [5, 6] are the most consistently performant, but they require the collection of paired training images containing the same content we would like to denoise, which is not always possible. In recent years this problem has been addressed by new self- and unsupervised methods [25, 31, 43, 32, 61, 59, 26, 4], which can be trained on individual (unpaired) noisy images, *e.g.*, the very images that are to be denoised. Two of the newest unsupervised techniques [26, 4], referred to as DivNoising and HDN [4], have demonstrated performance close to and sometimes exceeding that of supervised denoisers.

However, unsupervised methods require an additional ingredient during training. They rely on a mathematical description of the imaging noise, called a *noise model*. The noise model is estimate of the probability distribution, $p(\mathbf{x}|\mathbf{s})$, over the noisy observations $\mathbf{x}$ we should expect for a given underlying clean image $\mathbf{s}$. Noise models can be measured from calibration data [43], bootstrapped (using a self-supervised denoising algorithm) [32], or even co-learned on-the-fly while training a denoiser [26]. Most crucially, noise models are a property of the imaging setup – the camera/detector, amplifier *etc.*, but do not depend on the object that is being imaged. That is, once a noise model has been estimated for an imaging setup it can be reused again and again, opening the door for denoising in many practical applications.

Previous noise models used in this context are based on a conditional pixel-independence assumption. That is, the model assumes that for an underlying given clean image $\mathbf{s}$, noise is generated independently for each pixel in an *unstructured* way, similar to adding the result of separate dice rolls to each pixel without considering its neighbours. This assumption is reasonable for many imaging setups, such as for fluorescence microscopy, where noise is often thought of as a combination of Poisson shot noise and Gaussian readout noise [76]. For simplicity, we will refer to this type of noise model simply as *pixel-independent*.

Unfortunately, many imaging systems, such as photoacoustic (PA) imaging [77], do not adhere to this property and can produce structured noise. In practice, even in fluorescence microscopy the pixel-independence assumption does not always hold due to the camera's complex electronics, and many fluorescence microscopy setups suffer from noise that is partially structured. Figure 4.1 shows an example of simulated structured noise with a pattern close to what is produced by many scientific Complementary Metal–Oxide–Semiconductor (sCMOS) cameras [78].

When DivNoising methods are applied to data containing structured noise which is not accurately represented in their noise model, these methods usually fail to remove it[1]. While Prakash *et al*. [4] show that the effects of this problem can be mitigated by reducing the expressive power of their network, we find that this technique fails to remove noise featuring long range correlations.

Here, we present a new and principled way to address structured noise in the DivNoising/HDN framework. We present an autoregressive noise model that is capable of describing structured noise and thus enables HDN to remove it. We evaluate our method quantitatively on various simulated datasets and qualitatively on a PA dataset featuring highly structured noise. We publish our code as well as the our simulated noise datasets.[2]

In summary, our contributions are:

1. We present an autoregressive noise model capable of describing structured noise.
2. We demonstrate that HDN together with our noise model can effectively remove simulated structured noise in situations where the previously proposed approach [4] fails.
3. We qualitatively demonstrate structured noise removal on a real PA data.

[1] The same is true for self-supervised methods such as [25], which discusses this topic explicitly.

[2] Code and datasets can be found at `https://github.com/krulllab/autonoise`.

4.2 Related work

4.2.1 Self- and unsupervised methods for removing structured noise

Noise2Void (N2V) [25] is a self-supervised approach to removing unstructured noise relying on the assumption that the expected value of a noisy observed pixel, conditioned on the those surrounding it, is the true signal. Using what is known as a *blind-spot network*, a model is shown as input a patch of pixels with a random subset masked. It is trained to produce an output that is as close as possible to the pixels it did not see for which, under the aforementioned assumption, its best guess is something close to the true signal.

In the case of structured noise, that assumption is broken. Broaddus *et al*. [61] accommodated for this by masking not only the pixel that is to be predicted, but also masking all those for which the conditional expected value of the target pixel is not the true signal. A drawback of this approach is that one must first determine the distance and direction over which noise is correlated. Another is that a considerable amount of valuable information is sacrificed by masking.

In [4], Prakash *et al*. demonstrated that tuning the expressive power of a HDN can enable it to remove structured noise. This method – HDN$_{3-6}$ – is introduced in Sec. 2.3. Unfortunately, we find that HDN$_{3-6}$ does not work in all cases (see Fig. 4.1) and also comes at a cost; by inhibiting some levels of latent variables, the expressiveness of the model is reduced. Therefore, when we combine HDN with our autoregressive noise model, we keep all levels of latent variables activated to allow for maximum expressive power.

4.2.2 Noise modelling

In [79], Abdelhamed *et al*. proposed a deep generative noise model known as Noise Flow. It is based on the Glow [80] normalising flow architecture and can be trained for both density estimation and noise generation. In their paper, the authors demonstrated how this noise model could be applied to the problem of denoising by using it to synthesise clean and noisy image pairs. Those pairs could then be used to train a supervised denoising network.

A normalising flow based noise model could be used for the purposes of this chapter, but a recent review on deep generative modelling [81] found that autoregressive models perform slightly better in terms of log-likelihood. As will be seen later, this makes autoregressive noise models more suitable in a DivNoising framework.

4.3 Background

The unsupervised denoisers DivNoising [26] and HDN [4] (Sec. 2.3) are enabled by explicit noise models (Sec. 2.2). Before introducing our novel autoregressive noise model, we will recapitulate here the assumptions made by those currently employed.

4.3.1 Pixel-independent noise models

Until now, noise models used with DivNoising and HDN have been based on the assumption that when an image, $\mathbf{x}$, is recorded for any underlying signal, $\mathbf{s}$, noise occurs independently in each pixel. That is, the distribution factorises as

$$p_\eta(\mathbf{x}|\mathbf{s}) = \prod_{i=1}^{N} \prod_{j=1}^{M} p_\eta(x_{i,j}|s_{i,j}), \tag{4.1}$$

where $p_\eta(x_{i,j}|s_{i,j})$ is the distribution of possible noisy values for the pixel in row i and column j given an underlying clean pixel value at the same location. To describe the noise model for an entire image, we only need to characterise the much simpler 1-dimensional distributions for individual pixel values. These pixel noise models have been described non-parametrically using 2-dimensional histograms (using one dimension for the clean signal and one for the noisy observation) [43], or parametrically as an individual normal distribution [76] or a Gaussian Mixture Model (GMM) [32] parametrised by the pixel's signal $s_{i,j}$.

4.3.2 Signal-independent noise models

Although the models described in Eq. (4.1) are unable to capture dependencies between pixels, they can describe a dependency on the signal underlying each pixel. For many practical applications, this is essential. For example, fluorescence microscopy is often heavily affected by shot noise [76], which follows a signal-dependent Poisson distribution.

However, in this work, we will consider only a more basic case in which the noise does not depend on the signal and is purely additive. In this case, we can write

$$p_\eta(\mathbf{x}|\mathbf{s}) = p_\eta(\mathbf{n}), \tag{4.2}$$

with $\mathbf{n} = \mathbf{x} - \mathbf{s}$. For a pixel-independent noise model, Eq. (4.1) becomes

$$p_\eta(\mathbf{x}|\mathbf{s}) = \prod_{i=1}^{N} \prod_{j=1}^{M} p_\eta(n_{i,j}). \tag{4.3}$$

In Sec. 4.4.1, we will introduce our novel autoregressive noise model which assumes signal-independence but drops the pixel-independence assumption. We address the more general case of combined signal- and pixel-dependence in Chapter 6.

4.4 Method

4.4.1 Deep autoregressive models

Generally, the distribution of any high dimensional variable $\mathbf{v} = (v_1, \ldots, v_N)$ can be written as the product,

$$p(\mathbf{v}) = \prod_{i=1}^{N} p(v_i|v_1, \ldots, v_{i-1}), \tag{4.4}$$

of 1-dimensional distributions for each element conditioned on all previous elements.

Oord *et al*. [51] proposed using a CNN to apply this technique to images, $\mathbf{x}$, in an algorithm known as PixelCNN. Let i^* index pixels in a row-major ordering such that $< i^*$ indexes all the

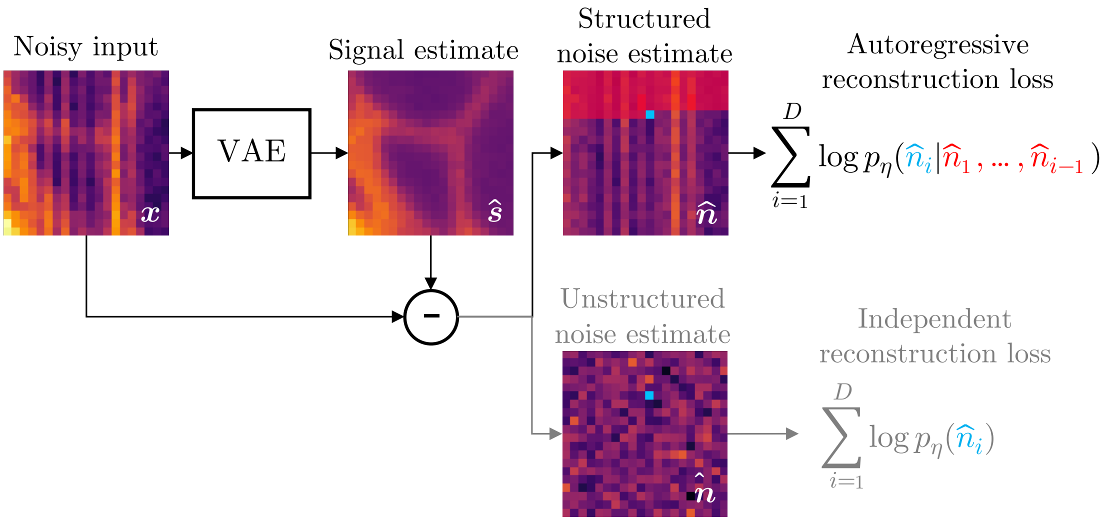
Noisy input
VAE
Signal estimate
Structured noise estimate
Autoregressive reconstruction loss
x
$\hat{s}$
$\hat{n}$
$\sum_{i=1}^{D} \log p_\eta(\hat{n}_i|\hat{n}_1, ..., \hat{n}_{i-1})$
-
Unstructured noise estimate
Independent reconstruction loss
$\hat{n}$
$\sum_{i=1}^{D} \log p_\eta(\hat{n}_i)$

Figure 4.2: **Our autoregressive noise model as a component in the DivNoising framework.** Divnoising trains a VAE to describe the distribution of noisy images $\mathbf{x}$. It does so by sampling clean images $\mathbf{s}$ and using a noise model as part of its loss function, called *reconstruction loss*. The reconstruction loss assess the likelihood of network output $\mathbf{s}$ giving rise to original noisy training image $\mathbf{x}$. It is defined as the logarithm of the noise model. In both cases, for the pixel-independent noise model and our autoregressive noise model, the reconstruction loss can be computed efficiently as a sum over pixels. For the pixel-independent noise model, this is done based on the conditional independence assumption by summing over the pixel noise models $\log p(n_{i,j})$, modelled as a Gaussian Mixture Model (GMM). In our autoregressive noise model we sum over the conditional distributions $p(\hat{n}_{i,j}|\hat{n}_1, \ldots, \hat{n}_{i-1})$ for the noise in each pixel conditioned on the previous pixels, *i.e.*, the pixels above and left. Our noise models describes these conditional distributions using a modified version of the PixelCNN [82] approach, which is implemented as an efficient fully convolutional network, outputting the parameters of a separate GMM for each pixel.

pixels above and directly to the left of pixel i^*. PixelCNNs models the distribution of images as

$$p(\mathbf{x}) = \prod_{i^*}^{N \times M} p(x_{i^*}|x_{<i^*}). \tag{4.5}$$

This model can be parametrised with a CNN by adequately shaping the network's receptive field. When applied to an image, the network outputs the parameters of the 1-dimensional conditional distribution for each pixel.

The model can also be conditioned on external information $\mathbf{c}$, such as a one-hot encoded label or another image, by passing $\mathbf{c}$ as an additional input to the network. The distribution of each pixel is then conditioned on preceding pixels and the external information,

$$p(\mathbf{x}|\mathbf{c}) = \prod_{i^*}^{N \times M} p(x_{i^*}|x_{<i^*}, \mathbf{c}). \tag{4.6}$$

4.4.2 Autoregressive noise models

Under the signal-independence assumption (Eq. (4.2)), a structured noise model can be implemented as an image model for the distribution of noise images $\mathbf{n}$. We use the PixelCNN approach to implement this model. Specifically, our noise model distribution factorises as,

$$p_\eta(\mathbf{n}) = \prod_{i^*}^{N \times M} p_\eta(n_{i^*}|n_{<i^*}). \tag{4.7}$$

where $p_\eta(n_{i^*}|n_{<i^*})$ are the conditional pixel distributions described by a PixelCNN network by outputting the parameters of a GMM for each pixel.

To train our autoregressive noise model, we require training images containing pure noise. In practice, such noise images might be derived from dark areas of the image, where the signal is close to zero, or could be explicitly recorded for the purpose, e.g. by imaging without a sample. Our loss function is

$$\mathcal{L}(\eta; \mathbf{n}) = -\sum_{i^*=1}^{N \times M} \log p_\eta(n_{i^*}|n_{<i^*}). \tag{4.8}$$

Once our noise model is pre-trained, we can proceed to our HDN model for denoising. We follow the training process described in [4] and use Eq. (2.27) as a loss. Note that this contains the noise model $\log p_\eta(\mathbf{x}|\mathbf{s} = f_\theta(\mathbf{z}))$. Considering Eq. (4.2), we can compute $\mathbf{n} = \mathbf{x} - f_\theta(\mathbf{z})$ and insert it into Eq. (2.27).

It would be possible to train a signal-dependent autoregressive noise model by passing an underlying signal as an additional input to the network as in Eq. (4.6). However, this would require paired training data. Unlike for pixel-independent noise models, this training data would not only need to represent the range of signal intensities that will be encountered, but also the signal patterns and structures that will be encountered. If the patterns in the signals, and therefore the patterns in noise variances, are too simplistic, the noise model will not be accurate for the real noisy data. Therefore, the paired data needed to pre-train a signal-dependent autoregressive noise model would be the same paired data that could simply be used to train a supervised denoiser. At this stage, the problem of removing structured and signal-dependent noise with unpaired noisy observations therefore remains open. We will return to this challenge in Chapter 6, where we restrict our assumptions about the structure of the noise. For now, we will demonstrate the effectiveness of deep autoregressive noise models for unsupervised denoising.

4.5 Experiments

We use a total of 5 datasets in our experiments, one is intrinsically noisy PA data and the other four are synthetically corrupted imaging data.

4.5.1 Synthetic noise datasets

While datasets of paired noisy and clean images are not needed to train our denoiser, they are needed to quantitatively evaluate the denoiser's performance using metrics such as PSNR. The method proposed here is currently only capable of removing signal-independent noise, with the extension to signal-dependent noise being left for future work. We are not aware of any real datasets of paired noisy and clean images that do not contain signal-dependent noise, and have therefore created synthetic pairs by adding signal-independent noise to clean images for the purpose of quantitative evaluation. The very noise images that were added to the clean images in the simulated datasets were used to train their noise models but this was only for convenience. Any dataset of noise recorded under the same conditions as the signal could be used.

Convallaria sCMOS Broaddus *et al*. [61] took 1,000 images of a stationary section of a *Convallaria* with size 1,024×1,024. Each image contained signal-dependent noise, but the average of the 1,000 images is an estimate of the ground truth. We normalised this ground truth and split it into patches of size 128×128. For each patch, we added the same sample from the standard normal distribution to the upper 64 pixels in a column, taking a different sample for every column, and then did the same for the lower 64 pixels. We then added pixel-independent Gaussian noise with a standard deviation of 0.3. This was an attempt to produce noise similar to the sCMOS noise shown in Figure 6 of [83].

Brain CT 2,486 clean Computed Tomography (CT) brain scan images were taken from Hssayeni [84] and centre cropped to size 256×256. Independent Gaussian noise was generated with a standard deviation of 110. This noise was smoothed by a Gaussian filter with a standard deviation of 1 vertically and 5 horizontally. More independent Gaussian noise with a standard deviation of 20 was added on top of that. Finally, we subtracted and shifted the noise to have zero mean. This noise was intended to be similar to the CT noise shown in Figure 3 of [85].

KNIST The Kuzushiji-MNIST dataset was taken from Clanuwat *et al*. [86]. The data was normalised before adding a value of 1 to diagonal lines to create a stripe pattern. Independent Gaussian noise with a standard deviation of 0.3 was then added on top. This was intended to demonstrate how HDN$_{3-6}$ with a pixel-independent noise model fails on long range, strong correlations while HDN with our noise model is successful.

4.5.2 Photoacoustic dataset

PA imaging is the process of detecting ultrasound waves as they are emitted by tissues that are being made to thermoelastically expand and contract by pulses of an infrared laser [87]. The resulting data is a time series, and noise samples can be acquired by taking a recording while the infrared laser is not pulsed.

This particular dataset is afflicted with structured noise (see Fig. 4.4) that is thought to have been caused by inter-pixel sensitivity variations. It consists of 468 observations of a signal and 200 observations of only noise, with size 128×128.

4.5.3 Training the noise model

The noise model used in experiments uses the architecture in van den Oord *et al*. [82], modified to output the parameters of a Gaussian mixture model. We used the same hyperparameters for each dataset. Those hyperparameters were 5 layers, 128 feature channels and a kernel size of 7. The output of the network was the parameters of a 10 component Gaussian mixture model for each pixel. The Adam optimiser with an initial learning rate of 0.001 was used, and learning rate was reduced by a factor of 0.99 every epoch. Every dataset was trained on for a maximum of 12,000 steps, but a patience of 10 on the validation loss was used to avoid overfitting on the training set. Images were randomly cropped to 64×64, except the kanji data which was trained on full images. All experiments used a batch size of 8.

4.5.4 Training HDN

The HDN architecture was based on that of Prakash *et al*. [4] and was kept the same for all experiments. Six hierarchical latent variables were used, each with 32 feature channels. There was a dropout probability of 0.2, and to prevent KL vanishing, the free bits approach [88] was used with a lambda of 0.5. The Adamax optimiser was used with a learning rate of 3×10^{-4} and learning rate was reduce by 0.5 when the validation loss plateaued for more than 10 epochs. The same patch and batch size as in the training of the noise model was used.

4.5.5 Denoising with autoregressive noise models

Each of the 4 datasets was denoised using HDN with a pixel-independent Gaussian noise model, HDN$_{3-6}$ with a pixel-independent Gaussian noise model and HDN with our autoregressive noise model. For each test image, 100 samples were generated from each trained model and averaged to produce an MMSE estimate. Each result is shown in Fig. 4.3, with peak signal-to-noise ratio calculated for the datasets where ground truth is available.

The highest PSNR was achieved by HDN with our noise model. For all of the datasets, HDN with a Gaussian pixel-independent noise model seemed to remove only the pixel-independent

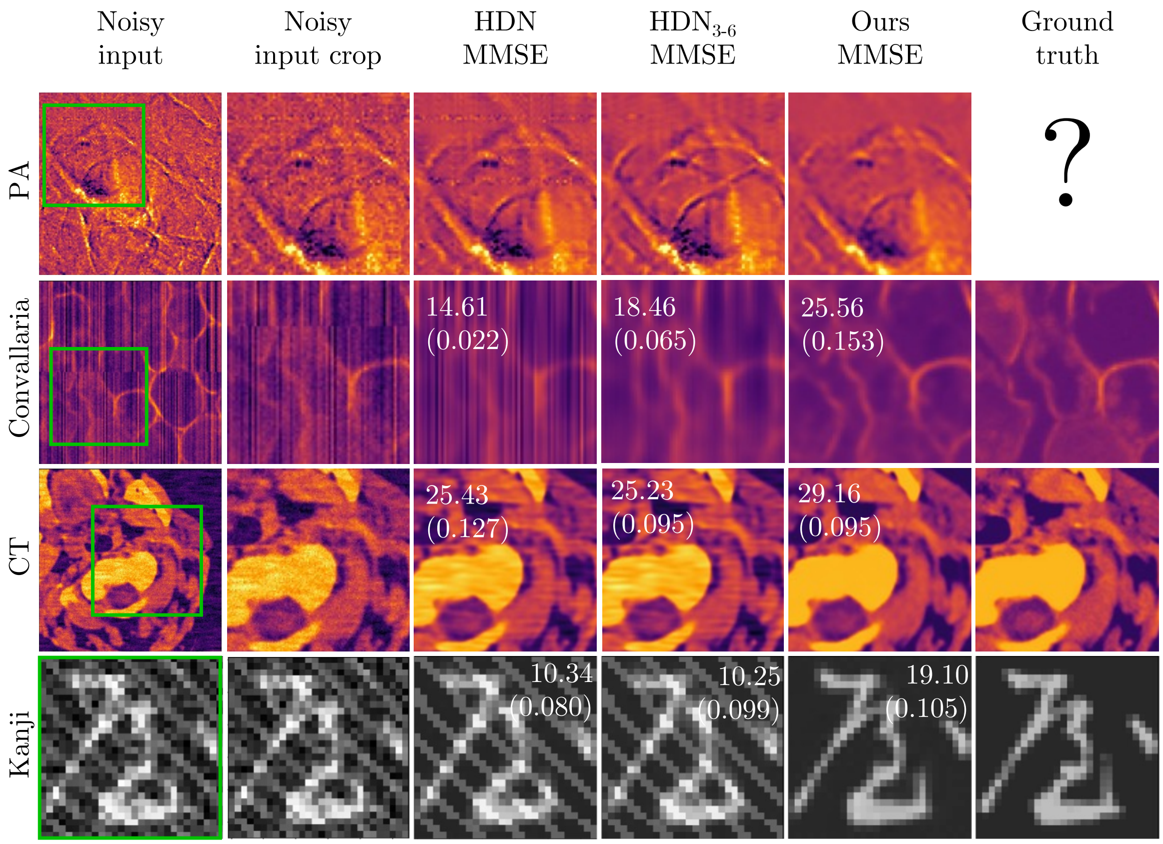

Figure 4.3: **Denoising results.** Here we compare the outputs of different methods on various datasets. The overlaid numbers indicate the mean PSNR values on the dataset after three experiments with the standard deviation in brackets. We find that HDN with a pixel-independent noise model is able to effectively remove some structured artefacts, by removing layers of the latent space space [4], but fails for larger scale structures, spanning over tens of pixels. In contrast, our method reliably removes all small- and large-scale structured noise.

component of the noise, while retaining the structured parts. In some cases, HDN_{3-6} manages to partially remove structured noise.

For the KNIST dataset, both HDN and HDN_{3-6} fail to remove the diagonal lines, which are completely removed by our structured noise model.

We believe that HDN_{3-6} is unable to remove these noise structures because they feature long range correlations, which are not only captured by the two lowest latent variables but also by others in the hierarchy, entangled with the signal.

Similarly, for the PA dataset, only our autoregressive noise model is able to remove the structured recording noise. Here, however, we find that our method produces a slightly blurred result. We attribute this to the limited amount of available noise model training data for this dataset. To avoid overfitting, we had to stop noise model training early in this case, which we believe leads to a sub-optimal end result.

4.5.6 Evaluating the noise model

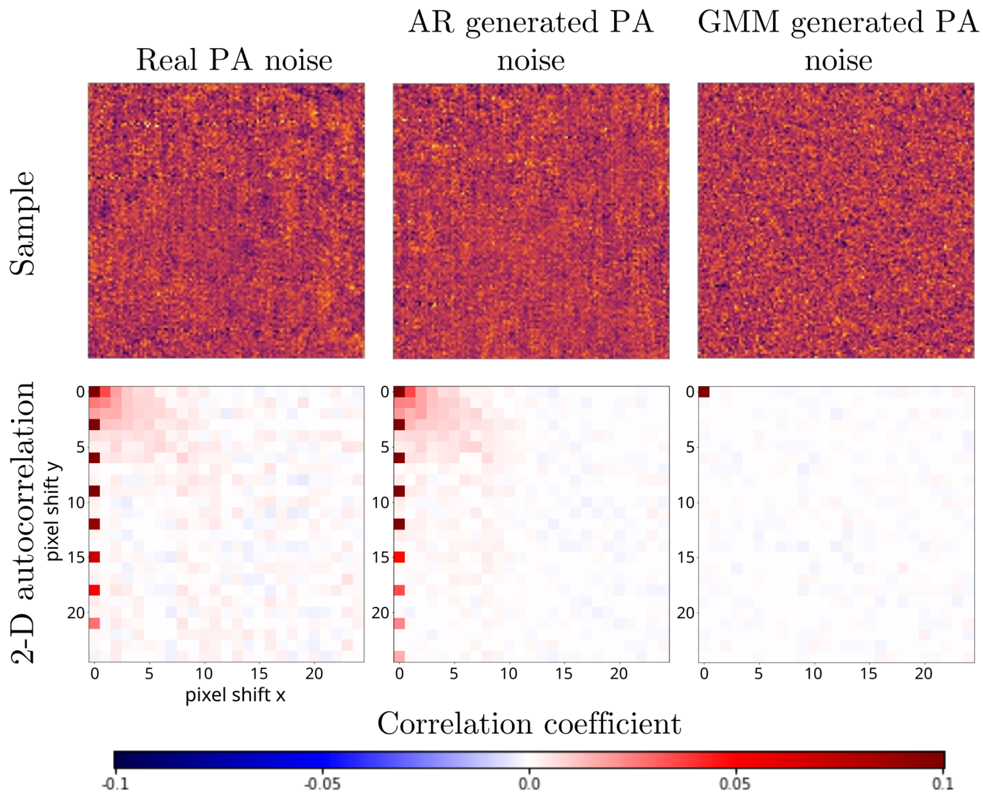

Figure 4.4: **Comparing the statistics of pixel-independent noise models and our new autoregressive model.** Here, we compare generated PA noise samples from our noise model (AR) and a pixel-independent GMM noise model to real PA noise. The auto-correlation function compares different shifted versions (pixel shift) of the noise images in both directions, characterising the dependencies between pixels values at various distances and directions, *i.e.*, the structure of the noise. As expected, the pixel-independent noise model is unable to capture any such dependencies present in the real noise. In contrast, our autoregressive noise model can faithfully capture and reproduce even longer range dependencies.

To show how the autoregressive noise model is able to capture dependencies across an image, we calculated the 2-dimensional autocorrelation of the real noise from the PA data, samples of noise generated from our autoregressive noise model and samples of noise generated by a pixel-independent noise model. Each of these autocorrelation graphs are shown in Fig. 4.4, along with an image of each type of noise for visual comparison.

4.5.7 Choice of autoregressive pixel ordering

Some might be concerned that the choice of autoregressive ordering should take into account the direction of dependencies in the noise, but, fundamentally, this is not the case. Eq. (4.7) generally holds for any distribution of images and also regardless of the used pixel order.

Take, for example, the simulated noise in the Convallaria sCMOS dataset which is designed to be correlated vertically but not horizontally. In the modelling of the noise in this dataset, the distribution of the possible values of one pixel will be more concentrated if it is a function of the other pixels in the same column. However, considering Eq. (4.7), the autoregressive model must sweep through the whole image one pixel at a time. Therefore, no matter if we choose a row-major or column-major ordering, for at least one pixel the distribution has to be computed without considering relevant correlated pixels. On the other hand, in both cases only one pixel in a column can be a function of all relevant, correlated pixels. Both a row-major and column-major ordering of pixels can achieve this if they have a large enough receptive field.

To demonstrate that there are no practical disadvantages arising from the choice of the pixel order, we ran the experiment on the Convallaria sCMOS dataset with transposed images, which corresponds to changing the pixel order. Fig. 4.5 shows the results of this experiment, where almost no perceptual difference between the MMSE of the two experiments can be detected and only a slight difference in mean PSNR is recorded.

4.6 Conclusion

We have presented a novel type of noise model to be used within the DivNoising framework that addresses structured noise and outperforms HDN$_{3-6}$ on highly structured, long range noise artefacts. Both the noise model and DivNoising framework can be trained without matched pairs of clean and noisy images. Instead, practitioners require a set of noise samples and the

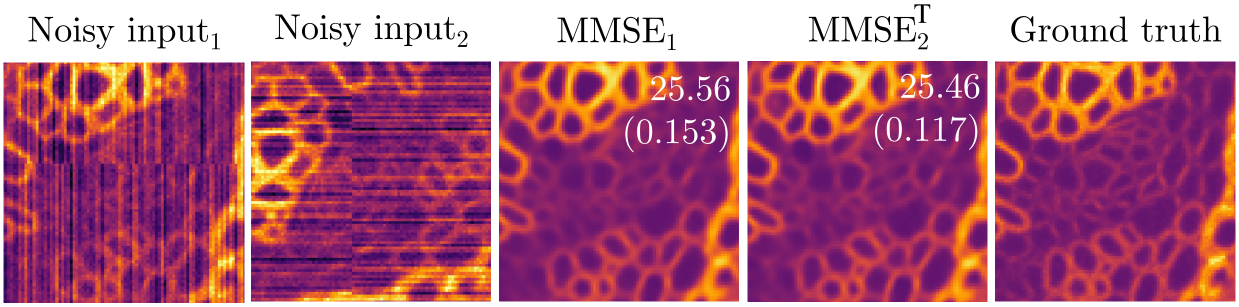

Figure 4.5: **Our noise model can capture noise patterns regardless of their orientation or the direction of pixel ordering.** To demonstrate this, we reran the experiment (including training of the noise model and VAE) on a transposed version of the Convallaria sCMOS dataset. This is equivalent to using a column-major ordering of pixels to train the noise model, while the orignal experiment used a row-major ordering. We compare denoising results carried out on the original Convallaria sCMOS dataset (Noisy input$_\mathbf{1}$, MMSE$_\mathbf{1}$) to the transposed version of the dataset (input$_\mathbf{2}$, MMSE$_\mathbf{2}$). We have transposed the result MMSE$_\mathbf{2}^T$ again to allow for easier comparison. The overlayed numbers indicate the average PSNR and its standard deviation (in brackets) over three reruns of the experiment.

images that are to be denoised. We believe this can potentially have great impact, by enabling applications with structured noise for which no paired data is available.

The key difference between our noise model and those that had been used before [43][26][32] is that ours evaluates the probability of a noise pixel conditioned on other pixels in the image, while previously used noise models evaluate the probability of each pixel independently.

Currently, our method is limited to signal-independent noise, which makes a direct application impossible for many settings, such as fluorescence microscopy, where data is usually affected by signal-dependent Poisson shot noise. However, we do believe that we have made the first step towards widely applied unsupervised removal of structured noise.

In Chapter 6, we extend this noise model to learn the distribution of signal-dependent structured noise by making assumptions about the nature of the noise structures.

5 HIGH THROUGHPUT ANALYSIS OF RARE NANOPARTICLES WITH DEEP-ENHANCED SENSITIVITY VIA UNSUPERVISED DENOISING

In the previous chapter, we presented a method to remove structured, signal-independent imaging noise using an unsupervised denoiser with a deep autoregressive noise model. In this chapter, we adapt that method to the light scattering signal of flow cytometry. By combining our denoiser with an optofluidic device tuned for nanoparticle detection, we realise a nanoparticle analyser that simultaneously achieves high scalability, throughput, and sensitivity levels.

This chapter is based on: [89] Y. Iwamoto*, **B. Salmon***, Y. Yoshioka, R. Kojima, A. Krull, and S. Ota, "High throughput analysis of rare nanoparticles with deep-enhanced sensitivity via unsupervised denoising," Nature Communications, vol. 16, no. 1, p. 1728, 2025.

*Equal contribution.

I, B. Salmon, with Y. Iwamoto, A. Krull and S. Ota conceived the idea and proactively promoted the project. Y. Iwamoto, Y. Yoshioka, A. Krull, S. Ota and I planned and performed the experiments and contributed to experimental data analysis. Y. Iwamoto, A. Krull, S. Ota and I contributed the denoising method development and data analysis. Y. Iwamoto, A. Krull, S. Ota and I drafted the manuscript, and all the authors provided feedback. Y. Yoshioka and R. Kojima contributed to preparation of the materials to analyse. Notably, I did not directly contribute to development of the nanoparticle analyser, sample preparation or data collection, although I did contribute to deciding what data should be collected.

5.1 Introduction

The rapidly evolving field of biological nanoparticle research, crucial to a wide range of biology [90, 91, 92], medicine [93, 94], and material engineering [95, 96], urgently demands methods that can analyze both the physical and chemical properties of individual particles at a scale. Many key applications in these fields require both high sensitivity and scalability to enable high-resolution and comprehensive analysis of heterogeneous populations [97, 98], including the identification of rare subpopulations within diverse samples. This requirement becomes more challenging with smaller particles, as efforts to enhance detection sensitivity often compromise throughput.

For instance, extracellular vesicles (EVs), nano-sized lipid cargoes secreted by living cells, are intensively studied to reveal their important roles as intercellular communication tools [99]. Accurate and scalable profiling of EVs based on both size and markers is essential [100, 101] to understand their diverse functions in physiological processes and to pinpoint specific populations vital in pathological processes [102, 103]. For small EVs, however, conventional high-sensitivity multi-parametric analysis techniques [104, 105, 106, 107, 108, 109, 110] struggle with scalability, leading to limited statistical accuracy in detecting minor subpopulations. Critically, these techniques require sufficiently purified samples to detect the rare marker EVs, creating a substantial barrier to exploring the enormous diagnostic potential of EVs in complex body fluids such as serum and plasma. The EV purification process, dependent on laborious procedures and expensive equipment such as ultracentrifuges and size-exclusion chromatography systems, severely limits practical clinical applications and may introduce variabilities depending on the separation methods [111]. Thus, there is a significant need for a multi-parametric nanoparticle analysis method that integrates high sensitivity with the ability to scale.

To address this challenge, here we develop a multiparametric nanoparticle analyser with unparalleled sensitivity, throughput, and scalability by applying the unsupervised deep learning-based denoising approach developed in Chapter 4 to data obtained from an optofluidic apparatus tailored for nanoparticle analysis. We name this analysis method Deep Nanometry (DNM). This unsupervised approach requires only an empty-water time series in addition to the particle time series to be denoised itself for training a deep learning-based model; the model removes instrument- and environment-specific noises and recovers very weak particle signals that were otherwise dismissed. DNM detects polystyrene beads as small as 30 nm based on theoretical fitting of experimental measurements within the size distribution of 40 nm beads at a throughput of over 100,000 events/second (Supplementary Fig. 9.1, Supplementary Fig. 9.10) and scale up to millions of events. This scalable measurement system enables us to detect rare EVs representing 0.002% of the 1,214,392 total particles detected in serum without purification. We demonstrate the cancer detection by accurately identifying CD9 and CD147 double-positive EVs [112] in 0.93% of colorectal cancer patient samples and 0.17% of healthy controls, achieving significant diagnostic accuracy at $p < 0.05$.

5.2 Results

5.2.1 Apparatus and denoising workflow for DNM

In DNM, we employ an optofluidic apparatus specifically tailored for nanoparticle measurement. Details of this analyser can be found in this chapter's accompanying publication, [89].

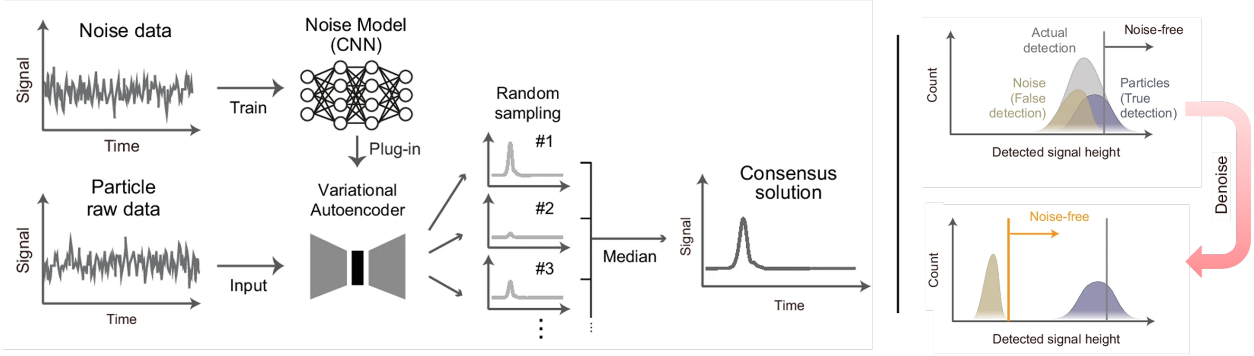

Figure 5.1: Conceptual diagram of the unsupervised denoising method. The first model, an autoregressive CNN, is trained to model background noise in the measurement system. The second model, incorporating the first, is a VAE trained to model the particle signals. Finally, we obtain a consensus solution by randomly sampling many possible solutions from the trained signal model and calculating their median. This denoising approach allows us to push down the noise floor and increase the separation between particle peaks and noise, ultimately enabling the detection of smaller particles.

In the analysis pipeline of DNM, we develop a deep learning-based one-dimensional (1D) unsupervised denoising method to identify and quantify very weak light scattering of nanoparticles buried in various background noises (Fig. 5.1, Supplementary Fig. 9.2(a)). Specifically, we recover noise-free clean particle signals (including the time-series signals between the detected particles) from a given noisy measurement. Our method is a 1D adaptation of the image denoising approach developed in Chapter 4 and has three distinct advantages over other denoising approaches. The first is that, by being deep learning-based, it allows us to automatically identify optimal parameters to remove complex noise that varies depending on environments and experimental and instrumental settings. Conventional design-based methods such as Gaussian filters require manual setting of parameters, making it practically difficult to identify the appropriate parameters for the complexity of the noise on each experiment [113, 23]. Secondly, by being unsupervised, clean, noise-free data are not necessary for training the system, and instead only the background noise and the very data to be denoised are necessary (Fig. 5.1, Supplementary Fig. 9.2(b)). In contrast to the common supervised approach [114, 5], which requires pairs of clean and noise-free data, unsupervised denoising is particularly impactful in nanoparticle analysis, where it is difficult to experimentally obtain noise-free data. Finally, our method acknowledges the ill-posed-ness of the denoising problem, where infinitely many possible clean signals can explain a given noisy measurement [115], by estimating the probability distribution of possible denoised signals for a given noisy measurement. This enables us to randomly sample denoised estimates.

Our denoising approach assumes that a measured noisy time series, $\mathbf{x} = \mathbf{s} + \mathbf{n}$, $\mathbf{x} \in \mathbb{R}^T$, is the sum of the measured signal intensity originating from the light scattered by particles (particle signal) $\mathbf{s} \in \mathbb{R}^T$ and signal-independent background noise $\mathbf{n} \in \mathbb{R}^T$, where T is time series length. The particle signal $\mathbf{s}$ is affected by shot noise with a variance that depends linearly on the unknown underlying clean signal. The background noise $\mathbf{n}$ has a fixed variance and is independent of the particle signal [116], and we therefore describe it as sampled from probability distribution $p(\mathbf{n})$. Our denoising method removes signal-independent noise to recover the particle signal. Even though it is currently not able to remove the remaining shot noise, we experimentally show in the following section that this approach leads to improved sensitivity.

To denoise our measurements, we approximate the probability distribution of possible particle signals for a given noisy measurement, $p(\mathbf{s}|\mathbf{x})$. Our first step in this process is to approximate $p(\mathbf{n})$ with an autoregressive deep learning-based probabilistic noise model $p_\eta(\mathbf{n})$ (with parameters

η) using a CNN with shifted convolutions [82]. This model is trained using samples $\mathbf{n} \sim p(\mathbf{n})$ containing only background noise, obtained by recording particle-free ultrapure water. The noise model then allows us to train a Hierarchical VAE [55] to approximate $p(\mathbf{s}|\mathbf{x})$ with a signal model $q_{\phi,\theta}(\mathbf{s}|\mathbf{x})$ (with parameters ϕ, θ), using samples of noisy measurements, $\mathbf{x} \sim p(\mathbf{x})$, obtained by recording nanoparticle suspensions. This training data is the very data to denoise. Once the whole system is trained, we feed noisy measurements $\mathbf{x}$ into our VAE and randomly sample a set of clean signals from the signal model, $q_{\phi,\theta}(\mathbf{s}|\mathbf{x})$ (Supplementary Fig. 9.2). Since the noise model assumes the signal intensity in its training data is zero throughout, it does not describe signal-dependent noise. Therefore, samples from the signal model will be shot noise-affected particle signals.

For evaluation, the random denoised samples from a trained signal model must be aggregated into a single consensus solution with which peak detection can be performed. We chose to use the point-wise median of the samples as it can be used to interpret the model's confidence for this task. At each time point, denoised random samples can be classified as either above or below the peak-detection threshold. If the median at that point in time is above the threshold, at least 50% of the denoised samples must be above. Therefore, DNM has predicted the probability distribution of denoised solutions to have at least 50% of its probability above the detection threshold. In other words, the model believes it is more likely that there is a particle than not.

For more certainty from the model, a lower percentile, for example the 10th, could be used as a consensus. This would demand that the model place 90% of its samples above the detection threshold to reveal a signal peak. Inversely, a higher percentile, for example the 90th, would require that the model place only 10% of its samples above the detection threshold to reveal a signal peak.

In contrast to the median, the mean of samples is more sensitive to outliers. Even if only a few percent of the noise-removed samples have a sufficiently high intensity above the detection threshold, this could distort the mean and lead to false detections.

5.2.2 Assessment of a 1D denoising method for nanoparticle analysis

Denoising methods have been typically assessed by feeding them noisy inputs and comparing their denoised outputs to corresponding clean ground-truth signals. In the case of nanoparticle detections, however, acquiring pairs of noisy inputs and clean ground-truth signals is intrinsically difficult. To mitigate this issue, we devised the optical setup shown in Fig. 5.2(a) for simultaneously recording pairs of time series with low and high Signal-to-Noise Ratios (SNRs). In the setup, the scattered photons derived from nanoparticles are first split using a beam splitter that reflects 10% of the signal and transmits 90%. Then, the reflected light is modulated by an attenuator, reducing its SNR, and serves as our noisy input time series. The transmitted light maintains a high SNR and serves as an approximation of the underlying clean signal, which we hereafter refer to as our ground-truth time series. Using this setup, we first record a time series containing only background noise through the attenuated arm for 1 s to train our noise model, $p_\eta(\mathbf{n})$. We then simultaneously record a noisy and ground-truth time series of 110 nm polystyrene nanoparticles for 1 s through the attenuated and ground truth arms, respectively; we train our signal model, $q_{\phi,\theta}(\mathbf{s}|\mathbf{x})$ with the noisy particle time series. Subsequently, we apply the trained signal model to denoise the noisy particle time series and compare the result to the ground truth. A local maximum peak detection algorithm is employed to identify the peak positions and heights within the noisy, denoised, and ground-truth time series (details in Sec. 5.4 Method). Also, see more details about denoising in Sec. 5.4 Method, Supplementary Secs. 9.1 and 9.2 and Supplementary Figs. 9.2 to 9.7 and 9.10.

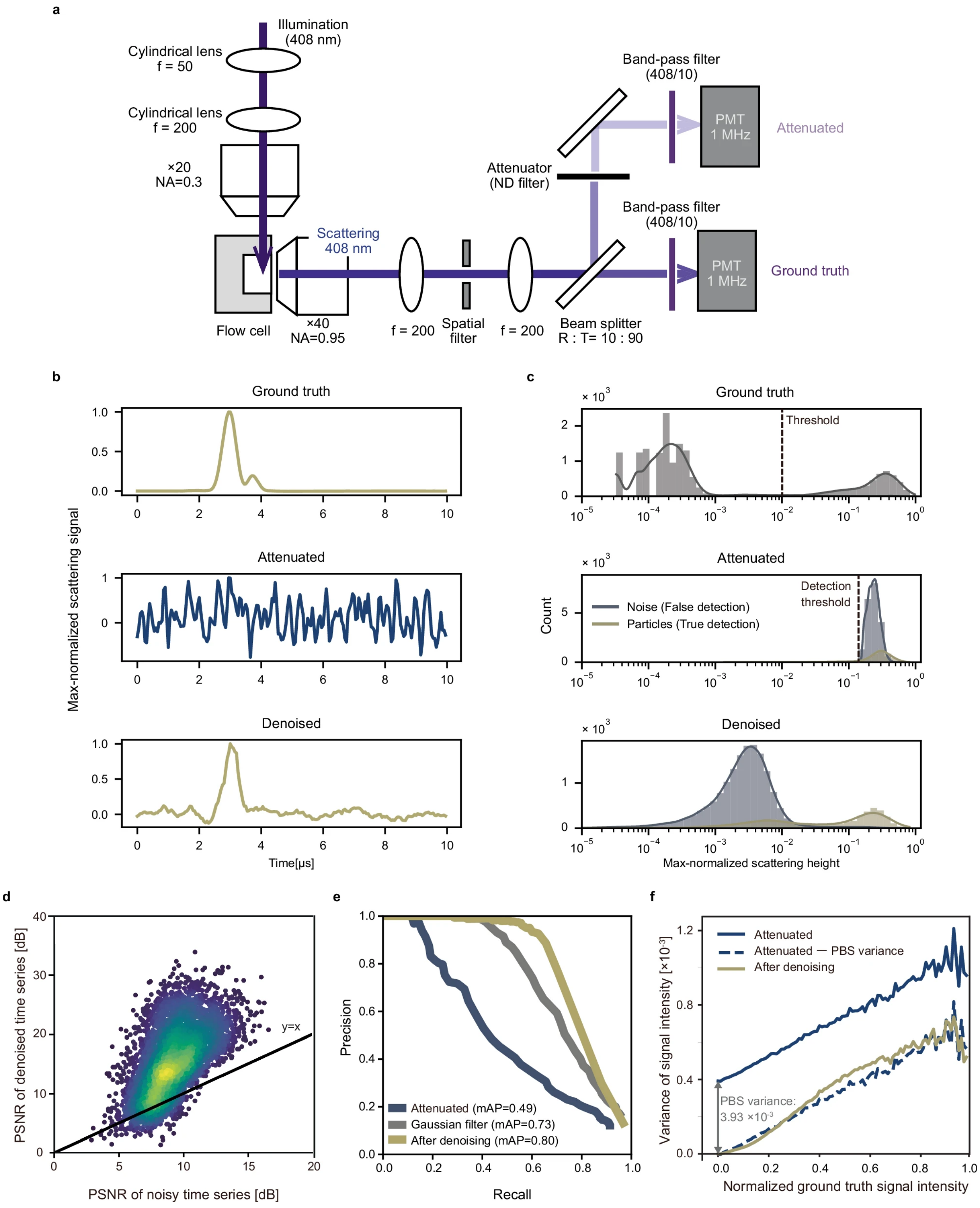

Figure 5.2: See next page for caption.

Figure 5.2: **(a)** Schematic of the optical setup for simultaneously obtaining a pair consisting of a noisy signal and a clean ground-truth signal derived from each nanoparticle. A beam splitter is used to split the scattering signals at a 90:10 ratio, with an absorbing ND filter placed in the 10% side path to further attenuate the signals. **(b)** Simultaneously acquired max-normalised time-series scattering signals derived from 110 nm beads, attenuated signals, and the denoised signal. **(c)** Histogram of the max-normalised scattering peak heights. In the ground-truth time series, peak heights exceeding a threshold of 0.01 (indicated with dashed lines) are considered true nanoparticles, and those lower than the threshold as noise. In the attenuated time series, peaks with a height exceeding 0.05 are considered detections (positives) and only those that coincide with the positions of the ground truth peaks (true nanoparticles) are considered correct detections (true positives). In the denoised time series, peaks at the same location as the positive peaks in the attenuated time series are considered as detections. **(d)** A Scatter plot comparing PSNRs before and after denoising illustrates the quality of the time series relative to the ground truth series (n = 5,010). Points above the line of y = x show improved PSNR, indicating an improved ability to detect particles, whereas points below this line represent a decrease in PSNR, suggesting that the particle signals are erroneously treated as noise and flattened. **(e)** Curve for evaluating peak detection Precision-Recall (PR). The colours indicate the time series generated before filtering, after the optimised Gaussian filter, and after denoising. The mean Average Precision (mAP) is the area under the PR curve. **(f)** The variance of the attenuated signal before denoising (blue line) and after denoising (olive line) as a function of the corresponding normalised ground truth signal intensity. The dashed blue line is the variance of the noisy attenuated signal minus the variance of signal-independent PBS-derived fluidic noise.

Fig. 5.2(b) displays a max-normalised ground truth, noisy and denoised time series from top to bottom. Despite the poor SNR of its input data, our denoiser produced an output remarkably similar to the ground truth. In the histograms obtained for max-normalised scattering height distributions, shown in Fig. 5.2(c), particles and the noise floor become distinguishable only after denoising.

Next, to evaluate the performance of denoising on the signal retrieval from noise, we calculate PSNRs [117] of the denoised time series relative to the ground-truth time series. In our analysis, we divide all time series equally into windows of equal length of 200 sample points and employ the ground truth to disregard windows containing no particle detections. Fig. 5.2(d) shows the PSNR of these windows in the original noisy data plotted on the horizontal axis, and those after denoising on the vertical axis. Points above the dashed line of y = x show improved PSNR, indicating improvements in the particle detection accuracy and subsequent recovery by denoising. Conversely, points below this line represent a decrease in PSNR, suggesting that the particles are erroneously treated as noise and flattened. We evaluate the quality of our denoising results by directly comparing the denoised time series with the corresponding ground-truth time series. We find that denoising improves the PSNR in the vast majority (87%) of cases. We achieved a median value of 15.3 dB, which exceeds both the 9.4 dB of the raw data and the 10.7 dB after Gaussian filtering (Fig. 5.2(d), Supplementary Fig. 9.5).

Finally, to evaluate the effect of the denoising on the ability to detect true particles, we utilise a Precision-Recall (PR) curve [118]. Precision – the number of true positives divided by the sum of true and false positives – and recall – the number of true positives divided by the sum of true positives and false negatives – are key metrics. We generate the PR curve by incrementally increasing the threshold intensity for peak identification from 0 to 1 in 1,000 steps. Notably, at a recall of 0.5, precision increased from 0.47 in the noisy data to 0.98 after denoising (Fig. 5.2(e). The mean Average Precision (mAP) [119] – the area under the PR curve – is 0.80 for our denoised data, significantly surpassing 0.50 for raw data and 0.73 for Gaussian-filtered data. These results suggest that the noise reduction capability of our denoising approach outperforms that of Gaussian filters and that unsupervised denoising realises accurate signal detection.

Furthermore, to thoroughly assess the denoising performances on the particle detection accuracy and the signal retrieval from noise as a function of SNR, we vary the optical density of the attenuator and repeated PR, mAP and PSNR calculations as shown in Supplementary Fig. 9.5. Resultantly, more significant improvements by denoising are observed when the raw data has a lower SNR around 1.4–3. In addition, by comparing the height distributions of peaks detected in the denoised time series and those in ground truth time series, we found that the general shape of the intensity distribution after denoising is preserved for SNRs ranging from high (107.7) to moderately weak value of 2.97 (see more detailed comparison in Supplementary Figs. 9.5 and 9.6). Also, see Supplementary Sec. 9.1 for a detailed discussion of a comparison to a supervised deep learning approach.

Additionally, we analysed the attenuated signal variance before and after denoising as a function of the ground-truth signal intensity to evaluate how closely DNM approaches the shot noise limit. We consider the shot noise limit to be the state in which there is only noise that depends on the optical signal and whose variance is proportional to the ground-truth signal intensity. In the analysis, we first max-normalised the ground-truth time series, then binned these normalised ground-truth signal intensities into 100 intervals. For each bin, we calculated the variance of the attenuated signal at corresponding time points. We plot the results of this analysis in Fig. 5.2(f). Before denoising, the variance of the signal from the attenuated arm increases approximately linearly with ground-truth signal intensity, matching the behaviour of shot noise. The y-intercept of the line – the variance of the noise when the ground-truth signal intensity is zero – is approximately equal to the variance of the phosphate-buffered saline (PBS)-derived fluidic noise. Since the variance of the sum of two independent random variables is additive [120], this indicates that the noise in the attenuated time series is the sum of the PBS-derived fluidic noise and shot noise. To make this clear, we subtracted the variance of the PBS-derived fluidic noise from that of the attenuated time series and plotted it as the dashed line in Fig. 5.2(f).

We then repeated this analysis on the variance of the denoised signal. The slope of the variance here is close to the slope before denoising, indicating that the shot noise indeed remains. However, the y-intercept is now close to zero, indicating the removal of the PBS-derived fluidic noise. The curve slightly deviates from the ideal linear shape due to imperfections in the denoising result.

5.2.3 Benchmarking the sensitivity, throughput, and scalability of DNM

In this section, by removing the beam splitter from the setup in Fig. 5.2(a), we benchmark the sensitivity, throughput, and scalability of DNM without a clean ground truth. As previously described, we train the denoising model using only the noise time series derived from pure water and the noisy signal time series of the actual particles to be detected.

In our initial benchmark, we focus on assessing the sensitivity of DNM to a series of nanoparticles: standard polystyrene (PS) beads with different sizes (27/40/60 nm) and serum-derived EVs.

Fig. 5.3(a) shows the time series signals produced by 27 nm particles before and after denoising. By detecting peaks of separately measured 27/40/60 nm particles before and after denoising, we obtain their scatter height histograms as shown in left and right panels of Fig. 5.3(b), indicating that DNM can resolve 27 nm and 40 nm nanoparticles. We observe a significant reduction in the PBS scatter height, confirming the detection of 27 nm scattered particles that were previously obscured by noise. Indeed, the number of detected peaks decreases by 40% with PBS alone and increases by 38% at 27 nm and by 5% at 40 nm (Fig. 5.3(c)). This reduction indicates that denoising reduces the false detection rate and improves the particle detection recovery. Our method outperforms standard flow cytometry in terms of the sensitivity and coefficient of variation metrics (Supplementary Fig. 9.8). In addition, Fig. 5.3(d) shows the relationship between particle size and denoised scattering height. Fitting using Rayleigh scattering theory [121] (Supplementary Sec. 9.3) produced good agreement, supporting the validity of the detected scattering values.

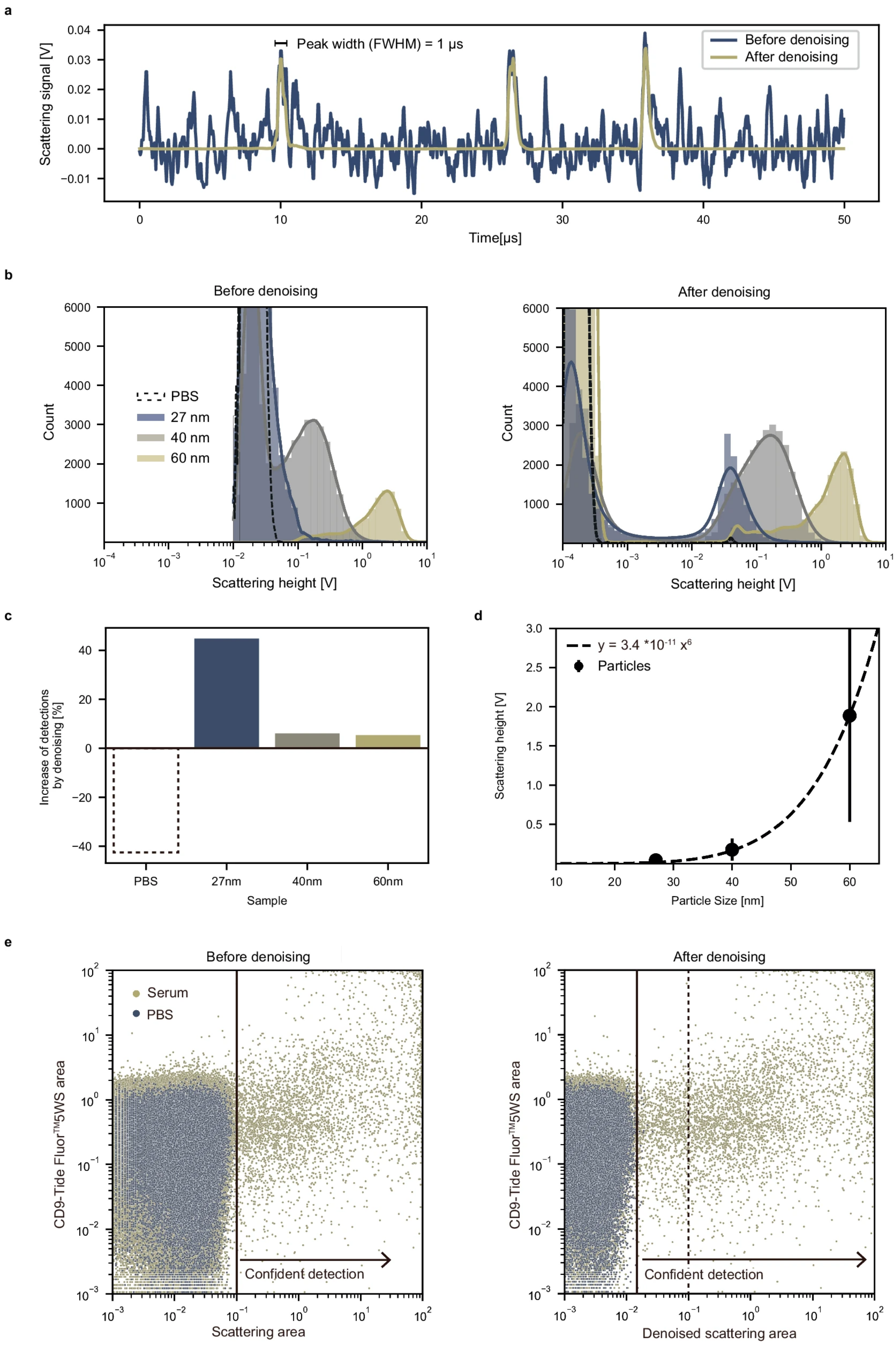

Figure 5.3: See next page for caption.

Figure 5.3: **(a)** Time series scattering signals produced before (blue line) and after (olive line) denoising when measuring 27 nm PS beads for a duration of 50 μs. **(b)** Scattering height histograms of PBS and 27 nm, 40 nm, and 60 nm beads detected by applying a constant trigger threshold to time series data. We consider the detections made in PBS (which does not contain real nanoparticles) to be purely the result of noise. After denoising, the scatter height of a portion of the particle sample is reduced, as is the PBS. This indicates that the denoising procedure separates the signal and noise that were not separated before denoising. In particular, a portion of the 27 nm detections become clearly separated from the PBS detections after denoising. **(c)** Increased ratio of the numbers of scattered peaks detected before and after denoising. Denoising reduces the number of detections in PBS but increases that in particle solutions, indicating that denoising reduces the false detection rate and improves the particle detection recovery. **(d)** Relationship between the particle size and average scattering height after denoising. Error bars indicate the standard deviation of the scattering heights of 12,636 peaks for 27 nm, 28,965 for 40 nm, and 17,299 for 60 nm. The dashed line represents a sixth-power nonlinear regression based on Rayleigh scattering theory [121] (Supplementary Note 4). **(e)** Scatter plot of the CD9-Tide Fluor$^($TM)5WS fluorescence area and scattering area for particles in the immunohistochemically labelled purified serum before and after denoising. By considering detections in PBS as background noise, we define the threshold for particle detection as black lines whereas a dashed line in the right panel represents the background noise level before denoising.

Finally, we measured the intensity of the scattering and fluorescence signals of purified biological particles labelled with anti-human CD9 antibody-conjugated Tide Fluor^{TM}5WS in serum. The left and right panels of Fig. 5.3(e) are plots of CD9-Tide Fluor^{TM}5WS area and scattering area before and after denoising the obtained time series, showing a shift of the noise area to the left. By setting a threshold for confident detection at a peak value exceeding 99.9% of the detected PBS peaks, it can be inferred that the denoising process has effectively enhanced the reliable detection of smaller particles. Collectively, in both the polystyrene and biological nanoparticle cases, we show that denoising reduces the number of false detections and increases the number of true detections, enhancing the accuracy of peak detection.

Next, we benchmark the throughput and scalability of DNM. Assessing the total number of detectable particles, herein referred to as scalability, is critical because the higher the scalability, the more low-concentration or low-positive targets will be detected, determining the detection limit of the system; the scalability is limited by both throughput and the time available for measurement. As a reference, we used a commercially available single-photon counting flow cytometer (SP-FCM), the NanoAnalyzer (NanoFCM Inc.), which exhibits high scattering and fluorescence sensitivities. In our method, the temporal width of each detected scattering peak is approximately 1 μs, theoretically allowing a maximum detection throughput of up to 1,000,000 events per second, assuming equal distances between particles. When we dilute the particle solution to a concentration where the particles arrive every 10 μs on average, considering the random arrival times of particles and the need to minimise coincident events, our system can confidently detect up to 100,000 events per second with minimal aborts (Fig. 5.3(a)). Experimentally, we achieved a throughput of 241,510 events per second by flowing a suspension of 110 nm PS beads and gating singlet events, which accounted for 96.8% of the total events in the scattering width versus area plot (Supplementary Fig. 9.1). Note that this detection throughput varies depending on the input concentration of particles.

The limit of the concentration detectable with DNM based only on scattered light is evaluated by measuring 110 nm polystyrene particles diluted with PBS in a stepwise manner. From the linear relationships observed between the number of particles detected and the dilution rate, as shown in Fig. 5.4(a), we conclude that the samples are detectable at concentrations of at least 2.82×10^3 particles/mL. We note that the detection limit (LoD) can ideally be determined by the mean and variance of the detections when blank samples are measured without the target particles; we experimentally detect no peaks in the PBS time series during the 5-min measurement period. Then, the LoD of simultaneous scattering and fluorescence measurement is assessed using immunohistochemically labelled EVs. In this experiment, we label EVs with anti-human CD9 antibody-conjugated Tide Fluor$^{\mathrm{TM}}$5WS in supernatants derived from human colorectal cancer cell line HCT116 cultures and dilute them 10^1 to 10^6 times with unstained serum. Among the detections defined by the scattering and fluorescence intensities, we assessed the counts and concentration of CD9-positive EVs in DNM and observe that they are proportional to the dilution rate of 10^{-5} (Fig. 5.4(b) and Supplementary Fig. 9.9). This dilution rate is equal to the CD9-positive concentration is 1.1×10^4 particles/mL and the detection rate of 11 particles/min; the

CD9-positive rate is estimated to be 0.04% (Fig. 5.4(c)). The detection limit derived from the antibodies in buffer control (background level of $+3\ \sigma$ for 10-min measurements) is 4 detections per minute or 4.0×10^3 particles/mL. The detection limit derived from blank PBS (background level of $+3\ \sigma$ for 10-minute measurements) is 0.68 detections/minute or 6.8×10^2 particles/mL. In contrast, when the same sample is measured by the SP-FCM for 1 min, we cannot detect CD9-positive particles at a 10-4 dilution rate (Fig. 5.4(b), (c)). The detection limit we can derive from this experiment for the SP-FCM was 3.87×10^5 detections/mL, with a CD9 positive rate of 6.98%.

5.2.4 Detecting rare EVs in serum

Large-scale measurement is critical for detecting an adequate number of rare EVs in nonpurified body fluids containing many other nanoparticles such as lipoproteins and debris [122, 123, 124, 125]. Currently, small-scale measurements require EV purification processes, which not only pose practical cost and time issues but also lead to the loss of EV population fractions and biased analyses [126, 127].

Here, we show that large-scale DNM enabled the direct detection of rare EVs from non-purified serum. In the experiment, we add anti-human CD9 antibody-conjugated Tide Fluor$^{\mathrm{TM}}$5WS and anti-human CD147 antibody-conjugated Alexa 488 to the serum of two donors, dilute the samples in a 50-fold manner in PBS, and detect them with DNM for 2 minutes. The detection threshold is set at SNR = 18 for confident detection, excluding 99.9% of the detected PBS peaks. An analysis of the serum obtained from one donor (Fig. 5.5(a)-(c)) reveals that CD9-positive EVs account for 0.05% of all confident detections (n = 1,214,392), corresponding to 2.7×10^5 detections/mL; CD147-positive EVs account for 0.01% of the total, corresponding to 7.2×10^4 detections/mL; and CD9 and CD147 double-positive EVs account for only 0.002% of the total, corresponding to 1.2×10^4 detections/mL.

Additionally, we demonstrate the power of large-scale DNM for detecting diagnostic EVs. Previous studies show that EVs originating from cancer [112, 128, 129, 130], depression [131], and other

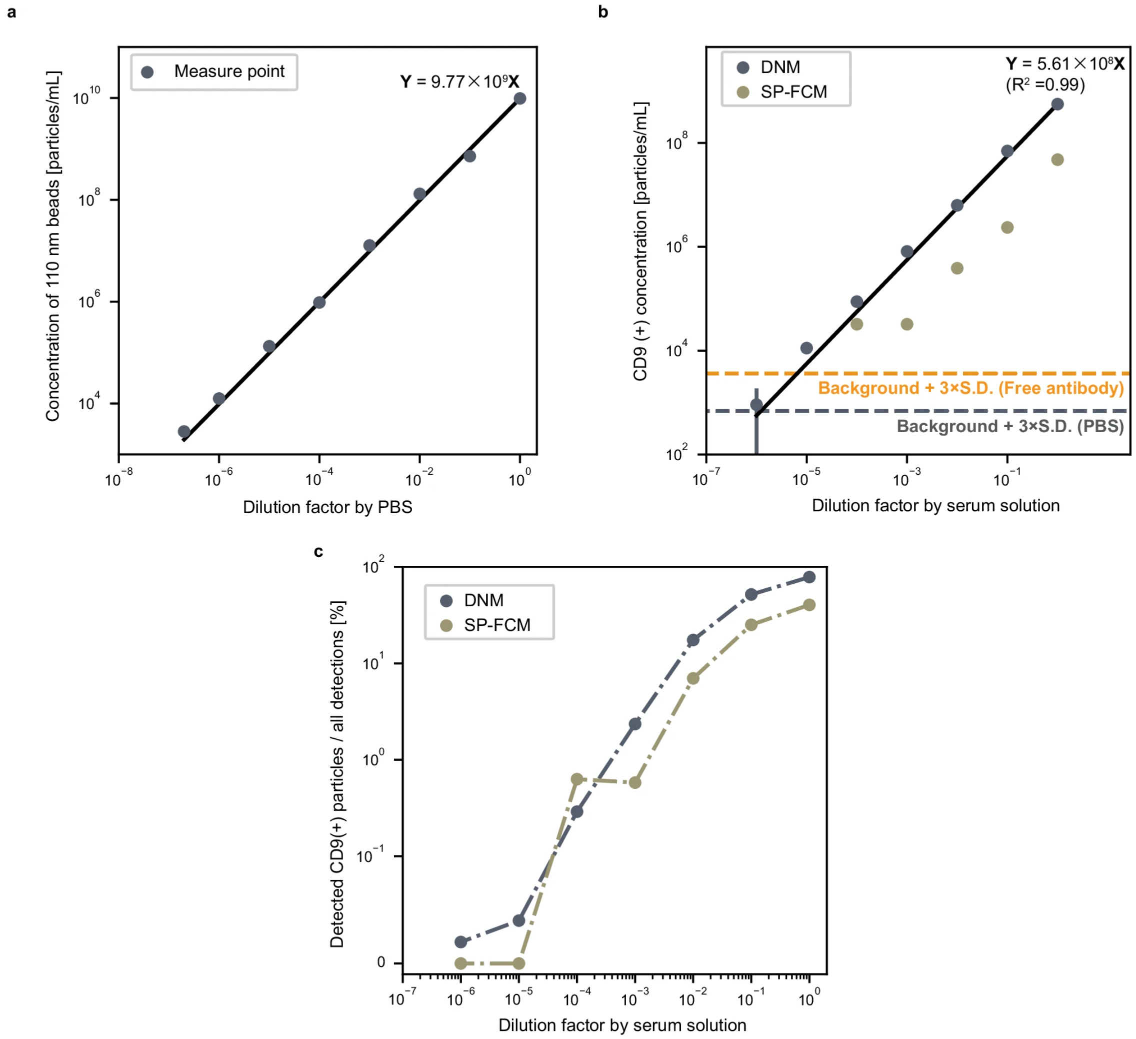

Figure 5.4: **(a)** Scatter plot of the detected concentrations and dilution factors of 110 nm particles (dilutions from 10^{-1} to 10^{-6}) (n = 3, mean, technical replicate). The linear relationship between the detected concentration and the dilution rate indicates the detection accuracy. **(b)** Scatter plot of the concentration of detected CD9 positive EVs and a factor of diluting them with unstained serum (10^{-1} to 10^{-6}). The y-axis is a log scale. The blue and olive points indicate the results of 4-minute DNM and 1-minute SP-FCM measurements, respectively. The black line indicates the linear regression result of the DNM measurements (n = 3, mean ± S.D., technical replicate). The black and orange dashed horizontal lines represent the PBS background mean ± 3×S.D. and the free antibody background mean ± 3×S.D., respectively. **(c)** Scatter plot of the average percentage of CD9 positive particles in the unstained serum and the dilution factor (10^{-1} to 10^{-6}) obtained from the same data used in **(b)**. As dilution increases, the proportion of CD9-positive particles decreases due to the increased presence of particles in the unstained serum. The y-axis is a symmetric log scale: the linear scale is from 0 to 0.1 and the log scale starts from 0.1. The blue and olive points indicate the results of the DNM and SP-FCM measurements, respectively.

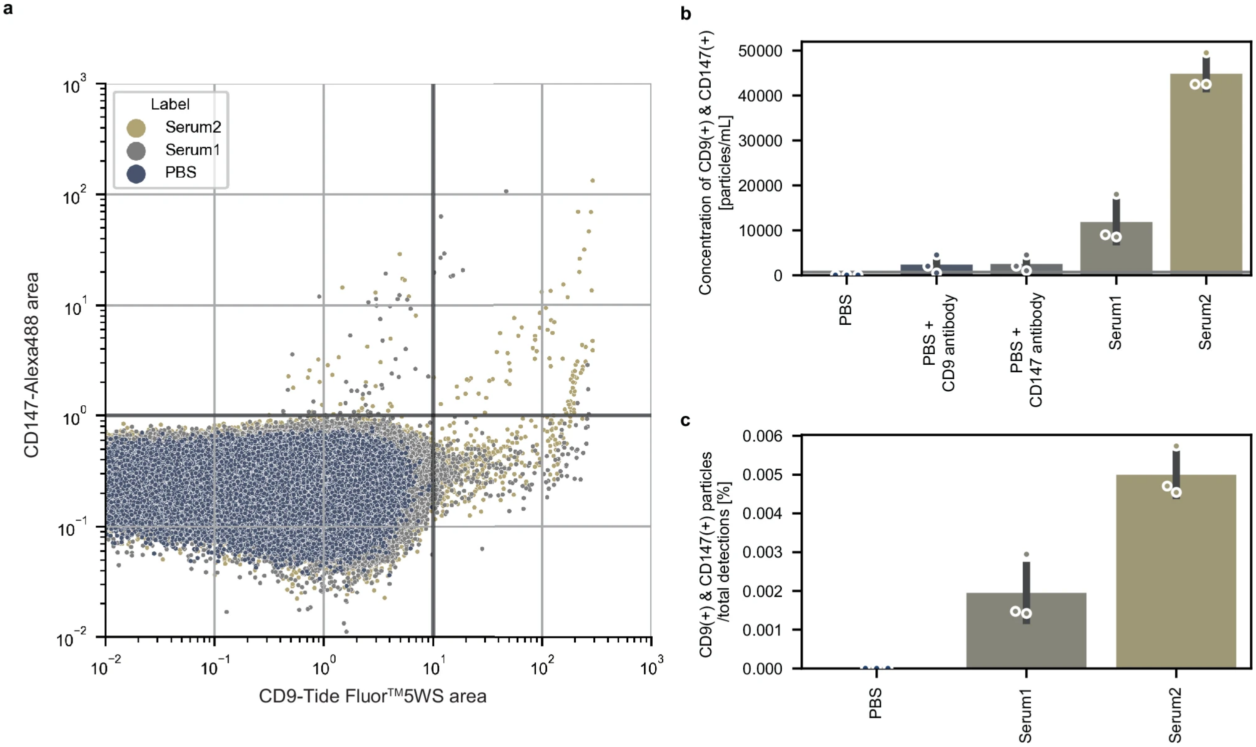

Figure 5.5: **(a)** Scatter plot of the CD147 and CD9 areas for particles in immunohistochemically labelled, nonpurified serum obtained after 2 minutes of measurement. PBS indicates the detected background noise and defines the thresholds (black lines). Serum samples 1 and 2 are nonpurified serum samples acquired from healthy individuals. Bar plot of **(b)** the concentration of CD9 - and CD147- positive particle detections per volume and **(c)** those per total detections in nonpurified serum (n = 3, mean ± S.D., technical replicates). Double positive EVs for CD9 and CD147 detected in serum samples 1 and 2 are compared to measurements of PBS only and PBS samples mixed with CD9 or CD147 antibodies as a control for background noise and false positives derived from antibody aggregation, respectively.

illnesses [132, 133] possess distinctive combinations of specific surface proteins. Increasing the marker combination generally decreases the number of positive detections. In other words, scaling up the analysis process enables the more accurate identification of disease-specific EVs through multiparametric analysis, a potential key to early disease detection. In this study, we focus on colorectal cancer, which has a high mortality rate of 40–50%, and its early detection leads to higher survival rates [134].

In the experiment, we purify nanoparticles from serum samples derived from five cancer patients and five healthy donors and label them with anti-human CD9 antibody-conjugated Tide Fluor$^{\mathrm{TM}}$5WS and anti-human CD147 antibody-conjugated Alexa 488. We analysed the ratio of

CD9 and CD147 double-positive particles, a well-known colorectal cancer biomarker [112], to CD9 single-positive EVs using DNM and the SP-FCM [107]. Figure 5.6(a), (b) show an example set of CD9 and CD147 scatter plots derived from DNM, showing that more CD9 and CD147 double positive EVs are observed in colorectal cancer patients.

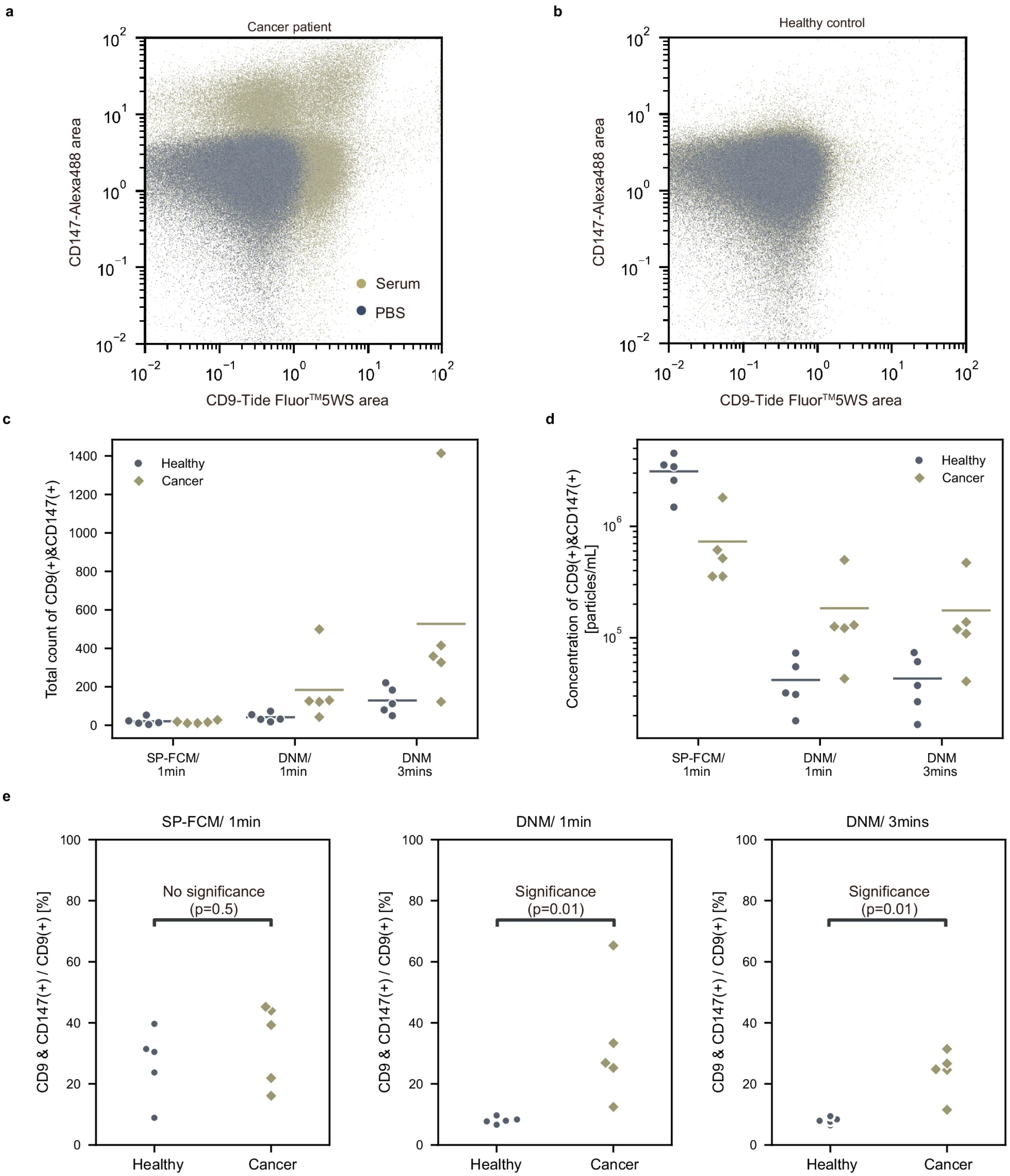

Figure 5.6: See next page for caption.

Figure 5.6: **(a)** Scatter plots of the CD147 area and CD9 area (n = 30,000) in cancer patient serum and **(b)** healthy control serum. **(c)** The total count and **(d)** the concentration of CD9- and CD147-positive EVs from cancer patients and healthy controls, obtained with the SP-FCM measurement for 1 min and DNM measurement for 1 min and 3 min, respectively. The bar shows mean (n = 5 for cancer patients, n = 5 for healthy controls, biological replicates). Sample flow rate of DNM is 1 μL/min and that of SP-FCM is 3.2 nL/min. **(e)** Comparison of the percentages of CD9 and CD147 positive EVs among CD9 positive EVs detected under different conditions. (n = 5 for cancer patients, n = 5 for healthy controls, biological replicates). DNM successfully detected a sufficient number of the targeted EVs that account for only 0.93% and 0.17% of the cancer patients and healthy control serum, respectively, leading to accurate cancer diagnosis results with significance ($p <$ 0.05). To evaluate p-value, we used the Mann-Whitney-Wilcoxon test, a nonparametric two-sided test with Bonferroni correction.

Again, the total number of detectable particles is critical for accurately detecting rare diagnostic EVs. When identical serum is applied to the SP-FCM for 1 min and to DNM for 1 min and 3 min, the mean total numbers of detected particles are 2,224, 31,610, and 97,175, respectively, as shown in Supplementary Fig. 9.11; the concentration of CD9 and CD147 double-positive particles were 1.9 $\times 10^6$ particle/mL, 1.1 $\times 10^5$ particle/mL, and 1.1 $\times 10^5$ particle/mL, respectively, as shown in Fig. 5.6(c), the mean total numbers of CD9 and CD147 double-positive particles were 19, 112, and 328, respectively, as shown in Fig. 5.6(d). Compared to the lower quantification limit of 25 particles estimated based on Poisson statistical simulations (Supplementary Fig. 9.12), CD9 and CD147 double positive particles purified from serum cannot be quantified by the SP-FCM but can be quantified by DNM. Note that the scalability and consequent accuracy difference can become even larger since the measurement throughput and time of DNM are much greater in practice: the throughput can increase up to 100,000 events/s at a higher particle concentration, and a stable measurement lasts for at least one hour. Finally, we compare the percentages of CD9 and CD147 double positive particles among the CD9 positive particles as cancer biomarkers (Fig. 5.6(e)). The SP-FCM reveal no significant differences between the cancer patients and healthy controls, whereas the DNM results significantly differed with $p <$ 0.05 and demonstrates superior diagnostic EV detection performance. This diagnostic sensitivity improvement achieves with DNM is specifically attributed to the ability of DNM to precisely count the low levels of CD9 and CD147 double positive EVs in healthy individuals.

5.3 Discussion

In summary, we developed a technique termed Deep Nanometry, which combines unsupervised deep learning for noise reduction with optofluidic technologies specialised for nanoparticle detection. DNM demonstrated the sensitive detection of polystyrene beads as small as 30 nm based on theoretical fitting of experimental measurements within the size distribution of 40 nm beads, achieving a high throughput detection of over 100,000 events per second and scalability of over a million event analyses. The scalable measurement of DNM enabled the purification-free detection of rare CD9 and CD147 double-positive EVs that account for 0.002% of all confident detections in serum; it also enabled the cancer diagnostic detection of CD9 and CD147 double-positive EVs among CD9 positive EVs, accounting for only 0.93% and 0.17% of the cancer patients and healthy control serum, respectively, with significance ($p <$ 0.05).

We foresee that the unsupervised 1D denoising technique we have developed can impact a wide range of technological fields, where detecting weak 1D signals is crucial. For instance, reducing instrument- or environment-specific background noise is highly beneficial for flow cytometers in general, from the traditional intensity-based to the image information-based [135]. Moreover, there is an increasing need to integrate new data modalities, such as Raman spectroscopy [136] and refractive index measurements, with high-throughput measurements to offer deeper insights into the molecular compositions and structures of the particles. The significance of denoising will be further emphasised as we advance in such measurement technology, given that the signals derived from these modalities are inherently weak.

An important milestone for future work will be developing a method capable of additionally removing signal-dependent shot noise. While in this work, the sensitivity was significantly improved by removing only the signal-independent noise as it was dominant in the scattered signal, we also observed that the signal-dependent noise in fluorescence data, for example, was too intense to ignore. We will return to this challenge in Chapter 7.

The sensitive, rapid, and scalable DNM may find diagnostic and therapeutic applications such as the multiparametric detections of EV-based biomarkers and the development and quality assurance of viral or drug delivery particles in pharmaceuticals and vaccines. Similarly, in research, DNM may provide a comprehensive and high-resolution understanding of EV populations in body fluids, opening avenues for dissecting and deciphering EV heterogeneity and the clinical significance of rare EVs.

5.4 Methods

For full reproducibility information, please refer to this chapter's associated publication [89] Y. Iwamoto, B. Salmon, Y. Yoshioka, R. Kojima, A. Krull, and S. Ota, "High throughput analysis of rare nanoparticles with deep-enhanced sensitivity via unsupervised denoising," Nature Communications, vol. 16, no. 1, p. 1728, 2025.

5.4.1 Data analysis workflow and gating strategy

For all analyses, we calculated height, area, and SNR from the time series data of scattering and fluorescence based on the peak positions detected in the scattering time series signals.

For peak detection, a local maximum-based peak detection algorithm was applied using the 'find peaks' function from the SciPy python library. This peak detection algorithm operates in two steps. First, it differentiates the time series signal and records the peak position at the point where the positive and negative derivative values are inverted. Then, based on the peak height which is the signal value at the detected peak position, we treat a peak as detected only if this height exceeded a detection threshold which is SNR = 2 in this work.

In Fig. 5.3, peaks detected in the ground-truth time series data with SNRs greater than 10 were treated as nanoparticle detections (ground truth labels). Then, a peak in the noise or noise-rejected time series was considered a true positive if it appeared within 20 points (1 μs) of the nearest peak in the ground-truth, and a false positive if it was further away than that. If there was no corresponding ground-truth peak within 20 time points in the noised or denoised time series, it was considered a false negative.

The SNR was calculated by dividing the signal height by the variance of the background noise signal values of 1,000 consecutively recorded points. The signal area was calculated by integrating the signal values contained in a total time window of 3 μs before and after the peak position.

5.4.2 Experimental procedure for denoising

In each experiment, first, we obtain a 1 second scattering time series from particle-free ultrapure water. Then, we acquire the scattering time series of the nanoparticle suspension for an arbitrary time. Following this, we use the ultrapure water time series to train a background noise model. We then train a signal model using the trained noise model and the sample suspension scattering time series. Finally, we use the trained signal model to remove noise from the sample suspension scattering time series.

5.4.3 Model architectures and training process

The models were trained with windows of the complete time series. This is because memory limitations prevented us from loading the entire time series into the GPU for conducting forward and backward passes through the CNNs. Instead, we exploited the repetitive nature of the data by assuming that the full time series could be modeled as the concatenation of smaller time series, each of which was sampled from the same distribution. Specifically, the utilized data were recordings of 1 s of a flow with a sample rate of 20 MHz, producing a total of 2×10^7 data points. Both networks learned to model windows containing 1×10^3 data points.

The noise model architecture consisted of a seven-layer causal CNN that utilized gated blocks from conditional PixelCNN decoders [82]. It created causal convolutions by shifting and employing dilation instead of downsampling. Specifically, layers 2, 4, and 6 used dilations of 2, 4, and 2, respectively. The convolutions had kernel sizes of 7 and 8 filter channels. The log-likelihood loss was calculated using a Gaussian mixture model with two components.

During training, the model used the Adamax [137] optimizer with a learning rate of 0.002, along with a scheduler that reduced the learning rate by a factor of 0.5 after 10 epochs with no negative log-likelihood changes. The data were split into non-overlapping windows with lengths of 1,000 and shuffled. An 80/20 split was used for training/validation, and the batch size was 32. The training process stopped when the negative log-likelihood plateaued for 50 epochs.

The VAE architecture was based on the Hierarchical DivNoising model [4] but with some modifications. We found that using 8 latent variables in the hierarchy instead of 6 [4] was more effective. However, increasing the number of latent variables beyond 8 did not provide any additional benefit. The network consisted of a bottom-up path with gated residual blocks that included batch normalization [138], ELU activation [139], convolution, and gating, as well as a top-down path that used gated residual blocks preceded by the sampling of a latent variable. The architecture contained 8 layers with resampling at every layer, and downsampling was achieved using strided convolutions and upsampling with nearest-neighbour interpolation. For the convolutions, 64 filters with kernel sizes of 3 were used.

During training, we used the Adamax optimiser [137] with a learning rate of 0.0003 and a scheduler that reduced the learning rate by a factor of 0.1 when the loss plateaued for 10 epochs. The data were split into windows with lengths of 1,000, shuffled, and divided based on a 90/10 training/validation split with a batch size of 32. We stopped the training process after the reconstruction loss plateaued for 50 epochs.

5.4.4 Evaluation metrics

Precision-recall (PR) curve and mean average precision (mAP)

PR curve represents the tradeoff between recall and precision depending on the detection threshold in peak detection. We refer to the area under this curve as the mAP, and a value close to 1 indicates an optimal detector.

Peak signal-to-noise ratio (PSNR)

PSNR represents the reproducibility of the signal to be evaluated relative to the clean signal in dB. The maximum signal value was divided by the mean squared error between the clean signal and the signal to be evaluated.

$$\mathrm{PSNR} = 10 \cdot \log_{10}\left(\frac{\text{Max peak height}^2}{\text{Mean square error}}\right) \tag{5.1}$$

The mean squared error is calculated as follows,

$$\text{Mean Square Error} = \frac{1}{T}\sum_{t=1}^{T}(s_t - \hat{s}_t)^2 \tag{5.2}$$

When a total of T data points are obtained, s_t represents the true value (signal intensity at the ground truth) at the t^{th} measured data point and $\hat{s}_t$ represents the estimated value (signal intensity after denoising).

Statistics and reproducibility

All experiments were performed at least twice to ensure reproducibility. The Mann-Whitney-Wilcoxon test two-sided with Bonferroni correction, a nonparametric test, was used for testing. $p >$ 0.05 was indicated as no significant difference (ns). No data were excluded from the analyses.

5.5 Data availability

The raw data of temporal signals before and after denoising and the table for the flow cytometric features, generated in this study have been deposited in the Zenodo database along with the analysis code and made publicly available with open access. Links to the data used in the figures of main manuscript are as follows: `https://doi.org/10.5281/zenodo.14616191`, `https://doi.org/10.5281/zenodo.10521246`, `https://doi.org/10.5281/zenodo.10521358`, `https://doi.org/10.5281/zenodo.10521083`59,60,61,62. A link to the data used in the Supplementary Figs. is as follows: `https://doi.org/10.5281/zenodo.14615945`63. The code for denoising and basic analysis of the measurement data was uploaded to GitHub [`https://zenodo.org/records/14610019`] and made publicly available with open access.

5.6 Code availability

The code for the denoiser, along with example Jupyter notebooks, can be found at https://github.com/krulllab/Deep_Nanometry.

6 UNSUPERVISED DENOISING FOR SIGNAL-DEPENDENT AND ROW-CORRELATED IMAGING NOISE

Up to this point, we have presented methods to remove **structured noise** in microscopy and flow cytometry, but we have been unable to remove structured and **signal-dependent** noise. This limitation stems from being practically unable to train an autoregressive model of such noise without clean signal data. In this chapter, we identify a common feature of structured noise across a variety of microscopy modalities: row-correlation. We take advantage of this characteristic to enable training of a signal-dependent autoregressive noise model using unpaired noisy images, *i.e.*, the data that trains the denoiser. The noise model can then be trained alongside the denoiser.

This chapter is based on: [140] **B. Salmon** and A. Krull, "Unsupervised denoising for signal-dependent and row-correlated imaging noise," in 2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), pp. 2379–2389, IEEE, 2025.

Computations in this chapter were performed using the University of Birmingham's BlueBEAR HPC service, which provides a High Performance Computing service to the University's research community. See `http://www.birmingham.ac.uk/bear` for more details.

Code and tutorials for applying this denoiser to new data can be found at `github.com/krulllab/COSDD`.

6.1 Introduction

Self- and unsupervised learning-based denoisers(*e.g.* [31, 25, 26, 4, 141]) can be trained directly on the data that is to be denoised and have substantially improved the practicality of denoising in scientific imaging. These methods separate the imaging noise from the underlying signal by making assumptions about the statistical nature of the noise. Typically, they assume the noise is (i) signal-independent (purely additive and occurs separate from the underlying signal) [73], or

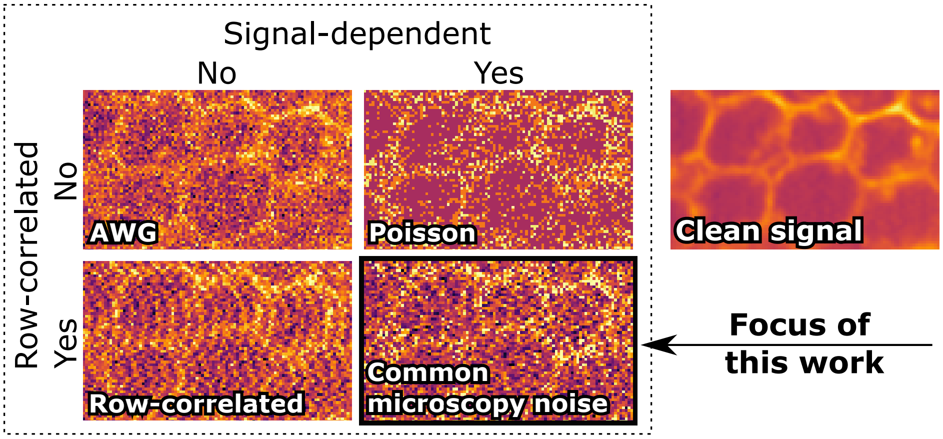

Figure 6.1: Noise is often assumed to be spatially uncorrelated and/or signal-independent, such as Additive White Gaussian (AWG) noise. These assumptions greatly facilitate self- and unsupervised denoising. Unfortunately, real noise in microscopy is usually neither. We present the first unsupervised denoising approach for this challenging scenario.

(ii) spatially uncorrelated (unstructured and occurs separately for each pixel) [25, 47, 26, 142, 60]. Examples of (i) and (ii) are in Fig. 6.1.

In practice, (i) is often broken by the presence of Poisson shot noise [22]. Moreover, many popular scientific cameras and imaging setups break (ii) by producing row- or column-correlated noise (see Sec. 6.3.1 **Row-Correlated Noise**). These types of noise cannot be removed with basic self- or unsupervised methods. Recently, variants of these methods for spatially correlated noise have been proposed, but they are limited to locally correlated noise and can come at the expense of reduced reconstruction quality [2, 4].

In this chapter, we present the first unsupervised deep learning-based denoiser capable of reliably removing signal-dependent noise that is correlated along rows or columns of pixels, as it commonly occurs in microscopy data. The approach is illustrated in Fig. 6.2. Our method requires neither examples of noise-free images, which can be impossible to obtain (*e.g.* [143]), nor pre-trained noise models [26, 73, 47] or hand-crafted priors [143, 19, 144]. Furthermore, we do not rely on blind-spot approaches [2] or subsampling [142, 1] techniques that degrade image quality and limit denoising performance.

Our method is based on the representation learning technique proposed by Chen *et al.* [145], who revealed how a VAE [52] prefers to avoid using latent variables to describe structures that could instead be modelled locally by its decoder. For an autoregressive (AR) decoder such as PixelCNN [82, 146], these are any inter-pixel dependencies that lie within its receptive field. Consequently, latent variables will only represent information that the decoder cannot model, *i.e.*, dependencies that span beyond its receptive field.

We take advantage of this behaviour by designing a decoder that is capable of modelling exactly what we want to be excluded from our latent variables – the imaging noise – while being incapable of modelling the remainder of the data. Specifically, we use an AR decoder that can only model axis-aligned structures because its receptive field spans only a row or column of the image. This trains our model to exclude row- or column-correlated noise from the latent variables, while encouraging it to include the statistics of the underlying signal. We then propose a second network, termed *signal decoder*, that is trained to map these latent variables back into image space, thus producing denoised images. Following Lehtinen *et al*. [6], who showed that noisy images can be used as training targets for denoisers, we train the signal decoder alongside the VAE using the original noisy data as targets.

Using real microscopy data recorded with various imaging modalities, we demonstrate that our denoiser achieves state-of-the-art results compared to other unsupervised methods. Furthermore, we show that the method is not sensitive to the length of the AR decoder's receptive field, as long as it is above a minimum, and that samples from the trained noise model have similar autocorrelation and signal-dependence characteristics to real noise.

6.2 Related work

6.2.1 Self-supervised denoising

Most self-supervised denoisers use the blind-spot approach [31, 141, 25, 60, 147], where a network learns a denoising function when trained to predict the value of a pixel from surrounding pixels. Another technique trains the network to predict a subset of randomly sampled pixels from another [142]. For photon-counting data, Krull *et al*. [148] proposed removing photons and training a network to predict them with a Poisson distribution. These techniques all exploit a particular property of spatially uncorrelated noise, which is that the noise in one pixel cannot be predicted from the noise in other pixels.

To address spatially correlated noise, Structured Noise2Void (SN2V) [2] extended the blind-spot approach to also mask pixels containing noise that is correlated with the noise in the pixel being predicted. Lee *et al*. [1] also extended the blind-spot approach by subsampling pixels to break up noise structures, making noise effectively spatially uncorrelated and ready for traditional blind-spot denoising. Noise2Void2 (N2V2) [3] constrained a blind-spot network's ability to reconstruct high-frequency content.

There are also methods that add two independent noise samples to already-noisy images, training networks to denoise by mapping between these doubly-noisy pairs [59, 149]. While theoretically applicable to signal dependent, spatially correlated noise, practical implementation has not been demonstrated. The obstacle lies in obtaining a suitable noise model to sample from, particularly when signal dependent and spatially correlated noise must be described by a covariance matrix ([149]).

6.2.2 Diffusion-based denoising

In recent years, diffusion models have been proposed as priors for solving inverse problems, such as denoising, in a Bayesian way [150, 151, 152]. However, such methods require clean training data and a known mathematical forward model of the corruption process during inference, both of which are unavailable in many microscopy applications. Very recently, methods have been proposed to remove at least the requirement for clean training data by assuming simple additive Gaussian noise [153] or focusing on inpainting problems [154].

6.2.3 GAN-based denoising

Another approach to denoising trains a conditional Generative Adversarial Network (GAN) [155] to model the distribution of clean images given noisy images. This can be done using a dataset of unpaired noisy and clean images, provided that the generator is forced to output an estimate of the clean signal underlying its noisy input, and not a random clean image. Various methods have been proposed to ensure this, including minimising the perceptual difference between the generator's input and output [156], or maximising their mutual information [157]. While these methods have more relaxed requirements than supervised denoisers, obtaining a dataset of clean images can be a challenge in microscopy, even if it is unpaired. This is because the clean images must follow the same distribution as the signal content in the noisy training data. That means, *e.g.*, the same cell type, imaging conditions, *etc.*.

6.2.4 VAE-based denoising

These methods train a latent variable model of noisy data using a VAE and a pre-trained explicit noise model. Through the use of the noise model, the VAE is trained to represent clean signals with its latent variables. In DivNoising [26], the noise model assumes the noise is spatially uncorrelated, *i.e.*, noise is generated in each pixel independently. Later, HDN [4] was proposed as an extension to DivNoising that used a VAE with a hierarchy of latent variables [55]. It was found that short-range spatially correlated noise structures were modelled by only the bottom levels of the hierarchy and could be removed by preventing these latent variables from using information from the encoded input. This technique is known as HDN$_{3-6}$.

In Chapter 4, we proposed a method to remove spatially correlated but signal-independent noise by replacing the pixel-independent noise model in HDN with a CNN-based AR noise model [82]. This noise model was pre-trained using samples of pure noise which can be obtained by, *e.g.*, imaging without light. The method was therefore able to remove noise of any spatial correlation, but could not be applied to signal-dependent noise.

6.3 Background

Before we come to describe our method, we will first formalise the type of noise that we aim to remove, then give some background on VAEs for representation learning.

6.3.1 Row-correlated noise

A popular choice for describing noise distributions is the pixel-independent noise model as it can describe Gaussian, Poisson or combined Gaussian-Poisson noise (Sec. 2.2). However, many imaging systems can produce noise that does not conform to the pixel-independence assumption, but is instead correlated along rows or columns of pixels. For example, Scanning Transmission Electron Microscopy (STEM) [158] is prone to line artefacts caused by the slow reaction time of readout electronics [159], and Laser Scanning Confocal Microscopy (LSCM) [33] has been shown to produce artefacts in the scanning direction [35]. Depending on their settings, Electron-Multiplying Charge-Coupled Device (EMCCD) [160] cameras can produce horizontally correlated readout noise [161, 2]. The sCMOS [162] cameras that are popular in optical microscopy [83] have separate amplifiers for each column of pixels, leading to correlated noise within each column [163]. In addition to microscopy, the detectors used in infrared (IR) imaging systems, *e.g.* microbolometers, also commonly use separate column amplifiers and suffer from similar noise structures [164]. Examples of noisy images from these modalities along with plots of their spatial autocorrelation and signal-dependence can be found in Fig. 6.3.

We propose to describe such noise as

$$p(\mathbf{x}|\mathbf{s}) = \prod_{i=1}^{N}\prod_{j=1}^{M} p(x_{i,j}|x_{i,<j}, \mathbf{s}), \tag{6.1}$$

where $p(x_{i,j}|x_{i,<j}, \mathbf{s})$ is the distribution of possible noisy pixel values $x_{i,j}$ conditioned on the signal, as well as all "previous" values in the same row, i. While this model can describe interactions between pixels within the same row, pixels in different rows are (conditionally) independent (given a signal $\mathbf{s}$). Note that in this formulation, a pixel $x_{i,j}$ can depend on not only the signal in pixel (i, j), as with shot noise, but on the entire image for more complex interactions. In this work, we address the type of noise described in Eq. (6.1), which we believe is a good model for many real scientific imaging data.

6.3.2 VAEs and the division of labour

A latent variable model, with parameters θ, defines a probability distribution, $p_\theta(\mathbf{x}) \propto p_\theta(\mathbf{x}|\mathbf{z})p_\theta(\mathbf{z})$, over observed variables $\mathbf{x}$ via latent variables $\mathbf{z}$. This can be used to represent a data generation process in which a value $\mathbf{z}$ is first sampled from the prior, $p_\theta(\mathbf{z})$, and a value $\mathbf{x}$ is sampled from the conditional distribution $p_\theta(\mathbf{x}|\mathbf{z})$.

A VAE can be used to simultaneously optimise the model's parameters and approximate the posterior, $p_\theta(\mathbf{z}|\mathbf{x})$, via minimisation of an upper bound on the negative marginal log-likelihood,

$$\mathcal{L}(\theta, \phi; \mathbf{x}) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[-\log p_\theta(\mathbf{x}|\mathbf{z}))] + KL(q_\phi(\mathbf{z}|\mathbf{x}))||p_\theta(\mathbf{z})), \tag{6.2}$$

where the second term on the RHS is the KL divergence [54] from the prior to an approximate posterior, $q_\phi(\mathbf{z}|\mathbf{x})$. In Eq. (6.2), the first term is known as the reconstruction error, the second term as the regularisation error, $q_\phi(\mathbf{z}|\mathbf{x})$ as the encoder, and $p_\theta(\mathbf{x}|\mathbf{z})$ as the decoder.

This loss function can be rewritten as

$$\mathcal{L}(\theta, \phi; \mathbf{x}) = -\log p_\theta(\mathbf{x}) + \mathrm{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z}|\mathbf{x})). \tag{6.3}$$

This formation shows that the goal of a VAE is to simultaneously maximise the marginal log-likelihood of the data while minimising the divergence from the true posterior distribution of latent variables to the approximate posterior distribution of latent variables.

When designing a latent variable model for image data, the decoder can be made autoregressive, where the distribution of each pixel is conditioned on the value of all pixels above and directly to the left, as well as on $\mathbf{z}$,

$$p_\theta(\mathbf{x}|\mathbf{z}) = \prod_{i^*=1}^{N\times M} p_\theta(x_{i^*}|x_{<i^*}, \mathbf{z}), \tag{6.4}$$

where i^* indexes pixels in a row-major order. We refer to $x_{<i^*}$ as the full AR *receptive field*, and its shape is shown in Fig. 6.2b. This modification is intended to make the model more expressive. However, the decoder is now powerful enough to model the entire data distribution locally. That

is, it is possible to have a model of the data, $p_\theta(\mathbf{x}) = \int_\mathbf{z} p_\theta(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z}$, where $p_\theta(\mathbf{x}|\mathbf{z}) = p_\theta(\mathbf{x})$. When we use a VAE to train this model, it is therefore unclear which aspects of the data should be encoded in $\mathbf{z}$ and which should be modelled by the decoder.

When using a VAE in practice, there seems to be a preference to model as much content as possible with the decoder. He *et al*. [165] argued that this behaviour is caused by the approximate posterior lagging behind the true posterior in the early stages of training, causing the parameters to get stuck in a local optimum where the decoder models the data without using the latent variables.

Alternatively, Chen *et al*. [145] reasoned that with most practical VAEs, ignoring the latent variables leads a tighter lower bound on the marginal log-likelihood. By only using the latent variables to express what the decoder cannot model, the true posterior is brought closer to the prior and can be more closely matched by the relatively inflexible approximate posterior, reducing the second term of Eq. (6.3). The authors then demonstrated how this behaviour can be used to control the division of labour in a VAE. Specifically, they designed an AR decoder that is capable of modelling information that they do not want captured in the latent variables, but is incapable of modelling the information that they do want captured in the latent variables.

In this chapter, we design an AR receptive field that only captures the correlations common in scientific imaging noise. This allows us to train a latent variable model of noisy image data where latent variables explain the signal content and the decoder models only the noise generation process. We then use the approximate posterior to sample latent variables, each representing one of the clean signals that could possibly underlie a given noisy image.

6.4 Method

We propose a VAE-based unsupervised image denoiser for noise that is both signal-dependent and correlated along rows or columns of pixels. It is trained using only noisy images and does not require a pre-trained noise model. Moreover, it does not require pixel blind-spots or subsampling. Instead, we restrict the receptive field of the AR decoder in a Hierarchical VAE [55] to a row or

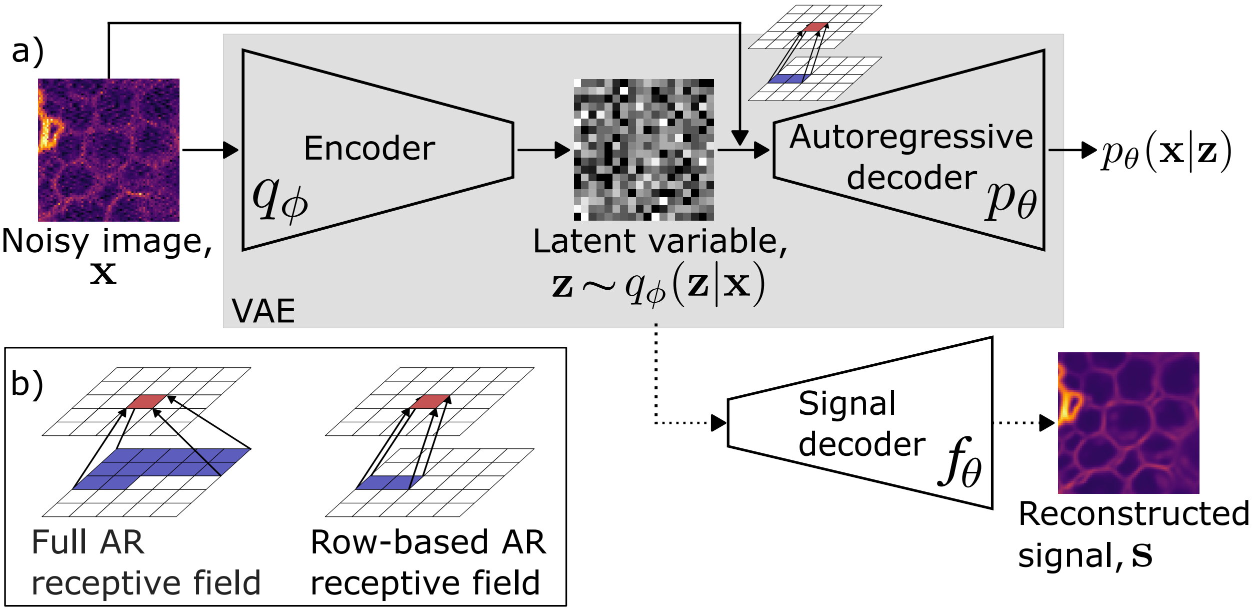

Figure 6.2: **(a)** A VAE [145] (solid arrows) is trained to model the distribution of noisy images $\mathbf{x}$. The autoregressive (AR) decoder models the noise component of the images, while the latent variable models only the clean signal component $\mathbf{s}$. In a second step (dashed arrows), our novel *signal decoder* is trained to map latent variables into image space, producing an estimate of the signal underlying $\mathbf{x}$. **(b)** To ensure that the decoder models only the imaging noise and the latent variables capture only the signal, we modify the AR decoder's receptive field. In a full AR receptive field (Eq. (6.4)), each output pixel (red) is a function of all input pixels located above and to the left (blue). In our decoder's row-based AR receptive field (Eq. (6.5)), each output pixel is a function of input pixels located in the same row, which corresponds to the row-correlated structure of imaging noise.

column of pixels, allowing the AR decoder to model the correlations of noise content but not the correlations of underlying signal. Following the insights of Chen *et al*. [145] (Sec. 6.3.2), the AR decoder will therefore learn to model noise, leaving the underlying signal to be encoded in the latent variable $\mathbf{z}$. We then propose a method for taking these latent variables and mapping them back into image space, obtaining estimates of clean signals. An outline of our method can be found in Fig. 6.2.

6.4.1 Autoregressive receptive fields for noise

Our AR decoder has a one-dimensional receptive field that is sufficient for modelling row- or column-correlated noise (Eq. (6.1)) as it occurs in microscopy (see Sec. 6.3.1 **Row-Correlated**

Noise) by spanning pixels in the same row or column as $x_{i,j}$. See Fig. 6.2b for a visual representation.

To remove row-correlated noise, the first step in our denoising process is to train a VAE to model noisy image data with the objective function in Eq. (6.2), where $p_\theta(\mathbf{x}|\mathbf{z})$ is factorised as,

$$p_\theta(\mathbf{x}|\mathbf{z}) = \prod_{i=1}^{N}\prod_{j=1}^{M} p_\theta(x_{i,j}|x_{i,<j}, \mathbf{z}). \tag{6.5}$$

To remove column-correlated noise, the factorization over pixels is perpendicular.

We find that this factorisation is insufficient for modelling signal content, which is highly correlated in all directions. Consequently, the VAE learns to encode signal content in its latent variables. The factorisation is however sufficient for modelling row-correlated noise content. Since a VAE's decoder will model everything that it can [145], noise will be modelled by the decoder. Our encoder, $q_\phi(\mathbf{z}|\mathbf{x})$, is now effectively a denoiser, mapping noisy images to a distribution over latent variables, where each latent variable corresponds to a possible clean signal.

An experimental investigation of the effects of changing receptive field size can be found in Sec. 6.5.2, and details on how we construct this receptive field with our AR decoder architecture can be found in the supplementary material.

6.4.2 Decoding the signal

Once our VAE has been trained, its latent space will represent clean signals, *i.e.*, each $\mathbf{z}$ contains all the information about an s. We would now like to use the model for denoising by inferring possible clean signals s for a given noisy image $\mathbf{x}$. Unfortunately, unlike previous methods [26, 4, 73], we cannot directly sample clean images from our encoder. Rather, samples from $q_\phi(\mathbf{z}|\mathbf{x})$ will directly *correspond* to a clean signal s. We denote the signal corresponding to a value of $\mathbf{z}$ as $\mathrm{s}(\mathbf{z})$. For an experimental validation of this deterministic relationship, please refer to the supplementary material. To obtain denoised images, we approximate $\mathrm{s}(\mathbf{z})$ with an additional regression network, termed *signal decoder*, $f_\theta(\mathbf{z}) \approx \mathrm{s}(\mathbf{z})$. In the following, we describe how $f_\theta(\mathbf{z})$ is trained despite not having access to training pairs $(\mathbf{z}, \mathbf{s}(\mathbf{z}))$.

In fact, training the signal decoder only requires pairs $(\mathbf{z}, \mathbf{x})$, where $\mathbf{x}$ is a noisy image and $\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})$. If we assume that the approximate posterior is accurate, such that $q_\phi(\mathbf{z}|\mathbf{x}) \approx p_\theta(\mathbf{z}|\mathbf{x})$, these pairs can be equivalently thought of as samples from the joint distribution, $(\mathbf{z}, \mathbf{x}) \sim p_\theta(\mathbf{z}, \mathbf{x})$. Least squares regression analysis tells us that optimising the signal decoder by minimising the squared L_2 norm of errors,

$$\mathcal{L}(\theta; \mathbf{x}) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} \left[||f_\theta(\mathbf{z}) - \mathbf{x}||_2^2 \right], \tag{6.6}$$

will train it to approximate $\mathbb{E}_{p_\theta(\mathbf{x}|\mathbf{z})}[\mathbf{x}]$ for any $\mathbf{z}$ [166]. By definition, $\mathbf{z}$ contains no more or less information about $\mathbf{x}$ than $\mathbf{s}(\mathbf{z})$, so the signal decoder will equivalently be approximating $\mathbb{E}_{p_\theta(\mathbf{x}|\mathbf{s}(\mathbf{z}))}[\mathbf{x}]$. Recalling that imaging noise is zero-centred (Sec. 6.3.1), the expected value of a noisy image given an underlying signal is *that signal*. Therefore, $f_\theta(\mathbf{z}) \approx \mathbb{E}_{p_\theta(\mathbf{x}|\mathbf{s}(\mathbf{z}))}[\mathbf{x}] = \mathbf{s}(\mathbf{z})$. This is similar to N2N [6], where the regressor for $\mathbf{s}$ is trained using noisy targets $\mathbf{x}$.

Even though the signal decoder would naturally be trained in a second stage after the main VAE is finished, in practice we co-trained it alongside the main VAE. At every training step, the sampled latent variable is fed to both decoders, but only the loss from the AR decoder is allowed to backpropagate to the encoder. This method of training is simply for convenience and we did not observe any changes in performance compared to a signal decoder that is trained separately, after the main VAE.

6.4.3 Inference

With the VAE and signal decoder trained, we can denoise an image $\mathbf{x}$ in a two-step process. We first sample a latent variable $\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})$, then obtain a clean image by decoding $\mathbf{s} = f_\theta(\mathbf{z})$. Similarly to [26, 4, 73], the result constitutes a random possible solution $\mathbf{s} \sim p(\mathbf{s}|\mathbf{x})$. To obtain a consensus solution, we follow [26, 4, 73] in averaging a large number of such samples. In the following experiments, all the results are the mean of 100 samples, both for our method and the baseline HDN. Please refer to the supplementary material for the inference time and PSNR of the mean of 1, 10 and 1,000 samples.

6.5 Experiments

Table 6.1: Comparison to baseline methods using PSNR, higher is better. The best results for methods that do not require paired images are printed in **bold**, and best results overall are underlined. HDN is a variant of HDN that is intended for spatially uncorrelated noise, but was applied to two datasets with spatially correlated noise by the original authors. These extra results are presented in brackets. Note that the *Mouse Nuclei* dataset contains spatially uncorrelated noise and that HDN failed to train on the *Mouse Actin* dataset. CARE and N2N are supervised denoisers requiring paired images, unlike the other baselines.

	EMCCD				LSCM	
	Conv. A	**Conv. B**	**Mouse Actin**	**Mouse Nuclei***	**Actin Conf.**	**Mito Conf.**
AP-BSN	22.30	24.10	29.55	35.41	26.89	24.94
SN2V	30.29	31.67	32.80	36.62	23.30	26.41
N2V2	29.36	36.72	34.23	35.88	26.71	25.89
HDN$_{3-6}$ (HDN)	31.41	37.21 (37.39)	– (34.12)	(36.87)	26.84	26.51
HDN$_{3-6}$ *large* (HDN *large*)	31.80	37.92	34.14	(38.12)	27.17	26.15
Ours *small*	34.42	38.99	36.50	39.56	27.35	27.49
Ours *large*	**37.49**	**44.10**	**39.23**	**42.98**	**27.41**	**27.50**
CARE	31.56	36.71	34.20	36.58	29.44	27.55
N2N	28.22	36.85	34.60	37.33	–	–

	Simulated	
	FFHQ Stripe	**FFHQ Checkerb.**
AP-BSN	28.22	13.19
SN2V	29.52	19.11
N2V2	33.08	29.22
HDN$_{3-6}$ (HDN)	32.64	29.61
HDN$_{3-6}$ *large* (HDN *large*)	34.54	25.51
Ours *small*	32.87	33.51
Ours *large*	**35.66**	**36.27**
CARE	36.46	36.89
N2N	36.18	36.43

6.5.1 Benchmarking denoising performance

Datasets

We tested the performance of our proposed denoiser on real noisy images captured by five different imaging modalities that commonly suffer from row-correlated noise. Two modalities have noisy

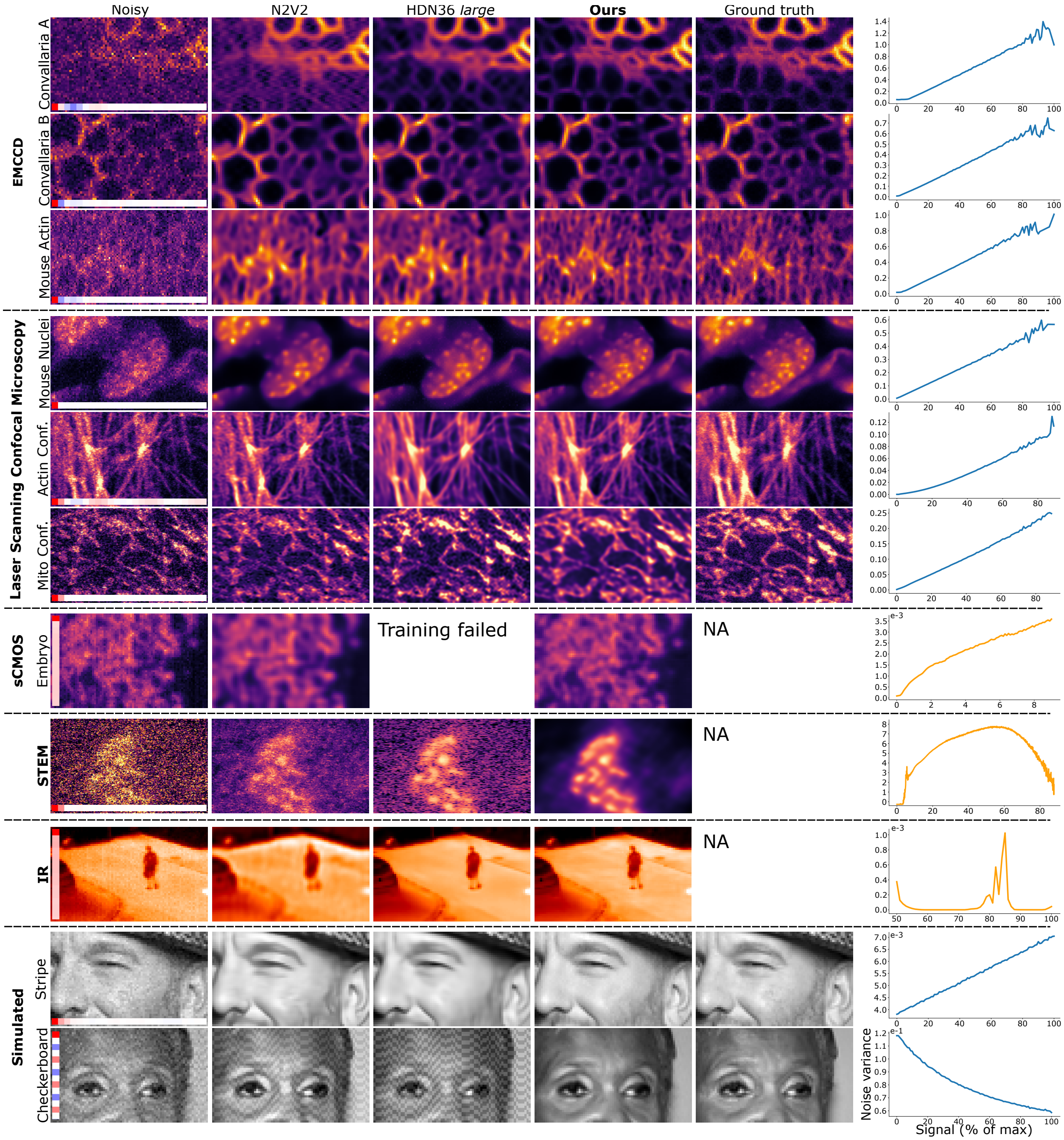

Figure 6.3: Visual results from our method and two unpaired baselines on all datasets. The spatial autocorrelation of the noise is overlaid on each noisy image, with red indicating a positive correlation and blue indicating a negative correlation. The direction of the correlation is given by the orientation of the autocorrelation bar. Additionally, the signal dependence of the noise in each dataset is shown in the right-hand column. The horizontal axis of these graphs is the clean signal intensity as a percentage of the maximum, while the vertical axis is the variance of noisy pixel values recorded for these signal intensities. Ground truth must be used to calculate signal dependence, so orange lines are used where denoised images from our method are used as pseudo-ground truth.

image datasets with known ground truth for quantitative evaluation. The method for obtaining ground truth can be found in the source publications. The first is the EMCCD sensor, for which we have three spinning disk confocal fluorescence microscopy datasets: *Convallaria A* [2], *Convallaria B* [50] and *Mouse Actin* [50]. The second is LSCM, for which we have three fluorescence datasets: *Mouse Nuclei* [50], *Actin Confocal* [167] and *Mito Confocal* [167]. Note that *Mouse Nuclei* contains spatially uncorrelated noise, but was included to demonstrate that the proposed method is still applicable to spatially uncorrelated noise without modification. For details on the size of the datasets and the train/test splits, see original publications. The remaining modalities have only noisy images. These are the sCMOS sensor for which we have one fluorescence dataset: *Embryo* [168], the microbolometer with one infrared imaging dataset: *IR* [169], and a STEM dataset: *STEM* [170].

In addition to real data, we created two datasets by corrupting the Flickr Faces HQ thumbnails dataset [171] with simulated noise. For *FFHQ - Stripes*, the images were corrupted by a combination of additive white Gaussian noise, Poisson shot noise, and additive white Gaussian noise that had undergone a horizontal Gaussian blur. For *FFHQ - Checkerboard*, the images were corrupted by a combination of a vertical checkerboard pattern and Gaussian noise with an inverse signal dependence. Details on the simulated noises are in the supplementary materials.

Baselines and architecture

We compare our method with other deep learning-based denoisers that can be applied to spatially correlated, signal-dependent noise and do not require paired images. These are the self-supervised denoisers SN2V [2], N2V2 [3] and Asymmetric PD BSN (AP-BSN) [1], and the unsupervised denoiser HDN$_{3-6}$ [4]. We use these as baseline methods along with CARE [5], which is trained using pairs of noisy and clean images, and N2N [6], which is trained using two noisy acquisitions of the same signal, on datasets with suitable training images available. For details on how each baseline was implemented, refer to the supplementary material.

Of the denoisers not requiring paired images, the performance of HDN$_{3-6}$ is the best, but the model implemented in the baseline's publication uses significantly fewer parameters than ours (7 million to 25 million). We therefore evaluate an additional version, termed HDN$_{3-6}$ *large*, with a similar number of parameters made by increasing the number of latent dimensions from 32 to 64.

It should be noted that the *Convallaria B* and *Mouse Actin* datasets had been treated as spatially uncorrelated by Prakash *et al*. [4] when HDN was tested. We also report those results in Sec. 6.5. It should also be noted that the noise in the *Mouse Nuclei* dataset is spatially uncorrelated, so was denoised by HDN and its higher parameter version, HDN *large*, which was made the same way as HDN$_{3-6}$ *large*.

Our method requires a choice of orientation and size for the AR decoder's receptive field. Orientation was determined by examining the spatial autocorrelation of noise samples, with these plots reported in Fig. 6.3, and following the ablation study in Sec. 6.5.2, we always used a receptive field length of 40 pixels.

As for our model's encoder, latent variables are produced by a Hierarchical VAE [55] with 14 levels to its hierarchy. In addition to the full-sized version, we evaluate a smaller version that requires approximately 6GB (as opposed to 20GB) of GPU memory to train. This was achieved by reducing the number of latent variables from 14 to 6 and reducing the number of latent dimensions from 64 to 32. We refer to these models as Ours *small* and Ours *large*, for the lower memory and higher memory versions respectively. See the supplementary materials for full architecture and training details.

Discussion

Quantitative results measuring the PSNR in dB are reported in Sec. 6.5, and qualitative results are reported in Fig. 6.3. Note that HDN$_{3-6}$ failed to train with the *Mouse Actin* and the *Embryo* dataset. Out of the methods that do not require paired images, Ours *large* achieved the highest PSNR across all datasets, even beating the supervised CARE and N2N on four of six microscopy datasets. Ours *small* then had the second highest PSNR for an unpaired method on all datasets except *FFHQ - Stripe* .

Turning to the qualitative results in Fig. 6.3, we see that Ours *large* denoised images from each dataset without leaving behind any artefacts, whereas HDN *large* could not remove the correlated component of the noise from the *STEM* or the *FFHQ - Checkerboard* dataset. SN2V left artefacts in *Convallaria A* and *FFHQ - Checkerboard*. It is unclear if the high frequency features that are visible in the *Mito Confocal* output of HDN *large* and N2V2, but not in our output, are true signal or artefacts of remaining noise, as the ground truth also contains a low level of noise.

6.5.2 Ablation study - receptive field size

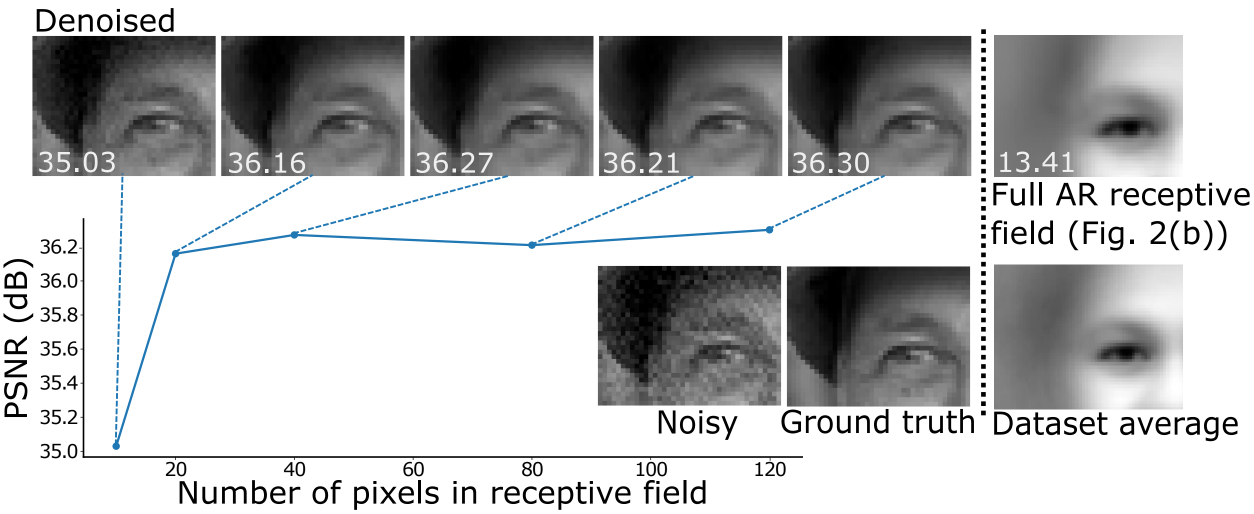

Figure 6.4: The *FFHQ - Checkerboard* dataset wass denoised 5 times by varying the number of pixels covered by the AR decoder's receptive field. The images show denoising results for each receptive field size with PSNR overlaid. Additionally, an image denoised using a full AR receptive field is included on the right. In this situation, the signal decoder is given completely uninformative inputs and learns to output the mean of the training dataset.

As stated in Sec. 6.4, to model the noise correlations addressed in this chapter, the receptive field of a VAE's AR decoder must span pixels in the same row or column as the pixel being predicted. In Fig. 6.4, the effect of receptive field size (number of pixels) is investigated by denoising the *FFHQ - Checkerboard* dataset using a range of receptive field lengths, from 10 to 120 pixels, and measuring the effect on PSNR. We also include the effect of training a model with a full AR receptive field, as shown in Fig. 6.2b.

This study shows that the AR decoder is able to model this spatially correlated noise, and therefore have it removed by the encoder, when the receptive field spans 40 pixels. Moreover, the AR decoder does not seem to model more of the signal as the receptive field grows. If it did, we would expect a steady drop in PSNR as denoised images lose signal content. However, as the image on the right of Fig. 6.4 shows, the signal will be modelled by an AR decoder with a full receptive field. Latent variables in this situation carry no information about $\mathbf{x}$, causing the signal decoder to minimize Eq. (6.6) by predicting the mean of the training set. This is expected, as when the latent variable $\mathbf{z}$ has no information about $\mathbf{x}$; $\mathbb{E}_{p_\theta(\mathbf{x}|\mathbf{z})}[\mathbf{x}] = \mathbb{E}_{p_\theta(\mathbf{x})}[\mathbf{x}]$.

6.5.3 Ablation study - noise reconstruction

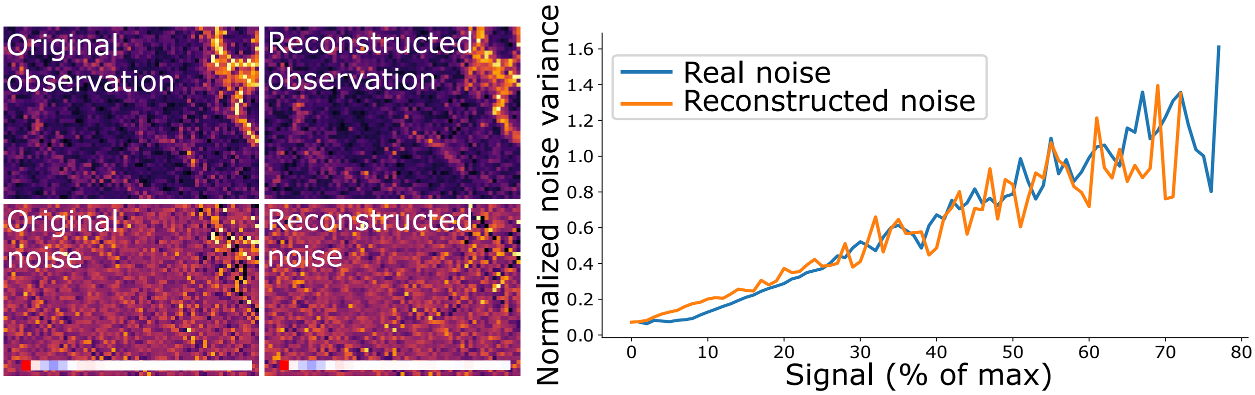

Figure 6.5: A noisy image from the *Convallaria A* dataset was encoded and decoded to produce a reconstructed observation and an artificial noise sample. The sampled noise has spatial autocorrelation and signal dependence that match those of the real noise, indicating that the AR decoder has learnt an accurate noise model.

If the VAE's AR decoder is modelling only the noise component of images, encoding an image with the VAE and sampling from the AR decoder should yield an image with the same underlying signal but a different random sample of noise. If the decoder's model of the noise is accurate, the sampled noise should exhibit similar autocorrelation and signal-dependence characteristics as the noise in the original image. An investigation into this is reported in Fig. 6.5, where a noisy image from the *Convallaria A* dataset was encoded by a trained VAE, then a latent variable was sampled and used by the VAE's AR decoder to sample a reconstruction of the original image.

The reconstructed noise exhibits spatial autocorrelation and signal dependence very similar to the real noise, indicating that our AR decoder has learnt an accurate model of the noise.

6.6 Conclusion

We proposed an unsupervised VAE-based denoising algorithm for signal-dependent noise that is correlated along rows or columns of pixels. It is trained without any clean data or a noise model. By engineering the receptive field of the AR decoder, the VAE's latent variables are encouraged to represent the signal content of an image while discarding the noise. We then presented a novel signal decoder that is trained to map this latent variable to an estimate of the clean image. The algorithm outperforms existing self- and unsupervised denoisers.

Our method is suited to noise with correlations that run parallel to the image axes. This commonly occurs in microscopy and lab users often cannot find suitable methods to remove it. We release our code open source and strongly believe that the scientific imaging community will apply and adapt our methods in a variety of applications.

While we have achieved our results using 1-dimensional receptive fields, some imaging modalities produce noise that is correlated in multiple directions, therefore requiring a differently shaped receptive field. We found that extending the receptive field to cover both a row and a column of pixels allows the AR decoder to model some aspects of the signal, making the technique unsuitable for removing noise correlated in two dimensions. However, we believe that techniques other than shaping the AR decoder's receptive field could be discovered to limit it's modelling capabilities. Furthermore, the method is limited to zero-centred noise by the signal decoder. This precludes, *e.g.*, removing salt-and-pepper noise or restoring images with saturated pixel intensities. A direction for future work could therefore be alternative methods for mapping latent variables back into image space. We hope that future work will further improve the theoretical understanding of the method and allow us to utilise its full potential.

7 DUAL ARM DETECTION FOR UNSUPERVISED DENOISING IN OPTICAL TIME SERIES MEASUREMENT

In the previous chapter, we proposed a method for training an unsupervised denoiser to remove **structured and signal-dependent noise** in a range of microscopy modalities. We utilised the fact that the structures present in noise are often distinct from the structures in the underlying signal. Unfortunately, the distinction is less clear for the structures in flow cytometry noise. Here, both signal and noise are correlated along the time axis – the only axis. To overcome this issue, we introduce a new axis, one along which signal is perfectly correlated but noise is perfectly uncorrelated. This requires inserting a beam splitter into the acquisition setup, effectively recording noisy-noisy pairs. The noise model co-training principal behind the denoiser in the previous chapter can then be applied to this data. The data analysed in this chapter was collected by Sadao Ota and Yuichiro Iwamoto from the University of Tokyo.

7.1 Introduction

Flow cytometry is a powerful tool for quantifying cellular and molecular characteristics. It has been shown to be effective in the diagnosis of disorders and diseases including immunodeficiencies [172], cancer [173, 89] and sepsis [174]. It works by suspending particles in a solution and flowing them single-file through a laser beam. Fluorescent probes attached to the particles are excited by the laser, causing them to emit photons that are subsequently amplified and detected by photomultiplier tubes. In addition to fluorescence signals, forward- and side-scattered light can be collected to provide additional structural and morphological information [175]. The result is a time series of light intensity measurements that describe the content of biological fluids [176].

One application that flow cytometry has great potential in is the detection of rare populations, enabling early disease diagnosis [177]. However, the sensitivity required for this task is limited by noise. Two noise sources dominate: dark current and shot noise [178]. Dark current can be misinterpreted as particle detections, while shot noise distorts the characteristics of a detection. For the weakest signals, shot noise can even cause no photons to be detected. Exacerbating the problem, noise is made to be correlated in time by the bandwidth limited detector, and temporally correlated additive read-out noise inhibits deconvolution of the impulse response.

Existing simple denoisers, such as Gaussian filters, are limited in their effectiveness and distort the resulting signal, and supervised deep learning denoisers are not applicable when no clean ground truth data is available. In Chapter 5, we introduced a VAE-based approach for unsupervised denoising. However, we were only able to denoise the scattering channel. This is because it is mainly affected by signal independent read-out noise, allowing us to collect calibration data to train a noise model.

In this chapter, we present an unsupervised deep learning-based method for estimating the signal underlying noisy measurements, enabling accuracy at the single photon level by removing signal-dependent Poisson shot noise and dark current. Our method requires no additional training or calibration data beyond the data that is to be denoised. Instead, we use a beam splitter setup to collect paired observations of the same underlying signal, each corrupted by statistically independent noise. The statistical characteristics of these observations allow us to apply a modified version of a denoiser that was recently proposed for microscopy images [140].

Furthermore, we improve the trustworthiness of denoised outputs by directly estimating aleatoric uncertainty. This method works by training an additional deep neural network to predict credible intervals over the range of values that the true denoised signal could exist in.

Experiments show that the performance of our method is close to that of a supervised approach [5] and exceeds an existing self-supervised approach [6]. The former represents the gold standard of performance in the impractical scenario that clean, ground-truth training data is available. The latter represents what is currently possible when using the same limited training data as us.

In summary, our contributions are:

- Combining a beam splitter acquisition setup with an unsupervised deep learning denoiser to remove signal-dependent and temporally correlated noise in an optical time series.

- Direct uncertainty estimation by predicting credible intervals over possible values of the denoised signal.

7.2 Background

Our method is built upon the deep learning denoiser proposed in the previous chapter, Correlated and Signal-Dependent Denoising (COSDD), which removes spatially correlated noise from microscopy images using only noisy training data. Here, we will give a brief recap of the method. COSDD trains a Hierarchical VAE [52, 55] to model noisy images, $\mathbf{x}$, but ensures that signal is modelled by the VAE's prior distribution, $p_\theta(\mathbf{z})$, over latent variables $\mathbf{z}$, and noise is modelled by the VAE's decoder, $p_\theta(\mathbf{x}|\mathbf{z})$. As a result, the encoder, $q_\phi(\mathbf{z}|\mathbf{x})$, acts as a denoiser, mapping noisy images to a distribution of latent variables that represent possible underlying signals. The loss function used to optimise parameters θ and ϕ is the ELBO [52],

$$\mathcal{L}(\theta, \phi; \mathbf{x}) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[-\log p_\theta(\mathbf{x}|\mathbf{z}))] + KL(q_\phi(\mathbf{z}|\mathbf{x}))\|p_\theta(\mathbf{z})), \tag{7.1}$$

where KL is the KL divergence.

This division of labour between the encoder and decoder is achieved by exploiting differences in the spatial correlation of signal and noise. In microscopy images, noise is often correlated along rows of pixels, or not at all, while signal is correlated in all directions. To take advantage of these characteristics, the VAE is equipped with an autoregressive (AR) decoder. In an AR decoder, the distribution of each pixel is conditioned on the value of previously generated pixels, as well as on $\mathbf{z}$. We refer to the elements of $\mathbf{x}$ that a pixel is conditioned on as the decoder's *receptive field*. In COSDD, the receptive field is restricted to a row of pixels. This gives the decoder the power to model noise but not signal. If a VAE's decoder has the power to model something, it will [145], so a row-based receptive field forces the decoder to model noise. Consequently, noise will not be represented by the latent variables. Signal, on the other hand, cannot be modelled by the decoder because its structures span outside the receptive field. This forces the VAE to use its latent variables to represent signal content. Once the VAE is trained, the encoder will randomly sample latent variables representing the clean signals that could underlie a given observation.

By distinguishing noise from signal via differences in their spatial correlation, the unsupervised learning-based COSDD performs comparably to supervised learning-based denoisers. However, in flow cytometry data, signal and noise cannot be distinguished so easily, as both can be correlated in time. To apply the principle behind COSDD, we introduce a new axis to the data, along which signal is perfectly correlated and noise is completely independent.

7.3 Method

7.3.1 Beam splitter

To collect suitable data for our denoiser, we use a flow cytometer with a beam splitter setup. This creates two observations with identical underlying signal and statistically independent noise. In our setup, the photons emitted by a fluorophore as it passes through the laser are sent through a beam splitter and redirected to two PMTs. The noise sources, shot noise and dark current, occur independently at each PMT. The bandwidth limited detectors then convolve the signal with an impulse response, making the noise correlated in time. They also add temporally correlated read-out noise, making deconvolution with the impulse response difficult.

Assuming equal rates of dark current, the mean signal at each PMT will be the same, and comprise of the quantity of interest – the rate of photons emitted by the fluorophore – scaled by the gain and summed with the rate of dark current. This quantity, as a function of time, will then be convolved with the impulse response, giving the mean of the observations which we refer to as our signal, $\mathbf{s}$. The paired observations are concatenated along the channel axis, giving us the final observation $\mathbf{x} \in \mathbb{R}^{C \times T}$, where $C = 2$ is the number of channels and T is the total time.

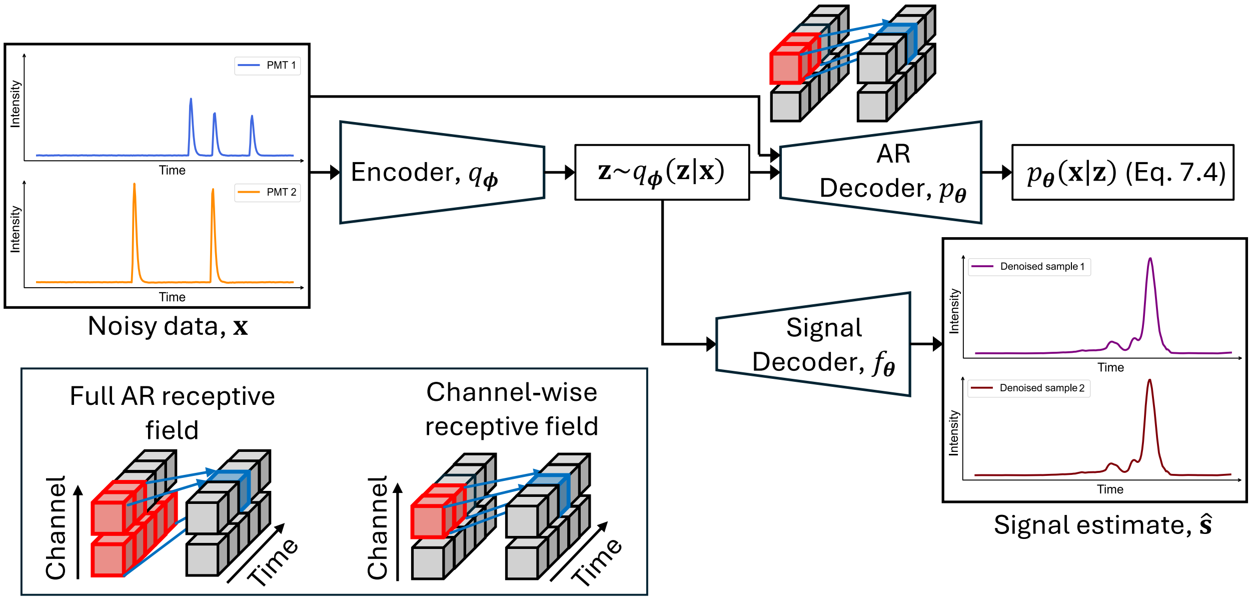
PMT 1
Intensity
Time
PMT 2
Intensity
Time
Noisy data, $\mathbf{x}$
Encoder, $q_{\boldsymbol{\phi}}$
$\mathbf{z}\sim q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$
AR Decoder, $p_{\boldsymbol{\theta}}$
$p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$ (Eq. 7.4)
Signal Decoder, $f_{\boldsymbol{\theta}}$
Denoised sample 1
Intensity
Time
Denoised sample 2
Intensity
Time
Signal estimate, $\hat{\mathbf{s}}$
Full AR receptive field
Channel
Time
Channel-wise receptive field
Channel
Time

Figure 7.1: **The architecture of our denoiser.** A pair of observations, $\mathbf{x}$, each containing the same underlying signal, are fed into a Hierarchical VAE. The VAE's decoder is autoregressive, and predicts each element of $\mathbf{x}$ by conditioning on previous elements of $\mathbf{x}$ in the same channel. Because it cannot see across channels, it cannot model signal, only noise. The latent variables therefore describe signal [145]. These latent variables are decoded back into time series by the signal decoder, which is trained using Eq. (6.6).

7.3.2 VAE denoising

We propose the following factorisation for an autoregressive decoder,

$$p_\theta(\mathbf{x}|\mathbf{z}) = \prod_{c=1}^{C}\prod_{t=1}^{T} p_\theta(x_{c,t}|x_{c,<t}, \mathbf{z}), \tag{7.2}$$

where channels are conditionally independent given the latent code, but can be temporally correlated. By conditioning the distribution of each element of $\mathbf{x}$ on previous values from the same channel, *i.e.*, from the same PMT, the decoder can perfectly model the noise content in $\mathbf{x}$. However, the signal content, *i.e.*, the timing and brightness of particles, must match between channels. Anything generated by the decoder will be generated differently in each channel, so it cannot effectively model signal. If we insert Eq. (7.2) into the ELBO loss (7.1), the only way for the VAE to effectively model data is to encode signal with its latent variables $\mathbf{z}$. The encoder,

$q_\phi(\mathbf{z}|\mathbf{x})$, therefore becomes a denoiser. The receptive field corresponding to Eq. (7.2) and the VAE are shown in Fig. 7.1.

We use Eq. (6.6) to train a signal decoder to map latent variables back into time series space. The target for the signal decoder is the concatenated observations $\mathbf{x}$, so it will predict the signal, $\mathbf{s} \in \mathbb{R}^{C \times T}$, underlying each of the two channels.

7.3.3 Direct denoising and uncertainty estimation

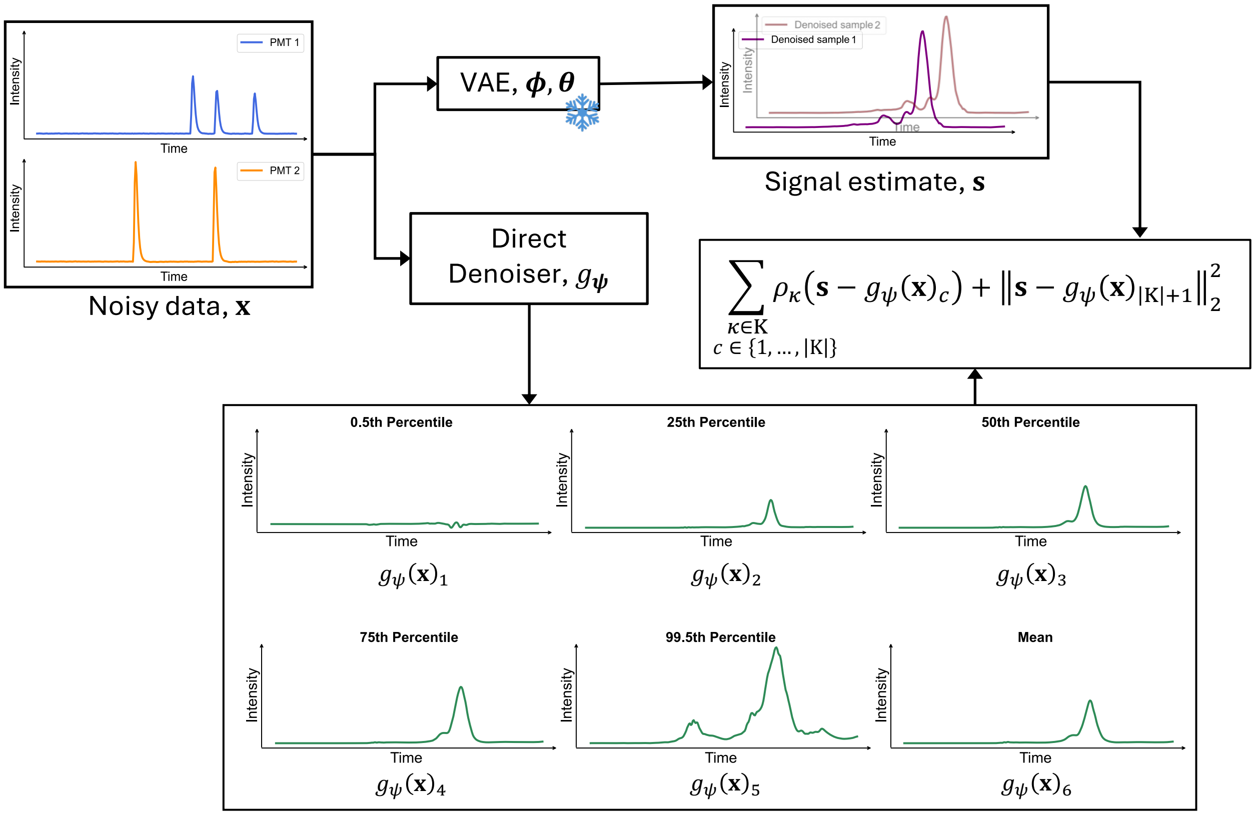

Figure 7.2: **The process of percentile estimation with a Direct Denoiser.** A pair of observations, $\mathbf{x}$, are fed into a pre-trained VAE and a training Direct Denoiser. The VAE randomly samples a possible clean signal estimate. This is the target for the Direct Denoiser. The Direct Denoiser returns $|K| + 1 = 6$ outputs. The first $|K| = 5$ are trained to minimise into the quantile regression loss (Eq. (7.3)) for each quantile in K. The $|K| + 1 = 6^{\text{th}}$ output is trained to minimise the mean squared error loss. Only the Direct Denoiser's parameters are updated to minimise these losses.

For a given noisy observation, a VAE-based denoiser [26, 4, 73, 140] randomly samples possible denoised signals. The variation of many samples can be inspected to assess the aleatoric uncertainty of the denoising problem. For a single consensus estimate, users will often take the mean of many samples.

Repeated sampling of denoised outputs for a large dataset takes a long time. A solution to this problem was presented in Chapter 3, where a second deep neural network regresses observations to denoised signals output by a VAE. This network is known as the *Direct Denoiser*, g_ψ. The choice of regression loss dictates what the Direct Denoiser will learn to predict. For example, the squared L_2 norm of errors will train it to predict the mean of all possible denoised signals, while the L1 norm will train it to predict the element-wise median of all possible signals.

Instead of settling for one prediction, we train a Direct Denoiser with multiple output channels, each minimising a different loss function. One is trained to minimise the squared L_2 norm of errors. The others are trained using the quantile regression loss function [179],

$$\rho_\kappa(\mathbf{s} - g_\psi(\mathbf{x})) = (\mathbf{s} - g_\psi(\mathbf{x})_c)(\kappa - \mathbb{1}_{\mathbf{s}-g_\psi(\mathbf{x})_c<0}), \tag{7.3}$$

where $\mathbb{1}$ is the indicator function, $0 < \kappa < 1$ is a choice of quantile and $g_\psi(\mathbf{x})_c$ is the cth channel of the prediction $g_\psi(\mathbf{x})$.

We choose the set of quantiles $K = \{0.005, 0.25, 0.5, 0.75, 0.995\}$. These allow us to plot the 99% and the 50% credible intervals over possible denoised signals, and predict the element-wise median signal. Users are free to select quantiles for a desired credible interval.

With $q_{\phi,\theta}(\mathbf{s}|\mathbf{x})$ as the distribution of denoised signals learned by the VAE (Sec. 3.3), the loss for our Direct Denoiser is

$$\mathcal{L}(\psi; \mathbf{x}) = \mathbb{E}_{q_{\phi,\theta}(\mathbf{s}|\mathbf{x})} \sum_{\substack{\kappa\in K \\ c\in\{1,\ldots,|K|\}}} \rho_\kappa(\mathbf{s} - g_\psi(\mathbf{x})_c) + \|\mathbf{s} - g_\psi(\mathbf{x})_{|K|+1}\|_2^2. \tag{7.4}$$

A graphical representation of this training process is shown in Fig. 7.2.

All results in Sec. 7.4 are computed using outputs of our Direct Denoiser. As our denoiser separately estimates the signal underlying each channel, we sum over the channels for a final denoised result.

7.4 Experiments

7.4.1 Architecture, datasets and baselines

The CNN architectures of our Hierarchical VAE and signal decoder are the same as in Chapter 6, but we use 6 levels in our VAE hierarchy and have one-dimensional convolutions instead of two. Our Direct Denoiser uses the same CNN as the Hierarchical VAE, but converted to be deterministic. This was achieved by replacing the random samples at each level of the approximate posterior with the distribution's mean prediction.

We evaluated our denoiser using two real flow cytometry datasets, *Data 1* and *Data 2*. They consist of two low SNR fluorescence channels, which are our noisy observations, and one high SNR fluorescence channel, which is our ground truth. All channels contain the same underlying signal, except the signal in the ground truth is stronger. We also collected a high SNR side-scatter channel. All particles in our data scatter light and emit fluorescence.

These datasets were made using a double beam splitter setup. After being emitted from the particle, light is passed through a 90/10 beam splitter, sending 90% of the signal to one PMT and creating our ground truth. The remaining 10% passes through a 50/50 beam splitter. For Data 1, this attenuated light becomes our paired noisy observations. For Data 2, the light ultimately passes through a neutral density filter with an optical density of 0.2 before becoming our paired observations. Both datasets are of 110-nanometre polystyrene nanoparticles, but Data 1 used a 250mW excitation power and Data 2 used a 50mW excitation power. The final 20% of each dataset was used for testing.

Our denoiser was compared to two learning-based baselines: a supervised learning denoiser [5] (*Supervised*) and the self-supervised learning denoiser *N2N* [6]. Both used the same UNet architecture as our Direct Denoiser and were also trained using the same multichannel loss function (Eq. (7.4)). The input for *Supervised* was both low SNR fluorescence channels, and its target was the high SNR fluorescence channel. The input for *N2N* was one of the low SNR channels and the target was the other. As *Supervised* is trained using ground-truth targets, it was not shown the test set during training.

Lastly, we compared our denoiser to a Gaussian filter with a standard deviation of 4 and the geometric mean of the two noisy channels. The intuition behind using the geometric mean is that, by multiplying channels together, only time points that are non-zero in both channels will remain, revealing simultaneous photons that could indicate the presence of a particle. Because of read-out noise, the intensity is never truly zero, so we first clamp the baseline of the observations to zero before multiplying them.

7.4.2 Evaluation

Table 7.1: Quantitative performance using mAP↑ and Range-Invariant Peak Signal-to-Noise Ratio (RI-PSNR)↑. The best method that does not require high SNR training data is shown in bold.

		Ours	Supervised	N2N	Gaussian filter	Geometric mean	Raw
mAP	Data 1	**0.8220**	0.8211	0.8218	0.7993	0.0253	0.8053
	Data 2	**0.802**	0.8055	0.7773	0.7394	0.0074	0.7318
RI-PSNR	Data 1	**18.86**	18.94	18.65	18.26	17.11	17.39
	Data 2	**18.74**	19.03	18.23	17.27	15.30	15.83

A good denoiser for flow cytometry should recover the signal of all particles and not reveal false positives. To assess how well particles are recovered, we apply a peak detection algorithm to our scattering data, revealing the true location of all particles. The same peak detection algorithm was applied to denoised data, varying the threshold height for a detection from the minimum of the data to the maximum. Following Iwamoto *et al*. [89], we do peak detection on 50th percentile predictions. From this we obtained true positive, false positive and false negative detections at varying thresholds and calculated the mAP of each denoiser. Results for this experiment can be found in upper Tab. 7.1. Our denoiser beats *N2N* on both datasets, and even beats *Supervised* on Data 2. We suspect that our denoiser is at an advantage as, being unsupervised, it can utilise

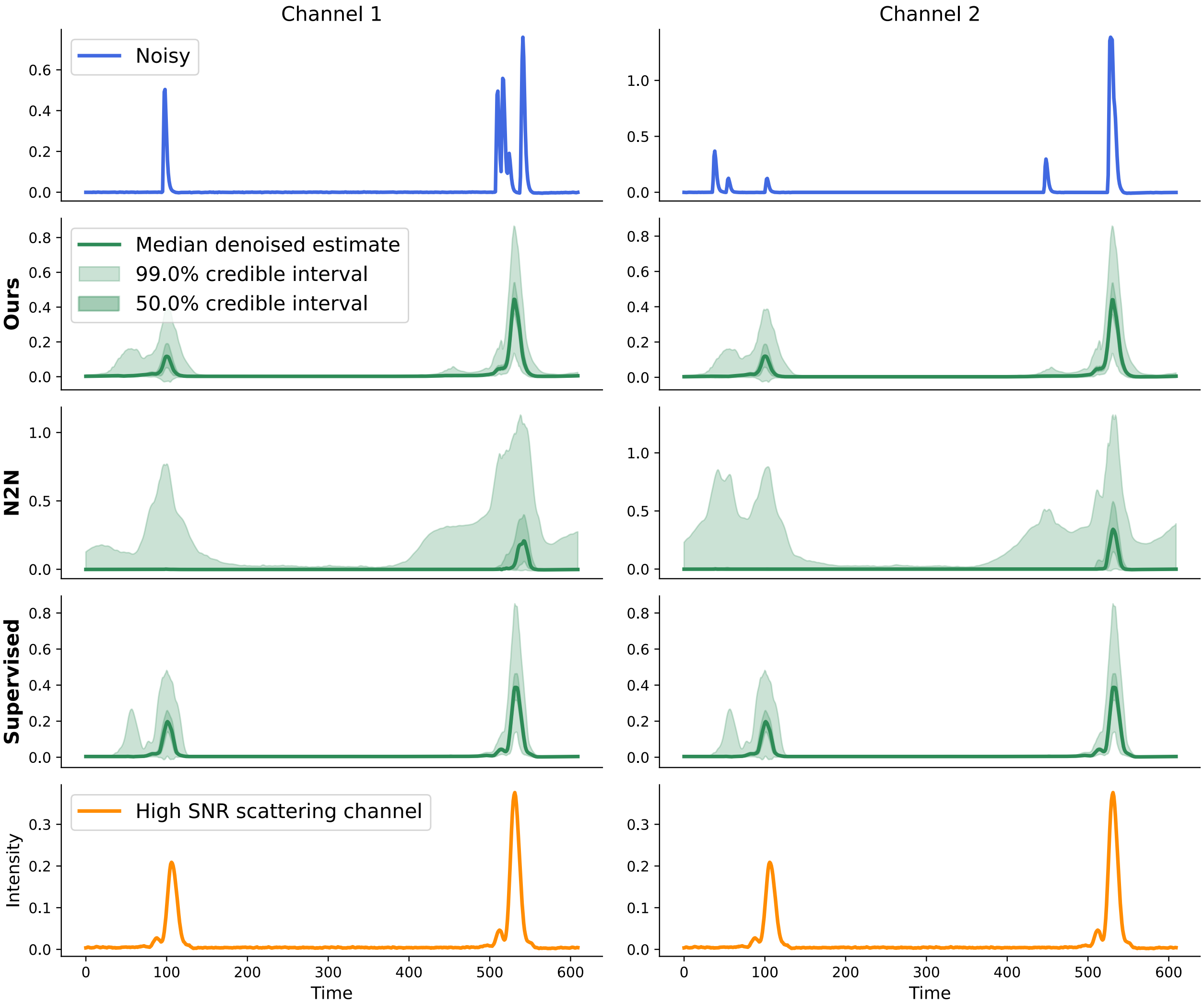

Figure 7.3: **Visual denoising results of Data 1.** In the top row are paired observations of the same underlying signal. The true location of particles is shown in the side-scattering channel on the bottom row. Middle rows are denoised outputs. We show the predicted median of denoised signals and the 99% and 50% credible intervals. These are the regions where the model is 99% and 50% certain that the true signal lies. Our denoiser and *Supervised* confidently recover both particles, but *N2N* misses the first.

the test data during training.

To assess the accuracy of recovered particle signals, we use the CAREamics library [180] to compute the Range-Invariant Peak Signal-to-Noise Ratio (RI-PSNR) from the ground truth to denoised signals. The predictions from the squared L_2 norm channel were used for this test. Following Iwamoto *et al*. [89], we crop a window of length 200 around each particle detection in the ground truth, and crop the same windows in the denoised time series. The RI-PSNR of each window is averaged to produce the final result. Results can be found in lower Tab. 7.1. Our denoiser again beats *N2N* and the non-deep learning baselines.

Lastly, we show denoised examples in Fig. 3.3. The first row of this plot has the two noisy observations. Subsequent rows show the output from our denoiser, *N2N* and *Supervised*. Each denoiser's estimate of the median and their predicted 99% and 50% credible intervals are shown. *N2N* is trained to target noisy data, so believes the true location of the signal could lie in a much larger region than ours, which instead predicts credible intervals over the true denoised signal. In the bottom row is the high SNR scattering channel, which reveals two particle detections. The first particle has been confidently revealed by our denoiser, but *N2N* only assigns a small probability to a detection.

7.5 Discussion

We paired an optical time series acquisition setup with an unsupervised deep learning denoiser to enable nanoparticle detection at extremely low signal intensities. The paired data we collect could be used to train an existing denoiser, N2N, but we demonstrate that ours recovers particle detections with greater accuracy. The reason for this is as follows. A practitioner will always use as much light as they can without causing enough photodamage to invalidate their observations. The amount of light that can be used is commonly known as a photon budget. For unpaired observations, the photon budget can be completely spent on each acquisition. For paired observations, the photon budget must be split between acquisitions. Each element of the paired observations will therefore have a greater noise intensity than the unpaired observation, while their sum will be equal to the unpaired observation (minus any inefficiencies).

N2N can only operate on each element of a pair individually, and must therefore contend with elevated noise levels. Our method can operate on both elements of the pair together, effectively halving the intensity of the noise that it must remove.

We believe that the denoiser presented here, and further developments of it, could be become a preferred method for utilising paired noisy-noisy observations.

8 CONCLUSION

8.1 Summary

This thesis identified some of the key obstacles that prevent routine application of deep learning-based denoisers to microscopy and flow cytometry data. Two of the foremost challenges were limited access to training data and structured noise. For microscopy and flow cytometry, these problems were interconnected. Structured noise could be removed, but only with paired training data could it be removed reliably and with accuracy. Without access to paired training data, structured noise required destructive blind-spot networks or impaired unsupervised denoisers, neither of which can be expected to perform as well as a supervised denoiser. One of the key contributions of this thesis was in Chapter 6, where an unsupervised deep learning-based denoiser was developed to remove structured and signal-dependent noise using only unpaired noisy training data. The specific type of structure tackled was row and column correlation, which is common in a range of microscopy modalities. This method was demonstrated to have closer performance to supervised deep learning-based denoisers than all other self- and unsupervised methods, and even surpassed supervised denoisers on some datasets.

The above denoising method is made possible by recognising and utilising structural differences in signal and noise in microscopy. For flow cytometry, signal and noise structures are much harder to distinguish. Nonetheless, two methods for removing structured noise in flow cytometry were developed. The first was in Chapter 5, where the structured background noise of the scattered light signal was removed. To avoid the need for paired training data, this method involved training a deep autoregressive model of light-scattering noise. Doing so required collecting data of pure noise, which is a relatively simple need that can be met by recording a particle-free flow. While this method was shown to enhance the sensitivity of flow cytometry to extra-cellular vesicles, it was unable to remove signal-dependent noise.

The signal-dependence challenge was therefore revisited in Chapter 7, where a beam splitter setup was utilised to create paired observations containing identical underlying signals but independent noise samples. This characteristic was used to train a denoiser with the same technique used in Chapter 6. The paired noisy-noisy training data here could have trained the supervised denoiser N2N [6], but N2N is not able to take advantage of both halves of the pair simultaneously, whereas our method can examine the correlation between each.

Beyond structured noise, another practical challenge for unsupervised denoisers is their prolonged inference time, where a practitioner must repeatedly sample random denoised signals to obtain a consensus solution. In Chapter 3, this challenge was overcome by training a supervised denoiser to predict the randomly sampled signals from an unsupervised denoiser. Depending on its loss function, the supervised denoiser could predict either the pixel-wise median or the mean of the denoised signals. It would thereby provide consensus estimates in a single forward pass.

We believe that the work here has taken a step towards unsupervised deep learning-based denoisers becoming a standard stage of life science observation. Integrating these denoisers with free and easy-to-use libraries such as CAREamics [180] or ZeroCostDL4Mic [114] will make it easier for researchers outside of computer science to adopt them. Already, the denoisers for scattered flow cytometry light [3] and row-correlated imaging noise [4] have been released with interactive-notebook examples. The implementation of the image denoiser was also integrated with the Direct Denoiser from Chapter 3, further easing its use.

Soon, unsupervised denoisers will smoothly integrate into researchers' workflows, and life scientists will be able to push their technology beyond the limits set by noise, trusting that they have the tools to accurately recover clean signals. The impact in live-cell imaging will be greatest, as the delicate dynamics of real processes will be observed in great detail without perturbation by illumination.

[3] github.com/krulllab/deep_nanometry

[4] github.com/krulllab/COSDD

8.2 Future work

Before unsupervised denoisers become mainstays of microscopy and flow cytometry, remaining challenges must be overcome. One challenge is their slow training time compared to regression-based denoisers. With the advent of multi-billion-parameter models and growing interest in implementing deep neural networks on edge devices, there has been a great deal of research into efficient deep learning [181]. It would be interesting to investigate whether recent developments in this field could be used to make unsupervised denoisers even less expensive to implement for researchers.

Another issue is interpretable uncertainty estimation. While randomly sampling signals is useful, manually examining a range of solutions for semantic differences is a subjective process, particularly for images. It could be beneficial to develop an uncertainty estimation method that is more meaningful than a per-pixel variance map, but more quantifiable than a collection of randomly sampled signals.

Lastly, it is crucial that the VAE's behaviour regarding its division of labour [145] is investigated further. In Chapters 6 and 7, we utilised this behaviour to design denoisers. Specifically, we used our knowledge of the dimensions along which noise is correlated in order to design an autoregressive receptive field suitable for modelling this noise. For images, we found that a row of pixels is sufficient for modelling common imaging noise, but is insufficient for modelling signal, and therefore enabled denoising. An outstanding question is: what other noise structures can be separated in the same way? Alternatively, we could ask how data could be designed to enable separate modelling of noise and signal, much like in Chapter 7. An interesting route to explore could be video denoising. Even if noise is highly correlated within a video frame, it is unlikely to be correlated between video frames. It may be possible to exploit this independence to train a denoiser.

The limits of the division of labour should also be investigated. For example, we found that a row of pixels is an insufficient receptive field for an autoregressive model of signal. We believe that this is because the inter-row correlation of signals describing natural phenomena is too strong to be ignored. That is, knowing the values of neighbouring pixels beyond the row massively reduces the uncertainty in the pixel being modelled. This is in contrast to row-correlated noise, where knowing the values of pixels beyond the row has no effect on the uncertainty in the noise of the pixel being modelled. However, imagine a simple signal that smoothly transitions from one brightness to another across the full width of the image. For this, a row-based receptive field would contain all relevant information for an accurate autoregressive model. It is therefore important to find the limit of how simple a signal can be before a row-based receptive field enables an accurate autoregressive model. This would provide more confidence when applying the denoiser. Another degenerate case could be caused by intense noise. Even for complex signals, it could be possible for noise to become so severe that inter-row correlations become irrelevant, and a pixel's signal intensity could be predicted no more accurately using inter-row information than using only intra-row information.

Ultimately, we believe that the VAE's division-of-labour preference, which was identified by Chen *et al*. [145], has the potential to power unsupervised learning-based image restoration in modalities beyond those addressed here, if it is investigated further.

REFERENCES

[1] W. Lee, S. Son, and K. M. Lee, "Ap-bsn: Self-supervised denoising for real-world images via asymmetric pd and blind-spot network," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17725–17734, 2022.

[2] C. Broaddus, A. Krull, M. Weigert, U. Schmidt, and G. Myers, "Removing structured noise with self-supervised blind-spot networks," in *2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)*, pp. 159–163, IEEE, 2020.

[3] E. Höck, T.-O. Buchholz, A. Brachmann, F. Jug, and A. Freytag, "N2v2–fixing noise2void checkerboard artifacts with modified sampling strategies and a tweaked network architecture," *arXiv preprint arXiv:2211.08512*, 2022.

[4] M. Prakash, M. Delbracio, P. Milanfar, and F. Jug, "Interpretable unsupervised diversity denoising and artefact removal," in *International Conference on Learning Representations*, 2022.

[5] M. Weigert, U. Schmidt, T. Boothe, A. Müller, A. Dibrov, A. Jain, B. Wilhelm, D. Schmidt, C. Broaddus, S. Culley, *et al.*, "Content-aware image restoration: pushing the limits of fluorescence microscopy," *Nature methods*, vol. 15, no. 12, pp. 1090–1097, 2018.

[6] J. Lehtinen, J. Munkberg, J. Hasselgren, S. Laine, T. Karras, M. Aittala, and T. Aila, "Noise2noise: Learning image restoration without clean data," in *International Conference on Machine Learning*, pp. 2965–2974, PMLR, 2018.

[7] K. R. Hardie, T. Self, and R. Markus, "Microscopy and microbiology: Moving forward together," in *Journal of Physics: Conference Series*, vol. 2877, p. 012111, IOP Publishing, 2024.

[8] C. Goetz, C. Hammerbeck, J. Bonnevier, J. Bonnevier, C. Hammerbeck, and C. Goetz, "Flow cytometry: definition, history, and uses in biological research," *Flow Cytometry Basics for the Non-Expert*, pp. 1–11, 2018.

[9] K. M. McKinnon, "Flow cytometry: an overview," *Current protocols in immunology*, vol. 120, no. 1, pp. 5–1, 2018.

[10] H. Balasubramanian, C. M. Hobson, T.-L. Chew, and J. S. Aaron, "Imagining the future of optical microscopy: everything, everywhere, all at once," *Communications Biology*, vol. 6, no. 1, p. 1096, 2023.

[11] E. Schaafsma, B. Zhang, Y. Zhao, and C. Cheng, "5.10 - cancer patient stratification based on patterns of immune infiltration," in *Comprehensive Precision Medicine (First Edition)* (K. S. Ramos, ed.), pp. 133–144, Oxford: Elsevier, first edition ed., 2024.

[12] P. P. Laissue, R. A. Alghamdi, P. Tomancak, E. G. Reynaud, and H. Shroff, "Assessing phototoxicity in live fluorescence imaging," *Nature methods*, vol. 14, no. 7, pp. 657–661, 2017.

[13] J. Icha, M. Weber, J. C. Waters, and C. Norden, "Phototoxicity in live fluorescence microscopy, and how to avoid it," *BioEssays*, vol. 39, no. 8, p. 1700003, 2017.

[14] Q. Zheng and L. D. Lavis, "Development of photostable fluorophores for molecular imaging," *Current Opinion in Chemical Biology*, vol. 39, pp. 32–38, 2017. Molecular Imaging Chemical Genetics and Epigenetics.

[15] L. A. Baker and J. L. Rubinstein, "Chapter fifteen - radiation damage in electron cryomicroscopy," in *Cryo-EM Part A Sample Preparation and Data Collection* (G. J. Jensen, ed.), vol. 481 of *Methods in Enzymology*, pp. 371–388, Academic Press, 2010.

[16] G. E. Stillman, "21 - optoelectronics," in *Reference Data for Engineers (Ninth Edition)* (W. M. Middleton and M. E. Van Valkenburg, eds.), p. 19, Woburn: Newnes, ninth edition ed., 2002.

[17] H. Tian, *Noise analysis in CMOS image sensors*. PhD thesis, Stanford University, 2000. Copyright - Database copyright ProQuest LLC; ProQuest does not claim copyright in the individual underlying works; Last updated - 2023-02-24.

[18] A. Buades, B. 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, pp. 60–65 vol. 2, 2005.

[19] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising with block-matching and 3d filtering," in *Image processing: algorithms and systems, neural networks, and machine learning*, vol. 6064, pp. 354–365, SPIE, 2006.

[20] R. C. Gonzalez, *Digital image processing*. Pearson education india, 2009.

[21] W. Meiniel, J.-C. Olivo-Marin, and E. D. Angelini, "Denoising of microscopy images: A review of the state-of-the-art, and a new sparsity-based method," *IEEE Transactions on Image Processing*, vol. 27, no. 8, pp. 3842–3856, 2018.

[22] R. F. Laine, G. Jacquemet, and A. Krull, "Imaging in focus: an introduction to denoising bioimages in the era of deep learning," *The international journal of biochemistry & cell biology*, vol. 140, p. 106077, 2021.

[23] H. C. Burger, C. J. Schuler, and S. Harmeling, "Image denoising: Can plain neural networks compete with bm3d?," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2392–2399, 2012.

[24] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Deep image prior," in *CVPR*, 2018.

[25] A. Krull, T.-O. Buchholz, and F. Jug, "Noise2void-learning denoising from single noisy images," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2129–2137, 2019.

[26] M. Prakash, A. Krull, and F. Jug, "Fully unsupervised diversity denoising with convolutional variational autoencoders," *arXiv preprint arXiv:2006.06072*, 2020.

[27] S. Izadi, D. Sutton, and G. Hamarneh, "Image denoising in the deep learning era," *Artificial Intelligence Review*, vol. 56, no. 7, pp. 5929–5974, 2023.

[28] S. Laine, T. Karras, J. Lehtinen, and T. Aila, "High-quality self-supervised deep image denoising," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[29] S. Deng, Y. Chen, W. Huang, R. Zhang, and Z. Xiong, "Unsupervised domain adaptation for em image denoising with invertible networks," *IEEE Transactions on Medical Imaging*, vol. 44, no. 1, pp. 92–105, 2025.

[30] T.-O. Buchholz, M. Jordan, G. Pigino, and F. Jug, "Cryo-care: Content-aware image restoration for cryo-transmission electron microscopy data," in *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*, pp. 502–506, IEEE, 2019.

[31] J. Batson and L. Royer, "Noise2self: Blind denoising by self-supervision," in *International Conference on Machine Learning*, pp. 524–533, PMLR, 2019.

[32] M. Prakash, M. Lalit, P. Tomancak, A. Krull, and F. Jug, "Fully unsupervised probabilistic noise2void," *arXiv preprint arXiv:1911.12291*, 2019.

[33] S. W. Paddock, "Principles and practices of laser scanning confocal microscopy," *Molecular biotechnology*, vol. 16, pp. 127–149, 2000.

[34] F. Helmchen and W. Denk, "Deep tissue two-photon microscopy," *Nature methods*, vol. 2, no. 12, pp. 932–940, 2005.

[35] G. Herberich, R. Windoffer, R. E. Leube, and T. Aach, "Signal and noise modeling in confocal laser scanning fluorescence microscopy," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2012: 15th International Conference, Nice, France, October 1-5, 2012, Proceedings, Part I 15*, pp. 381–388, Springer, 2012.

[36] European Machine Vision Association (EMVA), "EMVA 1288: Standard for Measurement and Presentation of Specifications for Machine Vision Sensors and Cameras," 2021. Release 4.0 General, 16 June 2021.

[37] A. D. Kiureghian and O. Ditlevsen, "Aleatory or epistemic? does it matter?," *Structural Safety*, vol. 31, no. 2, pp. 105–112, 2009. Risk Acceptance and Risk Communication.

[38] M. Ganaie, M. Hu, A. Malik, M. Tanveer, and P. Suganthan, "Ensemble deep learning: A review," *Engineering Applications of Artificial Intelligence*, vol. 115, p. 105151, 2022.

[39] S. D. Apte, *Introduction to Signals*, p. 1–23. Cambridge University Press, 2016.

[40] Y. Wang, X. Zhang, J. Xu, X. Sun, X. Zhao, H. Li, Y. Liu, J. Tian, X. Hao, X. Kong, Z. Wang, J. Yang, and Y. Su, "The development of microscopic imaging technology and its application in micro- and nanotechnology," *Frontiers in Chemistry*, vol. Volume 10 - 2022, 2022.

[41] M. Breese, E. Vittone, G. Vizkelethy, and P. Sellin, “A review of ion beam induced charge microscopy,” *Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms*, vol. 264, no. 2, pp. 345–360, 2007.

[42] S. Jeon, J. Kim, D. Lee, J. W. Baik, and C. Kim, “Review on practical photoacoustic microscopy,” *Photoacoustics*, vol. 15, p. 100141, 2019.

[43] A. Krull, T. Vicar, M. Prakash, M. Lalit, and F. Jug, “Probabilistic Noise2Void: Unsupervised Content-Aware Denoising,” *Front. Comput. Sci.*, vol. 2, p. 60, Feb. 2020.

[44] F. M. Dekking, *A Modern Introduction to Probability and Statistics: Understanding why and how*, pp. 181–194. Springer Science & Business Media, 2005.

[45] Stelzer, “Contrast, resolution, pixelation, dynamic range and signal-to-noise ratio: fundamental limits to resolution in fluorescence light microscopy,” *Journal of Microscopy*, vol. 189, no. 1, pp. 15–24, 1998.

[46] A. M. Wink and J. B. Roerdink, “Denoising functional mr images: a comparison of wavelet denoising and gaussian smoothing,” *IEEE transactions on medical imaging*, vol. 23, no. 3, pp. 374–387, 2004.

[47] A. Krull, T. Vičar, M. Prakash, M. Lalit, and F. Jug, “Probabilistic noise2void: Unsupervised content-aware denoising,” *Frontiers in Computer Science*, vol. 2, p. 5, 2020.

[48] E. Hüllermeier and W. Waegeman, “Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods,” *Machine learning*, vol. 110, no. 3, pp. 457–506, 2021.

[49] A. Foi, M. Trimeche, V. Katkovnik, and K. Egiazarian, “Practical poissonian-gaussian noise modeling and fitting for single-image raw-data,” *IEEE Transactions on Image Processing*, vol. 17, no. 10, pp. 1737–1754, 2008.

[50] M. Prakash, M. Lalit, P. Tomancak, A. Krul, and F. Jug, “Fully unsupervised probabilistic noise2void,” in *2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)*, pp. 154–158, IEEE, 2020.

[51] A. Van Oord, N. Kalchbrenner, and K. Kavukcuoglu, “Pixel recurrent neural networks,” in *International conference on machine learning*, pp. 1747–1756, PMLR, 2016.

[52] D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” *stat*, vol. 1050, p. 1, 2014.

[53] D. P. Kingma, M. Welling, *et al.*, “An introduction to variational autoencoders,” *Foundations and Trends® in Machine Learning*, vol. 12, no. 4, pp. 307–392, 2019.

[54] S. Kullback and R. A. Leibler, “On information and sufficiency,” *The annals of mathematical statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[55] C. K. Sønderby, T. Raiko, L. Maaløe, S. K. Sønderby, and O. Winther, “Ladder variational autoencoders,” *Advances in neural information processing systems*, vol. 29, 2016.

[56] B. Salmon and A. Krull, "Direct unsupervised denoising," in *2023 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pp. 3840–3847, 2023.

[57] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, 2017.

[58] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241, Springer, 2015.

[59] N. Moran, D. Schmidt, Y. Zhong, and P. Coady, "Noisier2noise: Learning to denoise from unpaired noisy data," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12064–12072, 2020.

[60] Y. Quan, M. Chen, T. Pang, and H. Ji, "Self2self with dropout: Learning self-supervised denoising from single image," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1890–1898, 2020.

[61] C. Broaddus, A. Krull, M. Weigert, U. Schmidt, and G. Myers, "Removing structured noise with self-supervised blind-spot networks," in *2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)*, pp. 159–163, 2020.

[62] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *stat*, vol. 1050, p. 9, 2015.

[63] T. G. Dietterich, "Ensemble methods in machine learning," in *International workshop on multiple classifier systems*, pp. 1–15, Springer, 2000.

[64] J. Gou, B. Yu, S. J. Maybank, and D. Tao, "Knowledge distillation: A survey," *International Journal of Computer Vision*, vol. 129, pp. 1789–1819, 2021.

[65] A. Shamilov, A. F. Yuzer, E. Agaoglu, and Y. Mert, "A method of obtaining distributions of transformed random variables by using the heaviside and the dirac generalized functions," *Journal of Statistical Research*, vol. 40, no. 1, pp. 23–34, 2006.

[66] H. Weisberg, *Central tendency and variability*. Sage, 1992.

[67] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[68] A. F. Agarap, "Deep learning using rectified linear units (relu)," *arXiv preprint arXiv:1803.08375*, 2018.

[69] O. Rukundo and H. Cao, "Nearest neighbor value interpolation," *arXiv preprint arXiv:1211.1768*, 2012.

[70] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization. 3rd international conference on learning representations, iclr 2015," *arXiv preprint arXiv:1412.6980*, vol. 9, 2015.

[71] V. Belagiannis, C. Rupprecht, G. Carneiro, and N. Navab, "Robust optimization for deep regression," in *Proceedings of the IEEE international conference on computer vision*, pp. 2830–2838, 2015.

[72] A. Mustafa, A. Mikhailiuk, D. A. Iliescu, V. Babbar, and R. K. Mantiuk, "Training a task-specific image reconstruction loss," in *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pp. 2319–2328, 2022.

[73] B. Salmon and A. Krull, "Towards structured noise models for unsupervised denoising," in *Computer Vision – ECCV 2022 Workshops* (L. Karlinsky, T. Michaeli, and K. Nishino, eds.), (Cham), pp. 379–394, Springer Nature Switzerland, 2023.

[74] F. Luisier, C. Vonesch, T. Blu, and M. Unser, "Fast interscale wavelet denoising of poisson-corrupted images," *Signal processing*, vol. 90, no. 2, pp. 415–427, 2010.

[75] C. Belthangady and L. A. Royer, "Applications, promises, and pitfalls of deep learning for fluorescence image reconstruction," *Nature methods*, pp. 1–11, 2019.

[76] Y. Zhang, Y. Zhu, E. Nichols, Q. Wang, S. Zhang, C. Smith, and S. Howard, "A poisson-gaussian denoising dataset with real fluorescence microscopy images," in *CVPR*, 2019.

[77] M. Xu and L. V. Wang, "Photoacoustic imaging in biomedicine," *Review of scientific instruments*, vol. 77, no. 4, p. 041101, 2006.

[78] H. P. Babcock, F. Huang, and C. M. Speer, "Correcting artifacts in single molecule localization microscopy analysis arising from pixel quantum efficiency differences in scmos cameras," *Scientific reports*, vol. 9, no. 1, pp. 1–10, 2019.

[79] A. Abdelhamed, M. A. Brubaker, and M. S. Brown, "Noise flow: Noise modeling with conditional normalizing flows," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3165–3173, 2019.

[80] D. P. Kingma and P. Dhariwal, "Glow: Generative flow with invertible 1x1 convolutions," *Advances in neural information processing systems*, vol. 31, 2018.

[81] S. Bond-Taylor, A. Leach, Y. Long, and C. G. Willcocks, "Deep generative modelling: A comparative review of vaes, gans, normalizing flows, energy-based and autoregressive models," *arXiv preprint arXiv:2103.04922*, 2021.

[82] A. Van den Oord, N. Kalchbrenner, L. Espeholt, O. Vinyals, A. Graves, *et al.*, "Conditional image generation with pixelcnn decoders," *Advances in neural information processing systems*, vol. 29, 2016.

[83] B. Mandracchia, X. Hua, C. Guo, J. Son, T. Urner, and S. Jia, "Fast and accurate scmos noise correction for fluorescence microscopy," *Nature communications*, vol. 11, no. 1, p. 94, 2020.

[84] M. Hssayeni, M. Croock, A. Salman, H. Al-khafaji, Z. Yahya, and B. Ghoraani, "Computed tomography images for intracranial hemorrhage detection and segmentation," *Intracranial Hemorrhage Segmentation Using A Deep Convolutional Model. Data*, vol. 5, no. 1, 2020.

[85] A. Parakh, C. An, S. Lennartz, P. Rajiah, B. M. Yeh, F. J. Simeone, D. V. Sahani, and A. R. Kambadakone, "Recognizing and minimizing artifacts at dual-energy ct," *Radiographics*, vol. 41, no. 2, p. 509, 2021.

[86] T. Clanuwat, M. Bober-Irizar, A. Kitamoto, A. Lamb, K. Yamamoto, and D. Ha, "Deep learning for classical japanese literature," *arXiv preprint arXiv:1812.01718*, 2018.

[87] P. Beard, "Biomedical photoacoustic imaging," *Interface focus*, vol. 1, no. 4, pp. 602–631, 2011.

[88] D. P. Kingma, T. Salimans, R. Jozefowicz, X. Chen, I. Sutskever, and M. Welling, "Improved variational inference with inverse autoregressive flow," *Advances in neural information processing systems*, vol. 29, 2016.

[89] Y. Iwamoto, B. Salmon, Y. Yoshioka, R. Kojima, A. Krull, and S. Ota, "High throughput analysis of rare nanoparticles with deep-enhanced sensitivity via unsupervised denoising," *Nature Communications*, vol. 16, no. 1, p. 1728, 2025.

[90] T. Sych, J. Schlegel, H. M. Barriga, M. Ojansivu, L. Hanke, F. Weber, R. Beklem Bostancioglu, K. Ezzat, H. Stangl, B. Plochberger, *et al.*, "High-throughput measurement of the content and properties of nano-sized bioparticles with single-particle profiler," *Nature Biotechnology*, vol. 42, no. 4, pp. 587–590, 2024.

[91] H. Lian, S. He, C. Chen, and X. Yan, "Flow cytometric analysis of nanoscale biological particles and organelles," *Annual Review of Analytical Chemistry*, vol. 12, no. 1, pp. 389–409, 2019.

[92] G. Bordanaba-Florit, F. Royo, S. G. Kruglik, and J. M. Falcón-Pérez, "Using single-vesicle technologies to unravel the heterogeneity of extracellular vesicles," *Nature Protocols*, vol. 16, no. 7, pp. 3163–3185, 2021.

[93] S. Gioria, F. Caputo, P. Urbán, C. M. Maguire, S. Bremer-Hoffmann, A. Prina-Mello, L. Calzolai, and D. Mehn, "Are existing standard methods suitable for the evaluation of nanomedicines: some case studies," *Nanomedicine*, vol. 13, no. 5, pp. 539–554, 2018.

[94] V. P. Chauhan and R. K. Jain, "Strategies for advancing cancer nanomedicine," *Nature materials*, vol. 12, no. 11, pp. 958–962, 2013.

[95] P. L. Mage, A. T. Csordas, T. Brown, D. Klinger, M. Eisenstein, S. Mitragotri, C. Hawker, and H. T. Soh, "Shape-based separation of synthetic microparticles," *Nature materials*, vol. 18, no. 1, pp. 82–89, 2019.

[96] Y. Wu, S. K. Campos, G. P. Lopez, M. A. Ozbun, L. A. Sklar, and T. Buranda, "The development of quantum dot calibration beads and quantitative multicolor bioassays in flow cytometry and microscopy," *Analytical biochemistry*, vol. 364, no. 2, pp. 180–192, 2007.

[97] D. Schraivogel, T. M. Kuhn, B. Rauscher, M. Rodríguez-Martínez, M. Paulsen, K. Owsley, A. Middlebrook, C. Tischer, B. Ramasz, D. Ordoñez-Rueda, *et al.*, "High-speed fluorescence image–enabled cell sorting," *Science*, vol. 375, no. 6578, pp. 315–320, 2022.

[98] S. Ryuzaki, T. Yasui, M. Tsutsui, K. Yokota, Y. Komoto, P. Paisrisarn, N. Kaji, D. Ito, K. Tamada, T. Ochiya, *et al.*, "Rapid discrimination of extracellular vesicles by shape distribution analysis," *Analytical Chemistry*, vol. 93, no. 18, pp. 7037–7044, 2021.

[99] C. Théry, K. W. Witwer, E. Aikawa, M. J. Alcaraz, J. D. Anderson, R. Andriantsitohaina, A. Antoniou, T. Arab, F. Archer, G. K. Atkin-Smith, *et al.*, "Minimal information for studies of extracellular vesicles 2018 (misev2018): a position statement of the international society for extracellular vesicles and update of the misev2014 guidelines," *Journal of extracellular vesicles*, vol. 7, no. 1, p. 1535750, 2018.

[100] A. Zijlstra and D. Di Vizio, "Size matters in nanoscale communication," *Nature cell biology*, vol. 20, no. 3, pp. 228–230, 2018.

[101] J. Kowal, G. Arras, M. Colombo, M. Jouve, J. P. Morath, B. Primdal-Bengtson, F. Dingli, D. Loew, M. Tkach, and C. Théry, "Proteomic comparison defines novel markers to characterize heterogeneous populations of extracellular vesicle subtypes," *Proceedings of the National Academy of Sciences*, vol. 113, no. 8, pp. E968–E977, 2016.

[102] E. Willms, C. Cabañas, I. Mäger, M. J. Wood, and P. Vader, "Extracellular vesicle heterogeneity: subpopulations, isolation techniques, and diverse functions in cancer progression," *Frontiers in immunology*, vol. 9, p. 738, 2018.

[103] M. Yáñez-Mó, P. R.-M. Siljander, Z. Andreu, A. Bedina Zavec, F. E. Borràs, E. I. Buzas, K. Buzas, E. Casal, F. Cappello, J. Carvalho, *et al.*, "Biological properties of extracellular vesicles and their physiological functions," *Journal of extracellular vesicles*, vol. 4, no. 1, p. 27066, 2015.

[104] M. Naiim, A. Boualem, C. Ferre, M. Jabloun, A. Jalocha, and P. Ravier, "Multiangle dynamic light scattering for the improvement of multimodal particle size distribution measurements," *Soft matter*, vol. 11, no. 1, pp. 28–32, 2015.

[105] M. Dehghani, S. M. Gulvin, J. Flax, and T. R. Gaborski, "Systematic evaluation of pkh labelling on extracellular vesicle size by nanoparticle tracking analysis," *Scientific Reports*, vol. 10, no. 1, p. 9533, 2020.

[106] V. Filipe, A. Hawe, and W. Jiskoot, "Critical evaluation of nanoparticle tracking analysis (nta) by nanosight for the measurement of nanoparticles and protein aggregates," *Pharmaceutical research*, vol. 27, pp. 796–810, 2010.

[107] S. Zhu, L. Ma, S. Wang, C. Chen, W. Zhang, L. Yang, W. Hang, J. P. Nolan, L. Wu, and X. Yan, "Light-scattering detection below the level of single fluorescent molecules for high-resolution characterization of functional nanoparticles," *ACS nano*, vol. 8, no. 10, pp. 10998–11006, 2014.

[108] L. Li, S. Wang, J. Xue, Y. Lin, L. Su, C. Xue, C. Mao, N. Cai, Y. Tian, S. Zhu, *et al.*, "Development of spectral nano-flow cytometry for high-throughput multiparameter analysis of individual biological nanoparticles," *Analytical Chemistry*, vol. 95, no. 6, pp. 3423–3433, 2023.

[109] J.-L. Fraikin, T. Teesalu, C. M. McKenney, E. Ruoslahti, and A. N. Cleland, "A high-throughput label-free nanoparticle analyser," *Nature nanotechnology*, vol. 6, no. 5, pp. 308–313, 2011.

[110] A. D. Kashkanova, M. Blessing, A. Gemeinhardt, D. Soulat, and V. Sandoghdar, "Precision size and refractive index analysis of weakly scattering nanoparticles in polydispersions," *Nature methods*, vol. 19, no. 5, pp. 586–593, 2022.

[111] R. E. Veerman, L. Teeuwen, P. Czarnewski, G. Güclüler Akpinar, A. Sandberg, X. Cao, M. Pernemalm, L. M. Orre, S. Gabrielsson, and M. Eldh, "Molecular evaluation of five different isolation methods for extracellular vesicles reveals different clinical applicability and subcellular origin," *Journal of extracellular vesicles*, vol. 10, no. 9, p. e12128, 2021.

[112] Y. Yoshioka, N. Kosaka, Y. Konishi, H. Ohta, H. Okamoto, H. Sonoda, R. Nonaka, H. Yamamoto, H. Ishii, M. Mori, *et al.*, "Ultra-sensitive liquid biopsy of circulating extracellular vesicles using exoscreen," *Nature communications*, vol. 5, no. 1, p. 3591, 2014.

[113] S. Chaudhary, S. Moon, and H. Lu, "Fast, efficient, and accurate neuro-imaging denoising via supervised deep learning," *Nature communications*, vol. 13, no. 1, p. 5165, 2022.

[114] L. von Chamier, R. F. Laine, J. Jukkala, C. Spahn, D. Krentzel, E. Nehme, M. Lerche, S. Hernández-Pérez, P. K. Mattila, E. Karinou, *et al.*, "Democratising deep learning for microscopy with zerocostdl4mic," *Nature communications*, vol. 12, no. 1, p. 2276, 2021.

[115] C. Zuo, J. Qian, S. Feng, W. Yin, Y. Li, P. Fan, J. Han, K. Qian, and Q. Chen, "Deep learning in optical metrology: a review," *Light: Science & Applications*, vol. 11, no. 1, pp. 1–54, 2022.

[116] L. de Rond, F. A. Coumans, J. A. Welsh, R. Nieuwland, T. G. van Leeuwen, and E. van der Pol, "Quantification of light scattering detection efficiency and background in flow cytometry," *Cytometry*, vol. 99, no. 7, p. 671, 2020.

[117] Q. Huynh-Thu and M. Ghanbari, "Scope of validity of psnr in image/video quality assessment," *Electronics letters*, vol. 44, no. 13, pp. 800–801, 2008.

[118] V. Raghavan, P. Bollmann, and G. S. Jung, "A critical investigation of recall and precision as measures of retrieval system performance," *ACM Transactions on Information Systems (TOIS)*, vol. 7, no. 3, pp. 205–229, 1989.

[119] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010.

[120] F. M. Dekking, C. Kraaikamp, H. P. Lopuhaä, and L. E. Meester, *Covariance and correlation*, pp. 135–150. London: Springer London, 2005.

[121] C. F. Bohren and D. R. Huffman, *Absorption and scattering of light by small particles*. John Wiley & Sons, 2008.

[122] J. W. Clancy, A. C. Boomgarden, and C. D'Souza-Schorey, "Profiling and promise of supermeres," *Nature cell biology*, vol. 23, no. 12, pp. 1217–1219, 2021.

[123] D. K. Jeppesen, Q. Zhang, J. L. Franklin, and R. J. Coffey, "Extracellular vesicles and nanoparticles: emerging complexities," *Trends in Cell Biology*, vol. 33, no. 8, pp. 667–681, 2023.

[124] Q. Zhang, D. K. Jeppesen, J. N. Higginbotham, R. Graves-Deal, V. Q. Trinh, M. A. Ramirez, Y. Sohn, A. C. Neininger, N. Taneja, E. T. McKinley, *et al.*, "Supermeres are functional extracellular nanoparticles replete with disease biomarkers and therapeutic targets," *Nature cell biology*, vol. 23, no. 12, pp. 1240–1254, 2021.

[125] H. Zhang, D. Freitas, H. S. Kim, K. Fabijanic, Z. Li, H. Chen, M. T. Mark, H. Molina, A. B. Martin, L. Bojmar, *et al.*, "Identification of distinct nanoparticles and subsets of extracellular vesicles by asymmetric flow field-flow fractionation," *Nature cell biology*, vol. 20, no. 3, pp. 332–343, 2018.

[126] J. Stam, S. Bartel, R. Bischoff, and J. C. Wolters, "Isolation of extracellular vesicles with combined enrichment methods," *Journal of Chromatography B*, vol. 1169, p. 122604, 2021.

[127] S. C. Jang, O. Y. Kim, C. M. Yoon, D.-S. Choi, T.-Y. Roh, J. Park, J. Nilsson, J. Lotvall, Y.-K. Kim, and Y. S. Gho, "Bioinspired exosome-mimetic nanovesicles for targeted delivery of chemotherapeutics to malignant tumors," *ACS nano*, vol. 7, no. 9, pp. 7698–7710, 2013.

[128] Y. Tian, L. Ma, M. Gong, G. Su, S. Zhu, W. Zhang, S. Wang, Z. Li, C. Chen, L. Li, *et al.*, "Protein profiling and sizing of extracellular vesicles from colorectal cancer patients via flow cytometry," *ACS nano*, vol. 12, no. 1, pp. 671–680, 2018.

[129] T. Yasui, T. Yanagida, S. Ito, Y. Konakade, D. Takeshita, T. Naganawa, K. Nagashima, T. Shimada, N. Kaji, Y. Nakamura, *et al.*, "Unveiling massive numbers of cancer-related urinary-microrna candidates via nanowires," *Science Advances*, vol. 3, no. 12, p. e1701133, 2017.

[130] A. Hoshino, H. S. Kim, L. Bojmar, K. E. Gyan, M. Cioffi, J. Hernandez, C. P. Zambirinis, G. Rodrigues, H. Molina, S. Heissel, *et al.*, "Extracellular vesicle and particle biomarkers define multiple human cancers," *Cell*, vol. 182, no. 4, pp. 1044–1061, 2020.

[131] H. Shin, Y. Kang, K. W. Choi, S. Kim, B.-J. Ham, and Y. Choi, "Artificial intelligence-based major depressive disorder (mdd) diagnosis using raman spectroscopic features of plasma exosomes," *Analytical chemistry*, vol. 95, no. 15, pp. 6410–6416, 2023.

[132] L. Hu, T. Zhang, H. Ma, Y. Pan, S. Wang, X. Liu, X. Dai, Y. Zheng, L. P. Lee, and F. Liu, "Discovering the secret of diseases by incorporated tear exosomes analysis via rapid-isolation system: itears," *ACS nano*, vol. 16, no. 8, pp. 11720–11732, 2022.

[133] M. Rezeli, O. Gidlöf, M. Evander, P. Bryl-Górecka, R. Sathanoori, P. Gilje, K. Pawłowski, P. Horvatovich, D. Erlinge, G. Marko-Varga, *et al.*, "Comparative proteomic analysis of extracellular vesicles isolated by acoustic trapping or differential centrifugation," *Analytical chemistry*, vol. 88, no. 17, pp. 8577–8586, 2016.

[134] R. L. Siegel, K. D. Miller, N. S. Wagle, and A. Jemal, "Cancer statistics, 2023," *CA: a cancer journal for clinicians*, vol. 73, no. 1, pp. 17–48, 2023.

[135] M. Ugawa, Y. Kawamura, K. Toda, K. Teranishi, H. Morita, H. Adachi, R. Tamoto, H. Nomaru, K. Nakagawa, K. Sugimoto, *et al.*, "In silico-labeled ghost cytometry," *Elife*, vol. 10, p. e67660, 2021.

[136] N. C. Lindquist, C. D. L. de Albuquerque, R. G. Sobral-Filho, I. Paci, and A. G. Brolo, "High-speed imaging of surface-enhanced raman scattering fluctuations from individual nanoparticles," *Nature nanotechnology*, vol. 14, no. 10, pp. 981–987, 2019.

[137] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[138] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*, pp. 448–456, pmlr, 2015.

[139] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (elus)," *arXiv preprint arXiv:1511.07289*, 2015.

[140] B. Salmon and A. Krull, "Unsupervised denoising for signal-dependent and row-correlated imaging noise," in *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 2379–2389, IEEE, 2025.

[141] M. Eom, S. Han, P. Park, G. Kim, E.-S. Cho, J. Sim, K.-H. Lee, S. Kim, H. Tian, U. L. Böhm, *et al.*, "Statistically unbiased prediction enables accurate denoising of voltage imaging data," *Nature Methods*, pp. 1–12, 2023.

[142] T. Huang, S. Li, X. Jia, H. Lu, and J. Liu, "Neighbor2neighbor: Self-supervised denoising from single noisy images," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 14781–14790, 2021.

[143] F. Wang, T. R. Henninen, D. Keller, and R. Erni, "Noise2atom: unsupervised denoising for scanning transmission electron microscopy images," *Applied Microscopy*, vol. 50, no. 1, pp. 1–9, 2020.

[144] S. Gu, L. Zhang, W. Zuo, and X. Feng, "Weighted nuclear norm minimization with application to image denoising," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2862–2869, 2014.

[145] X. Chen, D. P. Kingma, T. Salimans, Y. Duan, P. Dhariwal, J. Schulman, I. Sutskever, and P. Abbeel, “Variational lossy autoencoder,” in *International Conference on Learning Representations*, 2017.

[146] I. Gulrajani, K. Kumar, F. Ahmed, A. A. Taiga, F. Visin, D. Vazquez, and A. Courville, “Pixelvae: A latent variable model for natural images,” *arXiv preprint arXiv:1611.05013*, 2016.

[147] Z. Wang, J. Liu, G. Li, and H. Han, “Blind2unblind: Self-supervised image denoising with visible blind spots,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2027–2036, 2022.

[148] A. Krull, H. Basevi, B. Salmon, A. Zeug, F. Müller, S. Tonks, L. Muppala, and A. Leonardis, “Image denoising and the generative accumulation of photons,” in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1528–1537, 2024.

[149] T. Pang, H. Zheng, Y. Quan, and H. Ji, “Recorrupted-to-recorrupted: Unsupervised deep learning for image denoising,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2043–2052, 2021.

[150] H. Chung, J. Kim, M. T. Mccann, M. L. Klasky, and J. C. Ye, “Diffusion posterior sampling for general noisy inverse problems,” *arXiv preprint arXiv:2209.14687*, 2022.

[151] M. Elad, B. Kawar, and G. Vaksman, “Image denoising: The deep learning revolution and beyond—a survey paper,” *SIAM Journal on Imaging Sciences*, vol. 16, no. 3, pp. 1594–1654, 2023.

[152] Y. Song, L. Shen, L. Xing, and S. Ermon, “Solving inverse problems in medical imaging with score-based generative models,” in *NeurIPS 2021 Workshop on Deep Learning and Inverse Problems*.

[153] A. Aali, M. Arvinte, S. Kumar, and J. I. Tamir, “Solving inverse problems with score-based generative priors learned from noisy data,” in *2023 57th Asilomar Conference on Signals, Systems, and Computers*, pp. 837–843, IEEE, 2023.

[154] G. Daras, K. Shah, Y. Dagan, A. Gollakota, A. Dimakis, and A. Klivans, “Ambient diffusion: Learning clean distributions from corrupted data,” *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[155] M. Mirza and S. Osindero, “Conditional generative adversarial nets,” *arXiv preprint arXiv:1411.1784*, 2014.

[156] Z. Yang, C. Yan, and H. Chen, “Unpaired low-dose ct denoising using conditional gan with structural loss,” in *2021 International Conference on Wireless Communications and Smart Grid (ICWCSG)*, pp. 272–275, IEEE, 2021.

[157] W. Wang, F. Wen, Z. Yan, and P. Liu, “Optimal transport for unsupervised denoising learning,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 2, pp. 2104–2118, 2022.

[158] S. J. Pennycook and P. D. Nellist, *Scanning transmission electron microscopy: imaging and analysis*. Springer Science & Business Media, 2011.

[159] Y. Liao, "Practical electron microscopy and database," *An Online Book*, p. 1412, 2006.

[160] D. J. Denvir and E. Conroy, "Electron multiplying ccds," in *Opto-Ireland 2002: Optical Metrology, Imaging, and Machine Vision*, vol. 4877, pp. 55–68, SPIE, 2003.

[161] R. Boie and I. Cox, " An Analysis of Camera Noise ," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 14, pp. 671–674, June 1992.

[162] B. Moomaw, "Camera technologies for low light imaging: overview and relative advantages," *Methods in cell biology*, vol. 114, pp. 243–283, 2013.

[163] Z. Zhang, Y. Wang, R. Piestun, and Z.-L. Huang, "Characterizing and correcting camera noise in back-illuminated scmos cameras," *Optics Express*, vol. 29, no. 5, pp. 6668–6690, 2021.

[164] B. Dupont, A. Dupret, E. Belhaire, and P. Villard, "Fpn sources in bolometric infrared detectors," *IEEE Sensors Journal*, vol. 9, no. 8, pp. 944–952, 2009.

[165] J. He, D. Spokoyny, G. Neubig, and T. Berg-Kirkpatrick, "Lagging inference networks and posterior collapse in variational autoencoders," in *International Conference on Learning Representations*, 2019.

[166] T. Hastie, R. Tibshirani, J. H. Friedman, and J. H. Friedman, *The elements of statistical learning: data mining, inference, and prediction*, pp. 18–22. Springer, 2009.

[167] G. M. Hagen, J. Bendesky, R. Machado, T.-A. Nguyen, T. Kumar, and J. Ventura, "Fluorescence microscopy datasets for training deep neural networks," *GigaScience*, vol. 10, no. 5, p. giab032, 2021.

[168] A. K. Glaser, K. W. Bishop, L. A. Barner, E. A. Susaki, S. I. Kubota, G. Gao, R. B. Serafin, P. Balaram, E. Turschak, P. R. Nicovich, *et al.*, "A hybrid open-top light-sheet microscope for versatile multi-scale imaging of cleared tissues," *Nature methods*, vol. 19, no. 5, pp. 613–619, 2022.

[169] J. Portmann, S. Lynen, M. Chli, and R. Siegwart, "People detection and tracking from aerial thermal views," in *2014 IEEE international conference on robotics and automation (ICRA)*, pp. 1794–1800, IEEE, 2014.

[170] T. R. Henninen, M. Bon, F. Wang, D. Passerone, and R. Erni, "The structure of sub-nm platinum clusters at elevated temperatures," *Angewandte Chemie International Edition*, vol. 59, no. 2, pp. 839–845, 2020.

[171] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4401–4410, 2019.

[172] H. Kanegane, A. Hoshino, T. Okano, T. Yasumi, T. Wada, H. Takada, S. Okada, M. Yamashita, T.-w. Yeh, R. Nishikomori, *et al.*, "Flow cytometry-based diagnosis of primary immunodeficiency diseases," *Allergology International*, vol. 67, no. 1, pp. 43–54, 2018.

[173] J. A. DiGiuseppe and B. L. Wood, "Applications of flow cytometric immunophenotyping in the diagnosis and posttreatment monitoring of b and t lymphoblastic leukemia/lymphoma," *Cytometry Part B: Clinical Cytometry*, vol. 96, no. 4, pp. 256–265, 2019.

[174] P. Verma, A. Singh, R. Kushwaha, G. Yadav, S. P. Verma, U. S. Singh, H. D. Reddy, and A. Agarwal, "Early and effective diagnosis of sepsis using flow cytometry," *Journal of Laboratory Physicians*, vol. 15, no. 02, pp. 230–236, 2023.

[175] Y. Ding, A. E. Dulau-Florea, E. M. Groarke, B. A. Patel, D. B. Beck, P. C. Grayson, M. A. Ferrada, N. S. Young, K. R. Calvo, and R. C. Braylan, "Use of flow cytometric light scattering to recognize the characteristic vacuolated marrow cells in vexas syndrome," *Blood Advances*, vol. 7, no. 20, pp. 6151–6155, 2023.

[176] D. M. Betters, "Use of flow cytometry in clinical practice," *Journal of the advanced practitioner in oncology*, vol. 6, no. 5, p. 435, 2015.

[177] D. V. Voronin, A. A. Kozlova, R. A. Verkhovskii, A. V. Ermakov, M. A. Makarkin, O. A. Inozemtseva, and D. N. Bratashov, "Detection of rare objects by flow cytometry: imaging, cell sorting, and deep learning approaches," *International journal of molecular sciences*, vol. 21, no. 7, p. 2323, 2020.

[178] R. W. Engstrom, *Photomultiplier handbook*. RCA Corporation, 1980.

[179] R. Koenker, *Quantile regression*, vol. 38. Cambridge university press, 2005.

[180] CAREamics contributors, "Careamics."

[181] G. Menghani, "Efficient deep learning: A survey on making deep learning models smaller, faster, and better," *ACM Comput. Surv.*, vol. 55, Mar. 2023.

[182] Y. Luo and N. Mesgarani, "Tasnet: time-domain audio separation network for real-time, single-channel speech separation," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 696–700, IEEE, 2018.

[183] J. A. Welsh, G. J. Arkesteijn, M. Bremer, M. Cimorelli, F. Dignat-George, B. Giebel, A. Görgens, A. Hendrix, M. Kuiper, R. Lacroix, *et al.*, "A compendium of single extracellular vesicle flow cytometry," *Journal of extracellular vesicles*, vol. 12, no. 2, p. e12299, 2023.

[184] L. Pan, S. Shrestha, N. Taylor, W. Nie, and L. R. Cao, "Determination of x-ray detection limit and applications in perovskite x-ray detectors," *Nature communications*, vol. 12, no. 1, p. 5258, 2021.

[185] G. Long and J. Winefordner, "Limit of detection. a closer look at the iupac definition," *Analytical Chemistry - ANAL CHEM*, vol. 55, 09 2008.

[186] I. Recommendations, "Nomenclature in evaluation of analytical methods, including detection and quantification capabilities," *Pure Appl. Chem*, vol. 67, p. 1699, 1995.

[187] D. Misra, "Mish: A self regularized non-monotonic activation function," *arXiv preprint arXiv:1908.08681*, 2019.

[188] A. Vahdat and J. Kautz, "Nvae: A deep hierarchical variational autoencoder," *Advances in neural information processing systems*, vol. 33, pp. 19667–19679, 2020.

[189] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," *arXiv preprint arXiv:1511.07122*, 2015.

9 HIGH THROUGHPUT ANALYSIS OF RARE NANOPARTICLES WITH DEEP-ENHANCED SENSITIVITY VIA UNSUPERVISED DENOISING - SUPPLEMENTARY

9.1 Comparison to a supervised deep learning approach

To benchmark the performance of the unsupervised method in DNM, we compare it to a supervised deep learning-based denoiser. This supervised denoiser is a network trained using the attenuated data as input and the clean ground-truth data as target. In practice, such training data constituting both noisy data and noise-free data is much more difficult to obtain than the data required for the unsupervised method in DNM. DNM only requires a time series from particle-free ultrapure water, collected separately from the noisy time series that is to be denoised. For the supervised denoiser to also predict the point-wise median of possible solutions (Apparatus and denoising workflow for DNM), it is optimised by minimizing the L1 norm of the error between prediction and target. For a fair comparison, the architecture of the supervised network is the same as the architecture of the VAE in the unsupervised DNM, except stochastic layers are made deterministic by replacing random samples from the learned Gaussian distributions with the learned mean of the distributions. This makes the network a U-Net [58]. Unlike the unsupervised DNM, a denoiser trained in a supervised manner should not be evaluated using its training data, so we withheld 10% of the total dataset for evaluation and used it to evaluate both methods. The results in Supplementary Fig. 9.7 were obtained from this subset.

First, Supplementary Fig. 9.7(a) shows a max-normalised ground-truth time series, the corresponding attenuated time series before denoising, and the attenuated time series after denoising by both the supervised denoiser and the unsupervised denoiser in DNM. The supervised denoiser produced a result more closely resembling the shape of ground truth peak. In the histograms of max-normalised scattering height distributions in Supplementary Fig. 9.7(b), the supervised denoiser distinguishes more particle detections from noise than the104 unsupervised denoiser in DNM, which has some overlap between noise and particle scattering heights below 0.05.

Next, Supplementary Fig. 9.7(c) shows the scale-invariant peak signal-to-noise ratio (PSNR) [182] of the attenuated time series before and after denoising by both the supervised denoiser and the unsupervised denoiser of DNM. This was obtained by dividing all time series equally into windows of equal length of 200 sample points and employing the ground truth to disregard windows containing no particle detections. We find that the supervised denoiser improves scale-invariant PSNR in 79.6% of cases, compared to 76.0% of cases denoised by the unsupervised DNM. The supervised denoiser achieves a median scale-invariant PSNR value of 15.6 dB, while the unsupervised DNM achieves a median scale-invariant PSNR value of 15.2 dB. The raw attenuated data has a median scale-invariant PSNR value of 14.7 dB. Finally, the PR curve in Supplementary Fig. 9.7(d) was generated using the same procedure as Fig. 5.2(e) in the original manuscript. The two denoisers produce output with a similar precision up to a recall of 0.6. We observe that the precision of the supervised denoiser's output sharply drops at a recall of 0.8, compared to the precision of DNM's output sharply dropping at a recall of 0.7. The mAP of the supervised denoiser's output is 0.04 higher than the unsupervised DNM's output.

In conclusion, the supervised method outperforms DNM across metrics, indicating what could be possible with future methods. We believe that the main reason for this gap is the fact that the supervised approach can learn to remove signal independent noise as well as signal dependent shot noise, while our method is currently trained solely to remove signal independent noise. However, it should be again noted that the ground truth data required for supervised methods is rarely available in practice, and our unsupervised method did provide a substantial improvement.

9.2 Training and evaluation time

Our workstation consists of a 36 core Intel(R) Core(TM) i9-10980XE CPU @ 3.00GHz with 134GB of RAM running Ubuntu 22.04.4, Python 3.11.5 and pytorch-lightning 2.2.1. The GPU used for training is a NVIDIA GeForce RTX 3090 with 24GB of VRAM. The minimum VRAM required for training the model is 2GB. In our environment, total training time for the noise model and the VAE, using 1 second of data for each, takes 2.5 hours. Once the model is trained, denoising a 1 second time series by randomly sampling 100 denoised time series from the estimated signal distribution and computing the median of them takes 8.5 minutes.

9.3 Rayleigh scattering theory and fitting strategy

Using the Rayleigh theory [121], the scattering amplitude for small particles is expressed as,

$$\text{Scattering amplitude}(d) = A \times \sigma_{\text{scat}}(d, n_{\text{med}}, n_{\text{particle},\lambda}) = A \times \frac{2\pi^5 d^6 n_{\text{med}^4}}{3\lambda^4} \left| \frac{m^2-1}{m^2+2} \right| = cd^6, \tag{9.1}$$

where σ_{scat} is the scattering cross section, $m = \frac{n_{\text{particle}}}{n_{\text{med}}}$, d is the particle size, n_{med} is the refractive index of the medium, n_{particle} is the refractive index of the particle, and λ is the excitation wavelength. The scattering amplitude is proportional to the scattering cross section, scaled by a factor A. All of these, except d, are parameters of the experimental setup and can be reduced to the constant factor c. In the fitting, using the medians of the experimentally obtained scattering height for 40 nm and 60 nm particles, respectively, we obtained $c =$ 3.4×10^{-11} in Eq. (9.1).

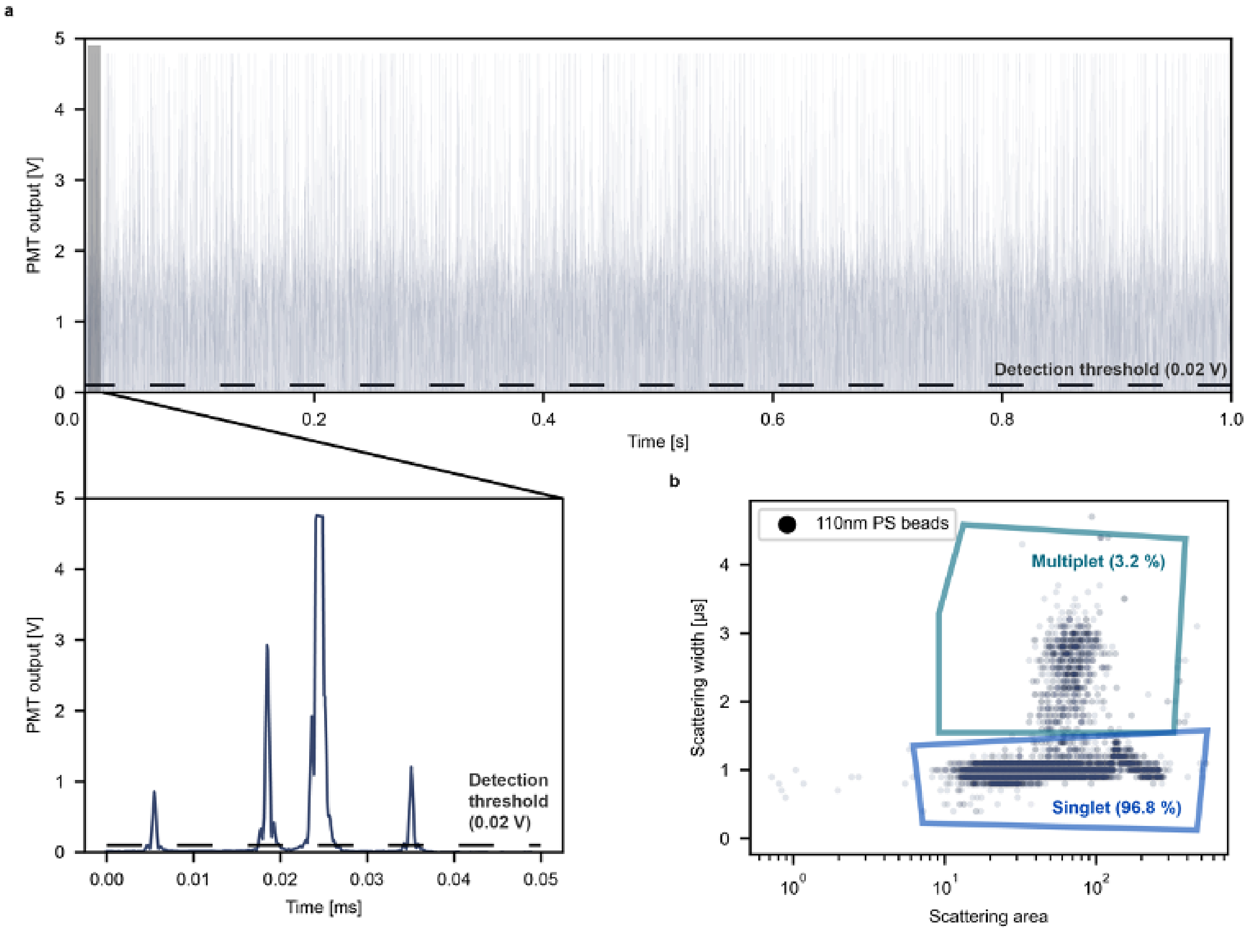

Figure 9.1: **Detection of 110nm polystyrene (PS) beads by DNM (a)** Time series of scattering signals of the PS beads measured for 1 second. The zoomed-up panel shows the time series scattering signal of the beads measured for 0.05 ms. **(b)** The scatter plot of scattering width and scattering area of the beads, obtained by applying a constant trigger threshold to the time series. We identified the singlet and multiplet detections by gating this plot, where we call the detection of individual particles a "singlet" detection and the simultaneous detection of two or more particles or aggregates of particles a "multiplet" detection. From this analysis, we obtained a detection throughput of the DNM as 241,510 events for one second, of which 96.8% were singlets and 0.6% were multiplets (n = 241,510).

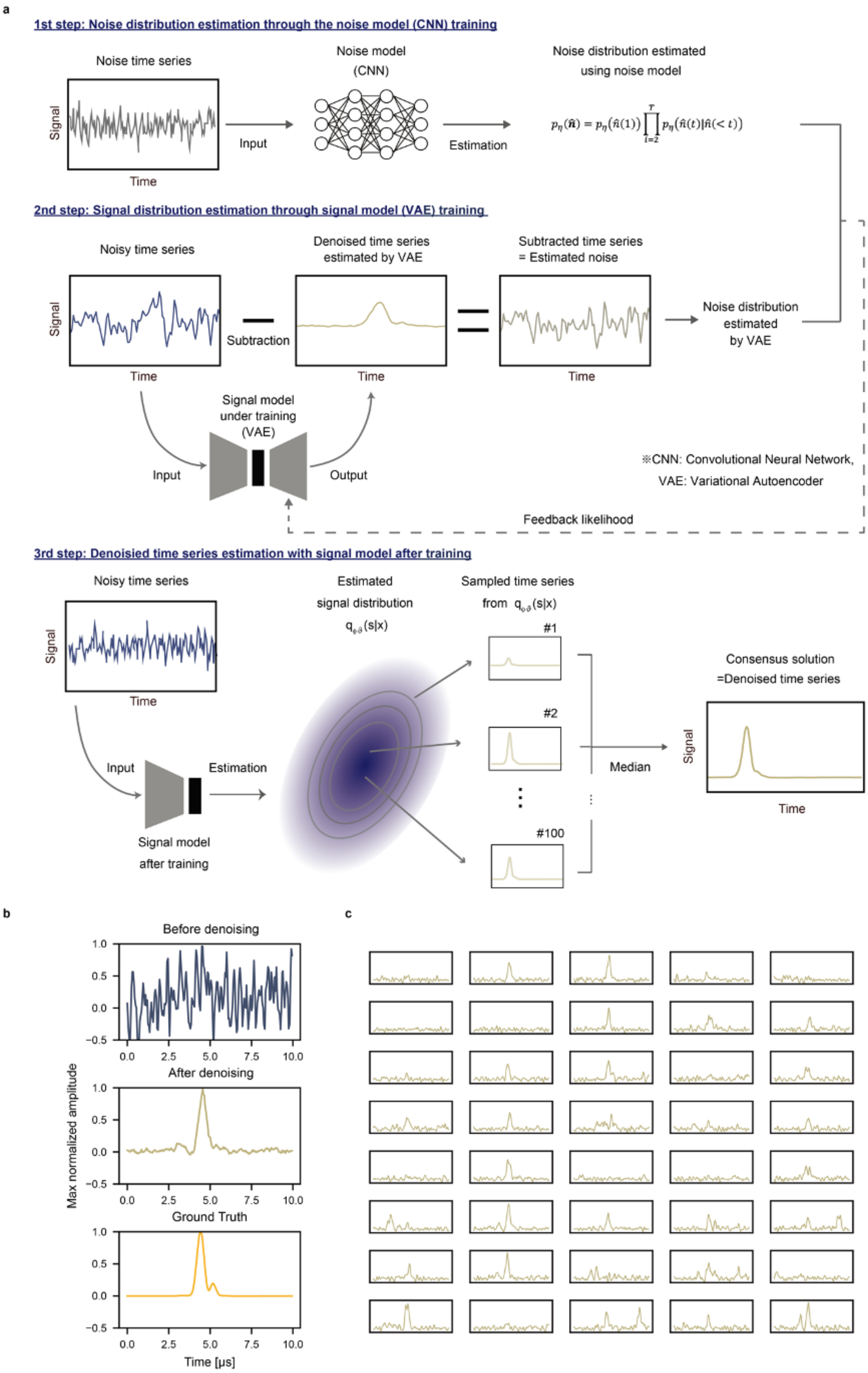

Figure 9.2: See next page for caption.

Figure 9.2: **Schematic of our unsupervised denoising process. (a)** Training and denoising scheme. In the first step, the signal-independent noise distribution $p(\mathbf{n})$ is estimated from the particle-free time series using an autoregressive CNN. Next, we train to estimate the clean signal distribution from the noisy time series containing particles using a VAE. At each training step, a noisy time series is fed to the VAE and a clean signal estimate is sampled. A corresponding noise estimate is then obtained by subtracting the clean signal estimate from the original noisy time series. The likelihood of this noise estimate is calculated using the pre-trained autoregressive noise model, $p_\eta(\mathbf{n})$. Finally, denoising is performed using the trained signal model. The signal distribution is estimated from the noisy time series, 100 denoised time series are randomly sampled from this distribution, and the median value is used as the denoising result (for more details, see Methods, Supplementary Note 1). **(b)** Simultaneously acquired time-series scattering signals derived from 110 nm beads, attenuated signals, and the denoised signal. **(c)** 40 time series out of 100 time series sampled from the signal distribution used to calculate the denoising results in **(b)**.

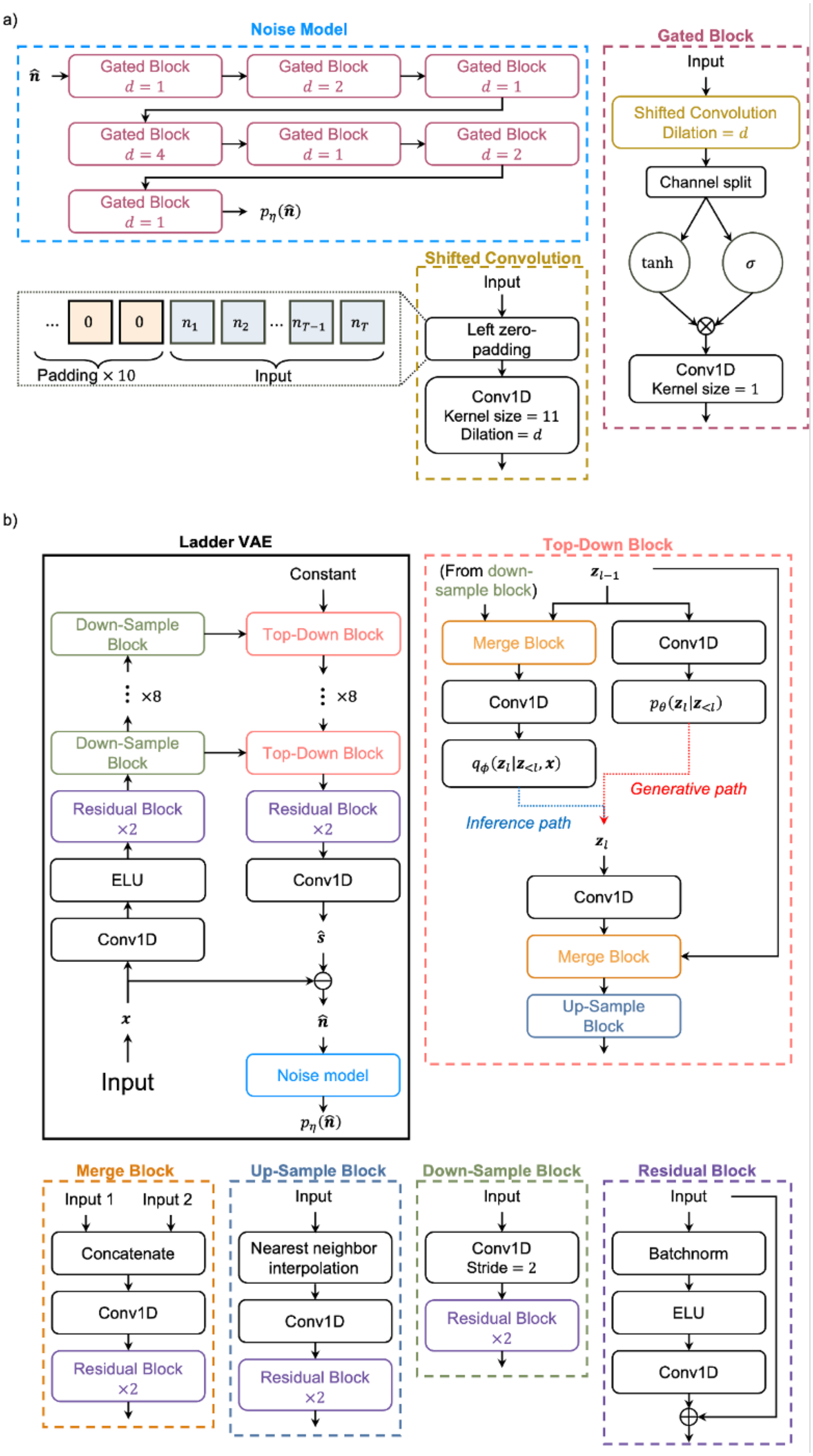

Figure 9.3: See next page for caption.

Figure 9.3: **The architecture of DNM (a) Noise model architecture.** The noise model is based on the PixelCNN architecture [82], adapted to one-dimensional data by using one-dimensional convolutions. The Shifted Convolution module enforces the autoregressive structure of the noise model. It pads the start of the input array with $k - 1$ zeros, where k is the size of the convolutional kernel. This means that the tth element of the output array is a function of the first t elements in the input array. The exception to this is in the first Shifted Convolution of the noise model, where the input is padded with k zeros, so that the tth element of the output array is a function of the first $t - 1$ elements in the input array. In the Channel Split, the input array is divided in two along the channel axis. The hyperbolic tangent and sigmoid functions are represented by $\tanh$ and σ respectively. $\bigotimes$ represents element-wise multiplication. **(b) Hierarchical VAE architecture.** This follows the architecture used by Prakash *et al*. [4]. The constant fed into the first top-down block is an array of zeros with the same dimensions as the data. The probability distributions $q_\phi(\mathbf{z}_l|\mathbf{z}_{<l}, \mathbf{x})$ and $p_\theta(\mathbf{z}_l|\mathbf{z}_{<l})$ are Gaussians. Their means are the first half of channels of the block's input and their standard deviations are the second half, with positivity enforced by a softplus function. All convolutions have a kernel size of 3. $\oplus$ and $\ominus$ represent element-wise addition and subtraction respectively. The concatenate module denotes concatenating arrays along the channel axis. ELU represents the Exponential Linear Unit activation function [139].

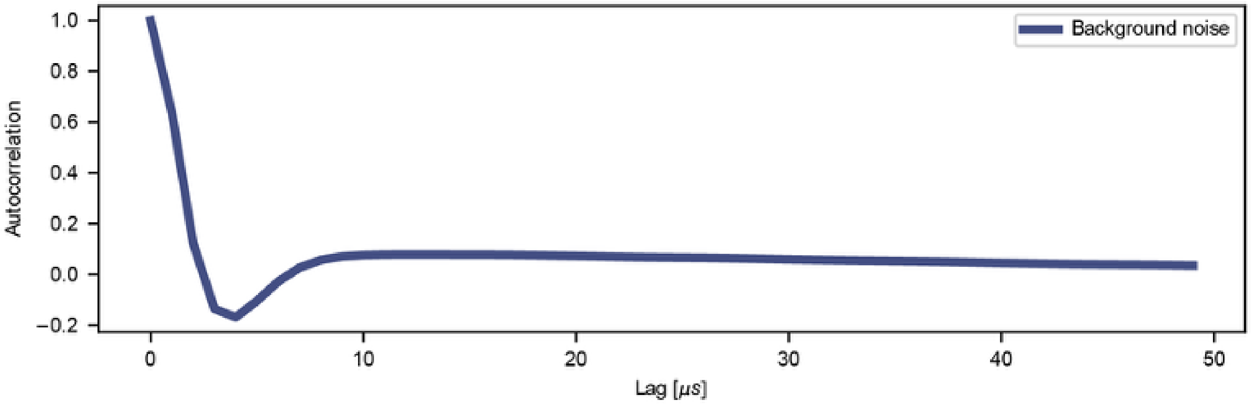

Figure 9.4: **Autocorrelation characteristics of the time-series of background noise.** Autocorrelation plot showing how the noise at each time point is correlated with the noise at previous time points.

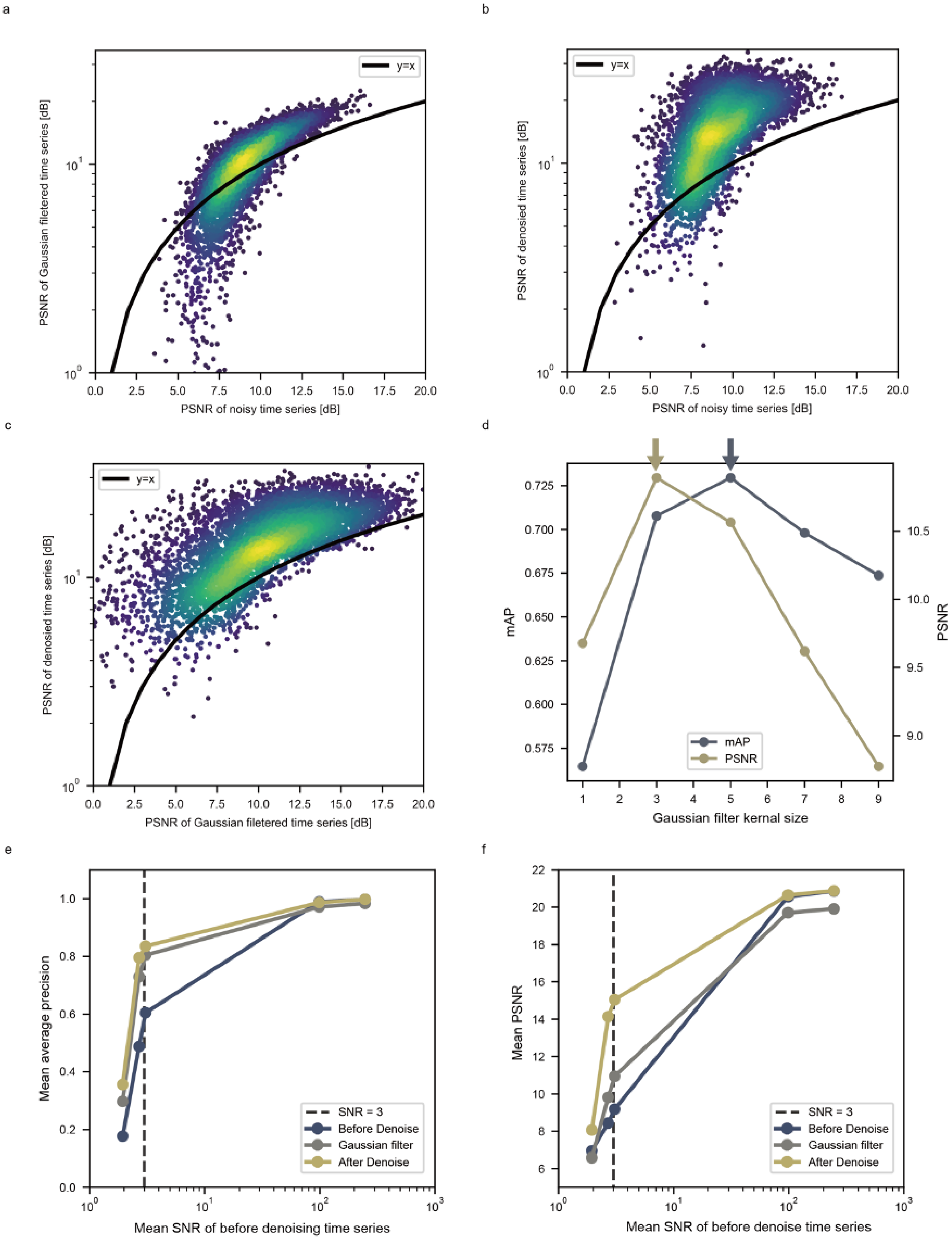

Figure 9.5: See next page for caption.

Figure 9.5: **Performances on the particle detection accuracy and the signal retrieval from noise with varying SNRs.** **(a)** Relationship between the PSNR of Gaussian filtered time series and the PSNR before denoising (n = 5,010). **(b)** Relationship between the PSNR of denoised time series and the PSNR before denoising. **(c)** Relationship between the PSNR of denoised time series and the PSNR of Gaussian filtered time series. **(d)** mAP and PSNR, obtained with the different kernel size of the Gaussian filter. The arrows indicate the highest mAP and PSNR obtained. **(e)** mAP, with error bars representing the standard errors, for different SNRs obtained from different attenuation rates. **(f)** Mean PSNRs, with error bars representing the standard errors, for different SNRs obtained from different attenuation rates.

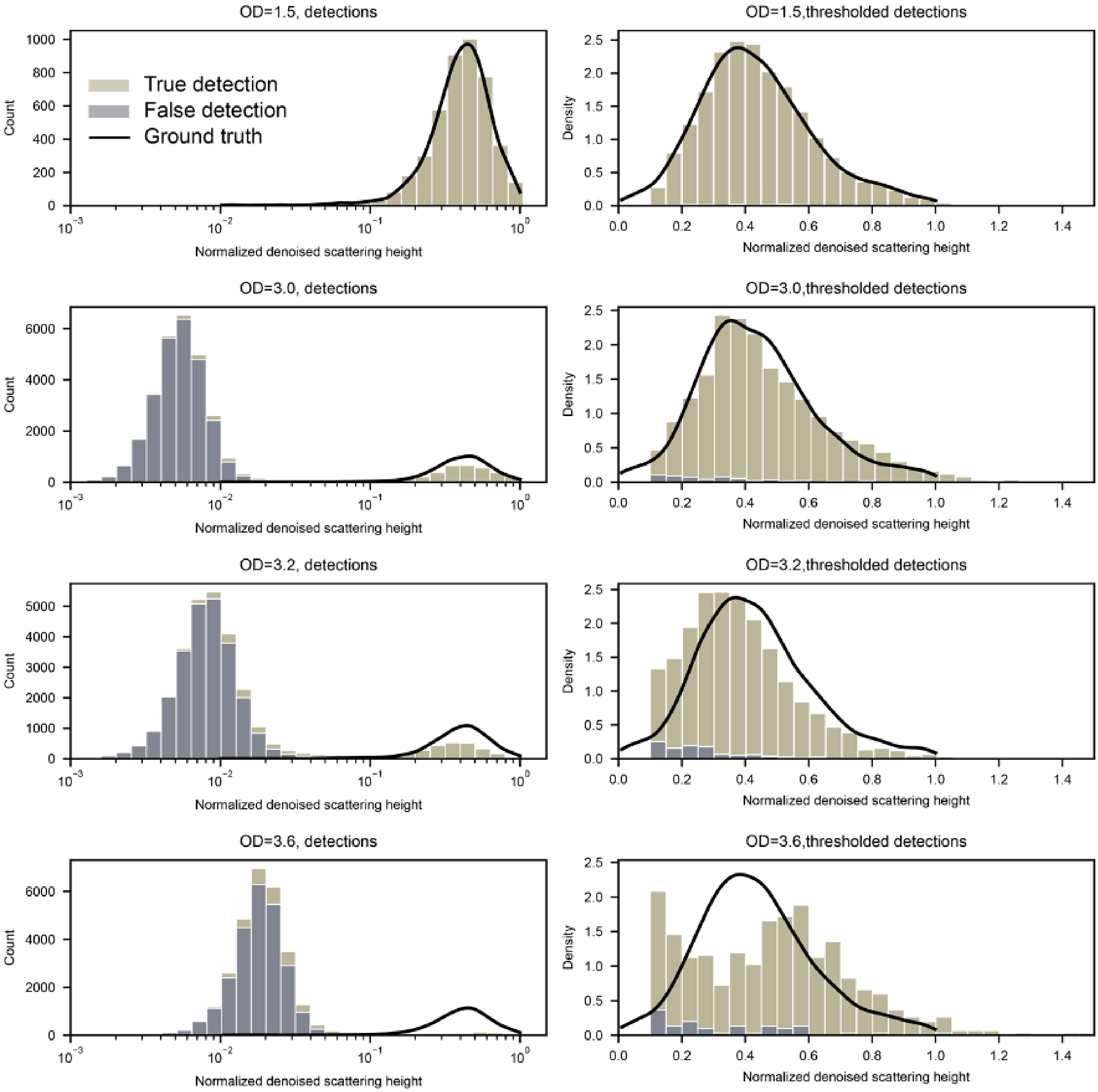

Figure 9.6: See next page for caption.

Figure 9.6: **Histograms of normalised denoised scattering peak heights at optical densities (OD) of 1.5, 3, 3.2, and 3.6, overlaid with a kernel density estimate (KDE) plot of the ground truth.** We consider peaks with a height >0.01 in the ground truth series true detections (olive) and all others false detections (blue). A black line represents the distribution of detections in the ground truth series. **Left panels** show stacked histograms with the x-axis on a logarithmic scale. We observe that the two distributions match well for attenuation OD 1.5 (median SNR of 107.77 before denoising), indicating that the denoising process preserves the scatter amplitude distribution under this condition. For the attenuation OD of 3.0 and 3.2 (median SNR of 3.05 and 2.97 before denoising, respectively) the distribution exhibits a bimodal pattern. The right mode corresponds to peaks that have been preserved by denoising and contains mostly true detections. The left mode corresponds to peaks that have been flattened by the denoiser and contains mostly false detections. With increasing OD, we observe that more and more true peaks are lost (incorrectly flattened). **Right panels** show a thresholded version of the data, rejecting detections below 0.1, as it may be done in a real applied setting to remove false detections. We show normalised probability density estimates with a linear scale on the x-axis. The result confirms that the general shapes of the distribution, which an experimenter without access to ground truth would measure, are well preserved at least for OD 1.5, 3.0, and 3.2 and less well preserved for the most extreme OD 3.6 (median SNR of 1.73).

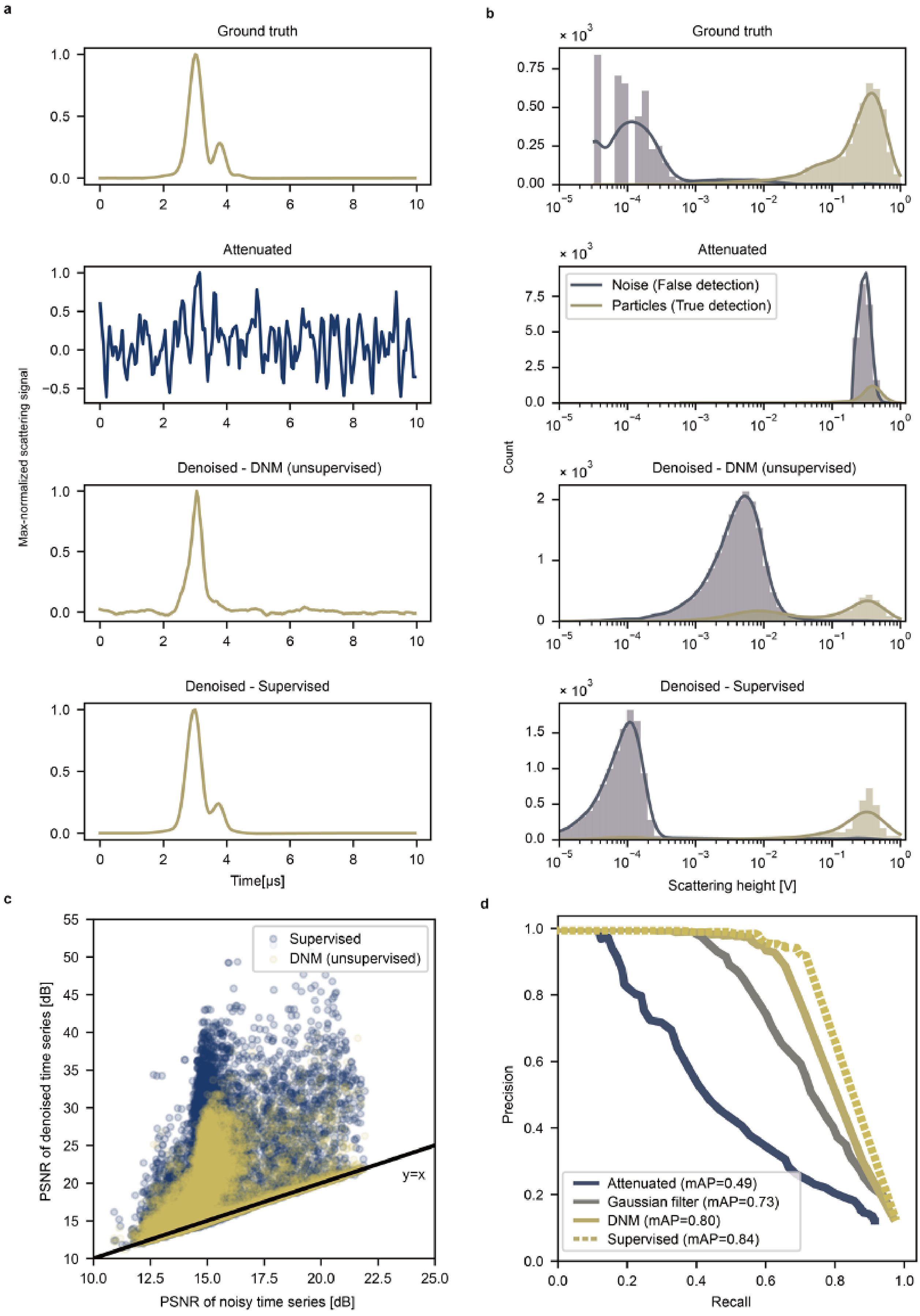

Figure 9.7: See next page for caption.

Figure 9.7: **Comparison of the supervised and unsupervised denoising results. (a)** Acquired max-normalised time-series scattering signals derived from 110 nm beads. The ground-truth signal, the attenuated signal before denoising, the attenuated signal after denoising by the unsupervised denoiser (DNM), and the attenuated signal after denoising by the supervised denoiser, respectively. **(b)** Histograms of the max-normalised scattering peak heights. In the ground truth times series, peak heights exceeding a threshold of 0.01 are considered true nanoparticles, and those lower than the threshold are noise. In the attenuated time series, peaks with a height exceeding 0.05 are considered detections (positives), and only those that coincide with the positions of the ground truth peaks (true nanoparticles) are considered correct detections (true positives). In the denoised time series, peaks at the same location as the positive peaks in the attenuated time series are considered as detections. **(c)** Scatter plot comparing PSNRs before and after supervised and unsupervised (DNM) denoising illustrates the quality of the time series relative to the ground truth series. Points above the line of y = x show improved PSNR, indicating an improved ability to detect particles. In contrast, points below this line represent a decrease in PSNR, suggesting that the particle signals are erroneously treated as noise and flattened. **(d)** Precision and recall (PR) curve for evaluating precision and recall of peak detection in time series denoised by the supervised model, added to Fig. 5.2(d) in the main text as a dashed olive line. The mAP is the area under the PR curve.

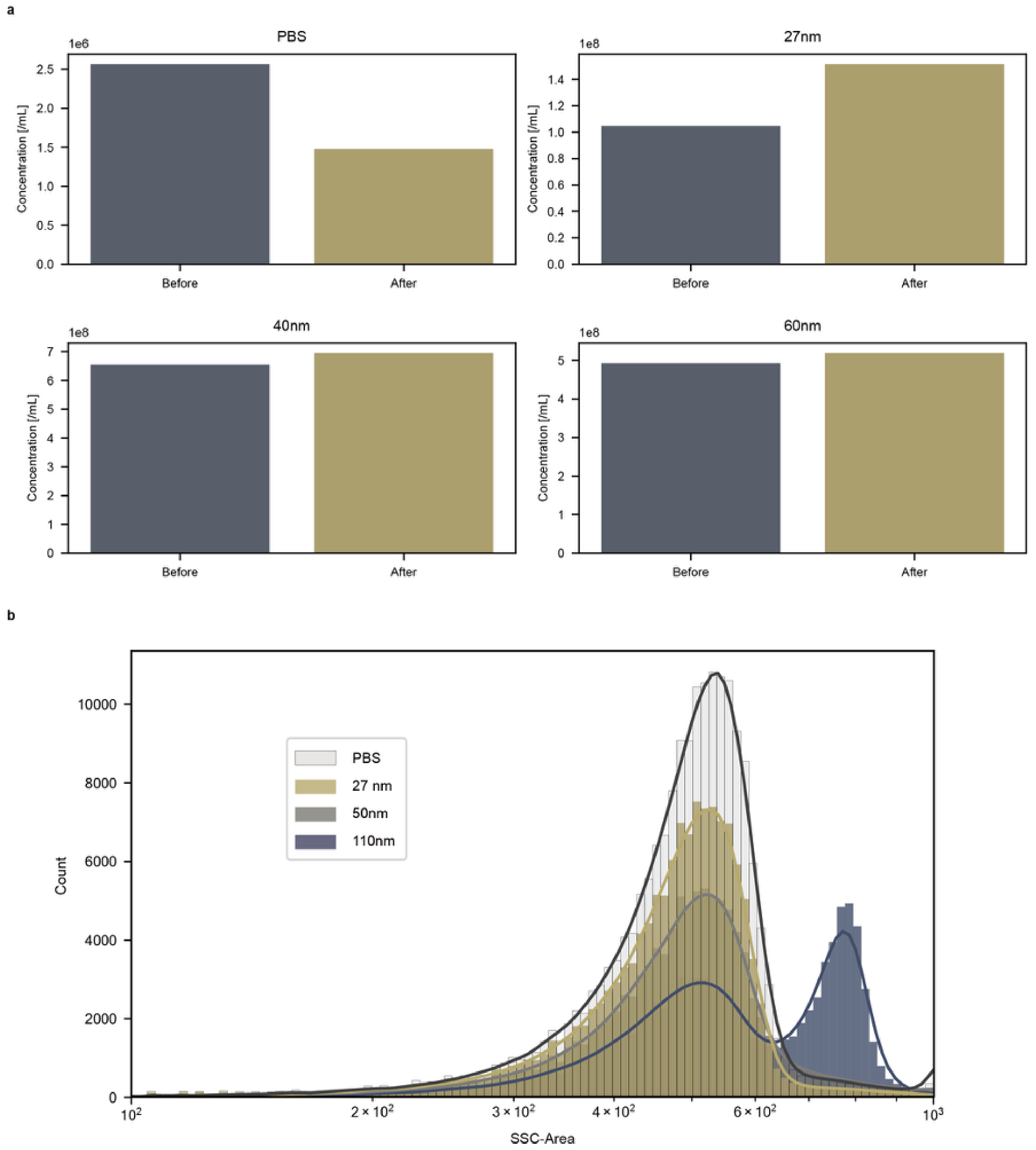

Figure 9.8: **Comparative analysis of the particle concentration changes observed before and after denoising. (a)** Variations in the detected particle concentration by size before and after denoising (blue: before denoising; olive: after denoising). The decrease in the number of detections at PBS and the increase in the number of detections at 27 nm suggest a decrease in false detection and an increase in true detection by DNM. **(b)** Nanoparticle scattering measurements obtained using the conventional flow cytometer (BD FACS Aria) (color: particle size). All but the 110 nm particles overlap the scattering area of the PBS and are not detected.

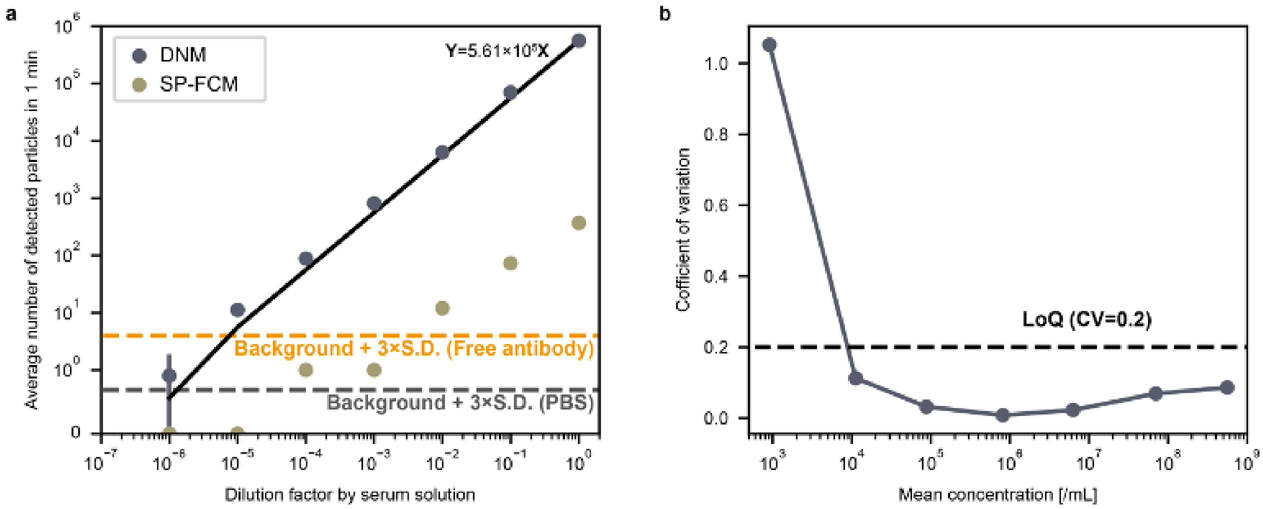

Figure 9.9: **Statistical analysis of scattering- and fluorescence-based detection of EVs subjected to stepwise dilution.** **(a)** Scatter plot of the average detection throughput of CD9(+) EVs and the factor of diluting them with unstained serum (10^{-1} to 10^{-6}). The y-axis is symmetric log scale. The blue and olive colors indicate the result of 3-minute DNM and 1-minute SP-FCM measurements, respectively. The black line indicates the linear regression result produced for DNM measurements (n = 3, mean ± S.D., technical replicate). The black and orange dashed horizontal lines represent the PBS background mean ± 3×S.D. and the free antibody background mean ± 3×S.D., respectively. Sample flow rate of DNM is 1 μL / min and that of SP-FCM is 3.2 nL/min. **(b)** Coefficient of variation (CV) for each estimated concentration. The dashed line denotes CV=0.2.

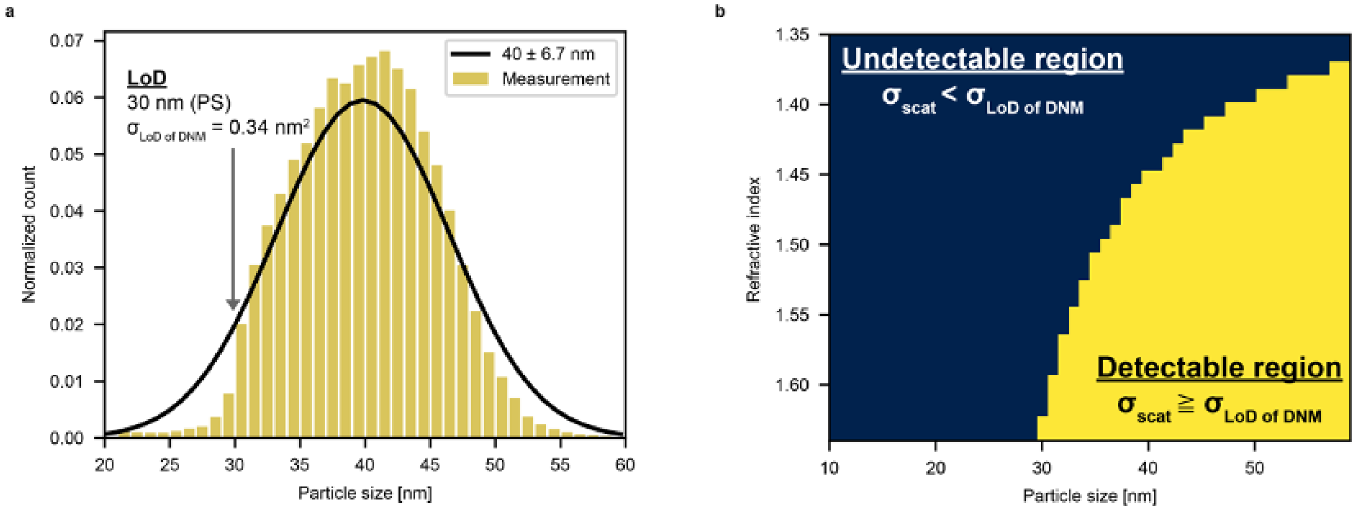

Figure 9.10: **The estimation of the limit of scattering detection (LoD). (a)** Comparison of the size distribution using the denoised data of 40 nm ± 6.7 nm PS beads (Thermo Fisher, F8795, n = 84,392) for estimating the LoD. First, we define the detection threshold value below which the 99.9% of 1-second PBS scattering time series values are included. Learning from the previous research [183], we compare the experimentally estimated size distribution of PS beads with that estimated based on the spec sheet provided by a manufacturer (as a ground truth distribution). To estimate the experimental size distribution, we converted the scatter height to particle size using Rayleigh scattering theory [121] and fitting (Supplementary Note 4). Herein, we normalized the experimentally measured histogram by dividing the counts by double the number of particles detected as 40 nm or larger. We defined the LoD as the smallest size at which the particle count ratio does not fall below 50% [184, 185, 186], and thereby obtained the LoD of 30 nm ($\sigma_{\text{LoD of DNM}} = 0.34$ nm^2). **(b)** Detectable size and refractive index range calculated based on scattering cross section of LoD. We calculated the particle size and refractive index combinations that are detectable by DNM based on the obtained $\sigma_{\text{LoD of DNM}} = 0.34$ nm^2). We calculated scattering cross sections for each particle size and refractive index pair based on Eq. (9.1), and determined the region where the cross sections are smaller than $\sigma_{\text{LoD of DNM}}$ undetectable (blue), and the region where the cross sections are larger detectable (yellow).

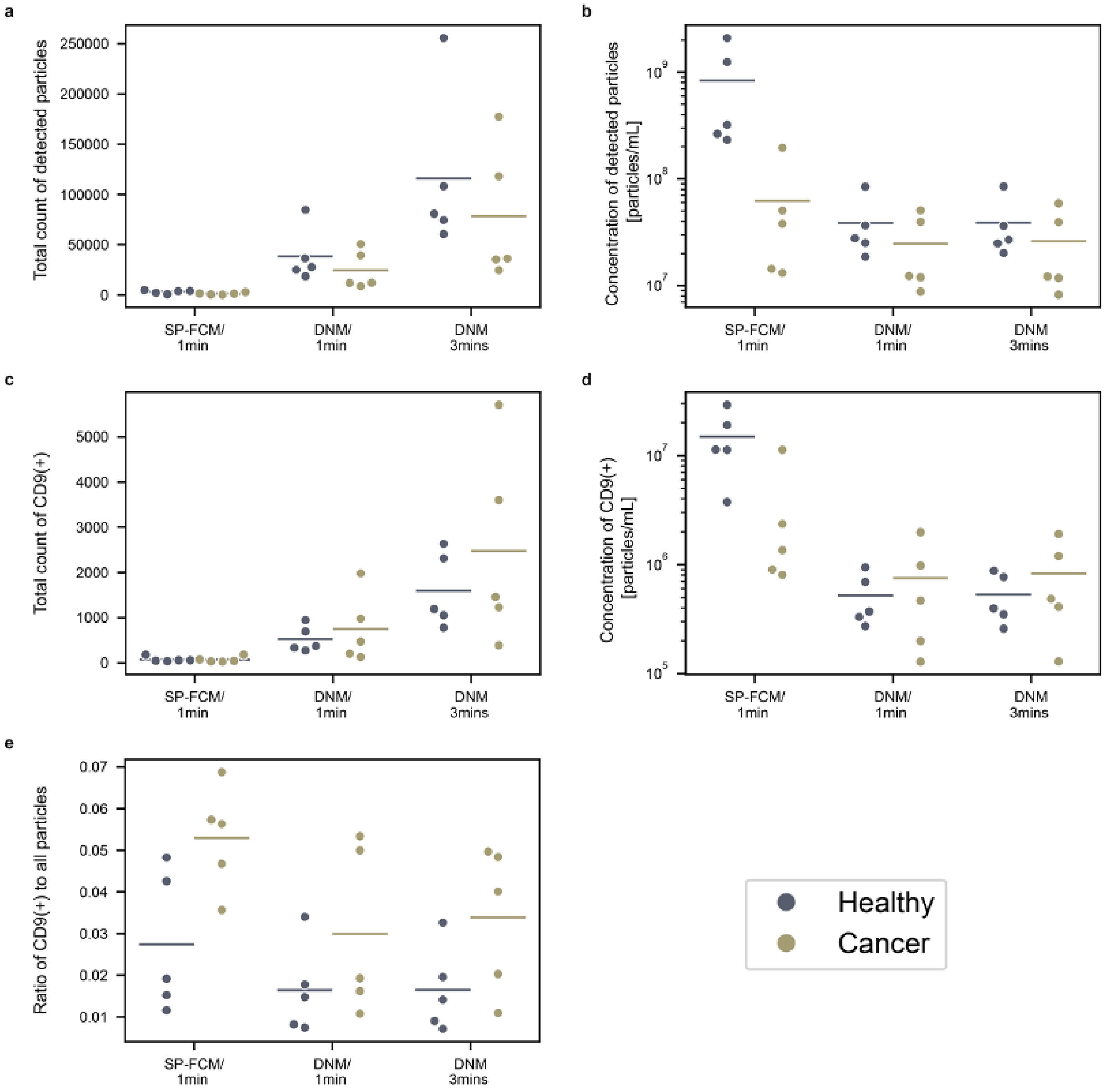

Figure 9.11: **Detection of EVs from cancer patients and healthy donors for colorectal cancer diagnosis.** We performed SP-FCM measurement for 1 minute and DNM measurement for 1 minute and 3 minutes, respectively for each sample (n = 5 for cancer patients, n = 5 for healthy controls, biological replicates). Plots of **(a)** the total count of detected particles, **(b)** the concentration of detected particles (/mL), **(c)** the total count of CD9-positive EVs, **(d)** the concentration of CD9-positive EVs (/mL), and **(e)** the ratio of CD9-positive particles in all detected particles.

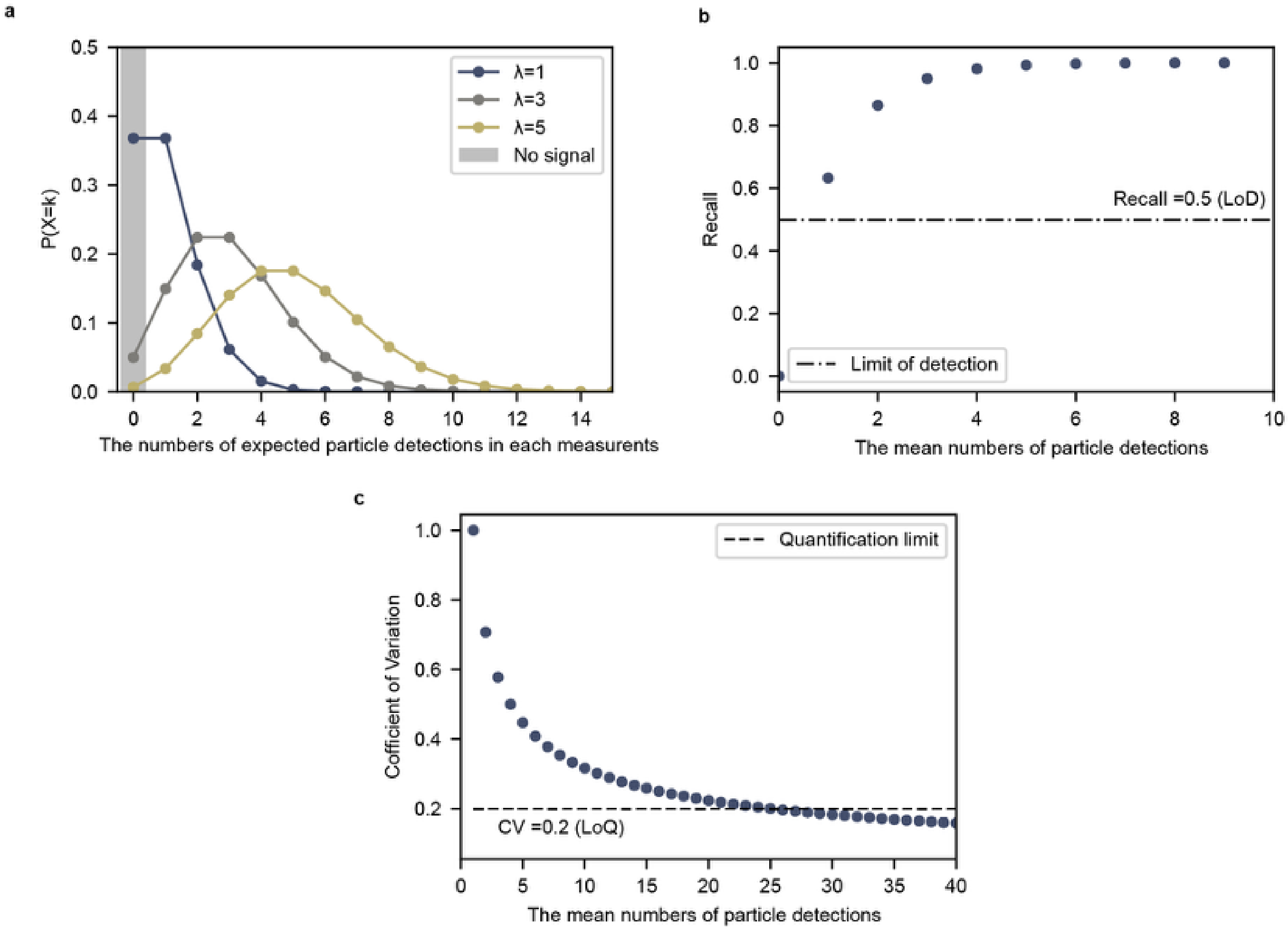

Figure 9.12: **Statistical analysis of the limit of detection (LoD) and quantification limit (LoQ). (a)** Distribution of the numbers of expected particle detection per measurement for different low-count conditions of λ: the expected mean number of particles per measurement. When the number of particle detection is sufficiently small, their distribution follows a Poisson distribution assuming each λ. When the number of detected particles is sufficiently large, it follows a normal distribution. **(b)** Plot shows the recalls with respect to the mean numbers of particle detections. The dotted line shows the case in which the recall $=$ 0.5, corresponding to the commonly used LoD: the lowest quantity or concentration of a component that can be reliably detected against noise. If the number of particle detections exceeds 0 and that for noise is zero, the number of particle detections measured is above LoD. **(c)** Plot shows the coefficient of variation (CV) with respect to the mean numbers of particle detections. The dotted line shows the case in which the CV $<$ 0.2, corresponding to the commonly used LoQ: the lowest quantity or concentration of a component that can be quantitatively. If the number of particle detections exceeds 24 and that for noise is zero, the number of particle detections measured is above LoQ.

10 UNSUPERVISED DENOISING FOR SIGNAL-DEPENDENT AND ROW-CORRELATED IMAGING NOISE - SUPPLEMENTARY

10.1 Latent variables represent clean images

The training of the signal decoder assumes that every sampled value of the latent variable $\mathbf{z}$ corresponds one clean image, or signal, $\mathbf{s}$. We denote the signal corresponding to a value of $\mathbf{z}$ by $\mathbf{s}(\mathbf{z})$. Using this relationship, the signal decoder, $f_\theta(\mathbf{z})$, can be trained to estimate

$$f_\theta(\mathbf{z}) \approx \mathbb{E}_{p_\theta(\mathbf{x}|\mathbf{z})}[\mathbf{x}] = \mathbb{E}_{p_\theta(\mathbf{x}|\mathbf{s}(\mathbf{z}))}[\mathbf{x}] = \mathbf{s}(\mathbf{z}). \tag{10.1}$$

Here, we provide an another way of viewing this deterministic relationship.

If the latent variables of our model truly represent only signals, then the AR decoder, $p_\theta(\mathbf{x}|\mathbf{z})$, must model only the noise generation process. Therefore, different random samples from the AR decoder for the same value of latent variable will be images with the same underlying signal and different random samples of noise. Since the noise is zero-centred, this allows us to produce an estimate of the signal by calculating the mean of many samples from the AR decoder. That is, if $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_L$ are L random samples from $p_\theta(\mathbf{x}|\mathbf{z})$,

$$\overline{\mathbf{x}} = \frac{1}{L}\sum_{l=1}^{L} \mathbf{x}_l \approx \mathbb{E}_{p_\theta(\mathbf{x}|\mathbf{z})}[\mathbf{x}] = \mathbf{s}(\mathbf{z}). \tag{10.2}$$

In this section, we experimentally verify Eq. (10.1) by estimating the signal underlying a noisy image $\mathbf{x}$ using both techniques; by passing a latent variable sample to the signal decoder and by averaging 10,000 noisy image samples from the AR decoder. If Eq. (10.1) is true, the two estimates of the signal should be nearly identical for the same value of latent variable.

Figure 10.1 shows the result of this experiment for two different random samples from the ap-

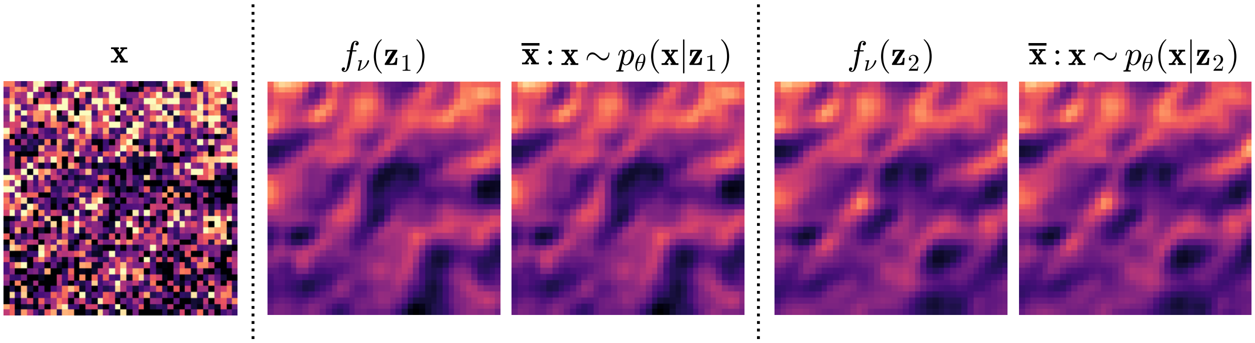
$\mathbf{x}$
$f_\nu(\mathbf{z}_1)$
$\overline{\mathbf{x}} : \mathbf{x} \sim p_\theta(\mathbf{x}|\mathbf{z}_1)$
$f_\nu(\mathbf{z}_2)$
$\overline{\mathbf{x}} : \mathbf{x} \sim p_\theta(\mathbf{x}|\mathbf{z}_2)$

Figure 10.1: Given a noisy image $\mathbf{x}$, we took two samples from the approximate posterior, $\mathbf{z}_1$ and $\mathbf{z}_2$. For each latent variable sample, we produced one estimate of the signal by passing it through the signal decoder f_θ and one by averaging 10,000 samples from the AR decoder, $p_\theta(\mathbf{x}|\mathbf{z})$.

proximate posterior, $\mathbf{z}_1$ and $\mathbf{z}_2$. The two estimates of $\mathbf{s}(\mathbf{z}_1)$ are visually very similar to each other, while exhibiting clear structural differences from the two estimates of $\mathbf{s}(\mathbf{z}_2)$.

10.2 Training and inference

10.2.1 Hyperparameters

Both the main VAE and the signal decoder were trained with an Adamax [137] optimiser with a learning rate of 0.002. Both learning rates decreased by a factor of 10 when the validation loss had plateaued for 50 epochs. The models for all datasets were trained for a maximum of 80,000 steps but stopped if validation loss had plateaued for 100 epochs.

For the non-simulated datasets, training images were randomly cropped to a size of 256×256 at each epoch and a batch size of 16 was used, but this was split into 4 virtual batches. For the *FFHQ - Stripe* and *FFHQ - Checkerboard* datasets, training images were kept at their original resolution of 128×128 and a batch size of 64 was used, but split into 16 virtual batches.

10.3 Hardware and software

The workstation used for this paper's experiments is a 36 core Intel(R) Core(TM) i9-10980XE CPU @ 3.00GHz with 134GB of RAM running Ubuntu 22.04.4, Python 3.11.5 and pytorch-lightning

2.2.1. The GPU used for training is a NVIDIA GeForce RTX 3090 with 24GB of VRAM. For all datasets, training required approximately 20GB of GPU memory with the *large* network and 6GB with the *small* network.

10.3.1 Times

Training a model for 80,000 steps with our hardware takes approximately 24 hours. After training, denoising 100 images of size 512x512 by randomly sampling 100 denoised estimates takes 13 minutes when using a batch size of 10. Increasing batch size made no improvement as the GPU is at 100% utilization.

10.3.2 PSNR of minimum mean square error estimates

Table 10.1: Our denoiser randomly samples clean images for a given noisy image. A consensus solution can be produced by averaging random samples. The PSNR of the mean of increasing sample sizes is shown below.

	EMCCD				**LSCM**	
No. Samples	**Conv. A**	**Conv. B**	**Mouse Actin**	**Mouse Nuclei***	**Actin Conf.**	**Mito Conf.**
1	36.54	42.33	37.43	42.25	26.81	22.62
10	37.39	43.90	39.03	42.91	27.37	23.39
100	37.49	44.10	39.23	42.98	27.42	27.50
1,000	37.50	44.10	39.24	42.99	27.44	27.52

	Simulated	
No. Samples	**FFHQ Stripe**	**FFHQ Checkerb.**
1	32.27	33.03
10	35.24	35.85
100	35.66	36.27
1,000	35.74	36.32

In addition to the quantitative results presented in the paper, the PSNR of the mean of 1, 10 and 1,000 random denoised samples can be found in Sec. 10.3.2.

10.4 Architecture

We proposed two network architectures for denoising, one *large* and one *small*. Each model consists of a hierarchical VAE [55], an autoregressive decoder [82] and our novel *signal decoder*. In the *large* network, the VAE has 14 levels in its hierarchy. The first 13 levels have 64 latent dimensions each, while the final level has 128 dimensions. The latent variable passed to the decoders is sampled from this final level. At each level on both the bottom-up path and the top-down path is a residual block consisting of two sets of a convolution followed by a batch normalization [138] followed by a Mish activation function [187]. Each residual block is followed by a gated block [188]. Resampling is performed at alternating levels. The *small* network is the same except that it has 6 levels to its hierarchy and half the latent dimensions.

The autoregressive decoder is built with eight layers of conditional PixelCNN blocks as proposed in [82], but we found the performance to be better with a ReLU activation function [68] than with gated units. The convolving kernels in the AR decoder have dimensions $1 \times k$, where k is the kernel size. In the first layer of the decoder, the input is padded with k zeros on its left-hand side, and then a convolution is applied. At all subsequent layers, the input features are padded with $k-1$ zeroes on the left-hand side. This results in a row-based autoregressive receptive field. For a column-based receptive field, the kernels have dimensions $k \times 1$ and padding is applied to the top of the input. For all of our experiments, $k = 5$. The convolutions in every other layer have dilated kernels [189] and all have 64 filters. The likelihood distribution is a Gaussian mixture model, with 3 components used for all datasets except the *FFHQ* datasets, for which 10 were used, and the *STEM* dataset, for which 5 were used.

The *signal decoder* is a convolutional neural network consisting of four 3×3 convolutions with 128 filters, each followed by a ReLU activation function.

10.5 Baselines

10.5.1 AP-BSN [1]

These models were trained using the code available at `https://github.com/wooseoklee4/AP-BSN` using hyperparamters detailed in the original publication. We used a stride factor of 5 for all datasets.

10.5.2 Structured Noise2Void [2] and N2V2 [3]

Both these model types were trained using the code available at `https://github.com/juglab/n2v`, using default hyperparameters found in the example notebooks for SN2V and hyperparameter values found in the original publication of N2V2. Following Broaddus *et al*. [2], SN2V masks should be as small as possible while covering pixels with a noise value that is highly predictive of the noise value in the target pixel. A trial and error test of the mask size for each dataset would be too computationally expensive, so we follow [2] and mask 4 pixels on each side of the target pixel for all datasets except *FFHQ - Checkerboard*. The structured component of noise in the *FFHQ - Checkerboard* dataset can theoretically be predicted by seeing only two pixels in the same column, so entire columns were masked here. The orientation of the pixel mask was determined by looking at the spatial autocorrelation in noise patches for each dataset. The *Mouse Nuclei* dataset is corrupted by unstructured noise, so was denoised with a single pixel mask.

10.5.3 HDN [4]

These models were trained using the code available at `https://github.com/juglab/HDN/` using default hyperparameters found in the example notebooks. HDN requires a pre-trained noise model. We followed Prakash *et al*. [4] and modelled the noise in each dataset using a Gaussian mixture model. The noise model parameters can be estimated from the training data of datasets with available ground truth. For datasets without ground truth, we trained the noise model using denoised images from our method as pseudo-ground truth.

10.5.4 CARE [5] and N2N [6]

Both of these model types were trained using the code available at `https://github.com/CSBDeep/CSBDeep`, using default hyperparameters and setting noisy images as target for N2N.

10.6 Simulated data

The Flickr Faces HQ thumbnails dataset [171], with resolution 128×128, was made greyscale by averaging across colour channels. For *FFHQ - Stripe*, the ground truth, $\mathbf{s}$, was scaled to have pixel values between 0 and 1, and Poisson noisy images were created as $\mathbf{x} = 0.002 \times \mathcal{P}(\mathbf{s}/0.002)$. Zero-mean Gaussian noise with a standard deviation of 0.02 was then added to these images. Finally, structured noise was created by applying a horizontal Gaussian blur with a standard deviation of 1 to white Gaussian noise with a standard deviation of 0.025 and added on top. For *FFHQ - Checkerboard*, we added noise with inverse signal dependence by sampling Gaussian noise from the distribution $\mathcal{N}(0, 0.15 \times 1/\mathbf{s})$. Then a vertical checkerboard pattern was added by subtracting 0.1 from two pixels and adding 0.1 to the next two pixels along columns. The starting point for the checkerboard was randomly sampled from a uniform distribution. For both *FFHQ* datasets, the final 1,000 images were designated as a test set.

10.7 Additional qualitative results

See overleaf for larger denoised images from each dataset.

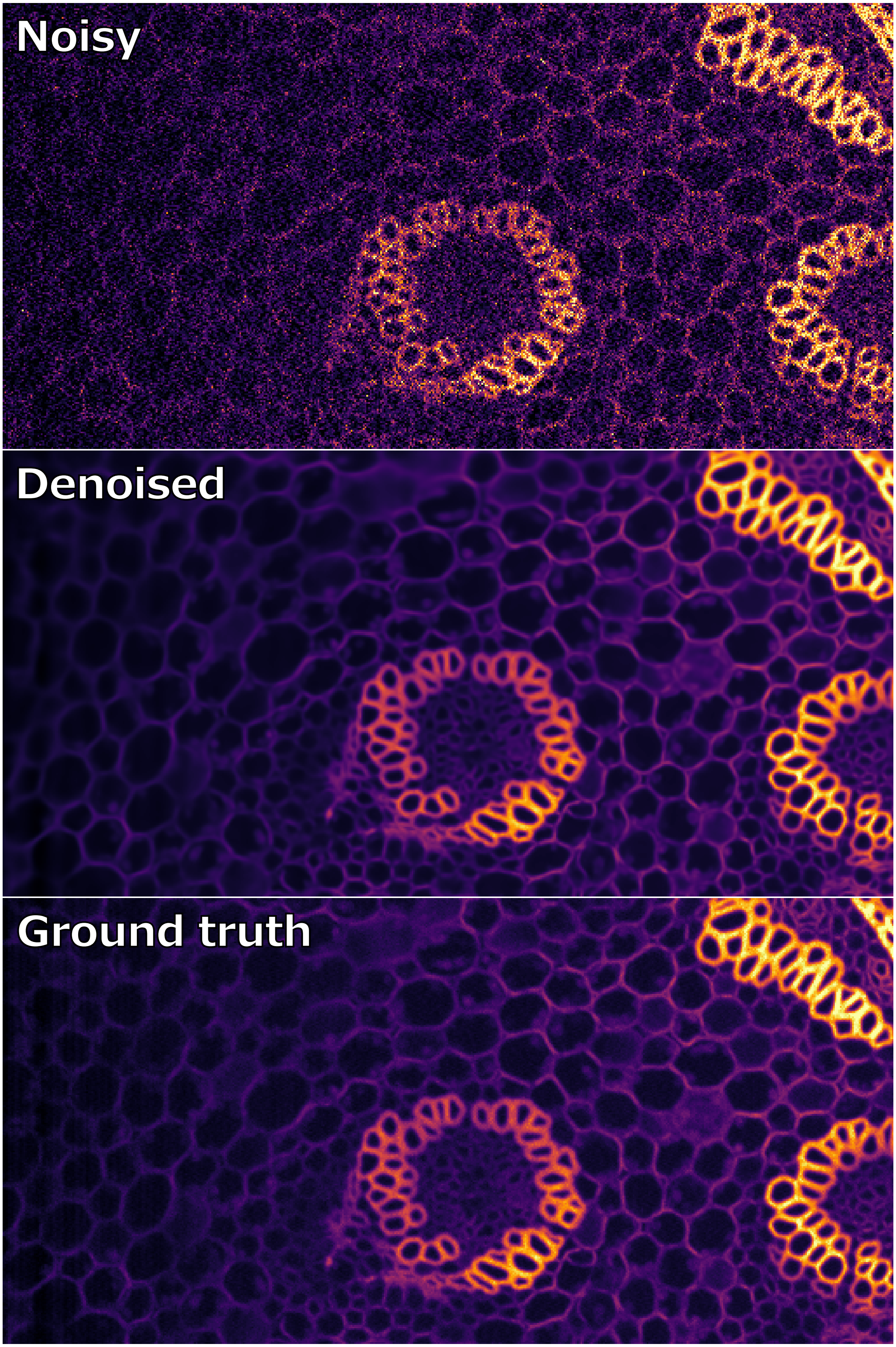

Figure 10.2: *Convallaria A*

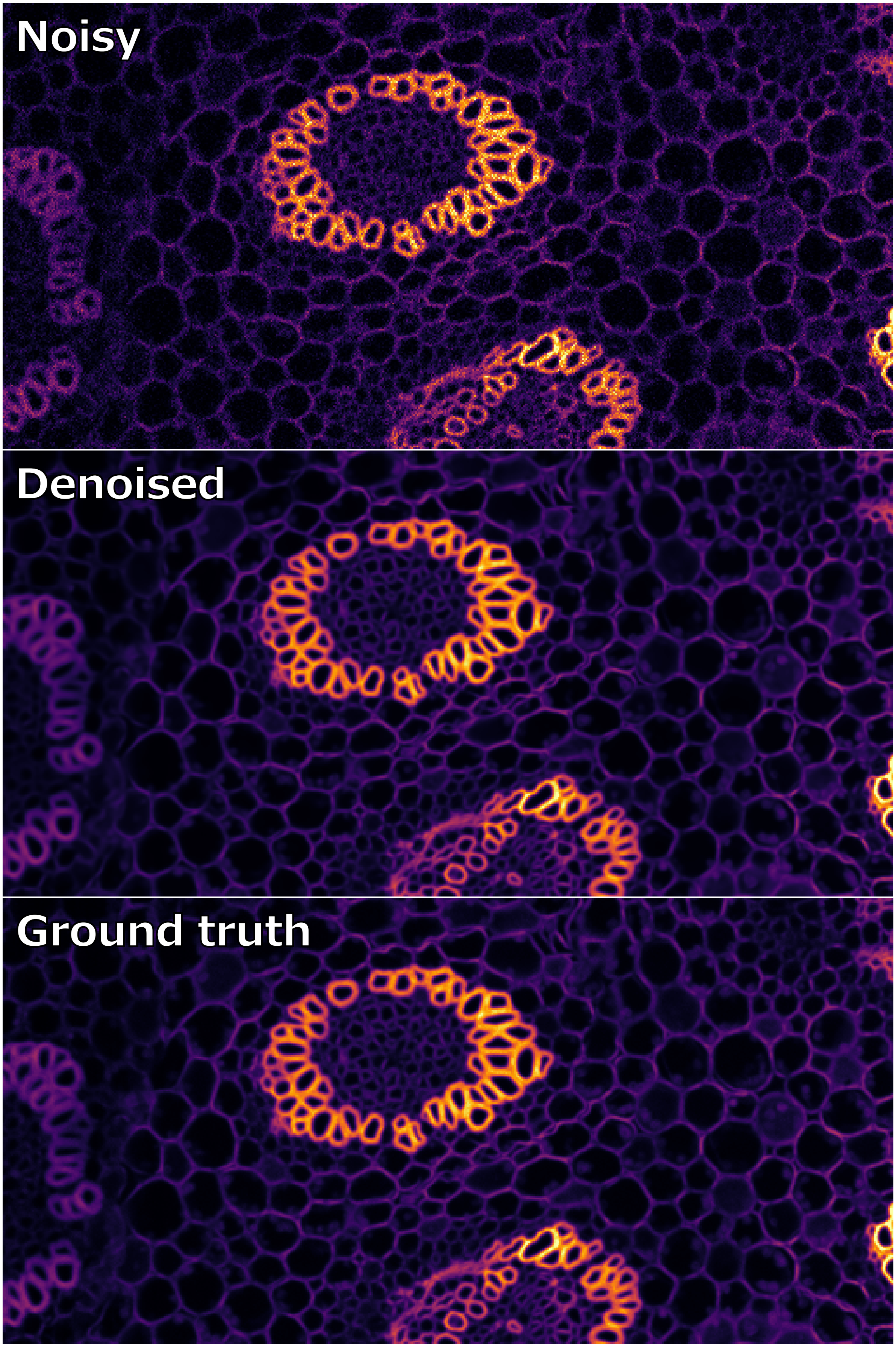

Figure 10.3: *Convallaria B*

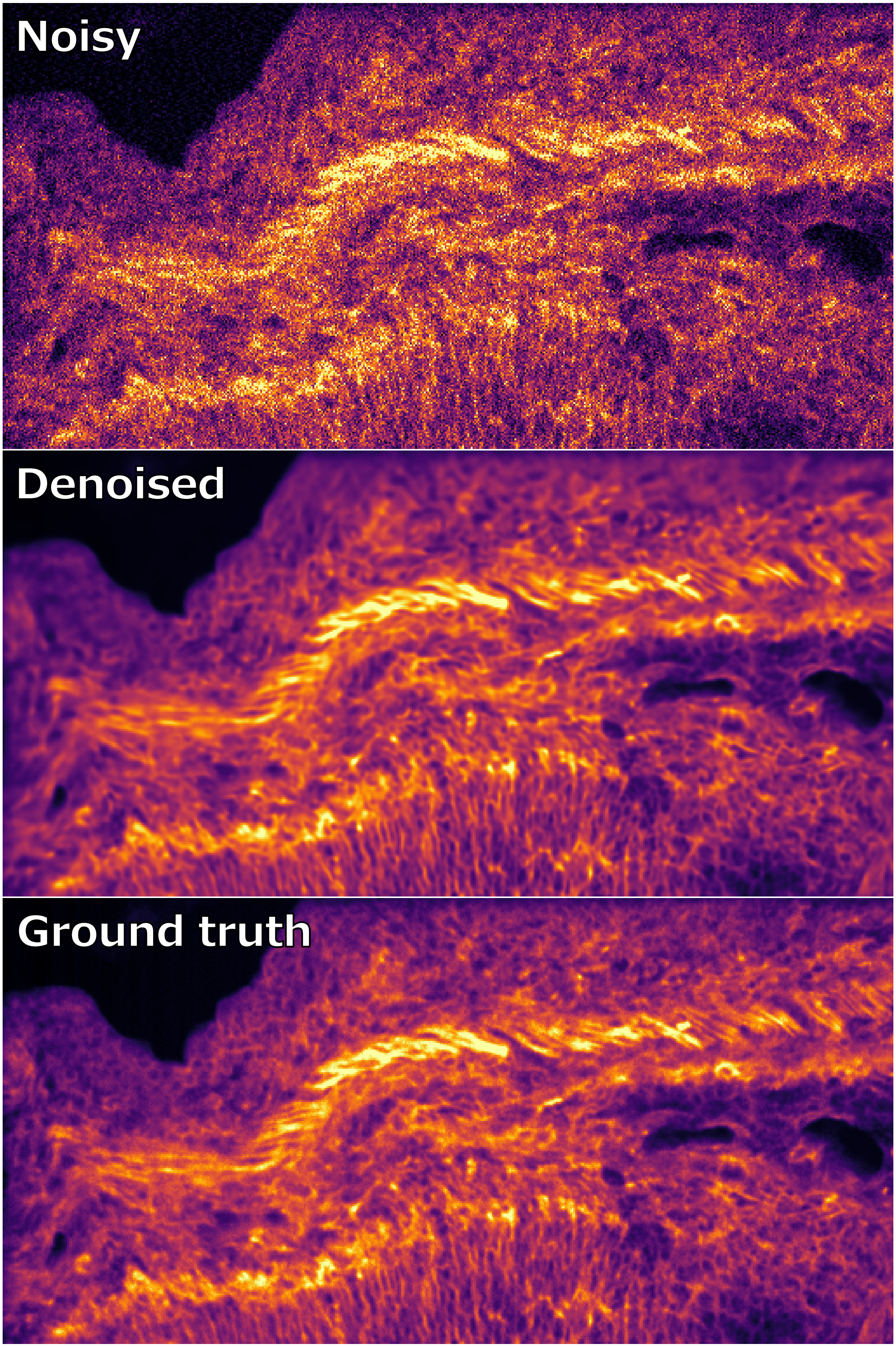

Figure 10.4: *Mouse Actin*

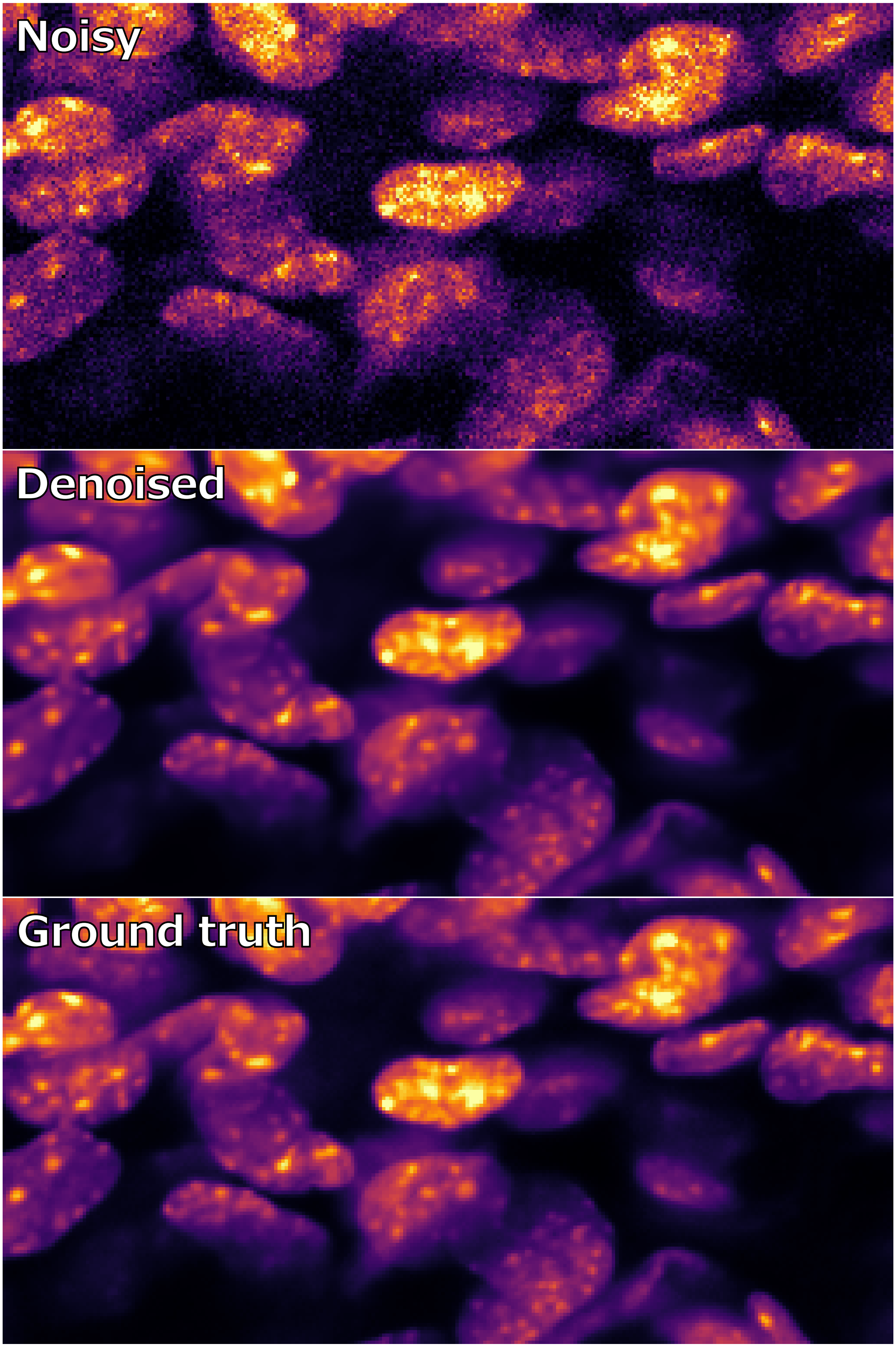

Figure 10.5: *Mouse Nuclei*

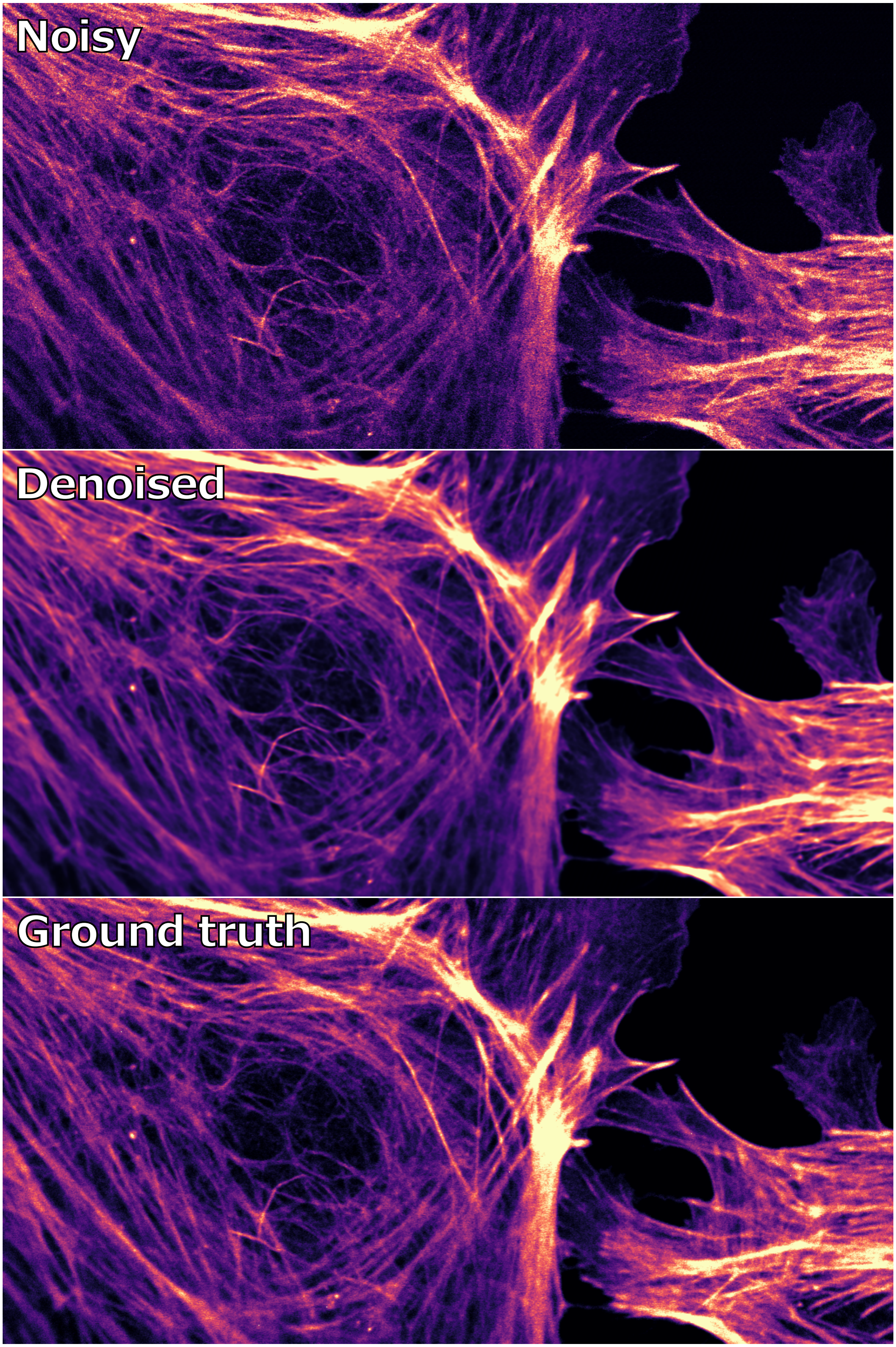

Figure 10.6: *Actin Confocal*

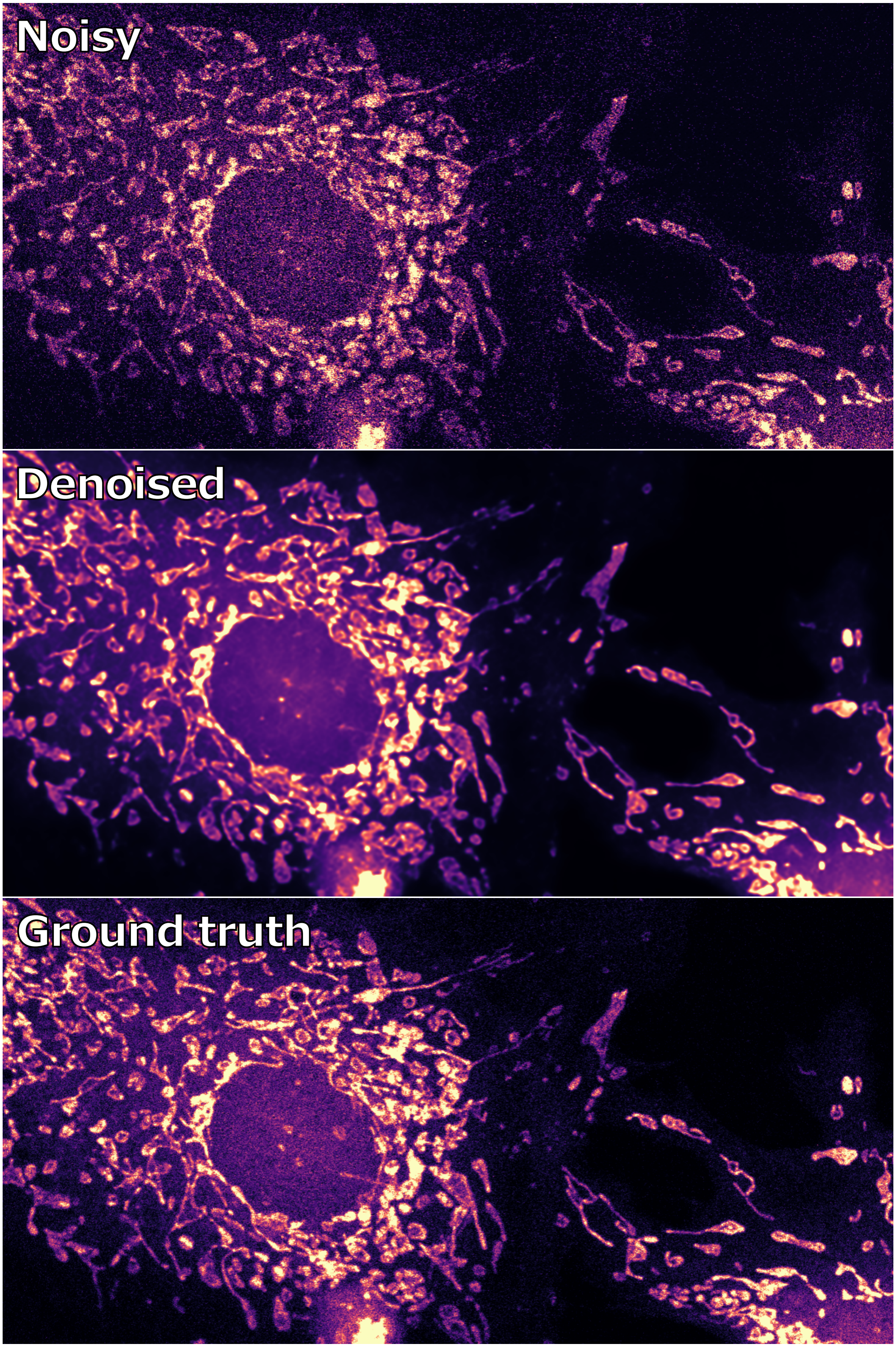

Figure 10.7: *Mito Confocal*

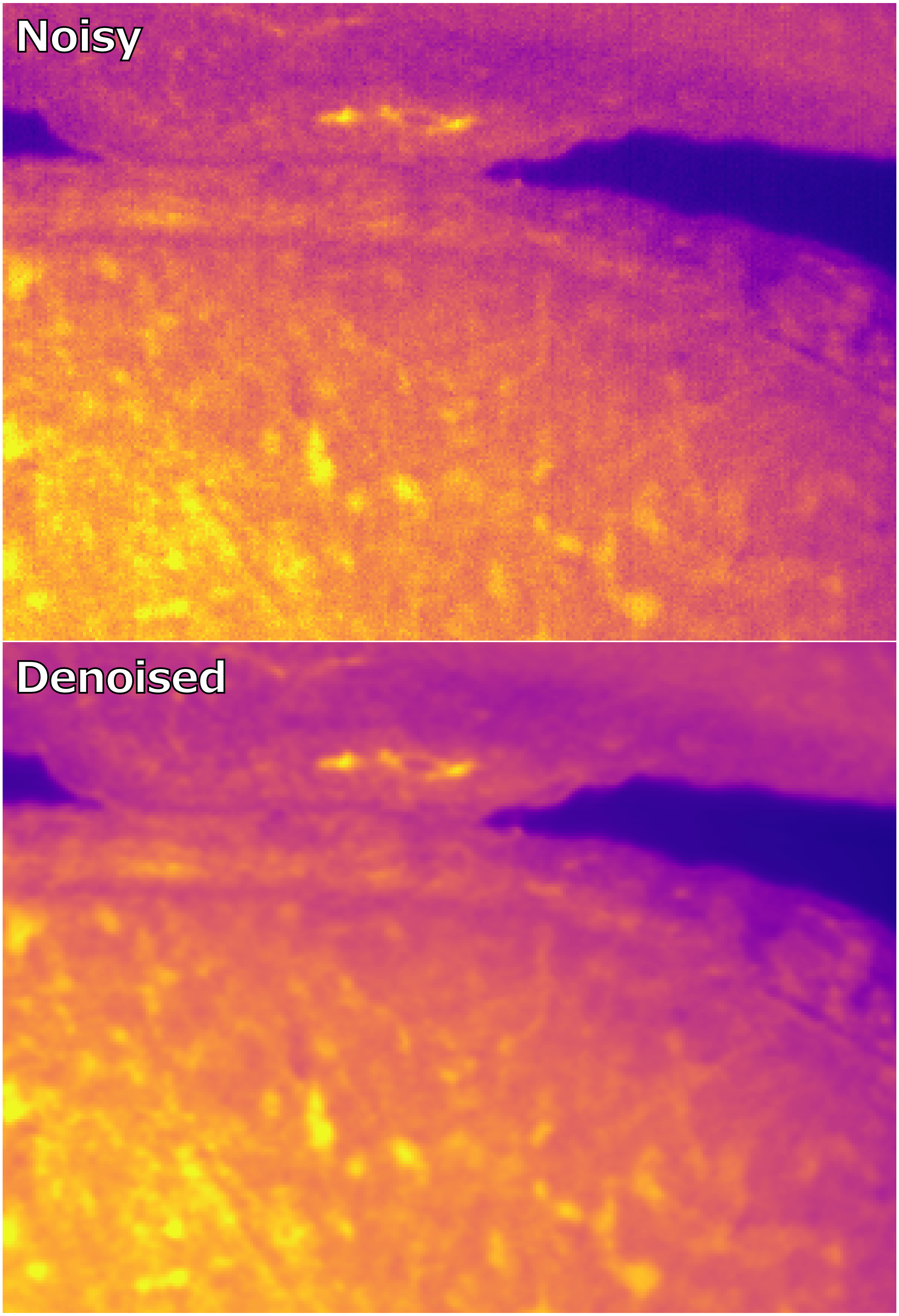

Figure 10.8: *Embryo*

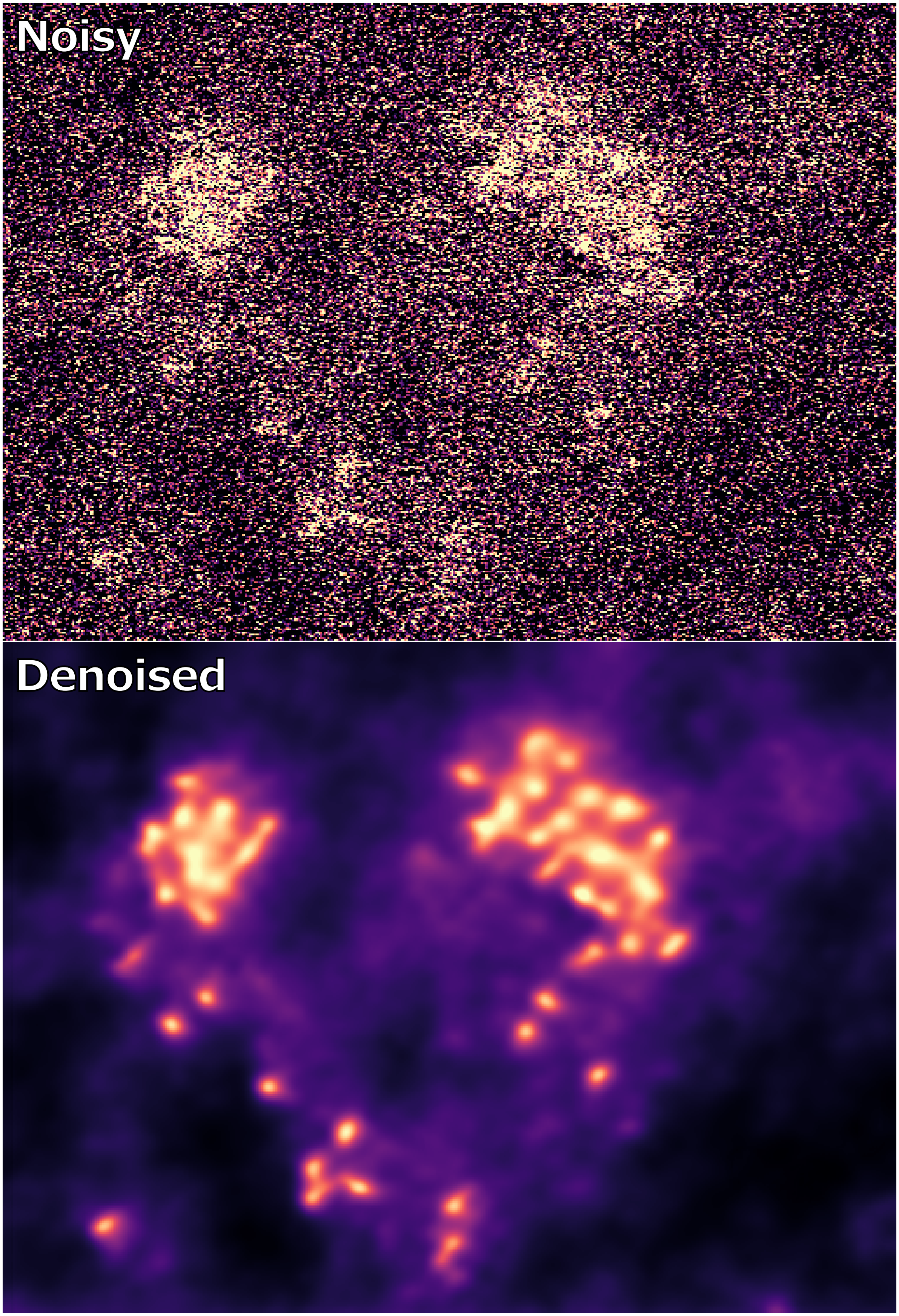

Figure 10.9: *STEM*

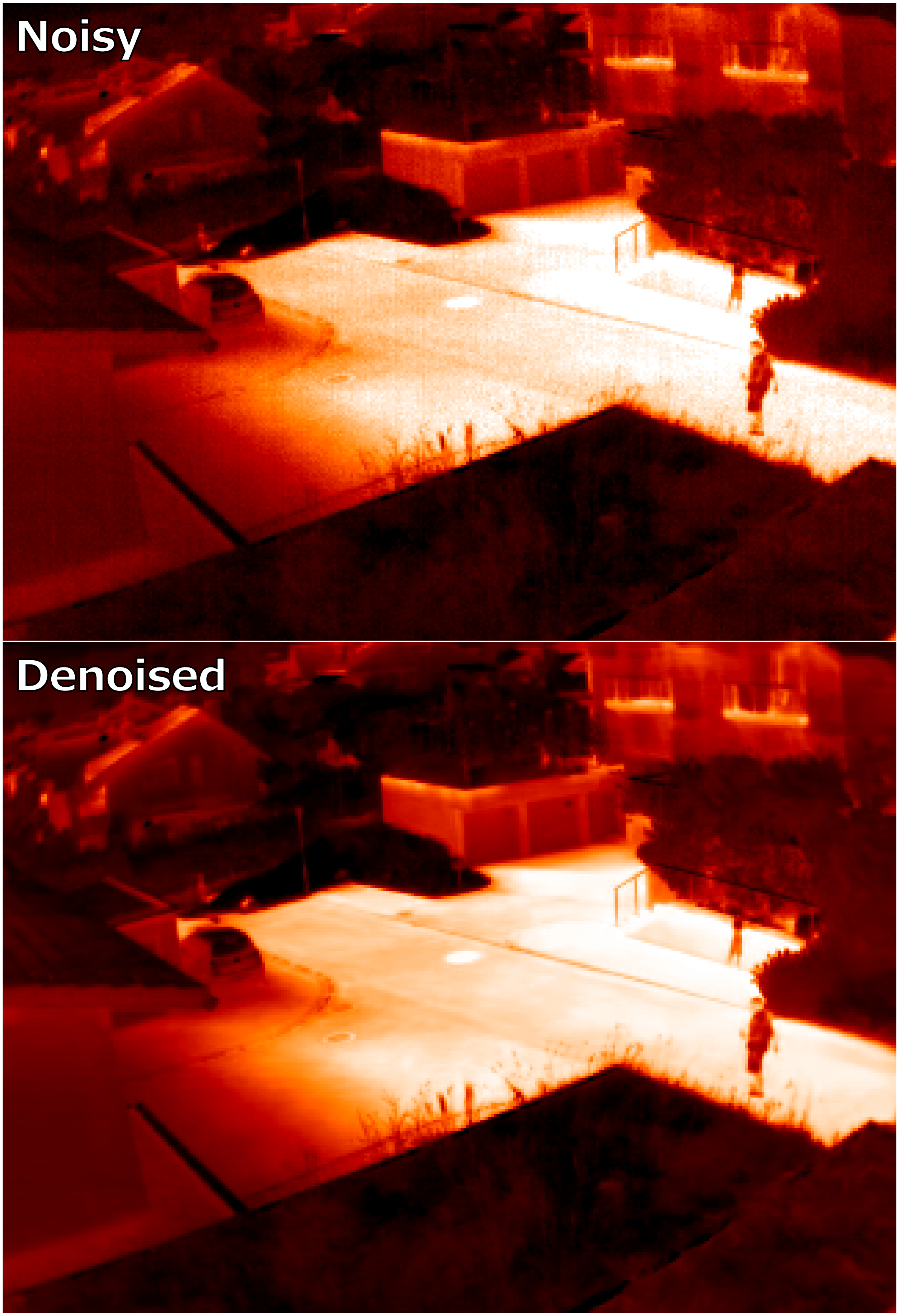

Figure 10.10: *IR*

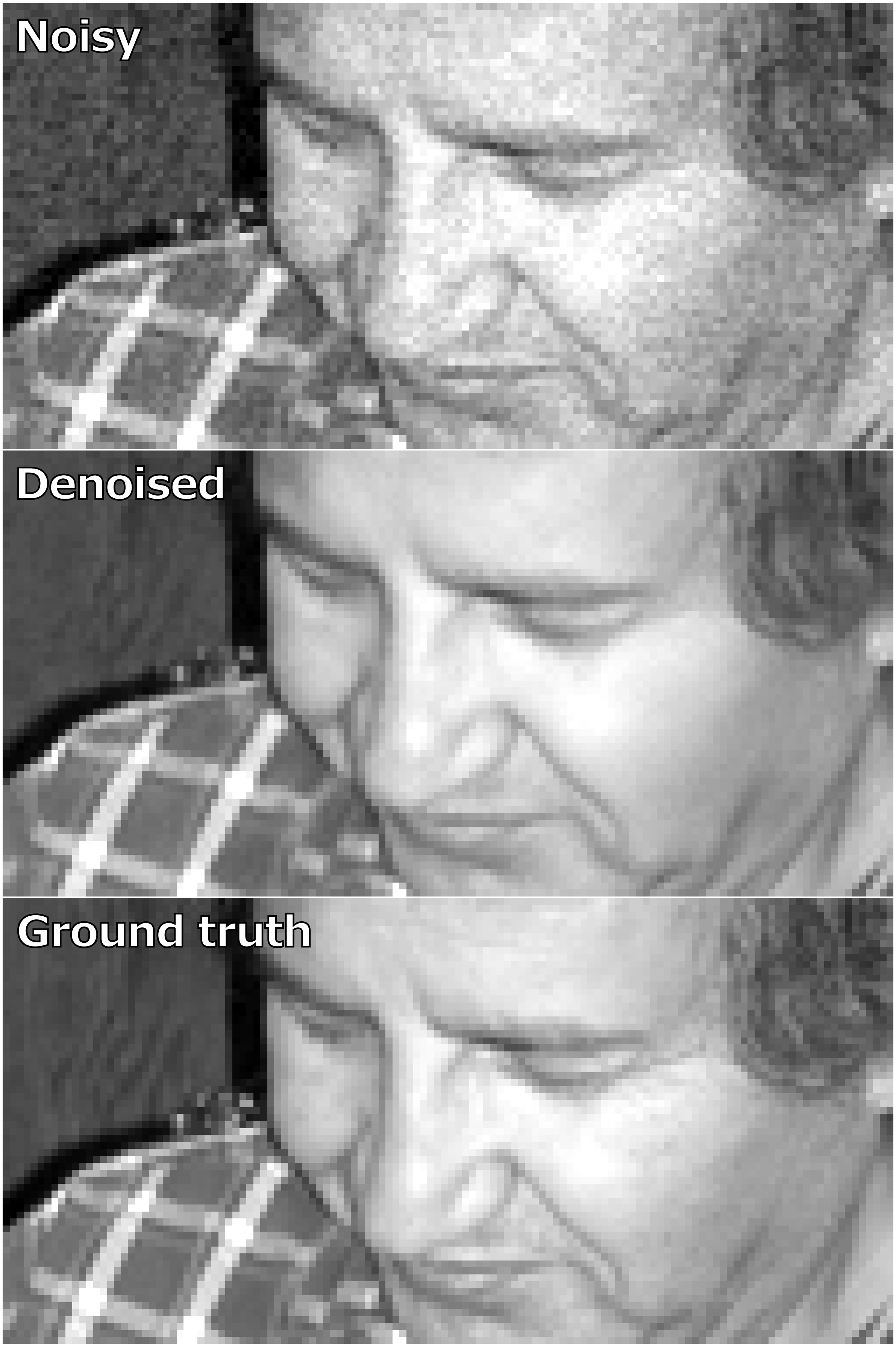

Figure 10.11: *FFHQ - Stripe*

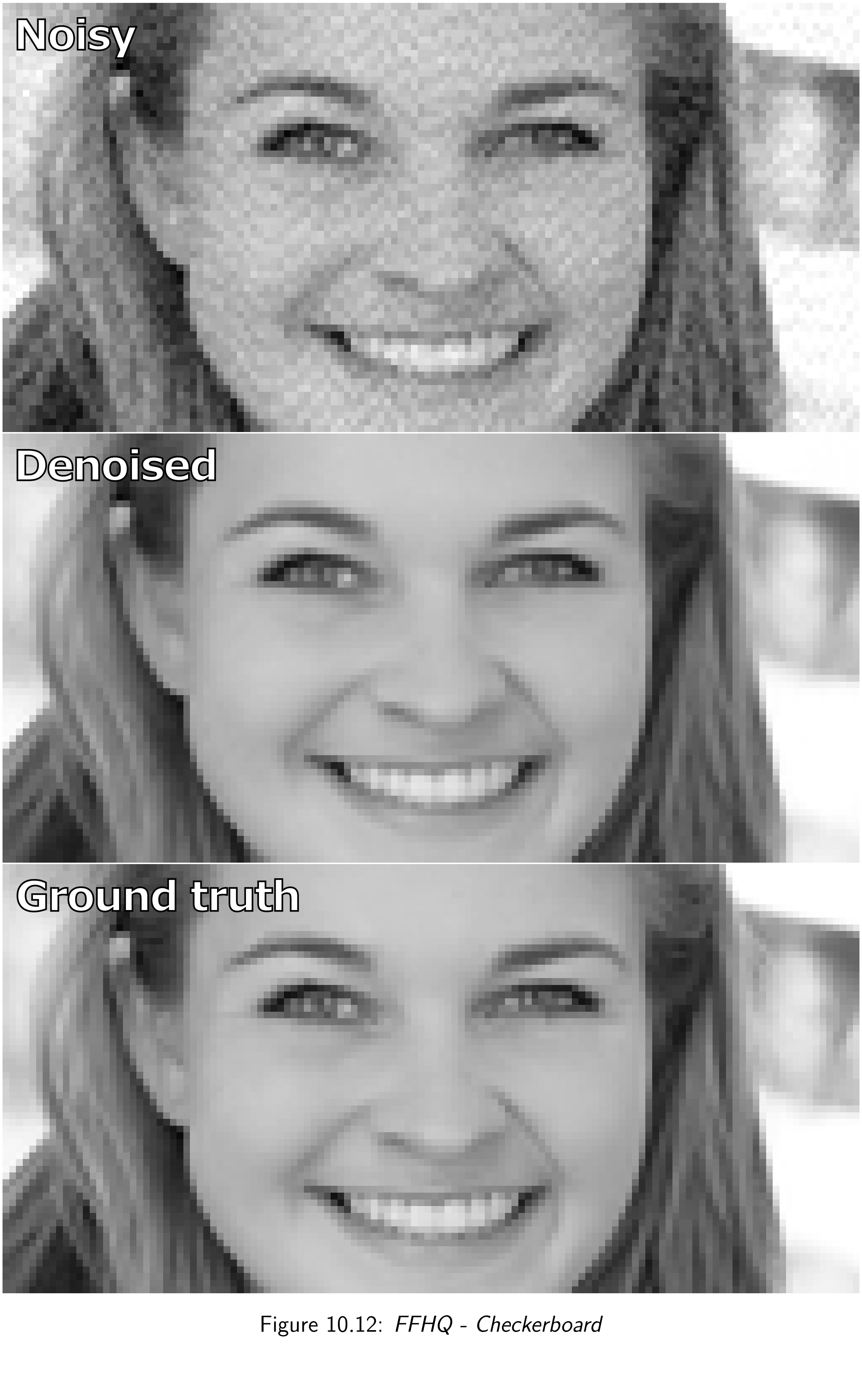

Figure 10.12: *FFHQ - Checkerboard*