

Text-guided diverse-expression diffusion model for molecule generation

Wenchao Weng(翁文超), Hanyu Jiang(蒋涵羽), Xiangjie Kong(孔祥杰), and Giovanni Pau

Citation: Chin. Phys. B, 2025, 34 (**5**): 050701. DOI: 10.1088/1674-1056/adbedd
Journal homepage: http://cpb.iphy.ac.cn; http://iopscience.iop.org/cpb

What follows is a list of articles you may be interested in

Literature classification and its applications in condensed matter physics and materials science by natural language processing

Siyuan Wu(吴思远), Tiannian Zhu(朱天念), Sijia Tu(涂思佳), Ruijuan Xiao(肖睿娟), Jie Yuan(袁洁), Quansheng Wu(吴泉生), Hong Li(李泓), and Hongming Weng(翁红明)

Chin. Phys. B, 2024, 33 (**5**): 050704. DOI: 10.1088/1674-1056/ad3c30

Preliminary abnormal electrocardiogram segment screening method for Holter data based on long short-term memory networks

Siying Chen(陈偲颖), Hongxing Liu(刘红星)

Chin. Phys. B, 2020, 29 (**4**): 040701. DOI: 10.1088/1674-1056/ab6d52

4.3 THz quantum-well photodetectors with high detection sensitivity

Zhenzhen Zhang(张真真), Zhanglong Fu(符张龙), Xuguang Guo(郭旭光), Juncheng Cao(曹俊诚)

Chin. Phys. B, 2018, 27 (**3**): 030701. DOI: 10.1088/1674-1056/27/3/030701

Thermal efficiency of the principal greenhouse gases

A. Y. Galashev, O. R. Rakhmanova

Chin. Phys. B, 2015, 24 (**1**): 010701. DOI: 10.1088/1674-1056/24/1/010701

SPECIAL TOPIC — Computational programs in complex systems

Text-guided diverse-expression diffusion model for molecule generation

Wenchao Weng(翁文超)[1,†], Hanyu Jiang(蒋涵羽)[2,†], Xiangjie Kong(孔祥杰)[1,‡], and Giovanni Pau[3]

[1] *College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou* 310014, *China*
[2] *Hangzhou Dianzi University ITMO Joint Institute, Hangzhou Dianzi University, Hangzhou* 310018, *China*
[3] *Faculty of Engineering and Architecture, Kore University of Enna, Italy*

(Received 18 November 2024; revised manuscript received 21 February 2025; accepted manuscript online 11 March 2025)

The task of molecule generation guided by specific text descriptions has been proposed to generate molecules that match given text inputs. Mainstream methods typically use simplified molecular input line entry system (SMILES) to represent molecules and rely on diffusion models or autoregressive structures for modeling. However, the one-to-many mapping diversity when using SMILES to represent molecules causes existing methods to require complex model architectures and larger training datasets to improve performance, which affects the efficiency of model training and generation. In this paper, we propose a text-guided diverse-expression diffusion (TGDD) model for molecule generation. TGDD combines both SMILES and self-referencing embedded strings (SELFIES) into a novel diverse-expression molecular representation, enabling precise molecule mapping based on natural language. By leveraging this diverse-expression representation, TGDD simplifies the segmented diffusion generation process, achieving faster training and reduced memory consumption, while also exhibiting stronger alignment with natural language. TGDD outperforms both TGM-LDM and the autoregressive model MolT5-Base on most evaluation metrics.

Keywords: molecule generation, diffusion model, AI for science

PACS: 07.05.Kf, 07.05.Mh, 07.05.Tp

DOI: 10.1088/1674-1056/adbedd **CSTR:** 32038.14.CPB.adbedd

1. Introduction

Molecular generation plays a critical role in fields such as chemistry, drug development, and materials science, especially in designing molecules that meet specific functional or structural requirements.[1] In recent years, with the rapid development of artificial intelligence, text-guided molecular generation tasks have gradually attracted widespread attention. This task aims to generate molecules that match given textual descriptions, integrating the advantages of natural language processing and molecular modeling, and providing a new avenue for precise molecular design.

With the rise of deep learning, generative models have become important tools for molecular generation. Early generative methods employed sequence-based molecular representations, such as simplified molecular input line entry system (SMILES).[2–4] Models like variational autoencoders (VAEs)[5] and recurrent neural networks (RNNs)[6] were used to learn the distribution of existing molecules and generate new molecules by sampling from the latent space.[7–9]

Currently, mainstream approaches mainly rely on two types of models: diffusion models[10] and autoregressive structures. These methods commonly utilize SMILES to represent molecules.[11–13] As a concise and easy-to-understand chemical structure representation, SMILES is widely used in molecular generation tasks due to its simplicity and operability. However, although each molecule is unique in its physical structure and chemical properties, the same molecule can be represented by multiple different SMILES strings, introducing ambiguity through this one-to-many mapping and increasing the learning difficulty for generation models, as shown in Fig. 1. In addition, minor modifications to SMILES strings can lead to invalid chemical structures, thereby increasing the difficulty of generating valid molecules. Especially when guided only by text descriptions, models often struggle to accurately capture textual information and generate molecules with the correct structure and function.

In response to this, existing methods use complex model architectures and increase training samples to improve accuracy, but these approaches affect the training and generation efficiency of the models. For example, MolT5[14] uses two monolingual corpora for pre-training. For the text modality, MolT5 selects the "Colossal Clean Crawled Corpus" (C4),[32] containing over 700 GB of English text, as the pre-training dataset, while for the molecular modality, it directly uses 100 million SMILES strings from Chemformer.[15] The vast amount of training data makes MolT5's pre-training highly de-

[†] These authors contributed equally to this work.
[‡] Corresponding author. E-mail: xjkong@ieee.org

© 2025 Chinese Physical Society and IOP Publishing Ltd. All rights, including for text and data mining, AI training, and similar technologies, are reserved.

manding in terms of computational power. TGM-DLM[11] is currently the model with the best overall performance in the field. Although TGM-DLM does not use additional datasets for pre-training, to compensate for the lack of validity in molecule generation, TGM-DLM adopts a two-stage generation strategy: the first stage focuses on generating molecules, and the second stage focuses on improving them to conform to SMILES syntax. This approach increases the generation validity of the model but clearly adds to training costs and slows down generation speed.

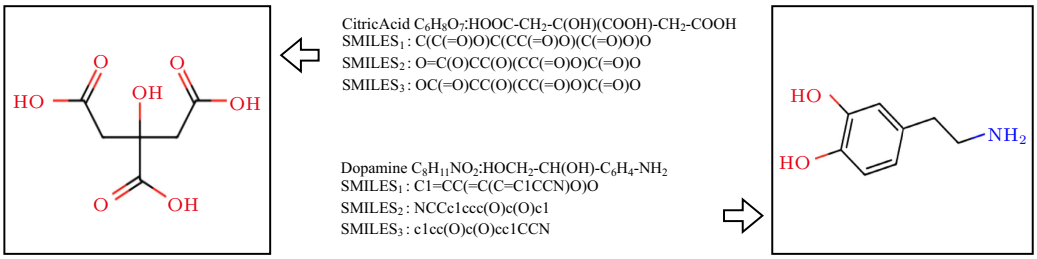


Fig. 1. Multiple SMILES correspond to one molecule.

To address the issue of model complexity caused by the uncertainty of single molecular representations, this paper proposes a text-guided diverse-expression diffusion model for molecule generation (TGDD). TGDD uses not only SMILES but also self-referencing embedded strings (SELFIES) to represent molecules. This method employs context-free grammar to encode molecular graphs into string formats, ensuring that any SELFIES string corresponds to a valid molecule, thereby significantly reducing the probability of generating invalid molecules. TGDD combines SMILES and SELFIES expressions to form a new joint molecular representation and utilizes text descriptions for guidance, iteratively optimizing token embeddings to achieve precise molecular mapping. Using this joint molecular representation, we can achieve excellent results with only a simple diffusion model.

In summary, the contributions of this paper can be summarized as follows. We propose a data augmentation method that converts the SMILES representation of molecules into SELFIES format, enriching the dataset and facilitating future joint learning efforts for diverse molecular representations. We propose a text-guided molecule generation diffusion language model based on diverse-expression (TGDD). TGDD innovatively performs joint learning of different representations of the same molecule, integrating SMILES and SELFIES. It achieves excellent performance with minimal memory usage and offers three output modes to cater to different application scenarios. Large-scale experiments on the dataset demonstrate that our model has achieved optimal results in understanding the correlation between natural language and molecular expressions.

2. Related work

2.1. One-dimensional molecular representation

One-dimensional molecular string representations, being the expression methods closest to natural language, serve as the cornerstone of high-dimensional molecular expressions and were the earliest molecular representation forms used in machine learning. Since the introduction of the SMILES,[2–4] its concise form and ease of parsing have been favored by researchers. However, the inherent syntactic deficiencies of SMILES often lead to lower validity in generation models based on SMILES. Designing more scientific molecular representation forms has thus become a hot research topic.

To accurately describe the stereochemical information of molecules, Isomeric SMILES[16] adds encoding for chiral centers and isotopic information, enabling precise representation of molecular stereostructures and isotopic labels.

The International Union of Pure and Applied Chemistry (IUPAC) and the National Institute of Standards and Technology (NIST) jointly developed the International Chemical Identifier (InChI).[17] InChI adopts a hierarchical string representation, providing standardized and normalized molecular descriptions, facilitating the sharing and retrieval of chemical information. However, InChI strings are lengthy and lack readability, making them unsuitable for direct use in generation models.

DeepSMILES[18] significantly improves validity by using single ring closure digits and adopting postfix notation to avoid the use of paired parentheses, making it suitable for machine learning of chemical structures.

SELFIES,[19] based on strict grammatical rules, ensures that any valid SELFIES string corresponds to a valid molecular structure. This greatly enhances the ability of generation models to produce valid molecules, addressing the inherent flaws of SMILES in the generation process.

The t-SMILES[20] performs a breadth-first search on a full binary tree formed from a fragmented molecular graph to generate SMILES-like strings for describing molecules. On labeled low-resource datasets, t-SMILES achieves higher novelty scores while maintaining reasonable similarity.

2.2. Diffusion models in language generation

Diffusion models, originally developed for generative modeling in the continuous domain, particularly for images, have recently been extended to handle discrete data such as natural language. These models, like denoising diffusion probabilistic models (DDPM) introduced by Ho *et al.*,[10] work by iteratively adding and then removing noise to data, effectively learning the data distribution through a diffusion process.

Applying diffusion models to language generation poses unique challenges due to the discrete nature of text.[21] Unlike images, where pixel values are continuous, words and tokens in language are discrete symbols. To bridge this gap, researchers have proposed methods to adapt diffusion models for discrete sequences.

One approach involves mapping discrete text data into continuous representations. For example, Diffusion-LM by Li *et al.*[22] employs a continuous diffusion process over embedded representations of text. The model integrates denoising diffusion techniques with transformer-based architectures,[23] enabling it to generate coherent and diverse text by learning to reverse the noising process applied to embeddings.

Another line of work focuses on defining diffusion processes directly over discrete variables. Jacob Austin *et al.*[24] introduced methods for discrete diffusion using categorical distributions, allowing the diffusion model to operate on discrete data without continuous relaxation. This approach has been applied to language tasks by modeling the diffusion over vocabulary tokens, facilitating applications in text generation and machine translation.

In the context of molecular generation, diffusion models have shown promise in generating complex molecular structures.[25,26] The parallels between molecular graphs and linguistic structures suggest that advances in diffusion models for language can inform and enhance molecular generative models, as both domains deal with discrete combinatorial spaces.

The application of diffusion models in language generation is an emerging and active area of research. Continued exploration is expected to lead to improvements in natural language processing tasks, including text completion, translation, and dialogue systems. As these models are further refined, they hold the potential to advance the state-of-the-art in generative language modeling.

3. Task formulation

The goal of text-guided molecular generation is to produce molecules that match the text description. Specifically, given a description text C of length L, where $C = [w_0, w_1, \ldots, w_L]$ represents all the words in the text and w_i indicates the i-th word, model M takes the text C as input. C guides the model to generate the target molecular representation X_0, as shown below:

$$X_0 = M(C). \tag{1}$$

4. Method

4.1. Data augmentation

Deep learning models typically output vectors, and establishing the relationship between these output vectors and the generation targets is a fundamental challenge for any generative model. TGDD addresses this by constructing a vocabulary that maps discrete numbers to molecular expressions. Subsequently, an embedding layer is employed to transform these discrete numbers into high-dimensional continuous vectors, as shown in Fig. 2.

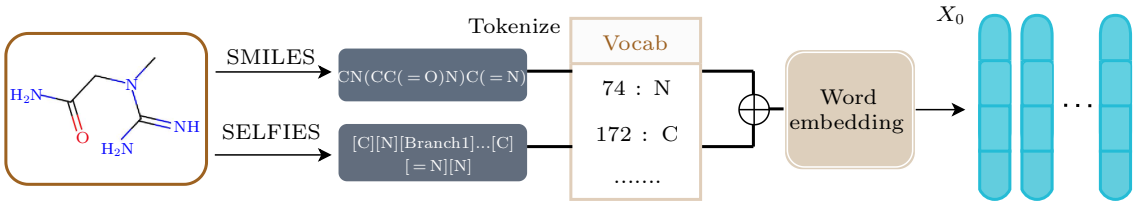


Fig. 2. The architecture of data augmentation.

4.1.1. Tokenizer

To maintain the integrity of atomic groups, we follow the standards of TGM-DLM by treating each character group with complete chemical significance as a single token. For example, silane is tokenized as [SiH4] rather than splitting it into (S, i, H, 4). Each unique token is numbered and stored as a vocabulary V. For a given molecule M, it can be initially serialized by traversing the vocabulary V as $M_0 = [S_1, S_2, \ldots, S_L]$, where S_i is an element in V representing the i-th token, and L is the length of the tokens after initial serialization. To standardize input and clarify valid information, M_0 is further padded into a sequence of length N as $M_1 = [[\text{SOS}], S_1, S_2, \ldots, S_L, [\text{EOS}], \ldots, S_N]$. Here, [SOS] and [EOS] are artificially added start and end tokens, and N is the specified uniform sequence length, with any shortfall filled with specific blank tokens [PAD], and any excess truncated, as shown in Fig. 3.

Original SMILES string:
CSCC[C@H](C(=O)[O-])N

SMILES tokenization:
[SOS], C, S, C, C, [C@H], (, C, (, =, O,), [O-],), N, [EOS], [PAD], ..., [PAD]

Corresponding SELFIES string:
[C][S][C][C][C@H1][Branch1][=Branch1][C][=Branch1][C][=O][O-1][N]

SELFIES tokenization:
[SOS], [C], [S], [C], [C], [C@H1], [Branch1], [=Branch1], [C], [=Branch1], [C], [=O], [O-], [N], [EOS], [PAD], ..., [PAD]

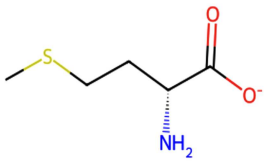


Fig. 3. Example of SMILES and SELFIES tokenization.

In the process of tokenizing the SMILES and SELFIES representations of each molecule, we systematically record each unique token to construct a joint vocabulary. In this process, we adhere to the genuine meaning of chemical symbols, rather than treat individual characters as the smallest tokens. We make efforts to retain the chemical significance of each token; for instance, tokens like "[C@H]" are preserved in their entirety to ensure the accurate representation of their structure and function.

Furthermore, we consolidate the common parts of different molecular representations in the middle of the vocabulary, while the unique tokens of different molecular representations are recorded at the beginning and the end of the vocabulary. Following this series of processes, we ultimately constructed a joint vocabulary consisting of 414 tokens, including 254 SMILES tokens, 314 SELFIES tokens, and 154 overlapping tokens.

Ultimately, the sequences of the two molecular expressions of the same molecule, after tokenization, will be combined in order to form a sequence M_2 with a length of 512.

4.1.2. Embedding

Each token S_i in the serialized molecular representation M_2 is mapped into a high-dimensional space through an embedding, forming a sequence of embedding vectors $\mathrm{Emb}(M_2) = [\mathrm{Emb}(s_0), \mathrm{Emb}(s_1), \ldots, \mathrm{Emb}(s_n)] \in \mathbb{R}^{d\times n}$, where d represents the embedding dimension and n represents the sequence length. Similarly, the input descriptive text is mapped to a high-dimensional space $\mathrm{Emb}(C)$ through SciBert.

4.2. Molecular generation

In the field of text-guided molecular generation, diffusion models learn how to gradually denoise a complete Gaussian noise into a target molecular representation. The key steps involve two processes: forward noise addition and reverse denoising. The forward noise addition process incrementally adds noise to the vectorized molecular representation until it becomes the complete Gaussian noise. In the reverse denoising process, the model, guided by text, gradually denoises the complete Gaussian noise into the corresponding target molecular representation vector. The noise addition process is fixed, while the denoising process is learnable. The overall process is shown in Fig. 4.

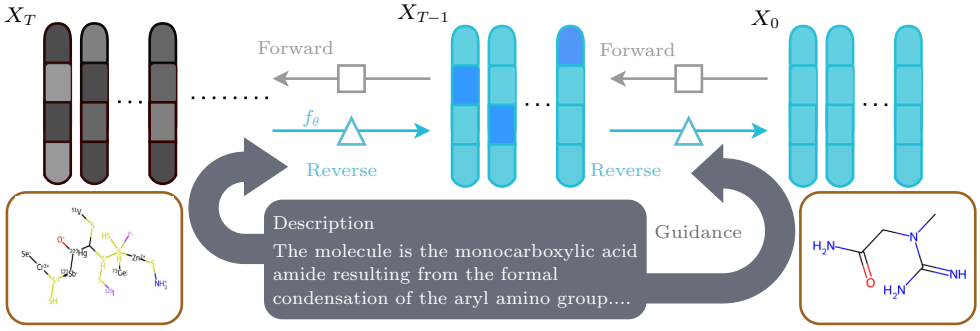


Fig. 4. The architecture of molecular generation. The forward process is the noise addition process, while the reverse process is the denoising process. The depth of the vector color represents the degree of noise influence.

4.2.1. Forward

The forward process first requires a random initial matrix X_0, which is sampled from a Gaussian distribution centered at $\mathrm{Emb}(M_2)$: $x_0 \sim \mathcal{N}(\mathrm{Emb}(M_2), \sigma_0 \boldsymbol{I})$. During the forward process, Gaussian noise is gradually added to X_0, and the purity of the added noise is controlled by a predefined constant $\beta_t \in [0,1]$, until X_0 becomes pure Gaussian noise X_T, with $x_T \sim \mathcal{N}(0, \boldsymbol{I})$. The formula for adding noise at each step is as follows:

$$q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) = \mathcal{N}(\boldsymbol{x}_t; \sqrt{1-\beta_t}\boldsymbol{x}_{t-1}, \beta_t \boldsymbol{I}). \tag{2}$$

4.2.2. Reverse

The traditional reverse process starts from the final result of the forward noising process X_T, and restores X_{T-1} based on X_T, gradually restoring back to X_0. However, the step-by-step restoration cannot guarantee the completeness of the process. Therefore, TGDD trains a neural network f_θ, which does not focus on the intermediate process from X_T to X_0, but instead focuses on how to restore X_0 from X_T. The denoising process is expressed as follows:

$$p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t) = \mathcal{N}\Bigg(\boldsymbol{x}_{t-1}; \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t} f_\theta(\boldsymbol{x}_t, t) + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\boldsymbol{x}_t, \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t \boldsymbol{I}\Bigg), \quad (3)$$

where $\alpha_t = \prod_{s=0}^{t}(1-\beta_s)$. By sequentially applying Eq. (2) starting from pure noise X_T, we can alternately sample X_{t-1}, and eventually obtain X_0.

The function f_θ is driven by a transformer model, which consists of T layers. Each layer's cross-attention block introduces the embedding vector $\mathrm{Emb}(C)$ from the text description. To better enrich the weights of each word segment, $\mathrm{Emb}(M_2)$ will undergo further information extraction through a multilayer perceptron (MLP):

$$\begin{aligned} &Q = W_Q^{(i)} z_t^{(i)}, \quad K = W_K^{(i)}\mathrm{MLP}(\mathrm{Emb}(C)), \\ &V = W_V^{(i)}\mathrm{MLP}(\mathrm{Emb}(C)), \\ &\mathrm{Attention}(Q,K,V) = V \cdot \mathrm{softmax}\left(\frac{K^\top Q}{\sqrt{d_2}}\right), \end{aligned} \quad (4)$$

where $W_{Q/K/V}^{(i)} \in \mathbb{R}^{d_1 \times d_1}$ represents the learnable parameters of the cross-attention block in the i-th layer.

To incorporate the temporal information of the diffusion step t, we mimic the method of positional embedding by expanding t into a vector $M(t)$ through an embedding layer. $M(t)$ and the positional embedding are equally considered in the input $I_t^{(0)}$ of the first layer of the transformer:

$$z_t^{(0)} = W_{\mathrm{in}} x_t + \mathrm{PosEmb} + M(t), \quad (5)$$

where PosEmb denotes the positional embedding. $W_{\mathrm{in}} \in \mathbb{R}^{d_2 \times d}$, $z_t^{(0)} \in \mathbb{R}^{d_2 \times n}$, and d_2 is the dimension of the transformer.

4.3. Output

In the training phase, TGDD accepts natural language descriptions of molecular properties as input and uses diverse-expression to represent the target molecule. The goal is to output a diverse-expression that is consistent with the target molecule. It is worth mentioning that we can easily parse the target molecule's SMILES and SELFIES expressions from the diverse-expression. Furthermore, SMILES and SELFIES can be converted into each other losslessly using the Python library provided by Krenn *et al*. Therefore, in the evaluation phase, TGDD's output can switch seamlessly between the three modes without any loss.

1. **TGDD-SMILES** The SMILES is parsed from the diverse-expression generated by the model and is included in the evaluation, where the evaluative entities are the parsed SMILES and the target molecule's SMILES.

2. **TGDD-SELFIES** The SELFIES is parsed from the diverse-expression generated by the model and is included in the evaluation, where the evaluative entities are the parsed SELFIES and the target molecule's SELFIES.

3. **TGDD-DIVERSE** The evaluation process prioritizes parsing SMILES from the diverse-expression generated by the model. If the parsed SMILES is valid, the evaluation follows the same procedure as TGDD-SMILES. If the parsed SMILES is invalid, the corresponding molecule for the invalid SMILES will have its SELFIES parsed from the diverse-expression generated by the model, and the evaluation will follow the same process as TGDD-SELFIES.

5. Experiment

5.1. Dataset and experimental settings

We use the publicly available dataset ChEBI-20,[14] which contains 33010 pairs of molecular descriptions. The dataset is split into training, validation, and test sets in a ratio of 8:1:1.

It is worth mentioning that since ChEBI-20 only includes molecular IDs, molecular SMILES, and text descriptions, and does not include SELFIES representations of the molecules, we enriched ChEBI-20 by adding the SELFIES representations of the molecules to the dataset using the Python library provided by Krenn *et al*.[19]

The maximum length of the serialized diverse-expression was limited to 512. Molecules exceeding this length were filtered out, which accounted for approximately 2% of the entire dataset. The joint vocabulary contained 417 tokens, including 3 special tokens ([SOS], [EOS], [PAD]), 254 SMILES tokens, and 314 SELFIES tokens. Notably, the 154 overlapping tokens represent the intersection of the two molecule representations after serialization.

Under our hardware conditions (AMD EPYC 7742 (256) @ 2.250 GHz CPU and one NVIDIA A100 GPU), training the model for 200000 steps takes 38 hours, and generating a single molecule only takes 0.7 s.

5.2. Metrics

To verify the effectiveness of the model, we use the following metrics to evaluate its performance:

BLEU[27] and **Levenshtein distance**:[28] Used to evaluate the similarity and differences between the generated SMILES strings and the labels.

MACCS FTS,[29] **RDK FTS**,[30] and **Morgan FTS**:[31] Evaluated by computing the average Tanimoto similarity between the fingerprints of the generated molecules and the label molecules.

Exact score and **validity**: The percentage of generated molecules that exactly match the label molecules, and the percentage of generated strings that are valid.

5.3. Baseline model

To better test the various capabilities of the model, we selected different types of models as comparison baselines, including large models, state-of-the-art models from the past, and popular models. The training and evaluation process for these models follows the setup where the model takes natural language descriptions of molecular properties as input and outputs a molecular representation in SMILES format. The goal is for the output SMILES to match exactly the target molecule's SMILES.

RNN:[6] A recurrent neural network with a four-layer GRU core and a bidirectional encoder.

Transformer:[23] A transformer trained on ChEBI-20, consisting of six encoder and decoder layers.

T5:[32] Models fine-tuned on two T5 public checkpoints: small and base.

MolT5:[14] The model was built upon a pre-trained T5 model and underwent additional pre-training on a combined dataset of C4 and ZINC-15 to enhance its knowledge in the domains of molecules and natural language. It was then fine-tuned on the ChEBI-20 dataset.

SciBert:[33] A pre-trained language model based on BERT,[34] which improves its performance on downstream scientific natural language processing tasks by undergoing unsupervised pre-training on a large corpus of scientific publications.

TGM-DLM:[11] It is an innovative approach that utilizes diffusion models to overcome the limitations of autoregressive methods. It updates the token embeddings within SMILES strings progressively through a two-phase diffusion generation process.

6. Experimental results

6.1. Performance comparison

Currently, most molecular generation models utilize SMILES as the molecular representation, and the baseline models listed in Table 1 are all based on SMILES. Therefore, the evaluation of the model is divided into two parts: one part focuses on analyzing the evaluation metrics of TGDD-SMILES in comparison with the baseline models, while the other part explores the impact of diverse-expression representation on the model's performance. This dual evaluation approach allows for a more comprehensive demonstration of TGDD's advantages, particularly in terms of enhancing diverse representations and generation capabilities.

Table 1. Performance metrics for different models.

Model	BLEU ↑	Exact ↑	Levenshtein ↓	MACCS FTS ↑	RDK FTS ↑	Validity ↑	Morgan FTS ↑
TGM-DLM	0.828	0.242	16.897	0.874	0.771	0.789	0.722
RNN[a)]	0.652	0.005	38.090	0.591	0.400	0.542	0.362
Transformer[b)]	0.499	0	57.660	0.480	0.320	0.906	0.217
T5-Small[c)]	0.741	0.064	27.703	0.704	0.578	0.608	0.525
MolT5-Small[d)]	0.755	0.079	25.988	0.703	0.568	0.721	0.517
T5-Base[c)]	0.762	0.069	24.950	0.731	0.605	0.660	0.545
MolT5-Base[d)]	0.769	0.081	24.458	0.721	0.588	0.772	0.529
SciBert[e)]	0.459	0.005	24.46	0.721	0.588	0.915	0.529
TGDD-SMILES	0.843	0.139	16.256	0.886	0.793	0.484	0.766

[a)]Ref. [6], [b)]Ref. [23], [c)]Ref. [32], [d)]Ref. [14], [e)]Ref. [33].

As shown in Table 1, TGDD-SMILES has achieved state-of-the-art performance across multiple evaluation metrics. Although its "Exact" is lower than that of TGM-DLM , it still far surpasses other baseline models, including large models. However, its validity is below the average level. Figure 5 visually presents the output molecules of TGDD-SMILES and TGM-DLM. Clearly, TGDD demonstrates exceptional potential in understanding the connection between natural language and molecular expressions. This means that TGDD can better comprehend the molecular information referred to in descriptive texts and construct the target molecules from scratch based on this natural language information. This capability highlights TGDD's advantages in molecular generation tasks. However, the incorporation of the SELFIES increases the model's requirements for learning the rules of different molecular representation forms, and there's still room for improvement for TGDD in this aspect.

6.2. Computation cost

The model size and generation speed are important metrics for evaluating the performance of generative models. We selected the three best-performing models from the baseline models and measured their parameter counts as a basis for assessing model size. At the same time, we compared them with

the previously best-performing model, TGM-DLM, to evaluate the average time taken to generate a single molecule on the same device.

Table 2. Comparison of parameters and generation time for different models.

Model	TGM-DLM	MolT5-Base	T5-Base	TGDD
Parameters	180M	220M	220M	50M
Generation time[a)]	1.2s	–	–	0.7s

[a)]The average time it takes to generate a molecule from its description under the same device conditions.

It is worth noting that Table 2 shows that the number of parameters for TGDD-SMILES is less than one-third of that for TGM-DLM, significantly reducing memory consumption. Tests have shown that under the same hardware conditions, TGDD-SMILES generates a molecule in 0.7 s on average, compared to TGM-DLM's average of 1.2 seconds per molecule, which considerably lowers the time cost.

6.3. Visualization

To visually demonstrate the differences between the outputs of TGDD and TGM-DLM and their targets, we visualized the results of TGDD-SMILES and TGM-DLM to showcase the models' strong capabilities in understanding descriptive texts about molecules.

Figure 5 visually presents the output molecules of TGDD-SMILES and TGM-DLM. Clearly, TGDD demonstrates exceptional potential in understanding the connection between natural language and molecular expressions. This means that TGDD can better comprehend the molecular information referred to in descriptive texts and construct the target molecules from scratch based on this natural language information. This capability highlights TGDD's advantages in molecular generation tasks.

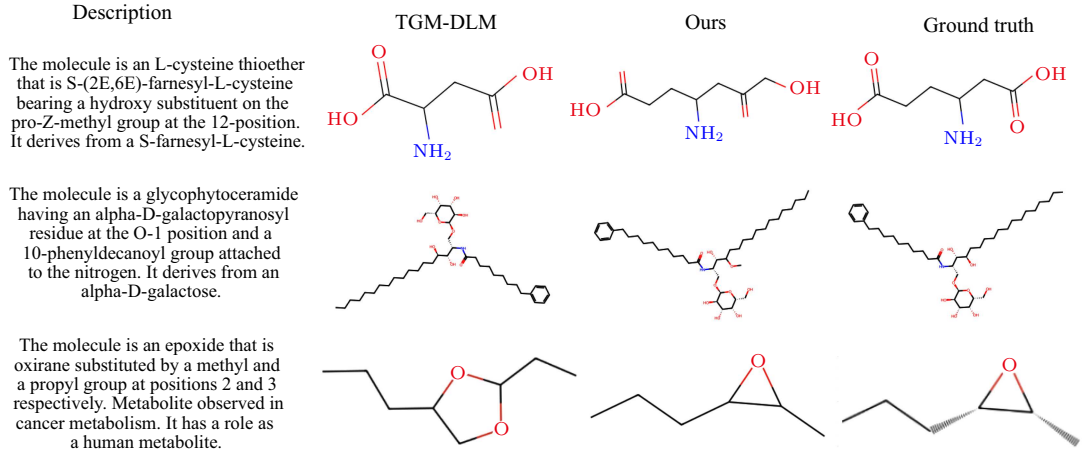


Fig. 5. Output molecule visualization comparison.

7. Ablation study

To verify the crucial role of diverse-expression in the model, we trained the models using only one type of molecular encoding with the same model parameters for TGDD-SMILES and TGDD-SELFIES (except for the maximum sequence length) and then discussed the final results.

In Table 3, a comparison between SMILES-Base and TGDD-SMILES, as well as between SELFIES-Base and TGDD-SELFIES, reveals that the integration of diverse molecular representations enhances the model's comprehension of the relationship between natural language descriptions and molecules. This improvement is evident in the metrics, including BLEU, MACCS FTS, RDK FTS, and Levenshtein. However, the additional molecular representation information inevitably increases the pressure on the model to distinguish between different representation methods, making the model's accuracy less than ideal. Particularly for TGDD-SELFIES, the model seems to have been influenced by SMILES syntax when understanding SELFIES syntax, leading to a decline in metrics. This aptly explains why the "Exact" metric of TGDD-SELFIES is not as good as that of SELFIES-base, but still higher than that of TGDD-SMILES.

Table 3. Comparison of single-molecule expressions and diverse-expression performance.

Model	BLEU ↑	Exact ↑	Levenshtein ↓	MACCS FTS ↑	RDK FTS ↑	Validity ↑	Morgan FTS ↑
SMILES-Base[a)]	0.809	0.15	18.552	0.877	0.791	0.487	0.759
TGDD-SMILES	0.843	0.139	16.256	0.886	0.793	0.484	0.766
SELFIES-Base[b)]	0.672	0.238	23.297	0.737	0.621	1	0.568
TGDD-SELFIES	0.681	0.156	22.304	0.75	0.637	0.999	0.558
TGDD-DIVERSE	0.74	0.152	20.455	0.793	0.699	0.998	0.606

[a)]Molecules using only SMILES, [b)]molecules using only SELFIES.

As for TGDD-DIVERSE, it seems more like a compromise between TGDD-SMILES and TGDD-SELFIES, combining both validity and natural language affinity.

8. Conclusion

In this paper, we introduce a text-guided diverse-expression diffusion model (TGDD) for molecule generation. TGDD jointly learns two molecular representations, SMILES and SELFIES, using a unified vocabulary to overcome the inherent limitations of single-representation models. This enables the model to better understand the connection between natural language descriptions and molecular structures. TGDD's three prediction modes allow the model to perform effectively in various scenarios, expanding its applicability. Experimental results demonstrate that TGDD significantly improves performance across multiple evaluation metrics.

Our research opens new avenues in the field of molecule generation by demonstrating the benefits of integrating diverse molecular expressions. Future work will focus on further expanding the range of molecular representations incorporated into the model, including graph-based structures and three-dimensional spatial coordinates. Additionally, we aim to apply TGDD to practical domains such as drug discovery and materials science, aiding researchers in the rapid generation of molecules that meet specific requirements and advancing innovation in related fields.

Program availability

The source code is openly available in the Science Data Bank at https://www.doi.org/10.57760/sciencedb.21097 and on Github at https://github.com/YHanJG/TGDD.

Acknowledgements

Project supported in part by the National Natural Science Foundation of China (Grant Nos. 62476247 and 62072409), the "Pioneer" and "Leading Goose" R&D Program of Zhejiang (Grant No. 2024C01214), and the Zhejiang Provincial Natural Science Foundation (Grant No. LR21F020003).

We thank Haisong Gong, the author of TGM-DLM, for providing the model information and improvement suggestions.

References

[1] Tang Y, Yang Z, Yao Y, Zhou Y, Tan Y, Wang Z, Pan T, Xiong R, Sun J and Wei G 2024 *Chin. Phys. B* **33** 030701
[2] Weininger D 1988 *Journal of Chemical Information and Computer Sciences* **28** 31
[3] Weininger D, Weininger A and Weininger J L 1989 *Journal of Chemical Information and Computer Sciences* **29** 97
[4] Weininger D 1990 *Journal of Chemical Information and Computer Sciences* **30** 237
[5] Kingma D P 2013 arXiv:1312.6114 [stat.ML]
[6] Cho K, Van Merrienboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H and Bengio Y 2014 arXiv:1406.1078 [cs.CL]
[7] 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 *ACS Central Science* **4** 268
[8] Olivecrona M, Blaschke T, Engkvist O and Chen H 2017 *Journal of Cheminformatics* **9** 48
[9] Popova M, Isayev O and Tropsha A 2018 *Science Advances* **4** eaap7885
[10] Ho J, Jain A and Abbeel P 2020 *Advances in Neural Information Processing Systems* **33** 6840
[11] Gong H, Liu Q, Wu S and Wang L 2024 *Proceedings of the AAAI Conference on Artificial Intelligence*,February 20–27, 2024, Vancouver, Canada, p. 109
[12] Irwin R, Dimitriadis S, He J and Bjerrum E J 2022 *Machine Learning: Science and Technology* **3** 015022
[13] Wang S, Guo Y, Wang Y, Sun H and Huang J 2019 *Proceedings of the 10th ACM international conference on bioinformatics, computational biology and health informatics*, September 7-10, 2019, Niagara Falls, NY, USA, p. 429
[14] Edwards C, Lai T, Ros K, Honke G, Cho K and Ji H 2022 *2022 Conference on Empirical Methods in Natural Language Processing* p. 375
[15] Irwin R, Dimitriadis S, He J and Bjerrum E J 2022 *Machine Learning: Science and Technology* **3** 015022
[16] Weininger D, Weininger A and Weininger J L 1989 *Journal of Chemical Information and Computer Sciences* **29** 97
[17] Heller S R, McNaught A, Pletnev I, Stein S and Tchekhovskoi D 2015 *Journal of Cheminformatics* **7** 23
[18] O'Boyle N and Dalke A 2018 chemrxiv.7097960.v1
[19] Krenn M, Häse F, Nigam A, Friederich P and Aspuru-Guzik A 2020 *Machine Learning: Science and Technology* **1** 045024
[20] Wu J N, Wang T, Chen Y, Tang L J, Wu H L and Yu R Q 2024 *Nat. Commun.* **15** 4993
[21] Yi Q, Chen X, Zhang C, Zhou Z, Zhu L and Kong X 2024 *PeerJ Computer Science* **10** e1905
[22] Li X, Thickstun J, Gulrajani I, Liang P S and Hashimoto T B 2022 *Advances in Neural Information Processing Systems* **35** 4328
[23] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez A N, Kaiser L and Polosukhin I 2017 *Proceedings of the* 31*st International Conference on Neural Information Processing Systems*, December 4–9, 2017, Long Beach, California, USA, p. 6000
[24] Austin J, Johnson D D, Ho J, Tarlow D and Van Den Berg R 2021 *Advances in Neural Information Processing Systems* **34** 17981
[25] Hoogeboom E, Satorras V G, Vignac C and Welling M 2022 *International conference on machine learning*, July 17–23, 2022 ,Baltimore, Maryland, USA, p. 8867
[26] Xu M, Yu L, Song Y, Shi C, Ermon S and Tang J 2022 arXiv:2203.02923 [cs.LG]
[27] Papinesi K 2002 *Proc. 40th Actual Meeting of the Association for Computational Linguistics (ACL)*, July 7–12, 2002, Philadelphia, Pennsylvania, p. 311
[28] Miller F P, Vandome A F and McBrewster J 2009 *Levenshtein Distance: Information theory, Computer science, String (computer science), String metric, Damerau Levenshtein distance, Spell checker, Hamming distance* (Alpha Press) 68
[29] Durant J L, Leland B A, Henry D R and Nourse J G 2002 *Journal of Chemical Information and Computer Sciences* **42** 1273
[30] Schneider N, Sayle R A and Landrum G A 2015 *Journal of Chemical Information and Modeling* **55** 2111
[31] Rogers D and Hahn M 2010 *Journal of Chemical Information and Modeling* **50** 742
[32] Raffel C, Shazeer N, Roberts A, Lee K, Narang S, Matena M, Zhou Y, Li W and Liu P J 2020 *Journal of Machine Learning Research* **21** 1
[33] Beltagy I, Lo K and Cohan A 2019 *Proceedings of the* 2019 *Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, November, 2019, Hong Kong, China, p. 3615
[34] Devlin J, Chang M W, Lee K and Toutanova K 2019 *Proceedings of the* 2019 *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Volume 1 (Long and Short Papers), June 2–7, 2019, Minneapolis, Minnesota, p. 4171