DriftRec: Adapting diffusion models to blind image restoration tasks

Simon Welker\textsuperscript{1,2,\textsuperscript{*}}, Henry N. Chapman\textsuperscript{2}, Timo Gerkmann\textsuperscript{1}

\textsuperscript{1}Signal Processing (SP), Universität Hamburg, Germany
\textsuperscript{2}Center for Free-Electron Laser Science, Deutsches Elektronen-Synchrotron DESY, Hamburg

simon.welker@uni-hamburg.de, henry.chapman@cfel.de, timo.gerkmann@uni-hamburg.de

Abstract

In this work, we utilize the high-fidelity generation abilities of diffusion models to solve blind image restoration tasks, using JPEG artifact removal at high compression levels as an example. We propose an elegant modification of the forward stochastic differential equation (SDE) of diffusion models to adapt them to restoration tasks and name our method DriftRec. Comparing DriftRec against an $L_2$ regression baseline with the same network architecture and a state-of-the-art technique for JPEG reconstruction, we show that our approach can escape both baselines' tendency to generate blurry images, and recovers the distribution of clean images significantly more faithfully while only requiring a dataset of clean/corrupted image pairs and no knowledge about the corruption operation. By utilizing the idea that the distributions of clean and corrupted images are much closer to each other than to a Gaussian prior, our approach requires only low levels of added noise, and thus needs comparatively few sampling steps even without further optimizations.

1. Introduction

Diffusion models have taken the world of machine learning by storm due to their unprecedented ability to generate high-fidelity images \cite{sohn2022improved, song2021score, song2021scorebased} and audio \cite{dhariwal2021diffusion, ho2022denoising, ramesh2022text2music}, relatively easy training, and great flexibility to effectively condition on a variety of user inputs \cite{song2021scorebased, song2021score, song2021denoising}. Previous works on diffusion models have largely concentrated on unconditional and conditional image generation. Recently, Bansal et al.\cite{bansal2022diffusion} and Daras et al.\cite{daras2022diffusion} have proposed to extend and improve the forward and reverse processes of diffusion models by employing known deterministic corruptions. However, they have only evaluated their ideas for unconditional generation tasks, rather than for faithfully inverting corruptions to restore plausible original images. Here, we go a different route by explicitly modifying and training diffusion models to restore plausible images from corrupted ones, called DriftRec.

In contrast to other works \cite{bansal2022diffusion, daras2022diffusion, song2021scorebased}, which make at least one of the following assumptions, DriftRec does not require the underlying corruption operator to be known, linear, nor from a continuous family of operators. Instead, it requires only a dataset of (clean image, corrupted image) pairs, as in classic supervised training. DriftRec utilizes an alternative way of combining a deterministic corruption with the usual Gaussian noise used in diffusion models, within the formalism of SDEs introduced by Song et al.\cite{song2020generative}. In this work, we consider JPEG compression with low and unknown JPEG Quality Factors (QF) from 0–20\% as an example corruption with a nonlinear and nondifferentiable corruption operator that we treat as being fully unknown to the restoration procedure at inference time. To the best of our knowledge, we present results for an input QF of 0 for the first time in the literature.

We propose a modification of the forward SDE of diffusion models to adapt them to such restoration tasks, building upon previous work from the speech processing literature \cite{dhariwal2021diffusion, song2021scorebased}. We extend their modification to a more general form and apply these ideas in the field of computer vision, to the nonlinear inverse problem of JPEG artifact removal.

1.1. Related work

For image restoration based on pretrained unconditional diffusion models, the formalism of Denoising Diffusion Restoration Models (DDRs) \cite{song2021scorebased} has been proposed. DDRs originally were only designed for linear inverse problems, but have very recently also been adapted to JPEG artifact removal by carefully considering JPEG encoding and decoding operators \cite{song2021scorebased}. The essential advantage of these methods is the ability to train only one unconditional diffusion model and use it for a variety of restoration tasks. This generality is advantageous from the standpoint of computational efficiency, however, such methods may not be optimally adapted to the task at hand in terms of reconstruction quality. Within this tradeoff between generality
and quality, we aim for the latter in this work.

In [28], the authors devise a diffusion model for image-to-image translation called Palette, and also briefly evaluate their approach for blind JPEG artifact removal with QFs as low as 5. In contrast to our work (see Sec. 2.2), Palette does not adapt the stochastic processes to the task, but simply uses the corrupted images as conditioning information to generate reconstructions, letting their sampling process start from an unstructured prior. Consequently, their approach uses 10 times as many DNN evaluations as our method, which likely impacts runtime performance by a similar factor. Palette also uses around 8.5 times as many DNN parameters (552 M) as our models.

For JPEG artifact removal specifically, recent methods often utilize knowledge about the corruption operation, such as Quality Factor estimates [12] or the quantization matrices stored in each image [9, 17]. Here, we intentionally avoid doing so to keep the model as general and applicable to corruptions that are not known or easily modeled, and to determine the achievable performance in such a fully blind setting.

2. Methods

2.1. SDE-based diffusion models

Following [31], the forward process of a diffusion model can be interpreted as a dynamical system following an SDE

\[ \text{d}x_t = f(x_t, t)\text{d}t + g(x_t, t)\text{d}w \]  

where \( x \in \mathbb{R}^{C \times H \times W} \) is the current image and \( w \) is a standard Wiener process of the same dimensionality as \( x \), and the process runs forward from \( t = t_e \) until the terminal process time \( T := 1 \), with \( t_e \gg 0 \) for numerical stability reasons [31]. Each image in the training dataset then represents the initial value \( x_0 \) of a particular realization of this SDE. Song et al. [31] showed that previous diffusion models in the discrete-time domain can be interpreted to follow either the so-called Variance Exploding (VE) SDE or the so-called Variance Preserving (VP) SDE. Both SDEs have the aim of progressively turning images into Gaussian white noise, thereby turning the intractable image distribution into a tractable prior. To generate images, one then samples an initial value from this prior and numerically solves the corresponding reverse SDE [1],

\[ \text{d}x_t = (-f(x_t, t) + g(t)\nabla_{x_t} \log p_t(x_t, t))\text{d}t + g(x_t, t)\text{d}\bar{w} \]  

where \( \bar{w} \) is a Wiener process running in reverse, and we assume an infinitesimal \( \text{d}t \) with positive sign. The only unknown term in (2) is the score \( \nabla_{x_t} \log p_t(x_t, t) \). A deep neural network called a score network \( S_\theta(x_t, t) \) is then trained to estimate this score, given the current process state \( x \) and time \( t \). By replacing the score with its learned approximation \( S_\theta \), we receive the so-called plug-in reverse SDE. To generate samples, this plug-in reverse SDE is then solved with numerical SDE solver schemes [18, 31, 32].

2.2. A family of task-adapted linear SDEs

Rather than turning the clean image distribution into pure noise, we adapt our forward process to turn the clean image
distribution into a noisy version of the corrupted image distribution. This has two purposes:

1. Instead of pure noise, this uses the corrupted image (plus tractable noise) as the initial value of the reverse SDE, thus achieving the task adaptation through the formulation of the process itself, as opposed to only providing the corrupted image as conditioning information.

2. Since the added Gaussian noise is white, it functions as a continual source of all possible spatial frequencies throughout the reverse process. The trained score model then filters these frequencies appropriately to generate plausible clean images from the target distribution without a loss of high-frequency detail.

Note that the distribution of noisy corrupted images is still tractable for the purposes of the restoration task, as a sample from it can be drawn by taking a corrupted image and adding Gaussian noise. Some other works also use similar initialization strategies, as a trick to speed up inference [17] or to generate images from rough sketches [22], but without adapting the SDE. While empirically this seems to work reasonably well, it is formally (and in principle, arbitrarily strongly) mismatched to the actual distributions encountered throughout the reverse process. Our SDE-based adaptation of the process largely avoids this problem, by ensuring that this initialization corresponds to the terminal distribution of the forward process, and thus the initial distribution of the reverse process.

We now propose the following family of linear forward SDEs to realize our idea:

\[ f(x_t, t) = \gamma F(t)(y - x_t), \quad g(t) = \nu \left( \frac{\sigma_{\text{max}}}{\sigma_{\text{min}}} \right)^{2t}, \quad (3) \]

where \( y \) is the corrupted image corresponding to \( x_0 \), \( \gamma \) is a stiffness hyperparameter controlling how strongly \( x_t \) is pulled towards \( y \), and \( F(t) \) controls the shape of the curve pulling \( x_t \) towards \( y \). \( \nu \) is a normalization factor determined to ensure that \( \sigma_T \approx \sigma_{\text{max}} \), where \( \sigma_t \) is the closed-form variance of the Gaussian process described by (3), see Appendix A in the supplementary material.

Intuitively, our family of SDEs combines the diffusion \( g \) of the VE SDE with an added drift term \( f \) that pulls \( x_t \) towards the corrupted image \( y \). It is interesting to note that the drift term \( f \) of the forward SDE (1) shows up with a negative sign in the reverse SDE (2). This term, \( -f \), continually pushes the process state \( x_t \) away from \( y \) during the reverse process. In Fig. 2, we illustrate this observation by showing both the forward and reverse process for one possible choice of SDE from our SDE family (3), see Sec. 2.3 for details. Without the score term, the state \( x_t \) of the reverse process would simply shoot off away from \( y \) and diverge. The score model can therefore be interpreted as a learned control strategy adapted to the problem and dataset, which must steer \( x_t \) towards a plausible sample \( x_0 \) in the presence of this repulsive action away from \( y \) due to \( -f \). A general connection between diffusion-based generative modeling and stochastic optimal control has very recently been discussed in the literature [3].

2.3. The two considered SDEs

In this work, we only consider the SDEs with \( F(t) = 1 \) and \( F(t) = t \) for simplicity – see Appendix A in the supplementary material for a more detailed discussion. The case of \( F(t) = 1 \) was previously proposed for similar tasks in the speech processing literature [37], and we refer to it as the Ornstein-Uhlenbeck Variance Exploding (OUVE) SDE. We also newly propose using \( F(t) = t \) here, which we call the \( t \)-squared Variance Exploding (TSDVE) SDE due to the decay towards \( y \) depending on \( t^2 \), see (4). To allow for efficient forward sampling in order to perform denoising score matching [31], we determine closed-form expressions of the mean \( \mu_t \) and variance \( \sigma_t \) of the Gaussian processes...
described by each SDE. Such closed forms are generally not available for most SDEs, but can be determined in our case since the drift terms we choose are affine functions of the state $x_t$, e.g., via \cite[eq. (5.50) and (5.53)]. The closed-form mean expressions of the two SDEs are:

$$\mu_{t}^{\text{OUVE}} = e^{-\gamma t} x_0 + (1 - e^{-\gamma t}) y,$$

(4)

$$\mu_{t}^{\text{TSDVE}} = e^{-\gamma t^2} x_0 + (1 - e^{-\gamma t^2}) y.$$

(5)

Both expressions describe a linear (pixel-wise) interpolation between $x_0$ and $y$, with the interpolation parameter controlled by an exponential (OUVE) or half-Gaussian-shaped (TSDVE) decay over time $t$. Fig. 2 illustrates the decay of the mean and simultaneous addition of noise for the OUVE SDE. We relegate the somewhat involved closed-form expressions for the variance to Appendix A in the supplementary material.

For both SDEs, $\mu_t \neq y$ for all finite $t$, which may seem like an issue since the aim was to have the process move towards $y$. However, letting $z \sim \mathcal{N}(0, I)$, it is only required that the distributions of $(\mu_t + \sigma_t z)$ and $(y + \sigma_t z)$ are close, so that the latter can function as a plausible initial value for the reverse sampling process. We can control how well the distributions of these two expressions match, either by increasing the stiffness $\gamma$ at the cost of potentially destabilizing the reverse process, or by increasing $\sigma_{\text{max}}$ to further smooth the density functions of both distributions at the cost of more reverse iterations.

2.4. Architecture and training procedure

We utilize the NCSN++ architecture \cite{NCSN++}, both for the $L_2$ regression baseline and as a score network. We train the $L_2$ regression baseline $R_0$ to recover $x_0$ given $y$ via an $L_2$ loss:

$$\theta^* = \arg \min_{\theta} \mathbb{E}_{t,(x_0,y)} \left[ \|R_0(y) - x_0\|^2_2 \right],$$

(6)

where we pass a constant dummy value of $t = 1$ to the time embedding layers \cite{NCSN++} of NCSN++ to avoid making any changes to the DNN that may affect the qualitative behavior of each layer. To train the score models $S_0$, we use the idea that diffusion models based on our linear SDEs can be trained by the same Denoising Score Matching \cite{DSM} target as in \cite{NCSN++}, but conditioned on $y$:

$$\theta^* = \arg \min_{\theta} \mathbb{E}_{t,(x_0,y),z,x_t} \left[ \|S_0(x_t, y, t) - \nabla_x \log p_{0t}(x_t|x_0,y)\|^2_2 \right],$$

(7)

where $p_{0t}(x_t|x_0,y) = \mathcal{N}(\mu_t, \sigma^2_I)$ is the so-called perturbation kernel, i.e., the distribution of the forward process at process time $t$. Here, $\mathcal{N}(\cdot, \cdot)$ denotes a Gaussian distribution, and $I$ is the identity matrix matching the total dimensionality of $x_0$. Due to $p_{0t}$ being a Gaussian distribution, a closed form of the score term can be derived as:

$$\nabla_x \log p_{0t}(x_t|x_0,y) = \frac{(x_t - \mu_t)}{\sigma^2},$$

(8)

which allows to write the training objective (7) as:

$$\theta^* = \arg \min_{\theta} \mathbb{E}_{t,(x_0,y),z,x_t} \left[ \|S_0(x_t, y, t) + z\|^2_2 \right],$$

(9)

simply by dropping the denominator. This has the nice effect that the target DNN output is distributed as a standard Gaussian for all process times $t$. The only changes in the objective (7) over the usual Denoising Score Matching objective are that the expectation is also calculated over $y$, and that $y$ is provided to the score network. We provide $y$ as an input to $S_0$ by concatenating $x_t$ and $y$ along the channel dimension, for which we can change the number of input channels of NCSN++ from 3 to 6, which increases the number of DNN parameters by only about 0.1% (see Tab. 1c).

3. Experiments

3.1. Dataset

We use the CelebA-HQ dataset \cite{CelebA}, resized to 256x256 and split into 24000 images for training, 1500 for validation and 4500 for testing. We set $x_0$ to be the clean images from the dataset. To generate $y$, we sample a JPEG Quality Factor uniformly at random from 0 to 30 for each image $x_0$ and training step, and then apply JPEG compression to $x_0$ using the Pillow Python library \cite{Pillow}. During evaluation, we set the JPEG quality values to a constant value across all compared images and model, but do not provide this quality value to any model. In this sense, the trained models must perform the reconstruction while being blind both to the amount and the internals of the used JPEG compression.

3.2. Training and process parameterization

For training, we use the AdamW optimizer \cite{AdamW}, and set the hyperparameters of the process and training as listed in Table 1. We chose $\gamma$ for the OUVE SDE via manual grid search in $\{1, 1.5, 2\}$, finding $\gamma = 1$ to generally perform best. We set $\gamma_{\text{TSDVE}} = 2\gamma$ for the TSDVE SDE to match the
4. Results and Discussion

For the following evaluations, we generate reconstructions from our DriftRec models with the Euler-Maruyama sampler [18] using $N = 100$ uniform discretization steps, and retrieve reconstructions from QGAC [9] and the $L_2$ baseline via a single forward pass.

4.1. Qualitative evaluation

In Figure 1, we show example reconstructions starting from JPEG QF 10, comparing QGAC and the $L_2$ baseline against DriftRec using the OUVE SDE. Qualitatively, both QGAC and the $L_2$ baseline reconstruct major features well, but fail to produce plausible high-frequency details. This is particularly visible for hair, facial hair and skin textures, and results in a painting-like look. In contrast, our approach reconstructs more natural-looking images and shows a remarkable ability to reconstruct natural skin and hair textures that are plausible given the corrupted image.

4.2. Quantitative evaluation

In Tab. 2, we quantitatively compare all methods for the quality factors $QF \in \{0, 5, 10, 20\}$, using the distribution-based metrics Fréchet Inception Distance (FID) [10] and Kernel Inception Distance (KID) [4], the average Structural Similarity (SSIM) [35] and the average of the Learned Perceptual Image Patch Similarity (LPIPS) [39]. We use the official code by Zhang et al. [39] to calculate LPIPS, the scikit-image Python library [33] to calculate SSIM, and the piq Python library [15] to calculate KID and FID, using default settings throughout. We evaluate all metrics on our test set of 4500 images.

Judging from KID and FID, both of our DriftRec methods consistently model the clean image distribution of CelebA-HQ significantly more faithfully than the other methods. QGAC even achieves worse values in this regard than the compressed images, and for QF 10 and 20, the $L_2$ baseline does too. It is important to point out that KID and FID only compare the distributions of extracted features from all test images against the features from the ground-truth distribution, and thus only judge how plausible the reconstructions are overall. They cannot be used directly to make a statement about the faithfulness of each single reconstruction to its ground-truth reference image. However, the reference-based perceptual metric LPIPS also shows a consistent advantage of both DriftRec methods in terms of perceptual quality for all QF values, which matches our subjective visual inspection.

For SSIM, on the other hand, the $L_2$ baseline achieves significant improvements throughout, whereas DriftRec only marginally improves SSIM compared to the compressed images. QGAC also achieves strong SSIM values and, overall, performs rather similarly to the $L_2$ baseline for QF 10 and 20. The low SSIM improvements of DriftRec are to be somewhat expected due to the generative and probabilistic nature of the method. DriftRec always attempts to generate high-frequency detail, even for very low QF where such details must be generated from very little information. QGAC and the $L_2$ baseline, meanwhile, resort to reconstructing especially blurry images, which nonetheless achieve decent SSIM scores. As has been previously noted [6], generated high-frequency content is generally disincentivized by classic reference-based metrics such as SSIM. SSIM has also been shown to not correlate well with human perceptual preference [6,20], with LPIPS scores being aligned better with human perception [23]. For subjective evaluation by the reader, we show further randomly chosen example images from all methods in the supplemen-

| SDE   | $\gamma$ | $\sigma_{\text{max}}$ | $\sigma_{\text{min}}$ | $t_e$ | $N$  |
|-------|----------|------------------------|------------------------|-------|-----|
| OUVE  | 1        | 0.3                    | 0.01                   | 0.01  | 100 |
| TSDVE | 2        | 0.3                    | 0.01                   | 0.01  | 100 |

(a) Process parameterization

| Learning rate | $2 \times 10^{-4}$ |
|---------------|--------------------|
| Batch size per GPU | 6               |
| Number of GPUs | 2                    |
| Max. epochs | 100                  |

(b) Training hyperparameters

| DriftRec | 65.6 M |
|----------|--------|
| $L_2$ Baseline | 65.5 M |
| QGAC [9] | 259.4 M |

(c) Number of DNN parameters in all compared methods.

Table 1. Parameters for the SDEs (a) and network training (b), and model size in number of parameters (c).
ranted, with respect to the drift term $f$.

The exploration of possible SDEs for restoration tasks is warranted, with respect to the drift term $f$, the diffusion term $g$ and the process parameterization.

4.3. Very low quality factor inputs

For very low input JPEG quality factors of 5 and 0, we show example images in Fig. 3 and Fig. 4. One can observe that DriftRec still generates mostly plausible fine structures in these difficult scenarios, whereas QGAC fully fails and the $L_2$ baseline overblurs heavily. The metrics in Tabs. 2c and 2d corroborate these observations. DriftRec is, however, somewhat affected by artifacts: in particular, it incorporates heavily quantized color values from the corrupted images into the reconstruction – see, for instance, the hair in Fig. 4 (top row). Colors also become somewhat mismatched to the ground truth, which should however not be too surprising as most color information is lost due to heavy quantization.

While both SDEs perform very similarly overall, the TSDVE SDE exhibits slightly better metrics than the OUVE SDE in these low-QF scenarios, see Tabs. 2c and 2d, but differences are not large. Nonetheless, we argue that further exploration of possible SDEs for restoration tasks is warranted, with respect to the drift term $f$, the diffusion term $g$ and the process parameterization.

4.4. Limitations

Our proposed method DriftRec makes a few assumptions that we would like to point out. First, DriftRec implicitly assumes that the clean image $x_0$ and the corrupted image $y$ are reasonably similar each other, so that interpolating linearly between them through the diffusion process – see Eqs. (4) and (5) – is sensible. Violations of this assumption, e.g., when the measurement $y$ is a Fourier transform of the clean image $x_0$, would require reparameterization or revision of the method, for example based on the ideas of Soft Diffusion [7].

Second, we have only presented results for a single dataset in this work. We have done so to determine achievable performance with our approach in a highly specialized setting without requiring long trainings on very large image datasets. Our trained models are therefore unlikely to adapt to images of very different objects than human faces, and may inpaint wrong structures such as hair or eyes by assuming that the target image is a picture of a human face. Nonetheless, DriftRec generally exhibits good reconstructions of non-facial textures present in the CelebA-HQ images, see for example Fig. 1 or further example images in Appendix B in the supplementary material. Furthermore, we point out that the $L_2$ baseline we evaluated is also baseline overblurs heavily. The metrics in Tabs. 2c and 2d corroborate these observations. DriftRec is, however, somewhat affected by artifacts: in particular, it incorporates heavily quantized color values from the corrupted images into the reconstruction – see, for instance, the hair in Fig. 4 (top row). Colors also become somewhat mismatched to the ground truth, which should however not be too surprising as most color information is lost due to heavy quantization.

While both SDEs perform very similarly overall, the TSDVE SDE exhibits slightly better metrics than the OUVE SDE in these low-QF scenarios, see Tabs. 2c and 2d, but differences are not large. Nonetheless, we argue that further exploration of possible SDEs for restoration tasks is warranted, with respect to the drift term $f$, the diffusion term $g$ and the process parameterization.

4.4. Limitations

Our proposed method DriftRec makes a few assumptions that we would like to point out. First, DriftRec implicitly assumes that the clean image $x_0$ and the corrupted image $y$ are reasonably similar each other, so that interpolating linearly between them through the diffusion process – see

| KID ↓ | FID ↓ | LPIPS ↓ | SSIM ↑ |
|-------|-------|---------|--------|
| JPEG_{20} | 8.63 | 21.17 | 0.08 | 0.90 |
| QGAC | 31.24 | 39.42 | 0.09 | 0.93 |
| $L_2$ baseline | 27.34 | 35.71 | 0.08 | 0.94 |
| TSDVE | 0.59 | 12.99 | 0.05 | 0.89 |
| OUVE | 0.57 | 12.97 | 0.05 | 0.89 |

(a) JPEG Quality Factor QF = 20

| KID ↓ | FID ↓ | LPIPS ↓ | SSIM ↑ |
|-------|-------|---------|--------|
| JPEG_{10} | 22.53 | 36.26 | 0.20 | 0.82 |
| QGAC | 46.89 | 53.97 | 0.14 | 0.90 |
| $L_2$ baseline | 38.18 | 45.92 | 0.13 | 0.90 |
| TSDVE | 2.32 | 15.72 | 0.08 | 0.83 |
| OUVE | 2.37 | 15.69 | 0.08 | 0.83 |

(b) JPEG Quality Factor QF = 10

| KID ↓ | FID ↓ | LPIPS ↓ | SSIM ↑ |
|-------|-------|---------|--------|
| JPEG_{5} | 66.75 | 79.61 | 0.38 | 0.71 |
| QGAC | 102.3 | 103.0 | 0.28 | 0.74 |
| $L_2$ baseline | 48.44 | 55.73 | 0.19 | 0.85 |
| TSDVE | 6.15 | 20.68 | 0.14 | 0.75 |
| OUVE | 6.89 | 21.31 | 0.14 | 0.75 |

(c) JPEG Quality Factor QF = 5

| KID ↓ | FID ↓ | LPIPS ↓ | SSIM ↑ |
|-------|-------|---------|--------|
| JPEG_{0} | 245.8 | 239.9 | 0.53 | 0.59 |
| QGAC | 289.8 | 254.3 | 0.71 | 0.35 |
| $L_2$ baseline | 55.24 | 62.27 | 0.26 | 0.78 |
| TSDVE | 21.28 | 37.07 | 0.24 | 0.64 |
| OUVE | 22.32 | 37.97 | 0.25 | 0.63 |

(d) JPEG Quality Factor QF = 0

Table 2. Distribution-based metrics (KID [4], FID [10]), the learned perceptual metric LPIPS [39], and SSIM [35] for 4500 CelebA-HQ test images, comparing corrupted and reconstructed test images of all methods to the ground truth for JPEG Quality Factors 20 (a), 10 (b), 5 (c) and 0 (d). KID scores are multiplied by 1000 for readability. Lower is better for all metrics except SSIM. Best values are listed in bold, and values worse than those of the corrupted images in dark red.
higher levels of Gaussian noise added on top. However, we also use the noiseless corrupted image $y$ as an input channel for our trained DNN, which is therefore not equipped to handle noise. The simplest, but likely still effective, strategy to alleviate this issue is to perform data augmentation by adding various expected levels of noise to this input channel during training.

### 4.5. Runtime performance and model size

As our method is based on diffusion models, which use iterative methods for sampling, it also requires multiple calls to the same DNN to retrieve a single reconstruction. We have used $N = 100$ iterations in this work using the Euler-Maruyama sampler [18], which requires the same number of $N_{DNN} = 100$ DNN evaluations. On a NVIDIA GeForce RTX 2080 Ti consumer GPU, we found that our choice of sampler and DNN requires around 70 seconds to reconstruct a batch of 16 images, resulting in a total runtime of about 5.5 hours for the entire test set of 4500 images. On the same hardware and image set, the compared method QGAC [9] requires only 310 seconds in total, and the $L_2$ baseline takes 400 seconds, with both methods using a single DNN pass for each image.

However, since we derived our diffusion-based method within the continuous formalism of SDEs, the choice of sampling procedure is extremely flexible and does not require retraining of the DNN. Recent research on diffusion models has made significant progress on sampling speed [14, 29, 38]. These techniques can in principle be combined with our method, but may require some careful
re-derivation since works typically make assumptions about the SDE drift term $f$ to be zero (VE SDE) or a simple multiplicative scaling of $x_t$ (VP SDE), which does not hold for our SDE adaptation.

Finally, pointing to Tab. 1c, we note that DriftRec has virtually the same number of DNN parameters as the $L_2$ baseline due to using the same DNN architecture with very minor modifications, see Sec. 2.4. Furthermore, our improved results over QGAC [9] are achieved with only about 25% of the DNN parameters.

5. Conclusions

Based on work from the speech processing literature [25, 37], we propose DriftRec, a method based on an elegant change to SDE-based diffusion models that adapts them to image restoration tasks. In contrast to existing diffusion-based restoration methods [2, 7, 16, 17], our approach is applicable to corruption operators that are both nonlinear and unavailable in closed form during reconstruction, and only requires a dataset of paired images. To demonstrate this, we consider the example task of JPEG artifact removal with very low quality factors. Comparing against an $L_2$ regression baseline using the same DNN architecture and a recent state-of-the-art method, we find that DriftRec restores images of perceptually higher quality and models the ground-truth image distribution significantly more faithfully. While currently requiring 100 times as many DNN passes than the compared methods due to the iterative nature of sampling with diffusion models, we expect that recent progress on efficient high-quality sampling for diffusion models, e.g. [36], will allow this number to drastically decrease and make the approach competitive in terms of runtime. Our idea of adapting the diffusion process can in principle be combined with most other techniques for designing, improving or specializing diffusion models, and therefore constitutes another building block in the growing diffusion model toolbox.

6. Ethics Statement

The methods presented in this paper carry ethical implications when used in domains such as video surveillance and identification. Our methods are trained on image datasets which may contain implicit biases in terms of skin color, age, gender, sexuality, or other identifying features. Since our methods generate enhanced images that represent samples from the underlying distribution of the training dataset which is biased, the resulting enhanced images will also be biased. Unchecked application of these methods – especially to low-quality input images – will thus lead to these implicit biases being reinforced. Any plans of applying such methods in fields such as person identification, video surveillance or law enforcement must therefore be subject to scrutiny by ethical researchers and regulators.

References

[1] Brian DO Anderson. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326, 1982. 2
[2] Arpit Bansal, Eitan Borgnia, Hong-Min Chu, Jie S Li, Hamid Kazemi, Furong Huang, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Cold diffusion: Inverting arbitrary image transforms without noise. *arXiv preprint arXiv:2208.09392*, 2022. 1, 8
[3] Julius Berner, Lorenz Richter, and Karen Ullrich. An optimal control perspective on diffusion-based generative modeling. In *NeurIPS 2022 Workshop on Score-Based Methods*, 2022. 3
[4] Mikołaj Binkowski, Dougal J. Sutherland, Michael Arbel, and Arthur Gretton. Demystifying MMD GANs. In *Int. Conf. on Learning Representations (ICLR)*, 2018. 5, 6
[5] Alex Clark. Pillow (PIL fork) documentation, 2015. 4
[6] Ryan Dahl, Mohammad Norouzi, and Jonathan Shlens. Pixel recursive super resolution. In *ICCV*, pages 5439–5448, 2017. 5
[7] Giannis Daras, Mauricio Delbracio, Hossein Talebi, Alexandros G Dimakis, and Peyman Milanfar. Soft diffusion: Score matching for general corruptions. *arXiv preprint arXiv:2209.05442*, 2022. 1, 6, 8
[8] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat GANs on image synthesis. *Advances in Neural Inf. Proc. Systems (NeurIPS)*, 34, 2021. 1
[9] Max Ehrlich, Larry Davis, Ser-Nam Lim, and Abhinav Shrivastava. Quantization guided JPEG artifact correction. In *ECCV*, pages 293–309. Springer, 2020. 2, 5, 7, 8
[10] 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. In *Advances in Neural Inf. Proc. Systems (NeurIPS)*, volume 30, 2017. 5, 6
[11] Rongjie Huang, Max W. Y. Lam, Jun Wang, Dan Su, Dong Yu, Yi Ren, and Zhou Zhao. FastDiff: A fast conditional diffusion model for high-quality speech synthesis. In *IJCAI*, 2022. 1
[12] Jiaxi Jiang, Kai Zhang, and Radu Timofte. Towards flexible blind JPEG artifacts removal. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4997–5006, 2021. 2
[13] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *Int. Conf. on Learning Representations (ICLR)*, 2018. 4
[14] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *arXiv preprint arXiv:2206.00364*, 2022. 7
[15] Sergey Katsyrulin, Jamil Zakirov, Denis Prokopenko, and Dmitry V. Dylov. PyTorch image quality: Metrics for image quality assessment, 2022. 5
[16] Bahajt Kawar, Michael Elad, Stefano Ermon, and Jiaming Song. Denoising diffusion restoration models. In *ICLR Workshop on Deep Generative Models for Highly Structured Data*, 2022. 1, 8
[17] Bahjat Kawar, Jiaming Song, Stefano Ermon, and Michael Elad. JPEG artifact correction using denoising diffusion restoration models. arXiv preprint arXiv:2209.11888, 2022.
1, 2, 3, 8

[18] P.E. Kloeden and E. Platenn. Numerical Solution of Stochastic Differential Equations. Stochastic Modelling and Applied Probability. Springer Berlin Heidelberg, 2011. 2, 5, 7

[19] Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. DiffWave: A versatile diffusion model for audio synthesis. Int. Conf. on Learning Representations (ICLR), 2021.

[20] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR), pages 4681–4690, 2017.

[21] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In Int. Conf. on Learning Representations (ICLR), 2019.

[22] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. SDEdit: Guided image synthesis and editing with stochastic differential equations. In Int. Conf. on Learning Representations (ICLR), 2022.

[23] Juan Carlos Mier, Eddie Huang, Hossein Talebi, Feng Yang, and Peyman Milanfar. Deep perceptual image quality assessment for compression. In IEEE International Conference on Image Processing (ICIP), pages 1484–1488. IEEE, 2021.

[24] 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.

[25] Julius Richter, Guillaume Carbajal, and Timo Gerkmann. Speech enhancement with stochastic temporal convolutional networks. ISCA Interspeech, pages 4516–4520, 2020.

[26] Julius Richter, Simon Welker, Jean-Marie Lemercier, Junlong Lay, and Timo Gerkmann. Speech enhancement and dereverberation with diffusion-based generative models. arXiv preprint arXiv:2208.05830, 2022.

[27] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR), pages 10684–10695, June 2022.

[28] 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.

[29] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In Int. Conf. on Learning Representations (ICLR), 2022.

[30] Roy Sheffer and Yossi Adi. I hear your true colors: Image guided audio generation. arXiv preprint arXiv:2211.03089, 2022.

[31] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. Int. Conf. on Learning Representations (ICLR), 2021.

[32] Simo Särkkä and Arno Solin. Applied Stochastic Differential Equations. Cambridge University Press, 2019.

[33] Stefan Van der Walt, Johannes L Schönberger, Juan Nunez-Iglesias, François Boulogne, Joshua D Warner, Neil Yager, Emmanuelle Gouillart, and Tony Yu. scikit-image: image processing in Python. PeerJ, 2:e453, 2014.

[34] Pascal Vincent. A connection between score matching and denoising autoencoders. Neural Computation, 23(7):1661–1674, 2011.

[35] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. IEEE Trans. on Image Proc., 13(4):600–612, 2004.

[36] Daniel Watson, William Chan, Jonathan Ho, and Mohammad Norouzi. Learning fast samplers for diffusion models by differentiating through sample quality. In Int. Conf. on Learning Representations (ICLR), 2021.

[37] Simon Welker, Julius Richter, and Timo Gerkmann. Speech enhancement with score-based generative models in the complex STFT domain. ISCA Interspeech, 2022.

[38] Qingsheng Zhang and Yongxin Chen. Fast sampling of diffusion models with exponential integrator. arXiv preprint arXiv:2204.13902, 2022.

[39] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR), pages 586–595, 2018.
A. Closed-form variance solutions of the OUVE and TSDVE SDEs

In the following, we provide the expressions for the diffusion-normalization factors $\nu$ and the solved variance of the Gaussian process, for the OUVE SDE and the TSDVE SDE. We utilized the software Mathematica 12.1 to solve the mean and variance ODEs [32, eq. (5.50), (5.53)] for the initial value $\sigma_0 = 0$. This choice is in contrast to [31], where the initial value $\sigma_0 = \sigma_{\text{min}}$ was used. Our reasoning for this change is that due to the influence of our added drift terms, the choice by Song et al. may result in nonmonotonous functions $\sigma_t$, particularly depending on the choice of $\gamma$ and the relative scale of $\sigma_{\text{max}}$ in comparison to $\sigma_{\text{min}}$. We admit that the convenient name of $\sigma_{\text{min}}$ may be misleading under this assumption, since it is not a “minimum sigma” under our assumptions. Nonetheless, we keep this parameter as a way to control the shape of the variance curve, and also keep its name in line with the previous literature.

For the OUVE SDE, we follow [31] to determine an approximate normalization factor

$$\nu_{\text{OUVE}} = \sqrt{2 \left( \gamma + \log \left( \frac{\sigma_{\text{max}}}{\sigma_{\text{min}}} \right) \right)}, \quad (12)$$

resulting in the variance

$$\sigma^2_{t,\text{OUVE}} = \sigma^2_{\text{min}} \left( \frac{\sigma_{\text{max}}}{\sigma_{\text{min}}} \right)^{2t} - e^{-2\gamma t}. \quad (13)$$

One may observe that the condition $\sigma_t = \sigma_{\text{max}}$ is only approximately fulfilled here, but the error is small when $\sigma_{\text{min}}$ is small and $\gamma$ is reasonably large: for our parameter choice, it is equal to

$$0.01^2(-e^{-2\cdot1.1}) \approx -1.4 \times 10^{-5} \quad (14)$$

and we therefore use this factor $\nu_{\text{OUVE}}$ due to its relative simplicity.

For the TSDVE SDE, we first solved for the unnormalized variance expression using $\nu = 1$:

$$\sigma^2_{t,\text{TSDVE,unnorm}} = \frac{\sigma^2_{\text{min}}}{\sqrt{\gamma}} e^{-\gamma t^2} \left[ e^{\gamma t^2 + 2t \log (\sigma_{\text{min}}/\sigma_{\text{max}})} D \left( \frac{t \gamma + \log (\sigma_{\text{min}}/\sigma_{\text{max}})}{\sqrt{\gamma}} \right) \right. \left. - D \left( \log \left( \frac{\sigma_{\text{max}}}{\sigma_{\text{min}}} \right) \right) \right]$$

where $D$ is the Dawson function. Solving for $\nu$ such that $\nu^2 \sigma^2_T = \sigma_{\text{max}}$ exactly, we retrieve

$$\nu_{\text{TSDVE}} = \sqrt{\frac{\sigma^2_{\text{max}}}{\sqrt{\gamma} e^{-\gamma \eta}}}, \quad (15)$$

$$\eta = e^{\gamma} \sigma^2_{\text{max}} D \left( \frac{\gamma + \log \left( \frac{\sigma_{\text{max}}}{\sigma_{\text{min}}} \right)}{\sqrt{\gamma}} \right) - \sigma^2_{\text{min}} D \left( \log \left( \frac{\sigma_{\text{max}}}{\sigma_{\text{min}}} \right) \right) \quad (16)$$

We then multiply the diffusion term $g(t)$ of the forward SDE by this $\nu_{\text{TSDVE}}$, finally resulting in

$$g(t) \equiv \nu_{\text{TSDVE}} \left( \frac{\sigma_{\text{max}}}{\sigma_{\text{min}}} \right)^{2t} \quad (17)$$

$$\implies \sigma^2_{t,\text{TSDVE}} = \nu^2_{\text{TSDVE}} \cdot \sigma^2_{t,\text{TSDVE,unnorm}} \quad (18)$$

Note that we are now already dealing with unwieldy functions such as the Dawson function just by setting $F(t) = t$ (TSDVE SDE). This was the principal reason for us to not investigate $F(t) = t^\alpha$ for $\alpha \geq 2$ or even more complex expressions for $F(t)$, though exploring a solution or approximation of $\sigma_t$ for these cases may be helpful: Increasing $\alpha$ keeps the process mean close to $x_0$ for longer early in the forward process and pushes it away from the corrupted $y$ stronger early in the reverse process, which may help the reverse process to result in higher-quality reconstructions. The slightly improved performance of the TSDVE SDE for low-quality input images (JPEG level 0 and 5, see Sec. 4.3) supports this idea to some degree.

B. More example images

In Figs. 5 to 8 we show a random selection of example images for the four evaluated JPEG QFs of 20, 10, 5 and 0 (respectively). To make the qualitative behavior of each method and each QF directly comparable, we compare all methods and all QF for a single image in Fig. 9.
Figure 5. Example images for the restoration task with JPEG QF 20.

Figure 6. Further example images for the restoration task with JPEG QF 10.
Figure 7. Further example images for the restoration task with JPEG QF 5.

Figure 8. Further example images for the restoration task with JPEG QF 0.
Figure 9. A direct comparison of all methods for a single image using input QF ∈ {0, 5, 10, 20}, demonstrating the qualitative behavior of all methods as the QF decreases.