International Journal of
Molecular Sciences

MDPI

Article

GraphGPT: A Graph Enhanced Generative Pretrained Transformer for Conditioned Molecular Generation

Hao Lu [†], Zhiqiang Wei [†], Xuze Wang, Kun Zhang and Hao Liu [*]

College of Computer Science and Technology, Ocean University of China, Qingdao 266100, China

[*] Correspondence: liu.hao@ouc.edu.cn

[†] These authors contributed equally to this work.

Abstract: Condition-based molecular generation can generate a large number of molecules with particular properties, expanding the virtual drug screening library, and accelerating the process of drug discovery. In this study, we combined a molecular graph structure and sequential representations using a generative pretrained transformer (GPT) architecture for generating molecules conditionally. The incorporation of graph structure information facilitated a better comprehension of molecular topological features, and the augmentation of a sequential contextual understanding of GPT architecture facilitated molecular generation. The experiments indicate that our model efficiently produces molecules with the desired properties, with valid and unique metrics that are close to 100%. Faced with the typical task of generating molecules based on a scaffold in drug discovery, our model is able to preserve scaffold information and generate molecules with low similarity and specified properties.

Keywords: molecular generation; generative pretrained transformer; graph neural networks

check for updates

Citation: Lu, H.; Wei, Z.; Wang, X.; Zhang, K.; Liu, H. GraphGPT: A Graph Enhanced Generative Pretrained Transformer for Conditioned Molecular Generation. *Int. J. Mol. Sci.* **2023**, *24*, 16761. https://doi.org/10.3390/ijms242316761

Academic Editor: Gennady Verkhivker

Received: 15 October 2023
Revised: 16 November 2023
Accepted: 23 November 2023
Published: 25 November 2023

1. Introduction

Drug discovery and development constitute a complex and challenging process demanding extensive time, resources, and domain expertise [1–4]. To address this issue, virtual screening methods have emerged as pivotal tools for efficiently identifying potential drug candidates [5–7]. Currently, the virtual screening library can be scaled up to the billion level and is gradually increasing [8–10]. However, since the lead compound does not meet certain essential drug properties, including topological polar surface area, lipophilicity, etc., subsequent molecular optimization is required, which may even lead to drug development failures. Therefore, condition-based molecular generation can not only build high-quality molecular virtual screening libraries but also plays an important role in lead compound optimization, which in turn, drives drug development.

Molecular generation models based on artificial intelligence (AI) have showcased immense potential and have become pivotal in de novo drug design [11,12]. These models encompass the Variational Autoencoder (VAE) [13], the Generative Adversarial Network (GAN) [14], and Reinforcement Learning (RL) [15]. The reward function of RL can quickly focus on a chemical region while causing instability in the model [16,17], e.g., LS-MolGen [18] and DrugEx v3 [19]. Transfer learning mitigates this problem but results in generated molecules that are similar to the molecules in the training set due to the rapid concentration on a certain target region [20]. Generative pretrained language models have exhibited tremendous potential across various domains [21–23]. Molecular generation methods based on language models have demonstrated impressive capabilities in SMILES [24] syntactic representation [25,26]. These methods grapple with a significant challenge: the inability to capture molecular topological information. Conventional sequence-to-sequence models often disregard the intrinsic spatial arrangement of atoms, resulting in an incomplete representation of molecular topology [27]. Transformer [28] and graph neural networks are currently combined in works like GraphTransformer [29] and the Structure-Aware Transformer (SAT) [30]. Nevertheless, unlike supervised learning methods, molecular generation models lack explicit structural information during the generation process [31]. Consequently, these models are unable to fully incorporate the structural features of molecules into the generation process, thereby limiting their capacity to generate molecules with specific topological properties.

In this work, we focus on the molecular generation task to produce molecules with designated properties, thereby expanding molecular screening libraries (Figure 1). We proposed a graph neural network augmented GPT model called GraphGPT. To compensate for the lack of topology of molecules characterized by SMILES, we introduce an innovative methodology that integrates graph-based representations into the sequence-to-sequence paradigm. This innovative approach enables our model to maintain and leverage crucial structural information during the molecular generation process. By amalgamating the advantages of graph-based representation and language modeling, our approach provides a more comprehensive and contextually enriched avenue for molecular generation.

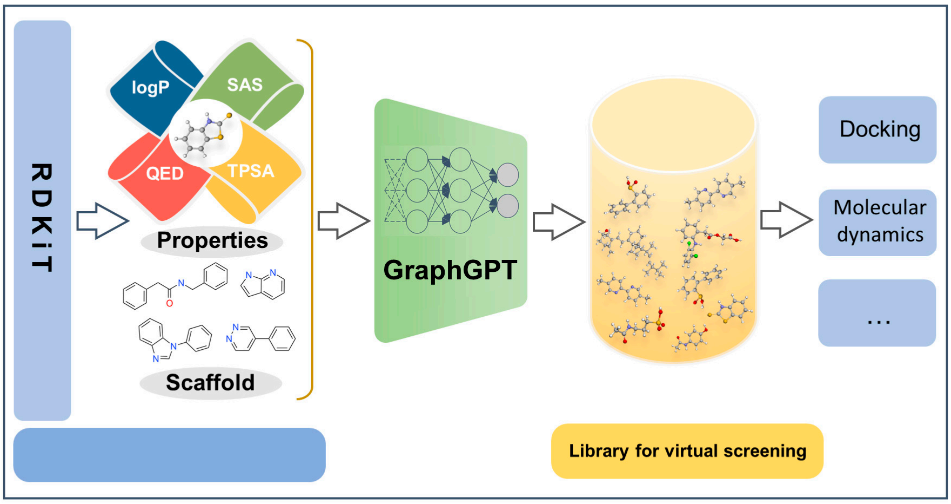

Figure 1. Condition-based molecular generation and downstream applications of virtual screening library.

2. Results

2.1. Molecular Generation Based on Properties

To test the ability of our model to generate molecules with a specific property, we specified a single property (Synthetic Accessibility Score (SAS)) [32], Quantitative Estimation of Drug-likeness (QED) [33], lipophilicity (logP) [34], and the Topological Polar Surface Area (TPSA) [35]) of the molecule to generate the molecule separately. We evaluated the performance of molecular generation using metrics that include validity, uniqueness, and novelty.

As shown in Table 1, we observed that both the MolGPT [36] and our model exhibit a relatively similar performance across the four properties. These metrics are close to one in terms of being valid and unique, indicating the capability of both models to generate high-quality molecular structures, with the majority of these molecules being unique. Furthermore, both models achieve a novelty value of one, indicating that the generated molecules are not present in the training dataset, thus mitigating overfitting concerns. Attention should also be paid to the standard deviation (SD) and mean absolute deviation (MAD) metrics. These metrics gauge the stability and consistency of the results. A smaller standard deviation and mean absolute deviation signify a more stable and consistent distribution of samples. It is evident that our model exhibits smaller standard deviations and mean absolute deviations in most SD/MAD metrics, indicating a greater stability compared to the MolGPT. This suggests that our model yields more consistent results across multiple experiments, displaying minimal performance fluctuations.

Table 1. Single property molecule generation, tested with the GuacaMol dataset.

	Model	Valid ↑	Unique ↑	Novelty ↑	SD ↓	MAD ↓
logP	MolGPT	**0.971**	0.998	1	0.31	0.23
	ours	0.969	0.998	1	**0.29**	**0.22**
TPSA	MolGPT	**0.972**	0.996	1	4.66	3.52
	ours	0.971	**0.997**	1	**4.21**	**3.31**
QED	MolGPT	**0.977**	0.995	1	0.20	0.13
	ours	0.968	**0.999**	1	**0.07**	**0.05**
SAS	MolGPT	0.975	**0.998**	1	0.20	**0.13**
	ours	**0.977**	0.996	1	**0.19**	0.14

In the context of generating a single property, our findings demonstrate (Figure 2) that the generated molecules exhibit desirable properties in terms of logP, SAS, and TPSA, affirming that the model effectively generates molecules with specific properties. It is noteworthy that some deviations were observed in the QED property, indicating a need for further refinement to ensure precise alignment with this metric. QED describes the drug-likeness of molecules, influenced by multiple underlying properties. As such, the model might face challenges in accurately controlling QED. These observations underscore the capability of our model in tailoring molecular properties while also highlighting the necessity to enhance its performance in generating molecules with specified QED.

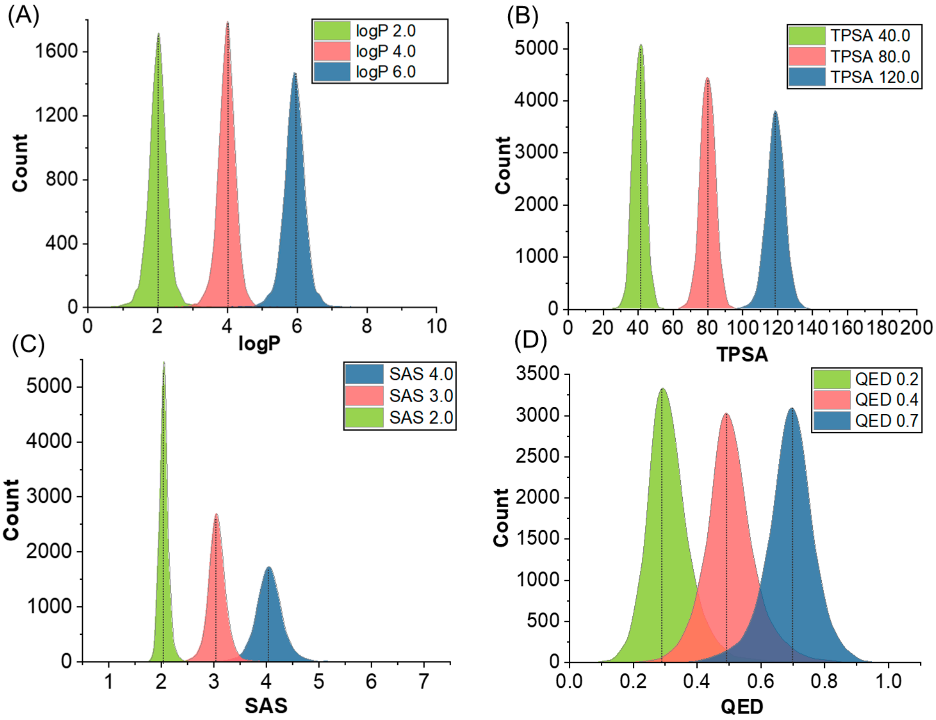

Figure 2. Generating property distribution from single-property molecules based on the model trained on the GuacaMol dataset; molecular properties include logP (**A**), TPSA (**B**), SAS (**C**), and QED (**D**). The legend in the upper right-hand corner represents the combinations of molecular properties that we defined.

Concurrently, we tested the performance of molecular generation based on multi properties. Table 2 and Figure 3 contain the evaluation results of two models, involving property and evaluation metrics such as validity, uniqueness, and novelty, and MAD/SD as a stability metric of the results. First of all, we can see that MolGPT and our model perform similarly under most combinations of properties, with scores very close to one for validity, uniqueness, and novelty, indicating that they are both capable of generating high-quality, unique, and novel molecular structures. MAD and SD are indicators used to assess the stability of molecular samples generated by our model. A smaller MAD and SD imply a more stable and consistent distribution of results. As can be seen from Table 2, the MAD and SD of our model are generally smaller for all combinations of properties. Our model is able to establish the relationship between molecular structure and molecular properties by integrating topological information, thereby demonstrating improved multi-property generation.

Table 2. Multi property molecule generation, tested with the GuacaMol dataset.

	Model	Valid ↑	Unique ↑	Novelty ↑	SD/MAD ↓		
					logP	SAS	TPSA
logP + SAS	MolGPT	**0.972**	**0.992**	1	0.340/**0.250**	0.210/**0.140**	
	ours	0.971	0.991	1	**0.331**/0.252	**0.208**/0.151	
SAS + TPSA	MolGPT	**0.971**	**0.988**	1		0.220/**0.150**	4.940/3.760
	ours	0.970	**0.988**	1		**0.217**/0.158	**4.705/3.647**
logP + TPSA	MolGPT	**0.965**	0.994	1	0.320/**0.240**		4.770/3.710
	ours	0.961	**0.995**	1	**0.314/0.240**		**4.575/3.607**
logP + TPSA + SAS	MolGPT	**0.973**	**0.969**	1	0.350/0.270	0.260/**0.180**	4.800/3.790
	ours	0.966	0.964	1	**0.335/0.259**	**0.247**/0.183	**4.461/3.532**

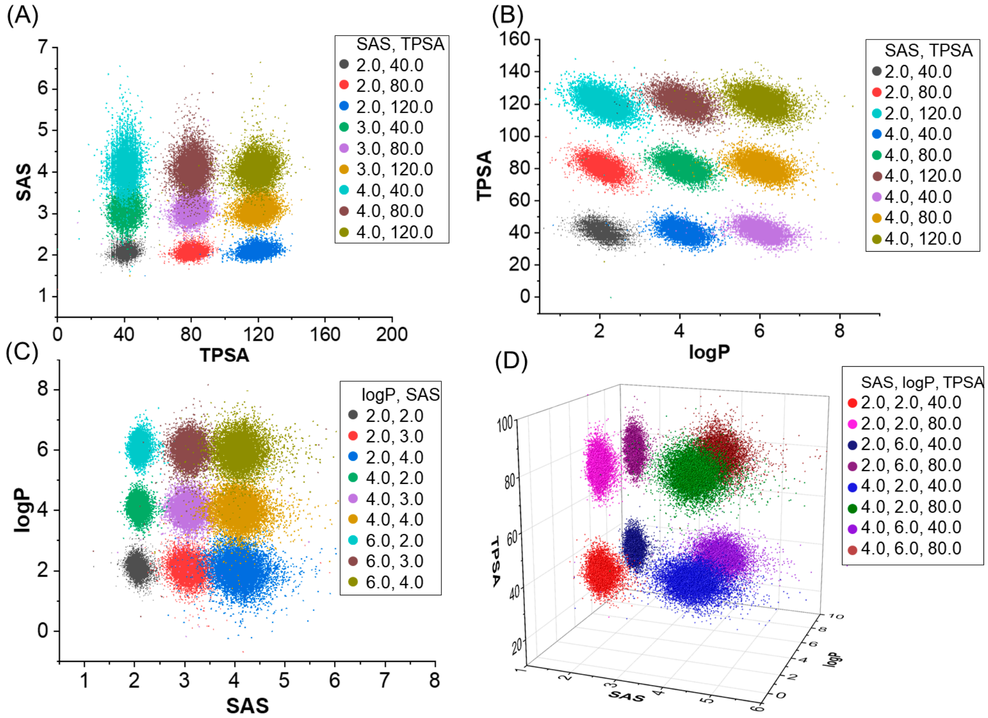

Figure 3. Distribution of molecular properties generated when specifying multiple properties using a model trained on the GuacaMol dataset. SAS and TPSA (**A**), logP and TPSA (**B**), SAS and logP (**C**), SAS, logP and TPSA (**D**). The legend in the upper right-hand corner represents the combinations of molecular properties that we defined.

In summary, although MolGPT and our model perform similarly in terms of validity, uniqueness, and novelty, our model is superior in terms of the stability of the results, i.e., the generated samples are more consistent and reliable across multiple experiments. It can be concluded that our model is better in this molecular generation task, and the generated molecular structures are not only of high quality but also show more stable performance in different specified properties experiments.

2.2. Molecular Generation Based on Properties and Scaffold

The molecular scaffold constitutes a pivotal element in drug design, influencing the structure, properties, and interactions of molecules. By strategically designing and modifying the molecular scaffold, drug molecules can attain specific biological activities, pharmacokinetics, and safety profiles, thereby laying a robust foundation for novel drug development. Therefore, we tested the performance of GraphGPT in generating molecules for a given scaffold and properties.

We adopted the five molecular scaffolds used by Bagal et al. [36], and tested the molecular generation based on these molecular scaffolds and single property. A boxplot was constructed by calculating the QED, logP, TPSA, and SAS of generated molecules, as illustrated in Figure 4A–D. Except for the presence of several outliers in the QED property, the molecular properties of the molecules generated based on the five molecular scaffolds were mostly within 1.5 of the interquartile range (IQR). Moreover, the means and medians of the various properties of the generated molecules were similar to those in the Moses dataset. Furthermore, Figure 4E indicates that the novelty of the molecules generated from the five molecular scaffolds exceeded 0.998, with valid samples of approximately 0.96. As shown in Section 2.4, the atom sizes of the molecular scaffold we used vary, with scaffold5 having up to 19 heavy atoms and scaffold2 having 9 heavy atoms. The uniqueness of the generated molecules based on both scaffold and single property exceeds 0.5, except for scaffold5. The unique metric was above 0.564 for the four sets of generated molecules, whereas scaffold5 exhibited a uniqueness value of 0.103. This difference of 0.65 compared to the corresponding indicator for scaffold2 might be attributed to the larger molecular size of scaffold5, resulting in a smaller space for the generated molecules.

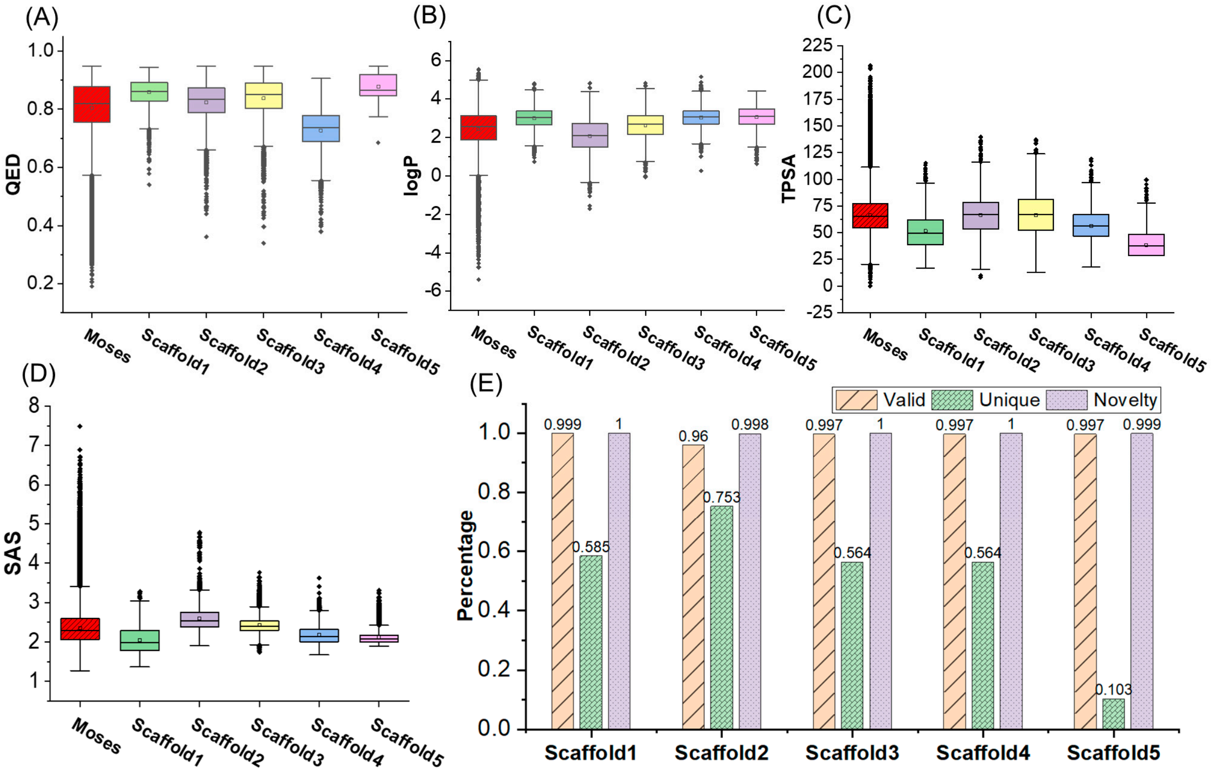

Figure 4. Properties of generated molecules based on five scaffolds and a single property. (**A**–**D**) are box-and-line plots for generated molecules with specified QED, logP, TPAS, and SAS, respectively. (**E**) is validity, uniqueness, and novelty of the generated molecules.

We visualize the properties of the generated samples in the scaffold and single property-based molecular generation experiments. As can be seen in Figure 5, the model is largely able to generate molecules according to a specified scaffold and a single property.

Similar to the single property generation experiment without a scaffold, the molecules generated with the specified QED are more dispersed. In the case of specifying the scaffold and three properties (Figure 6), the quality of molecule generation is poorer under SAS: 1.0, logP: 2.0, and TPSA: 40.0, which may be due to the lower property coverage of these properties for the training set molecules. The molecule generation results were better in other cases.

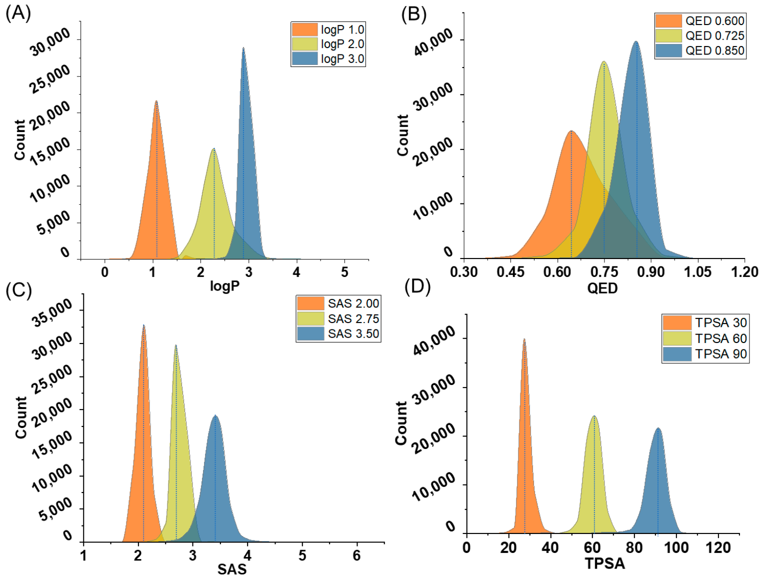

Figure 5. Distribution of molecular properties for molecular generation based on five molecular scaffolds and a single property, (**A**–**D**) are distribution of generated molecules specifying logP, QED, SAS and TPSA, respectively. The legend in the upper right-hand corner represents the combinations of molecular properties that we defined.

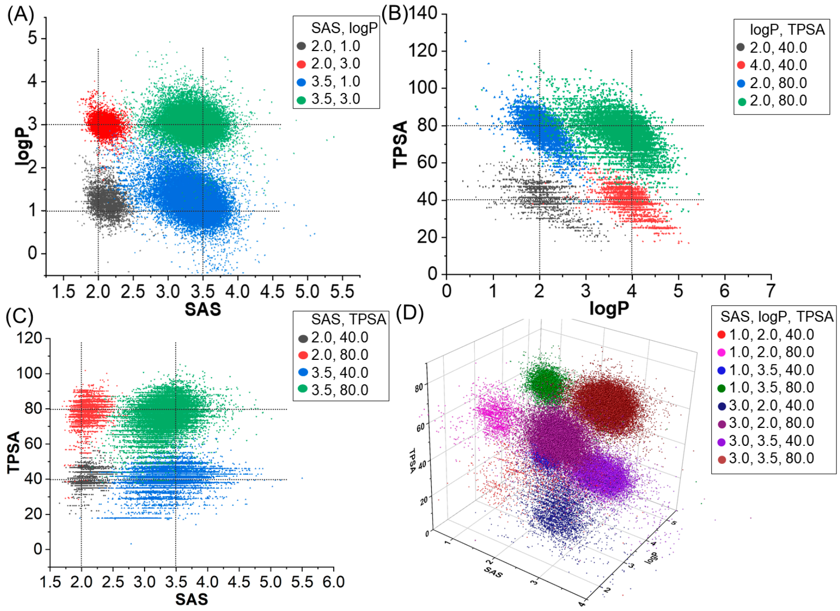

Figure 6. Molecular generation samples of multiple properties and scaffold. (**A**) SAS, LogP and scaffold. (**B**) LogP, TPSA, and scaffold. (**C**) SAS, TPSA, and scaffold. (**D**) SAS, LogP, TPSA, and scaffold. The legend in the upper right-hand corner represents the combinations of molecular properties that we defined.

2.3. Unconditional Molecular Generation

We conducted unconditional molecular generation on the GuacaMol [37] dataset, sampling 10,000 molecules and calculating various metrics for the generated molecules. The FCD [38] metric of the molecules significantly exceeded that of other models, reaching 1.009 (Table 3). Additionally, the KL divergence [39] metric matched that of the MolGPT model at 0.992, suggesting that it had a strong grasp on the training data distribution. While the validity of the generated molecules experienced a slight decrease, both uniqueness and novelty levels remained high. This indicates that the model, while capable of generating high-quality molecules, also learned the statistical characteristics of the trained molecules.

Table 3. Sampling of 10,000 molecules with different metrics for unconditional molecule generation based on a model trained on the GuacaMol dataset.

	Valid ↑	Unique ↑	Novelty ↑	FCD ↑	KL Divergence ↑
SMILES LSTM	0.959	**1**	0.912	0.913	0.991
AAE	0.822	**1**	0.998	0.529	0.886
Organ	0.379	0.841	0.687	0	0.267
VAE	0.87	0.999	0.974	0.863	0.982
MolGPT	**0.981**	0.998	**1**	0.907	**0.992**
ours	0.975	0.999	**1**	**1.009**	**0.992**

As shown in Table 4, when trained on datasets like the Moses dataset [40] containing drug-like small molecules, the model demonstrated a slight improvement in the validity metric while experiencing marginal reductions in uniqueness and novelty. The slight decrease in novelty is attributed to the model's improved ability to learn a more accurate representation of the molecules in the dataset, as a result of incorporating topological information. This leads to the generation of molecules that more closely resemble those in the training set. The IntDiv1 and IntDiv2 metrics for molecule diversity saw an increase. This demonstrates that our model can generate high-quality molecules in an unconstrained scenario.

Table 4. Sampling of 10,000 molecules with different metrics for unconditional molecule generation based on a model trained on the Moses dataset.

	Valid ↑	Unique ↑	Novelty ↑	IntDiv1 ↓	IntDiv2 ↓
charRNN	0.975	0.999	0.842	0.856	0.850
VAE	0.977	0.998	0.695	0.856	0.850
AAE	0.937	0.997	0.793	0.856	0.850
LatentGAN	0.897	0.997	**0.949**	0.857	0.850
JT-VAE	**1**	0.999	0.914	0.855	0.849
MolGPT	0.994	**1**	0.797	0.857	0.851
ours	0.995	0.999	0.787	**0.851**	**0.845**

2.4. Case Study

We employed the model trained on the Moses dataset for molecular generation, predefining the molecular scaffold, logP, and TPSA before generating molecules. Five molecular scaffolds were used to test the performance of molecular optimization. LogP was set to 2.0, and TPSA was set to 40.0. In other words, we aimed to generate molecules that retained their molecular scaffolds while achieving logP and TPSA values close to 2.0 and 40.0, respectively. Partially sampled molecules are illustrated in Figure 7; it can be observed that the generated molecules preserved the specified molecular scaffold while approximating the predetermined properties. Hence, our model can achieve molecular generation based on both molecular scaffold and properties.

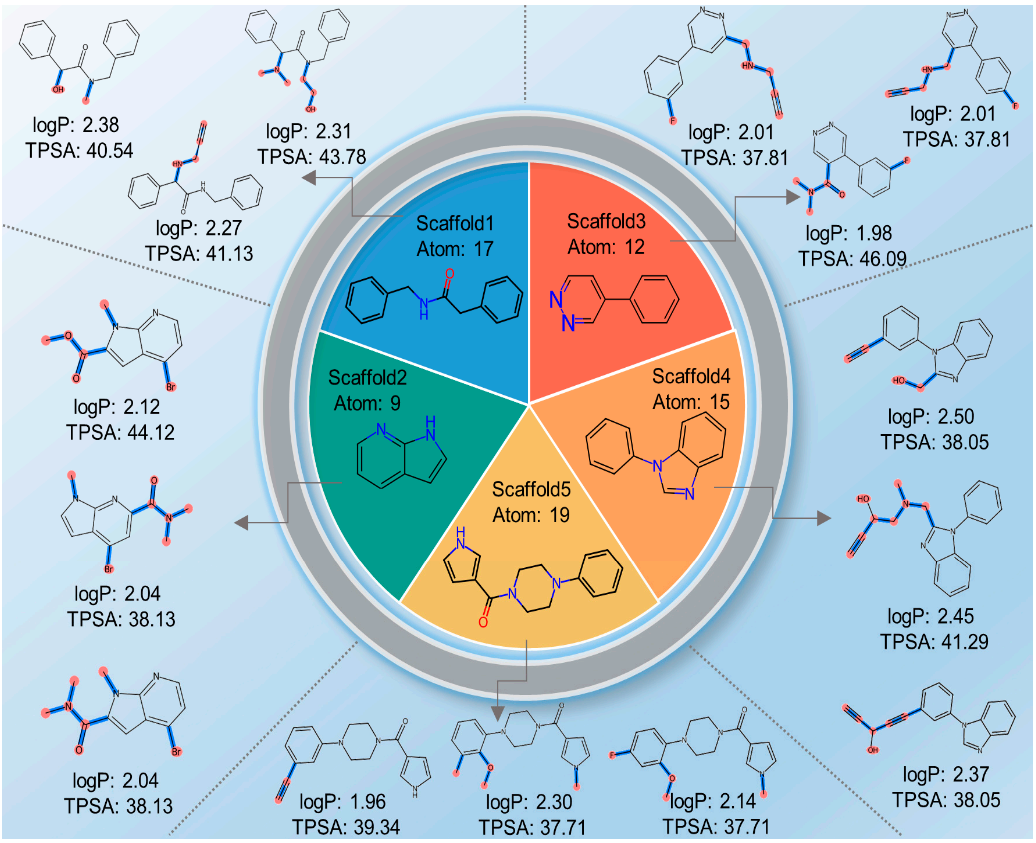

Figure 7. Samples of the generated molecules based on the five molecular scaffolds and properties (logP: 2, TPSA: 40.0). The generated bonds are represented in blue, and the atoms in light red.

2.5. Ablation Experiment

To verify the effect of different decoder layers as well as graph encoders on miniGPT, we performed ablation experiments. As shown in Table 5, three values of 40, 80, and 120 are set in the specified TPSA, and it can be seen that there is little difference in the validity, uniqueness, and novelty metrics. As the number of layers in the decoder increases, the standard deviation and MAD gradually become smaller, indicating the model's ability to simulate the properties of SMILES (the top three rows of Table 5). Considering that as the number of decoder layers increases the molecular validity, uniqueness, and novelty are close to one, we added graph encoders to the model, with eight decoders for runtime as well as efficiency reasons. The addition of the graph encoder reduces SD by 0.266 and MAD by 0.178 under the condition that all eight decoders are used (miniGPT_b and GraphGPT). The validity of the molecule is reduced by 0.001, which is acceptable for the sake of coincidence. After adding the graph encoder, the standard deviation of the numerator is even smaller, proving the effectiveness of the graph encoder.

Table 5. Impact of different number of decoder layers and graph encoders on the model.

	Graph Encoder	Decoder Layer	Valid ↑	Unique ↑	Novelty ↑	SD ↓	MAD ↓
miniGPT_a	×	4	0.946	0.999	1	5.017	3.806
miniGPT_b	×	8	0.972	0.997	1	4.474	3.485
miniGPT_c	×	12	0.977	0.996	1	4.240	3.238
GraphGPT	√	8	0.971	0.997	1	4.208	3.307

2.6. Attention Visualization

Figure 8 depicts an attention heatmap between tokens in the final layer of the model before and after encoding with a graph. The attention heatmap offers a visual interpretation of the representation of SMILES by the model. It is evident that, during the characterization of the molecule using GraphGPT, all tokens in the first part of the structure ("Cc1ccccc1") are strongly focused on the first "c", except for "C". This results in the formation of toluene and the presence of a significant number of benzene ring structures in the drug, which is consistent with chemical knowledge. Furthermore, the model places greater emphasis on the non-atomic tokens such as "1" and "-". This indicates the significance of accessory tokens in SMILES for the process of molecular generation. This behavior may be due to the model searching for signs of the conclusion of a functional group or other factors.

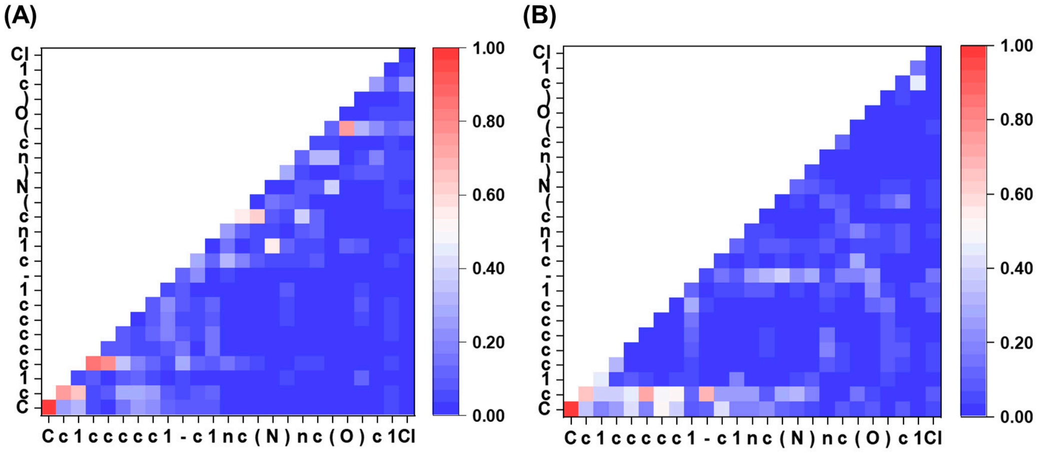

Figure 8. Attention heatmap between tokens in the last decoder layer. (**A**) Using the sequence encoder only, and (**B**) using GraphGPT.

3. Discussion

We use a GAT-based graph encoder to encode the molecular structure in order to solve the problem of missing topology in the sequence-based molecular generation process. The alignment of the graph encoder with the sequence encoder enables the fusion of structural information into the sequence encoder. Experiments demonstrate that the model is able to generate more accurate molecular properties with little degradation in performance such as molecular validity. There are also variants of SMILES such as SELFIES [41], R-SMILES [42], etc., which encode molecules, and in the future, we can try to use this sequence information for molecule generation. In addition, we did not try to take information such as target activity into account, which is the main area for our future research.

4. Methods and Materials

In this section, the overarching architecture of the model is initially presented. Subsequently, the structure of the graph-based encoder is expounded upon. Following this, the encoding of molecular scaffolds, properties, and molecular SMILES sequences are delineated through the utilization of a sequence encoder akin to the GPT framework. Lastly, the employed loss function is elucidated, facilitating the realization of molecular sequence generation fortified by graph-enhanced structures.

4.1. Overview of the Model

A sequence-to-sequence molecular generation model enhanced with graph structures was proposed. This model was capable of generating molecules based on molecular scaffolds and properties. As illustrated in Figure 9, the approach commenced by employing a graph-based encoder to encode the molecular structures, with the aim of capturing the inherent graph-related information of the molecules and thereby creating molecular encoding inclusive of graph-based structures. Subsequently, employing techniques from natural language processing, the molecules were represented using SMILES, facilitating the extraction of relationships between molecular properties and sequences, as well as capturing the syntactical format of SMILES. Lastly, a loss function was formulated and devised to seamlessly integrate the structure of molecules within the sequential data, thereby accomplishing the amalgamation of structural insights into the sequence generation process.

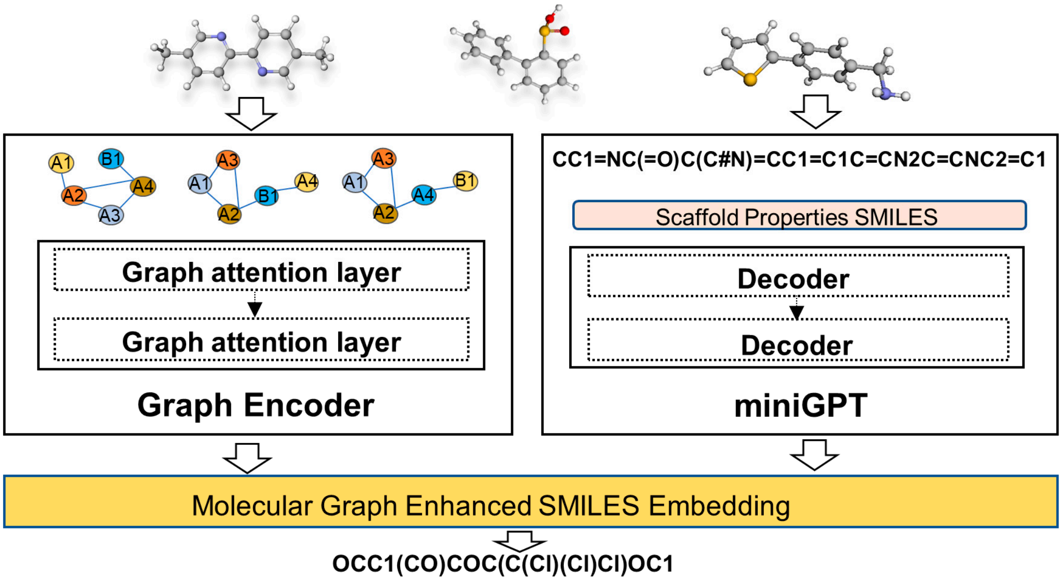
A1
B1
A2
A4
A3
A3
A1
B1
A4
A2
A3
A1
B1
A4
A2
Graph attention layer
Graph attention layer
Graph Encoder
CC1=NC(=O)C(C#N)=CC1=C1C=CN2C=CNC2=C1
Scaffold Properties SMILES
Decoder
Decoder
miniGPT
Molecular Graph Enhanced SMILES Embedding
OCC1(CO)COC(C(Cl)(Cl)Cl)OC1

Figure 9. Architecture of GraphGPT.

4.2. Graph Encoder

To address the deficiency in capturing the previous molecular topology within the molecular generation model, we introduced the Graph Attention Mechanism (GAT) [43] for encoding the structural aspects of molecules. As depicted in Figure 10, the atoms of a molecule were conceptualized as nodes within a graph, while the chemical bonds were construed as edges, collectively forming a graph-based representation of the entire molecule. Transformer-based molecular encoding calculates the impact of all atoms on the current atom. This results in even distant and unimportant atoms contributing to the representation of the current atom. As shown in Figure 10B, graph attention networks only consider the influence of directly linked atoms on the current atom, helping the model focus on more important atoms. Through the construction of molecular graphs and the utilization of the GAT model to learn the relationships between atoms, a more comprehensive grasp of the molecular topology could be achieved. During the training process, for each atomic node, GAT dynamically adjusted the weights based on the strengths of connections with neighboring atoms, thereby directing heightened attention toward atoms that bore more relevance. Consequently, the GAT model adeptly accentuated crucial connectivity patterns within the molecular representation, concurrently disregarding less significant elements, and thereby facilitating a more effective expression of the molecular topology.

We commenced by employing one-hot encoding to represent atomic attributes, encompassing atom type, degree, amount of hydrogen, and implicit valence. Subsequently, the interatomic relationships within the molecule were established, employing a two-layer graph attention for the encoding of graph structure. As expressed in Equation (1), the one-hot tensor of each atom was subjected to a linear transformation, projecting it into a higher-dimensional space. Here, h_j signifies the feature representation of atom j, W denotes the transformation matrix, and $||$ denotes the concatenation of the features of atom i and atom j. Following the application of the *LeakyReLU* activation function, the resultant $e_{i,j}$ emerged, representing the weight indicative of the influence of atom j on atom i. Equation (2) encapsulated the process of aggregating all connected atoms to atom i and subsequently normalizing the aggregated values. Incorporating a multi-head attention mechanism, as depicted in Equation (3), the elevated-dimensional feature representation of atom i was updated. The Sigmoid function *(σ)* was employed as the activation function, yielding the refined atom feature h'_j, which was subject to the dropout function. Following the passage of the molecule through the two layers of graph attention networks, a *ReLU* activation function was applied, succeeded by a global maximum pooling operation. It is pertinent to note that our graph encoder was exclusively operational during the training phase; during inference, its functionality was suspended due to the unavailability of molecular graph structural information.

$$e_{ij} = LeakyReLU(\alpha^T[Wh_i||Wh_j]) \tag{1}$$

$$a_{ij} = \frac{exp(e_{ij})}{\sum_{k=1}^{k\in N_i} exp(e_{ik})} \tag{2}$$

$$h'_i = \prod_{k=1}^{K} \sigma\left(\sum_{j\in\mathcal{N}_i} \alpha_{ij}^k W^k \vec{h}_j\right) \tag{3}$$

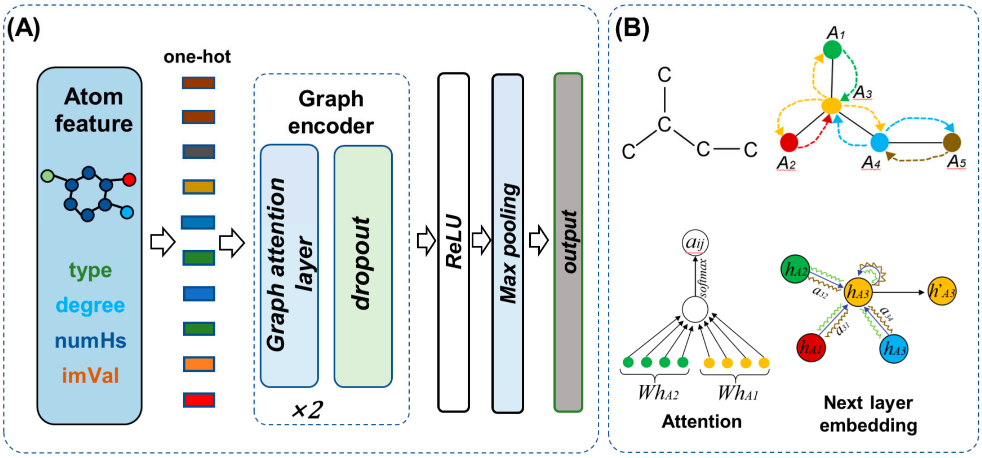

Figure 10. Details of graph encoder. (**A**) Graph encoding of molecules, type, degree, NumHs, and imVal refer to atom type, degree, amount of hydrogen, and implicit valence, respectively. (**B**) Graph attention mechanism.

4.3. GPT encoding of SMILES and Properties

The molecules were portrayed as sequences in the form of SMILES, serving as the input for the model. Additionally, RDKit [44] was utilized to procure molecular descriptors such as SAS, QED, logP, and TPSA. During the training phase, the molecular properties were concatenated with the respective molecular SMILES, enabling the model to discern the intricate associations existing between molecular properties and SMILES (Figure 11). In the inference stage, predefined molecular properties were specified, facilitating the achievement of conditional molecular generation.

In fact, our model employed a decoder module akin to the Transformer architecture, comprising 8 stacked decoders: a design reminiscent of architectures found within the GPT series. A comparison between the model we utilized and GPT-1 is provided in Table 6. For the sake of simplicity and efficiency, the number of decoder layers in the model as well as the attention header are compressed. An attention mechanism was employed to discern the influence of individual characters within the SMILES, thereby facilitating the feature updates. The calculation methodology for the attention mechanism is outlined in Equation (4), where Q, K, and V represent the Query, Key, and Value vectors, respectively, T represents the transpose, and d_k represents the Key vector dimension. After the sequence representation through the class GPT, the representation of the sequence is mapped to the same space as the representation of the graph encoder through a mapper, which is used with a fully connected representation in this study.

$$\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{4}$$

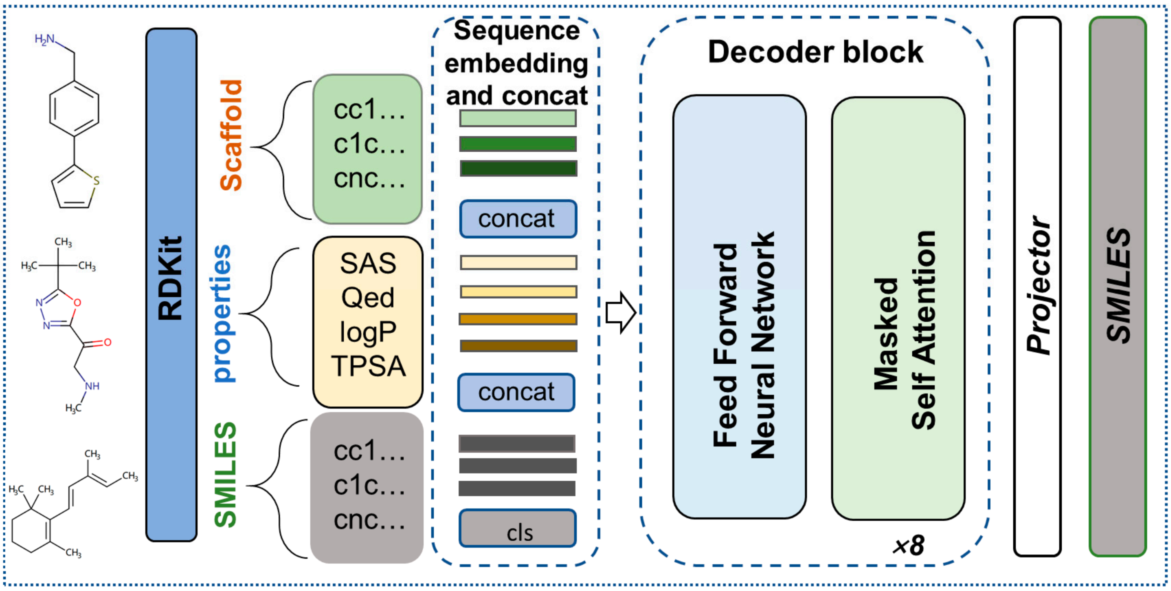

Figure 11. MiniGPT encoding of SMILES; properties and scaffold of molecules.

Table 6. Comparison of GPT-1 and GraphGPT.

	GPT-1	GraphGPT
Decoder layer	12	8
Attention header	12	8
Dimensions of vocab	768	256
Sequence length	512	100
Parameter	117 million	7.07 million

4.4. Optimum Objectives

In order to fuse the topological information, we employ a GAT in the molecule generation task and subsequently generate molecules with specific properties. As demonstrated in Equation (5), our formulated loss function encompasses two distinct components. The first loss function, denoted as L_{BT}, encapsulates the disparity between the molecular representation post-traversal through the graph encoder and SMILES sequence encoder. Given that these representations emanate from differing perspectives in molecular characterization, the features resulting from the graph encoder and the sequential encoder should exhibit close proximity. We constrained sequential encoding by employing the graph-encoded representation, thereby enabling the model to indirectly glean topological structural information inherent to the molecule. The second loss function, denoted as L_{ground}, encapsulates the divergence between the molecular predictions yielded by the model and the actual molecular structures. As depicted in Equation (6), the computation approach for measuring the gap between the graph structure encoding and the sequence encoding adopts the Barlow Twins loss function [45]. Here, λ serves as a hyperparameter, set to 0.005, and C_{ij} represents the correlation coefficient between the graph structure encoding and the sequence encoding, as calculated in Equation (7). Within these equations, b denotes a batch index, while i and j, respectively, index the graph encoder and sequence encoder. A and B correspond to the graph encoder and sequence encoder, respectively.

$$L = L_{\mathcal{BT}} + L_{groud} \tag{5}$$

$$L_{\mathcal{BT}} = \sum_i (1 - \mathcal{C}_{ii})^2 + \lambda \quad \sum_i \sum_{j \neq i} \mathcal{C}_{ij}{}^2 \tag{6}$$

$$\mathcal{C}_{ij} = \frac{\sum_b z^A_{b,i} z^B_{b,j}}{\sqrt{\sum_b \left(z^A_{b,i}\right)^2} \sqrt{\sum_b \left(z^B_{b,j}\right)^2}} \tag{7}$$

4.5. Dataset

We tested GraphGPT for the molecular generation task using two datasets: the GuacaMol dataset and the MOSES dataset. The GuacaMol dataset comprises a subset of 1.6 million molecules from the ChEMBL 24 database [46]. The molecular properties, such as molecular weight, LogP, and the number of rotatable bonds, exhibit heterogeneous distributions within this dataset. The MOSES dataset, containing 1.9 million lead-like compounds derived from the ZINC database [47], was created to represent molecule-like lead compounds. As a result, the molecular distribution in the MOSES dataset adheres more closely to desirable drug-like properties. Notably, the molecular properties within the MOSES dataset align more closely with those of actual drugs, featuring properties, such as logP lower than 7 and greater than 3.5. Given the wider range of molecular property distribution within the GuacaMol dataset, it was utilized for testing the generation of molecules with specified properties. In contrast, the MOSES dataset, closely mimicking attributes of real-world drugs, was employed to test the generation of molecules with designated scaffolds and properties. In both test scenarios, 10,000 molecules were generated using the model for evaluation. We employed the RDKit to calculate molecular properties and extract Bemis–Murcko scaffolds and four properties of the molecule, including Synthetic Accessibility Score, Quantitative Estimation of Drug-likeness, lipophilicity, and Topological Polar Surface Area.

4.6. Metrics

We used six metrics to assess the effectiveness as well as the diversity of generated molecules by the model, and here is what the metrics mean:

- Valid: Valid pertains to the valid portions within the generated molecules based on SMILES syntax and atomic valency rules. We consider a molecule valid when the generated SMILES can be analyzed using an RDKit. A high valid score indicates that the model has learned the accurate representation of molecules and their chemical properties.
- Unique: Unique specifies that it is a case of duplicates in the generated molecule. If the newly generated molecule has not been generated before, then it is considered ideal. A lower uniqueness score suggests that the model is generating repetitive or redundant molecules.
- Novelty: Novelty refers to the segments present in the generated valid and unique molecules that are absent in the training dataset. This metric is employed to determine whether the model is overfitting, signifying that it has memorized the training data without generalizing to unseen molecules.
- Internal Diversity (IntDivp): Internal Diversity evaluates the similarity between generated molecules. As shown in Equation (8), $s1$ and $s2$ denote two molecules, and T

represents Tanimoto similarity [48]. This entails similarity comparisons between all pairs of molecules within the generated set (S). The parameter p can be either 1 or 2.

$$IntDiv_p(S) = 1 - \sqrt[p]{\frac{1}{\mid S \mid^2} \sum_{s1,s2 \in S} T(s1, s2)^p} \tag{8}$$

- Frechet ChemNet Distance (FCD): This metric tests the similarity of the generated molecular data to the training molecular data. As shown in Equation (9), where μ_G is the mean and Σ_G is the covariance of the distribution G. In the same way as Bagal et al. [36] for the Guacamol data set, the final value is -0.2 power of FCD.

$$FCD(G, D) = \left|\left|\mu_G - \mu_D\right|\right|_2 + Tr(\Sigma_G + \Sigma_D - 2(\Sigma_G \Sigma_D)^{1/2}) \tag{9}$$

- KL Divergence: KL Divergence was computed using a plethora of physicochemical descriptors for both the generated molecules and the training set. Lower values denote a proficient learning of the distribution of these properties by the model. The calculation is shown in Equations (10) and (11). Here, k represents the kth properties.

$$D_{KL}(P, Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)} \tag{10}$$

$$S = \frac{1}{k} \sum_i^k \exp(-D_{\mathrm{KL},i}) \tag{11}$$

4.7. Baselines

We juxtaposed our model against seven distinct baseline models, encompassing Char-RNN [49], VAE [50], AAE [51], JTN-VAE [52], LatentGAN [53], ORGAN [54], and MolGPT.

5. Conclusions

The primary contribution of this work lies in the successful integration of graph structures into an NLP-inspired sequence-to-sequence framework. This novel approach not only harnesses the power of language modeling but also seamlessly captures and exploits the topological features of molecules. Our model demonstrates exceptional performance in generating molecules with higher validity, uniqueness, and diversity. Unconditional molecule generation, property molecule generation, and scaffold-based molecule generation experiments all demonstrate the performance of our model. Through extensive experimentation and evaluation of established datasets, we showcase GraphGPT making significant strides in achieving more advanced structure-aware molecular generation techniques. Our method facilitates the construction of large-scale molecular screening libraries and the generation of lead compounds with specified properties, thereby propelling advancements in AI-driven drug discovery and related fields.

Author Contributions: Conceptualization, H.L. (Hao Lu) and Z.W.; methodology, H.L. (Hao Lu); software, H.L. (Hao Lu); validation, H.L. (Hao Lu), H.L. (Hao Liu) and Z.W.; formal analysis, K.Z. and X.W.; investigation, K.Z.; resources, Z.W. and H.L. (Hao Liu); data curation, X.W. and K.Z.; writing—original draft preparation, H.L. (Hao Lu); writing—review and editing, H.L. (Hao Lu), Z.W. and H.L. (Hao Liu); visualization, K.Z.; supervision, H.L. (Hao Liu); project administration, Z.W. and H.L. (Hao Liu); funding acquisition, Z.W. and H.L. (Hao Liu). All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Key Research and Development Program of China (2021YFF0704000).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The Moses and GuacaMol datasets are openly available. Our model is available via: https://github.com/luhao27/GraphGPT. accessed on 22 November 2023.

Acknowledgments: The authors thank the anonymous reviewers for their valuable suggestions. We would like to thank the Marine Big Data Center of the Institute for Advanced Ocean Study of Ocean University of China for providing data conversion and storage. We thank DeepL and Quillbot for providing grammar checks and translations of parts of the text.

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

References

1. Cheng, Y.; Gong, Y.; Liu, Y.; Song, B.; Zou, Q. Molecular design in drug discovery: A comprehensive review of deep generative models. *Brief. Bioinform.* **2021**, *22*, bbab344. [CrossRef] [PubMed]
2. Li, Z.; Jiang, M.; Wang, S.; Zhang, S. Deep learning methods for molecular representation and property prediction. *Drug Discov. Today* **2023**, *27*, 103373. [CrossRef] [PubMed]
3. Urbina, F.; Lentzos, F.; Invernizzi, C.; Ekins, S. Dual use of artificial-intelligence-powered drug discovery. *Nat. Mach. Intell.* **2022**, *4*, 189–191. [CrossRef] [PubMed]
4. Nagra, N.S.; Lieven, V.D.V.; Stanzl, E.; Champagne, D.; Devereson, A.; Macak, M. The company landscape for artificial intelligence in large-molecule drug discovery. *Nat. Rev. Drug Discov.* 2023; *online ahead of print*. [CrossRef]
5. Chen, G.; Seukep, A.J.; Guo, M. Recent Advances in Molecular Docking for the Research and Discovery of Potential Marine Drugs. *Mar. Drugs* **2020**, *18*, 545. [CrossRef] [PubMed]
6. Pagadala, N.S.; Syed, K.; Tuszynski, J. Software for molecular docking: A review. *Biophys. Rev.* **2017**, *9*, 91–102. [CrossRef] [PubMed]
7. Ding, Y.; Wang, H.; Zheng, H.; Wang, L.; Zhang, G.; Yang, J.; Lu, X.; Bai, Y.; Zhang, H.; Li, J.; et al. Evaluation of drug efficacy based on the spatial position comparison of drug–target interaction centers. *Brief. Bioinform.* **2020**, *21*, 762–776. [CrossRef] [PubMed]
8. Zhang, X.; Zhang, O.; Shen, C.; Qu, W.; Chen, S.; Cao, H.; Kang, Y.; Wang, Z.; Wang, E.; Zhang, J.; et al. Efficient and accurate large library ligand docking with KarmaDock. *Nat. Comput. Sci.* **2023**, *3*, 789–804. [CrossRef]
9. Kuan, J.; Radaeva, M.; Avenido, A.; Cherkasov, A.; Gentile, F. Keeping pace with the explosive growth of chemical libraries with structure-based virtual screening. *Wires Comput. Mol.* **2023**, *13*, e1678. [CrossRef]
10. 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]
11. Sarkar, C.; Das, B.; Rawat, V.S.; Wahlang, J.B.; Nongpiur, A.; Tiewsoh, I.; Lyngdoh, N.M.; Das, D.; Bidarolli, M.; Sony, H.T. Artificial Intelligence and Machine Learning Technology Driven Modern Drug Discovery and Development. *Int. J. Mol. Sci.* **2023**, *24*, 2026. [CrossRef]
12. Westermayr, J.; Gilkes, J.; Barrett, R.; Maurer, R. High-throughput property-driven generative design of functional organic molecules. *Nat. Comput. Sci.* **2023**, *3*, 139–148. [CrossRef]
13. Liu, Q.; Allamanis, M.; Brockschmidt, M.; Gaunt, A.L. Constrained Graph Variational Autoencoders for Molecule Design. In Proceedings of the 2020 IEEE Symposium Series on Computational Intelligence SSCI, Canberra, Australia, 1–4 December 2020.
14. Kingma, D.P.; Welling, M. Auto-Encoding Variational Bayes. In Proceedings of the International Conference on Learning Representations ICLR, Banff, AB, Canada, 14–16 April 2014.
15. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative Adversarial Networks. In Proceedings of the 27th International Conference on Neural Information Processing Systems NIPS, Montreal, QC, Canada, 8–13 December 2014; Volume 2, pp. 2672–2680.
16. Zhou, Z.; Kearnes, S.; Li, L.; Zare, R.N.; Riley, P. Optimization of Molecules via Deep Reinforcement Learning. *Sci. Rep.* **2019**, *9*, 10752. [CrossRef]
17. Bilodeau, C.; Jin, W.; Jaakkola, T.; Barzilay, R.; Jensen, K.F. Generative models for molecular discovery: Recent advances and challenges. *Wires Comput. Mol.* **2022**, *12*, e1608. [CrossRef]
18. Li, S.; Hu, C.; Ke, S.; Yang, C.; Chen, J.; Xiong, Y.; Liu, H.; Hong, L. LS-MolGen: Ligand-and-Structure Dual-Driven Deep Reinforcement Learning for 411 Target-Specific Molecular Generation Improves Binding Affinity and Novelty. *J. Chem. Inf. Model.* **2023**, *63*, 4207–4215. [CrossRef]
19. Liu, X.; Ye, K.; van Vlijmen, H.W.T.; IJzerman, A.P.; van Westen, G.J.P. DrugEx v3: Scaffold-constrained drug design with graph transformer-based reinforcement learning. *J. Cheminform.* **2023**, *37*, 373–394. [CrossRef] [PubMed]
20. Wang, M.; Wang, Z.; Sun, H.; Wang, H.; Wang, J.; Shen, C.; Weng, G.; Chai, X.; Li, H.; Cao, D.; et al. Deep learning approaches for de novo drug design: An overview. *Curr. Opin. Struct. Biol.* **2022**, *72*, 135–144. [CrossRef]
21. Cai, C.; Wang, S.; Xu, Y.; Zhang, W.; Tang, K.; Ouyang, Q.; Lai, L.; Pei, J. Transfer Learning for Drug Discovery. *J. Med. Chem.* **2020**, *63*, 8683–8694. [CrossRef]
22. Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language Models are Few-Shot Learners. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 1877–1901.
23. OPENAI. GPT-4 Technical Report. *arXiv* **2023**, arXiv:2303.08774.

24. Weininger, D. SMILES, a chemical language and information system. *J. Chem. Inf. Comput. Sci.* **1988**, *28*, 31–36. [CrossRef]
25. Tysinger, E.P.; Rai, B.K.; Sinitskiy, A.V. Can We Quickly Learn to "Translate" Bioactive Molecules with Transformer Models. *J. Chem. Inf. Model.* **2023**, *63*, 1734–1744. [CrossRef]
26. Mokaya, M.; Imrie, F.; Van Hoorn, W.P.; Kalisz, A.; Bradley, A.R.; Deane, C.M. Testing the limits of SMILES-based de novo molecular generation with curriculum and deep reinforcement learning. *Nat. Mach. Intell.* **2023**, *5*, 386–394. [CrossRef]
27. Dwivedi, V.P.; Bresson, X. A Generalization of Transformer Networks to Graphs. In Proceedings of the AAAI 2021 Workshop on Deep Learning on Graphs: Methods and Applications, DGL-AAAI, Virtual, 8–9 February 2021.
28. Vaswani, A.; Shazeer, N.M.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention Is All You Need. In Proceedings of the 31st International Conference on 441 Neural Information Processing Systems, NIPS, Long Beach, CA, USA, 4–9 December 2017; pp. 6000–6010.
29. Ying, C.; Cai, T.; Luo, S.; Luo, S.; Zheng, S.; Ke, G.; He, D.; Shen, Y.; Liu, T.-Y. Do Transformers Really Perform Bad for Graph Representation? *Arxiv* **2021**, arXiv:2106.05234.
30. Chen, D.; O'Bray, L.; Borgwardt, K.M. Structure-Aware Transformer for Graph Representation Learning. In Proceedings of the International Conference on Machine Learning, ICML, Baltimore, MD, USA, 17–23 July 2022; Volume 162, pp. 3469–3489.
31. Luo, S.; Chen, T.; Xu, Y.; Zheng, S.; Liu, T.Y.; He, D.; Wang, L. One Transformer Can Understand Both 2D & 3D Molecular Data. In Proceedings of the International Conference on Learning Representations, ICLR, Kigali, Rwanda, 1–5 May 2023.
32. Ertl, P.; Schuffenhauer, A. Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions. *J. Cheminform.* **2009**, *1*, 8. [CrossRef]
33. Bickerton, G.R.; Paolini, G.V.; Besnard, J.; Muresan, S.; Hopkins, A.L. Quantifying the chemical beauty of drugs. *Nat. Chem.* **2012**, *4*, 90–98. [CrossRef]
34. Abraham, M.H.; Chadha, H.S.; Leitao, R.E.; Mitchell, R.C.; Lambert, W.J.; Kaliszan, R.; Nasal, A.; Haber, P. Determination of solute lipophilicity, as log P(octanol) and log P(alkane) using 481 poly(styrene–divinylbenzene) and immobilised artificial membrane stationary phases in reversed-phase high-performance liquid chromatography. *J. Chromatogr. A* **1997**, *766*, 35–47. [CrossRef]
35. Zhong, H.; Mashinson, V.; Woolman, T.A.; Zha, M. Understanding the Molecular Properties and Metabolism of Top Prescribed Drugs. *Curr. Top. Med. Chem.* **2013**, *13*, 1290–1307. [CrossRef]
36. Bagal, V.; Aggarwal, R.; Vinod, P.K.; Priyakumar, U.D. MolGPT: Molecular Generation Using a Transformer-Decoder Model. *J. Chem. Inf. Model.* **2022**, *62*, 2064–2076. [CrossRef] [PubMed]
37. Brown, N.; Fiscato, M.; Segler, M.H.S.; Vaucher, A.C. GuacaMol: Benchmarking Models for de Novo Molecular Design. *J. Chem. Inf. Model.* **2019**, *59*, 1096–1108. [CrossRef] [PubMed]
38. Preuer, K.; Renz, P.; Unterthiner, T.; Hochreiter, S.; Klambauer, G. Frechet ChemNet Distance: A Metric for Generative Models for Molecules in Drug Discovery. *J. Chem. Inf. Model.* **2018**, *58*, 1736–1741. [CrossRef] [PubMed]
39. Kullback, S.; Leibler, R.A. On information and sufficiency. *Ann. Math. Stat.* **1951**, *22*, 79–86. [CrossRef]
40. Polykovskiy, D.; Zhebrak, A.; Sanchez-Lengeling, B.; Golovabov, 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]
41. Krenn, M.; Häse, F.; Nigam, A.; Friederich, P.; Apuru-Guzik, A. Self-referencing embedded strings (SELFIES): A 100% robust molecular string representation. *Mach. Learn. Sci. Technol.* **2020**, *1*, 045024. [CrossRef]
42. Zhong, Z.; Song, J.; Feng, Z.; Liu, T.; Jia, L.; Yao, S.; Wu, M.; Hou, T.; Song, M. Root-aligned SMILES: A tight representation for chemical reaction prediction. *Chem. Sci.* **2022**, *13*, 9023–9034. [CrossRef]
43. Velickovic, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; Bengio, Y. Graph Attention Networks. International Conference on Learning Representations. *arXiv* **2018**, arXiv:1710.10903.
44. RDKit: Open-Source Cheminformatics. Available online: https://www.rdkit.org (accessed on 22 November 2023).
45. Zbontar, J.; Jing, L.; Misra, I.; LeCun, Y.; Deny, S. Barlow Twins: Self-Supervised Learning via Redundancy Reduction. *arXiv* **2021**, arXiv:2103.03230.
46. Mendez, D.; Gaulton, A.; Bento, A.P.; Chambers, J.; De Veij, M.; Felix, E.; Magarinos, M.P.; Mosquera, J.F.; Mutowo, P.; Nowotka, M.; et al. ChEMBL: Towards direct deposition of bioassay data. *Nucleic Acids Res.* **2019**, *47*, D930–D940. [CrossRef]
47. Sterling, T.; Irwin, J.J. ZINC 15—Ligand Discovery for Everyone. *J. Chem. Inf. Model.* **2015**, *55*, 2324–2337. [CrossRef]
48. Vogt, M.; Bajorath, J. Modeling Tanimoto Similarity Value Distributions and Predicting Search Results. *Mol. Inform.* **2017**, *67*, 1600131. [CrossRef]
49. Segler, M.H.S.; 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]
50. Gomez-Bombarelli, R.; Wei, J.N.; Duvenaud, D.; Hernandez-Lobato, J.M.; Sanchez-Lengeling, S.; Sheberla, D.; Aguilera-Lparraguirre, J.; Hirzel, T.D.; Adams, R.P.; Aspuru-Guzik, A. Automatic Chemical Design Using a Data-Driven Continuous Representation of Molecules. *ACS Cent. Sci.* **2018**, *4*, 268–276. [CrossRef]
51. Polykovskiy, D.; Zhebrak, A.; Vetrov, D.; Ivanenkov, Y.; Aladinskiy, V.; Mamoshina, P.; Bozdaganyan, M.; Aliper, A.; Zhavoronkov, A.; Kadurin, A. Entangled Conditional Adversarial Autoencoder for de Novo Drug Discovery. *Mol. Pharm.* **2018**, *15*, 4398–4405. [CrossRef]
52. Jin, W.; Barzilay, R.; Jaakkola, T. Junction Tree Variational Autoencoder for Molecular Graph Generation. *arXiv* **2018**, arXiv:1802.04364.

53. Prykhodko, O.; Johansson, S.V.; Kotsias, P.C.; Arus-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*, 74. [CrossRef]
54. Guimaraes, G.; Sanchez-Lengeling, B.; Outeiral, C.; Farias, P.L.C.; Aspuru-Guzik, A. Objective-Reinforced Generative Adversarial Networks (ORGAN) for Sequence Generation Models. *arXiv* **2018**, arXiv:1705.10843.

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.