Rethinking Deep Image Prior for Denoising

Yeonsik Jo[§]
LG AI Research
yeonsik.jo@lgresearch.ai

Se Young Chun
ECE, INMC, Seoul National University
sychun@snu.ac.kr

Jonghyun Choi[†]
GIST, South Korea
jhc@gist.ac.kr

Abstract

Deep image prior (DIP) serves as a good inductive bias for diverse inverse problems. Among them, denoising is known to be particularly challenging for the DIP due to noise fitting with the requirement of an early stopping. To address the issue, we first analyze the DIP by the notion of effective degrees of freedom (DF) to monitor the optimization progress and propose a principled stopping criterion before fitting to noise without access of a paired ground truth image for Gaussian noise. We also propose the 'stochastic temporal ensemble (STE)' method for incorporating techniques to further improve DIP's performance for denoising. We additionally extend our method to Poisson noise. Our empirical validations show that given a single noisy image, our method denoises the image while preserving rich textual details. Further, our approach outperforms prior arts in LPIPS by large margins with comparable PSNR and SSIM on seven different datasets.

1. Introduction

Deep neural network has been widely used in many computer vision tasks, yielding significant improvements over conventional approaches since AlexNet [18]. However, image denoising has been one of the tasks in which conventional methods such as BM3D [7] outperformed many early deep learning based ones [5, 47, 48] until DnCNN [51] outperforms it for synthetic Gaussian noise at the expense of massive amount of noiseless and noisy image pairs [51].

Requiring no clean and/or noisy image pairs, deep image prior (DIP) [42, 43] has shown that a randomly initialized network with hour-glass structure acts as a prior for several inverse problems including denoising, super-resolution, and inpainting with a single degraded image. Although DIP exhibits remarkable performance in these inverse problems, denoising is the particular task that DIP does not perform well, *i.e.*, a single run yields far lower PSNR than BM3D even for synthetic Gaussian noise set-up [42, 43]. Furthermore, for the best performance, one needs to monitor the PSNR (*i.e.*, the ground-truth clean image is required here) and stop the iterations before fitting to noise. Deep Decoder addresses the issue by proposing a strong structural regularization to allow longer iterations for the inverse problems including denoising [15]. However, it yields worse denoising performance than DIP due to low model complexity.

For better use of DIP for denoising without monitoring PSNR with a clean image, we first analyze the model complexity of the DIP by the notion of effective degrees of freedom (DF) [10, 12, 41]. Specifically, the DF quantifies the amount of overfitting (*i.e.*, optimism) of a chosen hypothesis (*i.e.*, a trained neural network model) to the given training data [10]. In other words, when overfitting occurs, the DF increases. Therefore, to prevent the overfitting of the DIP network to the noise, we want to suppress the DF over

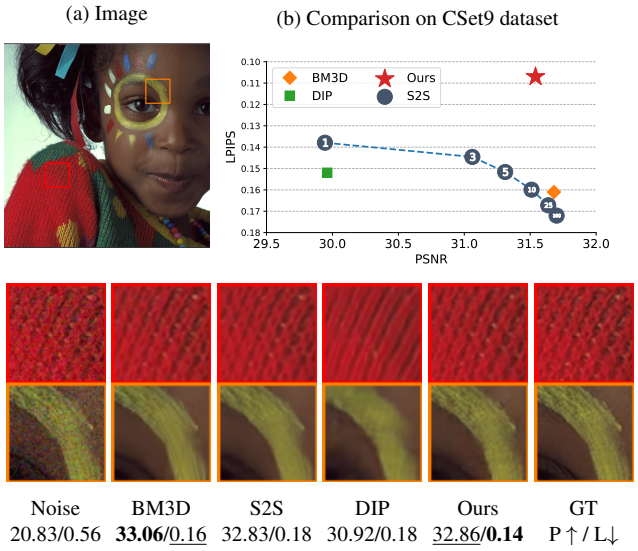


Figure 1: **Comparison of single image based denoising methods.** 'L↓' refers to LPIPS and it is lower the better. 'P↑' refers to PSNR and it is higher the better. Our method denoises an image while preserves rich details; showing the best LPIPS with comparable PSNR to Self2Self (S2S) [30]. Ours shows much better trade-off in PSNR and LPIPS than all other methods including all different ensembling attempts of state of the art (S2S). (Numbers in the circle of S2S denotes number of models in ensemble)

[§]: work done while with GIST. [†]: corresponding author.
Code: https://github.com/gistvision/DIP-denosing

iterations. But obtaining DF again requires a clean (ground truth) image. Fortunately, for the Gaussian noise model, there are approximations for DF without using a clean image; Monte-Carlo divergence approximations in Stein's unbiased risk estimator (SURE) (Eqs. 8, 9) (DF_{MC}).

Leveraging SURE and improvement techniques in DIP [43], we propose an objective with 'stochastic temporal ensembling (STE),' which mimics ensembling of many noise realizations in a single optimization run. On the proposed objective with the STE, we propose to stop the iteration when the proposed objective function crosses zero. The proposed method leads to much better solutions than DIP and outperforms prior arts for single image denoising. In addition, inspired by PURE formulation [20, 24], we extend our objective function to address the Poisson noise.

We empirically validate our method by comparing DIP based prior arts for denoising performance in various metrics that are suggested in the literature [13] such as PSNR, SSIM and learned perceptual image patch similarity (LPIPS) [54] on seven different datasets. LPIPS has been widely used in super resolution literature to complement PSNR, SSIM to measure the recovery power of details [21]. Since it is challenging for denoiser to suppress noise and preserve details together [4], we argue that LPIPS is another appropriate metric to evaluate denoisers. Note that it has not been widely used in denoising literature yet to analyze the denoising performance. Our method not only denoises the images but also preserves rich textual details, outperforming other methods in LPIPS with comparable classic measures including the PSNR and SSIM.

Our contributions are summarized as follows:

- Analyzing the DIP for denoising with effective degrees of freedom (DF) of a network and propose a loss based stopping criterion without ground-truth image.
- Incorporating noise regularization and exponential moving average by the proposed stochastic temporal ensembling (STE) method.
- Diverse evaluation in various metrics such as LPIPS, PSNR and SSIM in seven different datasets.
- Extending our method to Poisson noise.

2. Related work

2.1. Learning based methods

Learning-based denoising methods use a large number of clean-noisy image pairs to train a denoiser. In an early study, a neural network shows decent performance even though the noisy level is unknown, *i.e.*, blind noise setup [5]. Shortly afterwards, however, [28] has shown that most of early learning-based studies have often produced worse results than the classical technique such as BM3D [7]. But recently, DnCNN model with residual learning [51] outperforms BM3D. Then, several works are proposed to improve the computational efficiency, IRCNN [52] uses dilated convolution and FFDNet [53] uses downsampled subimages and noise level map.

2.2. Model based methods

Conventional model based methods do not need training but rely on an inductive bias given as a prior. The performance of model based methods depends on the chosen prior knowledge. There are several image priors such as total variation (TV) [34], Wavelet-domain processing [9] and BM3D [7]. Each prior assumes that the prior distribution is smoothness, low rank and self-similarity, respectively.

Image prior by deep neural networks. Ulyanov *et al*. [43] show that a randomly initialized convolutional neural network serves as an image prior and name it as deep image prior (DIP) and apply to several inverse problems.

Besides the broad usages, the performance of denoining with DIP is still disappointing because of "overfitting" to noise (see Sec. 3). There are several remedies for the noise overfitting of DIP [6, 15, 26, 42]. DIP-RED [26] combines a plug-and-play prior with DIP, which changes the converge point of DIP. GP-DIP [6] shows that DIP is asymptotically equivalent to a stationary Gaussian Process prior and introduces stochastic gradient Langevin dynamics (SGLD) [49].Deep decoder [15] utilizes underparameterized network based on the fact that overfitting is related to model complexity. Inspired by that, we systematically analyze the fitting of a network but improve the performance of DIP without sacrificing the network size.

Recently, Self2Self (S2S) [30] introduces self-supervised learning based on dropout and ensembling. Owing to model uncertainty from dropout, S2S generates multiple independent denoised instance and averages the outputs for low-variance solution. It outperforms existing solutions but needs extensive iteration with very low learning rate due to dropout. In addition, there is an approach to combine SURE [40] with DIP [27] (DIP-SURE). They share similarity to our work for both use SURE but we further extend it to propose a 'stochastic temporal ensembling,' which deviates from the original SURE formulation. Please find further discussion in Sec. 4.1.

2.3. Effective degrees of freedom

Effective degrees of freedom (DF) [10, 41] provides a quantitative analysis of the amount of fitting of a model to the training data. Efron shows that an estimate of optimism is difference of error on test and training data and relates it to a measure of model complexity deemed effective degrees of freedom [10]. Intuitively, it reflects the effective number of parameters used by a model in producing the fitted output [41]. We use the notion of DF to analyze and detect the overfitting of a network and propose our method.

2.4. Stein's unbiased risk estimator (SURE)

Stein's unbiased risk estimator [40] is a risk estimator for a Gaussian random variable. It is a useful tool for selecting a model or hyper-parameters in denoising problem, since it guarantees unbiasedness for risk estimator without a target vector [8, 50]. The analytic solution for SURE is only available for limited conditions; non-local mean or linear filter [44, 45]. When the closed form solution is not available, Ramani *et al*. [32] proposed a Monte Carlo-based SURE (MC-SURE) method to determine near-optimal parameters based on the brute-force search of the parameter space. As the SURE based method is limited to Gaussian noise [38], several works extend it to other types of noises including Poisson [24], Poisson-Gaussian [20], exponential family [11] or non-parametric noise model [35]. We also modify our objective to extend our method to Poisson noise by [20, 24, 37] (Sec. 4.3).

3. Preliminaries

Deep image prior (DIP). Let a noisy image $\mathbf{y} \in \mathbb{R}^N$ be modeled as

$$\mathbf{y} = \mathbf{x} + \mathbf{n}, \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^N$ be a noiseless image that one would like to recover and $\mathbf{n} \in \mathbb{R}^N$ be an $i.i.d.$ Gaussian noise such that $\mathbf{n} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$ where $\mathbf{I}$ is an identity matrix. Denoising can be formulated as a problem of predicting the unknown $\mathbf{x}$ from known noisy observation $\mathbf{y}$. Ulyanov *et al*. [43] argued that a network architecture naturally encourages to restore the original image from a degraded image $\mathbf{y}$ and name it as deep image prior (DIP). Specifically, DIP optimizes a convolutional neural network $\mathbf{h}$ with parameter $\boldsymbol{\theta}$ by a simple least square loss $\mathcal{L}$ as:

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \mathcal{L}(\mathbf{h}(\dot{\mathbf{n}}; \boldsymbol{\theta}), \mathbf{y}), \tag{2}$$

where $\dot{\mathbf{n}}$ is a random variable that is independent of $\mathbf{y}$. If $\mathbf{h}$ has enough capacity (*i.e*., sufficiently large number of parameters or architecture size) to fit to the noisy image $\mathbf{y}$, the output of model $\mathbf{h}(\dot{\mathbf{n}}; \hat{\boldsymbol{\theta}})$ should be equal to $\mathbf{y}$, which is *not* desirable. DIP uses the early stopping to obtained the results with best PSNR with clean images.

Effective degrees of freedom for DIP. The effective degrees of freedom [10, 41] quantifies the amount of fitting of a model to training data. We analyze the training of DIP by the effective degrees of freedom (DF) in Eq. 3 as a tool for monitoring overfitting to the given noisy image. the DF for the estimator $\mathbf{h}(\cdot)$ of $\mathbf{x}$ with input $\mathbf{y}$ can be defined as follows [14]:

$$\mathrm{DF}(\mathbf{h}) = \frac{1}{\sigma^2} \sum_{i=1}^{n} Cov(\mathbf{h}_i(\cdot), \mathbf{y}_i), \tag{3}$$

where $\mathbf{h}(\cdot)$ and $\mathbf{y}$ are a model (*e.g*., a neural network) and noise image respectively. σ is the standard deviation of the noise. $\mathbf{h}_i(\cdot)$ and $\mathbf{y}_i$ indicate the i^{th} element of corresponding vectors. For example, if the input to $\mathbf{h}(\cdot)$ is $\dot{\mathbf{n}}$ and $\mathbf{y}$ is a noisy image, *i.e*., $\mathbf{h}(\dot{\mathbf{n}})$, it is the DF for DIP. Note that $\mathbf{h}(\cdot)$ can take any input and we use $\mathbf{y}$ (instead of $\dot{\mathbf{n}}$) for our formulation.

Interestingly, the DF is closely related to the notion of *optimism* of an estimator $\mathbf{h}$, which is defined by the difference between test error and train error [14, 41] as:

$$\rho(\mathbf{h}) = \mathbb{E}\left[\mathcal{L}(\tilde{\mathbf{y}}, \mathbf{h}(\cdot)) - \mathcal{L}(\mathbf{y}, \mathbf{h}(\cdot))\right], \tag{4}$$

where $\mathcal{L}(\cdot)$ is a mean squared error (MSE) loss, $\tilde{\mathbf{y}}$ is another realization from the model (*i.e*., with different $\mathbf{n}$ in Eq. 1) that is independent of $\mathbf{y}$. In [41], it is shown that $\rho(\mathbf{h}) = 2\sum_{i=1}^{n} Cov(\mathbf{h}_i(\cdot), \mathbf{y}_i)$. Thus, combining with Eq. 3, it is straightforward to show that

$$2\sigma^2 \cdot \mathrm{DF}(\mathbf{h}) = \rho(\mathbf{h}). \tag{5}$$

It is challenging to compute the covariance since $\mathbf{h}(\cdot)$ is nonlinear (*e.g*., a neural network), gradually changing in optimization, and the $\rho(\mathbf{h})$ requires many pairs of noisy and clean (ground-truth) images to compute (note that it is an estimate). Here, we introduce a simple approximated degrees of freedom with a single ground-truth and call it as DF_{GT}. We derive the DF_{GT} as following:

$$2\sigma^2 \cdot \mathrm{DF}_{GT}(\mathbf{h}) \approx \mathcal{L}(\mathbf{x}, \mathbf{h}(\cdot)) - \mathcal{L}(\mathbf{y}, \mathbf{h}(\cdot)) + \sigma^2 \tag{6}$$

We describe a simple proof of the estimation in the supplementary material.

A large DF implies overfitting to the given input $\mathbf{y}$, which is not desirable. If DIP fits to $\mathbf{x}$, DF_{GT} becomes close to 0. The more the DIP is fitting to $\mathbf{y}$, the larger the DF is. We use the DF_{GT} to analyze the DIP optimization in empirical studies in Sec. 5.1.

4. Approach

To prevent the overfitting of DIP, we try to suppress the DF (Eq. 3) during the optimization without the access of ground-truth clean image $\mathbf{x}$. In Eq. 3, computing the DF is equivalent to the sum of the covariances for each element of the noise image $\mathbf{y}$ and the model output $\mathbf{h}(\cdot)$. There are a number of techniques to simply approximate the covariance computation in statistical learning literature such as AIC [1], BIC [36] and Stein's unbiased risk estimator (SURE) [40]. Both AIC and BIC, however, approximate the DF by counting the number of parameters of a model, so for usual over-parameterized deep neural networks, the approximations based on them could be incorrect [3]. Note that DF_{GT} cannot be used for optimizing model because it needs groud-truth clean image $\mathbf{x}$.

Here, we propose to use SURE to suppress the DF by deriving the DIP formulation using the Stein's lemma. The

Stein's lemma for a multivariate Gaussian vector $\mathbf{y}$ is [40]:

$$\frac{1}{\sigma^2}\sum_{i=1}^{n} Cov(\mathbf{h}_i(\mathbf{y}), \mathbf{y}_i) = \mathbb{E}\left[\sum_{i=1}^{n}\frac{\partial \mathbf{h}_i(\mathbf{y})}{\partial \mathbf{y}_i}\right]. \quad (7)$$

It simplifies the computation of DF from the covariances between $\mathbf{y}$ and $\mathbf{h}(\mathbf{y})$ to the expected partial derivatives at each point, which is well approximated in a number of computationally efficient ways [32, 39]. Note that the SURE which is denoted as $\eta(\mathbf{h}(\mathbf{y}), \mathbf{y})$, consists of Eq. 7 and the DIP loss (Eq. 2) with a modification of its input (from $\dot{\mathbf{n}}$ to $\mathbf{y}$) as:

$$\eta(\mathbf{h}(\mathbf{y}), \mathbf{y}) = \mathcal{L}(\mathbf{y}, \mathbf{h}(\mathbf{y})) + \underbrace{\frac{2\sigma^2}{N}\sum_{i=1}^{N}\frac{\partial \mathbf{h}_i(\mathbf{y})}{\partial(\mathbf{y})_i}}_{\text{divergence term}} - \sigma^2. \quad (8)$$

While the vanilla DIP loss encourages to fit the output of the model $\mathbf{h}$ to noisy image $\mathbf{y}$, Eq. (8) encourages to approximately fit it to clean image $\mathbf{x}$ without access to the $\mathbf{x}$.

However, it is still computationally demanding to use Eq. 8 as a loss for optimization with any gradient based algorithm due to the divergence term [32]. A Monte-Carlo approximation for Eq. 8 in [32] can be a remedy to the computation cost, but it introduces a hyper-parameter ϵ that has to be selected properly for the best performance on different network architectures and/or datasets. For not requiring to tune the hyper-parameter ϵ, we employed an alternative Monte-Carlo approximation for the divergence term [39] as:

$$\frac{1}{N}\sum_{i=1}^{N}\frac{\partial \mathbf{h}_i(\mathbf{y})}{\partial \mathbf{y}_i} \approx \frac{1}{N}\tilde{\mathbf{n}}^T\mathbf{J}_{\tilde{\mathbf{n}}^T\mathbf{h}(\mathbf{y})}, \quad (9)$$

where $\tilde{\mathbf{n}}$ is a standard normal random vector, *i.e.*, $\tilde{\mathbf{n}} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and the i^{th} element of the Jacobian $\mathbf{J}_{\tilde{\mathbf{n}}^T\mathbf{h}(\mathbf{y})}$ is $\partial\tilde{\mathbf{n}}^T\mathbf{h}(\mathbf{y};\theta)/\partial\mathbf{y}_i$. We denote this 'estimated degrees of freedom by Monte-Carlo' by DF_{MC} and will use it to monitor the DIP optimization without using the PSNR with the clean ground truth images (Sec. 4.2).

4.1. Stochastic temporal ensembling

To improve the fitting accuracy, DIP suggests several methods including noise regularization, exponential moving average [43]. We propose 'stochastic temporal ensembling (STE)' for better fitting performance by leveraging these methods to our objective.

Noise regularization on DIP. DIP shows that adding extra temporal noises to the input $\dot{\mathbf{n}}$ of function $\mathbf{h}(\cdot)$ at each iteration improves performance for the inverse problems including image denoising [43]. It is to add a noise vector γ, with $\gamma \sim N(0, \sigma_\gamma^2\mathbf{I})$ to the input of the function at every iteration of the optimization as:

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \mathcal{L}(\mathbf{h}(\dot{\mathbf{n}} + \gamma; \boldsymbol{\theta}), \mathbf{y}), \quad (10)$$

where $\dot{\mathbf{n}}$ is fixed but γ is sampled from Gaussian distribution with zero mean, standard deviation of σ_γ at every iteration. To estimate the $\mathbf{x}$ by Eq. 8, we replace the input of the model $\mathbf{h}(\cdot)$, $\dot{\mathbf{n}}$, with noisy image $\mathbf{y}$ (from Eq. 3 to Eq. 7). Interestingly, Eq. 10 becomes similar to the denoising auto-encoder (DAE), which prevents a model from learning a trivial solution by perturbing output of $\mathbf{h}$ [46].

Meanwhile, contractive autoencoder (CAE) [33] minimizes the Frobenius norm of the Jacobian and SURE and its variants minimize the trace of the Jacobian (Eq. 9) thus suppresses the DF. Since we assume that the different realizations of noise are independent, the off-diagonal elements of the matrix are zero, CAE is equivalent to SURE in terms of suppressing the DF. Alain *et al.* [2] later show that the DAE is a special case of the CAE when $\sigma_\gamma \to 0$. We can rewrite the Eq. 10 by using CAE formulation as:

$$\arg\min_{\boldsymbol{\theta}} \mathcal{L}(\mathbf{h}(\mathbf{y};\boldsymbol{\theta}), \mathbf{y}) + \sigma_\gamma^2\left\|\frac{\partial \mathbf{h}(\mathbf{y};\boldsymbol{\theta})}{\partial \mathbf{y}}\right\|_F^2 + o(\sigma_\gamma^2), \quad (11)$$

when $\sigma_\gamma \to 0$, where $o(\sigma_\gamma^2)$ is a high order error term from Taylor expansion. Thus, solving this optimization problem is equivalent to penalizing increase of DF. Here, the noise level σ_γ serves as a hyper-parameter for determining performance and it improves performance of DIP by using multiple level of σ_z at optimization of DIP. Thus, we further proposed to model σ_γ as a uniform random variable instead of a empirically chosen hyper-parameter such that

$$\sigma_\gamma \sim \mathcal{U}(0, b). \quad (12)$$

Exponential moving average. DIP further shows that averaging the restored images obtained in the last iterations improves the performance of denoising [43], which we refer to as 'exponential moving average (EMA).' It can be thought as an analogy to the effect of ensembling [31].

Stochastic temporal ensembling. Leveraging the noise regularization and the EMA, we propose a method called 'stochastic temporal ensembling (STE)' to improve the fitting performance of DIP loss. Specifically, we modify our formulation (Eq. 8) by allowing two noise observations, $\mathbf{y_1}$ for target of MSE loss and $\mathbf{y_2}$ for the input of the model, h, instead of one $\mathbf{y}$ by setting $\mathbf{y}_1 = \mathbf{y}$ and $\mathbf{y}_2 = \mathbf{y} + \gamma$ as:

$$\eta(\mathbf{h}(\mathbf{y_2}), \mathbf{y_1}) = \underbrace{\mathcal{L}(\mathbf{h}(\mathbf{y_2}), \mathbf{y_1})}_{\text{data fidelity}} + \underbrace{\frac{2\sigma^2}{N}\sum_{i=1}^{N}\frac{\partial \mathbf{h}_i(\mathbf{y_2})}{\partial(\mathbf{y_2})_i}}_{\text{regularization}} - \sigma^2, \quad (13)$$

where σ is a known noise level of $\mathbf{y_1}$ (same as Eq. 1), $\mathbf{h}_i(\mathbf{y_2})$ and $(\mathbf{y_2})_i$ are the i^{th} element of the vectors of $\mathbf{h}(\mathbf{y_2})$ and $\mathbf{y_1}$, respectively. Interestingly, Eq. 13 is equivalent to the formulation of extended SURE (eSURE) [55], which is shown to be a better unbiased estimator of the MSE with

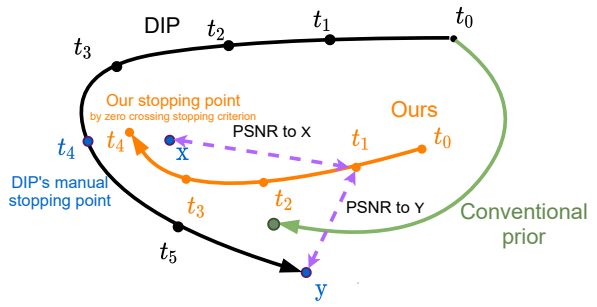


Figure 2: **Illustration of a solution trajectory of ours and DIP.** We consider the problem of reconstructing an image $\mathbf{x}$ from a degraded measurement $\mathbf{y}$. DIP finds its optimal stopping point (t_4) by early stopping. Ours changes DIP's solution trajectory from black to orange whose stopping point (t_5) is defined by a loss value (Sec.4.2) and is close to noiseless solution ($\mathbf{x}$).

the clean image $\mathbf{x}$. But there are a number of critical differences of ours from [55]. First, Our method does not require training, while Zhussip *et al*. [55] requires training with many noisy images. Because Zhussip *et al*. [55] use the fixed instance of γ, there is no effect of regularization from (Eq. 10), which gives reasonable performance gain(See Sec.5.2). This is our final objective function of DIP that stops automatically by a stopping criterion, described in the following section.

4.2. Zero-crossing stopping criterion

SURE works well if the model $\mathbf{h}$ satisfies the smoothness condition, *i.e*., $\mathbf{h}$ admits a well-defined second-order Taylor expansion [27, 38]. While a typical learning based denoiser satisfies this smoothness condition [38,55], the DIP network 'fits' to a target image (a noisy image in [42, 43] and an approximate clean image in our objective) and therefore there is no guarantee that the smoothness condition can be satisfied, especially when it has been converged.

We observed that the divergence term in our formulation (Eq. 13) increases at early iterations (*i.e*., before convergence) while it starts to diverge to $-\infty$ at later iterations (*i.e*., after convergence). This observation is consistent in all our experiments. Note that this divergence phenomenon was not reported in [27] because the DIP network with the SURE loss did not seem to be fully converged to recover the fine details with insufficient number of iterations. Based on this observation for our proposed objective, we propose 'zero crossing stopping criterion' to stop iteration when our objective function (Eq. 13) deviates from zero.

Solution trajectory. To help understand the difference between our method to DIP in optimization procedure, similar to Fig. 3 in [43], we illustrate DIP image restoration trajectory with that of our method in Fig. 2. DIP degrades the quality of the restored images by the overfitting. To obtain the solution close to the clean ground truth image, DIP uses early stopping (blue t_4). Our formulation has different training trajectory (orange) from DIP (black) and automatically stops the optimization by the zero crossing stopping (orange t_4). We argue that the resulting image by our formulation is in general closer to the clean image (blue $\mathbf{x}$) than the solution by DIP, which preserves more high frequency details than the solution by the DIP (Sec. 5.3) thanks to a better target to fit (an approximation of the clean $\mathbf{x}$ over a noisy image $\mathbf{y}$ and our proposed principled stopping criterion without using ground truth image). We empirically analyze this phenomenon with our proposed DF_{GT} and compare it to DF_{MC} in Sec. 5.1 and the supplementary material.

4.3. Extension to Poisson noise

As the SURE is limited to Gaussian noise [38], there are several attempts to extend it to other types of noises [11, 20, 35]. Here, we extend our formulation to Poisson noise as it is a useful model for noise in low-light condition. We modify our formulation (Eq. 13) to use Poisson unbiased risk estimator (PURE) [17, 20, 24] for Poisson noise as follows:

$$\mathcal{L}(\mathbf{h}(\mathbf{y}),\mathbf{y}) - \frac{\zeta}{N}\sum_{i=1}^{N}\mathbf{y}_i + \frac{2\zeta}{\dot{\epsilon}N}(\check{\mathbf{n}}\odot\mathbf{y})^{\mathbf{T}}(\mathbf{h}(\mathbf{y}+\dot{\epsilon}\check{\mathbf{n}}) - \mathbf{h}(\mathbf{y}))), \quad (14)$$

where $\check{\mathbf{n}}$ is a k-dimensional binary random variable whose element $\check{n}_i$ takes -1 or 1 with probability 0.5 for each, $\dot{\epsilon}$ is a small positive number and $\odot$ is a Hadamard product. We empirically validate the Poisson extension in Sec. 5.4.

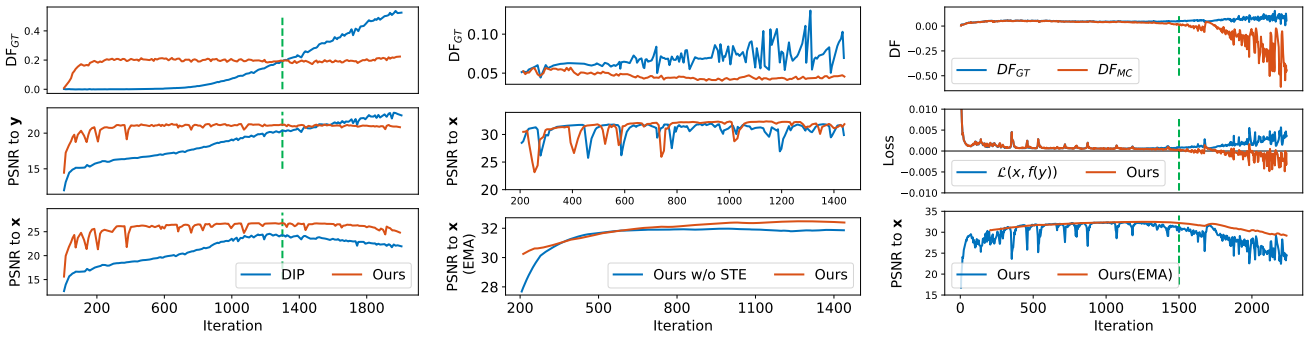
5. Experiments

Implementation details. For the σ, we use $\sigma = 15, 25, 50$, following [51], and $\sigma = 25$ for in-depth analysis. For the b in Eq. 12, we set it to the same value to the σ. RAdam opimizer [23] is used for training with learning rate 0.1. Details including network architectures and datasets are in the supplementary material.

Evaluation metrics. We use peak signal-to-noise ratio (PSNR), structured similarity (SSIM) and learned perceptual image patch similarity (LPIPS) [54]. The PSNR is widely used in denoising literature [16, 29, 51–53] but is recently argued that it is not an ideal metric as it values the oversmoothed results [21, 54]. For this reason, we compare the algorithms with LPIPS as an alternative measurement of human study. We use the publicly available pre-trained weights based on AlexNet by the authors [54]. We additionally report the performance of the peak PSNR during optimization of our method as a reference (denoted as 'Ours*').

5.1. Convergence analysis by DF_{GT}

Fig. 3a shows the DF_{GT}, PSNR to y and PSNR to x; DF_{GT} is the effective degrees of freedom with Ground Trues, PSNR to y and PSNR to x refer PSNR from the model output to y and x respectively. As optimization progresses, the degrees of freedom of DIP increases gradually



(a) Convergence analysis (b) Effect of the STE (c) Zero-crossing stopping criterion

Figure 3: **Learning analysis with DF$_{GT}$**. (a) As optimization progresses, degrees of freedom of DIP increase as fit to noisy observation. Ours does not overfit to noisy observation and shows consistently better performance thanks to Stein unbiased risk estimation. The green dashed line indicates the intersection between DIP and ours in DF$_{GT}$. (b) The proposed method makes optimization more stable than single instance one and this tendency also is observed on PSNR to $\mathbf{x}$, PSNR to $\mathbf{x}$(EMA). (c) Monte-carlo estimation DF$_{MC}$ of eSURE is stable for a considerable amount of steps but the error of estimation is soaring. It usually happens when the loss is already close to zero (the green dashed line on the plot). Thus, we propose to stop the optimization as soon as the loss reaches zero.

with PSNR to $\mathbf{x}$. But PSNR to $\mathbf{x}$ of DIP decreases from 1,300 iteration. In contrast, in ours, DF$_{GT}$ rises at the beginning of the iterations and stays at a certain value. Interestingly, the best stopping point for DIP is near the intersection between DIP and our method in DF$_{GT}$. It implies that the converged DF$_{GT}$ value by our method is near the optimal solution of DIP (t_4 of DIP in Fig. 2).

Fig. 3b shows the trajectory of two objectives on DF$_{GT}$; (1) ours w/o STE and (2) ours. As shown in Sec 4.1, STE suppresses the DF by minimizing the norm of the Jacobian, which is similar to trace of Jacobian (the DF from the Stein's lemma). Accordingly, ours suppresses the DF better than ours w/o STE in DF$_{GT}$ (Fig. 3b (top)). This tendency is also observed in 'PSNR to $\mathbf{x}$' and 'PSNR to $\mathbf{x}$ (EMA).'

The optimization progresses of DF_{GT} and DF_{MC} are shown in Fig. 3c. The DF_{MC} starts underestimating the DF_{GT} after a certain iteration and Eq. 13 ('Loss') becomes zero and it reaches the highest PSNR value. Thus, our proposed zero stopping criterion detects when DF_{MC} fails to estimate the DF_{GT}; when the loss crosses zero.

5.2. Quantitative analysis

Comparison to DIP variants. Table 1 shows the denoising results of several DIP based methods. Deep decoder (DD) [15] shows the worst performance in all metrics. We believe that DD mitigates overfitting problem with under-parameterized network in return for its performance. GP-DIP [6] outperforms DIP in PSNR and SSIM. It uses SGLD [49] to sample multiple instances of posterior distribution and average them which is similar to Self2Self [30]. This strategy may be useful for PSNR score but it may lose the texture of images, which leads to relatively low LPIPS score (see next section for more discussions). DIP-RED shows the best result apart from our method and its ablated version. Its plug-and-play overfitting prevention uses other denoising method as prior. Plug-and-play method might work with our method but it is beyond the scope of this paper. Note that all above methods except ours, DIP-SURE* and DIP stops optimization at the predefined number of iterations provided in the authors' codes.

Method	Overfit Prev.	PSNR (↑)	SSIM (↑)	LPIPS (↓)
DIP [43]	Early stopping	29.96	0.940	0.152
Deep Decoder [15]	Under-param.	26.94	0.889	0.377
DIP-RED [26]	Plug-and-play	30.88	0.932	0.197
GP-DIP [6]	SGLD	29.99	0.948	0.251
DIP-SURE* [27]	ZCSC	30.33	0.941	0.149
Ours w/o STE [55]	ZCSC	<u>31.34</u>	**0.955**	<u>0.108</u>
Ours	ZCSC	**31.54**	<u>0.953</u>	**0.107**

Table 1: **Comparison to DIP variants on CSet9 dataset** ($\sigma = 25$)**.** (↑): higher the better, (↓): lower the better. 'Overfit Prev.' refers to 'overfitting prevention method.' 'Under-param.' refers to 'under parameterized.' (Best values: in **bold**. The second best values: underlined). 'ZCSC' refers to the proposed zero crossing stopping criterion. 'DIP-SURE*' refers to [27] with ZCSC for fair comparison among the methods using the SURE formulation.

In particular, both DIP-SURE* [27] and 'Ours w/o STE' [55] are worse than ours even though they use SURE formulation. We argue that it is because they use a single noise realization. In addition, they are quite similar each other except DIP-SURE* depends on the ϵ as a hyperparameter while 'Ours w/o STE' does not have such hyperparameter (Sec. 4). The no need of hyperparameter tuning results in a noticeable gain by 'Ours w/o STE.' Note that original DIP-SURE depends on early stopping by monitoring PSNR with a clean image. For fair comparison, we use our stopping criterion to it and notate it as DIP-SURE*.

Comparison to the state of the arts. Table 2 shows comparative results with other single image denoisning methods in six datasets (four color and two gray-scale). The comparing methods includes CBM3D [7], DIP [42], Self2Self (S2S) [30]. Except for BM3D, all remaining methods are

Dataset	σ	PSNR (↑) BM3D [7]	DIP [43]	S2S [30]	Ours (Ours*)	SSIM (↑) BM3D [7]	DIP [43]	S2S [30]	Ours (Ours*)	LPIPS (↓) BM3D [7]	DIP [43]	S2S [30]	Ours (Ours*)
Color Image Datasets													
CSet9	15	33.83	31.83	33.24	33.83 (**34.07**)	0.972	0.960	0.968	0.973 (**0.975**)	0.111	0.114	0.135	**0.070** (0.077)
	25	31.68	29.96	31.72	31.54 (**31.88**)	0.956	0.940	0.956	0.953 (**0.960**)	0.161	0.152	0.173	**0.107** (0.118)
	50	28.92	27.42	**29.25**	28.90(29.03)	0.922	0.900	0.928	0.923 (**0.930**)	0.267	0.291	0.235	**0.181** (0.200)
CBSD68	15	33.51	31.48	32.78	33.43 (**33.56**)	0.961	0.941	0.956	0.961 (**0.963**)	0.081	0.081	0.102	0.060 (**0.057**)
	25	30.70	28.66	30.67	30.67 (**30.86**)	0.932	0.900	0.932	0.932 (**0.936**)	0.148	0.156	0.147	0.102 (**0.100**)
	50	27.37	25.70	**27.62**	27.43 (27.58)	0.871	0.832	0.879	0.873 (**0.881**)	0.298	0.329	0.244	**0.194** (0.198)
Kodak	15	34.41	32.17	33.70	34.35 (**34.49**)	0.962	0.941	0.958	0.961 (**0.963**)	0.104	0.105	0.118	0.080 (**0.077**)
	25	31.82	29.68	31.79	31.60 (**31.98**)	0.938	0.907	0.939	0.932 (**0.941**)	0.161	0.173	0.159	**0.117** (0.118)
	50	28.62	26.77	**29.08**	28.58 (28.76)	0.886	0.843	0.898	0.882 (**0.892**)	0.287	0.338	0.235	**0.203** (0.209)
McM	15	34.05	32.54	33.92	34.13 (**34.35**)	0.969	0.956	0.968	0.967 (**0.970**)	0.068	0.067	0.089	0.053 (**0.052**)
	25	31.66	30.09	**32.15**	31.89 (31.98)	0.950	0.929	**0.955**	0.950 (0.953)	0.107	0.123	0.117	**0.085** (**0.085**)
	50	28.51	27.06	**29.29**	28.83 (28.82)	0.910	0.882	**0.924**	0.913 (0.918)	0.207	0.252	0.178	**0.151** (0.162)
Gray-scale Image Datasets													
BSD68	15	31.07	28.83	30.62	30.98 (**31.21**)	0.872	0.812	0.858	0.873 (**0.882**)	0.147	0.163	0.163	**0.090** (0.099)
	25	28.57	26.59	28.60	28.40 (**28.78**)	0.801	0.734	0.801	0.800 (**0.818**)	0.226	0.262	0.197	**0.157** (0.159)
	50	25.61	24.13	25.70	25.75 (**25.81**)	0.686	0.625	0.687	0.696 (**0.708**)	0.363	0.443	0.313	**0.262** (0.282)
Set12	15	**32.36**	30.12	32.07	32.20 (32.26)	**0.895**	0.837	0.889	0.891 (0.894)	0.117	0.132	0.139	**0.084** (0.092)
	25	29.93	27.54	**30.02**	29.79 (29.76)	**0.850**	0.776	0.849	0.844 (0.848)	0.159	0.218	0.159	**0.122** (0.137)
	50	**26.71**	24.67	26.49	26.60 (26.47)	**0.768**	0.683	0.734	0.755 (0.760)	0.262	0.361	0.232	**0.208** (0.228)

Table 2: **Comparison to the state of the arts on single-image denoising algorithm.** (↑) denotes that higher is better and (↓) denotes the lower is better. Best performance is in bold. Second best is underlined. For DIP, we report the peak PSNR scores during the optimization.

based on convolutional neural network. For the network architecture for DIP, we use the same one to ours for fair comparison. We use slightly difference network for S2S since it needs the dropouts instead of batch normalization.

Our method outperforms all other single-image denoising methods in LPIPS, showing comparable PSNR and SSIM. Ours* exhibits best PSNR performance outperforming all comparing methods except S2S in high noise experiments. But Ours* loses some high frequency details to ours (see LPIPS). We believe that it is due to the exponential moving average (EMA) as it alleviates the instability of training (*i.e.*, rough solution space) that cannot be caught by PSNR. Ours performs well especially at low noise. We believe that the error by the MC estimation is smaller in the small noise set-up. Nevertheless, our method exhibits excellent performance in LPIPS and SSIM in almost all setups.

It is worth noting that S2S exhibits high LPIPS, especially in low noise ($\sigma = 15$) than all other methods despite being ahead of DIP in PSNR. Considering that MSE is the sum of squared bias and variance, we argue that the S2S achieves impressive PSNR results with significantly reduced variance and increased squared bias (*i.e.*, destroying textural details). It is clearly observed in Fig. 1-(b), where we show the results of S2S with various number of ensembles. As the number of ensembles increases, the PSNR also increases in return of the loss of LPIPS score. In contrast, our method achieves much better trade-off between LPIPS score and PSNR without ensembling.

Moreover, inference time of S2S on CSet9 is almost 35 hours without parallel processing whereas ours only takes 4 hours. Further speed up of S2S and ours are possible by parallel processing [30] but the gap would be maintained.

Although it is not quite fair to compare our method with learning-based ones including DnCNN [51], N2N [22], HQ-N2V [19], IRCNN [52] as we only use a single noisy observation, we additionally compare with them in the supplementary material for the space sake.

Noise scale	BM3D-VST [25]	DIP [43]	S2S [30]	Ours (Ours*)
$\zeta = 0.01$	30.50	30.99	**32.18**	32.00 (31.94)
$\zeta = 0.1$	21.57	23.54	22.84	24.87 (**24.94**)
$\zeta = 0.2$	18.48	21.43	20.10	22.85 (**22.90**)

Table 3: **Comparison to the state of the arts on Poisson noise (PSNR (dB)).** Best performance: bold. Second best: underlined.

5.3. Qualitative analysis

We present examples of denoised images in Fig. 4. In the first row, we observe that the results of CBM3D and S2S are over smoothed (having less high frequency details) than those by our method. DIP preserves textures but is much noisier than ours. Again, we observe that our results are in better trade-off between PSNR and LPIPS.

The second rows has higher noise level ($\sigma = 50$) than the first row. S2S and CBM3D show clean images with sharp edges. But they also make the English characters in the sign blurry. In contrast, our method preserves sharper details in the character in the sign while noises are mostly suppressed. More qualitative results are in supplement.

5.4. Extension to Poisson noise

Poisson noise is likely to occur in low light condition such as microscopic imaging. In [37], they use MNIST images for simulating this scenario. We conduct experiments of single-image Poisson denoising, and summarize the comparative results with BM3D-VST [25], DIP, and S2S in Table 3. Note that BM3D-VST is one of the most popular

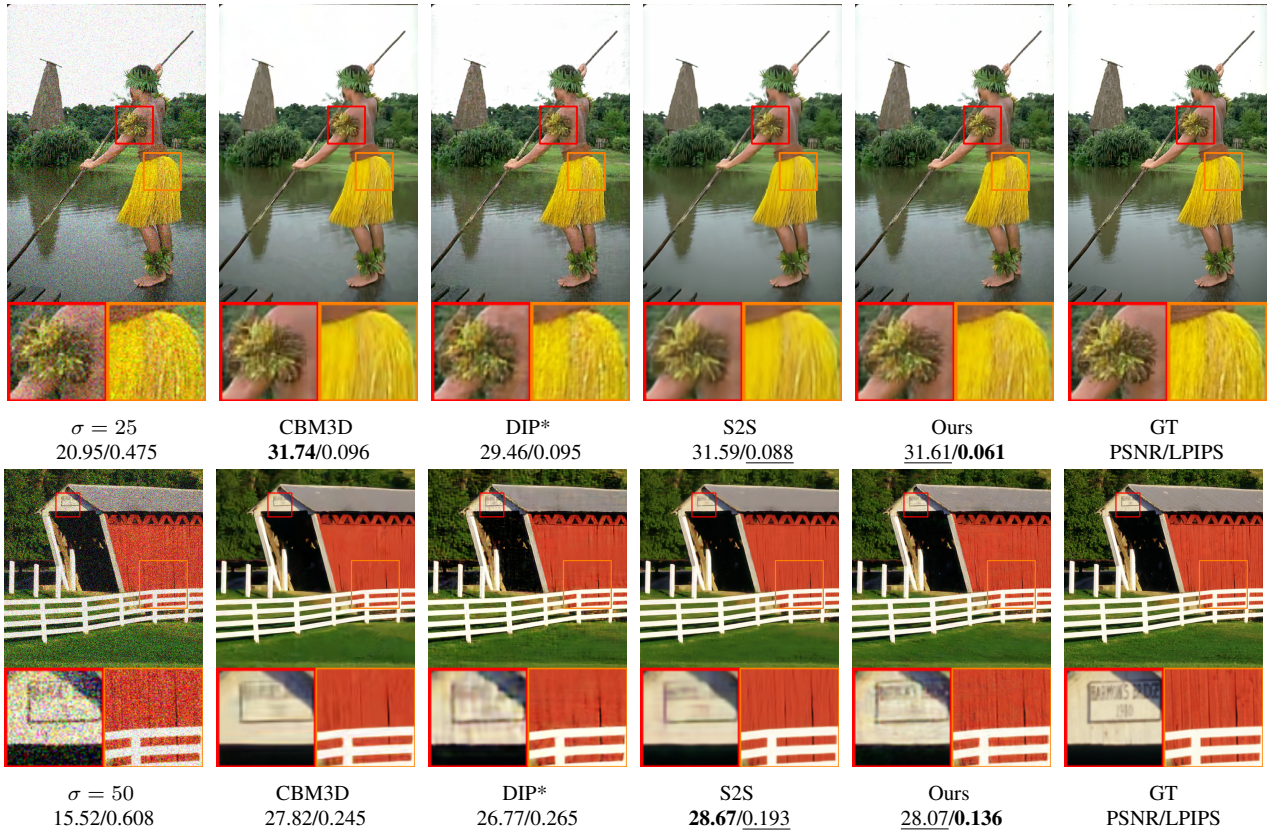


Figure 4: **Qualitative comparisons.** Best performance: bold. Second best: underlined. More results are in the supplement.

Figure 5: **Qualitative comparison on Poisson noise** ($\zeta = 0.2$)**.**

methods for Poisson denoising.

For low noise level ($\zeta = 0.01$), noise distribution becomes almost symmetric similar to Gaussian. So, our method does not perform well. But at higher level of noise, our method outperforms other methods. DIP shows better results than classic methods such as BM3D with VST [25], and the state of the art, S2S, in the higher noise. Our method outperforms all compared methods including the classic methods with VST (BM3D-VST) [25].

Fig. 5 shows the qualitative result of the Poisson noise setup. We observe that BM3D+VST images were considerably blurrier than other methods, and S2S also produce blurry image due to overfitting. DIP shows the second best result thanks to early stopping. In contrast, our method denoises holes in the images with detailed texture preserved without early stopping.

6. Conclusion

We investigate DIP for denoising by the notion of effective degrees of freedom to monitor the overfitting to noise and propose stochastic temporal ensembling (STE) and zero crossing stopping criterion to stop the optimization before it overfits without a clean image. We significantly improve the performance of Gaussian denoising by DIP without the manual early stopping and extend the method to Poisson denoising with PURE. Our empirical validation shows that the proposed method outperforms state-of-the-arts in LPIPS by large margins with comparable PSNR and SSIM, evaluated with the seven different datasets.

Acknowledgement. This work was partly supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No.2019R1C1C1009283) and Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2019-0-01842, Artificial Intelligence Graduate School Program (GIST)), (No.2019-0-01351, Development of Ultra Low-Power Mobile Deep Learning Semiconductor With Compression/Decompression of Activation/Kernel Data, 17%), (No. 2021-0-02068, Artificial Intelligence Innovation Hub) and was conducted by Center for Applied Research in Artificial Intelligence (CARAI) grant funded by DAPA and ADD (UD190031RD). The work of SY Chun was supported by Basic Science Research Program through National Research Foundation of Korea (NRF) funded by Ministry of Education (NRF-2017R1D1A1B05035810).

References

[1] Hirotogu Akaike. Ieee int symp info. In *Selected Papers of Hirotugu Akaike*. Springer New York, 1973. 3

[2] Guillaume Alain and Yoshua Bengio. What regularized auto-encoders learn from the data-generating distribution. *JMLR*, 2014. 4

[3] Ulrich Anders and Olaf Korn. Model selection in neural networks. *Neural Netw.*, 1999. 3

[4] Y. Blau and T. Michaeli. The perception-distortion tradeoff. In *CVPR*, 2018. 2

[5] HC. Burger, CJ. Schuler, and S. Harmeling. Image denoising: Can plain neural networks compete with bm3d? In *CVPR*, 2012. 1, 2

[6] Zezhou Cheng, Matheus Gadelha, Subhransu Maji, and Daniel Sheldon. A bayesian perspective on the deep image prior. In *CVPR*, 2019. 2, 6

[7] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image denoising by sparse 3d transform-domain collaborative filtering. *IEEE TIP*, 2007. 1, 2, 6, 7, 8

[8] David Donoho and Iain M. Johnstone. Adapting to unknown smoothness via wavelet shrinkage. *Journal of the American Statistical Association*, 1995. 3

[9] D. L. Donoho. De-noising by soft-thresholding. *IEEE Trans. Inf. Theory*, 1995. 2

[10] Bradley Efron. The estimation of prediction error. *Journal of the American Statistical Association*, 2004. 1, 2, 3

[11] Y. C. Eldar. Generalized sure for exponential families: Applications to regularization. *IEEE Transactions on Signal Processing*, 2009. 3, 5

[12] Tianxiang Gao and Vladimir Jojic. Degrees of freedom in deep neural networks. In *UAI*, 2016. 1

[13] Shuhang Gu and R. Timofte. A brief review of image denoising algorithms and beyond. In *VCIBA*. 2

[14] Trevor Hastie and Robert Tibshirani. Generalized additive models. *statistical science*, 1986. 3

[15] Reinhard Heckel and Paul Hand. Deep decoder: Concise image representations from untrained non-convolutional networks. In *ICLR*, 2019. 1, 2, 6

[16] Xixi Jia, Sanyang Liu, Xiangchu Feng, and Lei Zhang. Focnet: A fractional optimal control network for image denoising. In *CVPR*, 2019. 5

[17] K. Kim, S. Soltanayev, and S. Y. Chun. Unsupervised training of denoisers for low-dose ct reconstruction without full-dose ground truth. *IEEE JSTSP*, 2020. 5

[18] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, 2012. 1

[19] Samuli Laine, Tero Karras, Jaakko Lehtinen, and Timo Aila. High-quality self-supervised deep image denoising. In *Neurips*, 2019. 7

[20] Y. Le Montagner, E. D. Angelini, and J. Olivo-Marin. An unbiased risk estimator for image denoising in the presence of mixed poisson–gaussian noise. *IEEE TIP*, 2014. 2, 3, 5

[21] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. Photo-realistic single image super-resolution using a generative adversarial network. In *CVPR*, 2017. 2, 5

[22] Jaakko Lehtinen, Jacob Munkberg, Jon Hasselgren, Samuli Laine, Tero Karras, Miika Aittala, and Timo Aila. Noise2Noise: Learning image restoration without clean data. In *ICML*, 2018. 7

[23] Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. On the variance of the adaptive learning rate and beyond. In *ICLR*, 2020. 5

[24] F. Luisier, T. Blu, and M. Unser. Image denoising in mixed poisson–gaussian noise. *IEEE TIP*, 2011. 2, 3, 5

[25] M. Makitalo and A. Foi. Optimal inversion of the generalized anscombe transformation for poisson-gaussian noise. *IEEE TIP*, 2013. 7, 8

[26] Gary Mataev, Peyman Milanfar, and Michael Elad. Deepred: Deep image prior powered by red. In *ICCV Workshops*, 2019. 2, 6

[27] Christopher A. Metzler, Ali Mousavi, Reinhard Heckel, and Richard G. Baraniuk. Unsupervised learning with stein's unbiased risk estimator. In *BASP Workshops*, 2019. 2, 5, 6

[28] Tobias Plotz and Stefan Roth. Benchmarking denoising algorithms with real photographs. In *CVPR*, 2017. 2

[29] Tobias Plötz and Stefan Roth. Neural nearest neighbors networks. In *NeurIPS*, 2018. 5

[30] Yuhui Quan, Mingqin Chen, Tongyao Pang, and Hui Ji. Self2self with dropout: Learning self-supervised denoising from single image. In *CVPR*, 2020. 1, 2, 6, 7, 8

[31] I. Radosavovic, P. Dollár, R. Girshick, G. Gkioxari, and K. He. Data distillation: Towards omni-supervised learning. In *CVPR*, 2018. 4

[32] S. Ramani, T. Blu, and M. Unser. Monte-carlo sure: A black-box optimization of regularization parameters for general denoising algorithms. *TIP*, 2008. 3, 4

[33] Salah Rifai, Pascal Vincent, Xavier Muller, Xavier Glorot, and Yoshua Bengio. Contractive auto-encoders: Explicit invariance during feature extraction. In *ICML*, 2011. 4

[34] Leonid I. Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. In *11th CNLS*, 1992. 2

[35] B. Scholkopf, J. Platt, and T. Hofmann. Learning to be bayesian without supervision. In *NeurIPS*, 2007. 3, 5

[36] Gideon Schwarz. Estimating the Dimension of a Model. *The Annals of Statistics*, 1978. 3

[37] Shakarim Soltanayev and Se Young Chun. Training and refining deep learning based denoisers without ground truth data. *arXiv*, 2018. 3, 7

[38] Shakarim Soltanayev and Se Young Chun. Training deep learning based denoisers without ground truth data. In *NeurIPS*, 2018. 3, 5

[39] S. Soltanayev, R. Giryes, S. Y. Chun, and Y. C. Eldar. On divergence approximations for unsupervised training of deep denoisers based on stein's unbiased risk estimator. In *ICASSP*, 2020. 4

[40] C M Stein. Estimation of the mean of a multivariate normal distribution. *The Annals of Statistics*, 1981. 2, 3, 4

[41] Ryan Tibshirani. Degrees of freedom and model search. *Statistica Sinica*, 2014. 1, 2, 3

[42] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep image prior. In *CVPR*, 2018. 1, 2, 5, 6

[43] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep Image Prior. *IJCV*, 2020. 1, 2, 3, 4, 5, 6, 7, 8

[44] D. Van De Ville and M. Kocher. Sure-based non-local means. *IEEE Signal Processing Letters*, 2009. 3

[45] D. Van De Ville and M. Kocher. Nonlocal means with dimensionality reduction and sure-based parameter selection. *IEEE TIP*, 2011. 3

[46] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *ICML*, 2008. 4

[47] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *JMLR*, 2010. 1

[48] Y. Wang and J. Morel. Can a single image denoising neural network handle all levels of gaussian noise? *IEEE SP Letters*, 2014. 1

[49] Max Welling and Yee Whye Teh. Bayesian learning via stochastic gradient langevin dynamics. In *ICML*, 2011. 2, 6

[50] Xiao-Ping Zhang and M. D. Desai. Adaptive denoising based on sure risk. *IEEE SP Letters*, 1998. 3

[51] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising. *IEEE TIP*, 2017. 1, 2, 5, 7

[52] Kai Zhang, Wangmeng Zuo, Shuhang Gu, and Lei Zhang. Learning deep cnn denoiser prior for image restoration. In *CVPR*, 2017. 2, 5, 7

[53] Kai Zhang, Wangmeng Zuo, and Lei Zhang. Ffdnet: Toward a fast and flexible solution for CNN based image denoising. *IEEE TIP*, 2018. 2, 5

[54] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 2, 5

[55] Magauiya Zhussip, Shakarim Soltanayev, and Se Young Chun. Extending stein's unbiased risk estimator to train deep denoisers with correlated pairs of noisy images. In *Neurips*, 2019. 4, 5, 6