

SVNR: Spatially-variant Noise Removal with Denoising Diffusion

Naama Pearl^{†,1,2}, Yaron Brodsky¹, Dana Berman¹, Assaf Zomet¹, Alex Rav Acha¹, Daniel Cohen-Or^{†,1,3} and Dani Lischinski^{†,1,4}

¹Google Research, ²University of Haifa,

³Tel Aviv University, ⁴The Hebrew University of Jerusalem

Abstract

Denoising diffusion models have recently shown impressive results in generative tasks. By learning powerful priors from huge collections of training images, such models are able to gradually modify complete noise to a clean natural image via a sequence of small denoising steps, seemingly making them well-suited for single image denoising. However, effectively applying denoising diffusion models to removal of realistic noise is more challenging than it may seem, since their formulation is based on additive white Gaussian noise, unlike noise in real-world images. In this work, we present SVNR, a novel formulation of denoising diffusion that assumes a more realistic, spatially-variant noise model. SVNR enables using the noisy input image as the starting point for the denoising diffusion process, in addition to conditioning the process on it. To this end, we adapt the diffusion process to allow each pixel to have its own time embedding, and propose training and inference schemes that support spatially-varying time maps. Our formulation also accounts for the correlation that exists between the condition image and the samples along the modified diffusion process. In our experiments we demonstrate the advantages of our approach over a strong diffusion model baseline, as well as over a state-of-the-art single image denoising method.

1. Introduction

Image denoising, the task of removing unwanted noise from an image, while preserving its original features, is one of the most longstanding problems in image processing. Over the years, numerous image denoising techniques have been developed, ranging from traditional filtering-based methods to more recent deep learning-based approaches, e.g., [24, 10, 38, 9, 13].

In modern real-world digital photographs, noise most commonly arises from the imaging sensor, and is particu-

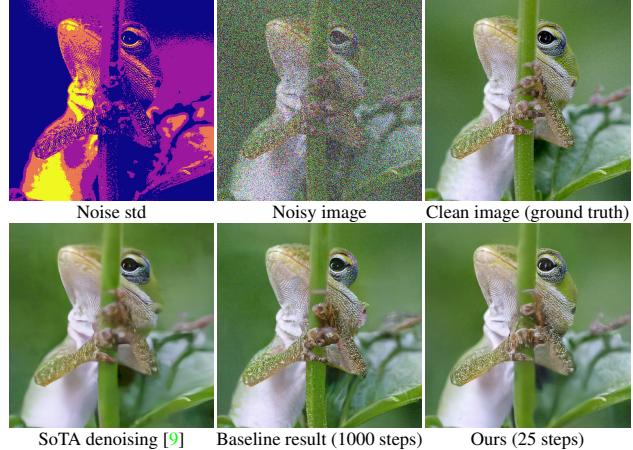


Figure 1: **Top:** spatially-variant standard deviation of noise (quantized), the resulting noisy image, and the ground truth clean image. Our SVNR formulation handles such noise by applying a pixel-wise time embedding. **Bottom:** state-of-the-art denoising methods manage to remove high levels of noise but over-smooth fine details. Diffusion based models are able to recover textures in the image even when they are hard to distinguish in the noisy image. SVNR yields clean images of higher fidelity (part of the lizard’s head is missing in the baseline result), while reducing the runtime ~ 10 .

larly evident when images are captured in low-light conditions. Yet, many of the proposed approaches make unrealistic assumptions regarding the noise and/or assess the denoising performance using metrics such as PSNR or SSIM. Such metrics struggle with the distortion-perception trade-off [4] as they are sensitive to pixel alignment and do not emphasize the restoration of fine details or high-frequency textures, which may be difficult to distinguish from noise.

In this paper, we propose a new denoising approach that leverages the natural image prior learned by today’s powerful diffusion-based generative models [15, 12]. Such models have been successfully applied to a variety of image restoration tasks [32, 30, 17, 18]. Furthermore, they pos-

[†] Performed this work while working at Google.

sess innate denoising capabilities, since the entire generation process is based on gradual denoising of images. Thus, one might expect that it should be possible to reconstruct a clean image simply by starting the diffusion process from the noisy input image. However, the diffusion process is based on additive white Gaussian noise (AWGN), while realistic noise models involve a signal-dependent component, the so-called shot-noise, which leads to higher noise levels in brighter parts of the image [20]. This violates the denoising diffusion formulation that associates a single scalar noise level (time) with each step, making it non-trivial to apply the diffusion process to realistic noise removal.

In this work, we present SVNR, a novel denoising diffusion formulation that handles *spatially-varying noise*, thereby enabling the reverse process to start from realistic noisy images, while significantly reducing the number of necessary diffusion steps.

Specifically, SVNR adapts the denoising diffusion framework to utilize the noisy input image as both the condition and the starting point. We assume a realistic signal-dependent noise model (Section 3.1), with a spatially-variant noise distribution. To cope with such a noise distribution, we adapt the diffusion process to allow each pixel to have its own time embedding, effectively assuming that the denoising time step is spatially-varying, rather than constant, across the image. We further present training and inference schemes that support such spatially-varying time maps. Our training scheme also accounts for correlation between the condition image and the samples of the diffusion process, which stems from the fact that the reverse process starts with the same image it is conditioned on.

The spatially-variant time embedding, together with the associated training scheme, enables using the noisy input image as both the condition and the starting point for the denoising process, yielding higher quality clean images (Fig. 1), while allowing significantly fewer denoising steps (Fig. 2). We demonstrate the power of the SVNR framework on simulated noisy images exhibiting a wide variety of noise levels and show its ability to generate fine details, such as fur and intricate textures. We show that our framework outperforms the standard conditioned diffusion baseline quantitatively, as well as visually, while avoiding the over-smoothing of a state-of-the-art single-image denoising method [9].

2. Background and Related Work

2.1. Image noise models

Cameras sensors convert incident photons to voltage readings, which are then converted to bits by an analog to digital converter (ADC). Throughout this process, noise is unavoidably added to the measurement, depending both on photon statistics and the sensor’s circuits. Sensor noise is

often modeled as a combination of two primary components [23]: shot noise, which originates from photon arrival statistics and is modeled as a Poisson process depending on signal intensity, and read noise, which is caused by imperfections in the readout circuitry and is modeled as a Gaussian noise with standard deviation σ_r .

2.2. Single image denoising

Early works for single image denoising used prior knowledge like non-local self-similarity in BM3D [10] or total variation [24].

Recently, convolutional neural networks (CNNs) have shown their success in single image denoising, as summarized in this comprehensive survey [13]. The following methods require a clean target image to train the CNNs. Initially, they were trained on synthetically added i.i.d. Gaussian noise, however that practice fails to generalize to real noisy images [27]. Later, datasets of real noisy images with their clean counterparts were collected (SIDD [1], RENOIR [2]), and are commonly used for denoising evaluation. As shown in [34], learning the noise distribution of real images via a GAN, which is used to synthesize noise for a denoising network, significantly improves performance. DnCNN [38] predicts the residual image (the noise) of a noisy image. Many works improved the performance by choosing better architectural components: SADNet [6] proposes a deformable convolution to adjust for different textures and noise patterns, HINet [9] introduces instance normalization block for image restoration tasks and NAFNet [8] suggests to replace non linear activation functions by element-wise multiplication between two sets of channels. Some methods iteratively solve the problem in a multi-scale architecture or in multiple iterations: MPRNet [37] proposes supervised attention block between the different stages to leverage the restored image features at different scales. Somewhat similarly to our work, FFDNet [39] employs a spatially-varying noise-map, and is able to remove non-uniform noise. However the architecture of FFDNet relies on downsampling and channel re-shuffle before applying a CNN to the image, which is different than the proposed approach.

Unlike the above works, which require clean target images, another line of works focuses on unsupervised or self-supervised solutions. According to N2N [19], the expected value of minimizing the objective with respect to clean samples is similar to minimizing it with respect to different noisy samples, and therefore clean images are not necessary. Further works designed different ways for data augmentation that achieve the same purpose. N2S [3], Noisier2noise [22], R2R [25], neighbor2neighbor [16] use different subsamples of the image as instances of the noisy image. IDR [41] added noise to the noisy image to create a noisier version which can be supervised by the noisy image.

2.2.1 Raw single image denoising / low light methods

Some methods take into account the image formation model and aim to denoise the raw image, where the pixel values directly relate to the number of incident photons and the noise can be better modeled. To tackle the task of low-light imaging directly, SID [7] introduces a dataset of raw short-exposure low-light images paired with corresponding long-exposure reference images. They train an end-to-end CNN to perform the majority of the steps of the image processing pipeline: color transformations, demosaicing, noise reduction, and image enhancement. Brooks *et al.* [5] present a technique to “unprocess” the image processing pipeline in order to synthesize realistic raw sensor images, which can be further used for training. Wei *et al.* [35] accurately formulate the noise formation model based on the characteristics of CMOS sensors. Punnappurath *et al.* [28] suggest a method that generates nighttime images from day images. Similarly, in the field of low light video, Monakhova *et al.* [21] learn to generate nighttime frames of video.

2.3. Diffusion models

The usage of diffusion models for generative tasks grew rapidly over the past years, and have shown great success in text-to-image generation (Imagen [31], DALL-E 2 [29]). Denoising is a key component of the diffusion process, offering a strong image prior for both restoration and generative tasks. SR3 [32] adapts denoising diffusion probabilistic models to solve the super resolution task, conditioned on the low resolution image. Palette [30] extended this idea to a general framework for image-to-image translation tasks, including colorization, inpainting, uncropping, and JPEG restoration. In our evaluation, we compare to this method as a baseline, where the noisy image is given as a prior, but without modifying the diffusion formulation. Kawar *et al.* [18, 17] solve linear inverse image restoration problems by sampling from the posterior distribution, based on a pre-trained denoising diffusion model. This approach is limited to linear problems, whereas a realistic noise model is signal-dependant and not additive Gaussian. In a concurrent work, Xie *et al.* [36] redefine the diffusion process to implement generative image denoising, however it is defined for different types of noise (Gaussian, Poisson) separately, while a realistic noise model is a combination of both.

3. Method

Our main goal in this work is to leverage the powerful denoising-based diffusion framework for noise removal. To this end, we adapt the framework to enable the noisy input image to be considered as a time step in the diffusion process. Accounting for the more complex nature of real camera noise, we propose a diffusion formulation that unifies realistic image noise with that of the diffusion process.

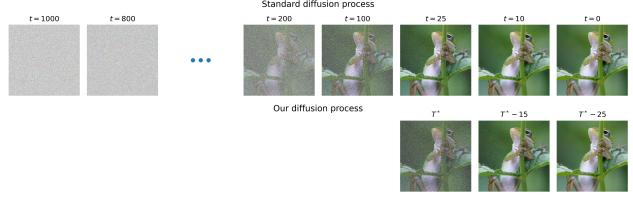


Figure 2: **Top:** standard forward diffusion process (2). The reverse denoising process starts from complete noise (left) and iterates for 1000 time-steps. **Bottom:** our diffusion formulation enables starting the reverse diffusion process from the noisy input image, requiring ~ 20 iterations.

In Section 3.1, we describe the camera noise model that we use, and in Sections 3.2–3.3 we propose a diffusion process that can incorporate such noisy images as its samples.

For a more realistic modeling of noisy images, we consider a raw-sensor noise model, which is not uniform across the image. This means that we cannot pair a step in the diffusion process with a single point in time. Instead, we pair each diffusion step with a spatially varying *time map*, where each pixel may have a different time encoding (Section 3.3). The training and the inference schemes are modified to support such time maps, as described in Section 3.4.

In particular, the starting point of the diffusion process is set to the noisy input image, and not to an i.i.d Gaussian noise. This has the additional advantage of significantly reducing the number of diffusion steps (~ 50 times fewer steps in our experiments), see Fig. 2. However, using the same noisy input image as both the condition and the starting point of the diffusion process, introduces another challenge: there is a correlation between the condition and the samples along the reverse diffusion process at inference time, a correlation that is not reflected in the training scheme. We address this challenge in Section 3.5, give a theoretical analysis of this phenomenon and propose a modified training scheme to overcome it.

Notation and setting: Below we use small italics (*e.g.*, x) to denote scalars, while bold roman letters (*e.g.*, \mathbf{x}) denote vectors. Images and other per-pixel maps are represented as vectors in $\mathbb{R}^{H \times W \times 3}$. In particular, $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ is a noise vector with the same dimensions, whose elements are sampled from $\mathcal{N}(0, 1)$. The operations $\mathbf{a} \cdot \mathbf{b}$ and $\frac{\mathbf{a}}{\mathbf{b}}$ between two vectors \mathbf{a} and \mathbf{b} , denote element-wise multiplication and division respectively.

3.1. Noise model

We adopt a noise model that is commonly used for sensor raw data [20, 26]. The noisy version $\mathbf{y} \in \mathbb{R}^{H \times W \times 3}$ of a

clean linear image $\mathbf{x}_0 \in \mathbb{R}^{H \times W \times 3}$ is given by:

$$\begin{aligned}\mathbf{y} &= \mathbf{x}_0 + \boldsymbol{\sigma}_p \cdot \boldsymbol{\epsilon}_y, \quad \boldsymbol{\epsilon}_y \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \\ \boldsymbol{\sigma}_p &\triangleq \sqrt{\sigma_r^2 + \sigma_s^2 \mathbf{x}_0},\end{aligned}\quad (1)$$

where $\boldsymbol{\epsilon}_y \in \mathbb{R}^{H \times W \times 3}$ and $\boldsymbol{\sigma}_p$ is the per-pixel standard deviation of the noise, defined as a combination of σ_r , the standard deviation for the *signal-independent* read-noise, and σ_s for the *signal-dependent* shot-noise. See Section 4.1 for further details regarding our experiments.

3.2. Diffusion process definition

Given a clean image \mathbf{x}_0 and a noise schedule $\{\beta_t\}_{t=1}^T$, the standard diffusion process of length T is given by:

$$\begin{aligned}q(\mathbf{x}_t | \mathbf{x}_{t-1}) &= \mathcal{N}\left(\mathbf{x}_t; \sqrt{1 - \beta_t} \mathbf{x}_{t-1}, \beta_t \mathbf{I}\right), \\ \bar{\alpha}_t &= \prod_{i=1}^t \alpha_i = \prod_{i=1}^t (1 - \beta_i), \\ q(\mathbf{x}_t | \mathbf{x}_0) &= \mathcal{N}\left(\mathbf{x}_t; \sqrt{\bar{\alpha}_t} \mathbf{x}_0, (1 - \bar{\alpha}_t) \mathbf{I}\right).\end{aligned}\quad (2)$$

Note that this formulation defines a Markovian process, i.e., the variance of \mathbf{x}_t along the process is constant (assuming $\mathbb{E}(\mathbf{x}_0) = \mathbf{0}$ and $\text{Var}(\mathbf{x}_0) = \mathbf{I}$). As the noise level increases, the stationary nature of \mathbf{x}_t is achieved by attenuating the clean signal by a factor of $\sqrt{\bar{\alpha}_t}$. To be able to refer to \mathbf{y} as a sample from the diffusion process, we need to overcome two obstacles. The first issue is that in our noise model, the signal is not attenuated, and the second is that our noise model uses a spatially-varying noise distribution. We first resolve the former issue and modify the diffusion process to be non-stationary, by considering a process which does not attenuate the signal:

$$\begin{aligned}q(\mathbf{x}_t | \mathbf{x}_{t-1}) &= \mathcal{N}(\mathbf{x}_t; \mathbf{x}_{t-1}, \eta_t \mathbf{I}), \\ q(\mathbf{x}_t | \mathbf{x}_0) &= \mathcal{N}(\mathbf{x}_t; \mathbf{x}_0, \gamma_t \mathbf{I}), \\ \gamma_t &= \sum_{i=1}^t \eta_i,\end{aligned}\quad (3)$$

for some noise schedule $\{\eta_t\}_{t=1}^T$. This process, where $\text{Var}(\mathbf{x}_t | \mathbf{x}_0) \rightarrow \infty$ as $t \rightarrow \infty$, is termed “Variance Exploding” by Song *et al.* [33].

We wish to keep the noise schedule similar to the original DDPM schedule [15]. Hence we choose the noise schedule η_t so that γ_t will be a scaled version of $1 - \bar{\alpha}_t$, that is, $\gamma_t = \lambda(1 - \bar{\alpha}_t)$ for some λ . This implies,

$$\eta_t = \lambda \beta_t \prod_{i=1}^{t-1} (1 - \beta_i). \quad (4)$$

This non-stationary forward process, yields a reverse pro-

cess of the same form as in the standard diffusion,

$$\begin{aligned}q(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{x}_0) &= \mathcal{N}(\mathbf{x}_{t-1}; \tilde{\mu}_t(\mathbf{x}_t, \mathbf{x}_0), \tilde{\eta}_t \mathbf{I}), \\ \tilde{\mu}_t(\mathbf{x}_t, \mathbf{x}_0) &= \frac{\gamma_{t-1}}{\gamma_t} \mathbf{x}_t + \frac{\eta_t}{\gamma_t} \mathbf{x}_0, \\ \tilde{\eta}_t &= \frac{\gamma_{t-1} \eta_t}{\gamma_t}.\end{aligned}\quad (5)$$

The fact that our noise model does not attenuate the clean signal \mathbf{x}_0 is reflected in the expression for $\tilde{\mu}_t$, that lacks the multiplication by the attenuation factor $\alpha, \bar{\alpha}$. More details can be found in the supplementary materials.

At inference time, the diffusion process should start with $\mathbf{x}_T = \mathbf{x}_0 + \sqrt{\lambda} \boldsymbol{\epsilon}_T$, $\boldsymbol{\epsilon}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Note that in our noise model one cannot start the reverse process from pure noise (as done in standard diffusion processes), since the signal is not attenuated to 0. However, since our goal is to start the reverse process from the input noisy image, this is not a concern.

3.3. Spatially-variant time embedding

Our noise schedule, Eq. (3), defines a noise level γ_t for every integer t between 0 and $T = 1000$. As in standard diffusion models, we can extend the definition of γ_t to non-integer t using interpolation. Thus, given a noise level σ^2 , we can find a time t at which this noise level is attained. Consider now our camera noise model, Eq. (1). Each pixel p has a different noise level $\boldsymbol{\sigma}_p^2(p)$, and thus a corresponding time value that yields this noise level. The maximum noise level over the three channels defines a time map $\mathbf{T}^* \in \mathbb{R}^{H \times W}$ for which $\gamma_{\mathbf{T}^*(p)} = \max_{c \in \text{R,G,B}} \boldsymbol{\sigma}_p^2(p_c)$. In other words, we think of each pixel as being at its own stage of the diffusion process. Note that the time map \mathbf{T}^* encodes the spatially-varying noise of the entire input image \mathbf{y} . Hence we denote

$$\mathbf{x}_{\mathbf{T}^*} \triangleq \mathbf{y}, \quad \boldsymbol{\epsilon}_{\mathbf{T}^*} \triangleq \boldsymbol{\epsilon}_y, \quad \gamma_{\mathbf{T}^*} \triangleq \max_{\text{R,G,B}} \boldsymbol{\sigma}_p^2. \quad (6)$$

In practice, when presented with a noisy image \mathbf{y} , we do not know the actual noise level $\boldsymbol{\sigma}_p$, even if σ_r and σ_s are known, since the original clean signal \mathbf{x}_0 is not available. Thus, we follow common practice [20] and estimate it using a clipped version of the noisy image, to obtain $\hat{\mathbf{T}}^*$ such that

$$\begin{aligned}\gamma_{\hat{\mathbf{T}}^*} &= \max_{\text{R,G,B}} \hat{\boldsymbol{\sigma}}_p^2 \\ \hat{\boldsymbol{\sigma}}_p^2 &= \sqrt{\sigma_r^2 + \sigma_s^2 \cdot \text{clip}(\mathbf{y}, 0, 1)}.\end{aligned}\quad (7)$$

A standard diffusion model receives as input both \mathbf{x}_t and a time value t , indicating the signal noise level over the entire image. An embedding vector of the time is then used to apply an affine transformation independently to each pixel feature in \mathbf{x}_t . By replacing t with a spatially-varying time map \mathbf{T}^* , and computing a different time embedding per

pixel, we can make the model dependent on the spatially-varying noise level σ_p . However, since each pixel can now be at a different stage of the diffusion process, it requires a different number of steps to reach time 0. Hence, we need to develop new training and inference schemes to account for this, which are presented below.

3.4. Training and inference schemes

Our diffusion model receives as input a noisy image y and a time map T^* . We present training and inference schemes that account for this change. Our algorithm is summarized in Algs. 1 and 2.

Note that the reverse diffusion process, Eq. (5), operates on each pixel independently. Thus, we can use the same reverse process even with a spatially-varying time step T^* . However, each pixel may require a different number of steps before reaching time 0. We handle this by stopping the reverse process once a pixel reaches a negative time. In other words, the time map after t_0 denoising steps will be $(T^* - t_0)^+ \triangleq \max\{T^* - t_0, 0\}$.

During training, given a clean image x_0 , we sample σ_r , σ_s , and a random noise $\epsilon_y = \epsilon_{T^*}$. The noisy image y is then generated according to the noise model Eq. (1), and the estimated induced time map \hat{T}^* is calculated by Eq. (7). Next, we sample a scalar t_0 between 0 and the maximal value of \hat{T}^* , and advance the times of all the pixels by t_0 steps, to obtain $\hat{t} = (\hat{T}^* - t_0)^+$. We then sample a random Gaussian noise $\epsilon_{\hat{t}}$ and construct a sample $x_{\hat{t}} = x_0 + \gamma_{\hat{t}} \epsilon_{\hat{t}}$ of the diffusion process according to Eq. (3). Note that $\gamma_{\hat{t}}$ is a matrix, so the noise level is spatially-varying. The network then tries to predict $\epsilon_{\hat{t}}$ from the diffusion sample $x_{\hat{t}}$, the time map \hat{t} , and the condition image y .

At inference time, we get a noisy image y and its σ_r , σ_s . First, we estimate the time map \hat{T}^* by Eq. (7). We feed the network with y as the condition image, \hat{T}^* as the time map, and $y = x_{T^*}$ as the diffusion sample. The network outputs an estimate of the noise $\epsilon_{\hat{T}^*}$, from which we can compute an estimate of the original image \hat{x}_0 . We then use the reverse process Eq. (5) (replacing x_0 by \hat{x}_0) to produce the next sample. Additionally, we promote the time map \hat{T}^* by one step, *i.e.*, we replace \hat{T}^* with $\hat{t} = (\hat{T}^* - 1)^+$. We then run the network with our new sample and the promoted \hat{t} (using the same condition y), and continue in this manner until we reach $\hat{t} = 0$ for all pixels.

Explicitly, the reverse process is performed by sampling a Gaussian noise $\epsilon_{\hat{t}-1} \sim \mathcal{N}(0, I)$ and computing

$$x_{\hat{t}-1} = \frac{\gamma_{\hat{t}-1}}{\gamma_{\hat{t}}} x_{\hat{t}} + \frac{\eta_{\hat{t}}}{\gamma_{\hat{t}}} \hat{x}_0 + \sqrt{\frac{\gamma_{\hat{t}-1} \eta_{\hat{t}}}{\gamma_{\hat{t}}}} \epsilon_{\hat{t}-1}, \quad (8)$$

where in $\hat{t} - 1$ we clip the negative values, and $\gamma_{\hat{t}}$, $\gamma_{\hat{t}-1}$, $\eta_{\hat{t}}$ are all vectors of the same dimension as x_0 , whose values depend on the initial noise in the image. To avoid further

denoising of pixels whose time has reached 0, we override their values after the prediction by the network.

Algorithm 1: Training diffusion initialized with y

```

1 for  $i = 1, \dots$  do
2   Sample  $x_0, \sigma_r, \sigma_s$ 
3   Sample  $y$  by Eq. (1)
4   Calculate  $\hat{T}^*$  by Eq. (7)
5   Sample  $t_0 \sim \mathcal{U}[0, \max(\hat{T}^*)]$ 
6   Set  $\hat{t} = \max\{\hat{T}^* - t_0, 0\}$ 
7   Calculate  $x_{\hat{t}}$  by Eq. (11)
8    $\hat{x}_0 = \text{SVNR}(y, x_{\hat{t}}, \hat{t})$ 
9   Calculate loss and update weights.

```

Algorithm 2: Inference by diffusion from y

Inputs: y, σ_r, σ_s

```

1 Calculate  $\hat{T}^*$  by Eq. (7)
2 Set  $\hat{t} = \hat{T}^*, x_{\hat{t}} = y$ 
3 while any( $\hat{t} > 0$ ) do
4    $\hat{x}_0 = \text{SVNR}(y, x_{\hat{t}}, \hat{t})$ 
5   Sample  $x_{(\hat{t}-1)^+}$  by Eq. (8)
6   Override pixels that will reach  $(t - 1)^+ = 0$ 
      with the values in  $\hat{x}_0$ . These values remain
      fixed for the rest of the process.
7   Set  $\hat{t} = (\hat{t} - 1)^+, x_{\hat{t}} = x_{(\hat{t}-1)^+}$ 

```

3.5. Noise correlation in the reverse process

Next, we discuss a phenomenon that arises when we initialize the process with the noisy input image *and* condition the process on it. The key observation is that throughout the reverse diffusion process, there is a correlation between the noise component of the diffusion sample x_t and the noise component of the condition image $y = x_{T^*}$.

When initializing the diffusion process with x_{T^*} , the first reverse step yields a sample x_{T^*-1} derived from Eq. (5). This sample is less noisy than x_{T^*} and can be explicitly written (given x_0) as

$$x_{T^*-1} = \frac{\gamma_{T^*-1}}{\gamma_{T^*}} x_{T^*} + \frac{\eta_{T^*}}{\gamma_{T^*}} x_0 + \sqrt{\frac{\gamma_{T^*-1} \eta_{T^*}}{\gamma_{T^*}}} \epsilon_{T^*-1}. \quad (9)$$

Using Eq. (1) it can be rewritten as a summation of x_0 and an additional noise term, which is a linear combination between the noise ϵ_{T^*} and the new sampled noise term ϵ_{T^*-1} ,

$$x_{T^*-1} = x_0 + \frac{\gamma_{T^*-1}}{\sqrt{\gamma_{T^*}}} \epsilon_{T^*} + \sqrt{\gamma_{T^*-1} \left(1 - \frac{\gamma_{T^*-1}}{\gamma_{T^*}}\right)} \epsilon_{T^*-1}. \quad (10)$$

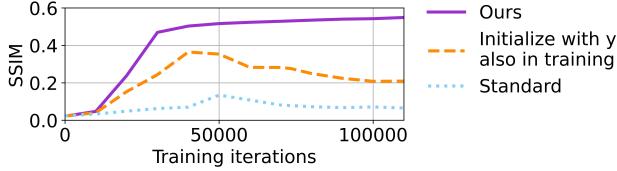


Figure 3: SSIM of validation during training. The standard training scheme (light blue) cannot restore the signal. Initializing the diffusion with the noisy image also in training (orange) partially solves the problem, but over time the network utilizes the two realizations of the noise (from the conditioned image and the diffusion sample) that are not available during inference. Our training scheme (purple) that relies on Eq.(11) yields stable training.

After t_0 inference steps, the time map is $\mathbf{t} = (\mathbf{T}^* - t_0)^+$ and \mathbf{x}_t can be written as

$$\begin{aligned}\mathbf{x}_t &= \mathbf{x}_0 + \frac{\gamma_t}{\sqrt{\gamma_{\mathbf{T}^*}}} \epsilon_{\mathbf{T}^*} + \sqrt{\gamma_t \left(1 - \frac{\gamma_t}{\gamma_{\mathbf{T}^*}}\right)} \epsilon_t, \\ &= \mathbf{x}_0 + \sqrt{\gamma_t} \tilde{\epsilon}_t.\end{aligned}\quad (11)$$

The full derivation can be found in the supplementary materials. The modified noise $\tilde{\epsilon}_t$ is a linear combination between the initial noise of $\epsilon_{\mathbf{T}^*}$ and another i.i.d noise term, ϵ_t ,

$$\tilde{\epsilon}_t = \sqrt{\frac{\gamma_t}{\gamma_{\mathbf{T}^*}}} \epsilon_{\mathbf{T}^*} + \sqrt{1 - \frac{\gamma_t}{\gamma_{\mathbf{T}^*}}} \epsilon_t. \quad (12)$$

This relationship describes the correlation between $\tilde{\epsilon}_t$, the noise component of the diffusion sample \mathbf{x}_t , and $\epsilon_{\mathbf{T}^*}$, the noise component of the condition image $\mathbf{y} = \mathbf{x}_{\mathbf{T}^*}$.

Because of the above correlation, at train time the network sees a different distribution than at inference time. During training, the noise of the diffusion sample \mathbf{x}_t consists entirely of noise sampled independently from $\epsilon_{\mathbf{T}^*}$. Hence, at train time, the \mathbf{x}_t and \mathbf{y} presented to the network are two independent degradations of the true signal \mathbf{x}_0 . This effect is made clearer when one considers the first step (*i.e.*, $t_0 = 0$). While at train time the network sees two independent samples of \mathbf{x}_0 noised with σ_p , at inference time the two images are the same.

Indeed, looking at the progress of inference error in Fig. 3, we see a sudden drop of quality, which can be explained by the fact that the network may be learning to utilize its two uncorrelated inputs, which does not generalize to the inference process.

A naive solution to this problem would be to drop the conditioning entirely, however, our ablation study shows that this yields deteriorated results. The experiments suggest that it stems mainly from the clipping of negative values, which violates the noise model.

Thus, we choose to pursue a different approach and modify the training scheme to explicitly account for this correla-

tion. Specifically, we propose to sample \mathbf{x}_t during training according to Eq. (11), in order to simulate a distribution of inputs that is similar to that of inference time. As noted above, a special case of this noise correlation is when $t_0 = 0$ and $\mathbf{y} = \mathbf{x}_{\mathbf{T}^*}$. We increase the probability of those cases to 1% of the training iterations.

4. Results

We test our method on natural images from the ImageNet dataset [11], corrupted by simulated noise that was generated by our noise model (Eq. (1)). For training we use the full training set of ImageNet, and for evaluation we use a subset of 2000 images from the ImageNet validation set.

We compare our results to a strong diffusion baseline, based on the framework of [32, 30], that was trained to solve the task of image denoising (conditioned on the noisy image), in addition to a state-of-the-art single image denoising method [9]. We report quantitative PSNR, SSIM, LPIPS [40] and FID [14] metrics for all of the models and datasets. While the former three metrics are used to compare pairs of images, the FID metric is used to compare entire distributions. We include this metric to asses the overall similarity between the distribution of the ground truth clean images and the distribution of the denoised results.

4.1. Data and implementation details

Noise simulation: The noise model in Eq. (1) is defined with respect to linear images. Hence, we first “linearize” the images by applying inverse gamma-correction and inverse white level. For white level values, during training we sample a value in the range $[0.1, 1]$, and use 0.5 during validation.

We train the network on a range of values for σ_r, σ_s and evaluate the method on fixed gain levels of an example camera, defined in [20]. Following [26], we consider a wider training region and higher gain levels in our evaluation. See Fig. 4 for the specific values used during training and evaluation.

To make the noisy images more realistic, we further clip the images at 0 after the addition of noise, as negative values are not attainable in real sensors. Our network seems to overcome this discrepancy between the theoretical model and the data distribution we use in practice. We do not clip the image at higher values, as it can be adjusted with exposure time. We use crops of 256×256 for training and a set of 2000 images for validation, cropped to the maximum square and resized to 1024×1024 . The noise is added after the resizing, so we do not change the noise distribution.

Implementation details: Before being fed into the network, the input noisy images are scaled to occupy the full range of $[-1, 1]$ to match the diffusion models assumption.

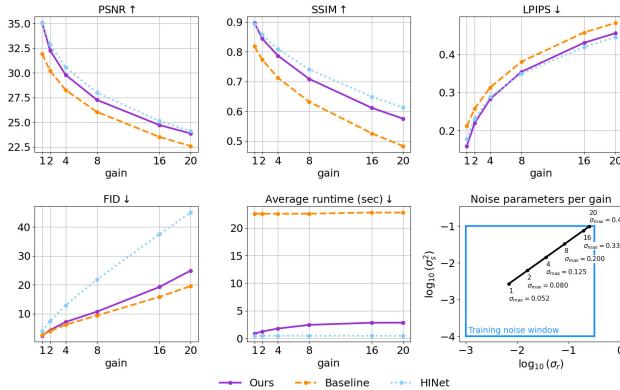


Figure 4: Quantitative results for simulated noise across different noise levels. We compare the diffusion baseline, a single image denoising method [9] and our method. The metrics we report are PSNR, SSIM, LPIPS [40] and FID [14]. In addition, average runtimes are presented for the diffusion methods. The noise is simulated using noise model in Eq. (1). During training, the noise parameters are sampled from the blue rectangle. At inference time, we use a set of fixed noise parameters that correspond to various gain levels of an example camera, as described in [20].

The noise standard deviation is scaled accordingly. The input to the network has 6 channels: 3 RGB channels of the noisy image y (condition) and 3 RGB channels of the sample in the diffusion process x_t . In addition, the network is also given as input the spatially-varying time map, which is computed from the known noise parameters σ_r, σ_s . At inference time the sample of the diffusion process is initialized with the noise image y and the estimated \hat{T}^* .

We fine-tune a fully-convolutional version of the Imagen model [31], disregarding the text components and conditioning it on the degraded input image, as done in [30, 32]. We use $\{\beta_t\}_{t=1}^T$ that are linearly spaced in the range $[0.02, 10^{-8}]$ and $T = 1000$ for the standard diffusion in Eq. (2), and $\lambda = 20$ for the modified noise schedule in Eq. (4). We train the network on 8 TPU-v4 chips, for $900K$ iterations and follow the training optimization of [31], with Adam optimizer and learning rate scheduler with linear warm-up followed by cosine decay. The training phase takes three days.

4.2. Results on ImageNet

We evaluate our method on a subset of 2000 images from the ImageNet dataset [11] and report metrics for noise levels corresponding to gains ranging from 1 to 20. Note that while the input to the network are “linearized” images, the metrics are calculated on the reprocessed images, *i.e.*, after readjusting the white level and reapplying the gamma correction. As mentioned before, we compare our results to a

strong diffusion baseline, as well as to HINet, a state-of-the-art single image denoising method [9]. For a fair comparison, we retrain HINet on the same dataset and noise levels that we used. Quantitative results for PSNR, SSIM, LPIPS and FID metrics are reported in Fig. 4, as well as the average runtime per example (in seconds).

Compared to the state-of-the-art model, our method (SVNR) shows slightly worse performance in all “pixel-to-pixel” metrics, while achieving a significantly better FID score. On the other hand, the baseline diffusion model outperforms our model in the FID metric but exhibits significantly worse results in all other metrics. This nicely demonstrates how our approach balances the perception-distortion trade-off [4]. We can see that the baseline diffusion model favours realistic images at the expense of lower fidelity to the clean signal, while the state-of-the-art model shows the best fidelity to the signal at the cost of drifting away from the input distribution. In contrast, SVNR manages to keep a relatively high signal fidelity without the significant distribution drift.

This can be further seen in Fig. 5 and Fig. 6, where we showcase denoising results of these three models for several inputs with noise gain of 16 (comparisons at other noise levels are included in the supplementary). Even at this relatively high noise level, all three models manage to remove most of the noise. However, the results of HINet suffer from considerable over-smoothing and lack high-frequency details. On the other hand, both SVNR and the baseline diffusion models manage to generate fine details. While the baseline diffusion model generally generates more details than SVNR, it eliminates less noise (top example) and furthermore, occasionally exhibits hallucinations (see the first two examples). We hypothesize that this difference between our method and the baseline stems from fine-tuning the baseline to adapt it to our diffusion noise model, Eq. (3). We conjecture that fine-tuning causes the model to lose some of its prior, instead allowing it to make more effective use of the underlying signal, by using the noisy image as the starting point.

Overall, we see that our method yields comparable performance to the state-of-the-art, while producing more realistic images. At the same time, our method retains more fidelity to the underlying signal and removes more noise than the baseline diffusion approach.

Since the diffusion baseline always starts from complete noise, its runtime is fixed (~ 22 seconds), regardless of the noise level in the input image. Starting the diffusion process from the noisy image in SVNR yields results in runtime that depends on the noise levels in the image, ranging from ~ 3 seconds to less than a second for the least noisy images.

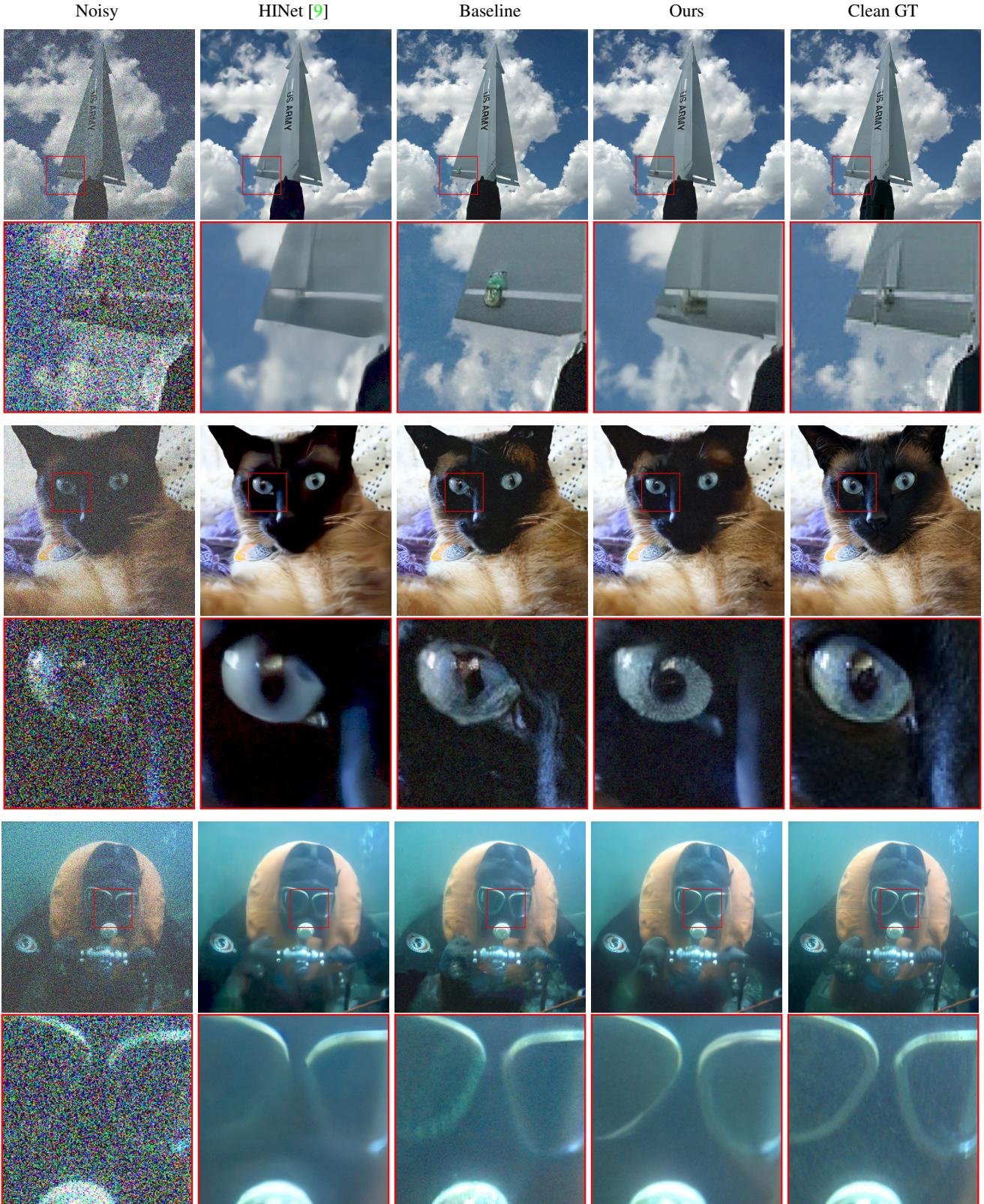


Figure 5: Comparison between different denoising methods on images with noise gain of 16.

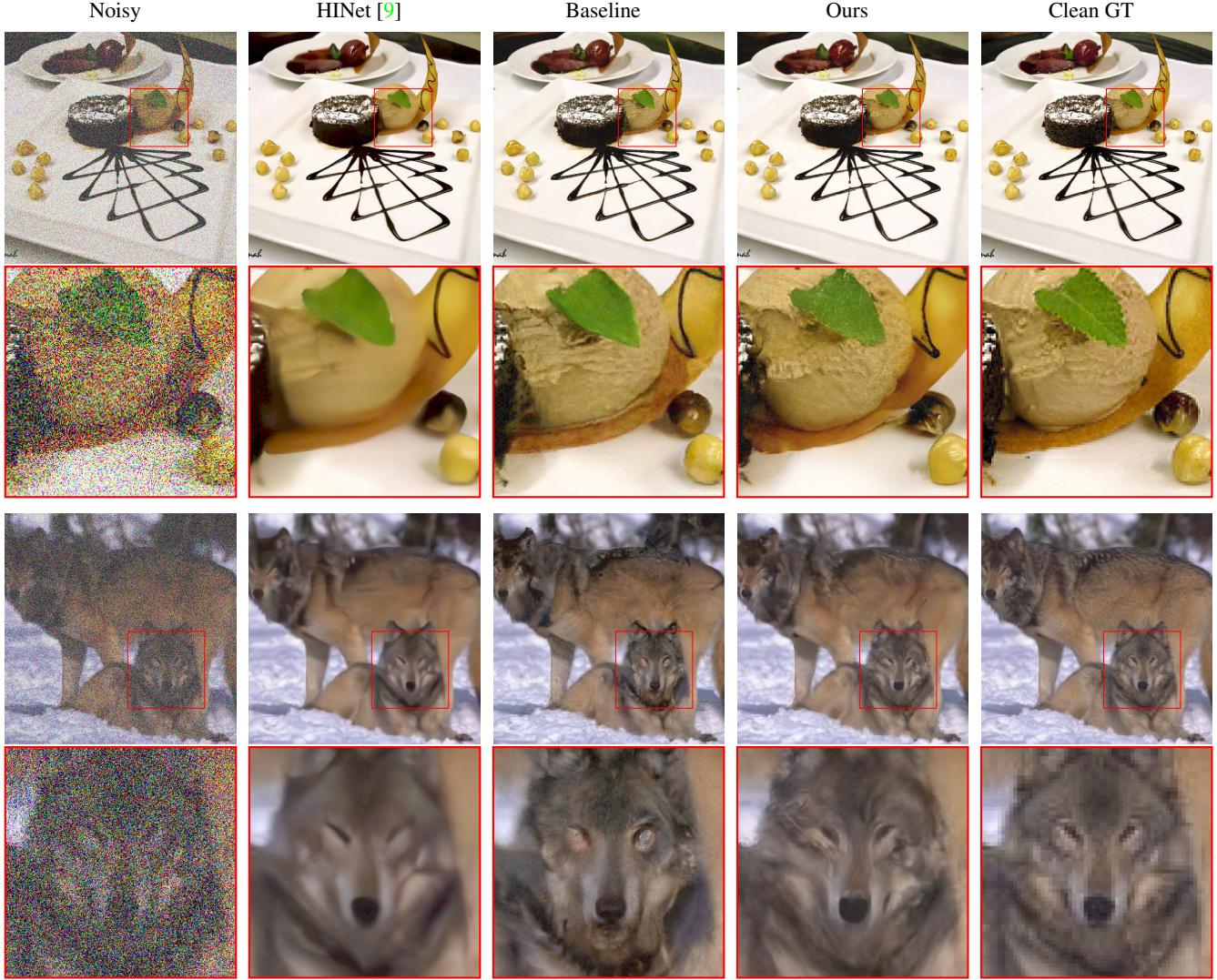


Figure 6: Comparison between different denoising methods on images with noise gain of 16.

4.3. Ablation

We validate the importance of different aspects of our approach by the ablation study in Table 1. We compare the results to the baseline diffusion model that is initialized with *complete noise* and conditioned on the noisy image (denoted A in the table) and to versions where diffusion is initialized with the *noisy input image* (denoted by B, C). When initializing the diffusion process with the noisy image, we consider unconditioned (B) and conditioned (C) variants.

The *unconditioned* variants differ in the type of their input images: B1, where the input values are clipped to avoid negative values; and B2, a variant where input images are allowed to have negative values. For the *conditioned* setup we consider three training schemes: C1, the standard training process, and two versions that try to handle the corre-

tion described in Section 3.5 – C2, a version that enforces the starting point of the diffusion x_{T^*} to be equal to the noisy input y in 1% of training iterations; and C3, our full SVNR framework that incorporates Eq. (11). All the ablation experiments are done with gain level 16, and the results are averaged over 80 images.

The comparison to the baseline A is discussed in the previous section. The *unconditioned* version B1 fails to restore the clean signal, mainly because it is not robust to the zero clipped values. When the original noisy image is not available during the process, the prediction of x_t at each diffusion step is shifted and “loses” the correct intensity levels. This is supported by the comparison with B2.

The standard *conditioned* version C1 emphasizes the importance of our training scheme that takes into account the

	PSNR ↑	SSIM ↑	LPIPS ↓
Initialized with complete noise			
A Conditioned (baseline)	23.76	0.46	0.441
Initialized with y			
B1 Unconditioned	15.71	0.41	0.508
B2 Unconditioned, without clipping	22.25	0.36	0.520
C1 Conditioned, standard training	12.59	0.07	0.759
C2 Conditioned, oversampling $\mathbf{x}_{T^*} = \mathbf{y}$	16.06	0.16	0.665
C3 SVNR	24.56	0.54	0.438

Table 1: Ablation study (under noise gain 16), averaged over 80 images. See Section 4.3 for details.

correlation between the two sources of noise. In C2, we practically apply Eq. (11) only for the first step of diffusion and only for 1% of the training iterations (as explained in Section 3.5, this is equivalent to training on samples with $\mathbf{x}_{T^*} = \mathbf{y}$), which slightly improves the results. However, to achieve good restoration, one must consider the correlation throughout the entire process, which is supported by the improved results achieved by our training scheme C3.

5. Conclusions

We have presented a new diffusion-based framework for the task of single image denoising, which leverages the natural rich image prior learned by generative denoising diffusion models. Our framework adapts denoising diffusion to utilize the noisy input image as both the condition and the starting point of the diffusion process. To enable the integration of a realistic noisy image as a sample in the diffusion process, we have proposed a novel denoising diffusion formulation that admits a spatially-variant time embedding, with supporting training and inference schemes.

We believe that this novel formulation can be potentially applied to any non-uniform noise distribution. Additionally, we have addressed a phenomenon that occurs when initializing and conditioning the diffusion process with the same noisy input image, and have mitigated it with a suitable training scheme. Our qualitative and quantitative results show improved handling of the distortion-perception trade-off, balancing faithful image reconstruction with generation of realistic fine details and textures. Furthermore, our formulation also significantly reduces the number of required diffusion steps. In the future, we aim to further distill the rich knowledge hidden in the backbone model, and expand the scope and applicability of our approach to complex real-world scenarios.

References

- [1] Abdelrahman Abdelhamed, Stephen Lin, and Michael S Brown. A high-quality denoising dataset for smartphone cameras. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1692–1700, 2018. [2](#)
- [2] Josue Anaya and Adrian Barbu. Renoir—a dataset for real low-light image noise reduction. *Journal of Visual Communication and Image Representation*, 51:144–154, 2018. [2](#)
- [3] Joshua Batson and Loic Royer. Noise2self: Blind denoising by self-supervision. In *International Conference on Machine Learning*, pages 524–533. PMLR, 2019. [2](#)
- [4] Yochai Blau and Tomer Michaeli. The perception-distortion tradeoff. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6228–6237, 2018. [1, 7](#)
- [5] Tim Brooks, Ben Mildenhall, Tianfan Xue, Jiawen Chen, Dillon Sharlet, and Jonathan T Barron. Unprocessing images for learned raw denoising. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11036–11045, 2019. [3](#)
- [6] Meng Chang, Qi Li, Huajun Feng, and Zhihai Xu. Spatial-adaptive network for single image denoising. In *European Conference on Computer Vision*, pages 171–187. Springer, 2020. [2](#)
- [7] Chen Chen, Qifeng Chen, Jia Xu, and Vladlen Koltun. Learning to see in the dark. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3291–3300, 2018. [3](#)
- [8] Liangyu Chen, Xiaojie Chu, Xiangyu Zhang, and Jian Sun. Simple baselines for image restoration. *European Conference on Computer Vision*, 2022. [2](#)
- [9] Liangyu Chen, Xin Lu, Jie Zhang, Xiaojie Chu, and Chengpeng Chen. Hinet: Half instance normalization network for image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 182–192, 2021. [1, 2, 6, 7, 8, 9, 14, 15, 16, 17](#)
- [10] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Transactions on image processing*, 16(8):2080–2095, 2007. [1, 2](#)
- [11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. [6, 7](#)
- [12] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat GANs on image synthesis. *Advances in Neural Information Processing Systems*, 34, 2021. [1](#)
- [13] Michael Elad, Bahjat Kawar, and Gregory Vaksman. Image denoising: The deep learning revolution and beyond—a survey paper. *arXiv preprint arXiv:2301.03362*, 2023. [1, 2](#)
- [14] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017. [6, 7](#)
- [15] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020. [1, 4, 13](#)
- [16] Tao Huang, Songjiang Li, Xu Jia, Huchuan Lu, and Jianzhuang Liu. Neighbor2neighbor: Self-supervised denoising from single noisy images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14781–14790, 2021. [2](#)

- [17] Bahjat Kawar, Michael Elad, Stefano Ermon, and Jiaming Song. Denoising diffusion restoration models. *arXiv preprint arXiv:2201.11793*, 2022. 1, 3
- [18] Bahjat Kawar, Gregory Vaksman, and Michael Elad. Stochastic image denoising by sampling from the posterior distribution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1866–1875, 2021. 1, 3
- [19] Jaakko Lehtinen, Jacob Munkberg, Jon Hasselgren, Samuli Laine, Tero Karras, Miika Aittala, and Timo Aila. Noise2noise: Learning image restoration without clean data. In *International Conference on Machine Learning*, pages 2965–2974. PMLR, 2018. 2
- [20] Ben Mildenhall, Jonathan T Barron, Jiawen Chen, Dillon Sharlet, Ren Ng, and Robert Carroll. Burst denoising with kernel prediction networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2502–2510, 2018. 2, 3, 4, 6, 7
- [21] Kristina Monakhova, Stephan R Richter, Laura Waller, and Vladlen Koltun. Dancing under the stars: video denoising in starlight. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16241–16251, 2022. 3
- [22] Nick Moran, Dan Schmidt, Yu Zhong, and Patrick Coady. Noisier2noise: Learning to denoise from unpaired noisy data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12064–12072, 2020. 2
- [23] Junichi Nakamura. *Image sensors and signal processing for digital still cameras*. CRC press, 2017. 2
- [24] Stanley Osher, Martin Burger, Donald Goldfarb, Jinjun Xu, and Wotao Yin. An iterative regularization method for total variation-based image restoration. *Multiscale Modeling & Simulation*, 4(2):460–489, 2005. 1, 2
- [25] Tongyao Pang, Huan Zheng, Yuhui Quan, and Hui Ji. Recorrupted-to-recorrupted: unsupervised deep learning for image denoising. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2043–2052, 2021. 2
- [26] Naama Pearl, Tali Treibitz, and Simon Korman. Nan: Noise-aware nerfs for burst-denoising. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12672–12681, 2022. 3, 6
- [27] Tobias Plotz and Stefan Roth. Benchmarking denoising algorithms with real photographs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1586–1595, 2017. 2
- [28] Abhijith Punnapurath, Abdullah Abuolaim, Abdelrahman Abdelhamed, Alex Levinstein, and Michael S Brown. Day-to-night image synthesis for training nighttime neural ips. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10769–10778, 2022. 3
- [29] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022. 3
- [30] Chitwan Saharia, William Chan, Huiwen Chang, Chris Lee, Jonathan Ho, Tim Salimans, David Fleet, and Mohammad Norouzi. Palette: Image-to-image diffusion models. In *ACM SIGGRAPH 2022 Conference Proceedings*, pages 1–10, 2022. 1, 3, 6, 7
- [31] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S Sara Mahdavi, Rapha Gontijo Lopes, et al. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv preprint arXiv:2205.11487*, 2022. 3, 7
- [32] Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J Fleet, and Mohammad Norouzi. Image super-resolution via iterative refinement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 1, 3, 6, 7
- [33] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. 4
- [34] Linh Duy Tran, Son Minh Nguyen, and Masayuki Arai. Gan-based noise model for denoising real images. In *Proceedings of the Asian Conference on Computer Vision*, 2020. 2
- [35] Kaixuan Wei, Ying Fu, Jiaolong Yang, and Hua Huang. A physics-based noise formation model for extreme low-light raw denoising. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2758–2767, 2020. 3
- [36] Yutong Xie, Minne Yuan, Bin Dong, and Quanzheng Li. Diffusion model for generative image denoising. *arXiv preprint arXiv:2302.02398*, 2023. 3
- [37] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Multi-stage progressive image restoration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14821–14831, 2021. 2
- [38] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE transactions on image processing*, 26(7):3142–3155, 2017. 1, 2
- [39] Kai Zhang, Wangmeng Zuo, and Lei Zhang. Ffdnet: Toward a fast and flexible solution for CNN based image denoising. *IEEE Transactions on Image Processing*, 2018. 2
- [40] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 6, 7
- [41] Yi Zhang, Dasong Li, Ka Lung Law, Xiaogang Wang, Hongwei Qin, and Hongsheng Li. Idr: Self-supervised image denoising via iterative data refinement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2098–2107, 2022. 2

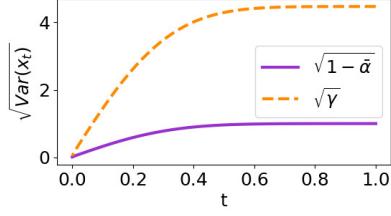


Figure 7: Schedules of standard deviation of added noise in the diffusion process (2), (3).

A. Proofs and derivations

A.1. Diffusion schedule

Below we show the derivation of the diffusion schedule $\eta_t = \lambda \beta_t \prod_{i=1}^{t-1} (1 - \beta_i)$ (Eq. (4) in the main paper), that is used in our diffusion noise model (Eq. (3) in the main paper). We require that

$$\gamma_t = \lambda (1 - \bar{\alpha}_t). \quad (13)$$

If follows from the definition of γ_t and $\bar{\alpha}_t$ that

$$\gamma_t = \sum_{i=1}^t \eta_i = \lambda \left(1 - \prod_{i=1}^t (1 - \beta_i) \right). \quad (14)$$

This implies for $t = 1, 2$:

$$\begin{aligned} \eta_1 &= \lambda (1 - (1 - \beta_1)) = \lambda \beta_1, \\ \eta_2 &= \lambda (1 - (1 - \beta_1)(1 - \beta_2)) - \eta_1 = \lambda \beta_2 (1 - \beta_1). \end{aligned} \quad (15)$$

For $t > 2$ we can derive the formula for η_t by observing that $\eta_t = \gamma_t - \gamma_{t-1}$, thus

$$\begin{aligned} \eta_t &= \gamma_t - \gamma_{t-1} = \lambda \left(1 - \prod_{i=1}^t (1 - \beta_i) \right) - \lambda \left(1 - \prod_{i=1}^{t-1} (1 - \beta_i) \right) \\ &= \lambda \left(\prod_{i=1}^{t-1} (1 - \beta_i) - \prod_{i=1}^t (1 - \beta_i) \right) = \lambda \cdot \left(\prod_{i=1}^{t-1} (1 - \beta_i) \right) \cdot (1 - (1 - \beta_t)) \\ &= \lambda \beta_t \prod_{i=1}^{t-1} (1 - \beta_i). \end{aligned} \quad (16)$$

The diffusion noise schedules for both the standard diffusion and the non-stationary diffusion are depicted in Fig. 7.

A.2. Reverse process

In this section we show the derivation of the reverse process as appears in Eq. (3) in Section 3.2 of the main paper. For small enough schedule steps η_t and given \mathbf{x}_0 , the reverse process probability is a Gaussian of the form:

$$q(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_{t-1}; \tilde{\boldsymbol{\mu}}(\mathbf{x}_t, \mathbf{x}_0), \tilde{\eta}_t \mathbf{I}). \quad (17)$$

Using Bayes' rule,

$$\begin{aligned}
q(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{x}_0) &= q(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{x}_0) \frac{q(\mathbf{x}_{t-1} | \mathbf{x}_0)}{q(\mathbf{x}_t | \mathbf{x}_0)} \\
&\stackrel{(a)}{\propto} \exp \left(-\frac{1}{2} \left(\frac{(\mathbf{x}_t - \mathbf{x}_{t-1})^2}{\eta_t} + \frac{(\mathbf{x}_{t-1} - \mathbf{x}_0)^2}{\gamma_{t-1}} - \frac{(\mathbf{x}_t - \mathbf{x}_0)^2}{\gamma_t} \right) \right) \\
&= \exp \left(-\frac{1}{2} \left(\frac{\mathbf{x}_t^2 - 2\mathbf{x}_t \mathbf{x}_{t-1} + \mathbf{x}_{t-1}^2}{\eta_t} + \frac{\mathbf{x}_{t-1}^2 - 2\mathbf{x}_0 \mathbf{x}_{t-1} + \mathbf{x}_0^2}{\gamma_{t-1}} - \frac{\mathbf{x}_t^2 - 2\mathbf{x}_0 \mathbf{x}_t + \mathbf{x}_0^2}{\gamma_t} \right) \right) \\
&\stackrel{(b)}{=} \exp \left(-\frac{1}{2} \left(\left(\frac{1}{\eta_t} + \frac{1}{\gamma_{t-1}} \right) \mathbf{x}_{t-1}^2 - 2 \left(\frac{\mathbf{x}_t}{\eta_t} + \frac{\mathbf{x}_0}{\gamma_{t-1}} \right) \mathbf{x}_{t-1} + \mathbf{f}(\mathbf{x}_t, \mathbf{x}_0) \right) \right) \\
&\stackrel{(c)}{\triangleq} \exp \left(-\frac{1}{2} \left(\frac{(\mathbf{x}_{t-1} - \tilde{\mu}_t)^2}{\tilde{\eta}_t} \right) \right), \tag{18}
\end{aligned}$$

where in (a) we use the forward definition in Eq. (3) in the main paper, in (b) we rearrange the expression as a polynomial of \mathbf{x}_{t-1} and define the free coefficient $\mathbf{f}(\mathbf{x}_t, \mathbf{x}_0) := \left(\frac{\mathbf{x}_t}{\eta_t} + \frac{\mathbf{x}_0}{\gamma_{t-1}} \right)^2 / \left(\frac{1}{\eta_t} + \frac{1}{\gamma_{t-1}} \right)$, and in (c) we denote,

$$\begin{aligned}
\tilde{\eta}_t &= \frac{1}{\left(\frac{1}{\eta_t} + \frac{1}{\gamma_{t-1}} \right)} = \frac{\gamma_{t-1} \eta_t}{\gamma_{t-1} + \eta_t} = \frac{\gamma_{t-1} \eta_t}{\gamma_t}, \\
\tilde{\mu}_t(\mathbf{x}_t, \mathbf{x}_0) &= \frac{\left(\frac{\mathbf{x}_t}{\eta_t} + \frac{\mathbf{x}_0}{\gamma_{t-1}} \right)}{\left(\frac{1}{\eta_t} + \frac{1}{\gamma_{t-1}} \right)} = \frac{\gamma_{t-1} \mathbf{x}_t + \eta_t \mathbf{x}_0}{\gamma_{t-1} + \eta_t} = \frac{\gamma_{t-1} \mathbf{x}_t + \eta_t \mathbf{x}_0}{\gamma_t} = \frac{\gamma_{t-1}}{\gamma_t} \mathbf{x}_t + \frac{\eta_t}{\gamma_t} \mathbf{x}_0. \tag{19}
\end{aligned}$$

The derivation of the loss function can be done by following the same approach as in [15], with the only difference being our different μ_t and η_t .

A.3. Noise correlation

In the following section we prove the noise correlation relationship described in Eq. (11) in Section 3.5 of the main paper. For clarity we summarize the background for this phenomenon here: We consider a noisy input image generated according to the noise model in Eq. (1), and calculate the induced time map \mathbf{T}^* . When initializing the reverse process with this noisy input, after k diffusion steps the time map is given by $\mathbf{t}_k = (\mathbf{T}^* - k)^+$. We wish to prove that the noise in $\mathbf{x}_{\mathbf{t}_k}$ can be written as a linear combination between the noise in the input image $\epsilon_{\mathbf{T}^*}$ and a new i.i.d. noise term $\epsilon_{\mathbf{t}_k}$. I.e.,

$$\mathbf{x}_{\mathbf{t}_k} = \mathbf{x}_0 + \frac{\gamma_{\mathbf{t}_k}}{\sqrt{\gamma_{\mathbf{T}^*}}} \epsilon_{\mathbf{T}^*} + \sqrt{\gamma_{\mathbf{t}_k} \left(1 - \frac{\gamma_{\mathbf{t}_k}}{\gamma_{\mathbf{T}^*}} \right)} \epsilon_{\mathbf{t}_k}. \tag{20}$$

We show this by induction. For $k = 0$, we ought to prove

$$\mathbf{x}_{\mathbf{t}_0} = \mathbf{x}_0 + \frac{\gamma_{\mathbf{t}_0}}{\sqrt{\gamma_{\mathbf{T}^*}}} \epsilon_{\mathbf{T}^*} + \sqrt{\gamma_{\mathbf{t}_0} \left(1 - \frac{\gamma_{\mathbf{t}_0}}{\gamma_{\mathbf{T}^*}} \right)} \epsilon_{\mathbf{t}_0}. \tag{21}$$

Since $\mathbf{t}_0 = \mathbf{T}^*$, this reduces to showing

$$\mathbf{x}_{\mathbf{T}^*} = \mathbf{x}_0 + \frac{\gamma_{\mathbf{T}^*}}{\sqrt{\gamma_{\mathbf{T}^*}}} \epsilon_{\mathbf{T}^*} + \sqrt{\gamma_{\mathbf{T}^*} \left(1 - \frac{\gamma_{\mathbf{T}^*}}{\gamma_{\mathbf{T}^*}} \right)} \epsilon_{\mathbf{t}_0} = \mathbf{x}_0 + \sqrt{\gamma_{\mathbf{T}^*}} \epsilon_{\mathbf{T}^*} \tag{22}$$

which is true by definition.

Now, suppose that Eq. (20) holds for $\mathbf{t}_k = (\mathbf{T}^* - k)^+$ with a new i.i.d. noise term $\epsilon_{\mathbf{t}_k}$. The next reverse step is $(\mathbf{t}_k - 1)^+$. For simplicity we omit the clipping notation. By the reverse process equation (Eq. (5) in the main paper) we can express $\mathbf{x}_{\mathbf{t}_{k-1}}$ as a function of $\mathbf{x}_{\mathbf{t}_k}, \mathbf{x}_0$ and a new i.i.d. noise term $\bar{\epsilon}_{\mathbf{t}_{k-1}}$. We use an equivalent reformulation for equation Eq. (5), noting that

$$\tilde{\eta}_t = \frac{\gamma_{t-1} \eta_t}{\gamma_t} = \frac{\gamma_{t-1} (\gamma_t - \gamma_{t-1})}{\gamma_t} = \gamma_{t-1} \left(1 - \frac{\gamma_{t-1}}{\gamma_t} \right). \tag{23}$$

Hence, we have

$$\mathbf{x}_{t_k-1} = \frac{\gamma_{t_k-1}}{\gamma_{t_k}} \mathbf{x}_{t_k} + \frac{\eta_{t_k}}{\gamma_{t_k}} \mathbf{x}_0 + \sqrt{\gamma_{t_k-1} \left(1 - \frac{\gamma_{t_k-1}}{\gamma_{t_k}}\right)} \bar{\epsilon}_{t_k-1}. \quad (24)$$

Plugging Eq. (20) into Eq. (24),

$$\begin{aligned} \mathbf{x}_{t_k-1} &= \frac{\gamma_{t_k-1}}{\gamma_{t_k}} \left[\mathbf{x}_0 + \frac{\gamma_{t_k}}{\sqrt{\gamma_{\mathbf{T}^*}}} \epsilon_{\mathbf{T}^*} + \sqrt{\gamma_{t_k} \left(1 - \frac{\gamma_{t_k}}{\gamma_{\mathbf{T}^*}}\right)} \epsilon_{t_k} \right] + \frac{\eta_{t_k}}{\gamma_{t_k}} \mathbf{x}_0 + \sqrt{\gamma_{t_k-1} \left(1 - \frac{\gamma_{t_k-1}}{\gamma_{t_k}}\right)} \bar{\epsilon}_{t_k-1} \\ &\stackrel{(a)}{=} \mathbf{x}_0 + \underbrace{\frac{\gamma_{t_k-1}}{\sqrt{\gamma_{\mathbf{T}^*}}} \epsilon_{\mathbf{T}^*}}_{\text{"Source noise"}} + \underbrace{\frac{\gamma_{t_k-1}}{\gamma_{t_k}} \sqrt{\gamma_{t_k} \left(1 - \frac{\gamma_{t_k}}{\gamma_{\mathbf{T}^*}}\right)} \epsilon_{t_k}}_{\text{"Uncorrelated noise"}} + \sqrt{\gamma_{t_k-1} \left(1 - \frac{\gamma_{t_k-1}}{\gamma_{t_k}}\right)} \bar{\epsilon}_{t_k-1} \\ &\stackrel{(b)}{=} \mathbf{x}_0 + \underbrace{\frac{\gamma_{t_k-1}}{\sqrt{\gamma_{\mathbf{T}^*}}} \epsilon_{\mathbf{T}^*}}_{\text{"Source noise"}}, \underbrace{\sqrt{\left(\frac{\gamma_{t_k-1}}{\gamma_{t_k}}\right)^2 \gamma_{t_k} \left(1 - \frac{\gamma_{t_k}}{\gamma_{\mathbf{T}^*}}\right) + \gamma_{t_k-1} \left(1 - \frac{\gamma_{t_k-1}}{\gamma_{t_k}}\right)}} \epsilon_{t_k-1}}_{\text{"Uncorrelated noise"}}, \bar{\epsilon}_{t_k-1} \\ &= \mathbf{x}_0 + \underbrace{\frac{\gamma_{t_k-1}}{\sqrt{\gamma_{\mathbf{T}^*}}} \epsilon_{\mathbf{T}^*}}_{\text{"Source noise"}}, \underbrace{\sqrt{\gamma_{t_k-1} \left[\left(\frac{\gamma_{t_k-1}}{\gamma_{t_k}} - \frac{\gamma_{t_k-1}}{\gamma_{\mathbf{T}^*}}\right) + \left(1 - \frac{\gamma_{t_k-1}}{\gamma_{t_k}}\right)\right]} \epsilon_{t_k-1}}_{\text{"Uncorrelated noise"}}, \bar{\epsilon}_{t_k-1} \\ &= \mathbf{x}_0 + \underbrace{\frac{\gamma_{t_k-1}}{\sqrt{\gamma_{\mathbf{T}^*}}} \epsilon_{\mathbf{T}^*}}_{\text{"Source noise"}}, \underbrace{\sqrt{\gamma_{t_k-1} \left(1 - \frac{\gamma_{t_k-1}}{\gamma_{\mathbf{T}^*}}\right)} \epsilon_{t_k-1}}_{\text{"Uncorrelated noise"}}, \bar{\epsilon}_{t_k-1} \end{aligned} \quad (25)$$

where in (a) we rearrange the noise term to separate between the noise of the input image and the noise terms that are uncorrelated to it and in (b) we use the property of summation of independent Gaussian variables and introduced a new i.i.d. noise term ϵ_{t_k-1} .

Finally, we wish to express \mathbf{x}_{t_k-1} with a single noise term, which will be used in the loss function. This is done again with summation of two independent variables,

$$\begin{aligned} \mathbf{x}_{t_k-1} &= \mathbf{x}_0 + \underbrace{\frac{\gamma_{t_k-1}}{\sqrt{\gamma_{\mathbf{T}^*}}} \epsilon_{\mathbf{T}^*}}_{\text{"Source noise"}}, \underbrace{\sqrt{\gamma_{t_k-1} \left(1 - \frac{\gamma_{t_k-1}}{\gamma_{\mathbf{T}^*}}\right)} \epsilon_{t_k-1}}_{\text{"Uncorrelated noise"}}, \bar{\epsilon}_{t_k-1} \\ &= \mathbf{x}_0 + \sqrt{\frac{\gamma_{t_k-1}^2}{\gamma_{\mathbf{T}^*}} + \gamma_{t_k-1} \left(1 - \frac{\gamma_{t_k-1}}{\gamma_{\mathbf{T}^*}}\right)} \tilde{\epsilon}_{t_k-1} \\ &= \mathbf{x}_0 + \sqrt{\gamma_{t_k-1}} \tilde{\epsilon}_{t_k-1}, \end{aligned} \quad (26)$$

and the expression for $\tilde{\epsilon}_{t_k-1}$ is given by division of the noise terms by $\sqrt{\gamma_{t_k-1}}$,

$$\tilde{\epsilon}_{t_k-1} = \underbrace{\sqrt{\frac{\gamma_{t_k-1}}{\gamma_{\mathbf{T}^*}}} \epsilon_{\mathbf{T}^*}}_{\text{"Source noise"}}, \underbrace{\sqrt{1 - \frac{\gamma_{t_k-1}}{\gamma_{\mathbf{T}^*}}} \epsilon_{t_k-1}}_{\text{"Uncorrelated noise"}}, \quad (27)$$

B. Results

We show a comparison of our method with the baseline diffusion model and a state-of-the-art denoising network [9], on images from ImageNet deteriorated by our noise model (1), under various noise levels. We show results from three noise levels, corresponding to camera gain levels of 1, 4, and 20 (recall that the results from gain level 16 were presented in Fig. 5 and Fig. 6 in the main paper).

In the lowest noise level (Fig. 8), the noise is mild and all models give comparable results. However, in darker areas (like the owl feathers), one can still identify some over-smoothing in the result of HiNet, and some remaining noise in the baseline result. In the middle noise level (Fig. 9), these artifacts are visible across all images, in both darker and brighter areas. We see that our model manages to balance between the generation of intricate details and the elimination of the noise. These phenomena are most evident in the highest noise level, depicted in Fig. 10.

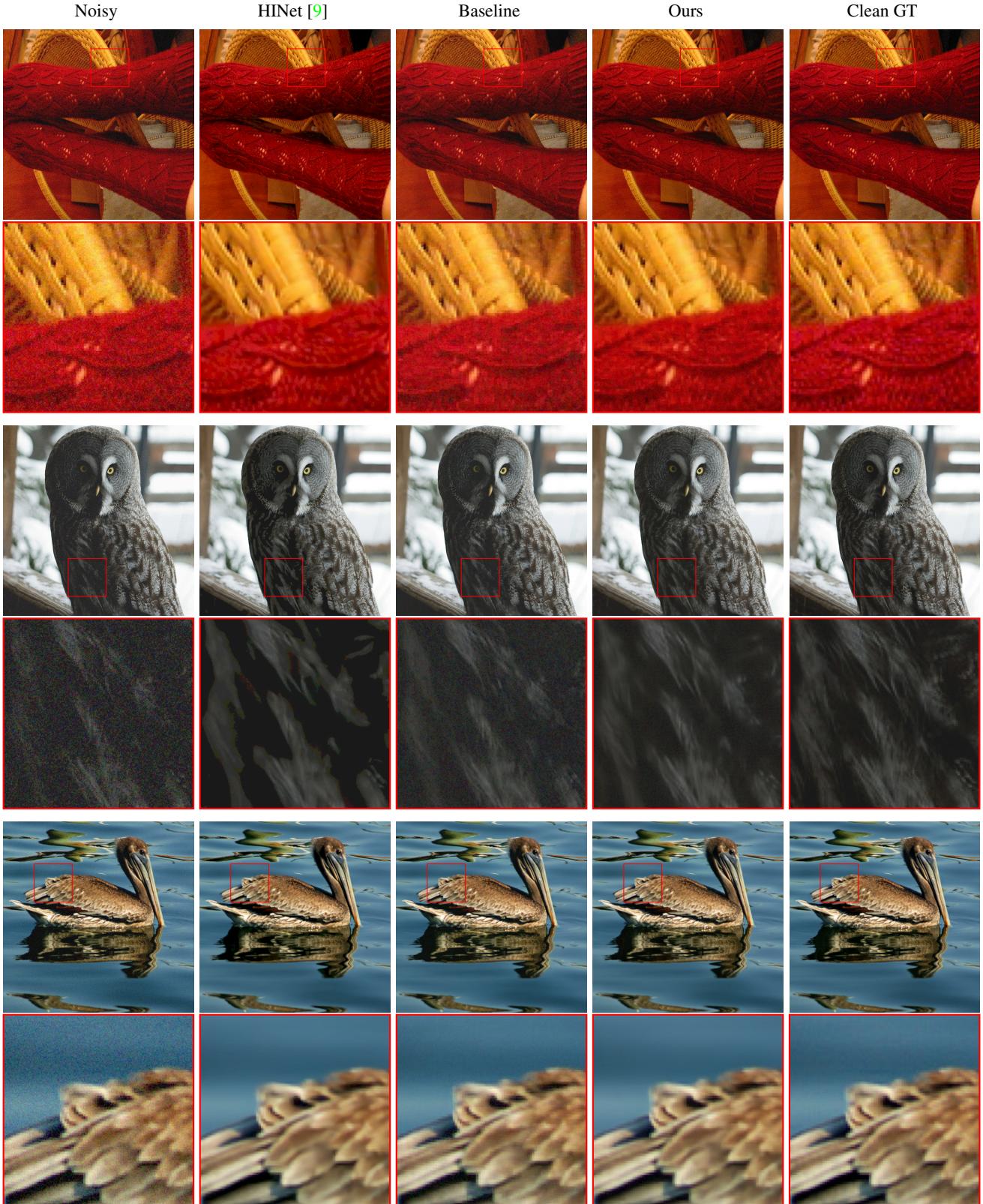


Figure 8: Comparison between different denoising methods on images with noise gain of 1. Some images are brightened for visualization.

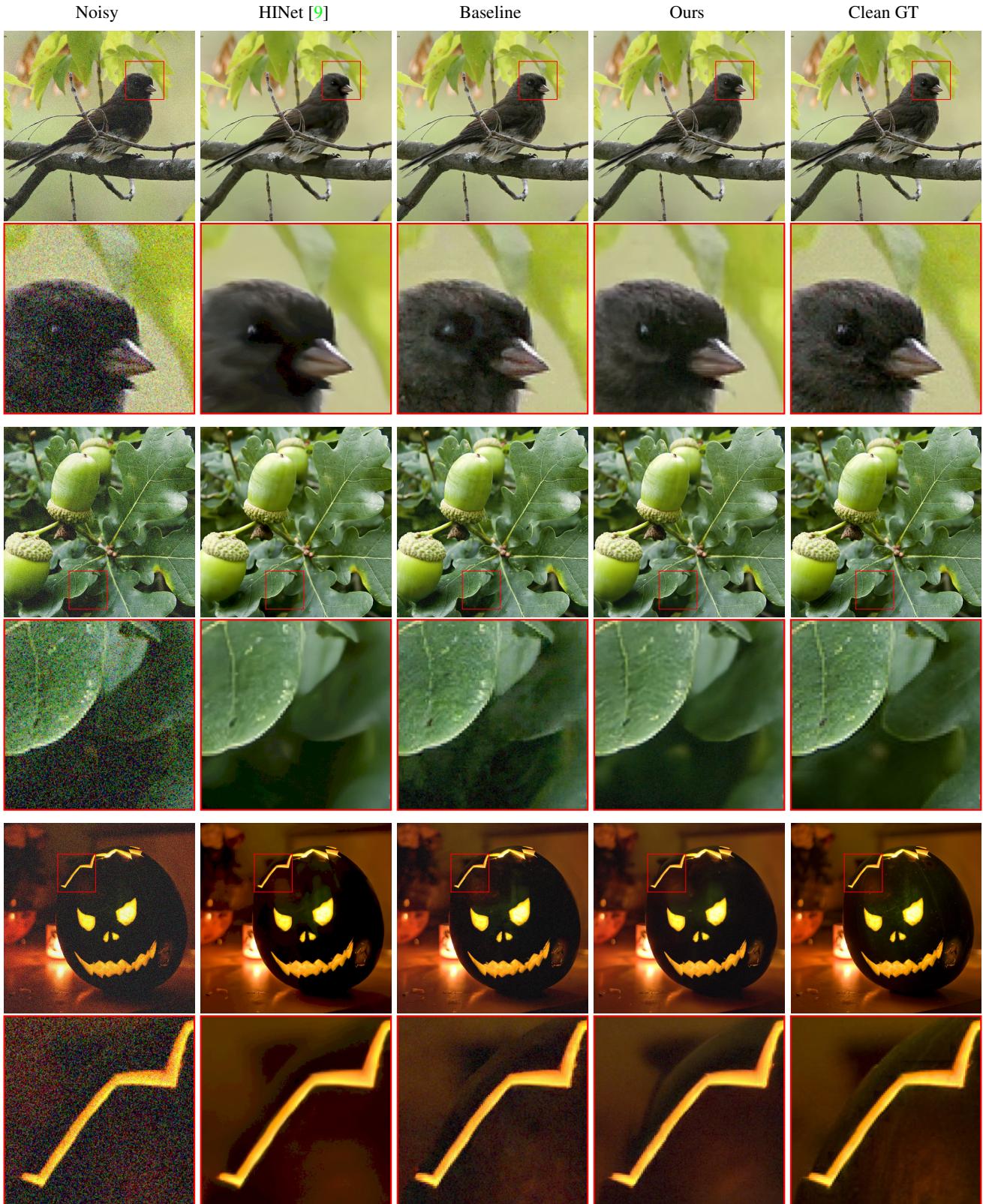


Figure 9: Comparison between different denoising methods on images with noise gain of 4.

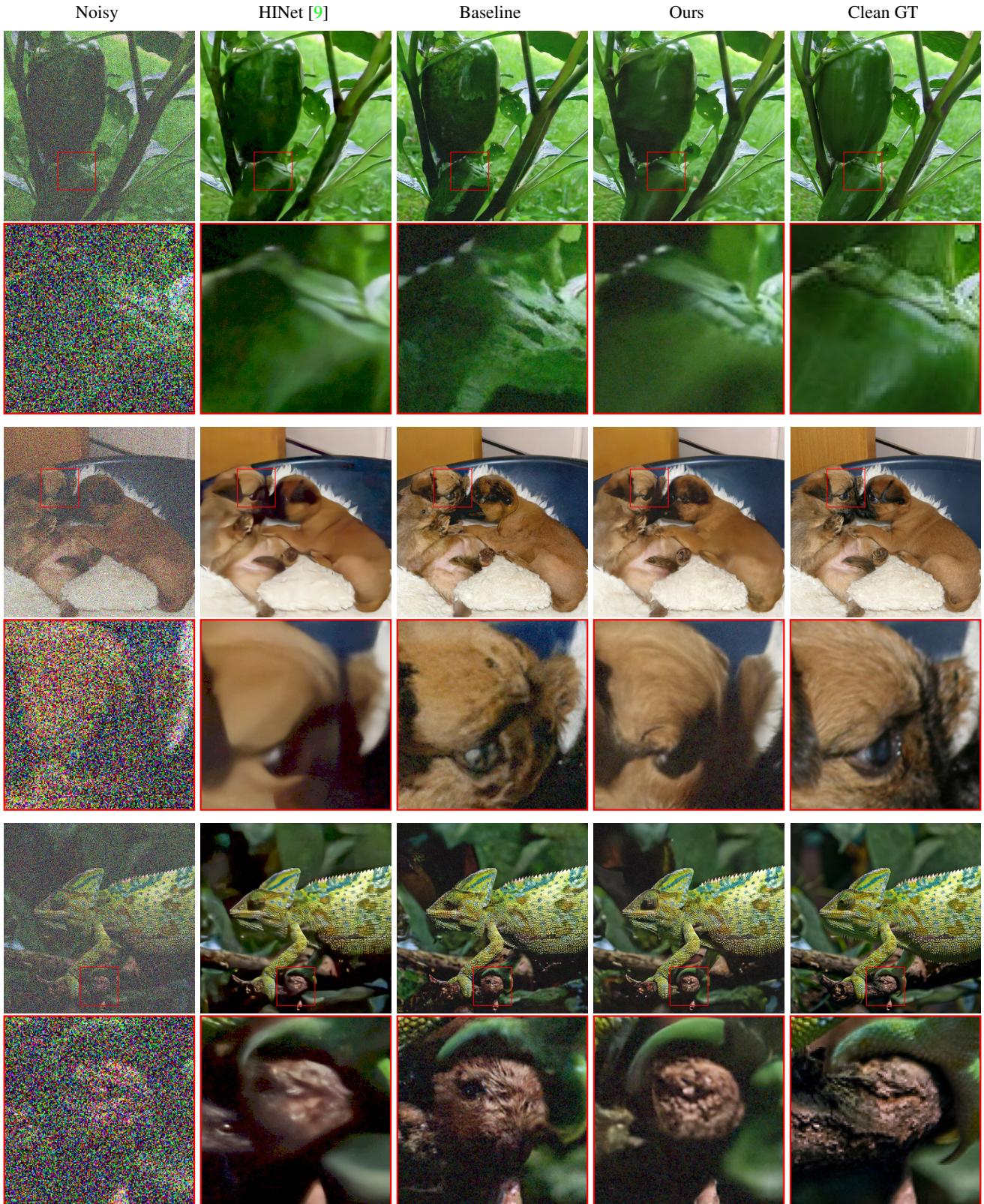


Figure 10: Comparison between different denoising methods on images with noise gain of 20.