molecules

MDPI

Article

CONSMI: Contrastive Learning in the Simplified Molecular Input Line Entry System Helps Generate Better Molecules

Ying Qian , Minghua Shi and Qian Zhang *

School of Computer Science and Technology, Shanghai Frontiers Science Center of Molecule Intelligent Syntheses, East China Normal University, 3663 North Zhongshan Road, Putuo District, Shanghai 200062, China; yqian@cs.ecnu.edu.cn (Y.Q.); 51215901061@stu.ecnu.edu.cn (M.S.)
* Correspondence: qzhang@cs.ecnu.edu.cn

Abstract: In recent years, the application of deep learning in molecular de novo design has gained significant attention. One successful approach involves using SMILES representations of molecules and treating the generation task as a text generation problem, yielding promising results. However, the generation of more effective and novel molecules remains a key research area. Due to the fact that a molecule can have multiple SMILES representations, it is not sufficient to consider only one of them for molecular generation. To make up for this deficiency, and also motivated by the advancements in contrastive learning in natural language processing, we propose a contrastive learning framework called CONSMI to learn more comprehensive SMILES representations. This framework leverages different SMILES representations of the same molecule as positive examples and other SMILES representations as negative examples for contrastive learning. The experimental results of generation tasks demonstrate that CONSMI significantly enhances the novelty of generated molecules while maintaining a high validity. Moreover, the generated molecules have similar chemical properties compared to the original dataset. Additionally, we find that CONSMI can achieve favorable results in classifier tasks, such as the compound–protein interaction task.

Keywords: deep learning; drug design; contrastive learning

check for updates

Citation: Qian, Y.; Shi, M.; Zhang, Q. CONSMI: Contrastive Learning in the Simplified Molecular Input Line Entry System Helps Generate Better Molecules. *Molecules* **2024**, *29*, 495. https://doi.org/10.3390/molecules29020495

Academic Editor: Bernard Maigret

Received: 15 November 2023
Revised: 12 January 2024
Accepted: 16 January 2024
Published: 19 January 2024

1. Introduction

Discovering new drugs and material molecules can bring tremendous social and technological progress. In particular, for some diseases that do not yet have effective treatment plans, new targeted drugs represent great hope. Discovering more drugs also provides a way to achieve personalized precision medicine [1]. However, drug discovery is a costly, time-consuming process with a high failure rate [2]. The estimated number of potential drug-like candidates ranges from 10^{23} to 10^{60} molecules [3], but only around 10^{8} molecules have been synthesized and investigated so far [4]. In molecular research, researchers are employing a synergy of genetic algorithms and object generation techniques for molecular screening [5–8]. These methods generally perform Pareto optimization on molecules to generate molecules with ideal properties. Additionally, they are utilizing deep learning generative models for a more accurate simulation of molecular distributions, showcasing the growing reliance on advanced artificial intelligence techniques in molecular research. Firstly, a generation model is used to simulate the molecular distribution using self-supervised learning, and then a series of molecules is generated using auto-regression starting from carbon atoms. With the thriving development of deep learning models in computer vision [9] and natural language processing (NLP) [10], these models are also employed to improve molecular distribution and, when combined with reinforcement learning and multimodal techniques, generate molecules with specific properties. There are two main ways to generate molecules: using graph-based methods and using SMILES notation methods. In addition, molecular generation methods based on SELFIES [11] have been very popular recently.

There is a lot of research on the graph-based method to generate molecules [12–14]. This method represents molecules as graph structures, where the nodes in the graph represent atoms and the edges represent chemical bonds. This representation is intuitive and can clearly present the atoms in the molecule and their connection relationships. MOLDR [14] decomposes the molecular graphs in the training dataset into subgraphs and reassembles them in different ways to generate new, optimized molecular graphs. The Junction Tree VAE (JT-VAE) [13] offers an alternative approach to molecular generation by representing molecules as graph tree structures. This representation allows for a more expressive and structured encoding of molecular compounds. One notable advantage of JT-VAE is its ability to guarantee the 100% validity of generated molecules. It achieves this by maintaining a vocabulary of molecular components that can be added at each junction of the molecule tree. By constructing molecules in a step-wise manner based on valid molecular components, JT-VAE ensures that every generated molecule adheres to the predefined rules and constraints of molecular validity. This design has three key limitations. Firstly, when using JT-VAE for attribute optimization, the task becomes more challenging because two molecules with the same connection tree may correspond to significantly different attributes. Secondly, the absence of consideration of the node order arrangement during the generation process can result in increased time consumption. Thirdly, due to the complexity of real-world drug molecules, generating substructures with less than 20 atoms is impractical [15].

The Simplified Molecular Input Line Entry System (SMILES) notation [16], which represents molecules as character strings, allows for the application of advanced deep learning models from the field of NLP for computational tasks involving molecules [17]. By treating molecules as sequences of characters, NLP models can be leveraged to analyze and generate molecular structures, enabling the exploration of chemical space using techniques inspired by language processing [18]. In the initial development of deep learning architectures for molecular generation, recurrent neural networks (RNNs) [19] were widely used with molecular SMILES representations [20,21]. These models were trained on extensive datasets of molecules and further refined through reinforcement learning [22,23] or transfer learning methods [21]. The goal was to generate molecules with specific properties and activities by guiding the model towards desired outcomes. These early approaches played a crucial role in advancing the field of deep learning for molecular generation and paved the way for subsequent research in the area. AE-based models have also played a significant role in molecular generation tasks used with SMILES. The molecule generative model based on VAE can generate various molecules with the required properties by learning the potential space following a specific probability distribution [24,25]. The AAE (adversarial auto-encoder) [26], on the other hand, incorporates adversarial training principles. By introducing a discriminator network, the AAE encourages the encoder to produce latent representations that closely resembled the true SMILES distribution [27]. This adversarial training improved the quality and diversity of generated molecules. generative adversarial networks (GANs) [28] have emerged as another popular approach for molecular design and generation used with SMILES representation. ORGANs (objective-reinforced generative adversarial networks) [29], as an early method of using the GAN model for molecular generation, guide the process of generation through the gradual construction process of generating molecules and through the use of reinforcement learning. However, the chemical feasibility of ORGAN-generated molecules is very low. LatentGAN [30] is a model that combines both an auto-encoder and a generative adversarial network for molecular generation. It starts by pre-training an auto-encoder using SMILES structures as input, and then trains a GAN to generate latent vectors for corresponding molecules. Similarly, LatentGAN faces the problem of poor validity and diversity of generated molecules.

Self-Referencing Embedded Strings (SELFIES) [11] represents a significant advancement in the field of molecular string representations, designed to address some of the limitations inherent in the traditional SMILES format. Compared to SMILES, SELFIES separates the information of branches and loops, improving the robustness of syntax. This robustness ensures that every generated string corresponds to a valid molecular structure, greatly benefiting machine learning applications in chemistry. This advancement is particularly notable in the context of machine learning applications for molecular generation. PASITHEA [31] is a SELFIES-based model for generating molecular structures, leveraging direct-gradient-based optimization techniques from computer vision. Researchers from the University of Toronto have developed STONED [32], a simple and efficient generative model that does not require training. This model is capable of enumerating structures and searching for transformative trajectories between molecules within a localized chemical space. SELFIES also has limitations, including potentially complex and less readable strings and higher computational demands for encoding and decoding processes.

Recently, there has been a lot of research on using the transformer [10] model to perform molecular generation tasks based on SMILES [33–35]. The transformer architecture comprises an encoder and a decoder for sequence tasks. It utilizes self-attention to capture dependencies and employs feed-forward networks. This design has powered advancements in natural language processing tasks like translation and generation. Generally speaking, researchers use a transformer encoder to encode molecule-related information, such as the three-dimensional structural information of molecules [34], molecule-related proteins [33], etc., and use a decoder to generate a SMILES. Of course, there are also cases where an encoder (Bert) is used alone for supervised learning tasks such as compound–protein interaction classification [36,37] or a decoder (GPT) is used alone for unconditional molecular generation [35]. MolGPT [35] has achieved high effectiveness in unconditional molecular generation, but its novelty is low, indicating severe overfitting of the model.

Besides enhancing the generation model, researchers are also exploring ways to more effectively use data. There are several methods to augment data for molecular generative models, and one of the most common approaches is SMILES enumeration [38]. SMILES enumeration means that a molecule can have multiple valid SMILES representations. Multiple SMILES representations of a molecule represent more comprehensive structural information of the molecule. Josep et al. [39] used randomized SMILES to expand data, effectively improving the molecular generative model compared to using canonical SMILES. Cheng-Kun Wu [40] also used SMILES enumeration to expand the dataset to improve the effectiveness of latent representation learning from molecules. However, the current SMILES enumeration method is only used for simple dataset expansion and has not been appropriately optimized for the concrete model.

In addition to simple data augmentation, researchers have recently used contrastive learning to learn better molecular representations for downstream tasks of molecules. The core of contrastive learning lies in constructing positive and negative sample pairs and utilizes the normalized temperature-scaled cross-entropy loss (NT-Xent) [41] to encourage the model to learn meaningful representations by maximizing agreement between positive pairs (augmented versions of the same image) and minimizing agreement between negative pairs (augmented versions of different images). Gathering positive instances typically encompasses various enhanced perspectives of identical data (such as data augmentation), while negative pairs commonly consist of the remaining samples within the mini-batch. Thus, the key to this matter is how to effectively enhance data. In the field of computer vision, approaches like SimCLR, proposed by Chen et al. [42], used image cropping, rotation, and other methods to enhance the data. In the textual domain, approaches like [43] utilize back-translation to enhance the data [44]. SimCSE [45] enters a sample into the model twice and then puts it through dropout twice, obtaining two different views that are mutually positive samples. In the molecular field, approaches like MolCLR [46] encompasses three molecular graph enhancement strategies for data enhancement: atomic masking, key deletion, and subgraph deletion. MoCL [47] combines the basis of ordinary graph data enhancement and domain knowledge to ensure that the representation of molecular graphs does not change during the enhancement process. SMICLR [48] uses different representations of molecules (SMILES representation and graph representation methods) as data augmentation for molecules.

We proposed a framework called CONSMI that utilizes the SMILES enumeration strategy as a data augmentation strategy for contrastive learning, and the trained representations achieved good results in self-supervised molecular generation and supervised compound–protein interaction prediction experiments. The molecular generation model is based on GPT, which effectively solves the problem of the overfitting of GPT-generated molecules compared to MolGPT [35]. The compound protein interaction model is based on a unified transformer and has achieved SOTA results on multiple datasets.

The contributions of this article are as follows:

1. We propose CONSMI, a contrastive learning framework that learns representation from a large molecular dataset.
2. The CONSMI framework combined with a transformer decoder generates more successful molecules.
3. The CONSMI framework combined with a transformer encoder achieves SOTA results on multiple datasets of compound–protein interactions.

2. Results and Discussion

In this section, we first demonstrate the performance of the molecular generation task. We compared the model with other state-of-the-art (SOTA) methods and conducted some interpretability analyses. Then, we demonstrated the performance of the model on some classification tasks. We refer to the GPT model with the CONSMI framework's pre-trained CONSMI embedding layer as CON-GPT, and indicate the two fine-tuning methods (frozen and unfrozen) after the model name. The models for comparative experiments have been introduced in Introduction.

2.1. Molecular Generation Results

As mentioned before, a good molecular generative model needs to generate more valid, unique, and novel molecules. Therefore, CON-GPT is evaluated in comparison to previous approaches using these evaluation criteria. Notably, JT-VAE utilizes graph representations as input, while the other approaches utilize SMILES.

The results on the Moses dataset are shown in Table 1. We conducted comparative experiments using the methods mentioned in the introduction: CharRNN, VAE, AAE, LatentGAN, JT-VAE, and MolGPT. Due to the fact that JT-VAE performs verification at every step of molecule generation, the validity of the model is 1. With the exception of JT-VAE, the MolGPT model achieves the highest validity score of 0.995 for molecule generation. However, its novelty score is only 0.781, indicating significant overfitting. It is important to note that both the validity and novelty of generated molecules are crucial. The CON-GPT model, whose SMILES embedding is pre-trained using the contrastive learning approach with the CONSMI framework, exhibits a validity score that is 0.04 lower compared to the MolGPT model. However, it achieves a higher novelty score of 0.834, indicating more valid and novel generated molecules. We found that the $IntDiv_1$ of the baseline method fluctuates around 0.855, while ours is around 0.850. Although the difference is not obvious, we found that GPT-based methods all have this problem, which is a point that we need to study in the future. The success rate of our CON-GPT exceeds that of all methods except JT-VAE and LatentGAN. LatentGAN has drawbacks related to GAN, such as unstable training and time consumption. This suggests that the CONSMI framework effectively learns the diversity of SMILES grammar, mitigates the overfitting issue observed in the MolGPT model, and provides a more improved deep molecular generative model based on SMILES.

We conducted experiments to compare two methods of fine-tuning the models: frozen (freezing the pre-training CONSMI embedding weights) and unfrozen (unfreezing the pre-training CONSMI embedding weights). We found that the freezing method outperforms the unfreezing method, despite our initial expectations. The CON-GPT model with unfrozen pre-training weights showed a slight decrease in validity (of 0.004) compared to the MolGPT model, but exhibited a slight increase in novelty (of 0.01). On the other hand, the CON-GPT model with frozen pre-training weights also experienced a decrease in validity (of 0.005) but showed a significant increase in novelty (of 0.053). Additionally, freezing the pre-training weights led to faster model training. These results demonstrate the strong feature extraction and generalization capabilities of our CONSMI framework.

Table 1. Comparison of the metrics for molecule generation using various approaches trained on the MOSES dataset.

Models	**Validity**	**Unique@10k**	**Novelty**	**Success Rate**	$IntDiv_1$
CharRNN	0.975	0.999	0.842	0.820	0.856
VAE	0.977	0.998	0.695	0.678	0.856
AAE	0.937	0.997	0.793	0.741	0.856
LatentGAN	0.897	0.997	0.949	0.849	0.857
JT-VAE	1.0	0.999	0.914	0.913	0.855
MolGPT	0.995	1.0	0.781	0.777	0.850
CON-GPT (unfrozen)	0.992	1.0	0.791	0.785	0.850
CON-GPT (frozen)	0.991	1.0	0.834	0.826	0.850

Our methods have been bolded.

In order to evaluate the generalization ability of our model, we conducted experiments on the GuacaMol dataset, which is a subset of the ChEMB dataset. The results on the GuacaMol dataset are shown in Table 2. It is worth noting that the pre-trained dataset and MOSES dataset used in our experiments are both subsets of the ZINC dataset. By assessing our model's performance on the GuacaMol dataset, we can learn how well it can generate molecules with desirable properties beyond the datasets it was specifically trained on. This evaluation provides valuable insights into the model's ability to generalize and produce high-quality molecules in diverse chemical spaces.

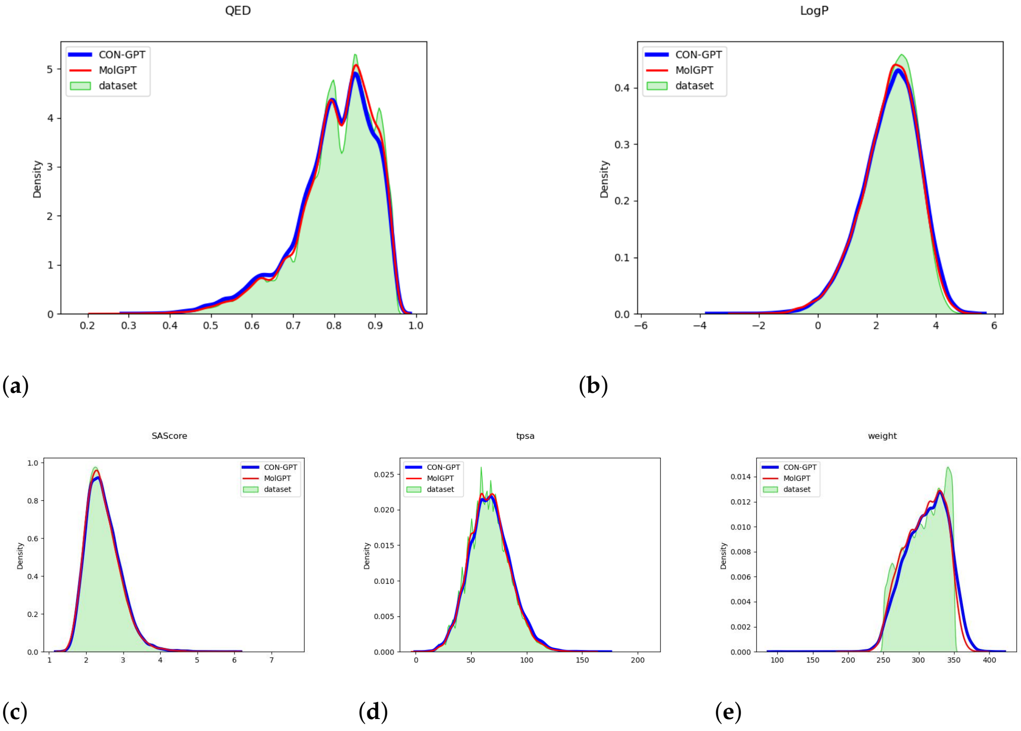
Table 2. Comparison of the metrics for molecule generation using various approaches trained on the GuacaMol dataset.

Models	**Validity**	**Unique**	**Novelty**	**Success Rate**
SMILES LSTM	0.959	1.0	0.912	0.875
VAE	0.870	0.999	0.974	0.847
AAE	0.822	1.0	0.998	0.820
MolGPT	0.979	0.998	0.958	0.936
CON-GPT (unfrozen)	0.968	0.999	0.968	0.936
CON-GPT (frozen)	0.961	0.999	0.975	0.936

Our methods have been bolded.

The experimental results on the GuacaMol dataset exhibit similarities to those obtained on the MOSES dataset. The GPT model enhanced with the CONSMI framework for pre-training demonstrates a higher uniqueness and novelty compared to the pure GPT model, albeit at the cost of a slight decrease in validity. The performance difference between the fine-tuning methods for unfrozen and frozen weights aligns with the observations on the MOSES dataset. Overall, there is still a slight advantage in generating valid and novel molecules, indicating a certain degree of generalization ability in our model. However, the leading advantage is not as pronounced as on MOSES, and we speculate that this may be due to the different sources of the GuacaMol dataset and our pre-trained dataset.

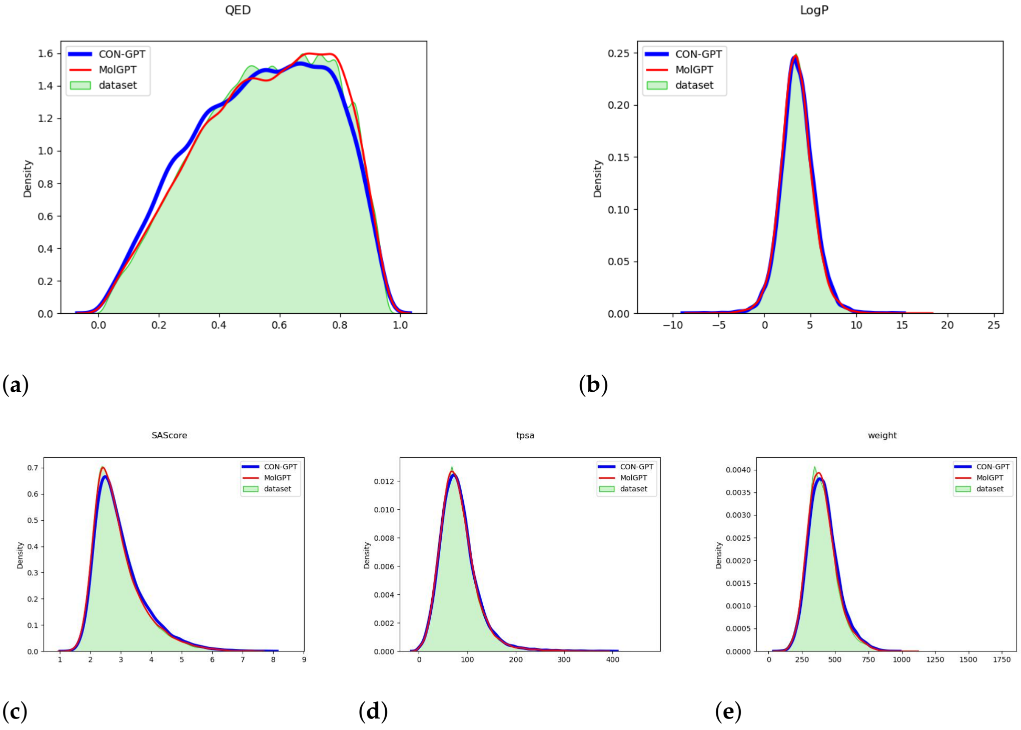
Figures 1 and 2 provide compelling evidence that the important molecular attributes (QED, LogP, SAScore, tpsa, weight) generated by the model closely match the distribution of the original dataset. These data were calculated by the RDKit library [49]. This result strongly suggests that the model has successfully learned the underlying distribution of molecular attributes present in the dataset.



(**a**) (**b**)

(**c**) (**d**) (**e**)

Figure 1. The distribution of molecular attributes generated by the model trained on the MOSES dataset. (**a**) QED (Quantitative Estimate of Drug-likeness). (**b**) LogP (Octanol-Water Partition Coefficient). (**c**) SAScore (Synthetic Accessibility Score). (**d**) TPSA (Topological Polar Surface Area). (**e**) Weight (Molecular Weight).



(**a**) (**b**)

(**c**) (**d**) (**e**)

Figure 2. The distribution of molecular attributes generated by the model trained on the GuacaMol dataset. (**a**) QED (Quantitative Estimate of Drug-likeness). (**b**) LogP (Octanol-Water Partition Coefficient). (**c**) SAScore (Synthetic Accessibility Score). (**d**) TPSA (Topological Polar Surface Area). (**e**) Weight (Molecular Weight).

Case Study We systematically examined all generated molecules with QED values exceeding 0.9 and assessed their similarity to the molecules in the Moses test set. As shown in

Table 3 and 4, We found that although the model has not seen the molecules in the test set, a substantial number of the generated molecules exhibited a similarity of over 0.9 to those in the test set.

Table 3. Example analysis of generated molecules: SMILES.

Number	Molecules Generated	Molecules in the Test Set
1	COc1ccc(NC(=O)N2CCN(C(=O)C3 CCCCC3)CC2)cc1	COc1ccc(NC(=O)N2CCN(C(=O)C3 CCCC3)CC2)cc1
2	Cc1nc2cc3c(cc2n1CC(=O)NC1CCCCC1) OCCO3	Cc1nc2cc3c(cc2n1CC(=O)NC1CCCC1) OCCO3
3	Cn1ccc(C(=O)Nc2cc(F)ccc2N2CCCC2)cc1=O	Cn1ccc(C(=O)Nc2cc(F)ccc2N2CCCCC2)cc1=O
4	Cc1cc(CN(C)C(=O)Nc2ccccc2N2CCCC2)no1	Cc1cc(CN(C)C(=O)Nc2ccccc2N2CCCCC2)no1
5	COc1ccccc1-c1noc(C(=O)N2CCCC2)c1N	COc1ccccc1-c1noc(C(=O)N2CCCCC2)c1N
6	CC(CC#N)N(C)C(=O)Nc1ccccc1N1CCCC1	CC(CC#N)N(C)C(=O)Nc1ccccc1N1CCCCC1

Table 4. Example analysis of generated molecules: molecular properties.

Number	Tanimoto Similarity	QED (Generated)	SAScore (Generated)	LogP (Generated)
1	0.958	0.916	1.839	2.952
2	0.957	0.940	2.318	2.565
3	0.957	0.941	2.184	2.377
4	0.956	0.950	2.102	3.247
5	0.952	0.935	2.284	2.168
6	0.952	0.925	2.685	3.053

Adjusting the Temperature We conducted an evaluation to understand how adjustments to the hyperparameter τ impact NT-Xent's ability to distinguish effectively between positive and negative samples. This assessment involved exploring a set of τ values that are frequently used in practice, specifically 0.05, 0.1, 0.5, and 1, to identify the most suitable τ value. The results, encapsulated in Table 5, display the validation errors corresponding to each τ value during the model's training process. We found that a τ of 0.1 yielded the most favorable results. Intriguingly, these results corroborate well with the existing literature, particularly concerning applications in molecular data [48].

Table 5. The impact of different τ values on generation tasks.

τ	0.05	0.10	0.50	1.00
Loss	0.16411	0.16326	0.16379	0.17299

2.2. Compound–Protein Interaction Results

In this section, we assessed the performance of the CONSMI architecture in the context of compound–protein interactions. In simpler terms, the task can be boiled down to a binary classification problem: determining whether there is an interaction between molecules and proteins or not. We used a transformer with two encoders and denoted our baseline model as UniT [50]. We substituted the original SMILES embedding layer with the trained CONSMI embedding Layer, referred to this as CON-UniT. We conducted comparisons between our model and some currently popular or highly effective models, such as GNN-CPI [51], TransformerCPI [36], Moltrans [37], and BCM-DTI [52]. GNN-CPI and TransformerCPI both utilize molecular graph representations of drugs and employ a CNN and a GCN (graph convolutional network), respectively, to encode the graphs. TransformerCPI and Moltrans, on the other hand, utilize transformer encoders to capture the chemical semantics. Additionally, Moltrans is a substructure-based DTI (drug–target interaction) prediction approach that applies byte pair encoding (BPE) to decompose drug and protein sequences into a set of explicit substructure sequences [52].

Tables 6 and 7 illustrate the notable performance improvements achieved by the UniT model with pre-training in compound–protein interaction classification tasks. The model exhibits a significantly higher F1 value, accuracy, and recall compared to the model without pre-training. Compared to the currently popular frameworks, our model excels in all aspects except for recall, where it does not always achieve the highest performance. However, it outperforms other models in terms of precision and demonstrates the remarkable ability to maintain a balance between precision and recall, even on the imbalanced DAVIS dataset [53]. This demonstrates the effectiveness of the pre-training approach in enhancing the model's ability to accurately classify compound–protein interactions. In addition, as seen in the table, a stand-alone transformer model does not necessarily outperform other models in terms of precision. However, the CON-UniT model demonstrates a significant competitive advantage. It is worth mentioning that the other models have undergone specific adjustments for the compound–protein interaction task, whereas our model achieved such impressive results with only the addition of pre-training to the original UniT. This observation further indicates that our CONSMI framework successfully learns meaningful representations of molecules. By leveraging the contrastive learning framework, our model is able to capture important features in the molecular data, leading to an improved performance in compound–protein interaction classification tasks. The effectiveness of the CONSMI framework highlights its ability to enhance the model's understanding and representation of molecular structures, thereby facilitating better predictions and classification accuracy.

Table 6. Experimental results of the compound–protein interaction classification task on the Celegans dataset.

Models	**F1**	**Precision**	**Recall**
GNN-CPI	0.933	0.938	0.929
TransformerCPI	0.952	0.952	0.953
Moltrans	0.954	0.947	0.962
BCM-DTI	0.969	0.967	**0.971**
UniT	0.964	0.966	0.961
CON-UniT	**0.969**	**0.972**	0.966

Top performed method in each metric is bold.

Table 7. Experimental results of the compound–protein interaction classification task on the DAVIS dataset.

Models	**F1**	**Precision**	**Recall**
GNN-CPI	0.658	0.647	0.669
TransformerCPI	0.584	0.46	0.8
Moltrans	0.306	0.185	**0.884**
BCM-DTI	0.611	0.853	0.476
UniT	0.841	0.844	0.837
CON-UniT	**0.868**	**0.874**	0.862

Top performed method in each metric is bold.

3. Methods

Here, we describe the method proposed in this work. We propose a contrastive learning pre-trained framework called CONSMI, which consists of a CONSMI embedding layer, a transformer encoder layer, and a projection head. We used the pre-trained CONSMI embedding layer for the molecular generation model CON-GPT. In order to demonstrate universality, we also used the CONSMI embedding layer for the classification model CON-UniT. CON-GPT is based on a transformer decoder, while CON-UniT is based on a transformer encoder.

3.1. CONSMI Framework

Here, we describe a contrastive learning framework using SMILES enumeration to learn more comprehensive potential representations of SMILES. As shown in Figure 3, (i) We first use the SMILES enumeration strategy to generate multiple different representations of a molecule. (ii) We then use a CONSMI embedding layer, a transformer encoder module, and a projection head with shared parameters to encode the input representations into latent space. (iii) We finally introduce a contrastive loss layer to calculate the contrastive loss in a batch of samples. The idea is to maximize the similarity of different SMILES representation vectors for the same molecule, while keeping the SMILES vectors of different molecules away from each other. The different SMILES enumeration representations of molecules can be obtained using the cheminformatics library RDKit [49].

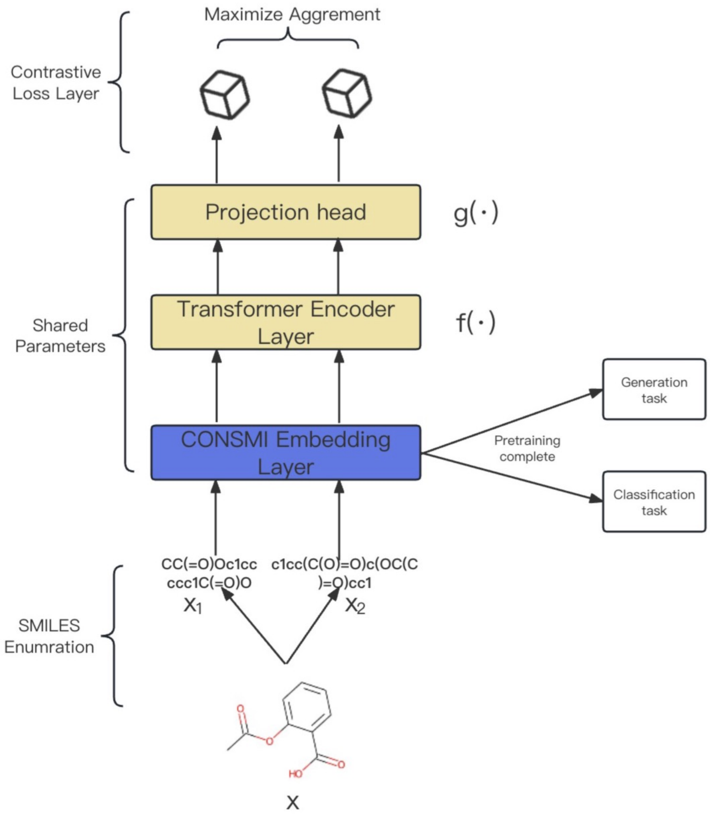


Figure 3. Overview of the CONSMI framework. Firstly, provide a molecule x. Obtain two SMILES representations representing x_1 and x_2 from x, and then input these together into the model. A transformer encoder module $f(\cdot)$ and a project head module $g(\cdot)$ are trained to maximize the agreement using a contrastive loss function. f includes several layers of transformer encoder layers, and g includes a multilayer perceptron (MLP).

For each input molecule x, we perform SMILES enumeration to randomly select two types x_1 and x_2. SMILES enumeration is completed using a function from the chemical informatics library RDKit [49]. As Figure 4 shows, the atomic order of molecules is disrupted when converting to the molfile format, so RDKit is used to generate SMILES from the molecules in the molfile. Changing the dorandom parameter to true will randomize the DFS transversal graph when generating SMILES.

The function $g(\cdot)$ within the projection head component takes the latent representation from the preceding module and maps it into an embedding space denoted as z. To implement the function $g(\cdot)$, we have adopted a multilayer perceptron (MLP) architecture which consists of a single hidden layer with rectified linear units (ReLUs) as the activation function, followed by a linear output layer. This type of architecture is commonly utilized in neural networks.

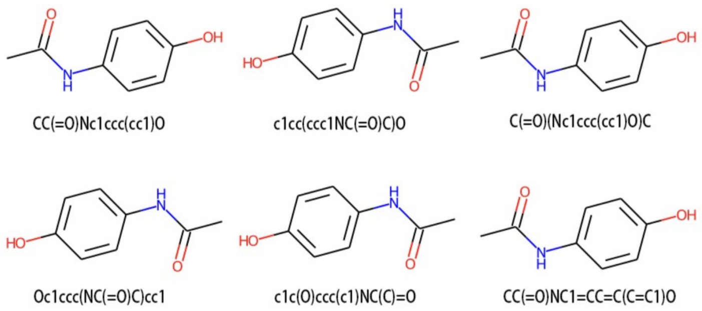


Figure 4. Visualization of different SMILES representations of the molecule acetaminophen, involving various transformations of its image. In principle, the DFS transversal graph that traverses all structures of acetaminophen has a certain degree of randomness. Since its images are generated according to specified rules by SMILES, each image is a different perspective from another images.

Following Chen et al. [42], this work adopted the normalized temperature-scaled cross-entropy loss (NT-Xent) shown below as $l_{i,j}$

$$l_{i,j} = -log\frac{exp(sim(z_i, z_j)/\tau)}{\sum_{k=1}^{2N} 1_{k \neq i} exp(sim(z_i, z_k)/\tau)}$$

where $sim(z_i, z_j) = z_i^T z_j / ||z_i||||z_j||$ (i.e., cosine similarity) and z_i and z_j are a positive pair (i.e., $g(r_i)$ and $g(r_j)$ of the same molecule). The function $1_{k \neq i}$ is an indicator function equal to 1 if $k \neq i$ (i.e., the negative pairs) and τ denotes the temperature parameter. In addition, since each molecule generates two different SMILES representations, the mini-batch includes $2N$ examples, which means there are $2(N-1)$ negative examples and 2 positive examples for each sample.

3.2. Generation Module

CON-GPT is composed of the generative pre-training transformer (GPT) model [54]. As shown in Figure 5, this module is built with N decoder blocks, where each block consists of a masked self-attention layer and a fully connected neural network. The self-attention layer produces a 256-dimensional vector, which serves as the input for the fully connected network. The hidden layer of the neural network generates a 1024-dimensional vector and applies the GELU (Gaussian error linear unit) activation function [55]. Its formula is as follows:

$$GELU(X) = x \times P(X \leq x) = x \times \phi(x), x \sim N(0,1)$$

where x is the input value, while X is a Gaussian random variable with zero mean and unit variance. $P(X \leq x)$ is the probability that X is less than or equal to the given value x. The final layer of the fully connected network outputs a 256-dimensional vector, which is subsequently fed into the next decoder block. The module uses auto-regressive patterns for generation. We first use the CONSMI framework for pre-training, and then bring the pre-trained CONSMI embedding layer over for molecular generation. After introducing pre-trained CONSMI embedding, we used two fine-tuning methods: The first is to freeze the CONSMI embedding weights and only train additional modules. The other option is to not freeze the CONSMI embedding weights and train them together with additional modules.

3.3. Classifier Module

CON-UniT is based on the unified transformer [50]. As shown in Figure 6, this module consists of two transformer encoders, one encoding the protein sequence and the other encoding the SMILES representation sequence of the molecule. After being encoded by the encoder, they are concatenated and mapped to the binary dimensional space. The internal structure of the encoder is similar to the decoder structure used in the generation module.

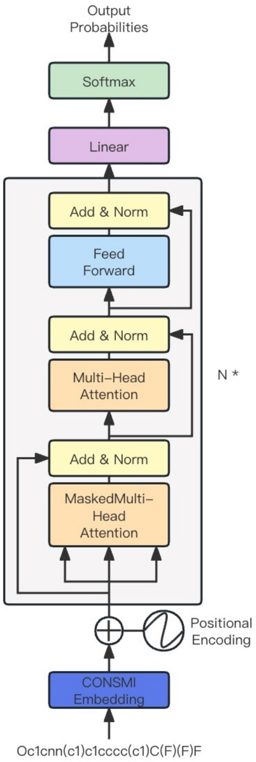


Figure 5. Overview of the generation module. N * means the module consists of a transformer decoder with N layers.

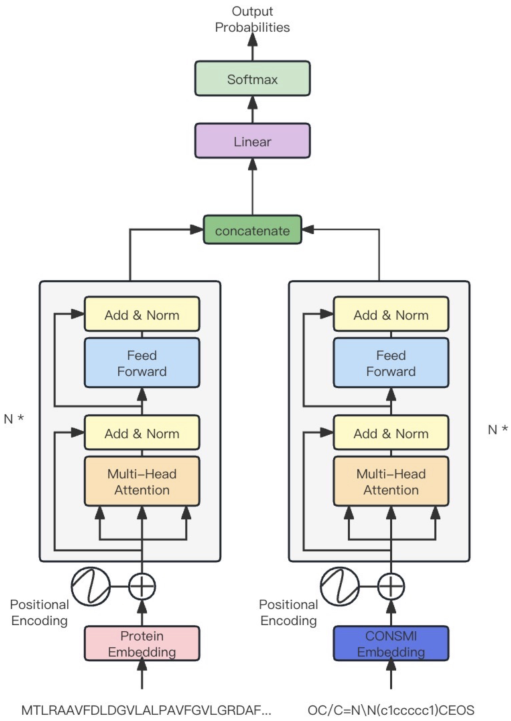


Figure 6. Overview of CON-UniT. N * means the module consists of two transformer encoders, each with N layers.

4. Experiment Configuration

In this section, we first introduce two datasets for molecular generation experiments and datasets for the compound–protein interaction (CPI) task. Then, we describe the process and evaluation indicators of the experiment. Finally, we provide a detailed overview of the CONSMI training process for SMILES representation learning, the molecule generation task, and the CPI task.

4.1. Datasets

We selected SMILES representations with molecular weights between 250 and 350, a logP not exceeding 3.5, and a SMILES sequence length not exceeding 100 from the ZINC Clean Leads [56] to form our pre-training dataset. There was a total of 37,956,795 SMILES sequences. We divided the training and testing sets randomly at 19:1. We hoped to pre-train molecules on a large dataset and learn more comprehensive molecular representations. MOSES [57] is a set of lead-like molecules extracted from the Zinc dataset, and its distribution is very similar to that of ideal drug molecules. We used the MOSES dataset to generate new drug-like molecules. GuacaMol [58] is a subset of the database ChEMBL [59] which contains 1.6 million molecules. It was used to verify the migration ability of the model on different molecular distributions.

We also used the Celegans [60] and DAVIS [53] datasets for experiments on compound–protein interactions. Every dataset was further divided into three sets with a ratio of 8:1:1 for training, validating, and testing, respectively.

4.2. Evaluation Metrics

The primary objective of our model is to generate a diverse set of molecules. To assess the quality of the generated molecules, we employed five distinct 2D-level metrics: Validity, Uniqueness, Novelty, Success Rate, and Internal Diversity. These metrics were used to evaluate and compare the structural integrity, uniqueness, and ability to introduce new chemical structures of generated molecules, as well as their diversity in the chemical space.

Validity was determined by utilizing RDKit's [49] molecular structure parser, which examines the valency of atoms and the consistency of bonds in aromatic rings. It assesses how accurately the generated molecules adhere to the rules and constraints of SMILES representations, ensuring proper atom connectivity and valence. Formally,

$$Validity = \frac{Number \quad of \quad valid \quad SMILES}{Number \quad of \quad generated \quad SMILES}$$

Uniqueness refers to the proportion of valid generated molecules that are distinct and not repetitive. A low uniqueness score indicates a higher frequency of duplicated or redundant molecules in the generated set. It reflects the model's ability, or lack thereof, to learn a diverse distribution of molecules during the generation process. In experiments based on the MOSES benchmark, we computed Unique@K and for the first K 10,000 valid molecules in the generated set.

$$Uniqueness = \frac{Number \quad of \quad distinct, \quad valid \quad SMILES}{Number \quad of \quad Valid \quad SMILES}$$

Novelty is defined as the proportion of generated molecules that do not exist in the training set. It measures the model's capability to produce new and unseen molecules that were not encountered during the training process. A low novelty score suggests a higher likelihood of overfitting, where the model predominantly reproduces molecules already present in the training set rather than generating novel compounds.

$$Novelty = \frac{Number \quad of \quad novel \quad SMILES \quad not \quad in \quad training \quad set}{Number \quad of \quad unique \quad generated \quad SMILES}$$

Success Rate is defined as the ratio of actual generation of available molecules, and from the perspective of unconditional generation of molecules, it should be the product of effectiveness, uniqueness, and novelty.

$$SuccessRate = Validity \times Uniqueness \times Novelty$$

Internal Diversity ($IntDiv_p$**)** was designed to measure the diversity of generated molecules and check for mode collapse or whether the model keeps generating similar structures. This involves calculating the mean power (p) of Tanimoto similarity (T) between fingerprints of all pairs of molecules ($s1$, $s2$) within the generated set (S).

$$IntDiv_p(S) = 1 - \sqrt[p]{\frac{1}{|S|^2} \sum_{s1,s2 \in S} T(s1,s2)^p}$$

For the classification task, we employed Precision, Recall, and the F1 score as evaluation metrics.

Precision measures the accuracy of the model's positive predictions. It is the proportion of correctly predicted positive samples out of all samples predicted as positive.

$$\text{Precision} = \frac{TP}{TP + FP}$$

where TP is the number of positive instances correctly predicted as positive by the model and FP is the number of negative instances incorrectly predicted as positive by the model.

Recall measures the coverage of positive samples by the model. It is the proportion of correctly predicted positive samples out of all actual positive samples.

$$\text{Recall} = \frac{TP}{TP + FN}$$

where FN is the number of positive instances incorrectly predicted as negative by the model.

F1 Score is the harmonic mean of precision and recall. It provides a comprehensive measure of a model's performance by considering both precision and recall.

$$\text{F1 Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

4.3. Training Details

The models were implemented based on Pytorch and trained on a GPU (Nvidia RTX3090) and a checkpoint was saved per epoch. We use the AdamW [61] optimizer, conducted a grid search in the interval of [0.0001, 0.01], and selected the value with the best performance in the validation set as the learning rate. We designated the target epoch as 100, and if there was no reduction in the loss of the validation set for 10 consecutive iterations, we saved the current model as the optimal one and concluded the training process. The batch size was 16, and the word vector dimension was 256. The parameters of the CON-GPT were consistent with those in MolGPT [35], so the data in the comparative experiment were directly used from the paper. We adopted the beam search procedure to generate multiple candidates. All generated candidates were canonicalized using RDkit and compared to the source molecules. The training settings for CON-UniT were the same as CONSMI, except that the batch size was changed to 128. The comparative data were collected from the paper on BCM-DTI [52].

5. Conclusions

In this work, we propose a contrastive learning pre-training framework called CONSMI, specifically designed for molecular SMILES representations. By leveraging SMILES enumeration as a data augmentation technique, we perform contrastive learning by using different SMILES representations of the same molecule as positive examples and different SMILES representations of different molecules as negative examples. The effectiveness of our framework is validated through experiments on the GPT model, showcasing its superior performance in molecular generation tasks.

The experimental results demonstrate that our pre-training framework significantly enhances the novelty and uniqueness of the generated molecules while maintaining a high level of validity. Our model is capable of generating more effective and novel molecules, while ensuring that their properties align with the distribution observed in the dataset. The evaluation conducted on both the MOSES dataset and the GuacaMol dataset further confirms the efficacy of our pre-training framework.

Moreover, our pre-training framework exhibits promising results in the compound–protein interaction task. The successful outcomes achieved across multiple datasets serve as further evidence that our framework facilitates the learning of improved molecular representations.

In summary, our proposed CONSMI framework contributes to the advancement of molecular generation tasks by enabling the generation of more effective and novel molecules while preserving their alignment with the dataset's molecular distribution. Additionally, the framework demonstrates its efficacy in the compound–protein interaction task, showcasing its ability to facilitate more comprehensive molecular representation learning.

Author Contributions: Conceptualization, methodology, Y.Q., M.S. and Q.Z.; validation, M.S.; formal analysis, Y.Q.; investigation, Q.Z.; writing—original draft, M.S.; writing—review and editing, Y.Q. and Q.Z.; supervision, Q.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The source code and datasets are available at https://github.com/WaterFlow1/CONSMI.git (accessed on 15 January 2024).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Lee, S.I.; Celik, S.; Logsdon, B.A.; Lundberg, S.M.; Martins, T.J.; Oehler, V.G.; Estey, E.H.; Miller, C.P.; Chien, S.; Dai, J.a. A machine learning approach to integrate big data for precision medicine in acute myeloid leukemia. *Nat. Commun.* **2018**, *9*, 42. [CrossRef]
2. Dimasi, J.A.; Grabowski, H.G.; Hansen, R.W. Innovation in the pharmaceutical industry: New estimates of R&D costs. *J. Health Econ.* **2016**, *47*, 20–33. [PubMed]
3. Polishchuk, P.G.; Madzhidov, T.I.; Varnek, A. Estimation of the size of drug-like chemical space based on GDB-17 data. *J. Comput.-Aided Mol. Des.* **2013**, *27*, 675–679. [CrossRef]
4. Sunghwan, K.; Thiessen, P.A.; Bolton, E.E.; Jie, C.; Gang, F.; Asta, G.; Lianyi, H.; Jane, H.; Siqian, H.; Shoemaker, B.A. PubChem Substance and Compound databases. *Nucleic Acids Res.* **2016**, *44*, D1202–D1213.
5. Yoshikawa, N.; Terayama, K.; Sumita, M.; Homma, T.; Oono, K.; Tsuda, K. Population-based de novo molecule generation, using grammatical evolution. *Chem. Lett.* **2018**, *47*, 1431–1434. [CrossRef]
6. Verhellen, J. Graph-based molecular Pareto optimisation. *Chem. Sci.* **2022**, *13*, 7526–7535. [CrossRef] [PubMed]
7. Lamanna, G.; Delre, P.; Marcou, G.; Saviano, M.; Varnek, A.; Horvath, D.; Mangiatordi, G.F. GENERA: A combined genetic/deep-learning algorithm for multiobjective target-oriented de novo design. *J. Chem. Inf. Model.* **2023**, *63*, 5107–5119. [CrossRef]
8. Creanza, T.M.; Lamanna, G.; Delre, P.; Contino, M.; Corriero, N.; Saviano, M.; Mangiatordi, G.F.; Ancona, N. DeLA-Drug: A deep learning algorithm for automated design of druglike analogues. *J. Chem. Inf. Model.* **2022**, *62*, 1411–1424. [CrossRef]
9. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets Advances in neural information processing systems. *arXiv* **2014**, arXiv:1406.2661.
10. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 1–11.
11. Krenn, M.; Häse, F.; Nigam, A.; Friederich, P.; Aspuru-Guzik, A. Self-referencing embedded strings (SELFIES): A 100% robust molecular string representation. *Mach. Learn. Sci. Technol.* **2020**, *1*, 045024. [CrossRef]
12. Lim, J.; Hwang, S.Y.; Kim, S.; Moon, S.; Kim, W.Y. Scaffold-based molecular design using graph generative model. *Chem. Sci.* **2020**, *11*, 1153–1164. [CrossRef] [PubMed]
13. Jin, W.; Barzilay, R.; Jaakkola, T. Junction tree variational autoencoder for molecular graph generation. In Proceedings of the International Conference on Machine Learning, PMLR, Stockholm, Sweden, 10–15 July 2018; pp. 2323–2332.

14. Yamada, M.; Sugiyama, M. Molecular Graph Generation by Decomposition and Reassembling. *ACS Omega* **2023**, *8*, 19575–19586. [CrossRef] [PubMed]
15. Yu, C.; Yongshun, G.; Yuansheng, L.; Bosheng, S.; Quan, Z. Molecular design in drug discovery: A comprehensive review of deep generative models. *Briefings Bioinform.* **2021**, *22*, bbab344.
16. Weininger, D. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *J. Chem. Inf. Comput. Sci.* **1988**, *28*, 31–36. [CrossRef]
17. Pathak, Y.; Laghuvarapu, S.; Mehta, S.; Priyakumar, U.D. Chemically interpretable graph interaction network for prediction of pharmacokinetic properties of drug-like molecules. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 873–880.
18. Chen, H.; Engkvist, O.; Wang, Y.; Olivecrona, M.; Blaschke, T. The rise of deep learning in drug discovery. *Drug Discov. Today* **2018**, *23*, 1241–1250. [CrossRef] [PubMed]
19. Jordan, M.I. Serial Order: A Parallel Distributed Processing Approach. *Adv. Psychol.* **1997**, *121*, 471–495.
20. Gupta, A.; Müller, A.T.; Huisman, B.J.H.; Fuchs, J.A.; Schneider, P.; Schneider, G. Erratum: Generative Recurrent Networks for De Novo Drug Design. *Mol. Inform.* **2018**, *37*, 1880141. [CrossRef]
21. Segler, M.H.; Kogej, T.; Tyrchan, C.; Waller, M.P. Generating focused molecule libraries for drug discovery with recurrent neural networks. *ACS Cent. Sci.* **2018**, *4*, 120–131. [CrossRef]
22. Popova, M.; Isayev, O.; Tropsha, A. Deep reinforcement learning for de novo drug design. *Sci. Adv.* **2018**, *4*, eaap7885. [CrossRef]
23. Olivecrona, M.; Blaschke, T.; Engkvist, O.; Chen, H. Molecular de novo design through deep reinforcement learning. *J. Cheminform.* **2017**, *9*, 1–14. [CrossRef]
24. Simonovsky, M.; Komodakis, N. GraphVAE: Towards Generation of Small Graphs Using Variational Autoencoders. In *Artificial Neural Networks and Machine Learning–ICANN 2018: 27th International Conference on Artificial Neural Networks, Rhodes, Greece, 4–7 October 2018*; Springer: Cham, Switzerland, 2018.
25. Jaechang, L.; Seongok, R.; Woo, K.J.; Youn, K.W. Molecular generative model based on conditional variational autoencoder for de novo molecular design. *J. Cheminform.* **2018**, *10*, 31.
26. Makhzani, A.; Shlens, J.; Jaitly, N.; Goodfellow, I.; Frey, B. Adversarial autoencoders. *arXiv* **2015**, arXiv:1511.05644.
27. Hong, S.H.; Lim, J.; Ryu, S.; Kim, W.Y. Molecular Generative Model Based On Adversarially Regularized Autoencoder. *J. Chem. Inf. Model.* **2019**, *60*, 29–36. [CrossRef] [PubMed]
28. Creswell, A.; White, T.; Dumoulin, V.; Arulkumaran, K.; Sengupta, B.; Bharath, A.A. Generative adversarial networks: An overview. *IEEE Signal Process. Mag.* **2018**, *35*, 53–65. [CrossRef]
29. Guimaraes, G.L.; Sanchez-Lengeling, B.; Outeiral, C.; Farias, P.L.C.; Aspuru-Guzik, A. Objective-reinforced generative adversarial networks (organ) for sequence generation models. *arXiv* **2017**, arXiv:1705.10843.
30. Prykhodko, O.; Johansson, S.V.; Kotsias, P.C.; Arús-Pous, J.; Bjerrum, E.J.; Engkvist, O.; Chen, H. A de novo molecular generation method using latent vector based generative adversarial network. *J. Cheminform.* **2019**, *11*, 1–13. [CrossRef]
31. Shen, C.; Krenn, M.; Eppel, S.; Aspuru-Guzik, A. Deep molecular dreaming: Inverse machine learning for de novo molecular design and interpretability with surjective representations. *Mach. Learn. Sci. Technol.* **2021**, *2*, 03LT02. [CrossRef]
32. Nigam, A.; Pollice, R.; Krenn, M.; dos Passos Gomes, G.; Aspuru-Guzik, A. Beyond generative models: Superfast traversal, optimization, novelty, exploration and discovery (STONED) algorithm for molecules using SELFIES. *Chem. Sci.* **2021**, *12*, 7079–7090. [CrossRef]
33. Grechishnikova, D. Transformer neural network for protein-specific de novo drug generation as a machine translation problem. *Sci. Rep.* **2021**, *11*, 321. [CrossRef]
34. Zheng, S.; Lei, Z.; Ai, H.; Chen, H.; Deng, D.; Yang, Y. Deep scaffold hopping with multimodal transformer neural networks. *J. Cheminform.* **2021**, *13*, 1–15. [CrossRef]
35. Bagal, V.; Aggarwal, R.; Vinod, P.; Priyakumar, U.D. MolGPT: Molecular generation using a transformer-decoder model. *J. Chem. Inf. Model.* **2021**, *62*, 2064–2076. [CrossRef]
36. Lifan, C.; Xiaoqin, T.; Dingyan, W.; Feisheng, Z.; Xiaohong, L.; Tianbiao, Y.; Xiaomin, L.; Kaixian, C.; Hualiang, J.; Mingyue, Z. TransformerCPI: Improving compound–protein interaction prediction by sequence-based deep learning with self-attention mechanism and label reversal experiments. *Bioinformatics* **2020**, *36*, 4406–4414.
37. Huang, K.; Xiao, C.; Glass, L.; Sun, J. MolTrans: Molecular Interaction Transformer for Drug Target Interaction Prediction. *Bioinformatics* **2021**, *37*, 830–836. [CrossRef] [PubMed]
38. Bjerrum, E.J. SMILES enumeration as data augmentation for neural network modeling of molecules. *arXiv* **2017**, arXiv:1703.07076.
39. Arús-Pous, J.; Johansson, S.V.; Prykhodko, O.; Bjerrum, E.J.; Tyrchan, C.; Reymond, J.L.; Chen, H.; Engkvist, O. Randomized SMILES strings improve the quality of molecular generative models. *J. Cheminform.* **2019**, *11*, 1–13. [CrossRef]
40. Wu, C.K.; Zhang, X.C.; Yang, Z.J.; Lu, A.P.; Hou, T.J.; Cao, D.S. Learning to SMILES: BAN-based strategies to improve latent representation learning from molecules. *Briefings Bioinform.* **2021**, *22*, bbab327. [CrossRef]
41. Hjelm, R.D.; Fedorov, A.; Lavoie-Marchildon, S.; Grewal, K.; Bachman, P.; Trischler, A.; Bengio, Y. Learning deep representations by mutual information estimation and maximization. *arXiv* **2018**, arXiv:1808.06670.
42. Chen, T.; Kornblith, S.; Norouzi, M.; Hinton, G. A simple framework for contrastive learning of visual representations. In Proceedings of the International Conference on Machine Learning (PMLR), Virtual Event, 13–18 July 2020; pp. 1597–1607.

43. Zhang, Y.; He, R.; Liu, Z.; Lim, K.H.; Bing, L. An unsupervised sentence embedding method by mutual information maximization. *arXiv* **2020**, arXiv:2009.12061.
44. Fang, H.; Wang, S.; Zhou, M.; Ding, J.; Xie, P. Cert: Contrastive self-supervised learning for language understanding. *arXiv* **2020**, arXiv:2005.12766.
45. Gao, T.; Yao, X.; Chen, D. Simcse: Simple contrastive learning of sentence embeddings. *arXiv* **2021**, arXiv:2104.08821.
46. Wang, Y.; Wang, J.; Cao, Z.; Barati Farimani, A. Molecular contrastive learning of representations via graph neural networks. *Nat. Mach. Intell.* **2022**, *4*, 279–287. [CrossRef]
47. Sun, M.; Xing, J.; Wang, H.; Chen, B.; Zhou, J. MoCL: Data-driven molecular fingerprint via knowledge-aware contrastive learning from molecular graph. In Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, Virtual Event, 14–18 August 2021; pp. 3585–3594.
48. Pinheiro, G.A.; Da Silva, J.L.; Quiles, M.G. Smiclr: Contrastive learning on multiple molecular representations for semisupervised and unsupervised representation learning. *J. Chem. Inf. Model.* **2022**, *62*, 3948–3960. [CrossRef] [PubMed]
49. Landrum, G. Rdkit documentation. *Release* **2013**, *1*, 4.
50. Singh, A.; Hu, R. UniT: Multimodal Multitask Learning with a Unified Transformer. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Virtual Conference, 11–17 October 2021.
51. Masashi, T.; Kentaro, T.; Jun, S. Compound-protein Interaction Prediction with End-to-end Learning of Neural Networks for Graphs and Sequences. *Bioinformatics* **2019**, *35*, 309–318.
52. Dou, L.; Zhang, Z.; Qian, Y.; Zhang, Q. BCM-DTI: A fragment-oriented method for drug–target interaction prediction using deep learning. *Comput. Biol. Chem.* **2023**, *104*, 107844. [CrossRef]
53. Davis, M.I.; Hunt, J.P.; Herrgard, S.; Ciceri, P.; Wodicka, L.M.; Pallares, G.; Hocker, M.; Treiber, D.K.; Zarrinkar, P.P. Comprehensive analysis of kinase inhibitor selectivity. *Nat. Biotechnol.* **2011**, *29*, 1046–1051. [CrossRef] [PubMed]
54. Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I. Language models are unsupervised multitask learners. *OpenAI Blog* **2019**, *1*, 9.
55. Hendrycks, D.; Gimpel, K. Gaussian error linear units (gelus). *arXiv* **2016**, arXiv:1606.08415.
56. Sterling, T.; Irwin, J.J. ZINC 15–ligand discovery for everyone. *J. Chem. Inf. Model.* **2015**, *55*, 2324–2337. [CrossRef]
57. Polykovskiy, D.; Zhebrak, A.; Sanchez-Lengeling, B.; Golovanov, S.; Tatanov, O.; Belyaev, S.; Kurbanov, R.; Artamonov, A.; Aladinskiy, V.; Veselov, M.; et al. Molecular sets (MOSES): A benchmarking platform for molecular generation models. *Front. Pharmacol.* **2020**, *11*, 565644. [CrossRef] [PubMed]
58. Brown, N.; Fiscato, M.; Segler, M.H.; Vaucher, A.C. GuacaMol: Benchmarking models for de novo molecular design. *J. Chem. Inf. Model.* **2019**, *59*, 1096–1108. [CrossRef] [PubMed]
59. Gaulton, A.; Bellis, L.J.; Bento, A.P.; Chambers, J.; Davies, M.; Hersey, A.; Light, Y.; McGlinchey, S.; Michalovich, D.; Al-Lazikani, B.; et al. ChEMBL: A large-scale bioactivity database for drug discovery. *Nucleic Acids Res.* **2012**, *40*, D1100–D1107. [CrossRef] [PubMed]
60. Liu, H.; Sun, J.; Guan, J.; Zheng, J.; Zhou, S. Improving compound–protein interaction prediction by building up highly credible negative samples. *Bioinformatics* **2015**, *31*, i221–i229. [CrossRef]
61. Loshchilov, I.; Hutter, F. Fixing weight decay regularization in adam. *arXiv* **2017**, arXiv:1711.05101.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.