A Survey on Generative Diffusion Models

Hanqun Cao, Cheng Tan, Zhangyang Gao, Yilun Xu, Guangyong Chen, Pheng-Ann Heng, *Senior Member, IEEE*, and Stan Z. Li, *Fellow, IEEE*

Abstract—Deep generative models are a prominent approach for data generation, and have been used to produce high quality samples in various domains. Diffusion models, an emerging class of deep generative models, have attracted considerable attention owing to their exceptional generative quality. Despite this, they have certain limitations, including a time-consuming iterative generation process and confinement to high-dimensional Euclidean space. This survey presents a plethora of advanced techniques aimed at enhancing diffusion models, including sampling acceleration and the design of new diffusion processes. In addition, we delve into strategies for implementing diffusion models in manifold and discrete spaces, maximum likelihood training for diffusion models, and methods for creating bridges between two arbitrary distributions. The innovations we discuss represent the efforts for improving the functionality and efficiency of diffusion models in recent years. To examine the efficacy of existing models, a benchmark of FID score, IS, and NLL is presented in a specific NFE. Furthermore, diffusion models are found to be useful in various domains such as computer vision, audio, sequence modeling, and AI for science. The paper concludes with a summary of this field, along with existing limitations and future directions. Summation of existing well-classified methods is in our Github: https://github.com/chq1155/A-Survey-on-Generative-Diffusion-Model.

Index Terms—Diffusion Model, advanced improvement on diffusion, diffusion application.

1 Introduction

How can we enable machines to possess human-like imagination? Deep generative models, including Variational Autoencoders (VAEs) [1–3], Energy-Based Models (EBMs) [4–6], Generative Adversarial Networks (GANs) [7, 8], normalizing flow [9–12], and diffusion models [13–15], have demonstrated remarkable potential in generating realistic samples. In this survey, our primary focus is on diffusion models, which represent the most advanced approach in this field. These models overcome the challenges associated with aligning posterior distributions in VAEs, dealing with the unstable adversarial objective in GANs, the computationally expensive training-time Markov Chain Monte Carlo (MCMC) methods in EBMs, and imposing network constraints as in normalizing flows. Consequently, diffusion models have garnered significant attention in various domains, such as computer vision [16–25], sequence modeling [26–29], audio processing [30–34], and AI for science [35–39]. Despite the considerable interest, a comprehensive taxonomy and analysis of the research progress on diffusion models are still lacking.

Diffusion models involve two interconnected processes: a predefined forward process that transforms the data distribution into a simpler prior distribution, such as a Gaussian, and a corresponding reverse process that utilizes a trained neural network to progressively undo the forward process through the simulation of Ordinary or Stochastic Differential Equations (ODE/SDE) [15, 40]. The forward process resembles a straightforward Brownian motion with time-varying coefficients [40]. The neural network is trained to estimate the score function using the denoising score-matching objective [41]. As a result, diffusion models offer a more stable training objective compared to the adversarial objective in GANs and exhibit superior generation quality in comparison to VAEs, EBMs, and normalizing flows [15, 42]. However, it is important to note that diffusion models inherently suffer from a more time-consuming sampling process compared to GANs or VAEs. This is due to the iterative transformation of the prior distribution into a complex data distribution through ODE/SDE/Markov processes, which necessitates a large number of function evaluations during the reverse process. Additional challenges include the instability of the reverse process, the high computational cost and restriction of training in the high-dimensional Euclidean space, and the difficulties associated with likelihood optimization. In response to these challenges, researchers have proposed various solutions. For instance, to accelerate the sampling process, advanced ODE/SDE solvers have been proposed [43–45], and model distillation strategies have been employed [46**?**]. Furthermore, new forward processes have been introduced to stabilize sampling [47–49] or achieve dimensionality reduction [50, 51]. Additionally, a recent line of work aims to utilize diffusion models to bridge arbitrary distributions, enabling tasks such as image-to-image translation and modeling of biological evolution [52, 53]. To provide clarity, we classify these improvements to diffusion models into four main categories: (1) **Sampling Acceleration**, (2) **Diffusion Process Design**, (3)

- *H. Cao is with the Department of Math, The Chinese University of Hong Kong, Hong Kong, China, also with Zhejiang Lab, Hangzhou, China, and the AI Lab, School of Engineering, Westlake University, Hangzhou, China. Email: 1155141481@link.cuhk.edu.hk.*
- *C. Tan and Z. Gao are with Zhejiang University, Hangzhou, China, and also with the AI Lab, School of Engineering, Westlake University, Hangzhou, China. Email: tancheng, gaozhangyang@westlake.edu.cn.*
- *Y. Xu is with Massachusetts Institute of Technology, Cambridge, Massachusetts, U.S. Email: ylxu@mit.edu.*
- *G. Chen is with Zhejiang Lab, Hangzhou, China. Email: gychen@zhejianglab.com.*
- *P.-A. Heng is with the Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong, China.*
- *Stan Z. Li is with the AI Lab, School of Engineering, Westlake University, Hangzhou, China. Email: Stan.ZQ.Li@westlake.edu.cn.*
- *H. Cao, C. Tan, and Z. Gao contributed equally to this work.*

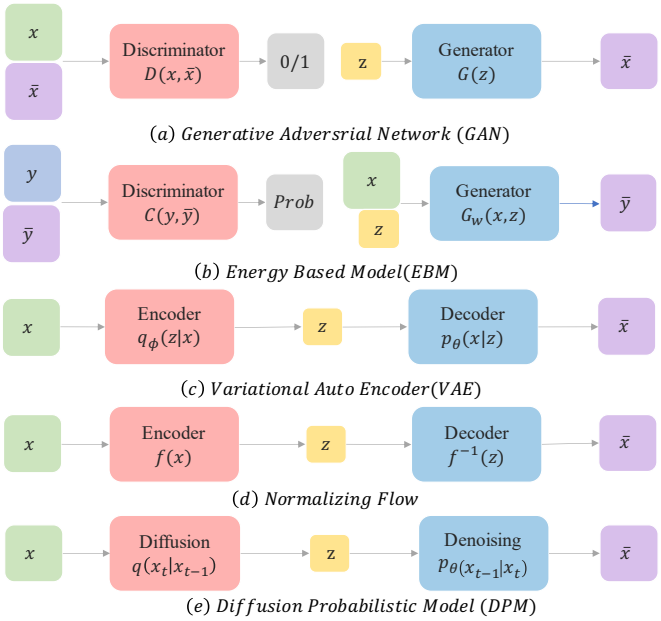


Fig. 1. The Generative Models Pipeline includes different methods such as GAN, EBM, VAE, NF, and Diffusion Model. **(a)** GAN [54] uses adversarial training to generate realistic samples. **(b)** EBM [55] matches conditions and samples using a generative discriminator. **(c)** VAE [56] projects the prior into a low-dimensional latent space for the decoder to sample from. **(d)** NF [57] uses reversible flow functions to transform inputs into latent variables and then back into samples. **(e)** Diffusion model adds noise to the original data and gradually converts it into a known noise distribution.

Likelihood Optimization, and (4) **Bridging Distributions**.

Hence, based on the wide range of applications along with multi-perspective thinking on algorithm improvement, we target to provide a detailed survey about current aspects of diffusion models. By classifying enhanced algorithms and applications in other domains, the core contributions of this review are as follows:

- Summarize essence mathematical formulation and derivation of fundamental algorithms in the field of diffusion model, including taking advantage of training strategy, and sampling algorithm.
- Present comprehensive and up-to-date classification of improved diffusion algorithms and divide them into four proposes, which are speed-up improvement, structure diversification, likelihood optimization, and dimension reduction.
- Provide extensive statements about the application of diffusion models on computer vision, natural language processing, bioinformatics, and speech processing which include domain-specialized problem formulation, related datasets, evaluation metrics, and downstream tasks, along with sets of benchmarks.
- Clarify current limitations of models and possible further-proof directions concerning the field of diffusion models.

2 Problem Statement

2.1 Notions and Definitions

2.1.1 Time and States

In the context of diffusion models, the entire process is carried out within a continuous or discrete timeline, where time is denoted as $t \in [0,1]$ or t_0^T, respectively. The states in this framework represent data distributions that describe the progression of diffusion models. The noise is incrementally introduced into the initial distribution, referred to as the starting state x_0, which is sampled from the data distribution p_0. Through a series of noise injections, the distribution gradually converges towards a known noise distribution $p_T(p_1)$, typically a Gaussian distribution, referred to as the prior state $x_T(x_1)$. The states that lie between the starting state and the prior state are referred to as intermediate states x_t, each associated with a marginal distribution p_t.
By following this approach, the diffusion models enable the exploration of the data distribution's evolution over time, allowing for the generation of samples that approximate the desired prior state x_T. The progression from the starting state to the prior state occurs through a sequence of intermediate states, with each state corresponding to a specific time point in the diffusion process.

2.1.2 Forward / Reverse Process, and Transition Kernel

In diffusion models, the transformation of the starting state into the manageable noise is carried out through the forward process F. Conversely, the reverse or denoised process R operates in the opposite direction, gradually denoising the prior state back to the starting state. Both processes involve the use of transition kernels to facilitate the exchange between any two states.
In the discrete framework of vanilla Denoising Diffusion Probabilistic Models (DDPM) [14], the forward process consists of a sequence of forward transition kernels within a Markov chain. Similarly, the reverse process evolves in a similar manner, employing a series of reverse transition kernels. These transition kernels dictate the transformation between consecutive states, enabling the diffusion process to unfold step by step. The discrete framework provides a discrete-time approximation of the continuous diffusion process, allowing for practical implementation and efficient computation.

$$F(x_0, \{\sigma_i\}_{i=1}^T) = F_T(x_{T-1}, \sigma_T) \cdots \circ F_t(x_{t-1}, \sigma_t) \cdots \circ F_1(x_0, \sigma_1) \quad (1)$$

$$R(x_T, \{\sigma_i\}_{i=1}^T) = R_1(x_1, \sigma_1) \cdots \circ R_t(x_t, \sigma_t) \cdots \circ R_T(x_T, \sigma_T) \quad (2)$$

where F_t / R_t are the forward / reverse transition kernels at time t, with the intermediate state x_{t-1} / x_t and the noise scale σ_t as inputs: $x_t = F_t(x_{t-1}, \sigma_t), x_{t-1} = R_t(x_t, \sigma_t)$.
The key distinction between diffusion models and normalizing flow lies in the variable noise scale, which governs the level of randomness within the process. As the noise scale approaches zero, the diffusion process tends towards the deterministic behavior observed in normalizing flow models. In normalizing flow, the transformation from the input distribution to the target distribution is achieved through a sequence of invertible mappings. On the other hand, diffusion models introduce noise and progressively

refine the distribution over time, allowing for a controlled transition towards the target distribution.

2.1.3 From discrete to continuous

Taking the perturbation kernel to sufficiently small, the pair of discrete processes (Eq. (1) and Eq. (2)) will generalize to continuous processes. [15] showed that diffusion models with discrete Markov chains [13, 14] can be integrated into a continuous Stochastic Differential Equation (SDE) framework, where the generative process is equivalent to reversing a fixed forward diffusion process. [15] also derives a reserve ODE marginally-equivalent to the reverse SDE. The continuous process enjoys better theoretical support, and opens the door for applying existing techniques in the ODE/SDE community to diffusion models.

2.2 Background

To access the improvement of diffusion models and its applications. In this sub-section, we introduce two foundation formulations of diffusion models. For each model, we present math formulations, training procedures, and sampling algorithms.

2.2.1 Denoised Diffusion Probabilistic Models

DDPM Forward Process: Based on the framework above, DDPM chooses a sequence of noise coefficients $\beta_1, \beta_2, ..., \beta_T$ for Markov transition kernels following specific patterns. The common choices are constant schedule, linear schedule, and cosine schedule. The constant schedule maintains a fixed value throughout the diffusion process, simplifying training but potentially resulting in less diverse or refined samples. In the linear schedule, coefficients progress linearly over time, gradually transitioning from high to low noise levels for improved sample refinement. The cosine schedule introduces periodic variation using a cosine function, enabling nuanced control of noise levels at different stages. This enhances exploration and refinement of the data distribution, leading to improved sample quality. According to [14], different noise schedules have no clear effects in experiments. The DDPM forward steps are defined as:

$$F_t(x_{t-1}, \beta_t) := q(x_t|x_{t-1}) := \mathcal{N}\left(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t \mathbf{I}\right) \quad (3)$$

for $t \in \{1, \ldots, T\}$. The idea of the choice of the forward kernels originally arises from the diffusion process in thermodynamics [13, 58]. By the composition of forward transition kernels from x_0 to x_T, we have the following Forward Diffusion Process that progressively adds Gaussian noises to the data through the Markov kernel $q(x_t|x_{t-1})$:

$$F(x_0, \{\beta_i\}_{i=1}^T) := q\left(x_{1:T} \mid x_0\right) := \prod_{t=1}^{T} q\left(x_t \mid x_{t-1}\right) \quad (4)$$

DDPM Reverse Process: Given the Forward Process above, we define the Reverse Process with learnable Gaussian transitions parameterized by θ [14] as follows

$$R_t(x_t, \Sigma_\theta) := p_\theta(x_{t-1}|x_t) := \mathcal{N}\left(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)\right) \quad (5)$$

The reverse process starts at $p(x_T) = \mathcal{N}(x_T; \mathbf{0}, \mathbf{I})$ and goes through a sequence of reverse steps from x_T to x_0:

$$R(x_T, \Sigma_\theta) := p_\theta\left(x_{0:T}\right) := p\left(x_T\right) \prod_{t=1}^{T} p_\theta\left(x_{t-1} \mid x_t\right) \quad (6)$$

DDPM aims to approximate the data distribution p_0 by the model distribution $p_\theta(x_0) = \int p_\theta(x_{0:T}) dx_{1:T}$ obtained via a sequence of denoising operations in the reverse process.

Diffusion Training Objective: To approximate the data distribution by the model distribution $p_\theta(x_0)$, diffusion models is trained to minimize variational bound on the negative log-likelihood (NLL), similar to VAEs [1]:

$$\begin{aligned}
\mathbb{E}\left[-\log p_\theta\left(\mathrm{x}_0\right)\right] &\leq \mathbb{E}_q\left[-\log \frac{p_\theta\left(\mathrm{x}_{0:T}\right)}{q\left(\mathrm{x}_{1:T} \mid \mathrm{x}_0\right)}\right] \\
&= \mathbb{E}_q\left[-\log p\left(\mathrm{x}_T\right) - \sum_{t \geq 1} \log \frac{p_\theta\left(\mathrm{x}_{t-1} \mid \mathrm{x}_t\right)}{q\left(\mathrm{x}_t \mid \mathrm{x}_{t-1}\right)}\right] \\
&= \mathbb{E}_q[\underbrace{D_{\mathrm{KL}}\left(q\left(x_T \mid x_0\right) \| p\left(x_T\right)\right)}_{L_T} \\
&+ \sum_{t>1} \underbrace{D_{\mathrm{KL}}\left(q\left(x_{t-1} \mid x_t, x_0\right) \| p_\theta\left(x_{t-1} \mid x_t\right)\right)}_{L_{t-1}} \\
&\underbrace{-\log p_\theta\left(x_0 \mid x_1\right)]}_{L_0} \\
&=: L
\end{aligned} \quad (7)$$

Here we use the notation in the DDPM paper [14], where L_T is the prior loss and L_0 is the reconstruction loss. $L_{1:T-1}$ is the sum of the divergence between the posterior of the forwarding step and the corresponding reversing step. In order to minimize the negative log-likelihood, the only item we can be used to train is $L_{1:T-1}$. By parameterizing the posterior $q(x_{t-1}|x_t, x_0)$ using Baye's rule, we have:

$$q\left(x_{t-1} \mid x_t, x_0\right) = \mathcal{N}\left(x_{t-1}; \tilde{\boldsymbol{\mu}}_t\left(x_t, x_0\right), \tilde{\beta}_t I\right) \quad (8)$$

where α_t is defined as $1 - \beta_t$, $\bar{\alpha}_t$ is defined as $\prod_{k=1}^{t} \alpha_k$. Keeping above parameterization as well as reparameterizing x_t as $x_t(x_0, \sigma)$, L_{t-1} can be regarded as an expectation of ℓ_2-loss between two mean coefficients:

$$L_{t-1} = \mathbb{E}_q\left[\frac{1}{2\sigma_t^2}\left\|\tilde{\boldsymbol{\mu}}_t\left(\mathrm{x}_t, \mathrm{x}_0\right) - \boldsymbol{\mu}_\theta\left(\mathrm{x}_t, t\right)\right\|^2\right] + C \quad (9)$$

which is linked to the denoising score-matching objective discuss in the next section. Simplifying L_{t-1} by reparameterizing μ_θ w.r.t ϵ_θ, we obtain the simplified training objective named L_{simple}:

$$L_{simple} := \mathbb{E}_{x_0, \epsilon}\left[\frac{{\beta_t}^2}{2{\sigma_T}^2 \alpha_t (1-\bar{\alpha}_t)}\right]\left\|\epsilon - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon)\right\|^2 \quad (10)$$

Most diffusion models until now use the training strategy of DDPM. But there exist some exceptions. Improved DDPM [59] proposes to combine L_{simple} with other auxiliary objectives. However, L_{simple} still takes the main effect of thes joint objective. After training, the neural prediction $\boldsymbol{\epsilon}_\theta$ is used in the reverse process for ancestral sampling.

2.2.2 Score SDE Formulation

Score SDE [15] extents the discrete-time scheme in DDPM to a unified continuous-time framework based on the stochastic differential equation. It not only presents the corresponding continuous set-up of DDPM based on score SDE but also proposes a density estimation ODE framework named probability flow ODE.

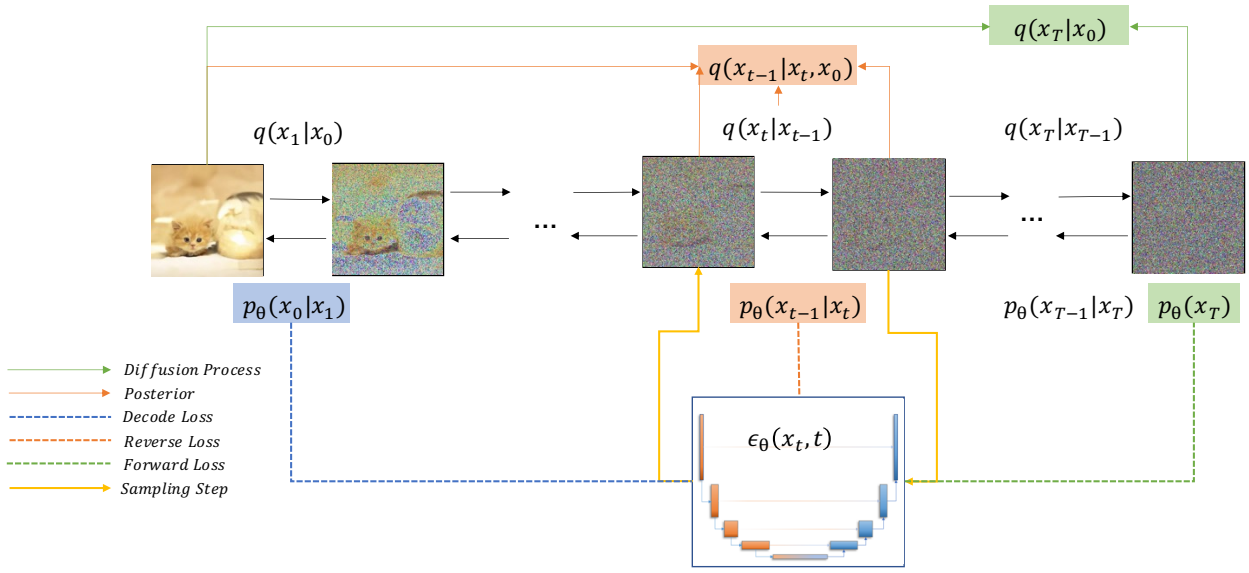


Fig. 2. **Pipeline of Denoised Diffusion Probabilistic Model.**

Forward SDE: In Song *et al.* [15], the diffusion process can be viewed as a continuous process through the len of Stochastic Differential Equation. And it is equal to the solution to Itô SDE [60], which is composed of a drift part for mean transformation and a diffusion coefficient for noise description. :

$$dx = f(x,t)dt + g(t)dw, t \in [0,T] \tag{11}$$

where w_t is the standard Wiener process/Brownian motion, $f(\cdot,t)$ is the drift coefficient of $x(t)$, and $g(\cdot)$ is the simplified version of diffusion coefficient assumed not dependent on x. We denote the marginal distribution at time t as $p_t(x)$. Consequently, p_T denotes the prior distribution which gains no information from p_0. When the coefficients are piece-wise continuous, the forward SDE equation admits a unique solution [61]. Based on the SDE frameworks, [15] showcases two kinds of forward process, termed Variation Preserving (VP) and Variation Explosion (VE) SDE, respectively:

$$\text{VP:} \quad \mathrm{d}x = -\frac{1}{2}\beta(t)x\ dt + \sqrt{\beta(t)}\mathrm{d}w$$

$$\text{VE:} \quad \mathrm{d}x = \sqrt{\frac{\mathrm{d}\left[\sigma^2(t)\right]}{\mathrm{d}t}}\ \mathrm{d}w$$

Notably, VP corresponds to the continuous extension of the DDPM framework (Sec 2.2.1), with $\beta(t)$ being the continuous-time variable of β_t.

Reversed SDE: The sampling of diffusion models is done via a corresponding reverse-time SDE of the forward process (Eq. (11)) [62]:

$$\mathrm{d}x = \left[f(x,t) - g(t)^2\nabla_x \log p_t(x)\right]\mathrm{d}\bar{t} + g(t)\mathrm{d}\overline{w}, t \in [0,T] \tag{12}$$

where $\bar{\cdot}$ denotes time traveling backward from T to 0 and $\nabla_x \log p_t(x)$ is the score function. We estimate the score of the transformed data distribution at time t, $\nabla_x \log p_t(x)$, via a neural network, $s_\theta(x,t)$. The training objective is a weighted sum of denosing score-matching objective [41]:

$$L := \mathbb{E}_t\{\lambda(t)\mathbb{E}_{x_0}\mathbb{E}_{q(x_t|x_0)}[\|s_\theta(x_t t) - \nabla_{x_t}\log p(x_t|x_0)\|_2^2]\} \tag{13}$$

where x_0 is sampled from the data distribution p_0 and $\lambda(t)$ is the positive weighting function to keep the time-dependent loss at the same magnitude [15]. $q(x_t|x_0)$ is the Gaussian transition kernel associated with the forward process in Eq. (11). For example, $q(x_t|x_0) = \mathcal{N}\left(x_t; x_0, \sigma^2(t)\mathbf{I}\right)$. One can show that the optimal solution in the denoising score-matching objective (Eq. (13)) equals the true score function $\nabla_x \log p_t(x)$ for almost all x,t. Additionally, the score function s_θ can be seen as reparameterization of the neural prediction $\boldsymbol{\epsilon}_\theta$ in the DDPM objective (Eq. (10)). [63] further shows that the score function in the forward process of diffusion models can be decomposed into three phases. When moving from the near field to the far field, the perturbed data get influenced by more modes in the data distribution.

Probability Flow ODE: probability flow ODE [15] is the continuous-time ODE that supports the deterministic process which shares the same marginal probability density with SDE. Inspired by Maoutsa *et al.* [64] and Chen *et al.* [65], any type of diffusion process can be derived into a special form of ODE. The corresponding probability flow ODE of Eq. (12) is

$$dx = \{f(x,t) - \frac{1}{2}g(t)^2\nabla_x \log p_t(x)\}dt \tag{14}$$

In contrast to SDE, probability flow ODE can be solved with larger step sizes as they have no randomness. Due to the advantages of ODE, several works such as PNDMs [66] and DPM-Solver [67] obtain faster sampling speed based on acceleration techniques of ODE.

2.3 Conditional Diffusion Models

Diffusion models are versatile in that they not only enable the generation of data samples from an unconditional

distribution p_0, but they can also produce samples from a conditional distribution $p_0(x|c)$ when a condition c is provided. This condition, c, could be a class label or text associated with the data x (for instance, as seen in Stable Diffusion [51]). During training, the score network $s_\theta(x, t, c)$ incorporates this condition as an input. There are also sampling algorithms tailored for conditional generation, such as classifier-free guidance [68] or classifier guidance [42] The following sections outlines several typical scenarios for conditional generation.

Labeled Condition Sampling with labeled conditions provides gradient guidance in each sampling step. Usually, an additional classifier with UNet Encoder architecture for generating condition gradients for specific labels is needed. The labels can be text or categorical label [42, 43, 69–71], binary label [72, 73], or extracted features [17, 74, 75]. It is firstly presented by [42], and current conditional sampling methods are similar in theory.

Unlabeled Condition Except for label-guidance sampling, unlabeled condition sampling only takes self-information as guidance. Conducting in a self-supervised manner [76, 77], it is often applied in denoising [78], paint-to-image [79], and inpainting [28] tasks.

3 Algorithm Improvement

Despite the impressive generation quality of diffusion models across various data modalities, there are substantial areas for improvement to enhance their deployment in real-world applications. For instance, diffusion models require a slow iterative sampling process in contrast with other generative models such as GANs and VAEs, and their forward process is directly defined in the high-dimensional pixel space. In this section, we outline four recent developments geared towards algorithmic improments on diffusion models: (1) Sampling Acceleration techniques (Section 3.1), aiming to expedite the vanilla ODE/SDE simulation of diffusion models; (2) New Forward Processes (Section 3.2), proposing improved versions of the original Brownian motion in pixel space to facilitate learning; (3) Likelihood Optimization techniques (Section 3.3), focusing on enhancing the diffusion ODE likelihood; (4) Bridging Distribution techniques (Section 3.4), employing concepts from diffusion models to bridge two distinct distributions.

3.1 Sampling Acceleration

Although diffusion models enjoy high-fidelity generation, the low sampling speed restrict its practical utility. To improve the sampling speed, advanced techniques can be divided into four categories, including distillations, training schedule improvement, training-free acceleration, and combining diffusion models with faster generative models. We give a brief overview of these methods in this section.

3.1.1 Knowledge Distillation

Knowledge distillation, a method for deriving efficient smaller networks by transferring "knowledge" from larger, complex teacher models to simpler student models, is an emerging trend [130, 131]. Distillation techniques in diffusion models aim to generate samples in fewer steps or smaller networks. Similar to traditional distillation methods for large pre-trained models, diffusion model distillation is based on the idea of alignment, which minimizes the differences between generated samples and corresponding original samples. Generally, the distillation processes in diffusion models are regarded as modifying trajectories (optimizing transportation costs) among different distributions. The optimal mappings in the fields bring shorter and more efficient paths, leading to less generation costs and controllable generation.

ODE Trajectory Keeping the "knowledge" from teacher models to student models with ODE formulation is equivalent to mapping prior distribution to targeted distribution with more efficient paths along the distribution field. [46] first applies this principle to enhance diffusion models by progressively distill the sampling trajectories. It distills the trajectories by straightening latent mappings in every two sampling steps (the acceleration rate is 2). To extend the straightening effects, TRACT [82], denoising student [80], and consistency models [132] improves the acceleration rate into 64 and 1024, respectively, directly estimating clean data from noisy samples at time T. Besides, RFCD [81] aligns sample features in training to enhance student model performance.

Additionally, the optimal trajectories can be obtained according to optimal transport [133]. By reducing the transportation cost among distributions as flow matching, Re-Flow [83] and [87] achieve one-step generation. DSNO [84] propose a neural operator for directly modeling the temporal paths. Consistency model [132], SFT-PG [85] and MMD-DDM [86] explore the optimized sampling trajectories by LPIPS, IPA and MMD, respectively.

SDE Trajectory It is still a challenging problem to distilling stochastic trajectories. Few works are proposed (referred to Section 3.4).

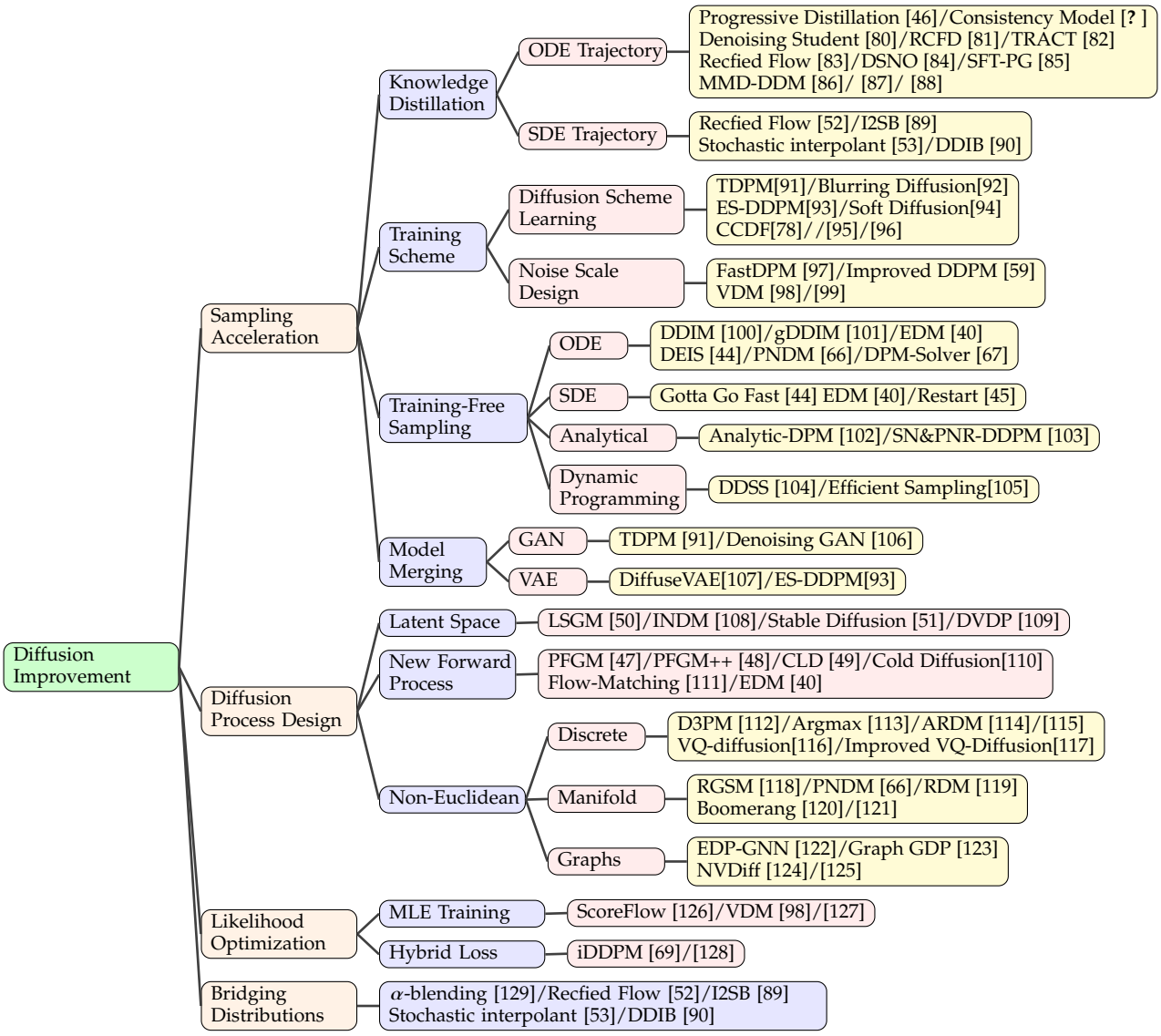
3.1.2 Training Schedule

Improving the training schedule means modifying traditional training settings, such as diffusion schemes and noise schemes, which are independent of sampling. Recent studies have shown the key factors in training schemes influencing learning patterns and models' performance. In this sub-section, we divide the training enhancement into two categories: diffusion scheme learning, and noise scale designing.

Diffusion Scheme Learning The forward diffusion process in diffusion models shares similarities with Variational Autoencoder (VAE) as it also projects data into latent spaces, where the diffusion pattern plays a crucial role in reverse decoding. However, diffusion models are more complex than VAEs as they encode data onto latent spaces with the same dimension, enabling higher expressiveness. Effective reverse decoding methods in diffusion models can be categorized into two approaches: encoding degree optimization and projecting approaches.

For encoding degree optimization, approaches such as CCDF [78] and Franzese et al. [95] establish optimization problems to minimize the Evidence Lower Bound (ELBO) by treating the number of diffusion steps as a variable [134, 135]. Truncation is another approach that balances generation speed and sample fidelity by sampling from less

TABLE 1
Classification of Improved Diffusion Techniques



diffused data in a one-step manner. TDPM [91] and ES DDPM [93] utilize truncation by sampling from implicitly learned generative distributions using GAN and CT [136].

$$\tilde{x} := R_{t^*0}(F_{0t^*}(x_0, \sigma_{0t^*}), \sigma_{t^*0}), \quad t^* \in [0, T] \tag{15}$$

On the other hand, projecting approaches focus on exploring the diversity of diffusion kernels. Works such as Soft diffusion [94] and blurring [92] diffusion models demonstrate that linear corruptions, such as blurring and masks, can serve as transition kernels.

Noise Scale Designing

In the traditional diffusion process, each transition step is determined by the injected noise, which is equivalent to a random walk on the forward and reversed trajectories. Therefore, designing the noise scale has the potential for reasonable generation and fast convergence. Unlike traditional Denoising Diffusion Probabilistic Models (DDPM), existing methods treat the noise scale as a learnable parameter throughout the entire process.

$$\mathrm{SNR(t)} := \alpha_t^2/\beta_t^2 = exp(\gamma_{\boldsymbol{\eta}}(t)), \quad \sigma_t^2 = \mathrm{sigmoid}\left(\gamma_{\boldsymbol{\eta}}(t)\right) \tag{16}$$

$$\begin{aligned} \mathcal{L}_T(\mathbf{x}) &= \frac{T}{2}\mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(0,\mathbf{I})}\left[(\mathrm{SNR}(s) - \mathrm{SNR}(t))\left\|\mathbf{x} - \hat{\mathbf{x}}_{\boldsymbol{\theta}}\left(\mathbf{z}_t; t\right)\right\|_2^2\right] \\ \mathcal{L}_\infty(\mathbf{x}) &= \frac{1}{2}\mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(0,\mathbf{I})}\int_{\mathrm{SNR}_{\min}}^{\mathrm{SNR}_{\max}}\left\|\mathbf{x} - \tilde{\mathbf{x}}_{\boldsymbol{\theta}}\left(\mathbf{z}_v, v\right)\right\|_2^2 dv \end{aligned} \tag{17}$$

Among the forward noise design methods, VDM [98] parameterizes the noise scale as a signal-to-noise ratio to connect the noise scale with training loss and model types. FastDPM [97] obtains the forward noise from discrete-time variables or a variance scalar, linking noise design to ELBO optimization. Regarding reverse noise design methods, improved DDPM [59] implicitly learns the reverse noise scale by training a hybrid loss that includes L_{simple} and L_{vlb}. Additionally, San Roman *et al.* employ a noise prediction network to directly update the reverse noise scale before conducting ancestral sampling in each step.

3.1.3 Training-Free Sampling

Training-free methods aims to leverage more advanced sampler to accelerate the sampling process of pre-trained diffusion models, eliminating the need for model re-training. This subsection categorizes these methods into several aspects: acceleration of the diffusion ODE and SDE samplers, analytical method and dynamic programming.

ODE Acceleration

[15] shows that the stochastic sampling process in DDPM has a marginally-equivalent probability ODE, which defines deterministic sampling trajectories from prior to data distribution. Given that ODE samplers have proven to generate less discretization error than their stochastic counterparts [15, 45], the bulk of preceding work on sampling acceleration has been ODE-centric. For example, a commonly used sampler DDIM [100] can be regarded as probability flow ODE [15]:

$$\mathrm{d}\bar{x}(t) = \epsilon_\theta^{(t)}\left(\frac{\bar{x}(t)}{\sqrt{\sigma^2+1}}\right)\mathrm{d}\sigma(t) \tag{18}$$

where σ_t is parameterized by $\sqrt{1-\alpha_t}/\sqrt{\alpha_t}$, and $\bar{x}$ is parameterized as $x/\sqrt{\alpha_t}$. Later works [44, 67] interpret DDIM as a product of applying an exponential integrator on the ODE of Variance Perserving (VP) diffusion [15]. Furthermore, Furthermore, advanced ODE solvers have been utilized in methods such as PNDM [66], EDM [40], DEIS [44], gDDIM [101], and DPM-Solver [67]. For example, EDM employs Heun's 2nd order ODE solvers, and DEIS/DPM-solver improves upon DDIM by numerically approximate the score functions within each discretized time interval. These methods significantly accelerate the sampling speed (diminishing the number of function evaluations, or NFE) compared to the original DDPM sampler, while still yielding high-quality samples.

SDE Acceleration

While ODE-based samplers tend to be faster, their performance eventually hits a ceiling. Conversely, SDE-based samplers, though more time-consuming, yield superior sample quality. Several works have focused on accelerating stochastic samplers' speed. For instance, Gotta Go Fast [137] leverages adaptive step size to accelerate SDE sampling, and EDM [40] combines higher-order ODE with Langevin-dynamics-like noise addition and removal, demonstrating that their proposed stochastic sampler significantly outperforms the ODE sampler on ImageNet-64. A recent work [45] reveals that although ODE-samplers involve smaller discretization errors, the stochasticity in SDE helps to contract accumulated errors. This leads to the the Restart Sampling algorithm [45], which blends the best aspects of both worlds. The sampling method alternates between adding significant noise during additional forward steps and strictly following a backward ODE, surpassing previous SDE and ODE samplers on standard benchmarks and the Stable Diffusion model [51], both in terms of speed and accuracy.

Analytical Method

Existing training-free sampling methods take reverse covariance scales as a hand-crafted sequence of noises without considering them dynamically. Starting from KL-divergence optimization, analytical methods set the reverse mean and covariance using the Monte Carlo method. Analytic-DPM [102] and extended Analytic-DPM [103] jointly propose optimal reverse solutions under correction for each state. Analytical methods enjoy a theoretical guarantee for the approximation error, but they are limited in particular distributions due to the pre-assumptions.

Dynamic Programming Adjustment

Dynamic programming (DP) achieves the traversal of all choices to find the optimized solution in a reduced time by memorization technique [138]. Assuming that each path from one state to another state shares the same KL divergence with others, dynamic programming algorithms explore the optimal traversal along the trajectory. Current DP-based methods [105, 139] take $O(T^2)$ of computational cost via optimizing the sum of ELBO losses.

3.1.4 Merging Diffusion and Other Generative Models

diffusion models can be integrated with other generative models like GANs or VAEs to expedite the sampling process. For instance, one could directly predict the clean data x_0 with a VAE [107] / GAN [106] from a noisy samples at intermediate time of diffusion sampling process. Another line of works employs a VAE [93] / GAN [91] to generate samples at intermediate diffusion time steps, then applies diffusion models to denoise these samples until time $t = 0$, thus allowing faster time traversing.

3.2 Diffusion Process Design

The original forward process in diffusion models is commonly perceived as Brownian motion within pixel space [14, 40], a potential sub-optimal choice for generative modeling. Researchers have been focusing on designing new diffusion processes that make the associated backward processes easier for neural networks to learn, more straight, or more robust. This endeavor has led to two main lines of work: the first aims to create latent spaces specifically for diffusion models (Section 3.2.1), and the second seeks to replace the original forward process with arguably improved ones operating in pixel space (Section 3.2.2). We also highlight diffusion processes specifically designed for non-Euclidean spaces, such as manifolds, discrete spaces, functional spaces, and graphs (Section 3.2.3).

3.2.1 Latent Space

Researchers have been delving into the training of diffusion models within a learned latent space to not only boost the expressiveness of neural networks, but also to create a more direct backward process. Notably, LSGM [50] and INDM [140] pioneer this approach by jointly training a diffusion model and a VAE or normalizing flow model, all centered around a common objective of the weighted denoising score-matching loss (L_{DSM} in Eq. (13)). The joint objective for the pair of encoder-decoder and diffusion model is as follows:

$$L \coloneqq L_{Enc}(z_0|x) + L_{Dec}(x|z_0) + L_{DSM}\left(\left(\{z_t\}_{t=0}^{T}\right)\right) \tag{19}$$

Here, z_0 represents the latent form of the original data x, while z_t is its perturbed counterpart. It is important to note that z_t is a function of the encoder, hence the L_{DSM} loss also updates the encoder's parameters. The joint objective is optimizing the ELBO or log-likelihood [50, 140]. This leads to a latent space that is simpler to learn from and to sample.

Additionally, the influential work, Stable Diffusion [51], divides the process into two distinct stages - the learning of the latent space of VAE and the training of the diffusion models with the text as conditional inputs. On a different note, DVDP [109] deconstructs the pixel space into several orthogonal components and modulates the attenuation of each component during image perturbation. This procedure can be interpreted as a dynamic form of image down-sampling and up-sampling.

3.2.2 New Forward Processes

While the adoption of latent space diffusion provides several advantages, it also introduces additional complexities to the diffusion framework and adds to the computational burden during training. Consequently, recent research has begun to explore the development of more robust and efficient generative models through the design of forward processes. Another physics-inspired generative models, PFGM [47], views the data as electric charges in an augmented space, enabling generative modeling by guiding a simple distribution along electric field lines toward the data distribution. In this model, the forward process is defined in the directions of electric field lines, and is shown to have more robust backward sampling than diffusion models. PFGM++ [48] extents PFGM with higher-dimensional augmented variables. Intriguingly, the process of interpolating between these two models uncovers a sweet spot, leading to new state-of-the-art performance in image generation. PFGM/PFGM++ also find broad applications in other field, such as antibody [141] and medical image [142] generation.

Also taking inspiration from physics, Dockhorn et al. [49] proposed a Critically-Damped Langevin Diffusion (CLD) model that augments the data with "velocity" variables. These variables interact according to Hamiltonian dynamics. Their model necessitates learning the score function of the conditional distribution of the velocity given the data. This task is simpler compared to directly learning the score functions of the data. Given the success of physics-inspired generative models such as diffusion models and PFGM, a recent work [143] presents a systematic way to convert physical processes into generative models.

Another line of works applies different corrupting process, resembling diffusion process. Cold Diffusion[110] uses arbitrary images transform, such as blurring, to construct the forward process, while [144] apply heat dissipation on the pixel space. Other works design better Gaussian perturbation kernels to improve training and sampling [40, 111].

3.2.3 Diffusion Models on non-Euclidean space

Discrete Space

Deep generative models have achieved significant advancements in fields such as natural language processing [145, 146], multi-modal learning [69, 147], and AI for science [148, 149], thanks to relevant architectures and innovative techniques. Among these accomplishments, processing discrete data – including sentences, residues, atoms, and vector-quantized data — is of critical importance. Several studies focus on applying diffusion models to such discrete spaces. We divide these works into processing text & categorical data, and vector-quantized data.

To process discrete data like text or atom type, D3PM [112] design transition kernels Q_t to define the forward process in discrete space:

$$q\left(x_t \mid x_{t-1}\right)=\operatorname{Cat}\left(x_t ; p=x_{t-1} \boldsymbol{Q}_t\right) \tag{20}$$

where Cat() denote categorical distribution. Multi-nominal diffusion [113] and ARDM [114] extended the categorical diffusion into multi-nomial data for generating language text & segmentation map and Lossless Compression.

To handle the multi-model problem such as text-to-image generation, text-to-3d generation, and text-to-image editing, vector-quantized (VQ) data is proposed to transform data into the codes. VQ data processing achieved great performance in autoregressive encoders [150]. Gu *et al.* [116] first applied diffusion techniques into VQ data, solving unidirectional bias as well as accumulation prediction error problems existing in VQ-VAE. Further text-to-image works such as Cohen *et al.* [151] & Improved VQ-Diffusion [117], text-to-pose works such as Xie *et al.* [152] & Guo *et al.* [153], and text-to-multimodal works such as Weinbach *et al.* [154] & Xu *et al.* [155] are built on this core idea. The forward process driven by the probability transition matrix Q and categorical representation vector v is defined by

$$q\left(x_t \mid x_{t-1}\right)=\boldsymbol{v}^{\top}\left(x_t\right) \boldsymbol{Q}_t \boldsymbol{v}\left(x_{t-1}\right) \tag{21}$$

Manifold

Most of the data structures in use today, such as images and videos, exist in Euclidean space. However, certain types of data in robotics [156], geoscience [157], and protein modeling [158] are defined within a Riemannian manifold [159], an environment where standard methods for Euclidean spaces may not apply. As a response, recent methodologies like RDM [119], RGSM [118], and Boomerang [120] have integrated diffusion sampling into the Riemannian manifold, thereby extending the score SDE framework [15]. Furthermore, pertinent theoretical works [66, 121] provide comprehensive support for manifold sampling.

Graph

According to [160], graph-based neural networks are becoming an increasingly popular trend due to the high expressiveness in the human pose [153], molecules [161], and proteins [39]. Many current methods apply diffusion theories to graph. In EDP-GNN [122], Pan *et al.* [125], and GraphGDP [123], graph data is processed through adjacency matrices for capturing the graph's permutation invariance. NVDiff [124] reconstructs node positions by reverse SDE.

Function

Dutordoir *et al.*, [162] proposes the first diffusion model sampling on the functional space. It captures infinite-dimensional distributions by sampling from joint posteriors.

3.3 Likelihood Optimization

Diffusion models [14] optimize the variational evidence lower bound (ELBO) due to the intractability of the log-likelihood. Nevertheless, optimizing the log-likelihood of general continuous-time diffusion models [15] remains challenging. A number of methods have been developed to enhance model likelihood. These methods build connections between denoising score-matching loss and maximum likelihood estimation (MLE) or design hybrid loss functions, which result in a better density estimation.

MLE Training of Diffusion Models
Three concurrent works—ScoreFlow [126], VDM [98], and [127]—establish a connection between the MLE training and the weighted denoising score-matching (DSM) objective in diffusion models, primarily through the use of the Girsanov theorem. For instance, ScoreFlow demonstrates that under a particular weighting scheme, the DSM objective provides an upper bound on the negative log-likelihood. This finding enables the approximate maximum likelihood training of score-based diffusion models (up to a constant that is independent of the neural network parameters).
Hybrid Loss
Rather than linking the denoising score-matching (DSM) to maximum likelihood training, some approaches propose hybrid loss designs to enhance the model likelihood. For example, Improved DDPM [59] suggests learning the variances of the reverse process through a straightforward reparameterization and a hybrid learning objective that combines the variational lower bound (L_{VLB}) and DSM:

$$L_{\text{hybrid}} = L_{\text{DSM}} + \lambda L_{\text{VLB}} \tag{22}$$

where λ is a hyper-parameter λ to balancing the two objectives. [128] shows that incorporating high-order score-matching loss helps to improve the log-likelihood.

3.4 Bridging Distributions

Diffusion models enable the transformation of a simple Gaussian distribution into more complex data distributions, but they are not capable of bridging two arbitrary distributions. This limitation is significant in applications such as image-to-image translation and cell distribution transportation. Drawing inspiration from the denoising score-matching objective and the stochastic processes inherent in diffusion models, several studies have focused on designing transportation maps between two distributions through SDE/ODE. For instance, α-blending [129] constructs a deterministic bridge via iterative blending and deblending, incorperating diffusion models as special cases when one of the end distributions is Gaussian. Rectified Flow [52] employs a similar formulation, introducing additional "reflow" to straighten the bridge, and [53] provides methods for constructing an ODE given general interpolant functions between two distributions.
Moreover, some researchers have turned to the Schrödinger Bridge for connecting two distributions [89], while others consider the Gaussian distribution as a junction and connect two diffusion ODEs [90].

4 Application

Benefiting from the powerful ability to generate realistic samples, diffusion models have been widely used in various fields such as computer vision [163], natural language processing, and bioinformatics.

4.1 Computer vision

4.1.1 Low-level vision

CMDE [18] empirically compared score-based diffusion methods in modeling conditional distributions of visual image data, introducing a multi-speed diffusion framework that outperformed vanilla conditional denoising estimator [164] in in-painting and super-resolution tasks. DDRM [165] proposed an efficient, unsupervised posterior sampling method for image restoration and demonstrated successful applications in super-resolution, deblurring, inpainting, and colorization of diffusion models. Palette [166] developed a diffusion-based framework for low-level vision tasks that demonstrated superior performance compared to GAN models. DiffC [167] proposed an unconditional generative approach that encoded and denoised corrupted pixels with a single diffusion model, showing the potential of diffusion models in lossy image compression. SRDiff [20] exploited the diffusion-based single-image super-resolution model and showed competitive results. RePaint [168] directly employed a pre-trained diffusion model as the generative prior for free-form inpainting and outperformed autoregressive and GAN methods under extreme tasks.

4.1.2 High-level vision

FSDM [21] is a few-shot generation framework that adapts quickly to various generative processes at test-time and performs well under few-shot generation with strong transfer capability. CARD [22] proposes denoising diffusion-based conditional generative models and pre-trained conditional mean estimator combinations for classification and regression. CLIP [169] and GLIDE [69] explore realistic image synthesis conditioned on the text, and DreamFusion [170] extends GLIDE's achievement into 3D space. LSGM trains a diffusion model in the latent space using a variational autoencoder framework. SegDiff [171] performs diffusion-based probabilistic encoding for image-level segmentation. Video diffusion [17] extends diffusion models in the time axis and performs video-level generation using a reconstruction-guided conditional sampling method.
VQ-Diffusion [23] shows superior performance on large datasets such as ImageNet and MSCOCO by exploring classifier-free guidance sampling for discrete diffusion models and presenting a high-quality inference strategy. DiffSCM [172] builds a deep structural model based on the generative diffusion model and achieves counterfactual estimation by inferring latent variables with deterministic forward diffusion and intervening in the backward process.

4.1.3 3D vision

[24] was an early work on diffusion-based 3D vision tasks. Motivated by the non-equilibrium thermodynamics, this work analogized points in point clouds as particles in a thermodynamic system and employed the diffusion process in point cloud generation, which achieved competitive performance. PVD [173] was a concurrent work on diffusion-based point cloud generation but performed unconditional generation without additional shape encoders, while a hybrid and point-voxel representation was employed for processing shapes. PDR [25] proposed a paradigm for diffusion-based point cloud completion that applied a diffusion model to generate a coarse completion based on the partial observation and refined the generated output by another network. To deal with point cloud denoising, [174] introduced a neural network to estimate the score of the distribution and denoised point clouds by gradient ascent.

4.1.4 Video modeling

Video diffusion [17] introduced the advances in diffusion-based generative models into the video domain. RVD [175] employed diffusion models to generate a residual to a deterministic next-frame prediction conditioned on the context vector. FDM [176] applied diffusion models to assist long video prediction and performed photo-realistic videos. MCVD [177] proposed a conditional video diffusion framework for video prediction and interpolation based on masking frames in a blockwise manner. RaMViD [178] extended image diffusion models to videos with 3D convolutional neural networks and designed a conditioning technique for video prediction, infilling, and upsampling.

4.1.5 Medical application

It is a natural choice to apply diffusion models to medical images. Score-MRI [179] proposed a diffusion-based framework to solve magnetic resonance imaging (MRI) reconstruction. [180] was a concurrent work but provided a more flexible framework that did not require a paired dataset for training. With a diffusion model trained on medical images, this work leveraged the physical measurement process and focused on sampling algorithms to create image samples that are consistent with the observed measurements and the estimated data prior. R2D2+ [181] combined diffusion-based MRI reconstruction and super-resolution into the same network for end-to-end high-quality medical image generation. [182] explored the application of the generative diffusion model to medical image segmentation and performed counterfactual diffusion.

4.2 Sequential modeling

4.2.1 Natural language processing

Benefited by the non-autoregressive mechanism of diffusion models, Diffusion-LM [26] took advantage of continuous diffusions to iteratively denoise noisy vectors into word vectors and performed controllable text generation tasks. Bit Diffusion [27] proposed a diffusion model for generating discrete data and was applied to image caption tasks.

4.2.2 Time series

To deal with time series imputation, CSDI [28] utilized score-based diffusion models conditioned on observed data. Inspired by masked language modeling, a self-supervised training procedure was developed that separates observed values into conditional information and imputation targets. SSSD [29] further introduced structured state space models to capture long-term dependencies in time series data. CSDE [222] proposed a probabilistic framework to model stochastic dynamics and introduced Markov dynamic programming and multi-conditional forward-backward losses to generate complex time series.

4.3 Audio

WaveGrad [30] and DiffWave [31] applied diffusion models to raw waveform generation and achieved excellent results. GradTTS [32] and Diff-TTS [233] generated mel features using diffusion models. DiffVC [232] addressed one-shot many-to-many voice conversion with a stochastic differential equation solver. DiffSinger [33] extended sound generation to singing voice synthesis. Diffsound [227] proposed a sound generation framework based on a discrete diffusion model to overcome unidirectional bias and accumulated errors. EdiTTS [34] used perturbations in the prior space to induce desired edits during denoising reversal. Guided-TTS [234] and Guided-TTS2 [235] used diffusion models in sound generation for text-to-speech. [236] combined voice diffusion models with a spectrogram-domain conditioning method for text-to-speech with unseen voices. InferGrad[239] improved the diffusion-based text-to-speech model for small inference steps. SpecGrad [237] adapted the time-varying spectral envelope of diffusion noise based on conditioning log-mel spectrograms. *ItoTTS* [238] unified text-to-speech and vocoder into a framework based on linear SDE. ProDiff [23] proposed a progressive and fast diffusion model for high-quality text-to-speech. BinauralGrad [228] explored the application of diffusion models in binaural audio synthesis.

4.4 AI for science

4.4.1 Molecular conformation generation

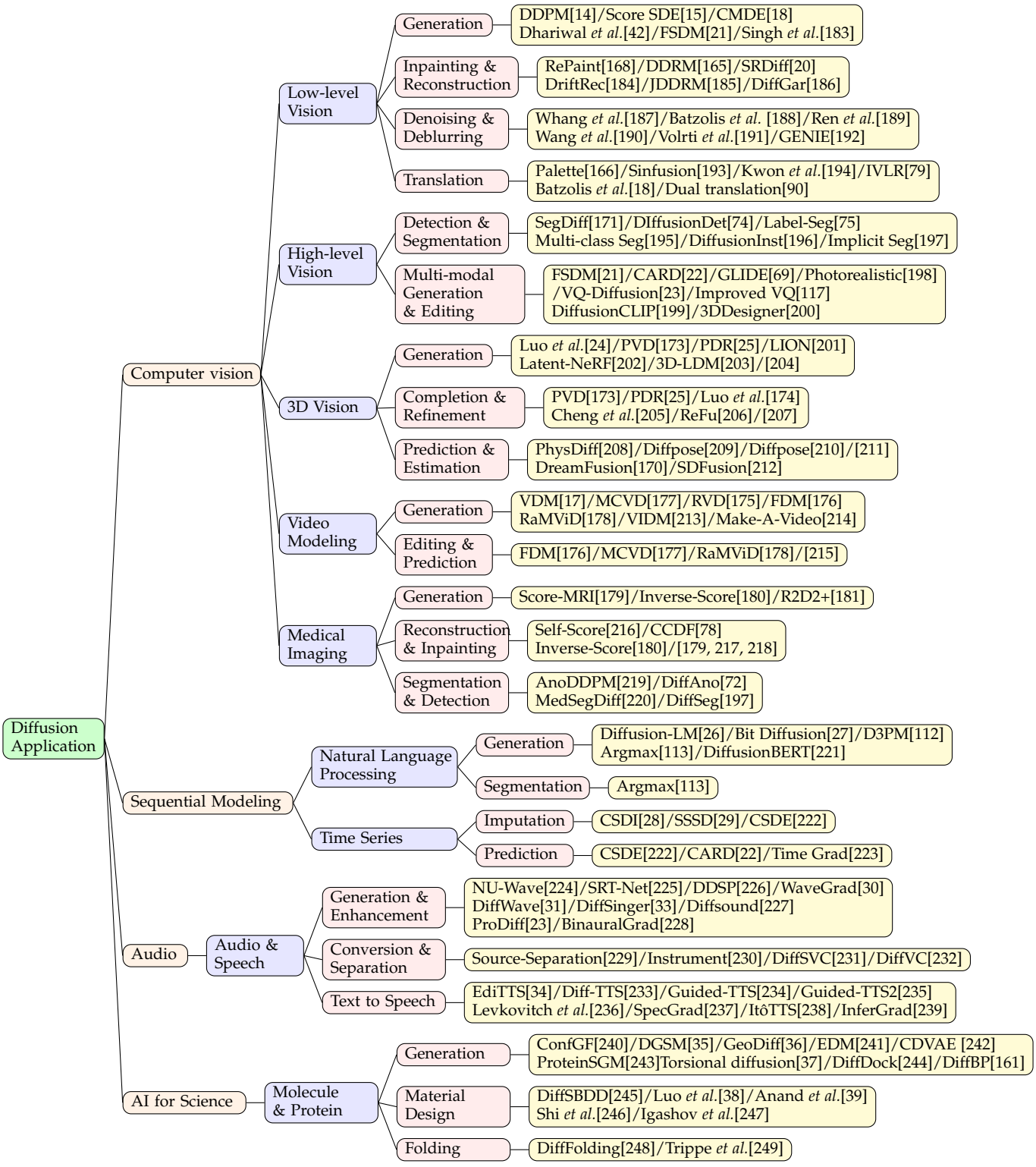
ConfGF [240] was an early work on diffusion-based molecular conformation generation models. While preserving rotation and translation equivariance, ConfGF generated samples by Langevin dynamics with physically inspired gradient fields. However, ConfGF only modeled local distances between the first-order, the second-order, and the third-order neighbors and thus failed to capture long-range interactions between non-bounded atoms. To tackle this challenge, DGSM [35] proposed to dynamically construct molecular graph structures between atoms based on their spatial proximity. GeoDiff [36] found that the model was fed with perturbed distance matrices during diffusion learning, which might violate mathematical constraints. Thus, GeoDiff introduced a roto-translational invariant Markov process to impose constraints on the density. EDM [241] further extended the above methods by incorporating discrete atom features and deriving the equations required for log-likelihood computation. Torsional diffusion [37] operated on the space of torsional angles and produced molecular conformations according to a diffusion process limited to the most flexible degrees of freedom. Based on previous geometric deep learning methods, DiffDock [244] conducts denoised score matching on transition, rotation, and torsion angle to generate drug conformation in protein-ligand complexes.

4.4.2 Material design

CDVAE [242] explored the periodic structure of stable material generation. To address the challenge that stable materials exist only in a low-dimensional subspace with all possible periodic arrangements of atoms, CDVAE designed a diffusion-based network as a decoder with output gradients leading to local minima of energy and updated atom types to capture specific local bonding preferences depending on the neighbors.

Inspired by the recent success of antibody modeling [250–252], the recent work [38] developed a diffusion-based generative model that explicitly targeted specific antigen

TABLE 2
Classification of Diffusion-based model Applications



structures and generated antibodies. The proposed method jointly sampled antibody sequences and structures and iteratively generated candidates in the sequence-structure space.

Anand *et al.* [39] introduced a diffusion-based generative model for both protein structure and sequence and learned the structural information that is equivariant to rotations and translations. ProteinSGM [243] formulated protein design as an image inpainting problem and applied conditional diffusion-based generation to precisely model the protein structure. DiffFolding [248] generates protein backbone concentrating on internal angles by traditional DDPM idea.

5 Conclusions & Discussions

The diffusion model becomes increasingly crucial to a wide range of applied fields. To utilize the power of the diffusion model, this paper provides a comprehensive and up-to-date review of several aspects of diffusion models using detailed insights on various attitudes, including theory, improved algorithms, and applications. We hope this survey serves as a guide for readers on diffusion model enhancement and its application.

Acknowledgments

This work is supported by the Science and Technology Innovation 2030 - Major Project (No. 2021ZD0150100) and the National Natural Science Foundation of China (No. U21A20427).

References

[1] C. Doersch, "Tutorial on variational autoencoders," *arXiv preprint arXiv:1606.05908*, 2016. (document), 2.2.1

[2] D. P. Kingma, M. Welling *et al.*, "An introduction to variational autoencoders," *Foundations and Trends® in Machine Learning*, 2019.

[3] A. Oussidi and A. Elhassouny, "Deep generative models: Survey," in *ISCV*. IEEE, 2018. (document)

[4] Y. LeCun, S. Chopra, R. Hadsell, M. Ranzato, and F. Huang, "A tutorial on energy-based learning," *Predicting structured data*, 2006. (document)

[5] J. Ngiam, Z. Chen, P. W. Koh, and A. Y. Ng, "Learning deep energy models," in *ICML*, 2011.

[6] A. G. ALIAS PARTH GOYAL, N. R. Ke, S. Ganguli, and Y. Bengio, "Variational walkback: Learning a transition operator as a stochastic recurrent net," *NIPS*, vol. 30, 2017. (document)

[7] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Communications of the ACM*, 2020. (document)

[8] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, "Generative adversarial networks: An overview," *IEEE Signal Process*, 2018. (document)

[9] L. Dinh, J. Sohl-Dickstein, and S. Bengio, "Density estimation using real nvp," *arXiv preprint arXiv:1605.08803*, 2016. (document)

[10] D. Rezende and S. Mohamed, "Variational inference with normalizing flows," in *ICML*, 2015.

[11] I. Kobyzev, S. J. Prince, and M. A. Brubaker, "Normalizing flows: An introduction and review of current methods," *IEEE TPAMI*, 2020.

[12] S. Bond-Taylor, A. Leach, Y. Long, and C. Willcocks, "Deep generative modelling: A comparative review of vaes, gans, normalizing flows, energy-based and autoregressive models." *IEEE TPAMI*, 2021. (document)

[13] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," in *ICML*, 2015. (document), 2.1.3, 2.2.1, 7

[14] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *NIPS*, 2020. 2.1.2, 2.1.3, 2.2.1, 2.2.1, 2.2.1, 3.2, 3.3, 2, 1, 5, 7

[15] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, "Score-based generative modeling through stochastic differential equations," *arXiv preprint arXiv:2011.13456*, 2020. (document), 2.1.3, 2.2.2, 2.2.2, 2.2.2, 3.1.3, 3.1.3, 3.2.3, 3.3, 2, 3, 3, 5, 7

[16] C. Saharia, W. Chan, H. Chang, C. A. Lee, J. Ho, T. Salimans, D. J. Fleet, and M. Norouzi, "Palette: Image-to-image diffusion models," 2021. (document)

[17] J. Ho, T. Salimans, A. Gritsenko, W. Chan, M. Norouzi, and D. J. Fleet, "Video diffusion models," 2022. 2.3, 4.1.2, 4.1.4, 2, 8

[18] G. Batzolis, J. Stanczuk, C.-B. Schönlieb, and C. Etmann, "Conditional image generation with score-based diffusion models," *arXiv preprint arXiv:2111.13606*, 2021. 4.1.1, 2, 8

[19] L. Theis, A. van den Oord, and M. Bethge, "A note on the evaluation of generative models," in *ICLR*, 2016, pp. 1–10.

[20] H. Li, Y. Yang, M. Chang, S. Chen, H. Feng, Z. Xu, Q. Li, and Y. Chen, "Srdiff: Single image super-resolution with diffusion probabilistic models," *Neurocomputing*, 2022. 4.1.1, 2, 8

[21] G. Giannone, D. Nielsen, and O. Winther, "Few-shot diffusion models," *arXiv preprint arXiv:2205.15463*, 2022. 4.1.2, 2, 8

[22] X. Han, H. Zheng, and M. Zhou, "Card: Classification and regression diffusion models," *arXiv preprint arXiv:2206.07275*, 2022. 4.1.2, 2, 8

[23] R. Huang, Z. Zhao, H. Liu, J. Liu, C. Cui, and Y. Ren, "Prodiff: Progressive fast diffusion model for high-quality text-to-speech," *arXiv preprint arXiv:2207.06389*, 2022. 4.1.2, 4.3, 2, 8

[24] S. Luo and W. Hu, "Diffusion probabilistic models for 3d point cloud generation," in *CVPR*, 2021, pp. 2837–2845. 4.1.3, 2, 7, 8

[25] Z. Lyu, Z. Kong, X. Xu, L. Pan, and D. Lin, "A conditional point diffusion-refinement paradigm for 3d point cloud completion," *arXiv preprint arXiv:2112.03530*, 2021. (document), 4.1.3, 2, 7, 8

[26] X. L. Li, J. Thickstun, I. Gulrajani, P. Liang, and T. B. Hashimoto, "Diffusion-lm improves controllable text generation," *arXiv preprint arXiv:2205.14217*, 2022. (document), 4.2.1, 2, 8

[27] T. Chen, R. Zhang, and G. Hinton, "Analog bits: Generating discrete data using diffusion models with self-

conditioning," *arXiv preprint arXiv:2208.04202*, 2022. 4.2.1, 2, 8

[28] Y. Tashiro, J. Song, Y. Song, and S. Ermon, "Csdi: Conditional score-based diffusion models for probabilistic time series imputation," *NIPS*, vol. 34, pp. 24 804–24 816, 2021. 2.3, 4.2.2, 2, 8

[29] J. M. L. Alcaraz and N. Strodthoff, "Diffusion-based time series imputation and forecasting with structured state space models," *arXiv preprint arXiv:2208.09399*, 2022. (document), 4.2.2, 2, 8

[30] N. Chen, Y. Zhang, H. Zen, R. J. Weiss, M. Norouzi, and W. Chan, "Wavegrad: Estimating gradients for waveform generation," in *ICLR*, 2020. (document), 4.3, 2, 8

[31] Z. Kong, W. Ping, J. Huang, K. Zhao, and B. Catanzaro, "Diffwave: A versatile diffusion model for audio synthesis," in *ICLR*, 2020. 4.3, 2, 8

[32] V. Popov, I. Vovk, V. Gogoryan, T. Sadekova, and M. Kudinov, "Grad-tts: A diffusion probabilistic model for text-to-speech," in *ICML*. PMLR, 2021, pp. 8599–8608. 4.3, 8

[33] J. Liu, C. Li, Y. Ren, F. Chen, and Z. Zhao, "Diffsinger: Singing voice synthesis via shallow diffusion mechanism," in *AAAI*, vol. 36, no. 10, 2022, pp. 11 020–11 028. 4.3, 2, 8

[34] J. Tae, H. Kim, and T. Kim, "Editts: Score-based editing for controllable text-to-speech," *arXiv preprint arXiv:2110.02584*, 2021. (document), 4.3, 2, 8

[35] S. Luo, C. Shi, M. Xu, and J. Tang, "Predicting molecular conformation via dynamic graph score matching," *NIPS*, vol. 34, pp. 19 784–19 795, 2021. (document), 4.4.1, 2, 8

[36] M. Xu, L. Yu, Y. Song, C. Shi, S. Ermon, and J. Tang, "Geodiff: A geometric diffusion model for molecular conformation generation," in *ICLR*, 2021. 4.4.1, 2, 8

[37] B. Jing, G. Corso, R. Barzilay, and T. S. Jaakkola, "Torsional diffusion for molecular conformer generation," in *ICLR*, 2022. 4.4.1, 2, 8

[38] S. Luo, Y. Su, X. Peng, S. Wang, J. Peng, and J. Ma, "Antigen-specific antibody design and optimization with diffusion-based generative models," *bioRxiv*, 2022. 4.4.2, 2, 8

[39] N. Anand and T. Achim, "Protein structure and sequence generation with equivariant denoising diffusion probabilistic models," *arXiv preprint arXiv:2205.15019*, 2022. (document), 3.2.3, 2, 4.4.2, 8

[40] T. Karras, M. Aittala, T. Aila, and S. Laine, "Elucidating the design space of diffusion-based generative models," *arXiv preprint arXiv:2206.00364*, 2022. (document), 1, 3.1.3, 3.2, 3.2.2, 6, 7

[41] P. Vincent, "A connection between score matching and denoising autoencoders," *Neural computation*, 2011. (document), 2.2.2

[42] P. Dhariwal and A. Nichol, "Diffusion models beat gans on image synthesis," *NIPS*, vol. 34, pp. 8780–8794, 2021. (document), 2.3, 2, 4

[43] C. Lu, Y. Zhou, F. Bao, J. Chen, C. Li, and J. Zhu, "Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models," *arXiv preprint arXiv:2211.01095*, 2022. (document), 2.3

[44] Q. Zhang and Y. Chen, "Fast sampling of diffusion models with exponential integrator," *arXiv preprint arXiv:2204.13902*, 2022. 1, 3.1.3, 5, 6, 7

[45] Y. Xu, M. Deng, X. Cheng, Y. Tian, Z. Liu, and T. Jaakkola, "Restart sampling for improving generative processes," *ArXiv*, vol. abs/2306.14878, 2023. (document), 1, 3.1.3, 3.1.3, 4, 7

[46] T. Salimans and J. Ho, "Progressive distillation for fast sampling of diffusion models," *arXiv*, 2022. (document), 3.1.1, 1, 6, 7

[47] Y. Xu, Z. Liu, M. Tegmark, and T. Jaakkola, "Poisson flow generative models," *ArXiv*, vol. abs/2209.11178, 2022. (document), 1, 3.2.2, 6, 7

[48] Y. Xu, Z. Liu, Y. Tian, S. Tong, M. Tegmark, and T. Jaakkola, "Pfgm++: Unlocking the potential of physics-inspired generative models," *ArXiv*, vol. abs/2302.04265, 2023. 1, 3.2.2, 6, 7

[49] T. Dockhorn, A. Vahdat, and K. Kreis, "Score-based generative modeling with critically-damped langevin diffusion," *arXiv preprint arXiv:2112.07068*, 2021. (document), 1, 3.2.2

[50] A. Vahdat, K. Kreis, and J. Kautz, "Score-based generative modeling in latent space," *NIPS*, vol. 34, pp. 11 287–11 302, 2021. (document), 1, 3.2.1, 3.2.1, 6, 7, 8

[51] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *CVPR*, 2022, pp. 10 684–10 695. (document), 2.3, 1, 3.1.3, 3.2.1

[52] X. Liu, C. Gong, and Q. Liu, "Flow straight and fast: Learning to generate and transfer data with rectified flow," *ArXiv*, vol. abs/2209.03003, 2022. (document), 1, 3.4

[53] M. S. Albergo and E. Vanden-Eijnden, "Building normalizing flows with stochastic interpolants," *ArXiv*, vol. abs/2209.15571, 2022. (document), 1, 3.4

[54] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," 2014. 1

[55] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. 1

[56] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013. 1

[57] G. Papamakarios, E. T. Nalisnick, D. J. Rezende, S. Mohamed, and B. Lakshminarayanan, "Normalizing flows for probabilistic modeling and inference." *J. Mach. Learn. Res.*, vol. 22, no. 57, pp. 1–64, 2021. 1

[58] C. Jarzynski, "Equilibrium free-energy differences from nonequilibrium measurements: A master-equation approach," *Physical Review E*, vol. 56, pp. 5018–5035, 1997. 2.2.1

[59] A. Q. Nichol and P. Dhariwal, "Improved denoising diffusion probabilistic models," in *ICML*, 2021. 2.2.1, 1, 3.1.2, 3.3, B.3, 5, 6, 7

[60] L. Arnold, "Stochastic differential equations," *New York*, 1974. 2.2.2

[61] B. Oksendal, *Stochastic differential equations: an introduction with applications*. Springer Science & Business Media, 2013. 2.2.2

[62] B. D. Anderson, "Reverse-time diffusion equation models," *Stochastic Processes and their Applications*, vol. 12, no. 3, pp. 313–326, 1982. 2.2.2

[63] Y. Xu, S. Tong, and T. Jaakkola, "Stable target field

for reduced variance score estimation in diffusion models," *ArXiv*, vol. abs/2302.00670, 2023. 2.2.2, 6

[64] D. Maoutsa, S. Reich, and M. Opper, "Interacting particle solutions of fokker–planck equations through gradient–log–density estimation," *Entropy*, vol. 22, no. 8, p. 802, 2020. 2.2.2

[65] R. T. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud, "Neural ordinary differential equations," *NIPS*, vol. 31, 2018. 2.2.2

[66] L. Liu, Y. Ren, Z. Lin, and Z. Zhao, "Pseudo numerical methods for diffusion models on manifolds," *arXiv preprint arXiv:2202.09778*, 2022. 2.2.2, 1, 3.1.3, 3.2.3, 3, 7

[67] C. Lu, Y. Zhou, F. Bao, J. Chen, C. Li, and J. Zhu, "Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps," *arXiv preprint arXiv:2206.00927*, 2022. 2.2.2, 1, 3.1.3, 3, 4, 6, 7

[68] J. Ho and T. Salimans, "Classifier-free diffusion guidance," *arXiv preprint arXiv:2207.12598*, 2022. 2.3, 5

[69] A. Nichol, P. Dhariwal, A. Ramesh, P. Shyam, P. Mishkin, B. McGrew, I. Sutskever, and M. Chen, "Glide: Towards photorealistic image generation and editing with text-guided diffusion models," *arXiv preprint arXiv:2112.10741*, 2021. 2.3, 1, 3.2.3, 4.1.2, 2, 8

[70] C. Meng, R. Gao, D. P. Kingma, S. Ermon, J. Ho, and T. Salimans, "On distillation of guided diffusion models," *arXiv preprint arXiv:2210.03142*, 2022.

[71] M. Hu, Y. Wang, T.-J. Cham, J. Yang, and P. N. Suganthan, "Global context with discrete diffusion in vector quantised modelling for image generation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 11 502–11 511. 2.3

[72] J. Wolleb, F. Bieder, R. Sandkühler, and P. C. Cattin, "Diffusion models for medical anomaly detection," *arXiv preprint arXiv:2203.04306*, 2022. 2.3, 2

[73] K. Packhäuser, L. Folle, F. Thamm, and A. Maier, "Generation of anonymous chest radiographs using latent diffusion models for training thoracic abnormality classification systems," *arXiv preprint arXiv:2211.01323*, 2022. 2.3

[74] S. Chen, P. Sun, Y. Song, and P. Luo, "Diffusiondet: Diffusion model for object detection," *arXiv preprint arXiv:2211.09788*, 2022. 2.3, 2

[75] D. Baranchuk, I. Rubachev, A. Voynov, V. Khrulkov, and A. Babenko, "Label-efficient semantic segmentation with diffusion models," *arXiv preprint arXiv:2112.03126*, 2021. 2.3, 2

[76] V. T. Hu, D. W. Zhang, Y. M. Asano, G. J. Burghouts, and C. G. Snoek, "Self-guided diffusion models," *arXiv preprint arXiv:2210.06462*, 2022. 2.3, 6

[77] C.-H. Chao, W.-F. Sun, B.-W. Cheng, and C.-Y. Lee, "Quasi-conservative score-based generative models," *arXiv preprint arXiv:2209.12753*, 2022. 2.3

[78] H. Chung, B. Sim, and J. C. Ye, "Come-closer-diffuse-faster: Accelerating conditional diffusion models for inverse problems through stochastic contraction," in *CVPR*, 2022. 2.3, 3.1.2, 1, 2, 7

[79] J. Choi, S. Kim, Y. Jeong, Y. Gwon, and S. Yoon, "Ilvr: Conditioning method for denoising diffusion probabilistic models," in *CVPR*, 2021, pp. 14 367–14 376. 2.3, 2

[80] E. Luhman and T. Luhman, "Knowledge distillation in iterative generative models for improved sampling speed," *arXiv*, 2021. 3.1.1, 1, 6, 7

[81] W. Sun, D. Chen, C. Wang, D. Ye, Y. Feng, and C. Chen, "Accelerating diffusion sampling with classifier-based feature distillation," *arXiv preprint arXiv:2211.12039*, 2022. 3.1.1, 1

[82] D. Berthelot, A. Autef, J. Lin, D. A. Yap, S. Zhai, S. Hu, D. Zheng, W. Talbot, and E. Gu, "Tract: Denoising diffusion models with transitive closure time-distillation," *arXiv preprint arXiv:2303.04248*, 2023. 3.1.1, 1

[83] X. Liu, C. Gong *et al.*, "Flow straight and fast: Learning to generate and transfer data with rectified flow," in *NeurIPS 2022 Workshop on Score-Based Methods*, 2022. 3.1.1, 1

[84] H. Zheng, W. Nie, A. Vahdat, K. Azizzadenesheli, and A. Anandkumar, "Fast sampling of diffusion models via operator learning," *arXiv preprint arXiv:2211.13449*, 2022. 3.1.1, 1

[85] Y. Fan and K. Lee, "Optimizing ddpm sampling with shortcut fine-tuning," *arXiv preprint arXiv:2301.13362*, 2023. 3.1.1, 1

[86] E. Aiello, D. Valsesia, and E. Magli, "Fast inference in denoising diffusion models via mmd finetuning," *arXiv preprint arXiv:2301.07969*, 2023. 3.1.1, 1

[87] S. Lee, B. Kim, and J. C. Ye, "Minimizing trajectory curvature of ode-based generative models," *arXiv preprint arXiv:2301.12003*, 2023. 3.1.1, 1

[88] C. Meng, R. Gao, D. P. Kingma, S. Ermon, J. Ho, and T. Salimans, "On distillation of guided diffusion models," *ArXiv*, vol. abs/2210.03142, 2022. 1

[89] G.-H. Liu, A. Vahdat, D.-A. Huang, E. A. Theodorou, W. Nie, and A. Anandkumar, "I2sb: Image-to-image schrödinger bridge," *ArXiv*, vol. abs/2302.05872, 2023. 1, 3.4

[90] X. Su, J. Song, C. Meng, and S. Ermon, "Dual diffusion implicit bridges for image-to-image translation," *arXiv preprint arXiv:2203.08382*, 2022. 1, 3.4, 2

[91] H. Zheng, P. He, W. Chen, and M. Zhou, "Truncated diffusion probabilistic models," *arXiv preprint arXiv:2202.09671*, 2022. 1, 3.1.2, 3.1.4, 5, 6, 7

[92] E. Hoogeboom and T. Salimans, "Blurring diffusion models," *arXiv preprint arXiv:2209.05557*, 2022. 1, 3.1.2

[93] Z. Lyu, X. Xu, C. Yang, D. Lin, and B. Dai, "Accelerating diffusion models via early stop of the diffusion process," *arXiv*, 2022. 1, 3.1.2, 3.1.4, 3, 4, 6, 7

[94] G. Daras, M. Delbracio, H. Talebi, A. G. Dimakis, and P. Milanfar, "Soft diffusion: Score matching for general corruptions," *arXiv preprint arXiv:2209.05442*, 2022. 1, 3.1.2

[95] G. Franzese, S. Rossi, L. Yang, A. Finamore, D. Rossi, M. Filippone, and P. Michiardi, "How much is enough? a study on diffusion times in score-based generative models." 3.1.2, 1, 6, 7

[96] V. Khrulkov and I. Oseledets, "Understanding ddpm latent codes through optimal transport," *arXiv preprint arXiv:2202.07477*, 2022. 1

[97] Z. Kong and W. Ping, "On fast sampling of diffusion

probabilistic models," *arXiv preprint arXiv:2106.00132*, 2021. 1, 3.1.2, 6, 7

[98] D. Kingma, T. Salimans, B. Poole, and J. Ho, "Variational diffusion models," *NIPS*, vol. 34, pp. 21 696–21 707, 2021. 1, 3.1.2, 3.3, 5, 7

[99] R. San-Roman, E. Nachmani, and L. Wolf, "Noise estimation for generative diffusion models," *arXiv preprint arXiv:2104.02600*, 2021. 1, 7

[100] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," in *ICLR*, 2020. 1, 3.1.3, 6, 7

[101] Q. Zhang, M. Tao, and Y. Chen, "gddim: Generalized denoising diffusion implicit models," *arXiv preprint arXiv:2206.05564*, 2022. 1, 3.1.3, 6, 7

[102] F. Bao, C. Li, J. Zhu, and B. Zhang, "Analytic-dpm: an analytic estimate of the optimal reverse variance in diffusion probabilistic models," *arXiv preprint arXiv:2201.06503*, 2022. 1, 3.1.3, 3, 4, 5, 6, 7

[103] F. Bao, C. Li, J. Sun, J. Zhu, and B. Zhang, "Estimating the optimal covariance with imperfect mean in diffusion probabilistic models," *arXiv preprint arXiv:2206.07309*, 2022. 1, 3.1.3, 3, 4, 5, 6, 7

[104] D. Watson, W. Chan, J. Ho, and M. Norouzi, "Learning fast samplers for diffusion models by differentiating through sample quality." 1, 4, 6, 7

[105] D. Watson, J. Ho, M. Norouzi, and W. Chan, "Learning to efficiently sample from diffusion probabilistic models," *arXiv*, 2021. 1, 3.1.3, 4, 7

[106] Z. Xiao, K. Kreis, and A. Vahdat, "Tackling the generative learning trilemma with denoising diffusion gans," *arXiv*, 2021. 1, 3.1.4, 6, 7

[107] K. Pandey, A. Mukherjee, P. Rai, and A. Kumar, "Diffusevae: Efficient, controllable and high-fidelity generation from low-dimensional latents." 1, 3.1.4, 3, 5, 6, 7

[108] D. Kim, B. Na, S. J. Kwon, D. Lee, W. Kang, and I.-C. Moon, "Maximum likelihood training of implicit nonlinear diffusion models," *arXiv preprint arXiv:2205.13699*, 2022. 1, 5, 7

[109] H. Zhang, R. Feng, Z. Yang, L. Huang, Y. Liu, Y. Zhang, Y. Shen, D. Zhao, J. Zhou, and F. Cheng, "Dimensionality-varying diffusion process," *arXiv preprint arXiv:2211.16032*, 2022. 1, 3.2.1

[110] A. Bansal, E. Borgnia, H.-M. Chu, J. S. Li, H. Kazemi, F. Huang, M. Goldblum, J. Geiping, and T. Goldstein, "Cold diffusion: Inverting arbitrary image transforms without noise," *arXiv preprint arXiv:2208.09392*, 2022. 1, 3.2.2

[111] Y. Lipman, R. T. Q. Chen, H. Ben-Hamu, M. Nickel, and M. Le, "Flow matching for generative modeling," *ArXiv*, vol. abs/2210.02747, 2022. 1, 3.2.2

[112] J. Austin, D. D. Johnson, J. Ho, D. Tarlow, and R. van den Berg, "Structured denoising diffusion models in discrete state-spaces," *NIPS*, 2021. 1, 3.2.3, 2, 5, 7, 8

[113] E. Hoogeboom, D. Nielsen, P. Jaini, P. Forré, and M. Welling, "Argmax flows and multinomial diffusion: Towards non-autoregressive language models," *NIPS*, 2021. 1, 3.2.3, 2, 7, 8

[114] E. Hoogeboom, A. A. Gritsenko, J. Bastings, B. Poole, R. v. d. Berg, and T. Salimans, "Autoregressive diffusion models," *arXiv preprint arXiv:2110.02037*, 2021. 1, 3.2.3, 7

[115] A. Campbell, J. Benton, V. De Bortoli, T. Rainforth, G. Deligiannidis, and A. Doucet, "A continuous time framework for discrete denoising models," *arXiv preprint arXiv:2205.14987*, 2022. 1, 7

[116] S. Gu, D. Chen, J. Bao, F. Wen, B. Zhang, D. Chen, L. Yuan, and B. Guo, "Vector quantized diffusion model for text-to-image synthesis," in *CVPR*, 2022, pp. 10 696–10 706. 1, 3.2.3, 7, 8

[117] Z. Tang, S. Gu, J. Bao, D. Chen, and F. Wen, "Improved vector quantized diffusion models," *arXiv preprint arXiv:2205.16007*, 2022. 1, 3.2.3, 2, 7, 8

[118] V. De Bortoli, E. Mathieu, M. Hutchinson, J. Thornton, Y. W. Teh, and A. Doucet, "Riemannian score-based generative modeling," *arXiv preprint arXiv:2202.02763*, 2022. 1, 3.2.3, 7

[119] C.-W. Huang, M. Aghajohari, A. J. Bose, P. Panangaden, and A. Courville, "Riemannian diffusion models," *arXiv preprint arXiv:2208.07949*, 2022. 1, 3.2.3, 7

[120] L. Luzi, A. Siahkoohi, P. M. Mayer, J. Casco-Rodriguez, and R. Baraniuk, "Boomerang: Local sampling on image manifolds using diffusion models," *arXiv preprint arXiv:2210.12100*, 2022. 1, 3.2.3

[121] X. Cheng, J. Zhang, and S. Sra, "Theory and algorithms for diffusion processes on riemannian manifolds," *arXiv preprint arXiv:2204.13665*, 2022. 1, 3.2.3

[122] C. Niu, Y. Song, J. Song, S. Zhao, A. Grover, and S. Ermon, "Permutation invariant graph generation via score-based generative modeling," in *AISTATS*. PMLR, 2020, pp. 4474–4484. 1, 3.2.3, 7

[123] H. Huang, L. Sun, B. Du, Y. Fu, and W. Lv, "Graphgdp: Generative diffusion processes for permutation invariant graph generation," *arXiv preprint arXiv:2212.01842*, 2022. 1, 3.2.3

[124] X. Chen, Y. Li, A. Zhang, and L.-p. Liu, "Nvdiff: Graph generation through the diffusion of node vectors," *arXiv preprint arXiv:2211.10794*, 2022. 1, 3.2.3

[125] T. Luo, Z. Mo, and S. J. Pan, "Fast graph generative model via spectral diffusion," *arXiv preprint arXiv:2211.08892*, 2022. 1, 3.2.3

[126] Y. Song, C. Durkan, I. Murray, and S. Ermon, "Maximum likelihood training of score-based diffusion models," *NIPS*, vol. 34, pp. 1415–1428, 2021. 1, 3.3, 7

[127] C.-W. Huang, J. H. Lim, and A. C. Courville, "A variational perspective on diffusion-based generative models and score matching," *NIPS*, 2021. 1, 3.3, 7

[128] C. Lu, K. Zheng, F. Bao, J. Chen, C. Li, and J. Zhu, "Maximum likelihood training for score-based diffusion odes by high-order denoising score matching," in *International Conference on Machine Learning*, 2022. 1, 3.3

[129] E. Heitz, L. Belcour, and T. Chambon, "Iterative α-(de)blending: a minimalist deterministic diffusion model," *ArXiv*, vol. abs/2305.03486, 2023. 1, 3.4

[130] R. G. Lopes, S. Fenu, and T. Starner, "Data-free knowledge distillation for deep neural networks," *arXiv preprint arXiv:1710.07535*, 2017. 3.1.1

[131] J. Gou, B. Yu, S. J. Maybank, and D. Tao, "Knowledge distillation: A survey," *IJCV*, 2021. 3.1.1

[132] Y. Song, P. Dhariwal, M. Chen, and I. Sutskever, "Con-

sistency models," *ArXiv*, vol. abs/2303.01469, 2023. 3.1.1
[133] C. Villani, "Topics in optimal transportation," *Graduate Studies in Mathematics*, 2003. 3.1.1
[134] H. Tsukamoto, S.-J. Chung, and J.-J. E. Slotine, "Contraction theory for nonlinear stability analysis and learning-based control: A tutorial overview," *Annual Reviews in Control*, 2021. 3.1.2
[135] N. V. Hung, S. Migórski, V. M. Tam, and S. Zeng, "Gap functions and error bounds for variational–hemivariational inequalities," *Acta Applicandae Mathematicae*, 2020. 3.1.2
[136] H. Zheng and M. Zhou, "Act: Asymptotic conditional transport," *arxiv*, 2020. 3.1.2
[137] A. Jolicoeur-Martineau, K. Li, R. Piché-Taillefer, T. Kachman, and I. Mitliagkas, "Gotta go fast when generating data with score-based models," *arXiv preprint arXiv:2105.14080*, 2021. 3.1.3, 5, 6
[138] R. Bellman, "Dynamic programming," *Science*, 1966. 3.1.3
[139] D. Watson, W. Chan, J. Ho, and M. Norouzi, "Learning fast samplers for diffusion models by differentiating through sample quality," in *ICLR*, 2021. 3.1.3
[140] D. Kim, B. Na, S. J. Kwon, D. Lee, W. Kang, and I.-c. Moon, "Maximum likelihood training of parametrized diffusion model," *arxiv*, 2021. 3.2.1, 3.2.1, 7
[141] C. Huang, Z. Liu, S. Bai, L. Zhang, C. Xu, Z. WANG, Y. Xiang, and Y. Xiong, "Pf-abgen: A reliable and efficient antibody generator via poisson flow," in *ICLR 2023-Machine Learning for Drug Discovery workshop*, 2023. 3.2.2
[142] R. Ge, Y. He, C. Xia, Y. Chen, D. Zhang, and G. Wang, "Jccs-pfgm: A novel circle-supervision based poisson flow generative model for multiphase cect progressive low-dose reconstruction with joint condition," 2023. 3.2.2
[143] Z. Liu, D. Luo, Y. Xu, T. Jaakkola, and M. Tegmark, "Genphys: From physical processes to generative models," *ArXiv*, vol. abs/2304.02637, 2023. 3.2.2
[144] S. Rissanen, M. Heinonen, and A. Solin, "Generative modelling with inverse heat dissipation," *ArXiv*, vol. abs/2206.13397, 2022. 3.2.2
[145] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *NIPS*, 2017. 3.2.3
[146] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018. 3.2.3
[147] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, "Hierarchical text-conditional image generation with clip latents," *arXiv preprint arXiv:2204.06125*, 2022. 3.2.3
[148] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko *et al.*, "Highly accurate protein structure prediction with alphafold," *Nature*, 2021. 3.2.3
[149] S. Ovchinnikov and P.-S. Huang, "Structure-based protein design with deep learning," *Current opinion in chemical biology*, 2021. 3.2.3
[150] A. Van Den Oord, O. Vinyals *et al.*, "Neural discrete representation learning," *NIPS*, vol. 30, 2017. 3.2.3
[151] M. Cohen, G. Quispe, S. L. Corff, C. Ollion, and E. Moulines, "Diffusion bridges vector quantized variational autoencoders," *arXiv preprint arXiv:2202.04895*, 2022. 3.2.3, 7
[152] P. Xie, Q. Zhang, Z. Li, H. Tang, Y. Du, and X. Hu, "Vector quantized diffusion model with codeunet for text-to-sign pose sequences generation," *arXiv preprint arXiv:2208.09141*, 2022. 3.2.3, 7, 8
[153] C. Guo, S. Zou, X. Zuo, S. Wang, W. Ji, X. Li, and L. Cheng, "Generating diverse and natural 3d human motions from text," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5152–5161. 3.2.3, 3.2.3
[154] S. Weinbach, M. Bellagente, C. Eichenberg, A. Dai, R. Baldock, S. Nanda, B. Deiseroth, K. Oostermeijer, H. Teufel, and A. F. Cruz-Salinas, "M-vader: A model for diffusion with multimodal context," *arXiv preprint arXiv:2212.02936*, 2022. 3.2.3
[155] X. Xu, Z. Wang, E. Zhang, K. Wang, and H. Shi, "Versatile diffusion: Text, images and variations all in one diffusion model," *arXiv preprint arXiv:2211.08332*, 2022. 3.2.3
[156] H. A. Pierson and M. S. Gashler, "Deep learning in robotics: a review of recent research," *Advanced Robotics*, 2017. 3.2.3
[157] R. P. De Lima, K. Marfurt, D. Duarte, and A. Bonar, "Progress and challenges in deep learning analysis of geoscience images," in *81st EAGE Conference and Exhibition 2019*. European Association of Geoscientists & Engineers, 2019. 3.2.3
[158] J. Wang, H. Cao, J. Z. Zhang, and Y. Qi, "Computational protein design with deep learning neural networks," *Scientific reports*, 2018. 3.2.3
[159] W. Cao, Z. Yan, Z. He, and Z. He, "A comprehensive survey on geometric deep learning," *IEEE Access*, 2020. 3.2.3
[160] L. Wu, H. Lin, Z. Gao, C. Tan, and S. Z. Li, "Self-supervised on graphs: Contrastive, generative, or predictive," *IEEE TKDE*, 2021. 3.2.3
[161] H. Lin, Y. Huang, M. Liu, X. Li, S. Ji, and S. Z. Li, "Diffbp: Generative diffusion of 3d molecules for target protein binding," *arXiv preprint arXiv:2211.11214*, 2022. 3.2.3, 2
[162] V. Dutordoir, A. Saul, Z. Ghahramani, and F. Simpson, "Neural diffusion processes," *arXiv preprint arXiv:2206.03992*, 2022. 3.2.3
[163] A. Ulhaq, N. Akhtar, and G. Pogrebna, "Efficient diffusion models for vision: A survey," *arXiv preprint arXiv:2210.09292*, 2022. 4
[164] Y. Song and S. Ermon, "Generative modeling by estimating gradients of the data distribution," *NIPS*. 4.1.1, 2, 3, 5, 7
[165] B. Kawar, M. Elad, S. Ermon, and J. Song, "Denoising diffusion restoration models," in *ICLR Workshop*, 2022. 4.1.1, 2, 8
[166] C. Saharia, W. Chan, H. Chang, C. Lee, J. Ho, T. Salimans, D. Fleet, and M. Norouzi, "Palette: Image-to-image diffusion models," in *ACM SIGGRAPH*, 2022,

pp. 1–10. 4.1.1, 2, 8
[167] L. Theis, T. Salimans, M. D. Hoffman, and F. Mentzer, "Lossy compression with gaussian diffusion," *arXiv preprint arXiv:2206.08889*, 2022. 4.1.1, 8
[168] A. Lugmayr, M. Danelljan, A. Romero, F. Yu, R. Timofte, and L. Van Gool, "Repaint: Inpainting using denoising diffusion probabilistic models," in *CVPR*, 2022, pp. 11 461–11 471. 4.1.1, 2, 8
[169] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *ICML*. PMLR, 2021, pp. 8748–8763. 4.1.2
[170] B. Poole, A. Jain, J. T. Barron, and B. Mildenhall, "Dreamfusion: Text-to-3d using 2d diffusion," *arXiv preprint arXiv:2209.14988*, 2022. 4.1.2, 2, 8
[171] T. Amit, E. Nachmani, T. Shaharbany, and L. Wolf, "Segdiff: Image segmentation with diffusion probabilistic models," *arXiv preprint arXiv:2112.00390*, 2021. 4.1.2, 2, 8
[172] P. Sanchez and S. A. Tsaftaris, "Diffusion causal models for counterfactual estimation," in *First Conference on Causal Learning and Reasoning*, 2021. 4.1.2
[173] L. Zhou, Y. Du, and J. Wu, "3d shape generation and completion through point-voxel diffusion," in *ICCV*, 2021, pp. 5826–5835. 4.1.3, 2, 8
[174] S. Luo and W. Hu, "Score-based point cloud denoising," in *ICCV*, 2021, pp. 4583–4592. 4.1.3, 2, 8
[175] R. Yang, P. Srivastava, and S. Mandt, "Diffusion probabilistic modeling for video generation," *arXiv preprint arXiv:2203.09481*, 2022. 4.1.4, 2, 8
[176] W. Harvey, S. Naderiparizi, V. Masrani, C. Weilbach, and F. Wood, "Flexible diffusion modeling of long videos," *arXiv preprint arXiv:2205.11495*, 2022. 4.1.4, 2, 8
[177] V. Voleti, A. Jolicoeur-Martineau, and C. Pal, "Mcvd: Masked conditional video diffusion for prediction, generation, and interpolation," *arXiv preprint arXiv:2205.09853*, 2022. 4.1.4, 2, 8
[178] T. Höppe, A. Mehrjou, S. Bauer, D. Nielsen, and A. Dittadi, "Diffusion models for video prediction and infilling," *arXiv preprint arXiv:2206.07696*, 2022. 4.1.4, 2, 8
[179] H. Chung and J. C. Ye, "Score-based diffusion models for accelerated mri," *Medical Image Analysis*, p. 102479, 2022. 4.1.5, 2, 8
[180] Y. Song, L. Shen, L. Xing, and S. Ermon, "Solving inverse problems in medical imaging with score-based generative models," in *ICLR*, 2021. 4.1.5, 2, 8
[181] H. Chung, E. S. Lee, and J. C. Ye, "Mr image denoising and super-resolution using regularized reverse diffusion," *arXiv preprint arXiv:2203.12621*, 2022. 4.1.5, 2, 8
[182] P. Sanchez, A. Kascenas, X. Liu, A. Q. O'Neil, and S. A. Tsaftaris, "What is healthy? generative counterfactual diffusion for lesion localization," *arXiv preprint arXiv:2207.12268*, 2022. 4.1.5
[183] J. Singh, S. Gould, and L. Zheng, "High-fidelity guided image synthesis with latent diffusion models," *arXiv preprint arXiv:2211.17084*, 2022. 2
[184] S. Welker, H. N. Chapman, and T. Gerkmann, "Driftrec: Adapting diffusion models to blind image restoration tasks," *arXiv preprint arXiv:2211.06757*, 2022. 2
[185] B. Kawar, J. Song, S. Ermon, and M. Elad, "Jpeg artifact correction using denoising diffusion restoration models," *arXiv preprint arXiv:2209.11888*, 2022. 2
[186] Y. Yin, L. Huang, Y. Liu, and K. Huang, "Diffgar: Model-agnostic restoration from generative artifacts using image-to-image diffusion models," *arXiv preprint arXiv:2210.08573*, 2022. 2
[187] J. Whang, M. Delbracio, H. Talebi, C. Saharia, A. G. Dimakis, and P. Milanfar, "Deblurring via stochastic refinement," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 16 293–16 303. 2
[188] G. Batzolis, J. Stanczuk, C.-B. Schönlieb, and C. Etmann, "Non-uniform diffusion models," *arXiv preprint arXiv:2207.09786*, 2022. 2
[189] M. Ren, M. Delbracio, H. Talebi, G. Gerig, and P. Milanfar, "Image deblurring with domain generalizable diffusion models," *arXiv preprint arXiv:2212.01789*, 2022. 2
[190] X. Wang, J.-K. Yan, J.-Y. Cai, J.-H. Deng, Q. Qin, Q. Wang, H. Xiao, Y. Cheng, and P.-F. Ye, "Super-resolution reconstruction of single image for latent features," *arXiv preprint arXiv:2211.12845*, 2022. 2
[191] V. Voleti, C. Pal, and A. Oberman, "Score-based denoising diffusion with non-isotropic gaussian noise models," *arXiv preprint arXiv:2210.12254*, 2022. 2
[192] T. Dockhorn, A. Vahdat, and K. Kreis, "Genie: Higher-order denoising diffusion solvers," *arXiv preprint arXiv:2210.05475*, 2022. 2
[193] Y. Nikankin, N. Haim, and M. Irani, "Sinfusion: Training diffusion models on a single image or video," *arXiv preprint arXiv:2211.11743*, 2022. 2
[194] G. Kwon and J. C. Ye, "Diffusion-based image translation using disentangled style and content representation," *arXiv preprint arXiv:2209.15264*, 2022. 2
[195] B. Kolbeinsson and K. Mikolajczyk, "Multi-class segmentation from aerial views using recursive noise diffusion," *arXiv preprint arXiv:2212.00787*, 2022. 2
[196] Z. Gu, H. Chen, Z. Xu, J. Lan, C. Meng, and W. Wang, "Diffusioninst: Diffusion model for instance segmentation," *arXiv preprint arXiv:2212.02773*, 2022. 2
[197] J. Wolleb, R. Sandkühler, F. Bieder, P. Valmaggia, and P. C. Cattin, "Diffusion models for implicit image segmentation ensembles," *arXiv preprint arXiv:2112.03145*, 2021. 2
[198] C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. Denton, S. K. S. Ghasemipour, B. K. Ayan, S. S. Mahdavi, R. G. Lopes *et al.*, "Photorealistic text-to-image diffusion models with deep language understanding," *arXiv preprint arXiv:2205.11487*, 2022. 2
[199] G. Kim and J. C. Ye, "Diffusionclip: Text-guided image manipulation using diffusion models," 2021. 2
[200] G. Li, H. Zheng, C. Wang, C. Li, C. Zheng, and D. Tao, "3ddesigner: Towards photorealistic 3d object generation and editing with text-guided diffusion models," *arXiv preprint arXiv:2211.14108*, 2022. 2
[201] X. Zeng, A. Vahdat, F. Williams, Z. Gojcic, O. Litany, S. Fidler, and K. Kreis, "Lion: Latent point diffu-

sion models for 3d shape generation," *arXiv preprint arXiv:2210.06978*, 2022. 2

[202] G. Metzer, E. Richardson, O. Patashnik, R. Giryes, and D. Cohen-Or, "Latent-nerf for shape-guided generation of 3d shapes and textures," *arXiv preprint arXiv:2211.07600*, 2022. 2

[203] G. Nam, M. Khlifi, A. Rodriguez, A. Tono, L. Zhou, and P. Guerrero, "3d-ldm: Neural implicit 3d shape generation with latent diffusion models," *arXiv preprint arXiv:2212.00842*, 2022. 2

[204] J. R. Shue, E. R. Chan, R. Po, Z. Ankner, J. Wu, and G. Wetzstein, "3d neural field generation using triplane diffusion," *arXiv preprint arXiv:2211.16677*, 2022. 2

[205] A.-C. Cheng, X. Li, S. Liu, M. Sun, and M.-H. Yang, "Autoregressive 3d shape generation via canonical mapping," *arXiv preprint arXiv:2204.01955*, 2022. 2, 8

[206] G. Shim, M. Lee, and J. Choo, "Refu: Refine and fuse the unobserved view for detail-preserving single-image 3d human reconstruction," in *Proceedings of the 30th ACM International Conference on Multimedia*, 2022, pp. 6850–6859. 2

[207] D. Wei, H. Sun, B. Li, J. Lu, W. Li, X. Sun, and S. Hu, "Human joint kinematics diffusion-refinement for stochastic motion prediction," *arXiv preprint arXiv:2210.05976*, 2022. 2

[208] Y. Yuan, J. Song, U. Iqbal, A. Vahdat, and J. Kautz, "Physdiff: Physics-guided human motion diffusion model," *arXiv preprint arXiv:2212.02500*, 2022. 2

[209] J. Gong, L. G. Foo, Z. Fan, Q. Ke, H. Rahmani, and J. Liu, "Diffpose: Toward more reliable 3d pose estimation," *arXiv preprint arXiv:2211.16940*, 2022. 2

[210] K. Holmquist and B. Wandt, "Diffpose: Multi-hypothesis human pose estimation using diffusion models," *arXiv preprint arXiv:2211.16487*, 2022. 2

[211] G. Tevet, S. Raab, B. Gordon, Y. Shafir, D. Cohen-Or, and A. H. Bermano, "Human motion diffusion model," *arXiv preprint arXiv:2209.14916*, 2022. 2

[212] Y.-C. Cheng, H.-Y. Lee, S. Tulyakov, A. Schwing, and L. Gui, "Sdfusion: Multimodal 3d shape completion, reconstruction, and generation," *arXiv preprint arXiv:2212.04493*, 2022. 2

[213] K. Mei and V. M. Patel, "Vidm: Video implicit diffusion models," *arXiv preprint arXiv:2212.00235*, 2022. 2

[214] U. Singer, A. Polyak, T. Hayes, X. Yin, J. An, S. Zhang, Q. Hu, H. Yang, O. Ashual, O. Gafni *et al.*, "Make-a-video: Text-to-video generation without text-video data," *arXiv preprint arXiv:2209.14792*, 2022. 2

[215] G. Kim, H. Shim, H. Kim, Y. Choi, J. Kim, and E. Yang, "Diffusion video autoencoders: Toward temporally consistent face video editing via disentangled video encoding," *arXiv preprint arXiv:2212.02802*, 2022. 2

[216] Z.-X. Cui, C. Cao, S. Liu, Q. Zhu, J. Cheng, H. Wang, Y. Zhu, and D. Liang, "Self-score: Self-supervised learning on score-based models for mri reconstruction," *arXiv preprint arXiv:2209.00835*, 2022. 2

[217] A. Jalal, M. Arvinte, G. Daras, E. Price, A. G. Dimakis, and J. Tamir, "Robust compressed sensing mri with deep generative priors," *Advances in Neural Information Processing Systems*, vol. 34, pp. 14 938–14 954, 2021. 2

[218] P. Rouzrokh, B. Khosravi, S. Faghani, M. Moassefi, S. Vahdati, and B. J. Erickson, "Multitask brain tumor inpainting with diffusion models: A methodological report," *arXiv preprint arXiv:2210.12113*, 2022. 2

[219] J. Wyatt, A. Leach, S. M. Schmon, and C. G. Willcocks, "Anoddpm: Anomaly detection with denoising diffusion probabilistic models using simplex noise," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 650–656. 2

[220] J. Wu, H. Fang, Y. Zhang, Y. Yang, and Y. Xu, "Medsegdiff: Medical image segmentation with diffusion probabilistic model," *arXiv preprint arXiv:2211.00611*, 2022. 2

[221] Z. He, T. Sun, K. Wang, X. Huang, and X. Qiu, "Diffusionbert: Improving generative masked language models with diffusion models," *arXiv preprint arXiv:2211.15029*, 2022. 2

[222] S. W. Park, K. Lee, and J. Kwon, "Neural markov controlled sde: Stochastic optimization for continuous-time data," in *ICLR*, 2021. 4.2.2, 2, 8

[223] K. Rasul, C. Seward, I. Schuster, and R. Vollgraf, "Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting," in *International Conference on Machine Learning*. PMLR, 2021, pp. 8857–8868. 2

[224] J. Lee and S. Han, "Nu-wave: A diffusion probabilistic model for neural audio upsampling," *arXiv preprint arXiv:2104.02321*, 2021. 2

[225] Z. Qiu, M. Fu, Y. Yu, L. Yin, F. Sun, and H. Huang, "Srtnet: Time domain speech enhancement via stochastic refinement," *arXiv preprint arXiv:2210.16805*, 2022. 2

[226] D.-Y. Wu, W.-Y. Hsiao, F.-R. Yang, O. Friedman, W. Jackson, S. Bruzenak, Y.-W. Liu, and Y.-H. Yang, "Ddsp-based singing vocoders: A new subtractive-based synthesizer and a comprehensive evaluation," *arXiv preprint arXiv:2208.04756*, 2022. 2

[227] D. Yang, J. Yu, H. Wang, W. Wang, C. Weng, Y. Zou, and D. Yu, "Diffsound: Discrete diffusion model for text-to-sound generation," *arXiv preprint arXiv:2207.09983*, 2022. 4.3, 2, 8

[228] Y. Leng, Z. Chen, J. Guo, H. Liu, J. Chen, X. Tan, D. Mandic, L. He, X.-Y. Li, T. Qin *et al.*, "Binauralgrad: A two-stage conditional diffusion probabilistic model for binaural audio synthesis," *arXiv preprint arXiv:2205.14807*, 2022. 4.3, 2, 8

[229] R. Scheibler, Y. Ji, S.-W. Chung, J. Byun, S. Choe, and M.-S. Choi, "Diffusion-based generative speech source separation," *arXiv preprint arXiv:2210.17327*, 2022. 2

[230] S. Han, H. Ihm, D. Ahn, and W. Lim, "Instrument separation of symbolic music by explicitly guided diffusion model," *arXiv preprint arXiv:2209.02696*, 2022. 2

[231] S. Liu, Y. Cao, D. Su, and H. Meng, "Diffsvc: A diffusion probabilistic model for singing voice conversion," in *IEEE ASRU*, 2021. 2, 8

[232] V. Popov, I. Vovk, V. Gogoryan, T. Sadekova, M. S. Kudinov, and J. Wei, "Diffusion-based voice conversion with fast maximum likelihood sampling scheme," in *ICLR*, 2021. 4.3, 2, 8

[233] M. Jeong, H. Kim, S. J. Cheon, B. J. Choi, and N. S.

Kim, "Diff-TTS: A Denoising Diffusion Model for Text-to-Speech," in *Proc. Interspeech 2021*, 2021, pp. 3605–3609. 4.3, 2, 8

[234] H. Kim, S. Kim, and S. Yoon, "Guided-tts: A diffusion model for text-to-speech via classifier guidance," in *ICML*. PMLR, 2022, pp. 11 119–11 133. 4.3, 2, 8

[235] S. Kim, H. Kim, and S. Yoon, "Guided-tts 2: A diffusion model for high-quality adaptive text-to-speech with untranscribed data," *arXiv preprint arXiv:2205.15370*, 2022. 4.3, 2, 8

[236] A. Levkovitch, E. Nachmani, and L. Wolf, "Zero-shot voice conditioning for denoising diffusion tts models," *arXiv preprint arXiv:2206.02246*, 2022. 4.3, 2, 8

[237] Y. Koizumi, H. Zen, K. Yatabe, N. Chen, and M. Bacchiani, "Specgrad: Diffusion probabilistic model based neural vocoder with adaptive noise spectral shaping," *arXiv preprint arXiv:2203.16749*, 2022. 4.3, 2, 8

[238] S. Wu and Z. Shi, "Itôtts and itôwave: Linear stochastic differential equation is all you need for audio generation," *arXiv e-prints*, pp. arXiv–2105, 2021. 4.3, 2, 8

[239] Z. Chen, X. Tan, K. Wang, S. Pan, D. Mandic, L. He, and S. Zhao, "Infergrad: Improving diffusion models for vocoder by considering inference in training," in *ICASSP*. IEEE, 2022, pp. 8432–8436. 4.3, 2

[240] C. Shi, S. Luo, M. Xu, and J. Tang, "Learning gradient fields for molecular conformation generation," in *ICML*. PMLR, 2021, pp. 9558–9568. 4.4.1, 2, 8

[241] E. Hoogeboom, V. G. Satorras, C. Vignac, and M. Welling, "Equivariant diffusion for molecule generation in 3d," in *ICML*. PMLR, 2022, pp. 8867–8887. 4.4.1, 2, 8

[242] T. Xie, X. Fu, O.-E. Ganea, R. Barzilay, and T. S. Jaakkola, "Crystal diffusion variational autoencoder for periodic material generation," in *ICLR*, 2021. 4.4.2, 2, 8

[243] J. S. Lee and P. M. Kim, "Proteinsgm: Score-based generative modeling for de novo protein design," *bioRxiv*, 2022. 2, 4.4.2, 8

[244] G. Corso, H. Stärk, B. Jing, R. Barzilay, and T. Jaakkola, "Diffdock: Diffusion steps, twists, and turns for molecular docking," *arXiv preprint arXiv:2210.01776*, 2022. 4.4.1, 2, 8

[245] A. Schneuing, Y. Du, C. Harris, A. Jamasb, I. Igashov, W. Du, T. Blundell, P. Lió, C. Gomes, M. Welling *et al.*, "Structure-based drug design with equivariant diffusion models," *arXiv preprint arXiv:2210.13695*, 2022. 2

[246] C. Shi, C. Wang, J. Lu, B. Zhong, and J. Tang, "Protein sequence and structure co-design with equivariant translation," *arXiv preprint arXiv:2210.08761*, 2022. 2

[247] I. Igashov, H. Stärk, C. Vignac, V. G. Satorras, P. Frossard, M. Welling, M. Bronstein, and B. Correia, "Equivariant 3d-conditional diffusion models for molecular linker design," *arXiv preprint arXiv:2210.05274*, 2022. 2

[248] K. E. Wu, K. K. Yang, R. v. d. Berg, J. Y. Zou, A. X. Lu, and A. P. Amini, "Protein structure generation via folding diffusion," *arXiv preprint arXiv:2209.15611*, 2022. 2, 4.4.2, 8

[249] B. L. Trippe, J. Yim, D. Tischer, T. Broderick, D. Baker, R. Barzilay, and T. Jaakkola, "Diffusion probabilistic modeling of protein backbones in 3d for the motif-scaffolding problem," *arXiv preprint arXiv:2206.04119*, 2022. 2

[250] W. Jin, J. Wohlwend, R. Barzilay, and T. S. Jaakkola, "Iterative refinement graph neural network for antibody sequence-structure co-design," in *ICLR*, 2021. 4.4.2

[251] T. Fu and J. Sun, "Antibody complementarity determining regions (cdrs) design using constrained energy model," in *SIGKDD*, 2022, pp. 389–399.

[252] W. Jin, R. Barzilay, and T. Jaakkola, "Antibody-antigen docking and design via hierarchical structure refinement," in *ICML*. PMLR, 2022, pp. 10 217–10 227. 4.4.2

[253] A. Borji, "Pros and cons of gan evaluation measures: New developments," *Computer Vision and Image Understanding*, vol. 215, p. 103329, 2022. B.1

[254] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," *NIPS*, vol. 29, 2016. B.1

[255] S. Kullback, *Information theory and statistics*. Courier Corporation, 1997. B.1

[256] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," in *NIPS*. Curran Associates, Inc. B.2

[257] A. Razavi, A. Van den Oord, and O. Vinyals, "Generating diverse high-fidelity images with vq-vae-2," *NIPS*, vol. 32, 2019. B.3

[258] T. M. Nguyen, A. Garg, R. G. Baraniuk, and A. Anandkumar, "Infocnf: Efficient conditional continuous normalizing flow using adaptive solvers," 2019. B.3

[259] Z. Ziegler and A. Rush, "Latent normalizing flows for discrete sequences," in *ICML*. PMLR, 2019, pp. 7673–7682. B.3

[260] J. Tomczak and M. Welling, "Vae with a vampprior," in *AISTATS*. PMLR, 2018, pp. 1214–1223. B.3

[261] O. Rybkin, K. Daniilidis, and S. Levine, "Simple and effective vae training with calibrated decoders," in *ICML*. PMLR, 2021, pp. 9179–9189. B.3

[262] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009. C

[263] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017. C

[264] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *ICCV*, December 2015. C

[265] F. Yu, A. Seff, Y. Zhang, S. Song, T. Funkhouser, and J. Xiao, "Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop," *arXiv preprint arXiv:1506.03365*, 2015. C

[266] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *CVPR*, June 2019. C

[267] Y. LeCun and C. Cortes, "MNIST handwritten digit database." C

[268] H. Chung, B. Sim, D. Ryu, and J. C. Ye, "Improving diffusion models for inverse problems using manifold constraints," *arXiv*. 4

[269] Y. Song, S. Garg, J. Shi, and S. Ermon, "Sliced score matching: A scalable approach to density and score estimation," in *Uncertainty in Artificial Intelligence*, 2020. 5

[270] Y. Song and S. Ermon, "Improved techniques for training score-based generative models," *NIPS*, 2020. 5, 7

[271] Q. Zhang and Y. Chen, "Diffusion normalizing flow," *NIPS*. 6, 7

[272] R. Gao, Y. Song, B. Poole, Y. N. Wu, and D. P. Kingma, "Learning energy-based models by diffusion recovery likelihood," *arXiv preprint arXiv:2012.08125*, 2020. 7

[273] Y. Song and D. P. Kingma, "How to train your energy-based models," *arXiv preprint arXiv:2101.03288*, 2021. 7

[274] V. De Bortoli, A. Doucet, J. Heng, and J. Thornton, "Simulating diffusion bridges with score matching," *arXiv preprint arXiv:2111.07243*, 2021. 7

[275] L. Zhou, Y. Du, and J. Wu, "3d shape generation and completion through point-voxel diffusion," in *ICCV*, 2021. 7

APPENDIX A
SAMPLING ALGORITHMS

In this section, we provide a brief guide on current mainstream sampling methods. We divide them into two parts: unconditional sampling and conditional sampling. For unconditional sampling, we present the original sampling algorithms for three landmarks. For conditional sampling, we divide them into the labeled condition and the unlabeled condition.

A.1 Unconditional Sampling

A.1.1 Ancestral Sampling

Algorithm 1 Ancestral Sampling [14]

$x_T \sim \mathcal{N}(0, I)$
for $t = T, ..., 1$ **do**
  $z \sim \mathcal{N}(0, I)$
  $x_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon_\theta \left(x_t, t \right) \right) + \sigma_t z$
end for
return x_0

A.1.2 Annealed Langevin Dynamics Sampling

Algorithm 2 Annealed Langevin Dynamics Sampling [164]

Initialize x_0
for $i = 1, ..., L$ **do**
  $\alpha_i \leftarrow \epsilon \cdot \sigma_i^2 / \sigma_L^2$
  for $t = 1, ..., L$ **do**
    $z_t \sim \mathcal{N}(0, I)$
    $\tilde{\mathbf{x}}_t = \tilde{\mathbf{x}}_{t-1} + \frac{\alpha_i}{2} \mathbf{s}_\theta \left(\tilde{\mathbf{x}}_{t-1}, \sigma_i \right) + \sqrt{\alpha_i} \mathbf{z}_t$
  end for
  $\tilde{x}_0 \leftarrow \tilde{x}_T$
end for
return $\tilde{x}_T$

A.1.3 Predictor-Corrector Sampling

Algorithm 3 Predictor-Corrector Sampling [15]

$x_N \sim \mathcal{N} \left(0, \sigma_{\max}^2 I \right)$
for $i = N - 1$ to 0 **do**
  $z \sim \mathcal{N}(0, I)$
  if Variance Exploding SDE **then**
    $x_i' \leftarrow x_{i+1} + \left(\sigma_{i+1}^2 - \sigma_i^2 \right) s_{\boldsymbol{\theta}} * (x_{i+1}, \sigma_{i+1})$
    $x_i \leftarrow x_i' + \sqrt{\sigma_{i+1}^2 - \sigma_i^2} z$
  else if Variance Preserving SDE **then**
    $x_i' \leftarrow \left(2 - \sqrt{1 - \beta_{i+1}} \right) x_{i+1} + \beta_{i+1} s_{\boldsymbol{\theta}} * (x_{i+1}, i+1)$
    $x_i \leftarrow x_i' + \sqrt{\beta_{i+1}} z$
  end if
  for $j = 1$ to M **do**
    $z \sim \mathcal{N}(0, I)$
    $x_i \leftarrow x_i + \epsilon_i s_{\boldsymbol{\theta}} * (x_i, \sigma_i) + \sqrt{2\epsilon_i} z$
  end for
end for
return x_0

A.2 Conditional Sampling

A.2.1 Labeled Condition

Algorithm 4 Classifier-guided Diffusion Sampling [42]

Input: class label y, gradient scale s
$x_T \sim \mathcal{N}(0, I)$
for $t = T, ..., 1$ **do**
  if DDPM Sampling **then**
    $\mu, \Sigma \leftarrow \mu_\theta(x_t), \Sigma_\theta(x_t)$
    $x_{t-q} \leftarrow$ sample from $\mathcal{N} \left(\mu + s \Sigma \nabla_{x_t} \log p_\phi \left(y \mid x_t \right), \Sigma \right)$
  end if
  if DDIM Sampling **then**
    $\hat{\epsilon} \leftarrow \epsilon_\theta \left(x_t \right) - \sqrt{1 - \bar{\alpha}_t} \nabla_{x_t} \log p_\phi \left(y \mid x_t \right)$
    $x_{t-1} \leftarrow \sqrt{\bar{\alpha}_{t-1}} \left(\frac{x_t - \sqrt{1-\bar{\alpha}_t} \hat{\epsilon}}{\sqrt{\bar{\alpha}_t}} \right) + \sqrt{1 - \bar{\alpha}_{t-1}} \hat{\epsilon}$
  end if
end for
return x_0

Algorithm 5 Classifier-free Guidance Sampling [68]

Input: guidance w, conditioning c, SNR $\lambda_1, ..., \lambda_T$
$z \sim \mathcal{N}(0, I)$
for $t = 1, ..., T$ **do**
  $\tilde{\boldsymbol{\epsilon}}_t = (1 + w) \boldsymbol{\epsilon}_\theta \left(\mathbf{z}_t, \mathbf{c} \right) - w \boldsymbol{\epsilon}_\theta \left(\mathbf{z}_t \right)$
  $\tilde{\mathbf{x}}_t = \left(\mathbf{z}_t - \sigma_{\lambda_t} \tilde{\boldsymbol{\epsilon}}_t \right) / \alpha_{\lambda_t}$
  $\mathbf{z}_{t+1} \sim \mathcal{N} \left(\tilde{\boldsymbol{\mu}}_{\lambda_{t+1} | \lambda_t} \left(\mathbf{z}_t, \tilde{\mathbf{x}}_t \right), \left(\tilde{\sigma}_{\lambda_{t+1} | \lambda_t} \right)^{1-v} \left(\sigma_{\lambda_t | \lambda_{t+1}}^2 \right)^v \right)$
end for
return z_{T+1}

A.3 Unlabeled Condition

Algorithm 6 Self-guided Conditional Sampling [76]

Input: guidance w, annotation map f_ψ, g_ϕ, dataset $\mathcal{D}$, label $\mathbf{k}$, segmentation label $\mathbf{k}_s$, image guidance $\hat{\mathbf{k}}$
$x_T \sim \mathcal{N}(0, I)$
for $t = T, ..., 1$ **do**
  $z \sim \mathcal{N}(0, I)$
  if Self Guidance **then**
    $\tilde{\epsilon} \leftarrow (1 - w) \boldsymbol{\epsilon}_\theta \left(\mathbf{x}_t, t \right) + w \boldsymbol{\epsilon}_\theta \left(\mathbf{x}_t, t; f_\psi \left(g_\phi(\mathbf{x}; \mathcal{D}); \mathcal{D} \right) \right)$
  else if Self-Labeled Guidance **then**
    $\tilde{\epsilon} \leftarrow \boldsymbol{\epsilon}_\theta \left(\mathbf{x}_t, \text{concat}[t, \mathbf{k}] \right)$
  else if Self-Boxed Guidance **then**
    $\tilde{\epsilon} \leftarrow \boldsymbol{\epsilon}_\theta \left(\text{concat}[\mathbf{x}_t, \mathbf{k}_s], \text{concat}[t, \mathbf{k}] \right)$
  else if Self-Segmented Guidance **then**
    $\tilde{\epsilon} \leftarrow \boldsymbol{\epsilon}_\theta \left(\text{concat}[\mathbf{x}_t, \mathbf{k}_s], \text{concat}[t, \hat{\mathbf{k}}] \right)$
  end if
  $x_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \tilde{\epsilon} \right) + \sigma_t z$
end for
return x_0

APPENDIX B
EVALUATION METRIC

B.1 Inception Score (IS)

The inception score is built on valuing the diversity and resolution of generated images based on the ImageNet dataset [253, 254]. It can be divided into two parts: diversity measurement and quality measurement. Diversity measurement denoted by p_{IS} is calculated w.r.t. the class entropy of generated samples: the larger the entropy is, the more diverse the samples will be. Quality measurement denoted by q_{IS} is computed through the similarity between a sample and the related class images using entropy. It is because the samples will enjoy high resolution if they are closer to the specific class of images in the ImageNet dataset. Thus, to lower q_{IS} and higher p_{IS}, the KL divergence [255] is applied to inception score calculation:

$$\begin{aligned} IS &= D_{KL}(p_{IS} \parallel q_{IS}) \\ &= \mathbb{E}_{x\sim p_{IS}}\left[\log \frac{p_{IS}}{q_{IS}}\right] \\ &= \mathbb{E}_{x\sim p_{IS}}[\log(p_{IS}) - \log(p_{IS})] \end{aligned} \tag{23}$$

B.2 Frechet Inception Distance (FID)

Although there are reasonable evaluation techniques in the Inception Score, the establishment is based on a specific dataset with 1000 classes and a trained network that consists of randomness such as initial weights, and code framework. Thus, the bias between ImageNet and real-world images may cause an inaccurate outcome. Furthermore, the number of sample batches is much less than 1000 classes, leading to a value

FID is proposed to solve the bias from the specific reference datasets. The score shows the distance between real-world data distribution and the generated samples using the mean and the covariance [256].

$$\text{FID} = \left\|\mu_r - \mu_g\right\|^2 + \text{Tr}\left(\Sigma_r + \Sigma_g - 2\left(\Sigma_r\Sigma_g\right)^{1/2}\right) \tag{24}$$

where μ_g, Σ_g are the mean and covariance of generated samples, and μ_r, Σ_r are the mean and covariance of real-world data.

B.3 Negative Log Likelihood (NLL)

According to Razavi *et al.*, [257] negative log-likelihood is seen as a common evaluation metric that describes all modes of data distribution. Lots of works on normalizing flow field [258, 259] and VAE field [260, 261] uses NLL as one of the choices for evaluation. Some diffusion models like improved DDPM [59] regard the NLL as the training objective.

$$\text{NLL} = \mathbb{E}\left[-\log p_\theta\left(x\right)\right] \tag{25}$$

APPENDIX C
BENCHMARKS

The benchmarks of landmark models along with improved techniques corresponding to **FID score**, **Inception Score**, and **NLL** are provided on diverse datasets which includes CIFAR-10 [262], ImageNet[263], and CelebA-64 [264]. In addition, some dataset-based performances such as LSUN[265], FFHQ[266], and MINST[267] are not presented since there is much less experiment data. The selected performance are listed according to NFE in descending order to compare for easier access.

C.1 Benchmarks on CelebA-64

TABLE 3
Benchmarks on CelebA-64

Method	NFE	FID	NLL
NPR-DDIM [103]	1000	3.15	-
SN-DDIM [103]	1000	2.90	-
NCSN [164]	1000	10.23	-
NCSN ++ [15]	1000	1.92	1.97
DDPM ++ [15]	1000	1.90	2.10
DiffuseVAE [107]	1000	4.76	-
Analytic DPM [102]	1000	-	2.66
ES-DDPM [93]	200	2.55	-
PNDM [66]	200	2.71	-
ES-DDPM [93]	100	3.01	-
PNDM [66]	100	2.81	-
Analytic DPM [102]	100	-	2.66
NPR-DDIM [103]	100	4.27	-
SN-DDIM [103]	100	3.04	-
ES-DDPM [93]	50	3.97	-
PNDM [66]	50	3.34	-
NPR-DDIM [103]	50	6.04	-
SN-DDIM [103]	50	3.83	-
DPM-Solver Discrete [67]	36	2.71	-
ES-DDPM [93]	20	4.90	-
PNDM [66]	20	5.51	-
DPM-Solver Discrete [67]	20	2.82	-
ES-DDPM [93]	10	6.44	-
PNDM [66]	10	7.71	-
Analytic DPM [102]	10	-	2.97
NPR-DDPM [103]	10	28.37	-
SN-DDPM [103]	10	20.60	-
NPR-DDIM [103]	10	14.98	-
SN-DDIM [103]	10	10.20	-
DPM-Solver Discrete [67]	10	6.92	-
ES-DDPM [93]	5	9.15	-
PNDM [66]	5	11.30	-

C.2 Benchmarks on ImageNet-64

C.3 Benchmarks on CIFAR-10 Dataset

APPENDIX D
DETAILS FOR IMPROVEMENT ALGORITHMS

APPENDIX E
TABLE OF NOTATION

TABLE 4
Benchmarks on ImageNet-64

Method	NFE	FID	IS	NLL
MCG [268]	1000	25.4	-	-
Analytic DPM [102]	1000	-	-	3.61
ES-DDPM [93]	900	2.07	55.29	-
Restart [45]	623	1.36	-	-
Efficient Sampling [105]	256	3.87	-	-
Analytic DPM [102]	200	-	-	3.64
NPR-DDPM [103]	200	16.96	-	-
SN-DDPM [103]	200	16.61	-	-
ES-DDPM [93]	100	3.75	48.63	-
DPM-Solver Discrete [67]	57	17.47	-	-
Restart [45]	39	2.38	-	-
ES-DDPM [93]	25	3.75	48.63	-
GGDM [104]	25	18.4	18.12	-
Analytic DPM [102]	25	-	-	3.83
NPR-DDPM [103]	25	28.27	-	-
SN-DDPM [103]	25	27.58	-	-
DPM-Solver Discrete [67]	20	18.53	-	-
ES-DDPM [93]	10	3.93	48.81	-
GGDM [104]	10	37.32	14.76	-
DPM-Solver Discrete [67]	10	24.4	-	-
ES-DDPM [93]	5	4.25	48.04	-
GGDM [104]	5	55.14	12.9	-

TABLE 5
Benchmarks on CIFAR-10 (NFE $\geq$ 1000)

Method	NFE	FID	IS	NLL
Improved DDPM [59]	4000	2.90	-	-
VE SDE [15]	2000	2.20	9.89	-
VP SDE [15]	2000	2.41	9.68	3.13
sub-VP SDE [15]	2000	2.41	9.57	2.92
DDPM [14]	1000	3.17	9.46	3.72
NCSN [164]	1000	25.32	8.87	-
SSM [269]	1000	54.33	-	-
NCSNv2 [270]	1000	10.87	8.40	-
D3PM [112]	1000	7.34	8.56	3.44
Efficient Sampling [137]	1000	2.94	-	-
NCSN++ [15]	1000	2.33	10.11	3.04
DDPM++ [15]	1000	2.47	9.78	2.91
TDPM [91]	1000	3.07	9.24	-
VDM [98]	1000	4.00	-	-
DiffuseVAE [107]	1000	8.72	8.63	-
Analytic DPM [102]	1000	-	-	3.59
NPR-DDPM [103]	1000	4.27	-	-
SN-DDPM [103]	1000	4.07	-	-
Gotta Go Fast VP [44]	1000	2.49	-	-
Gotta Go Fast VE [44]	1000	3.14	-	-
INDM [108]	1000	2.28	-	3.09

TABLE 6
Benchmarks on CIFAR-10 (NFE $<$ 1000)

Method	NFE	FID	IS	NLL
Diffusion Step [95]	600	3.72	-	-
ES-DDPM [93]	600	3.17	-	-
Diffusion Step [95]	400	14.38	-	-
Diffusion Step [95]	200	5.44	-	-
NPR-DDPM [103]	200	4.10	-	-
SN-DDPM [103]	200	3.72	-	-
Gotta Go Fast VP [44]	180	2.44	-	-
Gotta Go Fast VE [44]	180	3.40	-	-
LSGM [50]	138	2.10	-	-
PFGM [47]	110	2.35	-	-
DDIM [100]	100	4.16	-	-
FastDPM [97]	100	2.86	-	-
TDPM [91]	100	3.10	9.34	-
NPR-DDPM [103]	100	4.52	-	-
SN-DDPM [103]	100	3.83	-	-
DiffuseVAE [107]	100	11.71	8.27	-
DiffFlow [271]	100	14.14	-	3.04
Analytic DPM [102]	100	-	-	3.59
Efficient Sampling [137]	64	3.08	-	-
DPM-Solver [67]	51	2.59	-	-
DDIM [100]	50	4.67	-	-
FastDPM [97]	50	3.2	-	-
NPR-DDPM [103]	50	5.31	-	-
SN-DDPM [103]	50	4.17	-	-
Improved DDPM [59]	50	4.99	-	-
TDPM [91]	50	3.3	9.22	-
DEIS [137]	50	2.57	-	-
gDDIM [101]	50	2.28	-	-
DPM-Solver Discrete [67]	44	3.48	-	-
STF [63]	35	1.90	-	-
EDM [40]	35	1.79	-	-
PFGM++ [48]	35	1.74	-	-
Improved DDPM [59]	25	7.53	-	-
GGDM [104]	25	4.25	9.19	-
NPR-DDPM [103]	25	7.99	-	-
SN-DDPM [103]	25	6.05	-	-
DDIM [100]	20	6.84	-	-
FastDPM [97]	20	5.05	-	-
DEIS [137]	20	2.86	-	-
DPM-Solver [67]	20	2.87	-	-
DPM-Solver Discrete [67]	20	3.72	-	-
Efficient Sampling [137]	16	3.41	-	-
NPR-DDPM [103]	10	19.94	-	-
SN-DDPM [103]	10	16.33	-	-
DDIM [100]	10	13.36	-	-
FastDPM [97]	10	9.90	-	-
GGDM [104]	10	8.23	8.90	-
Analytic DPM [102]	10	-	-	4.11
DEIS [137]	10	4.17	-	-
DPM-Solver [67]	10	6.96	-	-
DPM-Solver Discrete [67]	10	10.16	-	-
Progressive Distillation [46]	8	2.57	-	-
Denoising Diffusion GAN [106]	8	4.36	9.43	-
GGDM [104]	5	13.77	8.53	-
DEIS [137]	5	15.37	-	-
Progressive Distillation [46]	4	3.00	-	-
TDPM [91]	4	3.41	9.00	-
Denoising Diffusion GAN [106]	4	3.75	9.63	-
Progressive Distillation [46]	2	4.51	-	-
TDPM [91]	2	4.47	8.97	-
Denoising Diffusion GAN [106]	2	4.08	9.80	-
Denoising student [80]	1	9.36	8.36	-
Progressive Distillation [46]	1	9.12	-	-
TDPM [91]	1	8.91	8.65	-

TABLE 7
Details for Improved Diffusion Methods

Method	Year	Data	Model	Framework	Training	Sampling	Code
Landmark Works							
DPM [13]	2015	RGB Image	Discrete	Diffusion	L_{simple}	Ancestral	[code]
DDPM [14]	2020	RGB Image	Discrete	Diffusion	L_{simple}	Ancestral	[code]
NCSN [164]	2019	RGB Image	Discrete	Score	L_{DSM}	Langevin dynamics	[code]
NCSNv2 [270]	2020	RGB Image	Discrete	Score	L_{DSM}	Langevin dynamics	[code]
Score SDE [15]	2020	RGB Image	Continuous	SDE	L_{DSM}	PC-Sampling	[code]
Improved Works							
Progressive Distill [46]	2022	RGB Image	Discrete	Diffusion	L_{simple}	DDIM Sampling	[code]
Denoising Student [80]	2021	RGB Image	Discrete	Diffusion	$L_{Distill}$	DDIM Sampling	[code]
TDPM [91]	2022	RGB Image	Discrete	Diffusion	$L_{DDPM\&GAN}$	Ancestral	-
ES-DDPM [93]	2022	RGB Image	Discrete	Diffusion	$L_{DDPM\&VAE}$	Conditional Sampling	[code]
CCDF [78]	2021	RGB Image	Discrete	SDE	L_{simple}	Langevin dynamics	[code]
Franzese's Model [95]	2022	RGB Image	Continuous	SDE	L_{DSM}	DDIM Sampling	-
FastDPM [97]	2021	RGB Image	Discrete	Diffusion	L_{simple}	DDIM Sampling	[code]
Improved DDPM [59]	2021	RGB Image	Discrete	Diffusion	L_{hybrid}	Ancestral	[code]
VDM [98]	2022	RGB Image	Both	Diffusion	L_{simple}	Ancestral	[code]
San-Roman's Model [99]	2021	RGB Image	Discrete	Diffusion	$L_{DDPM\&Noise}$	Ancestral	-
Analytic-DPM [102]	2022	RGB Image	Discrete	Score	$L_{Trajectory}$	Ancestral	[code]
NPR-DDPM [103]	2022	RGB Image	Discrete	Diffusion	$L_{DDPM\&Noise}$	Ancestral	[code]
SN-DDPM [103]	2022	RGB Image	Discrete	Score	L_{square}	Ancestral	[code]
DDIM [100]	2021	RGB Image	Discrete	Diffusion	L_{simple}	DDIM Sampling	[code]
gDDIM [101]	2022	RGB Image	Continuous	SDE&ODE	L_{DSM}	PC-Sampling	[code]
INDM [108]	2022	RGB Image	Continuous	SDE	$L_{DDPM\&Flow}$	PC-Sampling	-
Gotta Go Fast [44]	2021	RGB Image	Continuous	SDE	L_{DSM}	Improved Euler	[code]
DPM-Solver [67]	2022	RGB Image	Continuous	ODE	L_{DSM}	Higher ODE solvers	[code]
Restart [45]	2023	RGB Image	Continuous	SDE	L_{DSM}	2^{nd} Order Heun	[code]
EDM [40]	2022	RGB Image	Continuous	ODE	L_{DSM}	2^{nd} Order Heun	[code]
PFGM [47]	2022	RGB Image	Continuous	ODE	L_{DSM}	ODE-Solver	[code]
PFGM++ [48]	2023	RGB Image	Continuous	ODE	L_{DSM}	2^{nd} Order Heun	[code]
PNDM [66]	2022	Manifold	Discrete	ODE	L_{simple}	Multi-step & Runge-Kutta	[code]
DDSS [104]	2021	RGB Image	Discrete	Diffusion	L_{simple}	Dynamic Programming	-
GGDM [105]	2022	RGB Image	Discrete	Diffusion	L_{KID}	Dynamic Programming	-
Diffusion GAN [106]	2022	RGB Image	Discrete	Diffusion	$L_{DDPM\&GAN}$	Ancestral	[code]
DiffuseVAE [107]	2022	RGB Image	Discrete	Diffusion	$L_{DDPM\&VAE}$	Ancestral	[code]
DiffFlow [271]	2021	RGB Image	Discrete	SDE	L_{DSM}	Langevin & Flow Sampling	[code]
LSGM [50]	2021	RGB Image	Continuous	ODE	$L_{DDPM\&VAE}$	ODE-Slover	[code]
Score-flow [126]	2021	Dequantization	Continuous	SDE	L_{DSM}	PC-Sampling	[code]
PDM [140]	2022	RGB Image	Continuous	SDE	L_{Gap}	PC-Sampling	-
ScoreEBM [272]	2021	RGB Image	Discrete	Score	$L_{Recovery}$	Langevin dynamics	[code]
Song's Model [273]	2021	RGB Image	Discrete	Score	L_{DSM}	Langevin dynamics	-
Huang's Model [127]	2021	RGB Image	Continuous	SDE	L_{DSM}	SDE-Solver	[code]
De Bortoli's Model [274]	2021	RGB Image	Continuous	SDE	L_{DSM}	Importance Sampling	[code]
PVD [275]	2021	Point Cloud	Discrete	Diffusion	L_{simple}	Ancestral	[code]
Luo's Model [24]	2021	Point Cloud	Discrete	Diffusion	L_{simple}	Ancestral	[code]
Lyu's Model [25]	2022	Point Cloud	Discrete	Diffusion	L_{simple}	Farthest Point Sampling	[code]
D3PM [112]	2021	Categorical Data	Discrete	Diffusion	L_{hybrid}	Ancestral	[code]
Argmax [113]	2021	Categorical Data	Discrete	Diffusion	$L_{DDPM\&Flow}$	Gumbel sampling	[code]
ARDM [114]	2022	Categorical Data	Discrete	Diffusion	L_{simple}	Ancestral	[code]
Campbell's Model [115]	2022	Categorical Data	Continuous	Diffusion	L^{CT}_{simple}	PC-Sampling	[code]
VQ-diffusion [116]	2022	Vector-Quantized	Discrete	Diffusion	L_{simple}	Ancestral	[code]
Improved VQ-Diff [117]	2022	Vector-Quantized	Discrete	Diffusion	L_{simple}	Purity Prior Sampling	[code]
Cohen's Model [151]	2022	Vector-Quantized	Discrete	Diffusion	L_{simple}	Ancestral & VAE Sampling	[code]
Xie's Model [152]	2022	Vector-Quantized	Discrete	Diffusion	$L_{DDPM\&Class}$	Ancestral & VAE Sampling	-
RGSM [118]	2022	Manifold	Continuous	SDE	L_{DSM}	Geodesic Random Walk	-
RDM [119]	2022	Manifold	Continuous	SDE	L^{CT}_{simple}	Importance Sampling	-
EDP-GNN [122]	2020	Graph	Discrete	Score	L_{DSM}	Langevin dynamics	[code]

TABLE 8
Details for Diffusion Applications

Method	Year	Data	Framework	Downstream Task	Code
Computer Vision					
CMDE [18]	2021	RGB-Image	SDE	Inpainting, Super-Resolution, Edge to image translation	[code]
DDRM [165]	2022	RGB-Image	Diffusion	Super-Resolution, Deblurring, Inpainting, Colorization	[code]
Palette [166]	2022	RGB-Image	Diffusion	Colorization, Inpainting, Uncropping, JPEG Restoration	[code]
DiffC [167]	2022	RGB-Image	SDE	Compression	-
SRDiff [20]	2021	RGB-Image	Diffusion	Super-Resolution -	
RePaint [168]	2022	RGB-Image	Diffusion	Inpainting, Super-resolution, Edge to Image Translation	[code]
FSDM [21]	2022	RGB-Image	Diffusion	Few-shot Generation	-
CARD [22]	2022	RGB-Image	Diffusion	Conditional Generation	[code]
GLIDE [69]	2022	RGB-Image	Diffusion	Conditional Generation	[code]
LSGM [50]	2022	RGB-Image	SDE	UnConditional & Conditional Generation	[code]
SegDiff [171]	2022	RGB-Image	Diffusion	Segmentation	-
VQ-Diffusion [116]	2022	VQ Data	Diffusion	Text-to-Image Synthesis	[code]
DreamFusion [170]	2023	VQ Data	Diffusion	Text-to-Image Synthesis	[code]
Text-to-Sign VQ [152]	2022	VQ Data	Diffusion	Conditional Pose Generation	-
Improved VQ-Diff [117]	2022	VQ Data	Diffusion	Text-to-Image Synthesis	-
Luo's Model [24]	2021	Point Cloud	Diffusion	Point Cloud Generation	[code]
PVD [173]	2022	Point Cloud	Diffusion	Point Cloud Generation, Point-Voxel representation	[code]
PDR [25]	2022	Point Cloud	Diffusion	Point Cloud Completion	[code]
Cheng's Model [205]	2022	Point Cloud	Diffusion	Point Cloud Generation	[code]
Luo's Model[174]	2022	Point Cloud	Score	Point Cloud Denoising	[code]
VDM [17]	2022	Video	Diffusion	Text-Conditioned Video Generation	[code]
RVD [175]	2022	Video	Diffusion	Video Forecasting, Video compression	[code]
FDM [176]	2022	Video	Diffusion	Video Forecasting, Long-range Video modeling	-
MCVD [177]	2022	Video	Diffusion	Video Prediction, Video Generation, Video Interpolation	[code]
RaMViD [178]	2022	Video	SDE	Conditional Generation	-
Score-MRI [179]	2022	MRI	SDE	MRI Reconstruction	[code]
Song's Model [180]	2022	MRI, CT	SDE	MRI Reconstruction, CT Reconstruction	[code]
R2D2+ [181]	2022	MRI	SDE	MRI Denoising	-
Sequence Modeling					
Diffusion-LM [26]	2022	Text	Diffusion	Conditional Text Generation	[code]
Bit Diffusion [27]	2022	Text	Diffusion	Image-Conditional Text Generation	[code]
D3PM [112]	2021	Text	Diffusion	Text Generation	-
Argmax [113]	2021	Text	Diffusion	Test Segmantation, Text Generation	[code]
CSDI [28]	2021	Time Series	Diffusion	Series Imputation	[code]
SSSD [29]	2022	Time Series	Diffusion	Series Imputation	[code]
CSDE [222]	2022	Time Series	SDE	Series Imputation, Series Predicton	-
Audio & Speech					
WaveGrad [30]	2020	Audio	Diffusion	Conditional Wave Generation	[code]
DiffWave [31]	2021	Audio	Diffusion	Conditional & Unconditional Wave Generation	[code]
GradTTS [32]	2021	Audio	SDE	Wave Generation	[code]
Diff-TTS [233]	2021	Audio	Diffusion	non-AR mel-Spectrogram Generation, Speech Synthesis	-
DiffVC [232]	2022	Audio	SDE	Voice conversion	[code]
DiffSVC [231]	2022	Audio	Diffusion	Voice Conversion	[code]
DiffSinger [33]	2022	Audio	Diffusion	Singing Voice Synthesis	[code]
Diffsound [227]	2021	Audio	Diffusion	Text-to-sound Generation tasks	[code]
EdiTTS [34]	2022	Audio	SDE	fine-grained pitch, content editing	[code]
Guided-TTS [234]	2022	Audio	SDE	Conditional Speech Generation	-
Guided-TTS2 [235]	2022	Audio	SDE	Conditional Speech Generation	-
Levkovitch's Model [236]	2022	Audio	SDE	Spectrograms-Voice Generation	[code]
SpecGrad [237]	2022	Audio	Diffusion	Spectrograms-Voice Generation	[code]
ItoTTS [238]	2022	Audio	SDE	Spectrograms-Voice Generation	-
ProDiff [23]	2022	Audio	Diffusion	Text-to-Speech Synthesis	[code]
BinauralGrad [228]	2022	Audio	Diffusion	Binaural Audio Synthesis	-
AI For Science					
ConfGF [240]	2021	Molecular	Score	Conformation Generation	[code]
DGSM [35]	2022	Molecular	Score	Conformation Generation, Sidechain Generation	-
GeoDiff [36]	2022	Molecular	Diffusion	Conformation Generation	[code]
EDM [241]	2022	Molecular	SDE	Conformation Generation	[code]
Torsional Diff [37]	2022	Molecular	Diffusion	Molecular Generation	[code]
DiffDock [244]	2022	Molecular&protein	Diffusion	Conformation Generation, molecular docking	[code]
CDVAE [242]	2022	Protein	Score	Periodic Material Generation	[code]
Luo's Model [38]	2022	Protein	Diffusion	CDR Generation	-
Anand's Model [39]	2022	Protein	Diffusion	Protein Sequence and Structure Generation	-
ProteinSGM [243]	2022	Protein	SDE	de novo protein design	-
DiffFolding [248]	2022	Protein	Diffusion	Protein Inverse Folding	[code]

TABLE 9
Notions in Diffusion Systems

Notations	Descriptions
T	Discrete total time steps
t	Random time t
z_t	Random noise with normal distribution
ϵ	Random noise with normal distribution
$\mathcal{N}$	Normal distribution
β	Generalized process noise scale
β_t	Variance scale coefficients
$\beta(t)$	Continuous-time β_t
σ	Generalized process noise scale
σ_t	Noise scale of perturbation
$\sigma(t)$	Continuous-time σ_t
α_t	Mean coefficient defined as 1 - β_t
$\alpha(t)$	Continuous-time α_t
$\bar{\alpha}_t$	Cumulative product of α_t
$\gamma(t)$	Signal-to-Noise ratio
η_t	Step size of annealed Langevin dynamics
x	Unperturbed data distribution
$\tilde{x}$	Perturbed data distribution
x_0	Starting distribution of data
x_t	Diffused data at time t
$x_t^{'}$	Partly diffused data at time t
x_T	Random noise after diffusion
$F(x, \sigma)$	Forward/Diffusion process
$R(x, \sigma)$	Reverse/Denoised process
$F_t(x_t, \sigma_t)$	Forward/Diffusion step at time t
$R_t(x_t, \sigma_t)$	Reverse/Denoised step at time t
$q(x_t \mid x_{t-1})$	DDPM forward step at time t
$p(x_{t-1} \mid x_t)$	DDPM reverse step at time t
$f(x, t)$	Drift coefficient of SDE
$g(t)$	Simplified diffusion coefficient of SDE
$\mathcal{D}(x, t)$	Degrader at time t in Cold Diffusion
$\mathcal{R}(x, t)$	Reconstructor at time t in Cold Diffusion
$w, \bar{w}$	Standard Wiener process
$\nabla_x \log p_t(x)$	Score function w.r.t x
$\mu_\theta(x_t, t)$	Mean coefficient of reversed step
$\Sigma_\theta(x_t, t)$	Variance coefficient of reversed step
$\epsilon_\theta(x_t, t)$	Noise prediction model
$s_\theta(x)$	Score network model
L_0, L_{t-1}, L_T	Forward loss, reversed loss, decoder loss
L_{vlb}	Evidence Lower Bound
L_{vlb}^{CT}	Continuous evidence lower bound
L_{simple}	Simplified denoised diffusion loss
L_{simple}^{CT}	Continuous L_{simple}
L_{Gap}	Variational gap
L_{KID}	Kernel inception distance
$L_{Recovery}$	Recovery likelihood loss
L_{hybrid}	Hybrid diffusion loss
$L_{DDPM\&GAN}$	DPM ELBO and GAN hybrid loss
$L_{DDPM\&VAE}$	DPM ELBO and VAE hybrid loss
$L_{DDPM\&Flow}$	DPM ELBO and normalizing flow hybrid loss
L_{DSM}	Loss of denoised score matching
L_{ISM}	Loss of implicit score matching
L_{SSM}	Loss of sliced score matching
$L_{Distill}$	Diffusion distillation loss
$L_{DDPM\&Noise}$	DPM ELBO and reverse noise hybrid loss
L_{Square}	Noise square loss
$L_{Trajectory}$	Process optimization loss
$L_{DDPM\&Class}$	DPM ELBO and classification hybrid loss
θ	learnable parameters
ϕ	learnable parameters