Exploring Diffusion Models: A Survey in Super-Resolution and In-Painting Applications

Zarifis Stelios
Department of Electrical and Computer Engineering
National Technical University of Athens
Athens, Greece
Email: el20435@mail.ntua.gr

Abstract—This survey explores diffusion models and their concepts and applications in inpainting and super-resolution. Guided by the contributions of Yang, Sohl-Dickstein, Ho, Song, Lugmayr, Zhang, Rombach, Zhao, and Zhu, we delve into the fundamentals of Denoising Diffusion Probabilistic Models (DDPMs) and Score-Based Generative Modeling through Stochastic Differential Equations and explore their applications in super-resolution and inpainting tasks, through substantial models.

Index Terms—diffusion models, generative models, score-based generative models, stochastic differential equations, latent diffusion models, partial diffusion models, super-resolution, inpainting

I. Introduction

DIFFUSION models have emerged as groundbreaking deep generative models, challenging the traditional dominance of generative adversarial networks (GANs) in image synthesis. This survey provides an overview of diffusion models and explores their applications, focusing on super-resolution and image inpainting.

The survey begins with the core principles of diffusion models. It covers a spectrum of methodologies, including generative models, score-based generative models, and stochastic differential equations. The aim is to establish a foundational understanding of diffusion models.

Proceeding with the applications, the survey explores the principles of super-resolution and inpainting. Notable works are examined.

RePaint, a state-of-the-art image inpainting algorithm, utilizes Denoising Diffusion Probabilistic Models with a conditioning strategy. Blending a pretrained unconditional DDPM with a reference image and sampling unmasked regions during the reverse diffusion iterations, it generates high-quality results.

Coherent Image Inpainting, introduced by Zhang et al., contributes to the survey with the algorithm named CoPAINT. This algorithm maintains coherence in generation without violating constraints, providing a more robust approach to computing and drawing samples from the posterior distribution.

The survey also discusses Latent Diffusion Models proposed by Rombach et al., addressing computational challenges in likelihood-based models for high-resolution image synthesis. With a two-phase training strategy with an autoencoder and diffusion models, Latent Diffusion Models offer scalability while preserving perceptual equivalence in a lower-dimensional latent space.

Partial Diffusion Models (PartDiff), as presented by Zhao et al., also tackle computational costs associated with numerous denoising steps in Diffusion Probabilistic Models. The work suggests latent alignment during diffusion, significantly reducing denoising steps without compromising image quality, resulting in efficient image generation.

The Diffusion Plug-and-Play Image Restoration (DiffPIR), as proposed by Zhu et al., investigates the potential of diffusion models in the plug-and-play image restoration framework. This approach aims to generalize pretrained models, enabling them to adapt in various image restoration tasks. DiffPIR efficiently restores images across a broad spectrum of applications, emphasizing the adaptability and versatility of diffusion models.

II. Fundamentals of Diffusion Models

A. Deep Unsupervised Learning using Nonequilibrium Thermodynamics

The core concept of this study [1] draws inspiration from non-equilibrium thermodynamics. The central objective is to methodically and gradually disrupt the inherent structure present in a data distribution by employing an iterative forward diffusion process. Following this process, a model can acquiring insights into a reverse diffusion process designed to systematically reintroduce structure into the data.
Especially in microscopic contexts such as Brownian motion, where positional updates follow small Gaussian distributions over time, the reversibility of diffusion processes is particularly attainable.

This method gradually destroys the data structure using an algorithm that, at each timestep, employs a

Fig. 1: The analogy between a diffusion process and the dispersal of color in water.

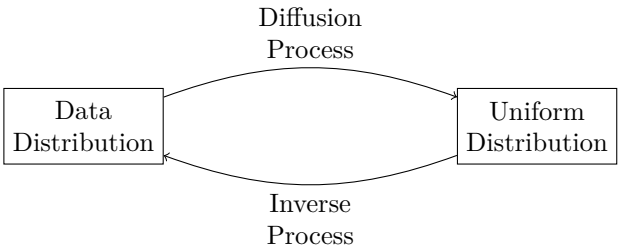

Fig. 2: Illustration of the diffusion and inverse processes in generative modeling.

Neural Network to learn the diffusion process. When the timestep is small enough, both the diffusion and inverse processes can be approximated by Gaussian distributions. The goal is to learn the mean and covariance for each transition kernel, which represents the stochastic process transforming the distribution from one state to the next. The algorithm employs a dual trajectory approach, using forward and reverse diffusion processes, to transform an initial distribution $q(\mathbf{x}^{(0)})$ into the target distribution $\pi(\mathbf{y})$. It is a self-supervised learning algorithm and involves the repeated application of a Markov diffusion kernel $T_\pi(\mathbf{y} \mid \mathbf{y}'; \beta)$. The various components of the algorithm are discussed.

1) Forward Trajectory: The algorithm defines a forward diffusion process transforming $q(\mathbf{x}^{(0)})$ into $\pi(\mathbf{y})$. This involves repeated application of a Markov diffusion kernel $T_\pi(\mathbf{y} \mid \mathbf{y}'; \beta)$. The forward trajectory is given by:

$$q(\mathbf{x}^{(0\cdots T)}) = q(\mathbf{x}^{(0)}) \prod_{t=1}^{T} q(\mathbf{x}^{(t)} \mid \mathbf{x}^{(t-1)})$$

2) Reverse Trajectory: The generative distribution describes the reverse trajectory:

$$p(\mathbf{x}^{(T)}) = \pi(\mathbf{x}^{(T)})$$

$$p(\mathbf{x}^{(0\cdots T)}) = p(\mathbf{x}^{(T)}) \prod_{t=1}^{T} p(\mathbf{x}^{(t-1)} \mid \mathbf{x}^{(t)})$$

3) Model Probability: The generative model's probability assignment to the data is an intractable integral. The solution results from the Annealed Importance Sampling (AIS) and the Jarzynski. It involves evaluating the relative probability of the forward and reverse trajectories.

$$\begin{aligned} p(\mathbf{x}^{(0)}) &= \int d\mathbf{x}^{(1\cdots T)} p(\mathbf{x}^{(0\cdots T)}) = \\ &= \int d\mathbf{x}^{(1\cdots T)} p\left(\mathbf{x}^{(0\cdots T)}\right) \frac{q\left(\mathbf{x}^{(1\cdots T)} \mid \mathbf{x}^{(0)}\right)}{q\left(\mathbf{x}^{(1\cdots T)} \mid \mathbf{x}^{(0)}\right)} \\ &= \int d\mathbf{x}^{(1\cdots T)} q\left(\mathbf{x}^{(1\cdots T)} \mid \mathbf{x}^{(0)}\right) \frac{p\left(\mathbf{x}^{(0\cdots T)}\right)}{q\left(\mathbf{x}^{(1\cdots T)} \mid \mathbf{x}^{(0)}\right)} \\ &= \int d\mathbf{x}^{(1\cdots T)} q\left(\mathbf{x}^{(1\cdots T)} \mid \mathbf{x}^{(0)}\right) \cdot \\ &\quad p\left(\mathbf{x}^{(T)}\right) \prod_{t=1}^{T} \frac{p\left(\mathbf{x}^{(t-1)} \mid \mathbf{x}^{(t)}\right)}{q\left(\mathbf{x}^{(t)} \mid \mathbf{x}^{(t-1)}\right)} \end{aligned}$$

The equality holds when β approaches infinitesimal values. In this limit, forward and reverse distributions over trajectories become identical, simplifying the calculations.

4) Training: Training maximizes the model log likelihood, which has a lower bound provided by Jensen's inequality.

$$\begin{aligned} L &= \int d\mathbf{x}^{(0)} q\left(\mathbf{x}^{(0)}\right) \log p\left(\mathbf{x}^{(0)}\right) \\ &= \int d\mathbf{x}^{(0)} q\left(\mathbf{x}^{(0)}\right) \cdot \log \left[\begin{array}{c} \int d\mathbf{x}^{(1\cdots T)} q\left(\mathbf{x}^{(1\cdots T)} \mid \mathbf{x}^{(0)}\right) \cdot \\ p\left(\mathbf{x}^{(T)}\right) \prod_{t=1}^{T} \frac{p\left(\mathbf{x}^{(t-1)} \mid \mathbf{x}^{(t)}\right)}{q\left(\mathbf{x}^{(t)} \mid \mathbf{x}^{(t-1)}\right)} \end{array} \right] \geq \\ &\geq \int d\mathbf{x}^{(0\cdots T)} q\left(\mathbf{x}^{(0\cdots T)}\right) \log \left[p\left(\mathbf{x}^{(T)}\right) \prod_{t=1}^{T} \frac{p\left(\mathbf{x}^{(t-1)} \mid \mathbf{x}^{(t)}\right)}{q\left(\mathbf{x}^{(t)} \mid \mathbf{x}^{(t-1)}\right)} \right] \end{aligned}$$

$$\begin{aligned} L \geq &- \sum_{t=2}^{T} \int d\mathbf{x}^{(0)} d\mathbf{x}^{(t)} q\left(\mathbf{x}^{(0)}, \mathbf{x}^{(t)}\right) \\ &D_{KL}\left(q\left(\mathbf{x}^{(t-1)} \mid \mathbf{x}^{(t)}, \mathbf{x}^{(0)}\right) || p\left(\mathbf{x}^{(t-1)} \mid \mathbf{x}^{(t)}\right)\right) \end{aligned}$$

$$+ H_q\left(\mathbf{X}^{(T)} \mid \mathbf{X}^{(0)}\right) - H_q\left(\mathbf{X}^{(1)} \mid \mathbf{X}^{(0)}\right) - H_p\left(\mathbf{X}^{(T)}\right).$$

The optimization process aims to maximize this lower bound by adjusting the reverse Markov transitions.

5) Multiplying Distributions: The algorithm efficiently multiplies distributions, facilitated by modified marginal distributions and diffusion steps.

6) Entropy of Reverse Process: The algorithm uses knowledge of the forward process to derive upper and lower bounds on the conditional entropy of each step in the reverse trajectory.

$$\begin{aligned} &H_q(\mathbf{X}^{(t)} \mid \mathbf{X}^{(t-1)}) + H_q(\mathbf{X}^{(t-1)} \mid \mathbf{X}^{(0)}) - H_q(\mathbf{X}^{(t)} \mid \mathbf{X}^{(0)}) \leq \\ &\leq H_q(\mathbf{X}^{(t-1)} \mid \mathbf{X}^{(t)}) \leq H_q(\mathbf{X}^{(t)} \mid \mathbf{X}^{(t-1)}) \end{aligned}$$

B. Denoising Diffusion Probabilistic Models (DDPMs)

Denoising Diffusion Probabilistic Models proposed by Ho et al. [2] are a class of generative models that leverage denoising processes to capture complex data distributions.

1) Notation: Diffusion models are latent variable models of the form $p_\theta(\mathbf{x}_0) := \int p_\theta(\mathbf{x}_{0:T})\, d\mathbf{x}_{1:T}$, where $\mathbf{x}_1, \ldots, \mathbf{x}_T$ are latents of the same dimensionality as the data $\mathbf{x}_0 \sim q(\mathbf{x}_0)$. The joint distribution $p_\theta(\mathbf{x}_{0:T})$ is the reverse process, and it is defined as a Markov chain with learned Gaussian transitions starting at $p(\mathbf{x}_T) = \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I})$:

$$p_\theta(\mathbf{x}_{0:T}) := p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t),$$
$$p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t) := \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t))$$

In Figure 3, we illustrate the forward and reverse diffusion processes.

What distinguishes diffusion models from other latent variable models is that the approximate posterior $q(\mathbf{x}_{1:T} \mid \mathbf{x}_0)$ (forward or diffusion process) is fixed to a Markov chain that gradually adds Gaussian noise to the data according to a variance schedule $\beta_1, \ldots, \beta_T$:

$$q(\mathbf{x}_{1:T} \mid \mathbf{x}_0) := \prod_{t=1}^{T} q(\mathbf{x}_t \mid \mathbf{x}_{t-1}),$$
$$q(\mathbf{x}_t \mid \mathbf{x}_{t-1}) := \mathcal{N}\left(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t \mathbf{I}\right)$$

Training is performed by optimizing the variational bound on negative log likelihood:

$$\mathbb{E}[-\log p_\theta(\mathbf{x}_0)] \leq \mathbb{E}_q\left[-\log \frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T} \mid \mathbf{x}_0)}\right] =$$
$$= \mathbb{E}_q\left[-\log p(\mathbf{x}_T) - \sum_{t \geq 1} \log \frac{p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t)}{q(\mathbf{x}_t \mid \mathbf{x}_{t-1})}\right] =: L$$

The forward process variances β_t can be learned through reparameterization or kept constant as hyperparameters. The choice of Gaussian conditionals in $p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t)$ is crucial for the expressive capacity of the reverse process (the ability of the model to represent a wide range of complex relationships and patterns within the data). This choice is significant, as both processes share the same functional form when β_t is small.

An interesting feature of the forward process is its capability to sample $\mathbf{x}_t$ at any timestep t in a closed form. Using the notations $\alpha_t := 1 - \beta_t$ and $\bar{\alpha}_t := \prod_{s=1}^{t} \alpha_s$, the closed-form expression is given by:

$$q(\mathbf{x}_t \mid \mathbf{x}_0) = \mathcal{N}\left(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I}\right)$$

Efficient training is therefore possible by optimizing random terms of L with stochastic gradient descent.

2) Diffusion models and denoising autoencoders: In the upcoming 2 sections, we explore the forward and reverse processes. Despite diffusion models initially appearing as a constrained class of latent variable models, their implementation offers substantial flexibility. Critical considerations are the choice of fixed forward process variances β_t, the model architecture and Gaussian distribution parameterization of the reverse process. The modeling decisions in the paper are guided by insights into trade-offs, justified by simplicity and supported by empirical results. To inform these decisions, a novel connection between diffusion models and denoising score matching is established.

3) Forward process and L_T: In the implementation, forward process variances β_t are all fixed constants, making the approximate posterior q parameter-free and therefore L_T constant, so it can be ignored during training.

4) Reverse process and $L_{1:T-1}$: The choices in $p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t))$ for $1 < t \leq T$ are explained. Ho et al. set $\boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t) = \sigma_t^2 \mathbf{I}$ to untrained time-dependent constants. Experiments showed that $\sigma_t^2 = \beta_t$ and $\sigma_t^2 = \tilde{\beta}_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t$ had similar results. To represent the mean $\boldsymbol{\mu}_\theta(\mathrm{x}_t, t)$, a parameterization is motivated by the following analysis of L_t. With $p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t) = \mathcal{N}\left(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \sigma_t^2 \mathbf{I}\right)$, L_{t-1} is writen as:

$$L_{t-1} = \mathbb{E}_q\left[\frac{1}{2\sigma_t^2}\left\|\tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0) - \boldsymbol{\mu}_\theta(\mathbf{x}_t, t)\right\|^2\right] + C$$

where C is a constant that does not depend on θ. It is obvious that the most appropriate choice of $\boldsymbol{\mu}_\theta$ is a model that predicts the forward process posterior mean $\tilde{\boldsymbol{\mu}}_t$. By the reparameterization $\mathbf{x}_t(\mathbf{x}_0, \boldsymbol{\epsilon}) = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ we can express $L_{t-1} - C$ as:

$$\begin{aligned}
&L_{t-1} - C = \\
&= \mathbb{E}_{\mathbf{x}_0, \boldsymbol{\epsilon}}\left[\frac{1}{2\sigma_t^2}\left\|\tilde{\boldsymbol{\mu}}_t\left(\mathbf{x}_t(\mathbf{x}_0, \boldsymbol{\epsilon}), \frac{1}{\sqrt{\bar{\alpha}_t}}\left(\mathbf{x}_t(\mathbf{x}_0, \boldsymbol{\epsilon}) - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}\right)\right)\right.\right. \\
&\left.\left. - \boldsymbol{\mu}_\theta(\mathbf{x}_t(\mathbf{x}_0, \boldsymbol{\epsilon}), t)\right\|^2\right] \\
&= \mathbb{E}_{\mathbf{x}_0, \boldsymbol{\epsilon}}\left[\frac{1}{2\sigma_t^2}\left\|\frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t(\mathbf{x}_0, \boldsymbol{\epsilon}) - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}\right)\right.\right. \\
&\left.\left. - \boldsymbol{\mu}_\theta(\mathbf{x}_t(\mathbf{x}_0, \boldsymbol{\epsilon}), t)\right\|^2\right]
\end{aligned}$$

From this analysis we can see that $\boldsymbol{\mu}_\theta$ should predict $\frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}\right)$, given $\mathbf{x}_t$. $\mathbf{x}_t$ is the input, so the following parameterization can be chosen:

$$\boldsymbol{\mu}_\theta(\mathbf{x}_t, t) = \tilde{\boldsymbol{\mu}}_t\left(\mathbf{x}_t, \frac{1}{\sqrt{\bar{\alpha}_t}}\left(\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t)\right)\right) =$$
$$= \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\right)$$

Here, ϵ_θ predicts ϵ, from $\mathbf{x}_t$

The sampling $\mathbf{x}_{t-1} \sim p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t)$ utilizes the reparameterization trick.

a) Reparameterization Trick: The reparameterization trick is crucial in diffusion models as it enables the training of generative models using stochastic gradient descent. It is applied to handle the sampling of noise variables. Let z be a latent variable, and ϵ be a noise

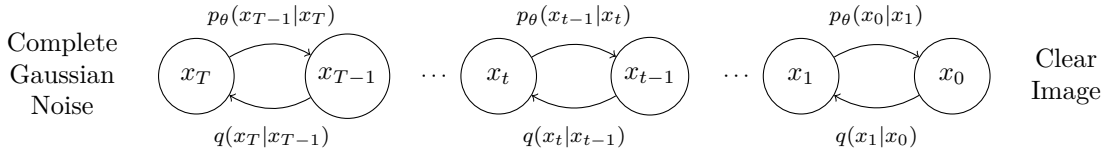

Fig. 3: Forward and Reverse Diffusion Process Markov Chain.

variable. The reparameterization trick expresses the sampling process as follows:

$$z = \mu + \sigma \cdot \epsilon$$

where μ and σ are the mean and standard deviation parameters, respectively. This reparameterization allows for the backpropagation of gradients as z is now deterministic with respect to μ, σ.
So the sampling of $\mathbf{x}_{t-1}$ is done as $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\right) + \sigma_t \mathbf{z}, \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and as a result, there is the following simplification:

$$L_{t-1} - C =$$
$$\mathbb{E}_{\mathbf{x}_0,\boldsymbol{\epsilon}}\left[\frac{\beta_t^2}{2\sigma_t^2\alpha_t(1-\bar{\alpha}_t)}\left\|\epsilon - \epsilon_\theta\left(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t\right)\right\|^2\right]$$

5) Data scaling and reverse process decoder: In the context of data scaling, Ho et al. adopt a standardized representation for image data, scaling integers from $\{0, 1, \ldots, 255\}$ linearly to $[-1, 1]$. This ensures uniform inputs for the neural network's reverse process, originating from the standard normal prior $p(\mathbf{x}_T)$. For computing discrete log likelihoods, the authors use an independent discrete decoder based on a Gaussian distribution $\mathcal{N}\left(\mathbf{x}_0; \boldsymbol{\mu}_\theta\left(\mathbf{x}_1, 1\right), \sigma_1^2\mathbf{I}\right)$.

C. Score-Based Generative Modeling Through Stochastic Differential Equations

Song et al. [3] introduced a novel approach to generative modeling using Stochastic Differential Equations (SDEs). Their key idea involves perturbing data with a continuous SDE, enabling the generation of samples through both forward and reverse diffusion processes, as seen in Figure 4. The SDE is defined as:

$$d\vec{x} = \underbrace{\vec{f}(\vec{x}, t)dt}_{\text{Deterministic Drift}} + \underbrace{g(t)d\vec{w}}_{\text{Stochastic Diffusion}}, \quad \vec{w} : \text{Brownian Motion}$$

And the reverse SDE is given by:

$$d\vec{x} = \left[\vec{f}(\vec{x}, t) - g^2(t)\underbrace{\nabla_{\vec{x}} \log p_t(\vec{x})}_{\text{Score function of distribution } p_t} dt\right] + g(t)d\vec{w}, \quad \vec{w} : \text{Brownian Motion}$$

Song et al. propose techniques for sampling from the reverse SDE, estimating score functions from data, and sampling with numerical SDE solvers. Additionally, they explore the conversion of SDEs to Ordinary Differential Equations (ODEs) for more efficient sampling. The controllable generation aspect involves using a control signal $\vec{y}$ to constrain the generation process.

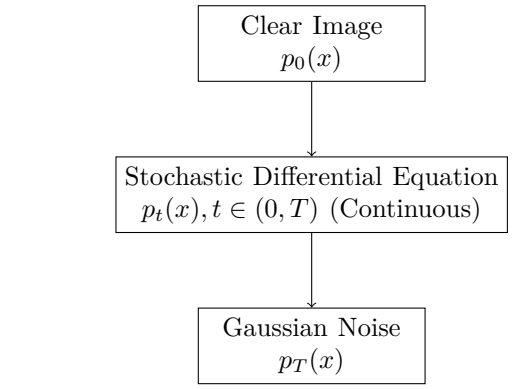

Fig. 4: Illustration of the generative process using a Stochastic Differential Equation (SDE).

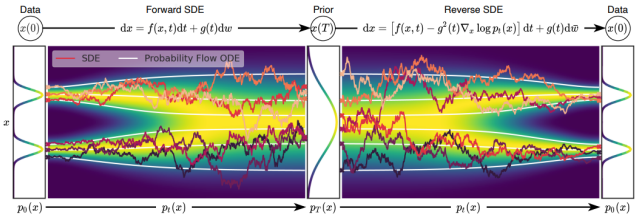

Fig. 5: Stochastic Differential Equation (SDE) Converted to Ordinary Differential Equation (ODE)

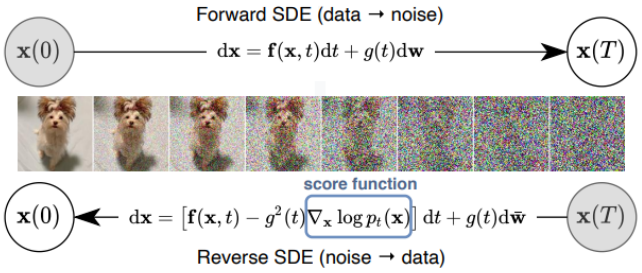

Fig. 6: Visualization of Diffusion Process as Stochastic Differential Equation

III. Image Inpainting Techniques and Applications

Image inpainting is the art of crafting a natural and comprehensive image from a reference that is only partially disclosed. In this challenge, diffusion models emerge as invaluable tools. With each step in the diffusion process, the reference image blends with the denoised image, symbolized by $\hat{X}_0^t$. This fusion of the reference image with the denoised counterpart acts as a guiding force, steering the diffusion process toward the creation of an image that retains the harmony between the masked and revealed areas. In this section, we will delve into two important works that have had a major impact on the field of image inpainting.

A. RePaint: Inpainting using Denoising Diffusion Probabilistic Models

In this work, Lugmayr et al. [4] propose RePaint, an innovative inpainting method based on Denoising Diffusion Probabilistic Models (DDPMs). Without relying on specific mask distributions during training, the method shows better performance across various types of masks. Utilizing a pretrained unconditional DDPM as a generative prior, the authors condition the generation process by sampling unmasked regions during reverse diffusion iterations. This approach enables RePaint to produce high-quality and diverse output images for any inpainting scenario. The authors introduce an improved denoising strategy that resamples iterations, resulting in semantically meaningful images. Through experiments on CelebA-HQ and ImageNet, the proposed model outperforms other state-of-the-art inpainting methods in terms of generalization and semantic quality.

Early Image Inpainting approaches (including low-level cues and neighbor-based methods) are discussed.

a) Deterministic Image Inpainting: involves GAN-based methods following an encoder-decoder architecture, adversarial training, and tailored losses for photo-realistic results.

b) Diverse Image Inpainting: addresses the deterministic nature of GAN-based methods, proposing VAE-based networks and autoregressive approaches for diversity and handling irregular masks.

c) Usage of Image Prior: explores StyleGAN [10] and non-trained generator network structures as priors, while the proposed method leverages a pretrained DDPM for generic image inpainting without specific training for the task.

d) Image Conditional Diffusion Models: , including early diffusion models and score-based formulations, are discussed, emphasizing the proposed method's comprehensive comparisons with top competing methods.

Ideas like guided synthesis, image-to-image translation, and concurrent works are also touched, highlighting the innovative approach of utilizing an unconditional DDPM for effortless generalization in free-form inpainting.

1) RePaint Method: In this section, the authors present the RePaint approach for image inpainting using an unconditional Denoising Diffusion Probabilistic Model (DDPM). Two primary components are discussed: conditioning the reverse diffusion process on known regions and a resampling strategy to enhance the process.

a) Conditioning on the Known Region: The inpainting objective involves predicting missing pixels in an image using a mask region. The authors leverage a trained unconditional DDPM and denote the ground truth image as x, unknown pixels as $m \odot x$, and known pixels as $(1-m) \odot x$. By altering the known regions during the reverse diffusion process, the authors give a formulation for one reverse step:

$$x_{t-1}^{\text{known}} \sim \mathcal{N}\left(\sqrt{\bar{\alpha}_t} x_0, (1-\bar{\alpha}_t)\mathbf{I}\right)$$
$$x_{t-1}^{\text{unknown}} \sim \mathcal{N}\left(\mu_\theta(x_t, t), \Sigma_\theta(x_t, t)\right)$$
$$x_{t-1} = m \odot x_{t-1}^{\text{known}} + (1-m) \odot x_{t-1}^{\text{unknown}}$$

Here, x_{t-1}^{known} is sampled using the known pixels, and x_{t-1}^{unknown} is sampled from the model, combining to form the new sample x_{t-1}, as illustrated in Figure 7.

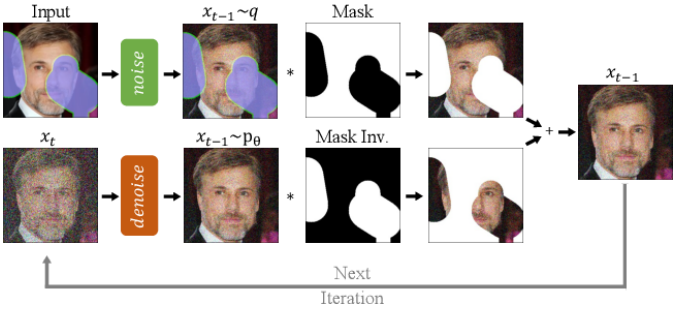

Fig. 7: RePaint adapts the standard denoising process to incorporate the content of the given image. At each step, the known region (top) is sampled from the input, while the inpainted section is sampled from the output of the Denoising Diffusion Probabilistic Model (DDPM) (bottom).

b) Resampling: It was observed that content matching the known regions lacked semantic correctness. To address this, a resampling strategy is introduced. The approach diffuses the output x_{t-1} back to x_t by sampling from a normal distribution:

$$x_t \sim \mathcal{N}\left(\sqrt{1-\beta_t} x_{t-1}, \beta_t \mathbf{I}\right)$$

This resampling strategy, with a defined jump length j, enhances harmonization and incorporates semantic information over the denoising process, as shown in Figure 8.

2) Experiments: Extensive experiments for face and generic inpainting are conducted, comparing RePaint to state-of-the-art solutions. Algorithm 1 outlines the inpainting process using the RePaint approach, incorporating conditioning on known regions and the

Fig. 8: More resampling steps in the RePaint method improve image harmony.

resampling strategy as described.

This novel approach utilizes the principles of DDPMs for effective inpainting, emphasizing its effective conditioning strategy. Comparison between "RePaint" and the other state-of-the-art models is presented in Table 1.

Algorithm 1: Inpainting using RePaint Approach

$x_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$;
for $t = T, \ldots, 1$ do
  for $u = 1, \ldots, U$ do
    if $t > 1$ then
      $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$;
      $x_{t-1}^{\text{known}} = \sqrt{\bar{\alpha}_t} x_0 + (1 - \bar{\alpha}_t)\epsilon$;
    end
    if $t > 1$ then
      $z \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$;
      $x_{t-1}^{\text{unknown}} = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon_\theta(x_t, t) \right) + \sigma_t z$;
    end
    $x_{t-1} = m \odot x_{t-1}^{\text{known}} + (1 - m) \odot x_{t-1}^{\text{unknown}}$;
    if $u < U$ and $t > 1$ then
      $x_t \sim \mathcal{N}(\sqrt{1 - \beta_{t-1}} x_{t-1}, \beta_{t-1}\mathbf{I})$;
    end
  end
end
return x_0;

B. Towards Coherent Image Inpainting Using Denoising Diffusion Implicit Models

As discussed in the previous section, diffusion models for inpainting applications condition the diffusion process by adding the reference image at each timestep $\hat{\mathrm{X}}_0^t$. Still, incoherent results may arise, such as inconsistent hair color for human images. This work tries to address this with an additional condition that enforces a perfect match between the resulting image and the revealed part of the reference: $r(\tilde{\mathrm{X}}_0) = s_0$. Optimization involves maximizing the posterior, solved using gradient descent:

$$\begin{aligned}\log p_\theta(\tilde{\mathrm{X}}_T \mid \mathcal{C}) &\approx -\frac{1}{2}\|\tilde{\mathrm{X}}_T\|_2^2 - \frac{1}{2\xi_T^2}\|s_0 - r(\tilde{\mathrm{X}}_0)\|_2^2 + C \\ &\approx -\frac{1}{2}\|\tilde{\mathrm{X}}_T\|_2^2 - \frac{1}{2\xi_T^2}\|s_0 - r\left(g_\theta(\tilde{\mathrm{X}}_T)\right)\|_2^2 + C\end{aligned}$$

The computational challenge of obtaining $r(\tilde{\mathrm{X}}_0)$ based on $\tilde{\mathrm{X}}_0$ is addressed by using a one-step approximation: $\tilde{\mathrm{X}}_0 \approx \hat{\mathrm{X}}_0^{(T)} = f_\theta(\tilde{\mathrm{X}}_T)$, allowing gradient descent to eventually solve $\tilde{\mathrm{X}}_T$.

CoPaint is introduced to ensure coherence without violating constraints, utilizing a Bayesian framework to systematically modify complete images. Approximating the posterior for intermediate images minimizes errors in inpainting constraints, gradually reducing approximation errors during the denoising process. Other models used for image inpainting address degraded image restoration using structures like auto-encoders, VAEs, GANs, or autoregressive transformers. Supervised diffusion inpainting methods, including PALETTE [11], GLIDE [12], LATENT DIFFUSION [6], CCDF [13], and a "predict-and-refine" model, require degradation-specific training. Unsupervised diffusion inpainting methods like BLENDED-DIFFUSION [14], REPAINT [4], RESAMPLING [15], and DPS [16] address visual inconsistencies and irreducible approximation errors. CoPAINT [5] contributes to unsupervised diffusion inpainting without modifying pre-trained models, introducing resampling and Bayesian frameworks for enhanced inpainting coherence.

1) Notation: We provide an overview of the diffusion model frameworks and notations employed by the paper. $\boldsymbol{X}_0$ represents a random vector of natural images, and DDIMs [17] aim to recover the distribution of $\boldsymbol{X}_0$ through progressively corrupted intermediate variables $\boldsymbol{X}_{1:T}$.

The forward diffusion process follows the denoising diffusion probabilistic models (DDPMs), involving a Markov process with Gaussian noise additions:

$$\begin{aligned}q\left(\boldsymbol{X}_{1:T} \mid \boldsymbol{X}_0\right) &= \prod_{t=1}^{T} q\left(\boldsymbol{X}_t \mid \boldsymbol{X}_{t-1}\right) \\ q\left(\boldsymbol{X}_t \mid \boldsymbol{X}_{t-1}\right) &= \mathcal{N}\left(\boldsymbol{X}_t; \sqrt{\alpha_t}\boldsymbol{X}_{t-1}, \beta_t \boldsymbol{I}\right)\end{aligned}$$

For the reverse diffusion process, DDIMs introduce an inference distribution q_σ with a matching conditional distribution for each intermediate variable:

$$\begin{aligned}q_\sigma\left(\boldsymbol{X}_{1:T} \mid \boldsymbol{X}_0\right) &= q_\sigma\left(\boldsymbol{X}_T \mid \boldsymbol{X}_0\right) \prod_{t=T}^{2} q_\sigma\left(\boldsymbol{X}_{t-1} \mid \boldsymbol{X}_t, \boldsymbol{X}_0\right) \\ q_\sigma\left(\boldsymbol{X}_T \mid \boldsymbol{X}_0\right) &= \mathcal{N}\left(\boldsymbol{X}_T; \sqrt{\bar{\alpha}_T}\boldsymbol{X}_0, \left(1 - \bar{\alpha}_T\right)\boldsymbol{I}\right) \\ q_\sigma\left(\boldsymbol{X}_{t-1} \mid \boldsymbol{X}_t, \boldsymbol{X}_0\right) &= \mathcal{N}\left(\boldsymbol{X}_{t-1}; \boldsymbol{\mu}_t, \sigma_t^2 \boldsymbol{I}\right)\end{aligned}$$

Where $\bar{\alpha} = \prod_{i=1}^{t} \alpha_i$ and σ_t^2 is a free hyperparameter, and

$$\boldsymbol{\mu}_t = \sqrt{\bar{\alpha}_{t-1}}\boldsymbol{X}_0 + \sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2}\frac{\boldsymbol{X}_t - \sqrt{\bar{\alpha}_t}\boldsymbol{X}_0}{\sqrt{1-\bar{\alpha}_t}}$$

The denoising process is derived from q_σ by replacing $\boldsymbol{X}_0$ with an estimated value of $\boldsymbol{X}_0$:

$$p_\theta\left(\boldsymbol{X}_T\right) = \mathcal{N}\left(\boldsymbol{X}_T; \mathbf{0}, \boldsymbol{I}\right)$$
$$p_\theta\left(\boldsymbol{X}_{t-1} \mid \boldsymbol{X}_t\right) = q_\sigma\left(\boldsymbol{X}_{t-1} \mid \boldsymbol{X}_t, \hat{\boldsymbol{X}}_0^{(t)}\right)$$

where $\hat{\boldsymbol{X}}_0^{(t)} = \boldsymbol{f}_\theta^{(t)}\left(\boldsymbol{X}_t\right)$ is produced by a (reparameterized) neural network predicting $\boldsymbol{X}_0$ from $\boldsymbol{X}_t$ by minimizing the mean squared error.

2) The CoPaint Algorithm: The objective of image inpainting is to create a complete and natural image based on a partially disclosed image. The generated image should precisely match the provided image in the areas that are revealed. Formally, let $\boldsymbol{r}(\cdot)$ be an operator that extracts the revealed subset of input dimensions, and s_0 represent the disclosed part of the reference image. The image inpainting task can be expressed as achieving a natural image under the constraint:

$$\mathcal{C} : \boldsymbol{r}\left(\tilde{\boldsymbol{X}}_0\right) = \boldsymbol{s}_0$$

a) A Prototype Approach: Considering a simple DDIM, where $\sigma_t = 0$ for all t, the inpainting requirement on $\tilde{\boldsymbol{X}}_0$ can be equivalently applied to $\tilde{\boldsymbol{X}}_T$, simplifying the image inpainting task to finding a suitable $\tilde{\boldsymbol{X}}_T$ guided by the posterior distribution:

$$p_\theta\left(\tilde{\boldsymbol{X}}_T \mid \mathcal{C}\right) \propto p_\theta\left(\tilde{\boldsymbol{X}}_T\right) \cdot p_\theta\left(\boldsymbol{r}\left(\tilde{\boldsymbol{X}}_0\right) = \boldsymbol{s}_0 \mid \tilde{\boldsymbol{X}}_T\right)$$

The logarithmic form of the posterior distribution:

$$\begin{aligned}\log p_\theta\left(\tilde{\boldsymbol{X}}_T \mid \mathcal{C}\right) \approx & -\frac{1}{2}\left\|\tilde{\boldsymbol{X}}_T\right\|_2^2 \\ & -\frac{1}{2\xi_T^2}\left\|\boldsymbol{s}_0 - \boldsymbol{r}\left(\boldsymbol{f}_\theta^{(T)}\left(\tilde{\boldsymbol{X}}_T\right)\right)\right\|_2^2 + C\end{aligned}$$

Hence, we have formulated a practical method to effectively determine $\tilde{X}_T$.

b) One-Step Approximation: The prototype approach encounters computational impracticality. To address this, a one-step generation is introduced: $\boldsymbol{f}_\theta^{(T)}\left(\tilde{\boldsymbol{X}}_T\right)$, providing a swift approximation of the final generation. This leads to an approximated conditional distribution: $p'_\theta\left(\boldsymbol{r}\left(\tilde{\boldsymbol{X}}_0\right) \mid \tilde{\boldsymbol{X}}_T\right)$, yielding the approximate posterior:

$$\begin{aligned}&\log p'_\theta\left(\tilde{\boldsymbol{X}}_T \mid \mathcal{C}\right) \\ =& -\frac{1}{2}\left\|\tilde{\boldsymbol{X}}_T\right\|_2^2 - \frac{1}{2\xi_T^2}\left\|\boldsymbol{s}_0 - \boldsymbol{r}\left(\boldsymbol{f}_\theta^{(T)}\left(\tilde{\boldsymbol{X}}_T\right)\right)\right\|_2^2 + C'\end{aligned}$$

To minimize the approximation gap, $\xi_T'^2$ should be set to:

$$\xi_T'^2 = \frac{1}{N}\mathbb{E}_{p_\theta}\left[\left\|\boldsymbol{r}\left(\boldsymbol{f}_\theta^{(T)}\left(\tilde{\boldsymbol{X}}_T\right)\right) - \boldsymbol{r}\left(\tilde{\boldsymbol{X}}_0\right)\right\|_2^2\right]$$

c) Denoising Successive Correction: To further enforce the inpainting constraint, the authors reintroduce non-deterministic elements into the DDIM procedure, correcting the approximation error across all intermediate variables through optimization. The proposed DDIM procedure samples $\tilde{\boldsymbol{X}}_{0:T}$ from the approximate posterior:

$$p'_\theta\left(\tilde{\boldsymbol{X}}_{0:T} \mid \mathcal{C}\right) = p'_\theta\left(\tilde{\boldsymbol{X}}_T \mid \mathcal{C}\right)\prod_{t=1}^{T} p'_\theta\left(\tilde{\boldsymbol{X}}_{t-1} \mid \tilde{\boldsymbol{X}}_t, \mathcal{C}\right)$$

For the computation of $p'_\theta\left(\tilde{\boldsymbol{X}}_{t-1} \mid \tilde{\boldsymbol{X}}_t, \mathcal{C}\right)$, Gaussian approximated distributions are used. The approximate posterior is now formulated as:

$$\begin{aligned}&\log p'_\theta\left(\tilde{\boldsymbol{X}}_{t-1} \mid \tilde{\boldsymbol{X}}_t, \mathcal{C}\right) \\ =& -\frac{1}{2\sigma_t^2}\left\|\tilde{\boldsymbol{X}}_{t-1} - \tilde{\boldsymbol{\mu}}_t\right\|_2^2 \\ & -\frac{1}{2\xi_{t-1}'^2}\left\|\boldsymbol{s}_0 - \boldsymbol{r}\left(\boldsymbol{f}_\theta^{(t-1)}\left(\tilde{\boldsymbol{X}}_{t-1}\right)\right)\right\|_2^2 + C'\end{aligned}$$

To obtain the final inpainting result, a greedy optimization procedure is employed to find samples of $\tilde{\boldsymbol{X}}_{0:T}$ that maximize $p'_\theta\left(\tilde{\boldsymbol{X}}_{0:T} \mid \mathcal{C}\right)$.

d) Additional Algorithmic Designs: While the algorithm effectively mitigates the one-step approximation error during the final denoising step, it acknowledges that errors in the early denoising steps may still impact the overall quality of generation. Therefore, the algorithm incorporates optional designs to further minimize the approximation error.

- Multi-Step Approximation: Addresses significant approximation errors in the early denoising steps by replacing the one-step approximation with a multi-step approach. In this method, the estimation of $\tilde{\boldsymbol{X}}_0$ involves undergoing multiple deterministic denoising steps at selected time intervals.
- Time Travel: Enhances the internal consistency of intermediate examples by periodically revisiting previous denoising steps. This is achieved by intentionally corrupting intermediate images. Algorithm 2 describes the Time Travel idea.

Figure 9 presents a visual comparison of the results obtained by the CoPaint algorithm alongside other inpainting methods, demonstrating the superior performance of CoPaint in generating high-quality inpainted images. Numeric comparison between CoPaint and the other state-of-the-art models is presented in Table 2.

IV. Super Resolution Methods and Their Applications

The upcoming sections will present three noteworthy contributions in the field of high-resolution image synthesis and restoration. Each work addresses specific challenges in the domain and proposes innovative solutions.

Recent advancements in high-resolution image synthesis

Algorithm 2: CoPAINT-TT Inpainting Algorithm

```
Initialize X̃_T ~ N(0, I);
t ← T, k ← K;
while t ≠ 0 do
    Optimize X̃_t using Equation 10-14 with G-step
     gradient descent and learning rate η_t;
    Generate X̃_{t−1} using Equation 4;
    t ← t − 1;
    if t mod τ = 0 and t ≤ T − τ then
        if k > 0 then
            // Time travel;
            Generate X̃_{t+τ} ~ q(X̃_{t+τ} | X̃_t);
            t ← t + τ − 1, k ← k − 1;
        end
        else
            k ← K;
        end
    end
end
return X̃_0;
```

(a) Input (b) BLENDED (c) DDRM (d) COPAINT

Fig. 9: CoPaint algorithm results comparison with other Inpainting models.

and restoration have addressed challenges in energy-intensive AutoRegressive Transformers for subtle image details, introducing solutions like Latent Diffusion Models. For super-resolution tasks, Partial Diffusion Models reduce computational complexity by using low-resolution images as surrogates, demonstrating significant denoising step reductions without sacrificing quality. In the realm of plug-and-play Image Restoration, DiffPIR explores the potential of diffusion models as generative denoiser priors, efficiently handling diverse degradation models and showcasing superior image restoration quality across various tasks. These innovations collectively signify strides toward more efficient and effective solutions in high-resolution image synthesis and restoration. In the subsequent sections, we will deepen into the specifics of each work, discussing methodologies, contributions, and experimental results.

A. High-Resolution Image Synthesis with Latent Diffusion Models

High-resolution image synthesis faces challenges with AutoRegressive (AR) Transformers which consume substantial energy for rendering subtle image details, due to their complex architecture and their staggering number of parameters (sometimes billions). The proposed solution, Latent Diffusion Models (LDMs), aims to mitigate these challenges by employing a two-phase training strategy.

In the first phase, an autoencoder is trained to learn an efficient, lower-dimensional latent space preserving perceptual equivalence to the original data. This addresses the computational complexity associated with likelihood-based models. Subsequently, in the second phase, Diffusion Models (DMs) are trained in this learned latent space, enhancing scalability without compromising synthesis quality.

In this task, various generative models for image synthesis have been explored. Generative Adversarial Networks (GANs) excel in perceptual quality and Likelihood-based methods, such as Variational Autoencoders (VAE) and flow-based models, focus on efficient density estimation. Autoregressive Models (ARM) achieve strong density estimation but are limited to low-resolution images due to computational demands. Latent Diffusion Models (LDMs) operate in a compressed latent space, reducing computational complexity in training.

The authors also include Two-Stage Image Synthesis approaches in their work. VQ-VAEs [18] and VQGANs utilize autoregressive models and adversarial objectives in the first stage, respectively. However, challenges arise with high compression rates in ARM training. LDMs address these tradeoffs by scaling more gently to higher-dimensional latent spaces, optimizing the mediation between a powerful first stage and high-fidelity reconstructions.

1) Method: To address the computational demands of training diffusion models for high-resolution image synthesis, a method that separates the compressive and generative learning phases is proposed. This involves using an autoencoding model for perceptual compression, which learns a low-dimensional latent space perceptually equivalent to the image space but with reduced computational complexity.

a) Perceptual Image Compression: The perceptual compression model is an autoencoder trained with a perceptual loss and a patch-based adversarial objective (Perceptual loss functions are designed to capture perceptual differences between images, such as content and style discrepancies, which are not always evident at the pixel level). Given an image $x \in \mathbb{R}^{H\times W\times 3}$ in RGB space, the encoder $\mathcal{E}$ produces a latent representation $z = \mathcal{E}(x)$, and the decoder $\mathcal{D}$ reconstructs the image as $\tilde{x} = \mathcal{D}(z) = \mathcal{D}(\mathcal{E}(x))$, where $z \in \mathbb{R}^{h\times w\times c}$. The encoder then downsamples the image by a factor $f = H/h = W/w$. Two regularization variants, KL-reg. and VQ-reg., are experimented with to avoid high-variance latent spaces.

b) Latent Diffusion Models: The effectiveness of Diffusion Models relies on a reweighted variant of the variational lower bound. Perceptual compression models, denoted as $\mathcal{E}$ and $\mathcal{D}$, generate an efficient, low-dimensional latent space that aligns with likelihood-based generative

models. The objective is:

$$L_{DM} = \mathbb{E}_{x,\epsilon\sim\mathcal{N}(0,1),t}\left[\|\epsilon - \epsilon_\theta\left(x_t, t\right)\|_2^2\right]$$

c) Conditioning Mechanisms: A general-purpose conditioning mechanism based on cross-attention is proposed, enhancing the flexibility of DMs as conditional image generators. The cross-attention mechanism allows DMs to be conditioned by inputs such as text or bounding boxes. Conditioning is achieved through a joint optimization process involving the encoder τ_θ and the denoising autoencoder ϵ_θ. This mechanism is effective for various input modalities and provides flexibility for tasks like text-to-image synthesis.

$$L_{LDM} = \mathbb{E}_{\mathcal{E}(x),\epsilon\sim\mathcal{N}(0,1),t}\left[\|\epsilon - \epsilon_\theta\left(z_t, t\right)\|_2^2\right]$$

The general architecture of a Latent Diffusion Model is shown in Figure 10.

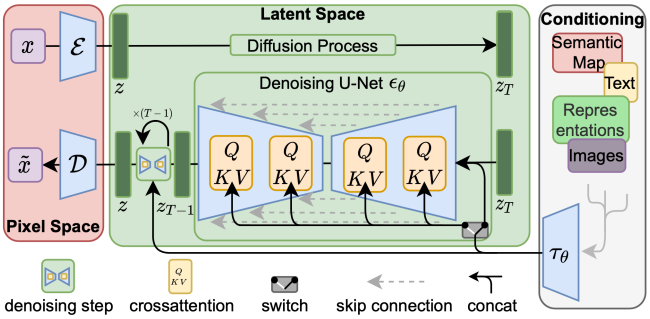

Fig. 10: Latent Diffusion Model Architecture.

2) Experiments: The experiments showed that $LDMs$ trained in VQ regularized latent spaces sometimes achieve better sample quality, even though the reconstruction capabilities of VQ regularized are usually worse. Also, small downsampling factors result in slow training progress, while overly large values cause stagnating fidelity after comparably few training steps.

a) On Perceptual Compression Tradeoffs: Latent Diffusion Models (LDMs) with different downsampling factors are analyzed. Smaller downsampling factors, such as LDM-1,2, exhibit slow training progress, while overly large values cause standstill in fidelity after a relatively small number of training steps. LDMs with downsampling factors between 4 and 16 strike a balance between efficiency and perceptually faithful results, outperforming pixel-based diffusion models.

b) Image Generation with Latent Diffusion: Unconditional models of 256^2 images trained on various datasets demonstrate state-of-the-art performance, achieving a new FID record on CelebA-HQ. The models outperform likelihood-based and GAN-based methods, showcasing the advantages of Latent Diffusion Models (LDMs).

c) Conditional Latent Diffusion: Introducing cross-attention based conditioning mechanisms to LDMs allows versatile conditioning modalities previously unexplored for diffusion models. The models demonstrate efficacy in text-to-image synthesis and image generation based on semantic layouts. LDMs consistently outperform GAN-based methods in Precision and Recall, emphasizing the benefits of their mode-covering likelihood-based training objective.

d) Super-Resolution with Latent Diffusion: LDMs prove efficient for super-resolution tasks by directly conditioning on low-resolution images. The models, particularly LDM-SR, demonstrate competitive performance, outperforming traditional approaches like SR3 [19] in FID.

e) Inpainting with Latent Diffusion: Also inpainting experiments have been conducted that compare LDMs with pixel-based and latent-based diffusion models, demonstrating the latter's speed-up and improvement in FID scores. LDMs with attention mechanisms outperform state-of-the-art inpainting models, providing diverse results favored by human subjects in a user study.

Tables 3 and 4 provide a summary of the experiments.

B. PartDiff: Image Super-resolution with Partial Diffusion Models

In the pursuit of efficient image super-resolution, the intricate computational demands posed by the Diffusion Process have been a significant challenge. The introduction of Partial Diffusion Models (PartDiff) by Zhao et al. presents a solution. By assuming that a low-resolution version of an image can serve as a proxy for its high-resolution counterpart, the computational cost of the diffusion process is notably reduced. PartDiff is based on the convergence of the latent states throughout the diffusion process, guiding the images towards an intermediate latent state that closely approximates the latent representation of the corresponding low-resolution image. During the generation process, denoising is selectively applied to a subset of steps, and the inclusion of latent alignment further refines the approximation, leading to a substantial reduction in denoising steps without compromising image quality. This innovation addresses the challenges posed by traditional diffusion models, such as SR3 and SRDiff [20], which, while effective in single-image super-resolution, are hindered by high computational costs.

In the broader context of image super-resolution research, traditional methods have tackled the challenge of generating high-resolution images from low-resolution inputs, relying on priors such as edge-based, statistical-based, and example-based approaches. Deep learning, particularly convolutional neural networks (CNNs) and generative adversarial networks (GANs), has shown promise but often struggles with producing clear, non-blurry images. Concurrently, diffusion probabilistic models like SR3 and SRDiff have demonstrated the potential to reverse gradual

noise processes, achieving high-quality samples. However, the computational overhead associated with numerous denoising steps in diffusion models has prompted various efforts to enhance their efficiency. PartDiff's distinct contribution lies in its targeted focus on image super-resolution, utilizing intermediate latent states to strategically reduce denoising steps.

1) Partial Diffusion Models for image Super Resolution:

a) Diffusing LR and HR Images: The process involves comparing the diffusion of low- and high-resolution images through forward processes denoted as $p\left(x_{1:T}^{LR} \mid x_0^{LR}\right)$ and $p\left(x_{1:T}^{HR} \mid x_0^{HR}\right)$. Here, x_t^{LR} and x_t^{HR} represent the latent states. The hypothesis suggests that convergence occurs at an intermediate step ($K < T$), where x_t^{HR} and x_t^{LR} become indistinguishable.

b) Partial Diffusion Models: In addressing the computational cost associated with numerous denoising steps, the authors propose Partial Diffusion Models. This model selectively executes a subset of denoising steps using x_K^{LR} as a proxy for x_K^{HR} in the reverse process, with $K < T$ as an intermediate step. Given a low-resolution image x_0^{LR} as input, the model initially diffuses it for K steps to obtain x_K^{LR}:

$$q\left(x_K^{LR} \mid x_0^{LR}\right) = \mathcal{N}\left(x_K; \sqrt{\alpha_K} x_0, (1-\bar{\alpha}_K)\,\mathbf{I}\right)$$

Subsequently, x_K^{LR} is employed as a surrogate for x_K^{HR}, initiating denoising from x_K^{HR} and progressing until reaching x_0^{HR}.

Figure 11 shows that the latent state of the low resolution image in an intermediate step can serve as a proxy for the high resolution image with some approximation error.

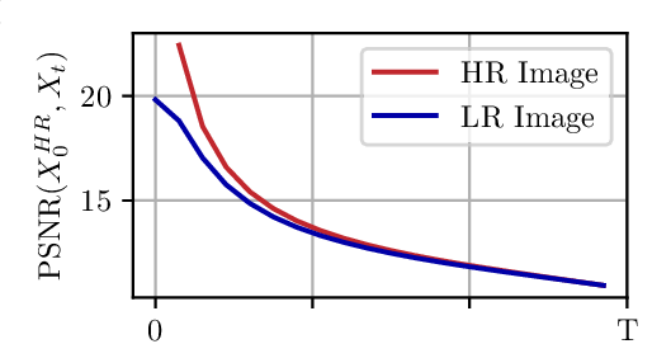

Fig. 11: PartDiff: SR latent state approximated with LR proxy.

c) Latent Alignment: To mitigate approximation errors stemming from subtle disparities between x_K^{LR} and x_K^{HR}, the Zhao et al. propose latent alignment. This technique gradually aligns the disparity between x_t^{LR} and x_t^{HR}. For each training iteration, a step index $t \in (0, K]$ is sampled, and a linear interpolation between latent states is performed:

$$q\left(\hat{x}_t \mid x_0^{LR}, x_0^{HR}\right) = \\ = \mathcal{N}\left(\hat{x}_t; \sqrt{\bar{\alpha}_t}\left(\lambda x_0^{HR} + (1-\lambda) x_0^{LR}\right), (1-\bar{\alpha}_t)\,\mathbf{I}\right)$$

Here, $\hat{x}_t$ represents the interpolated latent, and $\lambda = \frac{K-t}{K} \in [0,1]$ is a weight linearly increasing from 0 to 1. The forward diffusion posterior for $t < K$ is defined accordingly:

$$q\left(\hat{x}_{t-1} \mid x_t, x_0^{LR}, x_0^{HR}\right) = \mathcal{N}\left(x_{t-1}; \hat{\mu}_t\left(x_t, x_0\right), \hat{\beta}_t \mathbf{I}\right)$$

where

$$\hat{\mu}_t\left(x_t, x_0\right) = \lambda \tilde{\mu}_t\left(x_t^{HR}, x_0^{HR}\right) + (1-\lambda)\tilde{\mu}_t\left(x_t^{LR}, x_0^{LR}\right)$$

$$\hat{\beta}_t = \tilde{\beta}_t$$

The interpolated posterior serves as the target distribution in training the denoising model, and the loss function becomes:

$$L_{t-1} = D_{KL}\left(q\left(\hat{x}_{t-1} \mid x_t, x_0^{LR}, x_0^{HR}\right) \| p_\theta\left(x_{t-1} \mid x_t\right)\right)$$

The denoising model learns to gradually approach x_0^{HR} from x_K^{LR}.

The latent alignment is visualized in Figure 12

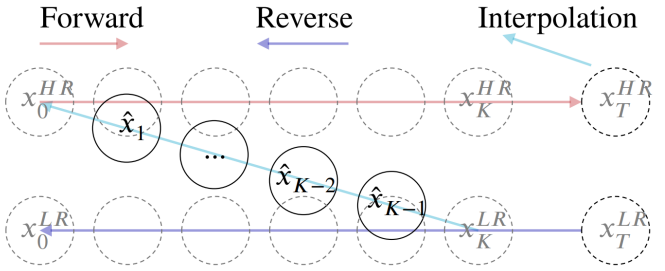

Fig. 12: PartDiff: Latent alignment used for mitigation of the approximation error.

2) Experiments: Implementation and experimental results are discussed in this section. Firstly, the implementation details are introduced, including datasets, model architecture, and training. Subsequently, experimental results on MRI and natural images are reported, comparing them with other super-resolution methods as well as with the proposed method and SR3. Finally, super-resolution methods are applied to downstream tasks.

a) Implementation details: A UNet [21] architecture similar to SR3 is adopted with minor modifications. The models are implemented using the PyTorch, with training parameters aligned with SR3 for fair comparisons.

b) Experiments on Clinical MRI: Super-resolution is tested on clinical prostate MRI scans. Results demonstrate the superiority of diffusion-based methods. Two settings, in-plane and through-plane super-resolution, are explored to address different aspects of MRI resolution enhancement.

c) Experiments on Natural Images: Super-resolution is applied to the ImageNet dataset, comparing the proposed method with other approaches. Quantitative evaluations, including FID and IS scores, reveal the realistic image generation capability of diffusion-based methods.

d) Image Classification: Super-resolution outputs are applied to downstream tasks, including zonal segmentation, face recognition, and image classification on ImageNet. Improved performance is demonstrated in zonal segmentation and face recognition tasks. In image classification, diffusion-based methods achieve the best top-1 and top-5 accuracies, showcasing their alignment with real images.

e) Application to Downstream Tasks: Super-resolution outputs are applied to downstream tasks, demonstrating improved performance in zonal segmentation and face recognition tasks. Image classification on ImageNet also shows enhanced accuracy. These findings highlight the practical utility of diffusion-based super-resolution in diverse applications.

Tables 5 and 6 show that "PartDiff" can compete with the state-of-the-art models.

C. Denoising Diffusion Models for Plug-and-Play Image Restoration

Plug-and-play Image Restoration (IR) has proven to be a flexible method for solving diverse inverse problems by leveraging off-the-shelf denoisers as implicit image priors. While existing methods predominantly focus on discriminative Gaussian denoisers, this paper explores the potential of diffusion models, known for their impressive performance in high-quality image synthesis, to serve as generative denoiser priors within the plug-and-play IR framework.

Recent studies have showcased the efficacy of plug-and-play IR methods in addressing various low-level vision tasks, including image denoising, super-resolution (SR), image deblurring, and inpainting, with outstanding results. However, existing approaches often limit themselves to task-specific Diffusion Models in Image Restoration. To overcome this limitation, the paper proposes a solution by employing the Alternating Direction Method of Multipliers (ADMM) or Half-Quadratic-Splitting (HQS) algorithm. This approach separates the data term, ensuring the solution adheres to the degradation process, and the prior term, enforcing the solution to follow the desired data distribution. The optimization problem is defined as:

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \frac{1}{2\sigma_n^2} \|\mathbf{y} - \mathcal{H}(\mathbf{x})\|^2 + \lambda \mathcal{P}(\mathbf{x}),$$

where $\mathbf{y}$ is the measurement of the ground truth $\mathbf{x}_0$ given the degradation model $\mathbf{y} = \mathcal{H}(\mathbf{x}_0) + \mathbf{n}$, $\mathcal{H}$ is a known degradation operator, σ_n is the known standard deviation of i.i.d. Gaussian noise $\mathbf{n}$, and $\lambda\mathcal{P}(\cdot)$ is the prior term with regularization parameter λ.

While traditional methods use denoisers like BM3D, recent attempts incorporate deep denoiser priors, showcasing the power of discriminative Gaussian denoisers. However, the limitations of denoisers parameterized by deep generative models, such as Generative Adversarial Networks (GANs), Normalizing Flows (NFs), and Variational Autoencoders (VAEs), hinder their effectiveness as plug-and-play priors. Recently, diffusion models have demonstrated the ability to generate high-quality images compared to previous generative models like GANs, VAEs, and NFs. Diffusion models define a forward diffusion process mapping data to noise and a reverse process generating images by gradually removing Gaussian noise, drawing inspiration from non-equilibrium thermodynamics. These models have achieved remarkable success in general inverse problems, including single-image SR, image inpainting, and noisy non-linear inverse problems.

Motivated by the flexibility of plug-and-play IR in utilizing off-the-shelf denoisers and recognizing the generative denoising capabilities of diffusion models, this paper introduces denoising diffusion models for plug-and-play IR, referred to as DiffPIR. Following the plug-and-play IR method proposed in, DiffPIR decouples the data term and the prior term, solving them iteratively within the diffusion sampling framework. The data term becomes independent, enabling DiffPIR to handle a wide range of degradation models, and the prior term is solved using an off-the-shelf diffusion model as a plug-and-play denoiser. Experimental results on different IR tasks, including SR, image deblurring, and image inpainting on FFHQ and ImageNet, demonstrate that DiffPIR efficiently restores images with superior quality.

1) Proposed Method: Conditional diffusion models play a pivotal role in conditional generation tasks, where the objective is to sample images from the posterior distribution $p(\mathbf{x} \mid \mathbf{y})$. In the context of Song et al.'s work, the conditional diffusion process is described by the equation (Bayes' Theorem):

$$\begin{aligned} \mathrm{d}\mathbf{x} = &\left[\mathbf{f}(\mathbf{x}, t) - g^2(t)\nabla_{\mathbf{x}} \left(\log p_t(\mathbf{x}) + \log p_t(\mathbf{y} \mid \mathbf{x})\right)\right] \mathrm{d}t \\ &+ g(t)\mathrm{d}\mathbf{w} \end{aligned}$$

where the posterior is decomposed into $p_t(\mathbf{x})$ and $p_t(\mathbf{y} \mid \mathbf{x})$. This allows the utilization of pre-trained unconditional diffusion models for conditional generation, augmented with an additional classifier.

Ho et al. introduced classifier-free diffusion guidance using $\mathbf{s}_\theta(\mathbf{x}, t, \mathbf{y}) = \nabla_{\mathbf{x}} \log p_t(\mathbf{x} \mid \mathbf{y})$ for image-conditional diffusion models. Following a similar approach, Saharia et al. trained image-conditional diffusion models for tasks such as super-resolution and image-to-image translation. Additionally, Nichol et al. [12] proposed text-guided diffusion models for generating photo-realistic images, where the hyperparameter λ in the diffusion equation serves as the guidance scale in classifier-free diffusion models.

a) Decoupling and Subproblems: The HQS algorithm is employed to decouple the data term and prior term in the inverse problem formulation, allowing iterative solutions of decoupled subproblems within the diffusion

sampling framework. Introducing an auxiliary variable $\mathbf{z}$, the problem is split into two subproblems:

$$\mathbf{z}_k = \arg\min_{\mathbf{z}} \frac{1}{2(\sqrt{\lambda/\mu})^2} \|\mathbf{z} - \mathbf{x}_k\|^2 + \mathcal{P}(\mathbf{z}),$$
$$\mathbf{x}_{k-1} = \arg\min_{\mathbf{x}} \|\mathbf{y} - \mathcal{H}(\mathbf{x})\|^2 + \mu\sigma_n^2 \|\mathbf{x} - \mathbf{z}_k\|^2,$$

where μ is the coefficient for the data-consistent constraint term.

b) Diffusion Models as Generative Denoiser Prior: Diffusion models, serving as a combination of generator and denoiser, are utilized as deep prior denoisers in the HQS algorithm. The generative power of diffusion models enables the solution of challenging inverse problems, shown in Algorithm 3 by approximating $\mathbf{z}_k$ as:

$$\mathbf{z}_k \approx \mathbf{x}_k + \frac{1-\bar{\alpha}_t}{\bar{\alpha}_t} \mathbf{s}_\theta(\mathbf{x}_k),$$

where $\bar{\alpha}_t$ is the known noise level. A visualization of this algorithm is shown in Figure 13.

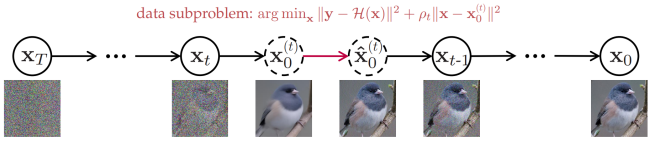

Fig. 13: DiffPIR sampling method: For every state x_t, following the prediction of the estimated $x_0^{(t)}$ by the diffusion model, the measurement y is incorporated by solving the data proximal subproblem. The next state x_{t-1} is derived by adding noise back and thus completing one step of reverse diffusion sampling.

Algorithm 3: DiffPIR Algorithm

Input: $\mathbf{s}_\theta, T, \mathbf{y}, \sigma_n, \{\bar{\sigma}_t\}_{t=1}^T, \zeta, \lambda$
Output: $\mathbf{x}_0$
Initialize $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, pre-calculate $\rho_t \triangleq \lambda\sigma_n^2/\bar{\sigma}_t^2$;
for $t = T$ to 1 do
  $\mathbf{x}_0^{(t)} = \frac{1}{\sqrt{\bar{\alpha}_t}}(\mathbf{x}_t + (1-\bar{\alpha}_t)\mathbf{s}_\theta(\mathbf{x}_t, t))$ // Predict $\hat{\mathbf{z}}_0$ with score model as denoiser;
  $\hat{\mathbf{x}}_0^{(t)} = \arg\min_{\mathbf{x}} \|\mathbf{y} - \mathcal{H}(\mathbf{x})\|^2 + \rho_t \left\|\mathbf{x} - \mathbf{x}_0^{(t)}\right\|^2$ // Solving data proximal subproblem;
  $\hat{\epsilon} = \frac{1}{\sqrt{1-\bar{\alpha}_t}}\left(\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\hat{\mathbf{x}}_0^{(t)}\right)$ // Calculate effective $\hat{\epsilon}(\mathbf{x}_t, \mathbf{y})$;
  $\epsilon_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$;
  $\mathbf{x}_{t-1} = \sqrt{\bar{\alpha}_{t-1}}\hat{\mathbf{x}}_0^{(t)} + \sqrt{1-\bar{\alpha}_{t-1}}\left(\sqrt{1-\zeta}\hat{\epsilon} + \sqrt{\zeta}\epsilon_t\right)$ // Finish one step reverse diffusion sampling;
end
return x_0

c) Analytic Solution to Data Subproblem: For image reconstruction tasks, a fast solution to subproblem based on the estimated z_0 is provided. When an analytical solution is not available, it is approximated using a first-order proximal operator method:

$$\hat{\mathbf{x}}_0^{(t)} \approx \mathbf{x}_0^{(t)} - \frac{\bar{\sigma}_t^2}{2\lambda\sigma_n^2} \nabla_{\mathbf{x}_0^{(t)}} \left\|\mathbf{y} - \mathcal{H}\left(\mathbf{x}_0^{(t)}\right)\right\|^2.$$

d) DiffPIR Sampling: The DiffPIR sampling method is introduced, obtaining an estimation $\hat{\mathbf{x}}_0^{(t)}(\mathbf{y})$ corrected by adding noise back to the noise level $t-1$, controlled by a hyperparameter ζ. This method, based on DDIM, is summarized in Algorithm 1, and Figure 2 illustrates the process.

e) Comparison to Related Works: DiffPIR is compared to related diffusion-based methods.
DDRM predicts $\mathbf{x}_0$ and adds noise to forward sample $\mathbf{x}_{t-1}$, limited to linear operators $\mathcal{H}$, while DiffPIR handles arbitrary degradation operators.
DPS uses Laplacian approximation for posterior sampling in general noisy inverse problems. DiffPIR, like other posterior sampling methods with diffusion models, handles measurements after each reverse diffusion step, supporting fast sampling.

f) Accelerated Generation Process: The quadratic sequence in DDIM is adapted for the sampling sequence, enabling accelerated generation. The sampling sequence, a subset of $[1, \ldots, N]$, supports fast sampling for better reconstructions.

2) Experiments:

a) Implementation Details: Extensive experiments were conducted on FFHQ 256×256 and ImageNet 256×256 datasets using pre-trained models. Linear noise schedules were employed with different sampling sequences. Degradation models included inpainting with various masks, Gaussian and motion blur, and super-resolution.

b) Quantitative Experiments: Metrics for comparison included Peak Signal-to-Noise Ratio (PSNR), Fréchet Inception Distance (FID), and Learned Perceptual Image Patch Similarity (LPIPS) distance. DiffPIR (20 and 100 NFEs) was compared with DDRM, DPS, and DPIR. Results for noisy and noiseless measurements demonstrated the superior performance of DiffPIR in terms of FID and LPIPS.

c) Qualitative Experiments: DiffPIR produced high-quality reconstructions with intricate details compared to DDRM and DPIR. It required fewer NFEs for faithful reconstruction compared to DPS. The method also showcased diversity in reconstructions.

d) Ablation Study: An ablation study investigated the effect of sampling steps (NFEs), the impact of t_{start} in the reverse diffusion process, and the effects of hyperparameters λ and ζ.

Tables 7 and 8 demonstrate that the "DiffPIR" exhibits remarkable results, outperforming state-of-the-art models in many senarios.

V. Conclusion

This survey systematically explores the landscape of diffusion models, beginning with foundational principles and extending into applications in image inpainting and super resolution. The review explores diffusion models that originate from the idea to use nonequilibrium thermodynamics to learn complex probability distributions, centering on DDPMs and SGMs. Building on this foundation, the survey explores image processing, examining key contributions like "RePaint" and "CoPaint" for image inpainting as well as latent and partial diffusion models for efficiency in super resolution applications and denoising diffusion models for plug-and-play image restoration senarios to utilize generic, big, pretrained models. Overall, this survey emphasizes the widespread impact of diffusion models in various image-related tasks, showcasing their versatility in the field of generative modeling.

References

Primary Sources

[1] J. Sohl-Dickstein, E. A. Weiss, N. Maheswaranathan, and S. Ganguli, Deep Unsupervised Learning using Nonequilibrium Thermodynamics, arXiv preprint arXiv:1503.03585 (2015).

[2] J. Ho, A. Jain, and P. Abbeel, Denoising Diffusion Probabilistic Models, arXiv preprint arXiv:2006.11239 (2020).

[3] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, Score-Based Generative Modeling through Stochastic Differential Equations, arXiv preprint arXiv:2011.13456 (2021).

[4] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, Luc Van Gool, RePaint: Inpainting using Denoising Diffusion Probabilistic Models, (2022), https://arxiv.org/abs/2201.09865.

[5] G. Zhang, J. Ji, Y. Zhang, M. Yu, T. Jaakkola, and S. Chang, Towards Coherent Image Inpainting Using Denoising Diffusion Implicit Models, arXiv preprint arXiv:2304.03322 (2023).

[6] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, High-Resolution Image Synthesis with Latent Diffusion Models, arXiv preprint arXiv:2112.10752 (2022).

[7] K. Zhao, A. L. Y. Hung, K. Pang, H. Zheng, and K. Sung, PartDiff: Image Super-resolution with Partial Diffusion Models, arXiv preprint arXiv:2307.11926 (2023).

[8] Y. Zhu, K. Zhang, J. Liang, J. Cao, B. Wen, R. Timofte, and L. Van Gool, Denoising Diffusion Models for Plug-and-Play Image Restoration, arXiv preprint arXiv:2305.08995 (2023).

[9] L. Yang, Z. Zhang, Y. Song, S. Hong, R. Xu, Y. Zhao, W. Zhang, B. Cui, and M.-H. Yang, Diffusion Models: A Comprehensive Survey of Methods and Applications, ACM Transactions on Computing, Vol. 1, No. 1, pp. 1-49, October 2023, doi: 10.1145/3626235. (GitHub: https://github.com/YangLing0818/Diffusion-Models-Papers-Survey-Taxonomy)

Secondary Sources

[10] T. Karras, S. Laine, and T. Aila, A Style-Based Generator Architecture for Generative Adversarial Networks, arXiv preprint arXiv:1812.04948 (2019), archivePrefix=arXiv, primaryClass=cs.NE.

[11] C. Saharia, W. Chan, H. Chang, C. A. Lee, J. Ho, T. Salimans, D. J. Fleet, and M. Norouzi, Palette: Image-to-Image Diffusion Models, arXiv preprint arXiv:2111.05826 (2022), archivePrefix=arXiv, primaryClass=cs.CV.

[12] A. Nichol, P. Dhariwal, A. Ramesh, P. Shyam, P. Mishkin, B. McGrew, I. Sutskever, and M. Chen, GLIDE: Towards Photorealistic Image Generation and Editing with Text-Guided Diffusion Models, arXiv preprint arXiv:2112.10741 (2022), archivePrefix=arXiv, primaryClass=cs.CV.

[13] H. Chung, B. Sim, and J. C. Ye, Come-Closer-Diffuse-Faster: Accelerating Conditional Diffusion Models for Inverse Problems through Stochastic Contraction, arXiv preprint arXiv:2112.05146 (2022), archivePrefix=arXiv, primaryClass=eess.IV.

[14] O. Avrahami, D. Lischinski, and O. Fried, Blended Diffusion for Text-driven Editing of Natural Images, In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), DOI: 10.1109/CVPR52688.2022.01767, June 2022.

[15] W.-F. Ku, W.-C. Siu, X. Cheng, and H. A. Chan, Intelligent Painter: Picture Composition With Resampling Diffusion Model, arXiv preprint arXiv:2210.17106 (2023), archivePrefix=arXiv, primaryClass=cs.CV.

[16] H. Chung, J. Kim, M. T. McCann, M. L. Klasky, and J. C. Ye, Diffusion Posterior Sampling for General Noisy Inverse Problems, arXiv preprint arXiv:2209.14687 (2023), archivePrefix=arXiv, primaryClass=stat.ML.

[17] J. Song, C. Meng, and S. Ermon, Denoising Diffusion Implicit Models, arXiv preprint arXiv:2010.02502 (2022), archivePrefix=arXiv, primaryClass=cs.LG.

[18] A. van den Oord, O. Vinyals, and K. Kavukcuoglu, Neural Discrete Representation Learning, arXiv preprint arXiv:1711.00937 (2018), archivePrefix=arXiv, primaryClass=cs.LG.

[19] C. Saharia, J. Ho, W. Chan, T. Salimans, D. J. Fleet, and M. Norouzi, Image Super-Resolution via Iterative Refinement, arXiv preprint arXiv:2104.07636 (2021), archivePrefix=arXiv, primaryClass=eess.IV.

[20] H. Li, Y. Yang, M. Chang, H. Feng, Z. Xu, Q. Li, and Y. Chen, SRDiff: Single Image Super-Resolution with Diffusion Probabilistic Models, arXiv preprint arXiv:2104.14951 (2021), archivePrefix=arXiv, primaryClass=cs.CV.

[21] O. Ronneberger, P. Fischer, and T. Brox, U-Net: Convolutional Networks for Biomedical Image Segmentation, arXiv preprint arXiv:1505.04597 (2015), archivePrefix=arXiv, primaryClass=cs.CV.

Appendix

CelebA-HQ	Wide		Narrow		Super-Resolve 2×		Altern. Lines		Half		Expand	
Methods	LPIPS↓	Votes [%]	LPIPS↓	Votes [%]	LPIPS↓	Votes [%]	LPIPS↓	Votes [%]	LPIPS↓	Votes [%]	LPIPS↓	Votes [%]
AOT [51]	0.104	11.6 ± 2.0	0.047	12.8 ± 2.1	0.714	1.1 ± 0.6	0.667	2.4 ± 1.0	0.287	9.0 ± 1.8	0.604	8.3 ± 1.7
DSI [33]	0.067	16.0 ± 2.3	0.038	22.3 ± 2.6	0.128	5.5 ± 1.4	0.049	5.1 ± 1.4	0.211	4.5 ± 1.3	0.487	4.7 ± 1.3
ICT [42]	0.063	27.6 ± 2.8	0.036	30.9 ± 2.9	0.483	4.2 ± 1.2	0.353	0.7 ± 0.5	0.166	12.7 ± 2.1	0.432	8.8 ± 1.8
DeepFillv2 [47]	0.066	23.9 ± 2.6	0.049	21.0 ± 2.5	0.119	9.8 ± 1.8	0.049	10.6 ± 1.9	0.209	4.1 ± 1.2	0.467	13.1 ± 2.1
LaMa [40]	0.045	41.8 ± 3.1	0.028	33.8 ± 3.0	0.177	5.5 ± 1.4	0.083	20.6 ± 2.5	0.138	35.6 ± 3.0	0.342	24.7 ± 2.7
RePaint	0.059	*Reference*	0.028	*Reference*	0.029	*Reference*	0.009	*Reference*	0.165	*Reference*	0.435	*Reference*

ImageNet	Wide		Narrow		Super-Resolve 2×		Altern. Lines		Half		Expand	
Methods	LPIPS↓	Votes [%]	LPIPS↓	Votes [%]	LPIPS↓	Votes [%]	LPIPS↓	Votes [%]	LPIPS↓	Votes [%]	LPIPS↓	Votes [%]
DSI [33]	0.117	31.7 ± 2.9	0.072	28.6 ± 2.8	0.153	26.9 ± 2.8	0.069	23.6 ± 2.6	0.283	31.4 ± 2.9	0.583	9.2 ± 1.8
ICT [42]	0.107	42.9 ± 3.1	0.073	33.0 ± 2.9	0.708	1.1 ± 0.6	0.620	6.6 ± 1.5	0.255	51.5 ± 3.1	0.544	25.6 ± 2.7
LaMa [40]	0.105	42.4 ± 3.1	0.061	33.6 ± 2.9	0.272	13.0 ± 2.1	0.121	9.6 ± 1.8	0.254	41.1 ± 3.1	0.534	20.3 ± 2.5
RePaint	0.134	*Reference*	0.064	*Reference*	0.183	*Reference*	0.089	*Reference*	0.304	*Reference*	0.629	*Reference*

Table 1: CelebA-HQ (top) and ImageNet (bottom) are subjected to a comparative analysis against state-of-the-art methods. The evaluation includes the computation of the LPIPS metric, where lower values indicate superior performance, and Votes across six distinct mask settings. The term 'Votes' denotes the ratio of votes concerning the referenced work.

Method	*Expand*		*Half*		*Altern*		*S.R.*		*Narrow*		*Wide*		*Text*		**Average**	
	LPIPS↓	Vote(%)↓	LPIPS↓	Vote(%)↓	LPIPS↓	Vote(%)↓	LPIPS↓	Vote(%)↓	LPIPS↓	Vote(%)↓	LPIPS↓	Vote(%)↓	LPIPS↓	Vote(%)↓	LPIPS↓	Vote(%)↓
CelebA-HQ																
BLENDED	0.557	82/80	0.228	64/72	0.047	12/30	0.269	78/86	0.078	54/64	0.102	46/58	0.011	18/12	0.185	51/57
DDRM	0.704	94/98	0.273	86/96	0.151	78/84	0.596	100/100	0.140	76/84	0.125	84/62	0.028	38/42	0.288	79/81
RESAMPLING	0.536	60/66	0.231	68/88	0.050	24/46	0.261	64/72	0.077	50/64	0.102	40/50	0.013	**-12**/8	0.181	42/56
REPAINT	0.496	24/18	0.199	2/12	**0.014**	**-32**/38	0.041	10/10	0.039	4/10	0.072	**-16/-32**	**0.006**	4/**-14**	0.124	**0**/6
DPS	**0.449**	**-16/-12**	0.261	28/32	0.166	58/72	0.182	60/82	0.160	72/52	0.181	30/28	0.152	58/60	0.222	41/45
DDNM	0.598	76/94	0.257	84/72	0.015	-2/**-2**	0.046	6/**0**	0.071	14/38	0.111	28/60	0.014	**-12**/10	0.158	27/39
COPAINT-FAST	0.483	10/34	0.203	44/20	0.057	10/2	0.084	20/6	0.068	16/10	0.096	20/4	0.036	14/-4	0.147	13/11
COPAINT	0.472	12/20	0.188	40/24	0.016	-6/-4	0.033	22/-4	0.040	20/14	0.071	24/-2	0.007	-12/-4	0.118	15/6
COPAINT-TT	0.464	0/0	**0.180**	**0/0**	**0.014**	0/0	**0.028**	**0/0**	**0.037**	**0/0**	**0.069**	0/0	**0.006**	0/0	**0.114**	**0/0**
ImageNet																
BLENDED	0.717	39/36	0.366	72/80	0.277	96/92	0.686	94/96	0.161	76/64	0.194	62/60	0.028	8/26	0.347	64/65
DDRM	0.730	58/44	0.385	78/64	0.439	92/100	0.822	92/100	0.211	84/84	0.231	86/72	0.060	32/44	0.411	75/71
RESAMPLING	0.704	38/40	0.353	58/86	0.259	72/88	0.624	94/98	0.151	66/64	0.183	76/66	0.028	22/26	0.329	61/67
REPAINT	0.706	36/36	0.323	4/24	0.103	50/22	0.209	70/66	**0.072**	32/2	0.156	24/36	0.014	22/18	0.226	34/29
DPS	0.673	38/44	0.512	82/72	0.474	100/100	0.511	96/95	0.447	94/98	0.468	96/92	0.438	92/96	0.503	87/86
DDNM	0.805	34/76	0.408	68/64	0.051	12/12	0.107	18/36	0.101	50/70	0.185	48/60	**0.012**	**-2/-20**	0.238	33/44
COPAINT-FAST	0.678	14/26	0.335	22/24	0.075	10/6	0.128	36/28	0.103	26/22	0.167	24/32	0.043	6/-2	0.218	15/19
COPAINT	0.640	-2/8	0.307	6/0	0.041	22/4	**0.069**	20/18	0.078	24/30	0.138	14/16	0.017	2/-10	0.184	12/9
COPAINT-TT	**0.636**	**0/0**	**0.294**	**0/0**	**0.039**	**0/0**	**0.069**	**0/0**	0.074	**0/0**	**0.133**	**0/0**	0.015	0/0	**0.180**	**0/0**

Table 2: In the analysis of CelebA-HQ (top) and ImageNet (bottom), the study presents quantitative results, including LPIPS objective metrics and subjective human vote differences compared to the COPAINT-TT method. Lower scores are preferred for both metrics. The vote differences, expressed as percentages relative to COPAINT-TT, reveal superior performance of certain baselines with negative values. Human tests encompass overall assessment (naturalness, restoration quality, and coherence) and coherence-only evaluations. Blue-marked numbers denote additional results.

CelebA-HQ 256 × 256				FFHQ 256 × 256			
Method	FID ↓	Prec. ↑	Recall ↑	**Method**	FID ↓	Prec. ↑	Recall ↑
DC-VAE [63]	15.8	-	-	ImageBART [21]	9.57	-	-
VQGAN+T. [23] (k=400)	10.2	-	-	U-Net GAN (+aug) [77]	10.9 (7.6)	-	-
PGGAN [39]	8.0	-	-	UDM [43]	5.54	-	-
LSGM [93]	7.22	-	-	StyleGAN [41]	4.16	0.71	0.46
UDM [43]	7.16	-	-	ProjectedGAN [76]	**3.08**	0.65	0.46
LDM-4 (ours, 500-s†)	**5.11**	0.72	0.49	*LDM-4* (ours, 200-s)	4.98	**0.73**	**0.50**

LSUN-Churches 256 × 256				LSUN-Bedrooms 256 × 256			
Method	FID ↓	Prec. ↑	Recall ↑	**Method**	FID ↓	Prec. ↑	Recall ↑
DDPM [30]	7.89	-	-	ImageBART [21]	5.51	-	-
ImageBART [21]	7.32	-	-	DDPM [30]	4.9	-	-
PGGAN [39]	6.42	-	-	UDM [43]	4.57	-	-
StyleGAN [41]	4.21	-	-	StyleGAN [41]	2.35	0.59	0.48
StyleGAN2 [42]	3.86	-	-	ADM [15]	1.90	**0.66**	**0.51**
ProjectedGAN [76]	**1.59**	0.61	0.44	ProjectedGAN [76]	**1.52**	0.61	0.34
*LDM-8** (ours, 200-s)	4.02	**0.64**	**0.52**	*LDM-4* (ours, 200-s)	2.95	**0.66**	0.48

Table 3: In the context of unconditional image synthesis, evaluation metrics are employed. CelebA-HQ results and FFHQ results are presented. The symbol † indicates N-s, denoting N sampling steps with the DDIM [84] sampler. The symbol ∗ indicates models trained in a KL-regularized latent space. Supplementary material contains additional results.

Text-Conditional Image Synthesis				
Method	FID ↓	IS↑	N_{params}	
CogView† [17]	27.10	18.20	4B	self-ranking, rejection rate 0.017
LAFITE† [109]	26.94	26.02	75M	
GLIDE* [59]	12.24	-	6B	277 DDIM steps, c.f.g. [32] $s = 3$
Make-A-Scene* [26]	**11.84**	-	4B	c.f.g for AR models [98] $s = 5$
LDM-KL-8	23.31	$20.03_{\pm 0.33}$	1.45B	250 DDIM steps
*LDM-KL-8-G**	12.63	$\mathbf{30.29}_{\pm \mathbf{0.42}}$	1.45B	250 DDIM steps, c.f.g. [32] $s = 1.5$

Table 4: In the evaluation of text-conditional image synthesis on the 256 × 256-sized MS-COCO dataset, the latent diffusion model attains parity with recent diffusion and autoregressive methods, despite utilizing significantly fewer parameters, incorporating 250 DDIM steps.

Method	×2 PSNR (↑)	×2 SSIM (↓)	×4 PSNR (↑)	×4 SSIM (↓)
Bicubic	32.0775	0.9226	24.6771	0.6808
Regression	32.7084	0.9109	25.3750	0.7112
SRGAN [31]	33.3979	0.9135	26.9268	0.7503
LIIF [30]	33.4791	0.9270	27.9112	0.7929
SR3 ($T = 100$)	**33.5382**	0.9301	28.2462	**0.8101**
PartDiff ($K = 25$)	33.5194	0.9287	28.2329	0.8002
PartDiff ($K = 50$)	33.5318	0.9295	**28.2463**	0.8093

Table 5: In-plane MRI super-resolution results on ProstateX dataset.

Method		T1W PSNR (↑)	T1W SSIM (↑)	T2W PSNR(↑)	T2W SSIM (↑)
Bicubic	×2	33.3657	0.7509	29.5263	0.9091
Regression	×2	34.4634	0.9265	32.7347	0.9035
SRGAN [31]	×2	36.0610	0.9434	36.2626	0.9365
LIIF [30]	×2	37.3901	0.9524	35.8739	0.9433
SR3 [15] ($T = 100$)	×2	**37.5087**	0.9670	**36.3017**	**0.9413**
PartDiff ($K = 25$)	×2	37.5109	0.9672	36.2859	0.9411
PartDiff ($K = 50$)	×2	37.5071	**0.9672**	36.2770	0.9411
Bicubic	×4	26.5617	0.8278	24.9121	0.6999
Regression	×4	29.0838	0.8095	27.3084	0.7199
SRGAN [31]	×4	30.2013	0.8444	29.6069	0.7443
LIIF [30]	×4	31.3574	0.8647	30.2751	0.8261
SR3 [15] ($T = 100$)	×4	31.5104	**0.8656**	**30.4349**	0.8348
PartDiff ($K = 25$)	×4	31.3581	0.8607	30.3109	0.8301
PartDiff ($K = 50$)	×4	**31.5149**	0.8617	30.1342	**0.8156**

Table 6: Quantitative performance of in-plane MRI super-resolution results on in-house prostate MRI dataset.

FFHQ		**Deblur (Gaussian)**			**Deblur (motion)**			**SR** (×4)		
Method	NFEs ↓	PSNR ↑	FID ↓	LPIPS ↓	PSNR ↑	FID ↓	LPIPS ↓	PSNR ↑	FID ↓	LPIPS ↓
DiffPIR	100	27.36	**59.65**	**0.236**	**26.57**	**65.78**	**0.255**	26.64	**65.77**	0.260
DPS [8]	1000	25.46	65.57	0.247	23.31	73.31	0.289	25.77	67.01	**0.256**
DDRM [32]	20	25.93	101.89	0.298	-	-	-	27.92	89.43	0.265
DPIR [57]	>20	**27.79**	123.99	0.450	26.41	146.44	0.467	**28.03**	133.39	0.456

ImageNet		**Deblur (Gaussian)**			**Deblur (motion)**			**SR** (×4)		
Method	NFEs ↓	PSNR ↑	FID ↓	LPIPS ↓	PSNR ↑	FID ↓	LPIPS ↓	PSNR ↑	FID ↓	LPIPS ↓
DiffPIR	100	22.80	**93.36**	**0.355**	**24.01**	**124.63**	**0.366**	23.18	**106.32**	0.371
DPS [8]	1000	19.58	138.80	0.434	17.75	184.45	0.491	22.16	114.93	0.383
DDRM [32]	20	22.33	160.73	0.427	-	-	-	23.89	118.55	**0.358**
DPIR [57]	>20	**23.86**	189.92	0.476	23.60	210.31	0.489	**23.99**	204.83	0.475

Table 7: In the evaluation of noisy quantitative results on FFHQ (top) and ImageNet (bottom), the average PSNR (dB), FID, and LPIPS metrics are computed for various methods in Gaussian deblurring, motion deblurring, and 4× SR.

FFHQ		**Inpaint (box)**		**Inpaint (random)**			**Deblur (Gaussian)**			**Deblur (motion)**			**SR** (×4)		
Method	NFEs ↓	FID ↓	LPIPS ↓	PSNR ↑	FID ↓	LPIPS ↓	PSNR ↑	FID ↓	LPIPS ↓	PSNR ↑	FID ↓	LPIPS ↓	PSNR ↑	FID ↓	LPIPS ↓
DiffPIR	20	35.72	0.117	34.03	30.81	0.116	30.74	46.64	0.170	37.03	20.11	0.084	29.17	58.02	0.187
DiffPIR	100	**25.64**	**0.107**	**36.17**	**13.68**	**0.066**	**31.00**	**39.27**	**0.152**	37.53	**11.54**	**0.064**	29.52	**47.80**	**0.174**
DPS [8]	1000	43.49	0.145	34.65	33.14	0.105	27.31	51.23	0.192	26.73	58.63	0.222	27.64	59.06	0.209
DDRM [32]	20	37.05	0.119	31.83	56.60	0.164	28.40	67.99	0.238	-	-	-	30.09	68.59	0.188
DPIR [57]	>20	-	-	-	-	-	30.52	96.16	0.350	**38.39**	27.55	0.233	**30.41**	96.16	0.362

Table 8: In the analysis of noiseless quantitative results on FFHQ, average PSNR (dB), FID, and LPIPS metrics are computed for various methods in inpainting, deblurring, and SR.