Posterior Sampling using Diffusion Models for HDR Reconstruction

Jamin Xie
Computational Imaging Lab, Stanford University
jmx@stanford.edu

Abstract

Posterior sampling methods offer a powerful framework for solving inverse problems such as High Dynamic Range (HDR) image reconstruction by leveraging pre-trained diffusion models as expressive priors. In this setting, reconstruction is guided by a single distorted measurement, and the task is framed as a nonlinear Bayesian inverse problem, bypassing the limitations of traditional multi-exposure or CNN-based approaches. However, prior diffusion-based sampling methods rely on denoising steps that primarily correct local errors, which limits their effectiveness for complex, nonlinear problems like HDR reconstruction.

Diffusion Annealing Posterior Sampling (DAPS) addresses this limitation by decoupling steps in the sampling trajectory and gradually annealing noise to draw samples from the true posterior, thereby enabling global error correction. We evaluate DAPS with both pixel-space and latent pre-trained diffusion models across three HDR reconstruction tasks—symmetric distortion, overexposure, and underexposure—on FFHQ and ImageNet datasets. Our results demonstrate that posterior sampling with diffusion models enables the generation of diverse, high-fidelity HDR images, showcasing its potential for realistic and training-free image reconstruction.

1. Introduction

When solving inverse problems, the goal is to recover the unknown true signal $\boldsymbol{x}_0 \in \mathbb{R}^n$ from observed measurements $\boldsymbol{y} \in \mathbb{R}^m$:

$$\boldsymbol{y} = \mathcal{A}(\boldsymbol{x}_0) + \boldsymbol{n} \tag{1}$$

where $\mathcal{A}$ represents some forward model or measurement operator (linear or nonlinear) and $\boldsymbol{n} \in \mathbb{R}^m$ is typically Gaussian noise [14, 19], however other noise models like Poisson noise are also common. We are typically interested in the case where $m < n$, which leads to an ill-posed inverse problem where we must constrain the solution space to find a meaningful solution because the measurement process is noisy and many-to-one [23].

Because the inverse problem is ill-posed, it is critical to define what the objective of solving the problem is. For many objectives, the Bayesian inverse problem framework is relevant, where the solution space is characterized by the posterior distribution $p(\boldsymbol{x}_0|\boldsymbol{y}) \propto p(\boldsymbol{y}|\boldsymbol{x}_0)p(\boldsymbol{x}_0)$ [23]. Even so, it is possible for example that we focus on maximizing the posterior distribution $p(\boldsymbol{x}_0|\boldsymbol{y})$, which leads to the maximum a posteriori (MAP) estimation [4]. However, we will focus on sampling approximately from the posterior distribution.

One nonlinear inverse problem where deep learning has been recently popularized is High Dynamic Range (HDR) imaging. HDR is important in image processing, computer graphics, and photography, and its general aim is to capture a wider range of intensity levels compared to Low Dynamic Range (LDR) images, creating more realistic visuals and improving contrast and detail [22]. Additionally, LDR limits downstream applications like image-based lighting and depth of field rendering, and an increasing number of consumer displays support HDR [21]. However, capturing HDR with special HDR cameras is currently prohibitively expensive for the general public. As a result, much of existing images are LDR, so one major goal of HDR imaging is to expand the dynamic range of images captured with LDR cameras.

There are two main paradigms of doing LDR-to-HDR conversion: multi-exposure and single-exposure. In multi-exposure, a set of LDR images of different exposures are combined to achieve HDR. However, multi-exposure suffers from difficulties in aligning reference images due to scene motion or requires specialized optical systems [22]. On the other hand, single-exposure HDR image reconstruction avoids these limitations, resulting in the potential to recover the full dynamic range of images from standard or legacy LDR cameras.

HDR reconstruction falls in the paradigm of generative modeling, from which there have been recent efforts to use posterior sampling in combination with diffusion or flow-based methods.

Furthermore, generative modeling in latent space has been demonstrated to perform better than pixel-based meth-

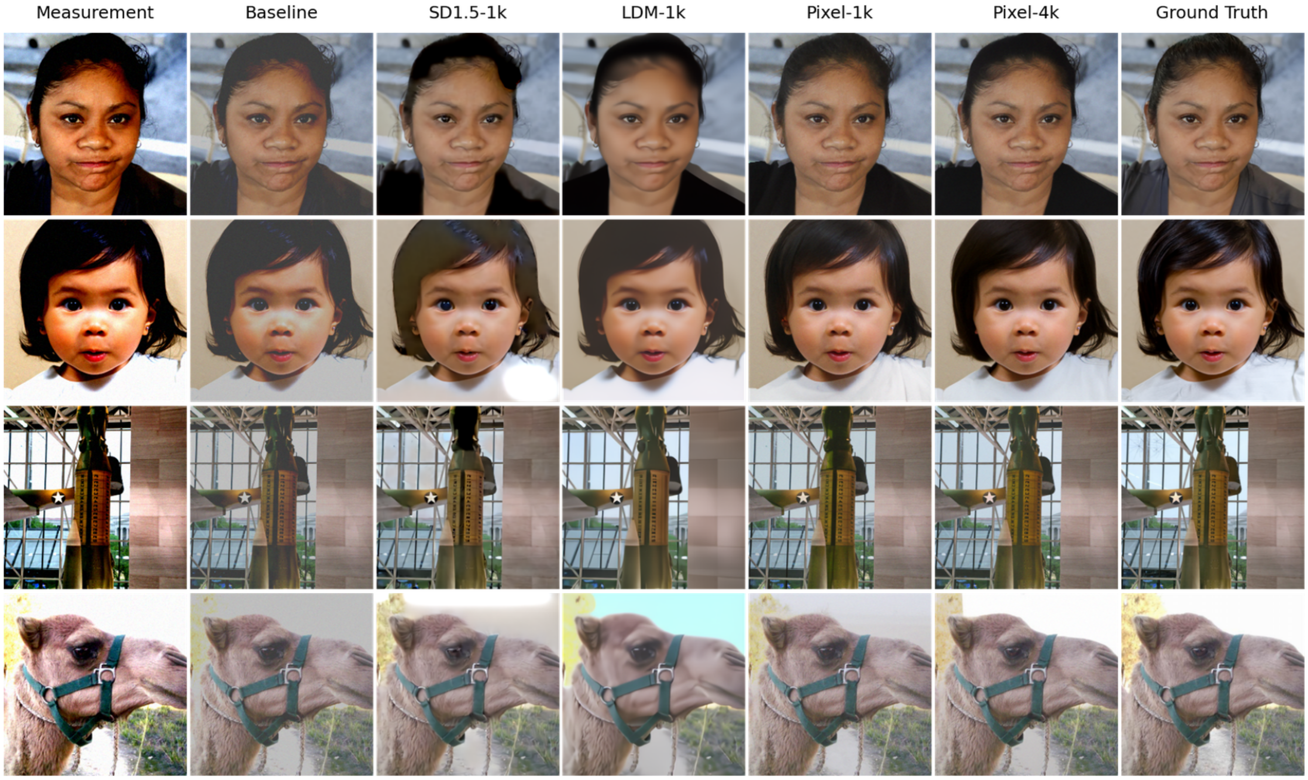


Figure 1. Comparison between DAPS sampling methods and baseline on the symmetric HDR reconstruction task. The first two rows are from FFHQ, and the bottom two rows are from ImageNet.

ods, exemplified by latent diffusion models (LDMs) [17]. Working in latent space attempts to improve flexibility, efficiency, scalability, and inference time and involves encoding and decoding samples between pixel and latent spaces.

1.1. Problem Statement

In this work, we will focus on single-exposure HDR reconstruction, or expanding the dynamic range given a single noisy and degraded measurement. The reconstruction will not be true HDR, but rather an LDR image with higher dynamic range than the input measurement. Using posterior sampling in conjunction with pre-trained diffusion models, we can perform reconstruction with inference only with needing further model training or fine tuning, thereby leveraging pre-trained models as powerful priors.

2. Background and Related Work

2.1. Classical HDR Methods

Before using diffusion models, many classical HDR methods were proposed for solving the inverse problem of single image HDR reconstruction. HDRCNN [7, 22] was an important approach in this domain, which utilized a hybrid LDR encoder and HDR decoder with deep convolutional neural networks (CNNs). Santos *et al*. [18] similarly used a CNN-based model, with a unique idea of masking saturated pixels, therefore achieving better performance on areas of the image with information loss due to overexposure. However, more recently CNNs have been phased out in favor of diffusion models.

2.2. Diffusion and Flow-based Models and Methods

Diffusion and flow models are similar paradigms of generative models that involve simulating stochastic differential equations (SDEs) and ordinary differential equations (ODEs) [10], which are similar except for SDEs being non-deterministic due to stochasticity. Despite being based on differential equations, they can be trained using neural networks [1] using score matching or flow matching.

In training these diffusion and flow models, various backbone architectures are used, although transformer-based architectures have demonstrated to be a superior backbone compared to convolution-based U-Nets. One well-known example is Stable Diffusion (SD). SD v1.5 [17] uses an autoencoder and U-Net backbone; SD 3 [8] replaces the U-Net with a transformer-based architecture [16] and uses flow-based sampling.

One recent approach LEDiff [21] utilizes the powerful

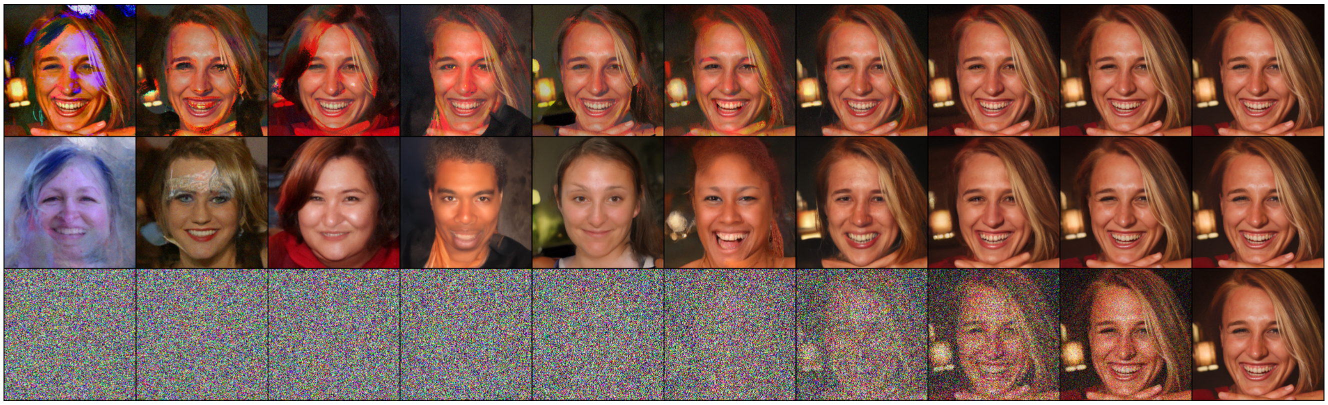


Figure 2. **Sample trajectory of DAPS for an FFHQ image.** The first row represents $\boldsymbol{x}_{0|\boldsymbol{y}}$, the second row $\hat{\boldsymbol{x}}_0(\boldsymbol{x}_t)$, and the third row $\boldsymbol{x}_t$. From left to right, the noise level goes from $\sigma = \sigma_T$ to $\sigma = \sigma_0 = 0$, which is the noise annealing process. Row 3 to 2 is reverse diffusion with the ODE solver. Row 2 to 1 is Langevin dynamics. Row 1 to 3 is forward diffusion. Reconstruction was done with Pixel-4k on the symmetric task.

variational autoencoder (VAE) and latent space of SD 3, merging exposure brackets in the latent space and fine-tuning a VAE to decode into a final HDR image. SD 3 and other pre-trained models are trained on a large amount of LDR samples and are especially useful in tasks such as inverse problems where re-training a full model is not efficient. Typically, only fine-tuning is needed at most.

2.3. Formulation of Diffusion Models

Diffusion models are typically score-based [20], and they generally work by using a stochastic differential equation (SDE) to smoothly and gradually transform a complex data distribution into a into a known, simple distribution like a Gaussian by slowly injecting noise, then using the corresponding reverse-time SDE to slowly remove the noise and recover the data distribution. The training process, known as score matching, learns the score function $\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x_t})$. The continuous time SDE that models the corruption process is [4, 9]

$$\mathrm{d}\boldsymbol{x}_t = \boldsymbol{f}(\boldsymbol{x}_t, t)\mathrm{d}t + g(t)\mathrm{d}\boldsymbol{W}_t \tag{2}$$

with $t = 0$ corresponding to the data distribution, $\boldsymbol{x}_0 \sim p_0$, the first term representing a deterministic drift function, and the second term representing a Wiener process responsible for the stochasticity. Following the SDE, possibly taking discrete steps in practice, we transform the data distribution into Gaussian noise. Denote with p_t the distribution at time t, so that p_T approximates a Gaussian or other simple distribution. Then, we can follow the reverse SDE [4] running backwards in time to sample from the data distribution, initialized with $\boldsymbol{x}_T \sim p_T$:

$$\mathrm{d}\boldsymbol{x}_t = \left(\boldsymbol{f}(\boldsymbol{x}_t, t) - g^2(t)\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x_t})\right)\mathrm{d}t + g(t)\mathrm{d}\boldsymbol{W}_t \tag{3}$$

where we can see the score function $\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x_t})$ that can be learned with a neural network [10]. The remarkable result is that we can initalize the process at $t = T$ and then sample from the data distribution $p(\boldsymbol{x}_0)$.

The drift coefficient $\boldsymbol{f}(\boldsymbol{x}_t, t)$, diffusion coefficient $g(t)$, and noise schedule are design choices. Two popular variants of the SDE are variance exploding (VE) and variance preserving (VP) [4, 11]. In particular, VE sets $\boldsymbol{f}(\boldsymbol{x}_t, t) = \boldsymbol{0}$ and $g(t) = \sqrt{\frac{\mathrm{d}\sigma_t^2}{\mathrm{d}t}} = \sqrt{2\dot{\sigma}_t\sigma_t}$ for a noise/variance schedule σ_t [4, 11, 23]. Following the VE formulation, we can derive that the marginal distribution at time t of the forward process is

$$\boldsymbol{X}_t = \boldsymbol{X}_0 + \sigma_t\boldsymbol{Z}, \boldsymbol{X}_0 \sim p(\boldsymbol{X}_0), \boldsymbol{Z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}), \tag{4}$$

which can be interpreted as adding Gaussian noise with standard deviation σ_t to the data. Note that marginal distribution $\boldsymbol{X}_t$ refers to the distribution over all possible $\boldsymbol{x}_t$ at time t. To sample from the data distribution $p(\boldsymbol{x}_0)$, we draw a sample from $\mathcal{N}(\boldsymbol{0}, \sigma_T^2\boldsymbol{I})$, then use the approximate score function obtained by training a diffusion model and the reverse SDE.

2.4. Bayesian Framework for Posterior Sampling

Diffusion models serve as powerful, expressive priors for $p(\boldsymbol{x_0})$, enabling sampling from the posterior distribution $p(\boldsymbol{x_0}|\boldsymbol{y})$ given a measurement without further training or fine-tuning. Recall that for an inverse problem, we have a forward model given by (1) as $\boldsymbol{y} = \mathcal{A}(\boldsymbol{x}_0) + \boldsymbol{n}$. Assuming $\boldsymbol{n} \sim \mathcal{N}(\boldsymbol{0}, \beta_{\boldsymbol{y}}^2\boldsymbol{I})$, in the Bayesian framework, we have $p(\boldsymbol{y}|\boldsymbol{x}_0) = \mathcal{N}(\mathcal{A}(\boldsymbol{x}_0), \beta_{\boldsymbol{y}}^2\boldsymbol{I})$.

By applying Bayes' theorem [4], we notice that

$$\nabla_{\boldsymbol{x}} \log p(\boldsymbol{x}|\boldsymbol{y}) = \nabla_{\boldsymbol{x}} \log p(\boldsymbol{x}) + \nabla_{\boldsymbol{x}} \log p(\boldsymbol{y}|\boldsymbol{x}) \tag{5}$$

Now we can modify the reverse SDE to approximately sample from the posterior distribution [23], plugging in the assumptions from the VE formulation:

$$\begin{aligned} \mathrm{d}\boldsymbol{x}_t &= -2\dot{\sigma}_t\sigma_t\nabla_{\boldsymbol{x}_t}\log p_t(\boldsymbol{x_t})\mathrm{d}t \\ &- 2\dot{\sigma}_t\sigma_t\nabla_{\boldsymbol{x}_t}\log p_t(\boldsymbol{y}|\boldsymbol{x_t})\mathrm{d}t + \sqrt{2\dot{\sigma}_t\sigma_t}\mathrm{d}\boldsymbol{W}_t \end{aligned} \tag{6}$$

We have a diffusion model to approximate the score $\nabla_{\boldsymbol{x}_t}\log p_t(\boldsymbol{x_t})$, so then the goal is estimate the intractable $\nabla_{\boldsymbol{x}_t}\log p_t(\boldsymbol{y}|\boldsymbol{x_t})$, which can be interpreted as the noisy likelihood.

One notable approach is termed Diffusion Posterior Sampling (DPS) [3], which estimates

$$\begin{aligned} p(\boldsymbol{y}|\boldsymbol{x_t}) &\approx p(\boldsymbol{y}|\mathbb{E}[\boldsymbol{x}_0|\boldsymbol{x}_t]), \text{where} \\ \mathbb{E}[\boldsymbol{x}_0|\boldsymbol{x}_t] &= \mathbb{E}_{\boldsymbol{x}_0\sim p(\boldsymbol{x}_0|\boldsymbol{x}_t)}[\boldsymbol{x}_0] \end{aligned} \tag{7}$$

is the posterior mean [3]. DPS proves to successfully solve general noisy inverse problems, however as it involves accurately solving the reverse SDE and therefore taking small time steps Δt, each reverse step only results in a local error correction in $\boldsymbol{x}_t$, so global errors are difficult to correct.

Other methods of posterior sampling have been tested with varying goals and success. Flow-Driven Posterior Sampling (FlowDPS) [14] uses flow matching and flow-based models instead of score-matching with diffusion models. FlowDPS can be seamlessly integrated into latent flow models, but it has mostly been tested on linear inverse problems so far, not nonlinear. Another notable method thats extends on DPS is BlindDPS [2], which seeks to solve blind inverse problems, where the forward measurement operator is unknown. BlindDPS jointly estimates the unknown image and the unknown operator, which is computationally intensive.

We would like to emphasize that all of these posterior sampling methods only involve sampling, and no additional training or fine-tuning is needed because we leverage powerful pre-trained models.

3. Methods

In our experiments, we focus on the Decoupled Annealing Posterior Sampling (DAPS) [23], which resolves the lack of global error correction that DPS [3] was unable to address. The restriction to local error corrections is primarily caused by restricting to solving the reverse SDE. Therefore, DAPS uses a noise annealing process instead of solving the reverse SDE, gradually reducing noise until $\boldsymbol{x}_0$ is sampled. The key proposition, proved in [23], is that we can sample from $p(\boldsymbol{x}_{t_2}|\boldsymbol{y})$ with noise level σ_{t_2} given any sample $\boldsymbol{x}_{t_1}$ at another noise level σ_{t_1}. Therefore we can start from $t = T$ and iteratively sample from $p(\boldsymbol{x}_t|\boldsymbol{y})$ with the noise level annealing down from σ_T to 0. An illustration of the DAPS process is in 2.

The first step is sampling $\boldsymbol{x}_{0|\boldsymbol{y}} \sim p(\boldsymbol{x}_0|\boldsymbol{x}_t,\boldsymbol{y})$, where $\boldsymbol{x}_t$ and $\boldsymbol{y}$ are known. Analyzing a probabilistic graph in [3], we notice that $\boldsymbol{y}$ and $\boldsymbol{x}_t$ are conditionally independent given $\boldsymbol{x}_0$. Hence we use Bayes' theorem to get

$$p(\boldsymbol{x}_0|\boldsymbol{x}_t,\boldsymbol{y}) \propto p(\boldsymbol{x}_0|\boldsymbol{x}_t)p(\boldsymbol{y}|\boldsymbol{x}_0) \tag{8}$$

$p(\boldsymbol{x}_0|\boldsymbol{x}_t)$ turns out to be intractable, so [23, 3] propose to approximate it with a Gaussian as follows:

$$p(\boldsymbol{x}_0|\boldsymbol{x}_t) \approx \mathcal{N}(\boldsymbol{x}_0;\hat{\boldsymbol{x}}_0(\boldsymbol{x}_t), r_t^2\boldsymbol{I}) \tag{9}$$

where $\hat{\boldsymbol{x}}_0(\boldsymbol{x}_t) = \mathbb{E}[\boldsymbol{x}_0|\boldsymbol{x}_t]$ is an estimator of $\boldsymbol{x}_0$ given $\boldsymbol{x}_t$ which can be interpreted as a unique posterior mean [3]. The variance r_t^2 is specified heuristically. To compute the posterior mean $\hat{\boldsymbol{x}}_0(\boldsymbol{x}_t)$, DAPS employs an Euler ODE solver that solves the probability flow ODE for the forward diffusion SDE given by $\mathrm{d}\boldsymbol{x}_t = \sqrt{2\dot{\sigma}_t\sigma_t}\mathrm{d}\boldsymbol{W}_t$, which as demonstrated in [11, 20] is

$$\mathrm{d}\boldsymbol{x}_t = -\dot{\sigma}_t\sigma_t\nabla_{\boldsymbol{x}_t}\log p(\boldsymbol{x}_t)\mathrm{d}t \tag{10}$$

Notice that the score $\nabla_{\boldsymbol{x}_t}\log p(\boldsymbol{x}_t)$ is exactly what the pre-trained diffusion model was trained to predict, so following [12], we use the pre-trained diffusion model to solve the ODE given $\boldsymbol{x_t}$ and t as initial values, thereby computing $\hat{\boldsymbol{x}}_0(\boldsymbol{x}_t)$. DAPS uses an Euler solver [11, 23] with noise schedule $\sigma_t = t$ and discretized steps from time t to 0. The exact number of ODE steps is a hyperparameter that can be tuned, as is the number of noise annealing steps from above.

Now using the Gaussian approximation for $p(\boldsymbol{x}_0|\boldsymbol{x}_t)$, assuming that the measurement noise is a Gaussian with variance $\beta_{\boldsymbol{y}}^2$, and continuing from the application of Bayes' theorem in (8), we sample from $\boldsymbol{x}_{0|\boldsymbol{y}}$ using Langevin dynamics [23], which has update rule

$$\begin{aligned} \boldsymbol{x}_0^{(j+1)} &= \boldsymbol{x}_0^{(j)} + \eta\nabla_{\boldsymbol{x}_0^{(j)}}\log p(\boldsymbol{x}_0^{(j)}|\boldsymbol{x}_t) \\ &+ \eta\nabla_{\boldsymbol{x}_0^{(j)}}\log p(\boldsymbol{y}|\boldsymbol{x}_0^{(j)}) + \sqrt{2\eta}\epsilon_j \\ &= \boldsymbol{x}_0^{(j)} - \eta\nabla_{\boldsymbol{x}_0^{(j)}}\frac{\left\|\boldsymbol{x}_0^{(j)} - \hat{\boldsymbol{x}}_0(\boldsymbol{x}_t)\right\|^2}{2r_t^2} \\ &- \eta\nabla_{\boldsymbol{x}_0^{(j)}}\frac{\left\|\mathcal{A}(\boldsymbol{x}_0^{(j)}) - \boldsymbol{y}\right\|^2}{2\beta_{\boldsymbol{y}}^2} + \sqrt{2\eta}\epsilon_j \end{aligned} \tag{11}$$

where η is the step size and ϵ_j is drawn from the unit Gaussian.

Now that we've sampled $\boldsymbol{x}_{0|\boldsymbol{y}}$, we can sample $\boldsymbol{x}_t \sim \mathcal{N}(\boldsymbol{x}_{0|\boldsymbol{y}}, \sigma_t^2\boldsymbol{I})$. We repeat this process for a number of noise annealing steps until $\boldsymbol{x}_0$ is sampled. In this process, $\boldsymbol{x}_t$ and $\boldsymbol{x}_{t+\Delta t}$ are conditionally independent given $\boldsymbol{x}_0$, hence the decoupled in decoupled noise annealing.

For the latent version of DAPS, LatentDAPS, the only main difference is that we apply an encoder at the beginning and apply a corresponding decoder to recover $\boldsymbol{x}_0$.

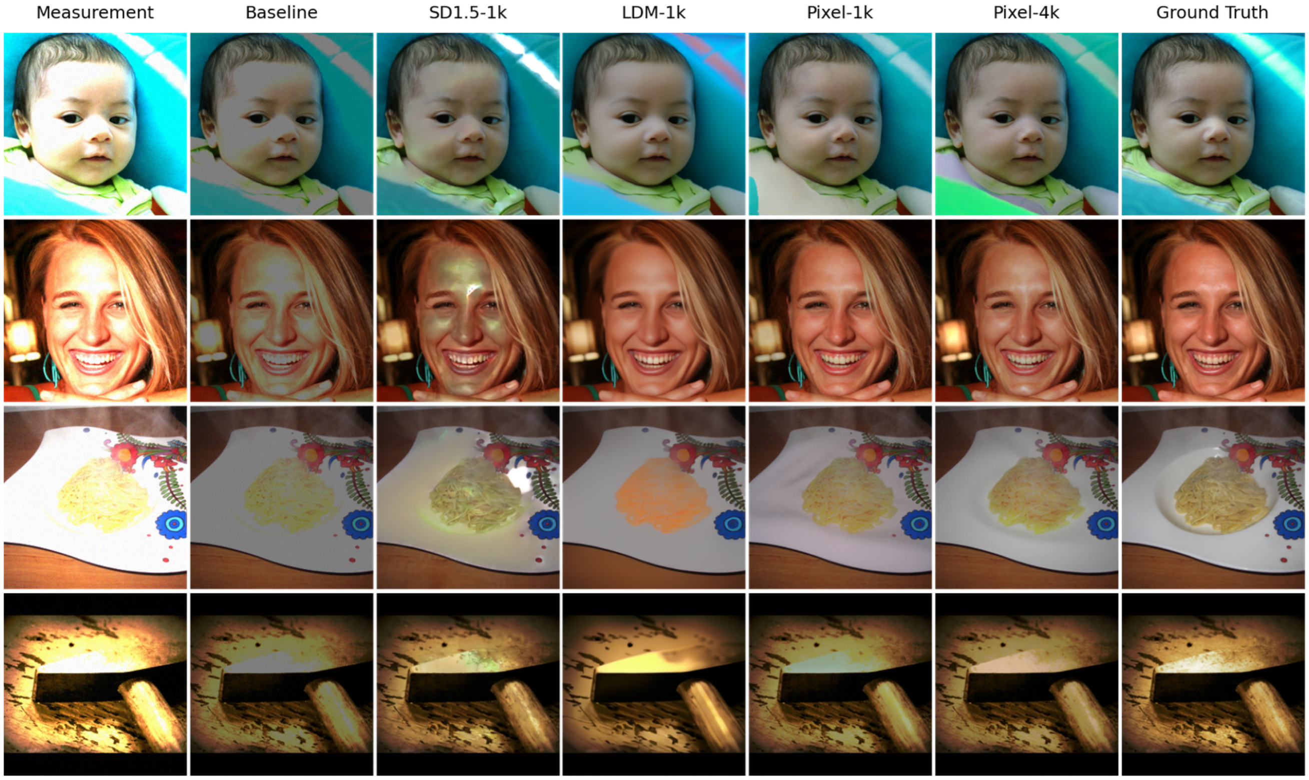


Figure 3. Comparison between DAPS sampling methods and baseline on the overexposed HDR reconstruction task. The first two rows are from FFHQ, and the bottom two rows are from ImageNet.

Code was adapted from the DAPS GitHub repository. We added new forward measurement operators, created a baseline method, and wrote scripts to generate figures and conduct the experiments. The method is written in PyTorch [15].

4. Dataset

Datasets for deep HDR imaging are particularly difficult to obtain, especially those that have paired LDR-HDR data, which is necessary for supervised training. Available datasets differ in exposure levels, size, resolution, and scene diversity. Also, some datasets only have HDR images, while others simulate a camera response function (CRF) to obtain LDR images or contain real-world LDR images with ground-truth HDR images.

On the other hand, in this study we focus on expanding the dynamic range of regular LDR images that are corrupted by a known forward measurement operator. We evaluate the DAPS method on the FFHQ 256×256 [12] and ImageNet 256×256 [5] datasets. We use 10 images from the each of the test datasets for sampling. Note that we do not need the training images, since we are using pre-trained models. For each test image, we apply a forward operator, which decreases the dynamic range of the image, and then apply DAPS to sample a reconstructed image.

5. Results

5.1. Pre-trained models

For the pre-trained diffusion models, we use both pixel-space and latent diffusion models [23]. For pixel-space, we use pre-trained models trained by [3] on the FFHQ dataset and [6] on the ImageNet dataset. For latent, we use the unconditional LDM-VQ4 trained on FFHQ and ImageNet by [17]. In addition, we use Stable Diffusion (SD) v1.5 [17], which is a state-of-the-art text-conditioned latent diffusion model (LDM), but we do not use text conditioning in our experiments, which means it functions as an unconditional model. Unlike the other models, SD v1.5 was not trained on FFHQ or ImageNet.

5.2. Posterior sampling

During sampling/inference, the time step discretization and noise schedule is from EDM [11]. The posterior sampling algorithm follows DAPS for pixel models and Latent-DAPS for latent models [23], where the latent version is similar but performs sampling in the latent space of VAEs.

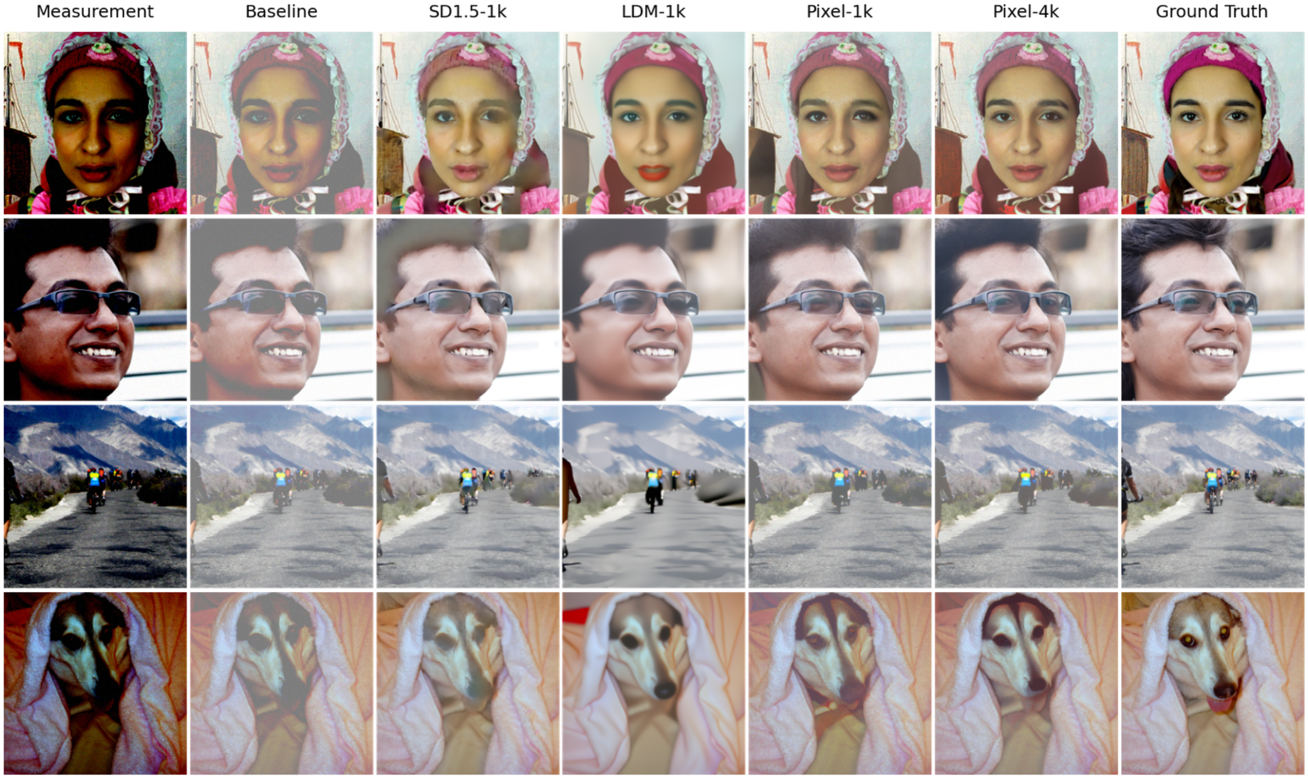


Figure 4. Comparison between DAPS sampling methods and baseline on the underexposed HDR reconstruction task. The first two rows are from FFHQ, and the bottom two rows are from ImageNet.

DAPS uses an Euler ODE solver during the sampling process to compute $\hat{\boldsymbol{x}}_0(\boldsymbol{x}_t)$.

5.3. Forward operators

Our experiments are focused on expanding dynamic range from a low dynamic range image to a higher dynamic ranger one, which we term as HDR reconstruction. We focus on scaling the dynamic range by a factor of 2, but in principle any factor greater than 1 could be experimented with, although higher factors are more difficult to achieve due to more limited information that must be hallucinated. We explore three different ways of framing the expanding dynamic range problem and propose three corresponding simple forward operators to produce simulated measurements, which create three nonlinear inverse problems/tasks for the methods to solve. All measurements are subject to white Gaussian noise with standard deviation $\beta_{\boldsymbol{y}} = 0.05$ for the pixel and latent models and $\beta_{\boldsymbol{y}} = 0.01$ for the Stable Diffusion model [23] to simulate noisy measurements. Note that in this case, the forward operator is known, whereas methods such as BlindDPS [2] are ”blind” to the forward model, which is unknown.

Let α represent the scale factor controlling distortion strength, defaulted to 2. First, consider symmetrical distortion, modeled as

$$\boldsymbol{y} \sim \mathcal{N}\left(\text{clip}\left(\alpha \boldsymbol{x}_0, -1, 1\right), \beta_{\boldsymbol{y}}^2 \boldsymbol{I}\right) \tag{12}$$

where we assume that input image and measurement are normalized to $[-1, 1]$. We similarly define two asymmetrical distortions as follows, with

$$\boldsymbol{y} \sim \mathcal{N}\left(\text{clip}\left(\alpha \cdot \frac{\boldsymbol{x}_0 + 1}{2}, 0, 1\right) \cdot 2 - 1, \beta_{\boldsymbol{y}}^2 \boldsymbol{I}\right) \tag{13}$$

representing an overexposed image and

$$\boldsymbol{y} \sim \mathcal{N}\left(\text{clip}\left(\alpha \cdot \frac{\boldsymbol{x}_0 - 1}{2}, -1, 0\right) \cdot 2 + 1, \beta_{\boldsymbol{y}}^2 \boldsymbol{I}\right) \tag{14}$$

representing an underexposed image.

We can interpret the symmetric case (12) as a poor quality camera that tends to overexpose highlights and underexpose shadows. In this case, assuming $\alpha = 2$, pixels with value in the middle 50% of the linear intensity range are not distorted but rather simply stretched to fit $[-1, 1]$, while values outside that range are clipped to the extreme tail values, resulting in lost information. Similarly, (13) represents an overexposed image with significant highlight clipping where pixels with value in the lower 50% are not distorted, and (14) represents an underexposed image with significant shadow clipping where pixels with value in the upper 50% are not distorted. Practically, the overexposed and underexposed operators can represent realistic camera captures with too long or short exposure times or other improperly calibrated settings like aperture and ISO.

5.4. Baseline

For all experiments, we compare the posterior sampling methods with a different generalized naive baseline inverse operator depending on the forward operator. Each corresponds to attempting to reconstruct all pixels that were not clipped by the forward operators while attempting a universal guess at the clipped pixels. The naive baselines corresponding to symmetric, overexposed, and underexposed, respectively, are

$$\hat{\boldsymbol{x}} = \frac{1}{\alpha}\boldsymbol{y} \tag{15}$$

$$\hat{\boldsymbol{x}} = \frac{1}{\alpha}(\boldsymbol{y} - (\alpha - 1)) \tag{16}$$

$$\hat{\boldsymbol{x}} = \frac{1}{\alpha}(\boldsymbol{y} + (\alpha - 1)) \tag{17}$$

For symmetric, overexposed, and underexposed, the range of pixel intensities of the input images that are not distorted by the application of the forward operator and thus perfectly reconstructed by the inverse operator up to a difference in injected Gaussian noise are $\left[-\frac{1}{\alpha}, \frac{1}{\alpha}\right]$, $\left[-1, \frac{2-\alpha}{\alpha}\right]$, and $\left[\frac{\alpha-2}{\alpha}, 1\right]$. We can reason the range for the symmetric case by noting that clipping is prevented when a pixel z satisfies

$$|\alpha z| \leq 1 \implies |z| \leq \frac{1}{\alpha}. \tag{18}$$

For overexposed, the clipping is prevented when

$$0 \leq \alpha \cdot \frac{z+1}{2} \leq 1 \implies -1 \leq z \leq \frac{2-\alpha}{\alpha}. \tag{19}$$

The reasoning for underexposed is similar. Note that the naive baseline cannot deal with the injected Gaussian measurement noise, so the non-distortion is exact up to this noise.

5.5. Hyperparameters and Setup

We use most of the default hyperparameters from DAPS [23] with some modifications. In particular, we experiment with different numbers of neural function evaluations (NFE), which is the product of the number of Euler ODE solver steps and the number of noise annealing steps, ranging from 50 to 4k. The choice of NFE is a direct tradeoff between cost and quality. We use 100 and 50 Langevin steps per denoising iteration for pixel and latent, respectively.

All sampling was done on Google Compute Engine using one NVIDIA L4 GPU with 24 GB of memory. Given the hardware constraints, we opted for a batch size of 2 throughout, testing on 10 images from each of FFHQ and ImageNet.

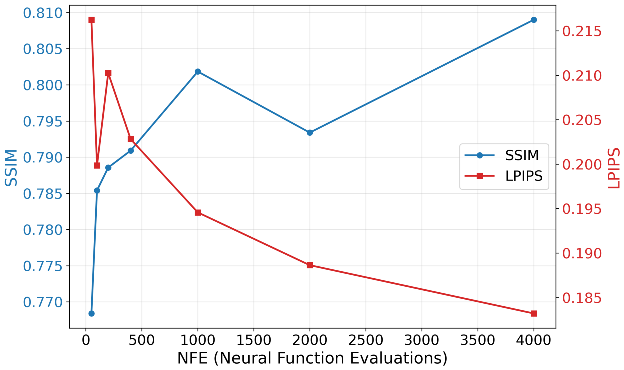


Figure 5. **Ablation study on NFE of DAPS.** The x-axis is the NFE. The evaluation uses 10 FFHQ images using pixel-space DAPS.

5.6. Metrics

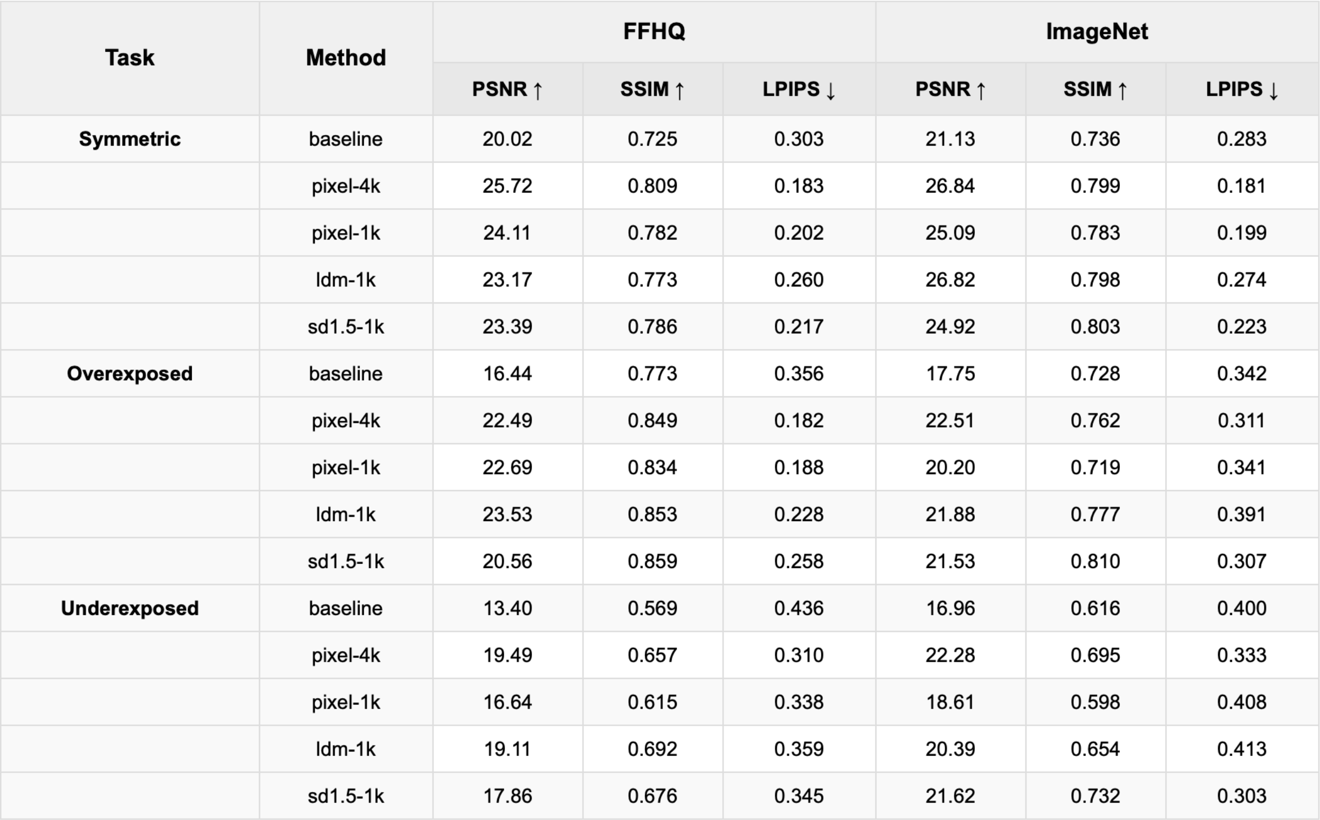
To evaluate the posterior sampling combined with the pre-trained models, we use the 10 images from the test datasets of FFHQ and ImageNet. Quantitative metrics include peak signal-to-noise-ratio (PSNR), structural similarity index measure (SSIM), and Learned Perceptual Image Patch Similarity (LPIPS) score [24]. PSNR measures pixel-wise fidelity (distortion/signal degradation), SSIM measures structural fidelity (visual perception), and LPIPS measures perceptual similarity, forming a comprehensive and complementary view of the reconstruction quality. These metrics are implemented in the piq [13] package with all images normalized to the range $[0, 1]$.

5.7. Main Results

We demonstrate qualitative results for all methods in symmetric, overexposed, and underexposed tasks with selected FFHQ and ImageNet images in 1, 3, 4. A table of average metrics for each configuration of task, method, and dataset is in 6.

5.8. Ablation Study

In 5, we run pixel-space DAPS on varying number of function evaluations (NFE) of the diffusion models ranging from 50 to 4k, with the number of ODE and annealing steps for each NFE setting following [23]. Larger NFE results in longer run time but better quality of reconstruction, as measured by LPIPS and SSIM, but there are diminishing returns as the NFE increases.



Task	Method	FFHQ			ImageNet		
		PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
Symmetric	baseline	20.02	0.725	0.303	21.13	0.736	0.283
	pixel-4k	25.72	0.809	0.183	26.84	0.799	0.181
	pixel-1k	24.11	0.782	0.202	25.09	0.783	0.199
	ldm-1k	23.17	0.773	0.260	26.82	0.798	0.274
	sd1.5-1k	23.39	0.786	0.217	24.92	0.803	0.223
Overexposed	baseline	16.44	0.773	0.356	17.75	0.728	0.342
	pixel-4k	22.49	0.849	0.182	22.51	0.762	0.311
	pixel-1k	22.69	0.834	0.188	20.20	0.719	0.341
	ldm-1k	23.53	0.853	0.228	21.88	0.777	0.391
	sd1.5-1k	20.56	0.859	0.258	21.53	0.810	0.307
Underexposed	baseline	13.40	0.569	0.436	16.96	0.616	0.400
	pixel-4k	19.49	0.657	0.310	22.28	0.695	0.333
	pixel-1k	16.64	0.615	0.338	18.61	0.598	0.408
	ldm-1k	19.11	0.692	0.359	20.39	0.654	0.413
	sd1.5-1k	17.86	0.676	0.345	21.62	0.732	0.303

Figure 6. **Quantitative evaluation on FFHQ and ImageNet on 3 HDR tasks using 5 different methods.** We report the mean metric over 10 samples. For PSNR and SSIM, higher is better; for LPIPS, lower is better. Pixel refers to DAPS in pixel-space with the appropriate pre-trained model, LDM refers to LatentDAPS with pre-trained latent models, and SD1.5 refers to Stable Diffusion v1.5. Baseline is the naive inverse operator. The number after each method indicates the NFE. Pixel-space DAPS with 4k NFE shows the best results for the most tasks. All models tended to struggle more with overexposed and underexposed.

5.9. Discussion

Forward operators All sampling methods struggled more with the asymmetrical dynamic range distortions compared to the symmetrical distortion operator, which suggests that strong clipping on either tail of the intensity distribution is more difficult to reconstruct than an equivalent amount of total clipping on both tails.

Comparison of methods All sampling methods performed better than the naive baseline significantly, however generally the pixel-space DAPS performed better than LatentDAPS, including the LDM and SD v1.5. Because SD v1.5 was not trained directly on the FFHQ or ImageNet and was designed primarily to use text conditioning, we expect it to perform worse. Pixel-space DAPS also sampled signifcantly faster than LatentDAPS, which can be attributed to the latent-space Langevin dynamics being more computationally intensive [23],

6. Conclusion and Future Work

In this study, we investigated the posterior sampling method DAPS on various HDR tasks, including increasing dynamic range by a factor of 2 in three different ways. We chose the forward measurement operators, but in principle any operator can be chosen depending on what distortion we want to model, and DAPS incorporates it in the sampling process. Posterior sampling using pre-trained diffusion models leverages powerful priors and does not require additional training, making it flexible and applicable to many linear and nonlinear inverse problems like HDR reconstruction.

Future work on this topic could include investigating expanding the dynamic range of LDR images to more extreme levels, which might require paired LDR-HDR data. Poisson noise could also be explored in conjuction with Gaussian noise, since Poisson noise is more prevalent in underexposed scenes.

7. Acknowledgements

Code was adapted from the DAPS GitHub repository: https://github.com/zhangbingliang2019/DAPS [23]. J.X. is the sole student working on this project. J.X. is grateful to Suyeon Choi (Computational Imaging Lab, suyeon@stanford.edu) for advising the project, helping with designing the project idea, and proofreading the paper.

References

[1] R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. Duvenaud. Neural Ordinary Differential Equations, 2019.

[2] H. Chung, J. Kim, S. Kim, and J. C. Ye. Parallel Diffusion Models of Operator and Image for Blind Inverse Problems, 2022.

[3] H. Chung, J. Kim, M. T. Mccann, M. L. Klasky, and J. C. Ye. Diffusion Posterior Sampling for General Noisy Inverse Problems, 2024.

[4] G. Daras, H. Chung, C.-H. Lai, Y. Mitsufuji, J. C. Ye, P. Milanfar, A. G. Dimakis, and M. Delbracio. A Survey on Diffusion Models for Inverse Problems, 2024.

[5] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[6] P. Dhariwal and A. Nichol. Diffusion Models Beat GANs on Image Synthesis, 2021.

[7] G. Eilertsen, J. Kronander, G. Denes, R. K. Mantiuk, and J. Unger. HDR image reconstruction from a single exposure using deep CNNs. *ACM Transactions on Graphics*, 36(6):1–15, 2017.

[8] P. Esser, S. Kulal, A. Blattmann, R. Entezari, J. Müller, H. Saini, Y. Levi, D. Lorenz, A. Sauer, F. Boesel, D. Podell, T. Dockhorn, Z. English, K. Lacey, A. Goodwin, Y. Marek, and R. Rombach. Scaling Rectified Flow Transformers for High-Resolution Image Synthesis, 2024.

[9] J. Ho, A. Jain, and P. Abbeel. Denoising Diffusion Probabilistic Models, 2020.

[10] P. Holderrieth and E. Erives. An Introduction to Flow Matching and Diffusion Models. 2025.

[11] T. Karras, M. Aittala, T. Aila, and S. Laine. Elucidating the Design Space of Diffusion-Based Generative Models. *Advances in Neural Information Processing Systems*, 35:26565–26577, 2022.

[12] T. Karras, S. Laine, and T. Aila. A Style-Based Generator Architecture for Generative Adversarial Networks, 2019.

[13] S. Kastryulin, J. Zakirov, D. Prokopenko, and D. V. Dylov. PyTorch Image Quality: Metrics for Image Quality Assessment, 2022.

[14] J. Kim, B. S. Kim, and J. C. Ye. FlowDPS: Flow-Driven Posterior Sampling for Inverse Problems, 2025.

[15] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library, 2019.

[16] W. Peebles and S. Xie. Scalable Diffusion Models with Transformers, 2023.

[17] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-Resolution Image Synthesis with Latent Diffusion Models, 2022.

[18] M. S. Santos, T. I. Ren, and N. K. Kalantari. Single Image HDR Reconstruction Using a CNN with Masked Features and Perceptual Loss. *ACM Transactions on Graphics*, 39(4), 2020.

[19] B. Song, S. M. Kwon, Z. Zhang, X. Hu, Q. Qu, and L. Shen. Solving Inverse Problems with Latent Diffusion Models via Hard Data Consistency, 2024.

[20] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole. Score-Based Generative Modeling through Stochastic Differential Equations, 2021.

[21] C. Wang, Z. Xia, T. Leimkuehler, K. Myszkowski, and X. Zhang. LEDiff: Latent Exposure Diffusion for HDR Generation, 2025.

[22] L. Wang and K.-J. Yoon. Deep Learning for HDR Imaging: State-of-the-Art and Future Trends. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(12):8874–8895, 2022.

[23] B. Zhang, W. Chu, J. Berner, C. Meng, A. Anandkumar, and Y. Song. Improving Diffusion Inverse Problem Solving with Decoupled Noise Annealing, 2024.

[24] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 586–595, 2018.