Abstract—With recent deep learning based approaches showing promising results in removing noise from images, the best denoising performance has been reported in a supervised learning setup that requires a large set of paired noisy images and ground truth data for training. The strong data requirement can be mitigated by unsupervised learning techniques, however, accurate modelling of images or noise variances is still crucial for high-quality solutions. The learning problem is ill-posed for unknown noise distributions. This paper investigates the tasks of image denoising and noise variance estimation in a single, joint learning framework. To address the ill-posedness of the problem, we present deep variation prior (DVP), which states that the variation of a properly learnt denoiser with respect to the change of noise satisfies some smoothness properties, as a key criterion for good denoisers. Building upon DVP and under the assumption that the noise is zero mean and pixel-wise independent conditioned on the image, an unsupervised deep learning framework, that simultaneously learns a denoiser and estimates noise variances, is developed. Our method does not require any clean training images or an external step of noise estimation, and instead, approximates the minimum mean squared error denoisers using only a set of noisy images. With the two underlying tasks being considered in a single framework, we allow them to be optimised for each other. The experimental results show a denoising quality comparable to that of supervised learning and accurate noise variance estimates.

Index Terms—Image denoising, unsupervised learning, noise level estimation, deep learning without ground truth.

I. INTRODUCTION

IMAGE denoising refers to the process of decoupling image signals and noise from the observed noisy data. It arises from a range of real-world applications where image acquisition systems often suffer from noise degradation. Image denoising has been investigated in the last decades and remains one of the active research topics in image processing. To decouple the signals of images and noise, a typical high-quality denoising process requires either explicitly or implicitly modelling the characteristics of images and noise, the structures of which are often complex.

Many classic image denoising approaches are designed based on assumptions of certain image priors, for example, about smoothness [23], [26], self-similarity [5], [10], and sparseness of image gradients [14], [31] or wavelet domains [9], [11]. With proper priors, they are shown to be effective in recovering small image details such as edges or textures. Recently, supervised deep learning (e.g., [8], [39]), which does not require explicit modelling of image priors but is built on the high capability of deep neural networks in describing image data, offers a promising alternative denoising solution. Typical deep neural network based denoisers are trained using pairs of ground truth and noisy images. In doing so, image distributions are captured and encoded in the deep neural networks, which output high-quality denoising results. Nevertheless, one major limitation of these approaches is the assumption that a sufficiently large set of clean images is available for training, and such an assumption can be too strong in practical applications.

The strong requirement for clean training images can be circumvented by unsupervised learning techniques. These techniques learn to separate clean images and noise using unpaired noisy and clean images (e.g., [7], [35]), or using solely noisy data, for example, from paired noisy observations of single images [21], a set of unpaired noisy images [3], [17], [25], [32], or a single image [34]. Given only single noisy images, the learning problems come with more ambiguities given the fact that noise models are not readily available from the data itself, in comparison to supervised learning where noise models can be derived from the given paired noisy-clean images and hence are known during the learning process. Therefore, to access optimal denoising quality, noise models (e.g., Gaussian distributions with known variance) are often assumed or a step of noise variance estimation is required.

Image denoising and noise variance (noise level) estimation are two closely connected tasks. An accurate estimate of noise variance is naturally a good criterion for justifying whether the size of noise being removed is proper, whereas a wrong estimate can lead to unexpected denoising effects such as residual noise or over-smoothing of the denoised images. Various noise estimation approaches have been developed for different types of noise, such as Gaussian noise [11], [16], [27] and Poisson noise [29]. In a common application setting of denoising, while the obtained noise levels can be used to configure relevant denoising algorithms, the errors in the denoised results cannot be directly fed back to the noise estimation, resulting in suboptimal denoising processes.

In the paper, we address the ill-posed problem of learning denoisers from noisy data without ground truth images or specific noise models given. We make the assumption that the noise is zero mean and pixel-wise independent conditioned on the clean images, and that the available data consists of...
samples of coupled signals (noisy images). The aim is to find a denoising model and estimate the level of noise from the samples simultaneously. Noise level estimation remains a challenging problem in image processing, which we will address in our algorithm. An end-to-end learning framework is proposed for handling both tasks. Specifically, we present a noise approximation based objective function that connects the mean squared error to a noise variance estimation problem, where the noise variance errors lead to interpretable artefacts in denoised outputs of the learnt denoiser. We introduce deep variation prior (DVP), which describes the smoothness properties of the variation of a learnt denoiser with respect to the change of noise. Our unsupervised framework incorporates the feedback of the denoisers (artefacts) and DVP to improve both the denoisers and noise variance estimates during the learning process.

In this work, we leverage the concept of the partially linear structure [17] for unsupervised denoising. However, different from [17] which relies on known noise variance, our approach not only learns a denoiser but also outputs estimated noise variances. In particular, it incorporates noise variance modelling in the learning process and iteratively improves the noise variance estimation based on the data and the learned denoisers. As a result, it can produce better denoising outcomes by reducing the errors in the estimated noise variance, as opposed to relying on a fixed noise variance. As a key difference from [17], we introduce deep variation prior as a criterion for training denoisers and use it to model the noise variance.

The main contributions of this work are summarised as follows. First, we propose a joint model for learning image denoisers and estimating noise variance from a set of unpaired noisy images only, hence removing the need for supervised data or prior knowledge on the level of noise. Second, our framework provides a theoretical approximation to the mean squared error function for denoising with single observations and under unknown noise variance. With such an approximation property, it can be extended to handling other inverse imaging problems where noise-free measurements are not available. Third, we introduce deep variation prior, a criterion for adjusting both noise variance estimates and denoisers based on the learnt denoisers during the learning process. This allows one to combine the knowledge of denoising and noise variance inference and avoids training multiple denoising models due to uncertain noise variances. Finally, we report experimental results showing accurate noise variance estimation as well as denoising with a quality comparable to supervised methods.

II. RELATED WORKS

Our work is relevant to both model-based and data-driven denoising, as well as noise variance estimation methods. We review related works in these areas.

Model-Based and Data-Driven Denoising: The existing denoising approaches lie in two main categories: model-based denoising and data-driven denoising. Typical classic model-based denoisers include Total Variation models [31], BM3D [10], Non-Local means [5] and Wavelet methods [9]. Prior knowledge about the underlying images to be restored is a core element of these approaches.

Recent powerful data-driven approaches are based on deep neural networks. In particular, convolutional network based models such as DnCNN [39] and TNRD [8] have shown promising results in image denoising. More recently, explicit image modelling has been considered in the design of deep architectures for denoising, leading to the state-of-the-art performance in supervised learning. For example, the SCUNet [37] combines local modelling and global modelling by utilising the residual connection, swim transformer block, and the UNet structure [30]. The GRL network architecture [22] considers explicit modelling of image hierarchies in the global, regional, and local range through different attention mechanisms. The MWDCNN approach [33] integrates modelling based on wavelet transformation in its architectures. Deep denoising models that take both the noise level and the noisy image as input are also developed to handle different noise levels, for example, in DRUNet [38]. Moreover, denoising has also been explored in the context of video sequences or multiple images. The bidirectional streaming video denoising framework [28] uses buffer blocks to captures temporal receptive fields and bidirectional temporal fusion, which incorporate information from both past and future frames. These models are data-driven and typically trained using pairs of noisy images and ground truth. Our work investigates prior knowledge about images and denoising operators for data-driven denoising, which leverages intrinsic structures of noisy data.

Unsupervised Learning for Denoising: Unsupervised learning methods have been studied for the tasks of image denoising, and they mitigate the strong data requirement of supervised approaches. One class of methods train denoisers using a set of unpaired noisy images and clean images. For example, [7] proposes an adversarial learning scheme that trains a network for predicting noise using extracted noisy patches, and the predicted noise is then combined with the clean images to train denoisers. This method has been extended to incorporate signal dependent noise [18]. In contrast, GDANet [36] trains two networks simultaneously for denoising and noise generation. The Self-Consistent GAN [35] proposes to train a network for extracting noise maps from noisy images, where the extracted noise maps are constrained by three self-consistency losses. These adversarial learning based methods are able to address different types of noises, including signal dependent and structured noise. However, they require the availability of clean images as training data, which is a crucial element for learning the underlying noise pattern.

Another class of unsupervised method relies on noisy training data. One of the early works on this line is the Noise2Noise [21] which requires pairs of two independent noisy observations of single images for training. This method approximates the mean squared error loss function with the noisy samples. Ground truth images are not needed but at the same time, the paired noisy samples explicitly imply information about the noise variance. The Noise2Self [3] offers an alternative way of unsupervised denoising that requires only a single noisy observation per image. It follows a blind-spot strategy which removes a subset of pixels of noisy
images in the input during training. While noise levels are not needed during training, Noise2Self does not take the full information of the noisy images as input. The gap between Noise2Self and supervised learning can be made smaller with a post-processing technique proposed in [20], where the outputs of the denoisers are combined with the known noise distributions and inputs for more accurate results.

Building on a partially linear structure that is learnable from single noisy images, an unsupervised framework is proposed in [17] for learning denoisers without removing pixels from images. The framework requires the noise variances data but information about the precise noise distribution is not needed. For Gaussian noise with a known noise level, the Recorrupted-to-Recorrupted [25] and SURE [32] methods learn denoising from unpaired noisy observations of images. The FBI-Denoiser method [6] trains a pixel-wise affine denoiser on noisy images that are normalised with the Generalized Anscombe Transformation [1]. The approach assumes the underlying noise is Poisson-Gaussian noise and requires training two different networks separately, the first one for noise level estimation and the other for denoising. The first one is trained by using off-the-self Gaussian noise variance estimation methods on the normalised images. The FBI-Denoiser is built on a special class of networks called blind-spot networks while the proposed method does not require using the blind-spot architectures.

The proposed method in this work is different from existing methods. It uses a set of single noisy observations of images without a known noise distribution for learning denoisers, and it combines denoising with explicit noise variance modelling, allowing these two components to be optimised for each other and removing the need for an external noise variance estimation process. With the proposed deep variation priors, we estimate simultaneously the denoisers and the noise variance from solely noisy data. Moreover, the proposed model is end-to-end and network-architecture agnostic.

Noise Variance Estimation: Understanding the noise variance in noisy images is a key step for various image processing tasks. Many existing noise variance estimation (NVE) approaches are developed by assuming Gaussian white noise where the noise is signal-independent. Similar to denoising, NVE needs prior information about the images and the noise. Based on the different behaviours of noise and signal in the wavelet domain, noise variance can be estimated by selecting proper wavelet coefficients [11]. Pimpalkhute et al. [27] propose to use edge detectors to remove edge information from the wavelet domain and improve the estimation accuracy using polynomial regression. NVE can also be done by using spatial filtering, for example, with the Laplacian matrices that capture local structural information [16]. Convolutional Neural Networks have also been used for NVE. Bahnemiri et al. [2] proposed SDNet for predicting a sigma-map (pixel-wise noise standard deviation) for non-stationary noise, where the SDNet is trained using on ground truth clean images. In the signal-dependent noise setting, NVE is more complex given the fact that the noise variance varies across the pixels. Foi et al. [12] develop a two-stage approach that first estimates local noise standard deviations on a set of selected regions and then fits a global parametric model for the noise variance. A patch-based method is proposed in [29] where patches are selected and clustered based on intensity and variances. Regression analysis is then carried out based on the noise variances computed within each cluster. NVE in this work, in contrast, is built upon the feedback of denoisers, and noise variances are iteratively updated according to the denoising quality.

Regularisation Parameter Selection: Regularisation parameter selection (RPS) has been used for finding denoising solutions with balanced data-fitting and regularisation in variational approaches. One type of method for RPS is based on Morozov’s discrepancy principle [4] which seeks regularisation parameters such that the distance between the obtained solution and the noisy image matches the underlying noise variance. The L-curve criterion [13] provides an alternative approach for RPS, which chooses the optimal parameters near the corner of the L-curve in the plot of the regularisation term versus the data-fitting term. With a similar strategy to these approaches, the parameters of the proposed joint models are decided based on the denoised results such that they meet the relevant denoising criteria.

III. Problem Formulation and the Main Approach

In image denoising, one aims to recover a clean image \( x \) given its noisy version \( y \in \mathbb{R}^m \), from the noise corruption model

\[
y = x + n,
\]

where \( m \) is the number of pixels, and \( n \) represents the noise vector. Both \( x \) and \( n \) are random vectors following some underlying distributions. In this work, we assume \( n \) has zero mean and is pixel-wise independent conditioned on \( x \). However, it can be either dependent or independent of \( x \). The distribution and variance of \( n \), however, are unknown. Throughout this paper, we use boldface letters (eg. \( x \)) to denote random variables/vectors while the specific realisations or observed values of these random vectors are denoted by non-boldface letters (eg. \( x \)).

A. The Learning Problems

To find a solution for (1), we aim to find a denoiser \( R : \mathbb{R}^m \rightarrow \mathbb{R}^m \) that maps \( y \) to \( x \). The denoiser \( R \) can be learnt from data, if a set of samples are available.

Paired Ground Truth and Noisy Images: The most basic scenario is supervised learning, where we assume that a set of noisy samples \( y^{(i)} \) and their associated clean images \( x^{(i)} \) are given. Mathematically, the denoising learning can be formulated as the problem of obtaining \( R \) from the data:

\[
\{(y^{(i)}, x^{(i)})\} \rightarrow R,
\]

where we expect \( R(y^{(i)}) \) to be equal or similar to \( x^{(i)} \).

In general, the exact recovery of \( x \) from \( y \) is often infeasible due to the fact that the problem (1) is underdetermined, and an ideal learning process is to find an estimate \( R(y) \) whose distance to \( x \) is minimal. A typical measure of their distance is the \( L_2 \) distance. Based on random variables \( x \) and \( y \), the
distance can be described by the mean squared error (MSE) defined by

$$\mathcal{L}(R) = \mathbb{E}\left(\|R(y) - x\|^2\right),$$

(2)

where the expectation is taken over $x$ and $n$. The minimiser of MSE gives the conditional mean estimator (also known as minimum MSE estimator) $\mathbb{E}(x | y)$. In summary, (P1) assumes that we have access to $\{(y^{(i)}, x^{(i)})\}$, which enables a direct estimation of MSE and hence $\mathbb{E}(x | y)$. In many practical situations, however, samples of clean image $x$ are not given.

**Paired Noisy Images:** Instead of relying on clean images for solving the problem (P1), one can estimate $R$ from paired noisy samples. More specifically, given independent observations $y^{(i,1)}, y^{(i,2)}$ of a single image $x$, we have the following learning problem

$$\{y^{(i,1)}, y^{(i,2)}\} \to R.$$  (P2)

This problem has been investigated in the Noise2Noise approach [21], which provides an estimate of the conditional mean $\mathbb{E}(x | y)$. It is worth mentioning that the paired observations in (P2) encode information about noise distributions which is helpful for learning $R$. As an example, the average of $(y^{(i,1)} - y^{(i,2)})^2/2$ ($i = 1, 2, \cdots$) provides an unbiased estimate of the noise variance.

**Single Noisy Images:** Further relaxation of the condition in the problem (P1) is to use solely a set of single noisy images. The learning problem, in this case, is described by

$$\{y^{(i)}\} \to R,$$  (P3)

where elements in the set $\{y^{(i)}\}$ are not linked explicitly to each other (therefore unpaired). Given only $\{y^{(i)}\}$, the learning problem (P3) is ill-posed as there exist infinitely many ways of interpreting the data. In particular, the conditional mean $\mathbb{E}(x | y)$ is not uniquely defined from the data.

In this work, we focus on the problem (P3). An important point to highlight is that the optimal denoisers depend on the information of how much noise is present in the noisy data, which is unknown and therefore needs to be estimated from the data. The aim is to develop a regularised learning solution providing high-quality denoising as well as accurate noise level information.

In the following, we first present a noise approximation objective function that connects denoisers to estimated noise variance (Subsection B), and then introduce deep variation prior which acts as a criterion for good denoisers and for accurate noise variance estimations (Subsection C). A summary of the proposed method is given in Subsection D.

### B. Noise Approximation Learning Objectives

Given samples of $y$, if there exists a denoiser $R$ such that the reconstructed image looks similar to a clean one, i.e., $R(y) \approx x$, then it implies an approximation to the noise

$$R_n(y) := y - R(y) \approx n.$$

Formally, the problem of minimising the MSE (2) is equivalent to minimising

$$\mathcal{L}(R) = \mathbb{E}\left(\|R_n(y) - n\|^2\right),$$

(3)

where we expect the output of $R_n(\cdot)$ to follow a similar distribution to the noise.

The objective function (3) reformulates the denoising problem as a noise estimation problem. From the viewpoint of (3), the correct amount of noise to be removed is characterised by the noise vector $n$, which reflects the true noise distribution.

One of the main obstacles to applying (3) in denoiser learning is that, in the unsupervised setting of (P3), neither the noise vector nor the noise distribution is accessible. This is in contrast to supervised learning where such information can be directly inferred from the noisy-clean pairs. To handle the unknown elements in the noise distribution, we start from a variant of (3) by relaxing the requirement of an explicit $n$, and then based on this, develop a joint learning framework for both denoising and noise variance estimation.

#### B.1 Auxiliary Vectors and Partially Linear Denoisers [17]:

Our approach to handling the denoising task is based on an extension of the previous work [17], in which the main idea is to exploit a partial linear structure in denoisers which helps to approximate the MSE. Next, we introduce the definition of the partially linear structure and how it can be used in the learning problems.

**Definition 1 (partially linear structure [17]):** A denoiser $R : \mathbb{R}^m \to \mathbb{R}^m$ has a partially linear structure if it can be decomposed into

$$R(\hat{y}) = g(x) + L\hat{n} + \hat{e},$$

(4)

where $\hat{n} = n + \alpha z$, $\hat{y} = y + \alpha z$, $\alpha$ is a constant, $z$ is an auxiliary random vector following a certain distribution, $g$ is a linear or nonlinear function, $L$ is a linear operator independent of $\hat{n}$, and $\hat{e}$ has a small variance. If $R$ has a partially linear structure, then we call it a partially linear denoiser.

The set of partially linear denoisers has been shown to contain good denoisers [17]. The intuition behind this definition (4) is that, given small $\hat{e}$ and bounded operator $L$, the operator $R$ is stable with respect to the change of noise. The partially linear structure can be useful for learning denoisers from data where ground truth images are missing. In particular, one can derive approximations of MSE under the set of partially linear denoisers, the details of which will be explained next.

We first assume that the random auxiliary vector $z$ satisfies

$$\mathbb{E}(z | n, x) = 0, \quad \text{Cov}(z | x) = M\text{Cov}(n | x),$$

(5)

where $\text{Cov}(z | x) := \text{Cov}(z, z | x)$ denotes the covariance matrix of $z$ conditioned on $x$, and $M$ is a matrix independent of $n$ and conditional on $x$. If both $z$ and $n$ are pixel-wise independent conditioned on $x$, then $M$ is a diagonal matrix. The definition of $z$ is general because of the following reasons.

- The vector $z$ is not restricted to a specific distribution (and is possibly dependent on $x$ and $n$).
- $\text{Cov}(z | x)$ can be an arbitrary covariance matrix, and $M$ always exists as long as $\text{Cov}(n | x)$ is non-singular.

It is worth mentioning that [17] considers only the case where $M = I$, hence assuming the noise variance $\text{Cov}(n | x)$ is known or can be estimated (while here $z$ can be defined without knowing the noise variance). Associated with $z$, a new
loss function of $R$ is defined without $n$ by
\[ \widehat{L}(R) = \mathbb{E}\left( \| R(y + \alpha z) - (y - z/\alpha) \|^2 \right), \]
(6)
where $\alpha$ is a small positive number, and the expectation is taken over $y$ and $z$. Importantly, the definition of $\widehat{L}$ does not require an explicit expression of $x$ or $n$.

In the special case where $M = I$, it has been shown in [17] that $\widehat{L}$ provides a good approximation to (3) if $R$ has a partially linear structure (Definition 1). Consequently, minimising the loss function (6) over the set of partially linear denoisers leads to denoisers with small values of MSE. However, the condition $M = I$ may not hold if $z$ is generated based on an inaccurate noise variance estimate.

**B.2 Noise Approximation Without Explicit $n$:** Returning to the discussion of the noise estimation problem in (3), we can now establish a connection between $\widehat{L}$ (defined without $n$) and the loss (3), in a generic setting, where $z$ is defined in (5) and $M$ is unknown. As we will see in the next theorem, the information of $M$ is encoded into the loss $\widehat{L}$ and its minimiser. For ease of presentation, here we assume that $\hat{e} = 0$.

**Theorem 1 (noise approximation):** Let $R$ be a denoiser satisfying the partially linear structure (4) with $\hat{e} = 0$ and assume that $n$ has zero mean, then $\widehat{L}$ (defined in (6)) satisfies
\[ \widehat{L}(R) = \mathbb{E}\left( \| R_n(\hat{y}) - Mn \|^2 \right) + c, \]
(7)
where $R_n(\hat{y}) := y - R(\hat{y})$, $c$ is a constant independent of $R$, and the expectation is taken over $z, x$ and $n$.

**Proof:** By the definitions of $\widehat{L}$ and $R_n(\hat{y})$,
\[ \widehat{L}(R) = \mathbb{E}\left( \| R_n(\hat{y}) - z/\alpha \|^2 \right) = \mathbb{E}\left( \| (R_n(\hat{y}) - Mn) + (Mn - z/\alpha) \|^2 \right). \]
(8)
According to (4), the first term on the right hand side of (8) can be rewritten as
\[ R_n(\hat{y}) - Mn = x - g(x) - L\hat{n} + (n - Mn). \]
Denoting by $(\cdot, \cdot)$ the inner product operator, we will show that the cross term $\mathbb{E}( (R_n(\hat{y}) - Mn, Mn - z/\alpha) )$ of (8) is constant in the following three steps.

(i) By the assumption $\mathbb{E}(n | x) = 0$ and $\mathbb{E}(z | x) = 0$,
\[ \mathbb{E}( (x - g(x), Mn - z/\alpha) ) = 0. \]
(ii) It follows from the assumption $\mathbb{E}(z | n) = 0$ that
\[ \mathbb{E}( (L\hat{n}, Mn - z/\alpha) ) = \mathbb{E}( (Ln + \alpha Lz, Mn - z/\alpha) ) = \mathbb{E}_x( \text{tr}(\text{Cov}(Ln,Mn | x)) - \text{tr}(\text{Cov}(Lz, z | x))) = \mathbb{E}_x \left( \text{tr}(L\text{Cov}(n | x) M^T) - \text{tr}(L\text{Cov}(z | x)) \right) = 0, \]
where $M^T$ denotes the transpose of $M$, $\text{tr}(\cdot)$ is the trace of a matrix, and the second and the last equalities follow from the condition (5).

(iii). Let $c_1 := \mathbb{E}( (n - Mn, Mn - z/\alpha) )$. This together with (i) and (ii) gives
\[ \mathbb{E}( (R_n(\hat{y}) - Mn, Mn - z/\alpha) ) = \mathbb{E}( (n - Mn, Mn - z/\alpha) ) = c_1, \]
(9)
which is independent of $R$.

Finally, let $c_2 := \mathbb{E}( \| Mn - z/\alpha \|^2 )$, and $c := c_1 + c_2$, then (8) and (9) give (7).

**Remark 1:** Comparing (7) with (3), we observe that (7) is a variant of (3). By Theorem 1, $\widehat{L} - c$ is an approximation to $\mathbb{L}$ given in (3) when $M$ is close to the identity matrix $I$.

**Remark 2:** It can be seen that (7) holds without regard to the noise types, and $\widehat{L}$ depends on the covariance of $z$ but not its distribution.

**Remark 3:** For the case with $\hat{e} \neq 0$, the distance between $\widehat{L}(R)$ and $\mathbb{E}( \| R_n(\hat{y}) - Mn \|^2 ) + c$ is small if $\mathbb{E}( \| \hat{e} \|^2 )$ is small, as required by the partially linear structure. However, the partially linear structure is essential, as otherwise the distance between $\widehat{L}(R)$ and $\mathbb{E}( \| R_n(\hat{y}) - Mn \|^2 ) + c$ can be large. Therefore, the partially linear structure will be used in the algorithms to ensure such proximity.

By Theorem 1, to obtain a denoiser that removes the correct amount of noise, we aim to have $M = I$ (i.e., $n$ and $z$ have the same conditional covariance), in which minimising $\widehat{L}$ under the condition of (4) is similar to finding $R_n(\hat{y})$ that best matches the noise $n$ in the mean squared distance. If $M \neq I$, the minimisation process fits $R_n(\hat{y})$ to a variant of $n$, and consequently, $R$ may not output clean images, however, $R$ encodes useful information of $M$ based on the relationship (7).

The formulations (6) and (7) therefore provide a theoretical foundation for simultaneous unsupervised denoising and noise variance estimation. Based on these formulations, one can learn a denoiser without knowing noise variance in advance, and the unknown noise variance is reflected in the denoised outputs, which then can be used to improve the denoiser.

More specifically, the idea is to minimise $\widehat{L}$ under the condition of (4) while at the same time adjusting the covariance $\text{Cov}(z | x)$ such that $M$ gets closer to $I$. To update $\text{Cov}(z | x)$ one needs to estimate $M$. One important question is how $M$ can be inferred from the denoising outputs. With the deep variation priors introduced in the next subsection, one can exploit the feedback of the denoising outputs.

**C. Deep Variation Prior**

The matrix $M$ is connected to $R_n$, based on the fact that the minimisation of the objective function (7) motivates a noise estimation satisfying
\[ R_n(\hat{y}) \approx Mn, \]
(10)
where $R_n$ (or the denoiser $R$) can be represented by a parameterised function, typically a deep neural network. Recall that matrix $M$ on the right hand side is unknown and needs to be estimated. However, for small $\alpha$ (hence $\hat{y} \approx y$), (10) implies an approximate linear correspondence (defined by $M$) between the outputs of $R_n$ and the noise, conditioned on $x$.

Nevertheless, $M$ cannot be directly computed from $R_n(\hat{y})$ because $a) n$ is unknown and $b)$ the two sides of (10) are not precisely equal. To study these problems, we define the variation of denoisers.
Definition 2 (variation of denoisers): Let $R$ be a denoiser as an operator from $\mathbb{R}^m$ to $\mathbb{R}^m$. The variation of $R$ with respect to noise is defined as

$$\delta R := R(y^{(2)}) - R(y^{(1)}),$$

where $y^{(1)}$ and $y^{(2)}$ are noisy versions of a single image.

Similarly, the variation of $R_n$ is defined as $\delta R_n = R_n(y^{(2)}) - R_n(y^{(1)}) = (y^{(2)} - y^{(1)}) + (R(y^{(1)}) - R(y^{(2)})) = \delta y - \delta R$, where $\delta y := y^{(2)} - y^{(1)}$. Here for the simplicity of illustration, we take the auxiliary vector $z$ as zero. With the definition of $\delta R$, for each image $x$, the condition (10) implies

$$\delta R \approx \delta y - M\delta y. \quad (11)$$

The variation of $R$ encodes information of $M$. For instance, if $M = \rho I$ for $\rho < 1$, i.e., the noise variance is underestimated, then $\delta R \approx (1 - \rho)\delta y$ contains some residual components of $\delta y$.

In comparison with (10), the formulation (11) does not involve explicit expressions of $n$ (which is unknown) and indicates an approximation of the matrix $M$ (and hence Cov$(n | x)$) from $\delta R$ and $\delta y$. Notice that the two sides of the approximate equality are not equal in general, because the noise usually cannot be completely separated from noisy images (i.e., $E[\|R_n(\hat{y}) - Mn|z\|^2] \neq 0$). However, the components of $\delta y - M\delta y$ still play a role in $\delta R$, subject to the uncertainty in the noise and image separation. The connection between $\delta R$ and $M$ can be better characterised by taking into account the uncertainty.

In the following part, we will focus on the variation of denoisers and investigate the properties of variation. We introduce deep variation prior (DVP), as a prior for $\delta R$ and more generally a criterion for denoising. The main idea is that, for a learnt denoiser $R$ that correctly approximates the clean images, the variation $\delta R$ is a piece-wise smooth function. Such a prior can be interpreted from the following two perspectives.

- For denoiser $R$ that correctly predicts smooth regions of the image, the predictions are expected to be minimally dependent on the changes of noise, and hence the variation $\delta R$ should be smooth on these regions as well.
- Both images and noise have non-smooth components. However, if $M$ is equal to the identity matrix $I$, then both image structures and noise structures are effectively removed from the right-hand side of (11). Therefore, the minimisation of $\tilde{\mathcal{L}}$ in (7) tends to remove these structures from $\delta R$.

Particularly, the variation $\delta R$ does not necessarily equal zero for $M = I$, but it is described as a function with smoothness properties, in contrast to the right-hand side of (11) which is identically zero.

The concept of DVP for learning denoisers in the unsupervised setting with estimated noise variances is illustrated in Figure 1. In particular, Figure 1 (c)-(e) contains examples of $\delta R$ for $R$ obtained by minimising $\tilde{\mathcal{L}}$ over the set of partially linear denoisers, under different estimated noise variance (here we set Cov$(z | x)$ to the estimated noise variance). The denoiser learnt with the true noise variance (Figure 1 (c)) leads to a variation image that is smooth except on a small set of locations. The variations of denoisers that are learnt with overestimated (or underestimated) noise variance are not smooth (see Figure 1 (d)-(e)). Intuitively, if $M \neq I$, then parts of the noise will remain in $\delta R$ according to the relationship (11). Typical non-smooth components of $\delta y$ lead to non-smooth noise patterns in $\delta R$ which violates the criterion of DVP. Therefore, to obtain a more accurate description of the relationship in (11), the variation $\delta R$ can be decomposed into two parts:

$$\delta R = \delta R^\parallel + \delta R^\perp, \quad (12)$$

where $R^\parallel$ is the learnt denoiser associated with the ideal case of $M = I$, and $R^\perp$ corresponds to the residual components of noise $(I - M)\delta y$ in $R$ (violating DVP), from which we estimate $M$ and learn better denoisers.

So far, we have established a connection between $M$ and $\delta R$ (see Equation (11)) and introduced DVP for describing the properties of $\delta R$. Recall that $M$ is unknown and needs to be estimated during the learning process (see Subsection III.B.2). In Section IV, we will explain how to estimate $M$ from $\delta R$ (under zero-mean and pixel-wise independent noise) which is then used to update the covariance of $z$. Before that, an overall picture of the proposed method is given below.

D. Summary of the Proposed Method

This subsection contains a summary of the proposed method, followed by the details and specifications of the learning framework presented in the next section.

Given a set of samples of noisy images $y$ and in the absence of both the ground truth images and known noise distributions, our method finds a denoiser through (6) (an approximation to the MSE) and an estimate of the noise variance through the random vector $z$ in (5). It is an iterative method, each iteration of which consists of two main steps:

(S1) Minimising the loss function $\tilde{\mathcal{L}}$ over the sets of partially linear denoisers $R$, i.e., with the partial linearity constraint (4) such that $\hat{e}$ is small.

(S2) Based on the obtained $R$, computing the variation $\delta R$, estimating the underlying matrix $M$ with DVP, and then using the estimated $M$ to update the covariance of $z$ such that the new $M$ is as close to the identity matrix $I$ as possible.

The two steps (S1) and (S2) are carried out alternatingly in a single learning process. The process aims to find an estimate of the noise variance (through updating $z$ such that $M \approx I$) and the associated denoiser (based on the approximate in (7) where $M \approx I$).

We underline that, instead of directly estimating noise variance from raw-data as in many classic approaches, our noise variance estimation is based on the learnt denoisers, hence incorporating the clean image distributions encoded therein as well as feedback from the denoisers. The interplay between the two components allows them to be optimised for each other. In other words, the updates of $R$ in (S1) enable better estimation of the noise variance in (S2), while the improved noise variance ($M$ is closer to $I$) contributes to more accurate denoisers implied by Theorem 1.
Fig. 1. Deep variation prior reflects how the outputs of a denoiser change as the noise changes. Best view in colour. Given a noisy image \( y(1) \) and its perturbed version \( y(2) \) (with different noise) in (a), the variation of a denoiser \( R \) is an image defined as 
\[
\delta R := R(y(2)) - R(y(1))
\]
as shown at the bottom of (b). The red dots in the plots in (c)-(e) illustrate how the variation \( \delta R \) is connected to the noise change 
\[
\delta y := y(2) - y(1)
\]
on one row of the image: (c) is the result of a denoiser that is trained with the true noise variance, and the variation (represented by the red dots) is smooth except in a small set of locations. The plots in (d) (resp. (e)) show the output of a denoiser that is learnt with underestimated (resp. overestimated) noise variance, and in contrast to (c), it is less smooth and has residual noise.

IV. DEEP LEARNING FRAMEWORK FOR DENOISING AND NOISE VARIANCE ESTIMATION

In the rest of the paper, we focus on pixel-wise independent noise, which simplifies the structure of \( M \). Unless specified otherwise, we assume that

(A1) The entries of the noise \( n \) are zero-mean and independent random variables, and \( n_i \) (the \( i \)th entry of \( n \)) is conditional on \( x_i \) but independent of pixel locations.

(A2) The vector \( z \) is zero-mean and pixel-wise independent conditioned on \( x \) and \( n \).

Therefore, \( \text{Cov}(z | x) \), \( \text{Cov}(n | x) \) and \( M \) are diagonal matrices. Let the conditional variance of \( z_i \) be denoted by \( \text{Var}(z_i | x_i) \), then
\[
\text{Cov}(z | x) = \text{diag}(\text{Var}(z | x))
\]
where \( \text{Var}(z | x) \) represents the vector of \( \text{Var}(z_1 | x_1), \ldots, \text{Var}(z_m | x_m) \), and \( \text{diag}(\cdot) \) converts a vector to a diagonal matrix. Similarly, \( \text{Cov}(n | x) = \text{diag}(\text{Var}(n | x)) \). Then the second equality in (5) is equivalent to

(A3) \( \text{Var}(z | x) = M \text{Var}(n | x) \).

The diagonal entries of \( M \) are \( M_{ii} = M_{ii}(x_i) \). For ease of presentation, we assume \( \text{Var}(n_i | x_i) \neq 0 \) for all \( i \), hence \( M \) exists for any distributions of \( z \).

Next, we first formulate the variance of \( z \) using parameterised functions. Then we present how to minimise (6) with constraints based on the generated vectors \( z \). Finally, the methods of how to update the variance of \( z \) based on the deviation \( \delta R \) and DVP, which includes an estimation of \( M \), will be discussed.

A. Formulating \( z \)

Let \( z \) be zero-mean and pixel-wise independent (conditional on \( y \)) with mean and variance given respectively by
\[
\mathbb{E}(z_i | y_i) = 0, \quad \text{Var}(z_i | y_i) = f(y_i),
\]
for any pixel \( i \), where \( f \) is a function to be determined.

Remark 4: By (13), the variance of \( z_i \) is a function of the observed value \( y_i \) instead of \( x_i \) and \( n_i \), hence samples of \( z_i \) can be generated without knowing the clean images. Clearly, the above formulation of \( z \) satisfies (A2), and (A3) holds.
for some existing $M$. A typical choice of $z_i$ is the Gaussian random variable $N(0, f(y_i))$.

Our goal is therefore to find a function $f(\cdot)$ such that $M$ is as close to $I$ as possible, as required by (S2) (see Subsection III-D). A necessary and sufficient condition for $M = I$ is given below.

**Proposition 2:** Assume that $z$ satisfies (13) and $M$ satisfies (A3). Then $M = I$ if and only if $f$ satisfies

$$
\mathbb{E}_{y_i}(f(y_i) \mid x_i) = \text{Var}(n_i \mid x_i),
$$

for any pixel $i$.

**Proof:** By (13), the conditional variance of $z_i$ is computed as

$$
\text{Var}(z_i \mid x_i) = \mathbb{E}_{y_i}(\text{Var}(z_i \mid y_i) \mid x_i)

= \mathbb{E}_{y_i}(f(y_i) \mid x_i),
$$
in which $\mathbb{E}_{y_i}$ means the expectation taken over $y_i$. This together with (A3) leads to a formulation of the diagonal entries of $M$:

$$
\mathbb{E}_{y_i}(f(y_i) \mid x_i) = M_{ii}\text{Var}(n_i \mid x_i), \quad i = 1, \ldots, m.
$$

Therefore the desired result follows from the fact that $M$ is a diagonal matrix. \hfill \Box

It is worth noting that the function $f$ satisfying (14) may take different forms depending on the noise distributions. For example, if $n$ is i.i.d. noise, then $f$ is a constant function equal to the noise variance. For Poisson noise with variance $\text{Var}(n_i \mid x_i) = \lambda x_i$ (where $\lambda$ is a Poisson distribution parameter), if we let $f(y) = \gamma y$, then $\mathbb{E}_{y_i}(f(y_i) \mid x_i) = \lambda x_i = \text{Var}(n_i \mid x_i)$.

In this work, we approximate $f$ with a low-dimensional space

$$
\left\{ f = \sum_{k=1}^{K} \beta_k f_k \mid \beta_k \in \mathbb{R} \right\}, \quad (15)
$$

where $\{f_k\}$ are basis functions. The number $K$ is typically small for computation considerations. The parameters $\{\beta_k\}$ are iteratively updated, as a step of (S2).

**B. Minimising Denoising Objectives**

Once $z$ is defined, the objective function $\tilde{L}$ given in (6) can be evaluated. In practice, we minimise its corresponding empirical risk over the given samples of $y$, which can be implemented by standard stochastic gradient descent methods. It is important to note that $\tilde{L}$ alone may not be a good approximation of the MSE loss (see Remark 3). However, one can approximate the MSE if $R$ has a partially linear structure, according to Theorem 1.

The partially linear structure (4) can be imposed following the penalty method proposed in [17]. Specifically, if $\tilde{y}^{(1)}, \tilde{y}^{(2)}, \ldots, \tilde{y}^{(P)}$ are variants of a noisy sample (associated with a single clean image) and they satisfy $\sum_{j=1}^{p} \tau_j \tilde{y}^{(j)} = 0$, then with the decomposition (4),

$$
\sum_{j=1}^{p} \tau_j R(\tilde{y}^{(j)}) = \tau_j e^{(j)},
$$

where $e^{(j)}$ is the residual term associated with $\tilde{y}^{(j)}$. A typical value of $P$ is 3. The above equality does not require explicit formulations of the function $g$ or the linear operator $L$, and hence an explicit decomposition of $R$ is not needed. Based on these observations, one can define a penalty term

$$
\mathcal{L}_c := \mathbb{E}_{y^{(1)}, \ldots, y^{(P)}} \left\{ W \sum_{j=1}^{P} \tau_j R(\tilde{y}^{(j)}) \right\} \quad (A3)
$$

for $\hat{e}$, where $W$ is a diagonal weighting matrix for balancing the loss over the pixels. To conclude, the overall loss function is

$$
\mathcal{L}_{all} = \tilde{L} + \gamma \mathcal{L}_c, \quad (16)
$$

where $\gamma$ is a weighting parameter.

**C. Updating the Variance of $z$**

The variance of $z$ is represented by the parameterised function $f$ in (15). Let the parameters at iteration $l$ be denoted by $\beta_k^{(l)}$, then the variance of $z$ is given by

$$
f^{(l)}(y) = \sum_{k=1}^{K} \beta_k^{(l)} f_k(y).
$$

Here $f_k(y)$ is a vector with entries $f_k(y_1), \ldots, f_k(y_m)$. To update $z$, we compute parameters $\{\beta_k^{(l+1)}\}$.

Recall that in (S2), $z$ is updated in order to meet the criterion $M \approx I$, which means $\text{Var}(n_i \mid x_i) \approx \mathbb{E}_{y_i}(f(y_i) \mid x_i)$ according to the condition (14). Motivated by this, $\{\beta_k\}$ are updated as follows. First, at iteration $l$, the variance $\text{Var}(n_i \mid x_i)$ can be expressed in terms of $f^{(l)}$ and $M$ (according to (A3) and the definition of $f^{(l)}$):

$$
\text{Var}(n \mid x) = M^{-1} \text{Var}(z \mid x) = M^{-1} \mathbb{E}_y(f^{(l)}(y) \mid x).
$$

Second, combining this with (14), we aim to solve

$$
\min_{\beta_1, \ldots, \beta_k} \mathbb{E}_y(f(y) \mid x) - M^{-1} \mathbb{E}_y(f^{(l)}(y) \mid x),
$$

(18)

to obtain new parameters $\{\beta_k^{(l+1)}\}$. To compute $\{\beta_k^{(l+1)}\}$, one needs to estimate the matrix $M$ and the expectations, the details of which are given next.

**C.1) Estimating $M$ from $\delta R$.** In what follows, we will explain in a few steps how $M$ can be estimated from $\delta R$.

First, define $R^\parallel$ as the minimizer of the MSE function:

$$
\tilde{L}^\parallel(R) = \mathbb{E} \left( \| R_n(\hat{y}) - n \|^2 \right).
$$

The function $\tilde{L}^\parallel$ is a special case of $\tilde{L}$ with $M = I$. As $R^\parallel$ is obtained with the correct noise variance, we assume that $\delta R^\parallel$ satisfies the DVP (Cf. Subsection III-C).

Second, let $R^*$ be the minimizer of $\tilde{L}$ associated with $z$ whose variance is $f^{(l)}(y)$ ($M$ is not necessarily equal to $I$ here). With $R^\parallel$ defined above, we decompose $R^*$ into the following form

$$
R^* = R^\parallel + R^\perp, \quad (19)
$$

Let $\delta R^\perp$ and $\delta R^\parallel$ be the deviation of $R^\parallel$ and $R^*$, respectively. So $\delta R^\parallel$ contributes to the non-smooth parts of $\delta R^*$ that violate DVP. Here $\delta R^\perp$ is a crucial component for estimating $M$, as mentioned in Subsection III-C and will be detailed next.
Third, one can show that the desired matrix M is linked to \( R^\perp \) by the following approximate decomposition:

\[
R^\perp(\hat{y}) \approx g^\perp(x) + L^\perp(n + \alpha z),
\]

(20)

where \( g^\perp \) is a function, and

\[
L^\perp = (I - M)(I + \alpha^2 M)^{-1}.
\]

(21)

The details of the derivation of \( L^\perp \) are given in Appendix A. Notice that \( M \) is a diagonal matrix, and so is \( L^\perp \). Consequently, taking the deviation of (20), \( L^\perp \) can be estimated by \( L^\perp \delta y \approx \delta R^\perp \). To estimate \( \delta R^\perp \), one can remove smooth components of \( \delta R \) from \( \delta R^* \). Specific details of the implementation will be given in the experimental section V.

In summary, based on (21), the matrix \( M \) can be computed as

\[
M = (\alpha^2 L^\perp + I)^{-1}(I - L^\perp).
\]

(22)

Clearly, if \( R^\perp = 0 \), then \( L^\perp = 0 \) and \( M = I \).

C.2 Estimating Conditional Expectations: Once \( M \) is computed, we use it to minimise (18). The conditional expectations in (18) are pixel independent location, i.e., for any pixel locations \( i \) and \( j \), \( \mathbb{E}_y(f(y) | x_i) = \mathbb{E}_y(f(y) | x_j) \) if \( x_i = x_j \). This is a consequence of the assumptions (A1) and (A2). Hence, given a sample \( y \) of \( y \) (associated with the clean image \( x \)), the conditional expectations are estimated as

\[
\mathbb{E}_y(h(y) | x_i = x_j) \approx \text{average}(\{h(y) | x_j = x_i\}),
\]

(23)

where \( h \in \{f, f^{(t)}\} \). The right-hand side of the above approximation is an average over different pixel locations of a single sample. To represent this in a compact form, we define a matrix \( A \), the entries of which are

\[
A_{ij} = 0 \quad \text{if} \quad x_i \neq x_j,
\]

\[
A_{ij} = 1/|\{i' | x_i = x_j\}| \quad \text{if} \quad x_i = x_j,
\]

in which \(|\cdot|\) denotes the cardinality of a set. In practice, the clean image \( x \) is unknown, so we approximate \( A \) by replacing \( x \) with its estimated version \( R^*(\hat{y}) \). With the definition of \( A \), we can rewrite the conditional expectation (23) as

\[
\mathbb{E}_y(h(y) | x = x) \approx Ah(y) \quad \text{for} \quad h \in \{f, f^{(t)}\}.
\]

Moreover, recall that \( M \) in (A3) is a diagonal matrix with diagonal entries \( M_{ii} = M_{ii}(x_i) \), i.e., the diagonal entries are also pixel location independent (but depend on the value of \( x_i \)). From the way the matrix \( A \) is defined, we have \( M^{-1} = \text{diag}(AM^{-1}1) \) where \( 1 \) is the vector of all ones. In the computations of (18), \( M^{-1} \) can be replaced with \( \text{diag}(AM^{-1}1) \) which is more robust to the errors in the computed \( M \). So far, we have derived the representations for the expectations- and \( M \)-terms in (18).

Finally, at iteration \( l \), the minimisation of (18) is carried out over a mini-batch of noisy images instead of the entire dataset. This, together with the approximated matrices and expectations, introduces randomness in the computed solutions. A common strategy to reduce the randomness is to let the sequence \( \beta_k^{(t)} \) be an experiential moving average of the computed solutions. With this practical consideration, the new parameters of \( \beta_k \) are computed as:

\[
\{\beta_k^{(t)}\} = \arg\min_{\{\beta_k\}} \left\{ A \sum_k \beta_k f_k(y) - \text{diag}(AM^{-1}1) Af^{(t)}(y) \right\}^2,
\]

(24)

where \( \nu > 0 \) is a constant.

Recalling that \( M \) was computed by (22), the algorithm (24) converges when \( R^* \) satisfies the DVP, as in this case \( R^\perp = 0 \) and therefore, \( L^\perp = 0 \) and the computed \( M \) is identity. If \( M = I \), then \( f^{(t)}(y) = \sum_{k=1}^K \beta_k^{(t)} f_k(y) \) is an unbiased estimate of noise variance \( \text{Var}(n_i | x_i) \), according to (14).

Therefore, we have shown how the denoiser is updated (in Subsection IV-B) and how the new variance of \( z \) is computed (in Subsection IV-C). The overall learning algorithm for simultaneous denoising and noise variance estimation is summarised in Algorithm 1. The outputs of the algorithm are the learnt denoiser and \( f^{(t)}(y) = \sum_{k=1}^K \beta_k^{(t)} f_k(y) \) which estimates \( \text{Var}(n_i | x_i) \) motivated by (14).

V. EXPERIMENTS

Experiments are carried out on benchmark denoising datasets as well as real microscopy images to evaluate the performance of the proposed approach. The rest of this section is structured as follows. First, the network architecture and training specifications, which are shared among all experiments unless specified otherwise, are given (Subsection V-A). Second, the results on benchmark denoising datasets are presented (Subsection V-B). Finally, we demonstrate the performance of the proposed approach in denoising real microscopy images where ground truth images for training are unavailable (Subsection V-C).

Algorithm 1 Learning Framework

Require: parameterised denoiser \( R \), \( \{\beta_k^{(0)}\} \), \( \gamma \). The initial variance of \( z \).

Results: learnt denoiser \( R^* \), the estimated variance of \( n \)

Set \( l = 0 \) and initialise \( R \).

While not converge do

Select randomly a mini-batch of noisy images \( \{y\} \).

For each \( y \), generate \( \hat{y}, \hat{y}^{(1)}, \ldots, \hat{y}^{(P)} \).

(51) update \( R \) (Subsection IV-B)

Carry out a gradient step of \( L \) all given in (16)

(52) update the variance of \( z \) (IV-C)

Compute \( \delta \) and then \( M \) from (22)

Compute \( \sum_{k=1}^K \beta_k^{(l+1)} \) by minimising (24) over the mini-batch

Set \( l \leftarrow l + 1 \)

End

Return the final denoiser \( R^* := R \) and \( f^{(t)}(\cdot) := \sum_{k=1}^K \beta_k^{(l)} f_k(\cdot) \) (for noise variance estimation).

1Source code will be made available at github.com.
A. Network Architectures and Training Specifications

The proposed approach is network-architecture agnostic. The denoiser \( R \) is generally defined as a parameterised operator and hence the algorithm applies to an arbitrarily selected network architecture for denoising. In all experiments, unless specified otherwise, the denoising architecture we use is the benchmark DnCNN [39].

Optimisation: The minimisation of the loss function (16) is implemented using the stochastic optimiser ADAM [19]. Each stochastic optimisation step contains a batch of 128 images, each of which is randomly cropped into the size of 40 × 40 pixels (and divided by 255 if the original scale of pixel values is from 0 to 255). Following the setting of [17], we first minimise \( \hat{L} \) for 2 × 10^3 optimisation steps with a learning rate of 0.001 and \( \alpha = 1 \). This is a pretraining step for reducing the computational cost because loss \( L_c \) is not evaluated at the moment. Then a further 6 × 10^4 optimisation steps are performed on the full loss \( L_{\text{all}} \), with a step decay of learning rate from 0.001 to 0.0001 and 0.000005. The parameter \( \alpha \), in this case, is randomly selected from [0.1, 0.5].

Loss Functions: The input images \( \hat{y} := y + \alpha \varepsilon \) (see Eq. (4)) where the given noisy images \( y \) as well as vectors \( \varepsilon \) drawn from normal distributions \( N(0, f^{(i)}(y)) \). The generated images \( \hat{y} \) are used to compute the loss function \( \hat{L} \).

To compute the loss \( L_c \), (in (16), one needs perturbed samples \( \hat{y}^{(1)} \), \( \hat{y}^{(2)} \) and \( \hat{y}^{(3)} \) of the noisy image \( y \) (setting \( P = 3 \)). Here we let \( \hat{y}^{(1)} = y \). The sample \( \hat{y}^{(2)} \) is equal to \( \hat{y} \) except on a randomly selected subset \( S \) of pixels. For each pixel location \( i \) in \( S \), the value \( \hat{y}^{(2)} \) is set to \( [\hat{y}]_i \) where (a) \( \hat{y}:= y + \alpha \varepsilon \) (\( \varepsilon \) is generated in the same way as \( \varepsilon \) ) and (b) \( \varepsilon \) is randomly selected from the 4-neighbours of \( i \). The permutation \( \theta := \left\{ \hat{y}^{(2)} \right\}_i | \left\{ \hat{y} \right\}_i \) can be large because of the edge structures in the images, so if \( \theta > 3 \sqrt{\text{var}} \) where \( \text{var} \) is the average variance of pixels of \( z \), we replace \( \left\{ \hat{y}^{(2)} \right\} \), \( \hat{y}^{(2)} \) with \( \hat{y}^{(3)} \) such that \( \hat{y}^{(3)} = \hat{y}^{(1)} + (1 - \tau_1) \hat{y}^{(2)} \) where \( \tau_1 \) is drawn from a uniform distribution in \( (0, 1) \). The diagonal entries of the weighting matrix \( W \) are \( W_{ii} = 0 \) if \( i \notin S \) and \( W_{ii} = 1/\left| (R(\hat{y}^{(1)}) - R(\hat{y}^{(2)}))_i \right| + \varepsilon_S \) if \( i \in S \). Here \( \varepsilon_S \) is a factor for preventing division by small numbers, and it is given by \( 0.1g_S \) where \( g_S \) is the mean squared distance between \( \hat{y}^{(1)} \) and \( \hat{y}^{(2)} \) restricted on the subset \( S \).

Updates of the Variance of \( z \): The update of \( \{\beta_1, \beta_2, \ldots, \beta_K\} \) (using (24)) is a crucial step of (22). Throughout the experiments, we set \( K = 2 \), and parameterise \( f \) with polynomials of degree one. The function \( f \) is initialised as a constant function based on a rough estimate of the noise variance; \( \beta_1 = \sigma^2 \) and \( \beta_2 = 0 \). We start updating \( \{\beta\} \) only after 6 × 10^3 optimisation steps of the full loss \( L_{\text{all}} \), as the \( R \) can be unstable at the beginning of the optimisation.

To update \( \{\beta_k\} \) requires first estimating \( \delta R^{\perp} \). For smooth regions of the variation \( \delta R^{\perp} \), we can compute \( \delta R^{\perp} \) by removing the smooth components of \( \delta R \) (i.e., \( \delta R^{\parallel} \)) according to the decomposition (12). The details of obtaining \( \delta R^{\perp} \) are as follows.

First, the variation of the denoiser is computed as \( \delta R = R(\hat{y}^{(2)}) - R(\hat{y}^{(1)}) \) where we have reused the perturbed samples \( \hat{y}^{(2)} \) and \( \hat{y}^{(1)} \). One of the benefits of doing so is computation effort saving because no extra forward passes of network \( R \) are needed (instead, reusing \( R(\hat{y}^{(1)}) \) and \( R(\hat{y}^{(2)}) \) that have been computed in (S1)).

Second, \( \delta R^\perp \) is computed from \( \delta R \). We compute and use only a subset of pixels in \( \delta R^\perp \) where \( \delta R^\parallel \) is smooth (details in the next step). For \( i \) within smooth regions of \( \delta R^\parallel \), \( \delta R^\perp \) is estimated by \( \delta R^\perp \}_i = \delta R}_i - \delta \hat{R}_i \) (where \( \delta \hat{R} \)) denotes the average of the 4-neighbours of \( \delta \hat{R} \).

Third, for the region selection required in the last step, we make use of the 4-neighbours of \( i \) where \( \delta \hat{R}^\parallel \}_j = \delta \hat{R}^\parallel \}_j \) (recalling that by design elements in \( S \) are spatially separated and hence \( \delta \hat{R}^\parallel \}_j = 0 \), and consequently, \( \delta \hat{R}^\perp \}_j \approx \delta \hat{R}^\perp \}_j \), and also \( \delta \hat{R}^\perp \}_l \approx \delta \hat{R}^\perp \}_l \) (here means the averaged 4-neighbours of \( \delta \hat{R}^\perp \}_l \)). A list of pixels is selected based on \( \delta \hat{R}^\perp \), with the following steps: i) we divide the range of pixel values in \( R(\hat{y}^{(1)}) \) into 10 intervals of equal lengths and divide \( S \) into 10 subsets accordingly, based on which intervals the pixels are in. ii) In each subset of \( S \), we remove its first 20% elements with the smallest values of \( \left\| \delta \hat{R} \right\| \) . This aims to avoid the instability of division by a small \( \left\| \delta \hat{R} \right\| \) in a later step. iii) In each of the new subsets, we empirically select the first 3% pixels with the smallest values of \( \left\| \delta \hat{R} \right\| \) . The resulting 10 subsets of \( S \) are used to estimate \( \delta R^\perp \), which is then used to estimate \( L^\perp \), as illustrated in subsection IV-C.

Lastly, with the estimated \( L^\perp \), the variance of \( z \) is updated by solving the least squares problem (24) restricted to the subsets of pixels where \( L^\perp \) is estimated.

Exponential Moving Average and Delayed Updates: The parameters \( \{\beta_k\} \) are updated within an exponential moving average (EMA) scheme (24). The parameter \( \nu \) controls the speed of the update. A larger \( \nu \) takes larger steps to the estimated \( \{\beta_k\} \), while a smaller \( \nu \) might lead to slower convergence but is more robust to the errors in estimation. In our experiments we choose \( \nu = 5 \times 10^{-4} \).

Finally, the size of \( S \) must not be too small in order to have robust solutions for (24). However, the size of \( S \) is restricted by the batch sizes and image sizes. A natural strategy to allow a larger sized \( S \) is to combine the data from multiple iterations. Specifically, we delay \( (S2) \) for a set of consecutive iterations, during which the data \( \hat{y}^{(1)}, \hat{y}^{(2)}, R(\hat{y}^{(1)}) \) and \( R(\hat{y}^{(2)}) \) are stored and merged together. The step \( (S2) \) is performed once in the last iteration, where the stored data is treated as a single batch (hence a larger size of \( S \) ). In our experiment we do one \( (S2) \) update per 5 iterations.

After training, the networks can be directly applied to a noisy image to perform denoising, hence the running time is equivalent to that of a forward pass of DnCNN (as used in our experiment).

B. Denoising and Noise Variance Estimation Results

Following [8] and [39], we use a dataset of 400 natural images for training where each image is of size 180 × 180. The denoising performance is evaluated on two different test
sets, namely, BSD68 (which consists of 68 image) [24] and 12 wildly used test images Set12, following the setting of [17], [39]. We consider various settings of noise (including Gaussian noise and Poisson noise at different levels). In addition to the training details given in subsection V-A, we set the noise and Poisson noise at different levels). We consider various settings of noise (including Gaussian noise with parameters \( \sigma \) = 1, 2, 10 and \( \lambda \) = 60, 30, 15, respectively. The noise variance parameters are initialised by \( \beta = \frac{\sigma^2}{\hat{\sigma}^2} \) with \( \hat{\sigma}^2 = 1/80, 1/40, 1/20 \) for noise parameters \( \lambda = 60, 30, 15 \), respectively. For each noise setting, a denoising model is trained using only the given noisy images, and their performance is reported and compared to other denoising approaches, such as BM3D [10], Noise2Self [3], SURE [32], DPLD [17], and the supervised baseline DnCNN [39]. For fair comparisons, all deep learning based approaches use the same network architectures as DnCNN [39].

1) Gaussian White Noise: We consider several different noise levels with standard deviations of \( \sigma = 5, 10, 25, 50, 100 \) (relative to image pixel values ranging from 0 to 255). For each noise level, the results, measured in both the Peak signal-to-noise ratio (PSNR) are presented in Table I. In comparison with BM3D [10], one of the best performing model-based Gaussian denoisers, our method (DVP) has a higher PSNR (by around 0.2 ~ 0.5 dB) for all noise levels and both test sets. Also, DVP outperforms Noise2Self [3] (by around 1 to 1.5 dB for smaller noise levels and around 0.3 for largest noise levels). We observe that DVP performs on par with the unsupervised method DPLD [17], which indicates that our methods are effective in extracting the noise variance from the data. However, DPLD is trained with the exact noise variance provided. We underline that in this experiment the results of BM3D, and DPLD are obtained with the help of the ground truth noise variance, and they may suffer from degradation in performance if the noise variance information is inaccurate. In contrast, the proposed approach was carried out without such information, hence offering a more robust solution. Compared with NBR2NBR [15] which does not require known noise variance for training, our method has higher PSNR especially for the high noise cases.

Remarkably, DVP reaches a denoising quality similar to that of the supervised baseline DnCNN [39], with a gap of around 0.1 or 0.2 dB. Note that the supervised DnCNN is trained with additional data, in form of noisy images and ground truth pairs. Compared to the supervised learning results with more advanced network architectures (MWDNN [33], DRUNet [40], SCUNet [37] and GLR [22]), a further improvement of up to 0.4 dB (for test set BSD68) or 0.8 dB (for the 12 wildly used test images) is observed.

Our approaches generated noise variance estimates during the learning process (from the noisy training images). The results are summarised in Table II, where the relative error means \( \frac{\sigma^2_{\text{estimated}} - \sigma^2}{\sigma^2} \) for an estimated variance \( \sigma^2_{\text{estimated}} \). Our approach is not specific to Gaussian noise, and in this case, we simply set \( \sigma^2_{\text{estimated}} = \beta_1 \) (where \( \beta_1 \) is the coefficient of the constant term of the computed \( f \) at the last iteration). The performance is compared to that of the other noise variance estimation methods, for each of which the results are obtained by averaging its estimated noise variances over the 400 training images. The table shows relative errors of less than 1% of our approach in all reported noise levels, and they are significantly smaller than those of the other methods. The obtained accurate noise variances are in good accordance with the similar denoising quality between DVP and DPLD [17] given in Table I, where the latter uses ground truth noise variance.

The qualitative results are presented in Figure 2, in which the denoised images by different methods for a standard test image Cameraman under Gaussian noise with \( \sigma = 25 \) are compared. The last three columns are denoised results specific
to the region of the noisy image that is highlighted by the red square (on the top-left of the figure). Our method is able to capture the smooth parts as well as the sharp edges of the images (as shown on the bottom-right of the figure), and it achieves a similar visual quality to DPLD [17]. By comparison, the BM3D method also recovers well the fine details but small artefacts are observed around the objects in the denoised images. The result of the Noise2Self method, in contrast, looks less smooth compared to the others.

2) Poisson Noise: The denoising results are compared in Table III. Similar to the Gaussian noise cases, the proposed method DVP remarkably outperforms Noise2Self. The performance gap between our method and DPLD [17] (which uses ground truth noise parameters) is small for the different values of parameter $\lambda$ and for both test sets (the BSD68 [24] and the 12 wildly used test images). As shown in the table, DVP outperforms DPLD-est, which is DPLD with estimated noise variance provided by an off-the-self noise variance Rakhshanfar et al. [29]. Moreover, the gap between DVP and the supervised baseline is around 0.1 to 0.2 dB in PSNR in all reported cases, which is much smaller than the gap for Noise2Self (from around 1 to 1.5 dB).

Noise distributions here are more complex compared to the Gaussian cases as they are signal dependent. Our method works also for the zero-mean signal-dependent noise, as the noise variance estimation results in Table IV show. In the table, MAE (mean absolute errors) and RMAE (relative mean absolute errors) are defined respectively by

$\sum_{l=0}^{255} [v_{\text{est}} - [v_{\text{GT}}]_l]|$ and $\sum_{l=0}^{255} |v_{\text{est}} - [v_{\text{GT}}]_l| \sum_{l=0}^{255} [v_{\text{GT}}]_l,$

where $[v_{\text{est}}]_l$ is the estimated variance at intensity level $l$ (ranging from 0 to 255), and $[v_{\text{GT}}]_l$ is the associated ground truth variance. As a comparison, we report in the table the results of two other noise variance estimation methods on the same dataset (400 noisy images). Our method DVP has RMAE around 5 times smaller than that of the other two methods for the three different values of $\lambda$. The accurate noise variance estimates are consistent with the good performance of DVP in denoising illustrated in Table III.
Fig. 3. Denoising results for microscopy images N2DH-GOWT1 (top row) and C2DL-MSC (bottom row). The images displayed in the first column are noisy images (i.e., inputs to denoisers). The other columns correspond to denoised results obtained by different methods on a sub-region of the noisy images highlighted by the red square in the first column.

### TABLE IV

| Noise parameters | MAE ($\times 10^3$) | RMAE (%) | MAE ($\times 10^3$) | RMAE (%) | MAE ($\times 10^3$) | RMAE (%) |
|------------------|---------------------|----------|---------------------|----------|---------------------|----------|
|                  | $\lambda = 60$     | $\lambda = 30$ | $\lambda = 15$     |          |                     |          |
| Original         | 1.695 4.238         | 1.317 6.585 | 1.003 10.026       |          |                     |          |
| Poi et al. [12]  | 1.857 4.643         | 0.957 4.787 | 0.647 6.468        |          |                     |          |
| DVP (Ours)       | 0.260 0.650         | 0.149 0.745 | 0.147 1.469        |          |                     |          |

### C. Denoising Real Microscopy Images

By design, our algorithm is suitable for learning denoising from noisy images and without prior knowledge about the noise variance. We apply the algorithm to real microscopy images, where the ground truth images and noise variance are unknown. Following the setting of [17], we consider two fluorescence microscopy datasets, namely, N2DH-GOWT1 (images of GFP transfected GOWT1 mouse stem cells) and C2DL-MSC (images of rat mesenchymal stem cells), each of which is a video composed of a number of noisy frames. The first dataset N2DH-GOWT1 contains 92 frames of the size 1024 $\times$ 1024, while the second one has 48 frames of size 992 $\times$ 832. The denoisers are trained on these noisy frames only. In addition to the training details given in subsection V-A, the weighting parameter $\gamma$ in (16) is set to 1, and the noise variance parameters are initialised by $\beta_1 = 0.01$ and $\beta_2 = 0$.

Examples of denoised results by the proposed approach are given in Figure 3. The results are obtained based on the same noisy frames of the videos that were used for training (as our method can be trained directly on the targeted images that need to be denoised). Samples of the noisy frames are displayed in the first column of the images, where the red squares highlight sub-regions for which the denoised results by different methods are demonstrated (see the last 4 columns). As shown in the last column of Figure 3, our approach recovers well the smooth regions of the images, despite the fact that such regions are not seen by the training process. The denoised results are visually close to that of DPLD [17], but we point out again that the latter requires a separate noise variance estimation step. Furthermore, in the figure, we also present the results of BM3D [10] and Noise2Self [3] as baselines. Our method is able to reveal more visual structure details of the underlying objects compared to the results of BM3D (second column) and Noise2Self (third column).

### VI. Conclusion

In this work, we propose an unsupervised deep learning method for jointly learning denoisers and estimating noise variance from a set of noisy images. Our method is based on deep variation priors, which state that the variation of denoisers with respect to the changes of noise follows some smoothness assumptions. We present deep variation priors as a criterion for high-quality denoisers in a data-driven setting, and such priors are combined with deep neural networks to explain a set of noisy training images, without having any ground truth images or noise variance beforehand. By doing so, our method achieves high-quality denoisers as well as accurate noise variance estimation. We demonstrate the power of the proposed method in denoising real microscopy images, where paired noisy and clean training images do not exist.

The proposed learning model is end-to-end, in contrast to other unsupervised denoising solutions that require a post-processing step or a separate process of noise estimation. In particular, our method leverages the error information of
denoising for better noise variance estimation, which in return improves the quality of denoisers.

**APPENDIX A**

**APPROXIMATING THE MATRIX $M$**

In subsection IV-C, we use the following relationship to estimate $M$:

$$R^\perp(\hat{y}) \approx g^\perp(x) + L^\perp(n + \alpha z),$$

where

$$L^\perp = (I - M)(I + \alpha^2 M)^{-1}. \quad (21)$$

We aim to show that this holds.

For simplicity, we assume that the condition of Theorem 1 holds. The desired formulation can be derived in the following steps.

(i). Denote by $C$ the set of denoisers satisfying (4) with $\hat{e} = 0$. For any $R^0 \in C$, rewrite the loss function (7) as

$$\hat{L}(R^0) = \mathbb{E}\left(\left\| R^1(\hat{y}) - x + (R^{0\perp}(\hat{y}) - (I - M)n) \right\|^2\right) + c,$$

$$= \mathbb{E}\left(\| p \|^2 \right) + 2\mathbb{E}(\langle p, q \rangle) + \mathbb{E}\left(\| q \|^2\right) + c,$$

where $R^{0\perp} := R^0 - R^1$, $p := R^1(\hat{y}) - x$ and $q := R^{0\perp}(\hat{y}) - (I - M)n$.

(ii). Next we show that the cross term $\mathbb{E}(\langle p, q \rangle)$ is independent of $R^{0\perp}$. Since $R^1, R^0 \in C$, one has $R^{0\perp} \in C$. Moreover, by definition, $R^1$ is the minimiser of $\mathbb{E}\left(\left\| R^1(\hat{y}) - x \right\|^2\right)$ over all $R^1 \in C$. The optimality of $R^1$ implies that

$$\mathbb{E}\left(\langle R^1(\hat{y}) - x, R^{0\perp}(\hat{y}) \rangle\right) = 0,$$

for any $R^{0\perp} \in C$. Therefore, $\mathbb{E}(\langle p, q \rangle) = \mathbb{E}(\langle p, (I - M)n \rangle)$ and it does not depend on $R^{0\perp}$.

(iii). By Equation (25) and the observation in (ii), the minimisation of $\hat{L}(R^0)$ is equivalent to

$$\min_{R^0 \in C} \hat{L}(R^0) = \mathbb{E}\left(\| p \|^2 \right) + 2\mathbb{E}(\langle p, q \rangle) + \min_{R^{0\perp} \in C} \mathbb{E}\left(\| q \|^2\right) + c.$$

Therefore, writing the minimiser of $\hat{L}$ as $R^* = R^\perp + R^\perp$ (Cf. Equation (19)), the term $R^\perp$ should satisfy

$$\mathbb{E}\left(\left\| R^\perp(\hat{y}) - (I - M)n \right\|^2\right) = \min_{R^{0\perp} \in C} \mathbb{E}\left(\| q \|^2\right). \quad (26)$$

(iv). Since $R^\perp \in C$, it can be written in the following form

$$R^\perp(\hat{y}) = g^\perp(x) + L^\perp(n + \alpha z).$$

Here $L^\perp$ depends on $x$. Plugging this into (26), $L^\perp$ approximately minimises

$$\mathbb{E}\left(\left\| g^\perp(x) + L^\perp(n + \alpha z) - (I - M)n \right\|^2\right) = \min_{R^{0\perp} \in C} \mathbb{E}\left(\| q \|^2\right).$$

Therefore,

$$L^\perp \approx L^\perp = (I - M)\text{Cov}(n|x) + \text{Cov}(n|x) + \alpha^2 \text{Cov}(z|x)^{-1},$$

which gives the desired statement.

**APPENDIX B**

**NOISE WITH NON-ZERO MEAN**

In this work, we focus on the denoising problems where the noise is zero-mean. That is, in Equation (1), the expectation of $n$ is zero conditioned on $x$. Next, we will discuss what happens when the mean of the underlying noise is non-zero. Define the following decomposition of the noise:

$$n = n_b + n_z,$$

where $n_b := \mathbb{E}(n|x)$ is the mean of the noise and $n_z$ has zero-mean.

Similar to Theorem 1, we have the following expression of $\hat{L}(R)$.

**Proposition 3:** Let $R$ be a denoiser satisfying (4) with $\hat{e} = 0$ and assume that $n$ satisfies (27), then $\hat{L}$ (defined in (6)) satisfies

$$\hat{L}(R) = \mathbb{E}\left(\| R_n(\hat{y}) - M_{n_z} \|^2\right) + c,$$

$$= \mathbb{E}\left(\| x + n_b - R(\hat{y}) + (n_z - M_{n_z}) \|^2\right) + c,$$

(28)

where $R_n(\hat{y}) := y - R(\hat{y})$, $c$ is a constant independent of $R$, and the expectation is taken over $z, x$ and $n$.

**Proof.** By the definitions of $\hat{L}$ and $R_n(\hat{y})$,

$$\hat{L}(R) = \mathbb{E}\left(\| R_n(\hat{y}) - z/\alpha \|^2\right)$$

$$= \mathbb{E}\left(\| R_n(\hat{y}) - M_{n_z} + (M_{n_z} - z/\alpha) \|^2\right). \quad (29)$$

According to (4), the first term on the right hand side of (29) can be rewritten as

$$R_n(\hat{y}) - M_{n_z} = x - g(x) - L_n + (n - M_{n_z})$$

$$= x - g(x) - L_n + n_b + (n_z - M_{n_z}).$$

Notice that $\langle n_b, (n_z - M_{n_z}) \rangle = 0$, so similar to the proof of Theorem 1, we can show that the cross term $\mathbb{E}(\langle R_n(\hat{y}) - M_{n_z}, M_{n_z} - z/\alpha \rangle)$ of (29) is equal to the constant $c_1 := \mathbb{E}(\langle n_z - M_{n_z}, M_{n_z} - z/\alpha \rangle$. Let $c_2 := \mathbb{E}(\| M_{n_z} - z/\alpha \|^2)$, and $c := c_1 + c_2$, then we have

$$\hat{L}(R) = \mathbb{E}\left(\| R_n(\hat{y}) - M_{n_z} \|^2\right) + c$$

$$= \mathbb{E}\left(\| x + n_b - R(\hat{y}) + (n_z - M_{n_z}) \|^2\right) + c, \quad (30)$$

as desired. \qed

The results of Proposition 3 can be interpreted from the following two perspectives.

First, the first equality $\hat{L}(R) = \mathbb{E}\left(\| R_n(\hat{y}) - M_{n_z} \|^2\right) + c$ looks similar to the Equation (7) in Theorem 1. The difference is that now $n$ in the squared term is replaced by $n_z$. This suggests that we can apply the same noise modelling framework discussed in sections (III) and (IV) to estimate the variance of $n_z$, which is equal to the variance of $n$, despite the fact that of the noise is not zero-mean.

Second, the result (28) also suggests that the bias of the noise $n_b$ may be contained in the output of the denoiser when minimising $\hat{L}$. In fact, in an ideal case where $M = I$, i.e., $\text{Cov}(z|x) = \text{Cov}(n|x) = \text{Cov}(n_z|x)$, we have

$$\hat{L}(R) = \mathbb{E}\left(\| x + n_b - R(\hat{y}) \|^2\right) + c,$$

(31)

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
which means that $R(\hat{y})$ would be fitted to $x + n_b$ instead of $x$. This results in the effect of partial denoising: the denoiser is trained to remove the zero-mean component $n_z$ of the noise only. On the other hand, this suggests that the scheme (where the loss $\hat{L}$ is minimised) still meaningfully removes part of the noise, when the zero-mean assumption is violated.

The biasedness of the noise can lead to difficulties in both denoising and noise variance estimation. Treatment of biasedness has been investigated in the literature. For example, in the setting of Poissonian-Gaussian noise, Gaussianisation techniques [12] is discussed when the noisy signal is biased.

**APPENDIX C**

**DENOISING RESULTS FOR POISSON NOISE**

This section contains the visual results of the denoising algorithms discussed in Subsection V-B.2. Figure 4 displays the results of Noise2Self [3], our approach (DVP) as well as the supervised method, where the images are taken from the set of 12 wildly used images.

As shown in the figure, DVP is able to recover the smooth regions (Cf. the image on the first row of the figure) as well as part of the textures (Cf. the image on the second row), and the visual quality is similar to that of the supervised method, which is consistent with the quantitative results reported in III (2). In comparison, the results of the Noise2Self method look less smooth.

**ACKNOWLEDGMENT**

The experiments in this work were carried out with support from the computational and data storage facilities of the Advanced Computing Research Centre, University of Bristol.

**REFERENCES**

[1] F. J. Anscombe, “The transformation of Poisson, binomial and negative-binomial data,” *Biometrika*, vol. 35, nos. 3–4, pp. 246–254, Dec. 1948.

[2] S. G. Bahnemiri, K. Egiazarian, and M. Ponomarenko, “Learning-based noise component map estimation for image denoising,” *IEEE Signal Process. Lett.*, vol. 29, pp. 1407–1411, 2022.

[3] J. Batson and L. Royer, “Noise2Self: Blind denoising by self-supervision,” in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 524–533.

[4] T. Bonesky, “Morozov’s discrepancy principle and Tikhonov-type functionals,” *Inverse Problems*, vol. 25, no. 1, Jan. 2009, Art. no. 015015.

[5] A. Buades, B. Coll, and J.-M. Morel, “A non-local algorithm for image denoising,” in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 2, Jun. 2005, pp. 60–65.

[6] J. Byun, S. Cha, and T. Moon, “FBI-denoiser: Fast blind image denoiser for Poisson–Gaussian noise,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 5764–5773.

[7] J. Chen, J. Chen, H. Chao, and M. Yang, “Image blind denoising with generative adversarial network based noise modeling,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3155–3164.

[8] Y. Chen and T. Pock, “Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1256–1272, Jun. 2016.

[9] R. R. Coifman and D. L. Donoho, “Translation-invariant de-noising,” in *Wavelets and Statistics*. New York, NY, USA: Springer, 1995, pp. 125–150.

[10] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, “Image denoising by sparse 3-D transform-domain collaborative filtering,” *IEEE Trans. Image Process.*, vol. 16, no. 8, pp. 2080–2095, Aug. 2007.

[11] D. L. Donoho and J. M. Johnstone, “Ideal spatial adaptation by wavelet shrinkage,” *Biometrika*, vol. 81, no. 3, pp. 425–455, Sep. 1994.

[12] A. Foi, M. Trimeche, V. Katkovnik, and K. Egiazarian, “Practical Poissonian–Gaussian noise modeling and fitting for single-image raw-data,” *IEEE Trans. Image Process.*, vol. 17, no. 10, pp. 1737–1754, Oct. 2008.

[13] P. C. Hansen and D. P. O’Leary, “The use of the L-curve in the regularization of discrete ill-posed problems,” *SIAM J. Sci. Comput.*, vol. 14, no. 6, pp. 1487–1503, 1993, doi: 10.1137/0914086.

[14] C. Huang, M. K. Ng, T. Wu, and T. Zeng, “Quaternion-based dictionary learning and saturation-value total variation regularization for color image restoration,” *IEEE Trans. Multimedia*, vol. 24, pp. 3769–3781, 2022.
[15] T. Huang, S. Li, X. Jia, H. Lu, and J. Liu, “Neighbor2Neighbor: Self-supervised denoising from single noisy images,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2021, pp. 14781–14790.

[16] J. Immerkær, “Fast noise variance estimation,” Comput. Vis. Image Understand., vol. 64, no. 2, pp. 300–302, Sep. 1996.

[17] R. Ke and C.-B. Schönlieb, “Unsupervised image restoration using partially linear denoisers,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 44, no. 9, pp. 5796–5812, Sep. 2022.

[18] D.-W. Kim, J. R. Chung, and S.-W. Jung, “GRDN: Grouped residual denoiser,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW), Jun. 2019, pp. 2086–2094.

[19] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” 2014, arXiv:1412.6980.

[20] A. Krull, T. Vicar, and F. Jug, “Probabilistic Noise2Void: Unsupervised content-aware denoising,” 2019, arXiv:1906.00651.

[21] J. Lehtinen et al., “Noise2Noise: Learning image restoration without clean data,” in Proc. Int. Conf. Mach. Learn., Jun. 2018, pp. 2965–2974.

[22] Y. Li et al., “Efficient and explicit modelling of image hierarchies for image restoration,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2023, pp. 18278–18289.

[23] L. I. Rudin, S. Osher, and E. Fatemi, “Nonlinear total variation based noise removal algorithms,” Phys. D, Nonlinear Phenomena, vol. 60, nos. 1–4, pp. 259–268, 1992.

[24] D. Martin, C. Fowlkes, D. Tal, and J. Malik, “A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics,” in Proc. 8th IEEE Int. Conf. Comput. Vis., Feb. 2001, pp. 416–423.

[25] M. Lindenbaum, M. Fischer, and A. Bruckstein, “On Gabor’s contribution to image enhancement,” Pattern Recognit., vol. 27, no. 1, pp. 1–8, 1994.

[26] P. Perona and J. Malik, “Scale-space and edge detection using anisotropic diffusion,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 12, no. 7, pp. 629–639, Jul. 1990.

[27] V. A. Pimpalkhute, R. Page, A. Kothari, K. M. Blurchand, and V. M. Kamble, “Digital image noise estimation using DWT coefficients,” IEEE Trans. Image Process., vol. 30, pp. 1962–1972, 2021.