

# A Survey on Graph Diffusion Models: Generative AI in Science for Molecule, Protein and Material

Mengchun Zhang\*, Maryam Qamar\*, Taegoo Kang, Yuna Jung, Chenshuang Zhang, Sung-Ho Bae†, Chaoning Zhang†

**Abstract**—Diffusion models have become a new SOTA generative modeling method in various fields, for which there are multiple survey works that provide an overall survey. With the number of articles on diffusion models increasing exponentially in the past few years, there is an increasing need for survey of diffusion models on specific fields. In this work, we are committed to conducting a survey on the graph diffusion models. Even though our focus is to cover the progress of diffusion models in graphs, we first briefly summarize how other generative modeling methods are used for graphs. After that, we introduce the mechanism of diffusion models in various forms, which facilitates the discussion on the graph diffusion models. The applications of graph diffusion models mainly fall into the category of AI-generated content (AIGC) in science, for which we mainly focus on how graph diffusion models are utilized for generating molecules and proteins but also cover other cases, including materials design. Moreover, we discuss the issue of evaluating diffusion models in the graph domain and the existing challenges.

**Index Terms**—Survey, Diffusion Models, Graph Generation, Molecule Generation, Protein Generation, Generative AI

## 1 INTRODUCTION

GRAPHS are ubiquitous in various scenarios. Due to the great expressive power of graphs, a growing number of studies have merged using machine learning for graph analysis and generation in recent years [1], [2], [3]. Graphs are broadly implemented in a considerable number of research domains, including social networks in social sciences [3], [4], molecular structures in chemistry and materials [5], knowledge bases [6] and protein interaction networks in biochemistry [7]. Graphs have an irregular structure with varying numbers of nodes and edges, which make it difficult to apply standard convolution or pooling operations [8], [9]. Moreover, graphs can be large in scale, requiring specialized techniques to process and analyze them efficiently [10], [11], [12]. Moreover, graphs exhibit diverse and heterogeneous properties, making it challenging to develop models that can generalize well across different types of graphs [13]. These challenges have motivated researchers to develop new approaches and architectures for deep learning on graph data, such as graph convolutional networks and graph attention networks, that can effectively capture the structure and properties of graphs [13], [14], [15], [16], [17].

Representing molecules and proteins as graphs enables researchers to harness the power of graph neural networks to learn their structural and functional properties, which can then be employed to create novel molecules and proteins with expected characteristics [18], [19]. Molecular and

protein generation tasks involve employing computational approaches to design novel molecules and proteins or optimize existing ones, which are more efficient and cost-effective than conventional experimental techniques [20], [21]. Researchers can create vast libraries of potential molecules and proteins by leveraging graph-based deep-learning techniques. This method can considerably accelerate the finding of novel drugs and materials and potentially revolutionize the field of molecular design [22], [23].

Diffusion models have become an essential tool in graph-based deep learning, providing a method to model the diffusion of information across the nodes and edges of a graph. With applications varying from representation learning to graph generation and dynamic graph processing, diffusion models have demonstrated versatility in addressing various challenges in graph-based deep learning. Furthermore, diffusion models have been shown to improve the scalability and efficiency of graph neural networks, specifically for large-scale graphs [20], [24]. They also provide a way to incorporate global information into the learning process, leading to more precise predictions and better performance on downstream tasks.

There are extensive works covering the graph generation techniques based on autoregressive, VAE, and GAN models [25], [26], [27], [28]. However, the literature lacks in a comprehensive review of this task regarding the emerging and promising diffusion models. In this survey, we strive to remedy this knowledge gap. We approach this by spelling out a brief summary of the aforementioned methods from graph generation perspective along with the challenges they face in section 2. The survey then progresses to the main topic of diffusion based approaches where section 3 introduces the diffusion models succinctly and then we move on to their deep discussion for graph generation in section 4 and 5, both from structural and application specific point of view. The survey then proceeds to explain the currently popular

\* indicates equal contribution

† Correspondence author(s)

- Mengchun Zhang and Chenshuang Zhang are with KAIST (South Korea). Maryam Qamar, Taegoo Kang, Yuna Jung, Sung-Ho Bae, Chaoning Zhang are with Kyung Hee University (South Korea).
- Chaoning Zhang (chaoningzhang1990@gmail.com) and Sung-Ho Bae (shbae@khu.ac.kr) are correspondence authors.

evaluation datasets and metrics in section 6, concluding with the challenges and research questions still open to the research community in section 7. By authoring this in-depth survey, we hope to provide a comprehensive resource for researchers interested in exploring the potential of diffusion models for graph generation and related tasks.

**Related survey works.** Multiple works [29], [30], [31] survey the progress of diffusion models in all fields. There are also multiple field-specific survey works on diffusion models, including text-to-image diffusion models [32], audio diffusion models [33]. Complementary to them, this work conducts a survey on graph diffusion models. From the perspective of AI-generated content (AIGC), this survey is also related to generative AI (see [34] for a survey) and ChatGPT (see [35] for a survey). Conceptually, diffusion models are also highly related to masked autoencoders (see [36] for a survey), which also perform denoising.

## 2 DEEP GRAPH GENERATION

Generative models such as the Erdos-Renyi [37], Watts-Strogatz [38], Barabasi-Albert [39], and R-MAT [40] observe manually specific significant properties of actual graphs and construct algorithms accordingly. While these models may successfully capture some features of actual graphs, they lack the flexibility to model different graph properties [4], [16], [41], [41]. Furthermore, generative models may struggle to generate graphs that indicate observed connectivity patterns or behaviors [42]. As a result, there is a need to develop more sophisticated generative models that can integrate a more comprehensive scope of graph properties and yield more realistic results [43].

In this section, we briefly review the deep graph generation models according to the algorithmic taxonomy. Regardless of data type, there are four popular generative modeling methods, namely, Auto-Regressive Models [44], Variational Auto-Encoders [45], Normalizing Flows [46], and Generative Adversarial Networks [47]. Here, we summarize the unified framework for deep graph generation models as shown in Figure 1.

### 2.1 Auto-Regressive Model

Auto-Regressive model (AR) factorizes the generation process into a sequence of steps, which predicts the next output based on the previous output in the sequence [44]. The most common objective function used in Auto-Regressive generation models is the maximum likelihood estimation (MLE), which leads the model to generate sequences that are similar to those in the training data [48], [49], [50]. In the field of graph generation, GraphRNN [25] uses two Recurrent Neural Networks (RNNs) in order to learn the distribution of numerous representations of graphs that vary with node orderings. Each RNN learns to generate the sequence of nodes and edges for graph structures including newly added nodes, so that it can capture the complex structural dependencies of the graph [51], [52]. Inspired by GraphRNN [25], MolecularRNN [26] adopts this framework to generate molecular structure. In addition, they use policy gradient optimization algorithm to produce molecules that satisfy certain conditions.

GraphAF [53] combines AR and Normalizing Flows, defining the sampling process in a sequential way using flow-based graph neural network. Similar to [26], they use reinforcement learning to generate valid and conditional molecules. Despite AR showing remarkable performance in graph generation, they suffer from designing permutation-invariant graph distribution since they treat graph as a sequence [25], [54], [55].

### 2.2 Variational Auto Encoder

Variational Auto Encoder (VAEs) consists of an encoder that maps the input data into a low-dimensional latent space and a decoder that maps latent representation back to the original data space [45], [56]. VAEs are optimized by minimizing the reconstruction error between the generated output and the input data, while regularizing the latent space to follow a prior probability distribution. VAE-based methods are categorized into two ways: one-shot [27], [57], [58] and sequential generation [59], [60]. For the one-shot type, VGAE [61] generates the edge probability based on the relationships between nodes' embedding. In GraphVAE [27], the decoder outputs a probabilistic fully connected graph in one-step. To find the best graph, they use approximate graph matching between nodes of graphs based on the similarities of their node pairs. When generating a graph, there is often a constraint demand that only specific combinations of the nodes and edges are valid in context [27], [55], [59]. RGVAE [57] generates a semantically valid graph by adding a penalty term that regularizes the output distribution of the decoder to the objective function of VAEs. Graphite [58] repeats the process that constructs an intermediate weighted graph and iteratively refines the reconstructed feature matrix via message passing. The final distribution of graph is obtained via the inner product of feature matrix. The other category is sequential generation method. CGVAE [59], node-by-node generator, repeats the process that chooses which node is connected to the current node and selects and labels edges between nodes to generate a graph. By contrast, JT-VAE [60], a motif-sequence-based generator, generates molecular graphs by a procedure that generates a tree-structured scaffold over chemical substructures first and then combines them into a molecule. But Autoencoder-based graph generation methods have limitations in that they are not scalable to be applied on large graphs.

### 2.3 Normalizing Flows

Normalizing Flow applies a sequence of invertible transformations to a simple probability distribution to model more complex probability distributions using encoder [46], [62], [63], [64]. Its decoder is essentially an inverse function of the encoder. GraphNVP [65] first adopts normalizing flows to molecular graph generation. To address the lack of validity constraint in the one-shot generation model, they propose a two-step generation process which is consisted of generating a graph structure and node features according to the generated structure. GNF [66] proposes a new graph neural network reversible for prediction and generation. They use graph auto-encoder that maps discrete graph structure to continuous

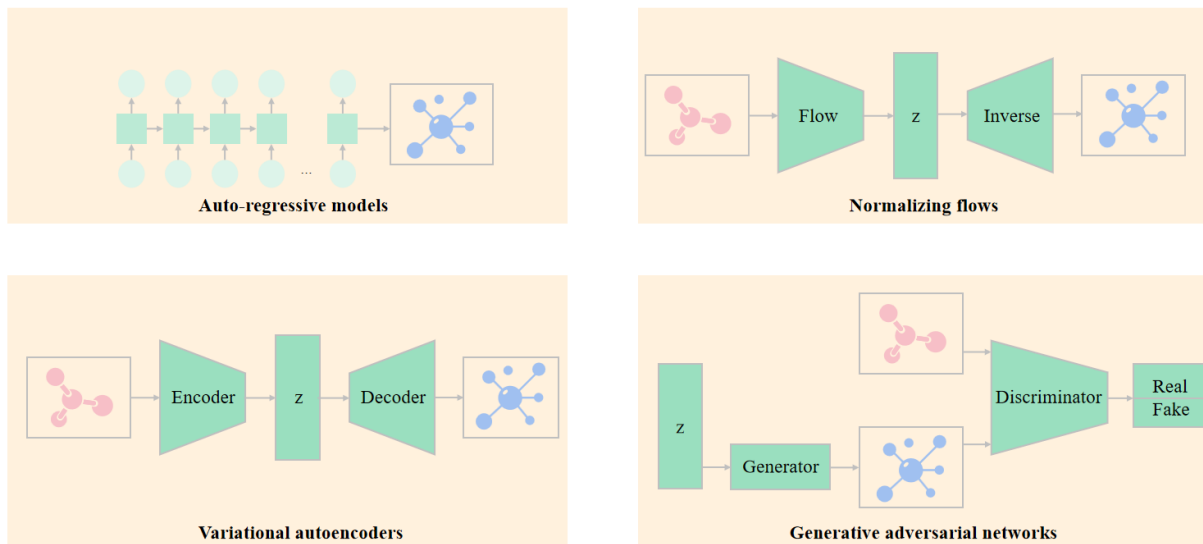


Fig. 1. Graph generation models for deep graph generation

graph embedding. Then GNF learns the distribution of graph embedding, which is obtained by a learned graph autoencoder. MoFlow [67] uses two invertible flow-based networks, one is for generating bonds and another is for generating atoms with given bonds. They use post-hoc method to correct the validity of the molecule. Unlike most previous methods, which map discrete graph structure to continuous latent variables, GraphDF [68] uses discrete latent variables using invertible transform. One of the critical limitations of Normalizing Flows is that there are constraints on generating valid graphs due to its one-shot nature [66].

## 2.4 Generative Adversarial Networks

Generative Adversarial Networks (GANs) are composed of two main components: a generator network that generates synthetic data for deceiving the discriminator, and a discriminator network that assesses the authenticity of the generated data [69], [70], [71], [72]. GAN-based methods utilize the property of discriminator to generate a graph to have desired properties. MolGAN [28] is the first address to generate a small molecular graph using GANs. The generator samples from the prior distribution and outputs the entire probabilistic graph at once. The discriminator distinguishes whether the molecule comes from the generator or the dataset. And they further utilize reward network for RL-based optimization towards desired chemical properties. There are also graph-to-graph translation methods that generate graphs with better properties from source graphs, such as VJTNN [73] and Mole-CycleGAN [74]. Unlike the above methods that generate graphs, there are methods of generating graphs via random walks that are invariant under node reordering. NetGAN [42] generates random walks that are plausible in the real graph, and then they are used to produce an adjacency matrix of a newly generated graph. But GAN-based methods are challenging for graph

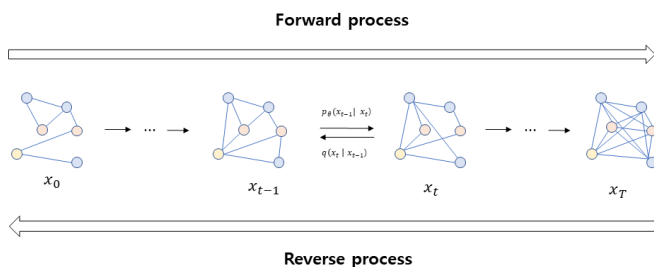


Fig. 2. Illustration of diffusion models for deep graph generation.

generation tasks because of the need for likelihood-based optimization.

## 3 BACKGROUND ON DIFFUSION MODELS

Diffusion models are a class of generative models that gradually introduce noise into data until it conforms to a prior distribution [75]. The models then learn to reverse this process to generate viable samples. By leveraging the power of diffusion models, researchers can create generative models that can accurately capture the underlying structure of complex datasets and produce high-quality, realistic samples.

There are three sub-types: denoising diffusion probabilistic models (DDPMs) [76], [77], [78], [79], [80], score-based generative models (SGMs) [81], [82], and stochastic differential equations (SDEs) [83], [84], [85], which differ in how they implement the forward and backward diffusion pass. Figure 2 presents a generic framework that is based on three categories.

### 3.1 Denoising Diffusion Probabilistic Models(DDPM)

A denoising diffusion probabilistic model (DDPM) is a generative model that can generate novel data samples from a given data distribution that employs two Markov chains [75].

**Forward process.** First, the original data is transformed into a simpler prior distribution by gradually adding noise over a fixed number of diffusion steps  $T$  according to a variance schedule  $\beta$ . This forward pass aims to transform the input data into a distribution that can be easily sampled using standard techniques such as sampling from a Gaussian distribution. Given a data distribution  $\mathbf{x}_0 \sim q(\mathbf{x}_0)$ , the forward process generates  $\mathbf{x}_T$  with transition kernel  $q(x_t|x_{t-1})$  and can be defined as follows [75]:

$$q(x_{1:T}|x_0) := \prod_{t=1}^T q(x_t|x_{t-1}), \quad (1)$$

$$q(x_t|x_{t-1}) := \mathcal{N}(x_t; \sqrt{1 - \beta_t}x_{t-1}, \beta_t I) \quad (2)$$

where  $\beta_t$  is a hyper-parameter. To simplify the discussion, we focus on the use of Gaussian noise as the transition kernels denoted as  $\mathcal{N}$  in 2. When  $\alpha_t := 1 - \beta_t$  and  $\bar{\alpha}_t := \prod_{s=0}^t \alpha_s$ :

$$q(x_t|x_0) := \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0, (1 - \bar{\alpha}_t)I) \quad (3)$$

**Reverse process.** To generate new samples from the data distribution, the reverse process is performed. The reverse pass makes use of a neural network trained to predict the noise that was added at each step in the forward pass. This pass removes the noise at each step in reverse order until the original data is recovered. To generate  $p_\theta(x_0)$  that follows the true data distribution  $q(x_0)$ , we start with  $p_\theta(T)$  and optimize the model with the following objective, as described in [75]:

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) := \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(x_t, t), \sigma_t^2 I) \quad (4)$$

As stated in [86], the model's optimization objective can be expressed as:

$$E_{t \sim \mathcal{U}(1, T), \mathbf{x}_0 \sim q(\mathbf{x}_0), \epsilon \sim \mathcal{N}(\mathbf{0}, I)} \lambda(t) \|\epsilon - \epsilon_\theta(\mathbf{x}_t, t)\|^2 \quad (5)$$

DiGress [76] and [77] adopt DDPM technique for graph diffusion though they add discrete noise instead of continuous Gaussian.

### 3.2 Score-Based Generative Models(SGMs)

Score-based generative models are a class of probabilistic models that employ a score function, also referred to as an energy function or an unnormalized probability density function, to illustrate the probability distribution of the data [87], [88].

The score function of a probability density function  $p(x)$  is defined as the gradient of the logarithm of the probability density,  $\nabla_x \log p(x)$ . A noise-conditional score network is a type of deep neural network that is trained to estimate the score function  $\nabla_{x_t} \log q(x_t)$ , which represents the gradient of the logarithm of the conditional probability density function  $q(x_t)$  of the data given the noise. The network is denoted as  $s_\theta(x, t)$ , where  $\theta$  represents the learnable parameters of the network and  $x$  means the input data at time  $t$ . The network is generally trained using maximum likelihood estimation or a variant thereof, such as noise-contrastive estimation or denoising score matching.

Employing denoising score matching and similar notations, as represented in 5, the training objective for the score network is given by:

$$E_{t \sim \mathcal{U}([1, T]), \mathbf{x}_0 \sim q(\mathbf{x}_0), \epsilon \sim \mathcal{N}(\mathbf{0}, I)} [\lambda(t) \|\epsilon + \sigma_t s_\theta(\mathbf{x}_t, t)\|^2] \quad (6)$$

The pioneer score-based graph generation model, EDP-GNN [81], leverages GIN [89] and makes use of annealed Langevin dynamic sampling to diffuse the edges. In contrast, NVDiff [82], utilizes an attention-based score generative model to exclusively diffuse the latent representation of node vectors.

### 3.3 Stochastic Differential Equations (SDEs)

Stochastic Differential Equations (SDEs) are a class of mathematical models that characterize the development of a system over time under the effect of random noise. A Score-based Stochastic Differential Equation (Score SDE) is a type of SDE where the drift term is defined as the negative gradient of a score function, and the diffusion term is a function of time [90].

**Forward SDE.** In mathematics, a forward SDE is a continuous-time dynamical system that describes the evolution of a state variable over time, where both deterministic and stochastic forces govern the evolution. The forward SDE can be defined as follows [90]:

$$dx = f(x, t)dt + g(t)d\bar{w} \quad (7)$$

**Reverse SDE.** A reverse stochastic differential equation (SDE) is a continuous-time dynamical system that progresses backward. It is generally employed to calculate the score function for the forward SDE, which can subsequently be operated to produce samples from the conditional distribution. The reverse SDE can be defined as follows [90]

$$dx = [f(x, t) - g(t)^2 \nabla_x \log p_t(x)]dt + g(t)d\bar{w} \quad (8)$$

The score function is estimated by parameterizing a score model  $s_\theta(x_t, t)$ , which involves generalizing the score matching objective in 9 to continuous time as follows [29]:

$$E_{t \sim \mathcal{U}[0, T], \mathbf{x}_t \sim q(\mathbf{x}_t|\mathbf{x}_0)} [\lambda(t) \|s_\theta(\mathbf{x}_t, t) - \nabla_{x_t} \log q_{0t}(\mathbf{x}_t|\mathbf{x}_0)\|^2] \quad (9)$$

GraphGDP [83], diffuses the graph edges via stochastic differential equations (SDE). For the sampling process tailored to graph data, the authors designed a position-enhanced graph score network employing the structure and position information of graphs and the noisy adjacency matrices. GDSS [84] formulated a system of two stochastic differential equations, one for each graph features generation and adjacency matrix generation in parallel for diffusing both nodes and edges. Another SDE based model GSDM [85], addressing the problem of Gaussian noise transforming the adjacency matrix into a dense one quickly, restricts the insertion of Gaussian noise to the eigenvalue matrix of the adjacency matrix.



## 4 GRAPH DIFFUSION MODELS

Multiple works [25], [27], [28], [42], [59], [61], [91], [92], [93], [94] have demonstrated various graph generation qualities depending on the sequence of node generation. Autoregressive models, in particular, perform optimally [83]. However, they are ineffective in capturing the permutation invariant properties. Diffusion models have come to rescue here, the first score-based model for permutation invariant graph generation, EDP-GNN [81], adapts GIN [89] and employs annealed Langevin dynamic sampling. It assumes undirected graphs, adding Gaussian noise only to the upper triangular part of the adjacency matrix for edge diffusion, and quantizes the continuous adjacency matrix generated when sampling. On predicting edgewise features, EDP-GNNs are empirically shown to be more expressive compared with vanilla ones. Additionally, sample quality is found to be comparable with state-of-the-art models.

GraphGDP [83] also attempted to address the generation of permutation invariant graphs, where state-of-the-art autoregressive models failed. However, the proposed continuous-time generative diffusion model employed stochastic differential equations (SDE) for edge diffusion. Moreover, for sampling, a score network is specifically designed utilizing the structure and position information.

Although diffusion models show great promise, the unique intrinsic properties of graph data call for a few adaptations in the standard diffusion process to make them fit for graph generation tasks. This section next details the modifications thus called for.

### 4.1 Discrete Diffusion

It is observed that the standard diffusion models have difficulties in capturing the structural properties of the graph data because it relies on a continuous Gaussian noise process which creates fully connected, noisy graphs where structural information is not defined [20]. DiGress [76], the first discrete graph generation model, features a Markov process noise model where, similar to the way noise is infused independently in each pixel in image diffusion, all progressive noise addition steps are performed independently on each node or edge. The denoising process involves training a graph transformer network to predict the clean graph from a noisy input. It delivers state-of-the-art performance on both molecular and non-molecular datasets. Meanwhile, another concurrent work [77] validates the improvement in the sampling quality and denoising efficiency with the use of discrete noise.

Furthermore, a work [78] in the protein design domain modifies the diffusion process by modeling it as random masking of a portion of the residues. This fraction is decided by linear interpolation between 0 and 1 depending on the time step  $t$ . For sampling, the same procedure is reversed, starting with the completely masked residues at time  $T$ . These works use the Markov transition probability matrix to implement discrete noise, and the objective becomes a simple cross-entropy loss.

### 4.2 Low Rank Diffusion

For a typical image diffusion model, data is corrupted by adding full-rank Gaussian noise [20], [95]. However,

diffusing graphs with full-rank noise is detrimental to the training of the score network. In the image diffusion model tasks, the early- and mid-diffusion process stages are not influenced by the inclusion of full-rank Gaussian noise in the image recognition [87], [90], [95]. Nevertheless, when perturbing the graph with full-rank Gaussian noise, the fatal noise is injected into the unsupported regions and is detrimental to the training of the score network [87], [96], [96]. It compromises graphic structure and feature representation. To resolve this issue, rather than doing a full-rank diffusion of the adjacency matrix, GSDM [85] restricts the insertion of Gaussian noise to the eigenvalue matrix, which is the spectral decomposition of the adjacency matrix. This approach is effective in both generic graph and molecule generation tasks.

### 4.3 Roto-Translation Equivariance and Invariance

With regards to molecular graph generation specifically, the distribution of molecules should be invariant to rotation and translation, that is, no matter what Euclidean transition is applied to the molecules, their probability distribution should not change [8], [97]. Thus generative models need to make this property true when generating 3D molecular data. The most commonly used method to satisfy these invariances is to employ a roto-translation equivariant score network. [98] proved that an invariant density can be produced by dynamics along an equivariant Gaussian Markov kernel beginning from an invariant standard density. This means that we can ensure the invariance property by making our neural networks equivariant. Consequently, in GeoDiff [98], an equivariant graph convolution network termed as graph field network (GNF) is designed. Inspired by the aforementioned principle, several works [79], [99], [100], [101] utilize either equivariant graph neural network [102] or its modification as their score network. Another trick, is that, zero center of mass (CoM) is also explored for this purpose, as it moves the center of atomic coordinates to zero, ensuring translation invariance. Another approach, ConfGF [103], formulates the problem as diffusing interatomic distances instead of atomic coordinates. Based on the observation that those two gradient fields are connected via chain rule, their score networks are roto-translation invariant.

## 5 APPLICATIONS

### 5.1 Molecules

Deep learning approaches are currently having a critical impact on molecular science, with an increasing number of studies developing deep learning models to analyze or generate molecules [104], [105]. These methods can capture high-level structural features and patterns in large data sets by utilizing artificial neural networks, which allows them to generalize across a wide range of molecular structures and properties [104]. For instance, in drug discovery, deep learning models illustrate considerable promise in identifying novel drug candidates with improved effectiveness and decreased side effects [106].

### 5.1.1 Molecular Conformation Generation

In the field of biological and chemical information, predicting molecular conformations is a fundamental problem [103]. Previous graph generation methods have suffered from several drawbacks that are detrimental to molecular generation. Among them, Markov chain Monte Carlo (MCMC) has high computational cost [107]; the deep generation model based on intermediate variables leads to implausible generation results due to the errors generated by the multi-stage generation process; and the gradient-based deep generation model violates mathematical theorems.

According to the conventional molecular dynamics simulation force field method, [103] presents ConfGF, a unique method for small molecule conformation generation. ConfGF addresses the issue of multistage cumulative error. The gradients of interatomic distances are first estimated by denoising score matching, and then the gradients of coordinates are obtained. They estimate the gradient field of log density of atomic coordinates and directly generate conformation through Langevin dynamics.

Past approaches [108], [109] generally neglect the interactions between long-range non-bonded atoms. Following the principle of ConfGF [103], DGSM [110] uses a score-matching method to directly evaluate the gradient field of the log density of atomic coordinates. They consider not only local interaction between bonded atoms but also long-range interaction between non-bonded atoms. To model long-range interactions in molecules, they augment graph structure dynamically during training and inference. When reconstructing graph structure, they assume that each atom only interacts with all atoms within the cutoff distance, i.e., only atoms within the cut-off distance were connected. The overall structure allows for effective training in an end-to-end manner.

GeoDiff [98] is a probabilistic model of the generated molecular conformations that considers each atom as a particle. To overcome multi-step errors caused by distances and conformations, GeoDiff [98] directly changes the atoms to ensure rotation-translation invariance. ConfGF [103] has accumulated errors in capturing roto-translation invariance using intermediate geometric variables (atom distances) when generating atomic coordinates. Therefore, GeoDiff [98] restores realistic conformation directly from the noise distribution. It learns the inversion directly by treating the diffusion process as a Markov chain.

The challenge of utilizing diffusion models on molecules lies in that the model does not work by adding the same Gaussian noise to every feature. Consequently, EDM [101], a novel method for generating 3D molecular structures, uses DDPM to add different noise for various features. EDM employs diffusion processes to sample molecular conformations by transforming initial points in space on both continuous coordinates and categorical atom types.

Unlike the prior diffusion process on Euclidean space, Torsional diffusion [111] formulates the diffusion process on the torus defined by torsion angles, as shown in Figure 3. The diffusion process over conformers operates only on the torsion angle, with other degrees of freedom fixed, such as bond lengths, bond angles, and cycle conformation. In reverse SDE, they leverage the extrinsic-to-intrinsic score model, which takes the conformer’s 3D structure in Eu-

clidean space as input, and predicts a score on a torsional space to generate a new conformer through an intrinsic torsional update. This has the advantage of being able to generate conformers through two orders of magnitude fewer denoising steps compared to Geodiff [98], sota Euclidean diffusion approach

### 5.1.2 Drug Discovery

Generative diffusion models allow the analysis of large databases of chemical structures and properties and the generation of new compounds optimally targeted for specific drug targets [60], [87]. By harnessing the power of diffusion-based generative models, a broad field of chemistry can be rapidly explored, and promising drug candidates can be identified that would be difficult to discover using traditional methods [112], [113].

A prevalent issue with existing models of molecular generation is the mutual correlation of interatomic distances and interatomic forces. For instance, the interatomic forces are dominated by the van der Waals force when the interatomic distance is less than a certain threshold and chemical bonding dominates the interatomic forces when the distance is greater than a certain threshold. MDM [114] addresses the issues of capturing the potential of interatomic forces and lack of diversity. MDM encodes the interatomic forces of different strengths by enhancing the potential interatomic forces with a dual equivalence encoder. To augment generative variety, they introduce latent variables in the process of diffusion and generation. MDM outperforms the state-of-the-art model EDM [101], which is unable to establish chemical bonds through atom pair distances, especially on drug-like datasets.

An effective paradigm for drug design is molecular fragment-based drug discovery, which is the design of linkers between unlinked fragments. DiffLinker [100] is the utilization of diffusion model to design molecular linkers. Conventional molecular linkers connect only pairs of fragments, while DiffLinker allows to ligate any number of fragments. The diffusion framework of DiffLinker is comparable to the EDM [101], and the denoising model is the same as the EGNN [115]. In addition, DiffLinker has an extra module to predict the linker size and automatically determine the atomic number of linkers and their connection points.

Structure-based drug design (SBDD) is the process of designing and optimizing chemical structures with the aim of candidate drug synthesis [116]. DiffSBDD [117] generates small molecule ligands with high affinity and particularity according to the protein pocket as a condition. The approach utilizes an equivariant DDPM to generate molecules and binding conformations specific to the protein pocket. The protein and ligand point clouds are represented as fully connected graphs and further processed by EGNNs. The prior and transition probabilities are modeled using an isotropic Gaussian to ensure translation equivariance. DiffSBDD is a powerful tool for generating high-quality ligands for specific protein targets.

It is insufficient to generate accurate molecules with the desired properties by simply using a single conditional generative model, such as EDM [101]. To enable more flexible and controllable generation of molecules, EEGSDE [118] formalizes the framework as an equivariant SDE to follow the

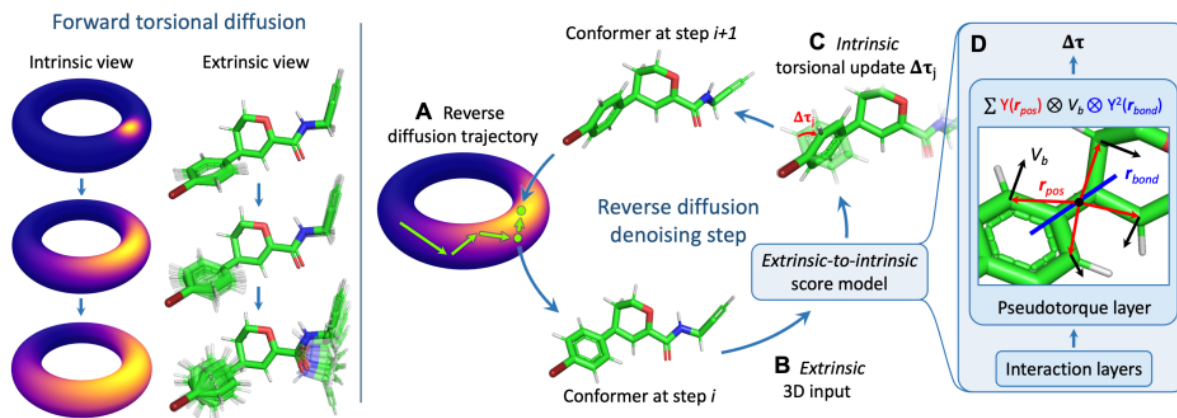


Fig. 3. Overview of torsional diffusion (figure obtained from [111])

guidance of the energy function which is carefully designed that the generated molecules are targeted to quantum properties, molecular structures, and even a combination of them in each step. Additionally, gradient descent is performed on energy function so that the generated molecules are encouraged to have low energy. Under the guidance of energy function, EEGSDE [118] benefit applications such as drug or material discovery over EDM [101]. (it can be written for section 4 because it can be compared by EDM (DDPM-based method)).

GCDM [119] proposes a novel method for generating 3D molecular structures using a diffusion process that considers the molecule's geometry. The model operates a graph convolutional neural network to encode the 2D structure of a molecule and then uses a diffusion process to generate a 3D structure compatible with the input 2D structure. The method is shown to create high-quality 3D structures that are competitive with state-of-the-art methods while requiring fewer computational resources. The proposed approach has potential applications in drug discovery and material design.

## 5.2 Protein Design

Technically, proteins are also a special kind of molecules. In contrast to the proteins discussed above, proteins often have a complex structure which consists of amino acid chains and they play pivotal roles in cellular processes. Many aspects of the protein generation problem have seen good progress but there remain certain challenges [20].

### 5.2.1 Structure and Sequence

The function of a protein is mainly determined by its three-dimensional structure. Considering the importance, one of the key scientific challenges in biotechnology is generating new proteins with specified functional structures and sequences [120], [121], [122], [123]. Traditional solutions call for expert participation, while, even though machine learning methods relinquished the need for expert help, they lack in generating large structures along with being computationally expensive and necessitating experimental validation [122]. Moreover, learning spatial ordering of the atoms is requisite to learn a viable chain structure in

protein generation tasks, recent diffusion models though successfully generate small molecules do not impose this and consequently are not up to the task.

A number of priors on the distribution of the atom positions constrain the 3D structure of a protein [78] proposes a diffusion based generative model able to construct feasible and diverse proteins. Specifically, to generate feature embeddings given the constraints, a transformer model with triangular self-attention is employed [124]. Taking into consideration the different natures of image and protein data, the attention process is replaced by Invariant Point Attention [122]. The output of the constraint network and the structure at time  $t$  are utilized as the conditions in the modified diffusion process. Specifically, diffusion is done via random masking of residues' fractions, and for sampling the process is reversed. Experiments are conducted for multiple tasks such as 3D design given the protein specification alone, modifying given structures, and joint modeling of structure and sequence showing promising performance [122], [125].

In contrast to modeling the protein backbone via 3D coordinates, a denoising diffusion model [126] mimicking the natural folding process, defines it via six inter-residue angles reflecting the arrangement of amino acid residues. Given the position of the current residue, each vector of these angles defines the relative position of backbone atoms in the subsequent residue. The shift and rotational invariance inherent in this representation eliminates the need for complicated equivariant networks enabling the authors to employ a simple bidirectional transformer architecture. A regression head is used for outputting the angles from the representations learned by the transformer network. Despite showing reasonable performance, the use of angles to represent protein backbone would potentially mean that early errors significantly affect the overall generated structure. Moreover, generated structures are also observed to have collisions.

### 5.2.2 Motif conditioned structure design

Apart from general protein structure design, the design of a stable scaffold/backbone to support a desired structural protein fragment, known as a motif is necessary, particu-

masking

predicting dependent on previous aa



larly, for the design of enzymes, and vaccines [79], features ProtDiff, which adapts the equivariant graph neural network [102] for the diffusion of protein backbones. It is the first machine-learning technique, which builds reasonably long protein backbones conditioned on motifs. The conditional sampling of scaffolds is approached as a Sequential Monte Carlo problem in this work, it implements SMCDiff as a particle filtering technique used to sample scaffolds that are more consistent with the motif. The approach is reported to generate valid scaffolds for the target motif. Additionally, unconditional sampling indicated the model’s reasonable capacity to capture the underlying distribution.

### 5.2.3 Antibody Design

Another important task in protein generation domain is the design of antibodies, the proteins playing a vital role in the human immune system [128]. The specificity of antibodies to the antigens is determined by the Complementarity Determining Regions (CDRs) [129], [130]. In order to create medicinal antibodies capable of effectively targeting specific antigens, it is crucial to design these CDRs [131]. Traditional methods involved complex energy functions and thus had been inefficient and prone to providing suboptimal solutions. Moreover, though recent deep learning methods have shown improved performance, these are still limited in their capacity to generate antibodies based on previously observed ones only. It is identified in [80] that a number of challenges for practical use scenario, that is, a generative model should output CDRs compatible with the 3D structure of antigen, consider position and orientation of amino acids, and given an antibody, be able to optimize its binding affinity. To this end, [80] proposes the first deep learning model DiffAb to perform joint design of antibody sequence and structure conditioned on the 3D structures of the antigen. It takes an antigen-antibody complex as input and diffuses the amino acid type, position, and orientation on CDRs.

### 5.2.4 Molecular Docking

Novel biological discoveries can benefit hugely from the advances in the generation of protein binding molecules, called ligands, and 3D structure of protein-ligand molecular complex [132]. DiffBP [99], a DDPM model, is proposed for generating ligands, and to account for the fact that in the molecular systems, the interactions among atoms are influenced by all other atoms. They generate molecules at the full atom level. For full-atom generation, though, it is necessary to assign the atom numbers of the binding molecules before diffusion and denoising process. Therefore, in the pre-generation process before the denoising, they use additional GNN that generates the atom numbers and the molecule’s CoM, and after that, denoise both 3D coordinates and element type of a molecule with an SE(3)-EGNN. DiffBP [99] can generate binding molecules with proper size, higher affinity with target protein, and roto-translation invariances. Unlike these methods, DiffDock [127] defines diffusion process over ligand poses (assignment of atomic positions in R3) corresponding to the degrees of freedom involved in molecular docking: ligand translations, ligand rotations, torsion angles. Additionally, they have high selective accuracy by leveraging the confidence model that

predicts confidence estimates on ligand poses generated through reverse SDE. The overview of DiffDock is shown in Figure 4. Moreover, for the protein-ligand complex generation, NeuralPLexer [133], an SDE-based generative model conditioned on the protein backbone templates and ligand molecular graphs is proposed. Particularly, the encoded inputs are forwarded to a contact predictor, which iteratively samples the distributions for binding interface proximity for each ligand. An SDE parameterized by the previously produced contact maps adds noise to the data. Protein and ligand structures are then jointly generated with an equivariant structure diffusion module (ESDM) by performing reverse diffusion of atom coordinates.

## 5.3 Others

**Materials Design.** Material is represented as the periodic arrangement of atoms in the 3D space [134]. Specifically, the Unit cell represents the infinite periodic structure of a material, which in turn is characterized by atom types, coordinates, and the periodic lattice. To generate stable materials, there are two key points: requirements 1) atom coordinates of material must exist in the local energy minimum defined by quantum mechanics, and requirements 2) The structure follows specific bonding preferences between different atom types. Moreover, the structure of the material should be permutation, translation, rotation, and periodic invariant. CDVAE [135] first encodes material onto a lower dimensional latent space, and then MLP property predictor predicts the number of atoms, periodic lattice, and the composition from latent representation  $z$ . Periodic GNN decoder, noise conditional score network (NCSN) [136], denoise atom type, and atom coordinate from  $z$ . The decoder outputs gradients leading atom coordinates to the energy local minimum and updates atom types to capture the specific local bonding preferences based on the neighbors. CDVAE [135] learns stability in making stable materials unstable by adding noise and then gradually denoising them into a stable material. The overview of CDVAE is shown in Figure 5.

**Mischaneous.** Early work in graph diffusion diffuses the adjacency matrix for edge diffusion, EDP-GNN [81], focusing on undirected graph generation adds noise only to the upper triangular part of adjacency matrix. Then, for making generated graphs discrete at sampling time, they threshold the sampled continuous adjacency matrix. Although methods diffusing only one structural element of graphs show promising results, they do not completely capture the relationship between nodes and edges, which is important for generating real-world graphs such as molecules. To overcome this issue, GDSS [84] diffuses both via two SDEs. It models the complex dependencies between nodes and edges in parallel. Since nodes and edges behave differently in terms of permutation, different modeling is desired, thus GDSS uses graph attention network for the denoising process. However, models learning node and edge representations independently ignore the interdependency between them [82]. To address these issues, NVDiff [82] diffuses only the latent representation of node vectors encoded via GraphVAE [61], capturing the structural information of graph, and uses an attention-based score generative model (SGM) as a prior to capture the local and global context within the graph.



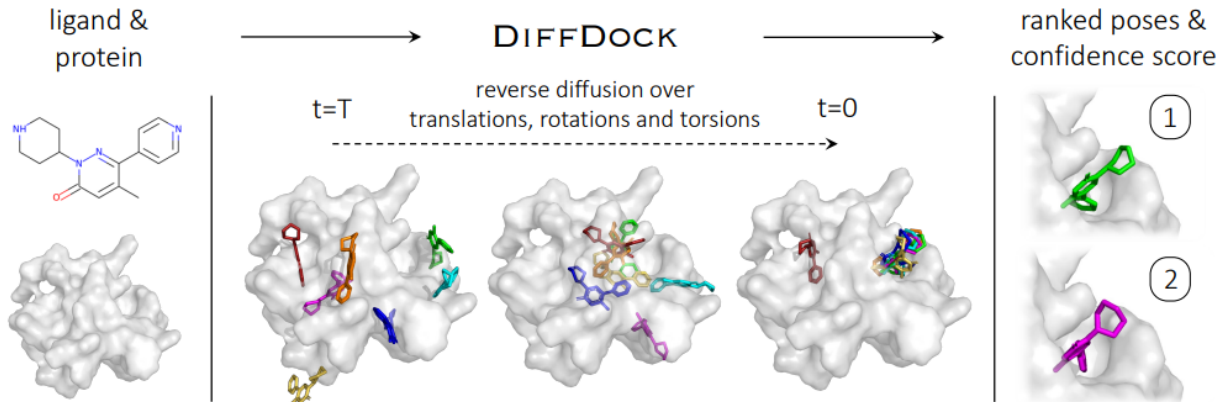


Fig. 4. Overview of DiffDock (figure obtained from [127])

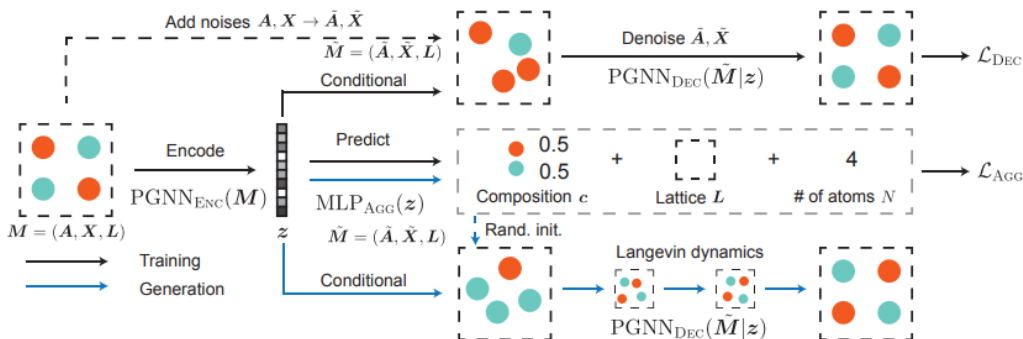


Fig. 5. Overview of CDVAE approach (figure obtained from [135])

TABLE 1  
Results on the Community-small and Ego-small datasets

Method	Datasets							
	Community-small				Ego-small			
	Deg.	Clus.	Orbit	Avg.	Deg.	Clus.	Orbit	Avg.
EDP-GNN [81]	0.053	0.144	0.026	0.074	0.052	0.093	0.007	0.050
GDSS [84]	0.021	0.024	0.007	0.017	0.045	0.086	0.007	0.046
GSDM [85]	0.011	0.015	0.001	0.009	-	-	-	-
NVDiff [82]	0.021	0.035	0.018	0.025	0.005	0.045	0.001	0.017
GraphGDP [83]	0.039	0.074	-	-	0.023	0.029	-	-

## 6 EVALUATION

Validating the effectiveness of generative models entails that both generated data and the learned distributions be evaluated [72], [137], [138]. For graph data, this is twofold difficult. First, unlike prediction problems which only need an evaluation of deterministic predictions, generative tasks necessitate evaluating the learned distributions [138], [139]. Second, evaluating graph-structured data is more complicated than assessing image or text data [4], [140]. We summarize the most generally operated graph generation metrics, including domain-independent and application-specific graph generation [141].

**Datasets.** Though there are many graph generation datasets, more specific to the target tasks, yet most representative

ones include 2D datasets, that is, Community-small, Ego-small, and Grid and 3D datasets, such as QM9, ZINC250k, and Enzymes as shown in Table 3. In detail, Community-small consists of 100 small community graphs with 12 to 20 nodes and Ego-small is a set of 200 citation ego graphs consisting of 4 to 18 nodes. The number of nodes in Grid, a dataset of 100 grid graphs, goes up to 400. Molecule generation datasets, QM9 and ZINC250k, contain more than a hundred thousand molecular graphs with the later comprising graphs as large as 38 nodes. Enzymes dataset, used in protein generation tasks, contains 600 protein graphs with the number of nodes ranging from 10 to 125.

### Metrics.

Most represented types of evaluation metrics in the

TABLE 2  
Results on the QM9 and ZINC250K datasets

Method	Datasets					
	QM9			ZINC250K		
	Val↑	NSPDK↓	FCD↓	Val↑	NSPDK↓	FCD↓
EDP-GNN [81]	47.52	0.005	2.680	82.97	0.049	16.737
GDSS [84]	95.72	0.003	2.900	97.01	0.019	14.656
GSDM [85]	99.9	0.003	2.650	92.70	0.017	12.956
NVDiff [82]	95.79	-	1.131	85.63	-	4.019

TABLE 3  
Datasets commonly used for graph generation models

Dataset	Dimensionality	Category	No. of Graphs (G)	No. of Nodes (N)
Community-small	2D	Social	100	11<N<20
Ego-small	2D	Social	200	3<N<18
Grid	2D	Grid	100	N<=400
QM9	3D	Bioinformatics/Molecular	130,831	3<N<29
ZINC250K	3D	Bioinformatics/Molecular	249,456	6<N<38
Enzymes	3D	Bioinformatics/Protein	600	9<N<125

graph generation literature are statistical, classifier-based, and related to quality. Evidently, **statistical** evaluations compare the statistics’ distributions of the generated graphs to that of real graphs via distance metrics such as Average Kullback-Leibler Divergence, Maximum Mean Discrepancy, etc. There are seven typical graph statistics [142], [143], including node degree distribution (Deg.) informing about the graph’s local connectivity patterns, clustering coefficient distribution (Clus.) measured as the ratio of a node’s potential number of triangles to the actual orbit count distribution (Orbit), the distribution of the 4-node orbit counts, useful in understanding the model’s capability to match higher-order graph statistics, and more. Whereas, in **Classifier-based** evaluation, a Graph Isomorphism Network (GIN) classifier trained on real graphs is tested on generated graphs via measuring either the classification accuracy of generated graphs or the Fréchet Inception Distance (FID) between the embeddings of generated and real graphs. However, as much as evaluating the similarity of real and generated graphs is critical, many applications like molecule generation demand some measure of **quality**. One is validity, which assesses whether certain requirements criteria, such as structural properties like being a tree or cycle, or chemical properties for molecule generation, are met. In Table 1 and Table 2, we summarize the results of some representative diffusion models tested in the graph domain. **Uniqueness** is another such metric; the fraction of graphs that are not sub-graph isomorphic to other generated graphs are computed to capture the diversity, key to some tasks, for instance, drug design. Novelty, on the other hand, measures the same but with regards to the real graphs to indicate generalizability.

## 7 CHALLENGES

Graph generation via diffusion models is a relatively emerging research area. Though several works have employed diffusion models for graph generation, the field still faces many

challenges and open research questions. In this section, we outline some of the foremost challenges.

**Evaluation Criterion.** Despite various widely adopted evaluation metrics, measuring and comparing the performance of graph generative models remains a major challenge given the lack of ground truth and universal metrics [144]. Current evaluations employing prior knowledge about graph data, for example, statistics and properties can not be relied on in real-world applications, in particular molecular generation, often requiring more expensive evaluations. The importance of these statistics and properties also varies with the target task. Thus, further research is needed to develop appropriate evaluation metrics.

**Diversity of Graphs.** Graphs are prevalent in many applications and a variety of data can be modeled via them. Each type of graph has unique characteristics and properties, which require different modeling approaches to generate them effectively. For instance, social networks have specific features, such as community structures and degree distributions, which must be considered when generating them [43], [145]. Similarly, biological networks require specialized models that incorporate the biochemical and physiological properties of the system being studied [20]. Despite this diversity of graph data, the current research on graph generation primarily focuses on molecular graphs used in drug discovery and materials science [146]. One of the reasons for this is the availability of many repositories of molecular graphs, such as the PubChem and ChEMBL databases. However, different graphs typically have significantly distinct natures and may need generation models to fulfill specific criteria accordingly. Despite this variety of graph data, research is mainly aimed at molecular graphs, partly due to the lack of ample repositories, overlooking the diversity of real graph data.

**Scalability.** Largely practiced diffusion of edges is not feasible for huge real-world graphs, as it necessitates a reverse diffusion of quadratically increasing steps for sampling new graphs. The cost of generating a new graph

by diffusion increases dramatically when the number of nodes grows [52]. This computational bottleneck makes it challenging to generate large-scale graphs using diffusion-based methods, limiting their application to large graphs common in the real world. Though efforts have been made to alleviate the problem by diffusing latent representation of nodes instead, the graph generation via diffusion is far from scalable and needs further exploration.

**Irregularity.** Graphs can have a highly irregular structure with variable numbers of nodes and edges, which can pose difficulties in specifying a diffusion process that can effectively seize the dynamics of the graph [147], [148]. In particular, designing a diffusion process that captures the complex dependencies between nodes and edges in a graph is challenging [149]. Additionally, the irregular structure of graphs can make it difficult to define a fixed diffusion kernel or transition operator suitable for modeling the dynamics of the graph [141]. As a result, adapting diffusion models for graph data requires developing new techniques for incorporating the irregular structure of graphs into the diffusion process, which is an active area of research.

**Interpretability.** Diffusion models depend on a score function to calculate the probability density of the data. However, the score function can be challenging to interpret and visualize in the context of graphs, which can restrict the model's interpretability and make it difficult to achieve insights into the underlying dynamics of the graphs [20], [25]. As a result, analyzing and comprehending the behavior of diffusion models in the context of graphs can be challenging.

## 8 CONCLUSION

This work conducts a review of the development of graph generation in generative AI. Specifically, we focus on its recent progress with graph diffusion models. Our survey work can help readers have a quick yet comprehensive understanding of graph diffusion models.

## REFERENCES

- [1] J. Sun, M. Ovsjanikov, and L. Guibas, "A concise and provably informative multi-scale signature based on heat diffusion," in *Computer graphics forum*, vol. 28, no. 5. Wiley Online Library, 2009, pp. 1383–1392. **1**
- [2] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013. **1**
- [3] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016. **1**
- [4] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," *Advances in neural information processing systems*, vol. 30, 2017. **1, 2, 9**
- [5] C. Chen, W. Ye, Y. Zuo, C. Zheng, and S. P. Ong, "Graph networks as a universal machine learning framework for molecules and crystals," *Chemistry of Materials*, vol. 31, no. 9, pp. 3564–3572, 2019. **1**
- [6] T. Hamaguchi, H. Oiwa, M. Shimbo, and Y. Matsumoto, "Knowledge transfer for out-of-knowledge-base entities: A graph neural network approach," *arXiv preprint arXiv:1706.05674*, 2017. **1**
- [7] D. Bacciu, F. Errica, A. Micheli, and M. Podda, "A gentle introduction to deep learning for graphs," *Neural Networks*, vol. 129, pp. 203–221, 2020. **1**
- [8] D. K. Duvenaud, D. Maclaurin, J. Iparraguirre, R. Bombarell, T. Hirzel, A. Aspuru-Guzik, and R. P. Adams, "Convolutional networks on graphs for learning molecular fingerprints," vol. 28, 2015. **1, 5**
- [9] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs," *arXiv preprint arXiv:1312.6203*, 2013. **1**
- [10] A. Kyrola, G. Blelloch, and C. Guestrin, "Graphchi: Large-scale graph computation on just a {PC}," in *Presented as part of the 10th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 12)*, 2012, pp. 31–46. **1**
- [11] G. Malewicz, M. H. Austern, A. J. Bik, J. C. Dehnert, I. Horn, N. Leiser, and G. Czajkowski, "Pregel: a system for large-scale graph processing," in *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, 2010, pp. 135–146. **1**
- [12] J. E. Gonzalez, Y. Low, H. Gu, D. Bickson, and C. Guestrin, "Powergraph: Distributed graph-parallel computation on natural graphs," in *Presented as part of the 10th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 12)*, 2012, pp. 17–30. **1**
- [13] W. L. Hamilton, R. Ying, and J. Leskovec, "Representation learning on graphs: Methods and applications," *arXiv preprint arXiv:1709.05584*, 2017. **1**
- [14] Z. Zhang, P. Cui, and W. Zhu, "Deep learning on graphs: A survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 1, pp. 249–270, 2020. **1**
- [15] X. Wang, H. Ji, C. Shi, B. Wang, Y. Ye, P. Cui, and P. S. Yu, "Heterogeneous graph attention network," in *The world wide web conference*, 2019, pp. 2022–2032. **1**
- [16] W.-L. Chiang, X. Liu, S. Si, Y. Li, S. Bengio, and C.-J. Hsieh, "Cluster-gcn: An efficient algorithm for training deep and large graph convolutional networks," in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2019, pp. 257–266. **1, 2**
- [17] J. Zhu, Y. Yan, L. Zhao, M. Heimann, L. Akoglu, and D. Koutra, "Beyond homophily in graph neural networks: Current limitations and effective designs," *Advances in Neural Information Processing Systems*, vol. 33, pp. 7793–7804, 2020. **1**
- [18] Z. Wu, B. Ramsundar, E. N. Feinberg, J. Gomes, C. Geniesse, A. S. Pappu, K. Leswing, and V. Pande, "Moleculenet: a benchmark for molecular machine learning," *Chemical science*, vol. 9, no. 2, pp. 513–530, 2018. **1**
- [19] A. Fout, J. Byrd, B. Shariat, and A. Ben-Hur, "Protein interface prediction using graph convolutional networks," *Advances in neural information processing systems*, vol. 30, 2017. **1**
- [20] J. Ingraham, V. Garg, R. Barzilay, and T. Jaakkola, "Generative models for graph-based protein design," *Advances in neural information processing systems*, vol. 32, 2019. **1, 5, 7, 10, 11**
- [21] M. Popova, O. Isayev, and A. Tropsha, "Deep reinforcement learning for de novo drug design," *Science advances*, vol. 4, no. 7, p. eaap7885, 2018. **1**
- [22] E. J. Bjerrum and R. Threlfall, "Molecular generation with recurrent neural networks (rnns)," *arXiv preprint arXiv:1705.04612*, 2017. **1**
- [23] Y. Jing, Y. Bian, Z. Hu, L. Wang, and X.-Q. S. Xie, "Deep learning for drug design: an artificial intelligence paradigm for drug discovery in the big data era," *The AAPS journal*, vol. 20, pp. 1–10, 2018. **1**
- [24] Y. Li, D. Tarlow, M. Brockschmidt, and R. Zemel, "Gated graph sequence neural networks," *arXiv preprint arXiv:1511.05493*, 2015. **1**
- [25] J. You, R. Ying, X. Ren, W. Hamilton, and J. Leskovec, "Graphrnn: Generating realistic graphs with deep auto-regressive models," in *International conference on machine learning*. PMLR, 2018, pp. 5708–5717. **1, 2, 5, 11**
- [26] M. Popova, M. Shvets, J. Oliva, and O. Isayev, "Molecularrnn: Generating realistic molecular graphs with optimized properties," *arXiv preprint arXiv:1905.13372*, 2019. **1, 2**
- [27] M. Simonovsky and N. Komodakis, "Graphvae: Towards generation of small graphs using variational autoencoders," in *International conference on artificial neural networks*. Springer, 2018, pp. 412–422. **1, 2, 5**
- [28] N. De Cao and T. Kipf, "Molgan: An implicit generative model for small molecular graphs," *arXiv preprint arXiv:1805.11973*, 2018. **1, 3, 5**
- [29] R. Yang, P. Srivastava, and S. Mandt, "Diffusion probabilistic modeling for video generation," *arXiv preprint arXiv:2203.09481*, 2022. **2, 4**



- [30] A. Ulhaq, N. Akhtar, and G. Pogrebna, "Efficient diffusion models for vision: A survey," *arXiv preprint arXiv:2210.09292*, 2022. **2**
- [31] H. Cao, C. Tan, Z. Gao, G. Chen, P.-A. Heng, and S. Z. Li, "A survey on generative diffusion model," *arXiv preprint arXiv:2209.02646*, 2022. **2**
- [32] C. Zhang, C. Zhang, M. Zhang, and I. S. Kweon, "Text-to-image diffusion models in generative ai: A survey," *arXiv preprint arXiv:2303.07909*, 2023. **2**
- [33] C. Zhang, C. Zhang, S. Zheng, M. Zhang, M. Qamar, S.-H. Bae, and I. S. Kweon, "A survey on audio diffusion models: Text to speech synthesis and enhancement in generative ai," *arXiv preprint arXiv:2303.13336*, 2023. **2**
- [34] C. Zhang, C. Zhang, S. Zheng, Y. Qiao, C. Li, M. Zhang, S. K. Dam, C. M. Thwal, Y. L. Tun, L. Huy, D. Kim, S.-H. Bae, L.-H. Lee, Y. Yang, H. T. Shen, I. S. Kweon, and C. S. Hong, "A complete survey on generative ai (aigc): Is chatgpt from gpt-4 to gpt-5 all you need?" *arXiv preprint arXiv:2303.11717*, 2023. **2**
- [35] C. Zhang, C. Zhang, C. Li, S. Zheng, Y. Qiao, S. K. Dam, M. Zhang, J. U. Kim, S. T. Kim, G.-M. Park, J. Choi, S.-H. Bae, L.-H. Lee, P. Hui, I. S. Kweon, and C. S. Hong, "One small step for generative ai, one giant leap for agi: A complete survey on chatgpt in aigc era," *researchgate DOI:10.13140/RG.2.2.24789.70883*, 2023. **2**
- [36] C. Zhang, C. Zhang, J. Song, J. S. K. Yi, K. Zhang, and I. S. Kweon, "A survey on masked autoencoder for self-supervised learning in vision and beyond," *arXiv preprint arXiv:2208.00173*, 2022. **2**
- [37] P. Erdős, A. Rényi *et al.*, "On the evolution of random graphs," *Publ. Math. Inst. Hung. Acad. Sci.*, vol. 5, no. 1, pp. 17–60, 1960. **2**
- [38] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks," *nature*, vol. 393, no. 6684, pp. 440–442, 1998. **2**
- [39] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *science*, vol. 286, no. 5439, pp. 509–512, 1999. **2**
- [40] D. Chakrabarti, Y. Zhan, and C. Faloutsos, "R-mat: A recursive model for graph mining," in *Proceedings of the 2004 SIAM International Conference on Data Mining*. SIAM, 2004, pp. 442–446. **2**
- [41] J. Leskovec, D. Chakrabarti, J. Kleinberg, C. Faloutsos, and Z. Ghahramani, "Kronecker graphs: an approach to modeling networks," *Journal of Machine Learning Research*, vol. 11, no. 2, 2010. **2**
- [42] A. Bojchevski, O. Shchur, D. Zügner, and S. Günnemann, "Netgan: Generating graphs via random walks," in *International conference on machine learning*. PMLR, 2018, pp. 610–619. **2, 3, 5**
- [43] Z. Yang, W. Cohen, and R. Salakhudinov, "Revisiting semi-supervised learning with graph embeddings," in *International conference on machine learning*. PMLR, 2016, pp. 40–48. **2, 10**
- [44] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *NeurIPS*, 2017. **2**
- [45] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013. **2**
- [46] L. Dinh, J. Sohl-Dickstein, and S. Bengio, "Density estimation using real nvp," *arXiv preprint arXiv:1605.08803*, 2016. **2**
- [47] 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*, vol. 63, no. 11, pp. 139–144, 2020. **2**
- [48] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," *Advances in neural information processing systems*, vol. 27, 2014. **2**
- [49] A. Radford, R. Jozefowicz, and I. Sutskever, "Learning to generate reviews and discovering sentiment," *arXiv preprint arXiv:1704.01444*, 2017. **2**
- [50] S. Reed, A. Oord, N. Kalchbrenner, S. G. Colmenarejo, Z. Wang, Y. Chen, D. Belov, and N. Freitas, "Parallel multiscale autoregressive density estimation," in *International Conference on Machine Learning*. PMLR, 2017, pp. 2912–2921. **2**
- [51] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," 2017, pp. 1263–1272. **2**
- [52] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec, "Graph convolutional neural networks for web-scale recommender systems," in *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 2018, pp. 974–983. **2, 11**
- [53] C. Shi, M. Xu, Z. Zhu, W. Zhang, M. Zhang, and J. Tang, "Graphaf: a flow-based autoregressive model for molecular graph generation," *arXiv preprint arXiv:2001.09382*, 2020. **2**
- [54] T. D. Bui, S. Ravi, and V. Ramavajjala, "Neural graph learning: Training neural networks using graphs," in *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, 2018, pp. 64–71. **2**
- [55] J. You, B. Liu, Z. Ying, V. Pande, and J. Leskovec, "Graph convolutional policy network for goal-directed molecular graph generation," *Advances in neural information processing systems*, vol. 31, 2018. **2**
- [56] C. Doersch, "Tutorial on variational autoencoders," *arXiv preprint arXiv:1606.05908*, 2016. **2**
- [57] T. Ma, J. Chen, and C. Xiao, "Constrained generation of semantically valid graphs via regularizing variational autoencoders," *Advances in Neural Information Processing Systems*, vol. 31, 2018. **2**
- [58] A. Grover, A. Zweig, and S. Ermon, "Graphite: Iterative generative modeling of graphs," in *International conference on machine learning*. PMLR, 2019, pp. 2434–2444. **2**
- [59] Q. Liu, M. Allamanis, M. Brockschmidt, and A. Gaunt, "Constrained graph variational autoencoders for molecule design," *Advances in neural information processing systems*, vol. 31, 2018. **2, 5**
- [60] W. Jin, R. Barzilay, and T. Jaakkola, "Junction tree variational autoencoder for molecular graph generation," 2018, pp. 2323–2332. **2, 6**
- [61] T. N. Kipf and M. Welling, "Variational graph auto-encoders," *arXiv preprint arXiv:1611.07308*, 2016. **2, 5, 8**
- [62] D. Rezende and S. Mohamed, "Variational inference with normalizing flows," in *International conference on machine learning*. PMLR, 2015, pp. 1530–1538. **2**
- [63] G. Papamakarios, T. Pavlakou, and I. Murray, "Masked autoregressive flow for density estimation," *Advances in neural information processing systems*, vol. 30, 2017. **2**
- [64] J. Ho, X. Chen, A. Srinivas, Y. Duan, and P. Abbeel, "Flow++: Improving flow-based generative models with variational dequantization and architecture design," in *International Conference on Machine Learning*. PMLR, 2019, pp. 2722–2730. **2**
- [65] K. Madhawa, K. Ishiguro, K. Nakago, and M. Abe, "Graphnvp: An invertible flow model for generating molecular graphs," *arXiv preprint arXiv:1905.11600*, 2019. **2**
- [66] J. Liu, A. Kumar, J. Ba, J. Kiros, and K. Swersky, "Graph normalizing flows," *Advances in Neural Information Processing Systems*, vol. 32, 2019. **2, 3**
- [67] C. Zang and F. Wang, "Moflow: an invertible flow model for generating molecular graphs," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 617–626. **3**
- [68] Y. Luo, K. Yan, and S. Ji, "Graphdf: A discrete flow model for molecular graph generation," in *International Conference on Machine Learning*. PMLR, 2021, pp. 7192–7203. **3**
- [69] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, "Generative adversarial networks: An overview," *IEEE signal processing magazine*, vol. 35, no. 1, pp. 53–65, 2018. **3**
- [70] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014. **3**
- [71] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015. **3**
- [72] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," *Advances in neural information processing systems*, 2016. **3, 9**
- [73] W. Jin, K. Yang, R. Barzilay, and T. Jaakkola, "Learning multi-modal graph-to-graph translation for molecular optimization," *arXiv preprint arXiv:1812.01070*, 2018. **3**
- [74] Ł. Maziarka, A. Pocha, J. Kaczmarski, K. Rataj, T. Danel, and M. Warchol, "Mol-cyclelegan: a generative model for molecular optimization," *Journal of Cheminformatics*, vol. 12, no. 1, pp. 1–18, 2020. **3**
- [75] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in Neural Information Processing Systems*, vol. 33, pp. 6840–6851, 2020. **3, 4**
- [76] C. Vignac, I. Krawczuk, A. Siraudin, B. Wang, V. Cevher, and P. Frossard, "Digress: Discrete denoising diffusion for graph generation," *arXiv preprint arXiv:2209.14734*, 2022. **3, 4, 5**

- [77] K. K. Haefeli, K. Martinkus, N. Perraudin, and R. Wattenhofer, "Diffusion models for graphs benefit from discrete state spaces," *arXiv preprint arXiv:2210.01549*, 2022. [3](#), [4](#), [5](#)
- [78] N. Anand and T. Achim, "Protein structure and sequence generation with equivariant denoising diffusion probabilistic models," *arXiv preprint arXiv:2205.15019*, 2022. [3](#), [5](#), [7](#)
- [79] B. L. Trippe, J. Yim, D. Tischer, T. Broderick, D. Baker, R. Barzilay, and T. Jaakkola, "Diffusion probabilistic modeling of protein backbones in 3d for the motif-scaffolding problem," *arXiv preprint arXiv:2206.04119*, 2022. [3](#), [5](#), [8](#)
- [80] S. Luo, Y. Su, X. Peng, S. Wang, J. Peng, and J. Ma, "Antigen-specific antibody design and optimization with diffusion-based generative models," *bioRxiv*, 2022. [3](#), [8](#)
- [81] 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. [3](#), [4](#), [5](#), [8](#), [9](#), [10](#)
- [82] X. Chen, Y. Li, A. Zhang, and L.-p. Liu, "Nvdiffr: Graph generation through the diffusion of node vectors," *arXiv preprint arXiv:2211.10794*, 2022. [3](#), [4](#), [8](#), [9](#), [10](#)
- [83] H. Huang, L. Sun, B. Du, Y. Fu, and W. Lv, "Graphgdp: Generative diffusion processes for permutation invariant graph generation," *arXiv preprint arXiv:2212.01842*, 2022. [3](#), [4](#), [5](#), [9](#)
- [84] J. Jo, S. Lee, and S. J. Hwang, "Score-based generative modeling of graphs via the system of stochastic differential equations," *arXiv preprint arXiv:2202.02514*, 2022. [3](#), [4](#), [8](#), [9](#), [10](#)
- [85] T. Luo, Z. Mo, and S. J. Pan, "Fast graph generative model via spectral diffusion," *arXiv preprint arXiv:2211.08892*, 2022. [3](#), [4](#), [5](#), [9](#), [10](#)
- [86] L. Yang, Z. Zhang, Y. Song, S. Hong, R. Xu, Y. Zhao, Y. Shao, W. Zhang, B. Cui, and M.-H. Yang, "Diffusion models: A comprehensive survey of methods and applications," *arXiv preprint arXiv:2209.00796*, 2022. [4](#)
- [87] Y. Song and S. Ermon, "Improved techniques for training score-based generative models," *Advances in neural information processing systems*, vol. 33, pp. 12 438–12 448, 2020. [4](#), [5](#), [6](#)
- [88] 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*. PMLR, 2020, pp. 574–584. [4](#)
- [89] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?" *arXiv preprint arXiv:1810.00826*, 2018. [4](#), [5](#)
- [90] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, "Score-based generative modeling through stochastic differential equations," in *International Conference on Learning Representations*, 2020. [4](#), [5](#)
- [91] Y. Li, O. Vinyals, C. Dyer, R. Pascanu, and P. Battaglia, "Learning deep generative models of graphs," *arXiv preprint arXiv:1803.03324*, 2018. [5](#)
- [92] R. Liao, Y. Li, Y. Song, S. Wang, W. Hamilton, D. K. Duvenaud, R. Urtasun, and R. Zemel, "Efficient graph generation with graph recurrent attention networks," *Advances in neural information processing systems*, vol. 32, 2019. [5](#)
- [93] H. Dai, A. Nazi, Y. Li, B. Dai, and D. Schuurmans, "Scalable deep generative modeling for sparse graphs," in *International Conference on Machine Learning*. PMLR, 2020, pp. 2302–2312. [5](#)
- [94] X. Chen, X. Han, J. Hu, F. J. Ruiz, and L. Liu, "Order matters: Probabilistic modeling of node sequence for graph generation," *arXiv preprint arXiv:2106.06189*, 2021. [5](#)
- [95] P. Dhariwal and A. Nichol, "Diffusion models beat gans on image synthesis," vol. 34, 2021, pp. 8780–8794. [5](#)
- [96] U. Alon and E. Yahav, "On the bottleneck of graph neural networks and its practical implications," *arXiv preprint arXiv:2006.05205*, 2020. [5](#)
- [97] P. B. Jørgensen, M. N. Schmidt, and O. Winther, "Deep generative models for molecular science," *Molecular informatics*, vol. 37, no. 1-2, p. 1700133, 2018. [5](#)
- [98] M. Xu, L. Yu, Y. Song, C. Shi, S. Ermon, and J. Tang, "Geodiff: A geometric diffusion model for molecular conformation generation," *arXiv preprint arXiv:2203.02923*, 2022. [5](#), [6](#)
- [99] 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. [5](#), [8](#)
- [100] I. Igashov, H. Stärk, C. Vignac, V. G. Satorras, P. Frossard, M. Welling, M. Bronstein, and B. Correia, "Equivariant 3d-conditional diffusion models for molecular linker design," *arXiv preprint arXiv:2210.05274*, 2022. [5](#), [6](#)
- [101] E. Hoogetboom, V. G. Satorras, C. Vignac, and M. Welling, "Equivariant diffusion for molecule generation in 3d," in *ICML*. PMLR, 2022, pp. 8867–8887. [5](#), [6](#), [7](#)
- [102] V. G. Satorras, E. Hoogetboom, and M. Welling, "E (n) equivariant graph neural networks," in *International conference on machine learning*. PMLR, 2021, pp. 9323–9332. [5](#), [8](#)
- [103] C. Shi, S. Luo, M. Xu, and J. Tang, "Learning gradient fields for molecular conformation generation," 2021, pp. 9558–9568. [5](#), [6](#)
- [104] W. P. Walters and R. Barzilay, "Applications of deep learning in molecule generation and molecular property prediction," *Accounts of chemical research*, vol. 54, no. 2, pp. 263–270, 2020. [5](#)
- [105] K. Atz, F. Grisoni, and G. Schneider, "Geometric deep learning on molecular representations," *Nature Machine Intelligence*, vol. 3, no. 12, pp. 1023–1032, 2021. [5](#)
- [106] A. Zhavoronkov, "Artificial intelligence for drug discovery, biomarker development, and generation of novel chemistry," pp. 4311–4313, 2018. [5](#)
- [107] S. Brooks, "Markov chain monte carlo method and its application," *Journal of the royal statistical society: series D (the Statistician)*, vol. 47, no. 1, pp. 69–100, 1998. [6](#)
- [108] M. Xu, S. Luo, Y. Bengio, J. Peng, and J. Tang, "Learning neural generative dynamics for molecular conformation generation," *arXiv preprint arXiv:2102.10240*, 2021. [6](#)
- [109] G. N. Simm and J. M. Hernández-Lobato, "A generative model for molecular distance geometry," *arXiv preprint arXiv:1909.11459*, 2019. [6](#)
- [110] S. Luo, C. Shi, M. Xu, and J. Tang, "Predicting molecular conformation via dynamic graph score matching," vol. 34, 2021, pp. 19 784–19 795. [6](#)
- [111] B. Jing, G. Corso, J. Chang, R. Barzilay, and T. Jaakkola, "Torsional diffusion for molecular conformer generation," *arXiv preprint arXiv:2206.01729*, 2022. [6](#), [7](#)
- [112] D. C. Elton, Z. Boukouvalas, M. D. Fuge, and P. W. Chung, "Deep learning for molecular design—a review of the state of the art," *Molecular Systems Design & Engineering*, vol. 4, no. 4, pp. 828–849, 2019. [6](#)
- [113] B. Sanchez-Lengeling and A. Aspuru-Guzik, "Inverse molecular design using machine learning: Generative models for matter engineering," *Science*, vol. 361, no. 6400, pp. 360–365, 2018. [6](#)
- [114] L. Huang, H. Zhang, T. Xu, and K.-C. Wong, "Mdm: Molecular diffusion model for 3d molecule generation," *arXiv preprint arXiv:2209.05710*, 2022. [6](#)
- [115] L. Wu, C. Gong, X. Liu, M. Ye, and Q. Liu, "Diffusion-based molecule generation with informative prior bridges," *arXiv preprint arXiv:2209.00865*, 2022. [6](#)
- [116] A. C. Anderson, "The process of structure-based drug design," *Chemistry & biology*, vol. 10, no. 9, pp. 787–797, 2003. [6](#)
- [117] A. Schneuing, Y. Du, C. Harris, A. Jamasb, I. Igashov, W. Du, T. Blundell, P. Lió, C. Gomes, M. Welling *et al.*, "Structure-based drug design with equivariant diffusion models," *arXiv preprint arXiv:2210.13695*, 2022. [6](#)
- [118] F. Bao, M. Zhao, Z. Hao, P. Li, C. Li, and J. Zhu, "Equivariant energy-guided sde for inverse molecular design," *arXiv preprint arXiv:2209.15408*, 2022. [6](#), [7](#)
- [119] A. Morehead and J. Cheng, "Geometry-complete diffusion for 3d molecule generation," *arXiv preprint arXiv:2302.04313*, 2023. [7](#)
- [120] M. AlQuraishi, "End-to-end differentiable learning of protein structure," *Cell systems*, vol. 8, no. 4, pp. 292–301, 2019. [7](#)
- [121] A. W. Senior, R. Evans, J. Jumper, J. Kirkpatrick, L. Sifre, T. Green, C. Qin, A. Židek, A. W. Nelson, A. Bridgland *et al.*, "Improved protein structure prediction using potentials from deep learning," *Nature*, vol. 577, no. 7792, pp. 706–710, 2020. [7](#)
- [122] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Židek, A. Potapenko *et al.*, "Highly accurate protein structure prediction with alphafold," *Nature*, vol. 596, no. 7873, pp. 583–589, 2021. [7](#)
- [123] J. Yang, I. Anishchenko, H. Park, Z. Peng, S. Ovchinnikov, and D. Baker, "Improved protein structure prediction using predicted interresidue orientations," *Proceedings of the National Academy of Sciences*, vol. 117, no. 3, pp. 1496–1503, 2020. [7](#)
- [124] A. Mohamed, D. Okhonko, and L. Zettlemoyer, "Transformers with convolutional context for asr," *arXiv preprint arXiv:1904.11660*, 2019. [7](#)
- [125] A. Rives, J. Meier, T. Sercu, S. Goyal, Z. Lin, J. Liu, D. Guo, M. Ott, C. L. Zitnick, J. Ma *et al.*, "Biological structure and function emerge from scaling unsupervised learning to 250 million protein

- sequences," *Proceedings of the National Academy of Sciences*, vol. 118, no. 15, p. e2016239118, 2021. **7**
- [126] K. E. Wu, K. K. Yang, R. v. d. Berg, J. Y. Zou, A. X. Lu, and A. P. Amini, "Protein structure generation via folding diffusion," *arXiv preprint arXiv:2209.15611*, 2022. **7**
- [127] G. Corso, H. Stärk, B. Jing, R. Barzilay, and T. Jaakkola, "Diffdock: Diffusion steps, twists, and turns for molecular docking," *arXiv preprint arXiv:2210.01776*, 2022. **8, 9**
- [128] J. Zhou, C. L. Theesfeld, K. Yao, K. M. Chen, A. K. Wong, and O. G. Troyanskaya, "Deep learning sequence-based ab initio prediction of variant effects on expression and disease risk," *Nature genetics*, vol. 50, no. 8, pp. 1171–1179, 2018. **8**
- [129] B. North, A. Lehmann, and R. L. Dunbrack Jr, "A new clustering of antibody cdr loop conformations," *Journal of molecular biology*, vol. 406, no. 2, pp. 228–256, 2011. **8**
- [130] M.-P. Lefranc, C. Pommié, M. Ruiz, V. Giudicelli, E. Foulquier, L. Truong, V. Thouvenin-Contet, and G. Lefranc, "Imgt unique numbering for immunoglobulin and t cell receptor variable domains and ig superfamily v-like domains," *Developmental & Comparative Immunology*, vol. 27, no. 1, pp. 55–77, 2003. **8**
- [131] J. Dunbar, K. Krawczyk, J. Leem, T. Baker, A. Fuchs, G. Georges, J. Shi, and C. M. Deane, "Sabdab: the structural antibody database," *Nucleic acids research*, vol. 42, no. D1, pp. D1140–D1146, 2014. **8**
- [132] D. L. Mobley and K. A. Dill, "Binding of small-molecule ligands to proteins: "what you see" is not always "what you get"," *Structure*, vol. 17, no. 4, pp. 489–498, 2009. **8**
- [133] Z. Qiao, W. Nie, A. Vahdat, T. F. Miller III, and A. Anandkumar, "Dynamic-backbone protein-ligand structure prediction with multiscale generative diffusion models," *arXiv preprint arXiv:2209.15171*, 2022. **8**
- [134] J. P. Glusker, M. Lewis, and M. Rossi, *Crystal structure analysis for chemists and biologists*. John Wiley & Sons, 1996, vol. 16. **8**
- [135] T. Xie, X. Fu, O.-E. Ganea, R. Barzilay, and T. Jaakkola, "Crystal diffusion variational autoencoder for periodic material generation," *arXiv preprint arXiv:2110.06197*, 2021. **8, 9**
- [136] Y. Song and S. Ermon, "Generative modeling by estimating gradients of the data distribution," vol. 32, 2019. **8**
- [137] D. J. Sutherland, H.-Y. Tung, H. Strathmann, S. De, A. Ramdas, A. Smola, and A. Gretton, "Generative models and model criticism via optimized maximum mean discrepancy," *arXiv preprint arXiv:1611.04488*, 2016. **9**
- [138] L. Theis, A. v. d. Oord, and M. Bethge, "A note on the evaluation of generative models," *arXiv preprint arXiv:1511.01844*, 2015. **9**
- [139] M. S. Sajjadi, O. Bachem, M. Lucic, O. Bousquet, and S. Gelly, "Assessing generative models via precision and recall," *Advances in neural information processing systems*, vol. 31, 2018. **9**
- [140] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, "Graph neural networks: A review of methods and applications," *AI Open*, vol. 1, pp. 57–81, 2020. **9**
- [141] S. Zhang, H. Tong, J. Xu, and R. Maciejewski, "Graph convolutional networks: a comprehensive review," *Computational Social Networks*, vol. 6, no. 1, pp. 1–23, 2019. **9, 11**
- [142] A. Clauset, M. E. Newman, and C. Moore, "Finding community structure in very large networks," *Physical review E*, vol. 70, no. 6, p. 066111, 2004. **10**
- [143] B. Karrer and M. E. Newman, "Stochastic blockmodels and community structure in networks," *Physical review E*, vol. 83, no. 1, p. 016107, 2011. **10**
- [144] K. Xu, L. Wu, Z. Wang, Y. Feng, M. Witbrock, and V. Sheinin, "Graph2seq: Graph to sequence learning with attention-based neural networks," *arXiv preprint arXiv:1804.00823*, 2018. **10**
- [145] D. Wang, P. Cui, and W. Zhu, "Structural deep network embedding," in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, 2016, pp. 1225–1234. **10**
- [146] R. Gómez-Bombarelli, J. N. Wei, D. Duvenaud, J. M. Hernández-Lobato, B. Sánchez-Lengeling, D. Sheberla, J. Aguilera-Iparraguirre, T. D. Hirzel, R. P. Adams, and A. Aspuru-Guzik, "Automatic chemical design using a data-driven continuous representation of molecules," *ACS central science*, vol. 4, no. 2, pp. 268–276, 2018. **10**
- [147] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *arXiv preprint arXiv:1710.10903*, 2017. **11**
- [148] P. Veličković, W. Fedus, W. L. Hamilton, P. Liò, Y. Bengio, and R. D. Hjelm, "Deep graph infomax," *ICLR (Poster)*, vol. 2, no. 3, p. 4, 2019. **11**
- [149] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2014, pp. 701–710. **11**