

# Diffusion Models in Vision: A Survey

Florinel-Alin Croitoru , Vlad Hondu , Radu Tudor Ionescu , Member, IEEE,  
and Mubarak Shah , Life Fellow, IEEE

(Survey Paper)

**Abstract**—Denoising diffusion models represent a recent emerging topic in computer vision, demonstrating remarkable results in the area of generative modeling. A diffusion model is a deep generative model that is based on two stages, a forward diffusion stage and a reverse diffusion stage. In the forward diffusion stage, the input data is gradually perturbed over several steps by adding Gaussian noise. In the reverse stage, a model is tasked at recovering the original input data by learning to gradually reverse the diffusion process, step by step. Diffusion models are widely appreciated for the quality and diversity of the generated samples, despite their known computational burdens, i.e., low speeds due to the high number of steps involved during sampling. In this survey, we provide a comprehensive review of articles on denoising diffusion models applied in vision, comprising both theoretical and practical contributions in the field. First, we identify and present three generic diffusion modeling frameworks, which are based on denoising diffusion probabilistic models, noise conditioned score networks, and stochastic differential equations. We further discuss the relations between diffusion models and other deep generative models, including variational auto-encoders, generative adversarial networks, energy-based models, autoregressive models and normalizing flows. Then, we introduce a multi-perspective categorization of diffusion models applied in computer vision. Finally, we illustrate the current limitations of diffusion models and envision some interesting directions for future research.

**Index Terms**—Denoising diffusion models, deep generative modeling, diffusion models, image generation, noise conditioned score networks, score-based models.

## I. INTRODUCTION

DIFFUSION models [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11] form a category of deep generative models which has recently become one of the hottest topics in computer vision (see Fig. 1), showcasing impressive generative capabilities, ranging from the high level of details to the diversity of the generated examples. We can even go as far as stating that

Manuscript received 10 September 2022; revised 31 December 2022; accepted 23 March 2023. Date of publication 27 March 2023; date of current version 4 August 2023. This work was supported by a grant of the Romanian Ministry of Education and Research, CNCS-UEFISCDI, under Grants PN-III-P2-2.1-PED-2021-0195 and 690/2022, within PNCDI III. (*Florinel-Alin Croitoru and Vlad Hondu contributed equally to this work.*) Recommended for acceptance by Y. J. Lee. (*Corresponding author: Radu Tudor Ionescu.*)

Florinel-Alin Croitoru, Vlad Hondu, and Radu Tudor Ionescu are with the Department of Computer Science, University of Bucharest, 030018 Bucharest, Romania (e-mail: alincroitoru97@gmail.com; vlad.hondu25@gmail.com; radu.ionescu@gmail.com).

Mubarak Shah is with the Center for Research in Computer Vision (CRCV), Department of Computer Science, University of Central Florida, Orlando, FL 32816 USA (e-mail: shah@crcv.ucf.edu).

Digital Object Identifier 10.1109/TPAMI.2023.3261988

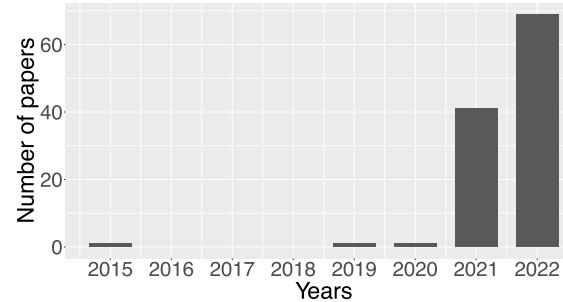


Fig. 1. The rough number of papers on diffusion models per year.

these generative models raised the bar to a new level in the area of generative modeling, particularly referring to models such as Imagen [12] and Latent Diffusion Models (LDMs) [10]. This statement is confirmed by the image samples illustrated in Fig. 2, which are generated by Stable Diffusion, a version of LDMs [10] that generates images based on text prompts. The generated images exhibit very few artifacts and are very well aligned with the text prompts. Notably, the prompts are purposely chosen to represent unrealistic scenarios (never seen at training time), thus demonstrating the high generalization capacity of diffusion models.

To date, diffusion models have been applied to a wide variety of generative modeling tasks, such as image generation [1], [2], [3], [4], [5], [6], [7], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], image super-resolution [10], [23], [24], [25], [26], [27], image inpainting [1], [3], [4], [10], [24], [26], [28], [29], [30], image editing [31], [32], [33], image-to-image translation [32], [34], [35], [36], [37], [38], among others. Moreover, the latent representation learned by diffusion models was also found to be useful in discriminative tasks, e.g., image segmentation [39], [40], [41], [42], classification [43] and anomaly detection [44], [45], [46]. This confirms the broad applicability of denoising diffusion models, indicating that further applications are yet to be discovered. Additionally, the ability to learn strong latent representations creates a connection to representation learning [47], [48], a comprehensive domain that studies ways to learn powerful data representations, covering multiple approaches ranging from the design of novel neural architectures [49], [50], [51], [52] to the development of learning strategies [53], [54], [55], [56], [57], [58].

According to the graph shown in Fig. 1, the number of papers on diffusion models is growing at a very fast pace. To outline the

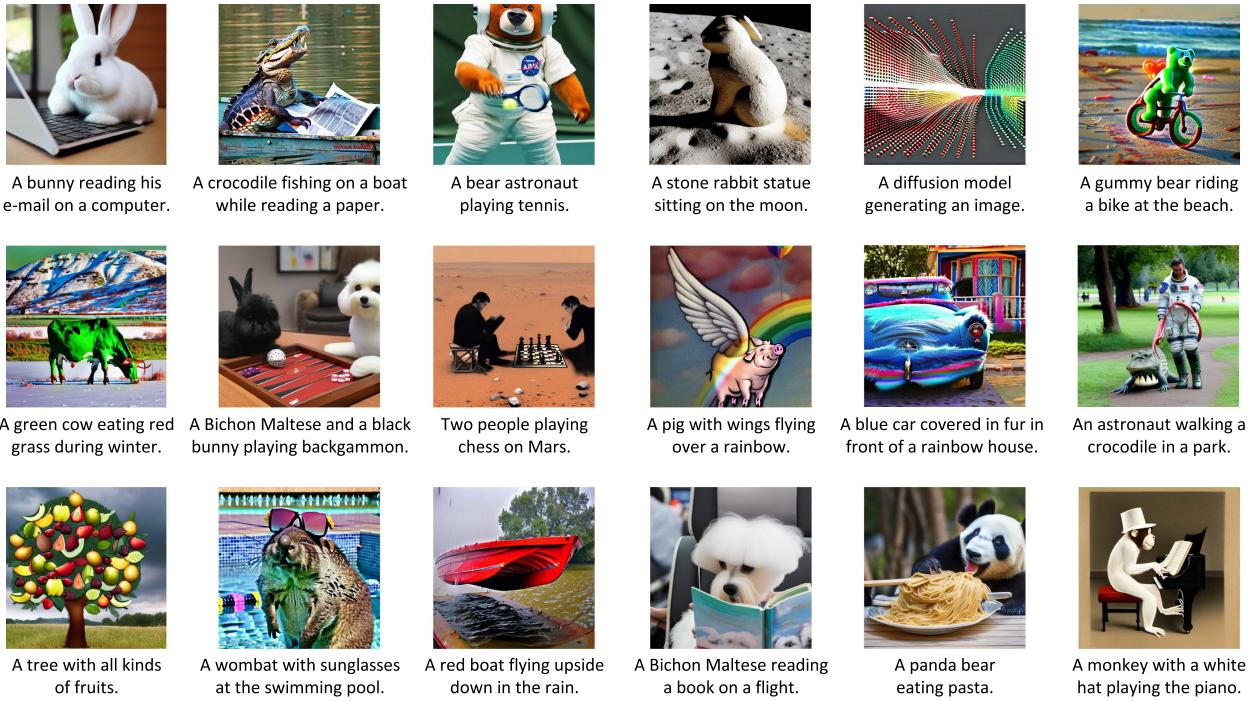


Fig. 2. Images generated by Stable Diffusion [10] based on various text prompts, via the <https://beta.dreamstudio.ai/dream> platform.

past and current achievements of this rapidly developing topic, we present a comprehensive review of articles on denoising diffusion models in computer vision. More precisely, we survey articles that fall in the category of generative models defined below. *Diffusion models* represent a category of deep generative models that are based on (i) a forward diffusion stage, in which the input data is gradually perturbed over several steps by adding Gaussian noise, and (ii) a reverse (backward) diffusion stage, in which a generative model is tasked at recovering the original input data from the diffused (noisy) data by learning to gradually reverse the diffusion process, step by step.

We underline that there are at least three sub-categories of diffusion models that comply with the above definition. The first sub-category comprises denoising diffusion probabilistic models (DDPMs) [1], [2], which are inspired by the non-equilibrium thermodynamics theory. DDPMs are latent variable models that employ latent variables to estimate the probability distribution. From this point of view, DDPMs can be viewed as a special kind of variational auto-encoders (VAEs) [50], where the forward diffusion stage corresponds to the encoding process inside VAE, while the reverse diffusion stage corresponds to the decoding process. The second sub-category is represented by noise conditioned score networks (NCSNs) [3], which are based on training a shared neural network via score matching to estimate the score function (defined as the gradient of the log density) of the perturbed data distribution at different noise levels. Stochastic differential equations (SDEs) [4] represent an alternative way to model diffusion, forming the third sub-category of diffusion models. Modeling diffusion via forward and reverse SDEs leads to efficient generation strategies as well as strong theoretical results [59]. This latter formulation (based

on SDEs) can be viewed as a generalization over DDPMs and NCSNs.

We identify several defining design choices and synthesize them into three generic diffusion modeling frameworks corresponding to the three sub-categories introduced above. To put the generic diffusion modeling framework into context, we further discuss the relations between diffusion models and other deep generative models. More specifically, we describe the relations to variational auto-encoders (VAEs) [50], generative adversarial networks (GANs) [52], energy-based models (EBMs) [60], [61], autoregressive models [62] and normalizing flows [63], [64]. Then, we introduce a multi-perspective categorization of diffusion models applied in computer vision, classifying the existing models based on several criteria, such as the underlying framework, the target task, or the denoising condition. Finally, we illustrate the current limitations of diffusion models and envision some interesting directions for future research. For example, perhaps one of the most problematic limitations is the poor time efficiency during inference, which is caused by a very high number of evaluation steps, e.g., thousands, to generate a sample [2]. Naturally, overcoming this limitation without compromising the quality of the generated samples represents an important direction for future research.

In summary, our contribution is twofold:

- Since many contributions based on diffusion models have recently emerged in vision, we provide a comprehensive and timely literature review of denoising diffusion models applied in computer vision, aiming to provide a fast understanding of the generic diffusion modeling framework to our readers.

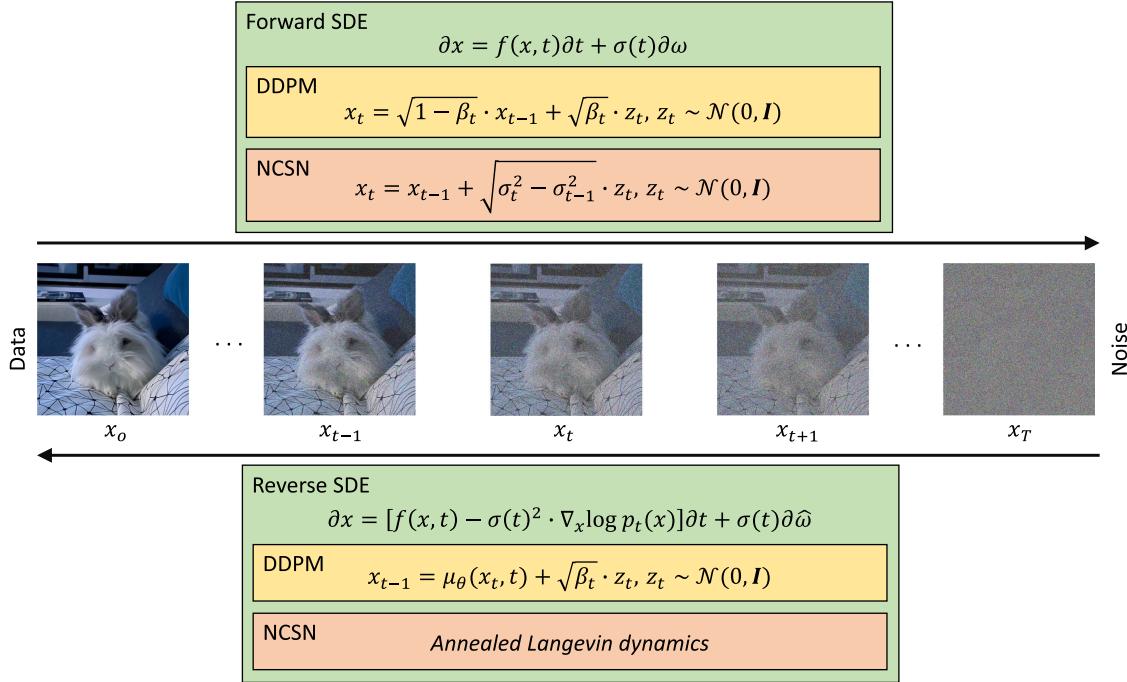


Fig. 3. A generic framework composing three alternative formulations of diffusion models based on: stochastic differential equations (SDEs), denoising diffusion probabilistic models (DDPMs) and noise conditioned score networks (NCSNs). In general, a diffusion model consists of two processes. The first one, called the forward process, transforms data into noise, while the second one is a generative process that reverses the effect of the forward process. This latter process learns to transform the noise back into data. We illustrate these processes for all three formulations. The forward SDE shows that a change over time in  $x$  is modeled by a function  $f$  plus a stochastic component  $\partial\omega \sim \mathcal{N}(0, \partial t)$  scaled by  $\sigma(t)$ . We underline that different choices of  $f$  and  $\sigma$  will lead to different diffusion processes. This is why the SDE formulation is a generalization of the other two. The reverse (generative) SDE shows how to change  $x$  in order to recover the data from pure noise. We keep the random component and modify the deterministic one using the gradients of the log probability  $\nabla_x \log p_t(x)$ , so that  $x$  moves to regions where the data density  $p(x)$  is high. DDPMs sample the data points during the forward process from a normal distribution  $\mathcal{N}(x_t; \sqrt{1-\beta_t} \cdot x_{t-1}, \beta_t \cdot \mathbf{I})$ , where  $\beta_t \ll 1$ . This iterative sampling slowly destroys information in data, and replaces it with Gaussian noise. The sampling is illustrated via the reparametrization trick (see details in Section II-A). The reverse process of DDPM also performs iterative sampling from a normal distribution, but the mean  $\mu_\theta(x_t, t)$  of the distribution is derived by subtracting the noise, estimated by a neural network, from the image at the previous step  $x_t$ . The variance is equal to the one used in the forward process. The initial image going into the reverse process contains only Gaussian noise. The forward process of NCSN simply adds normal noise to the image at the previous step. This can also be seen as sampling from a normal distribution  $\mathcal{N}(x_t; x_{t-1}, (\sigma_t^2 - \sigma_{t-1}^2) \cdot \mathbf{I})$ , with the mean being the image at the previous step. The reverse process of NCSN is based on an algorithm described in Section II-B. Best viewed in color.

- We devise a multi-perspective categorization of diffusion models, aiming to help other researchers working on diffusion models applied to a specific domain in quickly finding relevant related works in the respective domain.

## II. GENERIC FRAMEWORK

Diffusion models are a class of probabilistic generative models that learn to reverse a process that gradually degrades the training data structure. Thus, the training procedure involves two phases: the forward diffusion process and the backward denoising process.

The former phase consists of multiple steps in which low-level noise is added to each input image, where the scale of the noise varies at each step. The training data is progressively destroyed until it results in pure Gaussian noise.

The latter phase is represented by reversing the forward diffusion process. The same iterative procedure is employed, but backwards: the noise is sequentially removed, and hence, the original image is recreated. Therefore, at inference time, images are generated by gradually reconstructing them starting from random white noise. The noise subtracted at each

time step is estimated via a neural network, typically based on a U-Net architecture [65], allowing the preservation of dimensions.

In the following three subsections, we present three formulations of diffusion models, namely denoising diffusion probabilistic models, noise conditioned score networks, and the approach based on stochastic differential equations that generalizes over the first two methods. For each formulation, we describe the process of adding noise to the data, the method which learns to reverse this process, and how new samples are generated at inference time. In Fig. 3, all three formulations are illustrated as a generic framework. We dedicate the last subsection to discussing connections to other deep generative models.

### A. Denoising Diffusion Probabilistic Models (DDPMs)

**Forward Process.** DDPMs [1], [2] slowly corrupt the training data using Gaussian noise. Let  $p(x_0)$  be the data density, where the index 0 denotes the fact that the data is uncorrupted (original). Given an uncorrupted training sample  $x_0 \sim p(x_0)$ , the noised versions  $x_1, x_2, \dots, x_T$  are obtained according to the following

Markovian process:

$$p(x_t|x_{t-1}) = \mathcal{N}\left(x_t; \sqrt{1-\beta_t} \cdot x_{t-1}, \beta_t \cdot \mathbf{I}\right), \forall t \in \{1, \dots, T\}, \quad (1)$$

where  $T$  is the number of diffusion steps,  $\beta_1, \dots, \beta_T \in [0, 1]$  are hyperparameters representing the variance schedule across diffusion steps,  $\mathbf{I}$  is the identity matrix having the same dimensions as the input image  $x_0$ , and  $\mathcal{N}(x; \mu, \sigma)$  represents the normal distribution of mean  $\mu$  and covariance  $\sigma$  that produces  $x$ . An important property of this recursive formulation is that it also allows the direct sampling of  $x_t$ , when  $t$  is drawn from a uniform distribution, i.e.,  $\forall t \sim \mathcal{U}(\{1, \dots, T\})$ :

$$p(x_t|x_0) = \mathcal{N}\left(x_t; \sqrt{\hat{\beta}_t} \cdot x_0, (1 - \hat{\beta}_t) \cdot \mathbf{I}\right), \quad (2)$$

where  $\hat{\beta}_t = \prod_{i=1}^t \alpha_i$  and  $\alpha_t = 1 - \beta_t$ . Essentially, (2) shows that we can sample any noisy version  $x_t$  via a single step, if we have the original image  $x_0$  and fix a variance schedule  $\beta_t$ .

The sampling from  $p(x_t|x_0)$  is performed via a reparametrization trick. In general, to standardize a sample  $x$  of a normal distribution  $x \sim \mathcal{N}(\mu, \sigma^2 \cdot \mathbf{I})$ , we subtract the mean  $\mu$  and divide by the standard deviation  $\sigma$ , resulting in a sample  $z = \frac{x-\mu}{\sigma}$  of the standard normal distribution  $z \sim \mathcal{N}(0, \mathbf{I})$ . The reparametrization trick does the inverse of this operation, starting with  $z$  and yielding the sample  $x$  by multiplying  $z$  with the standard deviation  $\sigma$  and adding the mean  $\mu$ . If we translate this process to our case, then  $x_t$  is sampled from  $p(x_t|x_0)$  as follows:

$$x_t = \sqrt{\hat{\beta}_t} \cdot x_0 + \sqrt{(1 - \hat{\beta}_t)} \cdot z_t, \quad (3)$$

where  $z_t \sim \mathcal{N}(0, \mathbf{I})$ .

*Properties of  $\beta_t$ .* If the variance schedule  $(\beta_t)_{t=1}^T$  is chosen such that  $\hat{\beta}_T \rightarrow 0$ , then, according to (2), the distribution of  $x_T$  should be well approximated by the standard Gaussian distribution  $\pi(x_T) = \mathcal{N}(0, \mathbf{I})$ . Moreover, if each  $(\beta_t)_{t=1}^T \ll 1$ , then the reverse steps  $p(x_{t-1}|x_t)$  have the same functional form as the forward process  $p(x_t|x_{t-1})$  [1], [66]. Intuitively, the last statement is true when  $x_t$  is created with a very small step, as it becomes more likely that  $x_{t-1}$  comes from a region close to where  $x_t$  is observed, which allows us to model this region with a Gaussian distribution. To conform to the aforementioned properties, Ho et al. [2] choose  $(\beta_t)_{t=1}^T$  to be linearly increasing constants between  $\beta_1 = 10^{-4}$  and  $\beta_T = 2 \cdot 10^{-2}$ , where  $T = 1000$ .

*Reverse Process.* By leveraging the above properties, we can generate new samples from  $p(x_0)$  if we start from a sample  $x_T \sim \mathcal{N}(0, \mathbf{I})$  and follow the reverse steps  $p(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu(x_t, t), \Sigma(x_t, t))$ . To approximate these steps, we can train a neural network  $p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t))$  that receives as input the noisy image  $x_t$  and the embedding at time step  $t$ , and learns to predict the mean  $\mu_\theta(x_t, t)$  and the covariance  $\Sigma_\theta(x_t, t)$ .

In an ideal scenario, we would train the neural network with a maximum likelihood objective such that the probability assigned by the model  $p_\theta(x_0)$  to each training example  $x_0$  is as large as possible. However,  $p_\theta(x_0)$  is intractable because we have to marginalize over all the possible reverse trajectories to

compute it. The solution to this problem [1], [2] is to minimize a variational lower-bound of the negative log-likelihood instead, which has the following formulation:

$$\begin{aligned} \mathcal{L}_{vlb} = & -\log p_\theta(x_0|x_1) + KL(p(x_T|x_0)\|\pi(x_T)) \\ & + \sum_{t>1} KL(p(x_{t-1}|x_t, x_0)\|p_\theta(x_{t-1}|x_t)), \end{aligned} \quad (4)$$

where  $KL$  denotes the Kullback-Leibler divergence between two probability distributions. Upon analyzing each component, we can see that the second term can be removed because it does not depend on  $\theta$ . The last term shows that the neural network is trained such that, at each time step  $t$ ,  $p_\theta(x_{t-1}|x_t)$  is as close as possible to the true posterior of the forward process when conditioned on the original image. Moreover, it can be proven that the posterior  $p(x_{t-1}|x_t, x_0)$  is a Gaussian distribution, implying closed-form expressions for the  $KL$  divergences.

Ho et al. [2] propose to fix the covariance  $\Sigma_\theta(x_t, t)$  to a constant value and rewrite the mean  $\mu_\theta(x_t, t)$  as a function of noise, as follows:

$$\mu_\theta = \frac{1}{\sqrt{\alpha_t}} \cdot \left( x_t - \frac{1 - \alpha_t}{\sqrt{1 - \hat{\beta}_t}} \cdot z_\theta(x_t, t) \right). \quad (5)$$

These simplifications unlocked a new formulation of the objective  $\mathcal{L}_{vlb}$ , which measures, for a random time step  $t$  of the forward process, the distance between the real noise  $z_t$  and the noise estimation  $z_\theta(x_t, t)$  of the model:

$$\mathcal{L}_{simple} = \mathbb{E}_{t \sim [1, T]} \mathbb{E}_{x_0 \sim p(x_0)} \mathbb{E}_{z_t \sim \mathcal{N}(0, \mathbf{I})} \|z_t - z_\theta(x_t, t)\|^2, \quad (6)$$

where  $\mathbb{E}$  is the expected value, and  $z_\theta(x_t, t)$  is the network predicting the noise in  $x_t$ . We underline that  $x_t$  is sampled via (3), where we use a random image  $x_0$  from the training set.

The generative process is still defined by  $p_\theta(x_{t-1}|x_t)$ , but the neural network does not predict the mean and the covariance directly. Instead, it is trained to predict the noise from the image, and the mean is determined according to (5), while the covariance is fixed to a constant. Algorithm 1 formalizes the whole generative procedure.

## B. Noise Conditioned Score Networks (NCSNs)

The score function of some data density  $p(x)$  is defined as the gradient of the log density with respect to the input,  $\nabla_x \log p(x)$ . The directions given by these gradients are used by the Langevin dynamics algorithm [3] to move from a random sample  $(x_0)$  towards samples  $(x_N)$  in regions with high density. *Langevin dynamics* is an iterative method inspired from physics that can be used for data sampling. In physics, this method is used to determine the trajectory of a particle in a molecular system that allows interactions between the particle and the other molecules. The trajectory of the particle is influenced by a drag force of the system and by a random force motivated by the fast interactions between the molecules. In our case, we can think of the gradient of the log density as a force that drags a random sample through the data space into regions with high data density  $p(x)$ . The other term  $\omega_i$  accounts, in physics, for the random force, but

**Algorithm 1:** DDPM Sampling Method.**Input:**

$T$  – the number of diffusion steps.  
 $\sigma_1, \dots, \sigma_T$  – the standard deviations for the reverse transitions.

**Output:**

$x_0$  – the sampled image.

**Computation:**

- 1:  $x_T \sim \mathcal{N}(0, \mathbf{I})$
- 2: **For**  $t = T, \dots, 1$  **do**
- 3:   **if**  $t > 1$  **then**
- 4:      $z \sim \mathcal{N}(0, \mathbf{I})$
- 5:   **else**
- 6:      $z = 0$
- 7:      $\mu_\theta = \frac{1}{\sqrt{\alpha_t}} \cdot (x_t - \frac{1-\alpha_t}{\sqrt{1-\beta_t}} \cdot z_\theta(x_t, t))$
- 8:      $x_{t-1} = \mu_\theta + \sigma_t \cdot z$

for us, it is useful to escape local minima. Lastly, the value of  $\gamma$  weighs the impact of both forces, because it represents the friction coefficient of the environment where the particle resides. From the sampling point of view,  $\gamma$  controls the magnitude of the updates. In summary, the iterative updates of the Langevin dynamics are the following:

$$x_i = x_{i-1} + \frac{\gamma}{2} \nabla_x \log p(x) + \sqrt{\gamma} \cdot \omega_i, \quad (7)$$

where  $i \in \{1, \dots, N\}$ ,  $\gamma$  controls the magnitude of the update in the direction of the score,  $x_0$  is sampled from a prior distribution, the noise  $\omega_i \sim \mathcal{N}(0, \mathbf{I})$  addresses the issue of getting stuck in local minima, and the method is applied recursively for  $N \rightarrow \infty$  steps. Therefore, a generative model can employ the above method to sample from  $p(x)$  after estimating the score with a neural network  $s_\theta(x) \approx \nabla_x \log p(x)$ . This network can be trained via score matching, a method that requires the optimization of the following objective:

$$\mathcal{L}_{sm} = \mathbb{E}_{x \sim p(x)} \|s_\theta(x) - \nabla_x \log p(x)\|_2^2. \quad (8)$$

In practice, it is impossible to minimize this objective directly, because  $\nabla_x \log p(x)$  is unknown. However, there are other methods such as denoising score matching [67] and sliced score matching [68] that overcome this problem.

Although the described approach can be used for data generation, Song et al. [3] emphasize several issues when applying this method on real data. Most of the problems are linked with the manifold hypothesis. For example, the score estimation  $s_\theta(x)$  is inconsistent when the data resides on a low-dimensional manifold and, among other implications, this could cause the Langevin dynamics to never converge to the high-density regions. In the same work [3], the authors demonstrate that these problems can be addressed by perturbing the data with Gaussian noise at different scales. Furthermore, they propose to learn score estimates for the resulting noisy distributions via a single noise conditioned score network (NCSN). Regarding the sampling,

**Algorithm 2:** Annealed Langevin Dynamics.**Input:**

$\sigma_1, \dots, \sigma_T$  – a sequence of Gaussian noise scales.  
 $N$  – the number of Langevin dynamics iterations.  
 $\gamma_1, \dots, \gamma_T$  – the update magnitudes for each noise scale.

**Output:**

$x_0^0$  – the sampled image.

**Computation:**

- 1:  $x_T^0 \sim \mathcal{N}(0, \mathbf{I})$
- 2: **For**  $t = T, \dots, 1$  **do**
- 3:   **For**  $i = 1, \dots, N$  **do**
- 4:      $\omega \sim \mathcal{N}(0, \mathbf{I})$
- 5:      $x_t^i = x_t^{i-1} + \frac{\gamma_t}{2} \cdot s_\theta(x_t^{i-1}, \sigma_t) + \sqrt{\gamma_t} \cdot \omega$
- 6:    $x_{t-1}^0 = x_t^N$

they adapt the strategy in (7) and use the score estimates associated with each noise scale.

Formally, given a sequence of Gaussian noise scales  $\sigma_1 < \sigma_2 < \dots < \sigma_T$  such that  $p_{\sigma_1}(x) \approx p(x_0)$  and  $p_{\sigma_T}(x) \approx \mathcal{N}(0, \mathbf{I})$ , we can train an NCSN  $s_\theta(x, \sigma_t)$  with denoising score matching so that  $s_\theta(x, \sigma_t) \approx \nabla_x \log(p_{\sigma_t}(x))$ ,  $\forall t \in \{1, \dots, T\}$ . We can derive  $\nabla_x \log(p_{\sigma_t}(x))$  as follows:

$$\nabla_x \log p_{\sigma_t}(x_t | x) = -\frac{x_t - x}{\sigma_t^2}, \quad (9)$$

given that:

$$\begin{aligned} p_{\sigma_t}(x_t | x) &= \mathcal{N}(x_t; x, \sigma_t^2 \cdot \mathbf{I}) \\ &= \frac{1}{\sigma_t \cdot \sqrt{2\pi}} \cdot \exp\left(-\frac{1}{2} \cdot \left(\frac{x_t - x}{\sigma_t}\right)^2\right), \end{aligned} \quad (10)$$

where  $x_t$  is a noised version of  $x$ , and  $\exp$  is the exponential function. Consequently, generalizing (8) for all  $(\sigma_t)_{t=1}^T$  and replacing the gradient with the form in (9) leads to training  $s_\theta(x_t, \sigma_t)$  by minimizing the following objective,  $\forall t \in \{1, \dots, T\}$ :

$$\mathcal{L}_{dsm} = \frac{1}{T} \sum_{t=1}^T \lambda(\sigma_t) \mathbb{E}_{p(x)} \mathbb{E}_{x_t \sim p_{\sigma_t}(x_t | x)} \left\| s_\theta(x_t, \sigma_t) + \frac{x_t - x}{\sigma_t^2} \right\|_2^2, \quad (11)$$

where  $\lambda(\sigma_t)$  is a weighting function. After training, the neural network  $s_\theta(x_t, \sigma_t)$  will return estimates of the scores  $\nabla_x \log(p_{\sigma_t}(x_t))$ , having as input the noisy image  $x_t$  and the corresponding time step  $t$ .

At inference time, Song et al. [3] introduce the annealed Langevin dynamics, formally described in Algorithm 2. Their method starts with white noise and applies (7) for a fixed number of iterations. The required gradient (score) is given by the trained neural network conditioned on the time step  $T$ . The process continues for the following time steps, propagating the output of one step as input to the next. The final sample is the output returned for  $t = 0$ .

### C. Stochastic Differential Equations (SDEs)

Similar to the previous two methods, the approach presented in [4] gradually transforms the data distribution  $p(x_0)$  into noise. However, it generalizes over the previous two methods because, in its case, the diffusion process is considered to be continuous, thus becoming the solution of a stochastic differential equation (SDE). As shown in [69], the reverse process of this diffusion can be modeled with a reverse-time SDE which requires the score function of the density at each time step. Therefore, the generative model of Song et al. [4] employs a neural network to estimate the score functions, and generates samples from  $p(x_0)$  by employing numerical SDE solvers. As in the case of NCSNs, the neural network receives the perturbed data and the time step as input, and produces an estimation of the score function.

The SDE of the forward diffusion process  $(x_t)_{t=0}^T, t \in [0, T]$  has the following form:

$$\frac{\partial x}{\partial t} = f(x, t) + \sigma(t) \cdot \omega_t \iff \partial x = f(x, t) \cdot \partial t + \sigma(t) \cdot \partial \omega, \quad (12)$$

where  $\omega_t$  is Gaussian noise,  $f$  is a function of  $x$  and  $t$  that computes the drift coefficient, and  $\sigma$  is a time-dependent function that computes the diffusion coefficient. In order to have a diffusion process as a solution for this SDE, the drift coefficient should be designed such that it gradually nullifies the data  $x_0$ , while the diffusion coefficient controls how much Gaussian noise is added. The associated reverse-time SDE [69] is defined as follows:

$$\partial x = [f(x, t) - \sigma(t)^2 \cdot \nabla_x \log p_t(x)] \cdot \partial t + \sigma(t) \cdot \partial \hat{\omega}, \quad (13)$$

where  $\hat{\omega}$  represents the Brownian motion when the time is reversed, from  $T$  to 0. The reverse-time SDE shows that, if we start with pure noise, we can recover the data by removing the drift responsible for data destruction. The removal is performed by subtracting  $\sigma(t)^2 \cdot \nabla_x \log p_t(x)$ .

We can train the neural network  $s_\theta(x, t) \approx \nabla_x \log p_t(x)$  by optimizing the same objective as in (11), but adapted for the continuous case, as follows:

$$\begin{aligned} \mathcal{L}_{dsm}^* &= \\ &= \mathbb{E}_t \left[ \lambda(t) \mathbb{E}_{p(x_0)} \mathbb{E}_{p_t(x_t|x_0)} \| s_\theta(x_t, t) - \nabla_{x_t} \log p_t(x_t|x_0) \|_2^2 \right], \end{aligned} \quad (14)$$

where  $\lambda$  is a weighting function, and  $t \sim \mathcal{U}([0, T])$ . We underline that, when the drift coefficient  $f$  is affine,  $p_t(x_t|x_0)$  is a Gaussian distribution. When  $f$  does not conform to this property, we cannot use denoising score matching, but we can fallback to sliced score matching [68].

The sampling for this approach can be performed with any numerical method applied on the SDE defined in (13). In practice, the solvers do not work with the continuous formulation. For example, the Euler-Maruyama method fixes a tiny negative step  $\Delta t$  and executes Algorithm 3 until the initial time step  $t = T$  becomes  $t = 0$ . At step 3, the Brownian motion is given by  $\Delta \hat{\omega} = \sqrt{|\Delta t|} \cdot z$ , where  $z \sim \mathcal{N}(0, \mathbf{I})$ .

Song et al. [4] present several contributions in terms of sampling techniques. They introduce the Predictor-Corrector sampler which generates better examples. This algorithm first

---

### Algorithm 3: Euler-Maruyama Sampling Method.

---

**Input:**

$\Delta t < 0$  – a negative step close to 0.

$f$  – a function of  $x$  and  $t$  that computes the drift coefficient.

$\sigma$  – a time-dependent function that computes the diffusion coefficient.

$\nabla_x \log p_t(x)$  – the (approximated) score function.

$T$  – the final time step of the forward SDE.

---

**Output:**

$x$  – the sampled image.

---

**Computation:**

1:  $t = T$

2: **while**  $t > 0$  **do**

3:    $\Delta x = [f(x, t) - \sigma(t)^2 \cdot \nabla_x \log p_t(x)] \cdot \Delta t + \sigma(t) \cdot \Delta \hat{\omega}$

4:    $x = x + \Delta x$

5:    $t = t + \Delta t$

---

employs a numerical method to sample from the reverse-time SDE, and then uses a score-based method as a corrector, for example the annealed Langevin dynamics described in the previous subsection. Furthermore, they show that ordinary differential equations (ODEs) can also be used to model the reverse process. Hence, another sampling strategy unlocked by the SDE interpretation is based on numerical methods applied to ODEs. The main advantage of this latter strategy is its efficiency.

### D. Relation to Other Generative Models

We discuss below the connections between diffusion models and other types of generative models. We start with likelihood-based methods and finish with generative adversarial networks.

Diffusion models have more aspects in common with VAEs [50]. For instance, in both cases, the data is mapped to a latent space and the generative process learns to transform the latent representations into data. Moreover, in both situations, the objective function can be derived as a lower-bound of the data likelihood. Nevertheless, there are essential differences between the two approaches and, further, we will mention some of them. The latent representation of a VAE contains compressed information about the original image, while diffusion models destroy the data entirely after the last step of the forward process. The latent representations of diffusion models have the same dimensions as the original data, while VAEs work better when the dimensions are reduced. Ultimately, the mapping to the latent space of a VAE is trainable, which is not true for the forward process of diffusion models because, as stated before, the latent is obtained by gradually adding Gaussian noise to the original image. The aforementioned similarities and differences can be the key for future developments of the two methods. For example, there already exists some work that builds more efficient diffusion models by applying them on the latent space of a VAE [17], [18].

Autoregressive models [62], [70] represent images as sequences of pixels. Their generative process produces new samples by generating an image pixel by pixel, conditioned on the previously generated pixels. This approach implies a unidirectional bias that clearly represents a limitation of this class of generative models. Esser et al. [28] see diffusion and autoregressive models as complementary and solve the above issue. Their method learns to reverse a multinomial diffusion process via a Markov chain where each transition is implemented as an autoregressive model. The global information provided to the autoregressive model is given by the previous step of the Markov chain.

Normalizing flows [63], [64] are a class of generative models that transform a simple Gaussian distribution into a complex data distribution. The transformation is done via a set of invertible functions which have an easy-to-compute Jacobian determinant. These conditions translate in practice into architectural restrictions. An important feature of this type of model is that the likelihood is tractable. Hence, the objective for training is the negative log-likelihood. When comparing with diffusion models, the two types of models have in common the mapping of the data distribution to Gaussian noise. However, the similarities between the two methods end here, because normalizing flows perform the mapping in a deterministic fashion by learning an invertible and differentiable function. These properties imply, in contrast to diffusion models, additional constraints on the network architecture, and a learnable forward process. A method which connects these two generative algorithms is DiffFlow. Introduced in [71], DiffFlow extends both diffusion models and normalizing flows such that the reverse and forward processes are both trainable and stochastic.

Energy-based models (EBMs) [60], [61], [72], [73] focus on providing estimates of unnormalized versions of density functions, called energy functions. Thanks to this property and in contrast to the previous likelihood-based methods, this type of model can be represented with any regression neural network. However, due to this flexibility, the training of EBMs is difficult. One popular training strategy used in practice is score matching [72], [73]. Regarding the sampling, among other strategies, there is the Markov Chain Monte Carlo (MCMC) method, which is based on the score function. Therefore, the formulation from Section II-B of diffusion models can be considered to be a particular case of the energy-based framework, precisely the case when the training and sampling only require the score function.

GANs [52] were considered by many as state-of-the-art generative models in terms of the quality of the generated samples, before the recent rise of diffusion models [5]. GANs are also known as being difficult to train due to their adversarial objective [74], and often suffer from mode collapse. In contrast, diffusion models have a stable training process and provide more diversity because they are likelihood-based. Despite these advantages, diffusion models are still inefficient when compared to GANs, requiring multiple network evaluations during inference. A key aspect for comparison between GANs and diffusion models is their latent space. While GANs have a low-dimensional latent space, diffusion models preserve the original size of the images. Furthermore, the latent space of diffusion models is usually

modeled as a random Gaussian distribution, being similar to VAEs. In terms of semantic properties, it was discovered that the latent space of GANs contains subspaces associated with visual attributes [75]. Thanks to this property, the attributes can be manipulated with changes in the latent space [75], [76]. In contrast, when such transformations are desired for diffusion models, the preferred procedure is the guidance technique [5], [77], which does not exploit any semantic property of the latent space. However, Song et al. [4] demonstrate that the latent space of diffusion models has a well-defined structure, illustrating that interpolations in this space lead to interpolations in the image space. In summary, from the semantic perspective, the latent space of diffusion models has been explored much less than in the case of GANs, but this may be one of the future research directions to be followed by the community.

### III. A CATEGORIZATION OF DIFFUSION MODELS

We categorize diffusion models into a multi-perspective taxonomy considering different criteria of separation. Perhaps the most important criteria to separate the models are defined by (i) the task they are applied to, and (ii) the input signals they require. Furthermore, as there are multiple approaches in formulating a diffusion model, (iii) the underlying framework is another key factor for classifying diffusion models. Finally, the (iv) data sets used during training and evaluation are also of high importance, because they provide the means to compare different models on the same task. Our categorization of diffusion models according to the criteria enumerated above is presented in Table I.

In the remainder of this section, we present several contributions on diffusion models, choosing the target task as the primary criterion to separate the methods. We opted for this classification criterion as it is fairly well-balanced and representative for research on diffusion models, facilitating a quick grasping of related works by readers working on specific tasks. Although the main task is usually related to image generation, a considerable amount of work has been conducted to match and even surpass the performance of GANs on other topics, such as super-resolution, inpainting, image editing, image-to-image translation or segmentation.

#### A. Unconditional Image Generation

The diffusion models presented below are used to generate samples in an unconditional setting. Such models do not require supervision signals, being completely unsupervised. We consider this as the most basic and generic setting for image generation.

*1) Denoising Diffusion Probabilistic Models:* The work of Sohl-Dickstein et al. [1] formalizes diffusion models as described in Section II-A. The proposed network is based on a convolutional architecture with multi-scale convolutions.

Austin et al. [78] extend the approach of Sohl-Dickstein et al. [1] to discrete diffusion models, studying different choices for the transition matrices used in the forward process. Their results are competitive with previous continuous diffusion models for the image generation task.

TABLE I

OUR MULTI-PERSPECTIVE CATEGORIZATION OF DIFFUSION MODELS APPLIED IN COMPUTER VISION. TO CLASSIFY EXISTING MODELS, WE CONSIDER THREE CRITERIA: THE TASK, THE DENOISING CONDITION, AND THE UNDERLYING APPROACH (ARCHITECTURE). ADDITIONALLY, WE LIST THE DATA SETS ON WHICH THE SURVEYED MODELS ARE APPLIED. WE USE THE FOLLOWING ABBREVIATIONS IN THE ARCHITECTURE COLUMN: D3PM (DISCRETE DENOISING DIFFUSION PROBABILISTIC MODELS), DSB (DIFFUSION SCHRÖDINGER BRIDGE), BDDM (BILATERAL DENOISING DIFFUSION MODELS), PNDM (PSEUDO NUMERICAL METHODS FOR DIFFUSION MODELS), ADM (ABLATED DIFFUSION MODEL), D2C (DIFFUSION-DECODING MODELS WITH CONTRASTIVE REPRESENTATIONS), CCDF (COME-CLOSER-DIFFUSE-FASTER), VQ-DDM (VECTOR QUANTISED DISCRETE DIFFUSION MODEL), BF-CNN (BIAS-FREE CNN), FDM (FLEXIBLE DIFFUSION MODEL), RVD (RESIDUAL VIDEO DIFFUSION), RAMVID (RANDOM MASK VIDEO DIFFUSION)

| Paper                            | Task                         | Denoising Condition                   | Architecture              | Data Sets   |
|----------------------------------|------------------------------|---------------------------------------|---------------------------|---|
| Austin <i>et al.</i> [78]        | image generation             | unconditional                         | D3PM                      | CIFAR-10  |
| Bao <i>et al.</i> [19]           | image generation             | unconditional                         | DDIM, Improved DDPMP      | CelebA, ImageNet, LSUN Bedroom, CIFAR-10                    |
| Benny <i>et al.</i> [79]         | image generation             | unconditional                         | DDPM, DDIM                | CIFAR-10, ImageNet, CelebA                                  |
| Bond-Taylor <i>et al.</i> [80]   | image generation             | unconditional                         | DDPM                      | LSUN Bedroom, LSUN Church, FFHQ                             |
| Choi <i>et al.</i> [81]          | image generation             | unconditional                         | DDPM                      | FFHQ, AFHQ-Dog, CUB, Met-Faces                              |
| De <i>et al.</i> [82]            | image generation             | unconditional                         | DSB                       | MNIST, CelebA   |
| Deasy <i>et al.</i> [83]         | image generation             | unconditional                         | NCSN                      | MNIST, Fashion-MNIST, CIFAR-10, CelebA                      |
| Deja <i>et al.</i> [84]          | image generation             | unconditional                         | Improved DDPMP            | Fashion-MNIST, CIFAR-10, CelebA                             |
| Dockhorn <i>et al.</i> [20]      | image generation             | unconditional                         | NCSN++, DDPMP++           | CIFAR-10  |
| Ho <i>et al.</i> [2]             | image generation             | unconditional                         | DDPM                      | CIFAR-10, CelebA-HQ, LSUN                                   |
| Huang <i>et al.</i> [59]         | image generation             | unconditional                         | DDPM                      | CIFAR-10, MNIST   |
| Jing <i>et al.</i> [30]          | image generation             | unconditional                         | NCSN++, DDPMP++           | CIFAR-10, CelebA-256-HQ, LSUN Church                        |
| Jolicoeur <i>et al.</i> [85]     | image generation             | unconditional                         | NCSN                      | CIFAR-10, LSUN Church, Stacked-MNIST                        |
| Jolicoeur <i>et al.</i> [86]     | image generation             | unconditional                         | DDPM++, NCSN++            | CIFAR-10, LSUN Church, FFHQ                                 |
| Kim <i>et al.</i> [87]           | image generation             | unconditional                         | NCSN++, DDPMP++           | CIFAR-10, CelebA, MNIST                                     |
| Kingma <i>et al.</i> [88]        | image generation             | unconditional                         | DDPM                      | CIFAR-10, ImageNet  |
| Kong <i>et al.</i> [89]          | image generation             | unconditional                         | DDIM, DDPMP               | LSUN Bedroom, CelebA, CIFAR-10                              |
| Lam <i>et al.</i> [90]           | image generation             | unconditional                         | BDDM                      | CIFAR-10, CelebA  |
| Liu <i>et al.</i> [91]           | image generation             | unconditional                         | PNDM                      | CIFAR-10, CelebA  |
| Ma <i>et al.</i> [92]            | image generation             | unconditional                         | NCSN, NCSN++              | CIFAR-10, CelebA, LSUN Bedroom, LSUN Church, FFHQ           |
| Nachmani <i>et al.</i> [93]      | image generation             | unconditional                         | DDIM, DDPM                | CelebA, LSUN Church   |
| Nichol <i>et al.</i> [6]         | image generation             | unconditional                         | DDPM                      | CIFAR-10, ImageNet  |
| Pandey <i>et al.</i> [18]        | image generation             | unconditional                         | DDPM                      | CelebA-HQ, CIFAR-10   |
| San <i>et al.</i> [94]           | image generation             | unconditional                         | DDPM                      | CelebA, LSUN Bedroom, LSUN Church                           |
| Sehwag <i>et al.</i> [95]        | image generation             | unconditional                         | ADM                       | CIFAR-10, ImageNet  |
| Sohl-Dickstein <i>et al.</i> [1] | image generation             | unconditional                         | DDPM                      | MNIST, CIFAR-10, Dead Leaf Images                           |
| Song <i>et al.</i> [13]          | image generation             | unconditional                         | NCSN                      | FFHQ, CelebA, LSUN Bedroom, LSUN Tower, LSUN Church Outdoor |
| Song <i>et al.</i> [15]          | image generation             | unconditional                         | DDPM++                    | CIFAR-10, ImageNet 32×32                                    |
| Song <i>et al.</i> [7]           | image generation             | unconditional                         | DDIM                      | CIFAR-10, CelebA, LSUN                                      |
| Vahdat <i>et al.</i> [17]        | image generation             | unconditional                         | NCSN++                    | CIFAR-10, CelebA-HQ, MNIST                                  |
| Wang <i>et al.</i> [96]          | image generation             | unconditional                         | DDIM                      | CIFAR-10, CelebA  |
| Wang <i>et al.</i> [97]          | image generation             | unconditional                         | StyleGAN2, ProjectedGAN   | CIFAR-10, STL-10, LSUN Bedroom, LSUN Church, AFHQ, FFHQ     |
| Watson <i>et al.</i> [98]        | image generation             | unconditional                         | DDPM                      | CIFAR-10, ImageNet  |
| Watson <i>et al.</i> [8]         | image generation             | unconditional                         | Improved DDPMP            | CIFAR-10, ImageNet 64×64                                    |
| Xiao <i>et al.</i> [99]          | image generation             | unconditional                         | NCSN++                    | CIFAR-10  |
| Zhang <i>et al.</i> [71]         | image generation             | unconditional                         | DDPM                      | CIFAR-10, MNIST   |
| Zheng <i>et al.</i> [100]        | image generation             | unconditional                         | DDPM                      | CIFAR-10, CelebA, CelebA-HQ, LSUN Bedroom, LSUN Church      |
| Bordes <i>et al.</i> [101]       | conditional image generation | conditioned on latent representations | Improved DDPMP            | ImageNet  |
| Campbell <i>et al.</i> [102]     | conditional image generation | unconditional, conditioned on sound   | DDPM                      | CIFAR-10, Lakh Pianoroll                                    |
| Chao <i>et al.</i> [103]         | conditional image generation | conditioned on class                  | Score SDE, Improved DDPMP | CIFAR-10, CIFAR-100   |
| Dhariwal <i>et al.</i> [5]       | conditional image generation | unconditional, classifier guidance    | ADM                       | LSUN Bedroom, LSUN Horse, LSUN Cat                          |

TABLE I  
(CONTINUED)

| Paper                         | Task  | Denoising Condition   | Architecture                      | Data Sets   |
|-------------------------------|---|---|-----------------------------------|---|
| Ho <i>et al.</i> [104]        | conditional image generation  | conditioned on label  | DDPM                              | LSUN, ImageNet  |
| Ho <i>et al.</i> [77]         | conditional image generation  | unconditional, classifier-free guidance   | ADM                               | ImageNet 64×64, ImageNet 128×128                              |
| Karras <i>et al.</i> [105]    | conditional image generation  | unconditional, conditioned on class   | DDPM++, NCSN++, DDPM, DDIM        | CIFAR-10, ImageNet 64×64                                      |
| Liu <i>et al.</i> [106]       | conditional image generation  | conditioned on text, image, style guidance  | DDPM                              | FFHQ, LSUN Cat, LSUN Horse, LSUN Bedroom                      |
| Liu <i>et al.</i> [21]        | conditional image generation  | conditioned on text, 2D positions, relational descriptions between items, human facial attributes | Improved DDPM                     | CLEVR, Relational CLEVR, FFHQ                                 |
| Lu <i>et al.</i> [107]        | conditional image generation  | unconditional, conditioned on class   | DDIM                              | CIFAR-10, CelebA, ImageNet, LSUN Bedroom                      |
| Salimans <i>et al.</i> [108]  | conditional image generation  | unconditional, conditioned on class   | DDIM                              | CIFAR-10, ImageNet, LSUN                                      |
| Singh <i>et al.</i> [109]     | conditional image generation  | conditioned on noise  | DDIM                              | ImageNet  |
| Sinha <i>et al.</i> [16]      | conditional image generation  | unconditional, conditioned on label   | D2C                               | CIFAR-10, CIFAR-100, fMoW, CelebA-64, CelebA-HQ-256, FFHQ-256 |
| Ho <i>et al.</i> [34]         | image-to-image translation  | conditioned on image  | Improved DDPM                     | ctest10k, places10k   |
| Li <i>et al.</i> [37]         | image-to-image translation  | conditioned on image  | DDPM                              | Face2Comic, Edges2Shoes, Edges2Handbags                       |
| Sasaki <i>et al.</i> [110]    | image-to-image translation  | conditioned on image  | DDPM                              | CMP Facades, KAIST Multi-spectral Pedestrian                  |
| Wang <i>et al.</i> [36]       | image-to-image translation  | conditioned on image  | DDIM                              | ADE20K, COCO-Stuff, DIODE                                     |
| Wolleb <i>et al.</i> [38]     | image-to-image translation  | conditioned on image  | Improved DDPM                     | BRATS   |
| Zhao <i>et al.</i> [35]       | image-to-image translation  | conditioned on image  | DDPM                              | CelebaA-HQ, AFHQ  |
| Gu <i>et al.</i> [111]        | text-to-image generation  | conditioned on text   | VQ-Diffusion                      | CUB-200, Oxford 102 Flowers, MS-COCO                          |
| Jiang <i>et al.</i> [22]      | text-to-image generation  | conditioned on text   | Transformer-based encoder-decoder | DeepFashion-MultiModal  |
| Ramesh <i>et al.</i> [112]    | text-to-image generation  | conditioned on text   | ADM                               | MS-COCO, AVA  |
| Rombach <i>et al.</i> [11]    | text-to-image generation  | conditioned on text   | LDM                               | OpenImages, WikiArt, LAION-2B-en, ArtBench                    |
| Saharia <i>et al.</i> [12]    | text-to-image generation  | conditioned on text   | Imagen                            | MS-COCO, DrawBench  |
| Shi <i>et al.</i> [9]         | text-to-image generation  | unconditional, conditioned on text  | Improved DDPM                     | Conceptual Captions, MS-COCO                                  |
| Zhang <i>et al.</i> [113]     | text-to-image generation  | unconditional, conditioned on text  | DDIM                              | CIFAR-10, CelebA, ImageNet                                    |
| Daniels <i>et al.</i> [25]    | super-resolution  | conditioned on image  | NCSN                              | CIFAR-10, CelebA  |
| Saharia <i>et al.</i> [23]    | super-resolution  | conditioned on image  | DDPM++                            | FFHQ, CelebA-HQ, ImageNet-1K                                  |
| Avrahami <i>et al.</i> [114]  | image editing   | conditioned on image and mask   | DDPM, ADM                         | ImageNet, CUB, LSUN Bedroom, MS-COCO                          |
| Avrahami <i>et al.</i> [31]   | region image editing  | text guidance   | DDPM                              | PaintByWord   |
| Meng <i>et al.</i> [33]       | image editing   | conditioned on image  | Score SDE, DDPM, Improved DDPM    | LSUN, CelebA-HQ   |
| Lugmayr <i>et al.</i> [29]    | inpainting  | unconditional   | DDPM                              | CelebA-HQ, ImageNet   |
| Nichol <i>et al.</i> [14]     | inpainting  | conditioned on image, text guidance   | ADM                               | MS-COCO   |
| Amit <i>et al.</i> [42]       | image segmentation  | conditioned on image  | Improved DDPM                     | Cityscapes, Vaihingen, MoNuSeg                                |
| Baranchuk <i>et al.</i> [39]  | image segmentation  | conditioned on image  | Improved DDPM                     | LSUN, FFHQ-256, ADE-Bedroom-30, CelebA-19                     |
| Batzolis <i>et al.</i> [24]   | multi-task (inpainting, super-resolution, edge-to-image)                                | conditioned on image  | DDPM                              | CelebA, Edges2Shoes   |
| Batzolis <i>et al.</i> [115]  | multi-task (image generation, super-resolution, inpainting, image-to-image translation) | unconditional   | DDIM                              | ImageNet, CelebA-HQ, CelebA, Edges2Shoes                      |
| Blattmann <i>et al.</i> [116] | multi-task (image generation)   | unconditional, conditioned on text, class   | LDM                               | ImageNet  |
| Choi <i>et al.</i> [32]       | multi-task (image generation, image-to-image translation, image editing)                | conditioned on image  | DDPM                              | FFHQ, MetFaces  |
| Chung <i>et al.</i> [26]      | multi-task (inpainting, super-resolution, MRI reconstruction)                           | conditioned on image  | CCDF                              | FFHQ, AFHQ, fastMRI knee                                      |
| Esser <i>et al.</i> [28]      | multi-task (image generation, inpainting)   | unconditional, conditioned on class, image and text   | ImageBART                         | ImageNet, Conceptual Captions, FFHQ, LSUN                     |
| Gao <i>et al.</i> [117]       | multi-task (image generation, inpainting)   | unconditional, conditioned on image   | DDPM                              | CIFAR-10, LSUN, CelebA  |
| Graikos <i>et al.</i> [40]    | multi-task (image generation, image segmentation)                                       | conditioned on class  | DDIM                              | FFHQ-256, CelebA  |

TABLE I  
(CONTINUED)

| Paper                          | Task  | Denoising Condition                         | Architecture                   | Data Sets   |
|--------------------------------|---|---|--------------------------------|---|
| Hu <i>et al.</i> [118]         | multi-task (image generation, inpainting)   | unconditional, conditioned on image         | VQ-DDM                         | CelebA-HQ, LSUN Church                              |
| Khrulkov <i>et al.</i> [119]   | multi-task (image generation, image-to-image translation)   | conditioned on class                        | Improved DDPM                  | AFHQ, FFHQ, MetFaces, ImageNet                      |
| Kim <i>et al.</i> [120]        | multi-task (image translation, multi-attribute transfer)  | conditioned on image, portrait, stroke      | DDIM                           | ImageNet, CelebA-HQ, AFHQ-Dog, LSUN Bedroom, Church |
| Luo <i>et al.</i> [121]        | multi-task (point cloud generation, auto-encoding, unsupervised representation learning)                  | conditioned on shape latent                 | DDPM                           | ShapeNet  |
| Lyu <i>et al.</i> [122]        | multi-task (image generation, image editing)  | unconditional, conditioned on class         | DDPM                           | CIFAR-10, CelebA, ImageNet, LSUN Bedroom, LSUN Cat  |
| Preechakul <i>et al.</i> [123] | multi-task (latent interpolation, attribute manipulation)   | conditioned on latent representation        | DDIM                           | CelebA-HQ   |
| Rombach <i>et al.</i> [10]     | multi-task (super-resolution, image generation, inpainting)   | unconditional, conditioned on image         | VQ-DDM                         | ImageNet, CelebA-HQ, FFHQ, LSUN                     |
| Shi <i>et al.</i> [124]        | multi-task (super-resolution, inpainting)   | conditioned on image                        | Improved DDPM                  | MNIST, CelebA                                       |
| Song <i>et al.</i> [3]         | multi-task (image generation, inpainting)   | unconditional, conditioned on image         | NCSN                           | MNIST, CIFAR-10, CelebA                             |
| Kadkhodaie <i>et al.</i> [125] | multi-task (Spatial super-resolution, Deblurring, Compressive sensing, Inpainting, Random missing pixels) | conditioned on linear measurements          | BF-CNN                         | MNIST, Set5, Set68, Set14                           |
| Song <i>et al.</i> [4]         | multi-task (image generation, inpainting, colorization)   | unconditional, conditioned on image, class  | NCSN++, DDPM++                 | CelebA-HQ, CIFAR-10, LSUN                           |
| Hu <i>et al.</i> [126]         | medical image-to-image translation  | conditioned on image                        | DDPM                           | ONH   |
| Chung <i>et al.</i> [127]      | medical image generation  | conditioned on measurements                 | NCSN++                         | fastMRI knee  |
| Özbey <i>et al.</i> [128]      | medical image generation  | conditioned on image                        | Improved DDPM                  | IXI, Gold Atlas - Male Pelvis                       |
| Song <i>et al.</i> [129]       | medical image generation  | conditioned on measurements                 | NCSN++                         | LIDC, LDCT Image and Projection, BRATS              |
| Wolleb <i>et al.</i> [41]      | medical image segmentation  | conditioned on image                        | Improved DDPM                  | BRATS   |
| Sanchez <i>et al.</i> [130]    | medical image segmentation and anomaly detection  | conditioned on image and binary variable    | ADM                            | BRATS   |
| Pinaya <i>et al.</i> [44]      | medical image segmentation and anomaly detection  | conditioned on image                        | DDPM                           | MedNIST, UK Biobank Images, WMH, BRATS, MSLUB       |
| Wolleb <i>et al.</i> [45]      | medical image anomaly detection   | conditioned on image                        | DDIM                           | CheXpert, BRATS                                     |
| Wyatt <i>et al.</i> [46]       | medical image anomaly detection   | conditioned on image                        | ADM                            | NFBS, 22 MRI scans                                  |
| Harvey <i>et al.</i> [131]     | video generation  | conditioned on frames                       | FDM                            | GQN-Mazes, MineRL Navigate, CARLA Town01            |
| Ho <i>et al.</i> [132]         | video generation  | unconditional, conditioned on text          | DDPM                           | 101 Human Actions                                   |
| Yang <i>et al.</i> [133]       | video generation  | conditioned on video representation         | RVD                            | BAIR, KTH Actions, Simulation, Cityscapes           |
| Höppe <i>et al.</i> [134]      | video generation and infilling  | conditioned on frames                       | RaMVid                         | BAIR, Kinetics-600, UCF-101                         |
| Giannone <i>et al.</i> [135]   | few-shot image generation   | conditioned on image                        | Improved DDPM                  | CIFAR-FS, mini-ImageNet, CelebA                     |
| Jeanneret <i>et al.</i> [136]  | counterfactual explanations   | unconditional                               | DDPM                           | CelebA  |
| Sanchez <i>et al.</i> [137]    | counterfactual estimates  | conditional                                 | ADM                            | MNIST, ImageNet                                     |
| Kawar <i>et al.</i> [27]       | image restoration   | conditioned on image                        | DDIM                           | FFHQ, ImageNet                                      |
| Özdenizci <i>et al.</i> [138]  | image restoration   | conditioned on image                        | DDPM                           | Snow100K, Outdoor-Rain, RainDrop                    |
| Kim <i>et al.</i> [139]        | image registration  | conditioned on image                        | DDPM                           | Radboud Faces, OASIS-3                              |
| Nie <i>et al.</i> [140]        | adversarial purification  | conditioned on image                        | Score SDE, Improved DDPM, DDIM | CIFAR-10, ImageNet, CelebA-HQ                       |
| Wang <i>et al.</i> [141]       | semantic image generation   | conditioned on semantic map                 | DDPM                           | Cityscapes, ADE20K, CelebAMask-HQ                   |
| Zhou <i>et al.</i> [142]       | shape generation and completion   | unconditional, conditional shape completion | DDPM                           | ShapeNet, PartNet                                   |
| Zimmermann <i>et al.</i> [43]  | classification  | conditioned on label                        | DDPM++                         | CIFAR-10  |

Ho *et al.* [2] extend the work presented in [1], proposing to learn the reverse process by estimating the noise in the image at each step. This change leads to an objective that resembles the denoising score matching applied in [3].

On top of the work proposed by Ho *et al.* [2], Nichol *et al.* [6] introduce several improvements, observing that the linear noise schedule is suboptimal for low resolution. They show that it is required to learn the variance in order to improve the

performance of diffusion models in terms of log-likelihood. This change allows faster sampling, somewhere around 50 steps being required.

Song *et al.* [7] replace the Markov forward process used in [2] with a non-Markovian one. The generative process changes such that the model first predicts the normal sample, and then, it is used to estimate the next step in the chain. The change leads to a faster sampling procedure with a small impact on quality. The

resulting framework is known as the denoising diffusion implicit model (DDIM).

Sinha et al. [16] present the diffusion-decoding model with contrastive representations (D2C), a generative method which trains a diffusion model on the latent representation of an encoder. The framework, which is based on the DDPM architecture presented in [2], produces images by mapping the latent representations to images.

San-Roman et al. [94] propose a method to estimate the noise parameters given the current input during inference, requiring less steps. The authors employ VGG-11 to estimate the noise parameters, and DDPM [2] to generate images.

Nachmani et al. [93] suggest replacing the Gaussian noise distributions of the diffusion process with two other distributions, a mixture of two Gaussians and the Gamma distribution. They obtain better Fréchet Inception Distance (FID) values and faster convergence thanks to the Gamma distribution that has higher modeling capacity.

Lam et al. [90] propose to learn the noise scheduling for sampling. The noise schedule for training remains linear, as before. After training the score network, they assume it to be close to the optimal value and use it for noise schedule training.

Bond-Taylor et al. [80] present a two-stage process, where they apply vector quantization to images to obtain discrete representations, and use a transformer [143] to reverse a discrete diffusion process, where the elements are randomly masked at each step. The sampling process is faster because the diffusion is applied to a highly compressed representation, which allows fewer denoising steps (50–256).

Watson et al. [98] propose a dynamic programming algorithm that finds the optimal inference schedule, having a time complexity of  $\mathcal{O}(T)$ , where  $T$  is the number of steps. They conduct their image generation experiments on CIFAR-10 and ImageNet, using the DDPM architecture.

In a different work, Watson et al. [8] show how a reparametrization trick can be integrated within the backward process of diffusion models in order to optimize a family of fast samplers. Using the Kernel Inception Distance as loss function, they show how optimization can be done using stochastic gradient descent. Next, they propose a special parametrized family of samplers, which, using the same process as before, can achieve competitive results with fewer sampling steps.

Similar to Bond-Taylor et al. [80] and Watson et al. [8], [98], Xiao et al. [99] try to improve the sampling speed, while also maintaining the quality, coverage and diversity of the samples. Their approach is to integrate a GAN in the denoising process to discriminate between real samples (forward process) and fake ones (denoised samples from the generator), with the objective of minimizing the softened reverse KL divergence [144].

Kingma et al. [88] introduce a class of diffusion models that obtains state-of-the-art likelihoods on image density estimation. They add Fourier features to the input of the network to predict the noise, and investigate if the observed improvement is specific to this class of models.

Following the work presented in [117], Bao et al. [19] propose an inference framework that does not require training using non-Markovian diffusion processes. By first deriving an analytical

estimate of the optimal mean and variance with respect to a score function, and using a pretrained scored-based model to obtain score values, they show better results, while being 20 to 40 times more time-efficient.

Zheng et al. [100] suggest truncating the process at an arbitrary step, and propose a method to inverse the diffusion from this distribution by relaxing the constraint of having Gaussian random noise as the final output of the forward diffusion. To avoid starting the reverse process from a non-tractable distribution, an implicit generative distribution is used to match the distribution of the diffused data.

Deja et al. [84] analyze the backward process of a diffusion model and postulate that it is formed of two models, a generator and a denoiser. Thus, they propose to explicitly split the process into two components: the denoiser, via an auto-encoder, and the generator, via a diffusion model.

Wang et al. [97] start from the idea presented by Arjovsky et al. [145] and Sønderby et al. [146] to augment the input data of the discriminator by adding noise. This is achieved in [97] by injecting noise from a Gaussian mixture distribution composed of weighted diffused samples from the clean image at various time steps. The noise injection mechanism is applied to both real and fake images.

**2) Score-Based Generative Models:** Starting from a previous work [3], Song et al. [13] present several improvements which are based on theoretical and empirical analyses. They address both training and sampling phases. For training, the authors show new strategies to choose the noise scales and how to incorporate the noise conditioning into NCSNs [3]. For sampling, they propose to apply an exponential moving average to the parameters and select the hyperparameters for the Langevin dynamics such that the step size verifies a certain equation. Jolicoeur-Martineau et al. [85] introduce an adversarial objective along with denoising score matching to train score-based models. Furthermore, they propose a new sampling procedure called Consistent Annealed Sampling and prove that it is more stable than the annealed Langevin method. The suggested changes are tested on the architectures proposed in [2], [3], [13].

Song et al. [15] improve the likelihood of score-based diffusion models. They achieve this through a new weighting function for the combination of the score matching losses. In [82], the authors present a score-based generative model as an implementation of Iterative Proportional Fitting (IPF), a technique used to solve the Schrödinger bridge problem. This novel approach is tested on image generation, as well as data set interpolation, which is possible because the prior can be any distribution.

Vahdat et al. [17] train diffusion models on latent representations. They use a VAE to encode to and decode from the latent space, achieving up to 56 times faster sampling.

**3) Stochastic Differential Equations:** DiffFlow is introduced in [71] as a new generative modeling approach that combines normalizing flows and diffusion probabilistic models. From the perspective of diffusion models, the method has a sampling procedure that is up to 20 times more efficient, thanks to a learnable forward process which skips unneeded noise regions. The authors perform experiments using the same architecture as in [2].

Jolicoeur-Martineau et al. [86] introduce a new SDE solver that is between  $2\times$  and  $5\times$  faster than Euler-Maruyama and does not affect the quality of the generated images. The solver is evaluated in a set of image generation experiments using pretrained models from [3].

Wang et al. [96] present a new deep generative model based on Schrödinger bridge. This is a two-stage method, where the first stage learns a smoothed version of the target distribution, and the second stage derives the actual target.

Dockhorn et al. [20] employ a critically-damped Langevin diffusion process by adding another variable (velocity) to the data, which is the only source of noise in the process. Given the new diffusion space, the resulting score function is demonstrated to be easier to learn. The authors extend their work by developing a more suitable score objective called hybrid score matching, as well as a sampling method, by solving the SDE through integration. To address the limitations of high-dimensional score-based models caused by the Gaussian noise distribution, Deasy et al. [83] extend denoising score matching to generalize to the normal noising distribution. By adding a heavier tailed distribution, their comprehensive experiments show promising results, as the generative performance improves in certain cases (depending on the shape of the distribution). Jing et al. [30] try to shorten the duration of the sampling process of diffusion models by reducing the space onto which diffusion is realized, i.e., the larger the time step in the diffusion process, the smaller the subspace. The data is projected onto a finite set of subspaces, at specific times, each being associated with a score model. This results in reduced computational costs, while increasing performance. Kim et al. [87] propose to change the diffusion process into a non-linear one. This is achieved by using a trainable normalizing flow model which encodes the image in the latent space, where it can now be linearly diffused to the noise distribution. A similar logic is then applied to the denoising process. Ma et al. [92] aim to make the backward diffusion process more time-efficient, while maintaining the synthesis performance. Within the family of score-based diffusion models, they begin to analyze the reverse diffusion in the frequency domain, subsequently applying a space-frequency filter to the sampling process, which aims to integrate information about the target distribution into the initial noise sampling.

### B. Conditional Image Generation

We next showcase diffusion models that are applied to conditional image synthesis. The condition is commonly based on various source signals, in most cases some class labels being used. Some methods perform both unconditional and conditional generation, which are also discussed here.

1) *Denoising Diffusion Probabilistic Models:* Dhariwal et al. [5] introduce few architectural changes to improve the FID of diffusion models. They also propose classifier guidance, a strategy which uses the gradients of a classifier to guide the diffusion during sampling. Bordes et al. [101] examine representations resulting from self-supervised tasks by visualizing and comparing them to the original image. They also compare representations generated from different sources. The authors implement several

modifications to the U-Net architecture presented by Dhariwal et al. [5], such as adding conditional batch normalization layers, and mapping the vector representation through a fully connected layer. The method presented in [95] allows diffusion models to produce images from low-density regions of the data manifold. They use two new losses to guide the reverse process. The first loss guides the diffusion towards low-density regions, while the second enforces the diffusion to stay on the manifold. Kong et al. [89] define a bijection between the continuous diffusion steps and the noise levels. With the defined bijection, they are able to construct an approximate diffusion process which requires less steps. The method is tested using the previous DDIM [7] and DDPM [2] architectures on image generation. Pandey et al. [18] build a generator-refiner framework, where the generator is a VAE and the refiner is a DDPM conditioned by the output of the VAE. The latent space of the VAE can be used to control the content of the generated image because the DDPM only adds the details. After training the framework, the resulting DDPM is able to generalize to different noise types. Ho et al. [104] introduce Cascaded Diffusion Models (CDM), an approach to generate high-resolution images conditioned on ImageNet classes. The framework uses multiple diffusion models, where the first model from the pipeline generates low-resolution images conditioned on the image class. The subsequent models are responsible for generating images of increasingly higher resolutions, being conditioned on both the class and the low-resolution image. Benny et al. [79] study the advantages and disadvantages of predicting the image instead of the noise during the reverse process. They conclude that some of the discovered problems could be addressed by interpolating the two types of output. Choi et al. [81] investigate the impact of the noise levels on the visual concepts learned by diffusion models. They modify the conventional weighting scheme of the objective function to a new one that enforces diffusion models to learn rich visual concepts. Singh et al. [109] propose a novel method for conditional image generation. Instead of conditioning the signal at each sampling step, they present a method to condition the noise signal (from where the sampling starts). Using Inverting Gradients [147], the noise is injected with localization and orientation information of the conditioned class, while maintaining the same random Gaussian distribution. Studying the resembling functionality of diffusion and energy-based models, and leveraging the compositional structure of the latter models, Liu et al. [21] propose to combine multiple diffusion models for conditional image synthesis. In the reverse process, the composition of multiple diffusion models, each associated with a different condition, can be achieved either through conjunction or negation.

2) *Score-Based Generative Models:* The works of Song et al. [4] and Dhariwal et al. [5] on score-based conditional diffusion models based on classifier guidance inspired Chao et al. [103] to develop a new training objective which reduces the potential discrepancy between the score model and the true score. The loss of the classifier is modified into a scaled cross-entropy added to a modified score matching loss.

3) *Stochastic Differential Equations:* Ho et al. [77] introduce a guidance method that does not require a classifier. It just needs one conditional diffusion model and one unconditional version,

but they use the same model to learn both cases. The unconditional model is trained with the class identifier being equal to 0. Liu et al. [91] investigate the usage of conventional numerical methods to solve the reverse ODE formulation. They find that these methods return lower quality samples compared with the previous approaches. Thus, they introduce pseudo-numerical methods for diffusion models, dividing the numerical methods into two parts, the gradient part and the transfer part. The transfer part (standard methods have a linear transfer part) is replaced such that the result is as close as possible to the target manifold.

Tachibana et al. [148] address the slow sampling problem of DDPMs. They propose to decrease the number of sampling steps by increasing the order (from one to two) of the stochastic differential equation solver (denoising part). While preserving the network architecture and score matching function, they adopt the Itô-Taylor expansion scheme for the sampler, as well as substitute some derivative terms in order to simplify the calculation. Karras et al. [105] try to separate diffusion score-based models into individual components that are independent of each other. This separation allows modifying a single part without affecting the other units, thus facilitating the improvement of diffusion models. Using this framework, the authors present a sampling process that uses Heun's method as the ODE solver, which reduces the neural function evaluations while maintaining the FID score. Studying both unconditional and class-conditional image generation, Salimans et al. [108] propose a technique to reduce the number of sampling steps. They distill the knowledge of a trained teacher model, represented by a deterministic DDIM, into a student model with the same architecture, but halving the number of sampling steps. In other words, the student learns to take two consecutive steps of the teacher. Three model versions and two loss functions are explored to facilitate the distillation process and reduce the number of sampling steps (from 8192 to 4). Campbell et al. [102] propose a continuous-time formulation of denoising diffusion models capable of working with discrete data. The work models the forward continuous-time Markov chain diffusion process via a transition rate matrix, and the backward denoising process via a parametric approximation of the inverse transition rate matrix. The interpretation of diffusion models as ODEs proposed by Song et al. [4] is reformulated by Lu et al. [107] in a form that can be solved using an exponential integrator. Other contributions of Lu et al. [107] are an ODE solver that approximates the integral term of the new formulation using Taylor expansion (first order to third order), and an algorithm that adapts the time step schedule, being 4 to 16 times faster.

### C. Image-to-Image Translation

Saharia et al. [34] propose a diffusion model for image-to-image translation, focusing on four tasks: colorization, inpainting, uncropping and JPEG restoration. Notably, the proposed framework is the same across all four tasks, meaning that it does not suffer custom changes for each task. To translate unpaired sets of images, Sasaki et al. [110] propose a method based on two jointly trained diffusion models. During denoising, at every step, each model is also conditioned on the other's intermediate

sample. The aim of Zhao et al. [35] is to improve current image-to-image translation score-based diffusion models by utilizing data from a source domain with an equal significance. An energy-based function trained on both source and target domains is employed in order to guide the SDE solver. This leads to generating images that preserve the domain-agnostic features, while translating characteristics specific to the source domain to the target domain. Leveraging the power of pretraining, Wang et al. [36] train the GLIDE model [14] to obtain a rich semantic latent space. Starting from the pretrained version and replacing the head to adapt to any conditional input, the model is fine-tuned on some specific downstream generation tasks. Li et al. [37] introduce a diffusion model for image-to-image translation that is based on Brownian bridges, as well as GANs. The proposed process begins by encoding the image with a VQ-GAN [149]. Within the resulting quantized latent space, the diffusion process, formulated as a Brownian bridge, maps between the latent representations of the source domain and target domain. Finally, another VQ-GAN decodes the quantized vectors in order to synthesize the image in the new domain. Continuing their previous work proposed in [45], Wolleb et al. [38] extend their diffusion model by replacing the classifier with another model specific to the task. Thus, at every step of the sampling process, the gradient of the task-specific network is infused.

### D. Text-to-Image Synthesis

Perhaps the most impressive results of diffusion models are attained on text-to-image synthesis, where the capability of combining unrelated concepts, such as objects, shapes and textures, to generate unusual examples comes to light. To confirm this statement, we used Stable Diffusion [10] to generate images based on various text prompts, and the results are shown in Fig. 2.

Imagen is introduced in [12] as an approach for text-to-image synthesis. It consists of one encoder for the text sequence and a cascade of diffusion models for generating high-resolution images. These models are also conditioned on the text embeddings returned by the encoder. Moreover, the authors introduce a new set of captions (DrawBench) for text-to-image evaluations. Gu et al. [111] introduce the VQ-Diffusion model, a method for text-to-image synthesis that does not have the unidirectional bias of previous approaches. With its masking mechanism, the proposed method avoids the accumulation of errors during inference. The model has two stages, where the first stage is based on a VQ-VAE that learns to represent an image via discrete tokens, and the second stage is a discrete diffusion model that operates on the discrete latent space of the VQ-VAE. Avrahami et al. [31] present a text-conditional diffusion model conditioned on CLIP [150] image and text embeddings. This is a two-stage approach, where the first stage generates the image embedding, and the second stage (decoder) produces the final image conditioned on the image embedding and the text caption. Addressing the slow sampling inconvenience of diffusion models, Zhang et al. [113] focus their work on a new discretization scheme that reduces the error and allows a greater step size, i.e., a lower number of sampling steps. By using high-order polynomial extrapolations

in the score function and an Exponential Integrator for solving the reverse SDE, the number of network evaluations is drastically reduced, while maintaining the generation capabilities. Shi et al. [9] combine a VQ-VAE [151] and a diffusion model to generate images. Starting from the VQ-VAE, the encoding functionality is preserved, while the decoder is replaced by a diffusion model. Building on top of the work presented in [116], Rombach et al. [11] introduce a modification to create artistic images using the same procedure: extract the k-nearest neighbors in the CLIP [150] latent space of an image from a database, then generate a new image by guiding the reverse denoising process with these embeddings. As the CLIP latent space is shared by text and images, the diffusion can be guided by text prompts as well. Jiang et al. [22] present a framework to generate images of full-body humans with rich clothing representation given three inputs: a human pose, a text description of the clothes' shape, and another text of the clothing texture. The method encodes the former text prompt into an embedding vector and infuses it into an auto-encoder that generates a map of forms. Next, a diffusion-based transformer samples an embedded representation of the latter text prompt from multiple multi-level codebooks (each specific to a texture), a mechanism suggested in VQ-VAE [151].

### E. Image Super-Resolution

Saharia et al. [23] apply diffusion models to super-resolution. Their reverse process learns to generate high quality images conditioned on low-resolution versions. Daniels et al. [25] use score-based models to sample from the Sinkhorn coupling of two distributions. Their method models the dual variables with neural networks, then solves the problem of optimal transport. After training the neural networks, the sampling can be performed via Langevin dynamics and a score-based model.

### F. Image Editing

Meng et al. [33] employ diffusion models in various guided image generation tasks, i.e., stroke-based editing and image composition. Images are synthesized with a generic diffusion model by solving the reverse SDE, without requiring any custom data set or modifications for training. One of the first approaches for editing specific regions of images based on natural language descriptions is introduced in [31]. The regions to be modified are entered by the user via a mask. The method relies on CLIP guidance to generate an image according to the text input. After each denoising step, the mask is applied on the latent image, while also adding the noisy version of the original image.

Extending the work presented in [10], Avrahami et al. [114] apply latent diffusion models to edit images locally, using text. A VAE encodes the image and the adaptive-to-time mask (region to edit) into the latent space where the diffusion process occurs.

### G. Image Inpainting

Nichol et al. [14] train a diffusion model conditioned on text descriptions and study the effectiveness of classifier-free and CLIP-based guidance, obtaining better results with the first option. They also fine-tune the model for image inpainting,

unlocking image modifications based on text input. Lugmay et al. [29] present an inpainting method agnostic to the mask form. They use an unconditional diffusion model for this, but modify its reverse process. They produce the image at step  $t - 1$  by sampling the known region from the masked image, and the unknown region by applying denoising to the image obtained at step  $t$ .

### H. Image Segmentation

Baranchuk et al. [39] demonstrate how diffusion models can be used in semantic segmentation. Taking the feature maps (middle blocks) at different scales from the decoder of the U-Net (used in the denoising process) and concatenating them (upsampling the feature maps in order to have the same dimensions), they can be used to classify each pixel by further attaching an ensemble of multi-layer perceptrons. Amit et al. [42] apply diffusion probabilistic models to image segmentation by extending the U-Net encoder. The input image and the current estimated image are passed through two different encoders and combined together via summation. The result is then supplied to U-Net.

### I. Multi-Task Approaches

A series of diffusion models have been applied to multiple tasks, demonstrating a good generalization capacity across tasks. We discuss such contributions below.

Song et al. [3] present the noise conditional score network (NCSN), an approach which estimates the score function at different noise scales. For sampling, they introduce an annealed version of Langevin dynamics and use it to report results in image generation and inpainting. Kadkhodaie et al. [125] train a neural network to restore images corrupted with Gaussian noise, generated using random standard deviations that are restricted to a particular range. After training, the difference between the output of the neural network and the noisy image received as input is proportional with the gradient of the log-density of the noisy data. For image generation, the authors use the difference as gradient (score) estimation and sample from the implicit data prior of the network by employing an iterative method similar to annealed Langevin dynamics [3]. Another contribution of Kadkhodaie et al. [125] is to adapt the algorithm to linear inverse problems, such as deblurring and super-resolution.

The SDE formulation of diffusion models introduced in [4] generalizes over several previous methods [1], [2], [3]. Song et al. [4] present the forward and reverse diffusion processes as solutions of SDEs. The authors carry out experiments on image generation, inpainting and colorization. Batzolis et al. [115] introduce a new forward process in diffusion models, called non-uniform diffusion. This is determined by each pixel being diffused with a different SDE. The model, whose architecture is based on [2] and [4], is evaluated on unconditional synthesis, super-resolution, inpainting and edge-to-image translation.

Esser et al. [28] propose ImageBART, a generative model which learns to revert a multinomial diffusion process on compact image representations. ImageBART is evaluated on unconditional, class-conditional and text-conditional image generation, as well as local editing. Gao et al. [117] introduce diffusion

recovery likelihood, a new training procedure for energy-based models. They learn a sequence of energy-based models for the marginal distributions of the diffusion process. The authors run experiments on both image generation and inpainting. Batzolis et al. [24] analyze the previous score-based diffusion models on conditional image generation. Moreover, they present a new method for conditional image generation called conditional multi-speed diffusive estimator (CMDE). The approach is evaluated on inpainting, super-resolution and edge-to-image synthesis.

Liu et al. [106] introduce a framework that allows text, content and style guidance from a reference image. The core idea is to use the direction that maximizes the similarity between the representations learned for image and text. The image and text embeddings are produced by CLIP [150]. Choi et al. [32] propose a novel method for conditional image synthesis using unconditional diffusion models. Given a reference image, i.e., the condition, each sample is drawn closer to it by eliminating the low frequency content and replacing it with content from the reference image. The authors show how this method can be applied on various image-to-image translation tasks, e.g., paint-to-image, and editing with scribbles.

Hu et al. [118] propose to apply diffusion models on discrete representations given by a discrete VAE. They evaluate the idea in image generation and inpainting experiments, considering the CelebA-HQ and LSUN Church data sets.

Rombach et al. [10] introduce latent diffusion models, where the forward and reverse processes happen on the latent space learned by an auto-encoder. They also include cross-attention in the architecture, which brings further improvements on conditional image synthesis, super-resolution and inpainting. The method introduced by Preechakul et al. [123] contains a semantic encoder that learns a descriptive latent space. The output of this encoder is used to condition an instance of DDIM. Chung et al. [26] introduce an algorithm for sampling, which reduces the number of steps required for the conditional case. The approach is tested on inpainting, super-resolution, and magnetic resonance imaging (MRI) reconstruction. In [120], the authors fine-tune a pretrained DDIM to generate images according to a text description. They propose a local directional CLIP loss that basically enforces the direction between the generated image and the original image to be as close as possible to the direction between the reference (original domain) and target text (target domain). Starting from the formulation of diffusion models as SDEs of Meng et al. [33], Khrulkov et al. [119] investigate the latent space and the resulting encoder maps. As per Monge formulation, it is shown that these encoder maps are the optimal transport maps, but this is demonstrated only for multivariate normal distributions. Shi et al. [124] show how an unconditional score-based diffusion model can be formulated as a Schrödinger bridge, which can be solved using a modified version of Iterative Proportional Fitting. The previous method is reformulated to accept a condition, thus making conditional synthesis possible. The authors conduct experiments on super-resolution, inpainting, and biochemical oxygen demand, the latter task being inspired by Marzouk et al. [152].

Inspired by the Retrieval Transformer [153], Blattmann et al. [116] propose a new method to train diffusion models. First, a set of similar images is fetched from a database using a nearest neighbor algorithm. The images are further encoded by an encoder with fixed parameters and projected into the CLIP [150] feature space. Finally, the reverse process of the diffusion model is conditioned on this latent space. Lyu et al. [122] introduce a new technique to reduce the number of sampling steps, boosting the performance at the same time. The idea is to stop the diffusion process at an earlier step. As the sampling cannot start from a random Gaussian noise, a GAN or VAE model is used to encode the last diffused image into a Gaussian latent space. The aim of Graikos et al. [40] is to separate diffusion models into two independent parts, a prior (the base part) and a constraint (the condition). This enables models to be applied on various tasks without further training. The authors conduct experiments on conditional image synthesis and image segmentation.

#### J. Medical Image Generation and Translation

Wolleb et al. [41] introduce a method based on diffusion models for image segmentation within the context of brain tumor segmentation. The training consists of diffusing the segmentation map, then denoising it to obtain the original image. During the backward process, the brain MR image is concatenated into the intermediate denoising steps in order to be passed through the U-Net model, thus conditioning the denoising process on it. Song et al. [129] introduce a method for score-based models that is able to solve inverse problems in medical imaging, i.e., reconstructing images from measurements. The authors carry out multiple experiments on different medical image types, including computed tomography (CT), low-dose CT and MRI.

Within the area of medical imaging, but focusing on reconstructing the images from accelerated MRI scans, Chung et al. [127] propose to solve the inverse problem using a score-based diffusion model. A score model is pretrained only on magnitude images in an unconditional setting. Then, a variance exploding SDE solver [4] is employed in the sampling process. Özbeý et al. [128] propose a diffusion model with adversarial inference. In order to increase each diffusion step, and thus make fewer steps, inspired by [99], the authors employ a GAN model in the reverse process to estimate the denoised image at every step. The aim of Hu et al. [126] is to remove the speckle noise in optical coherence tomography (OCT) b-scans. The first stage is represented by a method called self-fusion, as described in [154], where additional b-scans close to the given 2D slice of the input OCT volume are selected. The second stage consists of a diffusion model whose starting point is the weighted average of the original b-scan and its neighbors.

#### K. Anomaly Detection in Medical Images

Auto-encoders are widely used for anomaly detection [155]. Since diffusion models can be seen as a particular type of VAEs, it seems natural to employ diffusion models for the same tasks as VAEs. So far, diffusion models have shown promising results in detecting anomalies in medical images, as further discussed below.

Wyatt et al. [46] train a DDPM on healthy medical images. The anomalies are detected at inference time by subtracting the generated image from the original image. The work also proves that using simplex noise instead of Gaussian noise yields better results for this type of task.

Wolleb et al. [45] propose a weakly-supervised method based on diffusion models for anomaly detection in medical images. Given two unpaired images, one healthy and one with lesions, the former is diffused by the model. Then, the denoising process is guided by the gradient of a binary classifier in order to generate the healthy image. Pinaya et al. [44] propose a diffusion model to detect and segment anomalies in brain scans. The images are encoded by a VQ-VAE [151], and the quantized latent representation is obtained from a codebook. The diffusion model operates in this latent space. By averaging the intermediate samples from median steps of the backward process and then applying a precomputed threshold map, a binary mask implying the anomaly location is created. Sanchez et al. [130] follow the same principle for detecting and segmenting anomalies in medical images: a diffusion model generates healthy samples which are then subtracted from the original images. The input image is diffused using the model, reversing the denoising equation and nullifying the condition. Then, the conditioned backward process is applied.

### L. Video Generation

The recent progress towards making diffusion models more efficient has enabled their application in the video domain. We next present works applying diffusion models to video generation.

Ho et al. [132] introduce diffusion models to the task of video generation. The authors adopt the 3D U-Net from [156], presenting results in unconditional and text conditional video generation. Yang et al. [133] generate videos frame by frame, using diffusion models. The reverse process is entirely conditioned on a context vector provided by a convolutional recurrent neural network. Höppe et al. [134] present random mask video diffusion (RaMViD), a method which can be used for video generation and infilling. The main contribution of their work is a novel strategy for training, which randomly splits the frames into masked and unmasked frames. The work of Harvey et al. [131] introduces flexible diffusion models, a type of diffusion model that can be used with multiple sampling schemes for long video generation. As in [134], the authors train a diffusion model by randomly choosing the frames used in the diffusion and those used for conditioning the process.

### M. Other Tasks

There are some pioneering works applying diffusion models to new tasks, which have been scarcely explored via diffusion modeling. We gather and discuss such contributions below.

Luo et al. [121] apply diffusion models on 3D point cloud generation, auto-encoding, and unsupervised representation learning. They derive an objective function from the variational lower bound of the likelihood of point clouds conditioned on a shape latent. Zhou et al. [142] introduce Point-Voxel Diffusion (PVD), a novel method for shape generation which applies diffusion on

point-voxel representations. The approach addresses the tasks of shape generation and completion. Zimmermann et al. [43] show a strategy to apply score-based models for classification. They add the image label as a conditioning variable to the score function and, thanks to the ODE formulation, the conditional likelihood can be computed at inference time. Thus, the prediction is the label with the maximum likelihood. Kim et al. [139] propose to solve the image registration task using diffusion models. This is achieved via two networks, a diffusion network, as per [2], and a deformation network that is based on U-Net, as described in [157]. Given two images (one static, one moving), the role of the former network is to assess the deformation between the two images, and feed the result to the latter network, which predicts the deformation fields, enabling sample generation. Jeanneret et al. [136] apply diffusion models for counterfactual explanations. The method starts from a noised query image, and generates a sample with an unconditional DDPM. With the generated sample, the gradients required for guidance are computed. Sanchez et al. [137] adapt the work of Dhariwal et al. [5] for counterfactual image generation. As in [5], the denoising process is guided by classifier gradients to generate samples from the desired counterfactual class. The key contribution is an algorithm that inverts the deterministic sampling procedure of [7] and maps each original image to a unique latent representation.

Nie et al. [140] demonstrate how a diffusion model can be used as a defensive mechanism for adversarial attacks. To optimize the computations of solving the reverse-time SDE, the adjoint sensitivity method of Li et al. [158] is used for the gradient score calculations. In the context of few-shot learning, an image generator based on diffusion is proposed by Giannone et al. [135]. Given a small set of images that condition the synthesis, a visual transformer encodes these, and the resulting context representation is integrated (via two different techniques) into the U-Net model employed in the denoising process. Wang et al. [141] present a framework based on diffusion models for semantic image synthesis. Leveraging the U-Net architecture of diffusion models, the input noise is supplied to the encoder, while the semantic label map is passed to the decoder using multi-layer spatially-adaptive normalization operators [159]. To restore images negatively affected by various weather conditions (e.g., snow, rain), Özdenizci et al. [138] demonstrate how diffusion models can be used. They condition the denoising process on the degraded image by concatenating it channel-wise to the denoised sample, at every time step. Formulating the task of image restoration as a linear inverse problem, Kawar et al. [27] propose the use of diffusion models. Inspired by Kawar et al. [160], the linear degradation matrix is decomposed using singular value decomposition, such that both the input and the output can be mapped onto the spectral space of the matrix where the diffusion process is carried out.

### N. Theoretical Contributions

Huang et al. [59] demonstrate how the method proposed by Song et al. [4] is linked with maximizing a lower bound on the marginal likelihood of the reverse SDE. Moreover, they verify

their theoretical contribution with image generation experiments on CIFAR-10 and MNIST.

#### IV. CLOSING REMARKS AND FUTURE DIRECTIONS

In this paper, we reviewed the advancements made by the research community in developing and applying diffusion models to various computer vision tasks. We identified three primary formulations of diffusion modeling based on: DDPMs, NCSNs, and SDEs. Each formulation obtains remarkable results in image generation, surpassing GANs while increasing the diversity of the generated samples. The outstanding results of diffusion models are achieved while the research is still in its early phase. Although we observed that the main focus is on conditional and unconditional image generation, there are still many tasks to be explored and further improvements to be realized.

*Limitations.* The most significant disadvantage of diffusion models remains the need to perform multiple steps at inference time to generate only one sample. Despite the important amount of research conducted in this direction, GANs are still faster at producing images. Other issues of diffusion models can be linked to the commonly used strategy to employ CLIP embeddings for text-to-image generation. For example, Ramesh et al. [112] highlight that their model struggles to generate readable text in an image and motivate the behavior by stating that CLIP embeddings do not contain information about spelling. Therefore, when such embeddings are used for conditioning the denoising process, the model can inherit this kind of issues.

*Future Directions.* To reduce the uncertainty level, diffusion models generally avoid taking large steps during sampling. Indeed, taking small steps ensures the data sample generated at each step is explained by the learned Gaussian distribution. A similar behavior is observed when applying gradient descent to optimize neural networks. Indeed, taking a large step in the negative direction of the gradient, i.e., using a very large learning rate, can lead to updating the model to a region with high uncertainty, having no control over the loss value. In future work, transferring update rules borrowed from efficient optimizers to diffusion models could perhaps lead to a more efficient sampling (generation) process.

Aside from the current tendency of researching more efficient diffusion models, future work can study diffusion models applied in other computer vision tasks, such as image dehazing, video anomaly detection, or visual question answering. Even if we found some works studying anomaly detection in medical images [44], [45], [46], this task could also be explored in other domains, such as video surveillance or industrial inspection.

An interesting research direction is to assess the quality and utility of the representation space learned by diffusion models in discriminative tasks. This could be carried out in at least two distinct ways. In a direct way, by learning some discriminative model on top of the latent representations provided by a denoising model, to address some classification or regression task. In an indirect way, by augmenting training sets with realistic samples generated by diffusion models. The latter direction might be

more suitable for tasks such as object detection, where inpainting diffusion models could do a good job at blending in new objects in images.

Recent diffusion models [132] have shown impressive text-to-video synthesis capabilities compared to the previous state of the art, significantly reducing the number of artifacts and reaching an unprecedented generative performance. However, we believe this direction requires more attention in future work, as the generated videos are rather short. Hence, modeling long-term temporal relations and interactions between objects remains an open challenge.

In future, the research on diffusion models can also be expanded towards learning multi-purpose models that solve multiple tasks at once [34]. Creating a diffusion model to generate multiple types of outputs, while being conditioned on various types of data, e.g., text, class labels or images, might take us closer to understanding the necessary steps towards developing artificial general intelligence (AGI).

#### REFERENCES

- [1] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, “Deep unsupervised learning using non-equilibrium thermodynamics,” in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 2256–2265.
- [2] J. Ho, A. Jain, and P. Abbeel, “Denoising diffusion probabilistic models,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2020, pp. 6840–6851.
- [3] Y. Song and S. Ermon, “Generative modeling by estimating gradients of the data distribution,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2019, pp. 11918–11930.
- [4] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, “Score-based generative modeling through stochastic differential equations,” in *Proc. Int. Conf. Learn. Representations*, 2021.
- [5] P. Dhariwal and A. Nichol, “Diffusion models beat GANs on image synthesis,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2021, pp. 8780–8794.
- [6] A. Q. Nichol and P. Dhariwal, “Improved denoising diffusion probabilistic models,” in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 8162–8171.
- [7] J. Song, C. Meng, and S. Ermon, “Denoising diffusion implicit models,” in *Proc. Int. Conf. Learn. Representations*, 2021.
- [8] D. Watson, W. Chan, J. Ho, and M. Norouzi, “Learning fast samplers for diffusion models by differentiating through sample quality,” in *Proc. Int. Conf. Learn. Representations*, 2021.
- [9] J. Shi, C. Wu, J. Liang, X. Liu, and N. Duan, “DiVAE: Photorealistic images synthesis with denoising diffusion decoder,” 2022, *arXiv:2206.00386*.
- [10] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, “High-resolution image synthesis with latent diffusion models,” in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 10684–10695.
- [11] R. Rombach, A. Blattmann, and B. Ommer, “Text-guided synthesis of artistic images with retrieval-augmented diffusion models,” 2022, *arXiv:2207.13038*.
- [12] C. Saharia et al., “Photorealistic text-to-image diffusion models with deep language understanding,” 2022, *arXiv:2205.11487*.
- [13] Y. Song and S. Ermon, “Improved techniques for training score-based generative models,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2020, pp. 12438–12448.
- [14] A. Nichol et al., “GLIDE: Towards photorealistic image generation and editing with text-guided diffusion models,” in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 16784–16804.
- [15] Y. Song, C. Durkan, I. Murray, and S. Ermon, “Maximum likelihood training of score-based diffusion models,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2021, pp. 1415–1428.
- [16] A. Sinha, J. Song, C. Meng, and S. Ermon, “D2C: Diffusion-decoding models for few-shot conditional generation,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2021, pp. 12533–12548.
- [17] A. Vahdat, K. Kreis, and J. Kautz, “Score-based generative modeling in latent space,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2021, pp. 11287–11302.

- [18] K. Pandey, A. Mukherjee, P. Rai, and A. Kumar, “VAEs meet diffusion models: Efficient and high-fidelity generation,” in *Proc. NeurIPS Workshop DGMs Appl.*, 2021.
- [19] F. Bao, C. Li, J. Zhu, and B. Zhang, “Analytic-DPM: An analytic estimate of the optimal reverse variance in diffusion probabilistic models,” in *Proc. Int. Conf. Learn. Representations*, 2022.
- [20] T. Dockhorn, A. Vahdat, and K. Kreis, “Score-based generative modeling with critically-damped Langevin diffusion,” in *Proc. Int. Conf. Learn. Representations*, 2022.
- [21] N. Liu, S. Li, Y. Du, A. Torralba, and J. B. Tenenbaum, “Compositional visual generation with composable diffusion models,” in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 423–439.
- [22] Y. Jiang, S. Yang, H. Qiu, W. Wu, C. C. Loy, and Z. Liu, “Text2Human: Text-driven controllable human image generation,” *ACM Trans. Graph.*, vol. 41, no. 4, pp. 1–11, 2022.
- [23] C. Saharia, J. Ho, W. Chan, T. Salimans, D. J. Fleet, and M. Norouzi, “Image super-resolution via iterative refinement,” 2021, *arXiv:2104.07636*.
- [24] G. Batzolos, J. Stanezuk, C.-B. Schönlieb, and C. Etmann, “Conditional image generation with score-based diffusion models,” 2021, *arXiv:2111.13606*.
- [25] M. Daniels, T. Maunu, and P. Hand, “Score-based generative neural networks for large-scale optimal transport,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2021, pp. 12955–12965.
- [26] H. Chung, B. Sim, and J. C. Ye, “Come-closer-diffuse-faster: Accelerating conditional diffusion models for inverse problems through stochastic contraction,” in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 12413–12422.
- [27] B. Kawar, M. Elad, S. Ermon, and J. Song, “Denoising diffusion restoration models,” in *Proc. DGM4HSD*, 2022.
- [28] P. Esser, R. Rombach, A. Blattmann, and B. Ommer, “ImageBART: Bidirectional context with multinomial diffusion for autoregressive image synthesis,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2021, pp. 3518–3532.
- [29] A. Lugmayr, M. Danelljan, A. Romero, F. Yu, R. Timofte, and L. Van Gool, “RePaint: Inpainting using denoising diffusion probabilistic models,” in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 11461–11471.
- [30] B. Jing, G. Corso, R. Berlinghieri, and T. Jaakkola, “Subspace diffusion generative models,” 2022, *arXiv:2205.01490*.
- [31] O. Avrahami, D. Lischinski, and O. Fried, “Blended diffusion for text-driven editing of natural images,” in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 18208–18218.
- [32] J. Choi, S. Kim, Y. Jeong, Y. Gwon, and S. Yoon, “ILVR: Conditioning method for denoising diffusion probabilistic models,” in *Proc. Int. Conf. Comput. Vis.*, 2021, pp. 14347–14356.
- [33] C. Meng, Y. Song, J. Song, J. Wu, J.-Y. Zhu, and S. Ermon, “SDEdit: Guided image synthesis and editing with stochastic differential equations,” in *Proc. Int. Conf. Learn. Representations*, 2021.
- [34] C. Saharia et al., “Palette: Image-to-image diffusion models,” in *Proc. ACM SIGGRAPH Conf.*, 2022, pp. 1–10.
- [35] M. Zhao, F. Bao, C. Li, and J. Zhu, “EGSDE: Unpaired image-to-image translation via energy-guided stochastic differential equations,” 2022, *arXiv:2207.06635*.
- [36] T. Wang et al., “Pretraining is all you need for image-to-image translation,” 2022, *arXiv:2205.12952*.
- [37] B. Li, K. Xue, B. Liu, and Y.-K. Lai, “VQBB: Image-to-image translation with vector quantized brownian bridge,” 2022, *arXiv:2205.07680*.
- [38] J. Wolleb, R. Sandkühler, F. Bieder, and P. C. Cattin, “The Swiss army knife for image-to-image translation: Multi-task diffusion models,” 2022, *arXiv:2204.02641*.
- [39] D. Baranchuk, I. Rubachev, A. Voynov, V. Khurkov, and A. Babenko, “Label-efficient semantic segmentation with diffusion models,” in *Proc. Int. Conf. Learn. Representations*, 2022.
- [40] A. Graikos, N. Malkin, N. Jovic, and D. Samaras, “Diffusion models as plug-and-play priors,” 2022, *arXiv:2206.09012*.
- [41] J. Wolleb, R. Sandkühler, F. Bieder, P. Valmaggia, and P. C. Cattin, “Diffusion models for implicit image segmentation ensembles,” in *Proc. Int. Conf. Med. Imag. Deep Learn.*, 2022, pp. 1336–1348.
- [42] T. Amit, E. Nachmani, T. Shaharbany, and L. Wolf, “SegDiff: Image segmentation with diffusion probabilistic models,” 2021, *arXiv:2112.00390*.
- [43] R. S. Zimmermann, L. Schott, Y. Song, B. A. Dunn, and D. A. Klindt, “Score-based generative classifiers,” in *Proc. NeurIPS Workshop DGMs Appl.*, 2021.
- [44] W. H. Pinaya et al., “Fast unsupervised brain anomaly detection and segmentation with diffusion models,” 2022, *arXiv:2206.03461*.
- [45] J. Wolleb, F. Bieder, R. Sandkühler, and P. C. Cattin, “Diffusion models for medical anomaly detection,” 2022, *arXiv:2203.04306*.
- [46] J. Wyatt, A. Leach, S. M. Schmon, and C. G. Willcocks, “AnoDDPM: Anomaly detection with denoising diffusion probabilistic models using simplex noise,” in *Proc. Conf. Comput. Vis. Pattern Recognit. Workshop*, 2022, pp. 650–656.
- [47] Y. Bengio, A. Courville, and P. Vincent, “Representation learning: A review and new perspectives,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013.
- [48] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.
- [49] G. E. Hinton and R. R. Salakhutdinov, “Reducing the dimensionality of data with neural networks,” *Science*, vol. 313, no. 5786, pp. 504–507, 2006.
- [50] D. P. Kingma and M. Welling, “Auto-Encoding Variational Bayes,” in *Proc. Int. Conf. Learn. Representations*, 2014.
- [51] I. Higgins et al., “Beta-VAE: Learning basic visual concepts with a constrained variational framework,” in *Proc. Int. Conf. Learn. Representations*, 2017.
- [52] I. Goodfellow et al., “Generative adversarial nets,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.
- [53] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, “Unsupervised learning of visual features by contrasting cluster assignments,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2020, pp. 9912–9924.
- [54] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, “A simple framework for contrastive learning of visual representations,” in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 1597–1607.
- [55] F.-A. Croitoru, D.-N. Grigore, and R. T. Ionescu, “Discriminability-enforcing loss to improve representation learning,” in *Proc. Conf. Comput. Vis. Pattern Recognit. Workshop*, 2022, pp. 2598–2602.
- [56] A. van den Oord, Y. Li, and O. Vinyals, “Representation learning with contrastive predictive coding,” 2018, *arXiv: 1807.03748*.
- [57] S. Laine and T. Aila, “Temporal ensembling for semi-supervised learning,” in *Proc. Int. Conf. Learn. Representations*, 2017.
- [58] A. Tarvainen and H. Valpola, “Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 1195–1204.
- [59] C.-W. Huang, J. H. Lim, and A. C. Courville, “A variational perspective on diffusion-based generative models and score matching,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2021, pp. 22863–22876.
- [60] Y. LeCun, S. Chopra, R. Hadsell, M. Ranzato, and F. J. Huang, “A tutorial on energy-based learning,” in *Predicting Structured Data*. Cambridge, MA, USA: MIT Press, 2006.
- [61] J. Ngiam, Z. Chen, P. W. Koh, and A. Y. Ng, “Learning deep energy models,” in *Proc. Int. Conf. Mach. Learn.*, 2011, pp. 1105–1112.
- [62] A. van den Oord, N. Kalchbrenner, L. Espeholt, K. Kavukcuoglu, O. Vinyals, and A. Graves, “Conditional image generation with Pixel-CNN decoders,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2016, pp. 4797–4805.
- [63] L. Dinh, D. Krueger, and Y. Bengio, “NICE: Non-linear independent components estimation,” in *Proc. Int. Conf. Learn. Representations*, 2015.
- [64] L. Dinh, J. Sohl-Dickstein, and S. Bengio, “Density estimation using Real NVP,” in *Proc. Int. Conf. Learn. Representations*, 2017.
- [65] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional networks for biomedical image segmentation,” in *Proc. 18th Int. Conf. Med. Image Comput. Comput. Assist. Interv.*, 2015, pp. 234–241.
- [66] W. Feller, “On the Theory of stochastic processes, with particular reference to applications,” in *Proc. 1st Berkeley Symp. Math. Statist. Probability*, 1949, pp. 403–432.
- [67] P. Vincent, “A connection between score matching and denoising autoencoders,” *Neural Comput.*, 2011, pp. 1661–1674.
- [68] Y. Song, S. Garg, J. Shi, and S. Ermon, “Sliced score matching: A scalable approach to density and score estimation,” in *Proc. Uncertainty Artif. Intell.*, 2019, Art. no. 204.
- [69] B. D. Anderson, “Reverse-time diffusion equation models,” *Stochastic Processes Appl.*, vol. 12, no. 3, pp. 313–326, 1982.
- [70] T. Salimans, A. Karpathy, X. Chen, and D. P. Kingma, “PixelCNN++: Improving the PixelCNN with discretized logistic mixture likelihood and other modifications,” in *Proc. Int. Conf. Learn. Representations*, 2017.
- [71] Q. Zhang and Y. Chen, “Diffusion normalizing flow,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2021, pp. 16280–16291.

- [72] K. Swersky, M. Ranzato, D. Buchman, B. M. Marlin, and N. Freitas, "On autoencoders and score matching for energy based models," in *Proc. Int. Conf. Mach. Learn.*, 2011, pp. 1201–1208.
- [73] F. Bao, K. Xu, C. Li, L. Hong, J. Zhu, and B. Zhang, "Variational (gradient) estimate of the score function in energy-based latent variable models," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 651–661.
- [74] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training GANs," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2016, pp. 2234–2242.
- [75] Y. Shen, J. Gu, X. Tang, and B. Zhou, "Interpreting the latent space of gans for semantic face editing," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 9240–9249.
- [76] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," in *Proc. Int. Conf. Learn. Representations*, 2016.
- [77] J. Ho and T. Salimans, "Classifier-free diffusion guidance," in *Proc. NeurIPS Workshop DGMs Appl.*, 2021.
- [78] J. Austin, D. D. Johnson, J. Ho, D. Tarlow, and R. van den Berg, "Structured denoising diffusion models in discrete state-spaces," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2021, pp. 17981–17993.
- [79] Y. Benny and L. Wolf, "Dynamic dual-output diffusion models," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 11482–11491.
- [80] S. Bond-Taylor, P. Hessey, H. Sasaki, T. P. Breckon, and C. G. Willcocks, "Unleashing transformers: Parallel token prediction with discrete absorbing diffusion for fast high-resolution image generation from vector-quantized codes," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 170–188.
- [81] J. Choi, J. Lee, C. Shin, S. Kim, H. Kim, and S. Yoon, "Perception prioritized training of diffusion models," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 11472–11481.
- [82] V. De Bortoli, J. Thornton, J. Heng, and A. Doucet, "Diffusion Schrödinger bridge with applications to score-based generative modeling," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2021, pp. 17695–17709.
- [83] J. Deasy, N. Simidjievski, and P. Liò, "Heavy-tailed denoising score matching," 2021, *arXiv:2112.09788*.
- [84] K. Deja, A. Kuzina, T. Trzcinski, and J. M. Tomczak, "On analyzing generative and denoising capabilities of diffusion-based deep generative models," 2022, *arXiv:2206.00070*.
- [85] A. Jolicoeur-Martineau, R. Piché-Taillefer, I. Mitliagkas, and R. T. des Combes, "Adversarial score matching and improved sampling for image generation," in *Proc. Int. Conf. Learn. Representations*, 2021.
- [86] A. Jolicoeur-Martineau, K. Li, R. Piché-Taillefer, T. Kachman, and I. Mitliagkas, "Gotta go fast when generating data with score-based models," 2021, *arXiv:2105.14080*.
- [87] D. Kim, B. Na, S. J. Kwon, D. Lee, W. Kang, and I.-C. Moon, "Maximum likelihood training of implicit nonlinear diffusion models," 2022, *arXiv:2205.13699*.
- [88] D. Kingma, T. Salimans, B. Poole, and J. Ho, "Variational diffusion models," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2021, pp. 21696–21707.
- [89] Z. Kong and W. Ping, "On fast sampling of diffusion probabilistic models," in *Proc. INNF+ Conf.*, 2021.
- [90] M. W. Lam, J. Wang, R. Huang, D. Su, and D. Yu, "Bilateral denoising diffusion models," 2021, *arXiv:2108.11514*.
- [91] L. Liu, Y. Ren, Z. Lin, and Z. Zhao, "Pseudo numerical methods for diffusion models on manifolds," in *Proc. Int. Conf. Learn. Representations*, 2022.
- [92] H. Ma, L. Zhang, X. Zhu, J. Zhang, and J. Feng, "Accelerating score-based generative models for high-resolution image synthesis," 2022, *arXiv:2206.04029*.
- [93] E. Nachmani, R. S. Roman, and L. Wolf, "Non-Gaussian denoising diffusion models," 2021, *arXiv:2106.07582*.
- [94] R. San-Roman, E. Nachmani, and L. Wolf, "Noise estimation for generative diffusion models," 2021, *arXiv:2104.02600*.
- [95] V. Schwag, C. Hazirbas, A. Gordo, F. Ozgenel, and C. Canton, "Generating high fidelity data from low-density regions using diffusion models," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 11492–11501.
- [96] G. Wang, Y. Jiao, Q. Xu, Y. Wang, and C. Yang, "Deep generative learning via Schrödinger bridge," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 10794–10804.
- [97] Z. Wang, H. Zheng, P. He, W. Chen, and M. Zhou, "Diffusion-GAN: Training GANs with diffusion," 2022, *arXiv:2206.02262*.
- [98] D. Watson, J. Ho, M. Norouzi, and W. Chan, "Learning to efficiently sample from diffusion probabilistic models," 2021, *arXiv:2106.03802*.
- [99] Z. Xiao, K. Kreis, and A. Vahdat, "Tackling the generative learning trilemma with denoising diffusion GANs," in *Proc. Int. Conf. Learn. Representations*, 2022.
- [100] H. Zheng, P. He, W. Chen, and M. Zhou, "Truncated diffusion probabilistic models," 2022, *arXiv:2202.09671*.
- [101] F. Bordes, R. Balestrieri, and P. Vincent, "High fidelity visualization of what your self-supervised representation knows about," *Trans. Mach. Learn. Res.*, 2022. [Online]. Available: <https://jmlr.org/tmlr/papers/bib/urfWb7VjmL.bib>
- [102] A. Campbell, J. Benton, V. De Bortoli, T. Rainforth, G. Deligiannidis, and A. Doucet, "A continuous time framework for discrete denoising models," 2022, *arXiv:2205.14987*.
- [103] C.-H. Chao et al., "Denoising likelihood score matching for conditional score-based data generation," in *Proc. Int. Conf. Learn. Representations*, 2022.
- [104] J. Ho, C. Saharia, W. Chan, D. J. Fleet, M. Norouzi, and T. Salimans, "Cascaded diffusion models for high fidelity image generation," *J. Mach. Learn. Res.*, vol. 23, no. 47, pp. 1–33, 2022.
- [105] T. Karras, M. Aittala, T. Aila, and S. Laine, "Elucidating the design space of diffusion-based generative models," 2022, *arXiv:2206.00364*.
- [106] X. Liu et al., "More control for free! Image synthesis with semantic diffusion guidance," 2021, *arXiv:2112.05744*.
- [107] C. Lu, Y. Zhou, F. Bao, J. Chen, C. Li, and J. Zhu, "DPM-Solver: A fast ODE solver for diffusion probabilistic model sampling in around 10 steps," 2022, *arXiv:2206.00927*.
- [108] T. Salimans and J. Ho, "Progressive distillation for fast sampling of diffusion models," in *Proc. Int. Conf. Learn. Representations*, 2022.
- [109] V. Singh, S. Jandial, A. Chopra, S. Ramesh, B. Krishnamurthy, and V. N. Balasubramanian, "On conditioning the input noise for controlled image generation with diffusion models," 2022, *arXiv:2205.03859*.
- [110] H. Sasaki, C. G. Willcocks, and T. P. Breckon, "UNIT-DDPM: Unpaired image translation with denoising diffusion probabilistic models," 2021, *arXiv:2104.05358*.
- [111] S. Gu et al., "Vector quantized diffusion model for text-to-image synthesis," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 10696–10706.
- [112] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, "Hierarchical text-conditional image generation with CLIP latents," 2022, *arXiv:2204.06125*.
- [113] Q. Zhang and Y. Chen, "Fast sampling of diffusion models with exponential integrator," 2022, *arXiv:2204.13902*.
- [114] O. Avrahami, O. Fried, and D. Lischinski, "Blended latent diffusion," 2022, *arXiv:2206.02779*.
- [115] G. Batzolis, J. Stanczuk, C.-B. Schönlieb, and C. Etmann, "Non-uniform diffusion models," 2022, *arXiv:2207.09786*.
- [116] A. Blattmann, R. Rombach, K. Oktay, and B. Ommer, "Retrieval-augmented diffusion models," 2022, *arXiv:2204.11824*.
- [117] R. Gao, Y. Song, B. Poole, Y. N. Wu, and D. P. Kingma, "Learning energy-based models by diffusion recovery likelihood," in *Proc. Int. Conf. Learn. Representations*, 2021.
- [118] M. Hu, Y. Wang, T.-J. Cham, J. Yang, and P. N. Suganthan, "Global context with discrete diffusion in vector quantised modelling for image generation," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 11502–11511.
- [119] V. Khrulkov and I. Oseledets, "Understanding DDPM latent codes through optimal transport," 2022, *arXiv:2202.07477*.
- [120] G. Kim, T. Kwon, and J. C. Ye, "DiffusionCLIP: Text-guided diffusion models for robust image manipulation," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 2426–2435.
- [121] S. Luo and W. Hu, "Diffusion probabilistic models for 3D point cloud generation," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 2837–2845.
- [122] Z. Lyu, X. Xu, C. Yang, D. Lin, and B. Dai, "Accelerating diffusion models via early stop of the diffusion process," 2022, *arXiv:2205.12524*.
- [123] K. Preechakul, N. Chathee, S. Wizadwongsu, and S. Suwajanakorn, "Diffusion autoencoders: Toward a meaningful and decodable representation," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 10619–10629.
- [124] Y. Shi, V. De Bortoli, G. Deligiannidis, and A. Doucet, "Conditional simulation using diffusion Schrödinger bridges," in *Proc. Uncertainty Artif. Intell.*, 2022, pp. 1792–1802.
- [125] Z. Kadkhodaie and E. P. Simoncelli, "Stochastic solutions for linear inverse problems using the prior implicit in a denoiser," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2021, pp. 13242–13254.
- [126] D. Hu, Y. K. Tao, and I. Oguz, "Unsupervised denoising of retinal OCT with diffusion probabilistic model," in *Proc. SPIE Med. Imag.*, 2022, pp. 25–34.
- [127] H. Chung and J. C. Ye, "Score-based diffusion models for accelerated MRI," *Med. Image Anal.*, vol. 80, 2022, Art. no. 102479.

- [128] M. Özbeý et al., “Unsupervised medical image translation with adversarial diffusion models,” 2022, *arXiv:2207.08208*.
- [129] Y. Song, L. Shen, L. Xing, and S. Ermon, “Solving inverse problems in medical imaging with score-based generative models,” in *Proc. Int. Conf. Learn. Representations*, 2022.
- [130] P. Sanchez, A. Kascenas, X. Liu, A. Q. O’Neil, and S. A. Tsaftaris, “What is healthy? Generative counterfactual diffusion for lesion localization,” in *Proc. DGM4MICCAI*, 2022.
- [131] W. Harvey, S. Naderiparizi, V. Masrani, C. Weilbach, and F. Wood, “Flexible diffusion modeling of long videos,” 2022, *arXiv:2205.11495*.
- [132] J. Ho, T. Salimans, A. A. Gritsenko, W. Chan, M. Norouzi, and D. J. Fleet, “Video diffusion models,” in *Proc. DGM4HSD*, 2022.
- [133] R. Yang, P. Srivastava, and S. Mandt, “Diffusion probabilistic modeling for video generation,” 2022, *arXiv:2203.09481*.
- [134] T. Höppe, A. Mehrjou, S. Bauer, D. Nielsen, and A. Dittadi, “Diffusion models for video prediction and infilling,” *Trans. Mach. Learn. Res.*, 2022. [Online]. Available: <https://jmlr.org/tmlr/papers/bib/lf0lr4AYM6.bib>
- [135] G. Giannone, D. Nielsen, and O. Winther, “Few-shot diffusion models,” in *Proc. NeurIPS 2022 Workshop Score-Based Methods*, 2022. [Online]. Available: <https://openreview.net/forum?id=rqKTms-YHAW>
- [136] G. Jeanneret, L. Simon, and F. Jurie, “Diffusion models for counterfactual explanations,” 2022, *arXiv:2203.15636*.
- [137] P. Sanchez and S. A. Tsaftaris, “Diffusion causal models for counterfactual estimation,” in *Proc. Conf. Causal Learn. Reasoning*, 2022, pp. 1–21.
- [138] O. Özdenizci and R. Legenstein, “Restoring vision in adverse weather conditions with patch-based denoising diffusion models,” 2022, *arXiv:2207.14626*.
- [139] B. Kim, I. Han, and J. C. Ye, “DiffuseMorph: Unsupervised deformable image registration along continuous trajectory using diffusion models,” 2021, *arXiv:2112.05149*.
- [140] W. Nie, B. Guo, Y. Huang, C. Xiao, A. Vahdat, and A. Anandkumar, “Diffusion models for adversarial purification,” in *Proc. Int. Conf. Mach. Learn.*, 2022, pp. 16805–16827.
- [141] W. Wang et al., “Semantic image synthesis via diffusion models,” 2022, *arXiv:2207.00050*.
- [142] L. Zhou, Y. Du, and J. Wu, “3D shape generation and completion through point-voxel diffusion,” in *Proc. Int. Conf. Comput. Vis.*, 2021, pp. 5826–5835.
- [143] A. Vaswani et al., “Attention is all you need,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 6000–6010.
- [144] M. Shannon et al., “Non-saturating GAN training as divergence minimization,” 2020, *arXiv: 2010.08029*.
- [145] M. Arjovsky and L. Bottou, “Towards principled methods for training generative adversarial networks,” in *Proc. Int. Conf. Learn. Representations*, 2017.
- [146] C. K. Sønderby, J. Caballero, L. Theis, W. Shi, and F. Huszár, “Amortised map inference for image super-resolution,” in *Proc. Int. Conf. Learn. Representations*, 2017.
- [147] J. Geiping, H. Bauermeister, H. Dröge, and M. Moeller, “Inverting gradients – How easy is it to break privacy in federated learning?,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2020, pp. 16937–16947.
- [148] H. Tachibana, M. Go, M. Inahara, Y. Katayama, and Y. Watanabe, “Itô-Taylor sampling scheme for denoising diffusion probabilistic models using ideal derivatives,” 2021, *arXiv:2112.13339*.
- [149] P. Esser, R. Rombach, and B. Ommer, “Taming transformers for high-resolution image synthesis,” in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 12873–12883.
- [150] A. Radford et al., “Learning transferable visual models from natural language supervision,” in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 8748–8763.
- [151] A. Van Den Oord, O. Vinyals, and K. Kavukcuoglu, “Neural discrete representation learning,” *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 6309–6318.
- [152] Y. Marzouk, T. Moseley, M. Parno, and A. Spantini, “Sampling via measure transport: An introduction,” in *Handbook of Uncertainty Quantification*. Berlin, Germany: Springer, 2016, pp. 1–41.
- [153] S. Borgeaud et al., “Improving language models by retrieving from trillions of tokens,” in *Proc. Int. Conf. Mach. Learn.*, 2022, pp. 2206–2240.
- [154] I. Oguz, J. D. Malone, Y. Atay, and Y. K. Tao, “Self-fusion for OCT noise reduction,” in *Proc. SPIE Med. Imag.*, SPIE, 2020, Art. no. 113130C.
- [155] R. T. Ionescu, F. S. Khan, M.-I. Georgescu, and L. Shao, “Object-centric auto-encoders and dummy anomalies for abnormal event detection in video,” in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 7842–7851.
- [156] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, “3D U-Net: Learning dense volumetric segmentation from sparse annotation,” in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervention*, 2016, pp. 424–432.
- [157] G. Balakrishnan, A. Zhao, M. R. Sabuncu, J. Guttag, and A. V. Dalca, “An unsupervised learning model for deformable medical image registration,” in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 9252–9260.
- [158] X. Li, T.-K. L. Wong, R. T. Chen, and D. Duvenaud, “Scalable gradients for stochastic differential equations,” in *Proc. Int. Conf. Artif. Intell. Statist.*, 2020, pp. 3870–3882.
- [159] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu, “Semantic image synthesis with spatially-adaptive normalization,” in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 2337–2346.
- [160] B. Kawar, G. Vaksman, and M. Elad, “SNIPS: Solving noisy inverse problems stochastically,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2021, pp. 21757–21769.



**Florinel-Alin Croitoru** received the master’s degree in artificial intelligence, in 2021 with a thesis on action spotting in football videos. He is currently working toward the PhD degree with the University of Bucharest, Romania. His domains of interest include machine learning, computer vision, and deep learning.



**Vlad Hondu** received the bachelor’s degree in mechatronic engineering from the University of Manchester and the graduation degree from Imperial College London. He received the MSc degree in Computing Science, with a Visual Computing and Robotics specialization, focusing on Artificial Intelligence. He is currently working toward the PhD degree with the University of Bucharest, Romania.



**Radu Tudor Ionescu** (Member, IEEE) received the PhD degree from the University of Bucharest in 2013. He is professor with the University of Bucharest, Romania. He receiving the 2014 Award for Outstanding Doctoral Research from the Romanian Ad Astra Association. His research interests include machine learning, computer vision, image processing, computational linguistics, and medical imaging. He published more than 110 articles at international venues (including CVPR, NeurIPS, ICCV, ACL, EMNLP, NAACL, TPAMI, IJCV, CVIU).



**Mubarak Shah** (Life Fellow, IEEE) is the UCF Trustee chair professor and the founding director with the Center for Research in Computer Vision, the University of Central Florida (UCF). He is a fellow of the NAI, AAAS, IAPR and SPIE. He is an editor of an international book series on video computing, was editor-in-chief of Machine Vision and Applications and an associate editor of *ACM Computing Surveys* and *IEEE Transactions on Pattern Analysis and Machine Intelligence*. His research interests include video surveillance, visual tracking, human activity recognition, visual analysis of crowded scenes, video registration, UAV video analysis, among others.