

Self-Supervised Poisson-Gaussian Denoising

Wesley Khademi¹, Sonia Rao², Clare Minnerath³, Guy Hagen⁴, and Jonathan Ventura¹

¹California Polytechnic State University ²University of Georgia ³Providence College ⁴University of Colorado Colorado Springs

1{wkhademi, jventu09}@calpoly.edu

Abstract

We extend the blindspot model for self-supervised denoising to handle Poisson-Gaussian noise and introduce an improved training scheme that avoids hyperparameters and adapts the denoiser to the test data. Self-supervised models for denoising learn to denoise from only noisy data and do not require corresponding clean images, which are difficult or impossible to acquire in some application areas of interest such as low-light microscopy. We introduce a new training strategy to handle Poisson-Gaussian noise which is the standard noise model for microscope images. Our new strategy eliminates hyperparameters from the loss function, which is important in a self-supervised regime where no ground truth data is available to guide hyperparameter tuning. We show how our denoiser can be adapted to the test data to improve performance. Our evaluations on microscope image denoising benchmarks validate our approach.

Fluorescence microscopy is a vital tool for understanding cellular processes and structures. Because fluorescence imaging with long exposure times or intense illumination may damage the cell sample through phototoxicity, fluorescence microscopy images are typically acquired under photon-limited conditions. However, safely imaging the cell using low light conditions and/or low exposure times unfortunately lowers the signal-to-noise ratio (SNR), hindering further analysis and interpretation of the resulting images.

The SNR is the product of a combination of factors, including exposure time, excitation intensity, and camera characteristics. In fluorescence microscopy, the noise is typically described by a Poisson-Gaussian model [6]. The goal of image denoising is to computationally increase the



Figure 1: An example of our self-supervised denoising result. Image from the Confocal Mice dataset [26].

image SNR (Figure 1). In contrast to traditional methods [4, 5, 2, 15, 3, 7] which denoise based on only the input image, learning-based methods learn to denoise from a dataset of example images.

In recent years, deep learning methods using convolutional neural networks have shown significant promise in learning-based fluorescence microscopy image denoising [25, 24]. However, the supervised approach to learning denoising faces practical limitations because it requires a large number of corresponding pairs of low SNR and high SNR images. When imaging live cells, for example, it is not possible to acquire paired low and high SNR images for training because a) the sample is moving and b) exposure to light causes photobleaching and ultimately kills the sample.

For these reasons, researchers have turned to selfsupervised approaches to denoising [21, 1, 9, 11]. In the self-supervised setting, the learner only has access to low SNR images. Of the recent approaches, blindspot neural networks [11] have shown the best performance. In this work, we address two shortcomings of blindspot neural networks for self-supervised denoising:

- We introduce a loss function appropriate for Poisson-Gaussian noise which is the standard model for microscope images;
- 2. We introduce an alternate training strategy which eliminates the need to regularize the loss function; this is critical in the self-supervised setting where no ground truth validation data is available to tune the regularization strength.

In the following, we survey related work on selfsupervised denoising (Section 1), review the blindspot neural network approach to self-supervised denoising (Section 2), introduce our new uncalibrated approach (Section 3), present the results of our evaluation and comparison to competing methods on benchmark datasets (Section 4), and provide conclusions and directions for future work (Section 5).

1. Related Work

1.1. Traditional methods

Many traditional methods for denoising such as BM3D [5], non-local means [3], and weighted non-nuclear norm minimizaiton [7] perform denoising by comparing the neighborhood of a pixel to other similar regions in the image. The advantage of learning-based methods is that they can also take advantage of examples from other images in the dataset beyond the input image to be denoised. Other methods such as total-variation denoising [4] enforce smoothness priors on the image which tend to lead to highly quantized results.

While most previous methods for denoising are designed for additive Gaussian noise; in the case of Poisson-Gaussian noise, a variance stabilizing transform [16] is applied to approximately transform the noise to be Gaussian. However, these methods are designed explicitly for Poisson-Gaussian noise [15].

1.2. Deep learning methods

At present, supervised deep learning methods for denoising [25, 24] typically far outperform traditional and selfsupervised methods in terms of peak signal-to-noise ratio (PSNR). Most supervised methods apply a fully convolutional neural network [14, 20] and simply regress to the clean image. Recently, several approaches to self-supervised denoising have been developed. Some methods [22] use as a loss function Stein's Unbiased Risk Estimate (SURE) [23, 19], which estimates the mean squared error (MSE) between a denoised image and the clean image without actually having access to the clean image. An analogous estimator for Poisson-Gaussian noise has been developed [12]. However, these methods require *a priori* knowledge of the noise level which is unrealistic in a practical setting. Our approach supports blind denoising and adaptively estimates the noise level at test time.

Lehtinen et al. [13] introduced a highly successful approach to self-supervised denoising called Noise2Noise. In this approach, the network learns to transform one noisy instantiation of a clean image into another; under the MSE loss function, the network learns to output the expected value of the data which corresponds to the clean image. While this method can achieve results very close to a supervised method, it requires multiple, corresponding noisy images and thus is similarly limited in application in the live cell microscopy context.

An alternate approach to self-supervised denoising which does not require multiple noise instantiations of the same clean image is to learn a filter which predicts the center pixel of the receptive field based on the surrounding neighborhood of noisy pixels. By training such a filter to minimize the MSE to the noisy input, the resulting filter will theoretically output the clean value [1, 9]. Laine et al. [11] refer to a neural network built around this concept as a "blindspot neural network." They improved upon the blindspot concept by extending it to a Bayesian context and introduced loss functions for pure Gaussian or Poisson noise, showing results very close to the supervised result when trained on synthetically noised data. However, their method requires a regularization term in the loss function which can't practically be tuned in the self-supervised setting; in our evaluation we found that the regularization strength indeed needs to be tuned for best results on different datasets. Our method avoids the need for regularization and outperforms the regularized version in our experiments.

Krull et al. [10] introduced Probabilistic Noise2Void (PN2V) which takes a non-parametric approach to modeling both the noise distribution and the network output; however, their approach requires paired clean and noisy images in order to calibrate the noise model. A recent follow-on work called PPN2V [18] estimates the noise model using a Gaussian Mixture Model (GMM) in a fully unsupervised manner. Again, this approach involves several hyperparameters controlling the complexity of the noise model which need to be tuned, while ours does not. Additionally, in our experiments, we show that our approach outperforms PPN2V on several datasets.

2. Self-supervised learning of denoising

The goal of denoising is to predict the values of a "clean" image $\mathbf{x} = (x_1, \ldots, x_n)$ given a "noisy" image $\mathbf{y} = (y_1, \ldots, y_n)$. Let Ω_{y_i} denote the neighborhood of pixel y_i , which does not include y_i itself. We make two assumptions that are critical to the setup of Noise2Void [9] and follow-on works: that the noise at each pixel is sampled independently, i.e. $p(y_i|x_1, \ldots, x_n) = p(y_i|x_i)$; and that each clean pixel is dependent on its neighborhood, a common assumption about natural images. The consequence of these assumptions is that Ω_{y_i} only gives information about x_i , not y_i . Therefore a network trained to predict y_i given Ω_{y_i} using a mean squared error loss will in fact learn to predict x_i [9, 1].

In this work, we take a probabilistic approach rather than trying to regress to a single value. Following Laine et al. [11] and Krull et al. [10], we can connect y_i to its neighborhood Ω_{y_i} by marginalizing out the unknown clean value x_i :

$$\underbrace{p(y_i|\Omega_{y_i})}_{\text{Noisy observation}} = \int \underbrace{p(y_i|x_i)}_{\text{Noise model}} \underbrace{p(x_i|\Omega_{y_i})}_{\text{Clean prior}} dx_i.$$
(1)

Since we only have access to observations of y_i for training, this formulation allows us to fit a model for the clean data by minimizing the negative log likelihood of the noisy data, i.e. minimizing a loss function defined as

$$\mathcal{L}^{\text{marginal}} = \sum_{i} -\log p(y_i | \Omega_{y_i}).$$
(2)

In the following we will drop the Ω_{y_i} to save space.

2.1. Poisson-Gaussian noise

In the case of Poisson-Gaussian noise, the noisy observation y_i is sampled by first applying Poisson corruption to x_i and then adding Gaussian noise which is independent of x_i . We have

$$y_i = aP(x_i/a) + N(0,b)$$
 (3)

where a > 0 is a scaling factor (related to the gain of the camera) and b is the variance of the Gaussian noise component, which models other sources of noise such as electric and thermal noise [6].

We apply the common approximation of the Poisson distribution as a Gaussian with equal mean and variance:

$$y_i \approx aN(x_i/a, x_i/a) + N(0, b) \tag{4}$$

$$= N(x_i, ax_i + b). \tag{5}$$

The noise model is then simply a Gaussian noise model whose variance is an affine transformation of the clean value. Note that in practice we allow b to be negative; this models the effect of an offset or "pedestal" value in the imaging system [6]. This general formulation encompasses both pure Gaussian (a = 0) and Poisson noise (b = 0).

2.2. Choice of prior

In order to implement our loss function (Equation 2) we need to choose a form for the prior $p(x_i|\Omega_{y_i})$ that makes the integral tractable. One approach is to use the conjugate prior of the noise model $p(y_i|x_i)$, so that the integral can be computed analytically. For example, Laine et al. [11] model the prior $p(x_i|\Omega_{y_i})$ as a Gaussian, so that the marginal is also a Gaussian. Alternatively, Krull et al. [10] take a nonparametric approach and sample the prior.

In this work, similar to Laine et al. [11] we model the prior as a Gaussian with mean μ_i and variance σ_i^2 . We replace the ax term in Equation 4 with $a\mu$ to make the integral in Equation 2 tractable; this approximation should be accurate as long as σ_i^2 is small. The marginal distribution of y_i is then

$$p(y_i) = \frac{1}{\sqrt{2\pi(a\mu_i + b + \sigma_i^2)}} \exp\left(-\frac{(y_i - \mu_i)^2}{2(a\mu_i + b + \sigma_i^2)}\right)$$
(6)

and the corresponding loss function is

$$\mathcal{L}^{\text{marginal}} = \sum_{i} \left(\frac{(y_i - \mu_i)^2}{a\mu_i + b + \sigma_i^2} + \log(a\mu_i + b + \sigma_i^2) \right)$$
(7)

2.3. Posterior mean estimate

At test time, μ_i is an estimate of the clean value x_i based on Ω_{y_i} , the neighborhood of noisy pixels around y_i . However, this estimate does not take into account the actual value of y_i which potentially provides useful information about x_i .

Laine et al. [11] and Krull et al. [10] suggest to instead use the expected value of the posterior to maximize the PSNR of the resulting denoised image. In our case we have

$$\hat{x}_i = \mathbb{E}[p(x_i|y_i)] = \frac{y_i \sigma_i^2 + (a\mu_i + b)\mu_i}{a\mu_i + b + \sigma_i^2}.$$
 (8)

Intuitively, when the prior uncertainty is large relative to the noise estimate, the formula approaches the noisy value y_i ; when the prior uncertainty is small relative to the noise estimate, the formula approaches the prior mean μ_i .

2.4. Blindspot neural network

In our approach, μ_i and σ_i^2 are the outputs of a blindspot neural network [11] and *a* and *b* are global parameters learned along with the network parameters.

The "blind-spot neural network" is constructed in such a way that the network cannot see input y_i when outputting the parameters for $p(x_i)$. The blindspot effect can be achieved in multiple ways. Noise2Void [9] and Noise2Self [1] replace a random subset of pixels in each batch and mask out those pixels in the loss computation. Laine et al. [11] instead construct a fully convolutional neural network in such

a way that the center of the receptive field is hidden from the neural network input. In our experiments we use the same blindspot neural network architecture as Laine et al. [11].

2.5. Regularization

In a practical setting, the parameters a and b of the noise model are not known *a priori*; instead, we need to estimate them from the data. However, an important issue arises when attempting to learn the noise parameters along with the network parameters: the network's prior uncertainty and noise estimate are essentially interchangeable without any effect on the loss function. In other words, the optimizer is free to increase a and b and decrease σ_i^2 , or vice-versa, without any penalty. To combat this, we add a regularization term to the per-pixel loss which encourages the prior uncertainty to be small:

$$\mathcal{L}^{\text{regularized}} = \mathcal{L}^{\text{marginal}} + \lambda \sum_{i} |\sigma_i|. \tag{9}$$

We found in our experiments that the choice of λ strongly affects the results. When λ is too high, the prior uncertainty is too small, and the results are blurry. When λ is too low, the prior uncertainty is too high, and the network does not denoise at all. Unfortunately, in the self-supervised setting, it is not possible to determine the appropriate setting of λ using a validation set, because we do not have ground truth "clean" images with which to evaluate a particular setting of λ .

3. Learning an uncalibrated model

This realization led us to adopt a different training strategy which defers the learning of the noise parameter models to test time.

In our uncalibrated model, we do not separate out the parameters of the noise model from the parameters of the prior. Instead, we learn a single variance value $\hat{\sigma_i}^2$ representing the total uncertainty of the network. Our uncalibrated loss function is then

$$\mathcal{L}^{\text{uncalibrated}} = \sum_{i} \left(\frac{(y_i - \mu_i)^2}{\hat{\sigma_i}^2} + \log(\hat{\sigma_i}^2) \right)$$
(10)

At test time, however, we need to know the noise parameters a and b in order to compute $\sigma_i^2 = \hat{\sigma_i}^2 - a\mu_i - b$ and ultimately compute our posterior mean estimate \hat{x}_i .

If we had access to corresponding clean and noisy observations (x_i and y_i , respectively) then we could fit a Poisson-Gaussian noise model to the data in order to learn a and b. In other words, we would find

$$a, b = \arg\min_{a,b} \sum_{i} \left(\frac{(y_i - x_i)^2}{ax_i + b} + \log(ax_i + b) \right).$$
(11)

As we are in a self-supervised setting, however, we do not have access to clean data. Instead, we propose to use the prior mean μ_i as a stand-in for the actual clean value x_i . This bootstrapping approach is similar to that proposed by Prakash et al. [18]; however, they fit a general parametric noise model to the training data where as we propose to fit a Poisson-Gaussian model to each image in the test set.

Our approach is summarized in the following steps:

- 1. Train a blindspot neural network to model the noisy data by outputting a mean and variance value at each pixel, using the uncalibrated loss (Equation 10).
- 2. For each test image:
 - i. Run the blindspot neural network with the noisy image as input to obtain mean μ_i and total variance $\hat{\sigma_i}^2$ estimate at each pixel.
 - ii. Determine the optimal noise parameters a, b by fitting a Poisson-Gaussian distribution to the noisy and psuedo-clean images given by the mean values of the network output (Equation 12).
 - iii. Calculate the prior uncertainty at each pixel as $\sigma_i^2 = \max(0.0001, \hat{\sigma_i}^2 a\mu_i b).$
 - iv. Use the noise parameters a, b and the calculated prior uncertainties σ_i^2 to compute the denoised image as the posterior mean estimate (Equation 8).

We believe our approach has two theoretical advantages over the bootstrap method proposed by Prakash et al. [18]. We can achieve a better fit to the data by training our system end-to-end, whereas Prakash et al. [18] impose a fixed noise model during training by first estimating the noise parameters and then training the network. Second, we estimate noise parameters for each image separately at test time, whereas Prakash et al. [18] estimate common noise parameters for all images first and fixes those parameters during training. Our approach allows for slight deviations in the noise parameters for each image, which might be more realistic for an actual microscope imaging system where the camera configuration slightly fluctuates between images.

4. Experiments and Results

4.1. Implementation details

Our implementation uses the Keras library with Tensorflow backend. We use the same blindspot neural network architecture as Laine et al. [11]. We use the Adam optimizer [8] with a learning rate of 0.0003 over 300 epochs, halving the learning rate when the validation loss plateaued. Each epoch consists of 50 batches of 128×128 crops from random images from the training set. For data augmentation

	Ground Truth	Bootstrap	Ground Truth	Bootstrap	Ground Truth	Bootstrap
Datasets	Loss		a		b	
Confocal Mice	-6.322	-6.257	0.0181	0.0196	-0.000203	-0.000232
Confocal Fish	-5.224	-5.122	0.0753	0.0723	-0.00101	-0.00084
Two-Photon Mice	-5.005	-4.979	0.0301	0.0296	-0.000559	-0.000491

Table 1: Quantitative comparison of fitting a Poisson-Gaussian noise model using the ground truth clean data or the denoised estimate from the prior. Values in the table are averages over the 50 test images from each dataset.

		Ground Truth	Estimated	Foi et al. [6]	Ground Truth	Estimated	Foi et al. [6]
λ	σ		a			b	
50	10	0.0200	0.0250	0.0252	0.00153	0.00135	0.000900
40	20	0.0250	0.0325	0.0759	0.00615	0.00586	0.000334
30	30	0.0333	0.0430	0.0971	0.0138	0.0134	0.00108
20	40	0.0500	0.0623	0.213	0.0246	0.0241	0.000373
10	50	0.100	0.117	0.313	0.0384	0.0378	0.00462

Table 2: Quantitative comparison of fitting a Poisson-Gaussian noise model on different noise levels of our synthetic Confocal MICE dataset. *Ground truth a,b* parameters correspond to the Poisson-Gaussian noise levels added to our synthetic dataset, *Estimated* represents the *a,b* parameters obtained using our bootstrapping technique, and *Foi et al.* represents the *a,b* parameters obtained using the noise estimation method in [6].

we apply random rotation (in multiples of 90 degrees) and horizontal/vertical flipping.

To fit the Poisson-Gaussian noise parameters at test time, we apply Nelder-Mead optimization [17] with (a = 0.01, b = 0) as the initialization point. We cut off data in the bottom 2% and top 3% of the noisy image's dynamic range before estimating the noise parameters.

4.2. Datasets

4.2.1 Synthetic Data

We generate a synthetic dataset using the ground truth images of the Confocal MICE dataset from the FMD benchmark [26] (described below). For training, we use the ground truth images from 19 of the 20 views and generate 50 noisy examples of each view by synthetically adding Poisson-Gaussian noise to the ground truth images using equation 3 where $a = 1/\lambda$ and $b = (\sigma/255)^2$. For testing, we use the ground truth image from the 20th view and generate 50 noisy examples by synthetically adding Poisson-Gaussian noise in the same manner as during training. To ensure our method works for a wide range of noise levels, we train/test our method on all combinations $(\lambda, \sigma) \in \{0, 10, 20, 30, 40, 50\} \times \{0, 10, 20, 30, 40, 50\}$.

4.2.2 Real Data

We evaluated our method on two datasets consisting of real microscope images captured with various imaging setups and types of samples. Testing on real data gives us a more

		Pseudo-Clean	Uncalibrated	Ground Truth	
λ	σ	PSNR	PSNR	PSNR	
50	10	36.95	37.05	37.25	
40	20	35.45	35.53	35.73	
30	30	34.10	34.15	34.33	
20	40	33.17	33.22	33.37	
10	50	31.98	32.03	32.14	

Table 3: PSNR comparison on different noise levels of our synthetic Confocal MICE dataset. *Pseudo-clean* is the result obtained before computing the posterior. *Uncalibrated* is the result from computing the posterior with the estimated noise parameters obtained from our bootstrapping technique. *Ground truth* represents computing the posterior with the true *a*,*b* parameters that correspond to the Poisson-Gaussian noise levels added into our synthetic dataset.

accurate evaluation of our method's performance in contrast to training and testing on synthetically noised data, since real data is not guaranteed to follow the theoretical noise model.

The fluoresence microscopy denoising (FMD) benchmark [26] consists of a total of 12 datasets of images captured using either a confocal, two-photon, or widefield microscope. We used the same subset of datasets (Confocal Mice, Confocal Fish, and Two-Photon Mice) used to evaluate PN2V [10] so that we could compare our results. Each dataset consists of 20 views of the sample with 50 noisy



Figure 2: Comparison of Poisson-Gaussian noise models fit to a noisy image from several datasets. Solid bars show histograms of the noisy values corresponding to a clean value of 20 (blue) and 50 (red). Curves show the pdfs of a Poisson-Gaussian distribution fit to the data using either ground truth clean data (dashed line) or pseudo-clean data (solid line).

Methods	λ	Confocal Mice	Confocal Zebrafish	Two-Photon Mice
Uncalibrated	-	37.97	32.26	33.83
Regularized	0.1	37.74	23.97	33.52
Regularized	1	37.64	27.44	33.56
Regularized	10	37.13	31.99	33.34

Table 4: Comparison between uncalibrated and regularized methods.

images per view. The 19th-view is withheld for testing, and the ground truth images are created by averaging the noisy images in each view. We trained a denoising model on the raw noisy images in each dataset separately.

Prakash et al. [18] evaluated PPN2V on three sequences from a confocal microscope, imaging Convallaria, Mouse Nuclei, and Mouse Actin. Each sequence consists of 100 noisy images and again the clean image is computed as the average of the noisy images. Whereas the FMD dataset provides 8-bit images clipped at 255, these images are 16-bit and thus are not clipped. Following their evaluation procedure, each method is trained on all 100 images and then tested on a crop of the same 100 images; this methodology is allowable in the self-supervised context since no label data is used during training.

4.3. Experiments

In the following we will refer to the competing methods under consideration as

- **Regularized** (Ours): Blindspot neural network trained using the regularized Poisson-Gaussian loss function (Equation 2) with regularization strength λ .
- Uncalibrated (Ours): Blindspot neural network trained using the uncalibrated loss function (Equation 10) with noise parameter estimation done adaptively at test time (Section 3).

- N2V: Noise2Void which uses the MSE loss function and random masking to create the blindspot effect [9].
- **PN2V**: Probabilistic Noise2Void same setup as N2V but uses a histogram noise model created from the ground truth data and a non-parametric prior [10].
- **Bootstrap GMM** and **Bootstrap Histogram**: PPN2V training same setup as PN2V but models the noise distribution using either a GMM or histogram fit to the Noise2Void output [18].
- U-Net: U-Net [20] trained for denoising in a supervised manner using MSE loss [24].
- N2N: Noise2Noise training using MSE loss [13].

4.3.1 Noise parameter estimation

We first evaluate whether our bootstrap approach to estimating the Poisson-Gaussian noise parameters is accurate in comparison to estimating the noise parameters using the actual ground truth clean values.

To evaluate our bootstrapping method, we compare the ground truth and estimated Poisson-Gaussian noise models fit for a test image in each dataset in the FMD benchmark [26]. Figure 2 shows that the Poisson-Gaussian pdfs generated using our bootstrapping technique closely match that of the Poisson-Gaussian pdfs generated from the ground truth images.

We further evaluate our approach by comparing the loss and estimated Poisson-Gaussian noise parameters obtained when using actual ground truth data or the pseudo-clean data generated in our bootstrap method. Table 1 shows that bootstrapping can provide an accurate estimation of noise parameters and result in a loss similar to that obtained from using ground truth clean data. Here the loss value is

$$\frac{1}{N}\sum_{i}\left(\frac{(y_i-x_i)^2}{ax_i+b} + \log(ax_i+b)\right)$$
(12)

	Confocal	Confocal	Two-Photon			
Methods	Mice	Zebrafish	Mice	Convallaria	Mouse Nuclei	Mouse Actin
Uncalibrated (Ours)	37.97	32.26	33.83	36.44	36.97	33.35
N2V	37.56	32.10	33.42	35.73	35.84	33.39
Bootstrap GMM	37.86	*	33.77	36.70	36.43	33.74
Bootstrap Histogram	36.98	32.23	33.53	36.19	36.31	33.61
PN2V	38.24	32.45	33.67	36.51	36.29	33.78
U-Net	38.38	32.93	34.35	36.71	36.58	34.20
N2N	38.19	32.93	34.33	-	-	-

Table 5: Quantitative comparison of our implementation and baseline methods on datasets provided by Zhang et al. [26] and Prakash et al. [18]. Methods above the solid line are fully unsupervised while those below it either require ground truth data or a noisy image pair. Bold numbers indicate the best performing method among the fully unsupervised methods. The * indicates a case where the Bootstrap GMM method failed to train (the loss became NaN before convergence).



Figure 3: Denoising results on images taken from the FMD dataset [26]. The missing image corresponds to the case where the Bootstrap GMM method failed to train.

where y_i is a pixel from the noisy image and x_i is a corresponding pixel from either the ground truth clean image or the pseudo-clean image.

We perform a similar evaluation on our synthetic dataset where instead of having to estimate the true noise parameters from fitting a Poisson-Gaussian noise model with the ground truth clean image we readily have available the true noise parameters that correspond to the level of synthetically added Poisson-Gaussian noise. Table 2 shows the true noise parameters as well as the ones obtained using our uncalibrated method and the method described by Foi et al. in [6]. Our method tends to overestimate the *a* parameter, whereas the estimate of the *b* parameter is consistently accurate. This is probably because a majority of the pixels in the Confocal MICE images are dark and thus there are not many good samples for fitting the level of Poisson noise, whereas every pixel can be effectively used to estimate the Gaussian noise no matter the underlying brightness. Unlike the method of Foi et al. [6] which obtains poor noise estimates most likely because of this, our method is still able to obtain a good estimate of the parameters by leveraging information from both the noisy image and our pseudo-clean image.

The effectiveness of fitting a Poisson-Gaussian noise model at test time is further evaluated in Table 3 which provides a comparison of peak signal-to-noise ratio (PSNR) on a subset of our synthetic dataset. Our method of estimating the noise parameters with our bootstrapping technique consistently improves the denoised results of the pseudo-clean image, but is ultimately bounded by the result obtained from



Figure 4: Visual comparison of results obtained by our proposed method and the fully unsupervised methods listed in Table 5. Noisy test images taken from PPN2V benchmark datasets [18].

using the pseudo-clean image along with the true noise parameters. Results for all noise parameter combinations are given in the supplemental material.

parison of denoising results on both benchmark datasets are shown in Figures 3 and 4.

4.3.2 Effect of regularization

To highlight the difficulties of hyperparameter tuning in the self-supervised context, we trained our uncalibrated model and several regularized models on the FMD datasets. We tested a regularization strength of $\lambda = 0.1, 1$, and 10.

The results are shown in Table 4. The test set PSNR of the regularized model varies greatly depending on the setting of λ , and indeed a different setting of λ is optimal for each dataset. This indicates that hyperparameter tuning is critical for the regularized approach, but it is not actually possible in a self-supervised context.

In contrast, our uncalibrated method outperforms the regularized method at any setting of λ , and does not require any hyperparameters.

4.3.3 Comparison to state-of-the-art

Next we present the results of our performance evaluation on the FMD and PPN2V benchmark datasets. Table 5 shows a comparison between our uncalibrated method and various competing methods, including self-supervised and supervised methods.

Between the fully unsupervised methods that do not require paired noisy images (our Uncalibrated method, N2V, Bootstrap GMM, and Bootstrap Histogram), our method outperforms the others on four out of six datasets. A com-

5. Conclusions and Future Work

Noise is an unavoidable artifact of imaging systems, and for some applications such as live cell microscopy, denoising is a critical processing step to support quantitative and qualitative analysis. In this work, we have introduced a powerful new scheme for self-supervised learning of denoising which is appropriate for processing of low-light images. In contrast to the state-of-the-art, our model handles Poisson-Gaussian noise which is the standard noise model for most imaging systems including digital microscopes. In addition, we eliminate the need for loss function regularization in our method, thus making self-supervised denoising more practically applicable. Our evaluation on real datasets show that our method outperforms competing methods in terms of the standard PSNR metric on many datasets tested.

Our work opens up new avenues in live-cell imaging such as extreme low-light imaging over long periods of time. Future work lies in extending our model to other noise models appropriate to other imaging modalities, and exploring whether our uncalibrated method could be combined with a non-parametric prior [10].

Acknowledgments This work was supported in part by NSF #1659788, NIH #1R15GM128166-01 and the UCCS Biofrontiers Center.

References

- Joshua Batson and Loic Royer. Noise2self: Blind denoising by self-supervision. In *The 36th International Conference on Machine Learning (ICML 2019)*, 2019.
- [2] Thierry Blu and Florian Luisier. The sure-let approach to image denoising. *IEEE Transactions on Image Processing*, 16(11):2778–2786, 2007.
- [3] Antoni Buades, Bartomeu Coll, and Jean-Michel Morel. Non-local means denoising. *Image Processing On Line*, 1:208–212, 2011.
- [4] Antonin Chambolle. An algorithm for total variation minimization and applications. *Journal of Mathematical imaging and vision*, 20(1-2):89–97, 2004.
- [5] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image denoising with block-matching and 3d filtering. In *Image Processing: Algorithms and Systems, Neural Networks, and Machine Learning*, volume 6064, page 606414. International Society for Optics and Photonics, 2006.
- [6] Alessandro Foi, Mejdi Trimeche, Vladimir Katkovnik, and Karen Egiazarian. Practical poissonian-gaussian noise modeling and fitting for single-image raw-data. *IEEE Transactions on Image Processing*, 17(10):1737–1754, 2008.
- [7] Shuhang Gu, Lei Zhang, Wangmeng Zuo, and Xiangchu Feng. Weighted nuclear norm minimization with application to image denoising. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2862–2869, 2014.
- [8] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2014.
- [9] 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.
- [10] Alexander Krull, Tomáš Vičar, Mangal Prakash, Manan Lalit, and Florian Jug. Probabilistic noise2void: Unsupervised content-aware denoising. *Frontiers in Computer Science*, 2:5, 2020.
- [11] Samuli Laine, Tero Karras, Jaakko Lehtinen, and Timo Aila. High-quality self-supervised deep image denoising. In Advances in Neural Information Processing Systems, pages 6968–6978, 2019.
- [12] Yoann Le Montagner, Elsa D Angelini, and Jean-Christophe Olivo-Marin. An unbiased risk estimator for image denoising in the presence of mixed poisson–gaussian noise. *IEEE Transactions on Image processing*, 23(3):1255–1268, 2014.
- [13] Jaakko 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*, 2018.
- [14] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

- [15] Florian Luisier, Thierry Blu, and Michael Unser. Image denoising in mixed poisson–gaussian noise. *IEEE Transactions* on image processing, 20(3):696–708, 2010.
- [16] Markku Makitalo and Alessandro Foi. Optimal inversion of the generalized anscombe transformation for poissongaussian noise. *IEEE transactions on image processing*, 22(1):91–103, 2012.
- [17] John A Nelder and Roger Mead. A simplex method for function minimization. *The computer journal*, 7(4):308–313, 1965.
- [18] Mangal Prakash, Manan Lalit, Pavel Tomancak, Alexander Krull, and Florian Jug. Fully unsupervised probabilistic noise2void. In *International Symposium on Biomedical Imaging (ISBI 2020)*, 2020.
- [19] Sathish Ramani, Thierry Blu, and Michael Unser. Montecarlo sure: A black-box optimization of regularization parameters for general denoising algorithms. *IEEE Transactions on image processing*, 17(9):1540–1554, 2008.
- [20] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. Unet: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.
- [21] Shakarim Soltanayev and Se Young Chun. Training deep learning based denoisers without ground truth data. In Advances in Neural Information Processing Systems, pages 3257–3267, 2018.
- [22] Shakarim Soltanayev and Se Young Chun. Training deep learning based denoisers without ground truth data. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 3257–3267. Curran Associates, Inc., 2018.
- [23] Charles M Stein. Estimation of the mean of a multivariate normal distribution. *The annals of Statistics*, pages 1135– 1151, 1981.
- [24] 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.
- [25] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155, 2017.
- [26] 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 *Proceedings of the IEEE Conference* on Computer Vision and Pattern Recognition, pages 11710– 11718, 2019.