Project Report : Denoising Diffusion Restoration Models Generative Modeling Class

Gabriel Fiastre, Shane Hoeberichts, Charbel Abi Hana, Maxime Alvarez
{gabriel.fiastre, shane.hoeberichts}@dauphine.eu,
{charbel.abi_hana, maxime.alvarez}@ens-paris-saclay.fr

Abstract

Recently, diffusion models have reached impressive performance on unconditional generative modeling becoming the new state-of-the-art. Their application to inverse problems could help solve difficult problems in many areas of science and engineering, however inverse problem solver face a trade-off between generality and efficiency with the extreme case of supervised models being able to solve only the inverse problems they have been trained for. In this project we study the Denoising Diffusion Restoration Models which are designed to solve general linear inverse problems while staying efficient, by exploiting pretrained diffusion models and applying the diffusion in the spectral space. Although DDRMs do not require to be retrained to adapt to new inverse problem, they cannot handle nonlinear problems, require to know the measurement function and need an efficient way to compute the SVD of the degradation matrix. We explore several extensions of DDRM and perform experiments using the limited hardware we dispose.

1. Introduction

Inverse problems are commonplace in science where measurements almost always introduce some noise or degradation of the original data. A general model for inverse problems is the following: $y = \mathcal{A}(x) + z$, where y is the measurement, $\mathcal{A}$ is the measurement process and z is noise introduced by the measurement. The link between measurement and samples is one-to-many making the inverse problem formulation ill-posed and difficult to solve.

Denoising Diffusion Restoration Models is a class of models capable of solving general linear inverse problems of shape $y = Hx + z$, where H is the degradation matrix. DDRM uses the SVD of the degradation matrix H to do the diffusion in the spectral space leveraging pretrained unconditional diffusion models. For a specific linear inverse problem and given the degradation matrix H, DDRM allows to sample from the posterior distribution $p(x|y)$, thus recovering the original data.

Extensions of the DDRM framework target nonlinear noise [9], using a manifold constraint [3], and applications to audio [12]. More recent work [2] give up the need for the SVD of the degradation matrix and can handle non-Gaussian measurement noise, by using an approximation of $p(y|x_t)$.

In this project we look at extending the DDRM method through multiple ways. By exploring the addition of a manifold constraint as detailed in *Improving Diffusion Models for Inverse Problems using Manifold Constraints* [3], by exploring the case where the degradation matrix is unknown but constrained to the deblurring task, and by extending to video deblurring task.

2. Denoising Diffusion Restoration Models

The DDRM method aims to solve general linear inverse problems by using denoising diffusion models.

2.1. Inverse Problems

General linear inverse problems can be posed as:

$$y = Hx + z \tag{1}$$

where $y \in \mathbb{R}^m$ is the noisy measurement, $x \in \mathbb{R}^n$ is the signal we would like to recover, $H \in \mathbb{R}^{m \times n}$ is the known linear degradation matrix specific to the inverse problem, and $z \sim \mathcal{N}(0, \sigma_{\mathbf{y}}^2 \mathbf{I})$ is additive Gaussian noise with a known variance.

We can learn the data distribution of $\mathbf{x}$ with a generative model $p_\theta(\mathbf{x})$. Using this model and Bayes' theorem we get $p_\theta(\mathbf{x}|\mathbf{y}) \propto p_\theta(\mathbf{x})p(\mathbf{y}|\mathbf{x})$. We can then sample from the posterior $p_\theta(\mathbf{x}|\mathbf{y})$ to recover the signal $\mathbf{x}$ given the measurement $\mathbf{y}$. The prior $p_\theta(\mathbf{x})$ is not conditioned on $\mathbf{y}$ and can so be learned in an unsupervised fashion. However, sampling from the posterior can require many iterations to produce a good sample. Another solution then is to approximate the posterior directly using supervised learning. In this case, the model would have to be trained for each specific inverse

problem and would hardly generalize to inverse problems not seen during training.

The DDRM method tries to get the best of both worlds by finding an unsupervised solution that also fits supervised learning objectives.

2.2. Denoising Diffusion Models

In order to learn the prior $p_\theta(\mathbf{x})$, we use a denoising diffusion model because they have demonstrated exceptional performances for unconditional generative modeling [4, 8].

The diffusion model used in DDRM is a Denoising Diffusion Implicit Model [14]. The main difference between DDPM [8] and DDIM is that the forward process is assumed Markovian in the former and non-Markovian in the later. This allow us to do use an accelerated generative process where instead of generating each sample $\mathbf{x}_{T:0}$ we generate a subset $\mathbf{x}_{\tau_1:\tau_S}$.

In DDPM, we have the following inference distribution, also called the forward or diffusion process:

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) = \prod_{t=0}^{T-1} q(\mathbf{x}_{t+1}|\mathbf{x}_t) \tag{2}$$

We then get the generative process p_θ, which is an approximation of the reverse forward process:

$$p_\theta(\mathbf{x}_{0:T}) = p_\theta(\mathbf{x}_T) \prod_{t=0}^{T-1} p_\theta(\mathbf{x}_t|\mathbf{x}_{t+1}) \tag{3}$$

But in DDIM, we instead use the following inference distribution:

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) = q(\mathbf{x}_T|\mathbf{x}_0) \prod_{t=0}^{T-1} q(\mathbf{x}_t|\mathbf{x}_{t+1}, \mathbf{x}_0) \tag{4}$$

To train a diffusion model we optimize the following ELBO objective [14]:

$$\sum_{t=1}^{T} \gamma_t \mathbb{E}_{(\mathbf{x}_0,\mathbf{x}_t)\sim q(\mathbf{x}_0)q(\mathbf{x}_t|\mathbf{x}_0)} \left[||\mathbf{x}_0 - f_\theta^{(t)}(\mathbf{x}_t)||_2^2 \right] \tag{5}$$

where $f_\theta^{(t)}$ is our neural network with parameters θ that learns to recover a noiseless observation from a noisy $\mathbf{x}_t$ and $\gamma_{1:T}$ positive coefficients dependent on $q(\mathbf{x}_{1:T}|\mathbf{x}_0)$.

2.3. DDRM

Because we want to solve inverse problems in DDRM and we have access to a measurement, we condition our generative process and the inference distribution on y:

$$p_\theta(\mathbf{x}_{0:T}, \mathbf{y}) = p_\theta(\mathbf{x}_T|\mathbf{y}) \prod_{t=0}^{T-1} p_\theta(\mathbf{x}_t|\mathbf{x}_{t+1}, \mathbf{y}) \tag{6}$$

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0, \mathbf{y}) = q(\mathbf{x}_T|\mathbf{x}_0, \mathbf{y}) \prod_{t=0}^{T-1} q(\mathbf{x}_t|\mathbf{x}_{t+1}, \mathbf{x}_0, \mathbf{y}) \tag{7}$$

In order to use the diffusion model to solve inverse problems, we consider the SVD of the degradation matrix H. The idea of using the SVD on the degradation matrix is an extension of the work from the same authors [10]. We tie the noise in the measurement $\mathbf{y}$ to the noise from the diffusion in $\mathbf{x}_{1:T}$ so that the final result $\mathbf{x}_0$ is coherent with the measurement. Because we consider a possibly noisy inverse problem, the diffusion model can perform denoising at the same time as solving the non-noisy inverse problem. Through the SVD of the degradation matrix, we can understand how the measurement affects the data in $\mathbf{x}$ and synthesize it using the diffusion process.

To exploit the information from the SVD, DDRM proposes the following construction: for each index of the spectral space (i) when the singular value is zero, $\boldsymbol{y}$ does not provide information and the update is unconditional generation, when the singular value is non-zero we want to incorporate the information $\boldsymbol{y}$ provides. We compare the noise in the spectral space $\frac{\sigma_{\boldsymbol{y}}}{s_i}$ to the noise in the diffusion model σ_t, we then incorporate information from $\boldsymbol{y}$ scaled differently depending on the noise in the spectral space and the diffusion model.

We change the variational distribution to take into account the information from the SVD:

$$q^{(T)}(\bar{\mathbf{x}}_t^{(i)}|\mathbf{x}_0, \mathbf{y}) = \begin{cases} \mathcal{N}(\bar{\boldsymbol{y}}^{(i)}, \sigma_T^2 - \frac{\sigma_{\boldsymbol{y}}^2}{s_i^2}) & \text{if } s_i > 0 \\ \mathcal{N}(\bar{\boldsymbol{x}}_0^{(i)}, \sigma_T^2) & \text{if } s_i = 0 \end{cases} \tag{8}$$

$$q^{(t)}(\bar{\mathbf{x}}_t^{(i)}|\mathbf{x}_{t+1}, \mathbf{x}_0, \mathbf{y}) = \begin{cases} \mathcal{N}(\bar{\boldsymbol{x}}_0^{(i)} + \sqrt{1-\eta^2}\sigma_t \frac{\bar{\boldsymbol{x}}_{t+1}^{(i)} - \bar{\boldsymbol{x}}_0^{(i)}}{\sigma_{t+1}}, \eta^2\sigma_t^2) & \text{if } s_i = 0 \\ \mathcal{N}(\bar{\boldsymbol{x}}_0^{(i)} + \sqrt{1-\eta^2}\sigma_t \frac{\bar{\boldsymbol{y}}^{(i)} - \bar{\boldsymbol{x}}_0^{(i)}}{\sigma_{\boldsymbol{y}}/s_i}, \eta^2\sigma_t^2) & \text{if } \sigma_t < \frac{\sigma_{\boldsymbol{y}}}{s_i} \\ \mathcal{N}((1-\eta_b)\bar{\boldsymbol{x}}_0^{(i)} + \eta_b\bar{\boldsymbol{y}}^{(i)}, \sigma_t^2 - \frac{\sigma_{\boldsymbol{y}}^2}{s_i^2}\eta_b^2) & \text{if } \sigma_t \geq \frac{\sigma_{\boldsymbol{y}}}{s_i}. \end{cases} \tag{9}$$

Where $\bar{\boldsymbol{x}} = \mathbf{V}^T\boldsymbol{x}_t$ and $\bar{\boldsymbol{y}} = \Sigma^\dagger\mathbf{U}^T\boldsymbol{y}$.

We get a trainable DDRM with parameters θ by replacing q by p_θ, and $\bar{\boldsymbol{x}}_0$ by $\bar{\boldsymbol{x}}_{\theta,t}$. The two hyperparameters η and η_b control the variance of the transitions.

We can then apply the same diffusion model, unconditioned on the inverse problem, and only modify the degradation matrix H and its SVD for different linear inverse problems. For that reason, optimized algorithms for computing the SVD are necessary. The authors of DDRM exploit the structure of H in specific linear inverse problems (denoising, inpainting, super resolution, deblurring, and colorization) to propose efficient SVD computations.

Some limitations of this method are that it works for general linear inverse problems but requires a memory-efficient SVD that is only available for denoising, inpainting, super resolution, deblurring, and colorization, and still requires to know the degradation matrix H at inference time which is unrealistic for real-world applications.

Several extensions to the DDRM framework exist. An extension to non-linear inverse problems, for JPEG Artifact Correction is proposed in [9], to noisy measurement with non-gaussian noise, or without using SVD [2].

Although DDRM may have been surpassed by other models on common inverse problem tasks [2], its speed is still attractive as it is orders of magnitude faster than the other models as seen in [11] (table 2).

3. Model improvement : Manifold Constraint Gradient

3.1. Method Overview

In *Improving Diffusion Models for Inverse Problems using Manifold Constraints* [3], the authors propose a method a correction term inspired by the manifold constraint which makes inverse-problem-solving models generate data in iterations closer to the underlying data manifold. Originally, these models would throw the generative sampling path off the data manifold which leads to accumulating errors.

The discrete form of the forward diffusion step is

$$\boldsymbol{x}_i = a_i \boldsymbol{x}_0 + b_i \boldsymbol{z}, \quad \boldsymbol{z} \sim \mathcal{N}(0, \boldsymbol{I}) \tag{10}$$

where $i \in \mathbb{N}$, and the reverse diffusion process is represented by

$$\boldsymbol{x}_{i-1} = \boldsymbol{f}(\boldsymbol{x}_i, \boldsymbol{s}_{\theta^*}) + g(\boldsymbol{x}_i)\boldsymbol{z}, \quad \boldsymbol{z} \sim \mathcal{N}(0, \boldsymbol{I}), \tag{11}$$

where $\boldsymbol{f}$, a_i, b_i and g are chosen according to whether we're using the Variance Exploding or Preserving SDE (Stochastic Differential Equation) score-function. $\boldsymbol{s}_\theta$ represents the θ-parametrized model output. For inverse linear problems where we wish to obtain the unknown $\boldsymbol{x} \in \mathbb{R}^{m \times n}$ from a measurement $\boldsymbol{y} \in \mathbb{R}^m$ and a linear degradation matrix $\boldsymbol{H} \in \mathbb{R}^{m \times n}$:

$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{z}, \tag{12}$$

where $\boldsymbol{z} \sim \mathcal{N}(0, \sigma_y^2 \boldsymbol{I})$ is an additive Gaussian Noise with known variance. To impose conditioning on the reverse diffusion process, generally, projection-based methods are used:

$$\boldsymbol{x}'_{i-1} = \boldsymbol{f}(\boldsymbol{x}_i, s_\theta) + g(\boldsymbol{x}_i)\boldsymbol{z}, \quad \boldsymbol{z} \sim \mathcal{N}(0, \boldsymbol{I}), \tag{13}$$

$$\boldsymbol{x}_{i-1} = \boldsymbol{A}\boldsymbol{x}'_{i-1} + \boldsymbol{b}_i, \tag{14}$$

where $\boldsymbol{A}, \boldsymbol{b}_i$ are functions of $\boldsymbol{H}, \boldsymbol{y}$, and $\boldsymbol{x}_0$. This projection-based method enables better generalization capability on the neural network since if we were to condition the score-function $\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x}|\boldsymbol{y})$ we would have to train different models for different tasks. The authors however impose the measurement constraint while using the unconditionally trained score function through the Bayes Rule $p(\boldsymbol{x}|\boldsymbol{y}) = p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{x})/p(\boldsymbol{y})$ which leads to

$$\nabla_{\boldsymbol{x}} \log p(\boldsymbol{x}|\boldsymbol{y}) = \nabla_{\boldsymbol{x}} \log p(\boldsymbol{x}) + \nabla_{\boldsymbol{x}} \log p(\boldsymbol{y}|\boldsymbol{x}). \tag{15}$$

Replacing in (13), we would obtain the following

$$\boldsymbol{x}'_{i-1} = \boldsymbol{f}(\boldsymbol{x}_i, \boldsymbol{s}_\theta) - \alpha \frac{\partial}{\partial \boldsymbol{x}_i} \|\boldsymbol{W}(\boldsymbol{y} - \boldsymbol{H}\boldsymbol{x}_i)\|_2^2 + g(\boldsymbol{x}_i)\boldsymbol{z}, \tag{16}$$

where α and $\boldsymbol{W}$ depend on the noise covariance. Finally, from Tweedie's formula:

$$\mathbb{E}[\boldsymbol{x}_0|\boldsymbol{x}_i] = (\boldsymbol{x}_i + b_i^2 \nabla_{\boldsymbol{x}_i} \log p(\boldsymbol{x}_i))/a_i. \tag{17}$$

where $\boldsymbol{x}_i \sim \mathcal{N}(a_i \boldsymbol{x}_0, b_i^2 I)$, the authors derive the manifold constraint gradient as the set constraint for $\boldsymbol{x}_i$ such that the gradient of the measurement term stays on the data manifold:

$$\boldsymbol{x} \in \mathcal{X}_i, \quad \text{where} \quad \mathcal{X}_i = \{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{x} = (\boldsymbol{x} + b_i^2 \boldsymbol{s}_\theta(\boldsymbol{x}, i))/a_i\} \tag{18}$$

Therefore, the reverse diffusion process is finally represented as:

$$\boldsymbol{x}'_{i-1} = \boldsymbol{f}(\boldsymbol{x}_i, \boldsymbol{s}_\theta) - \alpha \frac{\partial}{\partial \boldsymbol{x}_i} \|\boldsymbol{W}(\boldsymbol{y} - \boldsymbol{H}\hat{\boldsymbol{x}}_0(\boldsymbol{x}_i))\|_2^2 + g(\boldsymbol{x}_i)\boldsymbol{z}, \tag{19}$$

$$\boldsymbol{x}_{i-1} = \boldsymbol{A}\boldsymbol{x}'_{i-1} + \boldsymbol{b}. \tag{20}$$

In this section, we gave an overview of the Manifold Constraint Gradient method as given by its authors. We refer to section *4 Geometry of Diffusion Models and Manifold Constraint Gradient* for the theoretical explanation of the method. We aim to implement this method in the DDRM pipeline and study its effectiveness on the different experiments provided by the framework which will be demonstrated in the following sections.

3.2. Implementation in DDRM

In the reverse diffusion process in DDRM, we sample $\boldsymbol{x}_t$ (which is equivalent to $\boldsymbol{x}_{i-1}$ in the general discrete diffusion process, however, the authors of the DDRM paper opt to use the index i to represent the elements in the SVD-projected vectors) from $p_\theta^{(t)}(\bar{\boldsymbol{x}}_t^{(i)}|\boldsymbol{x}_{t+1}, \boldsymbol{y})$:

$$\begin{cases} \mathcal{N}(\bar{\boldsymbol{x}}_{\theta,t}^{(i)} + \sqrt{1-\eta^2}\sigma_t \frac{\bar{\boldsymbol{x}}_{t+1}^{(i)} - \bar{\boldsymbol{x}}_{\theta,t}^{(i)}}{\sigma_{t+1}}, \eta^2\sigma_t^2) & \text{if } s_i = 0 \\ \mathcal{N}(\bar{\boldsymbol{x}}_{\theta,t}^{(i)} + \sqrt{1-\eta^2}\sigma_t \frac{\bar{\boldsymbol{y}}^{(i)} - \bar{\boldsymbol{x}}_{\theta,t}^{(i)}}{\sigma_{\boldsymbol{y}}/s_i}, \eta^2\sigma_t^2) & \text{if } \sigma_t < \frac{\sigma_{\boldsymbol{y}}}{s_i} \\ \mathcal{N}((1-\eta_b)\bar{\boldsymbol{x}}_{\theta,t}^{(i)} + \eta_b \bar{\boldsymbol{y}}^{(i)}, \sigma_t^2 - \frac{\sigma_{\boldsymbol{y}}^2}{s_i^2}\eta_b^2) & \text{if } \sigma_t \geq \frac{\sigma_{\boldsymbol{y}}}{s_i}. \end{cases} \tag{21}$$

Figure 1. Mesurment signal: $\boldsymbol{y}$

Figure 2. DDRM colorization: $\boldsymbol{x}_0$

Figure 3. DDRM+MCG colorization: $\boldsymbol{x}_0$

Here, we are interested in the case where the corresponding singular value is zero then $\boldsymbol{y}$ does not provide any information to that index, therefore the update is essentially an unconditional generation step. We believe that also adding the constraint term in this index should help in overall performance by projecting the results of the MCG method compared to the DDRM approach. We would obtain for $s_i = 0$:

$$\bar{\boldsymbol{x}}_t^{(i)} = \bar{\boldsymbol{x}}_{\theta,t}^{(i)} + \sqrt{1-\eta^2}\sigma_t \frac{\bar{\boldsymbol{x}}_{t+1}^{(i)} - \bar{\boldsymbol{x}}_{\theta,t}^{(i)}}{\sigma_{t+1}} - \alpha \frac{\partial(\|(\boldsymbol{y} - \boldsymbol{H}\hat{\boldsymbol{x}_0})\|_2^2)}{\partial \bar{\boldsymbol{x}}_{t+1}} \tag{22}$$

3.3. Experiments

To study the impact of integrating the MCG method in the DDRM framework, we propose the following experiments; colorization, inpainting and denoising. We use the LSUN [17] dataset, specifically, the bedroom category. For all of the mentioned tasks, we use $\eta = 1$, $\eta_B = 0.8$ and $\sigma_{\boldsymbol{y}} = 0$. The number of NFE is set to 20 and we process batches of 6 images at a time. We experiment with values for the gradient coefficient in the interval $[1.0, 3.5]$. We study the PSNR metric (Peak Signal to Noise Ratio) which essentially is the maximum possible value (power) of a signal and the power of distorting noise that affects the quality of its representation. We obtain the results shown in table 1. We notice that for the colorization and denoising tasks, we obtain slightly higher PSNR values compared to the native DDRM model while for the inpainting, we obtain almost similar values. This shows that, originally, for the colorization and denoising tasks, we had some accumulation in error while performing the reverse diffusion generative path. Adding the manifold constraint helped in these tasks to keep the unconditional generation path to remain on the measurement data manifold. However, we don't notice an improvement in the inpainting task when choosing the gradient coefficient but we even observed high variance on the output PSNR when changing the gradient coefficient. The inpainting type chosen in these experiments is the *inpainting lorem* task where images have large text overlayed on the original image. This task is a bit different to the one used in the MCG method thus the parameter selection for $\boldsymbol{W}$ and α might differ. In addition, in our implementation, we project the obtained gradient into the spectral space, which also can contribute to the underwhelming performance boost of the MCG method in the DDRM framework.

Method	Colorization	Inpainting	Denoising
DDRM	24.52	**30.08**	51.57
DDRM+MCG	**25.22**	30.07	**51.65**

Table 1. PSNR on different tasks by DDRM native and DDRM+MCG

4. Limits and proposed extensions

Singular Value Decomposition. The core idea of DDRM is to compute the diffusion process in the SVD spectral space of the degradation matrix, making it more efficient than in the image space (very high dimensional) while making use of most of the high level information. This way, DDRM aims to find a trade-off between general unsupervised methods [2] and efficient but more restrictive supervised methods [5]. Despite the benefits of its efficiency and performance in the provided degradation models and settings, this method draws a very rigid framework with limited applications requiring heavy assumptions for several reasons. We will focus on one type of degradation, blur, and propose to explore the most important limitations of the DDRM framework through experimental extensions (with very limited hardware), demonstrating some flaws of the model and paving the way for further improvements.

Unknown kernel. DDRM provides the benefits of being run in an unsupervised setting and needing a known

degradation operator only at inference time [16]. However, real-life applications rarely provide the degradation operator even at inference time. We propose to extend the DDRM framework to the case where the degradation is completely unknown, called the task of **Blind-image deblurring**, hence paving the way to more general appliciations.

Linear Inverse Problem. DDRM poses the image restoration problem as a linear inverse problem and addresses only such scenario. Moreover it requires very strong assumptions on the degradation matrices as the efficient svd implementation relies on linear algebra tricks. Hence DDRM does not generalize in terms of degradation model and it sticks to very particular, simple, uniform degradation models. Real-life blur is often composed of non uniform, non trivial degradation operations [7]. We propose to explore the limits of the DDRM framework and take a first step to bridging the gap to the cases where image degradation is non uniform.

Real-life video domain. We explore generalization of DDRM to real-life data, trivial or non trivial motion blurs in the video domain, both adapting the DDRM framework to video sequences and experimenting its performance on complex natural blurs. We then discuss ways of improving performance by leveraging prior information (previous frames) in the diffusion process to make advantage of the video format.

5. Extension to Unknown Degradation

5.1. Blind-image deblurring

In the image restoration problem, the blur degradation is often modeled by the linear inverse problem ruled by a convolution [16], itself determined by a fixed, spatially non-varying blur kernel k, with H its associated Toeplitz matrix and Z some small additive gaussian noise:

$$Y = k * X + Z$$

ie.

$$Y = HX + Z \tag{23}$$

The degradation operator needs to be known at inference time for DDRM to compute its singular value decomposition and be able to perform the diffusion. In real life scenarios, such as camera movement or rotation blur, such degradations are image and context-specific, hence the degradation operator is unknown and even not generalizable itself to more than one single frame.

Having taken the measure of how restrictive this limitation is and inspired by previous works on degradation kernel estimation, we propose to estimate blur in the image at inference time and perform the diffusion conditional to the estimated degradation operator. Contrarily to the classic DDRM framework, an estimated degradation operator H_i will be assigned to each measurement y_0^i.

Given the limited resources we rely on, we can't train the diffusion but only make use of the ddrm model. Thus, we limit our work in building on the existing framework, exploring it's limits rather than aiming at performance gains.

5.2. Blur estimation

We first focus on degradation models with uniform unknown blur, as the formulation remains the inverse problem ennunciated earlier 23. Surprisingly, most of the previous work in the domain of uniform Blur kernel estimation is not made available, as it is mostly used for commercial purposes (even in academic research). The rare open source models use matlab and mostly non DL methods. We did not have the resources to use matlab code nor to train a model from scratch, and decided to use a non-uniform blur kernel estimator [1] instead, which demonstrated impressive performance on real-life data. The model, based on Adaptive decomposition uses a CNN to predict in a very efficient fashion $K = 25$ kernels $k_l \in \mathbb{R}^{33\times33}$ together with binary masks α_l for pixel-wise combinations of kernels. In other words each pixel is associated with a linear combination of degradation kernel, and for the vectorized writing, each pixel y_j of the measurement computes :

$$y_j = (\sum_{l=k}^{K} \alpha_j^l \times k_l) * x_j + j$$

We use the predicted non uniform masks and combine them to try and estimate the nearest uniform mask ruling the degradation of the image. For this purpose we proposed to reconstruct a single mask by thresholding the kernels applied to a sufficient part of the image and summing them. After a short finetunning it appears best when kernels applied to less than 2.5% of the image are eliminated. We leave as future work other possible combinations such as weighting the sum or adding a small learnable MLP module.

Our intuition was encouraged by studying the reconstructed kernel for a simple uniform separable motion blur on a random batch of BSD images. We estimate the kernel ruling the degradation :

$$\overline{k} = \sum_{l=1}^{K} \mathbb{1}(\sum \alpha^l \times \frac{x \times h}{100} > 2.5)$$

We then reconstruct the blurry image by convolution between the prediction and the original image

$$\overline{y} = \overline{k} * x$$

We get a average PSRN score of 149 between the original blurry image and the estimated blurry image, indicating

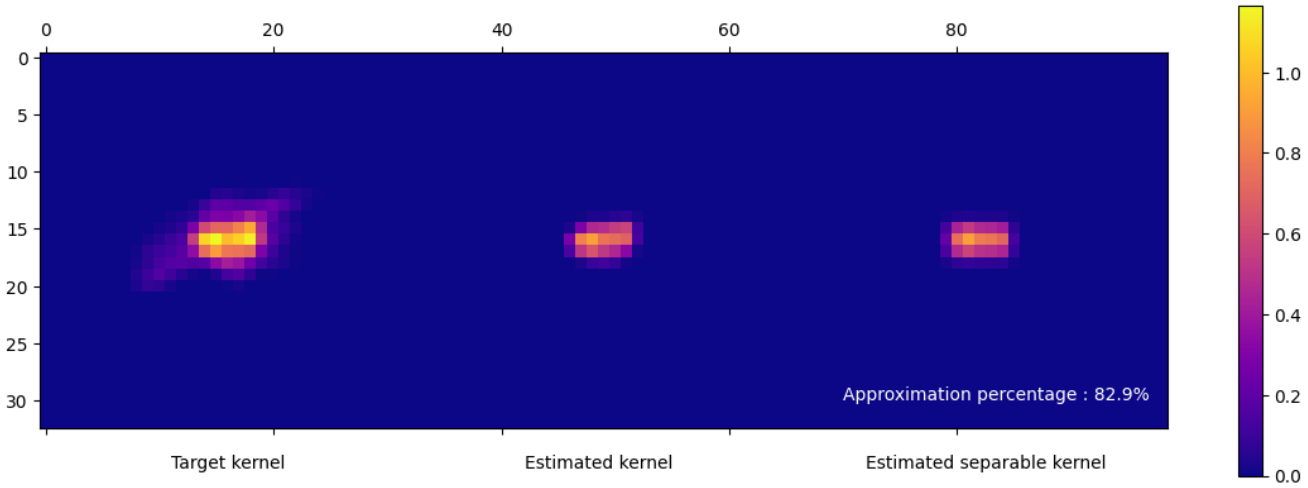

Figure 4. Kernel estimate

that a lot information is retained in the estimation and combination of the kernel. This first hint lead us to propose an end-to-end blind-image ddrm-based restoration method.

5.3. Singular value decomposition

Moreover, in order to have an efficient implementation of the SVD computation which can be very costly, DDRM relies on heavy assumptions on the kernel itself, making the application range of the model very limited : the degradation must be ruled by a 1D degradation kernel or a 2D separable degradation kernel, ie. of rank 1.

We compute the SVD of the reconstructed kernels and produce an estimate of rank 1 by keeping rows attributed to the highest singular value in order to have a separable kernel 4. There again we lose some information about the degradation, but because of the hardware limitations we must stick to the restricted use case of the DDRM diffusion. The Rank 1 estimate allows to keep the essential information as the reconstructed blurred image shares on average 150.0 of PSNR score with the original blurred image, but the separability implies that the kernel preserves only symmetric horizontal and vertical information, disregarding any sense of diagonality or orientation. In simple words, it would prevent the method from approximating the right kernel well enough on any camera rotation blur for instance.

Clearly the kernel is not perfectly approximated (e.g. 4) but some crucial information remains and might be sufficient to allow a first step towards bridging the gap to blind-image uniform deblurring. The estimated kernel being separable, we can apply the anisotropic (2D separable convolution) deblurring strategy from the original DDRM model, compute the degradation spectral value using the kroenecker product of the SVD of the kernels as in the paper and keep the same efficiency regarding the Diffusion.

Degradation model	Noise	PSNR	NFE/s
Ground Truth	✓	**30.98**	1.03
	×	**42.74**	1.05
2D Deblurring	✓	7.98	1.03
	×	6.63	1.06
Gaussian deblurring	✓	8.55	1.05
	×	**8.10**	1.06
Kernel estimation (ours)	✓	**10.14**	0.92
	×	7.43	0.93

Table 2. PSNR comparison on BSD with a challenging uniform motion blur degradation, without or with additive noise($\sigma_0 = 0.02$) and different degradation models : GT degradation operator, DDRM 2D deblurring, DDRM 1D deblurring and kernel estimation (ours)

The only additional costs are the estimation of the kernels, which is very efficient (around 0.2s per frame) and that the batch size must be 1 as each image has its associated degradation model.

Results on a challenging uniform motion blur are reported in 4. Clearly DDRM seems to generalize well to real-life data when using the GT kernels. On the other hands, it struggles when trying to run the diffusion with simple (wrong) degradation models from the paper, which was to be expected. Our proposed method with kernel estimates struggles too and the results are slightly better but still inconsistent and too sensitive to noise. The method still outperforms classic DDRM and opens the way for further improvement, mainly by further improving the kernel estimation phase.

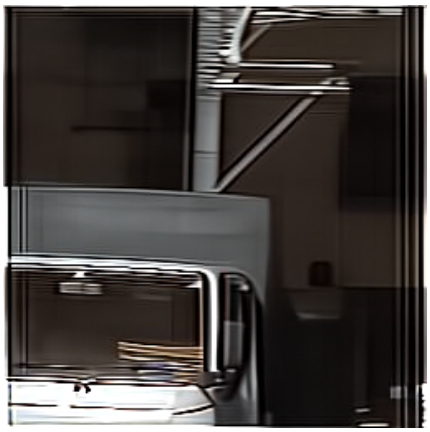

Figure 5. Deblurred Image with estimated kernel

Figure 6. Ground truth Image from the BSD dataset

Figure 7. Blurry image, BSD natural non uniform blur

5.4. Extension to Non-Uniform Blind Deblurring

We apply the same logic of estimating the nearest uniform kernel, but deploy it on the BSD dataset using natural, non-uniform blur and compare with DDRM in order to evaluate the results and test the generalization of DDRM to real-life data 8. In the same way as before, the model outperforms the naive DDRM baseline by a slight margin with much noise, but is very setup and noise-dependant. These results were to expect as this proposed model is only a first step towards bridging the gap with Blind-Image deblurring and non uniform real-world blur restoration.

Moreover, we seek to make better use of the estimated non uniform degradation and investigate the links to the inverse problem categorization. The degradation model can be formulated as the convolution :

$$y = (\sum_{l=1}^{K} \alpha_l k_l) * x + z$$

where x is the vectorized form of X and z the corresponding noise vector. It could be rewritten as a matrix product where H is the Toeplitz matrix associated with the linear combination of kernels. However, DDRM requires an efficient SVD computation, and such a Huge matrix SVDs can't be computed in less than $o(n^4)$ where n is the square image size. Moreover this computation needs a huge computational capabilities (16GB RAM). For small image crops (64x64), a single image deblurring takes around 1 hour. However, combinated with svd estimation algorithms such as stochastic SVD [13] could significantly reduce this computational costs, enabling to overcome what is currently the hardest limitation of the DDRM framework.

6. Extension to Video Format

Extending restoration models to video format is a natural progression in the field of image restoration, driven by the increasing demand for high-quality video content across various applications. Videos often suffer from various degradations, including motion blur, compression artifacts, and noise, which can significantly impact the visual quality and overall viewing experience. Traditional image restoration techniques are typically applied independently to each frame of a video, ignoring the temporal dependencies between consecutive frames. However, videos exhibit temporal coherence, where the content and characteristics of neighboring frames are often correlated. By extending restoration models to video format, we hope to leverage this temporal information to improve the accuracy and consistency of the restoration process. This approach has the potential to deliver enhanced video quality, reduce artifacts, and restore fine details, making it particularly beneficial in areas such as video surveillance, entertainment, and video analysis.

7. Beam-Splitter Deblurring Dataset (BSD)

The Beam-Splitter Deblurring (BSD) dataset [19] is a widely used benchmark dataset for evaluating image deblurring algorithms. The dataset contains a collection of images that have been degraded by motion blur caused by a beam-splitter setup. The blur in these images is non-uniform and complex, making it a challenging task for image restoration algorithms. The BSD dataset is particularly relevant to this project, as the goal is to extend existing deblurring models to handle non-uniform blurring. By evaluating the performance of these models on the BSD dataset, we can compare their effectiveness in restoring images that have been degraded by non-uniform blur. This will help us to identify the most effective techniques for handling this type of blur,

σ_y	Kernel Estimation	Gaussian	Uniform	2D
0.05	**12.9**	12.3	7.1	11.9
0.04	10.3	**11.5**	6.9	11.2
0.03	9.4	10.1	6.27	**10.6**
0.02	8.0	8.9	6.0	**9.34**
0.01	7.21	6.8	5.3	**7.5**
0	**6.2**	6.0	4.9	6.14

Figure 8. Total Average PSNR of DDRM on BSD in different degradation and noise settings.

and to develop new methods for improving the quality of image restoration.

8. Results

Naive DDRM - Our exploratory approach involves using the DDRM model to deblur images from the Beam-Splitter Deblurring (BSD) dataset, which contains non-uniformly blurred images. DDRM can't solve non-linear inverse problems, but this will provide a benchmark for other models and help us evaluate their effectiveness in handling non-uniform blur. Applying this naive DDRM to the non-uniform blur of BSD will also show the limits of the model with respect to non-linear inverse problems. As such, the DDRM model is run on the BSD test set by considering the blur from the dataset to be gaussian, anisotropic or uniform for different levels of noise observed in y. These results will provide a benchmark for the proposed Kernel Estimation method results.

Results - The results in Figure 8 show that, as expected, DDRM performs poorly on the BSD dataset. The Kernel Estimation presents comparable results with the gaussian and anisotropic (2D) degradation settings. The accuracy of all methods increase with increasing noise observed values.

9. Video Extension

9.1. Frame by Frame Approach

Applying an image deblurring restoration model frame by frame on a video serves as a valuable benchmark for evaluating the performance of other video restoration methods. This approach involves independently processing each frame of the video using the deblurring model designed for single images. While simplistic, it provides a baseline to compare against more sophisticated techniques. By evaluating the restoration quality frame by frame, we can assess the model's ability to mitigate blur artifacts, improve image sharpness, and restore fine details in each individual frame. This benchmark helps establish a performance reference point and highlights the limitations of treating video restoration as a frame-wise independent process. It also sets the stage for exploring more advanced techniques that leverage the temporal coherence and dependencies within the video to achieve more accurate and consistent restoration results.

9.2. Capturing Temporal Context

One approach to incorporate temporal information in DDRM for video restoration is to pass consecutive frames as input to the backbone model used to guide diffusion (here, UNet), rather than single frames. By considering the temporal context, the model can capture the motion and dynamics present in the video.

Another variation to enhance the DDRM for video restoration is to incorporate optical flow as additional input. Optical flow provides a dense correspondence field between consecutive frames, representing the pixel-level motion between the frames. By including optical flow information, the model could explicitly leverage the motion cues for better restoration results. Optical flow estimation has been extensively studied in computer vision, and various methods exist, such as the classic Lucas-Kanade algorithm or more recent deep learning-based approaches like FlowNet [6]. Optical flow guidance has been successfully applied in video deblurring tasks, as demonstrated by Su et al. [15]. They incorporate optical flow estimation into their deblurring model to account for motion blur caused by camera shake. By explicitly considering the optical flow, the model successfully reduces motion blur artifacts and improves the overall deblurring performance.

However, it is important to note that non-uniform blur can hinder the accuracy of optical flow estimation. The complex nature of non-uniform blur can distort image gradients and introduce inconsistencies, potentially affecting the reliability of the optical flow estimation; therefore, careful consideration and adaptation of the optical flow guidance approach would be needed. One idea to partially remedy this would be to iteratively use the restored prediction of the adjacent frame(s) of a video to condition the deblurring on the next frame. This method provides advantages other than just increasing the accuracy of the optical flow estimation; real blur in adjacent frames naturally tends to be similar due to their temporal dependency. By conditioning on the restoration predictions of previous frames, we hypothesise the deblurring model will be more accurate.

However, these approaches require re-training the backbone model due to the nature of DDRM and its operations in the spectral space; they will thus not be included in the results due to computational limits, but provide insights for future work.

10. Conclusion

In this project we explored the framework of Denoising Diffusion Restoration Models and it's limitations through experimental extensions. We proposed a first improvement to the DDRM model with the manifold constraint, and extended DDRM to a first step towards efficient restoration in the case of an unknown degradation blur, a non uniform degradation blur, or real-life deblurring.

Moreover, the project also ventured into the domain of video deblurring and restoration. Recognizing the temporal aspect introduced by video sequences, We further discussed possible methods to extend the DDRM framework to the video domain and potentially improving performance by considering consecutive frames and leveraging the knowledge gained from temporal dimension : we aimed to refine the restoration process iteratively, promoting temporal coherence and consistency in the restored videos.

We leave as future work further improvement of these extensions, notably the kernel estimations through more performing kernel estimators in the uniform degradation case, more efficient SVD computation or developping new degradation models. We also leave as future work the implementation of a video-conditioned diffusion restoration model based on DDRM by finetuning a backbone architecture such as ESTRNN [18] or similar.

Overall, this project has shed light on the potential avenues for extending and improving the DDRM model in the context of image and video deblurring, or more precisely on the limitation sof the model. The exploration of unknown blur kernels, non-uniform blur, and video restoration has provided valuable insights into the challenges and opportunities in the field. As further research continues to advance the state-of-the-art in generative modeling for image and video restoration, further enhancement of these experiment could contribute to the broader goal of achieving high-quality and robust restoration results in real-world applications.

References

[1] Guillermo Carbajal, Patricia Vitoria, Mauricio Delbracio, Pablo Musé, and José Lezama. Non-uniform motion blur kernel estimation via adaptive decomposition. *arXiv e-prints*, pages arXiv–2102, 2021. 5

[2] Hyungjin Chung, Jeongsol Kim, Michael T. Mccann, Marc L. Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems, 2023. 1, 3, 4

[3] Hyungjin Chung, Byeongsu Sim, Dohoon Ryu, and Jong Chul Ye. Improving diffusion models for inverse problems using manifold constraints, 2022. 1, 3

[4] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021. 2

[5] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(2):295–307, 2015. 4

[6] Philipp Fischer, Alexey Dosovitskiy, Eddy Ilg, Philip Häusser, Caner Hazırbaş, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks, 2015. 8

[7] Dong Gong, Jie Yang, Lingqiao Liu, Yanning Zhang, Ian Reid, Chunhua Shen, Anton van den Hengel, and Qinfeng Shi. From motion blur to motion flow: a deep learning solution for removing heterogeneous motion blur, 2016. 5

[8] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020. 2

[9] Bahjat Kawar, Jiaming Song, Stefano Ermon, and Michael Elad. Jpeg artifact correction using denoising diffusion restoration models. *arXiv preprint arXiv:2209.11888*, 2022. 1, 3

[10] Bahjat Kawar, Gregory Vaksman, and Michael Elad. Snips: Solving noisy inverse problems stochastically. *Advances in Neural Information Processing Systems*, 34:21757–21769, 2021. 2

[11] Xiangming Meng and Yoshiyuki Kabashima. Diffusion model based posterior sampling for noisy linear inverse problems. *arXiv preprint arXiv:2211.12343*, 2022. 3

[12] Eloi Moliner, Jaakko Lehtinen, and Vesa Välimäki. Solving audio inverse problems with a diffusion model. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023. 1

[13] Ohad Shamir. Fast stochastic algorithms for svd and pca: Convergence properties and convexity. In *International Conference on Machine Learning*, pages 248–256. PMLR, 2016. 7

[14] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 2

[15] Shuochen Su, Mauricio Delbracio, Jue Wang, Guillermo Sapiro, Wolfgang Heidrich, and Oliver Wang. Deep video deblurring, 2016. 8

[16] Li Xu and Jiaya Jia. Two-phase kernel estimation for robust motion deblurring. In *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part I 11*, pages 157–170. Springer, 2010. 5

[17] Fisher Yu, Yinda Zhang, Shuran Song, Ari Seff, and Jianxiong Xiao. LSUN: construction of a large-scale image dataset using deep learning with humans in the loop. *CoRR*, abs/1506.03365, 2015. 4

[18] Zhihang Zhong, Ye Gao, Yinqiang Zheng, and Bo Zheng. Efficient spatio-temporal recurrent neural network for video deblurring. In *European Conference on Computer Vision*, pages 191–207. Springer, 2020. 9

[19] Zhihang Zhong, Ye Gao, Yinqiang Zheng, Bo Zheng, and Imari Sato. Real-world video deblurring: A benchmark dataset and an efficient recurrent neural network. *International Journal of Computer Vision*, pages 1–18, 2022. 7