denoiSplit: a method for joint microscopy image splitting and unsupervised denoising

Ashesh Ashesh and Florian Jug

Fondazione Human Technopole, Viale Rita Levi-Montalcini 1, 20157 Milan, Italy
ashesh276@gmail.com, florian.jug@fht.org

Abstract. In this work, we present denoi𝕊plit, a method to tackle a new analysis task, *i.e.* the challenge of joint semantic image splitting and unsupervised denoising. This dual approach has important applications in fluorescence microscopy, where semantic image splitting has important applications but noise does generally hinder the downstream analysis of image content. Image splitting involves dissecting an image into its distinguishable semantic structures. We show that the current state-of-the-art method for this task struggles in the presence of image noise, inadvertently also distributing the noise across the predicted outputs. The method we present here can deal with image noise by integrating an unsupervised denoising subtask. This integration results in improved semantic image unmixing, even in the presence of notable and realistic levels of imaging noise. A key innovation in denoi𝕊plit is the use of specifically formulated noise models and the suitable adjustment of KL-divergence loss for the high-dimensional hierarchical latent space we are training. We showcase the performance of denoi𝕊plit across multiple tasks on real-world microscopy images. Additionally, we perform qualitative and quantitative evaluations and compare the results to existing benchmarks, demonstrating the effectiveness of using denoi𝕊plit: a single Variational Splitting Encoder-Decoder (VSE) Network using two suitable noise models to jointly perform semantic splitting and denoising.

1 Introduction

Fluorescence microscopy remains a cornerstone in the exploration of cellular and sub-cellular structures, enabling scientists to visualize biological processes at a remarkable level of detail [9, 22]. However, the ability to distinguish and analyze multiple structures within a single sample requires a multiplexed imaging protocol that requires extra time and effort [22]. To address these downsides and enable for more efficient and new types of investigation, a powerful method for semantic image splitting was recently introduced [1].

Building on this previous work [1], we address a key challenge that persisted: noise in microscopy images and its adverse effect on the quality of image-splitting predictions. Recognizing the need for a method capable of handling noisy input images while maintaining the integrity of the semantic splitting task, we introduce a technique that not only builds on the strengths of μSplit [1] but also

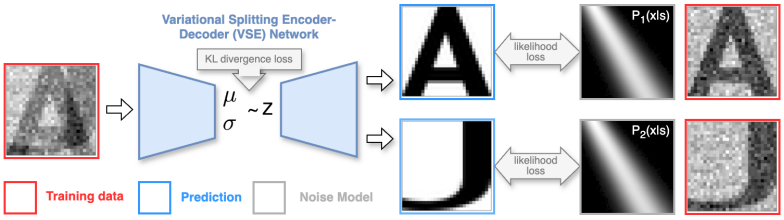


Fig. 1: Teaser Figure. In this work we use a variational encoder-decoder network to jointly solve an usupervised denoising and image splitting task and show that our approach outperforms existing baselines.

incorporates unsupervised denoising capabilities, for example as in [15, 19–21]. Figure 1 outlines the overall approach we are proposing.

Together, these ingredients lead to a new method denoi𝕊plit. It refines the process of image decomposition, ensuring that even under high levels of pixel noises present in the entire body of available training data, the semantic integrity of the semantically split image components (the predictions) is well preserved. Additionally, denoi𝕊plit can assess data uncertainty by sampling from the learned posterior of possible splitting solutions, followed by evaluating the inter-sample variability. In Section 4, we show how to use this possibility to predict the expected error denoi𝕊plit makes on a given input.

In summary, we believe that this work will open new avenues for the efficient and detailed analysis of complex biological samples, for example, in the context of fluorescent microscopy.

2 Related Work

2.1 Image Denoising

Image denoising is a task that has a long and exciting history. Classical methods, such as Non-Local Means [5] or BM3D [6], were frequently and very successfully used before neural network based approaches have been introduced towards the end of the last decade [14, 25–27].

The advent of deep learning saw people exploit different aspects of noise and the way networks learn to enable denoising. Noise, while usually undesired, is simultaneously much harder to predict, as was elegantly demonstrated in [24], leading to a zero-shot denoiser. In the case specific to pixel noises, *i.e.* all forms of noise that are independent per image pixel (given the signal at that pixel) [22], the impossibility to predict the noise was exploited in various ways, leading to important contributions such as Noise2Void [14], Noise2Self [3], or Self2Self [13]. Another well known approach close to this family of approaches is Noise2Noise [16], capable of denoising even more complex noises that can correlate beyond the confines of single pixels.

To further improve denoising performance, Probabilistic Noise2Void [15, 21] introduced, and DivNoising [20] and Hierarchical DivNoising (HDN) [19] reused

the idea of suitably measured or trained pixel noise models. Such noise models are, in essence, a collection of probability distributions mapping from a true pixel intensity to observed noisy pixel measurements (and vice versa).

2.2 Image Decomposition

Image decomposition is the inverse problem of splitting a given input image that is the superposition (*i.e.* the pixel-wise sum) of two constituent image channels. While the sum of two values is not uniquely invertible, if for each summand a prior on its value exists, even a unique solution can exist. In a similar vein, having learned structural priors of the appearance of the two constituent image channels, an input image (a grid of observed pixels that are each a sum of two values) can be split into two pixel grids such that each one satisfies the respective structural prior. In computer vision, reflection removal, dehazing, deraining *etc.* are some of the applications [2, 4, 7, 8] for which image splitting can be used.

More recently, image splitting in fluorescence microscopy was receiving heightened attention, probably because of the direct applicability and potential utility that a well-working approach can bring to this microscopy modality, which finds wide-spread use in biological investigations. In particular, a method called μSplit [1] demonstrated impressive image splitting performance on several datasets, suggesting that it is ready to be used in biological research projects.

However, μSplit requires relatively noise-free data for training and prediction, which limits its potential utility (see also Section 5 or Figure 2).

2.3 Uncertainty Calibration

The ability to correctly assess the quality of predictions is naturally useful. Ideally, a predictive system capable of co-predicting a confidence value has the property that the predicted confidence scales with the average error of the prediction. If the relationship between error and confidence is close to the identity, we call the uncertainty predictions of this system *calibrated*.

Early works tried to use the deviation of the prediction as a proxy of the network's confidence in the prediction [26]. Other works tried to calibrate the predicted standard deviation with the expected error (*i.e.* the RMSE) [17]. Earlier calibration works were mainly concerned with classification tasks [18]. However, in [17], these approaches were reformulated in the context of regression.

In [17], the authors propose a way to evaluate calibration. They train a separate branch to predict a standard deviation per pixel that expresses its prediction uncertainty. For evaluating the calibration quality, the authors clustered examples on the basis of the predicted standard deviation values. Within each cluster, the predicted uncertainty is then compared with the empirical uncertainty (RMSE loss). To further improve the calibration, the authors propose a simple, yet effective scaling methodology wherein they learn a scalar parameter on re-calibration data, *i.e.* a subset of data not included in the training data. (In our experiments, we use the validation data for this purpose.) This

scalar gets multiplied to the predicted uncertainty values, which then reduces the calibration error.

3 Problem Formulation

Lets denote a noise free dataset containing n pairs of images as $D = (C_1, C_2)$, with each C_i containing n images $C_i = (c_{i,j}|1 \leq j \leq n)$. Lets define a corresponding set of images $X = (x_j|1 \leq j \leq n)$, such that all $x_j = c_{1,j} + c_{2,j}$ are the pixel-wise sum of the two corresponding channel images.

Although D is typically not available (or even observable), in practice we can only observe noisy data, denoted here by $D^N = (C_1^N, C_2^N)$. Analogously to before, we define $X^N = (x_j^N|1 \leq j \leq n)$, such that $x_j^N = c_{1,j}^N + c_{2,j}^N$ are the pixel-wise sum of the noisy channel observations.

Given one $x_j^N \in X^N$ of D^N, the task at hand is to predict the noise free and unmixed tuple $(c_{1,j}, c_{2,j})$. We shall denote the predictions made by a trained denoiSplit network by $(\hat{c}_{1,j}, \hat{c}_{2,j})$.

Whenever above notions are used in a context that makes the j in the subscript redundant, we allow ourselves to omit them for brevity and readability.

For evaluation purposes, we will in later sections use high-quality microscopy datasets that contain minimal levels of noise as surrogates for D, X, C_1, and C_2, but we never use them during training, and only their noisy counterparts are used.

4 Our Approach

In the following sections we describe the main ingredients of denoiSplit, namely the hierarchical network structure we use (Section 4.1), the changed loss term for variational training of the splitting task (Section 4.2), the noise models we employ to enable the joint unsupervised denoising (Section 4.3), and an uncertainty calibration methodology allowing us to estimate the prediction error introduced by aleatoric uncertainty in a given input image (Section 4.4).

4.1 Network Architecture and Training Objective

In this work, we employ an altered Hierarchical VAE (HVAE) network architecture. HVAEs were originally described in [23] and later adapted for image denoising in [19] and for image splitting in [1]. In general terms, HVAEs learn a hierarchical latent space, with the lowest hierarchy level encoding detailed pixel-level structure, while higher hierarchy levels capture increasingly larger scale structures in the training data.

For denoiSplit, we modify the HVAE architecture so that it no longer remains an autoencoder. Instead, our outputs are the two unmixed channel images $(\hat{c}_1, \hat{c}_2)$, motivating us to call the resulting architecture a Variational Splitting Encoder-Decoder (VSE) Network (see Fig. 1).

Our objective is to maximize the likelihood over the noisy two channel dataset we train on, *i.e.*, finding decoder parameters $\boldsymbol{\theta}$ such that

$$\boldsymbol{\theta} = \arg\max_{\boldsymbol{\theta}} \sum_{1 \leq j \leq n} \log P(c_{1,j}^N, c_{2,j}^N; \boldsymbol{\theta}). \tag{1}$$

Using the modified evidence lower bound (ELBO), as proposed in [1] and assuming conditional independence of the two predictions $(\hat{c}_1, \hat{c}_2)$ given the latent space embedding, we maximize

$$E_{q(z|x;\boldsymbol{\phi})}[\log P(c_1^N|z;\boldsymbol{\theta}) + \log P(c_2^N|z;\boldsymbol{\theta})] - KL(q(z|x;\boldsymbol{\phi}), P(z)), \tag{2}$$

where $q(z|x;\boldsymbol{\phi})$ is the distribution parameterized by the output of the encoder network $\text{Enc}_{\boldsymbol{\phi}}(x)$, $P(c_i^N|z;\boldsymbol{\phi})$ is the distribution parameterized by the output of the decoder network $\text{Dec}_{\boldsymbol{\theta}}(z)$ and $KL()$ denotes the Kullback-Leibler divergence loss. As in [1], $P(z)$ factorizes over the different hierarchy levels in the network. Details about training, hyperparameters, μSplit, and its relationship with HDN and denoi$\mathbb{S}$plit can be found in Supp. *Sec. 2*.

Similar to the way noise models had been employed in the context of denoising [19], we model the two log likelihood terms $\log P(c_1^N|z;\boldsymbol{\theta})$ and $\log P(c_2^N|z;\boldsymbol{\theta})$ using noise models which we describe in detail in Section 4.3 and our open code repository[1].

4.2 Hierarchical KL Loss Weighing for Variational Training

In μSplit, the authors showed SOTA performance on a multitude of splitting tasks. However, used datasets were close to noise-free, making the task at hand simpler then the one we outlined in Section 3. When μSplit is trained on noisy datasets, the resulting channel predictions are themselves noisy. After analyzing this matter, we concluded that a modified KL loss can help reduce the amount of noise reconstructed by the decoder.

In more technical terms, let Z be a hierarchical latent space and $Z[i]$ denote the latent space embedding at i-th hierarchy level, having shape (c, h_i, w_i), with c being channel dimension, and h_i, w_i the height and width of the latent space embedding. Now let KL_i denote the KL-divergence loss tensor computed on $Z[i]$, which has the same shape as $Z[i]$ itself.

In μSplit, the corresponding scalar loss term kl_i is defined as $\text{kl}_i = \alpha \cdot \sum_{j,h,w} \frac{\text{KL}_i[j,h,w]}{h_i \cdot w_i}$, with α being a suitable constant. Observe that the denominator makes each kl_i be the average of all values in KL_i, making the respective values not scale with the size of $Z[i]$, even though lower hierarchy levels ($Z[i]$ for smaller i) have more entries. However, this also means that the KL loss for the individual pixels in these lower hierarchy levels is given less weight. Hence, smaller structures, such as noise itself, can more easily seep through such pixel-near hierarchy levels.

[1] https://github.com/juglab/denoiSplit

In this work, we diverge from this formulation and return to a more classical setup where we compute the scalar loss term for the i-th hierarchy level $Z[i]$ as

$$\mathrm{kl}_i = \alpha \cdot \sum_{j,h,w} \mathrm{KL}_i[j,h,w]. \tag{3}$$

The decisive difference is that this changed formulation gives more weight to the KL loss at lower hierarchy levels, leading to more strongly enforcing the Gaussian nature the KL loss enforces, and therefore hindering noise from being as easily represented during training. We refer to this architecture as *Altered μSplit* and show qualitative and quantitative results in Section 5 and Tables 1.

The next section extends on *Altered μSplit* by adding unsupervised denoising, adding the last ingredient to the denoi$\mathbb{S}$plit approach we present in this work.

4.3 Adding Suitable Pixel Noise Models

As briefly introduced in Section 2, pixel noise models are a collection of probability distributions mapping from a true pixel intensity to observed noisy pixel measurements (and vice-versa) [15]. They have previously been successfully used in the context of unsupervised denoising [19, 20] and we intend to employ them for this purpose also in the setup we are presenting here. We use the fact that, given a measured (noisy) pixel intensity, a pixel noise model returns a distribution over clean signal intensities and their respective probability of being the underlying true pixel value.

We incorporate this likelihood function into the loss of our overall setup, encouraging denoi$\mathbb{S}$plit to predict pixel intensities that maximize this likelihood and thereby values that are consistent with the noise properties of the given training data.

Since denoi$\mathbb{S}$plit, in contrast to existing denoising applications, predicts two images (the two unmixed channels), we employ two noise models and add two likelihood terms to our overall loss.

More formally, in VAEs [12] and HVAEs, the generative distribution over pixel intensities is modeled as a Gaussian distribution with its variance either clamped to 1 or also learned and predicted. We change our VSE Network to only predict the true pixel intensity and replace the Gaussian distribution mentioned above by the distributions defined in two noise models $P_1^{\mathrm{nm}}(c_1^N|c_1)$ and $P_2^{\mathrm{nm}}(c_2^N|c_2)$, one for each respective unmixed output channel. These noise models are pixel-wise independent, *i.e.*,

$$P_i{}^{\mathrm{nm}}(c_i^N|c_i) = \prod_k P_i^{\mathrm{nm}}(c_i^N[k]|c_i[k]),\ i \in \{1,2\}, \tag{4}$$

where $c_i^N[k]$ is the noisy pixel intensity for the k-th pixel and $c_i[k]$ the corresponding noise-free intensity value. This independence makes them particularly suitable for microscopy data where Poisson and Gaussian noise are the predominant pixel noises one desires to remove.

Since we now directly predict the noise-free pixel values, the output of the decoder can directly be interpreted as $\text{Dec}_{\boldsymbol{\theta}}(z) = (\hat{c}_1, \hat{c}_2)$ and the total loss for denoiSplit now becomes

$$E_{q(z|x;\boldsymbol{\phi})}[\log P^{\text{nm}}(c_1^N|\hat{c}_1) + \log P^{\text{nm}}(c_2^N|\hat{c}_2)] - KL(q(z|x;\boldsymbol{\phi}), P(z)). \quad (5)$$

In [20], two ways for the creation of noise models are described, and the decision to pick which method depends upon whether or not one has access to the microscope from which data was acquired. In Supp. *Sec. 5*, we describe the process of noise model generation and also compare performance between these two methodologies.

4.4 Computing Calibrated Data Uncertainties

The idea of calibration is for those network setups that produce both prediction and a measure of uncertainty for the prediction.

Networks that can co-assess the uncertainty of their predictions are called calibrated, when the predicted uncertainties are in line with the measured prediction error. To improve the calibration of a given system, one can find a suitable transformation from uncertainty predictions to measured errors (*e.g.*, the RMSE). After such a transformation is found, an ideal calibrated plot would be tightly fitting $y = x$, with y and x being the error and estimated uncertainty, respectively. See, for example, Figure 3. Since VSE networks, similar to VAEs, are variational inference systems, we can sample from their latent encoding and thereby sample from an approximate posterior distribution of possible solutions giving us the data uncertainty. In this section, our intention is to utilize this ability to predict a reliable uncertainty term for our results.

For this, we adapt the calibration methodology of [17]. In contrast to the approach described there, we propose to use the variability in posterior samples to estimate a pixel-wise standard deviation. More specifically, we sample $k = 50$ predictions for each input image and compute the pixel-wise standard deviations σ_1 and σ_2 for the two predicted image channels $\hat{c}_1$ and $\hat{c}_2$, respectively. This gives us uncertainty predictions.

Next, we calibrate these uncertainty predictions by scaling them appropriately with the help of two learnable scalars, α_1 and α_2. Following [17], we assume that pixel intensities come from a Gaussian distribution. The mean and standard deviation of this distribution are the pixel intensities of the MMSE prediction, *i.e.* the image obtained after averaging $k = 50$ predictions, and the scaled σ, respectively. We learn the scalars α_1 and α_2 by minimizing the negative log-likelihood over the recalibration dataset. It is important to note that the presented calibration procedure does not alter the original predictions but instead learns a mapping that best predicts the measured error.

To evaluate the quality of the resulting calibration, we sort the scaled standard deviations $\sigma_i \cdot s_i$ for each pixel in a predicted channel and build a histogram over $l = 30$ equally sized bins B_i^j. We then compute the root mean variance (RMV) and RMSE for each bin j and channel i as

$$\text{RMV}_i(j) = \sqrt{\frac{1}{|B_i^j|}\sum_{k\in B_i^j}(\sigma_i^k \cdot \alpha_i)^2} \qquad \text{RMSE}_i(j) = \sqrt{\frac{1}{|B_i^j|}\sum_{k\in B_i^j}(c_i[k] - \hat{c}_i[k])^2}$$

As in Section 3, $c_i[t]$ and $\hat{c}_i[t]$ denote the noise-free pixel intensity and the corresponding prediction for i-th channel. In Fig. 3, we plot the RMSE vs. RMV for multiple tasks, observing that the plots closely resemble the identity $y = x$. Following [17], we use the validation dataset for recalibration and show calibration plots on the test dataset.

5 Experiments and Results

5.1 Datasets

BioSR dataset We work primarily with BioSR dataset [11], a comprehensive dataset comprising fluorescence microscopy images of multiple cell structures. For our experiments, we have picked four structures, namely clathrin-coated pits (CCPs), microtubules (MTs), endoplasmic reticulum (ER), and F-actin.

Since the raw data quality is very high and only a small amount of image noise is present in the individual micrographs, we add Gaussian noise and Poisson noise of various levels to these raw data. The artificially noisy images are used to train denoiSplit, while the raw data is shown to convince the reader of the validity of our approach and to compute evaluation metrics (see Figs. 2 and 4 and Tab. 1).

Hagen et al. Actin-Mitochondria Dataset We picked the noisy Actin and Mitochondria channels from Hagen et al. [10], channels having real microscopy noise. For evaluation, we use the corresponding high-SNR (noise-free) channels provided in the dataset.

Synthetic Noise Levels We work with 4 levels of zero-mean Gaussian noise and two levels of Poisson noise. For Gaussian noise, we compute the standard deviation of the input data X^N for each of the tasks and scale the noise relative to one standard deviation. Specifically, the 4 scaling factors are $\{1, 1.5, 2, 4\}$. In cases where Poisson noise is added, and since it is already signal dependent, we use a constant factor of 1000 to hit a realistic-looking level of Poisson noise. We also consider the case where Poisson noise is not added, which we denote by the Poisson level of 0 in Tab. 1. To remove any remaining room for misinterpretations, we provide a pseudo-code for the synthetic noising procedure in the Supp. *Sec. 2*.

5.2 Baselines

We conducted all experiments with two baseline setups, μSplit and *HDN*$\oplus\mu$*Split*. In the original μSplit work [1], the authors introduce three architectures, each with a different trade-off between GPU efficiency, speed and performance. We

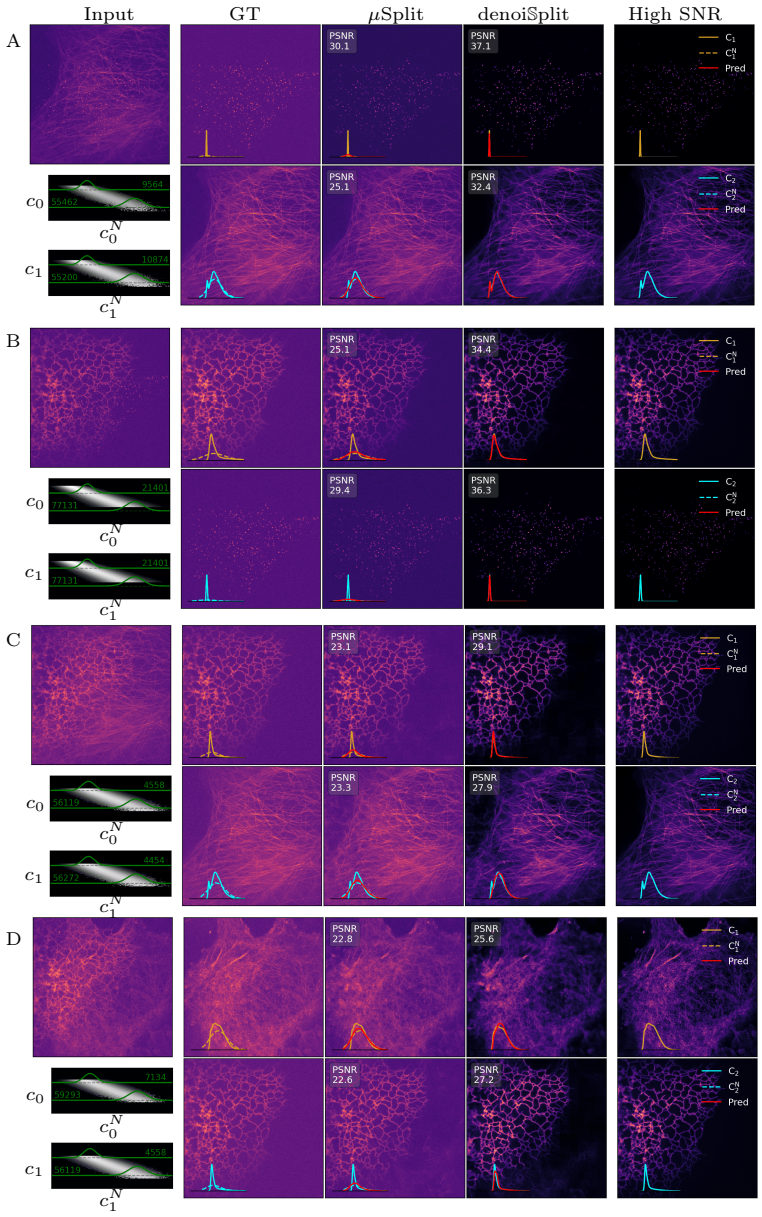


Fig. 2: Qualitative Results. We show examples of noisy inputs, individual noisy channel training data (GT), and predictions by one of the baselines (μSplit) and our own results obtained with denoiSplit for four tasks (A: MT vs. CCPs, B: ER vs. CCPs, C: MT vs. ER, and D: F-actin vs. ER). We show high SNR channel images (not used during training) and show PSNR values w.r.t. these images. Additionally, we plot histograms of various panels for comparison (see legend on the right). The bottom cell in the first column of each panel shows the used noise models (see main text for details). The superimposed plots (green) show the distribution of noisy observations (c_i^N) for two clean signal intensities.

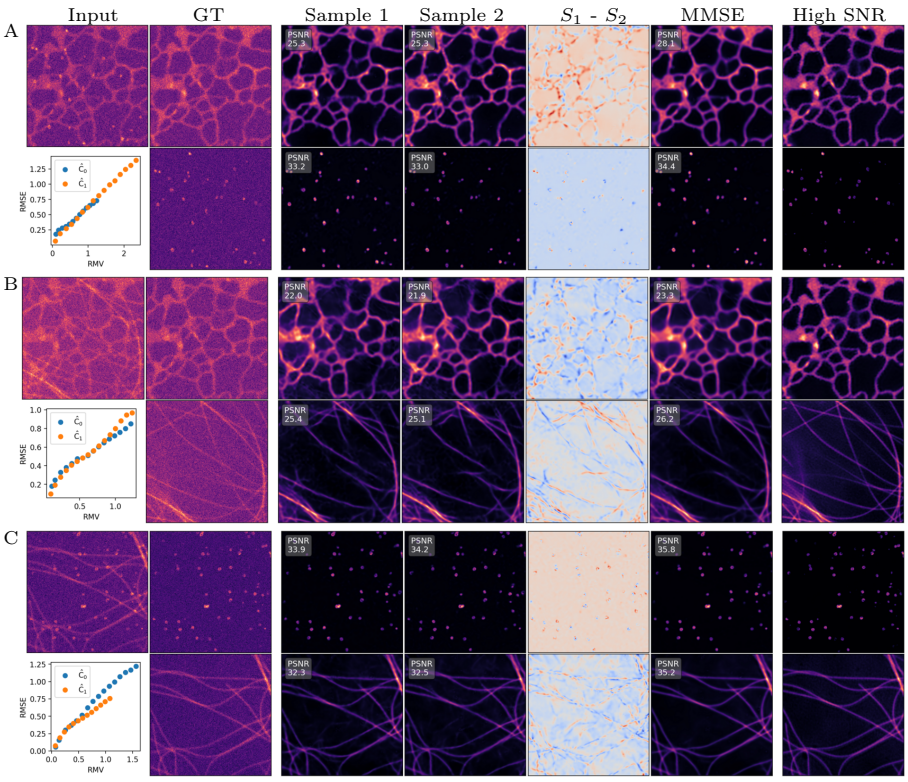


Fig. 3: Variational Sampling and Calibration. The VSE Network in denoiSplit is capable of sampling from a learned posterior. Here we show cropped inputs (256×256), two corresponding prediction samples, the difference between the two samples ($S_1 - S_2$), the MMSE prediction, and otherwise unused high SNR microscopy for three tasks, namely ER vs. CCPs, ER vs. MT, and CCPs vs. MT. The MMSE predictions are computed by averaging 50 samples. As before, we show PSNR w.r.t. high SNR patches. The dot plots in the first column show are calibration plots, showcasing that the error estimate we propose works well (see main text).

pick the most balanced variant, HVAE + *Regular-LC*, which we refer to as μSplit.

The second baseline, to which we refer to as *HDN⊕µSplit*, is a sequential application of Hierarchical DivNoising (HDN), one of the leading unsupervised denoising methods for microscopy datasets [19], and the μSplit setup from above. We first denoise all input images x_i^N and the respective two channel images $c_{1,j}^N, c_{2,j}^N$. Note that each set of the three kinds of image are denoised with a separately and specifically trained HDN.

Next, we use the denoised predictions of X^N, C_1^N, C_2^N to train a μSplit network as we did for the first baseline. The expectation from this baseline is to give denoised splitting results, which we also show in Fig. 4. We show the denoised HDN predictions in the supplement.

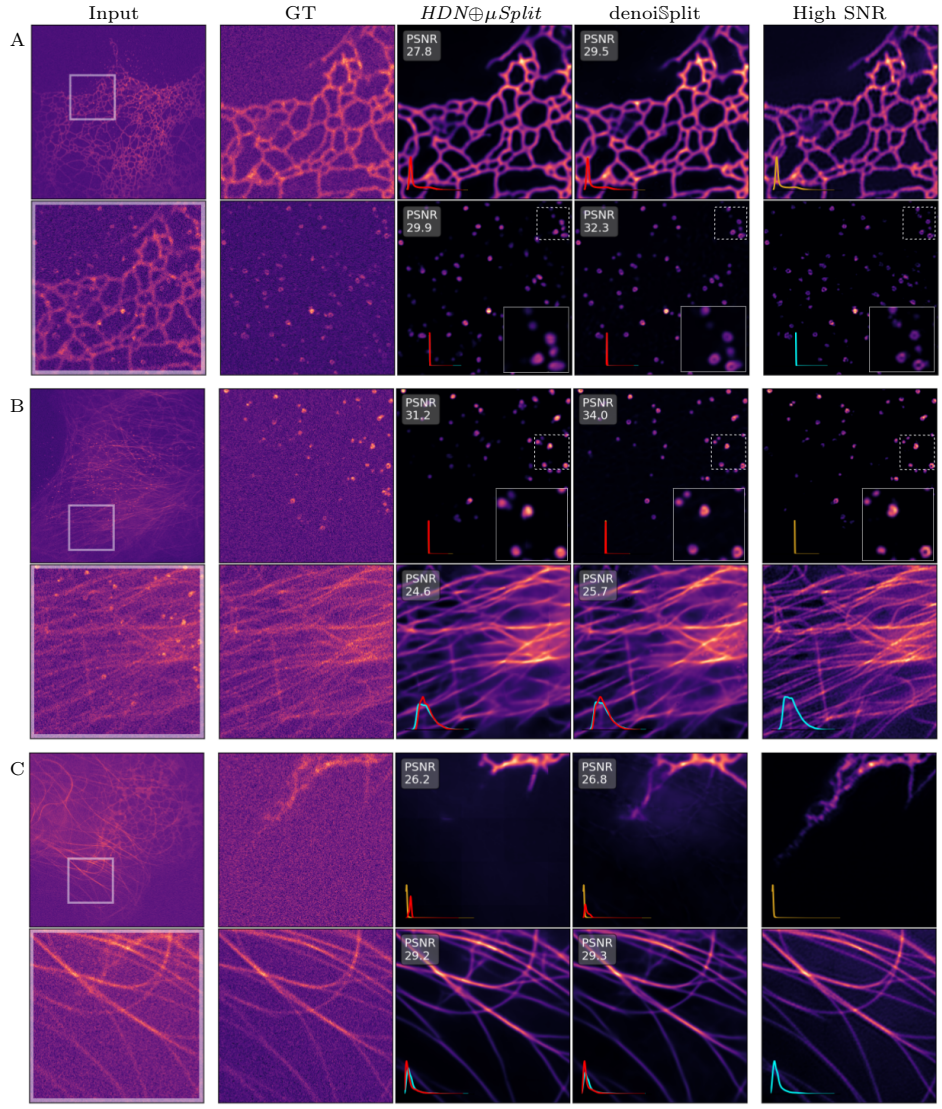
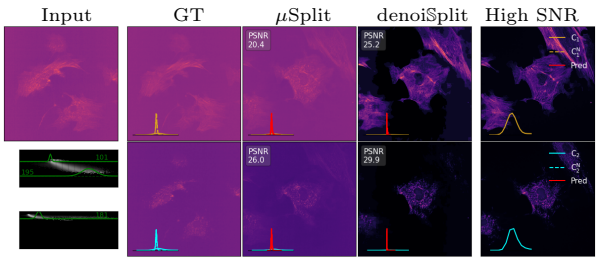


Fig. 4: Comparison to Sequential Baseline. For each panel (ER vs. CCPs, CCPs vs. MT, and ER vs. MT) we show the full input image and its (256×256) inset crop, corresponding noisy training data crops (GT), the results of the sequential denoising and splitting baseline (*HDN⊕μSplit*) and our end-to-end results obtained with denoiSplit. All predictions show the MMSE, obtained by averaging 50 sampled predictions. We show a few zoomed-in locations where the baseline under-performs. Note that such small differences might contribute little to evaluations via PSNR, but can make a huge difference for the downstream analysis of investigated biological structures contained in such microscopy data.

Task	Model	training [h]	Noise level parameters							
			$\lambda = 0$				$\lambda = 1000$			
			$\sigma = 1$	1.5	2	4	$\sigma = 1$	1.5	2	4
T1	μSplit	7	30.3	28.4	27.4	25.9	29.4	28.1	27.3	25.9
			0.853	0.748	0.66	0.42	0.844	0.750	0.667	0.437
	HDN⊕µSplit	11	37.3	34.9	33.8	29.4	36.3	34.3	33.3	29.4
			0.982	0.969	0.959	0.872	0.978	0.965	0.954	0.874
	Altered µSplit ***(ours)***	1.3	38.9	36.7	34.4	30.9	36.9	35.5	34.8	31.1
			0.988	**0.980**	0.965	0.909	0.982	0.974	**0.968**	**0.912**
	denoiSplit ***(ours)***	1.5	**39.7**	**36.8**	**35.4**	**31.1**	**37.9**	**36.3**	**35.0**	**31.2**
			0.989	0.978	**0.969**	**0.912**	**0.984**	**0.977**	0.967	**0.912**
T2	μSplit	6.3	26.0	23.9	22.7	21.0	25.2	23.7	22.7	21.1
			0.800	0.699	0.593	0.356	0.780	0.691	0.613	0.386
	HDN⊕µSplit	10	30.1	28.4	27.6	**25.3**	29.6	28.4	27.4	**25.2**
			0.909	0.873	0.845	**0.731**	**0.904**	0.874	0.835	**0.738**
	Altered µSplit ***(ours)***	1.5	30.4	28.8	26.9	23.4	**29.9**	27.4	**27.9**	24.4
			0.915	0.879	0.809	0.620	0.903	0.833	**0.845**	0.677
	denoiSplit ***(ours)***	1.6	**30.5**	**29.2**	**28.2**	25.1	**29.9**	**29.0**	27.0	24.8
			0.916	**0.886**	**0.860**	0.714	0.901	**0.885**	0.815	0.702
T3	μSplit	7.2	30.5	28.3	27.3	25.6	29.6	28.1	27.2	25.6
			0.880	0.793	0.713	0.46	0.877	0.800	0.722	0.476
	HDN⊕µSplit	11	38.4	35.9	34.3	29.3	36.8	34.9	33.8	29.3
			0.981	0.966	0.951	0.844	0.975	0.962	0.948	0.843
	Altered µSplit ***(ours)***	1.4	38.9	35.8	35.0	30.4	37.4	35.6	34.3	30.4
			0.985	0.968	0.960	0.867	0.979	0.968	0.953	0.865
	denoiSplit ***(ours)***	1.6	**40.1**	**37.3**	**35.7**	**30.6**	**38.1**	**36.6**	**35.2**	**30.7**
			0.986	**0.973**	**0.962**	**0.872**	**0.981**	**0.971**	**0.958**	**0.872**
T4	μSplit	7	25.9	24.3	23.6	22.4	25.2	24.2	23.5	22.4
			0.777	0.664	0.556	0.331	0.729	0.640	0.554	0.321
	HDN⊕µSplit	10.7	28.8	27.9	27.4	25.8	28.2	27.6	27.2	25.7
			0.852	0.817	0.790	**0.725**	0.840	0.810	0.787	0.716
	Altered µSplit ***(ours)***	1.3	29.4	28.5	27.5	25.9	**29.0**	27.8	27.3	25.8
			0.858	0.824	0.786	0.718	0.849	0.794	0.780	0.710
	denoiSplit ***(ours)***	1.5	**29.6**	**28.7**	**27.6**	**26.0**	**29.0**	**28.5**	**27.9**	**26.1**
			0.868	**0.835**	**0.787**	**0.725**	**0.854**	**0.828**	**0.799**	**0.727**

Table 1: Quantitative Results. We show quantitative evaluations for joint denoising and splitting experiments. The four corresponding tasks are abbreviated as T1: ER vs. CCPs; T2: ER vs. MT; T3: CCPs vs. MT, T4: F-actin vs. ER. For all experiments, we show the PSNR (sub-row 1) and MS-SSIM (sub-row 2) metrics across 8 noise levels: Gaussian noise levels of $\sigma \in \{1, 1.5, 2, 4\}$ and Poisson noise levels of $\lambda \in \{0, 1000\}$. The best performance per task and noise level is shown in bold. The third column additionally shows the training time on a single Tesla-V100 GPU (in hours). Not only does denoiSplit perform best, it does at the same time require considerably less training time.



Model	Eval.
μSplit	26.5
	0.872
HDN⊕μSplit	28.1
	0.887
Altered μSplit	31.1
	0.936
denoiSplit	31.0
	0.935

Fig. 5: Results on Actin *vs*. Mito Task: (Left) Here, qualitative evaluation of the different models on Hagen et al. [10] is shown. We also show High SNR channel images (not used during training) in last column and we show PSNR w.r.t. them. Noise models are shown in column one, second row. **(Right)** Quantitative evaluation of denoiSplit along with the baselines using PSNR (line 1) and range invariant MS-SSIM [26] (line 2, also see Supp. *Sec. 2* for details on the MS-SSIM variant). Note that HDN training in *HDN⊕μSplit* was quite unstable and so, we had to train it with a lower hierarchy count (3 as opposed to default 6).

5.3 Qualitative and Quantitative Evaluation of Results

We show the quality of results our methods can obtain in Figs. 2 to 5.

In Fig. 2, we show predictions on full input images (960×960 pixels). We can see that μSplit does unmix the given inputs and even partially reduces the noise. Still, the results by denoiSplit have a much higher resemblance to the high-SNR microscopy images shown in the rightmost column, even though they have never been presented during training (which was conducted only on noisy images, as shown in the second column). In Fig. 5, we show the results on Hagen et al. [10] dataset. Again, we observe denoiSplit outperforming μSplit and *HDN⊕μSplit*. Please refer to Supp. *Sec. 2* for more details.

In Fig. 3, we show zoomed 256×256 portions of full predictions to allow the reader to also appreciate the prediction quality of smaller structures contained in the data. Furthermore, we show two posterior samples (S_1 and S_2) and their highlighted differences ($S_1 - S_2$). The second to last column shows the average of 50 posterior samples (the approximate MMSE [20]). We show the calibration plot in the second row, first column of every panel where we can see a clear predictive (and close to linear) behavior of RMSE from RMV.

In Fig. 4, we also show 256×256 insets on inputs and results by *HDN⊕μSplit* and denoiSplit, showing that the fine details are better preserved by our proposed method. Note that these very details make all the difference when such methods are used on fluorescence microscopy data for the sake of downstream analysis.

For quantitative quality evaluations, we use the well known and established PSNR and MS-SSIM metrics [22] to evaluate all the results of our experiments and report these results in Tab. 1. These quantitative evaluations clearly show that our proposed methods improve considerably over μSplit and the sequential denoising and splitting baseline *HDN⊕μSplit*. In Supp. *Sec. 1 and 8*, we provide

results on more splitting tasks, including one failure mode. In Supp. *Sec. 3*, we show a proof-of-concept application to de-hazing and de-raining tasks.

6 Discussion and Conclusion

We present denoi𝕊plit, the first method that takes on the challenge of joint semantic image splitting and unsupervised denoising. This advancement in handling noise is crucial, considering the limitations microscopists face in acquiring high-SNR images, often due to practical constraints such as sample sensitivity and limitations in imaging technology. Unlike in a sequential approach of image denoising followed by training and applying μSplit on denoised data, denoi𝕊plit streamlines the process into a single end-to-end model, which on the one hand reduces the complexity and computational resources required for training and inference, and on the other hand leads to better results.

One of our methodological contributions is integrating noise models, originally developed for unsupervised denoising task, into the image-splitting setup. Given the fact that different microscope configurations produce images with different noise levels and Noise models can be made specific to each microscope configuration, the integration of noise models in our setup can go a long way in allowing a microscope-specific denoi𝕊plit setup thereby producing high-quality denoised and split predictions. Our work has additional interesting features, which we believe will improve its adoption among microscopists, *i.e.*, it supports variational sampling and calibration, allowing microscopists to observe multiple solutions for a given input and also to have an estimate of error for every predicted pixel.

In the future, we want to work on domain adaptation techniques to fine-tune existing models on noisy data from slightly different image domains or microscopy modalities. This will further ease the applicability of denoi𝕊plit for biomedical researchers. This continuous development aims to bridge the gap between computational imaging methods and the practicalities and limitations of modern microscopy, benefiting our overall goal of elevating the rate of scientific discovery in the life sciences by conducting cutting-edge methods research.

Acknowledgements

This work was supported by the European Commission through the Horizon Europe program (IMAGINE project, grant agreement 101094250-IMAGINE and AI4LIFE project, grant agreement 101057970-AI4LIFE) as well as the compute infrastructure of the BMBF-funded de.NBI Cloud within the German Network for Bioinformatics Infrastructure (de.NBI) (031A532B, 031A533A, 031A533B, 031A534A, 031A535A, 031A537A, 031A537B, 031A537C, 031A537D, 031A538A). Additionally, the authors also want to thank Damian Dalle Nogare of the Image Analysis Facility at Human Technopole for useful guidance and discussions and the IT and HPC teams at HT for the compute infrastructure they make available to us.

References

1. Ashesh, Krull, A., Sante, M.D., Pasqualini, F.S., Jug, F.: μSplit: image decomposition for fluorescence microscopy (2023)
2. Bahat, Y., Irani, M.: Blind dehazing using internal patch recurrence. In: 2016 IEEE International Conference on Computational Photography (ICCP). pp. 1–9 (May 2016)
3. Batson, J., Royer, L.: Noise2Self: Blind denoising by Self-Supervision pp. 1–16 (Jan 2019)
4. Berman, D., Treibitz, T., Avidan, S.: Non-local image dehazing. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1674–1682. IEEE (Jun 2016)
5. Buades, A., Coll, B., Morel, J.M.: A non-local algorithm for image denoising. 2005 IEEE computer society (2005)
6. Dabov, K., Foi, A., Katkovnik, V., Egiazarian, K.: Image denoising by sparse 3-D transform-domain collaborative filtering. IEEE Trans. Image Process. **16**(8), 2080–2095 (Aug 2007)
7. Dekel, T., Rubinstein, M., Liu, C., Freeman, W.T.: On the effectiveness of visible watermarks (2017)
8. Gandelsman, Y., Shocher, A., Irani, M.: "Double-DIP" : Unsupervised image decomposition via coupled deep-image-priors (2019), accessed: 2022-2-14
9. Ghiran, I.C.: Introduction to fluorescence microscopy. Methods Mol. Biol. **689**, 93–136 (2011)
10. Hagen, G.M., Bendesky, J., Machado, R., Nguyen, T.A., Kumar, T., Ventura, J.: Fluorescence microscopy datasets for training deep neural networks. Gigascience **10**(5) (May 2021)
11. Jin, L., Liu, J., Zhang, H., Zhu, Y., Yang, H., Wang, J., Zhang, L., Xu, Y., Kuang, C., Liu, X.: Deep learning permits imaging of multiple structures with the same fluorophores. bioRxiv (2023)
12. Kingma, D.P., Welling, M.: An introduction to variational autoencoders (Jun 2019)
13. Ko, J., Lee, S.: Self2Self+: Single-Image denoising with Self-Supervised learning and image quality assessment loss (Jul 2023)
14. Krull, A., Buchholz, T.O., Jug, F.: Noise2Void - learning denoising from single noisy images. arXiv **cs.CV**, 2129–2137 (Nov 2018)
15. Krull, A., Vicar, T., Prakash, M., Lalit, M., Jug, F.: Probabilistic Noise2Void: Unsupervised Content-Aware denoising. Frontiers in Computer Science **2**, 60 (Feb 2020)
16. Lehtinen, J., Munkberg, J., Hasselgren, J., Laine, S., Karras, T., Aittala, M., Aila, T.: Noise2Noise: Learning image restoration without clean data. arxiv.org (Mar 2018)
17. Levi, D., Gispan, L., Giladi, N., Fetaya, E.: Evaluating and calibrating uncertainty prediction in regression tasks. Sensors **22**(15) (2022)
18. Platt, J.: Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. Adv. Large Margin Classif. **10** (06 2000)
19. Prakash, M., Delbracio, M., Milanfar, P., Jug, F.: Interpretable unsupervised diversity denoising and artefact removal (Apr 2021)
20. Prakash, M., Krull, A., Jug, F.: DivNoising: Diversity denoising with fully convolutional variational autoencoders. ICLR 2020 (Jun 2020)
21. Prakash, M., Lalit, M., Tomancak, P., Krull, A., Jug, F.: Fully unsupervised probabilistic Noise2Void. arXiv **eess.IV** (Nov 2019)

22. Shroff, H., Testa, I., Jug, F., Manley, S.: Live-cell imaging powered by computation. Nat. Rev. Mol. Cell Biol. (Feb 2024)
23. Sønderby, C.K., Raiko, T., Maaløe, L., Sønderby, S.K., Winther, O.: Ladder variational autoencoders. Adv. Neural Inf. Process. Syst. **29**, 3738–3746 (Jan 2016)
24. Ulyanov, D., Vedaldi, A., Lempitsky, V.: Deep image prior. Int. J. Comput. Vis. **128**(7), 1867–1888 (Jul 2020)
25. Weigert, M., Royer, L., Jug, F., Myers, G.: Isotropic reconstruction of 3D fluorescence microscopy images using convolutional neural networks. In: Medical Image Computing and Computer-Assisted Intervention - MICCAI 2017. pp. 126–134. Springer International Publishing (2017)
26. Weigert, M., Schmidt, U., Boothe, T., M uuml ller, A., Dibrov, A., Jain, A., Wilhelm, B., Schmidt, D., Broaddus, C., Culley, S., Rocha-Martins, M., Segovia-Miranda, F., Norden, C., Henriques, R., Zerial, M., Solimena, M., Rink, J., Tomancak, P., Royer, L., Jug, F., Myers, E.W.: Content-aware image restoration: pushing the limits of fluorescence microscopy. Nature Publishing Group **15**(12), 1090–1097 (Dec 2018)
27. Zhang, K., Zuo, W., Chen, Y., Meng, D., Zhang, L.: Beyond a gaussian denoiser: Residual learning of deep CNN for image denoising. IEEE Trans. Image Process. **26**(7), 3142–3155 (Jul 2017)