Text-Guided Molecule Generation with Diffusion Language Model

Haisong Gong[1,2], **Qiang Liu**[1,2], **Shu Wu**[1,2*], **Liang Wang**[1,2]

[1]Center for Research on Intelligent Perception and Computing
State Key Laboratory of Multimodal Artificial Intelligence Systems
Institute of Automation, Chinese Academy of Sciences
[2]School of Artificial Intelligence, University of Chinese Academy of Sciences
gonghaisong2021@ia.ac.cn, {qiang.liu, shu.wu, wangliang}@nlpr.ia.ac.cn

Abstract

Text-guided molecule generation is a task where molecules are generated to match specific textual descriptions. Recently, most existing SMILES-based molecule generation methods rely on an autoregressive architecture. In this work, we propose the Text-Guided Molecule Generation with Diffusion Language Model (TGM-DLM), a novel approach that leverages diffusion models to address the limitations of autoregressive methods. TGM-DLM updates token embeddings within the SMILES string collectively and iteratively, using a two-phase diffusion generation process. The first phase optimizes embeddings from random noise, guided by the text description, while the second phase corrects invalid SMILES strings to form valid molecular representations. We demonstrate that TGM-DLM outperforms MolT5-Base, an autoregressive model, without the need for additional data resources. Our findings underscore the remarkable effectiveness of TGM-DLM in generating coherent and precise molecules with specific properties, opening new avenues in drug discovery and related scientific domains. Code will be released at: https://github.com/Deno-V/tgm-dlm.

Introduction

Molecules, the fundamental building blocks of matter, intricately shape the properties and functions of our world. Novel molecules hold profound significance across scientific realms (Yao et al. 2016; Reiser et al. 2022; Montoya and Persson 2017), motivating research in fields like chemistry, materials science, and biology. Central to this pursuit is drug discovery, where identifying molecules with specific properties, particularly interactions with proteins or enzymes, is paramount (Ferreira et al. 2015).

Traditional drug discovery's resource-intensive nature is being transformed by artificial intelligence (AI) (Paul et al. 2021). Among AI's applications, generating drug-like molecules has drawn attention (Bagal et al. 2021; You et al. 2018; Guan et al. 2023), along with bridging molecules and language (Zeng et al. 2022; Liu et al. 2022). In this paper, we focus on a novel task presented by Edwards et al. (2022): *text-guided de novo molecule generation*. This innovative endeavor blends natural language with molecular structures, generating molecules based on textual descriptions. This task empowers more comprehensive control over molecule generation and transcends prior limitations in AI-assisted molecule design.

One widely adopted approach in the scientific community to represent a molecule is the simplified molecular-input line-entry system (SMILES) (Weininger 1988; Weininger, Weininger, and Weininger 1989). SMILES provides a compact and human-readable representation of chemical structures. As Figure 1(a) shows, atoms and bonds are represented by characters and symbols, enabling the encoding of complex molecular structures as strings. Recent research has extensively explored SMILES-based molecule generation, treating SMILES strings as a form of language and applying natural language generation techniques to produce innovative molecules. Among these efforts, the majority of contemporary SMILES-based molecule generation methods rely on an autoregressive architecture, wherein models predict the next character based on previously generated ones (Bagal et al. 2021; Frey et al. 2022; Edwards et al. 2022; Irwin et al. 2022). Such methods often build upon autoregressive models like GPT (Floridi and Chiriatti 2020), T5 (Raffel et al. 2020) and BART (Lewis et al. 2020).

Despite their proven success, the autoregressive nature of existing methods brings forth inherent limitations, where the fixed generation order constrains the models' adaptability, particularly in settings that demand precise control over the generation process (Li et al. 2022). Bubeck et al. (2023) demonstrates that autoregressive models, including the state-of-the-art GPT-4, encounter significant difficulties when facing content generation tasks under *global constraints*. This arises because the autoregressive nature prevents the model from revising previously generated content, compelling it to focus on predicting content far ahead instead. Within the domain of SMILES-based molecule generation, the textual descriptions of molecules represent these crucial global constraints. These descriptions encapsulate vital molecular attributes, including scaffold structures, deeply embedded throughout the entire SMILES sequence. Consequently, the autoregressive architecture hinders accurate incorporation of crucial global constraints represented by textual description in SMILES-based molecule generation. This inherent limitation underscores the need for a novel generation paradigm that offers enhanced control.

[*] Corresponding Author

To address the aforementioned nature of autoregressive nature, we explore the utilization of diffusion models in SMILES-based molecule generation. Unlike autoregressive models, diffusion models (Ho, Jain, and Abbeel 2020) generate content iteratively and holistically. These models have exhibited remarkable aptitude in capturing complex data distributions and accommodating global constraints in the image generation field (Rombach et al. 2022; Yang et al. 2022). Inspired by these successes, we present the Text-Guided Molecule Generation with Diffusion Language Model (**TGM-DLM**), a novel approach that leverages diffusion model to update token embeddings within the SMILES string collectively and iteratively. In this process, TGM-DLM executes a two-phase diffusion generation. In the first phase, TGM-DLM iteratively optimizes the embedding features from random noise, guided by the text description. However, some molecules generated during this phase may suffer from invalid issues, with SMILES strings that cannot represent real molecules due to problems including unclosed rings, unmatched parentheses and valence errors. To address this, the second phase acts as a correction phase, where the model iteratively optimizes the invalid SMILES strings without text guidance, ultimately forming valid representations. To achieve these improvements, TGM-DLM is trained with two objectives: denoising embeddings using the text description and recovering uncorrupted SMILES strings by deliberately feeding the model with corrupted invalid ones during the latter diffusion generation steps.

In a nutshell, our main contributions can be listed as follows,

- We are the first to introduce diffusion language model to SMILES-based text-guided molecule generation, offering a powerful approach for coherent and precise molecule generation.
- We propose TGM-DLM, a novel method with a two-phase diffusion generation process, enabling the generation of coherent molecules guided by text descriptions.
- TGM-DLM showcases superior performance, notably surpassing MolT5-Base, an autoregressive generation model pretrained on the Colossal Clean Crawled Corpus (C4) (Raffel et al. 2020) and ZINC-15 (Sterling and Irwin 2015) dataset and fine-tuned on the ChEBI-20 dataset (Edwards, Zhai, and Ji 2021). Notably, these results are achieved without any additional data resources, highlighting the effectiveness of our model.

Related Work

SMILES-based Molecule Generation

Molecules, often analyzed computationally, can be represented in various formats, including SMILES, graphs, and 3D structures (Chen et al. 2023; Zhu et al. 2022). In the realm of SMILES-based molecule generation, early approaches employed RNN-based methods that transformed SMILES strings into one-hot vectors and were sampled step-by-step (Segler et al. 2018; Grisoni et al. 2020). VAE-based methods, like ChemVAE (Gómez-Bombarelli et al. 2018) and SD-VAE (Dai et al. 2018), harnessed paired encoder-decoder architectures to generate SMILES strings and establish latent molecular spaces.

In recent years, the progress in natural language generation has sparked a surge in interest in autoregressive methods for molecule generation. Prominent models like GPT (Floridi and Chiriatti 2020) have exhibited impressive capabilities. ChemGPT (Frey et al. 2022) and MolGPT (Bagal et al. 2021) leverage GPT-based architectures for molecule generation, with MolGPT focusing on generating molecules with specific attributes. Chemformer (Irwin et al. 2022), on the other hand, employs BART (Lewis et al. 2020) as its foundational framework for molecule generation. MolT5 (Edwards et al. 2022), closely aligned with our work, employs a pretrained T5 (Raffel et al. 2020) to facilitate text-guided SMILES-based molecule generation. Distinctively, our approach stands out as the pioneer in utilizing the diffusion architecture as opposed to the autoregressive method.

Diffusion Models for Language Generation

Diffusion models have achieved notable accomplishments in generating content across continuous domains, spanning images (Ho, Jain, and Abbeel 2020; Rombach et al. 2022) and audios (Kong et al. 2021). Adapting diffusion models for discrete domains, such as language generation, has engendered an array of exploratory strategies. Some techniques incorporate discrete corruption processes, replacing the continuous counterpart, and leverage categorical transition kernels, uniform transition kernels, and absorbing kernels (Hoogeboom et al. 2021b,a; He et al. 2022). In contrast, certain methodologies maintain diffusion steps within the continuous domain by transforming language tokens into word vectors, executing forward and reverse operations within these vectors (Li et al. 2022; Gong et al. 2022). Our work adheres to the latter paradigm, operating on word vectors, and significantly broadening the capabilities of diffusion models within the realm of SMILES-based molecule generation. While the exploration of diffusion models for language generation remains relatively under-explored, our approach methodically investigates its potential within the SMILES-based molecule generation domain, effectively underscoring its effectiveness.

Preliminary and Task Formulation

Diffusion Framework for Language Generation

In this section, we lay the foundation for our work by introducing a comprehensive framework for the application of diffusion models in continuous domains, modified from the work of (Li et al. 2022). The framework comprises four pivotal processes: embedding, forward, reverse, and rounding. Together, these processes enable the generation of coherent and meaningful language output, as depicted in Figure 1(b).

The embedding process is the initial step where a text sequence $W = [w_0, w_1, \ldots, w_n]$ is treated as a sequence of words. Each word w_i undergoes an embedding transformation to yield $\mathrm{Emb}(W) = [\mathrm{Emb}(w_0), \mathrm{Emb}(w_1), \ldots, \mathrm{Emb}(w_n)] \in \mathbb{R}^{d \times n}$, with n denoting sequence length and d representing the embedding dimension. The starting matrix for the forward process $\mathbf{x}_0$

emerges by sampling from a Gaussian distribution centered at $\mathrm{Emb}(W)$: $\mathbf{x}_0 \sim \mathcal{N}(\mathrm{Emb}(W), \sigma_0 \mathbf{I})$.

The forward process gradually adds noise to $\mathbf{x}_0$, ultimately leading to the emergence of pure Gaussian noise $\mathbf{x}_T \sim \mathcal{N}(0, \mathbf{I})$. The transition from $\mathbf{x}_{t-1}$ to $\mathbf{x}_t$ is defined as follows:

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

where $\beta_t \in [0, 1]$ regulates the noise scale added during diffusion time step t, which is predefined constant.

The reversal process commences from $\mathbf{x}_T$ and progressively samples $\mathbf{x}_{t-1}$ based on $\mathbf{x}_t$, reconstructing the original content. Traditionally, this is achieved through a neural network trained to predict $\mathbf{x}_{t-1}$ given $\mathbf{x}_t$. However, for enhanced precision in denoising $\mathbf{x}_t$ towards some specific word vectors, a neural network f_θ is trained to directly predict $\mathbf{x}_0$ from $\mathbf{x}_t$. In this way, the denoising transition used here from $\mathbf{x}_t$ to $\mathbf{x}_{t-1}$ can be formulated as:

$$\begin{aligned} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; &\frac{\sqrt{\overline{\alpha}_{t-1}}\beta_t}{1-\overline{\alpha}_t} f_\theta(\mathbf{x}_t, t) \\ &+ \frac{\sqrt{\overline{\alpha}_t}(1-\overline{\alpha}_{t-1})}{1-\overline{\alpha}_t}\mathbf{x}_t, \frac{1-\overline{\alpha}_{t-1}}{1-\overline{\alpha}_t}\beta_t) \end{aligned} \tag{2}$$

where $\overline{\alpha}_t = \prod_{s=0}^{t}(1-\beta_s)$. Sequentially applying Equation 2 to a pure noise $\mathbf{x}_T$ enables us to iteratively sample $\mathbf{x}_{t-1}$, thereby yielding $\mathbf{x}_0$.

The rounding process completes the framework by reverting the embedding matrix to the original text sequence. Each column vector of $\mathbf{x}_0$ is mapped to the word with the closest L-2 distance in terms of word embeddings. Thus, through the reverse and rounding processes, any given noise can be effectively denoised to a coherent text output.

Task Formulation

The objective of text-guided molecule generation is to craft a molecule that aligns with a provided textual description. In a formal context, let $C = [w_0, w_1, \ldots, w_m]$ represent the given text description, where w_i denotes the i-th word within the text, m is the length of the text sequence. Our goal revolves around the construction of a model $\mathcal{F}$ that accepts the text as input and yields the intended molecule as output, mathematically expressed as $M = \mathcal{F}(C)$.

Method

Overview

Building upon the framework outlined in the preceding section, TGM-DLM follows a similar structure, generating a molecule's SMILES string by iteratively denoising a pure Gaussian noise $\mathbf{x}_T$ through a two-phase reverse process, as depicted in Figure 2(a). The first phase constitutes a reverse process encompassing $T - B$ denoising steps. It initiates with $\mathbf{x}_T$ and progressively refines the embedding matrix under the guidance of the text description C. Once the first phase concludes, an embedding matrix $\mathbf{x}_B$ is derived. This matrix is then scrutinized to verify if it corresponds to a valid molecule. The transformation of $\mathbf{x}_B$ to a SMILES string is facilitated by a rounding process, with the SMILES string's validity determined using the RDKit toolkit[1]. Should $\mathbf{x}_B$ not correspond to any actual molecule, it proceeds to the second phase—known as the correction phase. In this phase, which also comprises B denoising steps, $\mathbf{x}_B$ is employed as the starting point, ultimately culminating in the final embedding matrix $\mathbf{x}_0$. The $\mathbf{x}_0$ is then rounded to SMILES string, serving as the model's output. In the upcoming sections, we delve into the specifics of each component of our design.

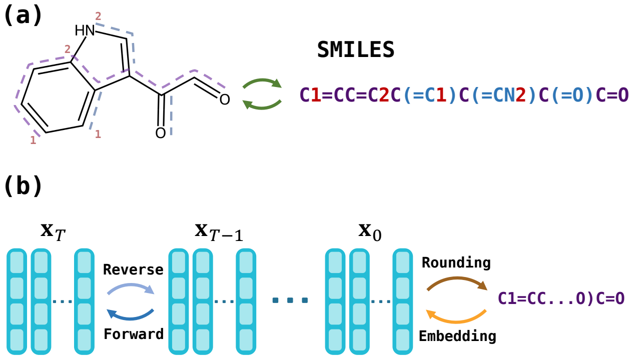

Figure 1: (a) Depiction of a molecule along with its corresponding SMILES representation. The main chain and side chains are colored purple and blue, respectively, both in the molecule graph and SMILES string. Ring numbering is highlighted in red. (b) The fundamental framework of the diffusion model for language generation. SMILES is treated as a sequence of language tokens. Through embedding and forward processes, the sequence transforms into pure noise. The reverse and rounding processes reconstruct the SMILES string from pure noise.

SMILES Tokenizer

Given our use of SMILES for molecular representation, each molecule M is treated as a sequence. Though the simplest approach would be to tokenize the SMILES string by individual characters, this approach poses significant challenges. It disrupts the unity of multi-character units and introduces ambiguity. For instance, the atom "scandium" represented in SMILES as [Sc] incorporates S and c, representing sulfur and aromatic carbon, respectively. Moreover, numeric characters within SMILES can either denote ring numbers or exist within atom groups like [NH3+], leading to ambiguity. Other methods, like those in autoregressive generation models, often rely on segmentation algorithms such as BPE (Gage 1994). In this study, we advocate treating each atom and atom group as a unified token, including additional tokens for bonds, ring numbers, parentheses, and special cases, thereby forming our vocabulary.

As a result, for a given molecule M, it is represented as a sequence list $M = [a_0, a_1, \cdots, a_n]$, where a_i represents the i-th token, and n is the maximum sequence length. For instance, the SMILES C(C(=O)O)[NH3+] is tokenized as [[SOS],C,(,C,(,=,O,),O,),[NH3+],[EOS],[PAD],...,[PAD]].

[1] https://rdkit.org

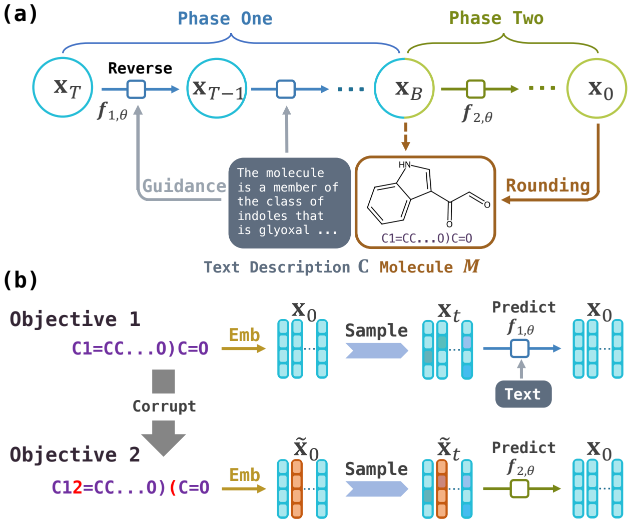

Figure 2: (a) Illustration of TGM-DLM's two-phase diffusion process. Phase one starts from pure noise, denoising $\mathbf{x}_t$ to $\mathbf{x}_B$ under text guidance. Phase two, without guidance, corrects phase one's outputs that can't be rounded to valid SMILES strings. (b) Two training objectives designed for TGM-DLM. The first objective entails denoising under text guidance, ensuring alignment with text descriptions. The second objective aims to enhance the model's ability to rectify invalid content, achieved by training it to recover embeddings from intentionally corrupted versions.

We pad every sequence to the maximum sequence length n.

After tokenization, the tokenized molecule M is ready for the embedding process, resulting in $\mathbf{x}_0$ via the sampling of $\mathbf{x}_0 \sim \mathcal{N}(\mathrm{Emb}(M), \sigma_0 \mathbf{I})$.

Phase One: Text-Guided Generation

The primary stage of diffusion generation encompasses a complete process involving $T-B$ diffusion steps. In contrast to generating molecules directly from unadulterated noise, the first phase of TGM-DLM incorporates text guidance to shape its generation process.

Several strategies exist to infuse the reverse process with the textual context. For instance, Li et al. (2022) employs an auxiliary classifier to steer the reverse process. Additionally, Rombach et al. (2022) enhanced diffusion models to serve as more versatile image generators, capable of being directed by semantic maps, textual descriptions, and images. This augmentation involves integrating the underlying model with a cross-attention mechanism. Drawing inspiration from this advancement, we incorporate cross-attention mechanism within TGM-DLM.

To introduce the text description, we utilize a pre-trained language model to map the text sequence C to its latent embeddings $\mathbf{C} \in \mathbb{R}^{d_1 \times m}$, where d_1 denotes the output embedding dimension of the language model. Our methods employs a Transformer model as the backbone, powering the neural network f_θ in Equation 2. We denote the function as $f_{1,\theta}$ to emphasize that it is used for the reverse process of phase one. Given the current diffusion state $\mathbf{x}_t$, diffusion step t, and text embeddings $\mathbf{C}$, $f_{1,\theta}$ predicts $\mathbf{x}_0$ as $\hat{\mathbf{x}}_0 = f_{1,\theta}(\mathbf{x}_t, t, \mathbf{C})$. We encode diffusion step t into the input of the first Transformer layer $\mathbf{z}_t^{(0)}$ using a technique akin to positional embedding (Vaswani et al. 2017):

$$\mathbf{z}_t^{(0)} = \mathbf{W}_{in}\mathbf{x}_t + \mathrm{PosEmb} + \mathrm{DE}(t) \tag{3}$$

where DE transforms t a vector, $\mathbf{W}_{in} \in \mathbb{R}^{d_2 \times d}$, $\mathbf{z}_t^{(0)} \in \mathbb{R}^{d_2 \times n}$, d_2 is the dimension for Transformer. PosEmb stands for the positional embedding.

The Transformer comprises L layers, each layer contains a cross-attention block, which introduces the text description into hidden states through the following mechanism:

$$\begin{aligned} \mathrm{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) &= \mathbf{V}\,\mathrm{softmax}(\frac{\mathbf{K}^T\mathbf{Q}}{\sqrt{d_2}}) \\ \mathbf{Q} &= \mathbf{W}_Q^{(i)} \mathbf{z}_t^{(i)} \\ \mathbf{K} &= \mathbf{W}_K^{(i)}\,\mathrm{MLP}(\mathbf{C}) \\ \mathbf{V} &= \mathbf{W}_V^{(i)}\,\mathrm{MLP}(\mathbf{C}) \end{aligned} \tag{4}$$

where, $\mathbf{W}_*^{(i)} \in \mathbb{R}^{d_2 \times d_2}$ represents the learnable parameters of the cross-attention block within the i-th layer, MLP is a multilayer perceptron.

After the process of phase one, we get the matrix $\mathbf{x}_B$ from pure noise $\mathbf{x}_T$ under the guidance of text description C. $\mathbf{x}_B$ is ready for the rounding process and being converted to SMILES string.

Phase Two: Correction

Following the diffusion process of phase one, the resulting SMILES strings obtained from $\mathbf{x}_B$ through the rounding process might occasionally fail to constitute a valid string. This underscores the need for a secondary phase—phase two—dedicated to rectifying such instances. As depicted in Figure 1(a), SMILES strings adhere to specific grammatical rules involving parentheses and numbers. Paired parentheses indicate branching on the main chain, while matching numbers represent bonds between atoms. To be valid, SMILES strings must satisfy these criteria, including meeting valence requirements for each atom.

By our observation, about three-quarters of the invalid SMILES strings generated by phase one fail to have paired parentheses and numbers. These erroneous SMILES strings have absorbed ample information from the text description, yet they contain minor inaccuracies in terms of rings and parentheses. Such issues can typically be rectified through minor adjustments involving the addition or removal of numbers and parentheses.

Unlike the condition-driven diffusion process of phase one, phase two's architecture closely mirrors its predecessor, employing B diffusion steps without text guidance, as adequate text information has already been acquired. Operating on the Transformer framework, phase two employs the denoising network $f_{2,\theta}$ as a post-processing module to refine $\mathbf{x}_B$ into a valid $\mathbf{x}_0$. It's important to note that this correction process introduces some distortion, affecting original information. Balancing effective correction with controlled distortion is achieved by tuning the steps taken during phase two, a topic explored in the upcoming experimental section.

Training

To endow the denoising network with the capability to execute both phase one and phase two, we equip TGM-DLM with two training objectives, as depicted in Figure 2(b).

Objective One: Denoising Training For the training of phase one, our approach centers around maximizing the variational lower bound of the marginal likelihood. Simplified and adapted from the works of Ho, Jain, and Abbeel (2020) and Li et al. (2022), the objective function compels $f_{1,\theta}$ to recover $\mathbf{x}_0$ at each step of the diffusion process:

$$\mathcal{L}_1(M, C) = \mathop{\mathbb{E}}_{q(\mathbf{x}_{0:T}|M)} \Big[\sum_{t=1}^{T} \|f_{1,\theta}(\mathbf{x}_t, t, \mathbf{C}) - \mathbf{x}_0\|^2 - \log p_\theta(M|\mathbf{x}_0) \Big] \tag{5}$$

where $p_\theta(M|\mathbf{x}_0)$ signifies the rounding process, characterized by a product of softmax distribution $p_\theta(M|\mathbf{x}_0) = \prod_{i=0}^{n} p(a_i|\mathbf{x}_{0[:,i]})$.

Objective Two: Corrective Training For the training of phase two, our approach deviates from predicting $\mathbf{x}_0$ from $\mathbf{x}_t$ and $\mathbf{C}$; instead, we compel the network to forecast $\mathbf{x}_0$ from a corrupted variant, denoted as $\tilde{\mathbf{x}}_t$, stemming from $\mathbf{x}_t$. As the primary objective of phase two revolves around rectifying unmatched rings and parentheses issues, we deliberately introduce such irregularities into $\mathbf{x}_t$.

In a conventional diffusion model training scenario, the acquisition of $\mathbf{x}_t$ involves the determination of $\mathbf{x}_0$ and the diffusion step t, followed by sampling via the equation:

$$\mathbf{x}_t = \sqrt{\overline{\alpha}_t}\mathbf{x}_0 + \sqrt{1 - \overline{\alpha}_t}\epsilon \tag{6}$$

where $\epsilon \sim \mathcal{N}(0, \mathbf{I})$.

To sample the corrupted $\tilde{\mathbf{x}}_t$, we initially infuse unmatched ring and parentheses complexities into the original molecule sequence M. This is achieved through a function Corrupt:

$$\tilde{M} = \mathrm{Corrupt}(M) \tag{7}$$

resulting in the altered molecule $\tilde{M}$. The function Corrupt incorporates a probability p of randomly adding or removing varying numbers of parentheses and ring numbers to disrupt the paired instances. Then, we transform $\tilde{M}$ to $\tilde{\mathbf{x}}_0$ through the embedding process, and $\tilde{\mathbf{x}}_t$ is sampled by:

$$\tilde{\mathbf{x}}_t = \sqrt{\overline{\alpha}_t}\tilde{\mathbf{x}}_0 + \sqrt{1 - \overline{\alpha}_t}\epsilon \tag{8}$$

The training loss for phase two becomes:

$$\mathcal{L}_2(M, \mathbf{C}) = \mathop{\mathbb{E}}_{q(\mathbf{x}_{0:T}|M)} \Big[\sum_{t=1}^{\tau} \|f_{2,\theta}(\tilde{\mathbf{x}}_t, t) - \mathbf{x}_0\|^2 + \sum_{t=\tau}^{T} \|f_{2,\theta}(\mathbf{x}_t, t, \mathbf{C}) - \mathbf{x}_0\|^2 - \log p_\theta(M|\mathbf{x}_0) \Big] \tag{9}$$

Note that when training $f_{2,\theta}$, we maintain the first $T - \tau$ reverse process steps in alignment with phase one training.

Algorithm 1: Training algorithm

```
repeat
    Sample $M$, $\mathbf{C}$ and $t$.
    Obtain $\mathbf{x}_0$ by embedding process
    if $t = 0$ then
        $g = \nabla_\theta(-\log p_\theta(M|\mathbf{x}_0))$
    else if phase two and $t < \tau$ then
        Obtain $\tilde{x}_t$ by Equation 7 and 8
        $g = \nabla_\theta\|f_{2,\theta}(\tilde{\mathbf{x}}_t, t) - \mathbf{x}_0\|^2$
    else
        $g = \nabla_\theta\|f_{*,\theta}(\mathbf{x}_t, t, \mathbf{C}) - \mathbf{x}_0\|^2$
    end if
    Take one step of optimization through gradient $g$
until converged
```

The introduction of corruption is confined only to the final τ steps. This strategic placement accounts for the fact that in the initial stages of the reverse process, $\mathbf{x}_t$ retains limited information, resembling noise, and any corruption introduced at this point could impede training and hinder convergence. See Algorithm 1 for the complete training procedure.

Experiment

In this section, we undertake experiments to assess the performance of our proposed TGM-DLM in text-guided molecule generation. Additionally, we delve into the impact of the second phase within our two-phase generation framework, a different training approach, and the outcome of integrating text guidance during the correction phase.

Experimental Setups

Dataset Given the nascent nature of our research focus, our evaluation centers on the ChEBI-20 dataset (Edwards et al. 2022), which is currently the sole publicly available dataset. This dataset encompasses a collection of 33,010 molecule-description pairs, which are separated into 80/10/10% train/validation/test splits. We adhere the data split setting in this paper.

Metrics Following previous work (Edwards et al. 2022), we employ nine metrics to evaluate the models.

- **SMILES BLEU** score and **Levenshtein** distance. Measure the similarity and distance of generated SMILES string to the ground truth molecule SMILES string.
- **MACCS FTS**, **RDK FTS** and **Morgan FTS**. Evaluate average Tanimoto similarity between generated and ground truth fingerprints.
- **Exact** score and **Validity**. The percentage of generated molecules that exactly match the ground truth and the percentage of generated strings that are valid.
- **FCD** and **Text2Mol** score. Measure latent information agreement using Fréchet ChemNet Distance (FCD) and assess relevance between text description and generated molecule using Text2Mol (Edwards, Zhai, and Ji 2021).

Model	BLEU$\uparrow$	Exact$\uparrow$	Levenshtein$\downarrow$	MACCS FTS$\uparrow$	RDK FTS$\uparrow$	Morgan FTS$\uparrow$	FCD$\downarrow$	Text2Mol$\uparrow$	Validity$\uparrow$
Ground Truth	1.000	1.000	0.000	1.000	1.000	1.000	0.00	0.609	1.000
Transformer	0.499	0.000	57.660	0.480	0.320	0.217	11.32	0.277	**0.906**
T5-Base	0.762	0.069	24.950	0.731	0.605	0.545	2.48	0.499	0.660
MolT5-Base	0.769	0.081	24.458	0.721	0.588	0.529	2.18	0.496	0.772
TGM-DLM$_{\text{w/o corr}}$	**0.828**	**0.242**	**16.897**	**0.874**	**0.771**	**0.722**	0.89	**0.589**	0.789
TGM-DLM	0.826	**0.242**	17.003	0.854	0.739	0.688	**0.77**	0.581	0.871

Table 1: Text-guided molecule generation results on ChEBI-20 test split. The results for Transformer, T5-Base and MolT5-Base are retrieved from (Edwards et al. 2022). We bold the best scores and underline the second-best scores. TGM-DLM$_{\text{w/o corr}}$ is the results generated by phase one, while TGM-DLM combines two phases, with the second phase increasing Validity by 8.2%.

Baselines Three baseline models are selected for comparison. All three models are autoregressive generation models.

- **Transformer** (Vaswani et al. 2017). A vanilla Transformer model with six encoder and decoder layers directly trained on the ChEBI-20 dataset.
- **T5-Base** (Raffel et al. 2020). A sequence-to-sequence generation model pre-trained on the C4 dataset, fine-tuned on ChEBI-20 dataset.
- **MolT5-Base** (Edwards et al. 2022). This model is initialized from a pre-trained T5 model, and further pre-trained on a combined dataset of C4 and ZINC-15 to gain domain knowledge in both molecules and natural languages. It is then fine-tuned on the ChEBI-20 dataset.

It's worth noting that the options for baseline models in the realm of text-based molecule generation task are limited. We choose the base versions of the T5 model and MolT5 model for a fair comparison, as language model's performance is positively correlated with the size of its parameters. The base version of T5/MolT5 comprises approximately 220M parameters, which is around 22% larger than TGM-DLM with about 180M parameters.

Implementation Details

We set the maximum sequence length for tokenized SMILES strings to $n = 256$. Molecule-description pairs with SMILES string lengths exceeding 256 were filtered out (approximately 1% of the entire dataset). SMILES vocabulary contained 257 tokens, with trainable token embeddings set at $d = 32$.We employed SciBERT (Beltagy, Lo, and Cohan 2019) as our frozen encoder for text descriptions, with an embedding dimension of $d_1 = 768$. The Transformer network for $f_{*,\theta}$ comprises $L = 12$ layers, and the hidden dimension is configured as $d_2 = 1024$. TGM-DLM is composed of approximately 180M trainable parameters.

During molecule generation, we adopt a uniform skipping strategy for reverse steps to enhance sampling efficiency. As a result, the practical number of sample steps is 200 for phase one and 20 for phase two. Note that the steps used for phase two are between 0 to τ. In this way, it only takes about 1.2 seconds on average to generate one molecule from its description on our hardware (AMD EPYC 7742 (256) @ 2.250GHz CPU and one NVIDIA A100 GPU).

During training, we set the total diffusion steps to $T = 2,000$ for both phase one and phase two. For phase two, τ is set to 400, and the corruption probability p is set to 0.4. We used Adam (Kingma and Ba 2015) as the optimizer, employing linear warm-up and a learning rate of 1e-4. We train separate denoising models for each phase using objective one and objective two, respectively. It is noteworthy that the two denoising models share the same architecture. We also experimented with training a single model using the second objective for both phases, and we present the results of this comparison in the subsequent section.

Overall Performance

We compare TGM-DLM with baseline models in Table 1. In general, our model outperforms all baseline models. While Transformer excels in Validity, its performance is poor in other metrics, highlighting limited efficacy. In contrast, our model excels in all other metrics. When compared to MolT5-base, our model exhibits a remarkable tripling of the exact match score, 18% to 36% improvement in FTS metrics. TGM-DLM generates molecules that are most similar to the ground truth molecules and exhibit the closest alignment to the text descriptions, as evidenced by the highest Text2Mol score. Notably, our model achieves these results without additional data or pre-training, distinguishing it from MolT5-Base. For more intuitive comparison, we showcase examples in Figure 3.

When comparing TGM-DLM with TGM-DLM$_{\text{w/o corr}}$, the latter represents results generated solely by phase one. We observe the successful rectification of invalid SMILES strings by phase two, boosting the Validity metric from 78.9% to 87.1%. However, this enhancement is accompanied by a slight reduction in other metrics. This can be attributed to two factors: firstly, the computation of metrics excluding BLEU, Exact, Levenshtein, and Validity considers only valid SMILES, thus the correction phase expands the metric scope. Secondly, the corrective phase inherently introduces subtle distortions in original molecular information, leading to a slight metric reduction.

Influence of Correction Phase

In this section, we investigate the impact of the two-phase design. A key factor is the number of steps employed during molecule sampling in phase two. As previously elucidated in the implementation section, we adopt a relatively short diffusion process for phase two. By varying the number of steps in phase two, we generate diverse variants of TGM-DLM.

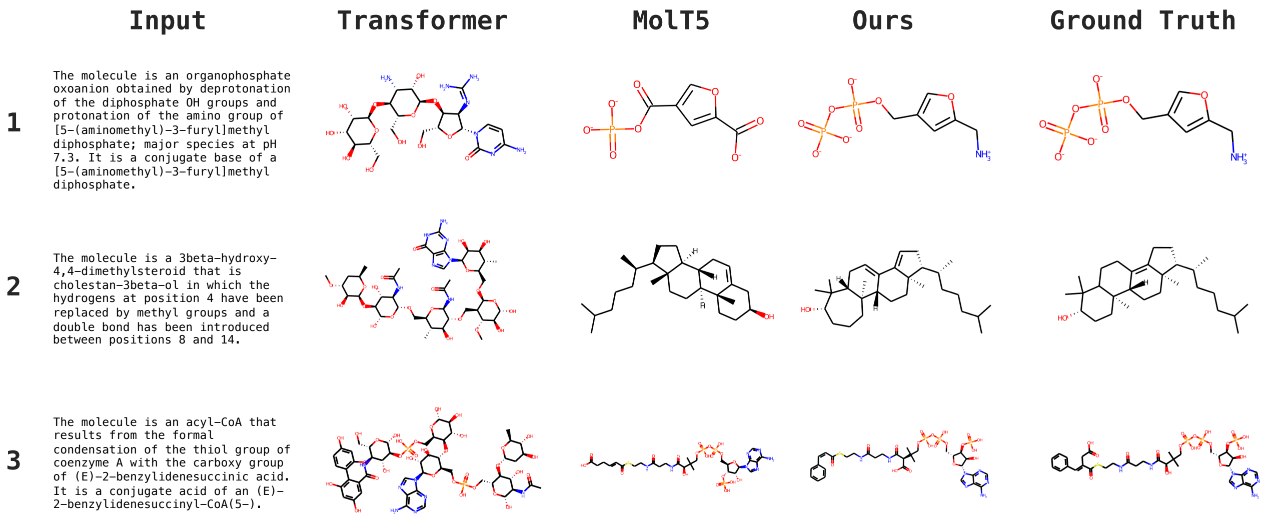
Input
Transformer
MolT5
Ours
Ground Truth
1
The molecule is an organophosphate oxoanion obtained by deprotonation of the diphosphate OH groups and protonation of the amino group of [5-(aminomethyl)-3-furyl]methyl diphosphate; major species at pH 7.3. It is a conjugate base of a [5-(aminomethyl)-3-furyl]methyl diphosphate.
2
The molecule is a 3beta-hydroxy-4,4-dimethylsteroid that is cholestan-3beta-ol in which the hydrogens at position 4 have been replaced by methyl groups and a double bond has been introduced between positions 8 and 14.
3
The molecule is an acyl-CoA that results from the formal condensation of the thiol group of coenzyme A with the carboxy group of (E)-2-benzylidenesuccinic acid. It is a conjugate acid of an (E)-2-benzylidenesuccinyl-CoA(5-).

Figure 3: Example of molecules generated by different models with the same input descriptions. Generated SMILES strings are converted to molecule graphs for better visualization.

Model	BLEU	MACCS FTS	Text2Mol	Validity
TGM-DLM$_{\text{w/o corr}}$	**0.828**	**0.874**	**0.589**	0.789
TGM-DLM$_{0.5\times}$	0.827	0.858	0.584	0.855
TGM-DLM$_{1\times}$	0.826	0.854	0.581	0.871
TGM-DLM$_{2\times}$	0.825	0.851	0.580	0.875
TGM-DLM$_{3\times}$	0.825	0.849	0.579	**0.883**
TGM-DLM$_{joint}$	0.819	0.846	0.576	0.855
TGM-DLM$_{text}$	0.827	0.869	0.588	0.824

Table 2: Comparison of TGM-DLM and its variants. Representative metrics are selected to display. TGM-DLM$_{2\times}$ is a variant with twice the number of phase two steps. TGM-DLM$_{joint}$ denotes the variant using a singular network trained for both phases. TGM-DLM$_{text}$ is a variant that incorporates text input in the correction phase.

For instance, TGM-DLM$_{2\times}$ employs twice the number of phase two steps. The findings are presented in Table 2. In general, validity increases as the number of phase two steps expands, albeit at the expense of other metrics. Given that only invalid SMILES strings from phase one are affected, adjusting the phase two step count allows us to flexibly strike a balance between Validity and the preservation of original information in erroneous SMILES.

Joint Training of Two Phases

As described in the implementation section, we separately train $f_{1,\theta}$ and $f_{2,\theta}$ with corresponding training objectives. Nevertheless, owing to the identical architecture of the two networks, we explore training a singular network for both phase one and phase two. This is achieved by employing Algorithm 1 while disregarding phase two condition at line 6. The results in Table 2 indicate that TGM-DLM$_{joint}$ performs less effectively than TGM-DLM. A potential explanation is that the correction training objective could influence the regular training objective.

Correction with Text Input

In TGM-DLM, the correction phase operates independently of text input. However, we explore an alternative approach where correction is performed with the utilization of text descriptions as input. The outcomes of this approach are presented in Table 2. From the table we can see that employing text input for correction does not yield the expected significant improvement in validity. In fact, our experiments reveal that augmenting the number of diffusion steps in phase two for TGM-DLM$_{text}$ fails to increase the validity beyond 82.4%. These findings indicate that TGM-DLM$_{text}$'s design is not well-suited for the correction phase, as the presence of text input may potentially impede the model's corrective ability by favoring adherence to textual guidance.

Conclusion

In this work, we present TGM-DLM, a novel diffusion model for text-guided SMILES-based molecule generation. This approach embeds SMILES sequences and employs a two-phase generation process. The first phase optimizes embeddings based on text descriptions, followed by a corrective second phase to address invalid SMILES generated by the first phase. Extensive experiments on ChEBI-20 dataset show TGM-DLM's consistent outperformance of autoregressive baselines, demonstrating better overall molecular attributes understanding, notably tripling the exact match score and 18% to 36% improvement in fingerprinting metrics over MolT5-Base. Remarkably, these advancements are attained without additional data sources or pre-training, highlighting TGM-DLM's effectiveness in text-guided molecule generation.

Acknowledgements

This work is jointly sponsored by National Natural Science Foundation of China (62206291, 62236010, 62141608).

References

Bagal, V.; Aggarwal, R.; Vinod, P.; and Priyakumar, U. D. 2021. MolGPT: molecular generation using a transformer-decoder model. *Journal of Chemical Information and Modeling*, 62(9): 2064–2076.

Beltagy, I.; Lo, K.; and Cohan, A. 2019. SciBERT: A Pretrained Language Model for Scientific Text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 3615–3620. Hong Kong, China: Association for Computational Linguistics.

Bubeck, S.; Chandrasekaran, V.; Eldan, R.; Gehrke, J.; Horvitz, E.; Kamar, E.; Lee, P.; Lee, Y. T.; Li, Y.; Lundberg, S.; et al. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. *ArXiv preprint*, abs/2303.12712.

Chen, D.; Zhu, Y.; Zhang, J.; Du, Y.; Li, Z.; Liu, Q.; Wu, S.; and Wang, L. 2023. Uncovering Neural Scaling Laws in Molecular Representation Learning. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Dai, H.; Tian, Y.; Dai, B.; Skiena, S.; and Song, L. 2018. Syntax-directed variational autoencoder for molecule generation. In *Proceedings of the international conference on learning representations*.

Edwards, C.; Lai, T.; Ros, K.; Honke, G.; Cho, K.; and Ji, H. 2022. Translation between molecules and natural language. *ArXiv preprint*, abs/2204.11817.

Edwards, C.; Zhai, C.; and Ji, H. 2021. Text2Mol: Cross-Modal Molecule Retrieval with Natural Language Queries. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 595–607. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics.

Ferreira, L. G.; Dos Santos, R. N.; Oliva, G.; and Andricopulo, A. D. 2015. Molecular docking and structure-based drug design strategies. *Molecules*, 20(7): 13384–13421.

Floridi, L.; and Chiriatti, M. 2020. GPT-3: Its nature, scope, limits, and consequences. *Minds and Machines*, 30: 681–694.

Frey, N.; Soklaski, R.; Axelrod, S.; Samsi, S.; Gomez-Bombarelli, R.; Coley, C.; and Gadepally, V. 2022. Neural scaling of deep chemical models.

Gage, P. 1994. A new algorithm for data compression. *C Users Journal*, 12(2): 23–38.

Gómez-Bombarelli, R.; Wei, J. N.; Duvenaud, D.; Hernández-Lobato, J. M.; Sánchez-Lengeling, B.; Sheberla, D.; Aguilera-Iparraguirre, J.; Hirzel, T. D.; Adams, R. P.; and Aspuru-Guzik, A. 2018. Automatic chemical design using a data-driven continuous representation of molecules. *ACS central science*, 4(2): 268–276.

Gong, S.; Li, M.; Feng, J.; Wu, Z.; and Kong, L. 2022. Diffuseq: Sequence to sequence text generation with diffusion models. *ArXiv preprint*, abs/2210.08933.

Grisoni, F.; Moret, M.; Lingwood, R.; and Schneider, G. 2020. Bidirectional molecule generation with recurrent neural networks. *Journal of chemical information and modeling*, 60(3): 1175–1183.

Guan, J.; Zhou, X.; Yang, Y.; Bao, Y.; Peng, J.; Ma, J.; Liu, Q.; Wang, L.; and Gu, Q. 2023. DecompDiff: Diffusion Models with Decomposed Priors for Structure-Based Drug Design.

He, Z.; Sun, T.; Wang, K.; Huang, X.; and Qiu, X. 2022. Diffusionbert: Improving generative masked language models with diffusion models. *ArXiv preprint*, abs/2211.15029.

Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising Diffusion Probabilistic Models. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Hoogeboom, E.; Gritsenko, A. A.; Bastings, J.; Poole, B.; Berg, R. v. d.; and Salimans, T. 2021a. Autoregressive diffusion models. *ArXiv preprint*, abs/2110.02037.

Hoogeboom, E.; Nielsen, D.; Jaini, P.; Forré, P.; and Welling, M. 2021b. Argmax flows and multinomial diffusion: Towards non-autoregressive language models. *ArXiv preprint*, abs/2102.05379.

Irwin, R.; Dimitriadis, S.; He, J.; and Bjerrum, E. J. 2022. Chemformer: a pre-trained transformer for computational chemistry. *Machine Learning: Science and Technology*, 3(1): 015022.

Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Kong, Z.; Ping, W.; Huang, J.; Zhao, K.; and Catanzaro, B. 2021. DiffWave: A Versatile Diffusion Model for Audio Synthesis. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; and Zettlemoyer, L. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 7871–7880. Online: Association for Computational Linguistics.

Li, X.; Thickstun, J.; Gulrajani, I.; Liang, P. S.; and Hashimoto, T. B. 2022. Diffusion-lm improves controllable text generation. *Advances in Neural Information Processing Systems*, 35: 4328–4343.

Liu, S.; Nie, W.; Wang, C.; Lu, J.; Qiao, Z.; Liu, L.; Tang, J.; Xiao, C.; and Anandkumar, A. 2022. Multi-modal molecule structure-text model for text-based retrieval and editing. *ArXiv preprint*, abs/2212.10789.

Montoya, J. H.; and Persson, K. A. 2017. A high-throughput framework for determining adsorption energies on solid surfaces. *npj Computational Materials*, 3(1): 14.

Paul, D.; Sanap, G.; Shenoy, S.; Kalyane, D.; Kalia, K.; and Tekade, R. K. 2021. Artificial intelligence in drug discovery and development. *Drug discovery today*, 26(1): 80.

Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1): 5485–5551.

Reiser, P.; Neubert, M.; Eberhard, A.; Torresi, L.; Zhou, C.; Shao, C.; Metni, H.; van Hoesel, C.; Schopmans, H.; Sommer, T.; et al. 2022. Graph neural networks for materials science and chemistry. *Communications Materials*, 3(1): 93.

Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; and Ommer, B. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10684–10695.

Segler, M. H.; Kogej, T.; Tyrchan, C.; and Waller, M. P. 2018. Generating focused molecule libraries for drug discovery with recurrent neural networks. *ACS central science*, 4(1): 120–131.

Sterling, T.; and Irwin, J. J. 2015. ZINC 15–ligand discovery for everyone. *Journal of chemical information and modeling*, 55(11): 2324–2337.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is All you Need. In Guyon, I.; von Luxburg, U.; Bengio, S.; Wallach, H. M.; Fergus, R.; Vishwanathan, S. V. N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, 5998–6008.

Weininger, D. 1988. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *Journal of chemical information and computer sciences*, 28(1): 31–36.

Weininger, D.; Weininger, A.; and Weininger, J. L. 1989. SMILES. 2. Algorithm for generation of unique SMILES notation. *Journal of chemical information and computer sciences*, 29(2): 97–101.

Yang, L.; Zhang, Z.; Song, Y.; Hong, S.; Xu, R.; Zhao, Y.; Shao, Y.; Zhang, W.; Cui, B.; and Yang, M.-H. 2022. Diffusion models: A comprehensive survey of methods and applications. *ArXiv preprint*, abs/2209.00796.

Yao, H.; Ye, L.; Zhang, H.; Li, S.; Zhang, S.; and Hou, J. 2016. Molecular design of benzodithiophene-based organic photovoltaic materials. *Chemical reviews*, 116(12): 7397–7457.

You, J.; Liu, B.; Ying, Z.; Pande, V. S.; and Leskovec, J. 2018. Graph Convolutional Policy Network for Goal-Directed Molecular Graph Generation. In Bengio, S.; Wallach, H. M.; Larochelle, H.; Grauman, K.; Cesa-Bianchi, N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, 6412–6422.

Zeng, Z.; Yao, Y.; Liu, Z.; and Sun, M. 2022. A deep-learning system bridging molecule structure and biomedical text with comprehension comparable to human professionals. *Nature communications*, 13(1): 862.

Zhu, Y.; Chen, D.; Du, Y.; Wang, Y.; Liu, Q.; and Wu, S. 2022. Featurizations matter: a multiview contrastive learning approach to molecular pretraining. In *ICML 2022 2nd AI for Science Workshop*.