

Image Denoising and the Generative Accumulation of Photons

Alexander Krull¹, Hector Basevi², Benjamin Salmon¹, Andre Zeug³, Franziska Müller³,
 Samuel Tonks¹, Leela Muppala¹, and Aleš Leonardis¹

¹Computer Science, University of Birmingham, UK

²Metabolism and Systems Research, University of Birmingham, UK

³Medizinische Hochschule Hannover, Germany

August 2, 2023

Abstract

We present a fresh perspective on shot noise corrupted images and noise removal. By viewing image formation as the sequential accumulation of photons on a detector grid, we show that a network trained to predict where the next photon could arrive is in fact solving the minimum mean square error (MMSE) denoising task. This new perspective allows us to make three contributions: i. We present a new strategy for self-supervised denoising, ii. We present a new method for sampling from the posterior of possible solutions by iteratively sampling and adding small numbers of photons to the image. iii. We derive a full generative model by starting this process from an empty canvas. We call this approach generative accumulation of photons (GAP). We evaluate our method quantitatively and qualitatively on 4 new fluorescence microscopy datasets, which will be made available to the community. We find that it outperforms supervised, self-supervised and unsupervised baselines or performs on-par.

1 Introduction

Scientific imaging techniques such as fluorescence microscopy have to limit the amount of light used to avoid damaging or destroying their sample [1]. As a result, the recorded images inevitably suffer from a certain degree of noise which has to be addressed in the downstream analysis. Images can be subject to a variety of differ-

ent types [2] of noise which can be alleviated by various technical means (*e.g.* [3]). However, there is a type of noise which is physically inevitable for most imaging set-ups in low-light conditions. It is referred to as Poisson *shot noise*.

Shot noise is the result of the particle nature of light. Even high-end scientific detectors and cameras that can accurately count the precise number of photons hitting each pixel cannot record a noise-free image. For a given light intensity the number of photons arriving at the detector is itself inherently random and follows a Poisson distribution. The effect is especially severe in microscopy applications, operating in low-light conditions.

The last decade has seen a number of deep learning-based computational methods designed to reduce noise after images have been recorded in order to allow for improved analysis of the data [2]. One of the first proposed methods, known as content-aware image restoration (CARE) [4], is based on training convolutional neural networks (CNNs) to learn a mapping from noisy images to clean images. Unfortunately, the method requires pairs of corresponding noisy and clean images during training, which can be hard to acquire in practice, rendering it inapplicable in many situations. However, other works have expanded on this line of research, enabling training with noisy image pairs [5] and even with unpaired noisy images, *e.g.* [6, 7, 8, 9].

While achieving impressive results, these supervised and self-supervised methods share a common shortcom-

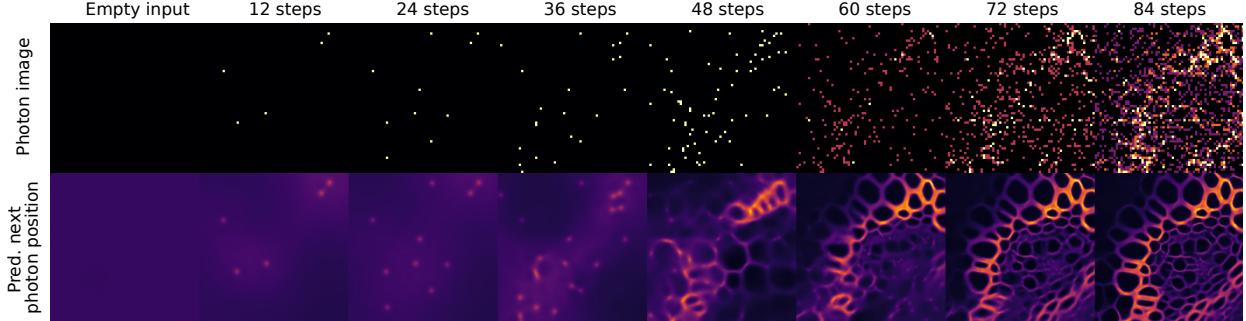


Figure 1: **Generative accumulation of photons (GAP):** Starting with an empty canvas, our method repeatedly predicts a map of probabilities of where the next photon might arrive and uses it to randomly place photons. We show the process for the *Neuro-PC* and *Conv-PC* datasets. Photon images have been down-sampled for better visibility.

ing: Denoising is inherently an ill-posed problem and given an image corrupted by a substantial amount of noise, it is not generally possible to recover the true underlying clean image. In fact, there is a posterior distribution of possible solutions that all might have led to the original noisy observation. When we view denoising as a regression problem like [4] aiming to learn a mapping to the clean image, we are in fact learning a mapping to a compromise between possible solutions, which may itself look different (often being more blurry) from real clean images.

This problem has been explored by Prakash *et al.* [10, 11], who proposed the idea of *diversity denoising* based on a variational autoencoder [12] (VAE). Instead of producing a single solution for each noisy image, Prakash *et al.* are able to sample possible solutions from an approximate posterior distribution of clean images.

Here, we take an entirely new perspective, focusing on shot noise and the denoising of shot noise corrupted images. Instead of viewing shot noise as a secondary corruption process applied to a clean image, we understand image formation as the sequential accumulation of photons and see any measured shot noise-affected image as a result of this process. We call this approach generative accumulation of photons (GAP).

We train a CNN to take a shot noise-affected image as input and predict a probability distribution over where the next photon might arrive. We show that, for normalized images, predicting the next photon position is identical to

denoising the image.

Based on this insight, we derive a novel method of training a self-supervised model for image denoising. Additionally, by understanding image generation as the accumulation of photons, we describe a new method for diversity denoising: we iteratively predict the distribution of the next photon position and randomly sample photons accordingly. Finally, by starting with an empty photon-free canvas, we are able to derive a full generative image model. The process is illustrated in Figures 1 and 2.

We introduce four new shot noise-corrupted microscopy datasets for evaluation that will be made available to the community. We evaluate our method quantitatively and qualitatively and find that it yields competitive results.

2 Related Work

2.1 Supervised denoising

Supervised denoising methods usually train CNNs using pairs of noisy and clean images to learn a mapping between the two. A common choice for the loss function is the mean square error (MSE) between the prediction and clean ground truth. Considering that there is a distribution of possible solutions, minimising the MSE loss corresponds to finding the expected value. Unfortunately obtaining clean ground truth data can be challenging or impossible for many applications and so supervised meth-

ods are often not applicable in the context of scientific imaging.

Lehtinen *et al.* introduced *Noise2Noise* [5], a partial solution to this problem. They showed that it is possible to replace the clean ground truth target with a second noisy version which might be more readily available. A network trained with this type of data will still find the same MMSE solution. While this presented a big step forward with respect to applicability, *Noise2Noise* still requires training pairs, which have to be collected for this purpose.

2.2 Self-supervised blind-spot denoising

Self-supervised blind-spot methods [6, 7, 13] suggest a training strategy that can do without paired training data, *i.e.* allowing training directly on the data that should be denoised, while still obtaining the same MMSE solution. The main idea is to block out individual pixels in order to use them as noisy targets (similar to *Noise2Noise*). These strategies rely on the assumption that imaging noise is conditionally pixel-independent given the underlying clean signal, making it not possible to predict the noise in a pixel from its surroundings. The downside of this approach is that, when making a prediction for a pixel, the network cannot make use of the pixel value itself, thus it is not making optimal use of the available information.

Our photon-based self-supervised denoising strategy is related to the blind-spot idea in that it removes part of the input image to use it as the target. However, instead of removing pixels, we are only removing individual photons, which means we are not facing the same problem of disregarded information.

2.3 VAE-based denoising

Another approach to image denoising has been suggested in [10]. The core idea is to use a variational autoencoder to describe the distribution of noisy images. By including a statistical model of the imaging noise as part of the decoder, the method allows us to: i. sample from an approximate posterior distribution of possible clean images, and ii. to sample clean images from scratch, functioning as a full generative model.

An extended method with a more powerful network architecture was presented in [11] under the name HDN. We see this method as our main competitor as it can be trained

from unpaired noisy data and, similarly to our method, can function as a generative model.

Unlike GAP which produces an MMSE denoising result in a single step, HDN produces MMSE results by repeated sampling and averaging from the posterior distribution.

2.4 Generative Image Models

Generative image models aim to describe a probability distribution over images, a highly challenging task, due to: i. the high dimensionality of the random variable (the number of pixels) and ii. due to the complex higher-order correlations between pixel values at different locations. As a result of ii. the distribution cannot easily be factorised into lower order terms and attempts to factorise using methods such as Markov random fields (MRFs) [14] have led to overly simplistic results that do not realistically describe the image distribution.

In recent years, a number of approaches to this problem have been highly successful. Latent variable models, such as generative adversarial networks (GANs) [15] and VAEs [12], or normalising flows [16], describe difficult distributions indirectly by starting with an easily modelled high dimensional latent variable (usually following a normal distribution) which is then deformed using convolutional neural networks (CNNs) to describe the distributions of interest.

A different approach to this is autoregressive modelling, as proposed by Van Oord *et al.* [17]. By viewing image generation as a sequential process in which the pixels of an image are thought to be generated one-at-a-time conditioned on all previous pixels. In this setup, the whole model can be formulated as a product of 1D conditional distributions over each pixel’s intensity value. Our method can be viewed as an autoregressive approach as we model image generation as a sequential process. However, we sample images by sequentially placing individual photons instead of drawing pixel values.

Finally, the current state-of-the-art approach to image modelling, *denoising diffusion models* [18], follows a similar approach by describing image generation as a sequence of steps. The process is inspired by physics and considers an image as a particle in a high dimensional space, diffusing away from its original position according to some noise distribution. To generate an image the de-

noising diffusion approach reverses the diffusion process by applying a sequence of denoising steps.

Denoising diffusion models iteratively reverse a diffusion process on clean images which typically involves Gaussian noise. This noise can be applied directly to the image [19, 18], or instead to a latent representation of the image corresponding to a pre-existing autoencoder [20]. In both cases, the diffusion noise distribution is unrelated to the noise distribution of the training data. The diffusion model learns to sample from the *noisy* training data distribution and so its samples contain this noise. In contrast, GAP learns to sample from a noisy training data distribution *and* to denoise this distribution. In addition, every iteration of GAP results in a physically valid noisy image.

Recent works have explored generalisations of diffusion models to broader families of corruption processes. Bansal *et al.* [21] focused on deterministic image corruptions. Daras *et al.* [22] focused on image corruptions which are linear with respect to the clean image. GAP focuses on shot noise, which is neither deterministic nor linear.

3 Method

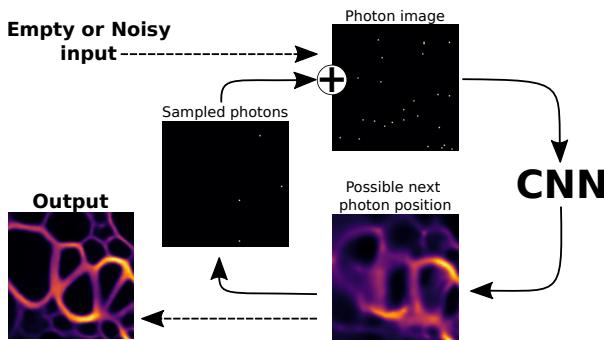


Figure 2: **Sampling algorithm:** Starting with a noisy or empty image, our method repeatedly predicts a map of where the next photon might arrive and uses it to randomly place a small number of photons.

3.1 Image Formation and Shot Noise

When we record an image, we usually project light onto a digital sensor, such as a CMOS or a CCD chip.¹ These chips contain many detector elements measuring the amount of light arriving at different locations on the chip. In our simplified model we assume that each of these detector elements corresponds to one pixel of the final image. When measuring the amount of light in each pixel, we treat light as discrete particles called photons. In an ideal case with a perfect detector, each pixel value in the final image corresponds to the number of photons that fell onto the pixel.

The result of this process is a *shot noise* corrupted image $\mathbf{x} = (x_1, \dots, x_n)$, where the photon count x_i in each pixel i , is independently drawn from a Poisson distribution

$$p(x_i|s_i) = \frac{s_i^{x_i} \exp(-s_i)}{x_i!}, \quad (1)$$

where s_i refers to the expected number of photons hitting the pixel i during the exposure, *i.e.* to the light intensity at the pixel – the quantity we were originally interested in measuring. We will refer to the vector $\mathbf{s} = (s_1, \dots, s_n)$ as the signal or as a clean image.

Since photons are hitting each pixel independently given a signal, we can describe the probability of observing a noisy image \mathbf{x} given a signal \mathbf{s} as

$$p(\mathbf{x}|\mathbf{s}) = \prod_{i=1}^n p(x_i|s_i). \quad (2)$$

We can now think of image formation as a two-step process. We can imagine an image being created by first drawing a clean image \mathbf{s} from a distribution $p(\mathbf{s})$ and then applying shot noise by drawing photon counts from Eq. 2 to create the shot noise-corrupted version.

3.2 The Denoising Task

Given noisy observation \mathbf{x} , denoising is defined as finding an estimate $\hat{\mathbf{s}}$ for the unknown clean image \mathbf{s} .

¹Some imaging technologies, especially those capable of counting photons, work by scanning the sample and recording one pixel at a time. Since this does not affect our model we will focus our explanation on camera-based systems for simplicity.

However, considering the process of image generation described above, finding the true signal may not be possible since many clean images can lead to the same noisy observation. We can use Bayes' theorem to write down a posterior distribution over possible clean images for a given noisy observation

$$p(\mathbf{s}|\mathbf{x}) \propto p(\mathbf{x}|\mathbf{s})p(\mathbf{s}). \quad (3)$$

Deep learning-based approaches (*e.g.* [4, 6]) often view denoising as a regression problem and use CNNs to try to directly learn a mapping from \mathbf{x} to \mathbf{s} . When such methods are trained with a mean squared error (MSE) loss function the optimal solution is the expectation

$$\hat{\mathbf{s}} = \int p(\mathbf{s}|\mathbf{x})\mathbf{s} d\mathbf{s}. \quad (4)$$

We call this the minimum mean squared error (MMSE) solution. This is a sensible way to find an estimate, but we should be aware that it constitutes a compromise between all possible \mathbf{s} .

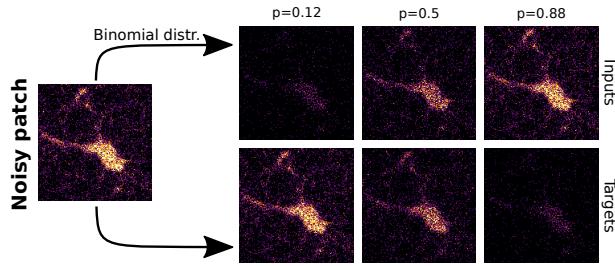


Figure 3: **Photon splitting:** We propose a new way of generating training pairs that requires only noisy data. We split the noisy image by randomly assigning photons to the input or target image. The number of photons assigned to the input is drawn from a binomial distribution, with a parameter p controlling the noise level. Remaining photons are assigned to the target.

3.3 Image Generation from Photon Sequences

Here, we take an alternative view on image generation. Instead of thinking of our pixel values as being drawn

from a Poisson distribution, we will derive an equivalent description, viewing image generation as a sequential process. Remembering that our observation \mathbf{x} is created by photons hitting our detector, we can imagine that it was created by an ordered sequence of photons $\mathbf{i} = (i_1, \dots, i_T)$, where i_t is the index of the pixel where photon t hit the detector. The index t simply refers to the position of the photon in the sequence, with the first photon arriving at $t = 1$ and the last one at $t = T$.

Assuming a known sequence of photons, we compute the resulting image by counting the number of photons hitting each pixel as

$$x_i = \sum_{t=1}^T \mathbb{1}(i_t = i), \quad (5)$$

where $\mathbb{1}(i_t = i)$ is the indicator function.

Considering a given signal \mathbf{s} and a known number T of photons, we can compute the probability of a sequence \mathbf{i} as

$$p(\mathbf{i}|\mathbf{s}, T) = \begin{cases} \prod_{t=1}^T p(i = i_t|\mathbf{s}) & T = |\mathbf{i}| \\ 0 & T \neq |\mathbf{i}| \end{cases}, \quad (6)$$

with the probability being 0 where the length $|\mathbf{i}|$ of the sequence does not match the number of photons T . Since the photons hit the detector independently, their position in the sequence does not matter and we can rewrite the probability as a product over pixels as

$$p(\mathbf{i}|\mathbf{s}, T) = \begin{cases} \prod_{i=1}^n p(i|\mathbf{s})^{x_i} & T = |\mathbf{i}| \\ 0 & T \neq |\mathbf{i}| \end{cases}. \quad (7)$$

where the probability $p(i|\mathbf{s})$ for a photon to hit a particular pixel i , given the signal, should be proportional to the light intensity at the pixel. Thus, we can compute it as the normalised signal at that pixel

$$p(i_t = i|\mathbf{s}) = \frac{s_i}{\sum_{j=1}^n s_j}. \quad (8)$$

However, if the clean signal is unknown the distribution will no longer factorise as easily as Eq. 7. Instead, we have to compute the probability of a sequence as

$$p(\mathbf{i}|T) = \begin{cases} \prod_{t=1}^T p(i = i_t|i_1, \dots, i_{t-1}, T) & T = |\mathbf{i}| \\ 0 & T \neq |\mathbf{i}| \end{cases}, \quad (9)$$

where the distribution $p(i = i_t | i_1, \dots, i_{t-1}, T)$ of the next possible photon location now depends on all previous photons. The order in which photons i_1, \dots, i_{t-1} arrived does not provide any information regarding the next photon position. By additionally considering that the next photon position does not depend on total photon number T nor on the order of previous photons, we can write

$$p(i = i_t | i_1, \dots, i_{t-1}, T) = p(i = i_t | \mathbf{x}_{t-1}), \quad (10)$$

where \mathbf{x}_{t-1} is the observed image at step $t - 1$ according to Eq. 5.

Equation 10 refers to the distribution over the next possible photon locations given a photon image \mathbf{x}_{t-1} . Before taking a closer look at how it can be computed, we would like to point out its significance. Together, Eq. 9 and Eq. 10 provide not only a way to calculate the probability of a sequence but also an iterative way to sample a sequence of photons and therefore images \mathbf{x}_T . Furthermore, for large T , we can expect \mathbf{x}_T to approach the clean image \mathbf{s} , when scaled correctly, so that Eq. 9 and Eq. 10 hold the key to the generation of clean images as well.

3.4 Predicting the Next Photon Location is MMSE Denoising for Normalised Signals

Let us now take a closer look at the distribution of possible next photon locations $p(i = i_t | \mathbf{x}_{t-1})$. We can rewrite Eq. 10 by marginalising over the unknown signal and using Eq. 8 as

$$p(i = i_t | \mathbf{x}_{t-1}) = \int p(\mathbf{s} | \mathbf{x}_{t-1}) p(i_t = i | \mathbf{s}, \mathbf{x}_{t-1}) d\mathbf{s} \quad (11)$$

$$= \int p(\mathbf{s} | \mathbf{x}_{t-1}) \frac{s_i}{\sum_{j=1}^n s_j} d\mathbf{s}. \quad (12)$$

We can see that the result is a weighted average of the possible normalized signals. We should expect that the distribution will be high entropy for small t , *i.e.*, when we have not yet observed many photons, and that it should become more concentrated and low entropy for large t . For very large t , the distribution should approach a normalised version of the signal (Eq. 8), because \mathbf{x}_t will give us more and more information on the underlying signal.

Interestingly, Eq. 12 closely resembles Eq. 4. In fact, if we were to consider only normalized signals with $\sum_{j=1}^n s_j = 1$ the two equations are identical, meaning that the task of predicting the next photon location is identical to denoising the image in an MMSE sense.

We will use a CNN to approximate $f_\theta(\mathbf{x}_{t-1}) \approx p(i = i_t | \mathbf{x}_{t-1})$, where θ are the network parameters. In section 3.5, we will discuss how we can train the CNN to achieve this task.

3.5 Learning to Predict the Next Photon Location

Based on the insight from section 3.4, we know that any model trained for MMSE denoising can approximate the distribution over the next photon location $p(i = i_t | \mathbf{x}_{t-1})$. Starting with normalised clean training images \mathbf{s}^k , the traditional way of creating training pairs is to simulate the corresponding noisy version \mathbf{x}^k . We can then train a denoiser network using a standard quadratic loss function, with \mathbf{x}^k as input and \mathbf{s}^k as target.

However, in many cases clean data is unavailable. Considering the task of predicting the next photon location suggests an alternative self-supervised approach by viewing the problem as a classification task learning the categorical distribution of possible photon positions. By using a softmax layer over pixels at the output of our network to ensure that outputs sum to one, we can use the standard cross-entropy loss. In principle, this would require only individual photon positions as target for each training image, just as classifiers are frequently trained using individual class labels for each training example. We could easily create such training pairs from unpaired noisy images \mathbf{x}^k by randomly removing a single photon and using it as target. The corresponding cross entropy loss is

$$L(\theta) = - \sum_{k=1}^m \sum_{i=1}^n \ln f_i(\mathbf{x}_{\text{inp}}^k; \theta) x_{\text{tar},i}^k, \quad (13)$$

where m is the number of training images, $\mathbf{x}_{\text{inp}}^k$ is the training image with one photon randomly removed and $\mathbf{x}_{\text{tar}}^k$ is a one-hot representation of the removed photon position.

However, we require training data at multiple noise levels to enable our network to predict an accurate approximation of $p(i = i_t | \mathbf{x}_{t-1})$ at different times t . To achieve this, we use a control parameter p and split the image \mathbf{x}^k

into two parts, $\mathbf{x}_{\text{inp}}^k$ and $\mathbf{x}_{\text{tar}}^k$. We can think of this process as simulating a shorter exposure time during image acquisition. Considering that \mathbf{x}^k was recorded with a certain exposure time τ , we can imagine what would be the result if we had instead recorded two images consecutively, with the first image being exposed for $p\tau$ and the second being exposed for $(1-p)\tau$. Considering, that the underlying signal remained fixed during the entire time, each of the photons that make up \mathbf{x}^k would end up in the first image with probability p and in the second image with probability $(1-p)$. To efficiently sample a split for parameter value $0 < p < 1$, we can determine each pixel value $x_{\text{inp},i}^k$ by drawing from binomial distribution using p and x_i^k as the distributions parameters, for success probability and number of trials, respectively. We can then compute the number photons in the target image as $x_{\text{inp},i}^k = x_i^k - x_{\text{inp},i}^k$. By changing the value p we can control the number of photons that are on average assigned to the input or target image respectively. The process is illustrated in Figure 3. We use a randomly selected p for each training patch to cover all levels of noise. We show in the Supplementary material that the loss formulation in Eq. 13 can still be used to maximise the likelihood of the training data even when $\mathbf{x}_{\text{tar}}^k$ is not a one-hot encoding of a single photon position but an image that contains an arbitrary number photons. In practice, we use a normalized variant that still maximizes likelihood of the data

$$L(\theta) = - \sum_{k=1}^m \frac{1}{n|\mathbf{x}_{\text{tar}}^k|} \sum_{i=1}^n \ln f_i(\mathbf{x}_{\text{inp}}^k; \theta) x_{\text{tar},i}^k, \quad (14)$$

where $|\mathbf{x}_{\text{tar}}^k|$ is the sum of photons in $\mathbf{x}_{\text{tar}}^k$.

3.6 Inference

MMSE denoising: To compute the MMSE denoising result $\hat{\mathbf{x}}$ for a noisy input image \mathbf{x} , we can simply apply our trained CNN. As shown in section 3.4, the resulting probability distribution corresponds to the MMSE estimate. However, since our network uses a final softmax layer, the output can only tell us about the normalized pixel intensities and not about the absolute ones. To obtain a scaled version, that is comparable to the results from N2V2 or HDN, we multiply our output with the number of photons in the input image.

Diversity denoising: To obtain a sample from the posterior of clean images, given a shot noise corrupted input, we use the iterative procedure illustrated in Figure 2. Starting with the original image, we repeatedly apply our network to obtain the distribution for the next photon position and add photons drawn from this distribution. Even though Eq. 9 contains a product over individual photons and we should in principle draw only a single photon at a time, we find that we can add multiple photons simultaneously while maintaining acceptable quality. In practice, we add 10% of the current photon count in each step, increase the total number of photons exponentially. A more detailed description of photon sampling can be found in the Supplementary materials.

Image generation: To obtain samples from our generative model, we follow the same process as diversity denoising, but start with a blank image.

4 Experiments

4.1 Network Architecture and Training

Here, we will only give a brief overview of the architecture and training procedure used for the experiments on microscopy data. A more detailed description can be found in the Supplementary material.

For all our experiments on microscopy datasets, we use a modified UNet [23] consisting of 6 levels, with a residual block at each level and skip connections. We use 28 feature channels in the first level and double the number of feature channels at each subsequent level. All our networks are trained using the ADAM [24] optimizer for 100 epochs. We use randomly cropped patches of 256×256 pixels, which are augmented 8-fold, using random flips and transpose operations. We use a batch size of 32.

4.2 Baselines

Supervised denoising uses the same network architecture as our method except for the softmax layer at the end. It is trained with the same hyperparameters but uses a MSE loss function. As for our method, we use 8-fold data augmentation.

N2V2 uses the implementation from *et al.* [13], with default hyper parameters and the default 64×64 training

patch size.

HDN uses the implementation from Prakash *et al.* [11], with default hyper parameters and the default 64×64 training patch size. HDN requires a model of the imaging noise, which is usually trained from data. Instead, because we know our data contains pure shot noise, we added an analytical Poisson noise model, accounting for shot noise.

HDN256 uses the implementation from Prakash *et al.* [11] but with increased network complexity to allow for a fairer comparison to our method. Specifically, we increase the dimensionality of the latent variables from 32 feature channels to 70, and the number of deterministic filters in the hidden units from 64 to 140. The method uses 256×256 pixel training patches. We use the same noise model as for HDN.

4.3 Photon Counting Datasets

While a number of denoising datasets are available in the microscopy domain (*e.g.* [25, 26]), none of them show purely shot noise corrupted data. To address this gap, we introduce four new quantitative datasets, including High-SNR ground truth data and one additional qualitative dataset that does not contain ground truth.

We use two photon-counting datasets that will be made available to the community. As a result, the recorded pixel intensities give a very accurate approximation of the photons hitting each pixel during the exposure.

The Conv-PC dataset We image 5 fields of view (FOV) repeatedly, 512 times at a resolution of 512×512 pixels. Each of the individual frames contains a substantial amount of shot noise. By summing the 512 images for each FOV, we obtain the high-SNR version. Four FOVs were used as training data for supervised denoising, the remaining one was used as test data.

The Neuro-PC dataset contains images of mouse neurons. The dataset is created from a z-stack of 2048×2048 pixels by using 2×2 binning in x- and y-direction direction and 4 times binning in the z-direction. We divided the images into non-overlapping 320×320 regions and rejected empty ones. To produce the corresponding low-SNR versions we reduced the photon count in each pixel to simulate a 1000-fold shorter exposure by using a binomial distribution with $p = 0.001$. We use every fourth frame as test set and keep the rest as training set for the

supervised baseline. All in all, this amounts to 133 images of size 320×320 , 33 of which are test images.

4.4 Single Molecule Localisation Microscopy

Single molecule localisation microscopy (SMLM) [27] data is produced differently from photon counting data but is subject to the same type of shot noise corruption. It uses a large set of images of the same field of view to detect and localise individual fluorescent emitters in each image. The resulting emitter locations are then stored in a list and can be binned in x and y to produce a 2D histogram/image containing the number of emitters in each bin/pixel.

The NPC-SM dataset was derived from single molecule localisation data published by Löschberger *et al.* in [28]. It shows the arrangement of the *gp210* protein around the nuclear pore complex (NPC). To create the dataset, we binned the detected emitter locations using a bin size of $20\text{nm} \times 20\text{nm}$ to produce the high-SNR data. To produce the corresponding low-SNR data we randomly reduced the detections by a factor of 20, using a binomial distribution for each pixel with $p = 0.05$. We use every 4th image as test set and keep the rest as training data for the supervised baseline. This amounts to a total of 33 images (24 for training and 9 for testing) of size 280×280 pixels.

The MT-SM dataset was derived from single molecule localisation data published by Jimenez *et al.* in [29]. It shows the arranged cells labeled for microtubules. To create the dataset, we binned the detected emitter locations using a bin size of $28\text{nm} \times 28\text{nm}$ to produce the high-SNR data. To produce the corresponding low-SNR data we randomly reduced the detections by a factor of 200, using a binomial distribution for each pixel with $p = 0.005$. We use every 4th image as test set and keep the rest as training data for the supervised baseline. This gives a total of 120 images (90 for training and 30 for testing) of size 640×640 pixels.

4.5 Denoising Performance

To evaluate the denoising performance of our method we train one network for our method and one for each baseline (N2V2, HDN, and HDN256). Since these methods

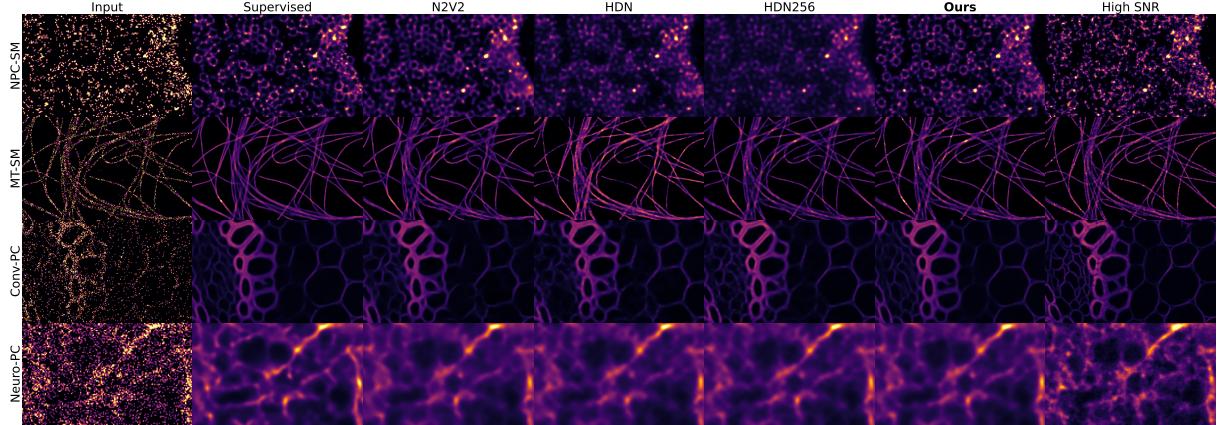


Figure 4: **Qualitative denoising results:** We show MMSE denoising results for all datasets and methods. All baselines except for the supervised one have been trained purely on low-SNR input data.

Table 1: Average peak signal to noise ratios (PSNRs) in dB (higher is better).

	Superv.	N2V	HDN	HDN256	Ours
NPC-SM	39.09	39.00	39.01	38.73	39.17
MT-SM	36.59	36.17	34.32	36.43	36.64
Conv-PC	20.42	23.85	23.77	24.32	24.84
Neuro-PC	31.74	32.61	32.41	32.56	32.63

do not require clean data, we can train them on the full low-SNR data, including the section used for testing. The supervised baseline, which requires clean training data, is trained only on the designated training section of the data. Quantitative and qualitative results can be found in table 1 and Figure 4.

We find that our method is on-par or outperforms the baselines and even the supervised approach. We believe that the reason for this might be that, depending on the data split, supervised methods might suffer from a mismatch between training and test distributions, which might be especially the case for the *Conv-PC* dataset where test and training data consist of different FOVs showing slightly different patterns.

4.6 Diversity Denoising

In Figure 5, we qualitatively evaluate the performance of our method for diversity denoising, that is, its abil-

ity to sample diverse possible clean images from single noisy input. To show the full range of possible results, we trained our method on the high-SNR data of the *Conv-PC* dataset.

We generate six different shot noise corrupted versions of an image at different noise levels/photon numbers and use them as input for the sampling procedure described in Figure 2, to generate three possible clean versions for each noisy input image. Noisy images with low photon counts can be explained by a broad range of possible clean images and yield highly diverse results. Increasing the photon count of the input image, we find that the differences in the sampled clean images become more subtle until only local structures differ.

4.7 Image Generation Performance

Finally, we evaluate our method for the use as a generative image model. We are especially interested in the setting where only low-SNR data is available for training and want to investigate how the distribution of the generated images will compare to the clean high-SNR data. We train our model as well as the HDN and HDN256 baselines on the low-SNR data for each dataset. We generate 10000 images of 256×256 pixels using our method (Figure 2) and HDN256. We then compute the FID [30] score against 10000 random crops of the augmented high SNR-data. We compute the scores using the *clean FID* [31]. For a fair comparison against the HDN baseline, trained

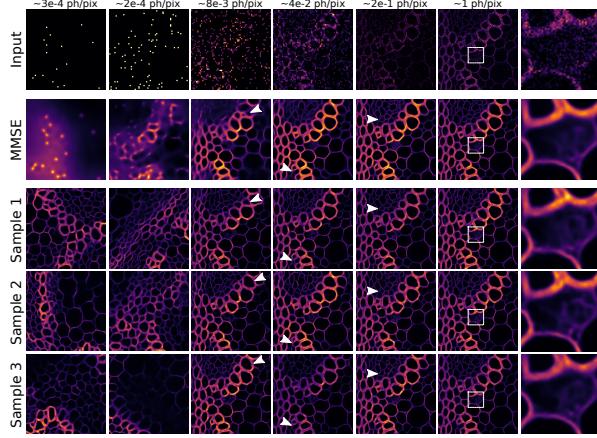


Figure 5: Diverse solutions for shot noise removal: GAP models can be used to remove shot noise by taking the noisy image as starting point and sequentially adding additional photons, until a clean image is produced. Less noisy inputs lead to less diverse predictions as more information about the clean image becomes available. The last column depicts a zoomed in region indicated by the dashed box. Arrows highlight structural differences in the samples.

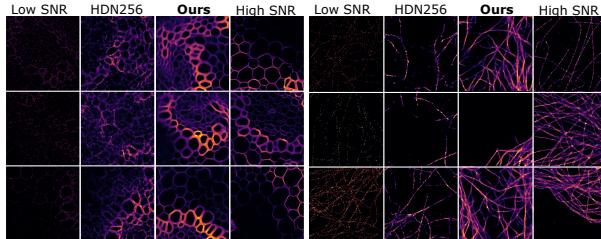


Figure 6: Comparison of sample quality: We show randomly selected generated samples for the *Conv-PC* and *MT-SM* datasets compared to randomly cropped real high- and low-SNR patches.

on 64×64 pixel patches, we compute the FID using 64×64 pixel patches against 64×64 crops of the high-SNR data. To compare against our method, we use random 64×64 crops from the 256×256 pixel patches generated by our method. Quantitative results can be found in Table 2. Qualitative results for 256×256 patches are

Table 2: We show the FID score [30] of 10k generated images compared to 10k random crops of high-SNR data. Note that all methods have been trained on low-SNR training data.

	64x64 pixels		256x256 pixels	
	HDN	Ours	HDN256	Ours
NPC-SM	146.48	146.92	204.34	95.32
MT-SM	130.20	138.72	132.55	84.49
Conv-PC	200.70	134.20	163.28	86.82
Neuro-PC	111.23	115.06	77.90	64.16

shown in Figure 6. We find that our method visually outperforms HDN and HDN256 and consistently achieves lower FID scores for 256×256 patches. For the smaller 64×64 patches FID results are less clear. We believe, that this is due to the fact, that larger structures are not captured at this patch size, and that our high-SNR data contains residual noise, which seems to be better represented by HDN.

5 Discussion and Conclusion

We have introduced a new perspective on shot noise-affected imaging and showed that it can be utilised for self-supervised MMSE denoising, obtaining diverse denoising solutions, and constructing generative models that can be trained with noisy data. We believe that this perspective might open the door to new applications in areas of microscopy where only shot noise-affected data is available. We also believe that our method can be extended to be used in a conditional setting for image-to-image translation, such as the prediction of fluorescence channels from bright-field images – a topic that has received much attention in the recent years [32, 33]. While our method is currently limited to data purely affected by shot noise, we hope that future work can extend the approach to be applicable in a more general setting. Finally, we applied GAP to two natural image datasets (see Supplementary material) with encouraging generative visual results. We believe GAP might be applicable as a generative model beyond microscopy.

Acknowledgements

We would like to thank Jeremy Pike for pointing us the single molecule localisation data and for the helpful discussions we had. The computations described in this paper 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. We used CaStLeS [34] and *Baskerville* resources.

References

- [1] Jaroslav Icha, Michael Weber, Jennifer C Waters, and Caren Norden. Phototoxicity in live fluorescence microscopy, and how to avoid it. *BioEssays*, 39(8):1700003, 2017.
- [2] Romain F Laine, Guillaume Jacquemet, and Alexander Krull. Imaging in focus: An introduction to denoising bioimages in the era of deep learning. *The International Journal of Biochemistry & Cell Biology*, 140:106077, 2021.
- [3] Donal J Denvir and Emer Conroy. Electron-multiplying ccd: the new ICCD. In *Low-Light-Level and Real-Time Imaging Systems, Components, and Applications*, volume 4796, pages 164–174. SPIE, 2003.
- [4] Martin Weigert, Uwe Schmidt, Tobias Boothe, Andreas Müller, Alexandr Dibrov, Akanksha Jain, Benjamin Wilhelm, Deborah Schmidt, Coleman Broaddus, Siân Culley, et al. Content-aware image restoration: pushing the limits of fluorescence microscopy. *Nature methods*, 15(12):1090–1097, 2018.
- [5] Jaakkko Lehtinen, Jacob Munkberg, Jon Hasselgren, Samuli Laine, Tero Karras, Miika Aittala, and Timo Aila. Noise2noise: Learning image restoration without clean data. In *International Conference on Machine Learning*, pages 2965–2974, 2018.
- [6] Alexander Krull, Tim-Oliver Buchholz, and Florian Jug. Noise2void-learning denoising from single noisy images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2129–2137, 2019.
- [7] Joshua Batson and Loic Royer. Noise2self: Blind denoising by self-supervision, 2019.
- [8] 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)*, pages 159–163, 2020.
- [9] Yuhui Quan, Mingqin Chen, Tongyao Pang, and Hui Ji. Self2self with dropout: Learning self-supervised denoising from single image. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1890–1898, 2020.
- [10] Mangal Prakash, Alexander Krull, and Florian Jug. Fully unsupervised diversity denoising with convolutional variational autoencoders. In *International Conference on Learning Representations*, 2020.
- [11] Mangal Prakash, Mauricio Delbracio, Peyman Milanfar, and Florian Jug. Interpretable unsupervised diversity denoising and artefact removal. In *International Conference on Learning Representations*, 2022.
- [12] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *stat*, 1050:1, 2014.
- [13] Eva Höck, Tim-Oliver Buchholz, Anselm Brachmann, Florian Jug, and Alexander Freytag. N2v2-fixing noise2void checkerboard artifacts with modified sampling strategies and a tweaked network architecture. *arXiv preprint arXiv:2211.08512*, 2022.
- [14] Stan Z Li. *Markov random field modeling in image analysis*. Springer Science & Business Media, 2009.
- [15] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *stat*, 1050:10, 2014.
- [16] Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International conference on machine learning*, pages 1530–1538. PMLR, 2015.
- [17] Aaron Van Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *International conference on machine learning*, pages 1747–1756. PMLR, 2016.
- [18] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.
- [19] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2256–2265, Lille, France, 07–09 Jul 2015. PMLR.

- [20] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022.
- [21] Arpit Bansal, Eitan Borgnia, Hong-Min Chu, Jie S Li, Hamid Kazemi, Furong Huang, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Cold diffusion: Inverting arbitrary image transforms without noise. *arXiv preprint arXiv:2208.09392*, 2022.
- [22] Giannis Daras, Mauricio Delbracio, Hossein Talebi, Alexandros G Dimakis, and Peyman Milanfar. Soft diffusion: Score matching for general corruptions. *arXiv preprint arXiv:2209.05442*, 2022.
- [23] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.
- [24] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. 3rd international conference on learning representations, iclr 2015. *arXiv preprint arXiv:1412.6980*, 9, 2015.
- [25] Yide Zhang, Yinhao Zhu, Evan Nichols, Qingfei Wang, Siyuan Zhang, Cody Smith, and Scott Howard. A poisson-gaussian denoising dataset with real fluorescence microscopy images. In *CVPR*, 2019.
- [26] Guy M Hagen, Justin Bendesky, Rosa Machado, Tram-Anh Nguyen, Tanmay Kumar, and Jonathan Ventura. Fluorescence microscopy datasets for training deep neural networks. *GigaScience*, 10(5):giab032, 2021.
- [27] Eric Betzig, George H Patterson, Rachid Sougrat, O Wolf Lindwasser, Scott Olenych, Juan S Bonifacino, Michael W Davidson, Jennifer Lippincott-Schwartz, and Harald F Hess. Imaging intracellular fluorescent proteins at nanometer resolution. *Science*, 313(5793):1642–1645, 2006.
- [28] Anna Löschberger, Sebastian van de Linde, Marie-Christine Dabauvalle, Bernd Rieger, Mike Heilemann, Georg Krohne, and Markus Sauer. Super-resolution imaging visualizes the eightfold symmetry of gp210 proteins around the nuclear pore complex and resolves the central channel with nanometer resolution. *Journal of cell science*, 125(3):570–575, 2012.
- [29] Angélique Jimenez, Karoline Friedl, and Christophe Leterrier. About samples, giving examples: Optimized single molecule localization microscopy. *Methods*, 174:100–114, 2020.
- [30] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.
- [31] Gaurav Parmar, Richard Zhang, and Jun-Yan Zhu. On aliased resizing and surprising subtleties in GAN evaluation. In *CVPR*, 2022.
- [32] Gyuhyun Lee, Jeong-Woo Oh, Mi-Sun Kang, Nam-Gu Her, Myoung-Hee Kim, and Won-Ki Jeong. Deepchs: Bright-field to fluorescence microscopy image conversion using deep learning for label-free high-content screening. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2018: 21st International Conference, Granada, Spain, September 16–20, 2018, Proceedings, Part II 11*, pages 335–343. Springer, 2018.
- [33] Samuel Tonks, Chih Hsu, Steve Hood, Ryan Musso, Ceriden Hopely, Minh Doan, Erin Edwards, Alexander Krull, and Iain Styles. Evaluation of virtual staining for high-throughput screenings. In *20th IEEE International Symposium on Biomedical Imaging*. IEEE, 2023.
- [34] Simon J Thompson, Stephanie EM Thompson, and Jean-Baptiste Cazier. Castles (compute and storage for the life sciences): A collection of compute and storage resources for supporting research at the University of Birmingham. *Zenodo*, 2019.

Image Denoising and the Generative Accumulation of Photons (Supplementary Material)

Alexander Krull¹, Hector Basevi², Benjamin Salmon¹, Andre Zeug³,
Franziska Müller³, Samuel Tonks¹, Leela Muppala¹, and
Aleš Leonardis¹

¹Computer Science, University of Birmingham, UK

²Metabolism and Systems Research, University of Birmingham, UK

³Medizinische Hochschule Hannover, Germany

August 1, 2023

1 Code and Data Availability

The code and datasets will be made publicly available here:
<https://github.com/krulllab/GAP>.

2 Comparing DDPM versus our Generative Model

In Figure 1, we qualitatively compare our results against a denoising diffusion model (DDPM) [1]. While DDPM models achieve impressive sample quality, they are, unlike our method not able to learn the generation of denoised images when trained on noisy data.

3 Additional Details on the Training Procedure

During training, we must ensure that our CNN is able to produce high quality predictions for a range of photon counts. To achieve this, we randomly pick a value for p for each training patch (see section 3.5 in the main paper). To achieve a good coverage over different noise levels/photon counts we use a concept we call pseudo-PSNR, explained in section 3.1. We sample uniformly from a range of pseudo-PSNR numbers and then compute the corresponding value for p accordingly. The process is described in section 3.2

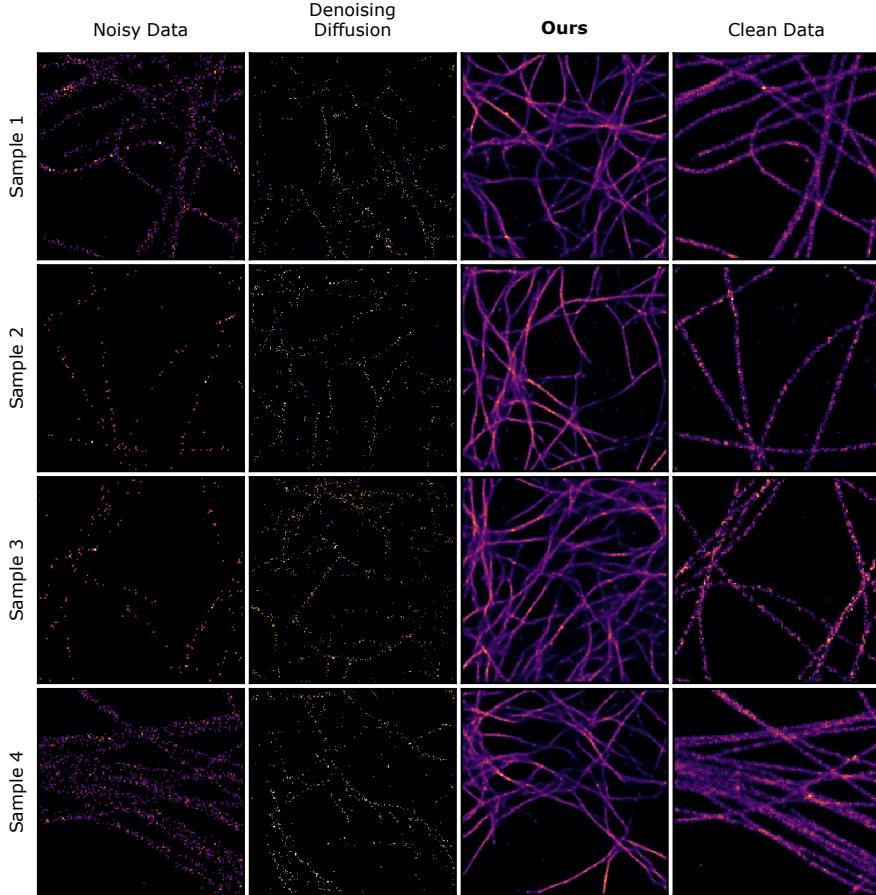


Figure 1: DDPM vs GAP: Randomly generated samples from DDPM and GAP compared to paired clean and noisy samples. DDPM [1] and our method were trained on 256x256 patches sampled from the low-SNR MT-SM dataset, each input was obtained via data augmentation strategies that included random cropping, vertical and horizontal flipping, and random transposing. DDPM was trained with default settings and a batch size of 4 on a single NVIDIA GTX1080Ti for 40,000 iterations.

3.1 The Pseudo-PSNR value

When imaging a static sample the resulting PSNR number of the recorded image depends on the amount of light that was allowed to hit the detector. We can expect a longer exposure time or stronger light intensity to produce a cleaner image with a higher PSNR value than an image recorded with a shorter exposure or reduced light intensity, in which the detector was allowed to collect fewer photons. We define the

intensity of an image as the average number of photons per pixel

$$\gamma = \frac{|\mathbf{x}|}{n}. \quad (1)$$

To compute the PSNR value of a noisy image \mathbf{x} one generally requires the correctly scaled version of the normalised clean ground truth image \mathbf{s} . We can compute the correctly scaled signal as $\bar{\mathbf{s}} = \gamma n \mathbf{s}$, which is then directly comparable to the noisy image \mathbf{x} . The equation used for this is

$$\text{PSNR}(\mathbf{x}, \bar{\mathbf{s}}) = 10 \log_{10} \frac{\bar{s}_{\max}^2}{\text{MSE}}, \quad (2)$$

where \bar{s}_{\max} is the maximum value of the absolute signal $\bar{\mathbf{s}}$, and MSE is the mean squared error between \mathbf{x} and $\bar{\mathbf{s}}$.

The idea of pseudo-PSNR is to directly compute the PSNR value we might expect for a shot noise corrupted image of a certain intensity, without requiring us to compare a noisy and clean image. We define the pseudo-PSNR value for intensity γ as the PSNR value we would expect for the shot noise corrupted version of a flat signal \mathbf{s} , with all pixel values being $s_i = \frac{1}{n}$. Based on the shape of the Poisson distribution, we should expect for such an image $\text{MSE} = \gamma$ and $\bar{s}_{\max} = \sqrt{\gamma}$. Based on Eq. 2, we calculate the pseudo PSNR as

$$\begin{aligned} \text{PSNR}_{\text{ps}}(\gamma) &= 10 \log_{10} \frac{\bar{s}_{\max}^2}{\text{MSE}} \\ &= 10 \log_{10} \frac{\gamma^2}{\gamma} \\ &= 10 \log_{10} \gamma \end{aligned} \quad (3)$$

We can invert Eq. 3 to compute the corresponding intensity γ for a given pseudo PSNR value as

$$\gamma = 10^{\frac{\text{PSNR}_{\text{ps}}}{10}}. \quad (4)$$

3.2 Training Pair Sampling

Before we can split out training patches into input and target using a binomial distribution (see section 3.5 in the main paper), we have to determine the success probability parameter p of the distribution. To achieve this, for each training patch, we first sample from a uniform distribution over pseudo PSNR values between a predefined minimum and maximum.

The goal is to set p , so that the average photon number (intensity) of the resulting input image corresponds to the drawn pseudo PSNR value. We compute the corresponding intensity γ using Eq. 4 from the randomly determined pseudo PSNR value. Then, we compute the corresponding success probability for the binomial distribution as

$$p = \frac{\gamma}{|\mathbf{x}|/n}, \quad (5)$$

such that the input image photon count after the split will correspond to the drawn pseudo PSNR. The result is then clipped to values below 0.99 to guarantee that at least 1% of photons is on average assigned to the target image.

We use the following intervals to sample the pseudo PSNR values: $[-40 : -5]$ for NPC-SM, $[-40, -10]$ for MT-SM, $[-40, -10]$ for Conv-PC, and $[-40, 20]$ for Neuro-PC.

4 Details on Photon Sampling Procedure

Here, we want to discuss the details of the photon sampling procedure. Depending on whether we perform image generation or diversity denoising, we initialise the process with an empty image or a noisy image \mathbf{x}_0 . We then apply our trained CNN to compute the probability distribution over the possible next photon positions

$$\bar{\mathbf{s}}_t = f(\mathbf{x}_t; \theta). \quad (6)$$

Because of our softmax output layer it is guaranteed that each pixel value $\bar{s}_{t,i} \geq 0$ and that

$$\sum_i^n \bar{s}_{t,i} = 1. \quad (7)$$

We then sample a set of new photons represented by the image $\mathbf{x}_t^{\text{new}}$, where each pixel value $\mathbf{x}_{t,i}^{\text{new}}$ holds the number of photons that will be added at location i . Each pixel value $x_{t,i}^{\text{new}}$ is drawn from a Poisson distribution with mean

$$\lambda_{t,i} = n\bar{s}_{t,i}\alpha_t, \quad (8)$$

where α_t controls how many photons will on average be sampled in total in $\mathbf{x}_t^{\text{new}}$. We set as

$$\alpha_t = \max(\beta \sum_i^n x_{t,i}, 1), \quad (9)$$

where the parameter β controls the rate at which the photon number increases on average. In our experiments, we set $\beta = 10\%$. The maximum operation ensures that the number of photons is increasing from the beginning even when starting with an empty image \mathbf{x}_0 . Finally, we compute the next photon count image as

$$\mathbf{x}_{t+1} = \mathbf{x}_t + \mathbf{x}_t^{\text{new}} \quad (10)$$

and repeat the process.

5 Detailed Description of Datasets

Here, we want to give additional information about the photon counting datasets we recorded.

5.1 Conv-PC dataset

Data was recorded from a *Convallaria majalis* rhizome section sample slide at a Leica TCS SP8 TPE DIVE with FALCON and the HC PL IRAPO 25x dipping objective. We used 850 nm excitation at 1% laser power, HyD-RLD detector, emission range of 600 – 650 nm, pixel size 0.6 x 0.6 μm , an 8 MHz resonant scanner and 4x averaging. We imaged 5 fields of view (FOVs), each containing 512x512x2048 voxels (xyt). Four FOVs were used as training data for supervised denoising, and the fifth FOV was used as test data.

We used a time binning of 4, resulting in datasets of 512x512x512 voxels, and named these low-SNR raw data 'trainingData.tif' and 'testData.tif' respectively. Furthermore, we summed these 512 frames to produce the high SNR ('ground truth') version and named it 'trainingDataGT.tif' and 'testDataGT.tif'. Each frame of the raw data contains a significant amount of image noise. All voxel values correspond to photon counts.

5.2 Neuro-PC dataset

Data was obtained from 11–14 week old male Mice (C57BL/6J background) stereotactically injected with AAV-hGFAP-5-HT4R-eGFP and AAV-hGFAP-tdTomato to the CA1 region of the hippocampus 3 weeks prior to experiment. Data was recorded from acute slices of the mouse hippocampus region at a Leica TCS SP8 TPE DIVE with FALCON, using following acquisition settings: HC PL IRAPO 25x dipping objective, excitation 920 nm at 15–30%, HyD-RLD detector, emission 490 – 560 nm (eGFP), and 560 – 650 nm (tdTomato), voxel size (0.1 x 0.1 x 0.5 μm), 4x averaging, scan speed 600 Hz. We are using only the tdTomato channel.

6 Network Architecture

We use a modified UNet [2] architecture, with skip connections and residual blocks. Each residual down-block and up-block consists of 3 3x3 convolutions with RELU activation functions after the second convolutions and at the end of the block. We use max-pooling for down-sampling and transposed convolutions for up-sampling.

Since we are training a single network to handle a range of different noise levels and photon counts in its input, normalizing the input is not trivial. To avoid normalization, we use a sinusoidal frequency encoding [3] applied to each pixel value at the input of our network. We use 10 different sinusoids with frequencies at different powers of 10.

7 Hyper Parameters

Since our datasets have differing sizes, we define one training epoch as 500 training steps. We use the first 90% of images in each dataset as training data and the last 10% as validation set. We use the ADAM optimizer [4]. We use an initial learning rate of 1e–4 and reduce the learning rate using the pytorch *ReduceLROnPlateau* scheduler with a patience of 10 by a factor of 2.

8 Details on the Loss Function

Here we show that the loss function from Eq. 13 in the main paper, which uses target images $\mathbf{x}_{\text{tar}}^k$ with multiple photons is equivalent to using single photons represented by one-hot-encoding images. We can write the loss as

$$\begin{aligned} L(\theta) &= - \sum_{k=1}^m \sum_{i=1}^n \ln f_i(\mathbf{x}_{\text{inp}}^k; \theta) x_{\text{tar},i}^k \\ &= - \sum_{k=1}^m \sum_{i=1}^n \ln f_i(\mathbf{x}_{\text{inp}}^k; \theta) \sum_{t=1}^T x_{\text{tar},i}^{k,t} , \end{aligned} \quad (11)$$

where $\sum_{t=1}^T x_{\text{tar},i}^{k,t}$ is the one-hot-encoding photon image for photon t from $\mathbf{x}_{\text{tar}}^k$. Note that the order of photons does not matter here. We can then continue to write

$$\begin{aligned} L(\theta) &= - \sum_{k=1}^m \sum_{i=1}^n \ln f_i(\mathbf{x}_{\text{inp}}^k; \theta) \sum_{t=1}^T x_{\text{tar},i}^{k,t} \\ &= - \sum_{k=1}^m \sum_{t=1}^T \sum_{i=1}^n \ln f_i(\mathbf{x}_{\text{inp}}^k; \theta) x_{\text{tar},i}^{k,t} . \end{aligned} \quad (12)$$

In this formulation it becomes clear that using target images with multiple photons is equivalent to replicating each input image T times and using it together with each of the corresponding single-photon target images. This corresponds to the same training data distribution as randomly sampling single photon targets.

9 Additional qualitative results

We show randomly selected samples for all datasets in Figure 2.

10 Natural image datasets

To demonstrate the potential of our method, we show randomly selected outputs of our generative model when applied on natural image datasets in Figure 3. To account for the greater complexity of these datasets, we trained 8 expert networks, each specialised on a sub-range of pseudo PSNR values. Each network is scaled up to 7 levels (instead of 6) and starting with 32 feature (instead of 28) channels. When generating images we switch between these expert networks as the image gains more and more photons. Apart from this, the approach is the same as for the microscopy data.

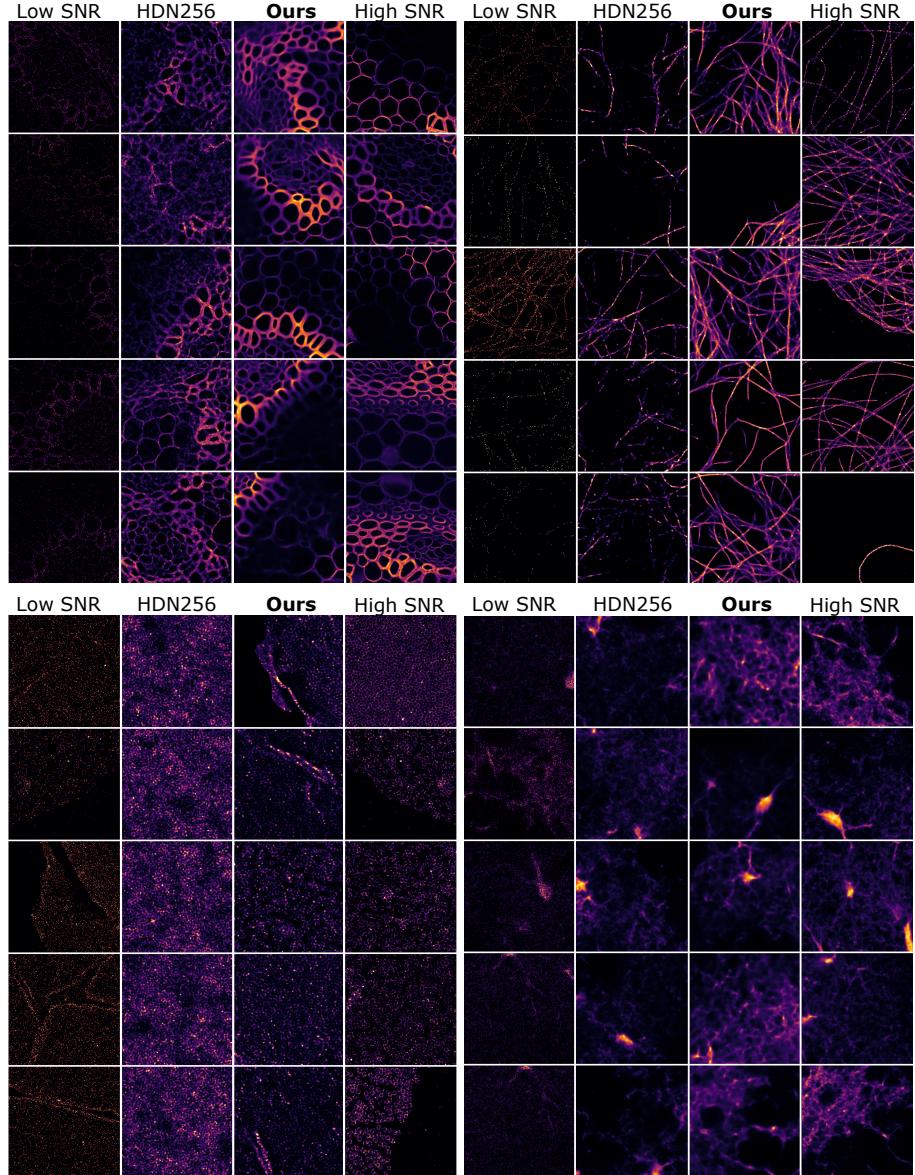


Figure 2: **Comparison of sample quality:** We show randomly selected generated samples for all datasets (top-left to bottom-right: *Conv-PC*, *MT-SM*, *NPC-SM*, *Neuro-PC*) compared to randomly cropped real high- and low-SNR patches.

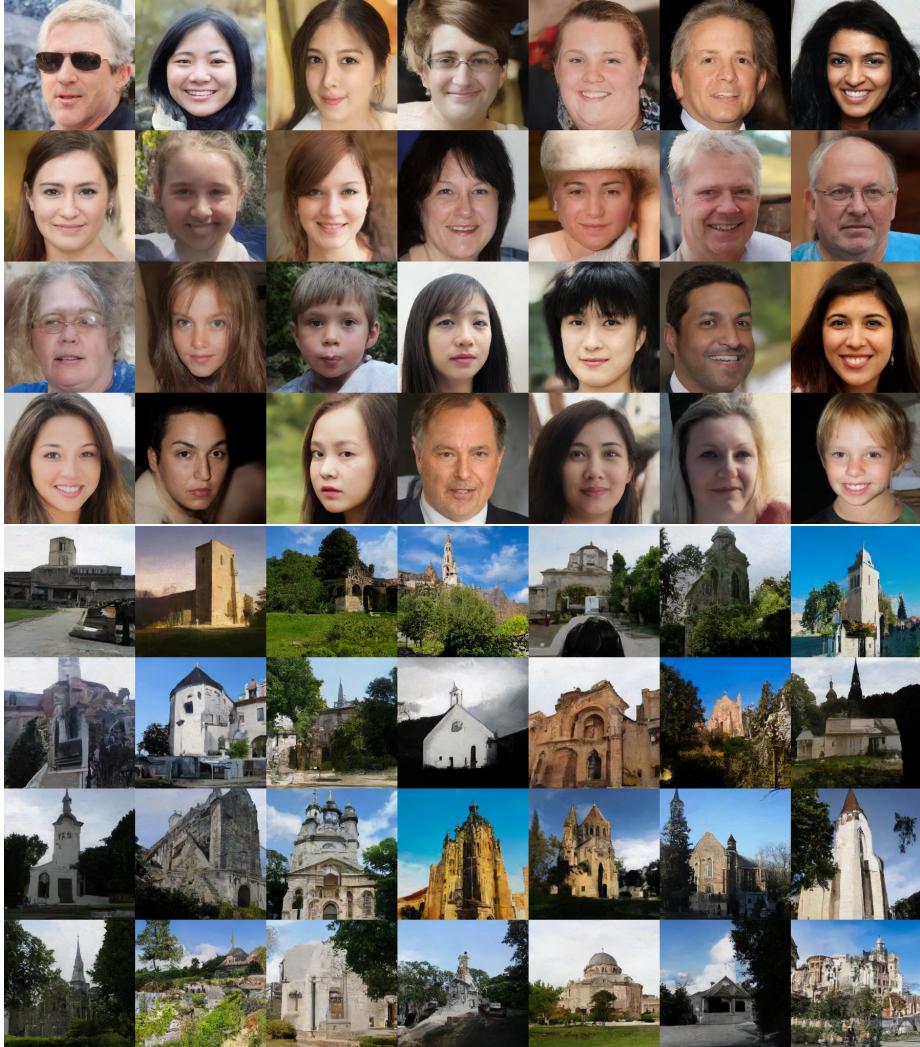


Figure 3: Randomly selected sample images generated by our model. Results of our method when trained on the 256×256 pixel versions of FFHQ dataset [5] and the LSUN-churches dataset [6].

References

- [1] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.
- [2] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.
- [3] Matthew Tancik, Pratul P Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, pages 7537–7547, 2020.
- [4] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. 3rd international conference on learning representations, iclr 2015. *arXiv preprint arXiv:1412.6980*, 9, 2015.
- [5] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019.
- [6] Fisher Yu, Yinda Zhang, Shuran Song, Ari Seff, and Jianxiong Xiao. LSUN: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.