Image inpainting and Super-Resolution using Diffusion Models and Conditional Flow Matching

Vladimir Radenkovic
University of Cambridge
vr375@cam.ac.uk

Abstract

Recent advancements in deep generative models have shown remarkable achievements in the realm of conditional image generation, particularly with diffusion models and the novel approach of conditional flow matching (CFM). While recent innovations, such as amortized Doob's h-transform in diffusion models and novel CFM variants, have shown promising results, a comprehensive evaluation of these models, especially in the context of image inpainting and super-resolution tasks, remains unexplored. This study aims to bridge this gap by providing empirical comparison of these models on the inverse problems of inpainting and super-resolution,offering insights into their performance in an uniform experimental setting. Furthermore, we conduct theoretical and implementation analyses of these. we hope to clarify the relative strengths and potential of these emerging, yet under explored techniques, providing a clearer understanding of their capabilities in generating conditional images.

1 Introduction

In recent years diffusion models have emerged as state of the art deep generative model aimed at estimating and sampling from an unknown data distribution conditioned on specific event [13, 5]. These models have excelled in tasks ranging from protein design to image generation. The emergence of novel *amortized* training methods, leveraging Doob's h-transform [3], marks a significant advancement as it offers both theoretically grounded approach to conditioning diffusion processes and more practical implementation by unifying conditional training and sampling under a single framework. The potential of this method has been demonstrated in image outpainting and protein motif scaffolding tasks.

Concurrently, recent developments [7, 1, 8] have introduced Conditional Flow Matching (CFM), a simulation-free method for training Continuous Normalizing Flow (CNF) models. CFM stands out as an efficient alternative to popular deep generative methods like diffusion models, enabling the training and sampling of CNFs along predefined probability paths. This approach is very flexible as it allows for creation of almost arbitrary probability paths, with recent emphasis on paths that encapsulate Optimal Transport (OT) displacement interpolation [7, 14].

Despite the potential of both diffusion models with amortized conditioning and conditional flow matching, a comprehensive study exploring these methodologies is lacking. Specifically, the application of these techniques on well explored, yet challenging tasks image inpainting and super-resolution has not been extensively researched under a consistent training and evaluation framework.

Submitted to 34th Conference on Neural Information Processing Systems (NeurIPS 2020). Do not distribute.

In this work, we address this gap by providing both detailed theoretical and empirical comparison of amortised conditioning of diffusion models and conditional flow matching, in context of image inpainting and super-resolution tasks. Through a detailed empirical evaluation, we demonstrate distinct capabilities, training and inference behaviors, and the inherent strengths and weaknesses of these advanced generative models under the uniform experimental conditions in conditional image generation.

2 Background

2.1 Conditional Generation with Deep Generative Models

Let $\mathbb{R}^d$ denote the data space with data points $\mathbf{x} \in \mathbb{R}^d$. The goal of deep generative models is to estimate and sample from an unknown data distribution, $p_{\text{data}}(\mathbf{x})$, using a simple prior distribution $p_0(\mathbf{x})$ such as the standard normal distribution $\mathcal{N}(x|0, I)$. In this work, we investigate two distinct classes of deep generative models that aim to learn a mapping from p_0 to p_{data}. We introduce the concept of a *probability density path* $p_t(\mathbf{x}) : [0, 1] \times \mathbb{R}^d \to \mathbb{R}> 0$ as a time dependent probability density function, i.e., $\int p_t(\mathbf{x})d\mathbf{x}$, where p_0 is the simple prior and $p_1 = p_{\text{data}}$ is the target data distribution.

Conditional image generation, following the framework outlined in [3, 2], is defined as the process of generating samples $\mathbf{x}$ that meet a specific condition $\mathbf{X}_0 \in B$ at a boundary time ($t = 0$ or $t = 1$). We consider events that are described by an equality constraint $\mathcal{A}(\mathbf{X}_0) = \mathbf{y}$ where $\mathcal{A}(\cdot) : \mathbb{R}^d \mapsto \mathbb{R}^n$ is the *forward measurement operator* and $\mathbf{y}$ is an observation. These are ill-poised inverse problems where we cannot exactly retrieve $\mathbf{x}$, as the events are many to one mappings. The objective is to estimate and sample from data distribution conditioned on hard constraint: $p_{\text{data}}(\mathbf{x}|\mathcal{A}(\mathbf{x}) = \mathbf{y})$. We asses the performance of the models on two particular inverse problems in image processing:

1. **Image Inpainting.** In this task, also known as Image Completion, generative models aim to fill the missing central patch of the image. For an image of size $N \times N$ the measurement model generates a mask of black pixels of size $K \times K$ at a random position in the image, where $K \leq N$. Prior research [13, 9] has demonstrated the potential of conditional generative models in effectively resolving this task.
2. **Image Super Resolution.** The objective is to generate a high-resolution image of size $N \times N$ from corresponding low-resolution image $K \times K$ where $K \leq N$. Research [12] has highlighted the efficacy of diffusion models in achieving high fidelity in this task.

2.2 Conditional Generation with Score Based Generative models

We focus on the Ornstein-Uhlenbeck (OU) forward noising process, $\mathbf{x}_t$ within the time interval $t \in [0, 1]$, defined by the stochastic differential equation (SDE):

$$d\mathbf{x} = -\frac{\beta(t)}{2}\mathbf{x}dt + \sqrt{\beta(t)}d\mathbf{w}, \quad \mathbf{x}_0 \sim p_{\text{data}} \tag{1}$$

, where $\beta(t) : \mathbb{R} \to \mathbb{R} > 0$ is the noise schedule of the process, defined as a linear function of t, and $\mathbf{w}$ is the standard d-dimensional Wiener process. The initial data points at $t = 0$ are drawn from the unknown data distribution $\mathbf{x}_0 \sim p_0(\mathbf{x}) = p_{\text{data}}(\mathbf{x})$, and the process evolves to standard normal distribution $\mathcal{N}(x|0, I)$ at terminal point $t = 1$, $\mathbf{x}_1 \sim p_1(\mathbf{x}) = \mathcal{N}(0, I)$. This noising process, known as the variance preserving (VP) SDE is equivalent with Denoising Diffusion Probabilistic Models (DDPM) [5], as established in [13].

Score based generative models aim to recover the data distribution from simple prior with the the reverse SDE corresponding to 1[13]:

$$d\mathbf{x} = -\beta(t)\left(\frac{1}{2}\mathbf{x} + \nabla_{\mathbf{x}_t}\log p_t(\mathbf{x}_t)\right)dt + \sqrt{\beta(t)}d\bar{\mathbf{w}}, \tag{2}$$

where $\bar{\mathbf{w}}$ is a standard reverse Wiener process and dt denotes reversed time. Diffusion models approximate the score function with a neural network $\mathbf{s}_\theta$ trained with denoising score matching:

$$\theta^* = \arg\min_\theta \mathbb{E}_{t\sim U(0,1),\mathbf{x}_t\sim p(\mathbf{x}_t|\mathbf{x}_0),\mathbf{x}_0\sim p_{\text{data}}}\left[\|\mathbf{s}_\theta(\mathbf{x}_t,t) - \nabla_{\mathbf{x}_t}\log p(\mathbf{x}_t|\mathbf{x}_0)\|^2\right], \tag{3}$$

Trained neural network $\mathbf{s}_{\theta^*}$ is used as approximation of score $\nabla_{\mathbf{x}_t}\log p_t(\mathbf{x}_t) \approx \mathbf{s}_{\theta^*}$ in backward SDE 2 to sample $\mathbf{x}_0 \sim p_{\text{data}}$ starting from sample from simple prior $\mathbf{x}_1 \sim p_1(\mathbf{x})$. In discrete settings with N bins we define $\mathbf{x}_i = \mathbf{x}(\frac{i}{N})$, $\beta_i = \beta(\frac{i}{N})$, and subsequently, $\alpha_i = 1 - \beta_i$, $\overline{\alpha}_i = \prod_{j=1}^{i}\alpha_i$ as in [5].

Doob's h-transform. Doob's h-transform provides an efficient and elegant way to condition the reverse SDE to hit an particular event at finite time T. Following Proposition 2.1 and Corollary 2.2 from [3], we outline how Doob's h-transform can be used to generate samples conditioned on $\mathcal{A}(\mathbf{X}_0) = \mathbf{y}$.

Proposition 1 *Consider the reverse SDE defined as*

$$d\mathbf{x} = b_t(\mathbf{x})dt + \sigma_t d\bar{\mathbf{w}}, \quad \mathbf{x}_T \sim p_T \tag{4}$$

where $b_t(\mathbf{x}_t)$ is the drift term and σ_t^2 represents the diffusion coefficient of the reverse SDE 2 with transitional densities $p_{t|s}$ and T is terminal time point. It then follows that the conditioned process $\mathbf{x}_t|\mathcal{A}(\mathbf{x_0})$ is a solution of

$$d\mathbf{h} = \left(b_t(\mathbf{h}) - \sigma_t^2\nabla_{\mathbf{h}_t}\log p_{0|t}(\mathcal{A}(\mathbf{x}_0) = \mathbf{y}|\mathbf{h}_t)\right)dt + \sigma_t^2 d\bar{\mathbf{w}}, \mathbf{x}, \quad \mathbf{x}_T \sim p_T \tag{5}$$

such that transitional densities of satisfy $p_{s|t}(\mathbf{h}_s|\mathbf{h}_t) = p_{s|t}(\mathbf{x}_s|\mathbf{x}_t, \mathcal{A}(\mathbf{x}_0) = \mathbf{y})$ and $p(\mathbf{h}_0) = p(\mathbf{x}_0|\mathcal{A}(\mathbf{x}_0) = \mathbf{y})$

By adding additional drift term in the backward SDE 2 the resulting conditioned process is guaranteed to provide samples $\mathbf{x} \sim p_{\text{data}}$ that satisfy the condition $\mathcal{A}(\mathbf{X}_0) = \mathbf{y}$ and this SDE is guaranteed to hit the condition within finite time T. The function $h(\mathbf{h}_t, t) = p_{0|t}(\mathcal{A}(\mathbf{X}_0) = \mathbf{y}|\mathbf{h}_t)$ is referred to as the *h-transform* [4, 11].

We note that the Doob's transformed SDE of a reversed OU process can be expressed as:

$$d\mathbf{h}_t = -\beta(t)\left(\frac{\mathbf{h}_t}{2} + \nabla_{\mathbf{h}_t}\log p_{t|0}(\mathbf{h}_t|\mathcal{A}(\mathbf{x}_0) = \mathbf{y})\right)dt + \sqrt{\beta(t)}\bar{\mathbf{w}}, \quad \mathbf{h}_T \sim p_T(\mathbf{x}) \tag{6}$$

[3] propose an objective for learning Doob's *h*-transform during training, rather than enforcing constraints during inference, as it is common in reconstruction guidance techniques. This is encapsulated in proposition 2.5 from [3]:

Proposition 2 *The minimiser of*

$$\theta^* = \arg\min_\theta \mathbb{E}_{\mathbf{y}\sim p(\mathcal{A},\mathbf{x}_0),\mathcal{A}\sim p,\mathbf{x}_0(t)\sim p_{data}}\left[\int_0^T \|\mathbf{s}_\theta(t,\mathbf{x}_t,\mathbf{y},\mathcal{A}) - \nabla_{\mathbf{x}_t}\log p_{t|0}(\mathbf{x}_t|\mathbf{x}_0)\|^2 dt\right], \tag{7}$$

is given by conditional score $\mathbf{s}_\theta(t,\mathbf{x}_t,\mathbf{y},\mathcal{A}) = \nabla_{\mathbf{h}_t}\log p_{t|0}(\mathbf{x}_t|\mathbf{x}_0)$

This approach is named *amortised* learning for conditional sampling, as the neural network $\mathbf{s}_\theta(t,\mathbf{x}_t,\mathbf{y},\mathcal{A})$ approximating the conditional score is amortised over $\mathcal{A}$ and $\mathbf{y}$, rather than having to learn separate networks for each condition, making it more efficient and effective. This approach is similar to 'classifier free guidance,' as both methods involve amortizing the scoring network over some auxiliary variable. However, the primary difference is that $\mathcal{A}$ in amortised learning is assumed to be known.

2.3 Conditional Generation with Continuous Normalizing Flows and Flow Matching

Let $\phi_t(\mathbf{x}) : [0,1]\times\mathbb{R}^d \to \mathbb{R}^d$ be a mapping called *flow* that maps samples $\mathbf{x}_0$ from simple prior density to samples $\mathbf{x}$ from distribution p_t. We introduce time a *time-dependent vector field* as $u_t(\mathbf{x}) : [0,1]\times\mathbb{R}^d \to \mathbb{R}^d$ that is used to construct $\phi_t(\mathbf{x})$ via an ordinary differential equation (ODE):

$$\frac{d}{dt}\phi_t(\mathbf{x}_0) = u_t(\phi_t(\mathbf{x}_0)) \tag{8}$$

$$\phi_0(\mathbf{x}_0) = \mathbf{x}_0 \tag{9}$$

Flow transforms simple prior density p_0 to data distribution p_t via the push-forward (or change of variables) operator $*$:

$$p_t(\mathbf{x}) = [\phi_t]_* p_0(\mathbf{x}) \tag{10}$$

$$= p_0(\phi_t^{-1}(\mathbf{x}))\text{det}\left[\frac{\partial \phi_t^{-1}}{\partial \mathbf{x}}(\mathbf{x})\right]. \tag{11}$$

A vector field u_t is said to generate a probability density path p_t if its flow ϕ_t satisfies equation 3. Probability density path p_t defined by p_0 and u_t is characterized by the *continuity equation*:

$$\frac{d}{dt}p_t(\mathbf{x}) + \text{div}(p_t(\mathbf{x})u_t(\mathbf{x})) = 0, \tag{12}$$

where div is divergence operator defined as $\text{div} = \sum_{i=1}^{d} \frac{\partial}{\partial x^i}$. This equation provides a necessary and sufficient condition that ensures that vector field u_t generates p_t. A vector field u_t is said to generate a probability density path p_t if its flow ϕ_t satisfies equation 12.

Continuous Normalizing Flows (CNF) are a class of deep generative models that model the flow ϕ_t with neural network modeling the vector field $v_t(\mathbf{x}, \theta)$. This neural network can be regressed to u_t by minimizing the Flow Matching (FM) objective:

$$\mathcal{L}_{\text{FM}}(\theta) = \mathbf{E}_{t,p_t(\mathbf{x})} \left\| v_t(\mathbf{x}) - u_t(\mathbf{x}) \right\|^2, \tag{13}$$

where θ denotes the learnable parameters of the vector field v_t, upon reaching zero loss, the learned CNF will generate $p_t(x)$. There are many choices of p_t and u_t that satisfy $p_1(\mathbf{x}) \approx q(x)$ and we do not have access to closed form u_t that generates desired p_t.

In [7] authors construct a target probability path via simpler *conditional probability path* $p_t(\mathbf{x}|\mathbf{x}_1)$, where x_1 is a data sample, that satisfies $p_0(\mathbf{x}|\mathbf{x}_1) = p_0(\mathbf{x})$ and $p_1(\mathbf{x}|\mathbf{x}_1) = \mathcal{N}(\mathbf{x}|\mathbf{x}_1, \sigma^2 I)$, where $\sigma > 0$ is sufficiently small standard deviation such that $p_1(\mathbf{x}|\mathbf{x}_1)$ is concentrated around $\mathbf{x}_1$. By marginalizing the $p_t(\mathbf{x}|\mathbf{x}_1)$ over $q(\mathbf{x}_1)$ we get the *marginal probability path*:

$$p_t(\mathbf{x}) = \int p_t(\mathbf{x}|\mathbf{x}_1) q(\mathbf{x}_1) d\mathbf{x}_1. \tag{14}$$

Next, we can define conditional vector field $u_t(\mathbf{x}|\mathbf{x}_1)$ that generates $p_t(\mathbf{x}|\mathbf{x}_1)$ and by by "marginalizing" over conditional vector fields we get *marginal vector field*:

$$u_t(\mathbf{x}) = \int u_t(\mathbf{x}|\mathbf{x}_1) \frac{p_t(\mathbf{x}|\mathbf{x}_1) q(\mathbf{x}_1)}{p_t(\mathbf{x})} d\mathbf{x}_1 \tag{15}$$

A key observation is that if $u_t(\mathbf{x}|\mathbf{x}_1)$ generates $p_t(\mathbf{x}|\mathbf{x}_1)$ then marginal vector field15 generates marginal probability path 14. By introducing conditional VFs we can break down the unknown and intractable marginal VF into simpler conditional VFs, that are much simpler to define as they only depend on a single data sample. [7] propose a novel *Conditional Flow Matching* (CFM) training objective for regressing the vector field u_t with a neural network v_t:

$$\mathcal{L}_{\text{CFM}}(\theta) = \mathbf{E}_{t,q(\mathbf{x}_1),p_t(\mathbf{x}|\mathbf{x}_1)} \left\| v_t(\mathbf{x}) - u_t(\mathbf{x}|\mathbf{x}_1) \right\|^2, \tag{16}$$

where $t \sim \mathcal{U}[0.1]$. This simpler objective will result in the same minima as the original objective as both objectives have identical gradients w.r.t. θ, meaning that optimizing the CFM objective is equivalent in expectation to optimizing the FM objective. This objective shows how to regress the marginal VF $u_t(\mathbf{x})$ with access only to samples from conditional probability path $p_t(\mathbf{x}|\mathbf{x}_1)$ and conditional VFs $u_t(\mathbf{x}|\mathbf{x}_1)$.

As in [7] we consider a general family of Gaussian conditional probability paths of form:

$$p_t(\mathbf{x}|\mathbf{x}_1) = \mathcal{N}(\mathbf{x}|\mu_t(\mathbf{x}_1), \sigma_t(\mathbf{x}_1)^2 I), \tag{17}$$

with $\mu_0(\mathbf{x}_1) = 0$, $\sigma_0(\mathbf{x}_1) = 1$, $\mu_1(\mathbf{x}_1) = \mathbf{x}_1$ and $\sigma_1(\mathbf{x}_1) = \sigma_{\text{min}}$. σ_{min} is set sufficiently small so that $p_1(\mathbf{x}|\mathbf{x}_1)$ is concentrated Gaussian distribution centered at x1. While there is an infinite number of

VFs that generate any Gaussian conditional probability paths 17, we focus on VFs that construct simple flow (conditioned on $\mathbf{x}_1$) in the form:

$$\phi_t(\mathbf{x}) = \sigma_t(\mathbf{x}_1)\mathbf{x} + \mu_t(\mathbf{x}_1). \tag{18}$$

Theorem 3 from [7] states that for Gaussian probability path defined in 17 and its corresponding flow map $\phi_t(\mathbf{x})$ 18 the unique VF that defines $\phi_t(\mathbf{x})$ has the form:

$$u_t(\mathbf{x}|\mathbf{x}_1) = \frac{\sigma_t'(\mathbf{x}_1)}{\sigma_t(\mathbf{x}_1)}(\mathbf{x} - \mu_t(\mathbf{x}_1)) + \mu_t'(\mathbf{x}_1), \tag{19}$$

where σ_t' and μ_t' denote the time derivative of σ_t and μ_t respectively.

This definition of $\phi_t(\mathbf{x})$ allows for arbitrary selection of differentiable functions $\sigma_t(\mathbf{x}_1)$ and $\mu_t(\mathbf{x}_1)$ that that satisfy the desired boundary conditions. In this work, we evaluate two specific choices of flow functions that generate two distinct probability paths based on the Wasserstein-2 optimal transport solution.

Optimal Transport Conditional VFs between conditional paths Mean and std of these conditional probability paths are changing linearly in time as follows:

$$\mu_t(\mathbf{x}_1) = t\mathbf{x}_1, \text{ and } \sigma_t(\mathbf{x}_1) = 1 - (1 - \sigma_{\text{min}})t. \tag{20}$$

According to theorem 3 in [7] this path is generated by the following conditional VF

$$u_t(\mathbf{x}|\mathbf{x}_1) = \frac{\mathbf{x}_1 - (1 - \sigma_{\text{min}})\mathbf{x}}{1 - (1 - \sigma_{\text{min}})t} \tag{21}$$

This conditional flow $\phi_t(\mathbf{x})$ not only generates simple and intuitive paths, but it is also the Optimal Transport (OT) displacement map between start and end conditional Gaussians $p_0(\mathbf{x}|\mathbf{x}_1)$ and $p_1(\mathbf{x}|\mathbf{x}_1)$. Following the notation from [14], we refer to CNF generated by these VFs as *Flow Matching from Gaussian (FM)* CNF.

Optimal Transport Conditional VFs between marginal paths In recent work [14] authors introduce conditional probability paths over joint distribution $q(\mathbf{x}_0, \mathbf{x_1})$ with marginals $q(\mathbf{x}_0)$ = $p_0(\mathbf{x}_0)$ and $q(\mathbf{x}_1)$ = $p_1(\mathbf{x}_1)$:

$$p_t(\mathbf{x}|\mathbf{x}_0, \mathbf{x}_1) = \mathcal{N}(\mathbf{x}|\mathbf{x}_1 t + (1-t)\mathbf{x}_0, \sigma^2 I). \tag{22}$$

They demonstrate that these PPs are generated with conditional VFs:

$$u_t(\mathbf{x}|\mathbf{x}_0, \mathbf{x}_1) = \mathbf{x}_1 - \mathbf{x}_0 \tag{23}$$

They demonstrate that conditional probability paths $p_t(\mathbf{x}|\mathbf{x}_0, \mathbf{x}_1)$, $u_t(\mathbf{x}|\mathbf{x}_0, \mathbf{x}_1)$ and joint distribution $q(\mathbf{x}_0, \mathbf{x_1})$ generate marginal probability paths with boundary conditions $p_1(\mathbf{x}) = q_1(\mathbf{x}) * \mathcal{N}(\mathbf{x}|0, \sigma^2 I)$ and $p_0(\mathbf{x}) = q_0(\mathbf{x}) * \mathcal{N}(\mathbf{x}|0, \sigma^2 I)$, where $*$ denotes convolution operation. As $\sigma \to 0$, the marginal VF $u_t(\mathbf{x})$ generates probability path that at $t = 1$ approach $q_1(\mathbf{x})$.

Furthermore, they introduce novel conditional flow function by defining joint marginal $q(\mathbf{x}_0, \mathbf{x_1})$ as 2-Wassestein optimal transport map π between marginals $q(\mathbf{x}_0)$ and $q(\mathbf{x}_1)$:

$$q(\mathbf{x}_0, \mathbf{x_1}) = \pi(\mathbf{x}_0, \mathbf{x_1}) \tag{24}$$

They demonstrate that by employing $p_t(\mathbf{x}|\mathbf{x}_0, \mathbf{x}_1)$ as defined in 22 and $u_t(\mathbf{x}|\mathbf{x}_0, \mathbf{x}_1)$ as in 23 with joint marginal $q(\mathbf{x}_0, \mathbf{x_1}) = \pi(\mathbf{x}_0, \mathbf{x_1})$, one can construct probability paths $p_t(\mathbf{x})$, that represent optimal transport map π between $p_0(\mathbf{x}_0) = q(\mathbf{x}_0)$ and $p_1(\mathbf{x}_1) = q(\mathbf{x}_1)$. Given the computational challenges of computing the transport plan π, the autors adopt the strategy wherein, for each batch of data $(\{\mathbf{x}_0^{(i)}\}_{i=1}^B, \{\mathbf{x}_1^{(i)}\}_{i=1}^B)$ they estimate minibatch approximation of π and sample pairs of points from the join distribution π_{batch} given by the OT plan between $\mathbf{x}_0$ and $\mathbf{x}_1$ pairs in the batch. When an entire dataset is processed, this approach computes the exact OT between source and target distributions. They demonstrate that the batch size can be much smaller than the dataset size for this approach to reach good performance. We refer to CNF models utilizing VFs as *Optimal Transport Conditional Flow Matching (OT-CFM)* CNFs.

Conditional Sampling of Continuous Normalizing Flows We train Continuous Normalizing Flows for generation of samples from $\mathbf{x} \sim q(\mathbf{x}_1)$ that satisfy $\mathcal{A}(\mathbf{x}) = \mathbf{y}$ by regressing $v_t(\mathbf{x}, \mathcal{A}, \mathbf{y}, \theta)$ over conditional vector field $u_t(\mathbf{x}|\mathcal{A}(\mathbf{x}) = \mathbf{y})$ that generates probability path conditioned on $\mathcal{A}(\mathbf{x}) = \mathbf{y}$, $p_t(\mathbf{x}|\mathcal{A}(\mathbf{x}) = \mathbf{y})$. Consequently, we define the new *CFM for Conditional Generation (CFMCG)* objective as follows:

$$\mathcal{L}_{\text{CFMCG}}(\theta) = \mathbf{E}_{t\sim U[0,1],\mathbf{y}\sim p(\mathcal{A},\mathbf{x}_1),\mathcal{A}\sim p,\mathbf{x}_1\sim q(\mathbf{x}_1|\mathcal{A},\mathbf{y}),\mathbf{x}\sim p_t(\mathbf{x}|\mathbf{x}_1)} \left\| v_t(\mathbf{x}, \mathcal{A}, \mathbf{y}, \theta) - u_t(\mathbf{x}|\mathbf{x}_1, \mathcal{A}, \mathbf{y}) \right\|^2, \tag{25}$$

Similarly to amortised learning for conditional sampling of diffusion models, this approach utilizes a single neural network to learn multiple vector fields that generate probability paths conditioned on events $\mathcal{A}(\mathbf{x}_1) = \mathbf{y}$.

3 Methodology and Experimental Setting

In this research, we evaluate diffusion models, representing score-based generative models, and Continuous Normalizing Flows (CNFs) utilizing both Flow Matching (FM) and Optimal Transport Conditional Flow Matching (OT-CFM) vector fields. Our focus is on assessing their performance on Image Inpainting and Image Super Resolution inverse problems. To fairly assess these models, we adopt the same neural network architecture and the consistent training and testing setting across all models. Our approach aligns closely with the experimental framework used in [3] for amortized conditioning on the image outpainting task.

We utilized MNIST [6] and FLOWERS dataset [10], where images from FLOWERS were centrally cropped to 64×64 pixels. For the image inpainting task, we use a mask of black pixels of size $K = 20$ for the FLOWERS dataset while for MNIST we use a mask of size $K = 14$. We set the value of the mask to -2 where the mask is 1, while for pixels for which the mask has value 0, we keep the original pixel value. For the image super resolution task we downsample FLOWER images to size 16×16 while MNIST images are downsampled from 28×28 to 7×7.

3.1 Model Training and Inference

Our diffusion models are developed by discretizing the Ornstein-Uhlenbeck (OU) process in section 2.2. We adopt the linear β-schedule with $\beta_0 = 10^{-4}$ and $\beta_N = 2 \cdot 10^{-2}$ and $N = 1000$ diffusion time steps. We employ an amortized h-transform for conditioning the model and the neural network $\mathbf{s}_\theta$ is trained to predict the added noise at each time step from image $\mathbf{x}_t$ at time step t and sampled $\mathcal{A}$ and $\mathbf{y}$ by minimizing the loss as in [3]:

$$\theta^* = \arg\min_\theta \mathbb{E}_{t,\mathbf{x}_0\sim p_{\text{data}},y\sim p(\mathcal{A},\mathbf{x}_\prime,\mathcal{A}\sim p,\epsilon} \left[\|\mathbf{s}_\theta(t, \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}, \mathbf{y}, \mathcal{A}) - \epsilon\|^2 \right], \tag{26}$$

where $\epsilon \sim \mathcal{N}(0, I)$. The model is conditioned in 90% of the training set, while in the remaining 10%, a replica of the original sample is provided. We explore two discretization strategies for VP-SDEs. The first is the discrete-time DDPM as defined by [Ho et al., 2020], which we refer to as AM-DDPM (Amortized Conditioning with Discrete-Time DDPM). The second model is obtained by applying the Euler-Maruyama discretization of the reverse SDE [13]. We label this approach as AM-EM Amortized Conditioning with Euler-Maruyama Discretization) Both of these discretization methods and their corresponding sampling procedures are provided in Algorithms 1 and 2.

We explore CFM models with FM and OT-CFM vector fields FM and OT-CFM, as they represent state-of-the-art Flow models and we set $\sigma_{\text{min}} = 0$. We train a neural network to minimize novel CFMCG objective defined in equation 25. We maintain the same batch size and number of epochs as in the diffusion model training to ensure a fair comparison. For inference, we employ both adaptive `dopir5` ODE solver and standard Euler Integration with 100 and 1000 steps.

Our models were trained for 10 and 100 epochs on MNIST and FLOWERS datasets respectively. We evaluated MNIST models twice per epoch, resulting in 20 total evaluations, while models trained on flowers were evaluated after every ten epochs, resulting in ten total evaluations. Both models were evaluated on 96 test images in each evaluation.

3.2 Model Configuration

We use the same UNET architecture for both noise ϵ_θ and VF v_θ models. Model configuration for the FLOWERS dataset consists of four downsampling blocks consisting of 2d convolutional layers of dimensionality 128, 256, 384, and 512, respectively, while the MNIST configuration contains three downsampling blocks with 2d convolutional layers of dimensionality 32, 64, and 64. Attention is applied to the middle layers of the UNet with four heads. SiLU activation function, without dropout and group normalization layers. This architecture takes concatenated images and conditional images with the same dimension. Model configuration for the FLOWERS dataset contains 68.159M parameters, while MNIST configuration contains 1.225 M parameters.

3.3 Performance Metrics

We evaluate the performance of models by employing two metrics: mean squared error (MSE) and the perceptual metric LPIPS between the original image (from which a patch was extracted) and the generated conditional sample. Additionally, for image super resolution task images, we also provide evaluations using the peak signal-to-noise ratio (PSNR) and structural similarity index measure (SSIM) as in [12]. Additionally, we investigate the number of neural network (function) evaluations (NFE) used in adaptive ODE solvers for both VFs. In our tables, we report mean and standard deviation of all metrics for evaluation over 96 test images after final epoch.

4 Results

4.1 MNIST

The results of models in tables 1 and 2 show that AM-DDPM model clearly outperformed other models in both tasks. Contrary to findings by [song ref] which find that Euler-Maruyama discretization improves the model performance, our evaluations reveal a noticeable under performance of this model when compared to AM-DDPM. We observe that FM model achieves second best performance in both tasks with significantly less neural network evaluations than AM-DDPM. Notably, the FM variant with adaptive solver (referred to as FM in our tables and figures) achieves more robust results than Euler Integration that exhibits significantly worse results in super-resolution task for both FM and OT-CFM models.

Figure 1 reveals that CNF models have much faster training convergence rate than Diffusion models, achieving low MSE and LPIPS even after first epoch and that additional training does not significantly improve model's performance. In contrast, diffusion models exhibit a more progressive loss reduction with Am-DDPM surpassing all CNF models before the fifth training epoch.

While OT-CFM model achieves good results in super-resolution task, it performers significantly worse than FM models in inpainting task. This suggests a potential sensitivity to minibatch estimations of the optimal transport map π. It suggests that larger batch sizes may be necessary for improving the robustness of these models. Furthermore, Figure 2 highlights that OT-CFM necessitates a substantially higher number of function evaluations for inpainting. Finally, in super-resolution task we observe that CNF models with adaptive ODE solver achieve significantly better results than the same models with Euler integration supporting the claim that adaptive solvers provide more robust samples.

4.2 FLOWERS

Informed by the robustness and good performance of adaptive ODE solver in MNIST experiments, in the following experiment we focus on FM and OT-CFM Models, including the same diffusion models.

Visual inspection of test samples for inpainting 6b shows that while models generally preserved the integrity of non-masked image areas, they uniformly colored the masked patch rather than accurately reconstructing it, expect from the OT-CFM model that simplified the patterns and colors and produced

Table 1: Quantitative results on **image image inpainting task** on **MNIST** dataset. Green represents the top result, Red represents the second-best result, and Blue represents the third-best result.

Model	MSE	LPIPS	NFE
AM-DDPM	$0.11_{\pm 0.07}$	$0.06_{\pm 0.03}$	$1000_{\pm 0}$
AM-EM	$0.16_{\pm 0.07}$	$0.17_{\pm 0.06}$	$1000_{\pm 0}$
FM	$0.15_{\pm 0.06}$	$0.2_{\pm 0.04}$	$50.0_{\pm 0.0}$
OT-CFM	$0.51_{\pm 0.15}$	$0.27_{\pm 0.06}$	$130.0_{\pm 3.46}$
FM_100	$0.15_{\pm 0.07}$	$0.16_{\pm 0.05}$	$100_{\pm 0.0}$
FM_1000	$0.16_{\pm 0.06}$	$0.17_{\pm 0.04}$	$1000_{\pm 0.0}$
OT-CFM_1000	$0.52_{\pm 0.16}$	$0.27_{\pm 0.06}$	$1000_{\pm 0.0}$
OT-CFM_100	$0.49_{\pm 0.14}$	$0.26_{\pm 0.06}$	$100_{\pm 0.0}$

Table 2: Quantitative results on **image super-resolution task** on **MNIST** dataset. Green represents the top result, Red represents the second-best result, and Blue represents the third-best result.

Model	MSE	LPIPS	PSNR	SSIM	NFE
AM-DDPM	$0.08_{\pm 0.04}$	$0.06_{\pm 0.02}$	$17.93_{\pm 2.67}$	$0.8_{\pm 0.09}$	$1000_{\pm 0}$
AM-EM	$0.16_{\pm 0.05}$	$0.21_{\pm 0.06}$	$14.16_{\pm 1.38}$	$0.49_{\pm 0.11}$	$1000_{\pm 0}$
FM	$0.1_{\pm 0.05}$	$0.1_{\pm 0.04}$	$16.66_{\pm 2.07}$	$0.71_{\pm 0.09}$	$56.0_{\pm 10.39}$
OT-CFM	$0.1_{\pm 0.05}$	$0.1_{\pm 0.04}$	$16.34_{\pm 1.99}$	$0.7_{\pm 0.09}$	$56.0_{\pm 10.39}$
OT-CFM	$0.52_{\pm 0.15}$	$0.27_{\pm 0.06}$	$9.02_{\pm 1.41}$	$0.2_{\pm 0.11}$	$100_{\pm 0}$
FM_1000	$0.91_{\pm 0.11}$	$0.48_{\pm 0.06}$	$6.46_{\pm 0.56}$	$0.02_{\pm 0.06}$	$1000_{\pm 0}$
OT-CFM_100	$0.52_{\pm 0.15}$	$0.27_{\pm 0.05}$	$9.0_{\pm 1.25}$	$0.2_{\pm 0.11}$	$100_{\pm 0}$
FM_100	$0.89_{\pm 0.13}$	$0.5_{\pm 0.05}$	$6.58_{\pm 0.64}$	$0.02_{\pm 0.06}$	$100_{\pm 0}$

images that do not match the given samples, thus failing in this task. AM-DDPM achieved highest performance, closely followed by FM, which requires only an average of 26 function evaluations per test sample. According to Figure 3, we observe a similar trend of rapid convergence across all models, with minimal loss achieved early in the training process.

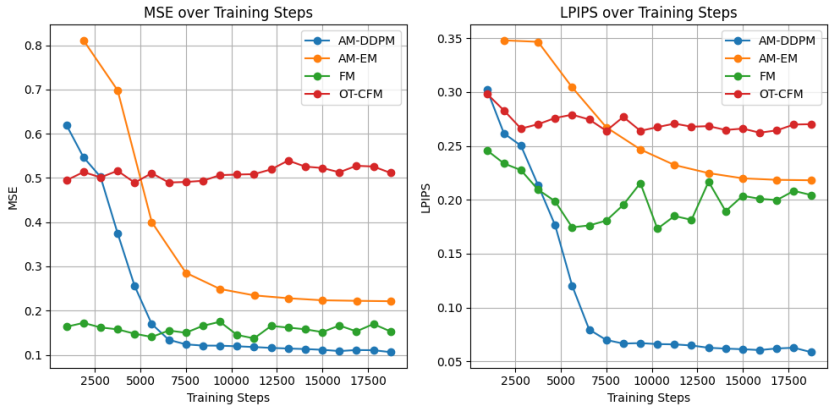
The super-resolution task yielded improved outcomes, with all models, expect for OT-CFM, producing samples that more faithfully resemble the original images. AM-DDPM, AM-EM and FM achieve similar results in visual image quality, supported by the almost the same value of LPIPS for all models. Remarkably, the FM model achieved these competitive results with highest SSIM and with an average of only 58 function evaluations. We observe similar early training convergence trend as in the inpainting task and we hypothesise that larger number of Diffusion steps would potentially improve the performance of Diffusion models and achieve more gradual loss reduction during training.

Table 3: Quantitative results on image super resolution task on FLOWERS. Green represents the top result, Red represents the second-best result, and Blue represents the third-best result.

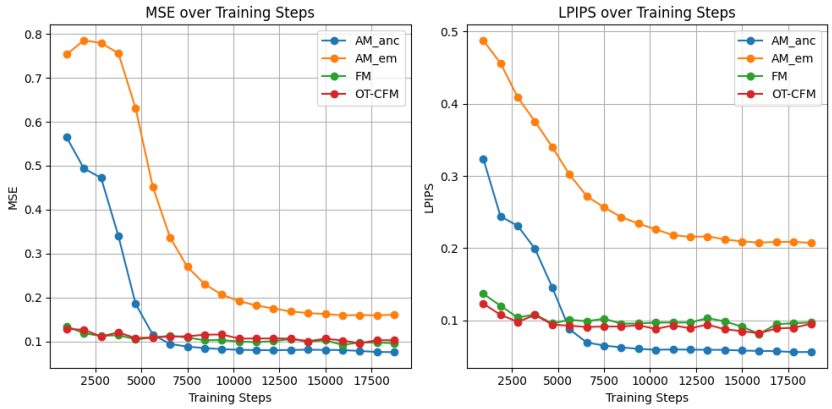
Model	MSE	LPIPS	NFE
AM-DDPM	$0.07_{\pm 0.07}$	$0.15_{\pm 0.06}$	$1000_{\pm 0}$
AM-EM	$0.12_{\pm 0.11}$	$0.26_{\pm 0.08}$	$1000_{\pm 0}$
FM	$0.07_{\pm 0.02}$	$0.3_{\pm 0.07}$	$26.0_{\pm 0.0}$
OT-CFM	$0.47_{\pm 0.16}$	$0.69_{\pm 0.06}$	$60.0_{\pm 3.46}$

5 Conclusion

In this study, we conducted an empirical comparison of two novel methods for conditional image generation: diffusion models conditioned with the theoretically grounded amortized h-transform and Continuous Normalizing Flow (CNF) models trained with Flow Matching (FM) vector fields. Our findings reveal that, across all tasks and datasets, Denoising Diffusion Probabilistic Models (DDPM) with amortized h-transform consistently achieved superior results. However, the CFM models with Optimal Transport Vector Fields between conditional paths (FM) demonstrated slightly lower performance but require significantly smaller number of neural network evaluations during



(a) Image Inpainting Task.



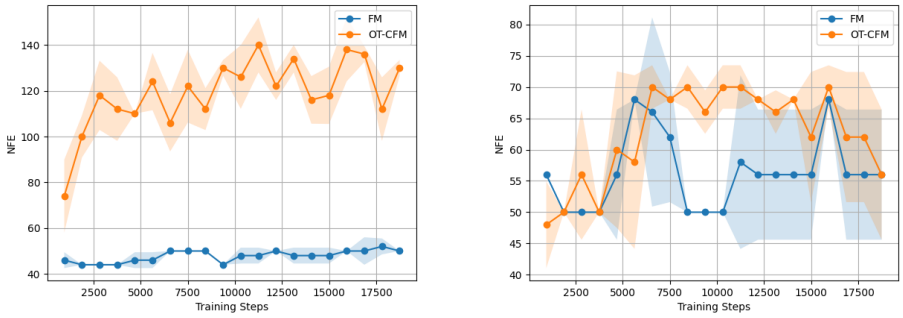
(b) Super Resolution Task.

Figure 1: Performance on MNIST: Evolution of MSE - left, and LPIPS - right, during training across AM-DDPM, AM-EM, FM, and OT-CFM models on the test set.

Table 4: Quantitative results on **image super-resolution task** on **FLOWERS** dataset. Green represents the top result, Red represents the second-best result, and Blue represents the third-best result.

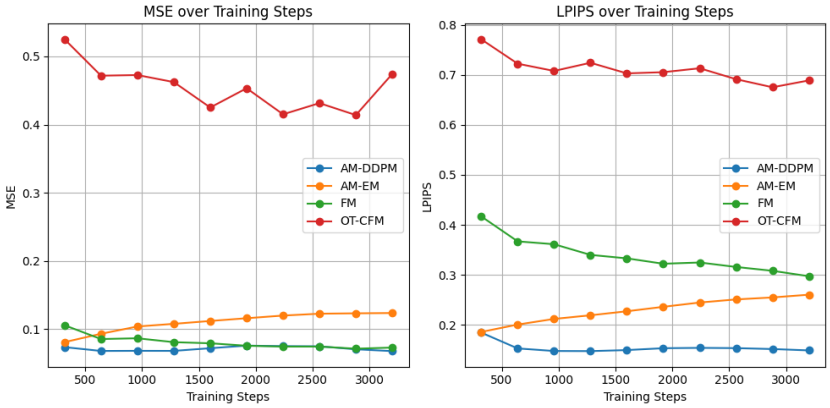
Model	MSE	LPIPS	PSNR	SSIM	NFE
AM-DDPM	$0.06_{\pm 0.04}$	$0.34_{\pm 0.07}$	$18.14_{\pm 2.36}$	$0.47_{\pm 0.16}$	$1000_{\pm 0}$
AM-EM	$0.11_{\pm 0.07}$	$0.34_{\pm 0.06}$	$15.61_{\pm 3.13}$	$0.33_{\pm 0.14}$	$1000_{\pm 0}$
FM	$0.1_{\pm 0.03}$	$0.33_{\pm 0.06}$	$15.57_{\pm 1.58}$	$0.51_{\pm 0.09}$	$58.0_{\pm 3.46}$
OT-CFM	$0.5_{\pm 0.13}$	$0.69_{\pm 0.08}$	$8.43_{\pm 1.15}$	$0.03_{\pm 0.03}$	$64.0_{\pm 3.46}$

inference. These models also exhibited faster, or comparable, training convergence rates relative to the DDPMs. This comparative study is first to compare and explore these novel methods in tasks of image inpainting and super-resolution. Future research will aim to scale up these approaches by applying them to more complex models and challenging tasks and extending training over greater number of diffusion steps, to provide more detailed evaluation and insights on the effectiveness of these models for conditional image generation.

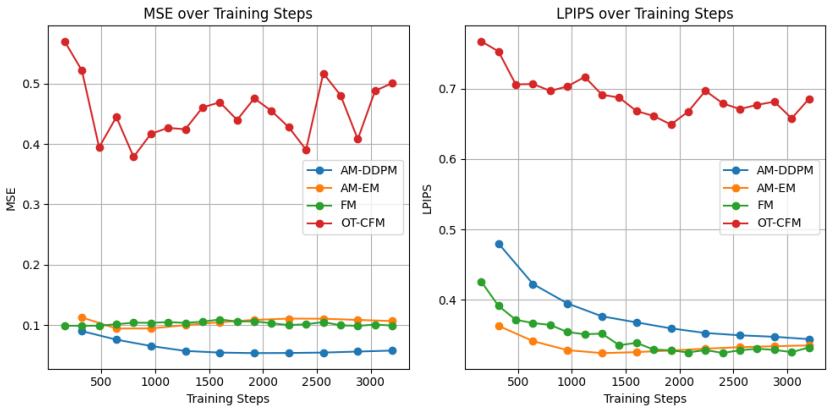


(a) Image Inpainting Task. (b) Super Resolution Task.

Figure 2: Number of Function Evaluations (NFE), on MNIST,for the FM and OT-CFM models on the test set during training: mean $\pm$ std.

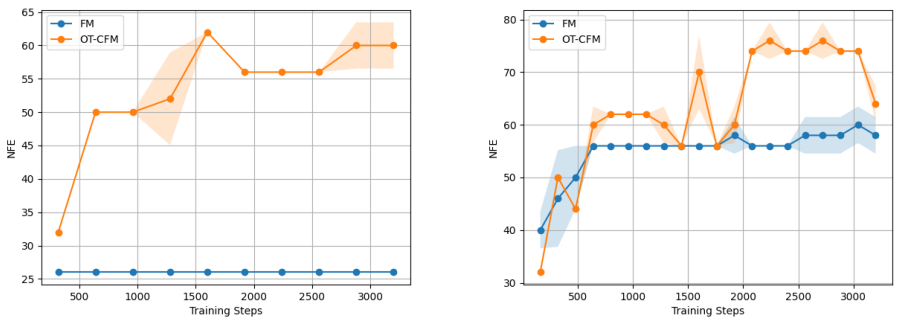


(a) Image Inpainting Task.



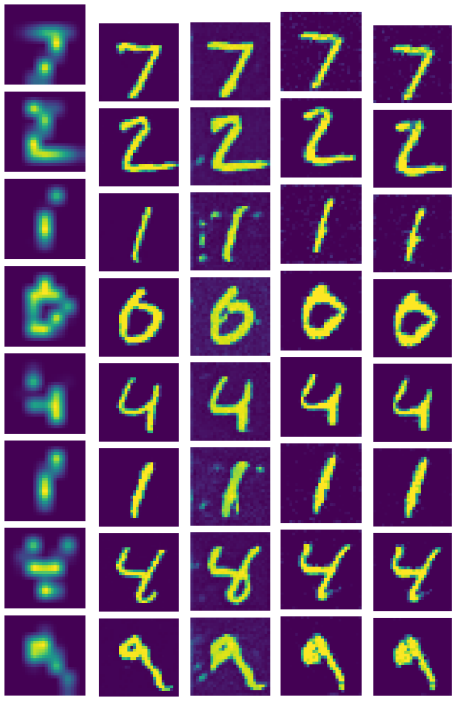
(b) Super Resolution Task.

Figure 3: Performance on FLOWERS: Evolution of MSE - left, and LPIPS - right, during training across AM-DDPM, AM-EM, FM, and OT-CFM models on the test set.

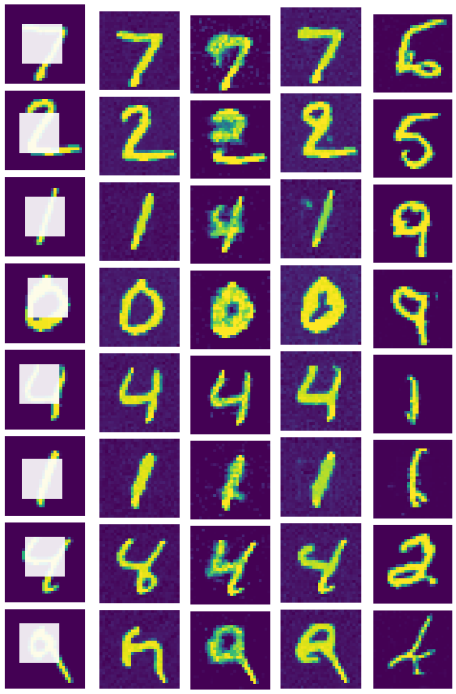


(a) Image Inpainting Task. (b) Super Resolution Task.

Figure 4: Number of Function Evaluations (NFE), on FLOWERS, for the FM and OT-CFM models on the test set during training: mean $\pm$ std.

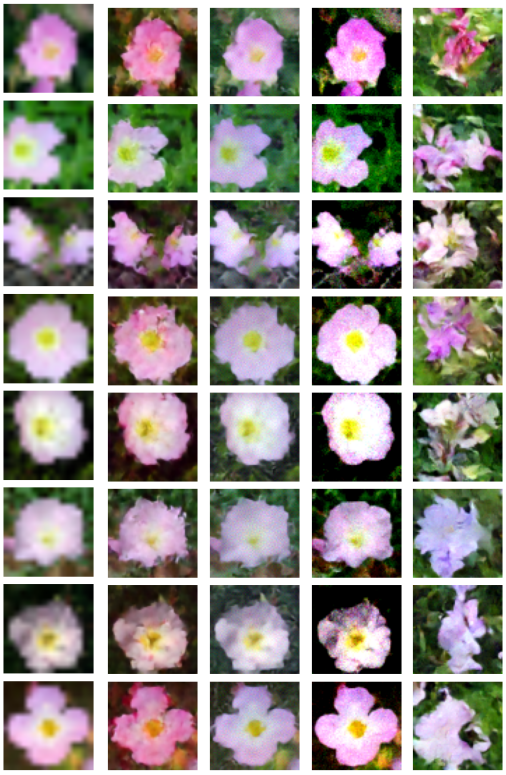


(a) Super Resolution test set samples.



(b) Inpainting test set samples.

Figure 5: Comparative Visualization of Super Resolution and Inpainting Tasks on the MNIST Dataset. From left to right: the input condition, followed by the outputs of AM-DDPM, AM-EM, FM, and OT-CFM models after the final training epoch.



(a) Super Resolution test set samples.

(b) Inpainting test set samples.

Figure 6: Comparative Visualization of Super Resolution and Inpainting Tasks on the MNIST Dataset. From left to right: the input condition, followed by the outputs of AM-DDPM, AM-EM, FM, and OT-CFM models after the final training epoch.

References

[1] Michael S. Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants, 2023.

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

[3] Kieran Didi, Francisco Vargas, Simon V Mathis, Vincent Dutordoir, Emile Mathieu, Urszula J Komorowska, and Pietro Lio. A framework for conditional diffusion modelling with applications in motif scaffolding for protein design, 2024.

[4] Jeremy Heng, Valentin De Bortoli, Arnaud Doucet, and James Thornton. Simulating diffusion bridges with score matching, 2022.

[5] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020.

[6] Yann LeCun, Corinna Cortes, and Christopher J.C. Burges. MNIST handwritten digit database. 2010.

[7] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling, 2023.

[8] Qiang Liu. Rectified flow: A marginal preserving approach to optimal transport, 2022.

[9] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models, 2022.

[10] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Indian Conference on Computer Vision, Graphics, and Image Processing*, December 2008.

[11] L. Chris G. Rogers and David Williams. *Diffusions, Markov Processes and Martingales: Volume 2, Itô Calculus*, volume 2. Cambridge University Press, 2000.

[12] Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J. Fleet, and Mohammad Norouzi. Image super-resolution via iterative refinement, 2021.

[13] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *CoRR*, abs/2011.13456, 2020.

[14] Alexander Tong, Kilian Fatras, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport, 2024.