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 have unlocked another profound realm of human creativity. By capturing and generalizing patterns within data, we have entered the epoch of all-encompassing Artificial Intelligence for General Creativity (AIGC). Notably, diffusion models, recognized as one of the paramount generative models, materialize human ideation into tangible instances across diverse domains, encompassing imagery, text, speech, biology, and healthcare. To provide advanced and comprehensive insights into diffusion, this survey comprehensively elucidates its developmental trajectory and future directions from three distinct angles: the fundamental formulation of diffusion, algorithmic enhancements, and the manifold applications of diffusion. Each layer is meticulously explored to offer a profound comprehension of its evolution. Structured and summarized approaches are presented here.

Index Terms—Diffusion Model, Deep Generative Model, Diffusion Algorithm, Diffusion Applications.

✦

1 INTRODUCTION

How can we enable machines to possess human-like imagination? Deep generative models, including Variational Autoencoders (VAEs) [1, 2], Energy-Based Models (EBMs) [3, 4], Generative Adversarial Networks (GANs) [5, 6], normalizing flows (NFs) [7, 8], and diffusion models [9–11], have demonstrated remarkable potential in generating realistic samples. Within this survey, our central emphasis lies on diffusion models, which epitomize the forefront of advancements within this domain. These models effectively surmount the obstacles entailed in aligning posterior distributions within VAEs, mitigating the instability inherent in adversarial objectives of GANs, addressing the computational burdens associated with Markov Chain Monte Carlo (MCMC) methods during training in EBMs, and enforcing network constraints akin to NFs. Consequently, diffusion models have garnered significant attention in various domains, including computer vision [12–14], natural language processing [15, 16], time series [17, 18], audio processing [19, 20], graph generation [21, 22], and bioinformatics [23, 24]. Despite the significant interest and attention garnered by diffusion models, there remains a notable absence of an up-to-date and comprehensive taxonomy and analysis encapsulating the research advancements made in this field. Diffusion models encompass two interconnected processes: a predefined forward process that maps the data distribution to a simpler prior distribution, often a Gaussian, and a

corresponding reverse process that employs a trained neural network to gradually reverse the effects of the forward process by simulating Ordinary or Stochastic Differential Equations (ODE/SDE) [11, 25]. The forward process resembles a straightforward Brownian motion with time-varying coefficients [25]. The neural network is trained to estimate the score function utilizing the denoising score-matching objective [26]. Consequently, diffusion models offer a more stable training objective compared to the adversarial objective employed in GANs and demonstrate superior generation quality when compared to VAEs, EBMs, and NFs [11, 27].

However, it is imperative to acknowledge that diffusion models inherently entail a more time-intensive sampling process compared to GANs and VAEs. This stems from the iterative transformation of the prior distribution into a complex data distribution through the utilization of ODE/SDE (Ordinary/Stochastic Differential Equations) or Markov processes, necessitating a substantial number of function evaluations during the reverse process. Furthermore, additional challenges encompass the instability of the reverse process, the computational demands and constraints associated with training in high-dimensional Euclidean space, and the intricacies involved in likelihood optimization. In response to these challenges, researchers have put forth diverse solutions. For instance, advanced ODE/SDE solvers have been proposed to expedite the sampling process [28–30], while model distillation strategies have been employed [14, 31] to achieve the same goal. Furthermore, novel forward processes have been introduced to enhance sampling stability [32–34] or facilitate dimensionality reduction [35, 36]. Additionally, a recent line of research endeavors to leverage diffusion models for efficiently bridging arbitrary distributions [37, 38]. To provide a systematic overview, we categorize these advancements into four principal domains: **Sampling Acceleration**, **Diffusion Process Design**, **Likelihood Optimization**, and **Bridging Distributions**. Moreover, this survey will comprehensively examine the diverse applications of diffusion models across different domains, including computer vision, natural language processing, healthcare, and beyond. It will explore

- H. Cao is with the Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong, China, also with Zhejiang Lab, Hangzhou, China. Email: 1155141481@link.cuhk.edu.hk.
- C. Tan and Z. Gao are 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.

how diffusion models have been successfully applied to tasks such as **Image Synthesis**, **Video Generation**, **3D Generation**, **Medical Analysis**, <mark>Text Generation</mark>, **Speech Synthesis**, **Time Series Generation**, **Molecule Design**, and <mark>Graph Generation</mark> . By highlighting these applications, we aim to showcase the practical utility and transformative potential of diffusion models in real-world scenarios.

The remaining sections are structured as follows: Section 2 provides an overview of the fundamental formulations and theories of diffusion models. Section 3 explores the algorithmic improvements made in the field, while Section 4 presents categorized applications based on the generation mechanism. Finally, in Section 5, we summarize the content, discuss connections with other diffusion surveys, and identify limitations and future directions for diffusion models.

2 PRELIMINARIES

2.1 Notions and Definitions

2.1.1 Time and States

In diffusion models, the process unfolds over a timeline, which can be either continuous or discrete. The states within this timeline represent data distributions that describe the model's progression. Noise is incrementally added to the initial distribution, denoted as the starting state x_0, which is sampled from the data distribution p_0. The distribution gradually converges towards a known noise distribution, typically Gaussian, referred to as the prior state x_T. The states between the starting and prior states are intermediate states x_t, each associated with a marginal distribution p_t. This enables diffusion models to explore the evolution of the data distribution over time and generate samples that approximate the prior state x_T. The progression occurs through a sequence of intermediate states, with each state mapping to a specific time point in the diffusion process.

2.1.2 Forward / Reverse Process, and Transition Kernel

In diffusion models, the forward process F transforms the starting state into prior Gaussian noise, while the reverse process R denoises the prior state back to the starting state using transition kernels. Following DDPM [10], the discrete formulation for diffusion model is generalized by defining transition kernels among diffusion and denoising processes.

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

$$R(x_T, \sigma) = R_1(x_1, \sigma_1) \cdots \circ R_t(x_t, \sigma_t) \cdots \circ R_T(x_T, \sigma_T) \quad (2)$$

where F_t and R_t denote the forward and reverse transition kernels at time t, with the noise scale σ_t from noise set σ. Unlike normalizing flow models, diffusion models incorporate variable noise, gradually refining the distribution for a controlled shift towards the target distribution, which provides a wider generation space and controllable generation. The discrete framework provides a discrete-time approximation of the continuous diffusion process, allowing for practical implementation and efficient computation.

2.1.3 From discrete to continuous

When the perturbation kernel is sufficiently small, the discrete processes (Eq. (1) and Eq. (2)) can be generalized to continuous processes. [11] showed that diffusion models with discrete Markov chains [9, 10] can be incorporated into a continuous Stochastic Differential Equation (SDE) framework, where the generative process reverses a fixed forward diffusion process. A reserve ODE marginally equivalent to the reverse SDE has also been derived [11]. 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

In this sub-section, we introduce three foundation formulations **Denoised Diffusion Probabilistic Models**, **Score SDE Formulation**, and **Conditional Diffusion Probabilistic Models**, establishing the connection to Section 3 and Section 4. The following math formulation can be regarded as specific form of the general framework in Section 2.1.2.

2.2.1 Denoised Diffusion Probabilistic Models (DDPM)

DDPM Forward Process: In the DDPM framework, a sequence of noise coefficients $\beta_1, \beta_2, ..., \beta_T$ for Markov transition kernels are chosen, following patterns like constant, linear, or cosine schedules, leading to improved sample quality. According to [10], the forward steps are defined as:

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

By the composition of forward transition kernels from x_0 to x_T, the Forward Diffusion Process, which adds Gaussian noises to the data through the Markov kernel $q(x_t|x_{t-1})$:

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

DDPM Reverse Process: The Reverse Process, with learnable Gaussian kernels parameterized by θ, is defined as:

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

μ_θ and Σ_θ are learnable mean and variance of the reverse Gaussian kernels, determined by reverse-step distribution p_θ. The sequence of reverse steps from x_T to x_0 is:

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

DDPM aims to approximate the data distribution p_0 by the joint probability distribution $p_\theta(x_0) = \int p_\theta(x_{0:T}) dx_{1:T}$.

Diffusion Training Objective: The training objective is equivalent to minimizing the variational bound on the negative log-likelihood by introducing KL-Divergence D_{KL}:

$$\mathbb{E}\left[-\log p_\theta(x_0)\right] \le \mathbb{E}_q[\underbrace{D_{KL}(q(x_T \mid x_0) \| p(x_T))}_{L_T}$$

$$+ \sum_{t>1} \underbrace{D_{KL}(q(x_{t-1} \mid x_t, x_0) \| p_\theta(x_{t-1} \mid x_t))}_{L_{t-1}}$$

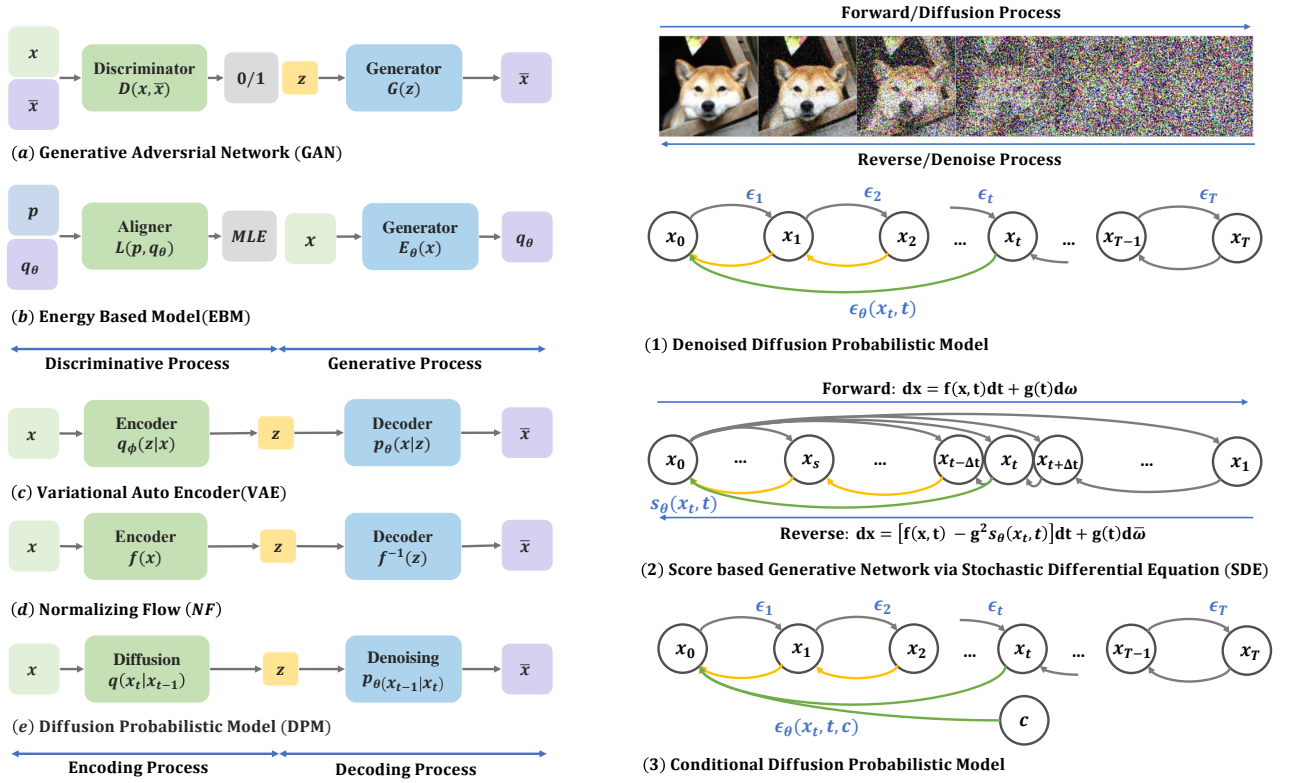$$- \underbrace{\log p_\theta(x_0 \mid x_1)}_{L_0}]$$

$$(7)$$

Fig. 1. **Overview of Diffusion Models. (Left)** Briefing of Generative Models. GAN and EBM firstly utilize generators to capture data distribution. By sampling from training data, the estimated distributions are corrected based on real distributions. The distribution capturing processes as well as discrimination criteria differ. VAE, NF, and DPM directly project real distribution into pre-defined distributions by encoding process. Instances are obtained by decoding the samples from pre-defined distributions. They apply distinct pre-defined distributions z and encoding & decoding processes. **(Right)** Simplified Formulations of Diffusion Models. General procedures follow the top-right figure. Data distributions are diffused into random Gaussian noise, and are reversed by denoising. **(1) DDPM**(Section 2.2.1) achieves step-by-step diffusion and denoising processes along discrete timeline. **(2) SDE**(Section 2.2.2) establishes continuous timeline, achieving inter-state translation by function-based Stochastic Differential Equations (SDE). **(3) CDPM**(Section 2.3) employs condition c in each sampling step of DPM to achieve controllable generation.

where L_T and L_0 denote the prior loss and the reconstruction loss; $L_{1:T-1}$ denoted the divergence sum between the posterior of the forward and reverse steps at the same time. Simplifying L_{t-1}, we obtain the simplified training objective named L_{simple} based on the posterior $q(x_{t-1}|x_t, x_0)$ as:

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

where $\tilde{\beta}_t$ depends on β_t. Keeping above parameterization and reparameterizing x_t as $x_t(x_0, \sigma)$, L_{t-1} is expressed as expectations of ℓ_2-loss between two mean coefficients:

$$L_{t-1} = \mathbb{E}_q\left[\frac{1}{2\sigma_t^2}\left\|\tilde{\mu}_t\left(x_t, x_0\right) - \mu_\theta\left(x_t, t\right)\right\|^2\right] + C \quad (9)$$

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

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

Most diffusion models use the DDPM training strategy. However, Improved DDPM proposes combining $Lsimple$ with other objectives. After training, the prediction network ϵ_θ is used in the reverse process for ancestral sampling.

2.2.2 Score SDE Formulation

Score SDE [11] extends the discrete-time scheme in DDPM to a continuous-time framework based on the stochastic differential equation. Also, it proposes additional deterministic sampling frameworks based on ODE formulation.

Forward SDE: ScoreSDE [11] connected continuous diffusion process and stochastic differential equations. The reverse process is linked with the solution to Itô SDE [39] composed of a drift term for mean shift and a Brownian motion for additional noising:

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

where w_t is the standard Wiener process, $f(\cdot, t)$ is $x(t)$'s drift coefficient, and $g(\cdot)$ is a simplified diffusion coefficient independent on x. $p_t(x)$ and p_T denote the marginal and prior distributions respectively. If coefficients are piecewise continuous, the forward SDE equation has a unique solution [40]. Two types of forward processes are proposed: Variation Preserving (VP) and Variation Explosion (VE) SDE. VP corresponds to the continuous extension of the

DDPM framework:

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

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

Reversed SDE: The sampling of diffusion models is done via a corresponding reverse-time SDE of the forward process (Eq. (11)) [41]:

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

where w_t is the standard Wiener process, $f(\cdot, t)$ is $x(t)$'s drift coefficient, and $g(\cdot)$ is a simplified diffusion coefficient. $p_t(x)$ and p_T are the marginal and prior distributions. If coefficients are piece-wise continuous, a unique solution exists for the forward SDE equation. [42]:

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

where x_0 is sampled from distribution p_0 and $\lambda(t)$ is the positive weighting function to keep the time-dependent loss at the same magnitude [11]. $q(x_t|x_0)$ 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 ϵ_{θ} in the DDPM objective (Eq. (10)). [43] 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 [11] supports the deterministic process which shares the same marginal probability density with SDE. Inspired by Maoutsa *et al.* [44] and Chen *et al.* [45], any type of diffusion process can be derived into a special form of ODE. The corresponding probability flow ODE of Eq. (12) is

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

In contrast to SDE, probability flow ODE can be solved with larger step sizes as they have no randomness. Thus, several works such as PNDMs [46] and DPM-Solver [47] obtain faster sampling speed based on advanced ODE solvers.

2.3 Conditional Diffusion Probabilistic Models

Diffusion models are versatile, capable of generating data samples from both unconditional p_0 and conditional $p_0(x|c)$ distributions, with c as a given condition such as a class label or text linked to data x[36]. The score network $s_{\theta}(x, t, c)$ integrates this condition during training. Various sampling algorithms, including classifier-free guidance[48] and classifier guidance [27], are designed for conditional generation.

Labeled Conditions Sampling with labeled conditions guides each sampling step's gradient. It typically requires an additional classifier with a UNet Encoder architecture to generate condition gradients for specific labels, which can be text, categorical, binary, or extracted features [12, 27, 28, 49–55]. The method, first presented by [27], underpins current conditional sampling techniques.

Unlabeled Conditions Unlabeled condition sampling uses self-information for guidance, often applied in a self-supervised manner [56, 57]. It is commonly used in denoising [58], paint-to-image [59], and inpainting tasks [17].

3 ALGORITHM IMPROVEMENT

Despite the high-quality generation of diffusion models across diverse data modalities, their real-world application could be improved. They necessitate a slow iterative sampling process, unlike other generative models like GANs and VAEs, and their forward process operates in high-dimensional pixel space. This section highlights four recent developments for enhancing diffusion models: **(1) Sampling Acceleration techniques** (Section 3.1) to speed up the standard ODE/SDE simulation; **(2) New Forward Processes** (Section 3.2) for improved Brownian motion in pixel space; **(3) Likelihood Optimization techniques** (Section 3.3) to enhance the diffusion ODE likelihood; **(4) Bridging Distribution techniques** (Section 3.4) that utilize diffusion model concepts to connect two distinct distributions.

3.1 Sampling Acceleration

Despite their high-fidelity generation, the practical utility of diffusion models is limited by their slow sampling speed. This section briefly overviews four advanced techniques to enhance sampling speed: distillations, training schedule optimization, training-free acceleration, and integration of diffusion models with faster generative models.
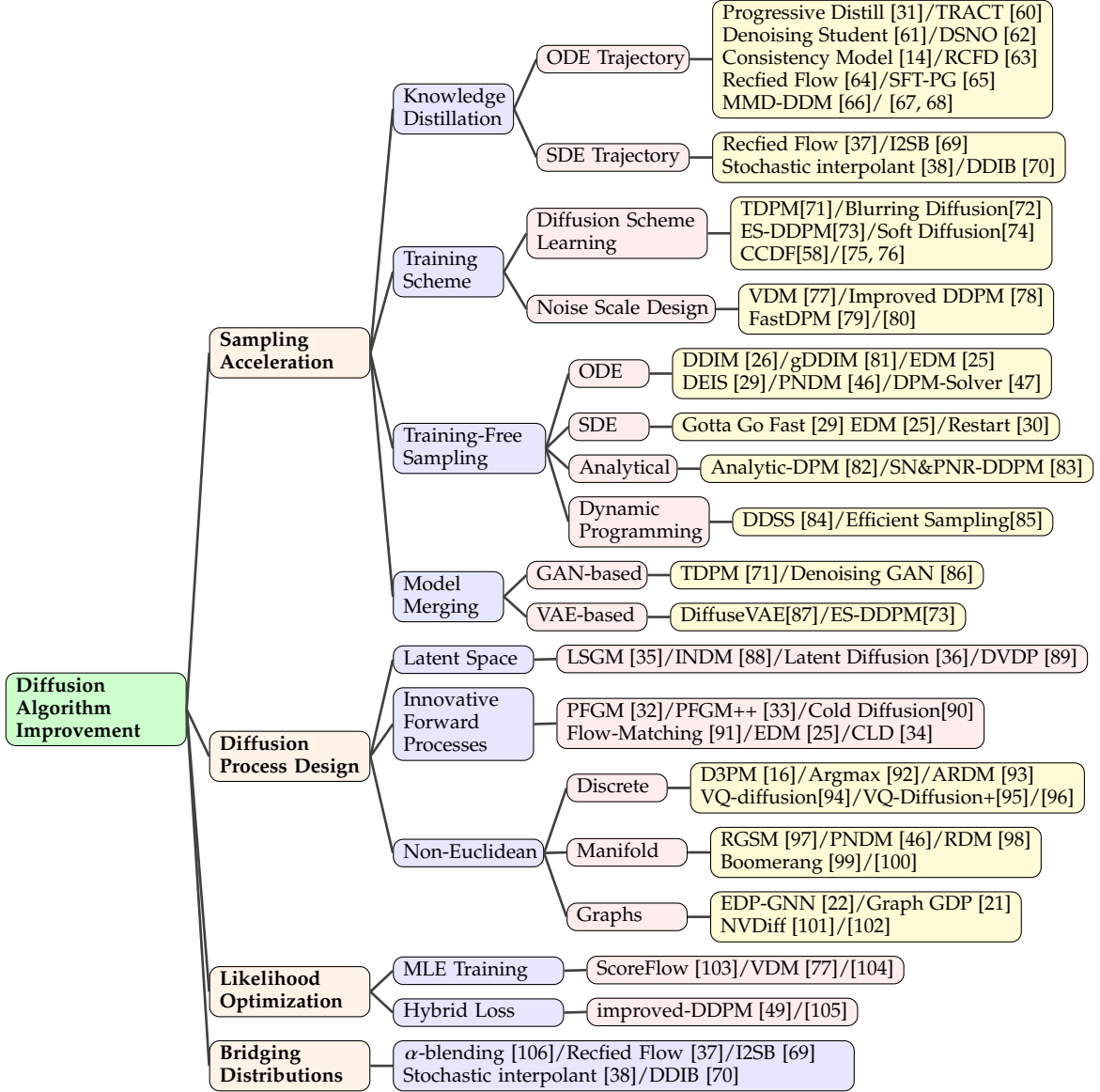
3.1.1 Knowledge Distillation

Knowledge distillation, a technique for transferring "knowledge" from larger to simpler models, is becoming increasingly popular [107, 108]. In diffusion models, the goal is to produce samples using fewer steps or smaller networks by aligning and minimizing the discrepancy between original and generated samples. Viewed as trajectory optimization across distributions, distillation offers optimal mappings for cost-effective and faster controllable generation.

ODE Trajectory Knowledge distillation from teacher to student models using ODE formulation parallels mapping prior distribution to target distribution via efficient paths across the distribution field. [31] first applied this principle to improve diffusion models by progressively distilling sampling trajectories, straightening latent mappings every two steps. TRACT [60], Denoising Student [61], and Consistency Models [14] extended this effect, increasing acceleration rates to 64 and 1024, by directly estimating clean data from noisy samples at time T. RFCD [63] enhances student model performance by aligning sample features during training.

Optimal trajectories can be obtained through optimal transport [109]. By minimizing transportation cost among distributions via flow matching, ReFlow [64] and [67] achieve one-step generation. DSNO [62] proposes a neural operator for direct temporal path modeling. Consistency Model [14], SFT-PG [65], and MMD-DDM [66] search ideal trajectories using LPIPS, IPA, and MMD, respectively.

SDE Trajectory Distilling stochastic trajectories is still challenging. Few works are proposed (referred to Section 3.4).

TABLE 1
Taxonomy of Improvements on Diffusion Algorithms (Section 3)

Diffusion Algorithm Improvement

- **Sampling Acceleration**
  - Knowledge Distillation
    - ODE Trajectory: Progressive Distill [31]/TRACT [60] / Denoising Student [61]/DSNO [62] / Consistency Model [14]/RCFD [63] / Recfied Flow [64]/SFT-PG [65] / MMD-DDM [66]/ [67, 68]
    - SDE Trajectory: Recfied Flow [37]/I2SB [69] / Stochastic interpolant [38]/DDIB [70]
  - Training Scheme
    - Diffusion Scheme Learning: TDPM[71]/Blurring Diffusion[72] / ES-DDPM[73]/Soft Diffusion[74] / CCDF[58]/[75, 76]
    - Noise Scale Design: VDM [77]/Improved DDPM [78] / FastDPM [79]/[80]
  - Training-Free Sampling
    - ODE: DDIM [26]/gDDIM [81]/EDM [25] / DEIS [29]/PNDM [46]/DPM-Solver [47]
    - SDE: Gotta Go Fast [29] EDM [25]/Restart [30]
    - Analytical: Analytic-DPM [82]/SN&PNR-DDPM [83]
    - Dynamic Programming: DDSS [84]/Efficient Sampling[85]
  - Model Merging
    - GAN-based: TDPM [71]/Denoising GAN [86]
    - VAE-based: DiffuseVAE[87]/ES-DDPM[73]
- **Diffusion Process Design**
  - Latent Space: LSGM [35]/INDM [88]/Latent Diffusion [36]/DVDP [89]
  - Innovative Forward Processes: PFGM [32]/PFGM++ [33]/Cold Diffusion[90] / Flow-Matching [91]/EDM [25]/CLD [34]
  - Non-Euclidean
    - Discrete: D3PM [16]/Argmax [92]/ARDM [93] / VQ-diffusion[94]/VQ-Diffusion+[95]/[96]
    - Manifold: RGSM [97]/PNDM [46]/RDM [98] / Boomerang [99]/[100]
    - Graphs: EDP-GNN [22]/Graph GDP [21] / NVDiff [101]/[102]
- **Likelihood Optimization**
  - MLE Training: ScoreFlow [103]/VDM [77]/[104]
  - Hybrid Loss: improved-DDPM [49]/[105]
- **Bridging Distributions**: α-blending [106]/Recfied Flow [37]/I2SB [69] / Stochastic interpolant [38]/DDIB [70]

3.1.2 Training Schedule

Improving the training schedule involves modifying traditional training settings, such as diffusion schemes and noise schemes, that are independent of sampling. Recent research has highlighted the crucial factors in training schemes that impact learning patterns and model performance. In this subsection, we categorize training enhancements into two main areas: diffusion scheme learning and noise scale design.

Diffusion Scheme Learning Diffusion models, which project data into latent spaces like Variational Autoencoders (VAEs), are more complex due to their higher expressiveness. Reverse decoding methods in these models can be divided into two approaches: encoding degree optimization and projecting approaches.

Encoding degree optimization methods, such as CCDF [58] and Franzese et al. [75], minimize the Evidence Lower Bound (ELBO) by treating the number of diffusion steps as a variable. Truncation, another approach, balances generation speed and sample fidelity by sampling from less diffused data in a one-step manner. TDPM [71] and ES DDPM [73] use truncation with GAN and CT [110]. Projecting approaches, like Soft diffusion [74] and blurring diffusion models [72], explore the diversity of diffusion kernels using linear corruptions such as blurring and masks.

Noise Scale Designing In traditional diffusion processes, each transition step is determined by injected noise, which is equivalent to a random walk on forward and reversed trajectories. Designing the noise scale can lead to reasonable generation and fast convergence. Unlike traditional DDPMs, existing methods treat the noise scale as a learnable parameter throughout the process.

Forward noise design methods like VDM [77] parameterize the noise scale as a signal-to-noise ratio, connecting it to training loss and model types. FastDPM [79] links noise design to ELBO optimization using discrete-time variables

or a variance scalar. For reverse noise design, improved DDPM [78] learns the reverse noise scale implicitly by training a hybrid loss, while San Roman *et al.* use a noise prediction network to update the reverse noise scale before ancestral sampling.

3.1.3 Training-Free Sampling

Training-free methods aim to leverage advanced samplers 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 methods, and dynamic programming.

ODE Acceleration [11] demonstrates 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 generate less discretization error than their stochastic counterparts [11, 30], most previous work on sampling acceleration has been ODE-centric. For instance, the widely-used sampler DDIM [26] can be regarded as a probability flow ODE [11]:

$$ d\bar{x}(t) = \epsilon_\theta^{(t)} \left(\frac{\bar{x}(t)}{\sqrt{\sigma^2 + 1}} \right) d\sigma(t) \qquad (15) $$

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 [29, 47] interpret DDIM as a product of applying an exponential integrator on the ODE of Variance Preserving (VP) diffusion [11]. Advanced ODE solvers have been utilized in methods such as PNDM [46], EDM [25], DEIS [29], gDDIM [81], and DPM-Solver [47]. For example, EDM employs Heun's 2nd order ODE solvers, and DEIS/DPM-solver improves upon DDIM by numerically approximating the score functions within each discretized time interval. These methods significantly accelerate the sampling speed (reducing the number of function evaluations, or NFE) compared to the original DDPM sampler while still yielding high-quality samples.

SDE Acceleration ODE-based samplers are faster but reach performance limits, while SDE-based samplers offer better sample quality despite being slower. Several works have focused on accelerating stochastic samplers' speed. Gotta Go Fast [111] uses adaptive step size for faster SDE sampling, while EDM [25] 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 [30] reveals that although ODE-samplers involve smaller discretization errors, the stochasticity in SDE helps to contract accumulated errors. This leads to the Restart Sampling algorithm [30], which blends the best aspects of both worlds. The sampling method alternates between adding significant noise by additional forward steps and strictly following a backward ODE, surpassing previous SDE and ODE samplers on standard benchmarks and the Stable Diffusion model [36], both in terms of speed and accuracy.

Analytical Method Existing training-free sampling methods treat 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 [82] and extended Analytic-DPM [83] jointly propose optimal reverse solutions under correction for each state. Analytical methods enjoy a theoretical guarantee for the approximation error, but they are limited to specific distributions due to their pre-assumptions.

Dynamic Programming Adjustment Dynamic programming (DP) achieves the traversal of all choices to find the optimized solution in a reduced time by using a memorization technique [112]. 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 [85, 113] take $O(T^2)$ of computational cost by optimizing the sum of ELBO losses.

3.1.4 Merging Diffusion and Other Generative Models

Diffusion models can be synergized with other generative models like Generative Adversarial Networks (GANs) or Variational Autoencoders (VAEs) to streamline the sampling process. For example, pristine data x_0 can be directly predicted through a VAE [87] or GAN [86] obtained from noisy samples during an intermediate phase of the diffusion sampling process. Moreover, a VAE [73] or GAN [71] can generate samples at intermediary diffusion time steps, which are then denoised by diffusion models until time $t = 0$ for faster time traversal.

3.2 Diffusion Process Design

The traditional forward process in diffusion models, often considered as Brownian motion in pixel space [10, 25], may be sub-optimal for generative modeling. Consequently, research efforts have been directed towards creating new diffusion processes that simplify and enhance the associated backward processes for neural networks. This path has bifurcated into developing latent spaces designed for diffusion models (Section 3.2.1) and replacing the conventional forward process with improved versions in pixel space (Section 3.2.2). Special attention is also given to diffusion processes specifically tailored for non-Euclidean spaces like manifolds, discrete spaces, functional spaces, and graphs (Section 3.2.3).

3.2.1 Latent Space

Researchers explore training diffusion models in a learned latent space to enhance neural networks and establish a more direct backward process. This approach is exemplified by LSGM [35] and INDM [114], which jointly train a diffusion model and a VAE or normalizing flow model. Both models share a common objective, the weighted denoising score-matching loss (L_{DSM} in Eq. (13)), to optimize the pair of encoder-decoder and diffusion model.

$$ L := L_{Enc}(z_0|x) + L_{Dec}(x|z_0) + L_{DSM}\left(\left(\{z_t\}_{t=0}^T\right)\right) \qquad (16) $$

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 [35, 114]. This leads to a latent space that is simpler to learn from and to sample. Influential work such as Stable Diffusion [36] separates the

process into two stages: learning the latent space of VAE and training diffusion models with text as conditional inputs. On a different note, DVDP [89] decomposes the pixel space into orthogonal components and dynamically adjusts the attenuation of each component during image perturbation, akin to dynamic image down-sampling and up-sampling.

3.2.2 Emerging Forward Processes

Latent space diffusion has advantages but also adds complexity and computational load to the framework. To address this issue, contemporary research explores forward process design for more robust and rddicient generative models. For instance, the Poisson Field Generative Model (PFGM)[32] treats data as electric charges in an augmented space, guiding a simple distribution along electric field lines towards the data distribution. The forward process in this model is defined in the electric field lines' directions, exhibiting more robust backward sampling than diffusion models. The PFGM++ [33] extends PFGM with higher-dimensional augmented variables, and an interpolation between these models reveals an optimal point, leading to state-of-the-art image generation. PFGM and PFGM++ also find applications in antibody [115] and medical image [116] generation. Dockhorn et al. [34] introduced the Critically-Damped Langevin Diffusion (CLD) model, which incorporates "velocity" variables interacting through Hamiltonian dynamics. The model simplifies learning the score function of the conditional velocity distribution, compared to directly learning the data's score functions. Given the success of physics-inspired generative models such as diffusion models and PFGM, a recent work [117] provides a systematic method to transform physical processes into generative models.

Other research explores alternative corrupting processes. For instance, Cold Diffusion[90] uses arbitrary image transformations like blurring for the forward process, while [118] applies heat dissipation in pixel space. Furthermore, there are efforts to enhance training and sampling with advanced Gaussian perturbation kernels [25, 91].

3.2.3 Diffusion Models on non-Euclidean space

Discrete Space Deep generative models have made considerable strides in various domains, such as natural language processing [119, 120], multi-modal learning [49, 121], and AI for science [122, 123]. A key achievement is the processing of discrete data, including sentences, residues, atoms, and vector-quantized data. Diffusion models are commonly used in these applications, focusing on text, categorical data, and vector-quantized data. D3PM [16] defines the forward process in discrete space, processing data like text or atom type, using transition kernels Q_t:

$$q(x_t \mid x_{t-1}) = \text{Cat}(x_t; p = x_{t-1}Q_t) \qquad (17)$$

where Cat() denotes a categorical distribution. This approach has been extended for generating language text, segmentation maps, and lossless compression [92, 93].

For multi-modal problems such as text-to-image generation and text-to-3d generation, vector-quantized (VQ) data transforms data into codes, achieving excellent performance in autoregressive encoders [124]. Diffusion techniques were first applied to VQ data by [94], addressing the unidirectional bias and accumulation prediction error in VQ-VAE.

This core idea has been utilized in further text-to-image, text-to-pose, and text-to-multimodal works [95, 125–129]. The forward process is defined by the probability transition matrix Q and categorical representation vector v:

$$q(x_t \mid x_{t-1}) = v^\top(x_t) Q_t v(x_{t-1}) \qquad (18)$$

Manifold Data structures like images and videos typically inhabit Euclidean space. However, certain data in fields like robotics [130], geoscience [131], and protein modeling [132] are defined within a Riemannian manifold [133]. Standard Euclidean methods may not apply in this environment. To address this, recent methodologies such as RDM [98], RGSM [97], and Boomerang [99] have incorporated diffusion sampling into the Riemannian manifold, extending the score SDE framework [11]. Theoretical works [46, 100] provide further support for manifold sampling.

Graph Graph-based neural networks are gaining popularity due to their expressiveness in human pose [127], molecules [134], and proteins [135] [136]. Current methods apply diffusion theories to graphs. Approaches like EDP-GNN [22], Pan *et al.* [102], and GraphGDP [21] process graph data via adjacency matrices to capture permutation invariance. NVDiff [101] reconstructs node positions using reverse SDE. **Function** Dutordoir *et al.*, [137] introduced the first diffusion model sampling in functional space, capturing infinite-dimensional distributions via joint posterior sampling.

3.3 Likelihood Optimization

While diffusion models [10] optimize the ELBO to overcome the intractability of the log-likelihood, the likelihood optimization is ignored, which is challenging for continuous-time diffusion models [11]. Two approaches including MLE Training (Section 3.3.1) and hybrid loss (Section 3.3.2) are designed to enhance likelihood training.

3.3.1 MLE Training

Three concurrent works—ScoreFlow [103], VDM [77], and [104] 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 [103] demonstrates that under a particular weighting scheme, the DSM objective provides an upper bound on the negative log-likelihood. This finding enables a neural-network parameter-independent approximation of score-based MLE.

3.3.2 Hybrid Loss

Instead of solely relying on maximum likelihood training, certain approaches introduce hybrid loss designs to improve the model likelihood in DSM. One such approach is Improved DDPM [78], which proposes learning the variances of the reverse process using a simple reparameterization technique and a hybrid learning objective that combines the variational lower bound and DSM. Additionally, [105] demonstrates that incorporating high-order score-matching loss contributes to enhancing the log-likelihood.

3.4 Bridging Distributions

Diffusion models excel at transforming simple Gaussian distributions but face challenges when bridging arbitrary distributions, particularly in areas like image-to-image translation and cell distribution transportation. Various approaches have been proposed to tackle this issue. One approach, known as α-blending [106], involves iterative blending and deblending to create a deterministic bridge. Diffusion models are treated as special cases when one end distribution is Gaussian. Another approach is Rectified Flow [37], which incorporates additional steps to straighten the bridge. Other methods, such as the one proposed in [38], suggest constructing an ODE with general interpolant functions between two distributions. Besides, others explore the utilization of the Schrödinger Bridge [69] or Gaussian distributions as junctions to connect two diffusion ODEs [70].

4 APPLICATION

Benefiting from the powerful ability to generate realistic samples, diffusion models have been widely used in various fields. In real-world applications, the key to unlocking the power of diffusion models lies in fitting the diffusion process, denoising process, and conditional sampling to the natural of a wide range of data. Inspired by this idea, the applications of diffusion are summarised as **Image Generation**, **3D Generation**, **Video Generation**, **Medical Analysis**, **Text Generation**, **Time Series Generation**, **Audio Generation**, **Molecule Design**, and **Graph Generation**.

4.1 Image Generation

Diffusion models have achieved remarkable performance on image generation, either on traditional class-conditioned or unconditional generation [10, 27, 138], or on more complicated text or image condition [36, 143], or their combinations [142]. Our discussion henceforth will concentrate on application settings that mimic real-world scenarios, categorizing applications according to the conditional inputs.

4.1.1 Text condition

Diffusion models demonstrate exceptional performance in text-to-image generation, capable of creating not only photorealistic images but also samples that closely adhere to user-provided textual inputs. Remarkable examples include Imagen [138], Stable Diffusion [36] and DALL-E 2 [121]. Built on top of existing diffusion architectures, these methods add a cross-attention layer to inject the sequence of text embeddings into the diffusion models. The experimental results show that such conditioning mechanism effectively blends the text information into the generated images.

In addition, the cross-attention conditional mechanism enables many training-free image editing by utilizing and manipulating the keys, values, or attention matrices in the cross-attention layers. For example, [139] changes the concepts in source images by swapping or adding new feature maps into the output of the cross-attention layers; [140] enables the customization of a new concept by learning a new text embedding as the input to the cross-attention layers. [141] enforces the cross-attention to attend to all subject tokens in the text prompt and enlarge their activations, encouraging the model to faithfully generate all subjects described in the text prompt.

4.1.2 Image condition

In addition to textual conditions, diffusion models also support image conditions, such as images to be edited, depth maps, or human skeletons, as conditional inputs. The underlying concept remains the same, which involves incorporating encoded image features into the diffusion backbone. The work by [142] introduces encoded features from the source image into the first convolutional layer to enable image conditioning, thereby allowing for image-to-image editing with text prompts. Similarly, [143] utilizes depth maps, Canny edges, or human skeletons to control the spatial layout of the generated images.

4.2 3D Generation

Broadly, there are two primary approaches to 3D generation by diffusion models. The first approach focuses on training these models directly with 3D data. However, due to the limited availability of 3D data, the second approach emphasizes generating 3D content by 2D diffusion priors.

4.2.1 3D Data Condition

Given the diverse range of 3D representations, such as NeRF, point clouds, voxels, Gaussian splatting, and more, diffusion models have been effectively applied across these various 3D representations. For instance, works such as [147, 150, 151] directly generate point clouds for 3D objects. In order to achieve efficient sampling, a hybrid point-voxel representation was employed for shape processing in PDR [144], introducing a new paradigm for point cloud completion. Building upon this research, Point-E [146] further incorporates image synthesis as an additional conditional input for point cloud diffusion models.

In contrast, Shape-E [145] utilizes diffusion models for the NeRF representation of 3D objects. Zero-1-to-3 [148] takes a different approach by training viewpoint-conditioned diffusion models to enable novel view synthesis. It then optimizes a NeRF based on the generated samples from different camera viewpoints. Based on this work, [149] further extends Zero-1-to-3 by incorporating a pose estimation stage.
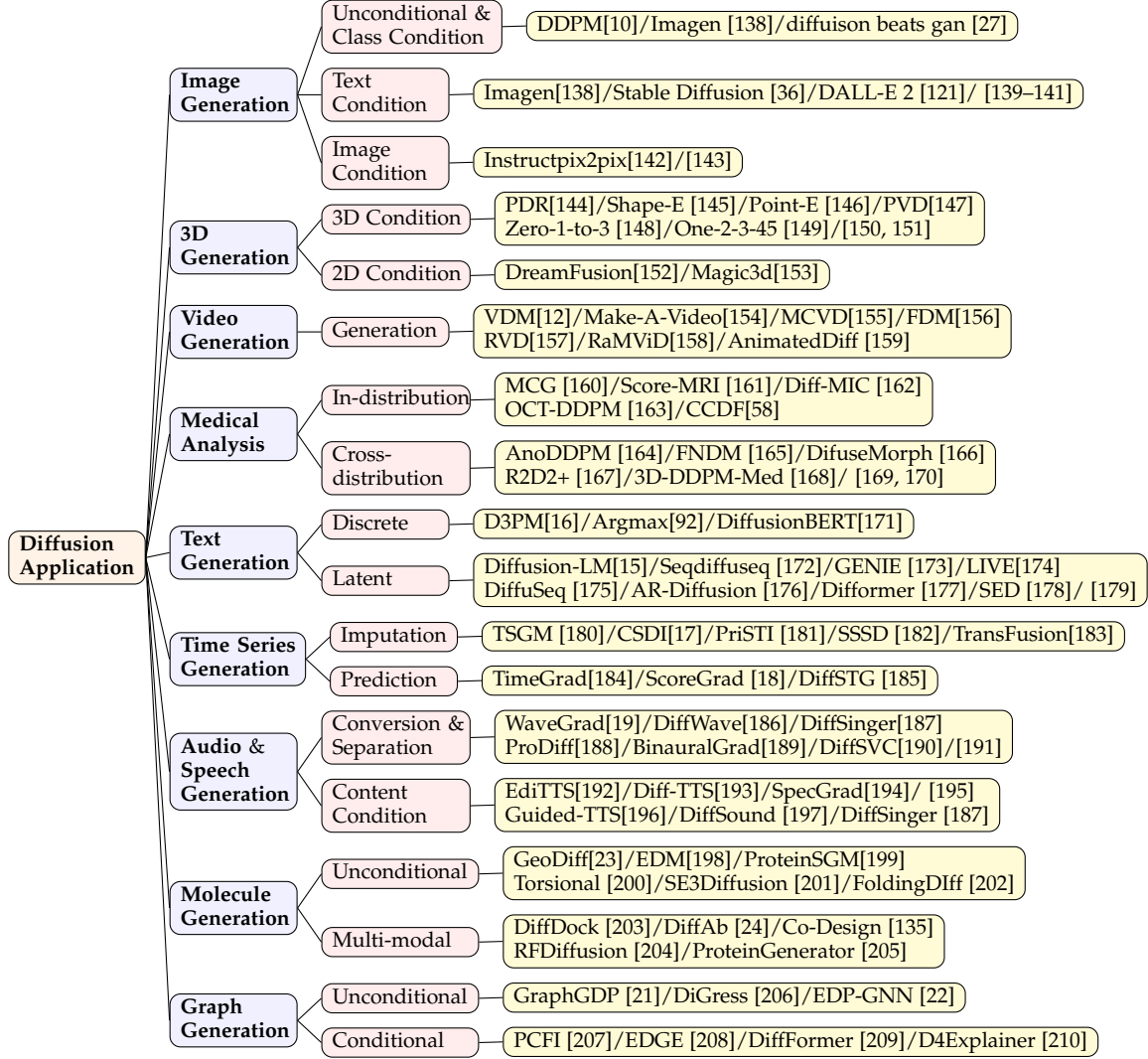
4.2.2 2D Diffusion prior

Another interesting line of works is aiming to *distill 3D from a 2D diffusion model*. Dreamfusion [152] smartly use the score distillation sampling (SDS) objective to distill a NeRF from a pre-trained text-to-image models. They optimize a randomly initialized NeRF via gradient descent such that the rendered images from different angles achieve low loss. [153] extends DreamFusion to a two-stage coarse-to-fine optimization framework, to accelerate the generation process.

4.3 Video Generation

Video diffusion models augment the 2D diffusion models for image generation with an additional time axis. The general idea is to add a temporal layer to explicitly model the cross-frame dependence in existing 2D diffusion structures. Representative works include Video Diffusion Models [12], Make-A-Video [154], AnimatedDiff [159], RVD [157], FDM [156], MCVD [155]. RaMViD [158] extends image diffusion models to videos with 3D convolutional neural networks and designed a conditioning technique for video prediction, infilling, and up-sampling.

TABLE 2
Classification of Diffusion-based model Applications(Section 4)

Image Generation	Unconditional & Class Condition	DDPM[10]/Imagen [138]/diffuison beats gan [27]
	Text Condition	Imagen[138]/Stable Diffusion [36]/DALL-E 2 [121]/ [139–141]
	Image Condition	Instructpix2pix[142]/[143]
3D Generation	3D Condition	PDR[144]/Shape-E [145]/Point-E [146]/PVD[147] Zero-1-to-3 [148]/One-2-3-45 [149]/[150, 151]
	2D Condition	DreamFusion[152]/Magic3d[153]
Video Generation	Generation	VDM[12]/Make-A-Video[154]/MCVD[155]/FDM[156] RVD[157]/RaMViD[158]/AnimatedDiff [159]
Medical Analysis	In-distribution	MCG [160]/Score-MRI [161]/Diff-MIC [162] OCT-DDPM [163]/CCDF[58]
	Cross-distribution	AnoDDPM [164]/FNDM [165]/DifuseMorph [166] R2D2+ [167]/3D-DDPM-Med [168]/ [169, 170]
Text Generation	Discrete	D3PM[16]/Argmax[92]/DiffusionBERT[171]
	Latent	Diffusion-LM[15]/Seqdiffuseq [172]/GENIE [173]/LIVE[174] DiffuSeq [175]/AR-Diffusion [176]/Difformer [177]/SED [178]/ [179]
Time Series Generation	Imputation	TSGM [180]/CSDI[17]/PriSTI [181]/SSSD [182]/TransFusion[183]
	Prediction	TimeGrad[184]/ScoreGrad [18]/DiffSTG [185]
Audio & Speech Generation	Conversion & Separation	WaveGrad[19]/DiffWave[186]/DiffSinger[187] ProDiff[188]/BinauralGrad[189]/DiffSVC[190]/[191]
	Content Condition	EdiTTS[192]/Diff-TTS[193]/SpecGrad[194]/ [195] Guided-TTS[196]/DiffSound [197]/DiffSinger [187]
Molecule Generation	Unconditional	GeoDiff[23]/EDM[198]/ProteinSGM[199] Torsional [200]/SE3Diffusion [201]/FoldingDIff [202]
	Multi-modal	DiffDock [203]/DiffAb [24]/Co-Design [135] RFDiffusion [204]/ProteinGenerator [205]
Graph Generation	Unconditional	GraphGDP [21]/DiGress [206]/EDP-GNN [22]
	Conditional	PCFI [207]/EDGE [208]/DiffFormer [209]/D4Explainer [210]

(Diffusion Application)

4.4 Medical Analysis

Diffusion models provide a solution to the challenges encountered in medical analysis, where acquiring large-scale, high-quality annotated datasets is challenging. These models demonstrate exceptional performance in tasks related to in-distribution analysis and cross-distribution generation.

4.4.1 In-distribution Analysis

Diffusion models are effective in various medical imaging tasks, leveraging their ability to accurately capture medical images with strong prior information. They have been successfully used in super-resolution [58, 160], classification [162], and noise robustness [161, 163]. For example, Score-MRI [161] accelerates MRI reconstruction using pixel guidance SDE sampling, while Diff-MIC [162] achieves accurate classification across multiple modalities with Dual-granularity guidance and Maximum-Mean Discrepancy. Additionally, MCG [160] proposes manifold correction during sampling for CT super-resolution, reducing errors and improving acceleration.

4.4.2 Cross-distribution Generation

Multimodal guidance has significantly improved generative capabilities in medical analysis. By integrating class-specific guidance [164, 165] and pixel-level guidance [165, 166, 169], unconditional denoising networks can perform image translation across different types of scarce images, including high-quality format images, healthy images, and unbiased images. Notable examples include FNDM [165], which enables accurate detection of brain anomalies through a non-Markovian framework with hybrid-condition guidance, and DiffuseMorph [166], which performs MR image registration using continuous diffusion sampling conditioned on moving and fixed image pairs. Moreover, there are promising methods for enriching training datasets with realistic medical images generated from a small number of high-quality samples [167, 168, 170]. For instance, Latent Diffusion Models trained on 31,740 samples have been used to synthesize a high-quality and semantically rich dataset with 100,000 instances, achieving an impressive FID score of 0.0076 [170].

4.5 Text Generation

Text generation plays a crucial role in bridging the gap between humans and advanced artificial intelligence by producing natural and coherent language. Autoregressive language models generate text sequentially, ensuring high semantic coherence but slower generation speed [211]. On the other hand, diffusion models enable parallel text generation, offering faster speed but relatively weaker semantic coherence [212, 213]. Two primary approaches, namely **Discrete Generation** and **Latent Generation**, are commonly used to address the challenge of generating discrete tokens.

4.5.1 Discrete Generation

Discrete generation approaches involves models taking discrete words as input and utilize advanced techniques, parameterization, and pre-trained models. Pioneering the connection between diffusion models and discrete generation, typical works including D3PM [16] and Argmax [92] treat words as categorical vectors. They establish forward and backward processes using a discrete transition matrix, considering the data to be generated as a stationary distribution. DiffusionBERT [171] combines diffusion models with pre-trained language models, showcasing improved text generation performance. Moreover, it introduces a novel noise schedule and explores the incorporation of time steps into BERT for reverse diffusion processes.

4.5.2 Latent Generation

The second approach focuses on generating text in the latent space of tokens, capturing the continuous nature of the diffusion process. It incorporates enhanced loss functions [172, 176, 177], diverse generation types [175, 178], and advanced model architectures [15, 173]. For instance, LM-Diffusion [15] introduces transformer-based graphical models for controllable generation, demonstrating superior performance in various text generation tasks. GENIE [173] presents a large-scale language model based on the diffusion framework, incorporating a novel Continuous Paragraph Denoise (CPD) loss for improved denoising and paragraph-level coherence. It showcases the potential of diffusion-based decoders for text generation and provides a strong foundation for future research. In addition to advanced conditional sampling, token-level capturing, and post-refinement, diffusion models in NLP are expected to enhance the modeling of embedding space [178, 213], establish connections with large pre-trained language models, and support cross-modality generations [174, 179, 212].

4.6 Time Series Generation

Accurate time series modeling is crucial for trend prediction, decision making, and real-time analysis. The diffusion model enhances this process with modules for time series data, enabling superior analysis and diverse generation [214]. Prior conditions can be categorized into inpainting tasks and prediction tasks based on different types of masking strategies. In inpainting tasks, observed states are used as prior conditions [17, 180–182], combined with context-based modules. CSDI proposed a self-supervised training framework based on bidirectional CNN modules, achieving substantial improvement in continuous generation of healthcare and environmental data [17]. For prediction tasks, prior states are transformed into user-defined features and latent embeddings, serving as self-conditions [18, 184, 185]. Combined with temporal-spatial modules, such as Graph UNet and RNN, DiffSTG and TimeGrad successfully achieve spatio-temporal probabilistic learning for time series [184, 185]. The success of time series generation hinges on the accurate modeling of time-dependent series and the incorporation of robust self-conditional guidance during sampling. These aspects point towards promising future advancements in the field [183, 185, 214].

4.7 Audio Generation

Synthesizing high-quality simulated speech has diverse applications in music composition, virtual reality, game development, and voice assistants, offering personalized and immersive audio experiences and improving human-computer interaction. Diffusion models, well-suited for handling the unique characteristics of audio data, utilize strong priors and effectively manage high-dimensional, time-dependent information. Speech generation relies on hybrid conditions, combining text and control tags to achieve specific semantics or sound features. Techniques such as WaveGrad [19], DiffSinger [187], and DiffSVC [190] use Mel-Spectrogram as conditional guidance, while BinualGrad [189] separates audio based on mono audio input. These methods form the foundation for general waveform generation, and additional features like loudness, melody, and phonetic posteriorgram enable controllable style generation [187, 190, 191, 215]. Text-based and music-based generation, including text-to-speech and acoustic generation, rely on spectrogram features. Diffusion models incorporate text and rhythm as latent variables, leveraging spectrogram features and multi-view labels during sampling. Guided-TTS [196] and Diff-TTS [193] employ components such as a speaker text encoder, duration predictor, and phoneme classifier for content generation and speech style guidance. Guide-TTS2 [195] extends this approach to untranscribed speech generation using a classifier-free speaker encoder. Additional guidance factors include emotion, noise level, and music style [187, 192, 194, 197].

4.8 Molecule Design

Molecules, as the fundamental building blocks of life, play a vital role in numerous biological processes. The design of functional molecules has long been a challenging and enduring problem [216]. Generative models have revolutionized molecular design by offering a more efficient alternative to the traditional, laborious methods of enumeration and experimental validation. By characterizing specific modal distributions and functional domains, generative models can produce novel and effective drug molecule structures, expanding the possibilities in drug design [217]. In the realm of drug discovery, diffusion models efficiently explore vast compound spaces, accelerating the search for potential drug candidates. This enhances the overall efficiency of the drug discovery process and reveals intricate compound relationships that contribute to a better understanding of drug mechanisms. The patterns observed in molecule design can be broadly categorized into unconditional generation and cross-modal generation.

4.8.1 Unconditional Generation

Unconditional molecule generation focuses on generating molecular structures using diffusion models, which offer speed and high-quality modeling capabilities. One approach is to generate the positions of molecules in three-dimensional space, capturing the conformation of molecules in space [23, 135, 198]. However, this approach may result in lower diversity and larger errors due to the non-uniform and irregular distribution of molecular three-dimensional structures. Alternatively, generating models that capture multiple features and the distribution of structural features in high-dimensional space can lead to more diverse distributions and interpretability [199–202]. [218] further introduces a repulsion force between samples to promote the diveristy.

4.8.2 Cross-modal Generation

In molecular design, cross-modal generation focuses on incorporating functionality as a condition. Diffusion-based methods excel at incorporating conditions and leveraging denoising models based on different modalities to enhance modeling capabilities. Sequence-based cross-modal generation methods utilize protein sequences and multiple sequence alignments (MSA) sequences to train denoising models, incorporating specific protein structural information and functional labels to guide the generation [205, 219]. Structure-based cross-modal methods leverage prior knowledge from structure prediction models to assist in precisely guided generation, combining protein sequences and functional information [204]. Molecular docking and antibody design methods utilize the structural priors of target molecules to guide the docking process and identify favorable binding configurations [24, 203]. These methods leverage the prior knowledge of target structures to enhance the generation and obtain promising conformations.

4.9 Graph Generation

The motivation for employing diffusion models to generate graphs stems from the aim to study and simulate diverse real-world networks and propagation processes. By doing so, it offers improved understanding and problem-solving capabilities for real-world issues. This approach empowers researchers to delve into the interactions and information propagation mechanisms within intricate systems, unveiling concealed patterns and correlations, and enabling the prediction of potential outcomes. The applications of this method encompass social network analysis, analysis of biological neural systems, as well as the generation and evaluation of graph datasets. In Section 3.2.3, we have previously mentioned the conventional methods for graph generation, which involve generating an adjacency matrix or node features through discrete diffusion [21, 22, 206]. However, these unconditionally generated graphs have limited scalability and lack practical applicability. As a result, the predominant approach in graph generation revolves around generating graphs based on specific conditions and requirements. Diffusion-based graph generation, guided by various specified conditions, facilitates the expansion of graph scale, refinement of graph features, and resolution of dataset-specific issues. PCFI [207] leverages partial graph features and utilizes shortest path distances to predict

pseudo confidence, serving as a guiding factor in the generation process. EDGE [208] and DiffFormer [209], on the other hand, utilize node degree and energy constraints, respectively, as conditions to enable discrete and continuous generation of adjacency matrices and latent embeddings, thereby broadening the range of generation possibilities. Moreover, D4Explainer [210] incorporates the distribution of graph data as a condition and combines distribution loss and counterfactual loss to explore counterfactual instances.

5 CONCLUSIONS & DISCUSSIONS

5.1 Conclusions

The diffusion model becomes increasingly crucial to fields of deep learning. 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 fundamental theories, improved algorithms, and applications. We aspire for this survey to serve as a comprehensive guide for readers, elucidating the advancements in diffusion model enhancement and offering valuable insights into its practical applications.

5.2 Comparison to Existing Surveys

There is several existing surveys in the field of diffusion model, including general survey [157], survey in diverse fields including vision [220], language processing [212, 213], audio [221], time series [214], medical analysis [222], and bioinformatics [223, 224], and surveys in diverse data structures [225, 226]. Compared to existing surveys, we conduct a comprehensive review with insights to broadly include algorithm enhancement and wide-range applications. Furthermore, we keep up-to-date updates of this field to track the latest improvements and maintain our GitHub Repository monthly for long-lasting analysis.

5.3 Limitations and Future Directions

5.3.1 Challenges Under Data Limitations

Except low inference speed, diffusion models often encounter difficulties in discerning patterns and regularities from low-quality data, leading to their inability to generalize to new scenarios or datasets. Additionally, handling large-scale datasets presents computational challenges such as extended training times, excessive memory usage, or failure to converge to the desired states, thus limiting the model's scale and complexity. Moreover, biased or uneven data sampling can restrict the model's capacity to generate outputs that are adaptable across diverse domains or demographics.

5.3.2 Controllable Distribution-based Generation

Improving the model's ability to understand and generate samples within specific distributions is essential for achieving better generalization with limited data. By focusing on identifying patterns and correlations in the data, the model can generate samples that closely match the training data and meet specific requirements. This requires effective data sampling, utilization techniques, and optimizing model parameters and structures. Ultimately, this enhanced understanding allows for more controlled and precise generation, leading to improved generalization performance.

5.3.3 Advanced Multi-modal Generation Leveraging LLMs

The future direction of diffusion models entails the advancement of multi-modal generation through the integration of Large Language Models (LLMs). This integration enables the model to generate outputs that encompass a combination of text, images, and other modalities. By incorporating LLMs, the model's understanding of the interplay between different modalities is enhanced, resulting in outputs that are more diverse and realistic. Moreover, LLMs significantly enhance the efficiency of prompt-based generation by effectively leveraging the connections between text and other modalities. Additionally, LLMs act as a catalyst for improving the diffusion model's generation capabilities, expanding the range of domains in which it can generate modalities.

5.3.4 Integration with Machine Learning Fields

Combining diffusion models with traditional machine learning theories offers new opportunities for enhancing performance in various tasks. Semi-supervised learning is particularly valuable in addressing the inherent challenges of diffusion models, such as generalization, and enabling effective conditional generation even with limited data. By utilizing unlabeled data, it strengthens the diffusion models' ability to generalize and achieve desirable performance when generating samples under specific conditions.

Furthermore, reinforcement learning plays a crucial role by employing fine-tuning algorithms to provide targeted guidance during the model's sampling process. This guidance ensures focused exploration and facilitates controlled generation. Additionally, incorporating additional feedback enriches reinforcement learning, leading to improved controllable conditional generation capabilities of the model.

ACKNOWLEDGMENTS

REFERENCES

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

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

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

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

[5] 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)

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

[7] D. Rezende and S. Mohamed, "Variational inference with normalizing flows," in *ICML*, 2015. (document)

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

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

[10] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *NeurIPS*, 2020. 2.1.2, 2.1.3, 2.2.1, 3.2, 3.3, 4.1, 2, 1, 5, 7

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

[12] J. Ho, T. Salimans, A. Gritsenko, W. Chan, M. Norouzi, and D. J. Fleet, "Video diffusion models," 2022. (document), 2.3, 4.3, 2, 8

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

[14] Y. Song, P. Dhariwal, M. Chen, and I. Sutskever, "Consistency models," *ArXiv*, vol. abs/2303.01469, 2023. (document), 3.1.1, 1

[15] X. Li, J. Thickstun, I. Gulrajani, P. S. Liang, and T. B. Hashimoto, "Diffusion-lm improves controllable text generation," *NeurIPS*, vol. 35, pp. 4328–4343, 2022. (document), 2, 4.5.2, 8

[16] J. Austin, D. D. Johnson, J. Ho, D. Tarlow, and R. Van Den Berg, "Structured denoising diffusion models in discrete state-spaces," *NeurIPS*, vol. 34, pp. 17 981–17 993, 2021. (document), 1, 3.2.3, 2, 4.5.1, 5, 7, 8

[17] Y. Tashiro, J. Song, Y. Song, and S. Ermon, "Csdi: Conditional score-based diffusion models for probabilistic time series imputation," *NeurIPS*, vol. 34, pp. 24 804–24 816, 2021. (document), 2.3, 2, 4.6, 8

[18] T. Yan, H. Zhang, T. Zhou, Y. Zhan, and Y. Xia, "Scoregrad: Multivariate probabilistic time series forecasting with continuous energy-based generative models," *arXiv preprint arXiv:2106.10121*, 2021. (document), 2, 4.6

[19] N. Chen, Y. Zhang, H. Zen, R. J. Weiss, M. Norouzi, and W. Chan, "Wavegrad: Estimating gradients for waveform generation," in *ICLR*, 2020. (document), 2, 4.7, 8

[20] 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. (document), 8

[21] H. Huang, L. Sun, B. Du, Y. Fu, and W. Lv, "Graphgdp: Generative diffusion processes for permutation invariant graph generation," *arXiv:2212.01842*, 2022. (document), 1, 3.2.3, 2, 4.9

[22] 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. (document), 1, 3.2.3, 2, 4.9, 7

[23] 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. (document), 2, 4.8.1, 8

[24] S. Luo, Y. Su, X. Peng, S. Wang, J. Peng, and J. Ma, "Antigen-specific antibody design and optimization with diffusion-based generative models for protein structures," in *NeurIPS*, 2022. (document), 2, 4.8.2, 8

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

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

[27] P. Dhariwal and A. Nichol, "Diffusion models beat gans on image synthesis," *NeurIPS*, vol. 34, pp. 8780–8794, 2021. (document), 2.3, 4.1, 2, 4

[28] 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:2211.01095*, 2022. (document), 2.3

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

[30] 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

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

[32] 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

[33] 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

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

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

[36] 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, 4.1, 4.1.1, 2

[37] 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

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

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

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

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

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

[43] 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

[44] 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

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

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

[47] 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:2206.00927*, 2022. 2.2.2, 1, 3.1.3, 3, 4, 6, 7

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

[49] 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:2112.10741*, 2021. 2.3, 1, 3.2.3, 8

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

[51] 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 *CVPR*, 2022, pp. 11 502–11 511.

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

[53] 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:2211.01323*, 2022.

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

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

[56] V. T. Hu, D. W. Zhang, Y. M. Asano, G. J. Burgh-

outs, and C. G. Snoek, "Self-guided diffusion models," *arXiv:2210.06462*, 2022. 2.3, 6

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

[58] 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, 1, 3.1.2, 2, 4.4.1, 7

[59] 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

[60] 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:2303.04248*, 2023. 3.1.1, 1

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

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

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

[64] 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

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

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

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

[68] 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

[69] 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

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

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

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

[73] 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

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

[75] 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." 1, 3.1.2, 6, 7

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

[77] D. Kingma, T. Salimans, B. Poole, and J. Ho, "Variational diffusion models," *NeurIPS*, vol. 34, pp. 21 696–21 707, 2021. 1, 3.1.2, 3.3.1, 5, 7

[78] A. Q. Nichol and P. Dhariwal, "Improved denoising diffusion probabilistic models," in *ICML*, 2021. 1, 3.1.2, 3.3.2, B.3, 5, 6, 7

[79] Z. Kong and W. Ping, "On fast sampling of diffusion probabilistic models," *arXiv:2106.00132*, 2021. 1, 3.1.2, 6, 7

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

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

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

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

[84] D. Watson, W. Chan, J. Ho, and M. Norouzi, "Learning fast samplers for diffusion models by differentiating through sample quality," 2022. 1, 4, 6, 7

[85] 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

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

[87] 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

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

[89] 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:2211.16032*, 2022. 1, 3.2.1

[90] 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:2208.09392*, 2022. 1, 3.2.2

[91] 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

[92] E. Hoogeboom, D. Nielsen, P. Jaini, P. Forré, and M. Welling, "Argmax flows and multinomial diffusion: Learning categorical distributions," *NeurIPS*, vol. 34, pp. 12 454–12 465, 2021. 1, 3.2.3, 2, 4.5.1, 7, 8

[93] E. Hoogeboom, A. A. Gritsenko, J. Bastings, B. Poole, R. v. d. Berg, and T. Salimans, "Autoregressive diffu-

sion models," *arXiv:2110.02037*, 2021. 1, 3.2.3, 7

[94] 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

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

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

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

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

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

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

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

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

[103] Y. Song, C. Durkan, I. Murray, and S. Ermon, "Maximum likelihood training of score-based diffusion models," *NeurIPS*, vol. 34, pp. 1415–1428, 2021. 1, 3.3.1, 7

[104] C.-W. Huang, J. H. Lim, and A. C. Courville, "A variational perspective on diffusion-based generative models and score matching," *NeurIPS*, 2021. 1, 3.3.1, 7

[105] 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 *ICML*, 2022. 1, 3.3.2

[106] 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

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

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

[109] C. Villani, "Topics in optimal transportation," *Graduate Studies in Mathematics*, 2003. 3.1.1

[110] H. Zheng and M. Zhou, "Act: Asymptotic conditional transport," *arxiv*, 2020. 3.1.2

[111] A. Jolicoeur-Martineau, K. Li, R. Piché-Taillefer, T. Kachman, and I. Mitliagkas, "Gotta go fast when generating data with score-based models," *arXiv:2105.14080*, 2021. 3.1.3, 5, 6

[112] R. Bellman, "Dynamic programming," *Science*, 1966. 3.1.3

[113] 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

[114] 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

[115] 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 MLDD workshop*, 2023. 3.2.2

[116] 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

[117] 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

[118] S. Rissanen, M. Heinonen, and A. Solin, "Generative modelling with inverse heat dissipation," *ArXiv*, vol. abs/2206.13397, 2022. 3.2.2

[119] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *NeurIPS*, 2017. 3.2.3

[120] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv:1810.04805*, 2018. 3.2.3

[121] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, "Hierarchical text-conditional image generation with clip latents," *arXiv:2204.06125*, 2022. 3.2.3, 4.1.1, 2

[122] 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

[123] S. Ovchinnikov and P.-S. Huang, "Structure-based protein design with deep learning," *Current opinion in chemical biology*, 2021. 3.2.3

[124] A. Van Den Oord, O. Vinyals *et al.*, "Neural discrete representation learning," *NeurIPS*, vol. 30, 2017. 3.2.3

[125] M. Cohen, G. Quispe, S. L. Corff, C. Ollion, and E. Moulines, "Diffusion bridges vector quantized variational autoencoders," *arXiv:2202.04895*, 2022. 3.2.3, 7

[126] P. Xie, Q. Zhang, Z. Li, H. Tang, Y. Du, and X. Hu, "Vector quantized diffusion model with code-unet for text-to-sign pose sequences generation," *arXiv:2208.09141*, 2022. 7, 8

[127] 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 *CVPR*, 2022, pp. 5152–5161. 3.2.3

[128] 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:2212.02936*, 2022.

[129] X. Xu, Z. Wang, E. Zhang, K. Wang, and H. Shi, "Versatile diffusion: Text, images and variations all in

one diffusion model," *arXiv:2211.08332*, 2022. 3.2.3

[130] H. A. Pierson and M. S. Gashler, "Deep learning in robotics: a review of recent research," *Advanced Robotics*, 2017. 3.2.3

[131] 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

[132] J. Wang, H. Cao, J. Z. Zhang, and Y. Qi, "Computational protein design with deep learning neural networks," *Scientific reports*, 2018. 3.2.3

[133] W. Cao, Z. Yan, Z. He, and Z. He, "A comprehensive survey on geometric deep learning," *IEEE Access*, 2020. 3.2.3

[134] 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

[135] N. Anand and T. Achim, "Protein structure and sequence generation with equivariant denoising diffusion probabilistic models," *arXiv:2205.15019*, 2022. 3.2.3, 2, 4.8.1, 8

[136] 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

[137] V. Dutordoir, A. Saul, Z. Ghahramani, and F. Simpson, "Neural diffusion processes," *arXiv:2206.03992*, 2022. 3.2.3

[138] 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:2205.11487*, 2022. 4.1, 4.1.1, 2

[139] A. Hertz, R. Mokady, J. Tenenbaum, K. Aberman, Y. Pritch, and D. Cohen-Or, "Prompt-to-prompt image editing with cross attention control," *arXiv preprint arXiv:2208.01626*, 2022. 4.1.1, 2

[140] N. Ruiz, Y. Li, V. Jampani, Y. Pritch, M. Rubinstein, and K. Aberman, "Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation," in *CVPR*, 2023, pp. 22 500–22 510. 4.1.1

[141] H. Chefer, Y. Alaluf, Y. Vinker, L. Wolf, and D. Cohen-Or, "Attend-and-excite: Attention-based semantic guidance for text-to-image diffusion models," *ACM Transactions on Graphics (TOG)*, vol. 42, no. 4, pp. 1–10, 2023. 4.1.1, 2

[142] T. Brooks, A. Holynski, and A. A. Efros, "Instruct-pix2pix: Learning to follow image editing instructions," in *CVPR*, 2023, pp. 18 392–18 402. 4.1, 4.1.2, 2

[143] L. Zhang, A. Rao, and M. Agrawala, "Adding conditional control to text-to-image diffusion models," in *ICCV*, 2023, pp. 3836–3847. 4.1, 4.1.2, 2

[144] Z. Lyu, Z. Kong, X. Xu, L. Pan, and D. Lin, "A conditional point diffusion-refinement paradigm for 3d point cloud completion," *arXiv:2112.03530*, 2021. 4.2.1, 2, 7, 8

[145] H. Jun and A. Nichol, "Shap-e: Generating conditional 3d implicit functions," *arXiv preprint arXiv:2305.02463*, 2023. 4.2.1, 2

[146] A. Nichol, H. Jun, P. Dhariwal, P. Mishkin, and M. Chen, "Point-e: A system for generating 3d point clouds from complex prompts," *arXiv preprint arXiv:2212.08751*, 2022. 4.2.1, 2

[147] L. Zhou, Y. Du, and J. Wu, "3d shape generation and completion through point-voxel diffusion," in *ICCV*, 2021, pp. 5826–5835. 4.2.1, 2, 8

[148] R. Liu, R. Wu, B. Van Hoorick, P. Tokmakov, S. Zakharov, and C. Vondrick, "Zero-1-to-3: Zero-shot one image to 3d object," in *ICCV (ICCV)*, October 2023, pp. 9298–9309. 4.2.1, 2

[149] M. Liu, C. Xu, H. Jin, L. Chen, Z. Xu, H. Su *et al.*, "One-2-3-45: Any single image to 3d mesh in 45 seconds without per-shape optimization," *arXiv preprint arXiv:2306.16928*, 2023. 4.2.1, 2

[150] S. Luo and W. Hu, "Score-based point cloud denoising," in *ICCV*, 2021, pp. 4583–4592. 4.2.1, 2, 8

[151] ——, "Diffusion probabilistic models for 3d point cloud generation," in *CVPR*, 2021, pp. 2837–2845. 4.2.1, 2, 7, 8

[152] B. Poole, A. Jain, J. T. Barron, and B. Mildenhall, "Dreamfusion: Text-to-3d using 2d diffusion," *arXiv:2209.14988*, 2022. 4.2.2, 2, 8

[153] C.-H. Lin, J. Gao, L. Tang, T. Takikawa, X. Zeng, X. Huang, K. Kreis, S. Fidler, M.-Y. Liu, and T.-Y. Lin, "Magic3d: High-resolution text-to-3d content creation," in *CVPR*, 2023, pp. 300–309. 4.2.2, 2

[154] 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:2209.14792*, 2022. 4.3, 2

[155] V. Voleti, A. Jolicoeur-Martineau, and C. Pal, "Mcvd-masked conditional video diffusion for prediction, generation, and interpolation," *NeurIPS*, vol. 35, pp. 23 371–23 385, 2022. 4.3, 2, 8

[156] W. Harvey, S. Naderiparizi, V. Masrani, C. Weilbach, and F. Wood, "Flexible diffusion modeling of long videos," *arXiv:2205.11495*, 2022. 4.3, 2, 8

[157] R. Yang, P. Srivastava, and S. Mandt, "Diffusion probabilistic modeling for video generation," *arXiv:2203.09481*, 2022. 4.3, 2, 5.2, 8

[158] T. Höppe, A. Mehrjou, S. Bauer, D. Nielsen, and A. Dittadi, "Diffusion models for video prediction and infilling," *arXiv:2206.07696*, 2022. 4.3, 2, 8

[159] Y. Guo, C. Yang, A. Rao, Y. Wang, Y. Qiao, D. Lin, and B. Dai, "Animatediff: Animate your personalized text-to-image diffusion models without specific tuning," *arXiv preprint arXiv:2307.04725*, 2023. 4.3, 2

[160] H. Chung, B. Sim, D. Ryu, and J. C. Ye, "Improving diffusion models for inverse problems using manifold constraints," *NeurIPS*, vol. 35, pp. 25 683–25 696, 2022. 2, 4.4.1, 4

[161] H. Chung and J. C. Ye, "Score-based diffusion models for accelerated mri," *Medical image analysis*, vol. 80, p. 102479, 2022. 2, 4.4.1, 8

[162] Y. Yang, H. Fu, A. I. Aviles-Rivero, C.-B. Schönlieb, and L. Zhu, "Diffmic: Dual-guidance diffusion network for medical image classification," in *MICCAI*. Springer, 2023, pp. 95–105. 2, 4.4.1

[163] D. Hu, Y. K. Tao, and I. Oguz, "Unsupervised denoising of retinal oct with diffusion probabilistic model,"

in *Medical Imaging 2022: Image Processing*, vol. 12032. SPIE, 2022, pp. 25–34. 2, 4.4.1

[164] J. Wyatt, A. Leach, S. M. Schmon, and C. G. Willcocks, "Anoddpm: Anomaly detection with denoising diffusion probabilistic models using simplex noise," in *CVPR*, 2022, pp. 650–656. 2, 4.4.2

[165] J. Li, H. Cao, J. Wang, F. Liu, Q. Dou, G. Chen, and P.-A. Heng, "Fast non-markovian diffusion model for weakly supervised anomaly detection in brain mr images," in *MICCAI*. Springer, 2023, pp. 579–589. 2, 4.4.2

[166] B. Kim, I. Han, and J. C. Ye, "Diffusemorph: unsupervised deformable image registration using diffusion model," in *ECCV*. Springer, 2022, pp. 347–364. 2, 4.4.2

[167] H. Chung, E. S. Lee, and J. C. Ye, "Mr image denoising and super-resolution using regularized reverse diffusion," *IEEE Transactions on Medical Imaging*, vol. 42, no. 4, pp. 922–934, 2022. 2, 4.4.2, 8

[168] Z. Dorjsembe, S. Odonchimed, and F. Xiao, "Three-dimensional medical image synthesis with denoising diffusion probabilistic models," in *Medical Imaging with Deep Learning*, 2022. 2, 4.4.2

[169] S. Gong, C. Chen, Y. Gong, N. Y. Chan, W. Ma, C. H.-K. Mak, J. Abrigo, and Q. Dou, "Diffusion model based semi-supervised learning on brain hemorrhage images for efficient midline shift quantification," in *International Conference on Information Processing in Medical Imaging*. Springer, 2023, pp. 69–81. 2, 4.4.2

[170] W. H. Pinaya, P.-D. Tudosiu, J. Dafflon, P. F. Da Costa, V. Fernandez, P. Nachev, S. Ourselin, and M. J. Cardoso, "Brain imaging generation with latent diffusion models," in *MICCAI Workshop on Deep Generative Models*. Springer, 2022, pp. 117–126. 2, 4.4.2

[171] 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, 4.5.1

[172] H. Yuan, Z. Yuan, C. Tan, F. Huang, and S. Huang, "Seqdiffuseq: Text diffusion with encoder-decoder transformers," *arXiv preprint arXiv:2212.10325*, 2022. 2, 4.5.2

[173] Z. Lin, Y. Gong, Y. Shen, T. Wu, Z. Fan, C. Lin, N. Duan, and W. Chen, "Text generation with diffusion language models: A pre-training approach with continuous paragraph denoise," in *ICML*. PMLR, 2023, pp. 21 051–21 064. 2, 4.5.2

[174] T. Tang, Y. Chen, Y. Du, J. Li, W. X. Zhao, and J.-R. Wen, "Learning to imagine: Visually-augmented natural language generation," *arXiv preprint arXiv:2305.16944*, 2023. 2, 4.5.2

[175] S. Gong, M. Li, J. Feng, Z. Wu, and L. Kong, "Diffuseq: Sequence to sequence text generation with diffusion models," in *ICLR*, 2022. 2, 4.5.2

[176] T. Wu, Z. Fan, X. Liu, Y. Gong, Y. Shen, J. Jiao, H.-T. Zheng, J. Li, Z. Wei, J. Guo *et al.*, "Ar-diffusion: Autoregressive diffusion model for text generation," *arXiv preprint arXiv:2305.09515*, 2023. 2, 4.5.2

[177] Z. Gao, J. Guo, X. Tan, Y. Zhu, F. Zhang, J. Bian, and L. Xu, "Difformer: Empowering diffusion model on embedding space for text generation," *arXiv preprint arXiv:2212.09412*, 2022. 2, 4.5.2

[178] R. Strudel, C. Tallec, F. Altché, Y. Du, Y. Ganin, A. Mensch, W. Grathwohl, N. Savinov, S. Dieleman, L. Sifre *et al.*, "Self-conditioned embedding diffusion for text generation," *arXiv preprint arXiv:2211.04236*, 2022. 2, 4.5.2

[179] J. Ye, Z. Zheng, Y. Bao, L. Qian, and Q. Gu, "Diffusion language models can perform many tasks with scaling and instruction-finetuning," *arXiv preprint arXiv:2308.12219*, 2023. 2, 4.5.2

[180] H. Lim, M. Kim, S. Park, and N. Park, "Regular time-series generation using sgm," *arXiv preprint arXiv:2301.08518*, 2023. 2, 4.6

[181] M. Liu, H. Huang, H. Feng, L. Sun, B. Du, and Y. Fu, "Pristi: A conditional diffusion framework for spatiotemporal imputation," *arXiv preprint arXiv:2302.09746*, 2023. 2

[182] J. M. Lopez Alcaraz and N. Strodthoff, "Diffusion-based time series imputation and forecasting with structured atate apace models," *Transactions on machine learning research*, pp. 1–36, 2023. 2, 4.6

[183] M. Fahim Sikder, R. Ramachandranpillai, and F. Heintz, "Transfusion: Generating long, high fidelity time series using diffusion models with transformers," *arXiv e-prints*, pp. arXiv–2307, 2023. 2, 4.6

[184] K. Rasul, C. Seward, I. Schuster, and R. Vollgraf, "Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting," in *ICML*. PMLR, 2021, pp. 8857–8868. 2, 4.6

[185] H. Wen, Y. Lin, Y. Xia, H. Wan, R. Zimmermann, and Y. Liang, "Diffstg: Probabilistic spatio-temporal graph forecasting with denoising diffusion models," *arXiv preprint arXiv:2301.13629*, 2023. 2, 4.6

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

[187] 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. 2, 4.7, 8

[188] 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:2207.06389*, 2022. 2, 8

[189] 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:2205.14807*, 2022. 2, 4.7, 8

[190] S. Liu, Y. Cao, D. Su, and H. Meng, "Diffsvc: A diffusion probabilistic model for singing voice conversion," in *IEEE ASRU*, 2021. 2, 4.7, 8

[191] 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. 2, 4.7, 8

[192] J. Tae, H. Kim, and T. Kim, "Editts: Score-based editing for controllable text-to-speech," *arXiv:2110.02584*, 2021. 2, 4.7, 8

[193] 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. 2, 4.7, 8

[194] Y. Koizumi, H. Zen, K. Yatabe, N. Chen, and M. Bacchiani, "Specgrad: Diffusion probabilistic model based neural vocoder with adaptive noise spectral shaping," *arXiv:2203.16749*, 2022. 2, 4.7, 8

[195] S. Kim, H. Kim, and S. Yoon, "Guided-tts 2: A diffusion model for high-quality adaptive text-to-speech with untranscribed data," *arXiv:2205.15370*, 2022. 2, 4.7, 8

[196] 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. 2, 4.7, 8

[197] 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:2207.09983*, 2022. 2, 4.7, 8

[198] E. Hoogeboom, V. G. Satorras, C. Vignac, and M. Welling, "Equivariant diffusion for molecule generation in 3d," in *ICML*. PMLR, 2022, pp. 8867–8887. 2, 4.8.1, 8

[199] J. S. Lee and P. M. Kim, "Proteinsgm: Score-based generative modeling for de novo protein design," *bioRxiv*, 2022. 2, 4.8.1, 8

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

[201] J. Yim, B. L. Trippe, V. De Bortoli, E. Mathieu, A. Doucet, R. Barzilay, and T. Jaakkola, "Se (3) diffusion model with application to protein backbone generation," *arXiv preprint arXiv:2302.02277*, 2023. 2

[202] 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:2209.15611*, 2022. 2, 4.8.1, 8

[203] G. Corso, H. Stärk, B. Jing, R. Barzilay, and T. Jaakkola, "Diffdock: Diffusion steps, twists, and turns for molecular docking," *arXiv:2210.01776*, 2022. 2, 4.8.2, 8

[204] J. L. Watson, D. Juergens, N. R. Bennett, B. L. Trippe, J. Yim, H. E. Eisenach, W. Ahern, A. J. Borst, R. J. Ragotte, L. F. Milles *et al.*, "De novo design of protein structure and function with rfdiffusion," *Nature*, pp. 1–3, 2023. 2, 4.8.2

[205] S. L. Lisanza, J. M. Gershon, S. W. K. Tipps, L. Arnoldt, S. Hendel, J. N. Sims, X. Li, and D. Baker, "Joint generation of protein sequence and structure with rosettafold sequence space diffusion," *bioRxiv*, pp. 2023–05, 2023. 2, 4.8.2

[206] C. Vignac, I. Krawczuk, A. Siraudin, B. Wang, V. Cevher, and P. Frossard, "Digress: Discrete denoising diffusion for graph generation," *arXiv:2209.14734*, 2022. 2, 4.9

[207] D. Um, J. Park, S. Park, and J. young Choi, "Confidence-based feature imputation for graphs with partially known features," in *ICLR*, 2022. 2, 4.9

[208] X. Chen, J. He, X. Han, and L.-P. Liu, "Efficient and degree-guided graph generation via discrete diffusion modeling," *arXiv preprint arXiv:2305.04111*, 2023. 2, 4.9

[209] Q. Wu, C. Yang, W. Zhao, Y. He, D. Wipf, and J. Yan, "Difformer: Scalable (graph) transformers induced by energy constrained diffusion," in *The Eleventh International Conference on Learning Representations*, 2022. 2, 4.9

[210] J. Chen, S. Wu, A. Gupta, and R. Ying, "D4explainer: In-distribution gnn explanations via discrete denoising diffusion," *arXiv preprint arXiv:2310.19321*, 2023. 2, 4.9

[211] D. W. Otter, J. R. Medina, and J. K. Kalita, "A survey of the usages of deep learning for natural language processing," *IEEE transactions on neural networks and learning systems*, vol. 32, no. 2, pp. 604–624, 2020. 4.5

[212] H. Zou, Z. M. Kim, and D. Kang, "Diffusion models in nlp: A survey," *arXiv preprint arXiv:2305.14671*, 2023. 4.5, 4.5.2, 5.2

[213] Y. Li, K. Zhou, W. X. Zhao, and J.-R. Wen, "Diffusion models for non-autoregressive text generation: A survey," *arXiv preprint arXiv:2303.06574*, 2023. 4.5, 4.5.2, 5.2

[214] L. Lin, Z. Li, R. Li, X. Li, and J. Gao, "Diffusion models for time series applications: A survey," *arXiv preprint arXiv:2305.00624*, 2023. 4.6, 5.2

[215] 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. 4.7, 8

[216] S. Min, B. Lee, and S. Yoon, "Deep learning in bioinformatics," *Briefings in bioinformatics*, vol. 18, no. 5, pp. 851–869, 2017. 4.8

[217] C. Bilodeau, W. Jin, T. Jaakkola, R. Barzilay, and K. F. Jensen, "Generative models for molecular discovery: Recent advances and challenges," *Wiley Interdisciplinary Reviews: Computational Molecular Science*, vol. 12, no. 5, p. e1608, 2022. 4.8

[218] G. Corso, Y. Xu, V. De Bortoli, R. Barzilay, and T. Jaakkola, "Particle guidance: non-iid diverse sampling with diffusion models," *arXiv preprint arXiv:2310.13102*, 2023. 4.8.1

[219] S. Alamdari, N. Thakkar, R. van den Berg, A. X. Lu, N. Fusi, A. P. Amini, and K. K. Yang, "Protein generation with evolutionary diffusion: sequence is all you need," *bioRxiv*, pp. 2023–09, 2023. 4.8.2

[220] F.-A. Croitoru, V. Hondru, R. T. Ionescu, and M. Shah, "Diffusion models in vision: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2023. 5.2

[221] C. Zhang, C. Zhang, S. Zheng, M. Zhang, M. Qamar, S.-H. Bae, and I. S. Kweon, "Audio diffusion model for speech synthesis: A survey on text to speech and speech enhancement in generative ai," *arXiv preprint arXiv:2303.13336*, 2023. 5.2

[222] A. Kazerouni, E. K. Aghdam, M. Heidari, R. Azad, M. Fayyaz, I. Hacihaliloglu, and D. Merhof, "Diffusion models for medical image analysis: A comprehensive survey," *arXiv preprint arXiv:2211.07804*, 2022. 5.2

[223] Z. Guo, J. Liu, Y. Wang, M. Chen, D. Wang, D. Xu, and J. Cheng, "Diffusion models in bioinformatics: A new wave of deep learning revolution in action," *arXiv preprint arXiv:2302.10907*, 2023. 5.2

[224] M. Zhang, M. Qamar, T. Kang, Y. Jung, C. Zhang, S.-H. Bae, and C. Zhang, "A survey on graph diffusion models: Generative ai in science for molecule, protein and material," *arXiv preprint arXiv:2304.01565*, 2023. 5.2

[225] W. Fan, C. Liu, Y. Liu, J. Li, H. Li, H. Liu, J. Tang, and Q. Li, "Generative diffusion models on graphs: Meth-

ods and applications," *arXiv preprint arXiv:2302.02591*, 2023. 5.2

[226] H. Koo and T. E. Kim, "Acomprehensive survey on generative diffusion models for structured data," *arXiv preprint arXiv:2306.04139*, 2023. 5.2

[227] Y. Song and S. Ermon, "Generative modeling by estimating gradients of the data distribution," *NeurIPS*, 2019. 2, 3, 5, 7

[228] A. Borji, "Pros and cons of gan evaluation measures: New developments," *Comput Vis Image Underst*, vol. 215, p. 103329, 2022. B.1

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

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

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

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

[233] 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

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

[235] 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

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

[237] Q. Zhang and Y. Chen, "Diffusion normalizing flow," *NeurIPS*, 2021. 6, 7

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

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

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

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

[242] B. Kawar, M. Elad, S. Ermon, and J. Song, "Denoising diffusion restoration models," in *ICLR Workshop*, 2022. 8

[243] 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. 8

[244] L. Theis, T. Salimans, M. D. Hoffman, and F. Mentzer, "Lossy compression with gaussian diffusion," *arXiv:2206.08889*, 2022. 8

[245] 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," *Neu-rocomputing*, 2022. 8

[246] 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. 8

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

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

[249] T. Amit, E. Nachmani, T. Shaharbany, and L. Wolf, "Segdiff: Image segmentation with diffusion probabilistic models," *arXiv:2112.00390*, 2021. 8

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

[251] Y. Song, L. Shen, L. Xing, and S. Ermon, "Solving inverse problems in medical imaging with score-based generative models," in *ICLR*, 2021. 8

[252] T. Chen, R. Zhang, and G. Hinton, "Analog bits: Generating discrete data using diffusion models with self-conditioning," *arXiv:2208.04202*, 2022. 8

[253] J. M. L. Alcaraz and N. Strodthoff, "Diffusion-based time series imputation and forecasting with structured state space models," *arXiv:2208.09399*, 2022. 8

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

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

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

[257] S. Luo, C. Shi, M. Xu, and J. Tang, "Predicting molecular conformation via dynamic graph score matching," *NeurIPS*, vol. 34, pp. 19 784–19 795, 2021. 8

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

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 [10]

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

A.1.2 Annealed Langevin Dynamics Sampling

Algorithm 2 Annealed Langevin Dynamics Sampling [227]

Initialize x_0
for $i = 1, ..., L$ **do**
  $\alpha_i \leftarrow \epsilon \cdot \sigma_i^2 / \sigma_L^2$
  for $t = 1, ..., L$ **do**
    $z_t \sim \mathcal{N}(0, I)$
    $\tilde{x}_t = \tilde{x}_{t-1} + \frac{\alpha_i}{2} s_\theta (\tilde{x}_{t-1}, \sigma_i) + \sqrt{\alpha_i} z_t$
  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 [11]

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

A.2 Conditional Sampling

A.2.1 Labeled Condition

Algorithm 4 Classifier-guided Diffusion Sampling [27]

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 $\mathcal{N} \left(\mu + s\Sigma \nabla_{x_t} \log p_\phi (y \mid x_t), \Sigma \right)$
  end if
  if DDIM Sampling **then**
    $\hat{\epsilon} \leftarrow \epsilon_\theta (x_t) - \sqrt{1 - \bar{\alpha}_t} \nabla_{x_t} \log p_\phi (y \mid x_t)$
    $x_{t-1} \leftarrow \sqrt{\bar{\alpha}_{t-1}} \left(\frac{x_t - \sqrt{1-\bar{\alpha}_t} \hat{\epsilon}}{\sqrt{\bar{\alpha}_t}} \right) + \sqrt{1 - \bar{\alpha}_{t-1}} \hat{\epsilon}$
  end if
end for
return x_0

Algorithm 5 Classifier-free Guidance Sampling [48]

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

A.3 Unlabeled Condition

Algorithm 6 Self-guided Conditional Sampling [56]

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**
    $\tilde{\epsilon} \leftarrow (1 - w)\epsilon_\theta (\mathbf{x}_t, t) + w\epsilon_\theta (\mathbf{x}_t, t; f_\psi (g_\phi(\mathbf{x}; \mathcal{D}); \mathcal{D}))$
  else if Self-Labeled Guidance **then**
    $\tilde{\epsilon} \leftarrow \epsilon_\theta (\mathbf{x}_t, \text{concat}[t, \mathbf{k}])$
  else if Self-Boxed Guidance **then**
    $\tilde{\epsilon} \leftarrow \epsilon_\theta (\text{concat}[\mathbf{x}_t, \mathbf{k}_s], \text{concat}[t, \mathbf{k}])$
  else if Self-Segmented Guidance **then**
    $\tilde{\epsilon} \leftarrow \epsilon_\theta \left(\text{concat}[\mathbf{x}_t, \mathbf{k}_s], \text{concat}[t, \hat{\mathbf{k}}] \right)$
  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 [228, 229]. 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 [230] is applied to inception score calculation:

$$
\begin{aligned}
IS &= D_{KL}(p_{IS} \parallel q_{IS}) \\
&= \mathbb{E}_{x \sim p_{IS}} \left[\log \frac{p_{IS}}{q_{IS}} \right] \\
&= \mathbb{E}_{x \sim p_{IS}} \left[\log(p_{IS}) - \log(p_{IS}) \right]
\end{aligned} \tag{19}
$$

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.

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.

$$
\text{FID} = \left\| \mu_r - \mu_g \right\|^2 + \text{Tr} \left(\Sigma_r + \Sigma_g - 2 \left(\Sigma_r \Sigma_g \right)^{1/2} \right) \tag{20}
$$

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.*, [231] negative log-likelihood is seen as a common evaluation metric that describes all modes of data distribution. Some diffusion models like improved DDPM [78] regard the NLL as measurement of distribution matching.

$$
\text{NLL} = \mathbb{E} \left[-\log p_\theta(x) \right] \tag{21}
$$

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 [232], ImageNet[233], and CelebA-64 [234]. 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 [83]	1000	3.15	-
SN-DDIM [83]	1000	2.90	-
NCSN [227]	1000	10.23	-
NCSN ++ [11]	1000	1.92	1.97
DDPM ++ [11]	1000	1.90	2.10
DiffuseVAE [87]	1000	4.76	-
Analytic DPM [82]	1000	-	2.66
ES-DDPM [73]	200	2.55	-
PNDM [46]	200	2.71	-
ES-DDPM [73]	100	3.01	-
PNDM [46]	100	2.81	-
Analytic DPM [82]	100	-	2.66
NPR-DDIM [83]	100	4.27	-
SN-DDIM [83]	100	3.04	-
ES-DDPM [73]	50	3.97	-
PNDM [46]	50	3.34	-
NPR-DDIM [83]	50	6.04	-
SN-DDIM [83]	50	3.83	-
DPM-Solver Discrete [47]	36	2.71	-
ES-DDPM [73]	20	4.90	-
PNDM [46]	20	5.51	-
DPM-Solver Discrete [47]	20	2.82	-
ES-DDPM [73]	10	6.44	-
PNDM [46]	10	7.71	-
Analytic DPM [82]	10	-	2.97
NPR-DDPM [83]	10	28.37	-
SN-DDPM [83]	10	20.60	-
NPR-DDIM [83]	10	14.98	-
SN-DDIM [83]	10	10.20	-
DPM-Solver Discrete [47]	10	6.92	-
ES-DDPM [73]	5	9.15	-
PNDM [46]	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 [160]	1000	25.4	-	-
Analytic DPM [82]	1000	-	-	3.61
ES-DDPM [73]	900	2.07	55.29	-
Restart [30]	623	1.36	-	-
Efficient Sampling [85]	256	3.87	-	-
Analytic DPM [82]	200	-	-	3.64
NPR-DDPM [83]	200	16.96	-	-
SN-DDPM [83]	200	16.61	-	-
ES-DDPM [73]	100	3.75	48.63	-
DPM-Solver Discrete [47]	57	17.47	-	-
Restart [30]	39	2.38	-	-
ES-DDPM [73]	25	3.75	48.63	-
GGDM [84]	25	18.4	18.12	-
Analytic DPM [82]	25	-	-	3.83
NPR-DDPM [83]	25	28.27	-	-
SN-DDPM [83]	25	27.58	-	-
DPM-Solver Discrete [47]	20	18.53	-	-
ES-DDPM [73]	10	3.93	48.81	-
GGDM [84]	10	37.32	14.76	-
DPM-Solver Discrete [47]	10	24.4	-	-
ES-DDPM [73]	5	4.25	48.04	-
GGDM [84]	5	55.14	12.9	-

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

Method	NFE	FID	IS	NLL
Improved DDPM [78]	4000	2.90	-	-
VE SDE [11]	2000	2.20	9.89	-
VP SDE [11]	2000	2.41	9.68	3.13
sub-VP SDE [11]	2000	2.41	9.57	2.92
DDPM [10]	1000	3.17	9.46	3.72
NCSN [227]	1000	25.32	8.87	-
SSM [235]	1000	54.33	-	-
NCSNv2 [236]	1000	10.87	8.40	-
D3PM [16]	1000	7.34	8.56	3.44
Efficient Sampling [111]	1000	2.94	-	-
NCSN++ [11]	1000	2.33	10.11	3.04
DDPM++ [11]	1000	2.47	9.78	2.91
TDPM [71]	1000	3.07	9.24	-
VDM [77]	1000	4.00	-	-
DiffuseVAE [87]	1000	8.72	8.63	-
Analytic DPM [82]	1000	-	-	3.59
NPR-DDPM [83]	1000	4.27	-	-
SN-DDPM [83]	1000	4.07	-	-
Gotta Go Fast VP [29]	1000	2.49	-	-
Gotta Go Fast VE [29]	1000	3.14	-	-
INDM [88]	1000	2.28	-	3.09

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

Method	NFE	FID	IS	NLL
Diffusion Step [75]	600	3.72	-	-
ES-DDPM [73]	600	3.17	-	-
Diffusion Step [75]	400	14.38	-	-
Diffusion Step [75]	200	5.44	-	-
NPR-DDPM [83]	200	4.10	-	-
SN-DDPM [83]	200	3.72	-	-
Gotta Go Fast VP [29]	180	2.44	-	-
Gotta Go Fast VE [29]	180	3.40	-	-
LSGM [35]	138	2.10	-	-
PFGM [32]	110	2.35	-	-
DDIM [26]	100	4.16	-	-
FastDPM [79]	100	2.86	-	-
TDPM [71]	100	3.10	9.34	-
NPR-DDPM [83]	100	4.52	-	-
SN-DDPM [83]	100	3.83	-	-
DiffuseVAE [87]	100	11.71	8.27	-
DiffFlow [237]	100	14.14	-	3.04
Analytic DPM [82]	100	-	-	3.59
Efficient Sampling [111]	64	3.08	-	-
DPM-Solver [47]	51	2.59	-	-
DDIM [26]	50	4.67	-	-
FastDPM [79]	50	3.2	-	-
NPR-DDPM [83]	50	5.31	-	-
SN-DDPM [83]	50	4.17	-	-
Improved DDPM [78]	50	4.99	-	-
TDPM [71]	50	3.3	9.22	-
DEIS [111]	50	2.57	-	-
gDDIM [81]	50	2.28	-	-
DPM-Solver Discrete [47]	44	3.48	-	-
STF [43]	35	1.90	-	-
EDM [25]	35	1.79	-	-
PFGM++ [33]	35	1.74	-	-
Improved DDPM [78]	25	7.53	-	-
GGDM [84]	25	4.25	9.19	-
NPR-DDPM [83]	25	7.99	-	-
SN-DDPM [83]	25	6.05	-	-
DDIM [26]	20	6.84	-	-
FastDPM [79]	20	5.05	-	-
DEIS [111]	20	2.86	-	-
DPM-Solver [47]	20	2.87	-	-
DPM-Solver Discrete [47]	20	3.72	-	-
Efficient Sampling [111]	16	3.41	-	-
NPR-DDPM [83]	10	19.94	-	-
SN-DDPM [83]	10	16.33	-	-
DDIM [26]	10	13.36	-	-
FastDPM [79]	10	9.90	-	-
GGDM [84]	10	8.23	8.90	-
Analytic DPM [82]	10	-	-	4.11
DEIS [111]	10	4.17	-	-
DPM-Solver [47]	10	6.96	-	-
DPM-Solver Discrete [47]	10	10.16	-	-
Progressive Distillation [31]	8	2.57	-	-
Denoising Diffusion GAN [86]	8	4.36	9.43	-
GGDM [84]	5	13.77	8.53	-
DEIS [111]	5	15.37	-	-
Progressive Distillation [31]	4	3.00	-	-
TDPM [71]	4	3.41	9.00	-
Denoising Diffusion GAN [86]	4	3.75	9.63	-
Progressive Distillation [31]	2	4.51	-	-
TDPM [71]	2	4.47	8.97	-
Denoising Diffusion GAN [86]	2	4.08	9.80	-
Denoising student [61]	1	9.36	8.36	-
Progressive Distillation [31]	1	9.12	-	-
TDPM [71]	1	8.91	8.65	-

TABLE 7
Details for Improved Diffusion Methods

Method	Year	Data	Model	Framework	Training	Sampling	Code
Landmark Works							
DPM [9]	2015	RGB Image	Discrete	Diffusion	L_{simple}	Ancestral	[code]
DDPM [10]	2020	RGB Image	Discrete	Diffusion	L_{simple}	Ancestral	[code]
NCSN [227]	2019	RGB Image	Discrete	Score	L_{DSM}	Langevin dynamics	[code]
NCSNv2 [236]	2020	RGB Image	Discrete	Score	L_{DSM}	Langevin dynamics	[code]
Score SDE [11]	2020	RGB Image	Continuous	SDE	L_{DSM}	PC-Sampling	[code]
Improved Works							
Progressive Distill [31]	2022	RGB Image	Discrete	Diffusion	L_{simple}	DDIM Sampling	[code]
Denoising Student [61]	2021	RGB Image	Discrete	Diffusion	$L_{Distill}$	DDIM Sampling	[code]
TDPM [71]	2022	RGB Image	Discrete	Diffusion	$L_{DDPM\&GAN}$	Ancestral	-
ES-DDPM [73]	2022	RGB Image	Discrete	Diffusion	$L_{DDPM\&VAE}$	Conditional Sampling	[code]
CCDF [58]	2021	RGB Image	Discrete	SDE	L_{simple}	Langevin dynamics	[code]
Franzese's Model [75]	2022	RGB Image	Continuous	SDE	L_{DSM}	DDIM Sampling	-
FastDPM [79]	2021	RGB Image	Discrete	Diffusion	L_{simple}	DDIM Sampling	[code]
Improved DDPM [78]	2021	RGB Image	Discrete	Diffusion	L_{hybrid}	Ancestral	[code]
VDM [77]	2022	RGB Image	Both	Diffusion	L_{simple}	Ancestral	[code]
San-Roman's Model [80]	2021	RGB Image	Discrete	Diffusion	$L_{DDPM\&Noise}$	Ancestral	-
Analytic-DPM [82]	2022	RGB Image	Discrete	Score	$L_{Trajectory}$	Ancestral	[code]
NPR-DDPM [83]	2022	RGB Image	Discrete	Diffusion	$L_{DDPM\&Noise}$	Ancestral	[code]
SN-DDPM [83]	2022	RGB Image	Discrete	Score	L_{square}	Ancestral	[code]
DDIM [26]	2021	RGB Image	Discrete	Diffusion	L_{simple}	DDIM Sampling	[code]
gDDIM [81]	2022	RGB Image	Continuous	SDE&ODE	L_{DSM}	PC-Sampling	[code]
INDM [88]	2022	RGB Image	Continuous	SDE	$L_{DDPM\&Flow}$	PC-Sampling	-
Gotta Go Fast [29]	2021	RGB Image	Continuous	SDE	L_{DSM}	Improved Euler	[code]
DPM-Solver [47]	2022	RGB Image	Continuous	ODE	L_{DSM}	Higher ODE solvers	[code]
Restart [30]	2023	RGB Image	Continuous	SDE	L_{DSM}	2nd Order Heun	[code]
EDM [25]	2022	RGB Image	Continuous	ODE	L_{DSM}	2nd Order Heun	[code]
PFGM [32]	2022	RGB Image	Continuous	ODE	L_{DSM}	ODE-Solver	[code]
PFGM++ [33]	2023	RGB Image	Continuous	ODE	L_{DSM}	2nd Order Heun	[code]
PNDM [46]	2022	Manifold	Discrete	ODE	L_{simple}	Multi-step & Runge-Kutta	[code]
DDSS [84]	2021	RGB Image	Discrete	Diffusion	L_{simple}	Dynamic Programming	-
GGDM [85]	2022	RGB Image	Discrete	Diffusion	L_{KID}	Dynamic Programming	-
Diffusion GAN [86]	2022	RGB Image	Discrete	Diffusion	$L_{DDPM\&GAN}$	Ancestral	[code]
DiffuseVAE [87]	2022	RGB Image	Discrete	Diffusion	$L_{DDPM\&VAE}$	Ancestral	[code]
DiffFlow [237]	2021	RGB Image	Discrete	SDE	L_{DSM}	Langevin & Flow Sampling	[code]
LSGM [35]	2021	RGB Image	Continuous	ODE	$L_{DDPM\&VAE}$	ODE-Slover	[code]
Score-flow [103]	2021	Dequantization	Continuous	SDE	L_{DSM}	PC-Sampling	[code]
PDM [114]	2022	RGB Image	Continuous	SDE	L_{Gap}	PC-Sampling	-
ScoreEBM [238]	2021	RGB Image	Discrete	Score	$L_{Recovery}$	Langevin dynamics	[code]
Song's Model [239]	2021	RGB Image	Discrete	Score	L_{DSM}	Langevin dynamics	-
Huang's Model [104]	2021	RGB Image	Continuous	SDE	L_{DSM}	SDE-Solver	[code]
De Bortoli's Model [240]	2021	RGB Image	Continuous	SDE	L_{DSM}	Importance Sampling	[code]
PVD [241]	2021	Point Cloud	Discrete	Diffusion	L_{simple}	Ancestral	[code]
Luo's Model [151]	2021	Point Cloud	Discrete	Diffusion	L_{simple}	Ancestral	[code]
Lyu's Model [144]	2022	Point Cloud	Discrete	Diffusion	L_{simple}	Farthest Point Sampling	[code]
D3PM [16]	2021	Categorical Data	Discrete	Diffusion	L_{hybrid}	Ancestral	[code]
Argmax [92]	2021	Categorical Data	Discrete	Diffusion	$L_{DDPM\&Flow}$	Gumbel sampling	[code]
ARDM [93]	2022	Categorical Data	Discrete	Diffusion	L_{simple}	Ancestral	[code]
Campbell's Model [96]	2022	Categorical Data	Continuous	Diffusion	L_{simple}^{CT}	PC-Sampling	[code]
VQ-diffusion [94]	2022	Vector-Quantized	Discrete	Diffusion	L_{simple}	Ancestral	[code]
Improved VQ-Diff [95]	2022	Vector-Quantized	Discrete	Diffusion	L_{simple}	Purity Prior Sampling	[code]
Cohen's Model [125]	2022	Vector-Quantized	Discrete	Diffusion	L_{simple}	Ancestral & VAE Sampling	[code]
Xie's Model [126]	2022	Vector-Quantized	Discrete	Diffusion	$L_{DDPM\&Class}$	Ancestral & VAE Sampling	-
RGSM [97]	2022	Manifold	Continuous	SDE	L_{DSM}	Geodesic Random Walk	-
RDM [98]	2022	Manifold	Continuous	SDE	L_{simple}^{CT}	Importance Sampling	-
EDP-GNN [22]	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 [13]	2021	RGB-Image	SDE	Inpainting, Super-Resolution, Edge to image translation	[code]
DDRM [242]	2022	RGB-Image	Diffusion	Super-Resolution, Deblurring, Inpainting, Colorization	[code]
Palette [243]	2022	RGB-Image	Diffusion	Colorization, Inpainting, Uncropping, JPEG Restoration	[code]
DiffC [244]	2022	RGB-Image	SDE	Compression	-
SRDiff [245]	2021	RGB-Image	Diffusion	Super-Resolution -	
RePaint [246]	2022	RGB-Image	Diffusion	Inpainting, Super-resolution, Edge to Image Translation	[code]
FSDM [247]	2022	RGB-Image	Diffusion	Few-shot Generation	-
CARD [248]	2022	RGB-Image	Diffusion	Conditional Generation	[code]
GLIDE [49]	2022	RGB-Image	Diffusion	Conditional Generation	[code]
LSGM [35]	2022	RGB-Image	SDE	UnConditional & Conditional Generation	[code]
SegDiff [249]	2022	RGB-Image	Diffusion	Segmentation	-
VQ-Diffusion [94]	2022	VQ Data	Diffusion	Text-to-Image Synthesis	[code]
DreamFusion [152]	2023	VQ Data	Diffusion	Text-to-Image Synthesis	[code]
Text-to-Sign VQ [126]	2022	VQ Data	Diffusion	Conditional Pose Generation	-
Improved VQ-Diff [95]	2022	VQ Data	Diffusion	Text-to-Image Synthesis	-
Luo's Model [151]	2021	Point Cloud	Diffusion	Point Cloud Generation	[code]
PVD [147]	2022	Point Cloud	Diffusion	Point Cloud Generation, Point-Voxel representation	[code]
PDR [144]	2022	Point Cloud	Diffusion	Point Cloud Completion	[code]
Cheng's Model [250]	2022	Point Cloud	Diffusion	Point Cloud Generation	[code]
Luo's Model[150]	2022	Point Cloud	Score	Point Cloud Denoising	[code]
VDM [12]	2022	Video	Diffusion	Text-Conditioned Video Generation	[code]
RVD [157]	2022	Video	Diffusion	Video Forecasting, Video compression	[code]
FDM [156]	2022	Video	Diffusion	Video Forecasting, Long-range Video modeling	-
MCVD [155]	2022	Video	Diffusion	Video Prediction, Video Generation, Video Interpolation	[code]
RaMViD [158]	2022	Video	SDE	Conditional Generation	-
Score-MRI [161]	2022	MRI	SDE	MRI Reconstruction	[code]
Song's Model [251]	2022	MRI, CT	SDE	MRI Reconstruction, CT Reconstruction	[code]
R2D2+ [167]	2022	MRI	SDE	MRI Denoising	-
Sequence Modeling					
Diffusion-LM [15]	2022	Text	Diffusion	Conditional Text Generation	[code]
Bit Diffusion [252]	2022	Text	Diffusion	Image-Conditional Text Generation	[code]
D3PM [16]	2021	Text	Diffusion	Text Generation	-
Argmax [92]	2021	Text	Diffusion	Test Segmantation, Text Generation	[code]
CSDI [17]	2021	Time Series	Diffusion	Series Imputation	[code]
SSSD [253]	2022	Time Series	Diffusion	Series Imputation	[code]
CSDE [254]	2022	Time Series	SDE	Series Imputation, Series Predicton	-
Audio & Speech					
WaveGrad [19]	2020	Audio	Diffusion	Conditional Wave Generation	[code]
DiffWave [186]	2021	Audio	Diffusion	Conditional & Unconditional Wave Generation	[code]
GradTTS [20]	2021	Audio	SDE	Wave Generation	[code]
Diff-TTS [193]	2021	Audio	Diffusion	non-AR mel-Spectrogram Generation, Speech Synthesis	-
DiffVC [215]	2022	Audio	SDE	Voice conversion	[code]
DiffSVC [190]	2022	Audio	Diffusion	Voice Conversion	[code]
DiffSinger [187]	2022	Audio	Diffusion	Singing Voice Synthesis	[code]
Diffsound [197]	2021	Audio	Diffusion	Text-to-sound Generation tasks	[code]
EdiTTS [192]	2022	Audio	SDE	fine-grained pitch, content editing	[code]
Guided-TTS [196]	2022	Audio	SDE	Conditional Speech Generation	-
Guided-TTS2 [195]	2022	Audio	SDE	Conditional Speech Generation	-
Levkovitch's Model [255]	2022	Audio	SDE	Spectrograms-Voice Generation	[code]
SpecGrad [194]	2022	Audio	Diffusion	Spectrograms-Voice Generation	[code]
ItoTTS [191]	2022	Audio	SDE	Spectrograms-Voice Generation	-
ProDiff [188]	2022	Audio	Diffusion	Text-to-Speech Synthesis	[code]
BinauralGrad [189]	2022	Audio	Diffusion	Binaural Audio Synthesis	-
AI For Science					
ConfGF [256]	2021	Molecular	Score	Conformation Generation	[code]
DGSM [257]	2022	Molecular	Score	Conformation Generation, Sidechain Generation	-
GeoDiff [23]	2022	Molecular	Diffusion	Conformation Generation	[code]
EDM [198]	2022	Molecular	SDE	Conformation Generation	[code]
Torsional Diff [200]	2022	Molecular	Diffusion	Molecular Generation	[code]
DiffDock [203]	2022	Molecular&protein	Diffusion	Conformation Generation, molecular docking	[code]
CDVAE [258]	2022	Protein	Score	Periodic Material Generation	[code]
Luo's Model [24]	2022	Protein	Diffusion	CDR Generation	-
Anand's Model [135]	2022	Protein	Diffusion	Protein Sequence and Structure Generation	-
ProteinSGM [199]	2022	Protein	SDE	de novo protein design	-
DiffFolding [202]	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
$\tilde{\alpha}_t$	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
$q(x_t\|x_{t-1})$	DDPM forward step at time t
$p(x_{t-1}\|x_t)$	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