Diffusion Models: A Comprehensive Survey of Methods and Applications

LING YANG and ZHILONG ZHANG, Peking University, China
YANG SONG, OpenAI, USA
SHENDA HONG, Peking University, China
RUNSHENG XU, University of California, Los Angeles, USA
YUE ZHAO, University of Southern California, USA
WENTAO ZHANG, Mila - Québec AI Institute, HEC Montréal, Canada
BIN CUI, Peking University, China
MING-HSUAN YANG, University of California at Merced and Yonsei University, USA and Korea

Diffusion models have emerged as a powerful new family of deep generative models with record-breaking performance in many applications, including image synthesis, video generation, and molecule design. In this survey, we provide an overview of the rapidly expanding body of work on diffusion models, categorizing the research into three key areas: efficient sampling, improved likelihood estimation, and handling data with special structures. We also discuss the potential for combining diffusion models with other generative models for enhanced results. We further review the wide-ranging applications of diffusion models in fields spanning from computer vision, natural language processing, temporal data modeling, to interdisciplinary applications in other scientific disciplines. This survey aims to provide a contextualized, in-depth look at the state of diffusion models, identifying the key areas of focus and pointing to potential areas for further exploration. Github: https://github.com/YangLing0818/Diffusion-Models-Papers-Survey-Taxonomy

CCS Concepts: • **Computing methodologies** → *Artificial intelligence*; *Computer vision*; *Natural language processing*; • **Applied computing** → *Life and medical sciences;*

Additional Key Words and Phrases: Generative models, diffusion models, score-based generative models, stochastic differential equations

ACM Reference format:
Ling Yang, Zhilong Zhang, Yang Song, Shenda Hong, Runsheng Xu, Yue Zhao, Wentao Zhang, Bin Cui, and Ming-Hsuan Yang. 2023. Diffusion Models: A Comprehensive Survey of Methods and Applications. *ACM Comput. Surv.* 56, 4, Article 105 (November 2023), 39 pages.
https://doi.org/10.1145/3626235

Z. Zhang contributed equally.

Authors' addresses: L. Yang, Z. Zhang, S. Hong, and B. Cui, Peking university: No.5 Yiheyuan Road Haidian District, Beijing 100871, China; e-mails: yangling@stu.pku.edu.cn, zhilong.zhang@bjmu.edu.cn, hongshenda@pku.edu.cn, bin.cui@pku.edu.cn; Y. Song, Open AI, 3180 18th St, San Francisco, California 94110, USA; e-mail: songyang@openai.com; R. Xu, University of California, 405 Hilgard Avenue, Los Angeles, CA 90095, USA; e-mail: rxx3386@ucla.edu; Y. Zhao, University of Southern California: Waite Phillips Hall, 3470 Trousdale Parkway, Los Angeles, CA 90089, USA; e-mail: yzhao010@usc.edu; W. Zhang, Mila, 6666 St-Urbain Street, Montreal, Quebec, H2S 3H1, Canada; e-mail: wentao.zhang@mila.quebec; M.-H. Yang, University of California at Merced: 5200 North Lake Rd. Merced, CA 95343, USA; e-mail: mhyang@ucmerced.edu.

0360-0300/2023/11-ART105 $15.00
https://doi.org/10.1145/3626235

1 INTRODUCTION

Diffusion models [87, 218, 223, 228] have emerged as the new state-of-the-art family of deep generative models. They have broken the long-time dominance of **generative adversarial networks (GANs)** [71] in the challenging task of image synthesis [54, 87, 223, 228] and have also shown potential in a variety of domains, ranging from computer vision [2, 11, 19, 23, 88, 90, 113, 115, 134, 150, 160, 173, 198, 200, 248, 270, 271, 283, 290], natural language processing [6, 93, 139, 205, 275], temporal data modeling [1, 33, 124, 191, 233, 262], multi-modal modeling [7, 186, 196, 199, 288], robust machine learning [18, 28, 112, 242, 273], to interdisciplinary applications in fields such as computational chemistry [3, 91, 104, 130, 132, 152, 258] and medical image reconstruction [25, 41–43, 48, 158, 177, 227, 259].

Numerous methods have been developed to improve diffusion models, either by enhancing empirical performance [165, 220, 224] or by extending the model capacity from a theoretical perspective [144, 145, 222, 228, 279]. Over the past two years, the body of research on diffusion models has grown significantly, making it increasingly challenging for new researchers to stay abreast of the recent developments in the field. Additionally, the sheer volume of work can obscure major trends and hinder further research progress. This survey aims to address these problems by providing a comprehensive overview of the state of diffusion model research, categorizing various approaches, and highlighting key advances.

In this paper, we first explain the foundations of diffusion models (Section 2), providing a brief but self-contained introduction to three predominant formulations: **denoising diffusion probabilistic models (DDPMs)** [87, 218], **score-based generative models (SGMs)** [223, 224], and **stochastic differential equations** (Score **SDEs**) [111, 222, 228]. Key to all these approaches is to progressively perturb data with intensifying random noise (called the "diffusion" process), then successively remove noise to generate new data samples. We clarify how they work under the same principle of diffusion and explain how these three models are connected and can be reduced to one another.

Next, we present a taxonomy of recent research that maps out the field of diffusion models, categorizing it into three key areas: efficient sampling (Section 3), improved likelihood estimation (Section 4), and methods for handling data with special structures (Section 5), such as relational data, data with permutation/rotational invariance, and data residing on manifolds. We further examine the models by breaking each category into more detailed sub-categories, as illustrated in Figure 1. In addition, we discuss the connections of diffusion models to other deep generative models (Section 6), including **variational autoencoders (VAEs)** [122, 194], **generative adversarial networks (GANs)** [71], normalizing flows [55, 174, 195], autoregressive models [239], and **energy-based models (EBMs)** [129, 226]. By combining these models with diffusion models, researchers have the potential to achieve even stronger performance.

Following that, our survey reviews six major categories of application that diffusion models have been applied to in the existing research (Section 7): computer vision, natural language processing, temporal data modeling, multi-modal learning, robust learning, and interdisciplinary applications. For each task, we provide a definition, describe how diffusion models can be employed to address it and summarize relevant previous work. We conclude our paper (Sections 8 and 9) by providing an outlook on possible future directions for this exciting new area of research.

2 FOUNDATIONS OF DIFFUSION MODELS

Diffusion models are a family of probabilistic generative models that progressively destruct data by injecting noise, then learn to reverse this process for sample generation. We present the intuition of diffusion models in Figure 2. Current research on diffusion models is mostly based on three

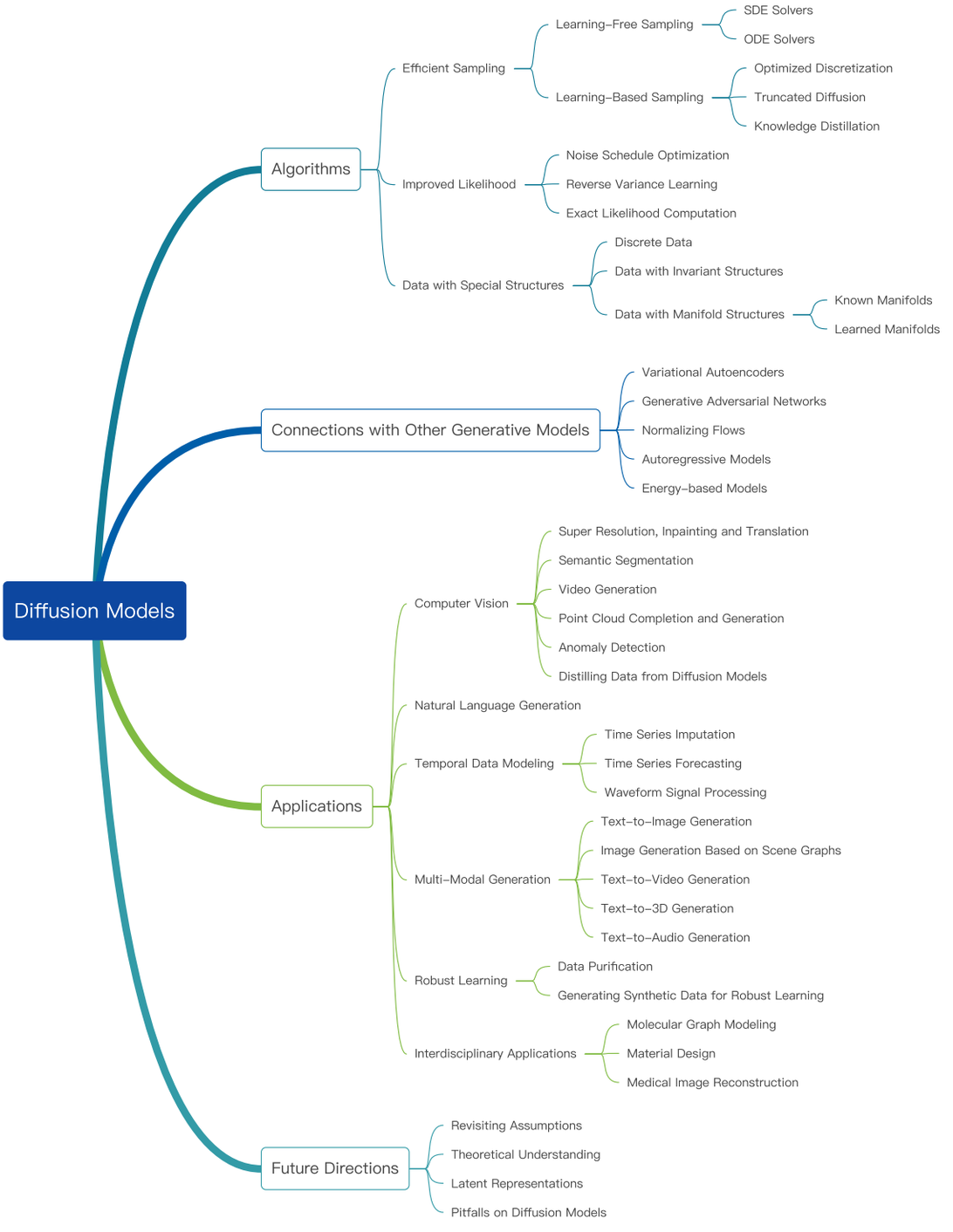


Fig. 1. Taxonomy of diffusion models variants (in Sections 3 to 5), connections with other generative models (in Section 6), applications of diffusion models (in Section 7), and future directions (in Section 8).

predominant formulations: denoising diffusion probabilistic models (DDPMs) [87, 165, 218], score-based generative models (SGMs) [223, 224], and stochastic differential equations (Score SDEs) [222, 228]. We give a self-contained introduction to these three formulations in this section, while discussing their connections with each other along the way.

2.1 Denoising Diffusion Probabilistic Models (DDPMs)

A *denoising diffusion probabilistic model* (DDPM) [87, 218] makes use of two Markov chains: a forward chain that perturbs data to noise, and a reverse chain that converts noise back to data.

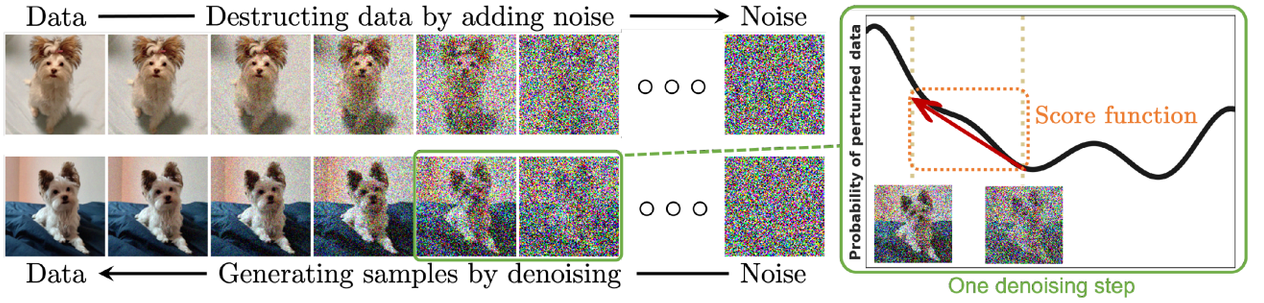


Fig. 2. Diffusion models smoothly perturb data by adding noise, then reverse this process to generate new data from noise. Each denoising step in the reverse process typically requires estimating the score function (see the illustrative figure on the right), which is a gradient pointing to the directions of data with higher likelihood and less noise.

The former is typically hand-designed with the goal to transform any data distribution into a simple prior distribution (*e.g.*, standard Gaussian), while the latter Markov chain reverses the former by learning transition kernels parameterized by deep neural networks. New data points are subsequently generated by first sampling a random vector from the prior distribution, followed by ancestral sampling through the reverse Markov chain [47].

Formally, given a data distribution $\mathbf{x}_0 \sim q(\mathbf{x}_0)$, the forward Markov process generates a sequence of random variables $\mathbf{x}_1, \mathbf{x}_2 \dots \mathbf{x}_T$ with transition kernel $q(\mathbf{x}_t \mid \mathbf{x}_{t-1})$. Using the chain rule of probability and the Markov property, we can factorize the joint distribution of $\mathbf{x}_1, \mathbf{x}_2 \dots \mathbf{x}_T$ conditioned on $\mathbf{x}_0$, denoted as $q(\mathbf{x}_1, \dots, \mathbf{x}_T \mid \mathbf{x}_0)$, into

$$q(\mathbf{x}_1, \dots, \mathbf{x}_T \mid \mathbf{x}_0) = \prod_{t=1}^{T} q(\mathbf{x}_t \mid \mathbf{x}_{t-1}). \tag{1}$$

In DDPMs, we handcraft the transition kernel $q(\mathbf{x}_t \mid \mathbf{x}_{t-1})$ to incrementally transform the data distribution $q(\mathbf{x}_0)$ into a tractable prior distribution. One typical design for the transition kernel is Gaussian perturbation, and the most common choice for the transition kernel is

$$q(\mathbf{x}_t \mid \mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t \mathbf{I}), \tag{2}$$

where $\beta_t \in (0, 1)$ is a hyperparameter chosen ahead of model training. We use this kernel to simplify our discussion here, although other types of kernels are also applicable in the same vein. As observed by Sohl-Dickstein et al. (2015) [218], this Gaussian transition kernel allows us to marginalize the joint distribution in Equation (1) to obtain the analytical form of $q(\mathbf{x}_t \mid \mathbf{x}_0)$ for all $t \in \{0, 1, \cdots, T\}$. Specifically, with $\alpha_t := 1 - \beta_t$ and $\bar{\alpha}_t := \prod_{s=0}^{t} \alpha_s$, we have

$$q(\mathbf{x}_t \mid \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I}). \tag{3}$$

Given $\mathbf{x}_0$, we can easily obtain a sample of $\mathbf{x}_t$ by sampling a Gaussian vector $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and applying the transformation

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}. \tag{4}$$

When $\bar{\alpha}_T \approx 0$, $\mathbf{x}_T$ is almost Gaussian in distribution, so we have $q(\mathbf{x}_T) := \int q(\mathbf{x}_T \mid \mathbf{x}_0) q(\mathbf{x}_0) \mathrm{d}\mathbf{x}_0 \approx \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I})$.

Intuitively speaking, this forward process slowly injects noise to data until all structures are lost. For generating new data samples, DDPMs start by first generating an unstructured noise vector from the prior distribution (which is typically trivial to obtain), then gradually remove noise therein by running a learnable Markov chain in the reverse time direction. Specifically, the

reverse Markov chain is parameterized by a prior distribution $p(\mathbf{x}_T) = \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I})$ and a learnable transition kernel $p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t)$. We choose the prior distribution $p(\mathbf{x}_T) = \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I})$ because the forward process is constructed such that $q(\mathbf{x}_T) \approx \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I})$. The learnable transition kernel $p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t)$ takes the form of

$$p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t), \Sigma_\theta(\mathbf{x}_t, t)) \tag{5}$$

where θ denotes model parameters, and the mean $\mu_\theta(\mathbf{x}_t, t)$ and variance $\Sigma_\theta(\mathbf{x}_t, t)$ are parameterized by deep neural networks. With this reverse Markov chain in hand, we can generate a data sample $\mathbf{x}_0$ by first sampling a noise vector $\mathbf{x}_T \sim p(\mathbf{x}_T)$, then iteratively sampling from the learnable transition kernel $\mathbf{x}_{t-1} \sim p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t)$ until $t = 1$.

Key to the success of this sampling process is training the reverse Markov chain to match the actual time reversal of the forward Markov chain. That is, we have to adjust the parameter θ so that the joint distribution of the reverse Markov chain $p_\theta(\mathbf{x}_0, \mathbf{x}_1, \cdots, \mathbf{x}_T) \coloneqq p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t)$ closely approximates that of the forward process $q(\mathbf{x}_0, \mathbf{x}_1, \cdots, \mathbf{x}_T) \coloneqq q(\mathbf{x}_0) \prod_{t=1}^{T} q(\mathbf{x}_t \mid \mathbf{x}_{t-1})$ (Equation (1)). This is achieved by minimizing the **Kullback-Leibler (KL)** divergence between these two:

$$\mathrm{KL}(q(\mathbf{x}_0, \mathbf{x}_1, \cdots, \mathbf{x}_T) \,||\, p_\theta(\mathbf{x}_0, \mathbf{x}_1, \cdots, \mathbf{x}_T)) \tag{6}$$

$$\overset{(i)}{=} -\mathbb{E}_{q(\mathbf{x}_0, \mathbf{x}_1, \cdots, \mathbf{x}_T)}[\log p_\theta(\mathbf{x}_0, \mathbf{x}_1, \cdots, \mathbf{x}_T)] + \text{const} \tag{7}$$

$$\overset{(ii)}{=} \underbrace{\mathbb{E}_{q(\mathbf{x}_0, \mathbf{x}_1, \cdots, \mathbf{x}_T)}\left[-\log p(\mathbf{x}_T) - \sum_{t=1}^{T} \log \frac{p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t)}{q(\mathbf{x}_t \mid \mathbf{x}_{t-1})} \right]}_{\coloneqq -L_{\mathrm{VLB}}(\mathbf{x}_0)} + \text{const} \tag{8}$$

$$\overset{(iii)}{\geq} \mathbb{E}\left[-\log p_\theta(\mathbf{x}_0)\right] + \text{const}, \tag{9}$$

where (i) is from the definition of KL divergence, (ii) is from the fact that $q(\mathbf{x}_0, \mathbf{x}_1, \cdots, \mathbf{x}_T)$ and $p_\theta(\mathbf{x}_0, \mathbf{x}_1, \cdots, \mathbf{x}_T)$ are both products of distributions, and (iii) is from Jensen's inequality. The first term in Equation (8) is the **variational lower bound (VLB)** of the log-likelihood of the data $\mathbf{x}_0$, a common objective for training probabilistic generative models. We use "const" to symbolize a constant that does not depend on the model parameter θ and hence does not affect optimization. The objective of DDPM training is to maximize the VLB (or equivalently, minimizing the negative VLB), which is particularly easy to optimize because it is a sum of independent terms, and can thus be estimated efficiently by Monte Carlo sampling [163] and optimized effectively by stochastic optimization [229].

Ho et al. (2020) [87] propose to reweight various terms in L_{VLB} for better sample quality and noticed an important equivalence between the resulting loss function and the training objective for **noise-conditional score networks (NCSNs)**, one type of *score-based generative models*, in Song and Ermon (2019) [223]. The loss in [87] takes the form of

$$\mathbb{E}_{t \sim \mathcal{U}[\![1,T]\!], \mathbf{x}_0 \sim q(\mathbf{x}_0), \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})}\left[\lambda(t) \left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\right\|^2\right] \tag{10}$$

where $\lambda(t)$ is a positive weighting function, $\mathbf{x}_t$ is computed from $\mathbf{x}_0$ and $\boldsymbol{\epsilon}$ by Equation (4), $\mathcal{U}[\![1, T]\!]$ is a uniform distribution over the set $\{1, 2, \cdots, T\}$, and $\boldsymbol{\epsilon}_\theta$ is a deep neural network with parameter θ that predicts the noise vector $\boldsymbol{\epsilon}$ given $\mathbf{x}_t$ and t. When $t \geq 2$, the component in Equation (10) is a denoising matching term that matches the denoising step $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ with $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$, while at $t = 1$, the corresponding component is a reconstruction term that predicts $\mathbf{x}_0$ from $\mathbf{x}_1$ [148]. This objective reduces to Equation (8) for a particular choice of the weighting function $\lambda(t)$, and has the same form as the loss of denoising score matching over multiple noise scales for training

score-based generative models [223], another formulation of diffusion models to be discussed in the next section.

2.2 Score-Based Generative Models (SGMs)

At the core of score-based generative models [223, 224] is the concept of *(Stein) score* (a.k.a. score or score function) [98]. Given a probability density function $p(\mathbf{x})$, its score function is defined as the gradient of the log probability density $\nabla_{\mathbf{x}} \log p(\mathbf{x})$. Unlike the commonly used *Fisher score* $\nabla_\theta \log p_\theta(\mathbf{x})$ in statistics, the Stein score considered here is a function of the data $\mathbf{x}$ rather than the model parameter θ. It is a vector field that points to directions along which the probability density function has the largest growth rate.

The key idea of score-based generative models (SGMs) [223] is to perturb data with a sequence of intensifying Gaussian noise and jointly estimate the score functions for all noisy data distributions by training a deep neural network model conditioned on noise levels (called a noise-conditional score network, NCSN, in [223]). Samples are generated by chaining the score functions at decreasing noise levels with score-based sampling approaches, including Langevin Monte Carlo [76, 108, 175, 223, 228], stochastic differential equations [107, 228], ordinary differential equations [111, 145, 222, 228, 279], and their various combinations [228]. Training and sampling are completely decoupled in the formulation of score-based generative models, so one can use a multitude of sampling techniques after the estimation of score functions.

With similar notations in Section 2.1, we let $q(\mathbf{x}_0)$ be the data distribution, and $0 < \sigma_1 < \sigma_2 < \cdots < \sigma_t < \cdots < \sigma_T$ be a sequence of noise levels. A typical example of SGMs involves perturbing a data point $\mathbf{x}_0$ to $\mathbf{x}_t$ by the Gaussian noise distribution $q(\mathbf{x}_t \mid \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \mathbf{x}_0, \sigma_t^2 I)$. This yields a sequence of noisy data densities $q(\mathbf{x}_1), q(\mathbf{x}_2), \cdots, q(\mathbf{x}_T)$, where $q(\mathbf{x}_t) \coloneqq \int q(\mathbf{x}_t)q(\mathbf{x}_0)\mathrm{d}\mathbf{x}_0$. A noise-conditional score network is a deep neural network $\mathbf{s}_\theta(\mathbf{x}, t)$ trained to estimate the score function $\nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t)$. Learning score functions from data (a.k.a. score estimate) has established techniques such as score matching [98], denoising score matching [188, 189, 240], and sliced score matching [225], so we can directly employ one of them to train our noise-conditional score networks from perturbed data points. For example, with denoising score matching and similar notations in Equation (10), the training objective is given by

$$\mathbb{E}_{t\sim\mathcal{U}[\![1,T]\!],\mathbf{x}_0\sim q(\mathbf{x}_0),\mathbf{x}_t\sim q(\mathbf{x}_t|\mathbf{x}_0)}\left[\lambda(t)\sigma_t^2 \left\|\nabla_{\mathbf{x}_t}\log q(\mathbf{x}_t) - \mathbf{s}_\theta(\mathbf{x}_t,t)\right\|^2\right] \tag{11}$$

$$\overset{(i)}{=}\mathbb{E}_{t\sim\mathcal{U}[\![1,T]\!],\mathbf{x}_0\sim q(\mathbf{x}_0),\mathbf{x}_t\sim q(\mathbf{x}_t|\mathbf{x}_0)}\left[\lambda(t)\sigma_t^2 \left\|\nabla_{\mathbf{x}_t}\log q(\mathbf{x}_t \mid \mathbf{x}_0) - \mathbf{s}_\theta(\mathbf{x}_t,t)\right\|^2\right] + \text{const} \tag{12}$$

$$\overset{(ii)}{=}\mathbb{E}_{t\sim\mathcal{U}[\![1,T]\!],\mathbf{x}_0\sim q(\mathbf{x}_0),\mathbf{x}_t\sim q(\mathbf{x}_t|\mathbf{x}_0)}\left[\lambda(t)\left\|-\frac{\mathbf{x}_t-\mathbf{x}_0}{\sigma_t} - \sigma_t\mathbf{s}_\theta(\mathbf{x}_t,t)\right\|^2\right] + \text{const} \tag{13}$$

$$\overset{(iii)}{=}\mathbb{E}_{t\sim\mathcal{U}[\![1,T]\!],\mathbf{x}_0\sim q(\mathbf{x}_0),\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I})}\left[\lambda(t)\left\|\boldsymbol{\epsilon} + \sigma_t\mathbf{s}_\theta(\mathbf{x}_t,t)\right\|^2\right] + \text{const}, \tag{14}$$

where (i) is derived by [240], (ii) is from the assumption that $q(\mathbf{x}_t \mid \mathbf{x}_0) = \mathbf{N}(\mathbf{x}_t; \mathbf{x}_0, \sigma_t^2\mathbf{I})$, and (iii) is from the fact that $\mathbf{x}_t = \mathbf{x}_0 + \sigma_t\boldsymbol{\epsilon}$. Again, we denote by $\lambda(t)$ a positive weighting function, and "const" a constant that does not depend on the trainable parameter θ. Comparing Equation (14) with Equation (10), it is clear that the training objectives of DDPMs and SGMs are equivalent, once we set $\boldsymbol{\epsilon}_\theta(\mathbf{x}, t) = -\sigma_t\mathbf{s}_\theta(\mathbf{x}, t)$.

For sample generation, SGMs leverage iterative approaches to produce samples from $\mathbf{s}_\theta(\mathbf{x}, T)$, $\mathbf{s}_\theta(\mathbf{x}, T-1), \cdots, \mathbf{s}_\theta(\mathbf{x}, 0)$ in succession. Many sampling approaches exist due to the decoupling of training and inference in SGMs, some of which are discussed in the next section. Here we introduce the first sampling method for SGMs, called **annealed Langevin dynamics (ALD)** [223]. Let N be the number of iterations per time step and $s_t > 0$ be the step size. We first initialize ALD with

$\mathbf{x}_T^{(N)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, then apply Langevin Monte Carlo for $t = T, T-1, \cdots, 1$ one after the other. At each time step $0 \leq t < T$, we start with $\mathbf{x}_t^{(0)} = \mathbf{x}_{t+1}^{(N)}$, before iterating according to the following update rule for $i = 0, 1, \cdots, N-1$:

$$\boldsymbol{\epsilon}^{(i)} \leftarrow \mathcal{N}(\mathbf{0}, \mathbf{I})$$
$$\mathbf{x}_t^{(i+1)} \leftarrow \mathbf{x}_t^{(i)} + \frac{1}{2} s_t \mathbf{s}_\theta(\mathbf{x}_t^{(i)}, t) + \sqrt{s_t}\boldsymbol{\epsilon}^{(i)}.$$

The theory of Langevin Monte Carlo [175] guarantees that as $s_t \to 0$ and $N \to \infty$, $\mathbf{x}_0^{(N)}$ becomes a valid sample from the data distribution $q(\mathbf{x}_0)$.

2.3 Stochastic Differential Equations (Score SDEs)

DDPMs and SGMs can be further generalized to the case of infinite time steps or noise levels, where the perturbation and denoising processes are solutions to stochastic differential equations (SDEs). We call this formulation *Score SDE* [228], as it leverages SDEs for noise perturbation and sample generation, and the denoising process requires estimating score functions of noisy data distributions.

Score SDEs perturb data to noise with a diffusion process governed by the following stochastic differential equation (SDE) [228]:

$$\mathrm{d}\mathbf{x} = \mathbf{f}(\mathbf{x}, t)\mathrm{d}t + g(t)\mathrm{d}\mathbf{w} \tag{15}$$

where $\mathbf{f}(\mathbf{x}, t)$ and $g(t)$ are diffusion and drift functions of the SDE, and $\mathbf{w}$ is a standard Wiener process (a.k.a. Brownian motion). The forward processes in DDPMs and SGMs are both discretizations of this SDE. As demonstrated in Song et al. (2020) [228], for DDPMs, the corresponding SDE is:

$$\mathrm{d}\mathbf{x} = -\frac{1}{2}\beta(t)\mathbf{x}dt + \sqrt{\beta(t)}\mathrm{d}\mathbf{w} \tag{16}$$

where $\beta(\frac{t}{T}) = T\beta_t$ as T goes to infinity; and for SGMs, the corresponding SDE is given by

$$\mathrm{d}\mathbf{x} = \sqrt{\frac{\mathrm{d}[\sigma(t)^2]}{\mathrm{d}t}}\mathrm{d}\mathbf{w}, \tag{17}$$

where $\sigma(\frac{t}{T}) = \sigma_t$ as T goes to infinity. Here we use $q_t(\mathbf{x})$ to denote the distribution of $\mathbf{x}_t$ in the forward process.

Crucially, for any diffusion process in the form of Equation (15), Anderson [4] shows that it can be reversed by solving the following reverse-time SDE:

$$\mathrm{d}\mathbf{x} = \left[\mathbf{f}(\mathbf{x}, t) - g(t)^2\nabla_{\mathbf{x}} \log q_t(\mathbf{x})\right]\mathrm{d}t + g(t)\mathrm{d}\bar{\mathbf{w}} \tag{18}$$

where $\bar{\mathbf{w}}$ is a standard Wiener process when time flows backwards, and $\mathrm{d}t$ denotes an infinitesimal negative time step. The solution trajectories of this reverse SDE share the same marginal densities as those of the forward SDE, except that they evolve in the opposite time direction [228]. Intuitively, solutions to the reverse-time SDE are diffusion processes that gradually convert noise to data. Moreover, Song et al. (2020) [228] prove the existence of an **ordinary differential equation (ODE)**, namely the *probability flow ODE*, whose trajectories have the same marginals as the reverse-time SDE. The probability flow ODE is given by:

$$\mathrm{d}\mathbf{x} = \left[\mathbf{f}(\mathbf{x}, t) - \frac{1}{2}g(t)^2\nabla_{\mathbf{x}} \log q_t(\mathbf{x})\right]\mathrm{d}t. \tag{19}$$

Both the reverse-time SDE and the probability flow ODE allow sampling from the same data distribution as their trajectories have the same marginals.

Once the score function at each time step t, $\nabla_{\mathbf{x}} \log q_t(\mathbf{x})$, is known, we unlock both the reverse-time SDE (Equation (18)) and the probability flow ODE (Equation (19)) and can subsequently generate samples by solving them with various numerical techniques, such as annealed Langevin dynamics [223] (*cf*., Section 2.2), numerical SDE solvers [107, 228], numerical ODE solvers [111, 145, 220, 228, 279], and predictor-corrector methods (combination of MCMC and numerical ODE/SDE solvers) [228]. Like in SGMs, we parameterize a time-dependent score model $\mathbf{s}_\theta(\mathbf{x}_t, t)$ to estimate the score function by generalizing the score matching objective in Equation (14) to continuous time, leading to the following objective:

$$\mathbb{E}_{t\sim\mathcal{U}[0,T],\mathbf{x}_0\sim q(\mathbf{x}_0),\mathbf{x}_t\sim q(\mathbf{x}_t|\mathbf{x}_0)}\left[\lambda(t)\left\|\mathbf{s}_\theta(\mathbf{x}_t,t)-\nabla_{\mathbf{x}_t}\log q_{0t}(\mathbf{x}_t\mid\mathbf{x}_0)\right\|^2\right], \tag{20}$$

where $\mathcal{U}[0,T]$ denotes the uniform distribution over $[0,T]$, and the remaining notations follow Equation (14).

Subsequent research on diffusion models focuses on improving these classical approaches (DDPMs, SGMs, and Score SDEs) from three major directions: faster and more efficient sampling, more accurate likelihood and density estimation, and handling data with special structures (such as permutation invariance, manifold structures, and discrete data). We survey each direction extensively in the next three sections (Sections 3 to 5).

3 DIFFUSION MODELS WITH EFFICIENT SAMPLING

Generating samples from diffusion models typically demands iterative approaches that involve a large number of evaluation steps. A great deal of recent work has focused on speeding up the sampling process while also improving quality of the resulting samples. We classify these efficient sampling methods into two main categories: those that do not involve learning (learning-free sampling) and those that require an additional learning process after the diffusion model has been trained (learning-based sampling).

3.1 Learning-Free Sampling

Many samplers for diffusion models rely on discretizing either the reverse-time SDE present in Equation (18) or the probability flow ODE from Equation (19). Since the cost of sampling increases proportionally with the number of discretized time steps, many researchers have focused on developing discretization schemes that reduce the number of time steps while also minimizing discretization errors.

3.1.1 SDE Solvers. The generation process of DDPM [87, 218] can be viewed as a particular discretization of the reverse-time SDE. As discussed in Section 2.3, the forward process of DDPM discretizes the SDE in Equation (16), whose corresponding reverse SDE takes the form of

$$\mathrm{d}\mathbf{x} = -\frac{1}{2}\beta(t)(\mathbf{x}_t - \nabla_{\mathbf{x}_t}\log q_t(\mathbf{x}_t))\mathrm{d}t + \sqrt{\beta(t)}\mathrm{d}\mathbf{w} \tag{21}$$

Song et al. (2020) [228] show that the reverse Markov chain defined by Equation (5) amounts to a numerical SDE solver for Equation (21).

Noise-Conditional Score Networks (NCSNs) [223] and **Critically-Damped Langevin Diffusion (CLD)** [59] both solve the reverse-time SDE with inspirations from Langevin dynamics. In particular, NCSNs leverage annealed Langevin dynamics (ALD, *cf*., Section 2.2) to iteratively generate data while smoothly reducing noise level until the generated data distribution converges to the original data distribution. Although the sampling trajectories of ALD are not exact solutions to the reverse-time SDE, they have the correct marginals and hence produce correct samples under the assumption that Langevin dynamics converges to its equilibrium at every noise level. The

method of ALD is further improved by **Consistent Annealed Sampling (CAS)** [108], a score-based MCMC approach with better scaling of time steps and added noise. Inspired by statistical mechanics, CLD proposes an augmented SDE with an auxiliary velocity term resembling underdamped Langevin diffusion. To obtain the time reversal of the extended SDE, CLD only needs to learn the score function of the conditional distribution of velocity given data, arguably easier than learning scores of data directly. The added velocity term is reported to improve sampling speed as well as quality.

The reverse diffusion method proposed in [228] discretizes the reverse-time SDE in the same way as the forward one. For any one-step discretization of the forward SDE, one may write the general form below:

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \mathbf{f}_i(\mathbf{x}_i) + \mathbf{g}_i \mathbf{z}_i, \quad i = 0, 1, \cdots, N-1 \tag{22}$$

where $\mathbf{z}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, $\mathbf{f}_i$ and $\mathbf{g}_i$ are determined by drift/diffusion coefficients of the SDE and the discretization scheme. Reverse diffusion proposes to discretize the reverse-time SDE similarly to the forward SDE, *i.e.*,

$$\mathbf{x}_i = \mathbf{x}_{i+1} - \mathbf{f}_{i+1}(\mathbf{x}_{i+1}) + \mathbf{g}_{i+1}\mathbf{g}_{i+1}^t \mathbf{s}_{\theta^*}(\mathbf{x}_{i+1}, t_{i+1}) + \mathbf{g}_{i+1}\mathbf{z}_i \quad i = 0, 1, \cdots, N-1 \tag{23}$$

where $\mathbf{s}_{\theta^*}(\mathbf{x}_i, t_i)$ is the trained noise-conditional score model. Song et al. (2020) [228] prove that the reverse diffusion method is a numerical SDE solver for the reverse-time SDE in Equation (18). This process can be applied to any types of forward SDEs, and empirical results indicate this sampler performs slightly better than DDPM [228] for a particular type of SDEs called the VP-SDE.

Jolicoeur-Martineau et al. (2021) [107] develop an SDE solver with adaptive step sizes for faster generation. The step size is controlled by comparing the output of a high-order SDE solver versus the output of a low-order SDE solver. At each time step, the high- and low-order solvers generate new sample $\mathbf{x}'_{\text{high}}$ and $\mathbf{x}'_{\text{low}}$ from the previous sample $\mathbf{x}'_{prev}$ respectively. The step size is then adjusted by comparing the difference between the two samples. If $\mathbf{x}'_{\text{high}}$ and $\mathbf{x}'_{\text{low}}$ are similar, the algorithm will return $\mathbf{x}'_{\text{high}}$ and then increase the step size. The similarity between $\mathbf{x}'_{\text{high}}$ and $\mathbf{x}'_{\text{low}}$ is measured by:

$$E_q = \left\| \frac{\mathbf{x}'_{\text{low}} - \mathbf{x}'_{\text{high}}}{\delta(\mathbf{x}', \mathbf{x}'_{\text{prev}})} \right\|^2 \tag{24}$$

where $\delta(\mathbf{x}'_{\text{low}}, \mathbf{x}'_{\text{prev}}) := \max(\epsilon_{abs}, \epsilon_{rel} \max(|\ \mathbf{x}'_{\text{low}}, |\ \mathbf{x}'_{\text{prev}}|))$, and ϵ_{abs} and ϵ_{rel} are absolute and relative tolerances.

The predictor-corrector method proposed in [228] solves the reverse SDE by combining numerical SDE solvers ("predictor") and iterative **Markov chain Monte Carlo (MCMC)** approaches ("corrector"). At each time step, the predictor-corrector method first employs a numerical SDE solver to produce a coarse sample, followed by a "corrector" that corrects the sample' marginal distribution with score-based MCMC. The resulting samples have the same time-marginals as solution trajectories of the reverse-time SDE, *i.e.*, they are equivalent in distribution at all time steps. Empirical results demonstrate that adding a corrector based on Langevin Monte Carlo is more efficient than using an additional predictor without correctors [228]. Karras et al. (2022) [111] further improve the Langevin dynamics corrector in [228] by proposing a Langevin-like "churn" step of adding and removing noise, achieving new state-of-the-art sample quality on datasets like CIFAR-10 [126] and ImageNet-64 [53].

3.1.2 ODE Solvers. A large body of works on faster diffusion samplers are based on solving the probability flow ODE (Equation (19)) introduced in Section 2.3. In contrast to SDE solvers, the trajectories of ODE solvers are deterministic and thus not affected by stochastic fluctuations. These

deterministic ODE solvers typically converge much faster than their stochastic counterparts at the cost of slightly inferior sample quality.

Denoising Diffusion Implicit Models (DDIM) [220] is one of the earliest work on accelerating diffusion model sampling. The original motivation was to extend the original DDPM to a non-Markovian case with the following Markov chain

$$q(\mathbf{x}_1, \ldots, \mathbf{x}_T \mid \mathbf{x}_0) = \prod_{t=1}^{T} q(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{x}_0) \tag{25}$$

$$q_\sigma(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_{t-1} | \tilde{\mu}_t(\mathbf{x}_t, \mathbf{x}_0), \sigma_t^2 \mathbf{I}) \tag{26}$$

$$\tilde{\mu}_t(\mathbf{x}_t, \mathbf{x}_0) := \sqrt{\overline{\alpha}_{t-1}}\mathbf{x}_0 + \sqrt{1 - \overline{\alpha}_{t-1} - \sigma_t^2} \cdot \frac{\mathbf{x}_t - \sqrt{\overline{\alpha}_t}\mathbf{x}_0}{\sqrt{1 - \overline{\alpha}_t}} \tag{27}$$

This formulation encapsulates DDPM and DDIM as special cases, where DDPM corresponds to setting $\sigma_t^2 = \frac{\hat{\beta}_{t-1}}{\hat{\beta}_t}\beta_t$ and DDIM corresponds to setting $\sigma_t^2 = 0$. DDIM learns a Markov chain to reverse this non-Markov perturbation process, which is fully deterministic when $\sigma_t^2 = 0$. It is observed in [111, 145, 201, 220] that the DDIM sampling process amounts to a special discretization scheme of the probability flow ODE. Inspired by an analysis of DDIM on a singleton dataset, **generalized Denoising Diffusion Implicit Models (gDDIM)** [280] proposes a modified parameterization of the score network that enables deterministic sampling for more general diffusion processes, such as the one in **Critically-Damped Langevin Diffusion (CLD)** [59]. PNDM [142] proposes a pseudo numerical method to generate sample along a specific manifold in $\mathcal{R}^N$. It uses numerical solver with nonlinear transfer part to solve differential equation on manifolds and then generates sample, which encapsulates DDIM as a special case.

Through extensive experimental investigations, Karras et al. (2022) [111] show that Heun's 2nd order method [5] provides an excellent trade off between sample quality and sampling speed. The higher-order solver leads to smaller discretization error at the cost of one additional evaluation of the learned score function per time step. Heun's method generates samples of comparable, if not better quality than Euler's method with fewer sampling steps.

Diffusion Exponential Integrator Sampler [279] and DPM-solver [145] leverage the semi-linear structure of probability flow ODE to develop customized ODE solvers that are more efficient than general-purpose Runge-Kutta methods. Specifically, the linear part of probability flow ODE can be analytically computed, while the non-linear part can be solved with techniques similar to exponential integrators in the field of ODE solvers. These methods contain DDIM as a first-order approximation. However, they also allow for higher order integrators, which can produce high-quality samples in just 10 to 20 iterations—far fewer than the hundreds of iterations typically required by diffusion models without accelerated sampling.

3.2 Learning-Based Sampling

Learning-based sampling is another efficient approach for diffusion models. By using partial steps or training a sampler for the reverse process, this method achieves faster sampling speeds at the expense of slight degradation in sample quality. Unlike learning-free approaches that use handcrafted steps, learning-based sampling typically involves selecting steps by optimizing certain learning objectives.

3.2.1 Optimized Discretization. Given a pre-trained diffusion model, Watson et al. (2021) [247] put forth a strategy for finding the optimal discretization scheme by selecting the best K time steps to maximize the training objective for DDPMs. Key to this approach is the observation that the

DDPM objective can be broken down into a sum of individual terms, making it well suited for dynamic programming. However, it is well known that the variational lower bound used for DDPM training does not correlate directly with sample quality [235]. A subsequent work, called Differentiable Diffusion Sampler Search [246], addresses this issue by directly optimizing a common metric for sample quality called the **Kernel Inception Distance (KID)** [17]. This optimization is feasible with the help of reparameterization [122, 194] and gradient rematerialization. Based on truncated Taylor methods, Dockhorn et al. (2022) [60] derive a second-order solver for accelerating synthesis by training an additional head on top of the first-order score network.

3.2.2 Truncated Diffusion. One can improve sampling speed by truncating the forward and reverse diffusion processes [155, 285]. The key idea is to halt the forward diffusion process early on, after just a few steps, and to begin the reverse denoising process with a non-Gaussian distribution. Samples from this distribution can be obtained efficiently by diffusing samples from pretrained generative models, such as variational autoencoders [122, 194] or generative adversarial networks [71].

3.2.3 Knowledge Distillation. Approaches that use knowledge distillation [147, 201] can significantly improve the sampling speed of diffusion models. Specifically, in Progressive Distillation [201], the authors propose distilling the full sampling process into a faster sampler that requires only half as many steps. By parameterizing the new sampler as a deep neural network, authors are able to train the sampler to match the input and output of the DDIM sampling process. Repeating this procedure can further reduce sampling steps, although fewer steps can result in reduced sample quality. To address this issue, the authors suggest new parameterizations for diffusion models and new weighting schemes for the objective function.

4 DIFFUSION MODELS WITH IMPROVED LIKELIHOOD

As discussed in Section 2.1, the training objective for diffusion models is a (negative) **variational lower bound (VLB)** on the log-likelihood. This bound, however, may not be tight in many cases [120], leading to potentially suboptimal log-likelihoods from diffusion models. In this section, we survey recent works on likelihood maximization for diffusion models. We focus on three types of methods: noise schedule optimization, reverse variance learning, and exact log-likelihood evaluation.

4.1 Noise Schedule Optimization

In the classical formulation of diffusion models, noise schedules in the forward process are handcrafted without trainable parameters. By optimizing the forward noise schedule jointly with other parameters of diffusion models, one can further maximize the VLB in order to achieve higher log-likelihood values [120, 165].

The work of iDDPM [165] demonstrates that a certain cosine noise schedule can improve log-likelihoods. Specifically, the cosine noise schedule in their work takes the form of

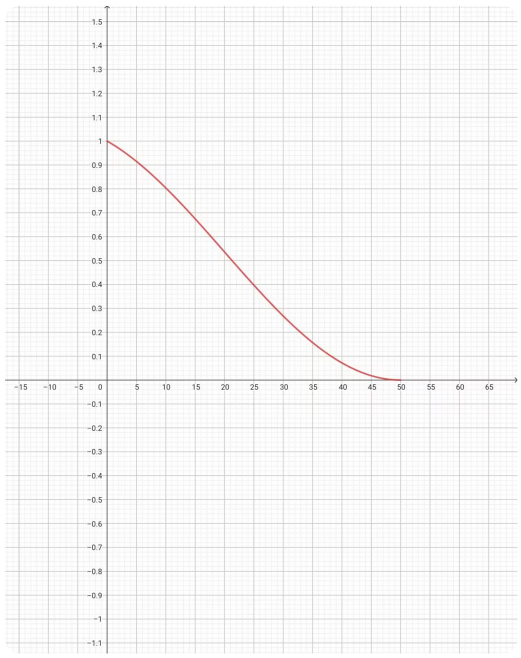
$$\bar{\alpha}_t = \frac{h(t)}{h(0)}, \quad h(t) = \cos\left(\frac{t/T + m}{1 + m} \cdot \frac{\pi}{2}\right)^2 \tag{28}$$

where $\bar{\alpha}_t$ and β_t are defined in Equation (2) and (3), and m is a hyperparameter to control the noise scale at $t = 0$. They also propose a parameterization of the reverse variance with an interpolation between β_t and $1 - \bar{\alpha}_t$ in the log domain.

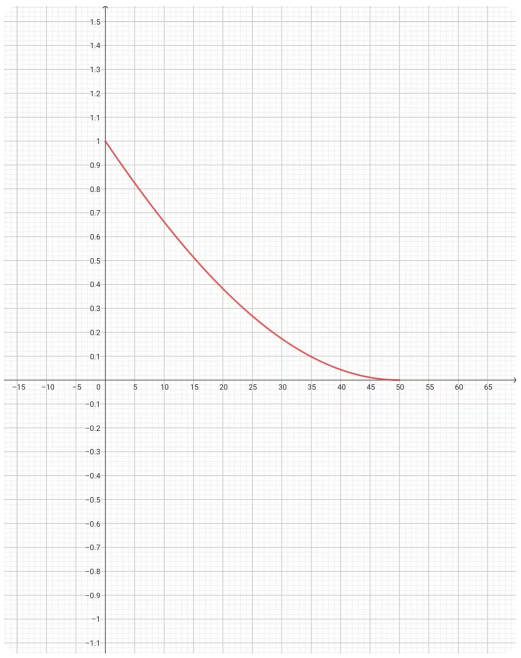
In **Variational Diffusion Models (VDMs)** [120], authors propose to improve the likelihood of continuous-time diffusion models by jointly training the noise schedule and other diffusion model parameters to maximize the VLB. They parameterize the noise schedule using a monotonic

DDPM中，T=50，$\bar{\alpha}_t$ 与s变化趋势

$m=0.2$



$m=2$



neural network $\gamma_\eta(t)$, and build the forward perturbation process according to $\sigma_t^2 = \text{sigmoid}(\gamma_\eta(t))$, $q(\mathbf{x}_t \mid \mathbf{x}_0) = \mathcal{N}(\bar{\alpha}_t \mathbf{x}_0, \sigma_t^2 \mathbf{I})$, and $\bar{\alpha}_t = \sqrt{(1-\sigma_t^2)}$. Moreover, authors prove that the VLB for data point $\mathbf{x}$ can be simplified to a form that only depends on the signal-to-noise ratio $\mathrm{R}(t) := \frac{\bar{\alpha}_t^2}{\sigma_t^2}$. In particular, the L_{VLB} can be decomposed to

$$L_{VLB} = -\mathbb{E}_{\mathbf{x}_0} \mathrm{KL}(q(\mathbf{x}_T|\mathbf{x}_0) \mid\mid p(\mathbf{x}_T)) + \mathbb{E}_{\mathbf{x}_0, \mathbf{x}_1} \log p(\mathbf{x}_0|\mathbf{x}_1) - L_D, \tag{29}$$

where the first and second terms can be optimized directly in analogy to training variational autoencoders. The third term can be further simplified to the following:

$$L_D = \frac{1}{2} \mathbb{E}_{\mathbf{x}_0, \epsilon \sim \mathcal{N}(0,\mathbf{I})} \int_{\mathrm{R}_{\min}}^{\mathrm{R}_{\max}} \|\mathbf{x}_0 - \tilde{\mathbf{x}}_\theta(\mathbf{x}_v, v)\|_2^2 \, dv, \tag{30}$$

where $\mathrm{R}_{\max} = R(1), \mathrm{R}_{\min} = R(T)$, $\mathbf{x}_v = \bar{\alpha}_v \mathbf{x}_0 + \sigma_v \epsilon$ denotes a noisy data point obtained by diffusing $\mathbf{x}_0$ with the forward perturbation process until $t = R^{-1}(v)$, and $\tilde{\mathbf{x}}_\theta$ denotes the predicted noise-free data point by the diffusion model. As a result, noise schedules do not affect the VLB as long as they share the same values at $\mathrm{R}_{\min}$ and $\mathrm{R}_{\max}$, and will only affect the variance of Monte Carlo estimators for VLB.

4.2 Reverse Variance Learning

The classical formulation of diffusion models assumes that Gaussian transition kernels in the reverse Markov chain have fixed variance parameters. Recall that we formulated the reverse kernel as $q_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t) = \mathcal{N}(\mu_\theta(\mathbf{x}_t, t), \Sigma_\theta(\mathbf{x}_t, t))$ in Equation (5) but often fixed the reverse variance $\Sigma_\theta(\mathbf{x}_t, t)$ to $\beta_t \mathbf{I}$. Many methods propose to train the reverse variances as well to further maximize VLB and log-likelihood values.

In iDDPM [165], Nichol and Dhariwal propose to learn the reverse variances by parameterizing them with a form of linear interpolation and training them using a hybrid objective. This results in higher log-likelihoods and faster sampling without losing sample quality. In particular, they parameterize the reverse variance in Equation (5) as:

$$\Sigma_\theta(\mathbf{x}_t, t) = \exp(\theta \cdot \log \beta_t + (1-\theta) \cdot \log \tilde{\beta}_t), \tag{31}$$

where $\tilde{\beta}_t := \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t} \cdot \beta_t$ and θ is jointly trained to maximize VLB. This simple parameterization avoids the instability of estimating more complicated forms of $\Sigma_\theta(\mathbf{x}_t, t)$ and is reported to improve likelihood values.

Analytic-DPM [10] shows a remarkable result that the optimal reverse variance can be obtained from a pre-trained score function, with the analytic form below:

$$\Sigma_\theta(\mathbf{x}_t, t) = \sigma_t^2 + \left(\sqrt{\frac{\overline{\beta}_t}{\alpha_t}} - \sqrt{\overline{\beta}_{t-1} - \sigma_t^2} \right)^2 \cdot \left(1 - \overline{\beta}_t \mathbb{E}_{q_t(\mathbf{x}_t)} \frac{||\nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t)||^2}{d} \right) \tag{32}$$

As a result, given a pre-trained score model, we can estimate its first- and second-order moments to obtain the optimal reverse variances. Plugging them into the VLB can lead to tighter VLBs and higher likelihood values.

4.3 Exact Likelihood Computation

In the Score SDE [228] formulation, samples are generated by solving the following reverse SDE, where $\nabla_{\mathbf{x}_t} \log p_\theta(\mathbf{x}_t, t)$ in Equation (18) is replaced by the learned noise-conditional score model

$\mathbf{s}_\theta(\mathbf{x}_t, t)$:

$$\mathrm{d}\mathbf{x} = f(\mathbf{x}_t, t) - g(t)^2 \mathbf{s}_\theta(\mathbf{x}_t, t)\mathrm{d}t + g(t)\mathrm{d}\mathbf{w}. \tag{33}$$

Here we use p_θ^{sde} to denote the distribution of samples generated by solving the above SDE. One can also generate data by plugging the score model into the probability flow ODE in Equation (19), which gives:

$$\frac{\mathrm{d}\mathbf{x}_t}{\mathrm{d}t} = \underbrace{f(\mathbf{x}_t, t) - \frac{1}{2} g^2(t) \mathbf{s}_\theta(\mathbf{x}_t, t)}_{:=\tilde{f}_\theta(\mathbf{x}_t, t)} \tag{34}$$

Similarly, we use p_θ^{ode} to denote the distribution of samples generated via solving this ODE. The theory of neural ODEs [34] and continuous normalizing flows [74] indicates that p_θ^{ode} can be computed accurately albeit with high computational cost. For p_θ^{sde}, several concurrent works [95, 144, 222] demonstrate that there exists an efficiently computable variational lower bound, and we can directly train our diffusion models to maximize p_θ^{sde} using modified diffusion losses.

Specifically, Song et al. (2021) [222] prove that with a special weighting function (likelihood weighting), the objective used for training score SDEs implicitly maximizes the expected value of p_θ^{sde} on data. It is shown that

$$\mathbf{D}_{KL}(q_0 \parallel p_\theta^{\text{sde}}) \leq \mathcal{L}(\theta; g(\cdot)^2) + \mathbf{D}_{KL}(q_T \parallel \pi), \tag{35}$$

where $\mathcal{L}(\theta; g(\cdot)^2)$ is the Score SDE objective in Equation (20) with $\lambda(t) = g(t)^2$. Since $\mathbf{D}_{KL}(q_0 \parallel p_\theta^{\text{sde}}) = -\mathbb{E}_{q_0} \log(p_\theta^{\text{sde}}) + \text{const}$, and $\mathbf{D}_{KL}(q_T \parallel \pi)$ is a constant, training with $\mathcal{L}(\theta; g(\cdot)^2)$ amounts to minimizing $-\mathbb{E}_{q_0} \log(p_\theta^{\text{sde}})$, the expected negative log-likelihood on data. Moreover, Song et al. (2021) and Huang et al. (2021) [95, 222] provide the following bound for $p_\theta^{\text{sde}}(\mathbf{x})$:

$$-\log p_\theta^{\text{sde}}(\mathbf{x}) \leq \mathcal{L}'(\mathbf{x}), \tag{36}$$

where $\mathcal{L}'(\mathbf{x})$ is defined by

$$\begin{aligned} \mathcal{L}'(\mathbf{x}) := \int_0^T \mathbb{E}\left[\frac{1}{2}||g(t)\mathbf{s}_\theta(\mathbf{x}_t, t)||^2 + \nabla \cdot (g(t)^2 \mathbf{s}_\theta(\mathbf{x}_t, t) - f(\mathbf{x}_t), \mathbf{t}) \,\middle|\, \mathbf{x_0} = \mathbf{x}\right] dt \\ - \mathbb{E}_{\mathbf{x}_T}[\log p_\theta^{\text{sde}}(\mathbf{x}_T) \mid \mathbf{x}_0 = x] \end{aligned} \tag{37}$$

The first part of Equation (37) is reminiscent of implicit score matching [98] and the whole bound can be efficiently estimated with Monte Carlo methods.

Since the probability flow ODE is a special case of neural ODEs or continuous normalizing flows, we can use well-established approaches in those fields to compute $\log p_\theta^{\text{ode}}$ accurately. Specifically, we have

$$\log p_\theta^{\text{ode}}(\mathbf{x}_0) = \log p_T(\mathbf{x}_T) + \int_{t=0}^T \nabla \cdot \tilde{f}_\theta(\mathbf{x}_t, t)\mathrm{d}t. \tag{38}$$

One can compute the one-dimensional integral above with numerical ODE solvers and the Skilling-Hutchinson trace estimator [97, 217]. Unfortunately, this formula cannot be directly optimized to maximize p_θ^{ode} on data, as it requires calling expensive ODE solvers for each data point $\mathbf{x}_0$. To reduce the cost of directly maximizing p_θ^{ode} with the above formula, Song et al. (2021) [222] propose to maximize the variational lower bound of p_θ^{sde} as a proxy for maximizing p_θ^{ode}, giving rise to a family of diffusion models called *ScoreFlows*.

Lu et al. (2022) [144] further improve ScoreFlows by proposing to minimize not just the vanilla score matching loss function, but also its higher order generalizations. They prove that $\log p_\theta^{\text{ode}}$

can be bounded with the first, second, and third-order score matching errors. Building upon this theoretical result, authors further propose efficient training algorithms for minimizing high order score matching losses and reported improved p_θ^{ode} on data.

5 DIFFUSION MODELS FOR DATA WITH SPECIAL STRUCTURES

While diffusion models have achieved great success for data domains like images and audio, they do not necessarily translate seamlessly to other modalities. Many important data domains have special structures that must be taken into account for diffusion models to function effectively. Difficulties may arise, for example, when models rely on score functions that are only defined on continuous data domains, or when data reside on low dimensional manifolds. To cope with these challenges, diffusion models have to be adapted in various ways.

5.1 Discrete Data

Most diffusion models are geared towards continuous data domains, because Gaussian noise perturbation as used in DDPMs is not a natural fit for discrete data, and the score functions required by SGMs and Score SDEs are only defined on continuous data domains. To overcome this difficulty, several works [6, 78, 93, 257] build on Sohl-Dickstein et al. (2015) [218] to generate discrete data of high dimensions. Specifically, VQ-Diffusion [78] replaces Gaussian noise with a random walk on the discrete data space, or a random masking operation. The resulting transition kernel for the forward process takes the form of

$$q(\mathbf{x}_t \mid \mathbf{x}_{t-1}) = \mathbf{v}^\top(\mathbf{x}_t)\mathbf{Q}_t\mathbf{v}(\mathbf{x}_{t-1}) \tag{39}$$

where $\mathbf{v}(\mathbf{x})$ is a one-hot column vector, and $\mathbf{Q}_t$ is the transition kernel of a lazy random walk. D3PM [6] accommodates discrete data in diffusion models by constructing the forward noising process with absorbing state kernels or discretized Gaussian kernels. Campbell et al. (2022) [24] present the first continuous-time framework for discrete diffusion models. Leveraging Continuous Time Markov Chains, they are able to derive efficient samplers that outperform discrete counterparts, while providing a theoretical analysis on the error between the sample distribution and the true data distribution.

5.2 Data with Invariant Structures

Data in many important domains have invariant structures. For example, graphs are permutation invariant, and point clouds are both translation and rotation invariant. In diffusion models, these invariances are often ignored, which can lead to suboptimal performance. To address this issue, several works [51, 169] propose to endow diffusion models with the ability to account for invariance in data.

Niu et al. (2020) [169] first tackle the problem of permutation invariant graph generation with diffusion models. They achieve this by using a permutation equivariant graph neural network [72, 206, 254], called EDP-GNN, to parameterize the noise-conditional score model. GDSS [105] further develops this idea by proposing a continuous-time graph diffusion process. This process models both the joint distribution of nodes and edges through a system of stochastic differential equations (SDEs), where message-passing operations are used to guarantee permutation invariance.

Similarly, Shi et al. (2021) [212] and Xu et al. (2021) [261] enable diffusion models to generate molecular conformations that are invariant to both translation and rotation. For example, Xu et al. (2022) [261] shows that Markov chains starting with an invariant prior and evolving with equivariant Markov kernels can induce an invariant marginal distribution, which can be used to enforce appropriate data invariance in molecular conformation generation. Formally, let $\mathcal{T}$ be a rotation or translation operation. Given that $p(\mathbf{x}_T) = p(\mathcal{T}(\mathbf{x}_T))$, $p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t) = p_\theta(\mathcal{T}(\mathbf{x}_{t-1}) \mid \mathcal{T}(\mathbf{x}_t))$, Xu

et al. (2022) [261] prove that the distribution of samples is guaranteed to be invariant to $\mathcal{T}$, that is, $p_0(\mathbf{x}) = p_0(\mathcal{T}(\mathbf{x}))$. As a result, one can build a diffusion model that generates rotation and translation invariant molecular conformations as long as the prior and transition kernels enjoy the same invariance.

5.3 Data with Manifold Structures

Data with manifold structures are ubiquitous in machine learning. As the manifold hypothesis [66] posits, natural data often reside on manifolds with lower intrinsic dimensionality. In addition, many data domains have well-known manifold structures. For instance, climate and earth data naturally lie on the sphere because that is the shape of our planet. Many works have focused on developing diffusion models for data on manifolds. We categorize them based on whether the manifolds are known or learned, and introduce some representative works below.

5.3.1 Known Manifolds. Recent studies have extended the Score SDE formulation to various known manifolds. This adaptation parallels the generalization of neural ODEs [34] and continuous normalizing flows [74] to Riemannian manifolds [143, 157]. To train these models, researchers have also adapted score matching and score functions to Riemannian manifolds.

The **Riemannian Score-Based Generative Model (RSGM)** [51] accommodates a wide range of manifolds, including spheres and toruses, provided they satisfy mild conditions. The RSGM demonstrates that it is possible to extend diffusion models to compact Riemannian manifolds. The model also provides a formula for reversing diffusion on a manifold. Taking an intrinsic view, the RSGM approximates the sampling process on Riemannian manifolds using a Geodesic Random Walk. It is trained with a generalized denoising score matching objective.

In contrast, the **Riemannian Diffusion Model (RDM)** [94] employs a variational framework to generalize the continuous-time diffusion model to Riemannian manifolds. The RDM uses a **variational lower bound (VLB)** of the log-likelihood as its loss function. The authors of the RDM model have shown that maximizing this VLB is equivalent to minimizing a Riemannian score-matching loss. Unlike the RSGM, the RDM takes an extrinsic view, assuming that the relevant Riemannian manifold is embedded in a higher dimensional Euclidean space.

5.3.2 Learned Manifolds. According to the manifold hypothesis [66], most natural data lies on manifolds with significantly reduced intrinsic dimensionality. Consequently, identifying these manifolds and training diffusion models directly on them can be advantageous due to the lower data dimensionality. Many recent works have built on this idea, starting by using an autoencoder to condense the data into a lower dimensional manifold, followed by training diffusion models in this latent space. In these cases, the manifold is implicitly defined by the autoencoder and learned through the reconstruction loss. In order to be successful, it is crucial to design a loss function that allows for the joint training of the autoencoder and the diffusion models.

The **Latent Score-Based Generative Model (LSGM)** [238] seeks to address the problem of joint training by pairing a Score SDE diffusion model with a **variational autoencoder (VAE)** [122, 194]. In this configuration, the diffusion model is responsible for learning the prior distribution. The authors of the LSGM propose a joint training objective that merges the VAE's evidence lower bound with the diffusion model's score matching objective. This results in a new lower bound for the data log-likelihood. By situating the diffusion model within the latent space, the LSGM achieves faster sample generation than conventional diffusion models. Additionally, the LSGM can manage discrete data by converting it into continuous latent codes.

Rather than jointly training the autoencoder and diffusion model, the **Latent Diffusion Model (LDM)** [196] addresses each component separately. First, an autoencoder is trained to produce a low-dimensional latent space. Then, the diffusion model is trained to generate latent codes.

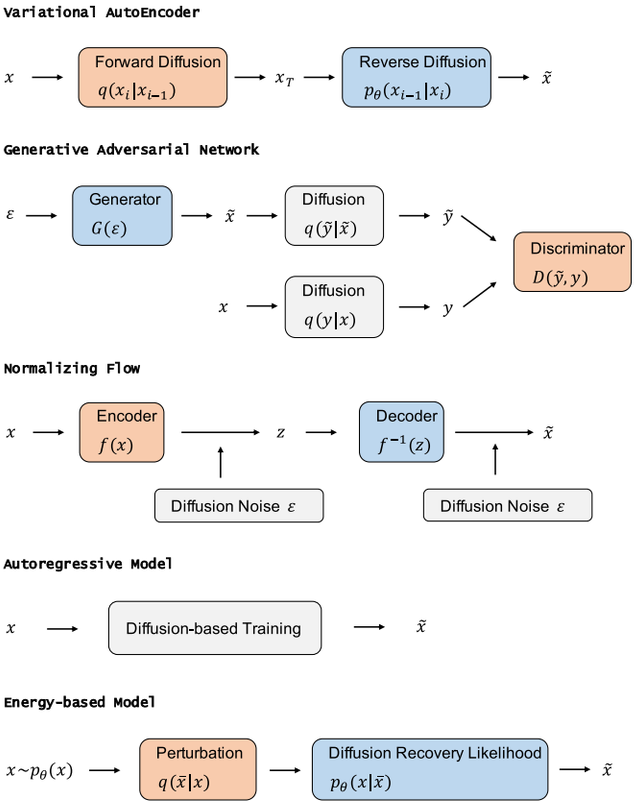


Fig. 3. Illustrations of works incorporating diffusion models with other generative models, such as : VAE [196] where a diffusion model is applied on a latent space, GAN [245] where noise is injected to the discriminator input, normalizing flow [278] where noise is injected in both forward and backward processes in the flow, autoregressive model [92] where the training objective is similar to diffusion models, and EBM [69] where a sequence of EBMs is learned by diffusion recovery likelihood.

DALLE-2 [186] employs a similar strategy by training a diffusion model on the CLIP image embedding space, followed by training a separate decoder to create images based on the CLIP image embeddings.

6 CONNECTIONS WITH OTHER GENERATIVE MODELS

In this section, we first introduce five other important classes of generative models and analyze their advantages and limitations. Then we introduce how diffusion models are connected with them, and illustrate how these generative models improve by incorporating diffusion models. We provide a schematic illustration in Figure 3, and quantitative (with **Frechet Inception Distance**, denoted as **FID**) and qualitative comparisons in Figure 4.

6.1 Variational Autoencoders and Connections with Diffusion Models

Variational Autoencoders [61, 123, 194] aim to learn both an encoder and a decoder to map input data to values in a continuous latent space. In these models, the embedding can be interpreted as a latent variable in a probabilistic generative model, and a probabilistic decoder can be formulated by a parameterized likelihood function. In addition, the data $\mathbf{x}$ is assumed to be generated by some unobserved latent variable $\mathbf{z}$ using conditional distribution $p_\theta(\mathbf{x} \mid \mathbf{z})$, and $q_\phi(\mathbf{z} \mid \mathbf{x})$ is used to approximately inference $\mathbf{z}$. To guarantee an effective inference, a variational Bayes approach is used to maximize the evidence lower bound:

$$\mathcal{L}(\phi, \theta; \mathbf{x}) = \mathbb{E}_{q(\mathbf{z}|\mathbf{x})} \left[\log p_\theta(\mathbf{x}, \mathbf{z}) - \log q_\phi(\mathbf{z} \mid \mathbf{x}) \right] \tag{40}$$

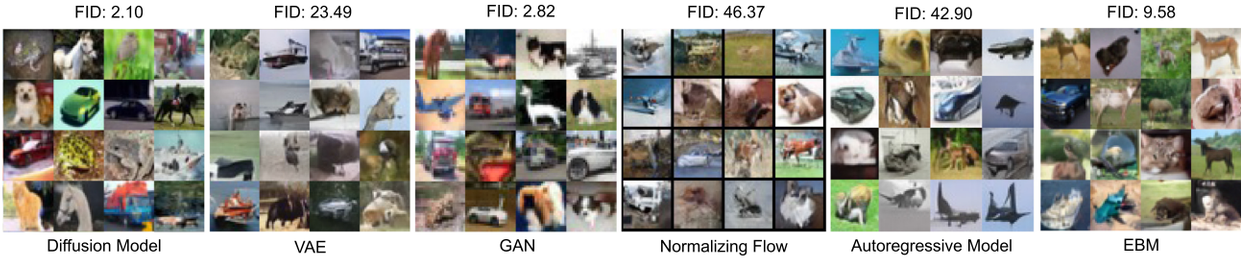


Fig. 4. Quantitative and qualitative comparison between diffusion models and other generative models on CIFAR10 dataset.

with $\mathcal{L}(\phi, \theta; \mathbf{x}) \leq \log p_\theta(\mathbf{x})$. Provided that the parameterized likelihood function $p_\theta(\mathbf{x} \mid \mathbf{z})$ and the parameterized posterior approximation $q_\phi(\mathbf{z} \mid \mathbf{x})$ can be computed in a point-wise way and are differentiable with their parameters, the ELBO can be maximized with gradient descent. This formulation allows flexible choice of encoder and decoder models. Typically, these models are represented by exponential family distributions whose parameters are generated by multi-layer neural networks.

The DDPM can be conceptualized as a hierarchical Markovian VAE with a fixed encoder. Specifically, DDPM's forward process functions as the encoder, and this process is structured as a linear Gaussian model (as described by Equation (2)). The DDPM's reverse process, on the other hand, corresponds to the decoder, which is shared across multiple decoding steps. The latent variables within the decoder are all the same size as the sample data.

In a continuous-time setting, Song et al. (2020) [228], Huang et al. (2021) [95], and Kingma et al. (2021) [120] demonstrate that the score matching objective may be approximated by the **Evidence Lower Bound (ELBO)** of a deep hierarchical VAE. Consequently, optimizing a diffusion model can be seen as training an infinitely deep hierarchical VAE—a finding that supports the common belief that Score SDE diffusion models can be interpreted as the continuous limit of hierarchical VAEs.

The Latent Score-Based Generative Model (LSGM) [238] furthers this line of research by illustrating that the ELBO can be considered a specialized score matching objective in the context of latent space diffusion. Though the cross-entropy term in the ELBO is intractable, it can be transformed into a tractable score matching objective by viewing the score-based generative model as an infinitely deep VAE.

6.2 Generative Adversarial Networks and Connections with Diffusion Models

Generative Adversarial Networks (GANs) [45, 71, 79] mainly consist of two models: a generator G and a discriminator D. These two models are typically constructed by neural networks but could be implemented in any form of a differentiable system that maps input data from one space to another. The optimization of GANs can be viewed as a minimax optimization problem with value function $V(G, D)$:

$$\min_G \max_D \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}(\mathbf{x})}[\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{z} \sim p_z(\mathbf{z})}[\log(1 - D(G(\mathbf{z})))]. \tag{41}$$

The generator G aims to generate new examples and implicitly model the data distribution. The discriminator D is usually a binary classifier that is used to identify generated examples from true examples with maximally possible accuracy. The optimization process ends at a saddle point that produces a minimum about the generator and a maximum about the discriminator. Namely, the goal of GAN optimization is to achieve Nash equilibrium [192]. At that point, the generator can be considered that it has captured the accurate distribution of real examples.

One of the issues of GAN is the instability in the training process, which is mainly caused by the non-overlapping between the distribution of input data and that of the generated data.

One solution is to inject noise into the discriminator input for widening the support of both the generator and discriminator distributions. Taking advantage of the flexible diffusion model, Wang et al. (2022) [245] inject noise to the discriminator with an adaptive noise schedule determined by a diffusion model. On the other hand, GAN can facilitate sampling speed of diffusion models. Xiao et al. (2021) [256] show that slow sampling is caused by the Gaussian assumption in the denoising step, which is justified only for small step sizes. As such, each denoising step is modeled by a conditional GAN, allowing larger step size.

6.3 Normalizing Flows and Connections with Diffusion Models

Normalizing flows [56, 193] are generative models that generate tractable distributions to model high-dimensional data [57, 121]. Normalizing flows can transform simple probability distribution into an extremely complex probability distribution, which can be used in generative models, reinforcement learning, variational inference, and other fields. Existing normalizing flows are constructed based on the change of variable formula [56, 193]. The trajectory in normalizing flows is formulated by a differential equation. In the discrete-time setting, the mapping from data $\mathbf{x}$ to latent $\mathbf{z}$ in normalizing flows is a composition of a sequence of bijections, taking the form of $F = F_N \circ F_{N-1} \circ \ldots \circ F_1$. The Trajectory $\{\mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_N\}$ in normalizing flows satisfies :

$$\mathbf{x}_i = F_i(\mathbf{x}_{i-1}, \theta),\ \mathbf{x}_{i-1} = F_i^{-1}(\mathbf{x}_i, \theta) \tag{42}$$

for all $i \leq N$.

Similar to the continuous setting, normalizing flows allow for the retrieval of the exact log-likelihood through a change of variable formula. However, the bijection requirement limits the modeling of complex data in both practical and theoretical contexts [44, 249]. Several works attempt to relax this bijection requirement [57, 249]. For example, DiffFlow [278] introduces a generative modeling algorithm that combines the benefits of both flow-based and diffusion models. As a result, DiffFlow produces sharper boundaries than normalizing flow and learns more general distributions with fewer discretization steps compared to diffusion probabilistic models.

6.4 Autoregressive Models and Connections with Diffusion Models

Autoregressive Models (ARMs) work by decomposing the joint distribution of data into a product of conditional distributions using the probability chain rule:

$$\log p(\mathbf{x}_{1:T}) = \sum_{t=1}^{T} \log p(x_t \mid \mathbf{x}_{<t}) \tag{43}$$

where $\mathbf{x}_{<t}$ is a shorthand for $x_1, x_2, \ldots, x_{t-1}$ [13, 127]. Recent advances in deep learning have facilitated significant progress for various data modalities [29, 161, 205], such as images [38, 239], audio [110, 170], and text [14, 20, 75, 159, 162]. Autoregressive models (ARMs) offer generative capabilities through the use of a single neural network. Sampling from these models requires the same number of network calls as the data's dimensionality. While ARMs are effective density estimators, sampling is a continuous, time-consuming process—particularly for high-dimensional data.

The **Autoregressive Diffusion Model (ARDM)** [92], on the other hand, is capable of generating arbitrary-order data, including order-agnostic autoregressive models and discrete diffusion models as special cases [6, 93, 219]. Instead of using causal masking on representations like ARMs, the ARDM is trained with an effective objective that mirrors that of diffusion probabilistic models. At the testing stage, the ARDM is able to generate data in parallel—enabling its application to a range of arbitrary-generation tasks.

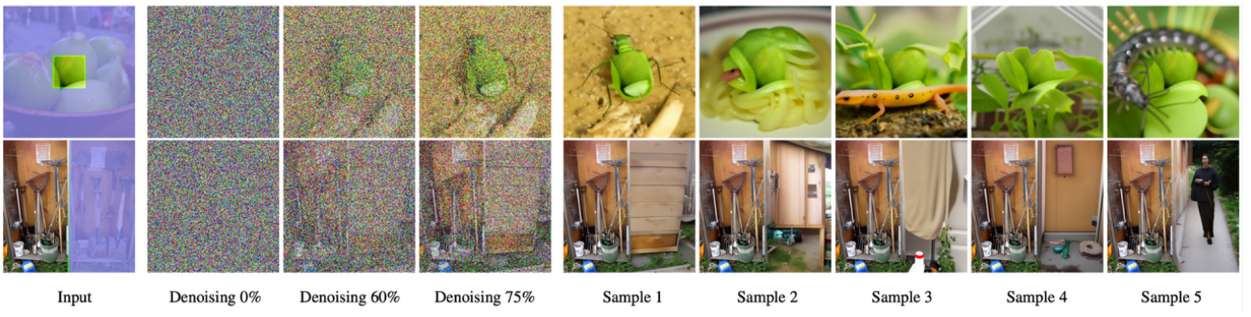


Fig. 5. Image inpainting results produced by RePaint [146].

6.5 Energy-based Models and Connections with Diffusion Models

Energy-based Models (EBMs) [119, 129, 226] can be viewed as one generative version of discriminators [101, 128, 131], while can be learned from unlabeled input data. Let $\mathbf{x} \sim p_{\text{data}}(\mathbf{x})$ denote a training example, and $p_\theta(\mathbf{x})$ denote a probability density function that aims to approximates $p_{\text{data}}(\mathbf{x})$. An energy-based model is defined as $p_\theta(\mathbf{x}) = \frac{1}{Z_\theta} \exp(f_\theta(\mathbf{x}))$, where $Z_\theta = \int \exp(f_\theta(\mathbf{x}))d\mathbf{x}$ is the partition function, which is analytically intractable for high-dimensional $\mathbf{x}$. For images, $f_\theta(\mathrm{x})$ is parameterized by a convolutional neural network with a scalar output. Salimans and Ho (2021) [202] compare both constrained score models and energy-based models for modeling the score of the data distribution, finding that constrained score models, i.e., energy based models, can perform just as well as unconstrained models when using a comparable model structure.

Although EBMs have a number of desirable properties, two challenges remain for modeling high-dimensional data. First, learning EBMs by maximizing the likelihood requires MCMC method to generate samples from the model, which can be very computationally expensive. Second, as demonstrated in [168], the energy potentials learned with non-convergent MCMC are not stable, in the sense that samples from long-run Markov chains can be significantly different from the observed samples, and thus it is difficult to evaluate the learned energy potentials. In a recent study, Gao et al. (2020) [69] present a diffusion recovery likelihood method to tractably learn samples from a sequence of EBMs in the reverse process of the diffusion model. Each EBM is trained with recovery likelihood, which aims to maximize the conditional probability of the data at a certain noise level, given their noisy versions at a higher noise level. EBMs maximize the recovery likelihood because it is more tractable than marginal likelihood, as sampling from the conditional distributions is much easier than sampling from the marginal distributions. This model can generate high-quality samples, and long-run MCMC samples from the conditional distributions still resemble realistic images.

7 APPLICATIONS OF DIFFUSION MODELS

Diffusion models have recently been employed to address a variety of challenging real-world tasks due to their flexibility and strength. We have grouped these applications into six different categories based on the task: computer vision, natural language processing, temporal data modeling, multi-modal learning, robust learning, and interdisciplinary applications. For each category, we provide a brief introduction to the task, followed by a detailed explanation of how diffusion models have been applied to improve performance.

7.1 Computer Vision

7.1.1 Super Resolution, Inpainting, and Translation. Generative models have been used to tackle a variety of image restoration tasks including super-resolution, inpainting, and translation [12, 53, 65, 100, 134, 173, 187, 283]. Image super-resolution aims to restore high-resolution images from

low-resolution inputs, while image inpainting revolves around reconstructing missing or damaged regions in an image.

Several methods make use of diffusion models for these tasks. For example, **Super-Resolution via Repeated Refinement (SR3)** [200] uses DDPM to enable conditional image generation. SR3 conducts super-resolution through a stochastic, iterative denoising process. The **Cascaded Diffusion Model (CDM)** [88] consists of multiple diffusion models in sequence, each generating images of increasing resolution. Both the SR3 and CDM directly apply the diffusion process to input images, which leads to larger evaluation steps.

In order to allow for the training of diffusion models with limited computational resources, some methods [196, 238] have shifted the diffusion process to the latent space using pre-trained autoencoders. The **Latent Diffusion Model (LDM)** [196] streamlines the training and sampling processes for denoising diffusion models without sacrificing quality.

For inpainting tasks, RePaint [146] features an enhanced denoising strategy that uses resampling iterations to better condition the image (see Figure 5). Meanwhile, Palette [198] employs conditional diffusion models to create a unified framework for four image generation tasks: colorization, inpainting, uncropping, and JPEG restoration.

Image translation focuses on synthesizing images with specific desired styles [100]. SDEdit [160] uses a **Stochastic Differential Equation (SDE)** prior to improve fidelity. Specifically, it begins by adding noise to the input image, then denoises the image through the SDE.

7.1.2 Semantic Segmentation. Semantic segmentation aims to label each image pixel according to established object categories. Generative pre-training can enhance the label utilization of semantic segmentation models, and recent work has shown that representations learned through DDPM contain high-level semantic information that is useful for segmentation tasks [11]. The few-shot method that leverages these learned representations has outperformed alternatives such as VDVAE [37] and ALAE [178]. Similarly, **Decoder Denoising Pretraining (DDeP)** [19] integrates diffusion models with denoising autoencoders [241] and delivers promising results on label-efficient semantic segmentation.

7.1.3 Video Generation. Generating high-quality videos remains a challenge in the deep learning era due to the complexity and spatio-temporal continuity of video frames [267, 276]. Recent research has turned to diffusion models to improve the quality of generated videos [90]. For example, the **Flexible Diffusion Model (FDM)** [83] uses a generative model to allow for the sampling of any arbitrary subset of video frames, given any other subset. The FDM also includes a specialized architecture designed for this purpose. Additionally, the **Residual Video Diffusion (RVD)** model [271] utilizes an autoregressive, end-to-end optimized video diffusion model. It generates future frames by amending a deterministic next-frame prediction, using a stochastic residual produced through an inverse diffusion process.

7.1.4 Point Cloud Completion and Generation. Point clouds are a critical form of 3D representation for capturing real-world objects. However, scans often generate incomplete point clouds due to partial observation or self-occlusion. Recent studies have applied diffusion models to address this challenge, using them to infer missing parts in order to reconstruct complete shapes. This work has implications for many downstream tasks such as 3D reconstruction, augmented reality, and scene understanding [150, 154].

Luo and Hu 2021 [149] have taken the approach of treating point clouds as particles in a thermodynamic system, using a heat bath to facilitate diffusion from the original distribution to a noise distribution (see Figure 6). Meanwhile, the **Point-Voxel Diffusion (PVD)** model [287] joins denoising diffusion models with the pointvoxel representation of 3D shapes. The **Point**

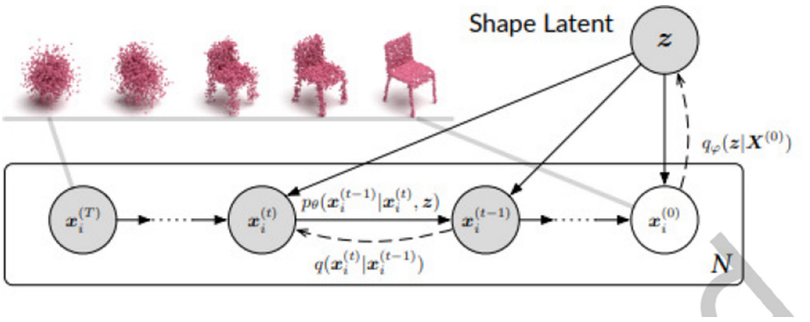


Fig. 6. The directed graphical model of the diffusion process for point clouds [149].

Diffusion-Refinement (PDR) model [154] uses a conditional DDPM to generate a coarse completion from partial observations; it also establishes a point-wise mapping between the generated point cloud and the ground truth.

7.1.5 Anomaly Detection. Anomaly detection is a critical and challenging problem in machine learning [207, 284] and computer vision [264]. Generative models have been shown to own a powerful mechanism for anomaly detection [8, 82, 255], modeling normal or healthy reference data. AnoDDPM [255] utilizes DDPM to corrupt the input image and reconstruct a healthy approximation of the image. These approaches may perform better than alternatives based on adversarial training as they can better model smaller datasets with effective sampling and stable training schemes. DDPM-CD [8] incorporates large numbers of unsupervised remote sensing images into the training process through DDPM. Changes of remote sensed images are detected by utilizing a pre-trained DDPM and applying the multi-scale representations from the diffusion model decoder.

7.1.6 Distilling Data from Diffusion Models. Distilling data from generative models can effectively advance various classification tasks [9, 243]. Recent works have begun to utilize diffusion models to achieve this goal for vision tasks [32, 84, 211]. For example, Trabucco et al. [236] adopt diffusion models to make effective data augmentation for few-shot image classification. He et al. [84] use text-to-image diffusion models to synthesize data for image recognition. In addition, Shao et al. [211] conduct diffusion-based feature augmentation for multiple instance learning in the high-resolution wide-field-of-view images.

7.2 Natural Language Processing

Natural language processing aims to understand, model, and manage human languages from different sources such as text or audio. Text generation has become one of the most critical and challenging tasks in natural language processing [99, 136, 137]. It aims to compose plausible and readable text in the human language given input data (e.g., a sequence and keywords) or random noise. Numerous approaches based on diffusion models have been developed for text generation. **Discrete Denoising Diffusion Probabilistic Models (D3PM)** [6] introduces diffusion-like generative models for character-level text generation [31]. It generalizes the multinomial diffusion model [93] through going beyond corruption processes with uniform transition probabilities. Large autoregressive **language models (LMs)** are able to generate high-quality text [20, 40, 184, 281]. To reliably deploy these LMs in real-world applications, the text generation process is usually expected to be controllable. It means we need to generate text that can satisfy desired requirements (e.g., topic, syntactic structure). Controlling the behavior of language models without re-training is a major and important problem in text generation [49, 114]. Although recent methods have achieved significant successes on controlling simple sentence attributes (e.g., sentiment) [125, 265], there is little progress on complex, fine-grained controls (e.g., syntactic structure). In order to tackle more

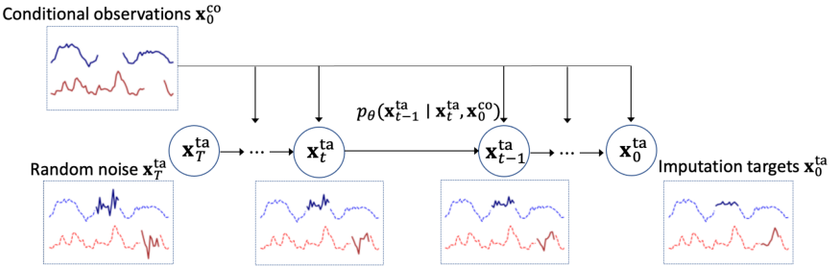


Fig. 7. The procedure of time series imputation with CSDI [233].

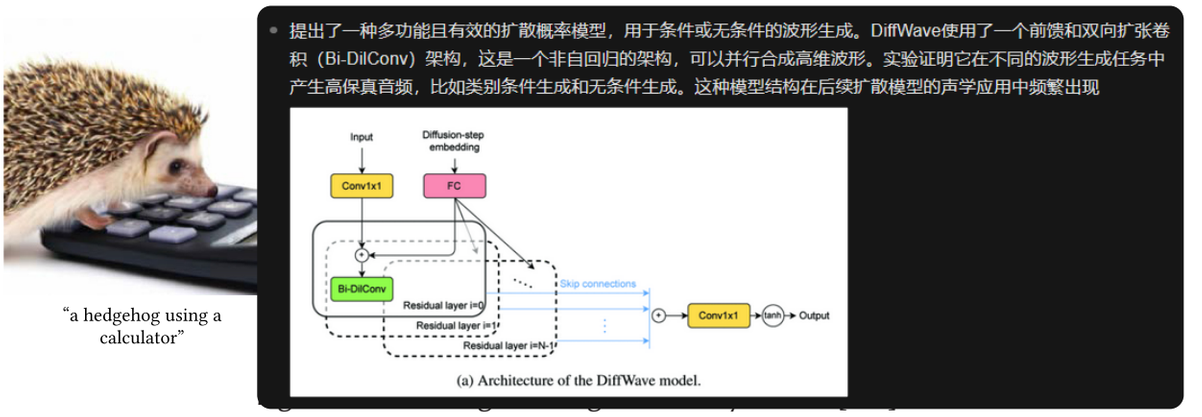
complex controls, Diffusion-LM [139] proposes a new language model based on continuous diffusion. Diffusion-LM starts with a sequence of Gaussian noise vectors and incrementally denoises them into vectors corresponding to words. The gradual denoising steps help produce hierarchical continuous latent representations. This hierarchical and continuous latent variable can make it possible for simple, gradient-based methods to accomplish complex control. Analog Bits [36] generates the analog bits to represent the discrete variables and further improves the sample quality with self-conditioning and asymmetric time intervals.

7.3 Temporal Data Modeling

7.3.1 Time Series Imputation. Time series data are widely used with many important real-world applications [64, 172, 267, 282]. Nevertheless, time series usually contain missing values for multiple reasons, caused by mechanical or artificial errors [215, 232, 272]. Recent years, imputation methods have been greatly developed for both deterministic imputation [26, 30, 153] and probabilistic imputation [67], including diffusion-based approaches. **Conditional Score-based Diffusion models for Imputation (CSDI)** [233] presents a novel time series imputation method that leverages score-based diffusion models (see Figure 7). Specifically, for the purpose of exploiting correlations within temporal data, it adopts the form of self-supervised training to optimize diffusion models. Its application in some real-world datasets reveals its superiority over previous methods. **Controlled Stochastic Differential Equation (CSDE)** [176] proposes a novel probabilistic framework for modeling stochastic dynamics with a neural-controlled stochastic differential equation. **Structured State Space Diffusion (SSSD)** [1] integrates conditional diffusion models and structured state-space models [77] to particularly capture long-term dependencies in time series. It performs well in both time series imputation and forecasting tasks.

7.3.2 Time Series Forecasting. Time series forecasting is the task of forecasting or predicting the future value over a period of time. Neural methods have recently become widely-used for solving the prediction problem with univariate point forecasting methods [171] or univariate probabilistic methods [203]. In the multivariate setting, we also have point forecasting methods [138] as well as probabilistic methods, which explicitly model the data distribution using Gaussian copulas [204], GANs [274], or normalizing flows [191]. TimeGrad [190] presents an autoregressive model for forecasting multivariate probabilistic time series, which samples from the data distribution at each time step through estimating its gradient. It utilizes diffusion probabilistic models, which are closely connected with score matching and energy-based methods. Specifically, it learns gradients by optimizing a variational bound on the data likelihood and transforms white noise into a sample of the distribution of interest through a Markov chain using Langevin sampling [223] during inference time.

7.3.3 Waveform Signal Processing. In electronics, acoustics, and some related fields, the waveform of a signal is denoted by the shape of its graph as a function of time, independent of its time



and magnitude scales. WaveGrad [33] introduces a conditional model for waveform generation that estimates gradients of the data density. It receives a Gaussian white noise signal as input and iteratively refines the signal with a gradient-based sampler. WaveGrad naturally trades inference speed for sample quality by adjusting the number of refinement steps, and makes a connection between non-autoregressive and autoregressive models with respect to audio quality. DiffWave [124] presents a versatile and effective diffusion probabilistic model for conditional or unconditional waveform generation. The model is non-autoregressive and is efficiently trained by optimizing a variant of variational bound on the data likelihood. Moreover, it produces high-fidelity audio in different waveform generation tasks, such as class-conditional generation and unconditional generation.

7.4 Multi-Modal Learning

7.4.1 Text-to-Image Generation. Vision-language models have attracted a lot of attention recently due to the number of potential applications [183]. Text-to-Image generation is the task of generating a corresponding image from a descriptive text [62]. Blended diffusion [7] utilizes both pre-trained DDPM [54] and CLIP [183] models, and it proposes a solution for region-based image editing for general purposes, which uses natural language guidance and is applicable to real and diverse images. DiffusionCLIP [117] carries out CLIP-guided text-driven image manipulation based on full inversion capability and high-quality image generation power of recent diffusion models. It finetunes the score function in the reverse diffusion process using a CLIP loss that controls the attributes of the generated image based on the text prompts. On the other hand, unCLIP (DALLE-2) [186] proposes a two-stage approach, a prior model that can generate a CLIP-based image embedding conditioned on a text caption, and a diffusion-based decoder that can generate an image conditioned on the image embedding. Recently, Imagen [199] proposes a text-to-image diffusion model and a comprehensive benchmark for performance evaluation. It shows that Imagen performs well against the state-of-the-art approaches including VQ-GAN+CLIP [46], Latent Diffusion Models [145], and DALL-E 2 [186]. Models based on classifier guidance [54] use the gradients of an extra classifier to improve the sampling quality of a diffusion model, whereas schemes based on classifier-free guidance [89] mix the score estimates of a conditional diffusion model and a jointly trained unconditional diffusion model. Inspired by the ability of these guided diffusion models [54, 89] to generate photorealistic samples and the ability of text-to-image models to handle free-form prompts, GLIDE [166] applies guided diffusion to the application of text-conditioned image synthesis as demonstrated in Figure 8. VQ-Diffusion [78] proposes a vector-quantized diffusion model for text-to-image generation, and it eliminates the unidirectional bias and avoids accumulative prediction errors.

Another interesting line of research is to leverage the pre-trained text-to-image diffusion model for complex or fine-grained control of synthesis results. DreamBooth [197] tackles the

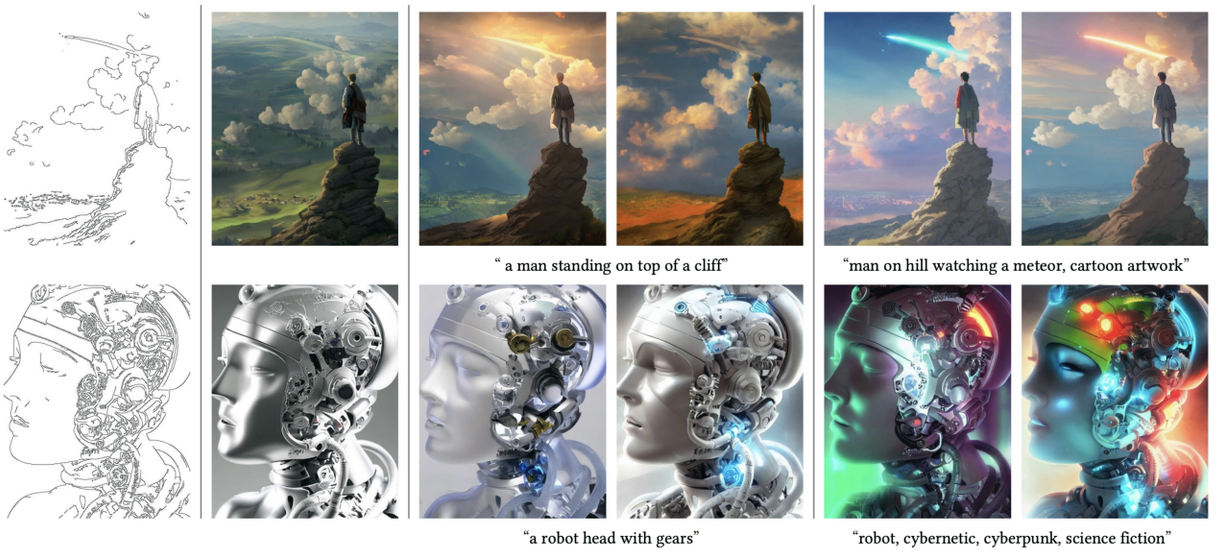


Fig. 9. ControlNet [277] controls stable diffusion with canny edges . The "automatic prompts" are generated by models (BLIP [135]) based on the default result images.

challenging problem of subject-driven generation to contextualize subjects and modify their properties based on a few images provided by users. It learns to associate a unique identifier with the input-specific subject by combining a pre-trained semantic prior with a class-specific prior preservation loss. Different from those image diffusion models conditioned on text prompts, ControlNet [277] adapts pre-trained large diffusion models to support additional semantic maps, like edge maps, segmentation maps, key points, shape normals, and depth cues. ControlNet utilizes a trainable copy of the original weights of the pre-trained diffusion model to avoid overfitting. The trainable copy and the original frozen model are connected with a special convolution layer, where the weights are initialized as zeros and no noise is added in the learning process. The generation results of ControlNet are demonstrated in Figure 9.

7.4.2 Image Generation Based on Scene Graphs. Despite text-to-image generation models have made significant progress, they struggle to faithfully reproduce complex sentences with many objects and relationships. Generating images from **scene graphs (SGs)** is an important and challenging task for generative models [106]. Existing methods [85, 106, 140] mainly predict an image-like layout from SGs, and then generate images based on the layout. However, such intermediate representations would lose some semantics in SGs. On the other hand, recent diffusion models [196] are not able to address this problem well. SGDiff [268] proposes the first diffusion model specifically for image generation from scene graphs and learns a continuous SG embedding to condition the latent diffusion model, which has been globally and locally semantically aligned between SGs and images by the designed masked contrastive pre-training. SGDiff can generate images that express complex relations in SGs better than both non-diffusion and diffusion methods. However, high-quality paired SG-image datasets are scarce. How to leverage large-scale text-image datasets to augment the training or provide a semantic diffusion prior to better initialization is still an open problem.

7.4.3 Text-to-Video Generation. Recent advances in text-to-image diffusion-based generation motivate the development of numerous text-to-video generation models [86, 216, 250]. Make-a-Video [216] extends a diffusion-based text-to-image model to text-to-video through a spatiotemporally factorized diffusion model. It leverages joint text-image prior to alleviate the need for paired text-video data and presents super-resolution strategies for high-definition, high-frame-rate

text-to-video generation. Imagen Video [86] generates high-definition videos by designing cascaded video diffusion models and transferring some findings that perform well in the text-to-image setting to video generation, including frozen T5 text encoder [185] and classifier-free guidance. Tune-a-Video [250] employs DDIM inversion [220] to provide structural guidance for sampling and proposes efficient attention tuning for improving temporal consistency. Most recently, FateZero [181] proposes temporal-consistent zero-shot text-to-video editing using a pre-trained text-to-image diffusion model. It fuses the attention maps in the DDIM inversion and generation processes to preserve the consistency of motion and structure during editing maximally.

7.4.4 Text-to-3D Generation. 3D content generation has been in high demand for a wide range of applications, including gaming, entertainment, and robotics simulation [141, 179]. Augmenting 3D content generation with natural language could considerably help with both novices and experienced artists. DreamFusion [179] adopts a pre-trained 2D text-to-image diffusion model to perform text-to-3D synthesis. It optimizes a randomly-initialized 3D model (a **Neural Radiance Field**, or **NeRF**) with a probability density distillation loss, which utilizes a 2D diffusion model as a prior for optimization of a parametric image generator. Latent-NeRF [164] brings the NeRF to the latent space, and guides text-to-3D sampling process with an abstract geometry that defines the coarse structure of the desired object. To obtain fast and high-resolution optimization of NeRF, Magic3D [141] proposes a two-stage diffusion framework built on cascaded low-resolution image diffusion prior and high-resolution latent diffusion prior. DATID-3D [116] is a domain adaptation method tailored for 3D generative models using text-to-image diffusion models, which can synthesize diverse images given text prompt without collecting additional information for the target domain.

7.4.5 Text-to-Audio Generation. Text-to-audio generation is the task to transform normal language texts to voice outputs [133, 252]. Grad-TTS [180] presents a novel text-to-speech model with a score-based decoder and diffusion models. It gradually transforms noise predicted by the encoder and is further aligned with text input by the method of Monotonic Alignment Search [182]. Grad-TTS2 [118] improves Grad-TTS in an adaptive way. Diffsound [263] presents a non-autoregressive decoder based on the discrete diffusion model [6, 218], which predicts all the mel-spectrogram tokens in every single step, and then refines the predicted tokens in the following steps. EdiTTS [231] leverages the score-based text-to-speech model to refine a mel-spectrogram prior that is coarsely modified. Instead of estimating the gradient of data density, ProDiff [96] parameterizes the denoising diffusion model by directly predicting the clean data.

7.5 Robust Learning

7.5.1 Data Purification. Robust learning is a class of defense methods that help models perform robustly robust against adversarial perturbations or noises [18, 167, 178, 242, 251, 273]. While adversarial training [156] is a standard defense method to counter adversarial attacks for image classifiers, adversarial purification has shown significant performance as an alternative defense method [273], which purifies attacked images into clean ones with a standalone purification model. Given an adversarial example, DiffPure [167] diffuses it with a small amount of noise following a forward diffusion process and then restores the clean image with a reverse generative process. **Adaptive Denoising Purification (ADP)** [273] demonstrates that an EBM trained with denoising score matching [240] can effectively purify attacked images within just a few steps. It further proposes an effective randomized purification scheme, injecting random noises into images before purification. **Projected Gradient Descent (PGD)** [18] presents a novel stochastic diffusion-based pre-processing robustification, which aims to be a model-agnostic adversarial defense and yield a

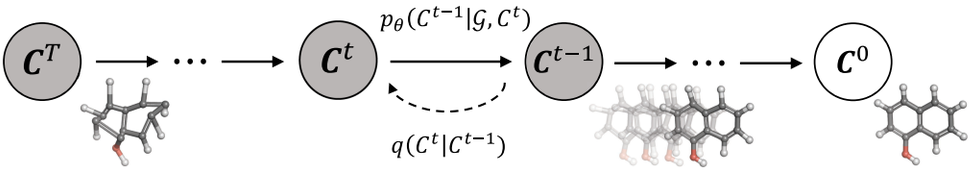


Fig. 10. Molecule-to-conformation diffusion process in GeoDiff [261].

high-quality denoised outcome. In addition, some approaches apply guided diffusion processes for advanced adversarial purification [242, 251].

7.5.2 Generating Synthetic Data for Robust Learning. Another use of diffusion models in robust learning is to generate synthetic data [73, 210, 237, 244]. For example, Wang et al. [244] employ the recent diffusion model to improve adversarial training, and the trained models achieve state-of-the-art performance using only generated data. Sehwag et al. [209] and Um and Ye [237] enforce the generation process of the diffusion models to focus on minority samples such that they can generate high-fidelity samples from low-density regions.

7.6 Interdisciplinary Applications

7.6.1 Molecular Graph Modeling. Graph Neural Networks [80, 254, 269, 286] and corresponding representation learning [81] methods have made significant advances [16, 234, 253, 260, 266, 289] in numerous areas ranging from property prediction [63, 70] to molecule generation [102, 109, 151, 213]. Recently, these molecular graph neural networks have been integrated with diffusion models to explore more intrinsic and informative properties. Torsional diffusion [104] presents a diffusion framework that makes operations on the space of torsion angles with a diffusion process on the hyperspace and an extrinsic-to-intrinsic scoring model. GeoDiff [261] demonstrates that Markov chains evolving with equivariant Markov kernels can produce an invariant distribution, and further designs blocks for the Markov kernels to preserve the desirable equivariance property (see Figure 10). Other methods incorporate the equivariance property into 3D molecule generation [91] and protein generation [3, 15]. Motivated by the classical force field methods for simulating molecular dynamics, ConfGF [212] estimates the gradient fields of log density of atomic coordinates in conformation generation.

7.6.2 Material Design. Solid state materials are the critical foundation of numerous key technologies [22]. **Crystal Diffusion Variational Autoencoder (CDVAE)** [258] incorporates stability as an inductive bias by proposing a noise conditional score network, which simultaneously utilizes permutation, translation, rotation, and periodic invariance properties. Luo et al. (2022) [152] model sequences and structures of complementarity-determining regions with equivariant diffusion, and explicitly target specific antigen structures to generate antibodies at atomic resolution.

7.6.3 Medical Image Reconstruction. An inverse problem aims to recover an unknown signal from observed measurements, and it is an important problem in medical image reconstruction of **Computed Tomography (CT)** and **Magnetic Resonance Imaging (MRI)** [41, 42, 177, 227, 259]. Song et al. (2021) [227] utilize a score-based generative model to reconstruct an image consistent with both the prior and the observed measurements. In [43] Chung and Ye (2022) train a continuous time-dependent score function with denoising score matching, and iterate between the numerical SDE solver and data consistency step for reconstruction at the evaluation stage. Recently, Peng et al. (2022) [177] perform MR reconstruction by gradually guiding the reverse-diffusion process given observed k-space signal, and propose a coarse-to-fine sampling algorithm for efficient sampling.

8 FUTURE DIRECTIONS

Research on diffusion models is in its early stages, with much potential for improvement in both theoretical and empirical aspects. As discussed in early sections, key research directions include efficient sampling and improved likelihood, as well as exploring how diffusion models can handle special data structures, interface with other types of generative models, and be tailored to a range of applications. In addition, we foresee that future research on diffusion models will likely expand to the following avenues.

Revisiting Assumptions. Numerous typical assumptions in diffusion models need to be revisited and analyzed. For example, the assumption that the forward process of diffusion models completely erases any information in data and renders it equivalent to a prior distribution may not always hold. In reality, complete removal of information is unachievable in finite time. It is of great interest to understand when to halt the forward noising process in order to strike a balance between sampling efficiency and sample quality [68]. Recent advances in Schrödinger bridges and optimal transport [35, 50, 52, 214, 221] provide promising alternative solutions, suggesting new formulations for diffusion models that are capable of converging to a specified prior distribution in finite time.

Theoretical Understanding. Diffusion models have emerged as a powerful framework, notably as the only one that can rival generative adversarial networks (GANs) in most applications without resorting to adversarial training. Key to harnessing this potential is an understanding of why and when diffusion models are effective over alternatives for specific tasks. It is important to identify which fundamental characteristics differentiate diffusion models from other types of generative models, such as variational autoencoders, energy-based models, or autoregressive models. Understanding these distinctions will help elucidate why diffusion models are capable of generating samples of excellent quality while achieving top likelihood. Equally important is the need to develop theoretical guidance for selecting and determining various hyperparameters of diffusion models systematically.

Latent Representations. Unlike variational autoencoders or generative adversarial networks, diffusion models are less effective for providing good representations of data in their latent space. As a result, they cannot be easily used for tasks such as manipulating data based on semantic representations. Furthermore, since the latent space in diffusion models often possesses the same dimensionality as the data space, sampling efficiency is negatively affected and the models may not learn the representation schemes well [103].

Pitfalls on Diffusion Models. Despite the fact that diffusion models can generate high-quality synthetic images and can be easily controlled and scaled, recent studies have shown several pitfalls in diffusion models that need to be addressed. For example, diffusion models would learn and even amplify the bias in the training dataset [21, 39], generate improper images [208], and suffer from privacy issues [27]. More efforts need to be made to address these issues in the near future [58, 208, 230].

9 CONCLUSION

In this paper, we present a comprehensive survey on diffusion models from various perspectives. We begin with a self-contained introduction to three fundamental formulations: DDPMs, SGMs, and Score SDEs. Next, we discuss recent efforts to improve diffusion models, highlighting three major directions: sampling efficiency, likelihood maximization, and new techniques for data with special structures. We also explore connections between diffusion models and other generative

models and outlined potential benefits of combining the two. A survey across six application domains illustrates the wide-ranging potential of diffusion models. Finally, we conclude this paper with discussions on challenging issues for future research in this field.

REFERENCES

[1] Juan Miguel Lopez Alcaraz and Nils Strodthoff. 2022. Diffusion-based time series imputation and forecasting with structured state space models. *arXiv preprint arXiv:2208.09399* (2022).

[2] Tomer Amit, Eliya Nachmani, Tal Shaharbany, and Lior Wolf. 2021. SegDiff: Image segmentation with diffusion probabilistic models. *arXiv preprint arXiv:2112.00390* (2021).

[3] Namrata Anand and Tudor Achim. 2022. Protein structure and sequence generation with equivariant denoising diffusion probabilistic models. *arXiv preprint arXiv:2205.15019* (2022).

[4] Brian D. O. Anderson. 1982. Reverse-time diffusion equation models. *Stochastic Processes and Their Applications* 12, 3 (1982), 313–326.

[5] Uri M. Ascher and Linda R. Petzold. 1998. *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*. SIAM.

[6] Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. 2021. Structured denoising diffusion models in discrete state-spaces. In *Advances in Neural Information Processing Systems*.

[7] Omri Avrahami, Dani Lischinski, and Ohad Fried. 2022. Blended diffusion for text-driven editing of natural images. In *IEEE Conference on Computer Vision and Pattern Recognition*. 18208–18218.

[8] Wele Gedara Chaminda Bandara, Nithin Gopalakrishnan Nair, and Vishal M. Patel. 2022. DDPM-CD: Remote sensing change detection using denoising diffusion probabilistic models. *arXiv preprint arXiv:2206.11892* (2022).

[9] Hritik Bansal and Aditya Grover. 2023. Leaving reality to imagination: Robust classification via generated datasets. In *International Conference on Learning Representations*.

[10] Fan Bao, Chongxuan Li, Jun Zhu, and Bo Zhang. 2021. Analytic-DPM: An analytic estimate of the optimal reverse variance in diffusion probabilistic models. In *International Conference on Learning Representations*.

[11] Dmitry Baranchuk, Andrey Voynov, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. 2021. Label-efficient semantic segmentation with diffusion models. In *International Conference on Learning Representations*.

[12] Georgios Batzolis, Jan Stanczuk, Carola-Bibiane Schönlieb, and Christian Etmann. 2021. Conditional image generation with score-based diffusion models. *arXiv preprint arXiv:2111.13606* (2021).

[13] Samy Bengio and Yoshua Bengio. 2000. Taking on the curse of dimensionality in joint distributions using neural networks. *IEEE Transactions on Neural Networks* (2000).

[14] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. *Journal of Machine Learning Research* 3 (2003), 1137–1155.

[15] Helen M. Berman, John Westbrook, Zukang Feng, Gary Gilliland, Talapady N. Bhat, Helge Weissig, Ilya N. Shindyalov, and Philip E. Bourne. 2000. The protein data bank. *Nucleic Acids Research* 28, 1 (2000), 235–242.

[16] Piotr Bielak, Tomasz Kajdanowicz, and Nitesh V. Chawla. 2021. Graph Barlow Twins: A self-supervised representation learning framework for graphs. *arXiv preprint arXiv:2106.02466* (2021).

[17] Mikołaj Bińkowski, Dougal J. Sutherland, Michael Arbel, and Arthur Gretton. 2018. Demystifying MMD GANs. In *International Conference on Learning Representations*.

[18] Tsachi Blau, Roy Ganz, Bahjat Kawar, Alex Bronstein, and Michael Elad. 2022. Threat model-agnostic adversarial defense using diffusion models. *arXiv preprint arXiv:2207.08089* (2022).

[19] Emmanuel Asiedu Brempong, Simon Kornblith, Ting Chen, Niki Parmar, Matthias Minderer, and Mohammad Norouzi. 2022. Denoising pretraining for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*. 4175–4186.

[20] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*.

[21] Blake Bullwinkel, Kristen Grabarz, Lily Ke, Scarlett Gong, Chris Tanner, and Joshua Allen. 2022. Evaluating the fairness impact of differentially private synthetic data. *arXiv preprint arXiv:2205.04321* (2022).

[22] Keith T. Butler, Daniel W. Davies, Hugh Cartwright, Olexandr Isayev, and Aron Walsh. 2018. Machine learning for molecular and materials science. *Nature* 559, 7715 (2018), 547–555.

[23] Ruojin Cai, Guandao Yang, Hadar Averbuch-Elor, Zekun Hao, Serge Belongie, Noah Snavely, and Bharath Hariharan. 2020. Learning gradient fields for shape generation. In *European Conference on Computer Vision*. 364–381.

[24] Andrew Campbell, Joe Benton, Valentin De Bortoli, Tom Rainforth, George Deligiannidis, and Arnaud Doucet. 2022. A continuous time framework for discrete denoising models. *arXiv preprint arXiv:2205.14987* (2022).

[25] Chentao Cao, Zhuo-Xu Cui, Shaonan Liu, Dong Liang, and Yanjie Zhu. 2022. High-frequency space diffusion models for accelerated MRI. *arXiv preprint arXiv:2208.05481* (2022).

[26] Wei Cao, Dong Wang, Jian Li, Hao Zhou, Lei Li, and Yitan Li. 2018. BRITS: Bidirectional recurrent imputation for time series. In *Advances in Neural Information Processing Systems*, Vol. 31.

[27] Nicholas Carlini, Jamie Hayes, Milad Nasr, Matthew Jagielski, Vikash Sehwag, Florian Tramèr, Borja Balle, Daphne Ippolito, and Eric Wallace. 2023. Extracting training data from diffusion models. *arXiv preprint arXiv:2301.13188* (2023).

[28] Nicholas Carlini, Florian Tramer, Krishnamurthy Dvijotham, and Kolter J. Zico. 2022. (Certified!!) Adversarial robustness for free! *arXiv preprint arXiv:2206.10550* (2022).

[29] Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T. Freeman. 2022. MaskGIT: Masked generative image transformer. In *IEEE Conference on Computer Vision and Pattern Recognition*. 11315–11325.

[30] Zhengping Che, Sanjay Purushotham, Kyunghyun Cho, David Sontag, and Yan Liu. 2018. Recurrent neural networks for multivariate time series with missing values. *Scientific Reports* 8, 1 (2018), 1–12.

[31] Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. 2013. One billion word benchmark for measuring progress in statistical language modeling. *arXiv preprint arXiv:1312.3005* (2013).

[32] Dong Chen, Xinda Qi, Yu Zheng, Yuzhen Lu, and Zhaojian Li. 2022. Deep data augmentation for weed recognition enhancement: A diffusion probabilistic model and transfer learning based approach. *arXiv preprint arXiv:2210.09509* (2022).

[33] Nanxin Chen, Yu Zhang, Heiga Zen, Ron J. Weiss, Mohammad Norouzi, and William Chan. 2020. WaveGrad: Estimating gradients for waveform generation. *arXiv preprint arXiv:2009.00713* (2020).

[34] Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. 2018. Neural ordinary differential equations. *arXiv preprint arXiv:1806.07366* (2018).

[35] Tianrong Chen, Guan-Horng Liu, and Evangelos Theodorou. 2021. Likelihood training of Schrödinger bridge using forward-backward SDEs theory. In *International Conference on Learning Representations*.

[36] Ting Chen, Ruixiang Zhang, and Geoffrey Hinton. 2022. Analog bits: Generating discrete data using diffusion models with self-conditioning. *arXiv preprint arXiv:2208.04202* (2022).

[37] Rewon Child. 2020. Very deep VAEs generalize autoregressive models and can outperform them on images. In *International Conference on Learning Representations*.

[38] Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. 2019. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509* (2019).

[39] Jaemin Cho, Abhay Zala, and Mohit Bansal. 2022. DALL-EVAL: Probing the reasoning skills and social biases of text-to-image generative models. *arXiv preprint arXiv:2202.04053* (2022).

[40] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. PaLM: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311* (2022).

[41] Hyungjin Chung, Eun Sun Lee, and Jong Chul Ye. 2022. MR image denoising and super-resolution using regularized reverse diffusion. *arXiv preprint arXiv:2203.12621* (2022).

[42] Hyungjin Chung, Byeongsu Sim, and Jong Chul Ye. 2022. Come-closer-diffuse-faster: Accelerating conditional diffusion models for inverse problems through stochastic contraction. In *IEEE Conference on Computer Vision and Pattern Recognition*. 12413–12422.

[43] Hyungjin Chung and Jong Chul Ye. 2022. Score-based diffusion models for accelerated MRI. *Medical Image Analysis* (2022), 102479.

[44] Rob Cornish, Anthony Caterini, George Deligiannidis, and Arnaud Doucet. 2020. Relaxing bijectivity constraints with continuously indexed normalising flows. In *International Conference on Machine Learning*. 2133–2143.

[45] Antonia Creswell, Tom White, Vincent Dumoulin, Kai Arulkumaran, Biswa Sengupta, and Anil A. Bharath. 2018. Generative adversarial networks: An overview. *IEEE Signal Processing Magazine* 35, 1 (2018), 53–65.

[46] Katherine Crowson, Stella Biderman, Daniel Kornis, Dashiell Stander, Eric Hallahan, Louis Castricato, and Edward Raff. 2022. VQGAN-CLIP: Open domain image generation and editing with natural language guidance. *arXiv preprint arXiv:2204.08583* (2022).
[47] Koller Daphne and Friedman Nir. 2009. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press.
[48] Salman U. H. Dar, Şaban Öztürk, Yilmaz Korkmaz, Gokberk Elmas, Muzaffer Özbey, Alper Güngör, and Tolga Çukur. 2022. Adaptive diffusion priors for accelerated MRI reconstruction. *arXiv preprint arXiv:2207.05876* (2022).
[49] Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. 2019. Plug and play language models: A simple approach to controlled text generation. In *International Conference on Learning Representations*.
[50] Valentin De Bortoli, Arnaud Doucet, Jeremy Heng, and James Thornton. 2021. Simulating diffusion bridges with score matching. *arXiv preprint arXiv:2111.07243* (2021).
[51] Valentin De Bortoli, Emile Mathieu, Michael Hutchinson, James Thornton, Yee Whye Teh, and Arnaud Doucet. 2022. Riemannian score-based generative modeling. *arXiv preprint arXiv:2202.02763* (2022).
[52] Valentin De Bortoli, James Thornton, Jeremy Heng, and Arnaud Doucet. 2021. Diffusion Schrödinger bridge with applications to score-based generative modeling. In *Advances in Neural Information Processing Systems*, Vol. 34. 17695–17709.
[53] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. ImageNet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*. 248–255.
[54] Prafulla Dhariwal and Alexander Nichol. 2021. Diffusion models beat GANs on image synthesis. In *Advances in Neural Information Processing Systems*, Vol. 34. 8780–8794.
[55] Laurent Dinh, David Krueger, and Yoshua Bengio. 2014. NICE: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516* (2014).
[56] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. 2017. Density estimation using Real NVP. In *International Conference on Learning Representations*.
[57] Laurent Dinh, Jascha Sohl-Dickstein, Hugo Larochelle, and Razvan Pascanu. 2019. A RAD approach to deep mixture models. *arXiv preprint arXiv:1903.07714* (2019).
[58] Tim Dockhorn, Tianshi Cao, Arash Vahdat, and Karsten Kreis. 2022. Differentially private diffusion models. *arXiv preprint arXiv:2210.09929* (2022).
[59] Tim Dockhorn, Arash Vahdat, and Karsten Kreis. 2021. Score-based generative modeling with critically-damped Langevin diffusion. In *International Conference on Learning Representations*.
[60] Tim Dockhorn, Arash Vahdat, and Karsten Kreis. 2022. GENIE: Higher-order denoising diffusion solvers. *Advances in Neural Information Processing Systems* (2022).
[61] Carl Doersch. 2016. Tutorial on variational autoencoders. *arXiv preprint arXiv:1606.05908* (2016).
[62] Yifan Du, Zikang Liu, Junyi Li, and Wayne Xin Zhao. 2022. A survey of vision-language pre-trained models. *arXiv preprint arXiv:2202.10936* (2022).
[63] David K. Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P. Adams. 2015. Convolutional networks on graphs for learning molecular fingerprints. In *Advances in Neural Information Processing Systems*, Vol. 28.
[64] Emadeldeen Eldele, Mohamed Ragab, Zhenghua Chen, Min Wu, Chee Keong Kwoh, Xiaoli Li, and Cuntai Guan. 2021. Time-series representation learning via temporal and contextual contrasting. *arXiv preprint arXiv:2106.14112* (2021).
[65] Patrick Esser, Robin Rombach, and Bjorn Ommer. 2021. Taming transformers for high-resolution image synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition*. 12873–12883.
[66] Charles Fefferman, Sanjoy Mitter, and Hariharan Narayanan. 2016. Testing the manifold hypothesis. *Journal of the American Mathematical Society* 29, 4 (2016), 983–1049.
[67] Vincent Fortuin, Dmitry Baranchuk, Gunnar Ratsch, and Stephan Mandt. 2020. GP-VAE: Deep probabilistic time series imputation. In *International Conference on Artificial Intelligence and Statistics*. 1651–1661.
[68] Giulio Franzese, Simone Rossi, Lixuan Yang, Alessandro Finamore, Dario Rossi, Maurizio Filippone, and Pietro Michiardi. 2022. How much is enough? A study on diffusion times in score-based generative models. *arXiv preprint arXiv:2206.05173* (2022).
[69] Ruiqi Gao, Yang Song, Ben Poole, Ying Nian Wu, and Diederik P. Kingma. 2020. Learning energy-based models by diffusion recovery likelihood. *arXiv preprint arXiv:2012.08125* (2020).
[70] Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. 2017. Neural message passing for quantum chemistry. In *International Conference on Machine Learning*. 1263–1272.
[71] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, Vol. 27. 139–144.

[72] Marco Gori, Gabriele Monfardini, and Franco Scarselli. 2005. A new model for learning in graph domains. In *International Joint Conference on Neural Networks*, Vol. 2. 729–734.
[73] Sven Gowal, Sylvestre-Alvise Rebuffi, Olivia Wiles, Florian Stimberg, Dan Andrei Calian, and Timothy A. Mann. 2021. Improving robustness using generated data. *Advances in Neural Information Processing Systems* 34 (2021), 4218–4233.
[74] Will Grathwohl, Ricky T. Q. Chen, Jesse Bettencourt, and David Duvenaud. 2019. Scalable reversible generative models with free-form continuous dynamics. In *International Conference on Learning Representations*.
[75] Alex Graves. 2013. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850* (2013).
[76] Ulf Grenander and Michael I. Miller. 1994. Representations of knowledge in complex systems. *Journal of the Royal Statistical Society: Series B (Methodological)* 56, 4 (1994), 549–581.
[77] Albert Gu, Karan Goel, and Christopher Re. 2021. Efficiently modeling long sequences with structured state spaces. In *International Conference on Learning Representations*.
[78] Shuyang Gu, Dong Chen, Jianmin Bao, Fang Wen, Bo Zhang, Dongdong Chen, Lu Yuan, and Baining Guo. 2022. Vector quantized diffusion model for text-to-image synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition*. 10696–10706.
[79] Jie Gui, Zhenan Sun, Yonggang Wen, Dacheng Tao, and Jieping Ye. 2021. A review on generative adversarial networks: Algorithms, theory, and applications. *IEEE Transactions on Knowledge and Data Engineering* (2021).
[80] William L. Hamilton, Rex Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*. 1025–1035.
[81] William L. Hamilton, Rex Ying, and Jure Leskovec. 2017. Representation learning on graphs: Methods and applications. *arXiv preprint arXiv:1709.05584* (2017).
[82] Songqiao Han, Xiyang Hu, Hailiang Huang, Mingqi Jiang, and Yue Zhao. 2022. ADBench: Anomaly detection benchmark. *arXiv preprint arXiv:2206.09426* (2022).
[83] William Harvey, Saeid Naderiparizi, Vaden Masrani, Christian Weilbach, and Frank Wood. 2022. Flexible diffusion modeling of long videos. *arXiv preprint arXiv:2205.11495* (2022).
[84] Ruifei He, Shuyang Sun, Xin Yu, Chuhui Xue, Wenqing Zhang, Philip Torr, Song Bai, and Xiaojuan Qi. 2022. Is synthetic data from generative models ready for image recognition?. In *International Conference on Learning Representations*.
[85] Roei Herzig, Amir Bar, Huijuan Xu, Gal Chechik, Trevor Darrell, and Amir Globerson. 2020. Learning canonical representations for scene graph to image generation. 210–227.
[86] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P. Kingma, Ben Poole, Mohammad Norouzi, David J. Fleet, and Tim Salimans. 2022. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303* (2022).
[87] Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, Vol. 33. 6840–6851.
[88] Jonathan Ho, Chitwan Saharia, William Chan, David J. Fleet, Mohammad Norouzi, and Tim Salimans. 2022. Cascaded diffusion models for high fidelity image generation. *Journal of Machine Learning Research* 23 (2022), 47–1.
[89] Jonathan Ho and Tim Salimans. 2022. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598* (2022).
[90] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. 2022. Video diffusion models. *arXiv preprint arXiv:2204.03458* (2022).
[91] Emiel Hoogeboom, Victor Garcia Satorras, Clement Vignac, and Max Welling. 2022. Equivariant diffusion for molecule generation in 3D. *arXiv print arXiv:2203.17003* (2022).
[92] Emiel Hoogeboom, Alexey A. Gritsenko, Jasmijn Bastings, Ben Poole, Rianne van den Berg, and Tim Salimans. 2021. Autoregressive diffusion models. In *International Conference on Learning Representations*.
[93] Emiel Hoogeboom, Didrik Nielsen, Priyank Jaini, Patrick Forré, and Max Welling. 2021. Argmax flows and multinomial diffusion: Learning categorical distributions. In *Advances in Neural Information Processing Systems*, Vol. 34. 12454–12465.
[94] Chin-Wei Huang, Milad Aghajohari, Joey Bose, Prakash Panangaden, and Aaron C. Courville. 2022. Riemannian diffusion models. *Advances in Neural Information Processing Systems* 35 (2022), 2750–2761.
[95] Chin-Wei Huang, Jae Hyun Lim, and Aaron C. Courville. 2021. A variational perspective on diffusion-based generative models and score matching. In *Advances in Neural Information Processing Systems*, Vol. 34. 22863–22876.
[96] Rongjie Huang, Zhou Zhao, Huadai Liu, Jinglin Liu, Chenye Cui, and Yi Ren. 2022. ProDiff: Progressive fast diffusion model for high-quality text-to-speech. *arXiv preprint arXiv:2207.06389* (2022).
[97] Michael F. Hutchinson. 1989. A stochastic estimator of the trace of the influence matrix for Laplacian smoothing splines. *Communications in Statistics-Simulation and Computation* 18, 3 (1989), 1059–1076.
[98] Aapo Hyvärinen. 2005. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research* 6 (2005), 695–709.

[99] Touseef Iqbal and Shaima Qureshi. 2020. The survey: Text generation models in deep learning. *Journal of King Saud University-Computer and Information Sciences* (2020).
[100] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. 2017. Image-to-image translation with conditional adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition*. 1125–1134.
[101] Long Jin, Justin Lazarow, and Zhuowen Tu. 2017. Introspective classification with convolutional nets. In *Advances in Neural Information Processing Systems*, Vol. 30. 823–833.
[102] Wengong Jin, Regina Barzilay, and Tommi Jaakkola. 2018. Junction tree variational autoencoder for molecular graph generation. In *International Conference on Machine Learning*. 2323–2332.
[103] Bowen Jing, Gabriele Corso, Renato Berlinghieri, and Tommi Jaakkola. 2022. Subspace diffusion generative models. *arXiv preprint arXiv:2205.01490* (2022).
[104] Bowen Jing, Gabriele Corso, Jeffrey Chang, Regina Barzilay, and Tommi Jaakkola. 2022. Torsional diffusion for molecular conformer generation. *arXiv preprint arXiv:2206.01729* (2022).
[105] Jaehyeong Jo, Seul Lee, and Sung Ju Hwang. 2022. Score-based generative modeling of graphs via the system of stochastic differential equations. *arXiv preprint arXiv:2202.02514* (2022).
[106] Justin Johnson, Agrim Gupta, and Li Fei-Fei. 2018. Image generation from scene graphs. In *IEEE Conference on Computer Vision and Pattern Recognition*. 1219–1228.
[107] Alexia Jolicoeur-Martineau, Ke Li, Rémi Piché-Taillefer, Tal Kachman, and Ioannis Mitliagkas. 2021. Gotta go fast when generating data with score-based models. *arXiv preprint arXiv:2105.14080* (2021).
[108] Alexia Jolicoeur-Martineau, Rémi Piché-Taillefer, Rémi Tachet des Combes, and Ioannis Mitliagkas. 2020. Adversarial score matching and improved sampling for image generation. *arXiv preprint arXiv:2009.05475* (2020).
[109] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A. A. Kohl, Andrew J. Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Michalina Pacholska, Tamas Berghammer, Sebastian Bodenstein, David Silver, Oriol Vinyals, Andrew W. Senior, Koray Kavukcuoglu, Pushmeet Kohli, and Demis Hassabis. 2021. Highly accurate protein structure prediction with AlphaFold. *Nature* 596, 7873 (2021), 583–589.
[110] Nal Kalchbrenner, Erich Elsen, Karen Simonyan, Seb Noury, Norman Casagrande, Edward Lockhart, Florian Stimberg, Aäron van den Oord, Sander Dieleman, and Koray Kavukcuoglu. 2018. Efficient neural audio synthesis. In *International Conference on Machine Learning*. 2410–2419.
[111] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. 2022. Elucidating the design space of diffusion-based generative models. *arXiv preprint arXiv:2206.00364* (2022).
[112] Bahjat Kawar, Roy Ganz, and Michael Elad. 2022. Enhancing diffusion-based image synthesis with robust classifier guidance. *arXiv preprint arXiv:2208.08664* (2022).
[113] Bahjat Kawar, Gregory Vaksman, and Michael Elad. 2021. Stochastic image denoising by sampling from the posterior distribution. In *International Conference on Computer Vision*. 1866–1875.
[114] Nitish Shirish Keskar, Bryan McCann, Lav R. Varshney, Caiming Xiong, and Richard Socher. 2019. CTRL: A conditional transformer language model for controllable generation. *arXiv preprint arXiv:1909.05858* (2019).
[115] Boah Kim, Inhwa Han, and Jong Chul Ye. 2021. DiffuseMorph: Unsupervised deformable image registration along continuous trajectory using diffusion models. *arXiv preprint arXiv:2112.05149* (2021).
[116] Gwanghyun Kim and Se Young Chun. 2023. DATID-3D: Diversity-preserved domain adaptation using text-to-image diffusion for 3D generative model. In *IEEE Conference on Computer Vision and Pattern Recognition*. 14203–14213.
[117] Gwanghyun Kim, Taesung Kwon, and Jong Chul Ye. 2022. DiffusionCLIP: Text-guided diffusion models for robust image manipulation. In *IEEE Conference on Computer Vision and Pattern Recognition*. 2426–2435.
[118] Sungwon Kim, Heeseung Kim, and Sungroh Yoon. 2022. Guided-TTS 2: A diffusion model for high-quality adaptive text-to-speech with untranscribed data. *arXiv preprint arXiv:2205.15370* (2022).
[119] Taesup Kim and Yoshua Bengio. 2016. Deep directed generative models with energy-based probability estimation. *arXiv preprint arXiv:1606.03439* (2016).
[120] Diederik Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. 2021. Variational diffusion models. In *Advances in Neural Information Processing Systems*, Vol. 34. 21696–21707.
[121] Diederik P. Kingma and Prafulla Dhariwal. 2018. Glow: Generative flow with invertible 1x1 convolutions. *arXiv preprint arXiv:1807.03039* (2018).
[122] Diederik P. Kingma and Max Welling. 2013. Auto-encoding variational Bayes. *arXiv preprint arXiv:1312.6114* (2013).
[123] Diederik P. Kingma and Max Welling. 2019. An introduction to variational autoencoders. *Foundations and Trends in Machine Learning* 12, 4 (2019), 307–392.
[124] Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. 2020. DiffWave: A versatile diffusion model for audio synthesis. *arXiv preprint arXiv:2009.09761* (2020).

[125] Ben Krause, Akhilesh Deepak Gotmare, Bryan McCann, Nitish Shirish Keskar, Shafiq Joty, Richard Socher, and Nazneen Fatema Rajani. 2020. GeDi: Generative discriminator guided sequence generation. *arXiv preprint arXiv:2009.06367* (2020).
[126] Alex Krizhevsky. 2009. Learning multiple layers of features from tiny images. (2009).
[127] Hugo Larochelle and Iain Murray. 2011. The neural autoregressive distribution estimator. In *International Conference on Artificial Intelligence and Statistics*.
[128] Justin Lazarow, Long Jin, and Zhuowen Tu. 2017. Introspective neural networks for generative modeling. In *International Conference on Computer Vision*. 2774–2783.
[129] Yann LeCun, Sumit Chopra, Raia Hadsell, Marc'Aurelio Ranzato, and Fujie Huang. 2006. A tutorial on energy-based learning. *Predicting Structured Data* (2006).
[130] Jin Sub Lee, Jisun Kim, and Philip M. Kim. 2022. ProteinSGM: Score-based generative modeling for de novo protein design. *bioRxiv* (2022), 2022–07.
[131] Kwonjoon Lee, Weijian Xu, Fan Fan, and Zhuowen Tu. 2018. Wasserstein introspective neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*. 3702–3711.
[132] Seul Lee, Jaehyeong Jo, and Sung Ju Hwang. 2022. Exploring chemical space with score-based out-of-distribution generation. *arXiv preprint arXiv:2206.07632* (2022).
[133] Alon Levkovitch, Eliya Nachmani, and Lior Wolf. 2022. Zero-shot voice conditioning for denoising diffusion TTS models. *arXiv preprint arXiv:2206.02246* (2022).
[134] Haoying Li, Yifan Yang, Meng Chang, Huajun Feng, Zhi hai Xu, Qi Li, and Yue ting Chen. 2022. SRDiff: Single image super-resolution with diffusion probabilistic models. *Neurocomputing* 479 (2022), 47–59.
[135] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. 2022. BLIP: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning*. PMLR, 12888–12900.
[136] Junyi Li, Tianyi Tang, Gaole He, Jinhao Jiang, Xiaoxuan Hu, Puzhao Xie, Zhipeng Chen, Zhuohao Yu, Wayne Xin Zhao, and Ji-Rong Wen. 2021. TextBox: A unified, modularized, and extensible framework for text generation. *arXiv preprint arXiv:2101.02046* (2021).
[137] Junyi Li, Tianyi Tang, Wayne Xin Zhao, and Ji-Rong Wen. 2021. Pretrained language models for text generation: A survey. *arXiv preprint arXiv:2105.10311* (2021).
[138] Shiyang Li, Xiaoyong Jin, Yao Xuan, Xiyou Zhou, Wenhu Chen, Yu-Xiang Wang, and Xifeng Yan. 2019. Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. In *Advances in Neural Information Processing Systems*, Vol. 32.
[139] Xiang Lisa Li, John Thickstun, Ishaan Gulrajani, Percy Liang, and Tatsunori B. Hashimoto. 2022. Diffusion-LM improves controllable text generation. *arXiv preprint arXiv:2205.14217* (2022).
[140] Yikang Li, Tao Ma, Yeqi Bai, Nan Duan, Sining Wei, and Xiaogang Wang. 2019. PasteGAN: A semi-parametric method to generate image from scene graph. *Advances in Neural Information Processing Systems* 32.
[141] Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. 2022. Magic3D: High-resolution text-to-3D content creation. *arXiv preprint arXiv:2211.10440* (2022).
[142] Luping Liu, Yi Ren, Zhijie Lin, and Zhou Zhao. 2021. Pseudo numerical methods for diffusion models on manifolds. In *International Conference on Learning Representations*.
[143] Aaron Lou, Derek Lim, Isay Katsman, Leo Huang, Qingxuan Jiang, Ser Nam Lim, and Christopher M. De Sa. 2020. Neural manifold ordinary differential equations. *Advances in Neural Information Processing Systems* 33 (2020), 17548–17558.
[144] Cheng Lu, Kaiwen Zheng, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. 2022. Maximum likelihood training for score-based diffusion ODEs by high order denoising score matching. In *International Conference on Machine Learning*. 14429–14460.
[145] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. 2022. DPM-solver: A fast ODE solver for diffusion probabilistic model sampling in around 10 steps. *arXiv preprint arXiv:2206.00927* (2022).
[146] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. 2022. Repaint: Inpainting using denoising diffusion probabilistic models. In *IEEE Conference on Computer Vision and Pattern Recognition*. 11461–11471.
[147] Eric Luhman and Troy Luhman. 2021. Knowledge distillation in iterative generative models for improved sampling speed. *arXiv preprint arXiv:2101.02388* (2021).
[148] Calvin Luo. 2022. Understanding diffusion models: A unified perspective. *arXiv preprint arXiv:2208.11970* (2022).
[149] Shitong Luo and Wei Hu. 2021. Diffusion probabilistic models for 3D point cloud generation. In *IEEE Conference on Computer Vision and Pattern Recognition*. 2837–2845.

[150] Shitong Luo and Wei Hu. 2021. Score-based point cloud denoising. In *International Conference on Computer Vision*. 4583–4592.
[151] Shitong Luo, Chence Shi, Minkai Xu, and Jian Tang. 2021. Predicting molecular conformation via dynamic graph score matching. In *Advances in Neural Information Processing Systems*, Vol. 34. 19784–19795.
[152] Shitong Luo, Yufeng Su, Xingang Peng, Sheng Wang, Jian Peng, and Jianzhu Ma. 2022. Antigen-specific antibody design and optimization with diffusion-based generative models for protein structures. *Advances in Neural Information Processing Systems* 35 (2022), 9754–9767.
[153] Yonghong Luo, Xiangrui Cai, Ying Zhang, Jun Xu, and Xiaojie Yuan. 2018. Multivariate time series imputation with generative adversarial networks. In *Advances in Neural Information Processing Systems*, Vol. 31.
[154] Zhaoyang Lyu, Zhifeng Kong, X. U. Xudong, Liang Pan, and Dahua Lin. 2021. A conditional point diffusion-refinement paradigm for 3D point cloud completion. In *International Conference on Learning Representations*.
[155] Zhaoyang Lyu, Xudong Xu, Ceyuan Yang, Dahua Lin, and Bo Dai. 2022. Accelerating diffusion models via early stop of the diffusion process. *arXiv preprint arXiv:2205.12524* (2022).
[156] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*.
[157] Emile Mathieu and Maximilian Nickel. 2020. Riemannian continuous normalizing flows. *Advances in Neural Information Processing Systems* 33 (2020), 2503–2515.
[158] Siyuan Mei, Fuxin Fan, and Andreas Maier. 2022. Metal inpainting in CBCT projections using score-based generative model. *arXiv preprint arXiv:2209.09733* (2022).
[159] Gábor Melis, Chris Dyer, and Phil Blunsom. 2018. On the state of the art of evaluation in neural language models. In *International Conference on Learning Representations*.
[160] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. 2021. SDEdit: Guided image synthesis and editing with stochastic differential equations. In *International Conference on Learning Representations*.
[161] Chenlin Meng, Jiaming Song, Yang Song, Shengjia Zhao, and Stefano Ermon. 2020. Improved autoregressive modeling with distribution smoothing. In *International Conference on Learning Representations*.
[162] Stephen Merity, Nitish Shirish Keskar, and Richard Socher. 2018. Regularizing and optimizing LSTM language models. In *International Conference on Learning Representations*.
[163] Nicholas Metropolis and Stanislaw Ulam. 1949. The Monte Carlo method. *Journal of the American Statistical Association* 44, 247 (1949), 335–341.
[164] Gal Metzer, Elad Richardson, Or Patashnik, Raja Giryes, and Daniel Cohen-Or. 2023. Latent-NeRF for shape-guided generation of 3D shapes and textures. In *IEEE Conference on Computer Vision and Pattern Recognition*. 12663–12673.
[165] Alexander Quinn Nichol and Prafulla Dhariwal. 2021. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*. 8162–8171.
[166] Alexander Quinn Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob Mcgrew, Ilya Sutskever, and Mark Chen. 2022. GLIDE: Towards photorealistic image generation and editing with text-guided diffusion models. In *International Conference on Machine Learning*. 16784–16804.
[167] Weili Nie, Brandon Guo, Yujia Huang, Chaowei Xiao, Arash Vahdat, and Anima Anandkumar. 2022. Diffusion models for adversarial purification. *arXiv preprint arXiv:2205.07460* (2022).
[168] Erik Nijkamp, Mitch Hill, Tian Han, Song-Chun Zhu, and Ying Nian Wu. 2019. On the anatomy of MCMC-based maximum likelihood learning of energy-based models. *arXiv preprint arXiv:1903.12370* (2019).
[169] Chenhao Niu, Yang Song, Jiaming Song, Shengjia Zhao, Aditya Grover, and Stefano Ermon. 2020. Permutation invariant graph generation via score-based generative modeling. In *International Conference on Artificial Intelligence and Statistics*. 4474–4484.
[170] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. 2016. WaveNet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499* (2016).
[171] Boris N. Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. 2019. N-BEATS: Neural basis expansion analysis for interpretable time series forecasting. In *International Conference on Learning Representations*.
[172] Irfan Pratama, Adhistya Erna Permanasari, Igi Ardiyanto, and Rini Indrayani. 2016. A review of missing values handling methods on time-series data. In *International Conference on Information Technology Systems and Innovation*, IEEE, 1–6.
[173] Muzaffer Özbey, Salman U. H. Dar, Hasan A. Bedel, Onat Dalmaz, Şaban Özturk, Alper Güngör, and Tolga Çukur. 2022. Unsupervised medical image translation with adversarial diffusion models. *arXiv preprint arXiv:2207.08208* (2022).
[174] George Papamakarios, Eric T. Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. 2021. Normalizing flows for probabilistic modeling and inference. *Journal of Machine Learning Research* 22, 57 (2021), 1–64.

[175] Giorgio Parisi. 1981. Correlation functions and computer simulations. *Nuclear Physics B* 180, 3 (1981), 378–384.
[176] Sung Woo Park, Kyungjae Lee, and Junseok Kwon. 2021. Neural Markov controlled SDE: Stochastic optimization for continuous-time data. In *International Conference on Learning Representations*.
[177] Cheng Peng, Pengfei Guo, S. Kevin Zhou, Vishal M. Patel, and Rama Chellappa. 2022. Towards performant and reliable undersampled MR reconstruction via diffusion model sampling. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*. 623–633.
[178] Stanislav Pidhorskyi, Donald A. Adjeroh, and Gianfranco Doretto. 2020. Adversarial latent autoencoders. In *IEEE Conference on Computer Vision and Pattern Recognition*. 14104–14113.
[179] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. 2022. DreamFusion: Text-to-3D using 2D diffusion. *arXiv preprint arXiv:2209.14988* (2022).
[180] Vadim Popov, Ivan Vovk, Vladimir Gogoryan, Tasnima Sadekova, and Mikhail Kudinov. 2021. Grad-TTS: A diffusion probabilistic model for text-to-speech. In *International Conference on Machine Learning*. 8599–8608.
[181] Chenyang Qi, Xiaodong Cun, Yong Zhang, Chenyang Lei, Xintao Wang, Ying Shan, and Qifeng Chen. 2023. FateZero: Fusing attentions for zero-shot text-based video editing. In *International Conference on Computer Vision*.
[182] Lawrence R. Rabiner. 1989. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE* 77, 2 (1989), 257–286.
[183] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*. 8748–8763.
[184] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog* 1, 8 (2019), 9.
[185] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research* 21, 1 (2020), 5485–5551.
[186] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. 2022. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125* (2022).
[187] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. 2021. Zero-shot text-to-image generation. In *International Conference on Machine Learning*. 8821–8831.
[188] Martin Raphan and Eero P. Simoncelli. 2007. Learning to be Bayesian without supervision. In *Advances in Neural Information Processing Systems*. 1145–1152.
[189] Martin Raphan and Eero P. Simoncelli. 2011. Least squares estimation without priors or supervision. *Neural Computation* 23, 2 (2011), 374–420.
[190] Kashif Rasul, Calvin Seward, Ingmar Schuster, and Roland Vollgraf. 2021. Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting. In *International Conference on Machine Learning*. 8857–8868.
[191] Kashif Rasul, Abdul-Saboor Sheikh, Ingmar Schuster, Urs M. Bergmann, and Roland Vollgraf. 2020. Multivariate probabilistic time series forecasting via conditioned normalizing flows. In *International Conference on Learning Representations*.
[192] Lillian J. Ratliff, Samuel A. Burden, and S. Shankar Sastry. 2013. Characterization and computation of local Nash equilibria in continuous games. In *Annual Allerton Conference on Communication, Control, and Computing*. 917–924.
[193] Danilo Rezende and Shakir Mohamed. 2015. Variational inference with normalizing flows. In *International Conference on Machine Learning*. 1530–1538.
[194] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. 2014. Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning*. 1278–1286.
[195] Oren Rippel and Ryan Prescott Adams. 2013. High-dimensional probability estimation with deep density models. *arXiv preprint arXiv:1302.5125* (2013).
[196] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *IEEE Conference on Computer Vision and Pattern Recognition*. 10684–10695.
[197] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. 2022. DreamBooth: Fine tuning text-to-image diffusion models for subject-driven generation. *arXiv preprint arXiv:2208.12242* (2022).
[198] Chitwan Saharia, William Chan, Huiwen Chang, Chris Lee, Jonathan Ho, Tim Salimans, David Fleet, and Mohammad Norouzi. 2022. Palette: Image-to-image diffusion models. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Proceedings*. 1–10.
[199] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L. Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, Jonathan Ho, David J. Fleet, and Mohammad Norouzi. 2022. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv preprint arXiv:2205.11487* (2022).

[200] Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J. Fleet, and Mohammad Norouzi. 2022. Image super-resolution via iterative refinement. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2022).

[201] Tim Salimans and Jonathan Ho. 2021. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*.

[202] Tim Salimans and Jonathan Ho. 2021. Should EBMs model the energy or the score?. In *International Conference on Learning Representations*.

[203] David Salinas, Michael Bohlke-Schneider, Laurent Callot, Roberto Medico, and Jan Gasthaus. 2019. High-dimensional multivariate forecasting with low-rank Gaussian copula processes. In *Advances in Neural Information Processing Systems*, Vol. 32.

[204] David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. 2020. DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting* 36, 3 (2020), 1181–1191.

[205] Nikolay Savinov, Junyoung Chung, Mikolaj Binkowski, Erich Elsen, and Aaron van den Oord. 2021. Step-unrolled denoising autoencoders for text generation. In *International Conference on Learning Representations*.

[206] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. 2008. The graph neural network model. *IEEE Transactions on Neural Networks* 20, 1 (2008), 61–80.

[207] Thomas Schlegl, Philipp Seeböck, Sebastian M. Waldstein, Ursula Schmidt-Erfurth, and Georg Langs. 2017. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In *International Conference on Information Processing in Medical Imaging*. 146–157.

[208] Patrick Schramowski, Manuel Brack, Björn Deiseroth, and Kristian Kersting. 2023. Safe latent diffusion: Mitigating inappropriate degeneration in diffusion models. In *IEEE Conference on Computer Vision and Pattern Recognition*. 22522–22531.

[209] Vikash Sehwag, Caner Hazirbas, Albert Gordo, Firat Ozgenel, and Cristian Canton. 2022. Generating high fidelity data from low-density regions using diffusion models. In *IEEE Conference on Computer Vision and Pattern Recognition*. 11492–11501.

[210] Vikash Sehwag, Saeed Mahloujifar, Tinashe Handina, Sihui Dai, Chong Xiang, Mung Chiang, and Prateek Mittal. 2021. Robust learning meets generative models: Can proxy distributions improve adversarial robustness?. In *International Conference on Learning Representations*.

[211] Zhuchen Shao, Liuxi Dai, Yifeng Wang, Haoqian Wang, and Yongbing Zhang. 2023. AugDiff: Diffusion based feature augmentation for multiple instance learning in whole slide image. *arXiv preprint arXiv:2303.06371* (2023).

[212] Chence Shi, Shitong Luo, Minkai Xu, and Jian Tang. 2021. Learning gradient fields for molecular conformation generation. In *International Conference on Machine Learning*. 9558–9568.

[213] Chence Shi, Minkai Xu, Zhaocheng Zhu, Weinan Zhang, Ming Zhang, and Jian Tang. 2020. GraphAF: A flow-based autoregressive model for molecular graph generation. *arXiv preprint arXiv:2001.09382* (2020).

[214] Yuyang Shi, Valentin De Bortoli, George Deligiannidis, and Arnaud Doucet. 2022. Conditional simulation using diffusion Schrödinger bridges. *arXiv preprint arXiv:2202.13460* (2022).

[215] Ikaro Silva, George Moody, Daniel J. Scott, Leo A. Celi, and Roger G. Mark. 2012. Predicting in-hospital mortality of ICU patients: The physionet/computing in cardiology challenge 2012. In *2012 Computing in Cardiology*. 245–248.

[216] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, and Oran Gafni. 2022. Make-a-Video: Text-to-video generation without text-video data. In *International Conference on Learning Representations*.

[217] John Skilling. 1989. The eigenvalues of mega-dimensional matrices. In *Maximum Entropy and Bayesian Methods*. 455–466.

[218] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. 2015. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*. 2256–2265.

[219] Xizewen Han, Huangjie Zheng, and Mingyuan Zhou. 2022. Card: Classification and regression diffusion models. *Advances in Neural Information Processing Systems* 35, (2022), 18100–18115.

[220] Jiaming Song, Chenlin Meng, and Stefano Ermon. 2020. Denoising diffusion implicit models. In *International Conference on Learning Representations*.

[221] Ki-Ung Song. 2022. Applying regularized Schrödinger-Bridge-based stochastic process in generative modeling. *arXiv preprint arXiv:2208.07131* (2022).

[222] Yang Song, Conor Durkan, Iain Murray, and Stefano Ermon. 2021. Maximum likelihood training of score-based diffusion models. In *Advances in Neural Information Processing Systems*, Vol. 34. 1415–1428.

[223] Yang Song and Stefano Ermon. 2019. Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems*, Vol. 32.

[224] Yang Song and Stefano Ermon. 2020. Improved techniques for training score-based generative models. In *Advances in Neural Information Processing Systems*, Vol. 33. 12438–12448.

[225] Yang Song, Sahaj Garg, Jiaxin Shi, and Stefano Ermon. 2019. Sliced score matching: A scalable approach to density and score estimation. In *The Conference on Uncertainty in Artificial Intelligence*. 204.
[226] Yang Song and Diederik P. Kingma. 2021. How to train your energy-based models. *arXiv preprint arXiv:2101.03288* (2021).
[227] Yang Song, Liyue Shen, Lei Xing, and Stefano Ermon. 2021. Solving inverse problems in medical imaging with score-based generative models. In *International Conference on Learning Representations*.
[228] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. 2020. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*.
[229] James C. Spall. 2012. Stochastic optimization. In *Handbook of Computational Statistics*. 173–201.
[230] Xuan Su, Jiaming Song, Chenlin Meng, and Stefano Ermon. 2022. Dual diffusion implicit bridges for image-to-image translation. In *International Conference on Learning Representations*.
[231] Jaesung Tae, Hyeongju Kim, and Taesu Kim. 2021. EdiTTS: Score-based editing for controllable text-to-speech. *arXiv preprint arXiv:2110.02584* (2021).
[232] Huachun Tan, Guangdong Feng, Jianshuai Feng, Wuhong Wang, Yu-Jin Zhang, and Feng Li. 2013. A tensor-based method for missing traffic data completion. *Transportation Research Part C: Emerging Technologies* 28 (2013), 15–27.
[233] Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon. 2021. CSDI: Conditional score-based diffusion models for probabilistic time series imputation. In *Advances in Neural Information Processing Systems*, Vol. 34. 24804–24816.
[234] Shantanu Thakoor, Corentin Tallec, Mohammad Gheshlaghi Azar, Rémi Munos, Petar Veličković, and Michal Valko. 2021. Bootstrapped representation learning on graphs. *arXiv preprint arXiv:2102.06514* (2021).
[235] Lucas Theis, Aäron van den Oord, and Matthias Bethge. 2015. A note on the evaluation of generative models. *arXiv preprint arXiv:1511.01844* (2015).
[236] Brandon Trabucco, Kyle Doherty, Max Gurinas, and Ruslan Salakhutdinov. 2023. Effective data augmentation with diffusion models. In *International Conference on Learning Representations*.
[237] Soobin Um and Jong Chul Ye. 2023. Don't play favorites: Minority guidance for diffusion models. *arXiv preprint arXiv:2301.12334* (2023).
[238] Arash Vahdat, Karsten Kreis, and Jan Kautz. 2021. Score-based generative modeling in latent space. In *Advances in Neural Information Processing Systems*, Vol. 34. 11287–11302.
[239] Aäron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. 2016. Pixel recurrent neural networks. In *International Conference on Machine Learning*. 1747–1756.
[240] Pascal Vincent. 2011. A connection between score matching and denoising autoencoders. *Neural Computation* 23, 7 (2011), 1661–1674.
[241] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. 2008. Extracting and composing robust features with denoising autoencoders. In *International Conference on Machine Learning*. 1096–1103.
[242] Jinyi Wang, Zhaoyang Lyu, Dahua Lin, Bo Dai, and Hongfei Fu. 2022. Guided diffusion model for adversarial purification. *arXiv preprint arXiv:2205.14969* (2022).
[243] Yufei Wang, Jiayi Zheng, Can Xu, Xiubo Geng, Tao Shen, Chongyang Tao, and Daxin Jiang. 2022. KnowDA: All-in-one knowledge mixture model for data augmentation in few-shot NLP. *arXiv preprint arXiv:2206.10265* (2022).
[244] Zekai Wang, Tianyu Pang, Chao Du, Min Lin, Weiwei Liu, and Shuicheng Yan. 2023. Better diffusion models further improve adversarial training. *arXiv preprint arXiv:2302.04638* (2023).
[245] Zhendong Wang, Huangjie Zheng, Pengcheng He, Weizhu Chen, and Mingyuan Zhou. 2022. Diffusion-GAN: Training GANs with diffusion. *arXiv preprint arXiv:2206.02262* (2022).
[246] Daniel Watson, William Chan, Jonathan Ho, and Mohammad Norouzi. 2021. Learning fast samplers for diffusion models by differentiating through sample quality. In *International Conference on Learning Representations*.
[247] Daniel Watson, Jonathan Ho, Mohammad Norouzi, and William Chan. 2021. Learning to efficiently sample from diffusion probabilistic models. *arXiv preprint arXiv:2106.03802* (2021).
[248] Jay Whang, Mauricio Delbracio, Hossein Talebi, Chitwan Saharia, Alexandros G. Dimakis, and Peyman Milanfar. 2022. Deblurring via stochastic refinement. In *IEEE Conference on Computer Vision and Pattern Recognition*. 16293–16303.
[249] Hao Wu, Jonas Köhler, and Frank Noe. 2020. Stochastic normalizing flows. In *Advances in Neural Information Processing Systems*, Vol. 33. 5933–5944.
[250] Jay Zhangjie Wu, Yixiao Ge, Xintao Wang, Weixian Lei, Yuchao Gu, Wynne Hsu, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. 2022. Tune-a-video: One-shot tuning of image diffusion models for text-to-video generation. *arXiv preprint arXiv:2212.11565* (2022).
[251] Quanlin Wu, Hang Ye, and Yuntian Gu. 2022. Guided diffusion model for adversarial purification from random noise. *arXiv preprint arXiv:2206.10875* (2022).

[252] Shoule Wu and Ziqiang Shi. 2022. Itôwave: Itô stochastic differential equation is all you need for wave generation. In *IEEE International Conference on Acoustics, Speech and Signal Processing*. 8422–8426.
[253] Shiwen Wu, Fei Sun, Wentao Zhang, Xu Xie, and Bin Cui. 2020. Graph neural networks in recommender systems: A survey. *Comput. Surveys* (2020).
[254] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S. Yu Philip. 2020. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems* 32, 1 (2020), 4–24.
[255] Julian Wyatt, Adam Leach, Sebastian M. Schmon, and Chris G. Willcocks. 2022. AnoDDPM: Anomaly detection with denoising diffusion probabilistic models using simplex noise. In *IEEE Conference on Computer Vision and Pattern Recognition*. 650–656.
[256] Zhisheng Xiao, Karsten Kreis, and Arash Vahdat. 2021. Tackling the generative learning trilemma with denoising diffusion GANs. *arXiv preprint arXiv:2112.07804* (2021).
[257] Pan Xie, Qipeng Zhang, Zexian Li, Hao Tang, Yao Du, and Xiaohui Hu. 2022. Vector quantized diffusion model with CodeUnet for text-to-sign pose sequences generation. *arXiv preprint arXiv:2208.09141* (2022).
[258] Tian Xie, Xiang Fu, Octavian-Eugen Ganea, Regina Barzilay, and Tommi S. Jaakkola. 2021. Crystal diffusion variational autoencoder for periodic material generation. In *International Conference on Learning Representations*.
[259] Yutong Xie and Quanzheng Li. 2022. Measurement-conditioned denoising diffusion probabilistic model for undersampled medical image reconstruction. *arXiv preprint arXiv:2203.03623* (2022).
[260] Minghao Xu, Hang Wang, Bingbing Ni, Hongyu Guo, and Jian Tang. 2021. Self-supervised graph-level representation learning with local and global structure. In *International Conference on Machine Learning*. 11548–11558.
[261] Minkai Xu, Lantao Yu, Yang Song, Chence Shi, Stefano Ermon, and Jian Tang. 2021. GeoDiff: A geometric diffusion model for molecular conformation generation. In *International Conference on Learning Representations*.
[262] Tijin Yan, Hongwei Zhang, Tong Zhou, Yufeng Zhan, and Yuanqing Xia. 2021. ScoreGrad: Multivariate probabilistic time series forecasting with continuous energy-based generative models. *arXiv preprint arXiv:2106.10121* (2021).
[263] Dongchao Yang, Jianwei Yu, Helin Wang, Wen Wang, Chao Weng, Yuexian Zou, and Dong Yu. 2022. Diffsound: Discrete diffusion model for text-to-sound generation. *arXiv preprint arXiv:2207.09983* (2022).
[264] Jie Yang, Ruijie Xu, Zhiquan Qi, and Yong Shi. 2021. Visual anomaly detection for images: A survey. *arXiv preprint arXiv:2109.13157* (2021).
[265] Kevin Yang and Dan Klein. 2021. FUDGE: Controlled text generation with future discriminators. *arXiv preprint arXiv:2104.05218* (2021).
[266] Ling Yang and Shenda Hong. 2022. Omni-granular ego-semantic propagation for self-supervised graph representation learning. *arXiv preprint arXiv:2205.15746* (2022).
[267] Ling Yang and Shenda Hong. 2022. Unsupervised time-series representation learning with iterative bilinear temporal-spectral fusion. In *International Conference on Machine Learning*. 25038–25054.
[268] Ling Yang, Zhilin Huang, Yang Song, Shenda Hong, Guohao Li, Wentao Zhang, Bin Cui, Bernard Ghanem, and Ming-Hsuan Yang. 2022. Diffusion-based scene graph to image generation with masked contrastive pre-training. *arXiv preprint arXiv:2211.11138* (2022).
[269] Ling Yang, Liangliang Li, Zilun Zhang, Xinyu Zhou, Erjin Zhou, and Yu Liu. 2020. DPGN: Distribution propagation graph network for few-shot learning. In *IEEE Conference on Computer Vision and Pattern Recognition*. 13390–13399.
[270] Ruihan Yang and Stephan Mandt. 2022. Lossy image compression with conditional diffusion models. *arXiv preprint arXiv:2209.06950* (2022).
[271] Ruihan Yang, Prakhar Srivastava, and Stephan Mandt. 2022. Diffusion probabilistic modeling for video generation. *arXiv preprint arXiv:2203.09481* (2022).
[272] Xiuwen Yi, Yu Zheng, Junbo Zhang, and Tianrui Li. 2016. ST-MVL: Filling missing values in geo-sensory time series data. In *International Joint Conference on Artificial Intelligence*.
[273] Jongmin Yoon, Sung Ju Hwang, and Juho Lee. 2021. Adversarial purification with score-based generative models. In *International Conference on Machine Learning*. 12062–12072.
[274] Jinsung Yoon, Daniel Jarrett, and Mihaela Van der Schaar. 2019. Time-series generative adversarial networks. In *Advances in Neural Information Processing Systems*, Vol. 32.
[275] Peiyu Yu, Sirui Xie, Xiaojian Ma, Baoxiong Jia, Bo Pang, Ruiqi Gao, Yixin Zhu, Song-Chun Zhu, and Ying Nian Wu. 2022. Latent diffusion energy-based model for interpretable text modelling. In *International Conference on Machine Learning*. 25702–25720.
[276] Sihyun Yu, Jihoon Tack, Sangwoo Mo, Hyunsu Kim, Junho Kim, Jung-Woo Ha, and Jinwoo Shin. 2022. Generating videos with dynamics-aware implicit generative adversarial networks. *arXiv preprint arXiv:2202.10571* (2022).
[277] Lvmin Zhang and Maneesh Agrawala. 2023. Adding conditional control to text-to-image diffusion models. *arXiv preprint arXiv:2302.05543* (2023).
[278] Qinsheng Zhang and Yongxin Chen. 2021. Diffusion normalizing flow. In *Advances in Neural Information Processing Systems*, Vol. 34. 16280–16291.

[279] Qinsheng Zhang and Yongxin Chen. 2022. Fast sampling of diffusion models with exponential integrator. *arXiv preprint arXiv:2204.13902* (2022).
[280] Qinsheng Zhang, Molei Tao, and Yongxin Chen. 2022. gDDIM: Generalized denoising diffusion implicit models. *arXiv preprint arXiv:2206.05564* (2022).
[281] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. OPT: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068* (2022).
[282] Wenrui Zhang, Ling Yang, Shijia Geng, and Shenda Hong. 2022. Cross reconstruction transformer for self-supervised time series representation learning. *arXiv preprint arXiv:2205.09928* (2022).
[283] Min Zhao, Fan Bao, Chongxuan Li, and Jun Zhu. 2022. EGSDE: Unpaired image-to-image translation via energy-guided stochastic differential equations. *arXiv preprint arXiv:2207.06635* (2022).
[284] Yue Zhao, Zain Nasrullah, and Zheng Li. 2019. PyOD: A Python toolbox for scalable outlier detection. *Journal of Machine Learning Research* 20 (2019), 1–7.
[285] Huangjie Zheng, Pengcheng He, Weizhu Chen, and Mingyuan Zhou. 2022. Truncated diffusion probabilistic models. *arXiv preprint arXiv:2202.09671* (2022).
[286] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. 2020. Graph neural networks: A review of methods and applications. *AI Open* 1 (2020), 57–81.
[287] Linqi Zhou, Yilun Du, and Jiajun Wu. 2021. 3D shape generation and completion through point-voxel diffusion. In *International Conference on Computer Vision*. 5826–5835.
[288] Ye Zhu, Yu Wu, Kyle Olszewski, Jian Ren, Sergey Tulyakov, and Yan Yan. 2022. Discrete contrastive diffusion for cross-modal and conditional generation. *arXiv preprint arXiv:2206.07771* (2022).
[289] Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. 2020. Deep graph contrastive representation learning. *arXiv preprint arXiv:2006.04131* (2020).
[290] Roland S. Zimmermann, Lukas Schott, Yang Song, Benjamin A. Dunn, and David A. Klindt. 2021. Score-based generative classifiers. *arXiv preprint arXiv:2110.00473* (2021).

Received 16 October 2022; revised 22 September 2023; accepted 26 September 2023