Supplemental Material: Self-Supervised Poisson-Gaussian Denoising

Wesley Khademi[1], Sonia Rao[2], Clare Minnerath[3], Guy Hagen[4], and Jonathan Ventura[1]

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

[1]{wkhademi, jventu09}@calpoly.edu

1. Derivation of Poisson-Gaussian Loss Function

In self-supervised denoising, we only have access to noisy pixels y_i and not the corresponding clean pixels x_i. Similar to a generative model, we use the negative log-likelihood of the training data as our loss function:

$$\mathcal{L}_i = -\log p(y_i) \tag{1}$$

However, for denoising we are interested in learning a model for $p(x_i)$, not $p(y_i)$. We relate $p(x_i)$ to $p(y_i)$ by marginalizing out x_i from the joint distribution:

$$p(y_i) = \int_{-\infty}^{\infty} p(y_i, x_i) dx_i \tag{2}$$
$$= \int_{-\infty}^{\infty} p(y_i|x_i)p(x_i) dx_i \tag{3}$$

In other words, we integrate $p(y_i, x_i) = p(y_i|x_i)p(x_i)$ over all possible values of the clean pixel x_i.

Here, $p(y_i|x_i)$ is simply our chosen noise model. $p(x_i)$ constitutes our prior belief about the value of x_i before we have seen an observation of y_i. We do not know what form the prior should take; it is essentially up to us to choose. Usually we use the conjugate prior of the noise model because this makes the integral tractable.

Our loss function term for pixel i will then be

$$\mathcal{L}_i = -\log \int_{-\infty}^{\infty} p(y_i|x_i)p(x_i) dx_i \tag{4}$$

1.1. Gaussian noise

For zero-centered Gaussian noise, $p(y_i|x_i)$ is the normal distribution centered at x_i with variance equal to σ_n^2. We choose $p(x_i)$ to be the normal distribution as well. Here we have the network output the parameters of the Gaussian, mean μ_i and std. dev. σ_i.

The marginalized pdf is derived as follows:

$$p(y_i) = \int_{-\infty}^{\infty} p(y_i|x_i)p(x_i) dx_i \tag{5}$$
$$= \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi\sigma_n^2}} \exp\left(-\frac{(y_i - x_i)^2}{2\sigma_n^2}\right) \cdot \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left(-\frac{(x_i - \mu_i)^2}{2\sigma_i^2}\right) dx_i \tag{6}$$
$$= \frac{1}{\sqrt{2\pi(\sigma_n^2 + \sigma_i^2)}} \exp\left(-\frac{(y_i - \mu_i)^2}{2(\sigma_n^2 + \sigma_i^2)}\right) \tag{7}$$

which we recognize as a Gaussian with mean μ_i and variance $\sigma_i^2 + \sigma_n^2$.

The loss function is then

$$\mathcal{L}_i = -\log p(y_i) \tag{8}$$
$$= -\log\left(\frac{1}{\sqrt{2\pi(\sigma_n^2 + \sigma_i^2)}} \exp\left(-\frac{(y_i - \mu_i)^2}{2(\sigma_n^2 + \sigma_i^2)}\right)\right) \tag{9}$$
$$= \frac{1}{2}\frac{(y_i - \mu_i)^2}{(\sigma_n^2 + \sigma_i^2)} + \frac{1}{2}\log 2\pi + \frac{1}{2}\log(\sigma_n^2 + \sigma_i^2). \tag{10}$$

Dropping the constant terms we have

$$\mathcal{L}_i = \frac{(y_i - \mu_i)^2}{(\sigma_n^2 + \sigma_i^2)} + \log(\sigma_n^2 + \sigma_i^2). \tag{11}$$

1.2. Poisson noise

For high enough values of X_i, the Poisson distribution $\mathcal{P}(\lambda)$ can be approximated by a Gaussian $\mathcal{N}(\lambda, \lambda)$ with mean and variance equal to λ. Using this idea, Laine et al. [2] adapt the above formulation for Gaussian noise to the Poisson noise case. However, they introduce an approximation in order to evaluate the integral.

Let a be the scaling factor s.t. $y/a \sim \mathcal{P}(x/a)$ where x and y are in the range $[0\ 1]$. The noise model using a normal approximation is $y = a(x/a + N(0, x/a)) = x + N(0, ax)$. The proper joint distribution for this model is thus

$$p(y_i)p(x_i) = \frac{1}{\sqrt{2\pi(ax_i)}} \exp\left(-\frac{(y_i - x_i)^2}{2(ax_i)}\right) \cdot \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left(-\frac{(x_i - \mu_i)^2}{2\sigma_i^2}\right) \tag{12}$$

However, Laine et al. replace the variance of the noise distribution with $a\mu_i$. This makes the integral tractable. They argue that this approximation is okay if σ_i^2 is small.

$$P(y_i) \approx \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi(a\mu_i)}} \exp\left(-\frac{(y_i - x_i)^2}{2(a\mu_i)}\right) \cdot \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left(-\frac{(x_i - \mu_i)^2}{2\sigma_i^2}\right) dx_i \tag{13}$$

$$= \frac{1}{\sqrt{2\pi(a\mu_i + \sigma_i^2)}} \cdot \exp\left(-\frac{(y_i - \mu_i)^2}{2(a\mu_i + \sigma_i^2)}\right) \tag{14}$$

which we recognize as a Gaussian with mean μ_i and variance $\sigma_i^2 + a\mu_i$.

Following the derivation above, our loss function is

$$\mathcal{L}_i = \frac{(y_i - \mu_i)^2}{(a\mu_i + \sigma_i^2)} + \log(a\mu_i + \sigma_i^2). \tag{15}$$

1.3. Poisson-Gaussian noise

Noise in microscope images is generally modeled as a Poisson-Gaussian process. The number of photons entering the sensor during the exposure time is assumed to follow a Poisson distribution, and other noise components such as the readout noise and thermal noise are captured by an additive Gaussian term.

We can easily extend the Poisson loss function above to Poisson-Gaussian by adding a noise variance b to the model. Following the derivation above, our loss function is

$$\mathcal{L}_i = \frac{(y_i - \mu_i)^2}{(a\mu_i + b + \sigma_i^2)} + \log(a\mu_i + b + \sigma_i^2). \tag{16}$$

2. Posterior mean estimate

The blind-spot network ignores the actual measured value for y_i when it makes a prediction for x_i. However, y_i contains extra information which can be used to improve our estimate of x_i.

Laine et al. [2] suggest to use the expected value of the posterior:

$$\hat{x_i} = \mathbb{E}[x_i|y_i] = \int_{-\infty}^{\infty} p(x_i|y_i)x_i dx_i = \frac{1}{Z}\int_{-\infty}^{\infty} p(y_i|x_i)p(x_i)x_i dx_i$$

where we have applied Baye's rule to relate $p(x_i|y_i)$ and $p(y_i|x_i)p(x_i)$ up to a normalizing constant Z, where

$$Z = \int_{-\infty}^{\infty} p(y_i|x_i)dx_i. \tag{17}$$

For a Gaussian with prior mean μ_i and variance σ_i^2, and noise variance σ_n^2, we have the following result:

$$\hat{x_i} = \frac{y_i\sigma_i^2 + \sigma_n^2\mu_i}{\sigma_i^2 + \sigma_n^2} \tag{18}$$

This same formula can be used for the Poisson or Poisson-Gaussian noise models (replacing σ_n^2 with $a\mu_i$ or $a\mu_i + b$, respectively).

3. Results

3.1. Gaussian noise

While our method is meant for denoising images with Poisson-Gaussian noise, we also test our methods ability to denoise pure Gaussian noise. We evaluate two variants of our noise estimation method by comparing the estimated a,b parameters to the ground truth parameters for every noise level, $\sigma \in \{10, 20, 30, 40, 50\}$, of our synthetic Gaussian noise Confocal MICE dataset. When we can make the assumption that Gaussian noise is the type of noise that exists in an image, we show that an improvement can be made in denoising quality by making a small change to our method to fit pure Gaussian noise parameters.

Table 1 provides results obtained on our synthetic dataset in which *Estimated* represents a,b parameters obtained using our Poisson-Gaussian noise fitting, while *G-Estimated* represents our estimated a,b parameters using our Gaussian noise fitting method which can be accomplished by simply fixing $a = 0$ for our Poisson-Gaussian noise fitting technique. While our Poisson-Gaussian noise fitting technique typically provides a better estimate of the b parameter than the Gaussian noise fitting method, our Poisson-Gaussian noise fitting obtains an incorrect estimate of a unlike our Gaussian noise fitting. This worse estimate of a contributes to an overall worse denoising quality than the quality obtained by estimating the noise parameters with Gaussian noise fitting which is shown in Table 2. This is because the Gaussian noise fitting parameters result in a more accurate estimate of the computed noise variance due to the a parameter being known.

3.2. Poisson noise

We perform the same denoising experiments on images with varying levels of Poisson noise. Once again, we show that we can improve the denoising quality by modifying our Poisson-Gaussian noise fitting technique to fit Poisson noise when we can make the assumption that pure Poisson noise is present in an image. This can be done by simply fixing $b = 0$ for our Poisson-Gaussian noise fitting technique.

Table 3 shows a comparison between ground truth and estimated a,b noise parameters for every noise level, $\lambda \in \{10, 20, 30, 40, 50\}$, of our synthetic Poisson noise Confocal MICE dataset. *Estimated* and *P-Estimated* represent our estimated a,b parameters using our Poisson-Gaussian and Poisson noise fitting techniques, respectively. While our Poisson-Gaussian noise fitting provides decent estimates of the a,b parameters, using the pure Poisson noise fitting can provide almost exact estimates of both parameters. Table 3 shows that our Poisson noise fitting technique results in a PSNR close to that obtained using the ground truth parameters. While the Poisson-Gaussian noise fitting results in a worse denoising quality than the Poisson noise fitting, an improvement in denoising quality is still obtained over the pseudo-clean image.

3.3. Poisson-Gaussian noise

To further evaluate our bootstrapping method, we compare our estimated a,b parameters to the ground truth ones for every noise level, $(\lambda, \sigma) \in \{10, 20, 30, 40, 50\} \times \{10, 20, 30, 40, 50\}$, of our synthetic Poisson-Gaussian noise Confocal MICE dataset. Table 5 shows the results obtained for all the Poisson-Gaussian noise levels. The ground truth a,b parameters correspond to the (λ, σ) noise level synthetically added to the clean images while the estimated a and b are the parameters learned from fitting a Poisson-Gaussian noise model using our bootstrapping technique.

We further evaluate our approach on our synthetic dataset by comparing the PSNR obtained by our bootstrapping method for all the noise level combinations. Table 6 compares the PSNR achieved by the pseudo-clean image which is the output of the blindspot neural network, the image obtained by our bootstrapping method, as well as the image obtained when the ground truth a,b parameters are known rather than learned.

3.4. Noise Parameter Estimation

To further study how effective our bootstrapping technique is we evaluate how the PSNR is affected by various levels of error in our estimates of the a,b parameters. Since the error in our estimation of the a parameter is different than the error in our estimation of the b parameter we select different percent errors for each parameter. The initial percent errors are chosen by computing the average percent error of our estimation of the noise parameters over all noise levels, $(\lambda, \sigma) \in \{0, 10, 20, 30, 40, 50\} \times \{0, 10, 20, 30, 40, 50\}$, of our synthetic Confocal MICE dataset. We then select the other two percent errors to be one standard deviation below and above the average percent errors.

Table 7 provides results of the PSNR obtained for the different percent errors in the a,b parameters on a Confocal MICE dataset with synthetically added Poisson-Gaussian noise ($\lambda/\sigma = 30$). As mentioned before, our estimation of the a parameter is much worse than the b parameter most likely due to the Confocal MICE dataset containing many dark pixels. Even when using the worst percent errors for our a,b parameters though we still obtain an increase in PSNR from the pseudo-clean image which shows that even a poor estimation using our method can still help denoise an image. We also note that the PSNR actually increases with an increase in percent error in the estimate of the b parameter, which can best be explained by our method always overestimating the a parameter and underestimating the b parameter. In practice, a worse estimate of the b parameter helps compensate for the overestimation of the a parameter, providing a better approximation of the noise variance.

3.5. BSD68

We also provide results for our method on the BSD68 dataset. We follow the experiment in [1] and train our model on 400 gray scale images and test on the gray scale version of the BSD68 dataset. Both the training and testing dataset have synthetically added zero mean Gaussian noise with standard deviation $\sigma = 25$.

We evaluate our method on the BSD68 dataset by comparing the PSNR obtained using BM3D, a supervised training method, Noise2Noise, Noise2Void, and three different outputs of our model: the blindspot neural network, the output obtained estimating the noise parameters, and the output obtained using the true noise parameters. While our method is not expected to perform better than the supervised training method or Noise2Noise, it should perform better than Noise2Void. Table 8 compares the different outputs of our model to the various state-of-the-art methods. We observe that the PSNR of the output of the blindspot neural network, or pseudo-clean image, is significantly worse than the PSNR of the output of the Noise2Void method, but when our Poisson-Gaussian noise fitting technique is used we outperform Noise2Void. Similar to Noise2Void, our method performs worse than BM3D which is most likely due to using too small of a training set to obtain the best possible pseudo-clean image from the blindspot neural network.

We further evaluate our method by comparing our estimated a,b parameters to the ground truth noise parameters. Table 9 shows that our method obtains only a decent estimate of the a,b parameters. As mentioned, this is probably due to the pseudo-clean image not being close enough to the ground truth clean image to provide a good estimate of the ground truth noise parameters from our Poisson-Gaussian noise fitting technique.

References

[1] 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.

[2] 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.

σ	Ground Truth	Estimated a	G-Estimated	Ground Truth	Estimated b	G-Estimated
10	0	0.00246	0	0.00153	0.00148	0.00165
20	0	0.00434	0	0.00615	0.00605	0.00636
30	0	0.00661	0	0.0138	0.0136	0.0141
40	0	0.00879	0	0.0246	0.0244	0.0250
50	0	0.00113	0	0.0384	0.0381	0.0389

Table 1: Estimated a,b parameters for varying levels of Gaussian noise using our Poisson-Gaussian and pure Gaussian noise fitting techniques.

σ	Pseudo-Clean PSNR	Uncalibrated PSNR	Uncalibrated w/ $a = 0$ PSNR	Ground Truth PSNR
10	39.25	39.50	39.58	39.61
20	36.73	36.84	36.96	37.02
30	35.04	35.10	35.26	35.30
40	33.72	33.76	33.92	33.95
50	32.74	32.79	32.91	32.94

Table 2: PSNR comparison between using our uncalibrated method with Poisson-Gaussian noise fitting and Gaussian noise fitting for estimating the a,b parameters of pure Gaussian noise.

λ	Ground Truth	Estimated a	P-Estimated	Ground Truth	Estimated b	P-Estimated
10	0.100	0.108	0.100	0	-0.000390	0
20	0.0500	0.0561	0.0500	0	-0.000246	0
30	0.0333	0.0380	0.0348	0	-0.000148	0
40	0.0250	0.0290	0.0258	0	-0.000152	0
50	0.0200	0.0239	0.0208	0	-0.000142	0

Table 3: Estimated a,b parameters for varying levels of Poisson noise using our Poisson-Gaussian and pure Poisson noise fitting techniques.

λ	Pseudo-Clean PSNR	Uncalibrated PSNR	Uncalibrated w/ $b = 0$ PSNR	Ground Truth PSNR
10	34.15	34.29	34.38	34.38
20	35.66	35.76	35.90	35.93
30	36.61	36.71	36.82	36.89
40	37.09	37.22	37.36	37.40
50	37.58	37.71	37.83	37.91

Table 4: PSNR comparison between using our uncalibrated method with Poisson-Gaussian noise fitting and Poisson noise fitting for estimating the a,b parameters of pure Poisson noise.

λ	σ	Ground Truth	Estimated a	Absolute Error	Ground Truth	Estimated b	Absolute Error
10	10	0.100	0.105	0.00537	0.00153	0.00116	0.000376
	20	0.100	0.103	0.00315	0.00615	0.00645	0.000296
	30	0.100	0.114	0.0135	0.0138	0.0133	0.000539
	40	0.100	0.121	0.0209	0.0246	0.0234	0.00122
	50	0.100	0.117	0.0177	0.0384	0.0379	0.000578
20	10	0.0500	0.0565	0.00651	0.00153	0.00128	0.000254
	20	0.0500	0.0593	0.00925	0.00615	0.00579	0.000356
	30	0.0500	0.0608	0.0108	0.0138	0.0134	0.000388
	40	0.0500	0.0623	0.0123	0.0246	0.0241	0.000405
	50	0.0500	0.0658	0.0158	0.0384	0.0378	0.000614
30	10	0.0333	0.0392	0.00587	0.00153	0.00134	0.000193
	20	0.0333	0.0408	0.00751	0.00615	0.00590	0.000247
	30	0.0333	0.0430	0.00975	0.0138	0.0134	0.000387
	40	0.0333	0.0457	0.0123	0.0246	0.0241	0.000504
	50	0.0333	0.0477	0.0144	0.0384	0.0378	0.000623
40	10	0.0250	0.0305	0.00551	0.00153	0.00131	0.000228
	20	0.0250	0.0325	0.00751	0.00615	0.00586	0.000291
	30	0.0250	0.0340	0.00897	0.0138	0.0135	0.000312
	40	0.0250	0.0367	0.0117	0.0246	0.0242	0.000405
	50	0.0250	0.0401	0.0151	0.0384	0.0377	0.000352
50	10	0.0200	0.0250	0.00508	0.00153	0.00135	0.000184
	20	0.0200	0.0267	0.00675	0.00615	0.00591	0.000288
	30	0.0200	0.0281	0.00807	0.0138	0.0137	0.000180
	40	0.0200	0.0297	0.00969	0.0246	0.0244	0.000198
	50	0.0200	0.0329	0.0129	0.0384	0.0381	0.000353

Table 5: Quantitative comparison of fitting a Poisson-Gaussian noise model on all the different Poisson-Gaussian noise levels of our synthetic Confocal MICE dataset.

λ	σ	Pseudo-Clean PSNR	Uncalibrated PSNR	Ground Truth PSNR
10	10	33.96	34.17	34.20
	20	33.60	33.75	33.82
	30	33.09	33.14	33.27
	40	32.38	32.42	32.55
	50	31.98	32.03	32.14
20	10	35.45	35.56	35.73
	20	34.68	34.75	34.93
	30	33.97	34.03	34.19
	40	33.17	33.22	33.37
	50	32.22	32.27	32.39
30	10	36.18	36.26	36.43
	20	35.30	35.37	35.56
	30	34.10	34.15	34.33
	40	33.16	33.22	33.37
	50	32.26	32.34	32.45
40	10	36.61	36.71	36.91
	20	35.45	35.53	34.71
	30	34.47	34.52	34.33
	40	33.30	33.37	33.52
	50	31.73	31.78	31.92
50	10	36.95	37.05	37.25
	20	35.83	35.90	36.10
	30	34.47	34.54	34.73
	40	33.52	33.57	33.72
	50	32.54	32.60	32.74

Table 6: Comparison of PSNR using our bootstrapping method on all the different Poisson-Gaussian noise levels of our synthetic Confocal MICE dataset.

% error a	% error b	Pseudo-Clean PSNR	Uncalibrated PSNR	Ground Truth PSNR
6	0	34.10	34.28	34.33
	4	34.10	34.31	34.33
	8	34.10	34.30	34.33
22	0	34.10	34.17	34.33
	4	34.10	34.19	34.33
	8	34.10	34.24	34.33
38	0	34.10	34.13	34.33
	4	34.10	34.14	34.33
	8	34.10	34.15	34.33

Table 7: Quantitative results on how our uncalibrated method's PSNR is affected by varying percent errors in our estimated a,b parameters. The results shown were done using Poisson-Gaussian noise with $\lambda = 30$ and $\sigma = 30$.

	BM3D	Traditional	N2N	N2V	Pseudo-Clean (Ours)	Uncalibrated (Ours)	Uncalibrated (Ours) w/ known b
BSD68	28.59	29.06	28.86	27.71	26.95	27.80	28.15

Table 8: PSNR results on the BSD68 dataset.

σ	Ground Truth a	Estimated a	Ground Truth b	Estimated b
25	0	0.00533	0.00961	0.00861

Table 9: Estimated noise parameters using our uncalibrated method on the BSD68 dataset.