SIAM J. Imaging Sciences
Vol. 18, No. 4, pp. 2580–2604

© 2025 Society for Industrial and Applied Mathematics

LURE: An Unsupervised Denoising Framework for Multiplicative Lognormal Noise*

Monalisa Bakshi†, Gayathri Venkat†, Nikhil Bisen†,
Chandra Sekhar Seelamantula‡, and Thierry Blu§

Abstract. Most image denoising problems focus on additive white Gaussian noise. In real-world imaging scenarios such as ultrasound, synthetic aperture radar, and optical-coherence tomography, the noise is multiplicative. Multiplicative noise has an extreme degradation effect compared to additive noise of the same variance. Further, in a practical imaging setting, one does not have access to the ground-truth clean images to train a deep neural network in a supervised fashion for image denoising. In this paper, we propose an unsupervised image denoising method for multiplicative noise. Specifically, we consider lognormal noise and develop an unbiased risk estimator of the mean-square error (MSE). We show that the resulting lognormal noise unbiased risk estimate, which we abbreviate as LURE, is an accurate estimator of the Oracle MSE. Computation of LURE involves the weighted trace of the Jacobian, which we estimate using a stochastic/Monte Carlo approximation method that is not only fast but also results in an accurate estimate of the MSE. The framework is flexible enough to accommodate a wide spectrum of denoisers—from wavelet denoising techniques to state-of-the-art deep learning techniques, subject to certain smoothness conditions on the denoiser. We deploy modern deep learning models such as U-Net, dilated-residual U-Net (DRUNet), and gradient step denoiser with DRUNet (GS-DRUNet) to establish the reliability of LURE. The performance measures used are peak signal-to-noise ratio (PSNR) and structural similarity index metric (SSIM). We show that minimizing Monte Carlo LURE in an unsupervised setting gives results that are on par with and sometimes even better than those obtained using the Oracle MSE loss in the supervised setting. We also provide comparisons with unsupervised despeckling techniques such as SAR2SAR, SAR-CNN, and Speckle2Void on real-world noisy images.

Key words. lognormal distribution, multiplicative noise, unbiased risk estimate, image denoising, LURE

MSC codes. 94A08, 68U10, 62M40, 62M45, 62H35

DOI. 10.1137/24M1700508

Downloaded 01/23/26 to 140.112.41.219 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

*Received by the editors October 7, 2024; accepted for publication (in revised form) August 6, 2025; published electronically December 2, 2025. The first, second, and third authors contributed equally to this work.

https://doi.org/10.1137/24M1700508

Funding: The fourth author was supported by ZEISS India and the Anusandhan National Research Foundation under SUPRA scheme (SPR/2023/000041). The fifth author was supported by the Yushan Fellow Program (MOE, Taiwan). In addition, the student authors received fellowship support as follows: Monalisa Bakshi — Wells Fargo International Private Limited; Gayathri Venkat — Axis Bank Centre for Mathematics and Computing, Indian Institute of Science; and Nikhil Bisen — Ministry of Education, Government of India.

†Spectrum Lab, Department of Electrical Engineering, Indian Institute of Science (IISc), Bengaluru, Karnataka, 560012 India (monalisab@iisc.ac.in, gayathrivenk@iisc.ac.in, nikhilbisen@iisc.ac.in).

‡Corresponding author. Department of Electrical Engineering, Indian Institute of Science (IISc), Bengaluru, Karnataka, 560012 India (css@iisc.ac.in).

§Department of Electrical Engineering, National Taiwan University, and Department of Electronic Engineering, Chinese University of Hong Kong, Hong Kong (thierry.blu@m4x.org).

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/23/26 to 140.112.41.219 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

1. Introduction.

In practical applications, it is often challenging to capture high-quality images due to background noise picked up by the acquisition device. Whether in medical diagnostics such as optical-coherence tomography (OCT) or in surveillance applications such as remote sensing, noise is ubiquitous, inherent to measurement, and poses a formidable challenge for downstream image processing and classification tasks. At its core, image denoising aims to restore clarity and quality of the acquired images, bridging the gap between the measurements and the ground truth. The noise profiles can be diverse—from additive to multiplicative, with a distribution ranging from Gaussian in the additive case, to more complicated distributions such as the lognormal or gamma distribution in the multiplicative case. Images acquired through coherent imaging techniques such as synthetic aperture radar (SAR) [1, 16, 17, 18], ultrasound imaging [35], laser imaging [36], and OCT [30] are often corrupted by noise that multiplies, rather than adds to, the signal, which is the underlying reflectance function. This type of noise is also referred to as "speckle," and therefore the corresponding denoising process is also referred to as "despeckling" in the literature. Multiplicative noise has a far more deteriorating effect on image quality than additive noise of the same variance, which makes the despeckling/denoising problem challenging. In stellar speckle interferometry, lognormal statistics have been shown to provide a better fit for wavefront perturbations than Gaussian models [2]. Local speckle contrast in laser-based imaging has been modeled as a lognormal variable, influenced by acquisition conditions and neighborhood size [14]. Beyond imaging, RT-PCR testing exhibits multiplicative lognormal noise, impacting sparse recovery methods like LASSO [12]. Given its relevance in some real-world applications, effective denoising techniques are essential for improving signal reconstruction and inference in the case of multiplicative lognormal noise.

Machine learning techniques for image denoising broadly fall into two categories—unsupervised and supervised methodologies, each presenting certain advantages and unique challenges. Supervised learning techniques typically perform better than their unsupervised counterparts, but require the ground-truth clean image, which may not always be available in practice. A review of deep learning techniques for image denoising can be found in [15, 19, 32]. Convolutional neural networks (CNNs) have been employed for despeckling by estimating the speckle from noisy images [5, 33]. Recent advancements in self-supervised learning methods for SAR image despeckling include techniques like the blind-spot CNNs and the self-supervised MERLIN approach [9]. These methods leverage noisy images alone for training, thus avoiding the need for clean reference images and achieving a performance comparable to supervised methods [9, 10, 25]. In the context of SAR image reconstruction, a Gibbs-sampling-based approach has been developed for speckle reduction and uncertainty quantification from phase history data [7]. Recent OCT image despeckling methods have employed CNNs and transformers for combining both local and global feature extraction [39]. These techniques do not come with any guarantees on the denoising performance.

In this paper, we address the problem of multiplicative noise suppression within an unsupervised learning framework without requiring access to the ground-truth images, with a performance that is competitive with the oracle case where ground-truth images are available.

1.1. Notation.

$\mathcal{R}^m$ denotes the vector space of m-dimensional column vectors with real entries. Scalar random variables are denoted using uppercase letters (e.g., X). Random vectors are denoted using uppercase letters in boldface (e.g., $\mathbf{X}$). The vector $\mathbf{Y} \in \mathcal{R}^m$ has

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/23/26 to 140.112.41.219 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

entries denoted by $\mathbf{Y} = [Y_1, Y_2, \ldots, Y_m]^\top$. Matrices are denoted in uppercase, in italics, and in boldface (e.g., the Jacobian matrix is denoted by $\boldsymbol{J}$). Images are considered as column vectors obtained from 2-D arrays by vectorization, which is sometimes also referred to as rasterization or flattening in the literature. A random variable N following the lognormal distribution with parameters μ and σ^2 is denoted by $N \sim \mathcal{LN}(\mu, \sigma^2)$. The denoiser, denoted by $\mathbf{f} : \mathcal{R}^m \to \mathcal{R}^m$, is given by $\mathbf{f}(\mathbf{Y}) = [f_1(\mathbf{Y}), f_2(\mathbf{Y}), \ldots, f_m(\mathbf{Y})]^\top$, which indicates that every pixel of the denoised output depends on the entire noisy image $\mathbf{Y}$. The expectation operator is denoted by $\mathbb{E}$ with a subscript denoting the random variable with respect to which the expectation is evaluated. For instance, $\mathbb{E}_N$ denotes expectation with respect to the random variable N.

1.2. Multiplicative noise model.

Consider an m-dimensional image $\mathbf{X} \in \mathcal{R}^m$, corrupted by noise $\mathbf{N} \in \mathcal{R}^m$, resulting in the output $\mathbf{Y} \in \mathcal{R}^m$, where the three quantities are related as

$$\mathbf{Y} = \mathbf{X} \odot \mathbf{N}, \tag{1.1}$$

where $\odot$ denotes elementwise multiplication. The elements of the noise vector $\mathbf{N}$ are assumed to be independent and identically distributed (i.i.d.) and following the lognormal distribution $\mathcal{LN}(\mu, \sigma^2)$ defined as

$$p_N(n) = \begin{cases} \dfrac{1}{n\sigma\sqrt{2\pi}} \exp\left(-\dfrac{(\ln n - \mu)^2}{2\sigma^2} \right), & n > 0, \\ 0, & n \le 0. \end{cases} \tag{1.2}$$

Unlike the frequently considered additive noise model, which is usually assumed to be stationary, the multiplicative counterpart is not stationary. This can be verified by considering the first-order and second-order statistics of the entries of $\mathbf{Y}$. Let Y_i, X_i, and N_i denote the ith entries of $\mathbf{Y}, \mathbf{X}$, and $\mathbf{N}$, respectively. Then, from (1.1), $Y_i = X_i\, N_i$, and considering that $\{N_i\}$ are i.i.d., the statistics of Y_i vary with the pixel index i because of the dependence on X_i, and hence the multiplicative noise model is not stationary and consequently more challenging.

The distribution of the lognormal multiplicative noise is specified by two parameters: μ and σ^2. We argue that the parameter μ is inconsequential to the denoising process and that only σ^2 is relevant. The irrelevance of μ can be established by considering the noisy image observation model $\mathbf{Y} = \mathbf{X} \odot \mathbf{N}$, which is indistinguishable from the model $\mathbf{Y} = e^\mu\, \mathbf{X} \odot \tilde{\mathbf{N}}$, where the image $\mathbf{X}$ is rescaled by the scalar e^μ and the entries of $\tilde{\mathbf{N}}$ follow a lognormal distribution with parameters 0 and σ^2. For image restoration or denoising problems, image recovery up to a global constant scale factor is acceptable. Also, for the purpose of displaying images, grayscale intensities are normalized to the range $[0, 255]$, which also takes care of the scale factor. Henceforth, without loss of generality, we assume that $\mu = 0$.

The objective is to process $\mathbf{Y}$ through a denoising operator $\mathbf{f} : \mathcal{R}^m \to \mathcal{R}^m$ such that the mean-square error (MSE) between the estimate $\hat{\mathbf{X}} = \mathbf{f}(\mathbf{Y})$ and the ground- truth $\mathbf{X}$, defined as follows, is minimized, i.e.,

$$\min_{\mathbf{f}} \mathbb{E}_{\mathbf{N}} \left\{ \|\mathbf{X} - \mathbf{f}(\mathbf{Y})\|^2 \right\}, \tag{1.3}$$

where $\mathbb{E}_{\mathbf{N}}$ denotes expectation with respect to the noise vector $\mathbf{N}$. The denoiser could be a simple linear filter or a sophisticated state-of-the-art deep neural network.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/23/26 to 140.112.41.219 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Minimizing the MSE also has the direct consequence of maximizing the peak signal-to-noise ratio (PSNR), which is the objective performance metric in image denoising applications, defined as follows:

$$\mathrm{PSNR} = 10\log_{10}\left(\frac{255^2}{\|\mathbf{X}-\hat{\mathbf{X}}\|^2}\right)\ \mathrm{dB}. \tag{1.4}$$

The other performance metric that is used for subjective quality assessment is the structural similarity index metric (SSIM) [34].

Developing the squares in the expression for the MSE, we get

$$\begin{aligned}\mathbb{E}_{\mathbf{N}}\left\{\|\mathbf{X}-\mathbf{f}(\mathbf{Y})\|^2\right\} &= \mathbb{E}_{\mathbf{N}}\{\|\mathbf{X}\|^2 + \|\mathbf{f}(\mathbf{Y})\|^2 - 2\mathbf{X}^\top\mathbf{f}(\mathbf{Y})\}\\ &= \|\mathbf{X}\|^2 + \mathbb{E}_{\mathbf{N}}\{\|\mathbf{f}(\mathbf{Y})\|^2\} - 2\,\mathbb{E}_{\mathbf{N}}\{\mathbf{X}^\top\mathbf{f}(\mathbf{Y})\}.\end{aligned} \tag{1.5}$$

In practice, one has access only to the noisy image $\mathbf{Y}$, and therefore exact computation of the MSE is not possible, and hence it must be estimated. The first term $\|\mathbf{X}\|^2$ does not affect the minimization in (1.3) with respect to the denoising function $\mathbf{f}$. The second term $\mathbb{E}_{\mathbf{N}}\{\|\mathbf{f}(\mathbf{Y})\|^2\}$ depends only on the noisy observation $\mathbf{Y}$ and the denoising function $\mathbf{f}$, and therefore it can be estimated in an unbiased fashion by computing $\|\mathbf{f}(\mathbf{Y})\|^2$. The critical hurdle in proceeding further is the cross-term $\mathbb{E}_{\mathbf{N}}\{\mathbf{X}^\top\mathbf{f}(\mathbf{Y})\}$, which cannot be directly computed because the ground-truth $\mathbf{X}$ is not available. However, it may be possible to find another function $\mathbf{g}(\mathbf{Y})$ such that $\mathbb{E}_{\mathbf{N}}\{\mathbf{X}^\top\mathbf{f}(\mathbf{Y})\} = \mathbb{E}_{\mathbf{N}}\{\mathbf{g}(\mathbf{Y})\}$. This is the principle behind an *unbiased risk estimator (URE).* The problem has been solved by developing Stein's unbiased risk estimator (SURE) in the additive Gaussian noise case [4] encountered in imaging in the presence of thermal and electronic noise; Poisson URE (PURE) in the case of Poisson noise encountered in low-light imaging using photon-counting devices [27]; and chi-square URE (CURE) in the case of chi-square noise encountered in magnetic resonance imaging (MRI) [37], etc. While SURE, PURE, and CURE have originally been employed in a wavelet-based denoising framework, recently, CNNs have also been trained in an unsupervised fashion using SURE as a surrogate to the Oracle MSE [24].

In this paper, we seek to develop a lognormal unbiased risk estimator (LURE) and employ it in an unsupervised setting to denoise multiplicative noise images.

1.3. Motivation for LURE.

Since the noise is multiplicative lognormal, a plausible approach is to convert it to additive noise by considering the logarithm of the noisy image, denoise the image using an off-the-shelf additive Gaussian noise suppression technique, and then exponentiate the result to obtain the denoised image. We show that such an approach yields suboptimal denoising performance. Table 1 shows the performance comparison of a U-Net trained to suppress additive Gaussian noise and the results obtained using U-Net trained

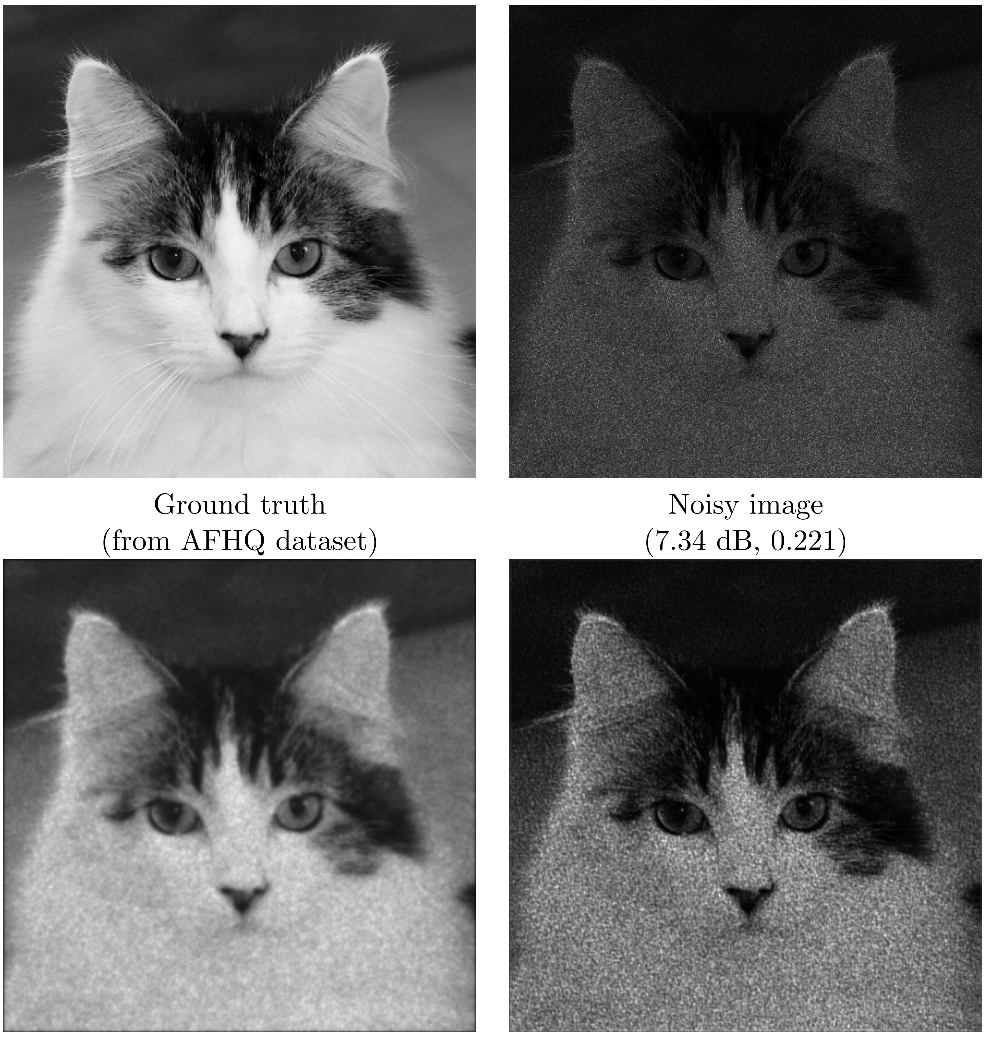
Table 1

Multiplicative image denoising performance by converting it to additive noise: Comparison of PSNR (dB) and SSIM (average values ± standard deviation) obtained using the U-Net by minimizing the Oracle MSE (additive noise model) and MC-LURE loss on the test set from the AFHQ dataset.

σ	**Input**		**MC-LURE**		**Oracle MSE**	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
0.3	9.73±2.14	0.274±0.07	21.53±2.82	0.638±0.072	13.4±2.55	0.419±0.082

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/23/26 to 140.112.41.219 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy



Ground truth
(from AFHQ dataset)

Noisy image
(7.34 dB, 0.221)

U-Net trained with MC-LURE
(22.47 dB, 0.647)

U-Net trained for additive noise
(12.09 dB, 0.333)

Figure 1. *A comparison of the denoising performance of U-Net trained with MC-LURE and U-Net trained with MSE for denoising images in additive Gaussian noise. In the latter case, log-transformed noisy images are denoised and the result is exponentiated. The numbers show (PSNR in dB, SSIM).*

with the Monte Carlo LURE (MC-LURE) developed in this paper (section 3) as the loss on the entire AFHQ dataset. The results show that MC-LURE produces significantly better results, with a PSNR improvement of 8 dB and SSIM improvement of 0.2 compared to additive noise suppression by considering a combination of logarithmic and exponential transformations. Figure 1 shows a subjective comparison of the denoising performance in both cases. Both objective and subjective results suggest that it is advisable to redesign and optimize the denoisers for the multiplicative noise scenario instead of converting the noise to additive and using an off-the-shelf additive noise suppression technique.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/23/26 to 140.112.41.219 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

1.4. Contributions.

The key objective in this paper is to develop an unsupervised learning framework for denoising images corrupted with multiplicative lognormal distributed noise. The ideal objective function to be optimized for maximizing the PSNR is the MSE. However, computing the MSE is impractical in scenarios where the ground truth is not available, which is precisely the scenario considered in this paper. To circumvent this problem, we derive the lognormal unbiased risk estimator (LURE), which can be computed solely based on the observations. We show that LURE closely follows the MSE. Consequently, optimizing LURE is a viable practical alternative to optimizing the MSE. LURE requires knowledge of the σ^2 parameter of the lognormal density, for which we propose a robust median-based estimator. Our derivations show that LURE requires access to the Jacobian of the denoiser, more precisely the weighted trace of the Jacobian, which must be computed as many times as the number of pixels in the image for every update of the parameters of the denoiser. This requires huge computation rendering the scheme impractical. We provide a stochastic alternative based on Monte Carlo sampling for estimating the weighted trace of the Jacobian. The Monte Carlo version of LURE, referred to as MC-LURE, is vastly superior to direct computation of LURE in the sense that it is practical and requires far fewer computations. For denoising, we consider several state-of-the-art denoising neural networks and optimize/train them using LURE as the loss function. Performance comparisons show that MC-LURE is on par with or even better than Oracle MSE at several noise levels, indicating its superior generalization capabilities. Effectively, the proposed unsupervised learning framework is a robust and accurate substitute to its supervised counterpart. Comparisons with state-of-the-art unsupervised denoising methods show that MC-LURE gives superior performance in a majority of the cases.

1.5. Outline of the paper.

Section 2 contains the derivation of LURE for the scalar and multivariate cases. Section 3 presents a computationally efficient strategy for LURE by means of Monte Carlo estimates of the weighted trace of the Jacobian of the denoiser. Further, we also present a robust approach for estimating the lognormal noise parameter σ. Section 4 presents details of the neural network training and results of experimental validation. Section 5 concludes the paper.

2. Lognormal unbiased risk estimator (LURE).

We first present the development of LURE in the scalar case, which is instructive and insightful, and then proceed with the development of LURE in the multivariate case.

Theorem 2.1 (scalar LURE). *Let $Y = XN$, where X is unknown, and let N be a random variable following the lognormal distribution $\mathcal{LN}(0,\sigma^2)$. Then*

$$\texttt{LURE}(f) \overset{\text{def.}}{=} e^{-2\sigma^2}Y^2 + f(Y)^2 - 2Y\mathcal{M}f(Y), \tag{2.1}$$

where the operator $\mathcal{M}$ is defined as follows:

$$\mathcal{M}f(Y) = e^{-\frac{\sigma^2}{2}} f(e^{-\sigma^2}Y) \tag{2.2}$$

is an unbiased estimator of the MSE, i.e., $\mathbb{E}_N\{\texttt{LURE}(f)\} = \mathbb{E}_N\{(X - f(Y))^2\}$, for any f with finite first- and second-order statistics.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Proof. Developing the squares in the expression for MSE gives

$$\mathbb{E}_N\{(X-f(Y))^2\}=\mathbb{E}_N\{X^2\}+\mathbb{E}_N\{f(Y)^2\}-2\mathbb{E}_N\{Xf(Y)\}.$$

The lognormal distribution defined in (1.2) has mean $\mathbb{E}_N\{N\} = e^{\frac{\sigma^2}{2}}$, second moment $\mathbb{E}_N\{N^2\}=e^{2\sigma^2}$, and variance $\mathrm{Var}\{N\}=e^{\sigma^2}[e^{\sigma^2}-1]$. Hence, $\mathbb{E}_N\{Y^2\}=\mathbb{E}_N\{X^2N^2\}\Rightarrow X^2=e^{-2\sigma^2}\mathbb{E}_N\{Y^2\}$.

The cross-term can be shown to be

$$\mathbb{E}_N\{Xf(Y)\}\overset{\text{def.}}{=}\int_{0^+}^{\infty}X\,f(Y)\,p_N(n)\,\mathrm{d}n=e^{\frac{3\sigma^2}{2}}\int_{0^+}^{\infty}X\,f(Xn)\,n\,p_N(ne^{\sigma^2})\,\mathrm{d}n,$$

by virtue of the property $p_N(n)=e^{\frac{3\sigma^2}{2}}n\,p_N(n\,e^{\sigma^2})$. By a change of variables $t=ne^{\sigma^2}$, we get

$$\mathbb{E}_N\{Xf(Y)\}=e^{-\frac{\sigma^2}{2}}\int_{0^+}^{\infty}Xt\,f(Xte^{-\sigma^2})\,p_N(t)\mathrm{d}t=\mathbb{E}_N\{Y\mathcal{M}f(Y)\},\tag{2.3}$$

where $\mathcal{M}f(Y)$ is as defined in (2.2). Hence, it follows that

$$\mathbb{E}_N\{(f(Y)-X)^2\}=\mathbb{E}_N\left\{e^{-2\sigma^2}Y^2+f(Y)^2-2Y\mathcal{M}f(Y)\right\}.\tag{2.4}$$

Therefore, $\mathtt{LURE}(f)\overset{\text{def.}}{=}e^{-2\sigma^2}Y^2+f(Y)^2-2Y\mathcal{M}f(Y)$ is an unbiased estimate of the MSE. ■

Theorem 2.2 (multivariate LURE). *Let* $\mathbf{Y}=\mathbf{X}\odot\mathbf{N}$, *where* $\mathbf{X},\mathbf{Y},\mathbf{N}\in\mathcal{R}^m$, $\odot$ *denotes elementwise multiplication,* $\mathbf{X}$ *is unknown, and the noise vector* $\mathbf{N}$ *has i.i.d. entries following the lognormal distribution* $\mathcal{LN}(0,\sigma^2)$. *Let* $\mathbf{f}:\mathcal{R}^m\to\mathcal{R}^m$ *denote the denoiser. Then*

$$\mathtt{LURE}(\mathbf{f})\overset{\text{def.}}{=}e^{-2\sigma^2}\|\mathbf{Y}\|^2+\|\mathbf{f}(\mathbf{Y})\|^2-2\mathbf{Y}^\top\mathcal{M}\mathbf{f}(\mathbf{Y})\tag{2.5}$$

is an unbiased estimator of the MSE, i.e., $E_{\mathbf{N}}\{\mathtt{LURE}(\mathbf{f})\}=\mathbb{E}_{\mathbf{N}}\{\|\mathbf{X}-\mathbf{f}(\mathbf{Y})\|^2\}$ *for any* $\mathbf{f}$ *with finite first- and second-order statistics. The operator* $\mathcal{M}\mathbf{f}(\mathbf{Y})$ *is defined as*

$$\mathcal{M}\mathbf{f}(\mathbf{Y})=[\mathcal{M}_1\mathbf{f}(\mathbf{Y}),\mathcal{M}_2\mathbf{f}(\mathbf{Y}),\ldots,\mathcal{M}_m\mathbf{f}(\mathbf{Y})]^\top,\textit{where}\tag{2.6}$$

$$\mathcal{M}_i\mathbf{f}(\mathbf{Y})=e^{-\frac{\sigma^2}{2}}\,f_i(Y_1,\ldots,Y_{i-1},e^{-\sigma^2}Y_i,Y_{i+1},\ldots,Y_m),\tag{2.7}$$

where, in turn, f_i *denotes the ith entry of* $\mathbf{f}$.

Proof. Expanding the MSE gives

$$\mathbb{E}_{\mathbf{N}}\{\|\mathbf{X}-\mathbf{f}(\mathbf{Y})\|^2\}=\mathbb{E}_{\mathbf{N}}\left\{\|\mathbf{X}\|^2+\|\mathbf{f}(\mathbf{Y})\|^2-2\mathbf{X}^\top\mathbf{f}(\mathbf{Y})\right\}.$$

Consider the following expectation and its simplification:

$$\mathbb{E}_{\mathbf{N}}\{\|\mathbf{Y}\|^2\}=\mathbb{E}_{\mathbf{N}}\left\{\sum_{i=1}^{m}X_i^2\,N_i^2\right\}=\sum_{i=1}^{m}X_i^2\,\mathbb{E}_{N_i}\left\{N_i^2\right\}=\sum_{i=1}^{m}X_i^2\,e^{2\sigma^2}=e^{2\sigma^2}\,\|\mathbf{X}\|^2$$

$$\Rightarrow\|\mathbf{X}\|^2=e^{-2\sigma^2}\,\mathbb{E}_{\mathbf{N}}\left\{\|\mathbf{Y}\|^2\right\}.\tag{2.8}$$

Downloaded 01/23/26 to 140.112.41.219 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Next, consider the cross-term

$$\mathbb{E}_{\mathbf{N}}\{\mathbf{X}^\top \mathbf{f}(\mathbf{Y})\} \overset{\text{def.}}{=} \sum_{i=1}^{m} \mathbb{E}_{N_1,N_2,\dots,N_m}\{X_i\, f_i(\mathbf{Y})\}$$

$$= \sum_{i=1}^{m} \int_{n_1} \cdots \int_{n_m} X_i f_i(\mathbf{Y})\, p_{\mathbf{N}}(n_1,n_2,\dots,n_m)\, \mathrm{d}n_1\, \mathrm{d}n_2 \dots \mathrm{d}n_m, \tag{2.9}$$

where the integrals are over $(0^+,+\infty)$. Invoking the i.i.d. assumption on the N_is, this term can be expressed as follows:

$$\mathbb{E}_{\mathbf{N}}\{\mathbf{X}^\top \mathbf{f}(\mathbf{Y})\} = \sum_{i=1}^{m} \int_{n_1} \cdots \int_{n_{i-1}} \int_{n_{i+1}} \cdots \int_{n_m} \prod_{\substack{j=1\\ j\neq i}}^{m} p_{N_j}(n_j)$$

$$\times \underbrace{\left(\int_{n_i} X_i\, f_i(Y_1,\dots,Y_{i-1},X_i n_i,Y_{i+1},\dots,Y_m)\, p_{N_i}(n_i)\, \mathrm{d}n_i \right)}_{\mathcal{I}} \prod_{\substack{j=1\\ j\neq i}}^{m} \mathrm{d}n_j. \tag{2.10}$$

By virtue of the property $p_{N_i}(n_i) = e^{\frac{3\sigma^2}{2}} n\, p_{N_i}(n_i\, e^{\sigma^2})$, the integral $\mathcal{I}$ can be expressed as

$$\mathcal{I} = e^{\frac{3\sigma^2}{2}} \int_{0^+}^{+\infty} X_i\, f_i(Y_1,\dots,Y_{i-1},X_i n_i,Y_{i+1},\dots,Y_m) n_i\, p_{N_i}(n_i e^{\sigma^2})\, \mathrm{d}n_i. \tag{2.11}$$

Substituting $t_i = n_i e^{\sigma^2}$ in the above integral gives

$$\mathcal{I} = e^{-\frac{\sigma^2}{2}} \int_{0^+}^{+\infty} X_i\, t_i\, f_i(Y_1,\dots,Y_{i-1},e^{-\sigma^2} X_i t_i,Y_{i+1},\dots,Y_m)\, p_{N_i}(t_i)\, \mathrm{d}t_i. \tag{2.12}$$

Substituting the expression for $\mathcal{I}$ in (2.10), we get

$$\begin{aligned}
\mathbb{E}_{\mathbf{N}}\{\mathbf{X}^\top \mathbf{f}(\mathbf{Y})\} &= \sum_{i=1}^{m} \int_{n_1} \cdots \int_{n_{i-1}} \int_{n_{i+1}} \cdots \int_{n_m} \prod_{\substack{j=1\\ j\neq i}}^{m} p_{N_j}(n_j) \\
&\quad \times e^{-\frac{\sigma^2}{2}} \left\{ \int_{0^+}^{+\infty} X_i t_i f_i(Y_1,\dots,Y_{i-1},e^{-\sigma^2} X_i t_i,Y_{i+1},\dots,Y_m)\, p_{N_i}(t_i)\, \mathrm{d}t_i \right\} \prod_{\substack{j=1\\ j\neq i}}^{m} \mathrm{d}n_j \\
&= \sum_{i=1}^{m} e^{-\frac{\sigma^2}{2}} \int_{n_1} \cdots \int_{n_m} X_i n_i f_i(Y_1,\dots,Y_{i-1},e^{-\sigma^2} Y_i,Y_{i+1},\dots,Y_m)_i \\
&\quad \times\ p_{\mathbf{N}}(n_1,\dots,n_i,\dots,n_m)\, \mathrm{d}n_1 \dots \mathrm{d}n_i \dots \mathrm{d}n_m,
\end{aligned}$$

where t_i, which is a dummy variable of integration, has been replaced with n_i. Effectively, we have

$$\mathbb{E}_{\mathbf{N}}\{\mathbf{X}^\top \mathbf{f}(\mathbf{Y})\} = \sum_{i=1}^{m} \mathbb{E}_{\mathbf{N}} \left\{ Y_i e^{-\frac{\sigma^2}{2}} f_i(Y_1,\dots,Y_{i-1},e^{-\sigma^2} Y_i,Y_{i+1},\dots,Y_m) \right\}. \tag{2.13}$$

Downloaded 01/23/26 to 140.112.41.219 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/23/26 to 140.112.41.219 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Following the definitions of $\mathcal{M}$ and $\mathcal{M}_i$ in (2.6) and (2.7), respectively, we can express

$$\mathbb{E}_{\mathbf{N}}\left\{\mathbf{X}^\top\mathbf{f}(\mathbf{Y})\right\} = \sum_{i=1}^{m}\mathbb{E}_{\mathbf{N}}\left\{Y_i\,\mathcal{M}_i\mathbf{f}(\mathbf{Y})\right\} = \mathbb{E}_{\mathbf{N}}\left\{\mathbf{Y}^\top\mathcal{M}\mathbf{f}(\mathbf{Y})\right\}. \tag{2.14}$$

Using (2.8) and (2.14), we can express the MSE as follows:

$$\mathbb{E}_{\mathbf{N}}\left\{\|\mathbf{X}-\mathbf{f}(\mathbf{Y})\|^2\right\} = \mathbb{E}_{\mathbf{N}}\{e^{-2\sigma^2}\,\|\mathbf{Y}\|^2 + \|\mathbf{f}(\mathbf{Y})\|^2 - 2\mathbf{Y}^\top\mathcal{M}\mathbf{f}(\mathbf{Y})\}. \tag{2.15}$$

Therefore, $\texttt{LURE}(\mathbf{f}) \overset{\text{def.}}{=} e^{-2\sigma^2}\,\|\mathbf{Y}\|^2 + \|\mathbf{f}(\mathbf{Y})\|^2 - 2\mathbf{Y}^\top\mathcal{M}\mathbf{f}(\mathbf{Y})$ is an unbiased estimate of the MSE. This completes the proof. ■

The usefulness of LURE stems from the fact that it depends on the observed noisy image $\mathbf{Y}$ and the processed image $\mathbf{f}(\mathbf{Y})$. Consider a denoiser function/network $\mathbf{f}$ parameterized by $\mathbf{\Theta}$. LURE enables optimization of the parameters in an unsupervised setting as follows:

$$\mathbf{\Theta}^* = \arg\min_{\mathbf{\Theta}}\texttt{LURE}(\mathbf{f}) \tag{2.16}$$

by considering only observed noisy images. This is an advantage, as in practice, noisy images are available, and it is not always possible to obtain the corresponding clean images. The noise parameter σ^2 can be estimated from the noisy image (as shown in section 3.2).

3. Monte Carlo LURE—a computationally efficient strategy. The denoiser $\mathbf{f}$ maps a noisy image to an estimate of the clean image. The expression for $\mathcal{M}_i\mathbf{f}(\mathbf{Y})$ in (2.7) shows that the value of $\mathcal{M}_i\mathbf{f}(\mathbf{Y})$ is effectively the output of the denoiser $\mathbf{f}$ corresponding to the input where the ith pixel is scaled by $e^{-\sigma^2}$. Therefore, computing $\mathcal{M}\mathbf{f}(\mathbf{Y})$ in (2.6) requires rescaling a pixel by $e^{-\sigma^2}$ and then computing the denoiser output, and this operation must be carried out for every pixel in the input noisy image in succession, which adds up to a huge amount of computation. For instance, if an image has 512 × 512 pixels, calculating LURE requires computing the denoiser output 2^{18} times, and that would be the case for one choice of the denoiser parameters. Since our objective is to use LURE to determine the optimal denoiser parameters, and a typical denoiser network such as the U-Net [20] has a few million parameters, direct deployment of LURE in its current form is not practically viable.

To surmount the computational hurdle, we develop an approximation to LURE by considering the series expansion of the exponential $e^{-\sigma^2}$ and a Taylor-series expansion of the denoiser $\mathbf{f}$, assuming that it is differentiable.

Consider the term

$$\mathbb{E}_{\mathbf{N}}\{\mathbf{Y}^\top\mathcal{M}\mathbf{f}(\mathbf{Y})\} = \sum_{i=1}^{m}\mathbb{E}_{\mathbf{N}}\left\{Y_i\,e^{-\frac{\sigma^2}{2}}\,f_i\left(Y_1,\ldots,Y_{i-1},e^{-\sigma^2}Y_i,Y_{i+1},\ldots,Y_m\right)\right\},$$

and invoke the approximation $e^{-\sigma^2} = 1-\sigma^2+\mathcal{O}(\sigma^4)$, which allows us to write

$$\mathbb{E}_{\mathbf{N}}\{\mathbf{Y}^\top\mathcal{M}\mathbf{f}(\mathbf{Y})\} \approx \sum_{i=1}^{m}\mathbb{E}_{\mathbf{N}}\left\{Y_i\,e^{-\frac{\sigma^2}{2}}\,f_i(Y_1,\ldots,Y_{i-1},(1-\sigma^2)Y_i,Y_{i+1},\ldots,Y_m)\right\}.$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/23/26 to 140.112.41.219 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

The approximation modified the scaled variable $Y_i e^{-\sigma^2}$ to the additively perturbed variable $Y_i - \sigma^2 Y_i$. Further, considering a differentiable $\mathbf{f}$ and approximating it using a Taylor-series expansion gives

$$\begin{aligned}\mathbb{E}_{\mathbf{N}}\{\mathbf{Y}^\top \mathcal{M}\mathbf{f}(\mathbf{Y})\} &\approx \sum_{i=1}^{m} \mathbb{E}_{\mathbf{N}}\left\{ Y_i e^{-\frac{\sigma^2}{2}} \left(f_i(Y_1,\ldots,Y_i,\ldots,Y_m) - \sigma^2 Y_i \frac{\partial f_i(Y_1,\ldots,Y_i,\ldots,Y_m)}{\partial Y_i} \right) \right\} \\ &= e^{-\frac{\sigma^2}{2}} \sum_{i=1}^{m} \mathbb{E}_{\mathbf{N}}\left\{ Y_i f_i(\mathbf{Y}) - \sigma^2 Y_i^2 \left[\boldsymbol{J}_{\mathbf{f}}\right]_{ii} \right\}, \end{aligned} \tag{3.1}$$

where $\boldsymbol{J}_{\mathbf{f}}$ denotes the Jacobian of the denoising transformation $\mathbf{f}$, and $[\boldsymbol{J}_{\mathbf{f}}]_{ii}$ denotes its (i,i)th entry. An approximation to LURE is obtained by substituting (3.1) into (2.15):

$$\text{LURE} \approx e^{-2\sigma^2} \|\mathbf{Y}\|^2 + \|\mathbf{f}(\mathbf{Y})\|^2 - 2e^{-\frac{\sigma^2}{2}} \left(\mathbf{Y}^\top \mathbf{f}(\mathbf{Y}) - \sigma^2 \sum_{i=1}^{m} Y_i^2 \left[\boldsymbol{J}_{\mathbf{f}}\right]_{ii} \right). \tag{3.2}$$

Comparing (3.2) with (2.15), we observe that the hurdle of having to compute LURE by perturbing the input image as many times as there are pixels in the image is now overcome. Instead, we are now required to compute the term $\sum_{i=1}^{m} Y_i^2 [\boldsymbol{J}_{\mathbf{f}}]_{ii}$, which depends on the Jacobian of the denoising function. Observing (3.2) closely, we see that what is needed is not the Jacobian explicitly, but the weighted trace of the Jacobian where the weights are the squares of the pixel intensities of the input noisy image. We show how this can be estimated using a Monte Carlo approach, similar to the estimation of the trace of the Jacobian in the additive Gaussian noise case [28].

3.1. Monte Carlo estimation of the weighted trace of the Jacobian.

Theorem 3.1. *Let $\mathbf{f} : \mathcal{R}^m \to \mathcal{R}^m$ be a differentiable function, and let $\boldsymbol{J}_{\mathbf{f}}$ denote its Jacobian. Then the following result holds:*

$$\sum_{i=1}^{n} Y_i^2 \left[\boldsymbol{J}_{\mathbf{f}}\right]_{ii} = \lim_{\varepsilon \to 0} \mathbb{E}_{\mathbf{B}} \left\{ (\mathbf{Y} \odot \mathbf{B})^\top \left(\frac{\mathbf{f}(\mathbf{Y} + \varepsilon(\mathbf{Y} \odot \mathbf{B})) - \mathbf{f}(\mathbf{Y} - \varepsilon(\mathbf{Y} \odot \mathbf{B}))}{2\varepsilon} \right) \right\}, \tag{3.3}$$

where $\odot$ denotes elementwise multiplication, and $\mathbf{Y}, \mathbf{B} \in \mathcal{R}^m$, with $\mathbf{B}$ having i.i.d. entries following the standard normal distribution $\mathcal{N}(0,1)$.

Proof. Effectively, the right-hand side of (3.3) corresponds to randomly perturbing $\mathbf{f}$ and computing a central difference and taking the limit as the perturbations become vanishingly small and the randomness is averaged out by the expectation operator.

The proof proceeds by considering the Taylor-series expansion for small ε:

$$\mathbf{f}(\mathbf{Y} + \varepsilon(\mathbf{Y} \odot \mathbf{B})) = \mathbf{f}(\mathbf{Y}) + \varepsilon \boldsymbol{J}_{\mathbf{f}}(\mathbf{Y})(\mathbf{Y} \odot \mathbf{B}) + \text{h.o.t.}, \tag{3.4}$$

where h.o.t. comprises the higher-order derivatives of $\mathbf{f}$. A similar expansion holds for $\mathbf{f}(\mathbf{Y} - \varepsilon(\mathbf{Y} \odot \mathbf{B}))$. Considering both expansions, we can express the central difference as follows:

$$\mathbf{f}(\mathbf{Y} + \varepsilon(\mathbf{Y} \odot \mathbf{B})) - \mathbf{f}(\mathbf{Y} - \varepsilon(\mathbf{Y} \odot \mathbf{B})) = 2\varepsilon \boldsymbol{J}_{\mathbf{f}}(\mathbf{Y})(\mathbf{Y} \odot \mathbf{B}) + \text{h.o.t.},$$

where h.o.t. comprise higher-order terms weighted by odd powers of ε only.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/23/26 to 140.112.41.219 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Premultiplying both sides with $(\mathbf{Y}\odot\mathbf{B})^\top$ gives

$$(\mathbf{Y}\odot\mathbf{B})^\top(\mathbf{f}(\mathbf{Y}+\varepsilon(\mathbf{Y}\odot\mathbf{B}))-\mathbf{f}(\mathbf{Y}-\varepsilon(\mathbf{Y}\odot\mathbf{B})))=2\varepsilon(\mathbf{Y}\odot\mathbf{B})^\top\boldsymbol{J}_{\mathbf{f}}(\mathbf{Y})(\mathbf{Y}\odot\mathbf{B})+\text{h.o.t.}$$

Taking expectation over $\mathbf{B}$ on both sides gives

$$\begin{aligned}&\mathbb{E}_{\mathbf{B}}\left\{(\mathbf{Y}\odot\mathbf{B})^\top\left(\frac{\mathbf{f}(\mathbf{Y}+\varepsilon(\mathbf{Y}\odot\mathbf{B}))-\mathbf{f}(\mathbf{Y}-\varepsilon(\mathbf{Y}\odot\mathbf{B}))}{2\varepsilon}\right)\right\}\\ &\quad=\mathbb{E}_{\mathbf{B}}\left\{(\mathbf{Y}\odot\mathbf{B})^\top\boldsymbol{J}_{\mathbf{f}}(\mathbf{Y})(\mathbf{Y}\odot\mathbf{B})\right\}+\text{h.o.t.},\end{aligned}\tag{3.5}$$

where h.o.t. comprise the even powers of ε.

Since the entries of $\mathbf{B}$ are i.i.d. and follow the standard normal distribution, the expectation on the right-hand side can be simplified as follows:

$$\begin{aligned}\mathbb{E}_{\mathbf{B}}\left\{(\mathbf{Y}\odot\mathbf{B})^\top\boldsymbol{J}_{\mathbf{f}}(\mathbf{Y})(\mathbf{Y}\odot\mathbf{B})\right\}&=\mathbb{E}_{\mathbf{B}}\left\{\sum_{i=1}^m Y_i^2b_i^2\,[\boldsymbol{J}_{\mathbf{f}}]_{ii}\right\}+\text{h.o.t.}\\ &=\sum_{i=1}^m Y_i^2\,[\boldsymbol{J}_{\mathbf{f}}]_{ii}+\text{h.o.t.}\end{aligned}\tag{3.6}$$

In the limit, as $\varepsilon\to 0$ in (3.5), the h.o.t. vanish, leading us to the desired result in (3.3). ■

The importance of this result lies in the fact that the weighted trace of the Jacobian of $\mathbf{f}$ can be estimated using a Monte Carlo approach as follows, by replacing ensemble averages with sample averages:

$$\sum_{i=1}^m Y_i^2\,[\boldsymbol{J}_{\mathbf{f}}]_{ii}\approx\frac{1}{nK}\sum_{i=1}^n\sum_{k=1}^K(\mathbf{Y}\odot\mathbf{B}_i)^\top\left(\frac{\mathbf{f}(\mathbf{Y}+\varepsilon_k(\mathbf{Y}\odot\mathbf{B}_i))-\mathbf{f}(\mathbf{Y}-\varepsilon_k(\mathbf{Y}\odot\mathbf{B}_i))}{2\varepsilon_k}\right),\tag{3.7}$$

where n random perturbations $\mathbf{B}_i$, where $n\ll m$, and K small values of $\varepsilon_k\in(0,1)$ are used to perturb the input to the denoiser. The larger the values of n and K, the more accurate the estimate will be. We found out experimentally that $n=1$ and $K=12$ gave sufficiently accurate estimates for practical denoising applications.

The choice of the central difference as opposed to the first-order forward or backward difference is motivated by a key observation pertaining to Taylor-series approximations with respect to the order of the residual. Whereas the forward/backward difference based Taylor-series approximation contains h.o.t. of all orders, the central difference based approximation contains h.o.t. of even orders only, which makes it more accurate than the forward/backward difference based approximation.

We are now equipped with all the results required to propose a practically viable technique for accurately estimating the MSE. This key result, referred to as Monte Carlo LURE (MC-LURE) is presented next.

Theorem 3.2 (MC-LURE). *Let* $\mathbf{Y}=\mathbf{X}\odot\mathbf{N}$, *where* $\mathbf{X},\mathbf{Y},\mathbf{N}\in\mathcal{R}^m$, $\odot$ *denotes pointwise multiplication,* $\mathbf{X}$ *is unknown, and the noise vector* $\mathbf{N}$ *has i.i.d. entries following the lognormal*

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/23/26 to 140.112.41.219 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

distribution $\mathcal{LN}(0,\sigma^2)$. Let $\mathbf{f}:\mathcal{R}^m \to \mathcal{R}^m$ denote the denoiser. Then a Monte Carlo estimate of the MSE is given by

$$\begin{aligned}\texttt{MonteCarloLURE} = e^{-2\sigma^2}\|\mathbf{Y}\|^2 + \|\mathbf{f}(\mathbf{Y})\|^2 \\ - 2e^{-\frac{\sigma^2}{2}}\left(\mathbf{Y}^\top\mathbf{f}(\mathbf{Y}) - \frac{\sigma^2}{nK}\sum_{i=1}^{n}\sum_{k=1}^{K}(\mathbf{Y}\odot\mathbf{B}_i)^\top\right. \\ \left.\times\left(\frac{\mathbf{f}(\mathbf{Y}+\varepsilon_k(\mathbf{Y}\odot\mathbf{B}_i)) - \mathbf{f}(\mathbf{Y}-\varepsilon_k(\mathbf{Y}\odot\mathbf{B}_i))}{2\varepsilon_k}\right)\right).\end{aligned} \tag{3.8}$$

3.2. Estimation of the noise parameter. Computation of MC-LURE requires knowledge of the σ^2 parameter of the lognormal distribution of noise, which must be estimated from the noisy image. We provide a median-based estimator along the lines originally proposed in [13] for Gaussian noise. We first provide the intuition behind the development of the estimator and then refine it by considering median computations.

Taking the elementwise logarithm of the noisy image gives $\ln(\mathbf{Y}) = \ln(\mathbf{X}) + \ln(\mathbf{N})$. Considering the forward/backward finite difference gives $\Delta\ln(\mathbf{Y}) = \Delta\ln(\mathbf{X}) + \Delta\ln(\mathbf{N})$, where Δ denotes the finite-difference operator. We resort to the frequently made assumption that the image $\mathbf{X}$ has a piecewise-constant structure, which is true of many natural images. Consequently, we observe that $\Delta\ln(\mathbf{X})$ contains impulses. On the other hand, the finite difference of noise $\Delta\ln(\mathbf{N})$ is Gaussian distributed with zero mean and variance $2\sigma^2$. Thus, $\Delta\ln(\mathbf{Y})$ comprises spikes in Gaussian noise. The spikes can be suppressed by computing the median of $|\Delta\ln(\mathbf{Y})|$. Considering the definition of the median of the Gaussian distribution, we obtain an estimator of σ:

$$\hat{\sigma} = \frac{\text{median}\{|\Delta\ln\mathbf{Y}|\}}{\sqrt{2}\times 0.6745}. \tag{3.9}$$

The median estimator can be refined by considering a combination of processing along rows and columns. For the purpose of derivation, we considered vectorized images, but in practice, images are two-dimensional arrays, and therefore one could apply the finite-difference operator along the rows (Δ_{row}), along the columns (Δ_{col}), or along the rows, followed by the columns ($\Delta_{\text{col}}\Delta_{\text{row}}$), and then compute an average of the median estimates as follows:

$$\hat{\sigma} = \left(\frac{\text{median}\{|\Delta_{\text{row}}\ln(\mathbf{Y})|\} + \text{median}\{|\Delta_{\text{col}}\ln(\mathbf{Y})|\} + \text{median}\{|\Delta_{\text{col}}\Delta_{\text{row}}\ln(\mathbf{Y})|\}}{3\sqrt{2}\times 0.6745}\right). \tag{3.10}$$

Table 2 shows estimates of σ obtained by the above method compared against the ground-truth value for four test images. For images where the piecewise-constant assumption holds, the estimate of σ is closer to the ground-truth value of σ.

Table 3 shows the PSNR and SSIM values obtained by minimizing the Oracle MSE and square of the MC-LURE loss for the three models over the entire AFHQ dataset. The improvements in PSNR and SSIM provided by the three models are significant over the input PSNR and SSIM. Further, the PSNR and SSIM values obtained using MC-LURE are competitive and even better than those obtained with the Oracle MSE: the average PSNR is within about 1 dB, and the average SSIM is within 0.02 of the values corresponding to the

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/23/26 to 140.112.41.219 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Table 2

Accuracy of the averaged median estimator of the noise parameter σ (3.10) for various ground-truth σ values (indicated in parentheses).

Image	**Estimated σ** (for $\sigma = 0.2$)	**Estimated σ** (for $\sigma = 0.4$)	**Estimated σ** (for $\sigma = 0.6$)	**Estimated σ** (for $\sigma = 0.8$)
Checkerboard	0.202	0.406	0.607	0.808
Cameraman	0.207	0.406	0.604	0.802
Fields	0.202	0.403	0.601	0.800
House	0.201	0.402	0.601	0.800

Table 3

PSNR (dB) and SSIM values (average ± standard deviation) obtained by training U-Net, DRUNet, and GS-DRUNet denoising models by minimizing the Oracle MSE and the square of the proposed MC-LURE loss function. The parameters of the lognormal distribution are $\mu = 0$ and $\sigma = 0.3$. The input PSNR and SSIM are 9.69 dB and 0.274, respectively. The values shown are obtained by validating over the entire AFHQ dataset. We observe that the denoising models provide significant improvements over the input PSNR and SSIM. The performance gains obtained by optimizing MC-LURE are competitive and in some cases even better than those obtained using the Oracle MSE.

Denoising network	**Oracle MSE**		**MC-LURE**	
	PSNR (dB)	SSIM	PSNR (dB)	SSIM
U-Net [20]	20.55±3.6	0.644±0.081	21.53±2.82	0.638±0.072
GS-DRUNet [21]	21.76±2.99	0.605±0.128	21.46±2.84	0.607±0.123
DRUNet [22]	19.62±3.2	0.608±0.116	21.56±3.04	0.631±0.111

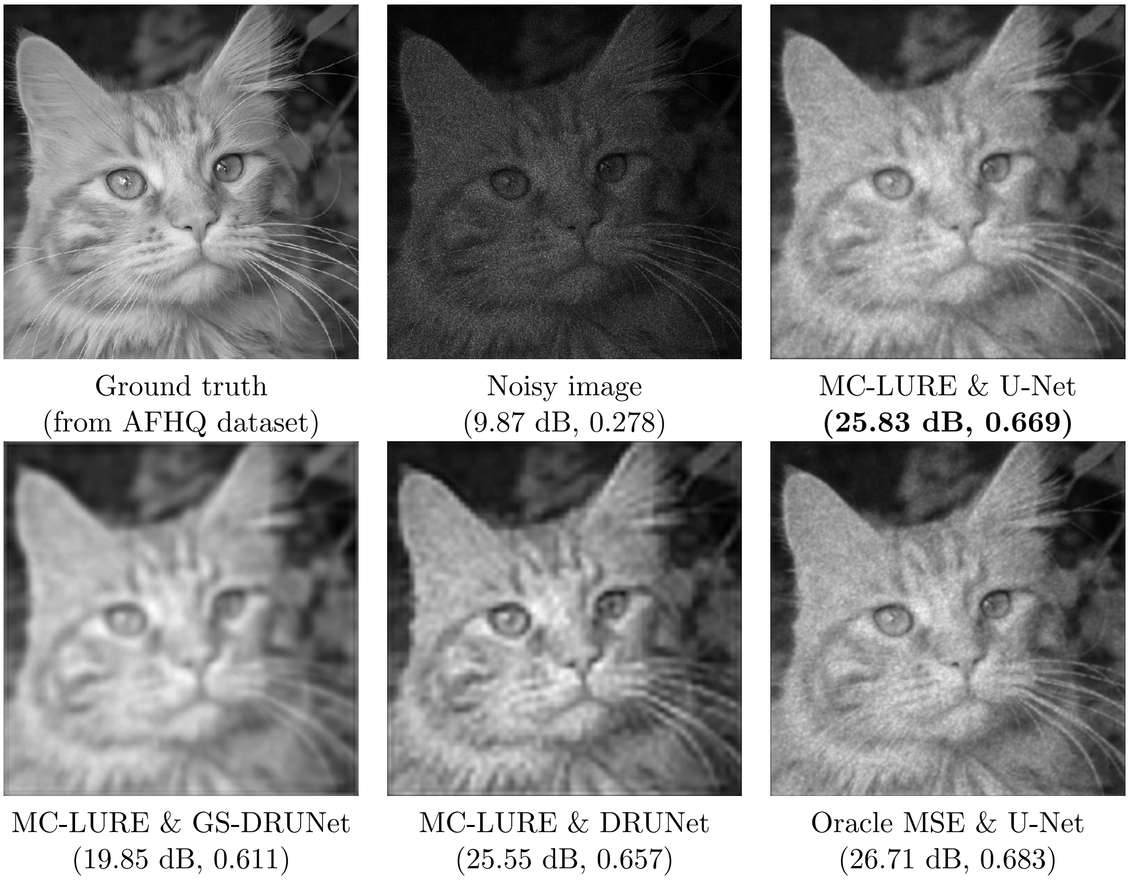
Oracle MSE. The superior performance of MC-LURE can be attributed to better generalizability, as it depends only on noisy observations, whereas the Oracle MSE with access to the ground-truth clean images may be causing the network to overfit to noise seen during training. This result testifies that the MC-LURE provides an accurate and a practically viable unsupervised alternative for training large neural networks with millions of parameters and yet give a competitive denoising performance that is comparable or even better than the oracle setting. Among the three models, U-Net provides superior denoising performance compared with DRUNet and GS-DRUNet. Figure 2 shows illustrative images output by the U-Net, DRUNet, and GS-DRUNet models. The visual quality of the images output by U-Net is better than that output by DRUNet and GS-DRUNet. For the subsequent experiments, we use U-Net for denoising in view of its relatively superior denoising performance.

4. Optimizing neural network denoisers based on MC-LURE.

For denoising, we consider prominent neural network denoisers—U-Net [20], dilated residual U-Net (DRUNet) [22], and gradient-step DRUNet (GS-DRUNet) [21]—and optimize them using MC-LURE. The number of learnable parameters in these networks is very large—U-Net: 7.69 million; DRUNet: 32.63 million, and GS-DRUNet: 17.01 million. We consider two paradigms—the *supervised* framework where the networks are trained using oracle knowledge of the ground-truth images and the MSE as the loss function; and the *unsupervised* framework where the networks are trained based on MC-LURE (3.8), without requiring access to the ground truth. Since MC-LURE is an estimate of the MSE, in practice, it can even go negative sometimes, and therefore the square of MC-LURE was found to be more appropriate for training denoising neural networks.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/23/26 to 140.112.41.219 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy


Ground truth (from AFHQ dataset)

Noisy image (9.87 dB, 0.278)

MC-LURE & U-Net **(25.83 dB, 0.669)**

MC-LURE & GS-DRUNet (19.85 dB, 0.611)

MC-LURE & DRUNet (25.55 dB, 0.657)

Oracle MSE & U-Net (26.71 dB, 0.683)

Figure 2. *A comparison of the denoising performance (PSNR in dB, SSIM) of U-Net, DRUNet, and GS-DRUNet optimized with MC-LURE loss. We observe that MC-LURE works best with U-Net.*

The training was carried out on the AFHQ dataset [6], which consists of 16,130 high-quality images at 512 × 512 resolution belonging to three different classes—`Cat`, `Dog`, and `Wild`. Out of the total of 16,130 images, 14,630 were used for training and 1,500 for testing. Noisy images were simulated by multiplying the clean images with lognormal distributed noise for various values of σ^2. The generated noisy images were used for training different models with the square of the MC-LURE loss given in (3.8). The Adam optimizer [23] was used with a learning rate of 10^{-6}, the value of which, like in all neural network training tasks, was experimentally determined. The batch size was 8 in the case of U-Net and 4 in the case of DRUNet and GS-DRUNet, with a smaller batch size in the case of DRUNet and GS-DRUNet given their large model sizes. The models were trained on NVIDIA RTX A6000 48 GB GPU. The performance is reported in terms of PSNR and SSIM.

4.1. Experimental results. Table 4 compares the PSNR and SSIM gains offered by MC-LURE versus Oracle MSE for various input noise levels on the training set (14,630 images). The denoising model is U-Net retrained for every input noise level for a range of noise levels. During training, the model optimized with MC-LURE does not rely on ground-truth clean

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/23/26 to 140.112.41.219 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Table 4

A comparison of PSNR (dB) and SSIM (average ± standard deviation) obtained using the U-Net model by minimizing the Oracle MSE and MC-LURE loss on the training set from the AFHQ dataset.

σ	**Input**		**MC-LURE**		**Oracle MSE**	
	PSNR (dB)	SSIM	PSNR (dB)	SSIM	PSNR (dB)	SSIM
0.1	17.99±3.39	0.648±0.103	24.44±3.2	0.772±0.08	23.14±3.28	0.787±0.079
0.2	12.24±2.49	0.417±0.093	23.28±3.06	0.704±0.071	22.63±3.69	0.712±0.086
0.3	9.69±2.12	0.274±0.07	21.53±2.79	0.638±0.072	20.61±3.57	0.644±0.082
0.4	8.3±1.95	0.184±0.05	20.05±2.79	0.539±0.073	19.76±3.19	0.535±0.082
0.5	7.47±1.85	0.127±0.038	20.2±2.45	0.503±0.07	20.58±2.86	0.597±0.088
0.6	6.95±1.79	0.089±0.03	19.49±2.36	0.481±0.071	19.27±2.98	0.575±0.088

Table 5

Comparison of PSNR (dB) and SSIM (average values ± standard deviation) obtained using the U-Net model by minimizing the Oracle MSE and MC-LURE loss on the test set from the AFHQ dataset.

σ	**Input**		**MC-LURE**		**Oracle MSE**	
	PSNR (dB)	SSIM	PSNR (dB)	SSIM	PSNR (dB)	SSIM
0.1	18.11±3.49	0.65±0.10	24.48±3.14	0.772±0.08	23.12±3.31	0.786±0.079
0.2	12.29±2.49	0.418±0.095	23.30±3.02	0.704±0.071	22.61±2.49	0.712±0.085
0.3	9.73±2.14	0.274±0.07	21.53±2.82	0.638±0.072	20.55±3.6	0.644±0.081
0.4	8.34±1.97	0.184±0.051	20.04±2.81	0.539±0.074	19.76±3.19	0.535±0.082
0.5	7.5±1.87	0.127±0.038	20.25±2.47	0.503±0.07	20.62±2.87	0.597±0.087
0.6	6.97±1.81	0.089±0.029	19.54±2.36	0.481±0.072	19.27±3.00	0.574±0.086

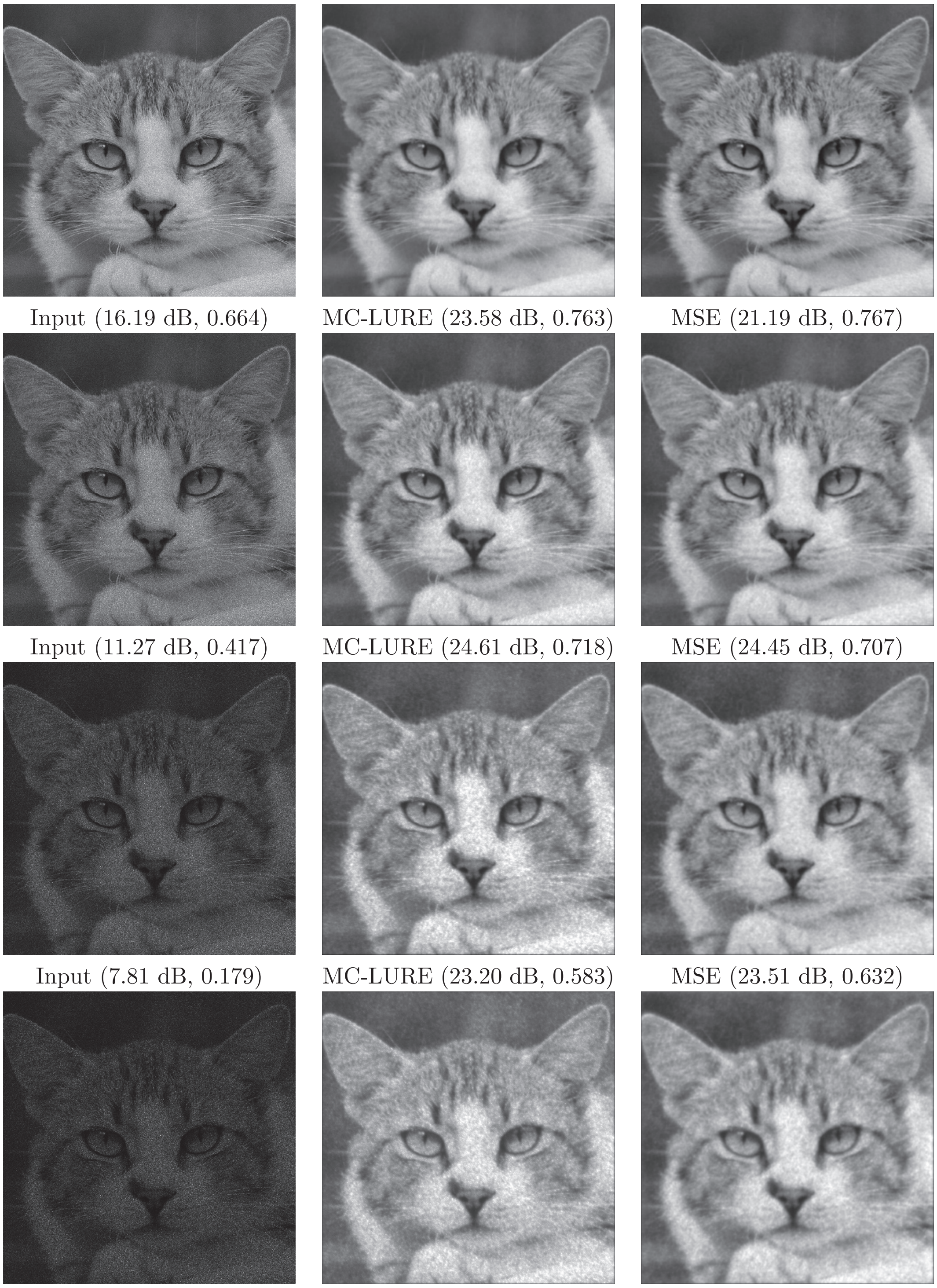
images. Table 5 shows the results obtained by evaluating the trained models on the test set (1,500 images). The performance achieved using MC-LURE is on par or even slightly better than that obtained using Oracle MSE. When the input PSNR is high, the accuracy of the Taylor-series approximations in the derivation of MC-LURE is high, and hence the performance is also on par with or even better than that obtained using Oracle MSE. At low input PSNRs, where the approximation error is high, the Oracle MSE has a marginally superior performance over MC-LURE. While the training data performance achieved with both Oracle MSE and MC-LURE is comparable, the results show that the generalizability of the model to unseen test set images is higher in the case of MC-LURE. Figure 3 shows representative results comparing MC-LURE with Oracle MSE for several input noise levels. All of these results point to the high relevance of MC-LURE for practical scenarios where ground-truth clean images are not available for training a denoising network.

We next train a U-Net model with 400k noisy instances synthetically generated from the AFHQ dataset. The images were resized to 128 × 128 due to computational constraints. Table 6 compares the PSNR and SSIM gains offered by this model comparing MC-LURE versus Oracle MSE for input noise with $\sigma = 0.2$. The results show that Oracle MSE has a marginally superior performance than MC-LURE in terms of PSNR and comparable SSIM.

4.2. Performance comparison with benchmark techniques. We carry out two types of performance comparisons. In the first type, we convert the multiplicative observation model to an additive one by taking the logarithm, denoising using an additive white Gaussian noise

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/23/26 to 140.112.41.219 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy



Input (16.19 dB, 0.664) MC-LURE (23.58 dB, 0.763) MSE (21.19 dB, 0.767)

Input (11.27 dB, 0.417) MC-LURE (24.61 dB, 0.718) MSE (24.45 dB, 0.707)

Input (7.81 dB, 0.179) MC-LURE (23.20 dB, 0.583) MSE (23.51 dB, 0.632)

Input (7.05 dB, 0.124) MC-LURE (21.64 dB, 0.516) MSE (22.54 dB, 0.594)

Figure 3. *A comparison of denoising performance of U-Net with MC-LURE loss vis-à-vis the Oracle MSE for different levels of the input lognormal noise. MC-LURE-driven U-Net gives competitive denoising performance with respect to the Oracle MSE.*

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/23/26 to 140.112.41.219 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Table 6

A comparison of PSNR (dB) and SSIM (average ± standard deviation) obtained using U-Net by minimizing the Oracle MSE and MC-LURE loss, trained on 400k noisy images synthesized from the AFHQ dataset.

σ	**Input**		**MC-LURE**		**Oracle MSE**	
	PSNR (dB)	SSIM	PSNR (dB)	SSIM	PSNR (dB)	SSIM
0.2	12.28±2.56	0.417±0.096	21.32±3.38	0.672±0.086	22.23±3.71	0.666±0.082

Table 7

A comparison of the denoising performance of the proposed MC-LURE-driven U-Net model with standard additive noise suppression techniques (denoising after log-transformation), using wavelet, BM3D, and DnCNN on the AFHQ data [6]. The performance metrics PSNR (dB) and SSIM show that the best results (highlighted in boldface) are obtained with MC-LURE-driven U-Net. The average input PSNR is 12.24 *with a standard deviation of* 2.49*, and the average input SSIM is* 0.417 *with a standard deviation of* 0.093 *on the test dataset.*

Denoising network	**Output PSNR (dB)**	**Output SSIM**
U-Net [20] with MC-LURE (ours)	$\mathbf{23.30 \pm 3.02}$	$\mathbf{0.704 \pm 0.071}$
Wavelet denoiser [3, 29]	10.79 ± 2.26	0.44 ± 0.08
BM3D [8]	10.67 ± 2.29	0.43 ± 0.10
DnCNN [38]	10.71 ± 2.27	0.37 ± 0.08

denoiser, and then exponentiating the result. These performance comparisons are with respect to additive noise suppression techniques.

The second set of comparisons pertains to benchmarking the performance of the U-Net denoiser with MC-LURE loss function vis-à-vis the state-of-the-art multiplicative noise suppression techniques operating in the unsupervised learning setting.

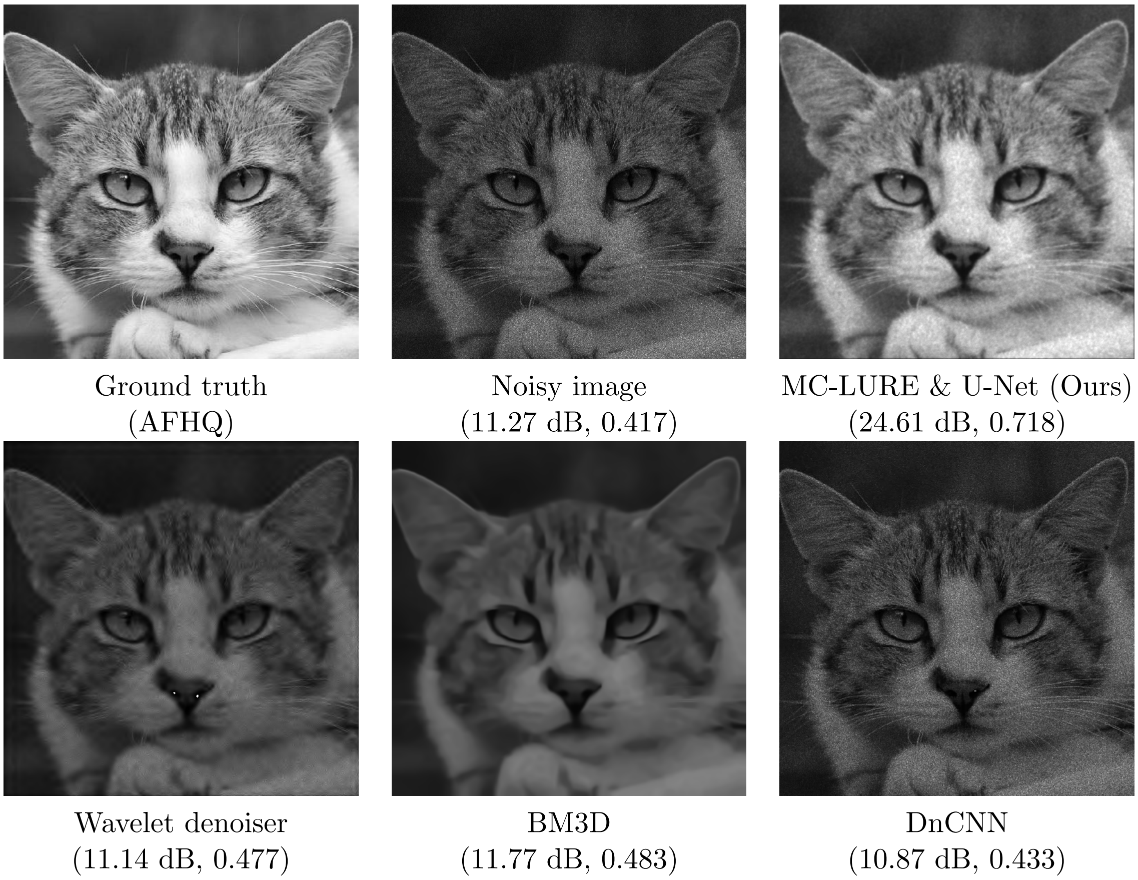
The validations are reported on images from the AFHQ dataset, the Sentinel-1 dataset [26], and the SEN1-2 dataset [31]. The standard performance measures are PSNR and SSIM. In the case of real-world noisy images from the Sentinel-1 and SEN1-2 datasets, we also report the contrast ratio (CR), which is computed as the ratio of the average intensity of the pixels in a region of higher intensity to that in a region of lower intensity. A higher CR indicates superior denoising performance.

For the first set of validations, we consider non-deep-learning denoising techniques such as wavelet denoising [29], BM3D [8], and a pretrained DnCNN with the Oracle MSE loss. Figures 4 and 5 show some illustrative results. The CR is superior for the proposed technique. Table 7 shows objective performance comparisons. The non-deep-learning methods and pretrained DnCNN model do not significantly improve the PSNR and SSIM, whereas U-Net trained with MC-LURE operates directly on the multiplicative noisy images and offers significant improvements. This experiment shows that it is advantageous to reoptimize denoisers for the multiplicative noise scenario instead of converting them to the additive type and using off-the-shelf pretrained additive denoisers. Even if the models were retrained to handle log-transformed images, exponentiation of the denoiser output is not guaranteed to results in the minimum MSE estimate. Since retraining is anyway inevitable to achieve good denoising performance, we recommend retraining using the MC-LURE loss function in the unsupervised setting instead of the log-transformation approach.

For the second set of validations, the unsupervised despeckling techniques chosen for comparison are SAR2SAR [10], Speckle2void [26], and SAR-CNN [11]. These models were

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/23/26 to 140.112.41.219 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy


Ground truth (AFHQ)

Noisy image (11.27 dB, 0.417)

MC-LURE & U-Net (Ours) (24.61 dB, 0.718)

Wavelet denoiser (11.14 dB, 0.477)

BM3D (11.77 dB, 0.483)

DnCNN (10.87 dB, 0.433)

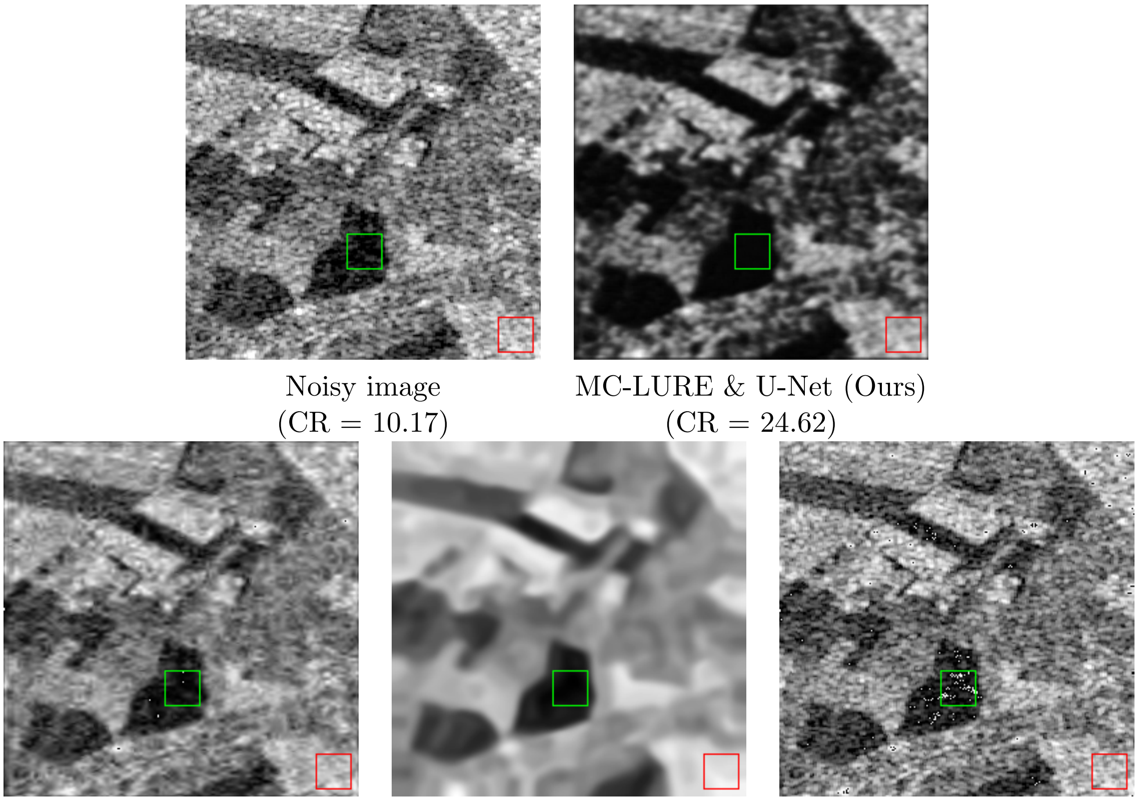
Figure 4. *A comparison of the denoising performance (PSNR, SSIM) of various techniques on an image taken from the* AFHQ *dataset (available at* https://www.kaggle.com/datasets/andrewmvd/animal-faces*)* [6]*—MC-LURE-driven U-Net with multiplicative lognormal noise versus off-the-shelf additive noise suppression techniques with log-transformed noise and denoising, followed by exponentiation. These results show that MC-LURE operating directly in the multiplicative noise domain gives performance superior to log-transformation followed by additive noise suppression.*

retrained on lognormal multiplicative noise-corrupted images from the AFHQ dataset. SAR-CNN required retraining on the lognormal noise data instead of the originally used gamma noise data. For SAR2SAR, retraining required an additional adaptation, which is a change of the loss function, from the log-likelihood function of the gamma distribution to that of the lognormal distribution. It can be verified that maximizing the log-likelihood results in MSE minimization. Since we are considering the noise to be independent at each pixel, the additional processing steps in the SAR2SAR pipeline that account for correlation of noise [10] are not needed. Speckle2Void was specifically developed to handle gamma-distributed multiplicative noise. The technique finds the best-fit shape and scale parameters, assuming that the input noise follows the gamma distribution. With these adaptations of the baselines to handle lognormal noise, the results obtained are shown in Figure 6 and Table 8.

The performance of the baselines, with the exception of Speckle2Void, has definitely improved with these adaptations, as opposed to taking them off the shelf and inferencing on the

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/23/26 to 140.112.41.219 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy



Noisy image (CR = 10.17)

MC-LURE & U-Net (Ours) (CR = 24.62)

Wavelet denoiser (CR = 9.22)

BM3D (CR = 18.56)

DnCNN (CR = 4.46)

Figure 5. *A comparison of the denoising performance of various techniques. The image is taken from the* `SEN1-2` *dataset* [31]. *The comparison is for MC-LURE-driven U-Net with multiplicative lognormal noise versus off-the-shelf additive noise suppression techniques with log-transformed noise and denoising, followed by exponentiation. These results show that MC-LURE operating directly in the multiplicative noise domain gives a performance superior to log-transformation followed by additive noise suppression. The MC-LURE output has superior visual quality and better contrast as measured by the contrast ratio (CR), which is the ratio of the average intensity in the red square to that in the green square.*

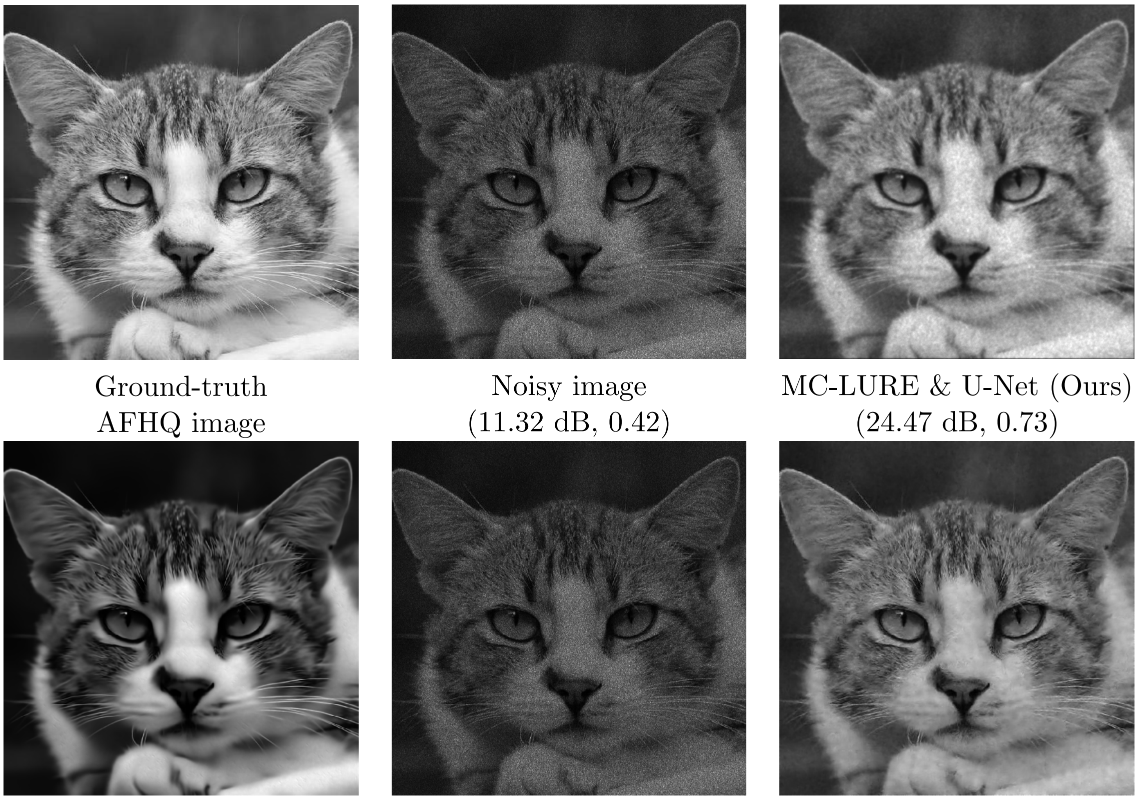
Table 8

A comparison of PSNR (dB) and SSIM (average ± standard deviation) obtained using U-Net by minimizing the MC-LURE loss, trained on 400*k noisy images synthesized from the AFHQ dataset. The average input PSNR on the test dataset is* 12.24 *dB with a standard deviation of* 2.49 *dB, and the average input SSIM is* 0.417 *with a standard deviation of* 0.093.

Denoising network	Output PSNR (dB)	Output SSIM
U-Net [20] with MC-LURE	$\mathbf{23.30 \pm 3.02}$	0.704 ± 0.071
SAR2SAR [10]	20.09 ± 5.04	$\mathbf{0.79 \pm 0.06}$
SAR-CNN [11]	17.09 ± 2.94	0.69 ± 0.08
Speckle2Void [26]	12.19 ± 2.90	0.423 ± 0.105

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/23/26 to 140.112.41.219 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy



Ground-truth AFHQ image

Noisy image (11.32 dB, 0.42)

MC-LURE & U-Net (Ours) (24.47 dB, 0.73)

SAR-CNN (13.85 dB, 0.61)

Speckle2Void (10.36 dB, 0.40)

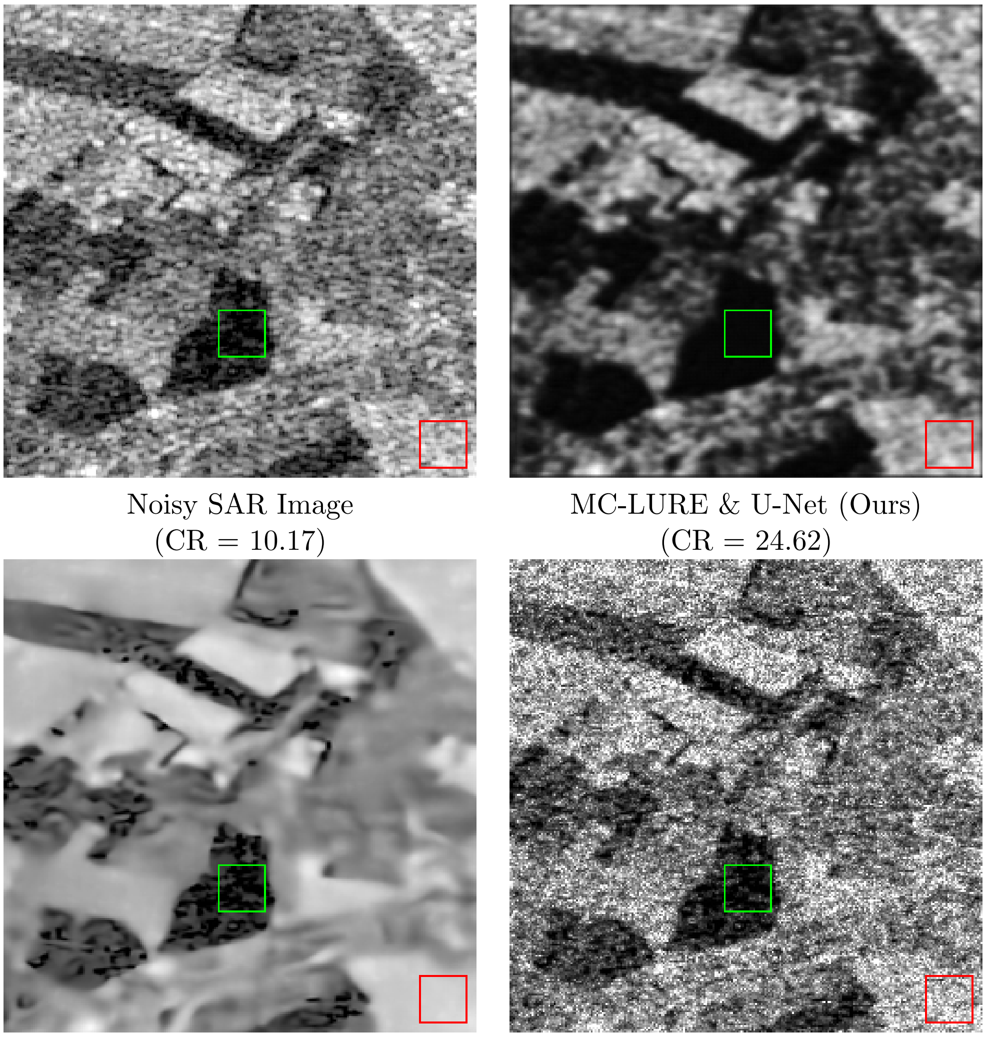
SAR2SAR (18.66 dB, 0.79)

Figure 6. *A performance comparison (PSNR, SSIM) of various unsupervised despeckling methods with the MC-LURE-driven U-Net model, on a synthetic noisy image from the AFHQ dataset (available at* https://www.kaggle.com/datasets/andrewmvd/animal-faces*)* [6]. *The neural network models for the benchmark techniques have been retrained for lognormal noise to ensure a fair comparison. We observe that the proposed technique gives significant improvement in PSNR and SSIM compared to the benchmark techniques. SAR-CNN and SAR2SAR offer improvements in PSNR and SSIM over the noisy image, although not to the same level of improvement as MC-LURE. On the other hand, Speckle2Void causes a mild reduction in PSNR and SSIM, which is due to the fact that the development of Speckle2Void is specific to gamma distributed noise, and when the noise distribution does not satisfy that assumption, the model estimates parameters of the nearest gamma distribution and employs them for denoising, which is not optimal when there is a distribution mismatch.*

lognormal noisy images. Speckle2Void did not yield any performance improvements despite having a built-in provision to find the best-fit scale and shape parameters. This is because the mathematical development of Speckle2Void is specific to gamma distributed multiplicative noise with known or estimated shape and scale parameters. When the noise distribution deviates from gamma, the performance was found to degrade drastically. Adapting Speckle2Void for the lognormal noise turned out to be a challenging task and a research problem in its own right, which requires a separate in-depth investigation.

In comparison, the U-Net model trained with MC-LURE gave significant performance improvements, as shown by the objective measures. In terms of visual assessment also, the proposed MC-LURE technique offers superior image quality.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.


Noisy SAR Image
(CR = 10.17)

MC-LURE & U-Net (Ours)
(CR = 24.62)

SAR-CNN
(CR = 9.91)

SAR2SAR
(CR = 9.46)

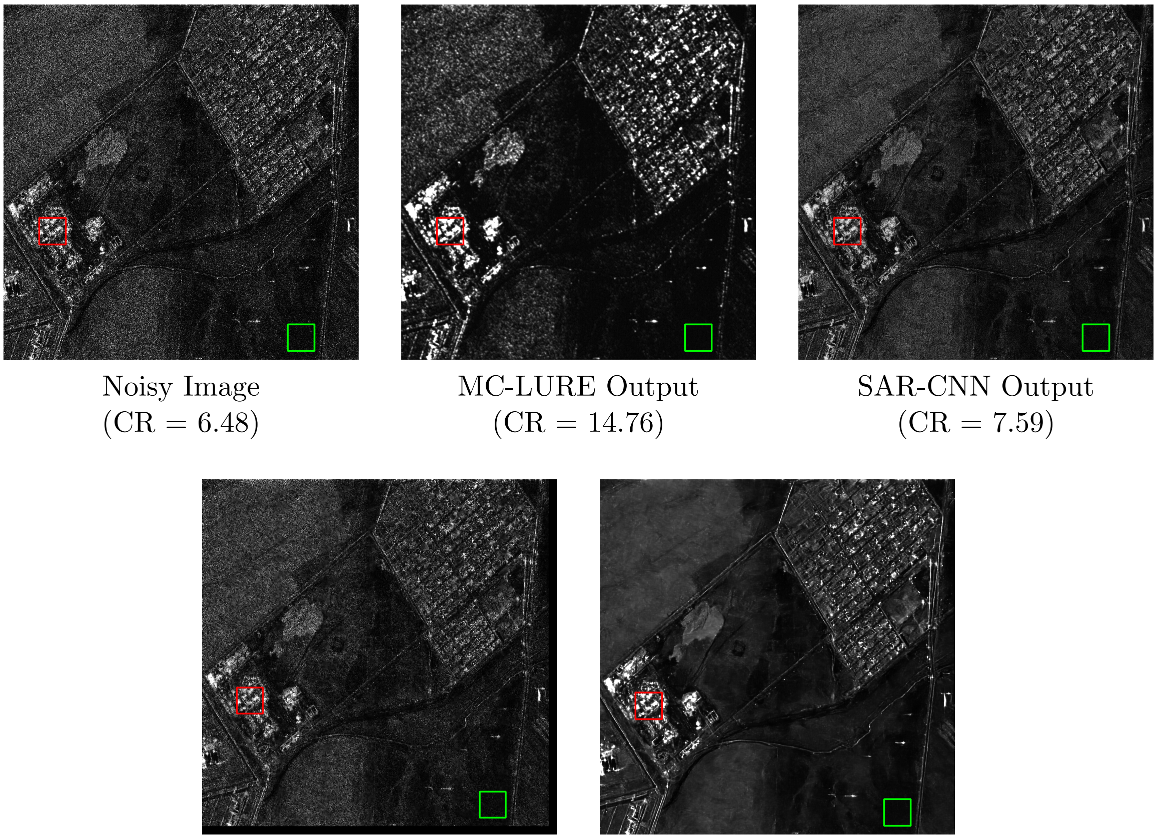
Figure 7. *A comparison of various unsupervised despeckling methods with the MC-LURE-driven U-Net model on an image from the SEN1-2 dataset* [31]. *The proposed U-Net model trained with MC-LURE offers the best denoising performance. The degree of noise suppression is particularly high in the regions of low intensity. The other techniques have a poorer degree of noise suppression. The result of Speckle2Void is not reported here because the code did not generate a reasonable output. This image is from a different source and not from that used in* [26]*, which may have caused some kind of a mismatch leading to poorer performance. The contrast ratio (CR) is computed as the ratio of average intensity in the red box to that in the green box.*

In the case of real-world SAR images, the contrast between the dark and bright regions is better in the case of the proposed technique (Figures 7 and 8). The degree of noise suppression is also high in the dark regions. The other techniques under comparison have a poorer degree of noise suppression. The proposed technique does not produce an overly smoothed output.

Downloaded 01/23/26 to 140.112.41.219 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/23/26 to 140.112.41.219 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy



Noisy Image
(CR = 6.48)

MC-LURE Output
(CR = 14.76)

SAR-CNN Output
(CR = 7.59)

SAR2SAR Output
(CR = 6.85)

Speckle2Void Output
(CR = 10.34)

Figure 8. *A comparison of various unsupervised despeckling methods with the MC-LURE-driven U-Net model on an image taken from the Sentinel-*1 *dataset used in* [26]. *The contrast ratio (CR) is computed as the ratio of the mean intensity of the bright region (box outlined in red) to that of the dark region (box outlined in green). The proposed MC-LURE technique gives a superior contrast. The output is oversmoothed in some of the baseline techniques.*

5. Conclusion.

We addressed the problem of image denoising in multiplicative noise conditions. Specifically, we considered the lognormal distribution for noise and developed the lognormal unbiased risk estimate of the MSE. Computation of the risk estimate requires estimation of the weighted trace of the Jacobian for which we proposed a computationally efficient Monte Carlo approximation technique. The μ parameter of the lognormal distribution has no effect on the denoising outcome. The σ parameter can be estimated robustly using a median-based finite-difference estimator. Experimental results showed that, despite the stochastic approximations involved in deriving MC-LURE, the quality of denoising, both in terms of PSNR and SSIM, is competitive and even better than that achieved using the Oracle MSE. The principal advantage of the proposed formalism is that it is ideal for the unsupervised setting, which is the practically relevant setting. Any state-of-the-art denoising neural network model could be optimized with MC-LURE as the loss function. We considered several types

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 01/23/26 to 140.112.41.219 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

of images for demonstrating the superior denoising capability of the proposed algorithm. We also showed that transforming the noise via the logarithm to an additive one and then using a pretrained additive Gaussian denoiser did not result in competitive denoising performance. The experimental results further showed that training the U-Net denoiser with the MC-LURE loss function gave better results than the benchmark despeckling techniques retrained on the lognormal noisy images. When tested on real-world SAR images, the MC-LURE technique offered competitive denoising performance.

Reproducibility. Our code is accessible at LURE GitHub, using which the results reported in this paper can be generated.

REFERENCES

[1] Y. Chan and V. Koo, *An introduction to synthetic aperture radar (SAR)*, Progress Electromagnet. Res. B, 2 (2008), pp. 27–60, https://doi.org/10.2528/PIERB07110101.

[2] A. Chelli, P. Lena, C. Roddier, F. Roddier, and F. Sibille, *Modulation transfer function for infrared stellar speckle interferometry: Evidence for a lognormal statistic*, Optica Acta: International J. Optics, 26 (1979), pp. 583–595, https://doi.org/10.1080/713820042.

[3] G. Chen, T. D. Bui, and A. Krzyzak, *Image denoising using neighbouring wavelet coefficients*, Integr. Comput-Aid. Eng., 12 (2005), pp. 99–107, https://doi.org/10.3233/ICA-2005-12108.

[4] L. Chen, L. Goldstein, and Q. Shao, *Fundamentals of Stein's method*, in Normal Approximation by Stein's Method, Springer, 2011, pp. 13–44.

[5] G. Chierchia, D. Cozzolino, G. Poggi, and L. Verdoliva, *SAR image despeckling through convolutional neural networks*, in IEEE International Geoscience and Remote Sensing Symposium (IGARSS), 2017, pp. 5438–5441.

[6] Y. Choi, Y. Uh, J. Yoo, and J.-W. Ha, *StarGAN v2: Diverse image synthesis for multiple domains*, in Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020.

[7] V. Churchill and A. Gelb, *Sampling-based spotlight SAR image reconstruction from phase history data for speckle reduction and uncertainty quantification*, SIAM/ASA J. Uncertain. Quantif., 10 (2022), pp. 1225–1249, https://doi.org/10.1137/20M1379721.

[8] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, *Image denoising by sparse 3-D transform-domain collaborative filtering*, IEEE Trans. Image Process., 16 (2007), pp. 2080–2095, https://doi.org/10.1109/TIP.2007.901238.

[9] E. Dalsasso, L. Denis, and F. Tupin, *As if by magic: Self-supervised training of deep despeckling networks with MERLIN*, IEEE Trans. Geosci. Remote Sens., 60 (2022), pp. 1–13.

[10] E. Dalsasso, L. Denis, and F. Tupin, *SAR2SAR: A semi-supervised despeckling algorithm for SAR images*, IEEE J. Sel. Top. Appl. Earth Observations Remote Sens., 14 (2021), pp. 4321–4329, https://doi.org/10.1109/JSTARS.2021.3071864.

[11] E. Dalsasso, X. Yang, L. Denis, F. Tupin, and W. Yang, *SAR image despeckling by deep neural networks: From a pre-trained model to an end-to-end training strategy*, Remote Sens., 12 (2020), 2636, https://doi.org/10.3390/rs12162636.

[12] R. Das, A. J. Ninan, A. Bhaskar, and A. Rajwade, *Performance bounds for lasso under multiplicative lognormal noise: Applications to pooled RT-PCR testing*, Signal Process., 214 (2024), 109233, https://doi.org/10.1016/j.sigpro.2023.109233.

[13] D. L. Donoho and I. M. Johnstone, *Ideal spatial adaptation by wavelet shrinkage*, Biometrika, 81 (1994), pp. 425–455, https://doi.org/10.1093/biomet/81.3.425.

[14] D. D. Duncan, S. J. Kirkpatrick, and R. K. Wang, *Statistics of local speckle contrast*, JOSA A, 25 (2008), pp. 9–15, https://doi.org/10.1364/JOSAA.25.000009.

[15] M. Elad, B. Kawar, and G. Vaksman, *Image denoising: The deep learning revolution and beyond—a survey paper*, SIAM J. Imaging Sci., 16 (2023), pp. 1594–1654, https://doi.org/10.1137/23M1545859.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

[16] A. Freeman, *SAR calibration: An overview*, IEEE Trans. Geosci. Remote Sens., 30 (1992), pp. 1107–1121, https://doi.org/10.1109/36.193786.

[17] G. Gao, *Statistical modeling of SAR images: A survey*, Sensors, 10 (2010), pp. 775–795, https://doi.org/10.3390/s100100775.

[18] R. Gens and J. Genderen, *Review article: SAR interferometry — issues, techniques, applications*, Int. J. Remote Sens., 17 (1996), pp. 1803–1835, https://doi.org/10.1080/014311696008948741.

[19] B. Goyal, A. Dogra, S. Agrawal, B. S. Sohi, and A. Sharma, *Image denoising review: From classical to state-of-the-art approaches*, Inform. Fusion, 55 (2020), pp. 220–244, https://doi.org/10.1016/j.inffus.2019.09.003.

[20] H. Huang, L. Lin, R. Tong, H. Hu, Q. Zhang, Y. Iwamoto, X. Han, Y. Chen, and J. Wu, *UNet 3+: A full-scale connected UNet for medical image segmentation*, in Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 2020, pp. 1055–1059.

[21] S. Hurault, A. Leclaire, and N. Papadakis, *Gradient step denoiser for convergent plug-and-play*, in International Conference on Learning Representations, 2021, https://openreview.net/forum?id=fPhKeld3Okz.

[22] M. Jafari, D. Auer, S. Francis, J. Garibaldi, and X. Chen, *DRU-Net: An efficient deep convolutional neural network for medical image segmentation*, in 17th IEEE International Symposium on Biomedical Imaging (ISBI), 2020, pp. 1144–1148.

[23] D. P. Kingma and J. Ba, *Adam: A method for stochastic optimization*, in International Conference on Learning Representations (ICLR), 2015, http://arxiv.org/abs/1412.6980.

[24] C. A. Metzler, A. Mousavi, R. Heckel, and R. Baraniuk, *Unsupervised Learning with Stein's Unbiased Risk Estimator*, preprint, arXiv:1805.10531, 2018.

[25] A. B. Molini, D. Valsesia, G. Fracastoro, and E. Magli, *Towards deep unsupervised SAR despeckling with blind-spot convolutional neural networks*, in IEEE International Geoscience and Remote Sensing Symposium (IGARSS 2020), pp. 2507–2510.

[26] A. B. Molini, D. Valsesia, G. Fracastoro, and E. Magli, *Speckle2void: Deep self-supervised SAR despeckling with blind-spot convolutional neural networks*, IEEE Trans. Geosci. Remote Sens., 60 (2022), pp. 1–17.

[27] Y. L. Montagner, E. D. Angelini, and J.-C. Olivo-Marin, *An unbiased risk estimator for image denoising in the presence of mixed Poisson–Gaussian noise*, IEEE Trans. Image Process., 23 (2014), pp. 1255–1268, https://doi.org/10.1109/TIP.2014.2300821.

[28] S. Ramani, T. Blu, and M. Unser, *Monte-Carlo SURE: A black-box optimization of regularization parameters for general denoising algorithms*, IEEE Trans. Image Process., 17 (2008), pp. 1540–1554, https://doi.org/10.1109/TIP.2008.2001404.

[29] S. Sardy, P. Tseng, and A. Bruce, *Robust wavelet denoising*, IEEE Trans. Signal Process., 49 (2001), pp. 1146–1152, https://doi.org/10.1109/78.923297.

[30] J. Schmitt, *Optical coherence tomography (OCT): A review*, IEEE J. Sel. Top. Quant. Electron., 5 (1999), pp. 1205–1215, https://doi.org/10.1109/2944.796348.

[31] M. Schmitt, *SEN*1-2, https://mediatum.ub.tum.de/1436631?style=full_text, 2018.

[32] C. Tian, L. Fei, W. Zheng, Y. Xu, W. Zuo, and C. W. Lin, *Deep learning on image denoising: An overview*, Neural Networks, 131 (2020), pp. 251–275, https://doi.org/10.1016/j.neunet.2020.07.025.

[33] P. Wang, H. Zhang, and V. M. Patel, *SAR image despeckling using a convolutional neural network*, IEEE Signal Process. Lett., 24 (2017), pp. 1763–1767, https://doi.org/10.1109/LSP.2017.2758203.

[34] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, *Image quality assessment: From error visibility to structural similarity*, IEEE Trans. Image Process., 13 (2004), pp. 600–612, https://doi.org/10.1109/TIP.2003.819861.

[35] P. Wells, *Ultrasound imaging*, Phys. Med. Biol., 51 (2006), pp. R83–R98, https://doi.org/10.1088/0031-9155/51/13/R06.

[36] P. Zakharov, A. Völker, A. Buck, B. Weber, and F. Scheffold, *Quantitative modeling of laser speckle imaging*, Opt. Lett., 31 (2006), pp. 3465–3467, https://doi.org/10.1364/OL.31.003465.

[37] C.-J. Zhang, X.-Y. Huang, and M.-C. Fang, *MRI denoising by NeighShrink based on chi-square unbiased risk estimation*, Artif. Intell. Med., 97 (2019), pp. 131–142, https://doi.org/10.1016/j.artmed.2018.12.001.

Downloaded 01/23/26 to 140.112.41.219 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

[38] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, *Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising*, IEEE Trans. Image Process., 26 (2017), pp. 3142–3155, https://doi.org/10.1109/TIP.2017.2662206.

[39] Q. Zhou, M. Wen, M. Ding, and X. Zhang, *Unsupervised despeckling of optical coherence tomography images by combining cross-scale CNN with an intra-patch and inter-patch based transformer*, Opt. Express, 30 (2022), pp. 18800–18820, https://doi.org/10.1364/OE.459477.

Downloaded 01/23/26 to 140.112.41.219 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.