BIG DATA MINING AND ANALYTICS
ISSN 2096-0654 10/15 pp142–155
DOI: 10.26599/BDMA.2023.9020009
Volume 7, Number 1, March 2024

Molecular Generation and Optimization of Molecular Properties Using a Transformer Model

Zhongyin Xu, Xiujuan Lei*, Mei Ma, and Yi Pan*

Abstract: Generating novel molecules to satisfy specific properties is a challenging task in modern drug discovery, which requires the optimization of a specific objective based on satisfying chemical rules. Herein, we aim to optimize the properties of a specific molecule to satisfy the specific properties of the generated molecule. The Matched Molecular Pairs (MMPs), which contain the source and target molecules, are used herein, and logD and solubility are selected as the optimization properties. The main innovative work lies in the calculation related to a specific transformer from the perspective of a matrix dimension. Threshold intervals and state changes are then used to encode logD and solubility for subsequent tests. During the experiments, we screen the data based on the proportion of heavy atoms to all atoms in the groups and select 12 365, 1503, and 1570 MMPs as the training, validation, and test sets, respectively. Transformer models are compared with the baseline models with respect to their abilities to generate molecules with specific properties. Results show that the transformer model can accurately optimize the source molecules to satisfy specific properties.

Key words: molecular optimization; transformer; Matched Molecular Pairs (MMPs); logD; solubility

1 Introduction

Generating new molecules with desirable properties is a substantial and challenging task in drug discovery. A drug requires balancing of multiple properties, including Absorption, Distribution, Metabolism, Elimination, and Toxicity (ADMET), physical and chemical properties. Finding molecules with specific features in a vast chemical environment, where the total number of potential drug candidates is presumed to be 10^{23}–10^{60}[1], is difficult.

• Zhongyin Xu, Xiujuan Lei, and Mei Ma are with School of Computer Science, Shaanxi Normal University, Xi'an 710119, China. E-mail: 20212687@snnu.edu.cn; xjlei@snnu.edu.cn; mm1016@qhnu.edu.cn.

• Yi Pan is with Faculty of Computer Science and Control Engineering, Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China. E-mail: yipan@siat.ac.cn.

* To whom correspondence should be addressed.

Manuscript received: 2023-01-06; revised: 2023-03-27; accepted: 2023-05-04

Molecular optimization improves the properties of initial molecules. Controlling some properties of initial drug molecules to create directional changes is crucial in increasing drug efficiency. Four aspects are typically involved in molecular generation. (1) Database selection involves choosing appropriate molecular data according to the task requirements. (2) Molecular representation is the translation of the selected molecular data into the input form that can be understood by the computer. (3) Selection of a suitable generative model is closely related to the chosen representation method and the performance of the model. (4) Evaluation metrics are used to assess the generated new molecules with appropriate indexes.

A series of molecular representations have been devised over the past few years. The *de novo* molecular design has two common representations. The first is a sequence-based representation. Two current examples of one-dimensional linear representations are SMILES and International CHemical Identifier (InCHI)[2]. Some improved molecular representations based on

SMILES[3, 4] have been recently proposed. The second representation is a graph-based representation, where each molecule can be represented as an undirected graph, with nodes and edges representing atoms and chemical bonds, respectively. Figure 1 shows a simple example of this representation.

Different molecular generative models, such as Recurrent Neural Networks (RNNs)[5–7], Variational AutoEncoders (VAEs)[8–14], Generative Adversarial Networks (GANs)[15–18], and flow-based models[19–21], have been recently used for molecular generation. Some novel approaches have emerged from these highly mature models. MGCVAE[22] can effectively generate drug-like molecules with target properties. Li et al.[23] generated molecules based on geometric convolution with specific protein constraints. GFVAE[24] inherited the advantages of VAE and flow-based methods. Luo et al.[25] proposed an autoregressive sampling scheme to generate 3D molecules. MolGPT[26] took masked self-attention[27] for generating drug-like molecules. Langevin et al.[28] introduced a new algorithm named scaffold-constrained molecular generation to perform scaffold-constrained molecular design. Zhang and Chen[29] proposed a novel molecular deep generative model that adopts an RNN architecture coupled with a ligand-protein interaction fingerprint as constraints. The sequence-to-sequence model with attention mechanism and the transformer model are employed in Ref. [30] to generate molecules with desirable properties. A further study[31] provides a general methodology for highly general structural modifications beyond Matched Molecular Pairs (MMPs).

The metrics of evaluating generative models of

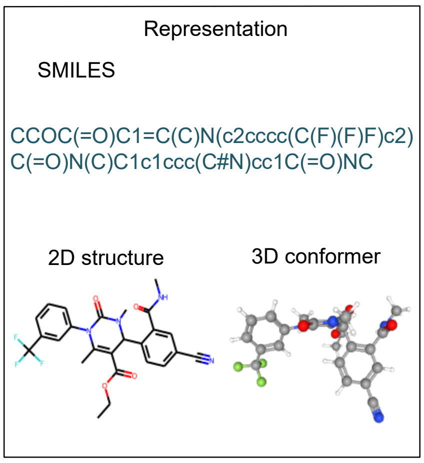

Representation
SMILES
CCOC(=O)C1=C(C)N(c2cccc(C(F)(F)F)c2)C(=O)N(C)C1c1ccc(C#N)cc1C(=O)NC
2D structure
3D conformer

Fig. 1 Example of molecular representation.

performance can be roughly classified into three catalogs according to different evaluation objects. (1) The evaluation metrics of the entire molecule set aim to assess the difference between the generated and test sets, such as the average Tanimoto similarity coefficient between two molecule sets, and generate the validity of the molecular set, novelty, and uniqueness. (2) Evaluation metrics for evaluating individual molecules in the molecular set are utilized in the following examples. Bickerton et al.[32] used Quantitative Estimate of Drug-likeness (QED), which is the concept of desirability, to measure drug-likeness. Frechet ChemNet Distance (FCD)[33] is a measure of distribution between training sets and generated molecules. Synthetic accessibility and ring sizes are considered in penalized logP[34]. (3) Assessment of generative models with benchmark suites is demonstrated in GuacaMol[35] and MOSES[36].

The issue of molecular optimization can be framed as a machine translation problem[37] in natural language processing, where a text is translated from one language to another. A way to translate initial molecules into target molecules with optimized properties based on SMILES must be determined. The datasets comprise a set of MMPs. The application of MMPs is a widely used design strategy by medicinal chemists due to its intuitive nature. The MMPs (including reverse transformations) are extracted from ChEMBL[38] using an open-source MMP tool[39]. A transformer is used in this study to generate new molecules with specific properties from initial molecules.

2 Method

2.1 Molecule and property representation

The SMILES representation of molecules[40] is used in models to train a set of MMPs. Figures 2a and 2c show examples.

logD and solubility are chosen as specific properties in this article. logD is the logarithm of the partition coefficient of a compound between an organic phase (e.g., octanol) and an aqueous phase (e.g., buffer) at a given pH. logD mainly affects the physicochemical and metabolic properties, as well as the activity and toxicity of compounds. Solubility is a type of physical property; for example, the maximum number of certain materials is dissolved in 100 g of solvent under a certain temperature. One of the most crucial characteristics in

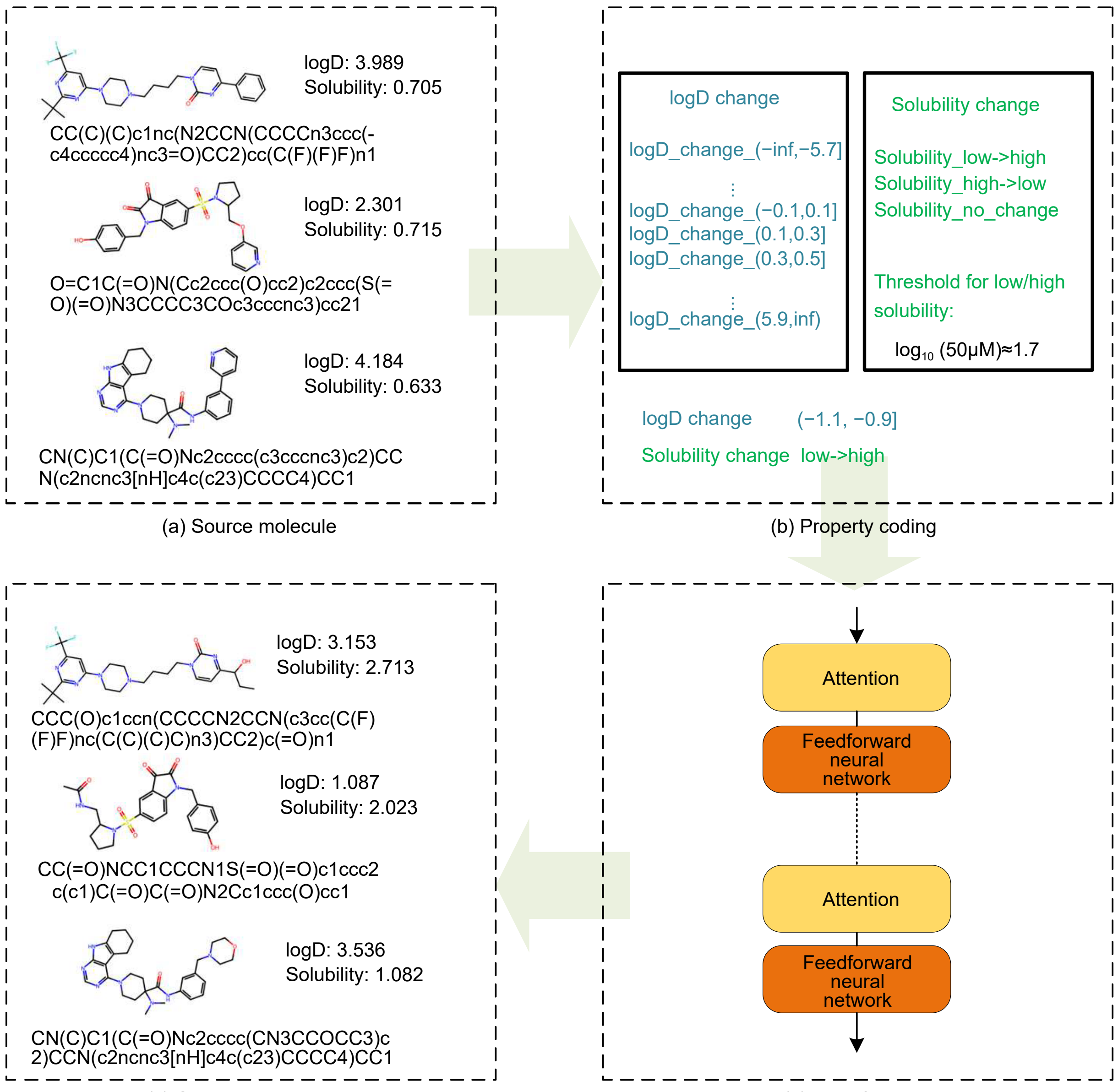


Fig. 2 Molecule and property representation.

the search for new drugs lies in their solubility. Compounds with low solubility can have numerous undesirable effects, such as poor absorption and bioavailability after oral administration of drugs and aggravated patient burden due to frequent high doses of medication. Moreover, several drug properties affecting ADMET are related to the two properties. Digitally labeling the features that have an important impact on the task is necessary to reduce the learning difficulty of models and enable them to learn the chemical structures of molecules and the complex relationships between the physics, chemistry, and biology of molecules effectively. Molecular properties are generally measured by quantified property fractions. Therefore, logD and solubility must be encoded as corresponding vectors[41] by one-hot encoding before inputting specific properties into models. The fractions for the properties of each drug are different. Requiring consistency for every property fraction is unreasonable when coding molecular properties, and a reasonable coding method must be defined. The solubility can be encoded in the following three states: from low to high, from high to low, and no change. Thresholds must be set to distinguish and measure these states. However, the value range of logD is relatively large and needs a detailed description; therefore, logD is encoded as the range interval. Figure 2b provides additional details.

Source molecules must be connected with the encoded SMILES with property changes in the input process to transform source molecules into target molecules with specific properties, and the generated target sequences are the SMILES of target molecules. The general process is shown in Fig. 3.

This task aims to provide a set of MMPs $\{A, B, C\}$, where A, B, and C represent the source molecule, the target molecule, and the property change between A and B, respectively. The model will learn a mapping $(A, C) \in \boldsymbol{A} \times \boldsymbol{C} \longrightarrow B \in \boldsymbol{B}$ during the training process, where $\boldsymbol{A} \times \boldsymbol{C}$ represents input space including all source molecules with property changes, and $\boldsymbol{B}$ represents output space including all target molecules. Given any group $(A, C) \in \boldsymbol{A} \times \boldsymbol{C}$, the model can generate a diverse set of target molecules with certain properties during the testing process.

2.2 Transformer architecture and calculation details

The transformer architecture will be comprehensively interpreted for the generation of molecules with specific properties proposed in this article. The transformer is also a classical encoder-decoder mode. The specific flow chart is shown in Fig. 4. The specific calculation process of the transformer is analyzed on the basis of matrix operation.

2.2.1 Encoder

The input part of the encoder comprises two parts: the word-coding matrix $I \in \mathbf{R}^{n \times l \times d}$ and the position-coding matrix $P \in \mathbf{R}^{n \times l \times d}$, where n, l, and d represent the number of source sequences, the maximum number of words, and the dimension of the word vector, respectively. The position-coding matrix P represents the position information of each word in a sentence.

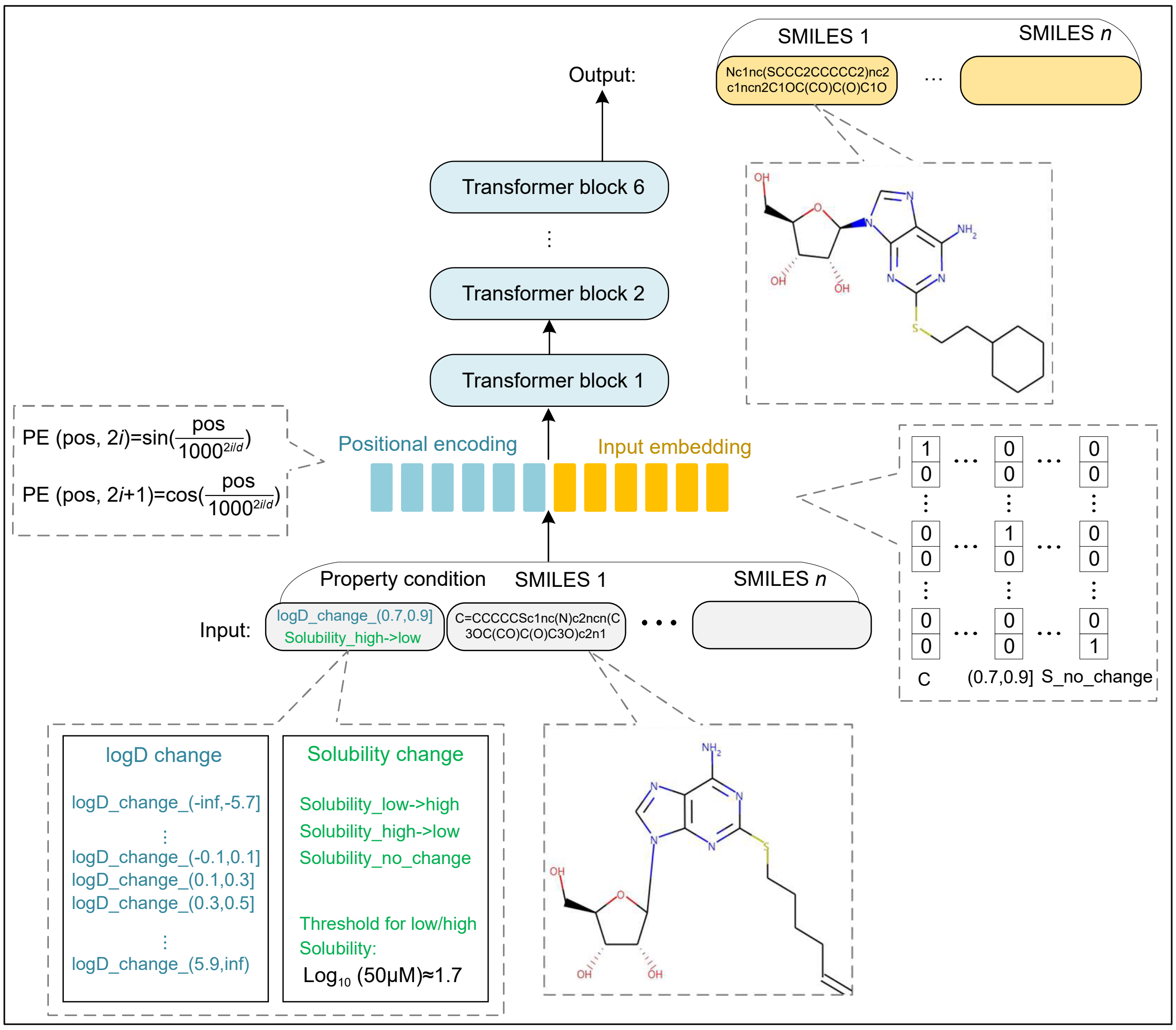


Fig. 3 Training process of molecular optimization work.

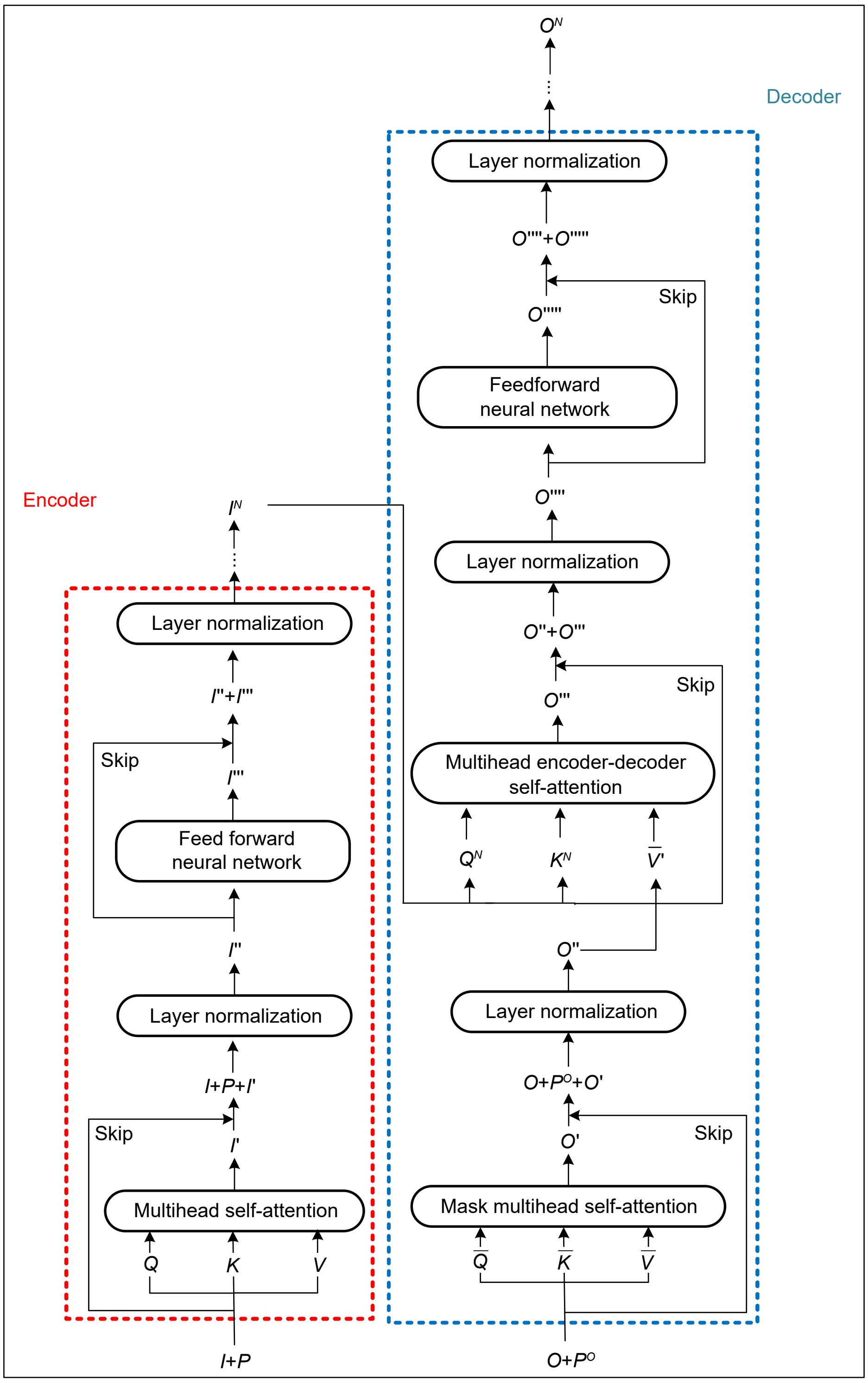


Fig. 4 Flowchart of the transformer block.

Numerous ways to generate position-coding vectors are available. The commonly used method is to generate a one-hot location code according to the position of a word in a sentence. Trigonometric functions are used in this article for position coding, as shown in the following:

$$\begin{cases} \mathrm{PE}\,(\mathrm{pos}, 2i) = \sin\left(\dfrac{\mathrm{pos}}{1000^{2i/d}}\right), \\ \mathrm{PE}\,(\mathrm{pos}, 2i+1) = \cos\left(\dfrac{\mathrm{pos}}{1000^{2i/d}}\right) \end{cases} \tag{1}$$

where PE represents the position encoding vector, pos represents the position of the word in the sentence, and i represents the position index of the encoding vector. The input matrix $I + P$ is transformed linearly to generate the matrices Q, K, and V. However, in the actual training process, $I + P$ is directly assigned to Q, K, and V. The word length is different. Therefore, padding, which uses zero to fill each word to the maximum word length, must be employed. Q, K, and V are inputted into the multihead self-attention module to calculate the attention distribution to obtain the matrix $I' \in \mathbf{R}^{n \times l \times d}$,

$$I' = \text{MultiHead}\,(Q, K, V) \tag{2}$$

The residual of the original input $I + P$ and attention distribution I' are then computed to obtain the output matrix $I + P + I' \in \mathbf{R}^{n \times l \times d}$.

Afterward, layer normalization is performed on $I + P + I' = \{x_{ijk}\}^{n \times l \times d}$ to obtain $I'' \in \mathbf{R}^{n \times l \times d}$, and the specific calculation is shown in the following:

$$\begin{aligned} &\mu^{ij} = \sum_{k=1}^{n} x_{ijk}, \\ &\sigma^{ij} = \sqrt[n]{\sum_{k=1}^{d} (x_{ijk} - \mu^{ij})^2}, \\ &\hat{x}_{ijk} = \frac{x_{ijk} - \mu^{ij}}{\sigma^{ij}}, \\ &i \in \{1, 2, \dots, n\}, j \in \{1, 2, \dots, l\}, \text{and } k \in \{1, 2, \dots, d\} \end{aligned} \tag{3}$$

I'' is inputted into a fully connected feedforward neural network to obtain $I''' \in \mathbf{R}^{n \times l \times d}$, and residual calculation between I'' and I''' is performed to acquire $I'' + I'''$.

Layer normalization operation is then performed on $I'' + I'''$. The above equation presents the detailed operation of a block. The number of blocks set in this training is six. The entire encoder module comprises six blocks, and the final output is obtained.

2.2.2 Decoder

The decoder input also comprises the following two parts: the word-coding matrix $O \in \mathbf{R}^{n_1 \times l_1 \times d}$ and the position-coding matrix $P^O \in \mathbf{R}^{n_1 \times l_1 \times d}$. In the model training process, the decoder's input word-coding matrix is the target sequence. In addition, the decoder's input is time sequential (that is, the output of the previous step is the input of the current step). Therefore, introducing the mask matrix in order is necessary to calculate the attention distribution.

The input matrix $O + P^O$ is transformed linearly to generate the matrices $\overline{Q}$, $\overline{K}$, and $\overline{V}$. However, in the actual training process, $O + P^O$ is directly assigned to $\overline{Q}$, $\overline{K}$, and $\overline{V}$. Similar to the encoder, 0 must be added to short word vectors.

$\overline{Q}$, $\overline{K}$, $\overline{V}$, and mask matrix M are then inputted into the mask multihead self-attention module to calculate attention distribution, and matrix $O' \in \mathbf{R}^{n_1 \times l_1 \times d}$ is obtained,

$$O' = \text{MaskMultiHead}\left(\overline{Q}, \overline{K}, \overline{V}, M\right) \tag{4}$$

Afterward, the residual between the original input $O + P^O$ and the attention distribution O' is computed to obtain the output matrix $O + P^O + O' \in \mathbf{R}^{n_1 \times l_1 \times d}$, and the layer normalization of $O + P^O + O'$ is then performed to acquire $O'' \in \mathbf{R}^{n_1 \times l_1 \times d}$.

The encoder information should be inputted into the decoder. Encoder output I^N can obtain Q^N and K^N, and O' can obtain $\overline{V}'$ through a linear transformation. The cross-attention distribution of Q^N, K^N, and $\overline{V}'$ can be used to obtain O''',

$$O''' = \text{MultiHead}\left(Q^N, K^N, \overline{V}'\right) \tag{5}$$

The cross-attention distribution synthesizes the output of the encoder and the intermediate result information of the decoder. However, in the actual training process, I^N is directly assigned to Q^N and K^N, and O'' is directly assigned to $\overline{V}'$. Next, the residual between O'' and O''' is calculated to obtain the output matrix $O'' + O'''$. The layer normalization operation is performed on $O'' + O'''$ to obtain O'''', and O'''' is inputted into a fully connected neural network to obtain O'''''. Residual operation is conducted to obtain $O'''' + O'''''$, and the layer normalization operation is finally conducted again.

The above is the detailed operation of a block. In this experiment, six blocks are stacked to form the entire decoder module, and the obtained output is $O^N \in \mathbf{R}^{n_1 \times l_1 \times d}$. The word in the vocabulary that currently predicts the maximum probability is identified, and the word vector is used as input for the next stage. The steps are repeated until the ”end” character is selected.

2.2.3 Multihead self-attention

Numerous works related to transformers do not comprehensively introduce the calculation method of multihead self-attention through examples. For the convenience of illustration, the head is set to 2, and the head of the actual training process is 8.

Take Fig. 5 as an example. Given the input vector a^1, a^2, $a^3 \in \mathbf{R}^{d_l \times 1}$, query vector $q^i \in \mathbf{R}^{d_k \times 1}$, key vector $k^i \in \mathbf{R}^{d_k \times 1}$, and value vector $v^i \in \mathbf{R}^{d_l \times 1}$ can be obtained by linear transformation through matrix $W^q \in \mathbf{R}^{d_k \times d_l}$, $W^k \in \mathbf{R}^{d_k \times d_l}$, and $W^v \in \mathbf{R}^{d_l \times d_l}$. A linear transformation is then performed on the query vector q^i through matrix $W^{q1} \in \mathbf{R}^{d_m \times d_k}$ and $W^{q2} \in \mathbf{R}^{d_m \times d_k}$ to obtain $q^{i1} \in \mathbf{R}^{d_m \times 1}$ and $q^{i2} \in \mathbf{R}^{d_m \times 1}$. Similarly, the same operation is performed on key vector k^i and value vector v^i to obtain $k^{i1} \in \mathbf{R}^{d_m \times 1}$, $k^{i2} \in \mathbf{R}^{d_m \times 1}$ and $v^{i1} \in \mathbf{R}^{(d_l/2) \times 1}$, $v^{i2} \in \mathbf{R}^{(d_l/2) \times 1}$, respectively. The specific calculation is as follows:

$$\begin{cases} q^{ih} = W^{qh} \cdot W^q \cdot a^i, \\ k^{ih} = W^{kh} \cdot W^k \cdot a^i, \\ v^{ih} = W^{vh} \cdot W^v \cdot a^i, \end{cases} \\ i \in \{1, 2, 3\},\ h \in \{1, 2\} \tag{6}$$

where h represents the number of heads.

The vectors are then integrated for the matrix calculation,

$$\begin{cases} Q^1 = (q^{11}, q^{21}, q^{31}) = W^{q1} \cdot W^q \cdot A, \\ K^1 = (k^{11}, k^{21}, k^{31}) = W^{k1} \cdot W^k \cdot A, \\ V^1 = (v^{11}, v^{21}, v^{31}) = W^{v1} \cdot W^v \cdot A, \end{cases}$$

$$\begin{cases} Q^2 = (q^{12}, q^{22}, q^{32}) = W^{q2} \cdot W^q \cdot A, \\ K^2 = (k^{12}, k^{22}, k^{32}) = W^{k2} \cdot W^k \cdot A, \\ V^2 = (v^{12}, v^{22}, v^{32}) = W^{v2} \cdot W^v \cdot A \end{cases} \tag{7}$$

For each head, the query and key vectors should be used to calculate the corresponding attention score. The calculation formula of the l components of the attention vector α^{ih} is as follows:

$$\alpha_l^{ih} = (q^{ih})^{\mathrm{T}} \cdot k^{ih}, \\ i, l \in \{1, 2, 3\},\ h \in \{1, 2\} \tag{8}$$

Then, the attention score matrix is obtained as follows:

$$\begin{cases} A^1 = (\alpha^{11}, \alpha^{21}, \alpha^{31}) = \dfrac{(K^1)^{\mathrm{T}} \cdot Q^1}{\sqrt{d_m}}, \\ A^2 = (\alpha^{12}, \alpha^{22}, \alpha^{32}) = \dfrac{(K^2)^{\mathrm{T}} \cdot Q^2}{\sqrt{d_m}} \end{cases} \tag{9}$$

The normalized attention distribution β^{ih} is obtained after the softmax layer,

$$\beta_j^{ih} = \frac{\mathrm{e}^{\alpha_j^{ih}}}{\sum_{n=1}^{3} \mathrm{e}^{\alpha_n^{ih}}}, \\ i, j \in \{1, 2, 3\},\ h \in \{1, 2\} \tag{10}$$

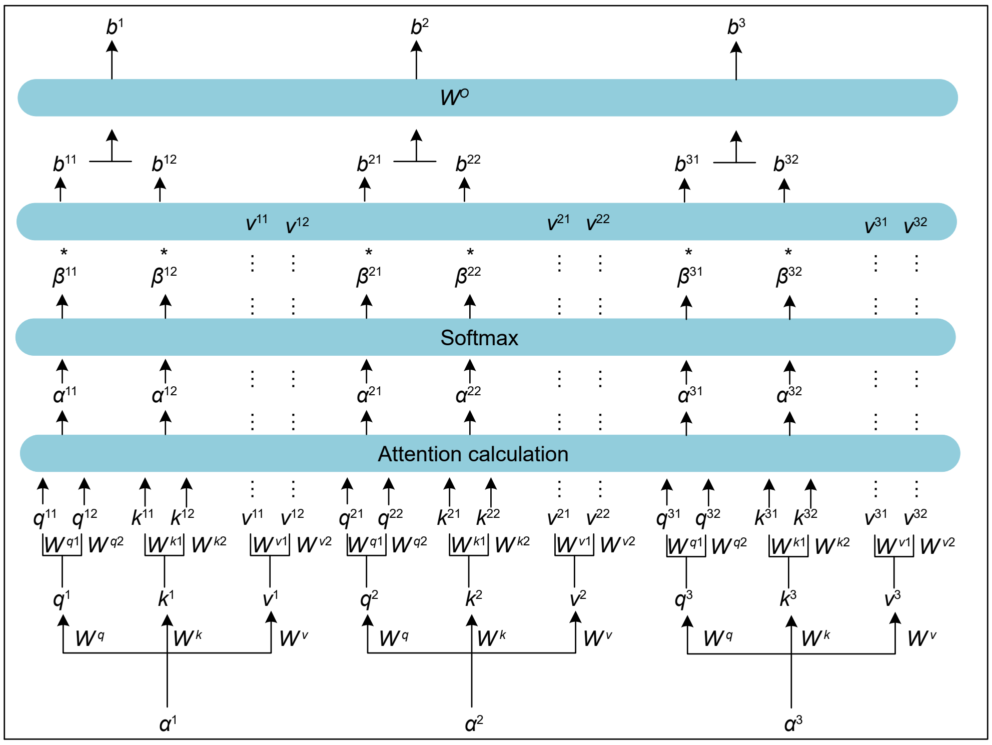


Fig. 5 Flow chart of multihead self-attention.

Attention distribution vector β^{ih} and value matrix V^h should be used for each head to obtain the output $b^{ih} \in \mathbf{R}^{(d_l/2)\times 1}$, as shown in the following:

$$b^{ih} = \sum_{l=1}^{3} \beta_l^{ih} \cdot v^{lh},$$
$$i \in \{1, 2, 3\},\ h \in \{1, 2\} \tag{11}$$

Concatenating the two head vectors,

$$B = \begin{pmatrix} b^{11} & b^{21} & b^{31} \\ b^{12} & b^{22} & b^{32} \end{pmatrix} \tag{12}$$

The given parameter matrix $W^O \in \mathbf{R}^{d_l \times d_l}$ is combined, and the output matrix is obtained as follows:

$$O = W^O \cdot B \in \mathbf{R}^{d_l \times 3} \tag{13}$$

Finally, the overall calculation can be obtained,

$$O = W^O \cdot \mathrm{Concat}\begin{pmatrix} V^1 \cdot \mathrm{softmax}\left(\dfrac{(K^1)^{\mathrm{T}} \cdot Q^1}{\sqrt{d_m}}\right) \\ V^2 \cdot \mathrm{softmax}\left(\dfrac{(K^2)^{\mathrm{T}} \cdot Q^2}{\sqrt{d_m}}\right) \end{pmatrix} \tag{14}$$

2.2.4 Additional transformer information

Specific parameters trained with the transformer are shown in Table 1, where H represents the number of heads in the multihead attention mechanism, and N represents the number of transformer blocks, as well as the training parameters of the Adam optimizer and the parameters set during the training process.

Using multihead, the parameter matrix can form multiple subspaces, and the overall size of the matrix remains the same. However, the dimension size corresponding to each head is changed. Therefore, the matrix can learn various pieces of information, but the computation amount is similar to that of a single head. In Eq. (14), when softmax is used to process attention scores, division by $\sqrt{d_m}$ is necessary to control the vanishing gradient problem effectively. Residuals and layer normalization can be used in the transformer. These steps can also be added to other training models to improve the training performance.

Table 1 Training parameters of transformer.

Parameter	Description	Value
H	Number of heads in self-attention	8
N	Number of transformer blocks	6
d_model	Dimensions of the input and output	256
d_ff	Extended dimension of the feedforward layer	2048
batch_size	Size of batch	128
num_epoch	Number of epoch	60
adam_beta1	Exponential decay rate for the 1st moment estimates	0.9
adam_beta2	Exponential decay rate for the 2nd moment estimates	0.98
adam_eps	Small value for numerical stability	10^{-9}

3 Experiment and Result

3.1 Baseline

The transformer is compared with the Seq2Seq[30], MGCVAE[22], and JT-VAE[13] baselines.

Transformer uses sequences to represent molecules. Seq2Seq also utilizes sequences, while MGCVAE and JT-VAE employ graphs to represent molecules. The three comparison methods and transformer belong to the encoder-decoder mode; therefore, the comparison experiment will be highly sufficient. The main purpose is to compare the efficiency of the transformer with other models in generating molecules that satisfy the required properties.

3.2 Data preparation

Transformer and baseline models are trained on a set of MMPs[42] extracted from ChEMBL[43] with property changes between source and target sequences. The selected molecules are standardized using MolVS[44]. The molecular pairs are processed in accordance with the following constraints: number of heavy atoms < 50; number of heavy atoms in the R group < 10; ratio of heavy atoms in the R group < 0.5; AZFilter = CORE[45] to filter out low-quality compounds. Each molecule's property values are within three standard deviations of all molecule property values. Then, 12 365, 1503, and 1570 MMPs are randomly sampled as the training, validation, and test sets, respectively, from full molecules.

3.3 Evaluation metrics

For each initial molecular test set, 10 unique and valid molecules are generated, and these generated molecules are not identical. The number of molecules satisfying the specific changes of logD and solubility is then calculated. The ADMET property prediction model[30] is used to compute the properties of generated molecules. The model error (test RMSE) is considered if a generated molecule satisfies its desirable properties. For logD, the error between the generated and target logD values is guaranteed to be less than 0.4. For solubility, a standard must also be established to frame the threshold range.

The generated and initial molecules are then analyzed, and the performance between the models is compared with the proportion of molecules satisfying specific properties. Figure 6 shows the property distribution of source and target molecules on the training set. Dark areas have highly concentrated molecules. The reverse transformation is represented in the data set. Thus, the distribution of the source molecule properties is comparable to that of the target molecule properties.

3.4 Conditional vs. unconditional transformer

This part of the test compares a conditional model with input source molecules and specific properties to an unconditional model with input source molecules only. From 1570 initial test molecules, 10 unique and valid molecules are generated, which must differ from the original molecules.

Figure 7 shows the performance of the conditional and unconditional transformers satisfying two specific properties. This experiment uses the *K*-sample Anderson-Darling test[46]. This test is employed to examine if a data sample came from a population with a specific distribution. This figure shows that the performance of the conditional transformer is better than that of the unconditional transformer. Of the 10 molecules generated per initial molecule, conditional and unconditional transformers averaged 6 and 3, respectively.

3.5 Transformer vs. baseline models

This experiment compares the performance of the transformer and the three other methods under the same test set and provides the same specific performance constraints.

Table 2 shows the ratio of generated molecules

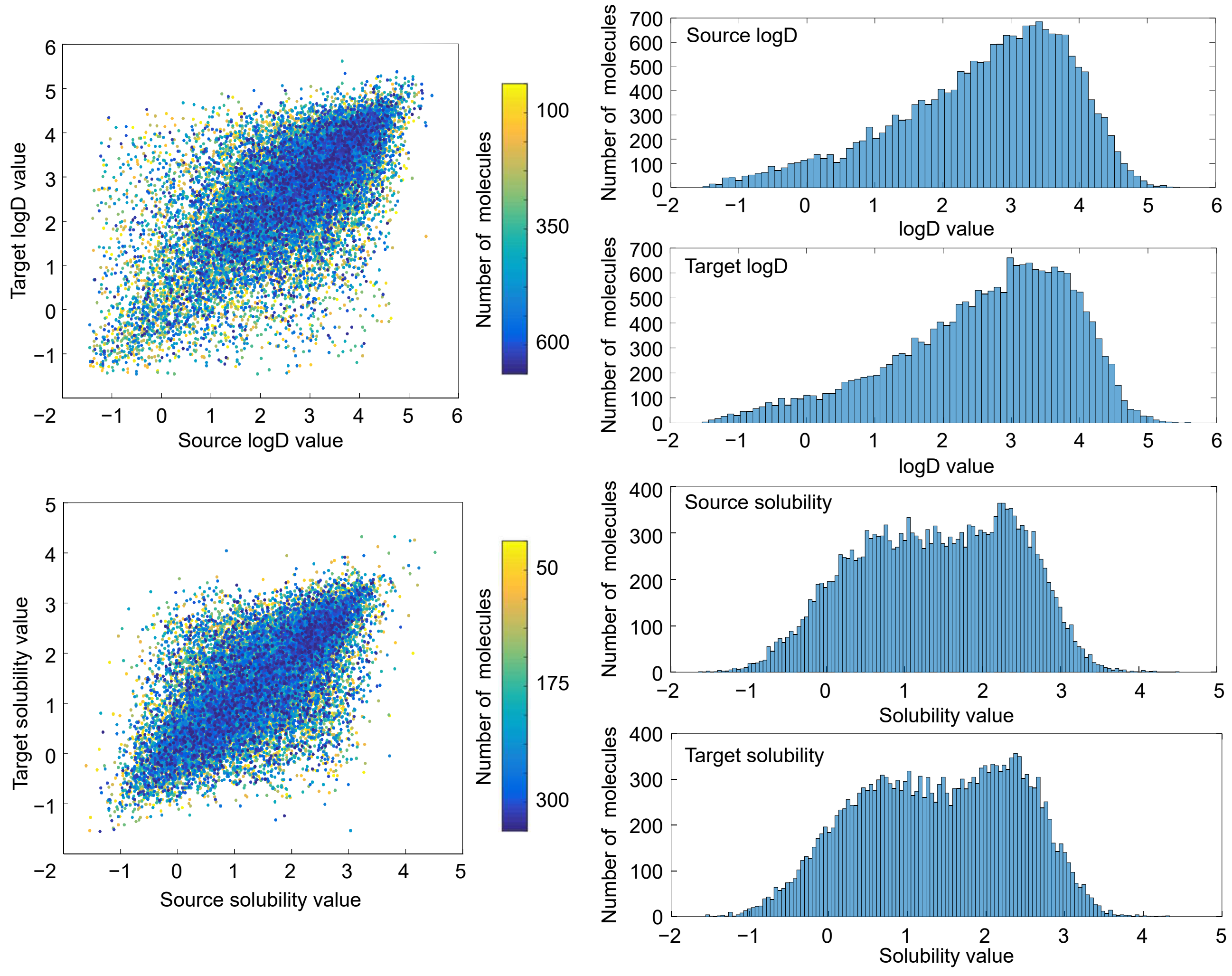


Fig. 6 Distributions and properties of source and target molecules in training.

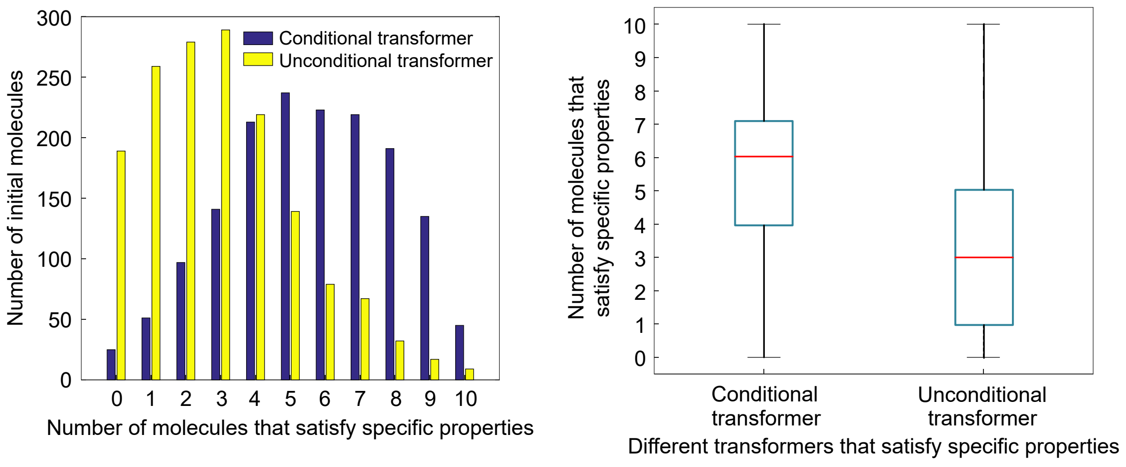


Fig. 7 Conditional transformer vs. unconditional transformer.

Table 2 Comparison of the ratio of generated molecules satisfying specific properties based on different methods.

(%)

Test set	Transformer	Seq2Seq	MGCVAE	JT-VAE
Original	**55.23**	44.93	54.03	54.56
$R < 0.5$	**94.45**	87.23	92.46	91.68

satisfying specific properties to all generated molecules for different methods. The second test set is selected under the condition that R group < 0.5 (the ratio of heavy atoms in R group < 0.5). The performance of the transformer is better than that of Seq2Seq. MGCVAE and JT-VAE also show good performance.

Moreover, the transformer is compared with baseline models based on the evaluation metrics of MOSES. Validity refers to the generation of satisfying the rules of chemical molecules that account for the proportion of all generated molecules. Uniqueness is the proportion of different molecules that are formed, and Unique@10 000 stands for the uniqueness value obtained per 10 000 generated molecules. Novelty refers to the proportion of generated molecules that are different from the source dataset of those in all generated molecules. FCD is calculated using the features of generated molecules and those molecules in the dataset. A low FCD value means that the model has successfully captured the statistical information of the dataset. IntDiv evaluates the chemical diversity of generated molecules and detects whether the model has a pattern collapse, and the value range is [0, 1].

As shown in Table 3 shows 100% validity because the formation of molecules requires the generation of 10 unique valid molecules for each initial molecule. Additional unique molecules can be generated through

Table 3 Comparison of transformer and baseline models based on MOSES.

Model	Validity	Unique@10000	Novelty	FCD/test	IntDiv
Transformer	1.000	0.996	**0.981**	**5.442**	0.867
Seq2Seq	1.000	0.979	0.963	8.560	0.866
MGCVAE	1.000	0.998	0.979	6.096	0.866
JT-VAE	1.000	**0.999**	0.964	6.139	0.867

JT-VAE, thus ensuring the diversity of molecules. Transformer achieves the best performance in novelty and FCD, ensuring the differentiation between generated and training set molecules as well as effectively capturing the distribution information of training set. The four methods for Internal Diversity (IntDiv) are the same, which can achieve high chemical diversity.

The numerical transformation of logD and solubility is analyzed. The test set is framed, and test molecules with high logD and low solubility are selected ($2 < \log D < 4.5$ and solubility < 1.7). Initial molecules must be optimized to produce new molecules with low logD and high solubility. As shown in Fig. 8, the properties of the source molecule are plotted on the left, and those of the predicted molecules are on the right (the predicted values of the properties of the 10 predicted molecules generated for each initial molecule are numerically averaged). The prediction of specific properties by a transformer can extensively widen the distribution of generated molecules, which markedly improves the diversity of molecules with specific properties. Most compounds become slightly soluble as logD increases, and a link is observed between logD and solubility[47]. Some molecules may not follow this pattern because other properties might affect

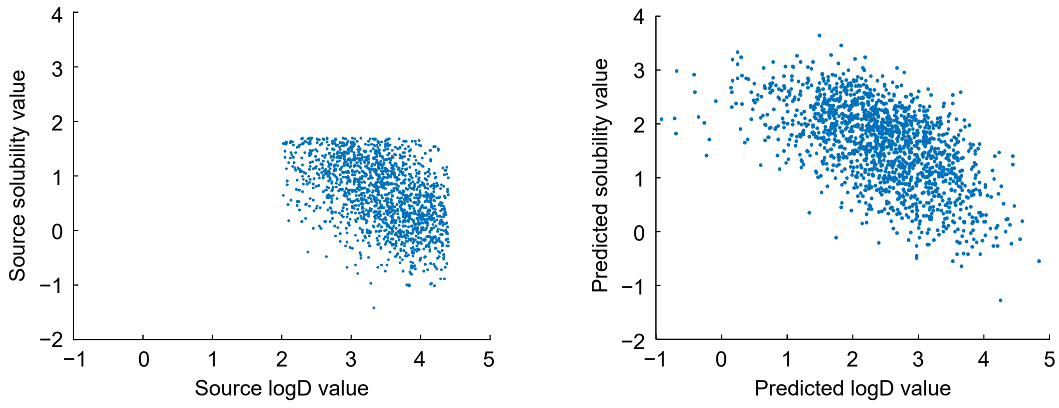


Fig. 8 Property distribution of initial molecules and generated molecules.

molecules. Future work may need to consider additional properties, such as stability and toxicity, comprehensively.

Figure 9 shows an example of different molecules with specific properties generated by transformer and baseline models concerning the initial molecule and the specific properties. Three molecules are selected from each model for observation. Each of the resulting molecules has some modifications to the original molecules and has the required specific properties. The molecules in the wireframe of the same color are identical. Thus, the molecules produced by the baseline and transformer models may overlap. Some details must still be uncovered. For example, the error of molecules generated by transformers MGCVAE and JT-VAE in specific properties is small, and the generated molecules can effectively satisfy the required properties. Overall, the transformer and other models can generate different sets of molecules with specific properties.

4 Conclusion and Challenge

The molecular property optimization problem can be defined as a machine translation problem, which

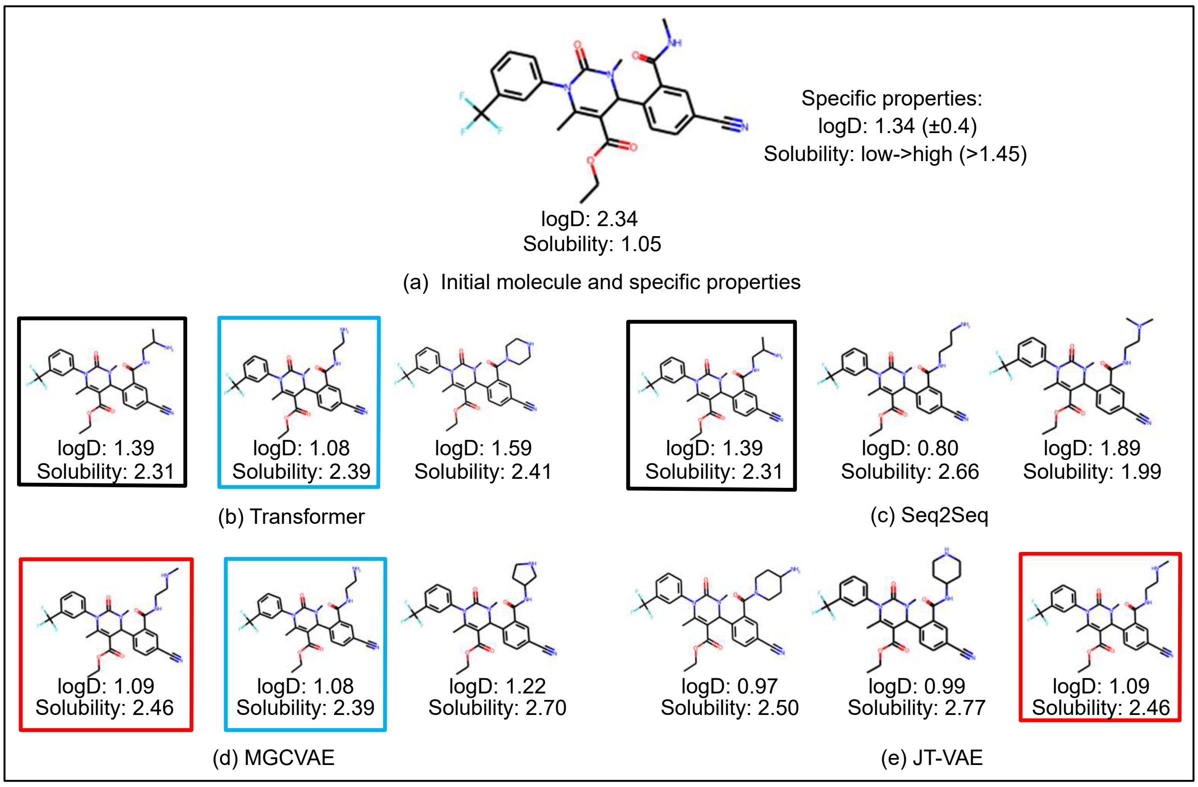

Specific properties:
logD: 1.34 (±0.4)
Solubility: low->high (>1.45)
logD: 2.34
Solubility: 1.05
(a) Initial molecule and specific properties
logD: 1.39 Solubility: 2.31
logD: 1.08 Solubility: 2.39
logD: 1.59 Solubility: 2.41
(b) Transformer
logD: 1.39 Solubility: 2.31
logD: 0.80 Solubility: 2.66
logD: 1.89 Solubility: 1.99
(c) Seq2Seq
logD: 1.09 Solubility: 2.46
logD: 1.08 Solubility: 2.39
logD: 1.22 Solubility: 2.70
(d) MGCVAE
logD: 0.97 Solubility: 2.50
logD: 0.99 Solubility: 2.77
logD: 1.09 Solubility: 2.46
(e) JT-VAE

Fig. 9 Examples of different molecules with specific properties generated from transformer and baseline models.

generates new molecules with specific properties by inputting the initial molecules and specific properties into the model. The optimization performance of the transformer is better than Seq2Seq and has its advantages compared with MGCVAE and JT-VAE. The transformer and these methods have produced different new molecules with specific properties.

Ten unique and valid molecules are generated in this article, satisfying specific properties for each initial molecule in the test set and ensuring their uniqueness from the initial molecules. The conditional and unconditional transformer models are compared. At least 50% of the conditionally constrained transformer can generate six molecules satisfying certain properties, while the unconditional transformer can only have three molecules. The optimization performance of the transformer and baseline models is also compared. The transformer ensures the difference between the generated molecules and the training set molecules, and can effectively capture the distribution information of the training set. The performance of the three baseline methods is different from that of the transformer. However, these methods still produce some molecules that satisfy specific properties. The optimized use of these models will help in substantially enriching different molecules. The current study encounter some challenges. A transformer is a model based on sequence representation. In the subsequent work, additional models based on the graph representation should also be included for comparison. In addition to logD and solubility, other specific properties should also be regarded to generate additional new molecules. These conditions and other difficulties will be considered in future research efforts.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (Nos. 62272288, 61972451, and U22A2041) and the Shenzhen Key Laboratory of Intelligent Bioinformatics (No. ZDSYS20220422103800001).

References

[1] P. G. Polishchuk, T. I. Madzhidov, and A. Varnek, Estimation of the size of drug-like chemical space based on GDB-17 data, *J. Comput. Aided Mol. Des.*, vol. 27, no. 8, pp. 675–679, 2013.

[2] S. Heller, A. McNaught, S. Stein, D. Tchekhovskoi, and I. Pletnev, InChI – the worldwide chemical structure identifier standard, *J. Cheminform.*, vol. 5, no. 1, p. 7, 2013.

[3] N. M. O'Boyle and A. Dalke, DeepSMILES: An adaptation of SMILES for use in machine-learning of chemical structures, doi:10.26434/chemrxiv.7097960.

[4] M. Krenn, F. Häse, A. K. Nigam, P. Friederich, and A. Aspuru-Guzik, Self-Referencing Embedded Strings (SELFIES): A 100% robust molecular string representation, *Mach. Learn. Sci. Technol.*, vol. 1, no. 4, p. 045024, 2020.

[5] E. J. Bjerrum and R. Threlfall, Molecular generation with recurrent neural networks (RNNs), arXiv preprint arXiv: 1705.04612, 2017.

[6] A. Gupta, A. T. Müller, B. J. H. Huisman, J. A. Fuchs, P. Schneider, and G. Schneider, Generative recurrent networks for *de novo* drug design, *Mol. Inform.*, vol. 37, nos. 1&2, p. 1700111, 2018.

[7] M. H. S. Segler, T. Kogej, C. Tyrchan, and M. P. Waller, Generating focused molecule libraries for drug discovery with recurrent neural networks, *ACS Cent. Sci.*, vol. 4, no. 1, pp. 120–131, 2018.

[8] R. Gómez-Bombarelli, J. N. Wei, D. Duvenaud, J. M. Hernández-Lobato, B. Sánchez-Lengeling, D. Sheberla, J. Aguilera-Iparraguirre, T. D. Hirzel, R. P. Adams, and A. Aspuru-Guzik, Automatic chemical design using a data-driven continuous representation of molecules, *ACS Cent. Sci.*, vol. 4, no. 2, pp. 268–276, 2018.

[9] J. Lim, S. Ryu, J. W. Kim, and W. Y. Kim, Molecular generative model based on conditional variational autoencoder for *de novo* molecular design, *J. Cheminform.*, vol. 10, p. 31, 2018.

[10] M. J. Kusner, B. Paige, and J. M. Hernández-Lobato, Grammar variational autoencoder, in *Proc. 34th Int. Conf. Machine Learning*, Sydney, Australia, 2017, pp. 1945–1954.

[11] H. Dai, Y. Tian, B. Dai, S. Skiena, and L. Song, Syntax-directed variational autoencoder for molecule generation, in *Proc. Int. Conf. Learning Representations*, https://doi.org/10.48550/arXiv.1802.08786, 2018.

[12] Q. Liu, M. Allamanis, M. Brockschmidt, and A. L. Gaunt, Constrained graph variational autoencoders for molecule design, in *Proc. 32nd Int. Conf. Neural Information Processing Systems*, Montréal, Canada, 2018, pp. 7806–7815.

[13] W. Jin, R. Barzilay, and T. Jaakkola, Junction tree variational autoencoder for molecular graph generation, in *Proc. 35th Int. Conf. Machine Learning*, Stockholm, Sweden, 2018, pp. 2323–2332.

[14] M. Simonovsky and N. Komodakis, GraphVAE: Towards generation of small graphs using variational autoencoders, in *Proc. 27th Int. Conf. Artificial Neural Networks*, Rhodes, Greece, 2018, pp. 412–422.

[15] G. L. Guimaraes, B. Sanchez-Lengeling, C. Outeiral, P. L. C. Farias, and A. Aspuru-Guzik, Objective-reinforced generative adversarial networks (ORGAN) for sequence generation models, arXiv preprint arXiv: 1705.10843, 2018.

[16] E. Putin, A. Asadulaev, Y. Ivanenkov, V. Aladinskiy, B. Sanchez-Lengeling, A. Aspuru-Guzik, and A. Zhavoronkov, Reinforced adversarial neural computer for *de novo* molecular design, *J. Chem. Inf. Model.*, vol. 58, no. 6, pp. 1194–1204, 2018.

[17] E. Putin, A. Asadulaev, Q. Vanhaelen, Y. Ivanenkov, A.

V. Aladinskaya, A. Aliper, and A. Zhavoronkov, Adversarial threshold neural computer for molecular *de novo* design, *Mol. Pharm.*, vol. 15, no. 10, pp. 4386–4397, 2018.

[18] N. De Cao and T. Kipf, MolGAN: An implicit generative model for small molecular graphs, arXiv preprint arXiv: 1805.11973, 2022.

[19] L. Dinh, D. Krueger, and Y. Bengio, NICE: Non-linear independent components estimation, arXiv preprint arXiv: 1410.8516, 2015.

[20] L. Dinh, J. Sohl-Dickstein, and S. Bengio, Density estimation using real NVP, arXiv preprint arXiv: 1605.08803, 2017.

[21] D. P. Kingma and P. Dhariwal, Glow: Generative flow with invertible 1x1 convolutions, arXiv preprint arXiv: 1807.03039, 2018.

[22] M. Lee and K. Min, MGCVAE: Multi-objective inverse design via molecular graph conditional variational autoencoder, *J. Chem. Inf. Model.*, vol. 62, no. 12, pp. 2943–2950, 2022.

[23] C. Li, J. Yao, W. Wei, Z. Niu, X. Zeng, J. Li, and J. Wang, Geometry-based molecular generation with deep constrained variational autoencoder, *IEEE Trans. Neural Netw. Learn. Syst.*, doi: 10.1109/TNNLS.2022.3147790.

[24] C. Ma and X. Zhang, GF-VAE: A flow-based variational autoencoder for molecule generation, in *Proc. 30th ACM Int. Conf. Information & Knowledge Management*, Virtual Event, Queensland, Australia, 2021, pp. 1181–1190.

[25] S. Luo, J. Guan, J. Ma, and J. Peng, A 3D generative model for structure-based drug design, arXiv preprint arXiv: 2203.10446, 2022.

[26] V. Bagal, R. Aggarwal, P. K. Vinod, and U. D. Priyakumar, MolGPT: Molecular generation using a transformer-decoder model, *J. Chem. Inf. Model.*, vol. 62, no. 9, pp. 2064–2076, 2022.

[27] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, Attention is all you need, in *Proc. 31st Int. Conf. Neural Information Processing Systems*, Long Beach, CA, USA, 2017. pp. 6000–6010.

[28] M. Langevin, H. Minoux, M. Levesque, and M. Bianciotto, Scaffold-constrained molecular generation, *J. Chem. Inf. Model.*, vol. 60, no. 12, pp. 5637–5646, 2020.

[29] J. Zhang and H. Chen, *De novo* molecule design using molecular generative models constrained by ligand-protein interactions, *J. Chem. Inf. Model.*, vol. 62, no. 14, pp. 3291–3306, 2022.

[30] J. He, H. You, E. Sandström, E. Nittinger, E. J. Bjerrum, C. Tyrchan, W. Czechtizky, and O. Engkvist, Molecular optimization by capturing chemist's intuition using deep neural networks, *J. Cheminform.*, vol. 13, no. 1, p. 26, 2021.

[31] J. He, E. Nittinger, C. Tyrchan, W. Czechtizky, A. Patronov, E. J. Bjerrum, and O. Engkvist, Transformer-based molecular optimization beyond matched molecular pairs, *J. Cheminform.*, vol. 14, no. 1, p. 18, 2022.

[32] G. R. Bickerton, G. V. Paolini, J. Besnard, S. Muresan, and A. L. Hopkins, Quantifying the chemical beauty of drugs, *Nat. Chem.*, vol. 4, no. 2, pp. 90–98, 2012.

[33] K. Preuer, P. Renz, T. Unterthiner, S. Hochreiter, and G. Klambauer, Fréchet ChemNet distance: A metric for generative models for molecules in drug discovery, *J. Chem. Inf. Model.*, vol. 58, no. 9, pp. 1736–1741, 2018.

[34] T. Fu, C. Xiao, and J. Sun, CORE: Automatic molecule optimization using copy & refine strategy, *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 1, pp. 638–645, 2020.

[35] N. Brown, M. Fiscato, M. H. S. Segler, and A. C. Vaucher, GuacaMol: Benchmarking models for de novo molecular design, *J. Chem. Inf. Model.*, vol. 59, no. 3, pp. 1096–1108, 2019.

[36] D. Polykovskiy, A. Zhebrak, B. Sanchez-Lengeling, S. Golovanov, O. Tatanov, S. Belyaev, R. Kurbanov, A. Artamonov, V. Aladinskiy, M. Veselov, et al., Molecular Sets (MOSES): A benchmarking platform for molecular generation models, *Front. Pharmacol.*, vol. 11, p. 565644, 2020.

[37] D. Bahdanau, K. Cho, and Y. Bengio, Neural machine translation by jointly learning to align and translate, arXiv preprint arXiv: 1409.0473, 2016.

[38] A. Gaulton, L. J. Bellis, A. P. Bento, J. Chambers, M. Davies, A. Hersey, Y. Light, S. McGlinchey, D. Michalovich, B. Al-Lazikani, et al., ChEMBL: a large-scale bioactivity database for drug discovery, *Nucleic Acids Res.*, vol. 40, no. D1, pp. D1100–D1107, 2012.

[39] A. Dalke, J. Hert, and C. Kramer, mmpdb: An open-source matched molecular pair platform for large multiproperty data sets, *J. Chem. Inf. Model.*, vol. 58, no. 5, pp. 902–910, 2018.

[40] D. Weininger, SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules, *J. Chem. Inf. Comput. Sci.*, vol. 28, no. 1, pp. 31–36, 1988.

[41] K. Yang, K. Swanson, W. G. Jin, C. Coley, P. Eiden, H. Gao, A. Guzman-Perez, T. Hopper, B. Kelley, M. Mathea, et al., Analyzing learned molecular representations for property prediction, *J. Chem. Inf. Model.*, vol. 59, no. 8, pp. 3370–3388, 2019.

[42] S. Turk, B. Merget, F. Rippmann, and S. Fulle, Coupling matched molecular pairs with machine learning for virtual compound optimization, *J. Chem. Inf. Model.*, vol. 57, no. 12, pp. 3079–3085, 2017.

[43] D. Mendez, A. Gaulton, A. P. Bento, J. Chambers, M. De Veij, E. Félix, M. P. Magariños, J. F. Mosquera, P. Mutowo, M. Nowotka, et al., ChEMBL: Towards direct deposition of bioassay data, *Nucleic Acids Res.*, vol. 47, no. D1, pp. D930–D940, 2019.

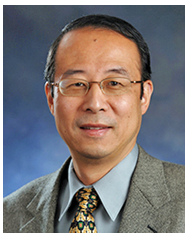
[44] M. Swain, MolVS: Molecule validation and standardization, https://pypi.org/project/Molvs, 2018.

[45] J. G. Cumming, A. M. Davis, S. Muresan, M. Haeberlein, and H. Chen, Chemical predictive modelling to improve compound quality, *Nat. Rev. Drug Discov.*, vol. 12, no. 12, pp. 948–962, 2013.

[46] F. W. Scholz and M. A. Stephens, *K*-sample Anderson-darling tests, *J. Am. Stat. Assoc.*, vol. 82, no. 399, pp. 918–924, 1987.

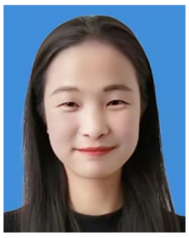
[47] J. B. Dressman and C. Reppas, In vitro-in vivo correlations for lipophilic, poorly water-soluble drugs, *Eur. J. Pharm. Sci.*, vol. 11, no. S2, pp. S73–S80, 2000.

Zhongyin Xu received the BEng degree from Shaanxi Normal University, Xi'an, China in 2016. He is currently a master student at School of Computer Science, Shaanxi Normal University, Xi'an, China. His current research interests include bioinformatics and intelligent computing, and deep learning.



Yi Pan received the BEng (computer engineering) and MEng (computer engineering) degrees from Tsinghua University, Beijing, China in 1982 and 1984, respectively, and the PhD degree (computer science) from University of Pittsburgh, USA in 1991. He is currently working as a dean and chair professor at Faculty of Computer Science and Control Engineering, Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China. He has published over 250 academic papers in SCI-indexed journals, including more than 100 papers in top IEEE/ACM transactions/journals. His publications have been cited more than 19 000 times, and his current H-index is 88. He has been awarded IEEE Distinguished Achievement Award, IEEE Distinguished Service Award, IEEE Transactions Best Paper Award, IEEE and other international conference best paper awards many times, IBM Professor Award four times, Andrew Mellon Award, and other awards. His research interests include parallel and distributed processing systems, Internet technology, and bioinformatics.

Xiujuan Lei received the MEng and PhD degrees from Northwestern Polytechnical University, Xi'an, China in 2001 and 2005, respectively. She is currently a professor at School of Computer Science, Shaanxi Normal University, Xi'an, China. Her research interests include bioinformatics, swarm intelligent optimization, data mining, and deep learning.



Mei Ma received the BEng and MEng degrees from Qinghai Normal University, Qinghai, China in 2010 and 2014, respectively. She is currently a PhD candidate at School of Computer Science, Shaanxi Normal University, Xi'an, China. Her current research interests include bioinformatics and intelligent computing, and deep learning.