236860 Digital Image Processing - Final Project

Paper: SNIPS - Solving Noisy Inverse Problems Stochastically[15]

Tom Dana and Fayrouz Azem

Abstract

The paper presents a novel stochastic algorithm named **SNIPS**, which addresses the field of noisy linear inverse problems, such as image deblurring, super-resolution, and compressive sensing. The algorithm uses a pre-trained minimum mean squared error (MMSE) Gaussian denoiser, and relies on ideas from Langevin dynamics and Newton's method to draw samples from the posterior distribution of the inverse problem. The paper demonstrates that the algorithm produces sharp and detailed samples.

1 Introduction

Noisy linear inverse problems is a family of image processing problems that includes denoising, inpainting, deblurring, super-resolution, compressive sensing, and other image recovery problems. A general linear problem can be expressed as:

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{z} \tag{1}$$

where $\mathbf{y}$ is the measurement, $\mathbf{x}$ is the original signal, $\mathbf{H}$ is the degradation operator and $\mathbf{z}$ is an additive, white Gaussian noise with zero-mean and variance σ_0^2 ($\mathbf{z} \sim \mathcal{N}(0, \sigma_0^2\mathbf{I})$). The goal of the task is to recover the signal $\mathbf{x}$ from the measurement $\mathbf{y}$. In the paper both $\mathbf{H}$ and σ_0 are assumed to be known.

The paper focuses on a method called Langevin Dynamics (explained in section 2.1) and aims to generalize recent work [14, 16, 29, 31] which was either proposed to solve noiseless inverse problems or addressed only specific kinds of noise. The proposed algorithm in the paper is intended to handle any noisy linear inverse problem. The paper shows that this extension is far from trivial, due to the involvement of the degradation operator $\mathbf{H}$, which makes it difficult to establish a relationship between $\mathbf{y}$ and $\mathbf{x}$, and the complex connection between the measurements' noise and the synthetic annealed Langevin noise.

2 Background

2.1 Langevin Dynamics

The *Langevin dynamics* algorithm [2, 26] suggests sampling from a probability distribution $p(\mathbf{x})$ using the following iterative rule:

$$\mathbf{x}_{t+1} = \mathbf{x}_t + \alpha \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t) + \sqrt{2\alpha}\mathbf{z}_t \tag{2}$$

where $\mathbf{z}_t \sim \mathcal{N}(0, \mathbf{I})$, α is an appropriately chosen small constant, and the term $\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t)$ is the score function. This process converges to a sample from the distribution $p(\mathbf{x})$ (after enough iterations and under mild conditions).

An extension of the previously mentioned algorithm is the *annealed Langevin dynamics* algorithm [29]. In this version, the score function in eq. (2) is replaced with a blurred version, $\nabla_{\tilde{\mathbf{x}}_t} \log p(\tilde{\mathbf{x}}_t)$, where $\tilde{\mathbf{x}}_t = \mathbf{x}_t + \mathbf{n}$ and $\mathbf{n} \sim \mathcal{N}(0, \sigma^2\mathbf{I})$ represents synthetically injected noise. This approach begins with a high noise level, gradually reduced to near zero, all while using a step size α dependent on the noise level. These modifications allow the algorithm to converge faster and achieve better overall performance. In [14] this formula is further developed and the blurred score function is given as:

$$\nabla_{\tilde{\mathbf{x}}_t} \log p(\tilde{\mathbf{x}}_t) = \frac{D(\tilde{\mathbf{x}}_t, \sigma) - \tilde{\mathbf{x}}_t}{\sigma^2} \tag{3}$$

where $D(\tilde{\mathbf{x}}_t, \sigma) = \mathbb{E}[\mathbf{x}|\tilde{\mathbf{x}}_t]$ is a minimizer of the MSE measure $\mathbb{E}[\|\mathbf{x} - D(\tilde{\mathbf{x}}_t, \sigma)\|_2^2]$ which can be approximated using a denoising neural network (MMSE denoiser).

3 Related Work

Over the years, many algorithms have been developed in an attempt to solve image restoration problems.

3.1 Classic and Deep Learning Algorithms

Most "classic" image processing algorithms try to find a good image prior that aims to regularize the inversion process and lead to visually pleasant results. Examples of such algorithms are sparsity-inspired techniques, where we try to find sparse representations of image patches [8, 9, 35], Gaussian mixture models, where the image is assumed to be drawn from a mixture of a finite number of Gaussian distributions with unknown parameters [37, 41], and methods relying on non-local self-similarity [4, 5]. Recently, deep learning algorithms, predominantly based on convolutional neural networks (CNNs), have been introduced and have demonstrated the ability to produce state-of-the-art results in various inverse problems such as denoising [18, 33, 40, 39], deblurring [17, 32], super-resolution [7, 12, 34], compressive sensing [38] and other tasks [11, 13, 20, 21, 25].

Despite the great progress compared to previous methods, many of these restoration algorithms still suffer from a critical issue: In cases of severe degradation, most recovery algorithms tend to produce washed out reconstructions that lack details. This is mainly due to the fact that most image restoration techniques aim to minimize the MSE loss $\|\mathbf{x}-\hat{\mathbf{x}}\|_2^2$ where $\mathbf{x}$ is the original image and $\hat{\mathbf{x}}$ is the restored one. When the degradation is severe, the reconstruction problem becomes highly ill-posed, i.e. there are many possible solutions for $\hat{\mathbf{x}}$. MMSE algorithms average all possible solutions of $\hat{\mathbf{x}}$, which often leads to the loss of fine details in the final output.

The work in [3] shows that such algorithms suffer from a perception-distortion trade-off, meaning that an algorithm that aims to minimize the error between $\mathbf{x}$ and $\hat{\mathbf{x}}$ will necessarily suffer from a compromised perceptual quality. As a result, only a limited perceptual improvement can be expected as long as the algorithms try to minimize the MSE.

3.2 GAN based algorithms

Generative Adversarial Networks (GANs) are generative models that try to learn the posterior distribution $p(\mathbf{x}|\mathbf{y})$, and can be used to produce samples from the distribution instead of its conditional mean. GANs have shown impressive results in generating realistic images [10, 24] and can be used to solve inverse problems while producing high-quality images [1, 22, 23]. A major disadvantage of GANs for inverse problems is that they mostly assume that the measures are noiseless, which is almost never the case.

3.3 Langevin Dynamics Based Algorithms

Another method for sampling from the posterior distribution, and the primary focus of this paper, is based on Langevin dynamics (see section 2.1). This iterative technique enables sampling from a given distribution by utilizing the *score function* - the gradient of the log of the probability density function (PDF). The work in [14, 29, 31] employs the annealed Langevin dynamics method both for image synthesis and for solving *noiseless* inverse problems. Their synthesis algorithm relies on an MMSE Gaussian denoiser to approximate a gradually blurred score function. Additionally, because of the assumption of noiseless measurements, the conditional score remains manageable in their approach to inverse problems.

4 Method

4.1 Problem Setting

The problem considered is the recovery of the signal $\mathbf{x} \in \mathbb{R}^N$ (where $\mathbf{x} \sim p(\mathbf{x})$ and $p(\mathbf{x})$ is unknown) from the observation $\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{z}$, where $\mathbf{y} \in \mathbb{R}^M, \mathbf{H} \in \mathbb{R}^{M\times N}, M \leq N, \mathbf{z} \sim \mathcal{N}(0, \sigma_0^2 I)$, and $\mathbf{H}, \sigma_0$ are known. The objective is to sample from the posterior $p(\mathbf{x}|\mathbf{y})$. However, since the score function $\nabla_x \log p(\mathbf{x}|\mathbf{y})$ is unavailable, the paper aims instead to sample from a blurred posterior distribution $p(\tilde{\mathbf{x}}|\mathbf{y})$, where $\tilde{\mathbf{x}} = \mathbf{x} + n$ and $n \sim \mathcal{N}(0, \sigma^2 I)$, with the noise level σ starting very high, and decreasing towards near-zero. In order to get a manageable derivation of the blurred score function, the sampling is done in the SVD domain. Denote the singular value decomposition (SVD) of $\mathbf{H}$ as $\mathbf{H} = U\Sigma V^T$, where $U \in \mathbb{R}^{M\times M}$ and $V \in \mathbb{R}^{N\times N}$ are orthogonal matrices, and $\Sigma \in \mathbb{R}^{M\times N}$ is a rectangular diagonal matrix containing the singular values of $\mathbf{H}$, denoted as $\{s_j\}_{j=1}^M$ in descending order ($s_1 > s_2 \cdots > s_{M-1} \geq 0$). For convenience, the paper also defines $s_j = 0$ for $M < j \leq N$, i.e., $\mathbf{H}$ has exactly M singular values greater than 0.

We notice the following equality:

$$p(\tilde{\mathbf{x}}|\mathbf{y}) = p(\tilde{\mathbf{x}}|U^T y) = p(V^T\tilde{\mathbf{x}}|U^T\mathbf{y}) \tag{4}$$

This holds because $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrices. Multiplying $\mathbf{y}$ by $\mathbf{U}^T$ neither adds nor removes information and multiplying $\tilde{\mathbf{x}}$ by $\mathbf{V}^T$ does not alter its probability distribution. Therefore, we can sample from $p(\mathbf{V}^T\tilde{\mathbf{x}}|\mathbf{U}^T\mathbf{y})$,

and then multiply the result by $\mathbf{V}$ to get a valid sample from $p(\tilde{\mathbf{x}}|\mathbf{y})$ (i.e., convert the sample back to the original space). For simplicity, denote $\mathbf{y}_T = \mathbf{U}^T\mathbf{y}, \mathbf{z}_T = \mathbf{U}^T\mathbf{z}, \mathbf{x}_T = \mathbf{V}^T\mathbf{x}, \mathbf{n}_T = \Sigma\mathbf{V}^T\mathbf{n}$ and $\tilde{\mathbf{x}}_T = \mathbf{V}^T\tilde{\mathbf{x}}$. With these notations, eq. (1) becomes

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{z} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T\mathbf{x} + \mathbf{z},$$

which leads (after multiplying by $\mathbf{U}^T$) to

$$\mathbf{y}_T = \mathbf{\Sigma}\tilde{\mathbf{x}}_T - \mathbf{n}_T + \mathbf{z}_T. \tag{5}$$

Next, the paper defines a sequence of noise levels $\{\sigma_i\}_{i=1}^{L+1}$ such that $\sigma_1 > \sigma_2 > \cdots > \sigma_L > \sigma_{L+1} = 0$, where σ_1 is high and σ_L is close to zero. For simplicity, the paper requires that $\forall i : \sigma_i s_j \neq \sigma_0$, although SNIPS works well even if such i exists. Using this notation, the paper next defines $\{\tilde{\mathbf{x}}_i\}_{i=1}^{L+1}$, a sequence of noisy versions of $\mathbf{x}$ with noise level σ_i respectively. Instead of defining these noise additions naively as independent of $\mathbf{z}$, which results in a conditional score term that cannot be calculated analytically, the paper defines the noise levels as carved gradually from z. The paper defines $\tilde{\mathbf{x}}_{L+1} = \mathbf{x}$, and $\forall i : 1 \leq i \leq L : \tilde{\mathbf{x}}_i = \tilde{\mathbf{x}}_{i+1} + \eta_i$, where $\eta_i \sim \mathcal{N}(0, (\sigma_i^2 - \sigma_{i+1}^2)\mathbf{I})$. This results in $\tilde{\mathbf{x}}_i = \mathbf{x} + \mathbf{n}_i$, where $\mathbf{n}_i = \sum_{k=1}^{L} \eta_k \sim \mathcal{N}(0, \sigma_i^2\mathbf{I})$.

Now, the paper defines the statistical dependencies between the measurements' noise $\mathbf{z}$ and the artificial noise vectors η_i. Because both noises η_i and $\mathbf{z}$ are Gaussian with uncorrelated entries, so are the components of the vectors $\mathbf{\Sigma}\mathbf{V}^T\eta_i, \mathbf{\Sigma}\mathbf{V}^T\mathbf{n}_i$ and $\mathbf{z}_T$. Focusing on a single entry j in these vectors, for which $s_j > 0$, and denoting these entries as $\eta_{T,i}, n_{T,i}$ and z_T respectively, $\eta_{T,i}$ is constructed such that

$$\mathbb{E}[\eta T_i \cdot z_T] = \begin{cases} \mathbb{E}[\eta_{T,i}^2] & \text{for } i \geq i_j \\ \mathbb{E}[(z_T - n_{T,i_j})^2] & \text{for } i = i_j - 1 \\ 0 & \text{otherwise.} \end{cases}$$

The above implies that the layers of noise $\eta_{T,L+1}, ..., \eta_{T,i_j-1}$ are all dependant of z_T, and afterwards $\eta_{T,i}$ become independent of z_T. For $s_j = 0$, the above simplifies to $\mathbb{E}[\eta T_i \cdot z_T] = 0$ for all i, which implies no statistical dependency between the given and the synthetic noises. Consequently, it can be shown that the overall noise in eq. (5) satisfies

$$(\mathbf{\Sigma}\mathbf{V}^T\mathbf{n}_i)_j = n_{T,i} - z_T \sim \begin{cases} \mathcal{N}(0, s_j^2\sigma_i^2 - \sigma_0^2) & \text{if } \sigma_i s_j > \sigma_0 \\ \mathcal{N}(0, \sigma_0^2 - s_j^2\sigma_i^2) & \text{otherwise,} \end{cases} \tag{6}$$

The top option refers to high values of the annealed Langevin noise which are stronger than z_T. In this case, $n_{T,i}$ contains all z_T and an additional independent portion of noise. The bottom option refers to lower values of the annealed Langevin noise, which are weaker than the measurements' noise and thus are fully contained in z_T, with the difference being Gaussian and independent.

4.2 Derivation of the Conditional Score Function

The above derivations show that the noise in eq. (5) is a zero-mean Gaussian, with uncorrelated entries, known variance, and is independent of $\tilde{\mathbf{x}}_i$. Thus, eq. (5) can be used for deriving the measurements part of the conditional score function. For simplicity, denote $\tilde{\mathbf{x}}_T = \mathbf{V}^T\mathbf{x}_i, \tilde{\mathbf{x}} = \mathbf{x}_i, \mathbf{n} = \mathbf{n}_i$, and turn to calculate $\nabla_{\tilde{\mathbf{x}}_T} \log p(\tilde{\mathbf{x}}|\mathbf{y}_T)$. $\tilde{\mathbf{x}}_T$ is splitted to three parts: $\tilde{\mathbf{x}}_{T,0}$, $\tilde{\mathbf{x}}_{T,>}$ and $\tilde{\mathbf{x}}_{T,<}$, which correspond to the entries j for which $s_j = 0, 0 < \sigma_i s_j < \sigma_0$ and $\sigma_i s_j > \sigma_0$ respectively. This partition of entries is non-overlapping and fully covering. Similarly, each vector $\mathbf{v} \in \mathbb{R}^N$ is partitioned into $\mathbf{v}_0, \mathbf{v}_<, \mathbf{v}_>$, which are the entries of $\mathbf{v}$ corresponding to $\tilde{\mathbf{x}}_{T,0}$, $\tilde{\mathbf{x}}_{T,<}$ and $\tilde{\mathbf{x}}_{T,>}$ respectively. Also $\mathbf{v}_{\not 0}, \mathbf{v}_{\not <}$ and $\mathbf{v}_{\not >}$ are defined as the complement entries of the three former definitions respectively. Using these notations, the authors were able to prove that

$$\nabla_{\tilde{\mathbf{x}}_T} \log p(\tilde{\mathbf{x}}_T|\mathbf{y}_T) = \mathbf{\Sigma}^T \left|\sigma_0^2\mathbf{I} - \sigma_i^2\Sigma\Sigma^T\right|^{\dagger} (\mathbf{y}_T - \Sigma\tilde{\mathbf{x}}_T) + \left(\mathbf{V}^T\nabla_{\tilde{\mathbf{x}}} \log p(\tilde{\mathbf{x}})\right)\Big|_{\not >} \tag{7}$$

where $(\mathbf{v})|_{\not >}$ is the vector $\mathbf{v}$ with zeros in its entries that correspond to $\mathbf{v}_>$. Note that the vector $\nabla_{\tilde{\mathbf{x}}} \log p(\tilde{\mathbf{x}})$ can be estimated using a neural network or a pre-trained MMSE denoiser, and all other elements can be obtained by the SVD decomposition of $\mathbf{H}$.

5 The Proposed Algorithm

This section of the paper begins by highlighting the issue of a slowed-down algorithm, it details that employing the conditional score function from eq. (7), enables the Langevin dynamics algorithm, whether using a constant or annealed step size, to converge to a sample from $p(\tilde{\mathbf{x}}_T|\mathbf{y}_T)$. However, to achieve effective convergence, a particularly small step size is required due to the varying progression rates of the entries of $\tilde{\mathbf{x}}_T$, which are influenced by their

respective singular values. This difference, combined with uniformly distributed noise variance across all entries, creates an unbalanced signal-to-noise ratio, significantly slowing down the algorithm.

To mitigate this problem, the paper suggests using a step-size vector $\alpha_i \in \mathbb{R}^N$. Denoting $\mathbf{A}_i = diag(\alpha_i)$, it obtains an updated formula for a Langevin dynamics algorithm:

$$\mathbf{V}^T\tilde{\mathbf{x}}_i = \mathbf{V}^T\tilde{\mathbf{x}}_{i-1} + c \cdot \mathbf{A}_i \cdot \nabla_{\mathbf{V}^T\tilde{\mathbf{x}}_i} \log p(\mathbf{V}^T\tilde{\mathbf{x}}_i|\mathbf{y}_T) + \sqrt{2c}\mathbf{A}_i^{1/2} \cdot z_i \tag{8}$$

The conditional score function is estimated as outlined in subsection 4.2, with c as a constant. For setting the step sizes in $\mathbf{A}_i$'s diagonal, the paper uses a strategy inspired by Newton's optimization method, which accelerates convergence to local maxima. This approach involves an update formula similar to Equation 8, but omits the additional noise $\mathbf{z}_i$ and uses $\mathbf{A}_i$ as the negative inverse of the Hessian's diagonal approximation from $\log p(\mathbf{V}^T\tilde{\mathbf{x}}_i|\mathbf{y}_T)$. Additionally, the conditional score function is calculated using equation 7 and a neural network. This combination of Langevins dynamics and Newton's method, slightly modified from a similar strategy in [28] where a Quasi-Newton method was used for approximation, enables the analytical calculation of a diagonal approximation of the negative inverse Hessian and thus obtaining the following:

$$(\alpha_i)_j = \begin{cases} \sigma_i^2 & s_j = 0 \\ \sigma_i^2 - \frac{\sigma_0^2}{s_j^2} & \sigma_i s_j > \sigma_0 \\ \sigma_i^2 \cdot (1 - s_j^2 \frac{\sigma_i^2}{\sigma_0^2}) & 0 < \sigma_i s_j < \sigma_0 \end{cases} \tag{9}$$

Utilizing the specified step sizes, the update formula presented in Equation 8, the conditional score function from Equation 7, and a neural network $(\tilde{\mathbf{x}}, \sigma)$ that estimates the score function $\nabla_{\tilde{\mathbf{x}}} \log p(\tilde{\mathbf{x}})$, the paper obtains a tractable iterative algorithm for sampling $p(\tilde{\mathbf{x}}_L|y)$, where the noise in $\tilde{\mathbf{x}}_L$ is sufficiently negligible to be considered as sampling from the ideal image manifold.

Algorithm 1 SNIPS Algorithm

Input: $\{\sigma_i\}_{i=1}^L, c, \tau, \mathbf{y}, \mathbf{H}, \sigma_0$
$\mathbf{U}, \Sigma, \mathbf{V} \leftarrow \text{svd}(\mathbf{H})$ ▷ SVD of degradation matrix
Initialize x_0 with random noise $U[0, 1]$
for $i \leftarrow 1$ to L **do** ▷ Loop over all noise levels
  $(\mathbf{A}_i)_0 \leftarrow \sigma_i^2\mathbf{I}$
  $(\mathbf{A}_i)_< \leftarrow \sigma_i^2 \cdot (\mathbf{I} - \frac{\sigma_i^2}{\sigma_0^2}\Sigma_<\Sigma_<^T)$
  $(\mathbf{A}_i)_> \leftarrow \sigma_i^2\mathbf{I} - \sigma_0^2\Sigma_>^\dagger{\Sigma_>^\dagger}^T$
  for $t \leftarrow 1$ to τ **do**
    Draw $z_t \sim N(0, \mathbf{I})$
    $d_t \leftarrow \Sigma^T \cdot \left|\sigma_0^2\mathbf{I} - \sigma_i^2\Sigma\Sigma^T\right|^\dagger \cdot (\mathbf{U}^T\mathbf{y} - \Sigma\mathbf{V}^T\mathbf{x}_{t-1}) + (\mathbf{V}^T \cdot s(\mathbf{x}_{t-1}, \sigma_i))|_{\not>}$
    $x_t \leftarrow \mathbf{V}\left(\mathbf{V}^T\mathbf{x}_{t-1} + c\mathbf{A}_i\mathbf{d}_t + \sqrt{2c}\mathbf{A}_i^{\frac{1}{2}}\mathbf{z}_t\right)$
  end
  $x_0 \leftarrow x_\tau$
end
Output: $\mathbf{x_0}$

6 Results

The paper employs the NCSNv2 network [30] to estimate the prior distribution's score function. Three distinct NCSNv2 models are trained on different datasets: 64x64 pixel images from CelebA dataset [19], 128x128 pixel images from LSUN bedrooms dataset [36], and LSUN 128x128 pixel images of towers. SNIPS is evaluated on these sets for image deblurring, super-resolution, and compressive sensing, running the algorithm eight times per test and analyzing both individual samples and their means as approximations of the MMSE solution, $\mathbb{E}[\mathbf{x}|\mathbf{y}]$.

The paper applies various image processing techniques using SNIPS. For image deblurring, a uniform 5x5 blur kernel and additive white Gaussian noise ($\sigma_0 = 0.1$) are used on CelebA dataset images, yielding visually diverse results. Super resolution involves downscaling images using block averaging filters of 2x2 or 4x4 pixels, adding noise, and displaying results on LSUN and CelebA. Compressive sensing uses random projection matrices to compress images by 25%, 12.5%, and 6.25%, with greater compression leading to more varied reconstructions. The experiments measure the average PSNR for each of the eight samples and their means, indicating a 2.4 dB improvement in PSNR for the means, despite less visual appeal. This result is consistent with the theory in [3], which states that

the difference in PSNR between posterior samples and the conditional mean (the MMSE estimator) should be 3dB. SNIPS outperforms RED[27] in deblurring, with significant improvements in PSNR and LPIPS, a perceptual quality metric.

6.1 Assessing Faithfulness to the Measurements

A valid solution to an inverse problem must be visually appealing and consistent with the underlying prior distribution of images, and should maintain the relationship given in the problem setting. To verify these criteria, visual appeal is judged by the realistic appearance of processed images. To check measurement fidelity, the reconstructed images ($\hat{\mathbf{x}}$) are degraded using $\mathbf{H}$, and the deviation from the original measurements $\mathbf{y}$ is calculated as $\mathbf{y} - \mathbf{H}\hat{x}$. This difference should ideally be additive white Gaussian noise with a standard deviation of σ_0. This is validated by measuring the empirical standard deviation, performing the Pearson-D'Agostino test of normality, and checking the Pearson correlation coefficient among neighboring entries to ensure that they are not correlated. The algorithm's performance is confirmed as it consistently matches these statistical expectations in nearly all the tests, demonstrating its effectiveness in solving inverse problems.

7 Experiments

7.1 Paper's Experiments Reproduction

In this section, we reproduced the experiments shown in the paper and checked if the results match. In the first experiment, we tried to reproduce the results shown in Table 1 in the original paper [15]. We can see in table 1 that the results we got are similar, but a little worse than the original results (around 0.5-0.8dB less). While we didn't find any particular reason, we can assume that the paper conducted these specific experiments with value of σ different than 0.1 (which we used), as the paper doesn't state exactly which value is used. Another option could be that we used different images, which can lead to different results. In fig. 1 and fig. 2 we added samples from our experiments. Besides that, we conducted similar experiments with different σ_0 values (fig. 3, fig. 4, fig. 5).

Table 1: PSNR results for different inverse problems on 8 images from CelebA. We ran SNIPS 8 times with $\sigma_0 = 0.1$ and obtained 8 samples. The average PSNR for each of the samples is in the first column, while the average PSNR for the mean of the 8 samples for each image is in the second column. In parenthesis we added the original values from the paper.

Problem	**Sample PSNR (Orig.)**	**Mean PSNR (Orig.)**
Uniform deblurring	24.94 (25.54)	27.22 (28.01)
Super resolution (by 2)	25.04 (25.58)	27.42 (28.03)
Super resolution (by 4)	21.26 (21.90)	23.48 (24.31)
Compressive sensing (by 25%)	25.04 (25.68)	27.27 (28.06)
Compressive sensing (by 12.5%)	21.74 (22.34)	23.98 (24.67)

7.2 Further Experiments

In addition to the experiments mentioned above, we also conducted further tests on images from a different dataset - ImageNet [6]. We selected a few samples from this dataset and performed similar experiments using σ_0 values of 0.1 and 0.04. We chose to run these specific experiments because the paper clearly indicates that the algorithm does not work well with datasets it was not trained on, and we wanted to compare these results with those on datasets the SNIPS model is trained on (e.g. CelebA). As shown in the samples we provided (fig. 6, fig. 8, fig. 7), the results are significantly worse than the corresponding results on CelebA images. In addition to the smearing in the output images, we can observe hints of human faces, which occur because the model was trained on the CelebA dataset. This limitation can be solved by using an MMSE denoiser trained specifically on the ImageNet dataset instead of the CelebA dataset.

Another experiment we conducted involved using different types of additive noise other than Gaussian noise with $\sigma = 0.1$. We experimented with Poisson noise, salt-and-pepper noise, and speckle noise. The results of these experiments offer insight into how SNIPS performs under different noise conditions compared to the original results using additive Gaussian noise. The results are presented below in table 2, table 3, and table 4, respectively.

Table 2: PSNR results for various inverse problems with additive Poisson noise on 8 images from the CelebA dataset. SNIPS was run 8 times, producing 8 samples. The first column shows the average PSNR for each individual sample, while the second column displays the average PSNR for the mean of the 8 samples for each image. In parenthesis we added the original values from the paper.

Problem	**Sample PSNR (Orig.)**	**Mean PSNR (Orig.)**
Uniform deblurring	25.87 (25.54)	28.28 (28.01)
Super resolution (by 2)	26.38 (25.58)	29.17 (28.03)
Super resolution (by 4)	22.17 (21.90)	24.44 (24.31)
Compressive sensing (by 25%)	25.89 (25.68)	28.61 (28.06)
Compressive sensing (by 12.5%)	22.04 (22.34)	24.49 (24.67)

Table 3: PSNR results for various inverse problems with additive salt and pepper noise on 8 images from the CelebA dataset. SNIPS was run 8 times, producing 8 samples. The first column shows the average PSNR for each individual sample, while the second column displays the average PSNR for the mean of the 8 samples for each image. In parenthesis we added the original values from the paper.

Problem	**Sample PSNR (Orig.)**	**Mean PSNR (Orig.)**
Uniform deblurring	23.57 (25.54)	25.65 (28.01)
Super resolution (by 2)	20.74 (25.58)	22.37 (28.03)
Super resolution (by 4)	18.86 (21.90)	20.5 (24.31)
Compressive sensing (by 25%)	20.14 (25.68)	21.03 (28.06)
Compressive sensing (by 12.5%)	18.41 (22.34)	19.73 (24.67)

Table 4: PSNR results for various inverse problems with additive speckle noise on 8 images from the CelebA dataset. SNIPS was run 8 times, producing 8 samples. The first column shows the average PSNR for each individual sample, while the second column displays the average PSNR for the mean of the 8 samples for each image. In parenthesis we added the original values from the paper.

Problem	**Sample PSNR (Orig.)**	**Mean PSNR (Orig.)**
Uniform deblurring	25.67 (25.54)	28.01 (28.01)
Super resolution (by 2)	26.09 (25.58)	28.71 (28.03)
Super resolution (by 4)	21.93 (21.90)	24.17 (24.31)
Compressive sensing (by 25%)	23.15 (25.68)	25.03 (28.06)
Compressive sensing (by 12.5%)	20.37 (22.34)	22.2 (24.67)

As shown in the tables, Poisson noise resulted in the highest PSNR values among all noise types tested, with noticeable improvements in both sample PSNR and mean PSNR compared to the original Gaussian noise results. Poisson noise typically models the natural variability in photon counts and is commonly observed in low-light conditions. The improved PSNR suggests that SNIPS handles this type of noise effectively. This can be attributed to the fact that Poisson noise is signal-dependent and follows a distribution that somewhat resembles Gaussian noise at higher counts. Because SNIPS is optimized for Gaussian noise, it can still perform well under Poisson conditions due to the noise's statistical properties aligning reasonably well with those SNIPS is trained to handle. The fact that the Poisson noise is signal-dependent (varies with the signal) indicates that it often matches the statistical properties of the images better than Gaussian noise (which has constant variance across all signal levels). This closer alignment with natural image structures can make the denoising or reconstruction task more manageable for SNIPS and thus result in better results.

Salt and pepper noise produced the lowest PSNR values across all experiments, indicating that SNIPS struggles the most with this type of noise. Salt and pepper noise, characterized by random occurrences of extreme values (black and white pixels), creates high-contrast noise patterns and abrupt changes that are not well modeled by the Gaussian MMSE denoiser that SNIPS relies on. This causes this kind of noise to be challenging for algorithms like SNIPS, which are designed to handle smoother, more predictable variations.

Speckle noise yielded moderate PSNR values, generally performing better than salt and pepper noise but not reaching the levels observed with Poisson noise. For example, for super-resolution (by 2), the mean PSNR was 28.71, slightly better than the Gaussian result of 28.03. Speckle noise is a multiplicative noise that arises from interference of waves and manifests as a granular pattern over the image. This multiplicative nature makes Speckle noise more complex than additive Gaussian noise, but it still maintains some degree of regularity and structure.

SNIPS performs moderately well with speckle noise because, unlike salt and pepper noise, speckle patterns do not cause extreme, isolated pixel values. The noise is distributed across the image in a way that SNIPS can somewhat mitigate, although not as effectively as Gaussian noise. The multiplicative nature means that the noise scales with intensity, which is more manageable for SNIPS compared to the unpredictable spikes from salt and pepper noise.

We can conclude that SNIPS shows robust adaptability to noise types that have properties aligning with Gaussian assumptions, like Poisson noise, but struggles with noise types that deviate significantly, such as salt and pepper noise. These results highlight the need for SNIPS to incorporate more sophisticated denoising techniques or pre-processing steps when dealing with more erratic noise patterns, such as salt and pepper or highly structured noise like speckle.

8 Conclusion and Future Work

The paper concludes by presenting SNIPS as an innovative stochastic algorithm for addressing noisy linear inverse problems through annealed Langevin dynamics and Newton's method, supported by a pre-trained Gaussian MMSE denoiser. SNIPS effectively generates diverse, high-quality samples from the posterior distribution, ensuring their relevance to the data. Its methodology incorporates a detailed choice of annealed noise and SVD decomposition of the degradation operator to reduce dependency in measurements, successfully applied to image deblurring, super-resolution, and compressive sensing.

Looking forward, the paper identifies key areas for improvement in SNIPS: the resource-intensive SVD decomposition limits scalability; its efficacy with general content images is restricted by the denoiser's characteristics; and the high number of iterations required for processes like super-resolution, which suggests a need for optimizing the algorithm's efficiency.

9 Review of the Paper

In this section, we will critically evaluate the structure and clarity of the paper, discussing the way of presentation of the arguments and methodologies in terms of their comprehensibility and logical coherence.

We found the paper to offer a unique perspective on the application of stochastic algorithms for solving noisy linear inverse problems. Yet, it presents several challenges in terms of accessibility and clarity. First, some derivations appear to be more complex than necessary, which may obscure the underlying principles rather than elucidate them. The introduction extensively discusses the properties of the proposed method before adequately introducing it, leading to potential confusion about the foundational concepts.

The lack of experimental comparisons with established baselines is a significant omission. Typically, visual comparisons and metrics such as MSE and SSIM are essential for evaluating the effectiveness of new methods against previous work. Such comparisons would greatly aid in understanding where and how the current method improves upon or falls short of existing techniques.

Section 3 is notably technical and complex, yet it fails to communicate clearly why access to the true score function is unattainable, which is crucial for comprehending the subsequent solutions proposed. Additionally, while detailed derivations are relegated to the appendix, the paper does not sufficiently guide the reader through the significance of key equations (7-11), leaving gaps in the narrative that links theoretical development to practical application.

From a personal standpoint, as an undergraduate student with no prior exposure to Langevin dynamics, the mathematical content of the paper felt particularly daunting. A more intuitive presentation of each sophisticated equation, perhaps by prefacing them with accessible explanations or analogies, would have made the material more approachable and the authors' intentions clearer. This adjustment would significantly enhance the educational value of the paper for novices in the field.

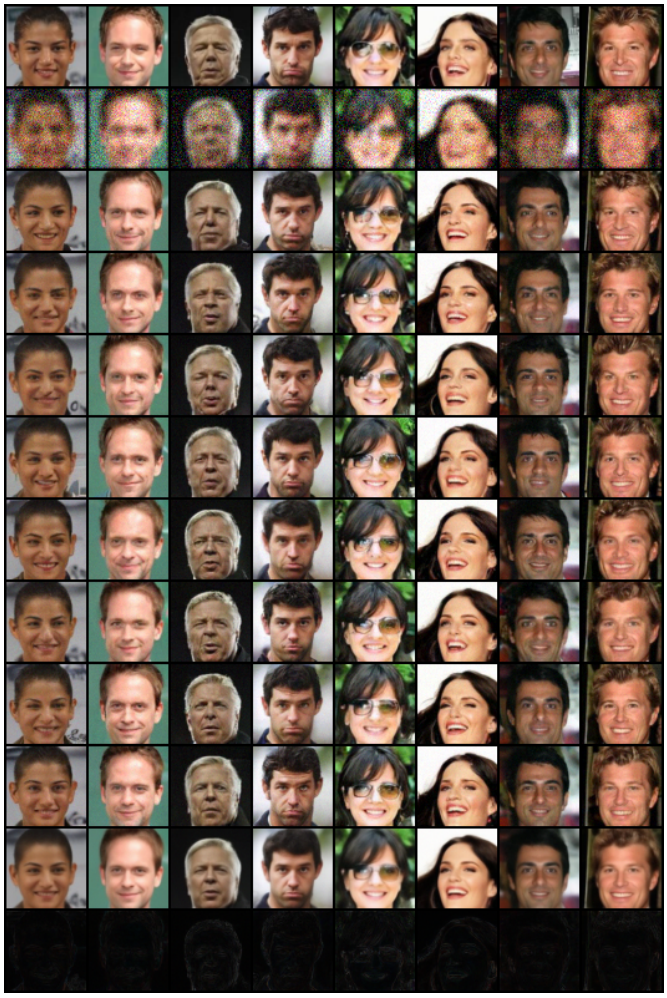

Figure 1: Uniform deblurring results on CelebA images with additive noise of $\sigma_0 = 0.1$. row 1: original images. row 2: degraded images. rows 3-10: 8 samples from the algorithm. row 11: mean sample. row 12: std of the samples.

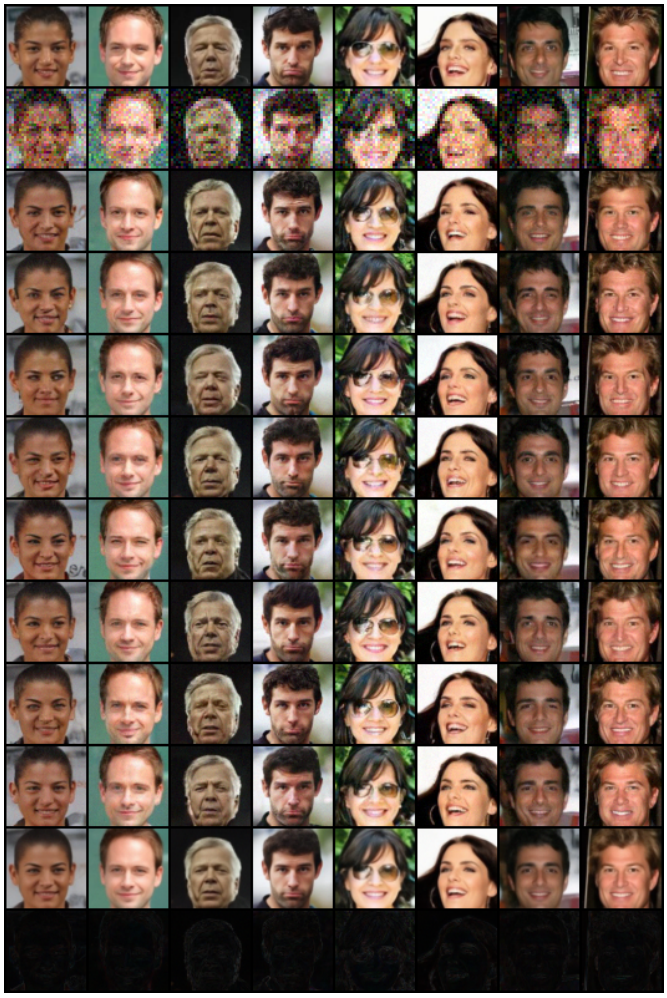

Figure 2: Super-Resolution (down-scaling $2:1$ by plain averaging) results on CelebA images with additive noise of $\sigma_0 = 0.1$. row 1: original images. row 2: degraded images. rows 3-10: 8 samples from the algorithm. row 11: mean sample. row 12: std of the samples.

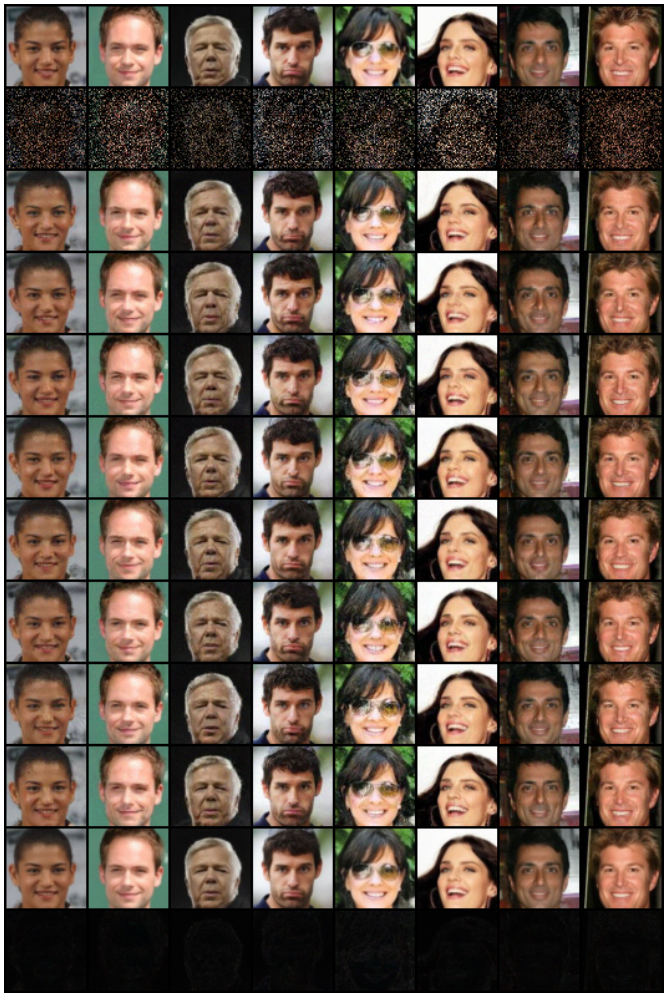

Figure 3: Compressive Sensing by 25% results on CelebA images with additive noise of $\sigma_0 = 0.04$. row 1: original images. row 2: degraded images. rows 3-10: 8 samples from the algorithm. row 11: mean sample. row 12: std of the samples.

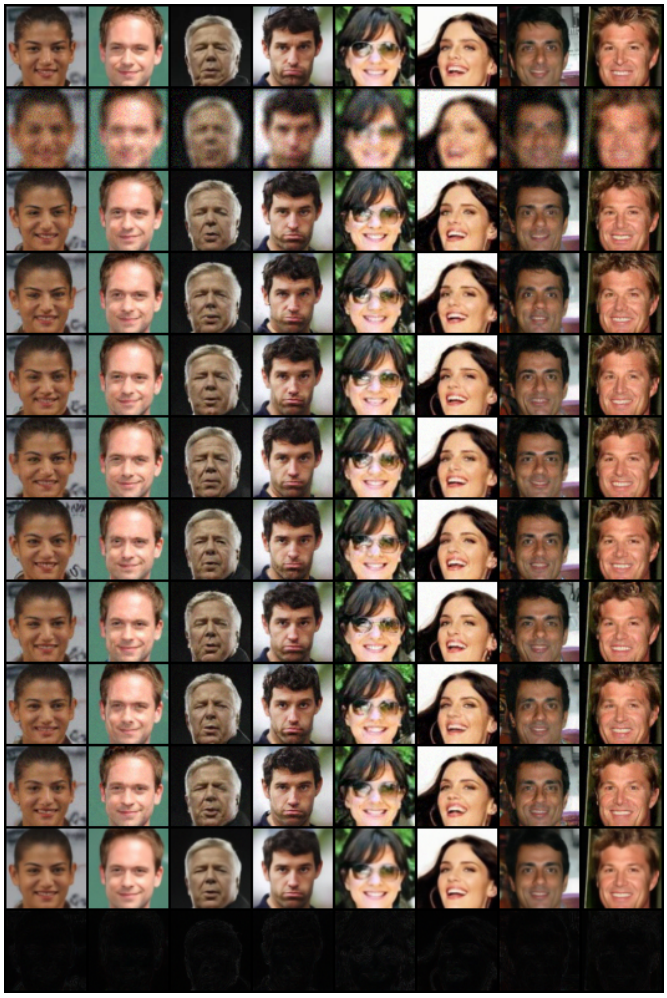

Figure 4: Gaussian deblurring results on CelebA images with additive noise of $\sigma_0 = 0.04$. row 1: original images. row 2: degraded images. rows 3-10: 8 samples from the algorithm. row 11: mean sample. row 12: std of the samples.

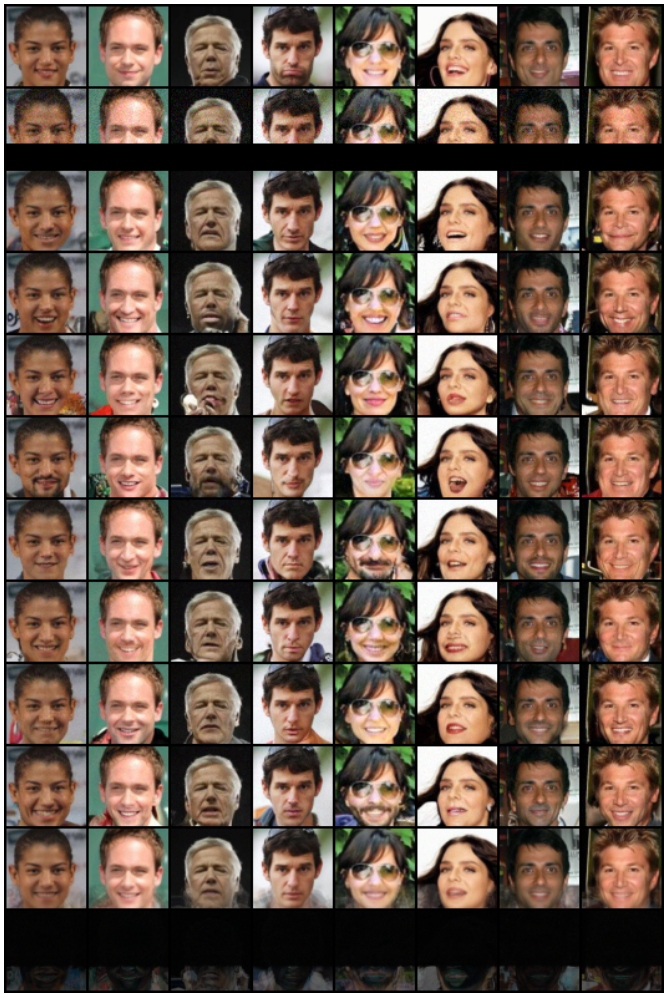

Figure 5: Inpainting results on CelebA images with additive noise of $\sigma_0 = 0.04$. row 1: original images. row 2: degraded images. rows 3-10: 8 samples from the algorithm. row 11: mean sample. row 12: std of the samples.

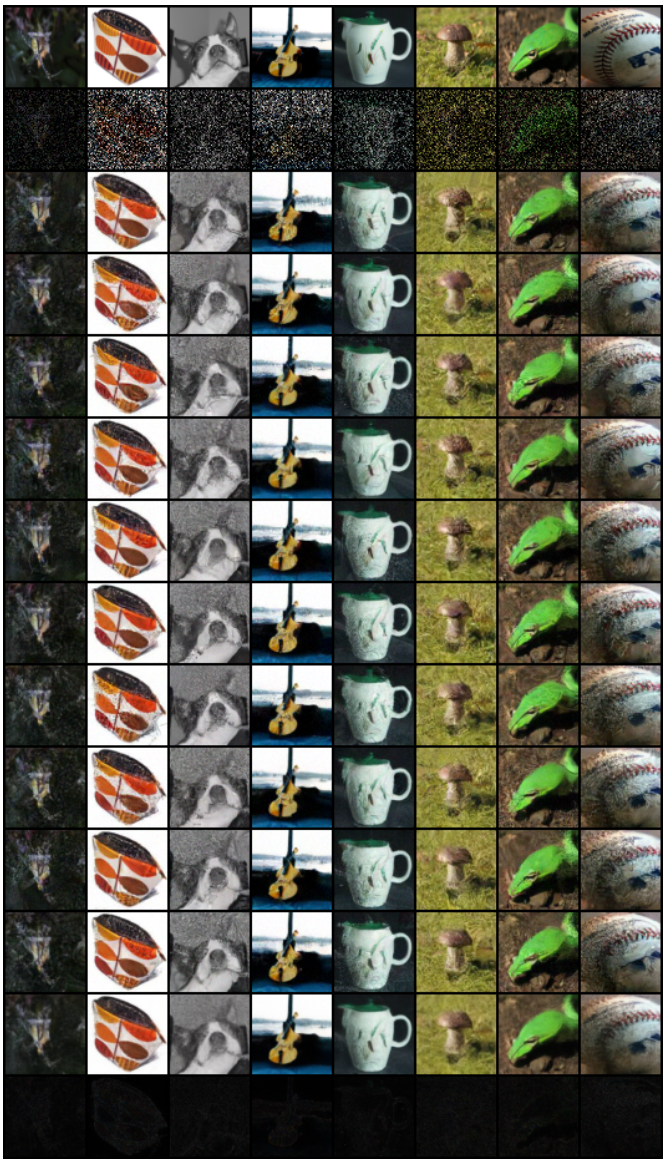

Figure 6: Compressive Sensing results on ImageNet images with additive noise of $\sigma_0 = 0.04$. row 1: original images. row 2: degraded images. rows 3-12: 10 samples from the algorithm. row 11: mean sample. row 12: std of the samples.

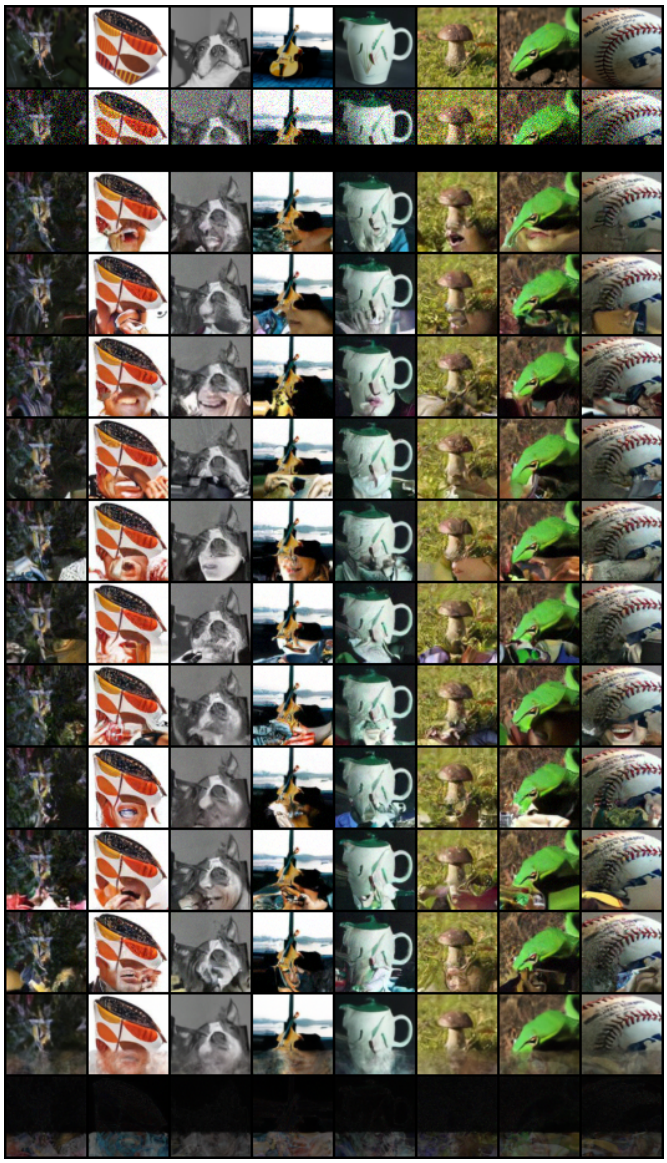

Figure 7: Inpainting results on ImageNet images with additive noise of $\sigma_0 = 0.1$. row 1: original images. row 2: degraded images. rows 3-12: 10 samples from the algorithm. row 11: mean sample. row 12: std of the samples.

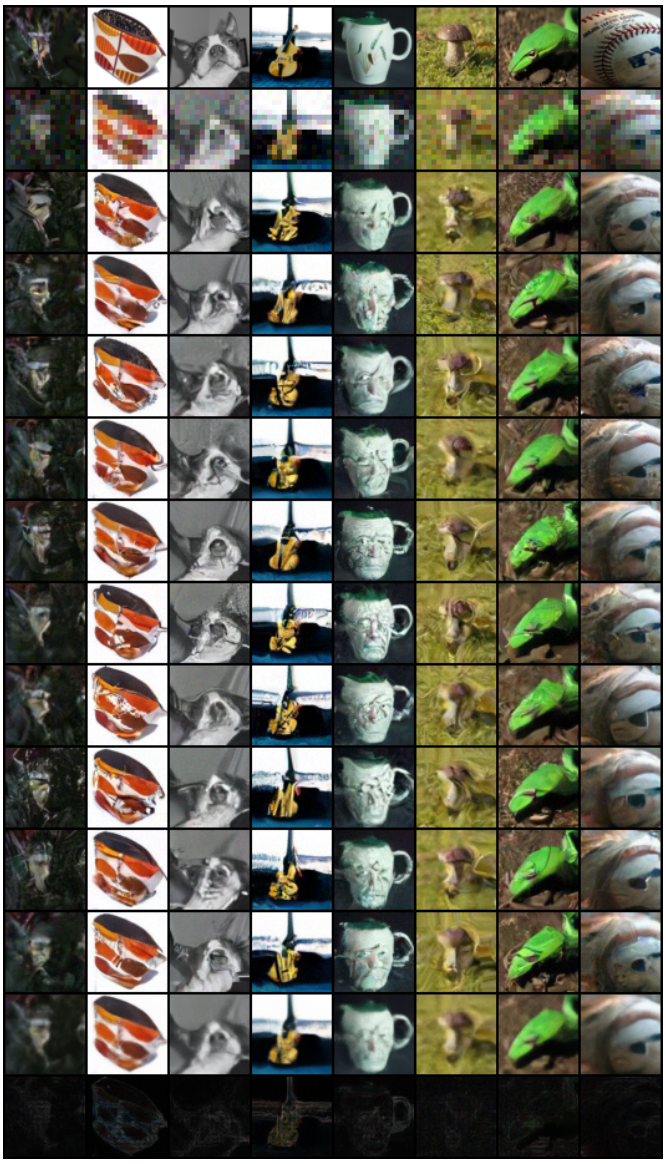

Figure 8: Super-Resolution (down-scaling 4 : 1 by plain averaging) results on ImageNet images with additive noise of $\sigma_0 = 0.04$. row 1: original images. row 2: degraded images. rows 3-12: 10 samples from the algorithm. row 11: mean sample. row 12: std of the samples.

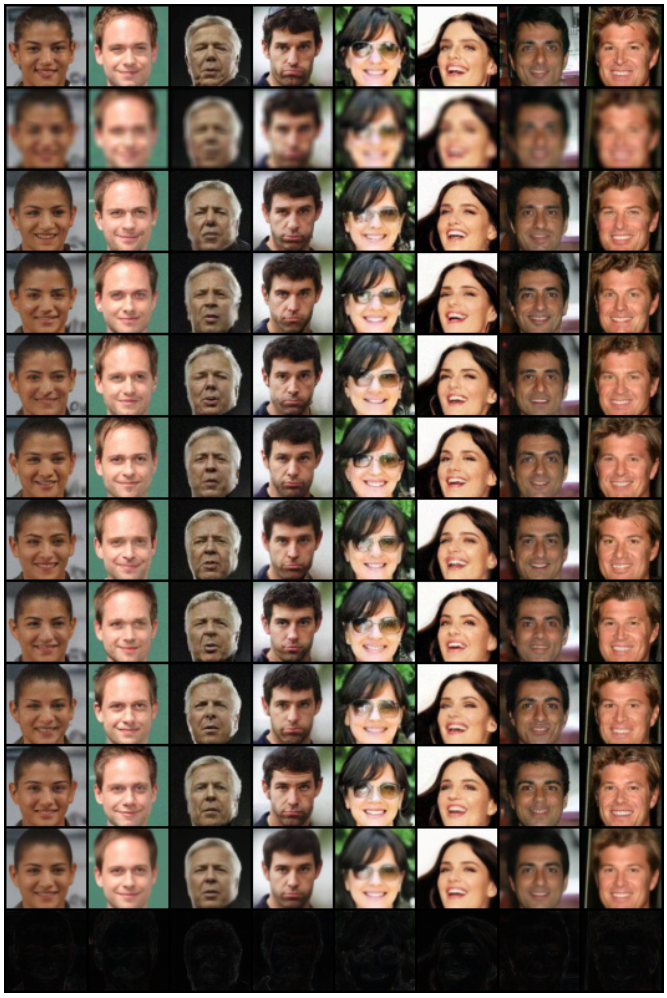

Figure 9: Uniform deblurring results on CelebA images with additive Poisson noise. row 1: original images. row 2: degraded images. rows 3-10: 8 samples from the algorithm. row 11: mean sample. row 12: std of the samples.

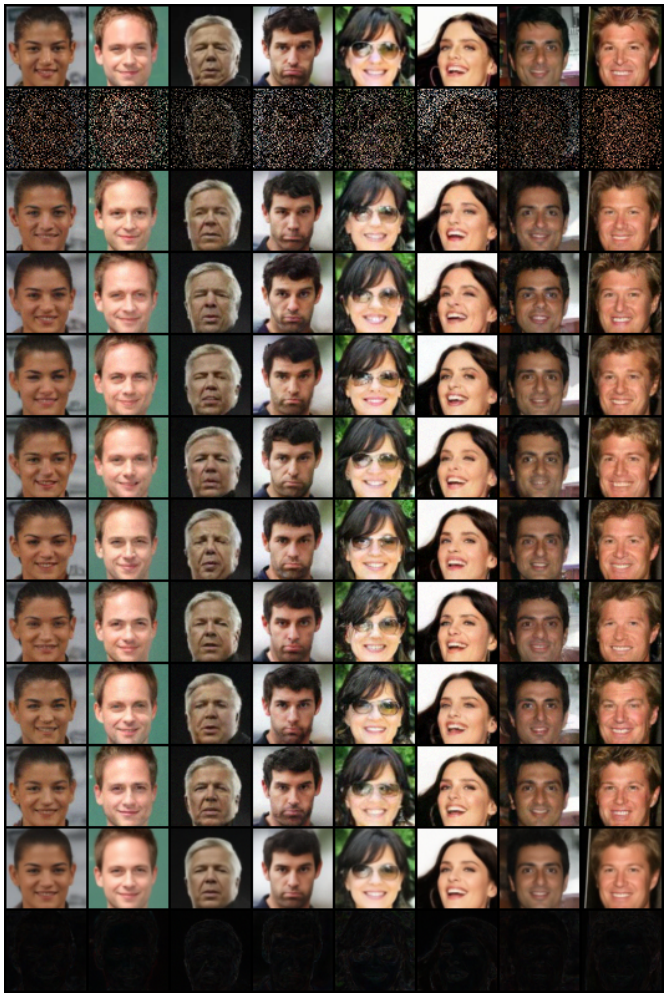

Figure 10: Compressive Sensing by 25% results on CelebA images with additive Poisson noise. row 1: original images. row 2: degraded images. rows 3-10: 8 samples from the algorithm. row 11: mean sample. row 12: std of the samples.

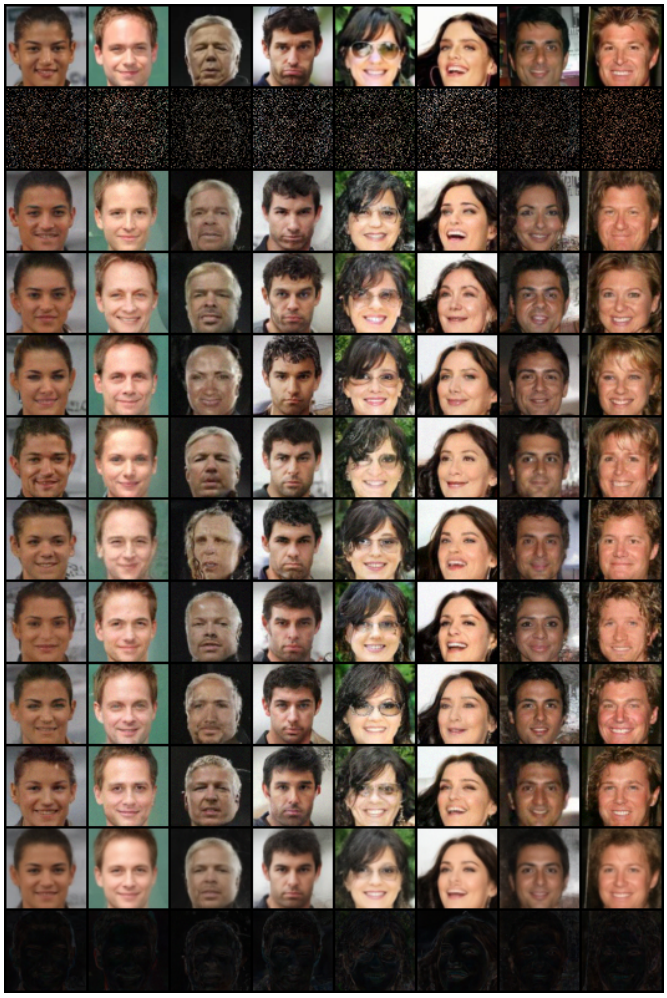

Figure 11: Compressive Sensing by 12.5% results on CelebA images with additive Poisson noise. row 1: original images. row 2: degraded images. rows 3-10: 8 samples from the algorithm. row 11: mean sample. row 12: std of the samples.

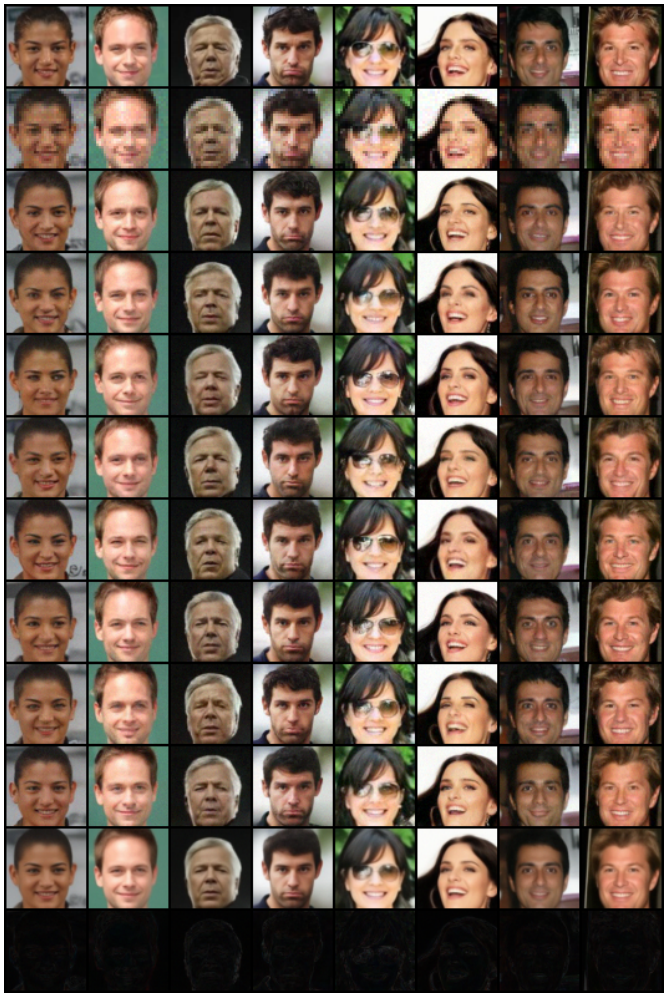

Figure 12: Super-Resolution (down-scaling 2:1 by plain averaging) results on CelebA images with additive Poisson noise. row 1: original images. row 2: degraded images. rows 3-10: 8 samples from the algorithm. row 11: mean sample. row 12: std of the samples.

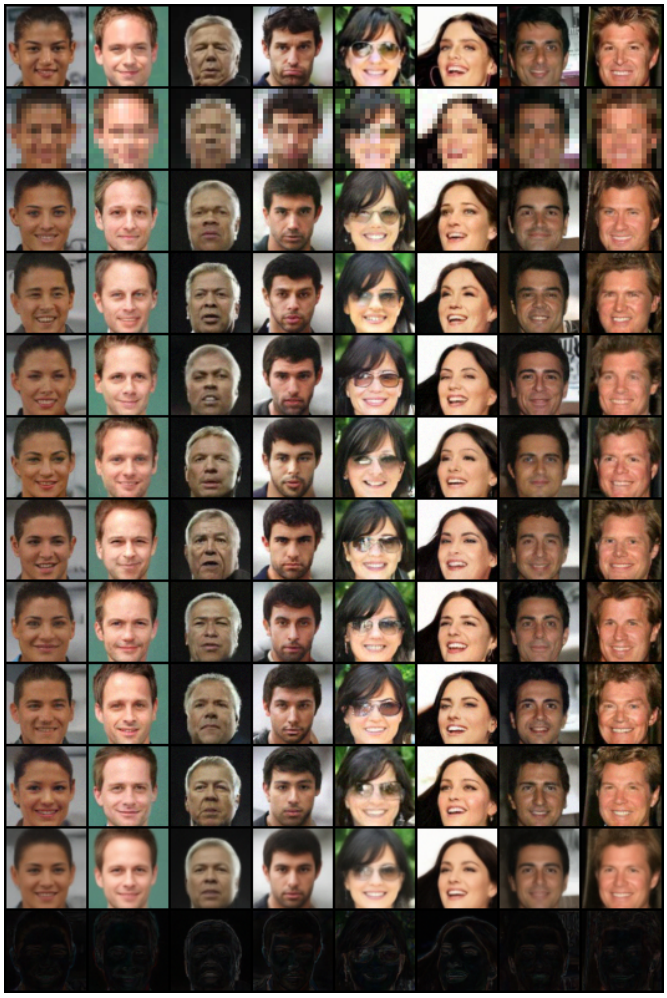

Figure 13: Super-Resolution (down-scaling 4:1 by plain averaging) results on CelebA images with additive Poisson noise. row 1: original images. row 2: degraded images. rows 3-10: 8 samples from the algorithm. row 11: mean sample. row 12: std of the samples.

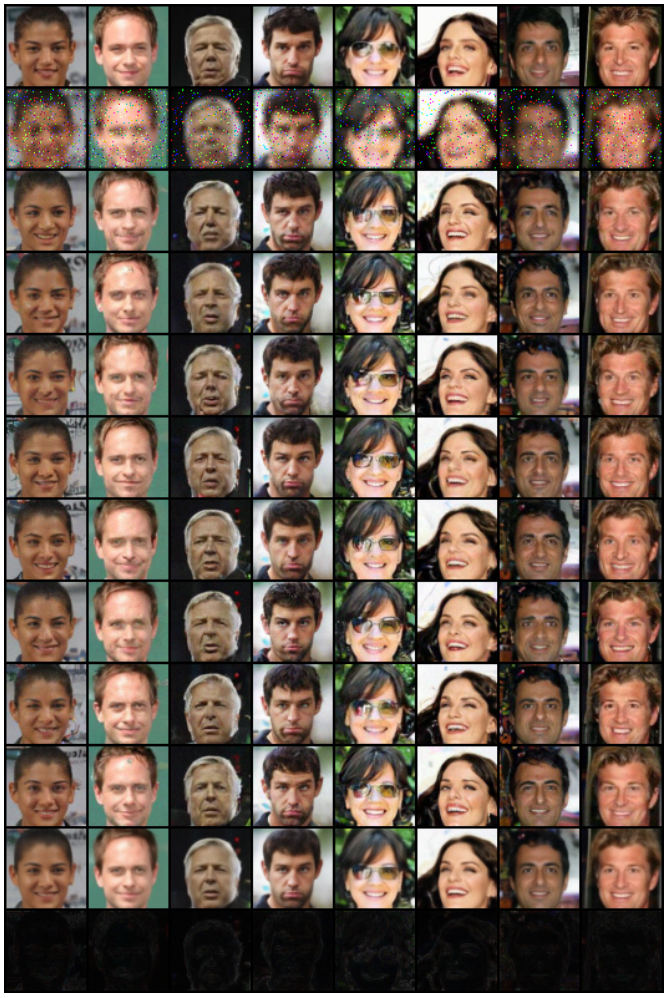

Figure 14: Uniform deblurring results on CelebA images with additive salt and pepper noise. row 1: original images. row 2: degraded images. rows 3-10: 8 samples from the algorithm. row 11: mean sample. row 12: std of the samples.

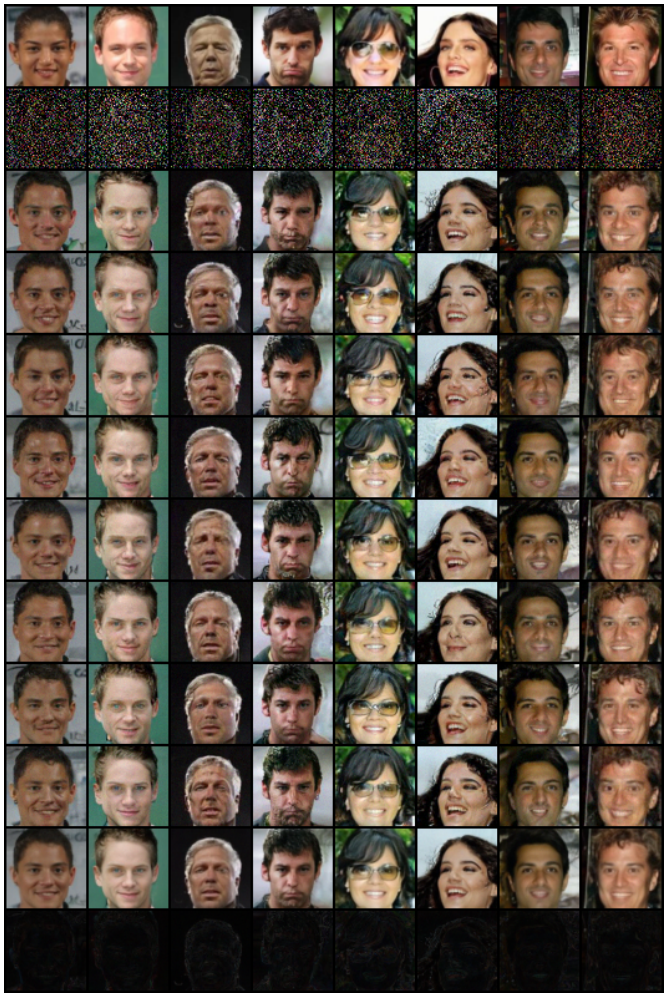

Figure 15: Compressive Sensing by 25% results on CelebA images with additive salt and pepper noise. row 1: original images. row 2: degraded images. rows 3-10: 8 samples from the algorithm. row 11: mean sample. row 12: std of the samples.

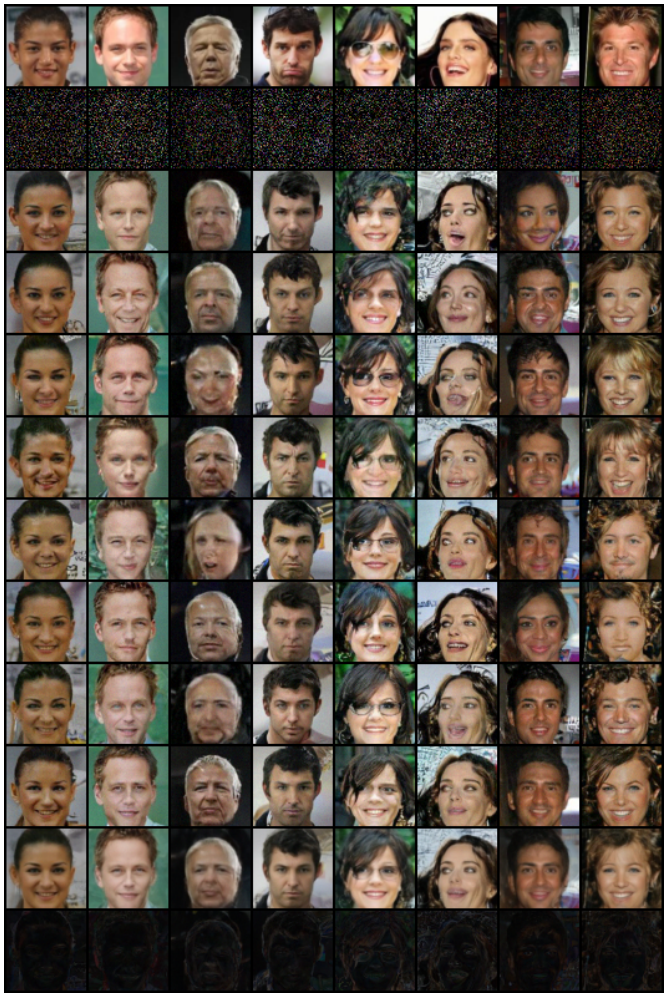

Figure 16: Compressive Sensing by 12.5% results on CelebA images with additive salt and pepper noise. row 1: original images. row 2: degraded images. rows 3-10: 8 samples from the algorithm. row 11: mean sample. row 12: std of the samples.

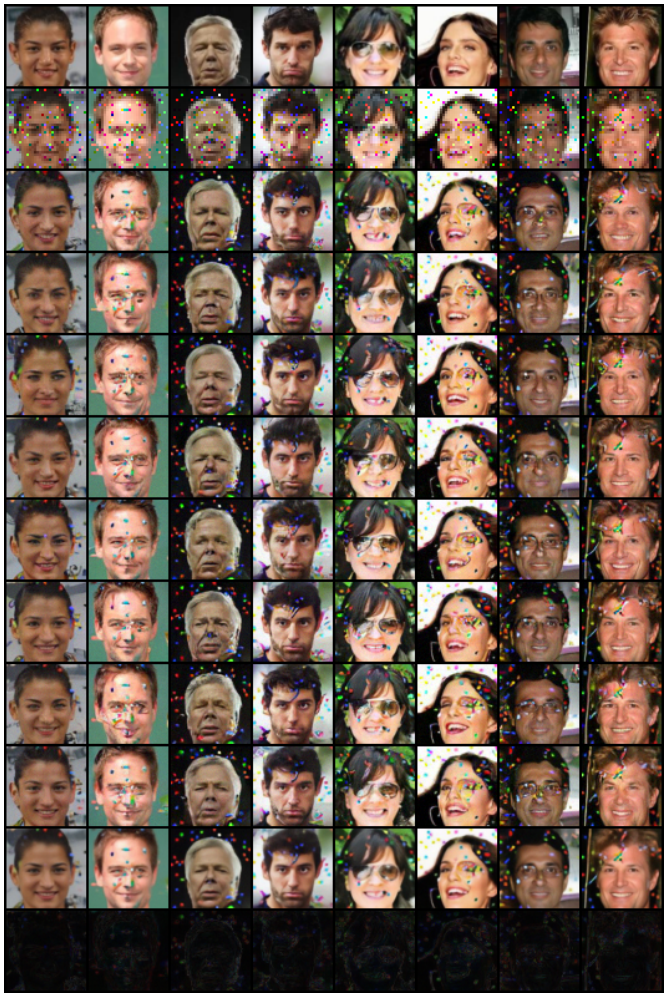

Figure 17: Super-Resolution (down-scaling 2:1 by plain averaging) results on CelebA images with additive salt and pepper noise. row 1: original images. row 2: degraded images. rows 3-10: 8 samples from the algorithm. row 11: mean sample. row 12: std of the samples.

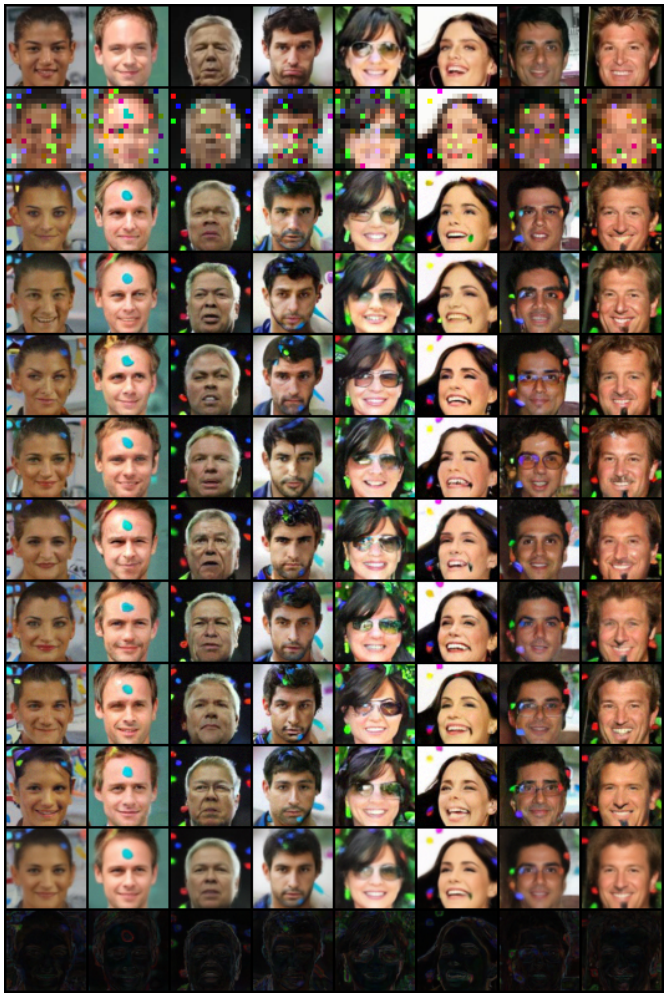

Figure 18: Super-Resolution (down-scaling 4:1 by plain averaging) results on CelebA images with additive salt and pepper noise. row 1: original images. row 2: degraded images. rows 3-10: 8 samples from the algorithm. row 11: mean sample. row 12: std of the samples.

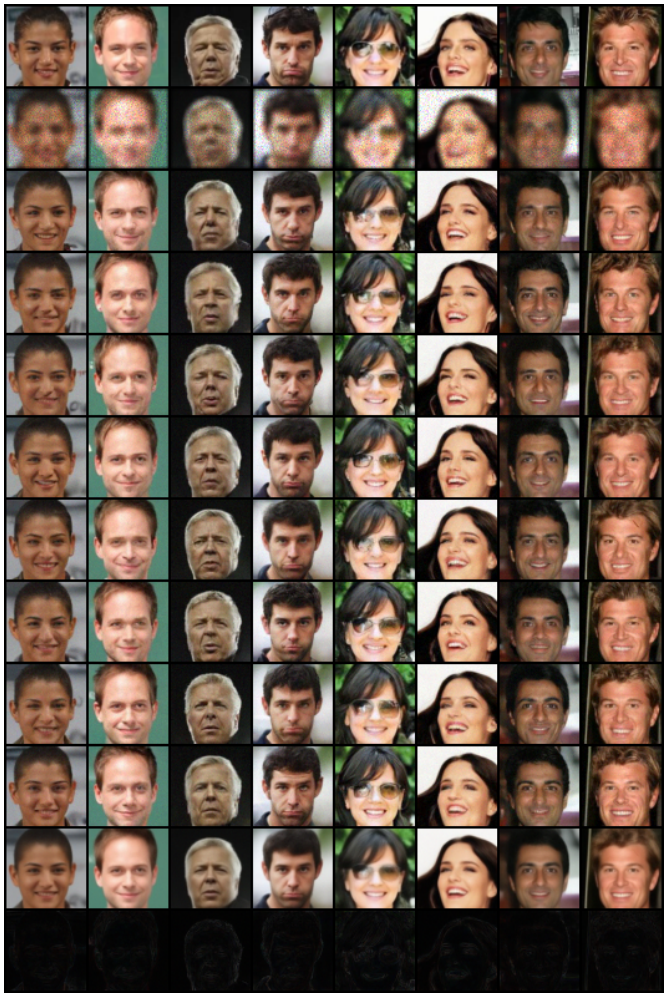

Figure 19: Uniform deblurring results on CelebA images with additive speckle noise. row 1: original images. row 2: degraded images. rows 3-10: 8 samples from the algorithm. row 11: mean sample. row 12: std of the samples.

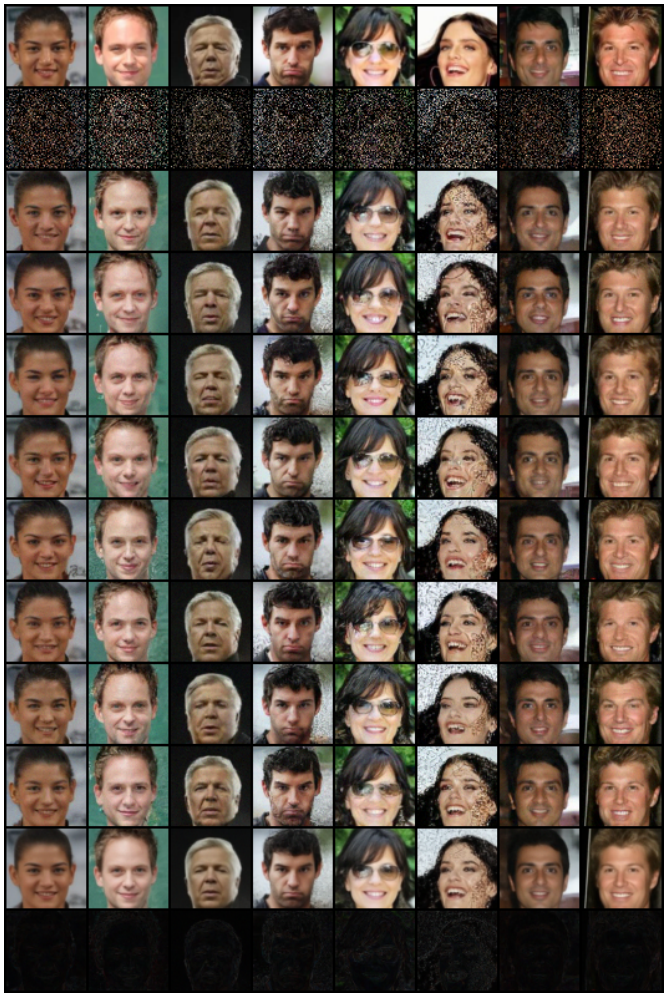

Figure 20: Compressive Sensing by 25% results on CelebA images with additive speckle noise. row 1: original images. row 2: degraded images. rows 3-10: 8 samples from the algorithm. row 11: mean sample. row 12: std of the samples.

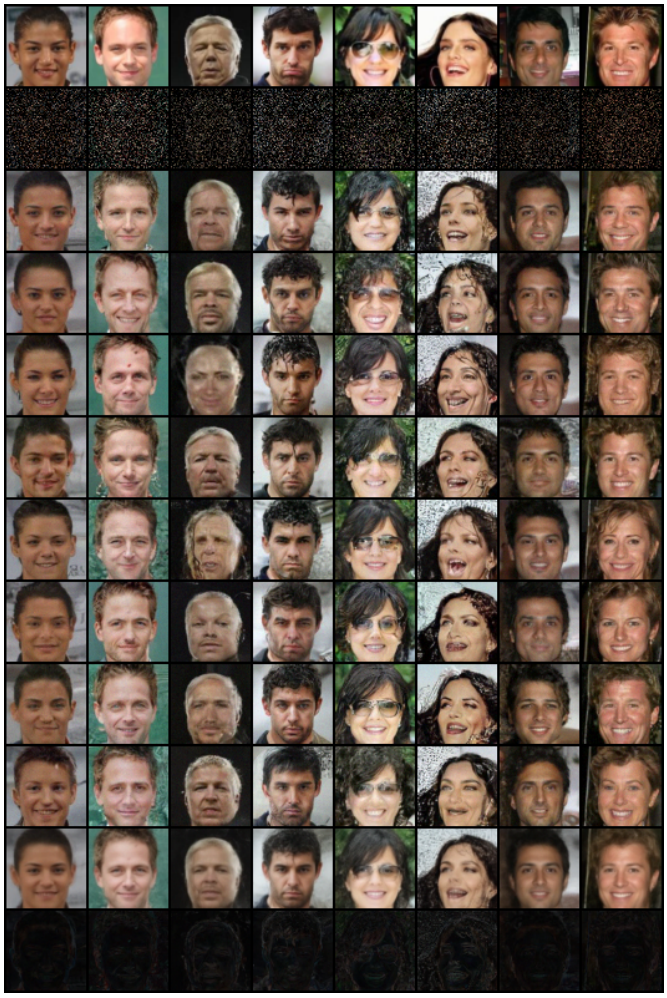

Figure 21: Compressive Sensing by 12.5% results on CelebA images with additive speckle noise. row 1: original images. row 2: degraded images. rows 3-10: 8 samples from the algorithm. row 11: mean sample. row 12: std of the samples.

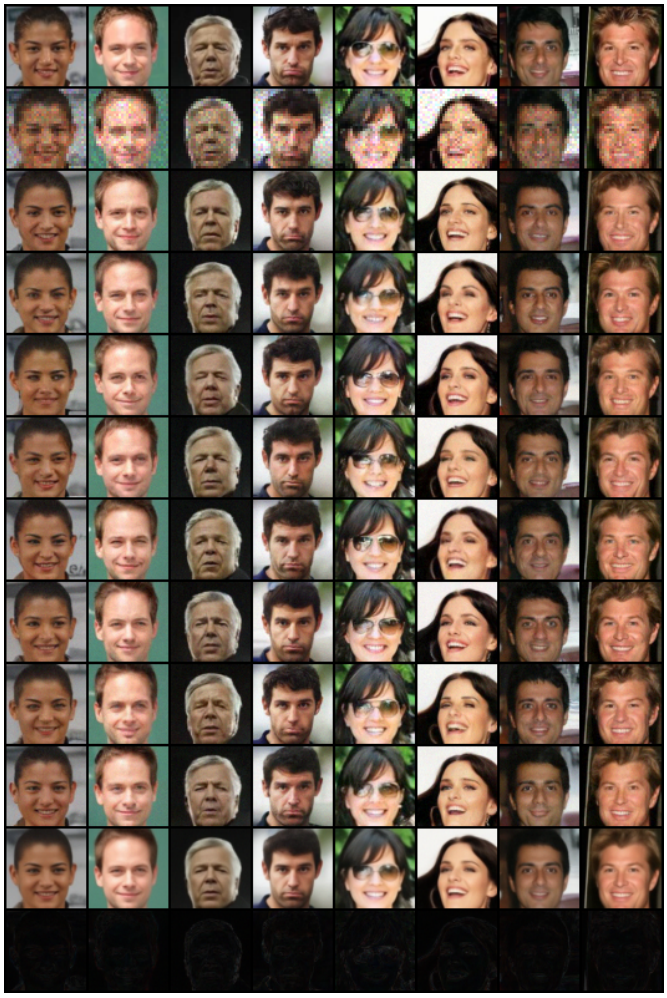

Figure 22: Super-Resolution (down-scaling 2:1 by plain averaging) results on CelebA images with additive speckle noise. row 1: original images. row 2: degraded images. rows 3-10: 8 samples from the algorithm. row 11: mean sample. row 12: std of the samples.

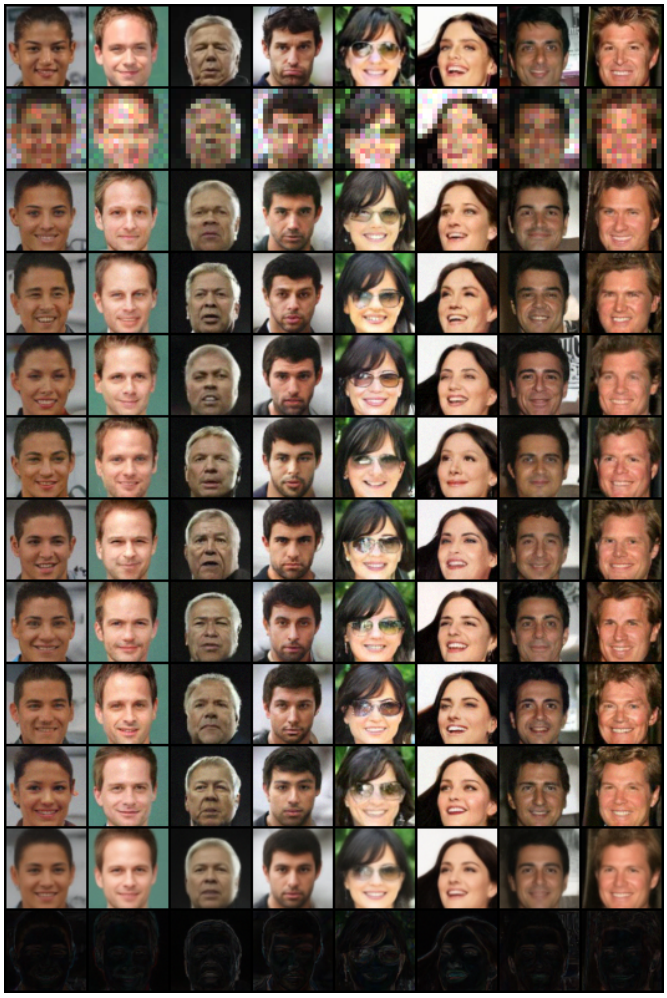

Figure 23: Super-Resolution (down-scaling 4:1 by plain averaging) results on CelebA images with additive speckle noise. row 1: original images. row 2: degraded images. rows 3-10: 8 samples from the algorithm. row 11: mean sample. row 12: std of the samples.

References

[1] Yuval Bahat and Tomer Michaeli. "Explorable super resolution". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 2716–2725.

[2] Julian Besag. "Markov Chain Monte Carlo for statistical inference". In: *Center for Statistics and the Social Sciences* 9 (2001), pp. 24–25.

[3] Yochai Blau and Tomer Michaeli. "The perception-distortion tradeoff". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 6228–6237.

[4] Antoni Buades, Bartomeu Coll, and J-M Morel. "A non-local algorithm for image denoising". In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. Vol. 2. IEEE. 2005, pp. 60–65.

[5] Aram Danielyan, Vladimir Katkovnik, and Karen Egiazarian. "BM3D frames and variational image deblurring". In: *IEEE Transactions on Image Processing* 21.4 (2011), pp. 1715–1728.

[6] Jia Deng et al. "ImageNet: A large-scale hierarchical image database". In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. 2009, pp. 248–255. DOI: 10.1109/CVPR.2009.5206848.

[7] Chao Dong, Chen Change Loy, and Xiaoou Tang. "Accelerating the super-resolution convolutional neural network". In: *European Conference on Computer Vision*. Springer. 2016, pp. 391–407.

[8] Weisheng Dong et al. "Nonlocally centralized sparse representation for image restoration". In: *IEEE Transactions on Image Processing* 22.4 (2012), pp. 1620–1630.

[9] Michael Elad and Michal Aharon. "Image denoising via sparse and redundant representations over learned dictionaries". In: *IEEE Transactions on Image Processing* 15.12 (2006), pp. 3736–3745.

[10] Ian Goodfellow et al. "Generative Adversarial Nets". In: *Advances in Neural Information Processing Systems*. Vol. 27. Curran Associates, Inc., 2014.

[11] Harshit Gupta et al. "CNN-based projected gradient descent for consistent CT image reconstruction". In: *IEEE Transactions on Medical Imaging* 37.6 (2018), pp. 1440–1453.

[12] Muhammad Haris, Gregory Shakhnarovich, and Norimichi Ukita. "Deep back-projection networks for super-resolution". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 1664–1673.

[13] Chang Min Hyun et al. "Deep learning for undersampled MRI reconstruction". In: *Physics in Medicine & Biology* 63.13 (2018), p. 135007.

[14] Zahra Kadkhodaie and Eero P Simoncelli. "Solving linear inverse problems using the prior implicit in a denoiser". In: *arXiv preprint arXiv:2007.13640* (2020).

[15] Bahjat Kawar, Gregory Vaksman, and Michael Elad. "SNIPS: Solving noisy inverse problems stochastically". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 21757–21769.

[16] Bahjat Kawar, Gregory Vaksman, and Michael Elad. "Stochastic Image Denoising by Sampling from the Posterior Distribution". In: *arXiv preprint arXiv:2101.09552* (2021).

[17] Orest Kupyn et al. "DeblurGAN-v2: Deblurring (orders-of-magnitude) faster and better". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 8878–8887.

[18] Stamatios Lefkimmiatis. "Non-local color image denoising with convolutional neural networks". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 3587–3596.

[19] Ziwei Liu et al. "Deep learning face attributes in the wild". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2015, pp. 3730–3738.

[20] Alice Lucas et al. "Using deep neural networks for inverse problems in imaging: beyond analytical methods". In: *IEEE Signal Processing Magazine* 35.1 (2018), pp. 20–36.

[21] Michael T McCann, Kyong Hwan Jin, and Michael Unser. "Convolutional neural networks for inverse problems in imaging: A review". In: *IEEE Signal Processing Magazine* 34.6 (2017), pp. 85–95.

[22] Sachit Menon et al. "PULSE: Self-supervised photo upsampling via latent space exploration of generative models". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 2437–2445.

[23] Shichong Peng and Ke Li. "Generating Unobserved Alternatives: A Case Study through Super-Resolution and Decompression". In: *arXiv preprint arXiv:2011.01926* (2020).

[24] Alec Radford, Luke Metz, and Soumith Chintala. "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks". In: *4th International Conference on Learning Representations*. 2016.

[25] Saiprasad Ravishankar, Jong Chul Ye, and Jeffrey A Fessler. "Image reconstruction: From sparsity to data-adaptive methods and machine learning". In: *Proceedings of the IEEE* 108.1 (2019), pp. 86–109.

[26] Gareth O Roberts, Richard L Tweedie, et al. "Exponential convergence of Langevin distributions and their discrete approximations". In: *Bernoulli* 2.4 (1996), pp. 341–363.

[27] Yaniv Romano, Michael Elad, and Peyman Milanfar. "The little engine that could: Regularization by denoising (RED)". In: *SIAM Journal on Imaging Sciences* 10.4 (2017), pp. 1804–1844.

[28] Umut Simsekli et al. "Stochastic Quasi-Newton Langevin Monte Carlo". In: *International Conference on Machine Learning*. PMLR, 2016, pp. 642–651.

[29] Yang Song and Stefano Ermon. "Generative modeling by estimating gradients of the data distribution". In: *Advances in Neural Information Processing Systems*. 2019, pp. 11918–11930.

[30] Yang Song and Stefano Ermon. "Improved techniques for training score-based generative models". In: *Advances in Neural Information Processing Systems*. 33. 2020.

[31] Yang Song et al. "Score-Based Generative Modeling through Stochastic Differential Equations". In: *International Conference on Learning Representations*. 2021.

[32] Maitreya Suin, Kuldeep Purohit, and AN Rajagopalan. "Spatially-attentive patch-hierarchical network for adaptive motion deblurring". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 3606–3615.

[33] Gregory Vaksman, Michael Elad, and Peyman Milanfar. "LIDIA: Lightweight learned image denoising with instance adaptation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 2020, pp. 524–525.

[34] Xintao Wang et al. "ESRGAN: Enhanced super-resolution generative adversarial networks". In: *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*. 2018.

[35] Jianchao Yang et al. "Image super-resolution via sparse representation". In: *IEEE transactions on image processing* 19.11 (2010), pp. 2861–2873.

[36] Fisher Yu et al. "LSUN: Construction of a large-scale image dataset using deep learning with humans in the loop". In: *arXiv preprint arXiv:1506.03365* (2015).

[37] Guoshen Yu, Guillermo Sapiro, and Stéphane Mallat. "Solving inverse problems with piecewise linear estimators: From Gaussian mixture models to structured sparsity". In: *IEEE Transactions on Image Processing* 21.5 (2011), pp. 2481–2499.

[38] Jian Zhang and Bernard Ghanem. "ISTA-Net: Interpretable optimization-inspired deep network for image compressive sensing". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 1828–1837.

[39] Kai Zhang, Wangmeng Zuo, and Lei Zhang. "FFDNet: Toward a fast and flexible solution for CNN-based image denoising". In: *IEEE Transactions on Image Processing* 27.9 (2018), pp. 4608–4622.

[40] Kai Zhang et al. "Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising". In: *IEEE Transactions on Image Processing* 26.7 (2017), pp. 3142–3155.

[41] Daniel Zoran and Yair Weiss. "From learning models of natural image patches to whole image restoration". In: *2011 International Conference on Computer Vision*. IEEE. 2011, pp. 479–486.