Towards Structured Noise Models for Unsupervised Denoising

Anonymous ECCV submission

Paper ID 100

Abstract. The introduction of unsupervised methods in denoising has shown that unpaired noisy data can be used to train denoising networks, which can not only produce a high quality results but also enable us to sample multiple possible diverse denoising solutions. However, these systems rely on a probabilistic description of the imaging noise–a noise model. Until now, imaging noise has been modelled as pixel-independent in this context. While such models often capture shot noise and read-out noise very well, they are unable to describe many of the complex patterns that occur in real life application. Here, we introduce an novel learning-based autoregressive noise model to describe imaging noise and show how it can enable unsupervised denoising for settings with complex structured noise patterns. We explore different ways to train a model for real life imaging noise and show that our deep autoregressive noise model has the potential to greatly improve denoising quality in structured noise datasets. We showcase the capability of our approach on various simulated datasets and on real photo-acoustic imaging data.

Keywords: denoising, deep learning, autoregressive, noise, diverse solutions, VAE, photo-acoustic imaging

1 Introduction

Whenever we attempt to acquire an image $\mathbf{s}$, using a microscope or any other recording device, we should generally expect that the result $\mathbf{x}$ will not perfectly correspond to to the signal. Instead, our measurement will be be subject to the random inaccuracies of the recording process, resulting in what is referred to as noise. We can define noise $\mathbf{n} = \mathbf{x} - \mathbf{s}$ as the difference between the corrupted observation and the true signal. Noise is especially prevalent in sub-optimal imaging conditions, such as when imaging with only a small amount of light. As a result, noise often becomes the limiting factor in life science imaging, operating right at the boundary of what is possible with current technology. The algorithmic removal of noise (*denoising*) can thus be a vital tool, enabling new previously unfeasible experimental setups [4, 16]. Given a noisy image $\mathbf{x}$, we can think of the denoising task as finding an estimate $\hat{\mathbf{s}}$ that is close to the true clean image $\mathbf{s}$.

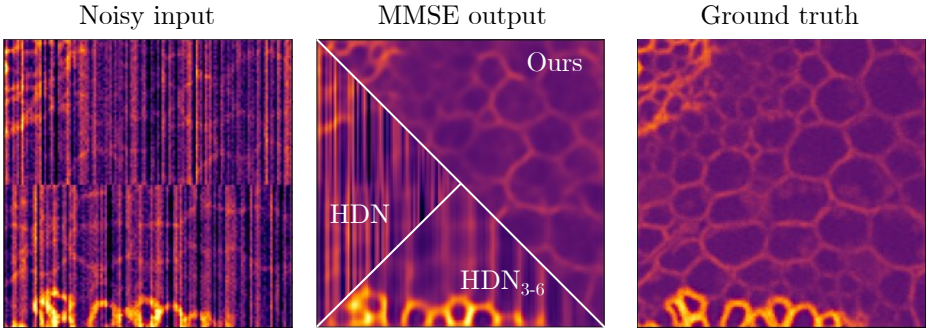

Fig. 1. Comparing HDN with our novel autoregressive noise model to HDN and HDN$_{3-6}$ with the standard pixel-independent noise model. Structured noise can be observed in many imaging modalities. Here, the simulated striped pattern in the noise is designed to mimic noise real noise as it frequently 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.

Consequently, since the introduction of digital image processing, a plethora of denoising methods has been proposed [7, 10, 18], to name a few. The last decade however, has seen a revolution of the field, with machine learning (ML) emerging as the technology capable of producing the most accurate results [4, 16]. Traditional supervised ML-based methods [28] view denoising as a regression problem, *i.e.*, they attempt to learn a function, mapping noisy images $\mathbf{x}$ to the true clean signal $\mathbf{s}$, based on previously collected training data of noisy-clean-image-pairs.

Despite its success, supervised learning of this form comes with an important caveat, the acquisition of training data can be impractical. Originally, the approach requires us to collect paired clean and noisy images of the same content type we would like to denoise. This is not always possible. Although the problem was partially alleviated by Lehtinen *et al.* [17], showing that pairs of corresponding noisy images are sufficient, the collection of paired data has remained an obstacle for many practical applications.

Only in recent years has this problem been addressed by new self- and unsupervised methods [14, 3, 15, 25, 6, 20, 24, 23], 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 [24, 23], referred to as DivNoising and HDN [23], provide an additional benefit. They do not produce a single estimate of the true signal, but instead allow us to sample different possible solutions for a noisy input image.

However, to achieve this, these methods require an additional ingredient during training. They rely on a mathematical description of the imaging noise, called *noise model*. The noise model is description 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 [15], bootstrapped (using a self-supervised denoising algorithm) [25], or even co-learned on-the-fly while training the denoiser [24]. 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.

However, 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 [30]. For simplicity, we will refer to this type of noise simply as *pixel-independent* noise.

Unfortunately, many imaging systems, such as computed tomography (CT) [9] or photo acoustic imaging (PA) [29], do not adhere to this property and can produce structured noise. In practice, even in fluorescence microscopy the *conditional pixel-independence* assumption does not always hold, due to the camera's complex electronics. Many fluorescence microscopy setups suffer from noise that is partially structured. Figure 1 shows an example of simulated structured noise with a pattern close to what is produced by many sCMOS cameras [2].

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]. Even though, Prakash *et al.* [23] show that the effects of this problem can in practice 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 framework. We present an autoregressive noise model that is capable of describing structured noise and thus enabling DivNoising to remove it. We evaluate our method quantitatively on various simulated datasets and qualitatively on a PA dataset featuring highly structured noise.

In summary, our contributions are:

1. We present an autoregressive noise model capable of describing structured noise.
2. We demonstrate that DivNoising together with our noise model can effectively remove simulated structured noise in situations where the previously proposed approach [23] fails.

[1] The same is true for self-supervised methods such as [14], which discusses this topic explicitly

3. We qualitatively demonstrate structured noise removal on a real PA data.

2 Related Work

2.1 Self- and Unsupervised Methods for Removing Structured Noise

Noise2Void [14] is a self-supervised approach to removing pixel-independent 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 the one in the centre masked. It is trained to produce an output that is as close as possible to the pixel 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*. [6] 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.

As mentioned previously, in [23], Prakash *et al*. demonstrated that tuning the expressive power of a DivNoising based method enables it to remove some cases of structured noise. This method is described in more detail in Section 3.1.

2.2 Noise Modelling

In [1], Abdelhamed *et al*. proposed a deep generative noise model known as Noise Flow. It is based on the Glow [12] 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 paper, but a recent review on deep generative modelling [5] found that auto-regressive models perform slightly better in terms of log-likelihood. As will be seen later, this makes auto-regressive noise models more suitable in a DivNoising framework.

3 Background

Here, we want to give a brief recap of the methods our approach relies on. We will begin with the DivNoising Prakash *et al*. [24] and its extension [23] (HDN), which is the framework our method is built upon. We will then discuss the currently used pixel-independent noise models, which are a component in DivNoising and HDN and which we will later compare against our novel autoregressive replacement. Finally, we will have a brief look at deep autoregressive models, which provide the backbone for our noise model.

3.1 DivNoising and HDN

Training DivNoising requires two ingredients, the data that needs to be denoised and a pre-trained or measured noise model, $p_\eta(\mathbf{x}|\mathbf{s})$. We will discuss the noise model in more detail in Section 3.2.

Instead of directly providing an estimate $\hat{\mathbf{s}}$ for a noisy image, DivNoising allows us to sample possible solutions $\mathbf{s}^k$ from an approximate *posterior* distribution $p(\mathbf{s}|\mathbf{x})$, *i.e.*, from the distribution of possible clean images given the noisy input. To obtain a sensible single estimate, we can average a large number of these samples to produce the *minimum mean square error* (MMSE) estimate

$$\hat{\mathbf{s}} = \frac{1}{K}\sum_{k=1}^{K}\mathbf{s}^k, \tag{1}$$

which is comparable to the single solution provided by a supervised denoising network.

DivNoising works by training a variational autoencoder (VAE) to approximate the distribution of training images $\mathbf{x}$. VAEs are latent variable models. That is, they can model a difficult and high dimensional distributions by introducing an unobserved latent variable $\mathbf{z}$ following a known prior distribution. In DivNoising $p(\mathbf{z})$ is assumed to be a standard normal distribution. DivNoising describes the distribution of noisy images as

$$\log p_{\theta,\eta}(\mathbf{x}) = \log \int p_\eta\left(\mathbf{x}|\mathbf{s} = g_\theta(\mathbf{z})\right)p(\mathbf{z})d\mathbf{z}, \tag{2}$$

where $g_\theta : \mathbb{R}^d \rightarrow \mathbb{R}^{D>d}$ is a convolutional neural network (CNN) called *decoder* that maps from the space of latent variables to the space of signals. We use θ to denote the parameters of the decoder network. Once trained, the decoder warps the simple distribution $p(\mathbf{z})$ to the potentially highly complex distribution of clean images. Even though this is an extremely expressive model, training of the parameters θ is challenging due to the intractable integral Eq. 2. In practice, a VAE can be trained by maximising the variational lower bound

$$\log p_{\theta,\eta}(\mathbf{x}) \geq \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\eta(\mathbf{x}|\mathbf{s} = g_\theta(\mathbf{z}))] - D_{KL}[q_\phi(\mathbf{z}|\mathbf{x}) \parallel p(\mathbf{z})], \tag{3}$$

where D_{KL} is the Kullback-Liebler divergence, and $q_\phi(\mathbf{z}|\mathbf{x})$ is a parametric distribution in latent space, implemented by a second CNN, called the *encoder*. The encoder network takes a noisy image $\mathbf{x}$ as input and outputs the parameters of the distribution. The encoder, ϕ, and the decoder, θ, are trained in tandem by maximising Eq. 3 based on a set of noisy training images.

Once trained, DivNoising can be used to denoise an image $\mathbf{x}$ by processing it with the encoder, drawing a sample $\mathbf{z}^k$ in latent space from $q_\phi(\mathbf{z}|\mathbf{x})$, and finally decoding the sample $g_\theta(\mathbf{z}^k)$ to obtain sampled solution $\mathbf{s}^k$. The resulting sampled solutions can then be combined to produce an MMSE estimate using Eq. 1.

The original DivNoising shows impressive performance in many cases, but struggles when applied to highly complex datasets, which contain diverse patterns and shapes. In these cases, the results tend to be blurry or contain artifacts.

The reason for this is that DivNoising trains a full model of the image distribution and this is a challenging task for complex datasets. Due to it's architecture, DivNoising performs especially poor for images that contain a lot of high frequency information.

As a side effect of this, DivNoising was found to at times remove structured noise even when using a pixel-independent noise model [23]. However, this comes at the cost of a blurred denoising result.

In [23], the power of DivNoising was improved with the use of the LadderVAE architecture [26]. This version is known as Hierarchical DivNoising (HDN). The main difference between a LadderVAE and a typical VAE is that the latent variable $\mathbf{z}$ is replaced with a hierarchy of latent variables $\mathbf{z} = \{\mathbf{z}_1, \mathbf{z}_2, \dots \mathbf{z}_n\}$ where each $\mathbf{z}_i$ is conditionally dependent upon all $\mathbf{z}_{i+1}, \dots, \mathbf{z}_n$, so that the prior distribution factorises as:

$$p_\theta(\mathbf{z}) = p_\theta(\mathbf{z}_n) \prod_{i=1}^{n-1} p_\theta(\mathbf{z}_i | \mathbf{z}_{i+1}, \dots, \mathbf{z}_n), \tag{4}$$

and the approximate posterior factorises as:

$$q_\phi(\mathbf{z}|\mathbf{x}) = q_\phi(\mathbf{z}_n|\mathbf{x}) \prod_{i=1}^{n-1} q_\phi(\mathbf{z}_i | \mathbf{z}_{i+1}, \dots, \mathbf{z}_n, \mathbf{x}). \tag{5}$$

With these changes, the variational lower bound to the log likelihood is now:

$$\begin{aligned} \log p_\theta(\mathbf{x}) \geq\ & \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\eta(\mathbf{x}|\mathbf{s} = g_\theta(\mathbf{z}))] \\ & - D_{KL}[q_\phi(\mathbf{z}_n|\mathbf{x}) \parallel p(\mathbf{z}_n)] \\ & - \sum_{i=1}^{n-1} \mathbb{E}_{q_\phi(\mathbf{z}_i|\mathbf{x})}[q_\phi(\mathbf{z}_i|\mathbf{z}_{i+1}, \dots, \mathbf{z}_n, \mathbf{x}) \parallel p_\theta(\mathbf{z}_i|\mathbf{z}_{i+1}, \dots, \mathbf{z}_n)] \end{aligned} \tag{6}$$

The authors found that with HDN the denoising capability is greatly improved, especially for complex high detail datasets. However, when HDN is used with a pixel-independent noise model, it will usually also faithfully reconstruct any structured noise instead of removing it. Prakash *et al.* were able to address this problem in some cases by not conditioning the distribution of the lowest latent variables in the hierarchy on $\mathbf{x}$. They noticed that it was through this conditioning that the model passed information about the structured noise to the output, so by severing the connection, the signal estimate was produced without the structured artifacts.

In their experiments, Prakash *et al.* mostly used HDN with six latent variables in the hierarchy, and when tackling structured noise they would alter the distribution of the first two. We refer this altered model as HDN_{3-6} for the remainder of this paper.

We find that HDN_{3-6} does not work in all cases (see Figure 1) and also comes at a cost. By removing some levels of latent variables we also reduce the expressiveness the model. Consequently, when we combine HDN with our autoregressive noise model, and keep all levels of latent variables activated to allow for maximum expressive power.

3.2 Pixel-Independent Noise Models

Noise models, as they have until now been used with DivNoising and HDN are based on the assumption that when an image is recorded for any underlying signal $\mathbf{s}$ noise is occurring independently in each pixel i. That is, the distribution factorises over the pixels of the image as

$$p_\eta(\mathbf{x}|\mathbf{s}) = \prod_{i=1}^{N} p_\eta(x_i|s_i), \tag{7}$$

where $p_\eta(x_i|s_i)$ correspond to the distributions of possible noisy pixel values given an underlying clean pixel value at the same location i. This means that to describe the noise model for an entire image $p_\eta(\mathbf{x}|\mathbf{s})$, we only need to characterise the much simpler 1-dimensional distributions for individual pixel values $p_\eta(x_i|s_i)$. These pixel noise models have been described with the help of 2-dimensional histograms (using one dimension for the clean signal and one for the noisy observation) [15], or parametrically using individual normal distributions [30] or Gaussian mixture models [25] parameterised by the pixel's signal s_i.

3.3 Signal-Independent Noise Models (a Simplification)

Even though the models described in Eq. 7 are unable to capture dependencies on other pixels, importantly, they are able to describe a dependency on the signal at the pixel itself. For many practical applications this is essential. For example, fluorescence microscopy is often heavily influenced by Poisson shot noise [30], following a distribution that depends on the pixel's signal.

However, here, we will in this work 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

$$p_\eta(\mathbf{x}|\mathbf{s}) \equiv p_\eta(\mathbf{n}), \tag{8}$$

with $\mathbf{n} = \mathbf{x} - \mathbf{s}$, turning Eq. 7 into

$$p_\eta(\mathbf{x}|\mathbf{s}) = \prod_{i=1}^{N} p_\eta(n_i), \tag{9}$$

Allowing us to fully characterise the noise model by defining a single 1-dimensional distribution $p_\eta(n_i)$ describing the noise at the pixel level.

In Section 4, we will introduce our novel autoregressive noise model, which will allow us to get rid of the pixel-independence assumption. However, within the scope of this work we are still operating under the assumption of signal-independence (Eq. 8), leaving the more general case of combined signal- and pixel-independence for future work.

3.4 Deep Autoregressive Models

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

$$p(\mathbf{v}) = \prod_{i=1}^{N} p(v_i | v_1, \ldots, v_{i-1}) \tag{10}$$

of 1-dimensional distributions for each element $p(v_i|v_1, \ldots, v_{i-1})$ conditioned on all previous elements.

Oord *et al.* [21] proposed using a CNN to apply this technique to image data known as Gated PixelCNN, or, for brevity, PixelCNN. The authors suppose a row-wise ordering of the pixels in the image and model the distribution $p(v_i|v_1, \ldots, v_{i-1})$ for each pixel conditioned on all pixels above and to the left of it using a CNN with an adequately shaped receptive field. When applied to the image, the network outputs a the parameters of the 1-dimensional conditional distribution for each pixel.

4 Methods

Considering the signal-independence assumption (Eq. 8), we can see that 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. 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. We denote these noise training images as $\mathbf{n}^j$.

To train our noise model based on Eq. 10, we use the following loss function

$$p_\eta(\mathbf{n}^j) = \sum_{i=1}^{N} \log p_\eta(n_i | n_1^j, \ldots, n_{i-1}^j), \tag{11}$$

where $p_\eta(n_i|n_1^j, \ldots, n_{i-1}^j)$ are the conditional pixel distributions described by our PixelCNN for pixel i by outputting the parameters of a Gaussian mixture model for each pixel.

Once our noise model is trained, we can proceed to our HDN model for denoising. We follow the training process as described in [23] and use Eq. 6 as training loss. Note that this contains the noise model $\log p_\eta(\mathbf{x}|\mathbf{s} = g_\theta(\mathbf{z}))$.

Considering Eq. 8, we can compute $\hat{\mathbf{n}} = \mathbf{x} - g_\theta(\mathbf{z})$ and insert it into Eq. 6, this time keeping the parameters η fixed.

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.

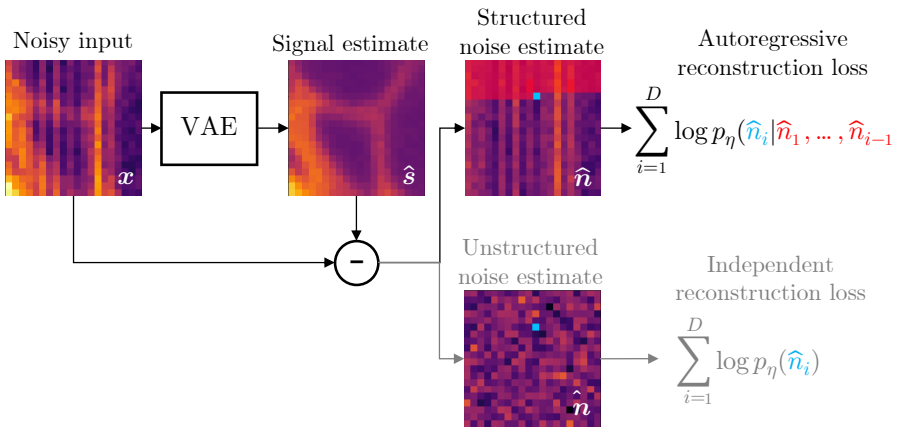

Fig. 2. Our autoregressive noise model as a component in the Divnoising framework. Divnoising trains a VAE to describe the distribution of noisy images $\mathbf{x}$. It does so by sampling clean images $\hat{\mathbf{s}}$ and using a noise model as part of its loss function, called *reconstruction loss*. The reconstruction loss assess the likelihood of network output $\hat{\mathbf{s}}$ 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. The pixel-independent noise model this is done based on the conditional independence assumption by summing over the pixel noise models $\log p(\hat{n}_i)$, modelled as a Gaussian mixture model. In our autoregressive noise model we sum over the conditional distributions $p(\hat{n}_i|\hat{n}_1, \ldots, \hat{n}_{i-1})$ of for the noise in each pixel conditioned on the previous pixels, *i.e.*, the pixels above and left. Our noise models describes these conditional distribution using a modified version of the PixelCNN [27] approach, which is implemented as an efficient fully convolutional network, outputting the parameters of a separate Gaussian mixture model for each pixel.

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 peak signal-to-noise ratio (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.* [6] took 1000 images of a stationary section of a *Convallaria* with size 1024×1024. Each image contained signal-dependent noise, but the average of the 1000 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 [19].

Brain CT 2486 clean CT brain scan images were taken from Hssayeni [11] 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 shifted the noise to have zero mean. This noise was intended to be similar to the CT noise shown in Figure 3 of [22].

KNIST The Kuzushiji-MNIST dataset was taken from Clanuwat *et al.* [8]. 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.

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. 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 Figure 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.

5.3 Training the Noise Model

The noise model used in experiments was based on the architecture in van den Oord *et al.* [21]. We used the same hyperparameters for each dataset. Those hyperparameters were 5 layers, 128 feature channels and a kernel size of 5. 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 0.0001 was used, and learning rate was reduced by 0.99 every epoch. Every dataset was trained on for a maximum of 12000 steps, but a patience of 2 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.

5.4 Training HDN

The architecture the HDN was based on that Prakash *et al.* [23] and was kept the same for all experiments. 6 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 [13] was used with a lambda of 0.5. The Adamax optimiser was used with a learning rate of 0.0003 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.

5.5 Denoising with Autoregressive Noise Models

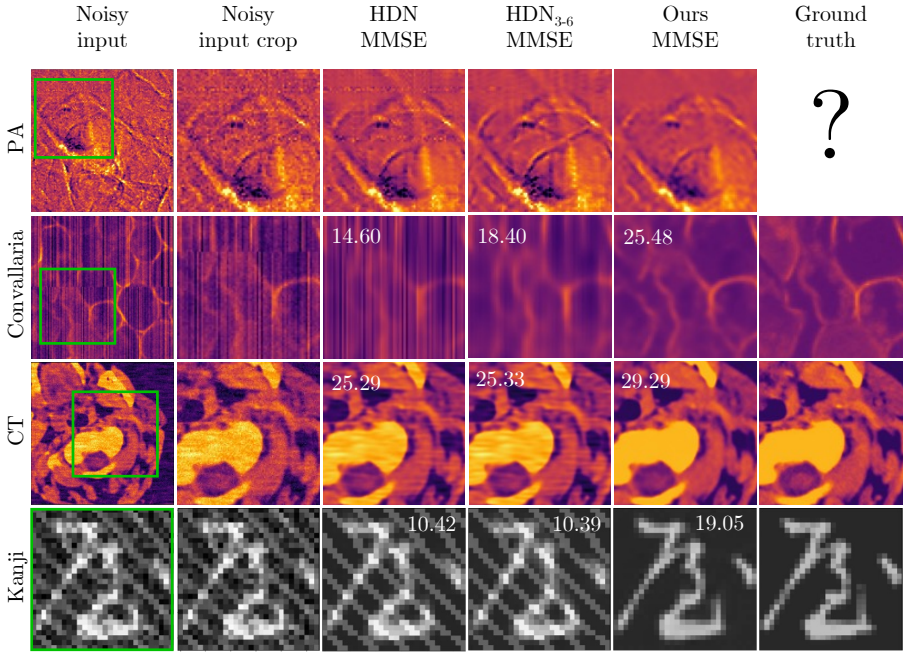

Fig. 3. Denoising results. Here we compare the outputs of different methods on various datasets. The overlayed numbers indicate the average PSNR values on the dataset. We find that HDN with a pixel-independent noise model is able to effectively remove some structured artifacts, by removing layers of the latent space space [23], but fails for larger scale structures, spanning over tens of pixels. In contrast, our method reliably removes all small- and large-scale structured noise.

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 Figure 3, with peak signal-to-noise ratio calculated for the datasets where ground truth is available.

Results from our HDN with our noise model has the highest PSNR on the three datasets with ground truth. It also removed artefacts in the PA image that HDN_{3-6} could not.

5.6 Evaluating the Noise Model

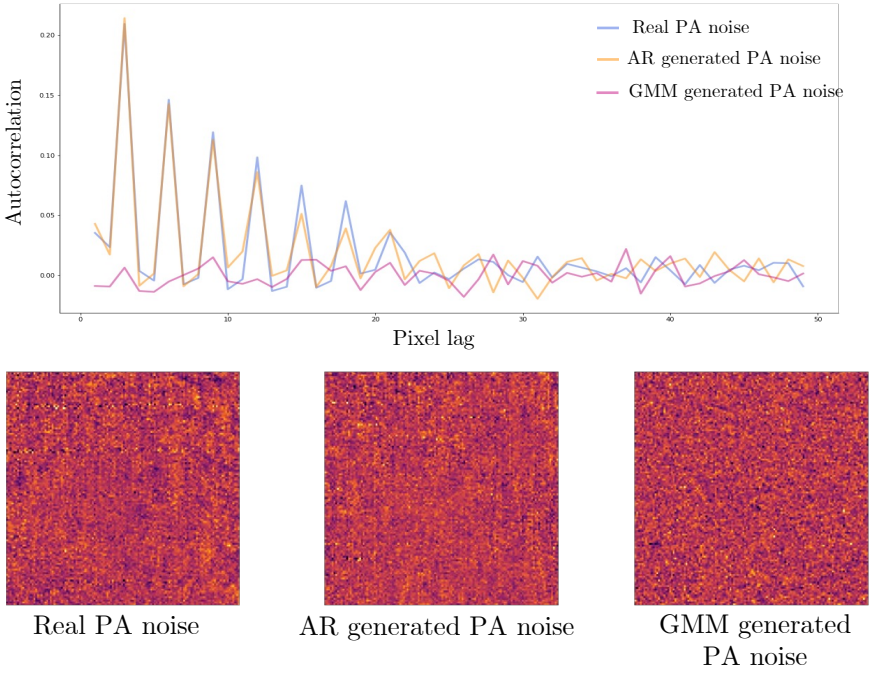

Fig. 4. Comparing the statistics of pixel-independent noise models and our new autoregressive model. Here, we compare generated PA samples from our noise model (AR) (orange), a Gaussian mixture pixel-independent noise model (GMM) (pink), and real noise (blue). The auto-correlation function compares different horizontally shifted versions (pixel lag) of the noise images, characterising the dependencies between pixels values at various distances, *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 horizontal auto-correlation of a sample of

real noise from the PA data, a sample of noise generated from our autoregressive noise model and a sample of noise generated by a Gaussian mixture noise model. Each of these samples are in Figure 4. along with a graph comparing the autocorrelation of the three.

6 Conclusion

We have presented a novel type of noise model to be used within the DivNoising framework that addresses signal-independent, structured noise and outperforms HDN_{3-6} on highly structured, long range noise artefacts.

The key difference between our noise model and those that had been used before [15][24][25] 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 individually and do not take into account inter-pixel correlations.

We plan to extend this noise model to learn the distribution of signal-dependent noise in the future, but in such a way that it can be trained without matched clean-and-noisy pairs.

7 Acknowledgements

Removed for double-blind review.

References

1. Abdelhamed, A., Brubaker, M.A., Brown, M.S.: Noise flow: Noise modeling with conditional normalizing flows. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3165–3173 (2019)
2. Babcock, H.P., Huang, F., Speer, C.M.: Correcting artifacts in single molecule localization microscopy analysis arising from pixel quantum efficiency differences in scmos cameras. Scientific reports **9**(1), 1–10 (2019)
3. Batson, J., Royer, L.: Noise2self: Blind denoising by self-supervision (2019)
4. Belthangady, C., Royer, L.A.: Applications, promises, and pitfalls of deep learning for fluorescence image reconstruction. Nature methods pp. 1–11 (2019)
5. Bond-Taylor, S., Leach, A., Long, Y., Willcocks, C.G.: Deep generative modelling: A comparative review of vaes, gans, normalizing flows, energy-based and autoregressive models. arXiv preprint arXiv:2103.04922 (2021)
6. Broaddus, C., Krull, A., Weigert, M., Schmidt, U., Myers, G.: Removing structured noise with self-supervised blind-spot networks. In: 2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI). pp. 159–163 (2020)
7. Buades, A., Coll, B., Morel, J.M.: 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. Ieee (2005)
8. Clanuwat, T., Bober-Irizar, M., Kitamoto, A., Lamb, A., Yamamoto, K., Ha, D.: Deep learning for classical japanese literature. arXiv preprint arXiv:1812.01718 (2018)
9. Cnudde, V., Boone, M.N.: High-resolution x-ray computed tomography in geosciences: A review of the current technology and applications. Earth-Science Reviews **123**, 1–17 (2013)
10. Dabov, K., Foi, A., Katkovnik, V., Egiazarian, K.: Image denoising by sparse 3-d transform-domain collaborative filtering. IEEE Transactions on image processing **16**(8), 2080–2095 (2007)
11. Hssayeni, M., Croock, M., Salman, A., Al-khafaji, H., Yahya, Z., Ghoraani, B.: Computed tomography images for intracranial hemorrhage detection and segmentation. Intracranial Hemorrhage Segmentation Using A Deep Convolutional Model. Data **5**(1) (2020)
12. Kingma, D.P., Dhariwal, P.: Glow: Generative flow with invertible 1x1 convolutions. Advances in neural information processing systems **31** (2018)
13. Kingma, D.P., Salimans, T., Jozefowicz, R., Chen, X., Sutskever, I., Welling, M.: Improved variational inference with inverse autoregressive flow. Advances in neural information processing systems **29** (2016)
14. Krull, A., Buchholz, T.O., Jug, F.: Noise2void-learning denoising from single noisy images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2129–2137 (2019)
15. Krull, A., Vicar, T., Prakash, M., Lalit, M., Jug, F.: Probabilistic Noise2Void: Unsupervised Content-Aware Denoising. Front. Comput. Sci. **2**, 60 (Feb 2020)
16. Laine, R.F., Jacquemet, G., Krull, A.: Imaging in focus: an introduction to denoising bioimages in the era of deep learning. The International Journal of Biochemistry & Cell Biology **140**, 106077 (2021)
17. Lehtinen, J., Munkberg, J., Hasselgren, J., Laine, S., Karras, T., Aittala, M., Aila, T.: Noise2noise: Learning image restoration without clean data. In: International Conference on Machine Learning. pp. 2965–2974 (2018)

18. Luisier, F., Vonesch, C., Blu, T., Unser, M.: Fast interscale wavelet denoising of poisson-corrupted images. Signal processing **90**(2), 415–427 (2010)
19. Mandracchia, B., Hua, X., Guo, C., Son, J., Urner, T., Jia, S.: Fast and accurate scmos noise correction for fluorescence microscopy. Nature communications **11**(1), 1–12 (2020)
20. Moran, N., Schmidt, D., Zhong, Y., Coady, P.: 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)
21. Van den Oord, A., Kalchbrenner, N., Espeholt, L., Vinyals, O., Graves, A., et al.: Conditional image generation with pixelcnn decoders. Advances in neural information processing systems **29** (2016)
22. Parakh, A., An, C., Lennartz, S., Rajiah, P., Yeh, B.M., Simeone, F.J., Sahani, D.V., Kambadakone, A.R.: Recognizing and minimizing artifacts at dual-energy ct. Radiographics **41**(2), 509 (2021)
23. Prakash, M., Delbracio, M., Milanfar, P., Jug, F.: Interpretable unsupervised diversity denoising and artefact removal. In: International Conference on Learning Representations (2022), https://openreview.net/forum?id=DfMqlB0PXjM
24. Prakash, M., Krull, A., Jug, F.: Fully unsupervised diversity denoising with convolutional variational autoencoders. In: International Conference on Learning Representations (2020)
25. Prakash, M., Lalit, M., Tomancak, P., Krull, A., Jug, F.: Fully unsupervised probabilistic noise2void. arXiv preprint arXiv:1911.12291 (2019)
26. Sønderby, C.K., Raiko, T., Maaløe, L., Sønderby, S.K., Winther, O.: Ladder variational autoencoders. Advances in neural information processing systems **29** (2016)
27. Van Oord, A., Kalchbrenner, N., Kavukcuoglu, K.: Pixel recurrent neural networks. In: International conference on machine learning. pp. 1747–1756. PMLR (2016)
28. Weigert, M., Schmidt, U., Boothe, T., Müller, A., Dibrov, A., Jain, A., Wilhelm, B., Schmidt, D., Broaddus, C., Culley, S., et al.: Content-aware image restoration: pushing the limits of fluorescence microscopy. Nature methods **15**(12), 1090–1097 (2018)
29. Xu, M., Wang, L.V.: Photoacoustic imaging in biomedicine. Review of scientific instruments **77**(4), 041101 (2006)
30. Zhang, Y., Zhu, Y., Nichols, E., Wang, Q., Zhang, S., Smith, C., Howard, S.: A poisson-gaussian denoising dataset with real fluorescence microscopy images. In: CVPR (2019)