DiffSeqMol: A Non-Autoregressive Diffusion-Based Approach for Molecular Sequence Generation and Optimization

Zixu Wang[1,*], Yangyang Chen[1,*], Xiucai Ye[1,#]

[1]Department of Computer Science, University of Tsukuba, Tsukuba 3058577, Japan

* Equal Contribution.

Corresponding Author

Email addresses: yexiucai@cs.tsukuba.ac.jp

Biographical Note:

Zixu Wang is currently a PhD student at the University of Tsukuba, Japan. His research interests include machine learning and drug discovery.

Yangyang Chen received an M.S. degree in Computer Technology from Hunan University, China, in 2021. She is currently a PhD student at the University of Tsukuba, Japan. Her research interests include bioinformatics and deep learning.

Xiucai Ye received a Ph.D. degree in computer science from the University of Tsukuba, Tsukuba, Japan, in 2014. She is currently an Assistant Professor with the Department of Computer Science, and Center for Artificial Intelligence Research (C-AIR), University of Tsukuba. Her current research interests include feature selection, clustering, machine learning, and bioinformatics.

Abstract

The application of deep generative models for molecular discovery has witnessed a significant surge in recent years. Currently, the field of molecular generation and molecular optimization

is predominantly governed by autoregressive models regardless of how molecular data is represented. However, an emerging paradigm in the generation domain is diffusion models, which treat data non-autoregressively and has achieved significant breakthroughs in areas such as image generation. The potential and capability of diffusion models in molecular generation and optimization tasks remain largely unexplored. In order to investigate the potential applicability of diffusion models in the domain of molecular exploration, we proposed DiffSeqMol, a molecular sequence generation model, underpinned by diffusion process. DiffSeqMol distinguishes itself from traditional autoregressive methods by its capacity to draw samples from random noise and direct generating the entire molecule. Through experiment evaluations, we demonstrated that DiffSeqMol can achieve, even surpass, the performance of established state-of-the-art models on unconditional generation tasks and molecular optimization tasks.
Keywords: Diffusion model, Molecule Generation, Molecule Optimization

Introduction

Exploration of chemical space is a critical undertaking in contemporary scientific research, encompassing various applications in drug discovery [1] and materials science [2]. However, traditional methods for exploring chemical space have limitations in terms of their efficiency and ability to generate novel and valuable compounds. Artificial Intelligence (AI) has the potential for revolutionizing the field of chemical discovery by facilitating more efficient and effective exploration of the vast and mysterious chemical universe [3, 4]. In this regard, deep learning models have emerged as a highly promising solution for addressing complicated biomedical-related challenges [5-8].

Autoregressive language models such as Recurrent Neural Networks (RNNs) [9] and Transformers [10] have played a significant role in the advancement of the language generation field. Researchers have found that these models not only excel in natural language but also demonstrate notable performance in the field of molecular generation. A majority of molecular sequence generation methods employ Simplified Molecular Input Line Entry System (SMILES) [11] strings as their molecular representation, and generate molecular sequences token by token. These models have found extensive use across a myriad of generation tasks, inclusive of enrich existing pharmacological libraries [12], generating molecules guided by some physical or chemical properties [13], among others. Furthermore, they have been employed for ligand-based de novo design [14, 15], a pursuit highly related to real-world drug development.

Variational Auto Encoders (VAEs) [16] are also widely adopted by researchers as a molecular generation model. The encoding-decoding structure of VAEs makes it well-suited for generation methods based on graph representation. They firstly encode the molecular graph into the latent space, and then autoregressively decode molecular features [17], either atom by atom or fragment by fragment. These methods allow researchers to pre-set some chemical constraints such as valence check, thereby generating valid molecules. Due to the strong interpretability of graph representation, they have been widely adopted in the sphere

of molecular optimization. Nevertheless, they are not without its limitations. One of the main challenges faced by methods relied on graph representation is the difficulty in efficiently modeling the chemical space [18]. However, autoregressive language models have proven to be highly effective in capturing sequential dependencies and surpass graph-based methods in terms of their ability to fit distributions. Furthermore, generating valid and chemically meaningful molecular graphs requires enforcing certain pre-defined constraints.

In summary, current molecular generation methods are primarily based on autoregressive models regardless of how molecular data is represented. These models are trained on specific subsets of chemical space, enabling them to generate molecules that bear resemblance to their training sets [14, 15], or optimize existing compounds to enhance molecular properties.

In recent years, the advent of diffusion models [19, 20] has revolutionized generation tasks, making notable progress especially in the domains of image, video [21] and audio [22] generation tasks. These models [23] are designed to non-autoregressively generate high-quality samples through denoising random noises and modeling the underlying probability distribution of the source data. Their outstanding performance in the vision and audio tasks demonstrates their potential for high-quality data synthesis. However, the development of diffusion models for text generation [24-26] is still at a preliminary stage, falling behind the progress observed in other modalities. Exploration of biomedical data is even scarcer. Currently, only a small number of studies [27, 28] have applied diffusion models to sequential protein data and their results are far from satisfactory.

To bridge the gap and establish a connection between diffusion generative models and the realm of molecular generation, we propose a diffusion-based generation model, dubbed DiffSeqMol, for molecular sequence generation. DiffSeqMol is different from existing canonical molecular generation methodologies in two ways: (1) In contrast to conventional molecular generation models, DiffSeqMol starts with a sequence of gaussian noise signals and progressively denoising these signals into vectors that align with molecular tokens; (2) Different from conventional generation approaches which generation molecular substructure/tokens in an autoregressive left-to-right manner, DiffSeqMol generates all molecular tokens in parallel. Since the molecular elements are highly interdependent in a molecule, the placement of a single atom affects the location and type of every other atom in the molecule. So, the simultaneous generation of all parts of a molecule increases the complexity of the learning, which is a more complex task compared to learning conditional probabilities in an autoregressive way. In this study, we evaluated the capacity of the diffusion-based molecular sequence generation model by apply DiffSeqMol on two unconditional molecular generation benchmarks and four molecular optimization benchmarks. The experiment findings show promising results, demonstrating that DiffSeqMol can rival conventional molecular generation models in terms of performance on molecule-related generation tasks.

Methods

Problem Statement and Background

In sequence-based molecular generation, the necessary preliminary step entails preprocessing of the molecular data. Initially, the molecular data is represented in its SMILES format. Subsequently, these molecular strings are divided and tokenized by character-based segmentation methods.

The objective of the molecular generation task is to design models that can sampling $\boldsymbol{w} = [\boldsymbol{w}_1, \boldsymbol{w}_2, \cdots, \boldsymbol{w}_n]$ from a trained molecular generation model $p_{mol}(\boldsymbol{w})$, where $\boldsymbol{w}$ is the sequence of discrete molecular tokens and w_i represents the i^{th} token. Correspondingly, the molecular optimization task aims to design models, represented as $p_{opmol}(\boldsymbol{w^y}|\boldsymbol{w^x})$, that can improve some predesigned optimization targets while preserve the original molecular structure as much as possible, where $\boldsymbol{w^x}$ denotes target molecule and $\boldsymbol{w^y}$ represents the optimized molecule. Optimization targets encompass a broad range of properties including, but not limited to, physicochemical properties of the molecule and affinity towards the protein receptor.

The conventional approaches employed for molecular generation or optimization tasks typically operate in an autoregressive left-to-right manner. They gradually complete molecules based on the partial molecular sequence generated so far. A typical unconditional autoregressive generation model can be expressed as follows:

$$p_{mol}(\boldsymbol{w}) = p_{mol}(\boldsymbol{w}_1)\prod_{i=2}^{n} p_{mol}(\boldsymbol{w}_i|\boldsymbol{w}_{<i})\,. \tag{1}$$

As for optimization task, the target molecule is also considered as the input of the generation model. A typical autoregressive molecular optimization model can be expressed as:

$$p_{opmol}(\boldsymbol{w^y}|\boldsymbol{w^x}) = p_{opmol}(\boldsymbol{w}_1^{\boldsymbol{y}})\prod_{i=2}^{n} p_{opmol}(\boldsymbol{w}_i^{\boldsymbol{y}}|\boldsymbol{w}_{<i}^{\boldsymbol{y}}, \boldsymbol{w^x})\,. \tag{2}$$

DiffSeqMol: a Diffusion-Based Molecular Sequence Generation Model

Diffusion models [20] are a class of generative model which are specifically designed to remove noise and generate high-quality samples by modeling the underlying probability distribution of the target data. These models leverage the principles of diffusion to gradually transform an initial noisy distribution into the target distribution of data. Ho et al. [19] have further refined the approach, successfully achieving the comparable image quality with state-of-the-art image generation models, such as Generative Adversarial Networks (GANs)

[29] and VAEs [16]. The subsequent works has made substantial advancements in enhancing the capability of diffusion models and diffusion-based models have achieved great performance across many areas. Inspired by diffusion model designed for textual data [24, 26], we extended the application of these models to sequential molecular data generation and molecular optimization tasks, dubbed DiffSeqMol.

Specifically, the diffusion process can be regarded as a discrete-time Markov process. It contains two processes, the forward diffusion process and the backward diffusion step. The forward diffusion process starts with an initial state $\boldsymbol{s_0}$, where $\boldsymbol{s_0}$ is the initial data distribution of the original data. Then, the diffusion model gradually adds Gaussian noises to $\boldsymbol{s_0}$ in the forward diffusion process according to the predetermined variance schedule $\beta_1, \cdots, \beta_T$, and β_t controls the noise level at time step t. The latent variable $\boldsymbol{s_t}$ is totally determined by its previous time step s_{t-1} at the time step t, and its formular is expressed as follows:

$$q(\boldsymbol{s_t}|\, \boldsymbol{s_{t-1}}) = \mathcal{N}\big(\, \boldsymbol{s_t};\, \sqrt{1-\beta_t}\boldsymbol{s_{t-1}}, \beta_t \boldsymbol{I}\big). \tag{3}$$

So, the data sample $\boldsymbol{s_0}$ gradually loses its distinguishable features as the time steps become large. Finally, the distribution of $\boldsymbol{s_T}$ is equivalent to an isotropic Gaussian distribution as $T \to \infty$. Correspondingly, for any time step t, $\boldsymbol{s_t}$ can be sampled directly from $\boldsymbol{s_0}$ in a closed form:

$$q(\boldsymbol{s_t}|\, \boldsymbol{s_0}) = \mathcal{N}\left(\boldsymbol{s_t}; \sqrt{1-\bar{\beta}_t}\boldsymbol{s_0}, \bar{\beta}_t \boldsymbol{I}\right), \tag{4}$$

where $\bar{\beta}_t = 1 - \prod_{i=0}^{t}(1-\beta_i)$. Usually, β_t gradually become larger while time steps increase, so $\beta_1 < \beta_2 < \cdots < \beta_T$.

The objective of diffusion models is to reverse the aforementioned forward process. By achieve a reversal of the forward process, we can recreate molecular data from an isotropic Gaussian noise input $\boldsymbol{s_T} \sim \mathcal{N}(0, I)$. So, in the backward process, the model gradually denoise the data and try to recover the data $\boldsymbol{s_0}$ by denoising from $\boldsymbol{s_T}$ with each step expressed by follows:

$$p_\theta(\boldsymbol{s_{t-1}}|\boldsymbol{s_t}) = \mathcal{N}\big(\boldsymbol{s_{t-1}}; \mu_\theta(\boldsymbol{s_t}, t), \Sigma_\theta(\boldsymbol{s_t}, t)\big), \tag{5}$$

where $\mu_\theta(\cdot)$ and $\Sigma_\theta(\cdot)$ are the predicted mean and covariance of the forward step $q(\boldsymbol{s_t}|\, \boldsymbol{s_{t-1}})$, and θ represents the parameters of the neural network used in the diffusion model. Empirically, Ho et al. [19] found that learning $\Sigma_\theta(\cdot)$ often leads to unstable training process and poorer sample results. Therefore, we follow their setting and choose to fix $\Sigma_\theta(\boldsymbol{s_t}, t) = \beta_t \boldsymbol{I}$ during training.

The objective loss function of the diffusion model is the variational lower-bound (LVB) of $log\, p_\theta\, (\boldsymbol{s_0})$:

$$\mathcal{L}_{lvb}(\boldsymbol{s_0}) = E_{q(\,\boldsymbol{s_{1:T}}|\boldsymbol{s_0}\,)}\left[log\, \frac{q(\,\boldsymbol{s_T}\mid \boldsymbol{s_0}\,)}{p_\theta(\boldsymbol{S_T})} + \sum_{t=2}^{T} log\, \frac{q(\,\boldsymbol{s_{t-1}}\mid \boldsymbol{s_0}, \boldsymbol{s_t}\,)}{p_\theta(\,\boldsymbol{s_{t-1}}\mid \boldsymbol{s_t}\,)} - log\, p_\theta\,(\,\boldsymbol{s_0}\mid \boldsymbol{s_1}\,)\right]. \tag{6}$$

Ho et al. [19] simplified and reweighted the $\mathcal{L}_{lvb}(\boldsymbol{s_0})$ function and it can be expressed by a simple version:

$$\mathcal{L}_{simple}(\boldsymbol{s_0}) = E_{s_0,t}\left[\frac{1}{2\sigma_t^2}\|\tilde{\mu}_t(\boldsymbol{s_t}, \boldsymbol{s_0}) - \mu_\theta(\boldsymbol{s_t}, t)\|^2\right], \tag{7}$$

where $\tilde{\mu}_t(\boldsymbol{s_t}, \boldsymbol{s_0})$ represents the mean of the posterior $q(\boldsymbol{s_{t-1}}|\boldsymbol{s_0}, \boldsymbol{s_t})$. However, this formulation is insufficient for sequential generation tasks and cannot generate sequence data

effectively. So, we followed Li et al. [24] and reparametrized the $\mu_\theta(\boldsymbol{s_t}, t)$ term as:

$$\mu_\theta(\boldsymbol{s}_t, t) = \tilde{\mu}_t(\boldsymbol{s}_t, \hat{\boldsymbol{s}}_0), \tag{8}$$

where $\hat{\boldsymbol{s}}_0$ is the prediction results of our diffusion model $f_\theta(\cdot)$, *i.e.*, $\hat{\boldsymbol{s}}_0 = f_\theta(\boldsymbol{s}_t, t)$. In particular, $f_\theta(\cdot)$ is a six-layer twelve-head attention network [10] in this work, as shown in Figure 1(c). The final objective function for diffusion model can be expressed as:

$$\mathcal{L}_{diff}(\boldsymbol{s}_0) = E_{\boldsymbol{s}_0,t}\left[\frac{1}{2\sigma_t^2}\|\boldsymbol{s}_0 - \hat{\boldsymbol{s}}_0\|^2\right]. \tag{9}$$

DiffSeqMol for Unconditional Molecule Generation

To enable the application of continuous diffusion models in molecule generation task, we firstly tokenized the sequence of molecular data in SMILES formulation by BPE (Byte Pair Encoding [30]). We used an opensource BPE provided on Hugging Face[1]. Then, an embedding layer $EMB(\boldsymbol{w}_i)$ is needed to map the discrete molecular tokens into a continuous space since the function $\mathcal{L}_{diff}$ is designed for continuous situations, as shown in previous nature language generation tasks [24]. Therefore, each molecule is encoded into continuous features, $EMB(\boldsymbol{w}) = [EMB(\boldsymbol{w}_1), EMB(\boldsymbol{w}_2), \cdots, EMB(\boldsymbol{w}_n)] \in R^{n\times d}$ in the forward diffusion process. After that, the embeddings of molecular word tokens are further perturbed by an additional Gaussian noise as follows:

$$q(\boldsymbol{s}_0|\boldsymbol{w}) = \mathcal{N}(\boldsymbol{s}_0; EMB(\boldsymbol{w}), \beta_0\boldsymbol{I}). \tag{10}$$

In the diffusion backward process, a trainable rounding layer [26] is designed to learn molecular token embeddings as follows:

$$p_\theta(\boldsymbol{w}|\boldsymbol{s}_0) = \prod_{i=1}^{n} p_\theta(\boldsymbol{w}_i|\boldsymbol{s}_{0,i}), \tag{11}$$

where $p_\theta(\boldsymbol{w}_i|\boldsymbol{s}_{0,i})$ is a Softmax distribution and its parameter is equal to $EMB(\cdot)$. The diagram of the overall unconditional generation process is shown in the Figure 1(a). The final objective function of DiffSeqMol for unconditional molecule generation should be able to simultaneous learn the parameters of diffusion model and embeddings of molecular tokens, which can be formulated as follows:

$$\mathcal{L}_{uncond\_dsm}(\boldsymbol{w}) = E_{\boldsymbol{w}}\left[\mathcal{L}_{diff}(\boldsymbol{s}_0) + ||EMB(\boldsymbol{w}) - f_\theta(\boldsymbol{s}_1, 1)||^2 - log\, p_\theta(\boldsymbol{w} \mid \boldsymbol{s}_0)\right], \tag{12}$$

where the first term is the loss function of continuous diffusion model, the second and the third term are designed to build the mutual mapping between discrete tokens and continuous features. Given the application of the attention layer within the diffusion model, all tokens'features are simultaneous denoised and DiffSeqMol can model the entire molecular sequence simultaneously. In the inference stage, DiffSeqMol is capable of generating novel molecules by gradually denoising randomly sampled Gaussian noises (Equation 5).

DiffSeqMol for Molecule Optimization

Conditional molecular optimization tasks bear resemblance to Nature Language

[1] huggingface.co/seyonec/PubChem10M_SMILES_BPE_450k

Translation tasks in Natural Language Processing. Sequential models for molecular optimization tasks need to deal with data in a sequence-to-sequence manner. So, to equip DiffSeqMol with molecule optimization ability, subtle modifications were introduced to the input of the diffusion model. Particularly, we treated target molecules as reference conditions. Moreover, during training, both the target molecule and the optimized molecule are inputted into the diffusion model. We followed the partial noising strategy proposed by [26]. In specific, give a pair of molecular sequence denoted as $[\boldsymbol{w}^x, \boldsymbol{w}^y]$, where $\boldsymbol{w}^x$ is the target molecule and $\boldsymbol{w}^y$ represents the optimized molecule, we concatenation them together, which can be formulated as follows:

$$EMB(\boldsymbol{w}^x,\ \boldsymbol{w}^y) = \left[EMB(\boldsymbol{w}_1^x), \cdots, EMB(\boldsymbol{w}_n^x),\ EMB\left(\boldsymbol{w}^{sep}\right),\ EMB\left(\boldsymbol{w}_1^y\right), \cdots, EMB\left(\boldsymbol{w}_m^y\right)\ \right], \quad (13)$$

where $\boldsymbol{w}^{sep}$ represents the separation token which help model in differentiating between the conditional molecule and the optimized molecule, and $EMB(\boldsymbol{w}^x,\ \boldsymbol{w}^y) \in R^{(n+m+1)\times d}$. This step can adapt pair-wise molecular data into the unconditional model without changing its fundamental architecture. However, the model is designed to avoid corrupting features from target molecules during the diffusion forward process, since $\boldsymbol{w}^x$ forms the target of the molecular optimization task. Therefore, for each step in the optimization version of forward process, noise is only introduced to tokens related to $\boldsymbol{w}^y$ (*i.e.*, all tokens after the separation token $\boldsymbol{w}^{sep}$). Since the nature of attention layer used in $f_\theta(\cdot)$, the semantic relationship between the conditional information ($\boldsymbol{w}^x$) and the corrupted molecule features ($\boldsymbol{w}^y$) can be learnt simultaneously. This noising strategy permits the sampling of optimized molecules based on the targe molecule $\boldsymbol{w}^x$. The diagram of the overall molecular optimization process is shown in the Figure 1(b). Finally, the loss function of the optimization-version of DiffSeqMol can be expressed as:

$$\mathcal{L}_{cond\_dsm}(\boldsymbol{w}^x, \boldsymbol{w}^y) = E_{\boldsymbol{w}^x,\boldsymbol{w}^y}\left[\mathcal{L}_{diff}(\boldsymbol{s}_0) + ||EMB(\boldsymbol{w}^x, \boldsymbol{w}^y) - f_\theta(\boldsymbol{s}_1, 1)||^2 - log\, p_\theta\left(\boldsymbol{w}^x, \boldsymbol{w}^y \mid \boldsymbol{s}_0\right)\right]. \quad (14)$$

Although the nosing injection step only affect features related to $\boldsymbol{w}^y$, DiffSeqMol learns the mapping step between the discrete tokens and continuous features for all the input tokens. In this way, the reconstruction layer of $\boldsymbol{w}^x$ and $\boldsymbol{w}^y$ incorporate information from each sides simultaneously. In the inference stage, DiffSeqMol can optimize the target molecules by gradually denoising randomly sampled Gaussian noises (Equation 5). Different from the conventional autoregressive optimization model, DiffSeqMol generates the optimized molecules in one shot，which poses a more substantially challenge. Therefore, we implement an iterative process of molecular optimization 5 times to achieve peak effectiveness.

Experiments and results

Tasks and Datasets

DiffSeqMol is primarily designed for unconditional molecular generation tasks and molecular optimization tasks. Therefore, we conducted two typical unconditional molecular generation tasks and four molecular optimization tasks.

High Penalized LogP Molecule Generation Task (unconditional generation task 1): This task is designed for a widely used benchmark assessments for drug discovery in real-world situation. The objective is to generate molecules that exhibit high penalized LogP, a parameter that measures the solubility and synthetic accessibility of a compound. All molecules in this dataset have a good pLogP score above 4.0 and collected by Flam-Shepherd et al. [18]. This dataset was screened from ZINC15 [31] database and has 34.7 atoms on average.

Multi-distribution Molecule Generation Task (unconditional generation task 2): This task is designed to assess whether the generative model can cope with data from complicated molecular property distribution. This dataset was collected from several dataset (GDB [32], ZINC [31], CEP [33] and POLYMERS [34]) by Flam-Shepherd et al. [18]. It has 31.1 atoms on average.

Penalized LogP Optimization Task (molecule optimization task 1&2): This task is designed for optimizing the Penalized LogP (plogP) score of the target molecules under the constriction of molecule similarity. The model's objective is to optimize a target molecule, denoted as x, and output an optimized compound y, such that $plogP(mol^x) < plogP(mol^y)$, under two different similarity constraints, 0.4 (optimization task 1) and 0.6 (optimization task 2). These datasets were collected by Jin et al. [17].

QED Optimization Task (molecule optimization task 3): This task is designed for optimizing the drug likeness (QED) score of the target molecules under the constriction of molecule similarity. In this optimization task, models need to optimize an input molecule x, whose qed score is within $[0.7, 0.8]$, into a higher score range $[0.9, 1.0]$. The similarity constraint requires that the similarity between mol^x and mol^y must exceed or equal to 0.4. This dataset was constructed by Jin et al. [17].

DRD2 Optimization Task (molecule optimization task 4): This task is designed for optimizing the biological activity against a biological target named the dopamine type 2 receptor (DRD2) under the constriction of molecule similarity. The score of the biological activity is evaluated from a pretrained model by Olivercrona et al. [35]. In this optimization task, models need to optimize an input molecule x, whose DRD2 score is below 0.05, into a higher score range above 0.5 (Molecules whose DRD2 score are above 0.5 are often considered as active compounds against DRD2). The similarity constraint requires that the similarity between mol^x and mol^y must exceed or equal to 0.4. This dataset was constructed by Jin et al. [17].

Unconditional Molecular Generation tasks

Each unconditional generation task's performance was evaluated by contrasting the property distribution of training molecules datasets and the distribution learned by different models. A histogram was employed to indicate the property of molecules from datasets, while a Gaussian Kernel Density Estimator (KDE) was used to mimic the property distribution. For each property, we plotted KDE of different models under the same bandwidth parameter. For the two tasks of unconditional generation, we generated $10k$ (thousand) molecules for each model and used the generated molecules to calculate their physiochemical properties and metrics. We removed duplicates from the generated molecule sets. Then the Wasserstein

distance between the generated molecules and training set was calculated to provide a quantitative assessment of model's ability to learn the molecular property distribution of the training set. For the evaluation of distribution fitting ability of different models, we considered the following molecular properties: (1) octanol–water partition coefficient (LogP), (2) synthetic accessibility score (SA), (3) drug-likeness (QED) and (4) natural product likeness (NP). Besides the distribution fitting ability, we also discussed Validity, Uniqueness and Novelty of different methods, please refer to the supplementary material.

In this section, we compared DiffSeqMol with the following baselines: autoregressive sequence generation model, CG-VAE, JT-VAE and Hier-VAE. (1) Traditional autoregressive sequence generation model: Nowadays, huge amount of drug-related works adopt conventional RNN models [9] or Transformer models [10]. They generate molecules token by token. Since we found the performance of these two different models on generation tasks is almost same [36], we only showed the results of a vanilla RNN models [37] and used "Sequence" to represent it on the figure. (2) CG-VAE [38]: This is a VAE-based graph generation model that generates molecules in an autoregressive, atom-by-atom manner. (3) JT-VAE [17]: this is a constrained graph variational autoencoder model that autoregressively generates molecules based on simple substructures. (4) Hier-VAE [39]: this is a hierarchical graph variational autoencoder model that autoregressively generates molecules based on structural motifs.

The results of the unconditional generation tasks are depicted in Figure 2. As for the High Penalized LogP Molecule Generation Task, Figure 2(a) illustrates the property distribution of all generated molecules and Figure 2(b) shows the Wasserstein distance in relation to the different molecular properties derived from the real data. It is evident that the CG-VAE model barely captures the distribution of the original data, particularly in the SA and NP regions, and is largely disconnected from the main characteristics of the training data. Although other graph-based methods show improvement compared to CG-VAE, they still fall short of representing the true distribution accurately. In contrast, the conventional sequence model outperforms all other models, exhibiting the lowest distance. This phenomenon that the distribution learning ability of autoregressive language model is far better than graph-based methods have already been discussed in the previous works [18]. Our proposed method, DiffSeqMol, closely aligns with the primary distribution of LogP as depicted in Figure 2(a), albeit with a slightly higher value than the autoregressive sequence model in Figure 2(b). This indicates that DiffSeqMol performs well in fitting the main part of the property distribution but is slightly inferior to the autoregressive sequence model in fitting the details. This phenomenon is more obvious on other molecular properties. Compared to autoregressive methods, DiffSeqMol involves directly modeling the entire molecule, which significantly increases the level of difficulty. This leads to DiffSeqMol falling short in its ability to fit distributions compared to autoregressive sequence methods.

As for the Multi-distribution Molecule Generation Task, the distribution of molecules in this dataset shows multiple peaks, suggesting the presence of distinct subgroups, shown in Figure 2(c, d). Similar to the former task, the graph-based generative model continues to face challenges in fitting the multi-peak distribution. Despite the lower distance values observed in JT-VAE, it struggles to accurately capture the distribution of SA. In addition, it is worth noting that although other methods (excluding CGVAE), achieved lower Wasserstein

distances, none of them were capable of accurately fitting all the property distributions to perfection. Once again, the autoregressive sequence model demonstrates a clear advantage over graph-based models in this specific scenario. Our proposed method, DiffSeqMol, achieved comparable results to the JT-VAE and Hier-VAE on this dataset.

It is worth to note that only a limited amount of work has explored generating the whole molecule at once [40-43] and they are early explorations in the molecular generation field. These models are only effective on datasets with extremely small numbers of atoms (such as molecules with atom smaller than 9), and their results lagged far behind those of autoregressive-based methods. DiffSeqMol successfully fills this gap and achieves comparable or even better performance to mainstream autoregressive graph methods. Compared to pure autoregressive sequence models, there is still some distance. Some randomly generated molecules by DiffSeqMol can be found in Supplement Fig 1, which shows the capability of DiffSeqMol to generate diverse styles of molecules. Overall, our proposed diffusion-based model has the potential to be considered as a candidate for unconditional molecular generation models.

Molecular Optimization Tasks

For each molecule optimization task, model performance was evaluated in accordance with the specific task objectives. For task 1 and task 2, model performance was measured based on the degree of property improvement. Specifically, for each molecule mol^x in the test set, the target model was deployed for 20 random samples. The optimized molecule mol^y which achieves the highest plogP improvement and preserves the original molecule structure $similarity(mol^x, mol^y) > \delta$ was selected. After that, the average plogP improvement over the whole test set was reported on table 1. For task 3 and task 4, model evaluation was based on the optimization success rate. For each molecule mol^x in the test set, it was randomly sampled 20 times by the target model. If any molecule mol^y in the 20 molecular optimization candidates reached the target property line, it was considered as a success. After that, the average success rate over the whole test set was reported on table 2.

In this section, DiffSeqMol was compared with the following baselines: (1) models described in the unconditional experiment; (2) GCPN [44], an autoregressive graph generation model based on reinforcement learning; (3) MMPA [45], a chemical rule-based generation model which presumed additivity of chemical properties and (4) JTNN [46], an autoregressive graph-to-graph structure which generates molecules structure by structure.

The results of all the models are presented in Table 1 and Table 2. Our model DiffSeqMol demonstrates the highest performance in both cases. As for two Penalized LogP optimization tasks, DiffSeqMol achieves the highest improvement in performance. Relative to the previous best-performing model Hier-VAE, our model exhibits an obviously improvement (nearly 10%) on the plogP optimization (0.6 similarity constraint) and a modest improvement on the 0.4 similarity constraint dataset. These results demonstrated that DiffSeqMol is effective in identifying the parts of a molecule that require optimization and making corresponding improvements. To give readers a more intuitive understanding, we randomly selected some optimization results to display in Figure 3. The parts that have been optimized compared to

the target molecule were highlighted with dashed circles. Similar results are also observed in QED optimization task and DRD2 optimization task. DiffSeqMol achieves a 4% enhancement of successful rate in the QED task and a 7% improvement in the DRD2 optimization task compared with the previous best-performing model Hier-VAE.

Visualization of the backward process

Unlike conventional autoregressive generation models, DiffSeqMol directly generates the entire molecular expression. To offer readers insight into the process by which DiffSeqMol generates molecules, we visualized the intermediate molecular results during the backward process, as exhibited in Figure 4. In Figure 4, part (a) represents the specific length of the molecular tokens during the backward process, while part (b) represents the intermediate generated results. Due to the excessively long length of the early intermediate results, we truncated them and show the head part and the tail part. As the denoising process progresses, the length of the molecule gradually decreases. Especially during the first half of the backward process, the intermediate results experience a sharp decrease in length from around 170 to around 30. Meanwhile, DiffSeqMol incrementally converts a portion of the tokens in the latter half of the intermediate results into "<pad>" tokens. In the latter half of the backward process, the length of the molecule stabilizes. DiffSeqMol's focus transitions from length determination to the selection of suitable molecular tokens pertinent to the target task. The process of selecting molecular tokens roughly takes up 40% of the total backward process, ranging from 50%T to 10%T of the backward process. Afterwards, the molecular results stabilize and DiffSeqMol outputs the final molecules.

Conclusion

The advent of diffusion models has brought a considerable advancement to generation tasks. In this work, we explored the possibility of applying diffusion models to sequential molecular generation. DiffSeqMol achieves comparable level of performance to existing graph-based methods in unconditional molecular generation tasks. Additionally, it demonstrates notable results in optimization tasks. The implementation of a non-autoregressively sequential diffusion model can subsequently open new pathways to traverse the expansive chemical space and to discover novel molecules.

Code Availability: The code is available in: https://github.com/viko-3/DiffSeqMol.
Data Availability: All the data used in this paper are from publicly available datasets. For the dataset used in unconditional generation tasks, you can refer to Flam-Shepherd et al. (https://www.nature.com/articles/s41467-022-30839-x). For the dataset used in molecular

optimization tasks, you can refer to Jin et al. (https://arxiv.org/abs/2002.03230).

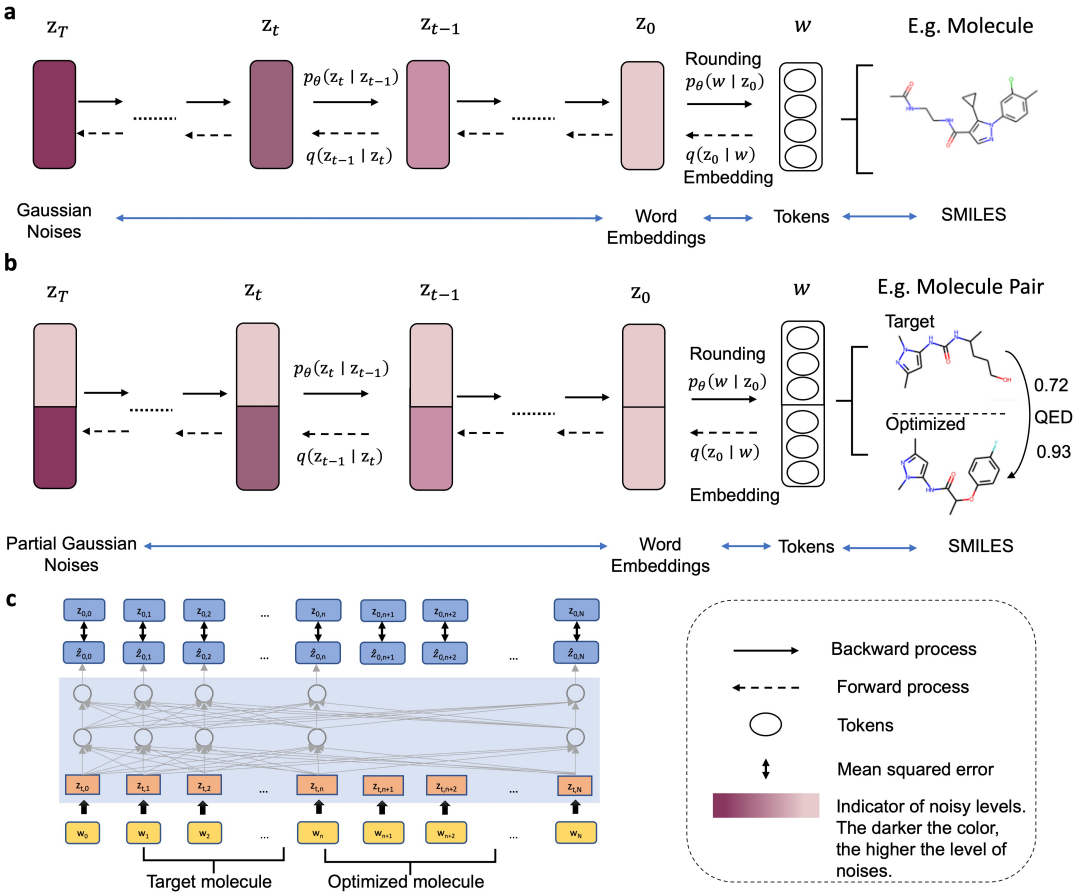

Figure 1. (a) The diagram of the overall unconditional generation process of DiffSeqMol. (b) The diagram of the overall molecular optimization process of DiffSeqMol. (c) The diagram of the attention network used in DiffSeqMol for molecular optimization. This network is designed for molecular token prediction (Equation 9). As for unconditional generation tasks, we eliminated the tokens of the optimized molecules.

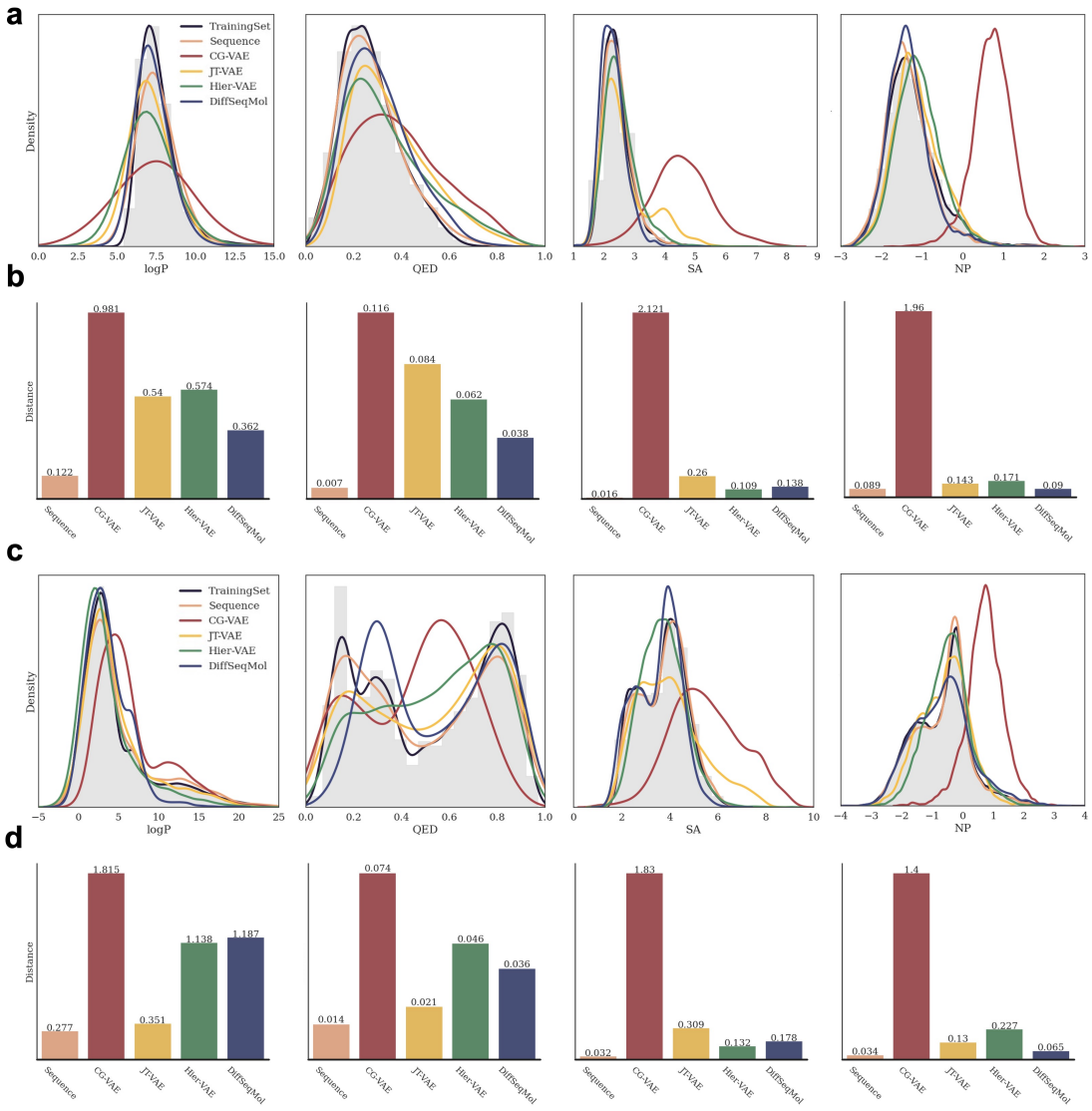

Figure 2. (a) Molecular property distribution calculated from samples generated by models trained on High Penalized LogP Molecule Generation Task (unconditional generation task 1).(b) The Wasserstein distance between the generated molecules and the training set, corresponding to the properties presented in (a). (c) Molecular property distribution calculated from samples generated by models trained on Multi-distribution Molecule Generation Task (Unconditional generation task 2). (d) The Wasserstein distance between the generated molecules and the training set, corresponding to the properties presented in (c).

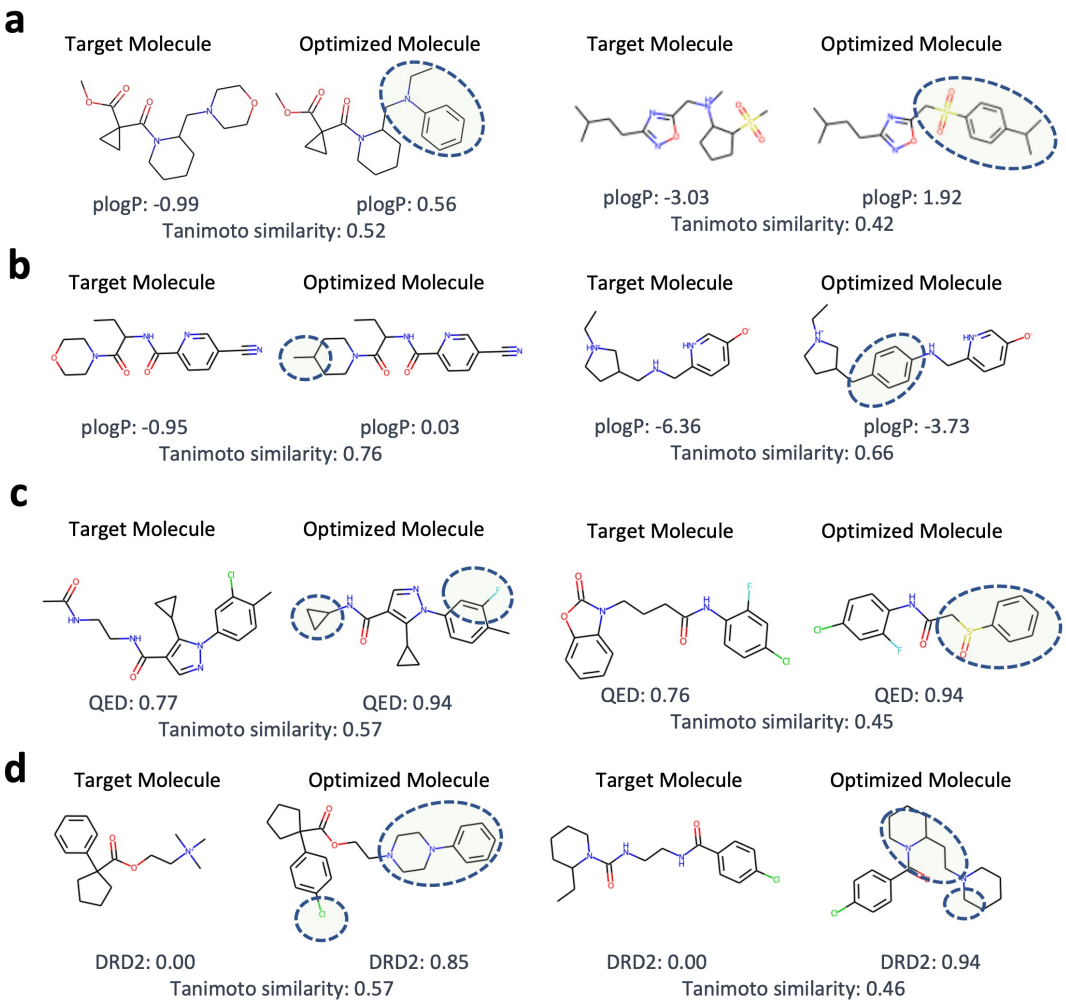

Figure 3. Randomly selected optimization results. (a) Randomly sampled from Penalized LogP Optimization Task (molecule optimization task 1). (b) Randomly sampled from Penalized LogP Optimization Task (molecule optimization task 2). (c) Randomly sampled from QED Optimization Task (molecule optimization task 3). (d) Randomly sampled from DRD2 Optimization Task (molecule optimization task 4).

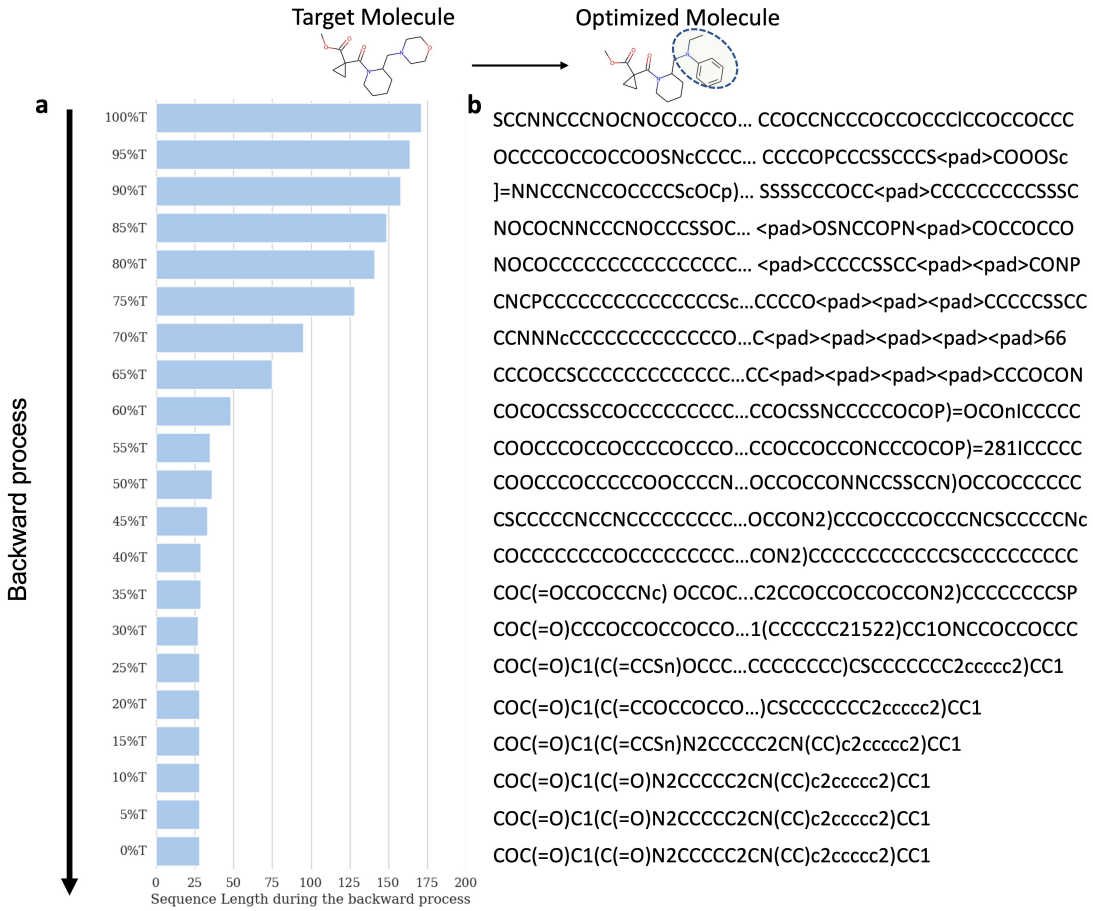

Figure 4. The intermediate molecular results during the backward process. We toke the molecular optimization result in Figure 3(a) as an illustrative example. (a) The length of the intermediate molecular tokens. (b) Truncated intermediate molecular tokens, corresponding to the temporal position in (a).

Table 1 Penalized LogP Optimization Results (Higher improvement scores of plogp reflects better performance.)

Method	plogp (Sim > 0.4) Improvement	plogp (Sim > 0.6) Improvement
JT-VAE	1.03 ± 1.39	0.28 ± 0.79
CG-VAE	0.61 ± 1.09	0.25 ± 0.74
GCPN	2.49 ± 1.30	0.79 ± 0.63
MMPA	3.29 ± 1.12	1.65 ± 1.44
Seq2Seq	3.37 ± 1.75	2.33 ± 1.17
JTNN	3.55 ± 1.67	2.33 ± 1.24
Hier-VAE	3.98 ± 1.46	2.49 ± 1.09
DiffSeqMol	**4.00 ± 1.55**	**2.72 ± 1.09**

Table 2 QED Optimization Results & DRD2 Optimization Results (Higher successful rates reflect better performance.)

Method	QED (Successful Rate)	DRD2 (Successful Rate)
JT-VAE	8.8%	3.4%
CG-VAE	4.8%	2.3%
GCPN	9.4%	4.4%
MMPA	32.9%	46.4%
Seq2Seq	58.5%	75.9%
JTNN	59.9%	77.8%
Hier-VAE	76.9%	85.9%
DiffSeqMol	**80.0%**	**92.2%**

Reference

1. Zeng, X., et al., *Deep generative molecular design reshapes drug discovery.* Cell Reports Medicine, 2022: p. 100794.
2. Butler, K.T., et al., *Machine learning for molecular and materials science.* Nature, 2018. **559**(7715): p. 547-555.
3. Meng, Y., et al., *A weighted bilinear neural collaborative filtering approach for drug repositioning.* Briefings in bioinformatics, 2022. **23**(2): p. bbab581.
4. Pan, X., et al., *Deep learning for drug repurposing: Methods, databases, and applications.* Wiley interdisciplinary reviews: Computational molecular science, 2022. **12**(4): p. e1597.
5. Jin, J., et al., *iDNA-ABF: multi-scale deep biological language learning model for the interpretable prediction of DNA methylations.* Genome biology, 2022. **23**(1): p. 1-23.
6. Jiang, Y., et al., *Explainable deep hypergraph learning modeling the peptide secondary structure prediction.* Advanced Science, 2023. **10**(11): p. 2206151.
7. Wang, R., et al., *DeepBIO: an automated and interpretable deep-learning platform for high-throughput biological sequence prediction, functional annotation and visualization analysis.* Nucleic Acids Research, 2023. **51**(7): p. 3017-3029.
8. Yang, X., et al., *CACPP: A Contrastive Learning-Based Siamese Network to Identify Anticancer Peptides Based on Sequence Only.* Journal of Chemical Information and Modeling, 2023.
9. Elman, J.L., *Finding structure in time.* Cognitive science, 1990. **14**(2): p. 179-211.
10. Vaswani, A., et al., *Attention is all you need.* Advances in neural information processing systems, 2017. **30**.
11. Weininger, D., *SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules.* Journal of chemical information and computer sciences, 1988. **28**(1): p. 31-36.
12. Segler, M.H., et al., *Generating focused molecule libraries for drug discovery with recurrent neural networks.* ACS central science, 2018. **4**(1): p. 120-131.
13. Kotsias, P.-C., et al., *Direct steering of de novo molecular generation with descriptor conditional recurrent neural networks.* Nature Machine Intelligence, 2020. **2**(5): p. 254-265.
14. Chen, Y., et al., *Deep generative model for drug design from protein target sequence.* Journal of Cheminformatics, 2023. **15**(1): p. 38.
15. Wang, J., et al., *De novo molecular design with deep molecular generative models for PPI inhibitors.* Briefings in Bioinformatics, 2022. **23**(4): p. bbac285.
16. Kingma, D.P. and M. Welling, *Auto-encoding variational bayes*, in *International Conference on Learning Representations, ICLR*. 2013.
17. Jin, W., R. Barzilay, and T. Jaakkola. *Junction tree variational autoencoder for molecular graph generation*. in *International conference on machine learning*. 2018. PMLR.
18. Flam-Shepherd, D., K. Zhu, and A. Aspuru-Guzik, *Language models can learn complex molecular distributions.* Nat Commun, 2022. **13**(1): p. 3293.

19. Ho, J., A. Jain, and P. Abbeel, *Denoising diffusion probabilistic models*. Advances in Neural Information Processing Systems, 2020. **33**: p. 6840-6851.
20. Sohl-Dickstein, J., et al. *Deep unsupervised learning using nonequilibrium thermodynamics*. in *International Conference on Machine Learning*. 2015. PMLR.
21. Ho, J., et al., *Video diffusion models.* arXiv preprint arXiv:2204.03458, 2022.
22. Kong, Z., et al., *Diffwave: A versatile diffusion model for audio synthesis.* arXiv preprint arXiv:2009.09761, 2020.
23. Rombach, R., et al. *High-resolution image synthesis with latent diffusion models*. in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022.
24. Li, X., et al., *Diffusion-lm improves controllable text generation.* Advances in Neural Information Processing Systems, 2022. **35**: p. 4328-4343.
25. Gao, Z., et al., *Difformer: Empowering Diffusion Model on Embedding Space for Text Generation.* arXiv preprint arXiv:2212.09412, 2022.
26. Gong, S., et al., *Diffuseq: Sequence to sequence text generation with diffusion models*, in *International Conference on Learning Representations, ICLR*. 2023.
27. Liu, S., et al., *A Text-guided Protein Design Framework.* arXiv preprint arXiv:2302.04611, 2023.
28. Ni, B., D.L. Kaplan, and M.J. Buehler, *Generative design of de novo proteins based on secondary-structure constraints using an attention-based diffusion model.* Chem, 2023.
29. Creswell, A., et al., *Generative adversarial networks: An overview.* IEEE signal processing magazine, 2018. **35**(1): p. 53-65.
30. Sennrich, R., B. Haddow, and A. Birch, *Neural machine translation of rare words with subword units.* arXiv preprint arXiv:1508.07909, 2015.
31. Irwin, J.J. and B.K. Shoichet, *ZINC− a free database of commercially available compounds for virtual screening.* Journal of chemical information and modeling, 2005. **45**(1): p. 177-182.
32. Blum, L.C. and J.-L. Reymond, *970 million druglike small molecules for virtual screening in the chemical universe database GDB-13.* Journal of the American Chemical Society, 2009. **131**(25): p. 8732-8733.
33. Hachmann, J., et al., *The Harvard clean energy project: large-scale computational screening and design of organic photovoltaics on the world community grid.* The Journal of Physical Chemistry Letters, 2011. **2**(17): p. 2241-2251.
34. St. John, P.C., et al., *Message-passing neural networks for high-throughput polymer screening.* The Journal of chemical physics, 2019. **150**(23): p. 234111.
35. Olivecrona, M., et al., *Molecular de-novo design through deep reinforcement learning.* Journal of cheminformatics, 2017. **9**(1): p. 1-14.
36. Chen, Y., et al., *Molecular language models: RNNs or transformer?* Briefings in Functional Genomics, 2023.
37. Polykovskiy, D., et al., *Molecular sets (MOSES): a benchmarking platform for molecular generation models.* Frontiers in pharmacology, 2020. **11**: p. 565644.
38. Liu, Q., et al., *Constrained graph variational autoencoders for molecule design.* Advances in neural information processing systems, 2018. **31**.
39. Jin, W., R. Barzilay, and T. Jaakkola. *Hierarchical generation of molecular graphs using*

structural motifs. in *International conference on machine learning*. 2020. PMLR.

40. Simonovsky, M. and N. Komodakis. *Graphvae: Towards generation of small graphs using variational autoencoders*. in *Artificial Neural Networks and Machine Learning–ICANN 2018: 27th International Conference on Artificial Neural Networks, Rhodes, Greece, October 4-7, 2018, Proceedings, Part I 27*. 2018. Springer.
41. De Cao, N. and T. Kipf, *MolGAN: An implicit generative model for small molecular graphs.* arXiv preprint arXiv:1805.11973, 2018.
42. Ma, T., J. Chen, and C. Xiao, *Constrained generation of semantically valid graphs via regularizing variational autoencoders.* Advances in Neural Information Processing Systems, 2018. **31**.
43. Flam-Shepherd, D., T.C. Wu, and A. Aspuru-Guzik, *MPGVAE: improved generation of small organic molecules using message passing neural nets.* Machine Learning: Science and Technology, 2021. **2**(4): p. 045010.
44. You, J., et al., *Graph convolutional policy network for goal-directed molecular graph generation.* Advances in neural information processing systems, 2018. **31**.
45. Dalke, A., J. Hert, and C. Kramer, *mmpdb: An open-source matched molecular pair platform for large multiproperty data sets.* Journal of chemical information and modeling, 2018. **58**(5): p. 902-910.
46. Jin, W., et al., *Learning multimodal graph-to-graph translation for molecular optimization.* arXiv preprint arXiv:1812.01070, 2018.