# Fast Diffusion EM: a diffusion model for blind inverse problems with application to deconvolution

Charles Laroche
GoPro & MAP5
charles.laroche@u-paris.fr

Andrés Almansa
CNRS & Université Paris Cité
andres.almansa@parisdescartes.fr

Eva Coupete
GoPro
ecoupete@gopro.com

## Abstract

*Using diffusion models to solve inverse problems is a growing field of research. Current methods assume the degradation to be known and provide impressive results in terms of restoration quality and diversity. In this work, we leverage the efficiency of those models to jointly estimate the restored image and unknown parameters of the degradation model such as blur kernel. In particular, we designed an algorithm based on the well-known Expectation-Minimization (EM) estimation method and diffusion models. Our method alternates between approximating the expected log-likelihood of the inverse problem using samples drawn from a diffusion model and a maximization step to estimate unknown model parameters. For the maximization step, we also introduce a novel blur kernel regularization based on a Plug & Play denoiser. Diffusion models are long to run, thus we provide a fast version of our algorithm. Extensive experiments on blind image deblurring demonstrate the effectiveness of our method when compared to other state-of-the-art approaches. Our code is available at https://github.com/claroche-r/FastDiffusionEM.*

## 1. Introduction

Image restoration aims to recover information that has been obscured by various degradations such as blur, noise, or compression artifacts. Deep-learning-based methods have revolutionized the field of image restoration by achieving impressive results in various tasks. They leverage the power of deep neural network architectures to learn a mapping between training data [11, 54, 56]. This data-driven approach allows deep-learning models to capture intricate patterns and relationships within the image data, enabling them to restore images with superior quality and perceptual fidelity [27, 50]. On the other hand, model-based approaches express the image restoration problem as an inverse problem and exploit the degradation process structure to design regularizations and optimization algorithms to find the optimal reconstruction [36]. They usually offer more control, flexibility, and interpretability. However, model-based approaches highly rely on the knowledge of the degradation forward process limiting their usefulness in practical applications. Some strategies try to bring the best of both worlds such as Plug-and-Play methods or deep unfolding networks [21, 22, 25, 40, 52]. One of the challenges behind inverse problems comes from their ill-posedness. In fact, for a single degraded image, there generally exist multiple plausible solutions. A common approach is to generate a single restored image that minimizes the mean squared error, but it does not allow the models to generate or hallucinate high-quality details [41, 49]. There is a growing interest in the field of image restoration to design models that can generate all the space of plausible solutions. Those models include Generative Adversarial Networks [15, 32] , conditional or PnP Diffusion Models [24, 41, 42] or Langevin dynamics [26]. This growing interest in diverse restoration is motivated by the impressive perceptual quality obtained by such methods. In particular, diffusion models that were first introduced for image synthesis tasks [19, 20, 43] are now used for a large diversity of tasks such as inverse problem solving [7, 24, 44]. In the field of blind deconvolution, it

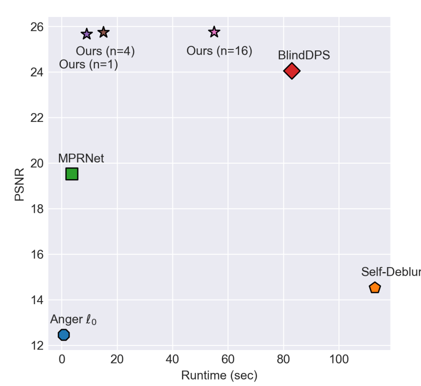



Figure 1. Performance comparison of the different models using the PSNR metric depending on the runtime, "Ours" corresponds to Fast EM ΠGDM method.

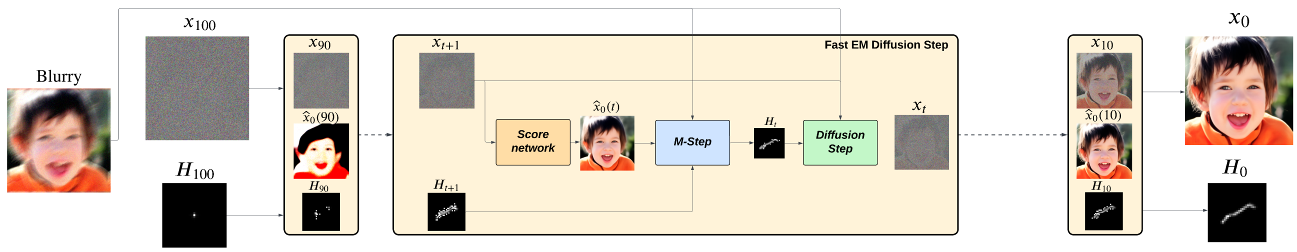



Figure 2. Overview of the method and evolution of the current estimates. We start with random noise and apply the diffusion process. The blurry image intervenes both for the guidance and for the M-step which estimates the blur kernel.

is common to use Bayesian methods to jointly estimate the blur kernel and the restored image [3, 30, 36, 37]. The kernel estimation highly relies on the restoration method that is used and it generally requires the restoration method to produce a sharp image. To do so, image regularizations such as TV, $\ell_0$ on the gradient can be used but they tend to over-sharpen the restored image leading to unpleasant results. Even with the sharp and blurry pairs, it is not easy to estimate any type of blur kernels without efficient regularization. Common regularization on the kernels are the $\ell_1$ norm [6], positivity, the sum to one constraint, and in some cases Gaussian constraints [4]. Some recent works also use deep neural networks such as normalizing flows to parameterize the kernels [28]. Motivated by the impressive quality of diffusion models for both estimated conditional distribution and returning high-quality images, it is natural to believe that they could be used in the context of kernel estimation. Also, a pioneer work [6] that combines parallel diffusion models for the kernel and image exhibits impressive results. Estimating the kernel and image is jointly done in the diffusion process using gradient descent on the forward model. Similarly, methods based on Monte Carlo sampling proposed parameters estimation derived from the Expectation-Maximization (EM) algorithm [13, 17], or the SAPG algorithm [12, 46]. Those methods are very efficient but Monte Carlo sampling is time-consuming. Also, the problem of kernel estimation is a complex problem so those methods highly depend on the regularization imposed in the *M-step* of the EM algorithm.

Motivated by the efficiency of diffusion models, we propose a diffusion model that solves the maximum a-posteriori estimator for blind deconvolution. Derived from the classical Expectation-Maximization algorithm, our model alternately estimates the expected value of the log-likelihood using samples drawn from a diffusion model and maximizes this quantity using half-quadratic splitting. In addition, we also propose a novel kernel regularization in a Plug & Play fashion. Finally, we proposed a fast version of our algorithm to facilitate the use of our method in real-world scenarios. Our experiments show that our proposed solution improves both in terms of fidelity and computational efficiency pushing the Pareto optimal curve further to the origin (Figure 1).

## 2. Background

Let us suppose that our deblurring problem fits the classical inverse problem formulation:

$$y = Hx + n \quad \text{with} \quad n \sim \mathcal{N}(0, \sigma^2) \tag{1}$$

where $x$ is the clean image we want to estimate, $y$ is the blurry and noisy image and $H$ is the degradation operator, a convolution operator in the case of deconvolution. We suppose that we are in the real-world case where we only have access to the blurry image $y$ and the noise level $\sigma$ to reconstruct both the clean image and the blur kernel $H$. In such a setting, a common approach to estimate the blur kernel is to compute the marginalized maximum a-posteriori (MAP) estimator of the inverse problem described in Equation (1):

$$\begin{aligned} H_{MAP} &= \arg\max_H p(H|y) = \arg\max_H p(y|H)p(H) \quad (2) \\ &= \arg\max_H \left[ \log \left( \int p(y|H,x)p(x)dx \right) + \log(p(H)) \right], \end{aligned}$$

with $p(x)$ a natural image prior, $p(y|H,x)$ the likelihood of the blurry image and $p(H)$ the kernel's prior distribution. This MAP estimator cannot be solved easily since the marginalization in the clean image $x$ is not tractable. Expectation-Maximization (EM) [10, 31] is an iterative algorithm that computes the MAP estimator for the parameters of a statistical model ($H$ in our case). It is very convenient when the model contains unobserved or missing data. The EM algorithm consists of two main steps. An *E-step* that computes the expected log-likelihood given the current model parameter estimates and an *M-step*, that maximizes this expected log-likelihood to update the estimated parameters. The whole algorithm alternates between the *E-step* and *M-step* until convergence. In the case of deblurring, the parameter we want to estimate is the blur kernel $H$ and our unobserved data are the clean images associated with the blurry image $y$ and the estimated blur kernel $H$. The EM

algorithm can be summarized as follows in such setting:
*E-Step:*

$$Q(H, H_l) = E_{x\sim p(x|y,H_l)}[\log(p(y|x, H)) + \log(p(x))] \tag{3}$$

*M-Step:*

$$H_{l+1} = \arg\max_H [Q(H, H_l) + \log(p(H))] \tag{4}$$

This formulation is very convenient but in many applications (including blind deblurring), the expected log-likelihood in Equation (3) cannot be computed explicitly, and even taking posterior samples $x \sim p(x|y, H_l)$ is challenging. Our method proposes to approximate the expectation in the *E-step* by an empirical mean in Monte-Carlo EM fashion [48] and to use a diffusion model to obtain posterior samples.

**Diffusion models for posterior sampling:** To learn $p(x_0)$ the distribution of the data, diffusion models define a family of distributions $p(x_t)$ by gradually adding Gaussian noise of variance $\beta(t)$ to samples of $p(x_0)$ until the distribution $p(x_T)$ reduces to a standard Gaussian with zero mean. For discrete timesteps $t \in [\![0, T]\!]$, we can define a Markov transition kernel $p(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta(t)}x_{t-1}, \beta(t)I)$ between two consecutive discrete timestamps. In the general continuous case, [45] described the forward noising process with the following stochastic differentiable equation (SDE) :

$$dx_t = -\frac{\beta(t)}{2}x_t dt + \sqrt{\beta(t)}dw \tag{5}$$

where $w(t)$ is the d-dimensional Wiener process. The reverse SDE of this process [2] can be written as:

$$dx_t = [-\frac{\beta(t)}{2}x_t - \beta(t)\nabla_{x_t}\log\pi(x_t)]dt + \sqrt{\beta(t)}d\bar{w} \tag{6}$$

with $dt$ corresponding to time running backwards and $d\bar{w}$ to the standard Wiener process running backwards. In the case of inverse problems, we want to use diffusion models to generate the posterior distribution $\pi(x_t) = p(x_t|y, H)$. Using Bayes' rule Equation (6) becomes:

$$\begin{aligned} dx_t = \Big[&-\frac{\beta(t)}{2}x_t - \beta(t)\left(\nabla_{x_t}\log p(x_t)\right. \\ &\left. + \nabla_{x_t}\log p(y|x_t, H)\right)\Big]dt + \sqrt{\beta(t)}d\bar{w} \end{aligned} \tag{7}$$

The main problem behind this equation is that in inverse problems, we have a relation between $y$ and $x_0$ but not between $x_t$ and $y$. Marginalizing in $x_0$, we obtain:

$$p(y|x_t) = \int p(y|x_0)p(x_0|x_t)dx_0 \tag{8}$$

that is intractable. The main challenge of non-blind diffusion for posterior sampling is to compute or approximate this integral. In our work, we conduct experiments with DPS [7] and ΠGDM [44] that use different approximations for this integral. Both approximations are based on the mean of $p(x_0|x_t)$, namely:

$$\widehat{x}_0(t) := E[x_0|x_t].$$

DPS approximates $p(x_0|x_t)$ by a delta function

$$p(x_0|x_t) \approx \delta_{\widehat{x}_0(t)}(x_0) \tag{9}$$

whereas ΠGDM approximates $p(x_0|x_t)$ by a Gaussian distribution

$$p(x_0|x_t) \approx \mathcal{N}(x_0|\widehat{x}_0(t), r_t^2) \tag{10}$$

with $r_t$ a hyper-parameter. Both approximations allow us to solve the marginal in Equation (8) analytically and obtain explicit expressions for $\nabla_{x_t}\log p(y|x_t)$ as detailed below.

As a recall, one property of diffusion models is that we can express the noisy measurement $x_t$ in the forward model using the original sample $x_0$:

$$x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon \tag{11}$$

with $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{i=1}^{t}\alpha_i$.
Using a noise predictor $\epsilon(x_t, t)$, we can thus estimate $\widehat{x}_0(t) = E[x_0|x_t]$ at each step $t$ using:

$$\widehat{x}_0(t) = \frac{1}{\sqrt{\bar{\alpha}_t}}(x_t - \sqrt{1-\bar{\alpha}_t}\epsilon(x_t, t)). \tag{12}$$

Equivalently, we can use a score network $s(x_t, t)$ using Tweedie's identity:

$$s(x_t, t) = \nabla_{x_t}\log p(x_t) = -\frac{1}{\sqrt{1-\bar{\alpha}_t}}\epsilon(x_t, t). \tag{13}$$

Using DDPM [19] to discretize the unconditional reverse diffusion process (6) we obtain the update rule

$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(x_t + \beta_t s(x_t, t)\right) + \tilde{\sigma}_t\mathcal{N}(0, I) \tag{14}$$

where $\tilde{\sigma}_t = \sqrt{\beta_t}$ or $\sqrt{\frac{(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\beta_t}$. To simulate the *conditional* reverse diffusion process (7), we just have to add the likelihood term to the score

$$\begin{aligned} x_{t-1} = &\frac{1}{\sqrt{\alpha_t}}\left(x_t + \beta_t\left[s(x_t, t) + \nabla_{x_t}\log p(y|x_t)\right]\right) \\ &+ \tilde{\sigma}_t\mathcal{N}(0, I) \end{aligned} \tag{15}$$

Using Equation (12), the DPS [7] approximation for $p(x_0|x_t)$ leads to the following formula for the gradient of the log-likelihood:

$$\nabla_{x_t} \log p(y|x_t) = -\frac{1}{\sigma^2} \nabla_{x_t} \|y - H\widehat{x}_0(t)\|_2^2 \tag{16}$$

Similarly, the ΠGDM [44] approximation leads to the following gradient for the log-likelihood:

$$\nabla_{x_t} \log p(y|x_t) = \tag{17}$$
$$\left( (y - H\widehat{x}_0(t))^T (r_t^2 HH^T + \sigma^2 I)^{-1} H \left( \frac{\partial \widehat{x}_0(t)}{\partial x_t} \right) \right)^T$$

DPS and ΠGDM derive different guidance terms for the inverse problem. While the DPS approximation leads to a gradient that is easily implemented for any degradation operator $H$ using automatic differentiation, the ΠGDM approximated gradient of Equation (17) is much more complex to estimate for a general operator $H$ because it requires the computation of its pseudo-inverse. On the other hand, the ΠGDM approximation is more precise and thus leads to stronger guidance which is very important for kernel estimation. We summarize in Algorithm 1 the diffusion process for inverse problems when the degradation operator $H$ is known. This case covers both DPS and ΠGDM. The pseudo-code is written using DDPM but is not limited to this particular diffusion scheme. To compensate for the fact that the first estimations of $x_t$ are uncertain, it is common to set $\zeta_t = \sqrt{\bar{\alpha}_t}$, instead of the theoretical $\zeta_t = 1$.

## 3. Method

Our method proposes to solve the MAP of the blur kernel from a blurry and potentially noisy image. We estimate the MAP estimator in an EM fashion. Iteratively, we first draw samples from the posterior distribution knowing the current kernel estimate using a diffusion model. It corresponds to the *E-step* of the EM algorithm. Then, we update our estimated kernel with the *M-step* by maximizing the expected log-likelihood on the previously computed samples. To efficiently model the kernels' distribution, we use a Plug & Play kernel denoiser to regularize our MAP estimator.

### 3.1. E-step: Non-blind diffusion

The *E-step* of the EM algorithm consists in evaluating the expectation from Equation (3). Instead of computing its exact value, we propose to approximate it using random samples in a Monte-Carlo EM fashion. To draw the random samples, we use a non-blind diffusion model. Since the diffusion model targets $p(x|y, H_l)$, sampling several images leads to a good approximation of the expectation. The number $n$ of samples used to approximate the expectation is a hyperparameter of the method. Having many samples leads to a slow but accurate estimation while having only one sample is equivalent to the Stochastic EM algorithm [35]. In practice, the *E-step* reduces to:
*Drawing samples*

$$\boldsymbol{x} = (x^1, ..., x^n) \sim p(x_0|y, H_l) \tag{18}$$

and *updating*

$$\widehat{Q}(H, H_l) = \frac{1}{n} \sum_{i=1}^{n} \log(p(y|x^i, H)). \tag{19}$$

The samples can be drawn by $n$ parallel runs of Algorithm 1, and the empirical mean $\widehat{Q}(H, H_l) \approx Q(H, H_l)$ approaches the expected value in Equation (3) as $n \to \infty$. Unlike in Equation (3), we remove the term in $p(x)$ from $\widehat{Q}(H, H_l)$ here since it does not affect the maximization in the blur kernel $H$.

### 3.2. M-step: Kernel estimation

The *M-step* computes the MAP estimator of the blur kernel using the estimated samples from the *E-step* as measurements. From equations (1), (4) and (19) this step can be summarized as:

$$H_{l+1} = \arg\max_H \hat{Q}(H, H_l) + \log(p(H)) \tag{20}$$

$$H_{l+1} = \arg\min_H \frac{1}{2n\sigma^2} \sum_{i=1}^{n} \|y - Hx^i\|_2^2 + \lambda\Phi(H) \tag{21}$$

where (21) is obtained using Equation (1) and (19). Common choices for $\Phi(.)$ are $\ell_2$ or $\ell_1$ regularizations on top of the simplex constraints on the blur kernel (non-negative values that add up to one). Despite being quite efficient when the blurry image does not have noise, they generally fail to provide good quality results when the noise increases. On the other side, Plug & Play regularizations have become more and more popular for many image restoration tasks. By training a deep denoiser on Gaussian denoising, one can obtain a powerful regularization in the domain on which the denoiser was trained. Generally, we train the denoiser on a dataset of natural images leading to a regularization on natural images. Here, we propose to train a denoiser on a dataset of blur kernels to build a Plug & Play regularization for the blur kernels. We observed that this approach leads to a kernel estimation algorithm that is more efficient and robust to noise, see Figure 4. To solve Equation (21), we use the Half-Quadratic Splitting (HQS) optimization scheme:

$$Z_{j+1} = \arg\min_Z \frac{1}{2\sigma^2 n} \sum_{i=1}^{n} \|Zx^i - y\|_2^2 + \frac{\beta}{2}\|Z - K_j\|_2^2 \tag{22}$$

$$K_{j+1} = \arg\min_K \lambda\Phi(K) + \frac{\beta}{2}\|K - Z_{j+1}\|_2^2 \tag{23}$$

**Algorithm 1** Diffusion model for deblurring

**Require:** $y, \sigma, H, T, (\zeta_t)_t$
**Ensure:** A posterior sample $x_0 \sim p(x_0|y, H)$
  $x_T \leftarrow \mathcal{N}(0, I)$
  **for** $t = T$ **to** 1 **do**
    $\widehat{\epsilon} \leftarrow \epsilon(x_t, t)$
    $\widehat{x}_0 = \frac{1}{\sqrt{\bar{\alpha}_t}}(x_t - \sqrt{1-\bar{\alpha}_t}\widehat{\epsilon})$
    // DPS or ΠGDM approx. using $\hat{x}_0$
    $g \leftarrow \nabla_{x_t} \log p(y|x_t, H)$ ▷ Equation (16) or (17)
    // Compute conditional score $s = \nabla_{x_t} \log p(x_t|y, H)$
    $s \leftarrow \zeta_t g - \frac{1}{\sqrt{1-\bar{\alpha}_t}}\hat{\epsilon}$ ▷ Bayes rule and Tweedie
    // DDPM update rule
    $z \leftarrow \mathcal{N}(0, I)$
    $x_{t-1} \leftarrow \frac{1}{\sqrt{\alpha_t}}(x_t + \beta_t s) + \tilde{\sigma}_t z$
  **end for**
  **return** $x_0$

For the deconvolution problem, Equation (22) can easily be solved in the Fourier domain (more details on the computations can be found in Appendix B). Equation (23) corresponds to the regularization step. It corresponds to the MAP estimator of a Gaussian denoising problem on the variable $Z_{j+1}$. The main idea behind Plug & Play regularization is to replace this regularization step with a pre-trained denoiser $\mathcal{D}$ Mean Squared Error (MSE) loss. This substitution can be done thanks to the close relationship that exists between the MAP and the MMSE estimator of a Gaussian denoising problem [16]. Eventually, the *M-step* consists of the following iterations:

$$Z_{j+1} = \mathcal{F}^{-1}\left(\frac{\mathcal{F}(y)\sum_{i=1}^{n}\overline{\mathcal{F}(x^i)} + n\beta\sigma^2\mathcal{F}(K_j)}{\sum_{i=1}^{n}\mathcal{F}(x^i)\overline{\mathcal{F}(x^i)} + n\beta\sigma^2}\right) \tag{24}$$

$$K_{j+1} = \mathcal{D}_{\sqrt{\lambda/\beta}}(Z_{j+1}). \tag{25}$$

While complex decreasing schemes for $\beta$ are often used to help HQS converge [53], we observed that using a constant $\beta$ was sufficient in our case. For the denoiser architecture, we use a simple DnCNN [54] with 5 blocks and 32 channels. In addition to the noisy kernel, we also give the noise level as an extra channel to the network to control the denoising intensity. Eventually, the complete Diffusion EM algorithm alternates between sampling from the non-blind diffusion model and the HQS algorithm for the kernel estimation. In all our experiments, we use $L = 10$ EM iterations. See Algorithm A.1 in the supplementary.

### 3.3. Fast EM diffusion

The diffusion EM algorithm requires running a diffusion model at each step of the EM algorithm to produce a set of $n$ particles. Executing diffusion models is time-consuming, particularly in cases where inverse problems are addressed using score guidance, as the guidance must be applied to the full-size image, precluding the utilization of acceleration techniques like latent diffusion [38]. Consequently, the diffusion EM algorithm's execution time becomes excessively long, significantly restricting its practical applicability.

**Algorithm 2** Fast EM DPS / ΠGDM

**Require:** $y, \sigma, H_T, T$
**Ensure:** $H \approx \arg\min_H p(y|H)$ and $x_0^i \sim p(x_0|y, H)$
  $\boldsymbol{x_T} \leftarrow (\mathcal{N}(0, I), ..., \mathcal{N}(0, I)) \in (\mathbb{R}^{h*w*3})^n$
  **for** $t = T$ **to** 1 **do**
    $\widehat{\boldsymbol{\epsilon}} \leftarrow \epsilon(\boldsymbol{x}_t, t)$
    $\widehat{\boldsymbol{x}}_0 = \frac{1}{\sqrt{\bar{\alpha}_t}}(\boldsymbol{x}_t - \sqrt{1-\bar{\alpha}_t}\widehat{\boldsymbol{\epsilon}})$
    $H_{t-1}$ = *M-step*$(y, \widehat{\boldsymbol{x}}_{\boldsymbol{0}}, \sigma)$ ▷ Iterate (24) and (25)
    // DPS or ΠGDM approx. using $\hat{x}_0$
    $\boldsymbol{g} \leftarrow \nabla_{\boldsymbol{x}_t} \log p(y|\boldsymbol{x}_t, H_{t-1})$ ▷ Equation (16) or (17)
    // Compute conditional score $s = \nabla_{x_t} \log p(x_t|y, H)$
    $\boldsymbol{s} \leftarrow \zeta_t \boldsymbol{g} - \frac{1}{\sqrt{1-\bar{\alpha}_t}}\hat{\boldsymbol{\epsilon}}$ ▷ Bayes rule and Tweedie
    // DDPM update rule
    $\boldsymbol{z} \leftarrow (\mathcal{N}(0, I), ..., \mathcal{N}(0, I)) \in (\mathbb{R}^{h*w*3})^n$
    $\boldsymbol{x}_{t-1} \leftarrow \frac{1}{\sqrt{\alpha_t}}(\boldsymbol{x}_t + \beta_t \boldsymbol{s}) + \tilde{\sigma}_t \boldsymbol{z}$
  **end for**
  **return** $\boldsymbol{x_0}$, $H_0$

To bypass this problem, we propose a fast version of diffusion EM that incorporates the *M-step* directly into the diffusion process, thereby reducing the number of required diffusion model runs to just one. To do so, we use the $n$ current samples $x_t^i \sim p(x_t|y, H)$ to build an approximation of $Q(H, H_t)$ at each timestep $t$, as follows. First, we use the current distribution estimates $p(x_0|x_t)$ (Equations (9) and (10) for DPS, resp. ΠGDM approximations) for each timestep $t$ to approximate the posterior $p(x_0|y, H)$ by (discretized) marginalization on $x_t$:

$$p(x_0|H, y) = \int p(x_0|x_t)p(x_t|y, H)dx_t \tag{26}$$

$$\approx \sum_{i=1}^{n} p(x_0|x_t^i)p(x_t^i|y, H) \tag{27}$$

$$= \frac{1}{n}\sum_{i=1}^{n} p(x_0|x_t^i) =: q_t(x_0|y, H). \tag{28}$$

Then, using this approximation, the *E-step* at timestep $t$ of the diffusion process is reformulated as follows:

$$Q(H, H_t) = E_{x\sim p(x|y,H_t)}[\log(p(y|x, H)] \tag{29}$$

$$\approx E_{x\sim q_t(x_0|y,H_t)}[\log(p(y|x, H)] \tag{30}$$

Since the distribution $q_t(x_0|y, H)$ progressively converges to the distribution $p(x_0|y, H)$ as $t \to 0$, we have a finer and finer estimation of the expected log-likelihood and thus, the blur kernel, through the iterations.
Finally, the *E-step* reduces in the case of the DPS approxi-

| Metric type | | Reference metrics | | | | No-reference metrics | | Kernel error | |
|---|---|---|---|---|---|---|---|---|---|
| ↓ Method \ Metric → | Time (sec/img) | PSNR ↑ | SSIM ↑ | LPIPS ↓ | FID ↓ | NIQE ↓ | BRISQUE ↓ | MSE kernel ↓ | $\mathcal{L}_{reblur}$ ↓ |
| DPS* | 58sec | 25.81 | 0.76 | 0.34 | 3.46 | 6.28 | 23.52 | ✗ | ✗ |
| ΠGDM* | 5sec | 27.65 | 0.81 | 0.34 | 4.50 | 7.49 | 30.32 | ✗ | ✗ |
| Anger $\ell_0$ | 0.73sec | 12.46 | 0.13 | 0.8 | 233.08 | 12.55 | 50.51 | 5.1e-5 | 1.1e-2 |
| Self-Deblur | 1min53sec | 14.53 | 0.15 | 0.69 | 44.83 | 14.16 | 49.28 | 3.6e-4 | 3.5e-2 |
| MPRNet | 3.7sec | 19.52 | 0.42 | 0.54 | 21.26 | 7.9 | 25.44 | ✗ | ✗ |
| Blind DPS | 1min23 | 24.05 | 0.73 | **0.34** | **2.66** | **6.17** | **20.72** | 3.9e-5 | 5.6e-3 |
| EM ΠGDM (n=1) | 1min30sec | 23.4 | 0.71 | 0.43 | 6.05 | 8.81 | 41.19 | 6.1e-5 | 5.3e-3 |
| EM ΠGDM (n=4) | 2min30sec | 23.21 | 0.71 | 0.4 | 5.43 | 8.23 | 38.02 | 5e-5 | 5.3e-3 |
| EM ΠGDM (n=16) | 9min10sec | 23.09 | 0.71 | 0.39 | 5.11 | 7.91 | 35.42 | 4.1e-5 | 5.3e-3 |
| Fast EM DPS (n=1) | 1min41 | 24.68 | 0.75 | **0.34** | 3.23 | 6.34 | 23.03 | 9e-6 | 5.1e-3 |
| Fast EM ΠGDM (n=1) | 9sec | 25.66 | 0.79 | **0.34** | 4.26 | 7.48 | 30.33 | 1.1e-5 | 5.1e-3 |
| Fast EM ΠGDM (n=4) | 15sec | 25.74 | **0.8** | **0.34** | 4.31 | 7.42 | 30.15 | **6e-6** | **5e-3** |
| Fast EM ΠGDM (n=16) | 55sec | **25.75** | **0.8** | **0.34** | 4.28 | 7.46 | 29.61 | 1.1e-5 | **5e-3** |

Table 1. Model comparison on FFHQ synthetic dataset. Models with a "*" correspond to *non-blind* models used as baselines. Best *blind* models are in **bold** while second best are underlined. Note that baselines do not count for best model rankings.

mation (9) to:

$$\widehat{Q}(H, H_t) = E_{x\sim q_t(x_0|y,H_t)}[\log(p(y|x,H)] \quad (31)$$
$$= \frac{-1}{2\sigma^2 n}\sum_{i=1}^{n} \|H\widehat{x}_0^i(t) - y\|_2^2. \quad (32)$$

In this case, the *M-step* is equivalent to the classical diffusion EM *M-step* of Equation (21) but applied in the current estimate $\widehat{x}_0^i(t)$ instead of the real sample $x^i$. In the case of the ΠGDM approximation (10), we have:

$$\widehat{Q}(H, H_t) = \frac{-1}{2\sigma^2 n}\sum_{i=1}^{n} E_{x\sim\mathcal{N}(\widehat{x}_0^i(t), r_t^2)}[\|Hx - y\|_2^2]. \quad (33)$$

The computations for the *M-step* in that case are left in Appendix D. Eventually, the only difference between the fast EM diffusion algorithm and a classical non-blind diffusion model is that we first estimate the blur kernel before applying the guidance. Our algorithm demonstrates comparable computational efficiency to non-blind diffusion algorithms, as the computation of the *M-step* negligibly impacts the overall diffusion process. The algorithm's pseudo-code can be found in Algorithm 2. Note that in the pseudo-code, the $n$ particles are treated as a batch directly in the $x_t$. To point out this difference, all the variables that are seen as a batch are written in bold.

# 4. Experiments

## 4.1. Experimental settings

We test our algorithm on the first 1000 validation images of the widely used FFHQ [23] 256x256 dataset that we degrade with random motion blur kernels computed using [14] and random Gaussian noise with noise level $\sigma \in \{5, 10, 20\}$. We also provide some results on DIV2K [1] dataset. To achieve a fair comparison, we use the code and pre-trained weights provided by the authors of Blind DPS. For ΠGDM, there is no public code so we re-implemented the model using the Blind DPS code backbone. In our experiments, we observed that DPS needs more iterations to properly converge in comparison to ΠGDM. Indeed, the DPS run needs 1000 iterations while we only use 100 iterations for ΠGDM. For the kernel estimation, we use a bias-free FFDNet [55] denoiser trained on a dataset of motion blur kernels for the Plug & Play regularization. At test time, the *M-step* consists of 10 HQS iterations with hyperparameters $\lambda = 1$ and $\beta = 1e5$. We provide experiments with different numbers of particles for both the Diffusion EM algorithm and the Fast diffusion EM algorithm. We use $n \in \{1, 4, 16\}$. All the models are evaluated on a single A100 GPU.

## 4.2. Compared methods

To test the efficiency of our method, we compare it to state-of-the-art models for deconvolution. We chose to compare against both optimization-based methods, deep learning approaches, and diffusion models to cover all the existing approaches. More specifically, we compare our method to [3] which is a MAP-based method for kernel estimation that uses $\ell_0$ norm on the gradient of the image as an image prior and $\ell_2$ norm to regularize the kernel. We also compare to self-deblur [37] which is a blind deconvolution method that provides both image reconstruction and kernel estimation based on Deep Image Prior. We provide comparisons with MPRNet [51] which is a multi-scale deep learning architecture design for image restoration problems that has proven its efficiency in deblurring. Finally, we compare our kernel estimation methods to Blind DPS [6] which consists of two parallel diffusion models that jointly model the restored image and its corresponding blur kernel. We also computed the results of the non-blind model DPS and ΠGDM to highlight the loss of quality between the blind and non-blind models. For all the methods, we used the

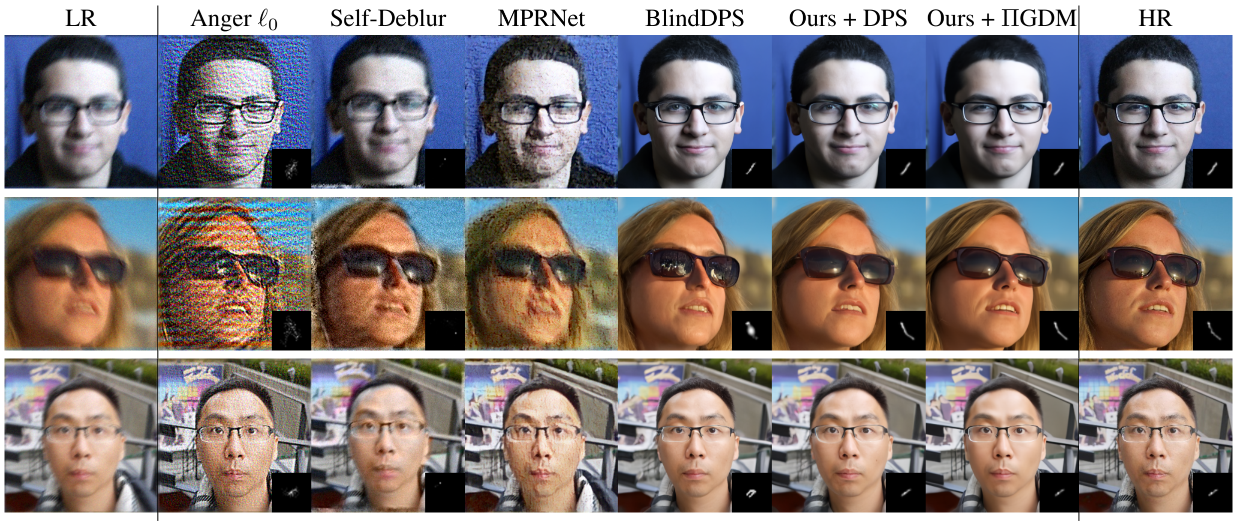



Figure 3. Visual comparison of the different models on a degraded version of the FFHQ 256x256 dataset. Ours correspond to Fast EM.

source code and pre-trained weights provided by the author.

### 4.3. Quantitative results

Table 1 shows the results of the different models on FFHQ synthetic dataset. We compute both classical metrics with full or reduced reference such as PSNR, SSIM [47], LPIPS [57] and FID [18], no-reference metrics to measure perceptual quality such as NIQE [34] and BRISQUE [33] and kernel metrics such as the Mean-Squared Error (MSE) on the reconstructed kernel. We also measure the consistency of the estimated image $\widehat{x}$ and kernel $\widehat{H}$ with the forward model by means of:

$$\mathcal{L}_{reblur}(y, \widehat{x}, \widehat{H}) = \|\widehat{H}\widehat{x} - y\|_2^2 - \sigma^2 M \tag{34}$$

where $M = 3hw$ is the number of elements in vector $x$. We observe that classical optimization-based approaches such as Anger $\ell_0$ [3] and Self-Deblur [37] fail to estimate the blur and reconstruct the image efficiently. The main problem with those approaches is that they fail to produce pleasant results in the presence of noise. While Anger $\ell_0$ [3] produces results with over-sharpened noise, Self-Deblur [37] completely fails to both estimate the kernel and deblur the image. MPRNet produces better results but with artifacts due to the noise, it also fails to recover high-frequency details which is a common problem when using deep-learning models trained on mean-squared error. Diffusion-based models seem to be the most efficient. Blind DPS ranks best among the no-reference perceptual metrics and FID while ranking below our model both for reference metrics and kernel estimation. Figure 3, shows some example images where we can notice the sharpness and high quality of Blind DPS results. In our experiments, we observed that Blind DPS sometimes fails to efficiently estimate the blur kernel, especially in the presence of noise. We also noticed that on some images Blind DPS was producing sharper results than our model, even with a worst kernel prediction which is surprising since we use the same diffusion model. Yet, the fact that our model has better full-reference metrics and better measurement consistency points out the fact that Blind DPS hallucinates more details. We also conducted experiments on deblurring images from DIV2K dataset while keeping the same FFHQ-trained score model for testing. In that particular case, the prior of the score model does not match the distribution of the test images so the model won't be able to hallucinate accurate details. Some visual results of those experiments can be found in Figure 5. Those experiments showed that our model and especially the one based on ΠGDM diffusion produces sharper results. It highlights the fact that Blind DPS and DPS, in general, have weaker guidance than ΠGDM, so it requires a more accurate score model which can be a limitation in practice since training a score model on the space of natural images is not an easy task. During our experiments, we realized that Fast Diffusion EM was both faster and better in terms of quality than

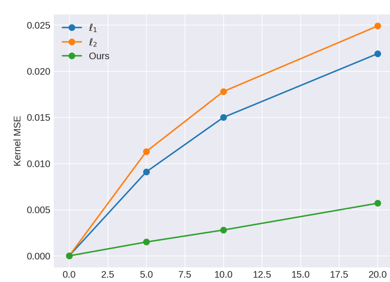
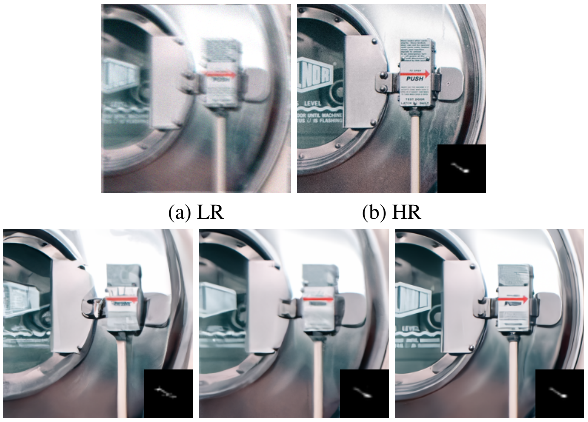



Figure 4. Comparison of the efficiency of the different kernel regularizations depending on the noise level $\sigma \in [0, 20]$. The vertical axis shows the mean MSE over the whole FFHQ dataset for kernel estimation from a noisy and blurred observation of a known image.



(a) LR (b) HR

(c) Blind DPS (d) Fast EM DPS (e) Fast EM ΠGDM

Figure 5. Visual comparison on out-of-distribution images. The score network is trained on FFHQ dataset while we test on DIV2K.

| | n-samples | Runtime | PSNR | PSNR SA |
|---|---|---|---|---|
| Diffusion EM | n=1 | 1min30sec | 23.4 | 23.4 |
| | n=4 | 2min30sec | 23.21 | 23.43 |
| | n=16 | 9min10sec | 23.09 | 23.37 |
| Fast Diffusion EM | n=1 | 9sec | 25.66 | 25.66 |
| | n=4 | 15sec | 25.74 | 26.14 |
| | n=16 | 55sec | 25.75 | 26.16 |

Table 2. Influence of the number of samples used to estimate the *E-step* in Fast EM ΠGDM. The image PSNR is computed on the first image of the batch.

Diffusion EM. Indeed, Diffusion EM is sometimes stuck in the no blur solution while we never observed this problem for Fast Diffusion EM. In terms of metrics, both Fast EM DPS and Fast EM ΠGDM have better reference metrics than all the other methods, and for any number of particles. We observed better performance and faster runtime with the ΠGDM model, probably because it has stronger guidance, thus, it is easier for the *M-step* to estimate the blur kernel. Fast EM ΠGDM performance in no-reference metrics NIQE and BRISQUE is worse than the other diffusion-based methods: BlindDPS and Fast EM DPS have indeed slightly sharper results, but they are less accurate and less consistent (see the hallucinations of BlindDPS in the second line in Figure 3). In terms of runtime, our ΠGDM-based model ranks best among diffusion models but it is significantly slower than MPRNet and Anger $\ell_0$.

### 4.4. Ablation studies

In this section, we discuss the efficiency of the different blocks of our algorithm. We first provide some additional results that show the efficiency of the proposed Plug & Play-based kernel regularization. Next, we study the influence of the number of samples used to estimate the *E-step* on the quality of the final results. To compare the efficiency of our regularization, we compared it against the $\ell_1$ and $\ell_2$ regularizations. To do so, we use our FFHQ synthetic dataset and estimate the blur kernel in the non-blind setting where the sharp and blurry images are both known. We compute the MSE of the reconstructed kernel for several noise levels. For all the regularizations, we used the same optimization scheme, HQS, and fine-tuned the hyper-parameters of the regularizations separately. Figure 4 shows the obtained results. We observed that our regularization is significantly better in the presence of noise and the loss of quality between $\sigma = 5$ and $\sigma = 20$ is very small. Finally, we also investigated the influence of the number of samples in our algorithms. We observed in Table 1 and Table 2 that increasing the number of samples increases the image reconstruction and kernel estimation accuracy. Using all the samples, we can also compute the PSNR on the average of the samples produced by the model. We refer to this metric as the "PSNR SA" in Table 2. Usually, the PSNR SA gives a higher PSNR than the PSNR on a single image, even if the average image is less sharp. We also observed that in the case of Diffusion EM, increasing the number of samples lowers the PSNR but improves all the other metrics. Averaging several samples is also possible with methods such as Blind-DPS, the main difference is that in our approach, all the samples have the same guidance at each diffusion step since we estimate a single kernel for all the samples. In Blind-DPS, all the samples have their respective kernels.

## 5. Conclusion

In this article, we present a novel approach for blind deconvolution based on diffusion models. In particular, we designed Diffusion EM, an algorithm based on the Expectation-Maximization algorithm. This algorithm consists of an *E-step*, which approximates the expected value of the log-likelihood using a diffusion model, and an *M-step*, which maximizes this expected log-likelihood with respect to the unknown parameters (the blur kernel). For the *M-step*, we introduced a novel kernel regularization based on a Plug & Play denoiser. The diffusion EM algorithm is slow since it requires running a diffusion model several times. We propose an acceleration of the algorithm that directly injects the EM iterations into the diffusion process (leveraging the intermediate diffusion steps as approximate posterior samples). We observed that this Fast EM diffusion model reaches better performance than the original diffusion EM algorithm while being significantly faster. Finally, we demonstrate the efficiency of our approach both quantitatively and visually. We compare our approach to state-of-the-art methods for blind deconvolution and provide several ablation studies that highlight the performance of our regularization and model and give insights into the behavior of the model. In its current form, our algorithm is limited to deconvolution. Future research will address more general blind deblurring problems. [5,9]. Faster diffusion models such as latent diffusion [8, 39] or diffusion bridges [29] could also benefit our method.

## References


[1] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, July 2017. 6

[2] Brian D.O. Anderson. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12, 1982. 3

[3] Jérémy Anger, Gabriele Facciolo, and Mauricio Delbracio. Blind Image Deblurring using the l0 Gradient Prior. *Image Processing On Line (IPOL)*, 2019. 2, 6, 7

[4] Sefi Bell-Kligler, Assaf Shocher, and Michal Irani. Blind Super-Resolution Kernel Estimation using an Internal-GAN. In *Advances in Neural Information Processing Systems (NIPS)*, 2019. 2

[5] Guillermo Carbajal, Patricia Vitoria, José Lezama, and Pablo Musé. Blind Motion Deblurring With Pixel-Wise Kernel Estimation via Kernel Prediction Networks. *IEEE Transactions on Computational Imaging*, 9:928–943, aug 2023. 8

[6] Hyungjin Chung, Jeongsol Kim, Sehui Kim, and Jong Chul Ye. Parallel Diffusion Models of Operator and Image for Blind Inverse Problems. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2, 6

[7] Hyungjin Chung, Jeongsol Kim, Michael T. Mccann, Marc L. Klasky, and Jong Chul Ye. Diffusion Posterior Sampling for General Noisy Inverse Problems. In *International Conference on Learning Representations (ICLR)*, 2023. arXiv:2209.14687 [cs, stat]. 1, 3, 4

[8] Hyungjin Chung, Jong Chul Ye, Peyman Milanfar, and Mauricio Delbracio. Prompt-tuning latent diffusion models for inverse problems. *arXiv:2310.01110*, 2023. 8

[9] Valentin Debarnot and Pierre Weiss. Deep-Blur : Blind Identification and Deblurring with Convolutional Neural Networks. *Preprint hal-03687822*, 2022. 8

[10] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977. 2

[11] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Learning a Deep Convolutional Network for Image Super-Resolution. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *European Conference on Computer Vision (ECCV)*, 2014. 1

[12] Gersende Fort, Edouard Ollier, and Adeline Samson. Stochastic proximal-gradient algorithms for penalized mixed models. *Statistics and Computing*, 29(2):231–253, 2019. 2

[13] Angela F Gao, Jorge C Castellanos, Yisong Yue, Zachary E Ross, and Katherine L Bouman. DeepGEM: Generalized Expectation-Maximization for Blind Inversion. In *Advances in Neural Information Processing Systems (NIPS)*, 2021. 2

[14] Fabien Gavant, Laurent Alacoque, Antoine Dupret, and Dominique David. A physiological camera shake model for image stabilization systems. In *SENSORS, IEEE*, pages 1461–1464, 2011. 6

[15] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Nets. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems (NIPS)*, 2014. 1

[16] Rémi Gribonval. Should penalized least squares regression be interpreted as maximum a posteriori estimation? *IEEE Transactions on Signal Processing*, 2011. 5

[17] Bichuan Guo, Yuxing Han, and Jiangtao Wen. AGEM: Solving Linear Inverse Problems via Deep Priors and Sampling. In *Advances in Neural Information Processing Systems*, 2019. 2

[18] 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 I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. 7

[19] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising Diffusion Probabilistic Models. In *Advances in Neural Information Processing Systems (NIPS)*, 2020. 1, 3

[20] Jonathan Ho and Tim Salimans. Classifier-Free Diffusion Guidance. In *NeurIPS Workshop on Deep Generative Models and Downstream Applications*, 2022. arXiv:2207.12598 [cs]. 1

[21] Samuel Hurault, Arthur Leclaire, and Nicolas Papadakis. Gradient Step Denoiser for convergent Plug-and-Play. In *International Conference on Learning Representations (ICLR'22)*, International Conference on Learning Representations, Online, United States, Apr. 2022. 1

[22] Ulugbek S. Kamilov, Charles A. Bouman, Gregery T. Buzzard, and Brendt Wohlberg. Plug-and-play methods for integrating physical and learned models in computational imaging: Theory, algorithms, and applications. *IEEE Signal Processing Magazine*, 40(1):85–97, 2023. 1

[23] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, 2019. 6

[24] Bahjat Kawar, Michael Elad, Stefano Ermon, and Jiaming Song. Denoising Diffusion Restoration Models. In *Advances in Neural Information Processing Systems (NIPS)*, 2022. 1

[25] Charles Laroche, Andrés Almansa, and Matias Tassano. Deep Model-Based Super-Resolution With Non-Uniform Blur. In *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2023. 1

[26] Rémi Laumont, Valentin De Bortoli, Andrés Almansa, Julie Delon, Alain Durmus, and Marcelo Pereyra. Bayesian imaging using plug & play priors: When langevin meets tweedie. *SIAM Journal on Imaging Sciences*, 2022. 1

[27] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. SwinIR: Image Restoration Using Swin Transformer. In *IEEE/CVF International Conference on Computer Vision (CVPR)*, 2021. 1

[28] Jingyun Liang, Kai Zhang, Shuhang Gu, Luc Van Gool, and Radu Timofte. Flow-based kernel prior with application to blind super-resolution. In *Proceedings of the IEEE/CVF*

*Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2
[29] Guan-Horng Liu, Arash Vahdat, De-An Huang, Evangelos A. Theodorou, Weili Nie, and Anima Anandkumar. $I^2$sb: Image-to-image schrödinger bridge, 2023. 8
[30] Ziwei Luo, Haibin Huang, Lei Yu, Youwei Li, Haoqiang Fan, and Shuaicheng Liu. Deep Constrained Least Squares for Blind Image Super-Resolution. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2
[31] Geoffrey J. McLachlan and Thriyambakam Krishnan. *The EM algorithm and extensions*. Wiley series in probability and statistics. Wiley, Hoboken, NJ, 2. ed edition, 2008. 2
[32] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *Arxiv*, 2014. 1
[33] Anish Mittal, Anush Krishna Moorthy, and Alan Conrad Bovik. No-reference image quality assessment in the spatial domain. *IEEE Transactions on Image Processing*, 21(12):4695–4708, 2012. 7
[34] Anish Mittal, Rajiv Soundararajan, and Alan C. Bovik. Making a "completely blind" image quality analyzer. *IEEE Signal Processing Letters*, 20(3):209–212, 2013. 7
[35] Søren Feodor Nielsen. The stochastic em algorithm: Estimation and asymptotic results. *Bernoulli*, 2000. 4
[36] Daniele Perrone and Paolo Favaro. Total Variation Blind Deconvolution: The Devil Is in the Details. In *IEEE/CVF International Conference on Computer Vision (CVPR)*, 2014. 1, 2
[37] Dongwei Ren, Kai Zhang, Qilong Wang, Qinghua Hu, and Wangmeng Zuo. Neural Blind Deconvolution Using Deep Priors. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2, 6, 7
[38] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022. 5
[39] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, June 2022. 8
[40] Ernest Ryu, Jialin Liu, Sicheng Wang, Xiaohan Chen, Zhangyang Wang, and Wotao Yin. Plug-and-Play Methods Provably Converge with Properly Trained Denoisers. In *International Conference on Machine Learning (ICML)*, 2019. 1
[41] 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 (TPAMI)*, 2023. 1
[42] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *Proceedings of the 32nd International Conference on Machine Learning*, 2015. 1
[43] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising Diffusion Implicit Models. In *International Conference on Learning Representations (ICLR)*, 2021. 1
[44] Jiaming Song, Arash Vahdat, Morteza Mardani, and Jan Kautz. Pseudoinverse-Guided Diffusion Models for Inverse Problems. In *International Conference on Learning Representations (ICLR)*, 2023. 1, 3, 4
[45] 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 (ICLR)*, 2021. 3
[46] Ana Fernandez Vidal, Valentin De Bortoli, Marcelo Pereyra, and Alain Durmus. Maximum likelihood estimation of regularisation parameters in high-dimensional inverse problems: an empirical Bayesian approach. Part I: Methodology and Experiments. *SIAM Journal on Imaging Sciences*, 13(4):1945–1989, nov 2019. 2
[47] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 2004. 7
[48] Greg C. G. Wei and Martin A. Tanner. A monte carlo implementation of the em algorithm and the poor man's data augmentation algorithms. *Journal of the American Statistical Association*, 85:699–704, 1990. 3
[49] Jay Whang, Mauricio Delbracio, Hossein Talebi, Chitwan Saharia, Alexandros G. Dimakis, and Peyman Milanfar. Deblurring via Stochastic Refinement. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 1
[50] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Multi-Stage Progressive Image Restoration. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1
[51] 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 (CVPR)*, pages 14821–14831, June 2021. 6
[52] Kai Zhang, Luc Van Gool, and Radu Timofte. Deep Unfolding Network for Image Super-Resolution. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1
[53] Kai Zhang, Yawei Li, Wangmeng Zuo, Lei Zhang, Luc Van Gool, and Radu Timofte. Plug-and-play image restoration with deep denoiser prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 5
[54] 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*, 2017. Conference Name: IEEE Transactions on Image Processing. 1, 5
[55] Kai Zhang, Wangmeng Zuo, and Lei Zhang. FFDNet: Toward a fast and flexible solution for cnn-based image denoising. In *IEEE Transaction on Image Processing*, 2018. 6
[56] Kai Zhang, Wangmeng Zuo, and Lei Zhang. Learning a Single Convolutional Super-Resolution Network for Multi-

ple Degradations. In *EEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1

[57] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 7