

Geometry-Based Molecular Generation With Deep Constrained Variational Autoencoder

Chunyan Li^{ID}, Junfeng Yao^{ID}, Wei Wei^{ID}, Senior Member, IEEE, Zhangming Niu^{ID}, Xiangxiang Zeng, Senior Member, IEEE, Jin Li^{ID}, and Jianmin Wang^{ID}

Abstract—Finding target molecules with specific chemical properties plays a decisive role in drug development. We proposed GEOM-CVAE, a constrained variational autoencoder based on geometric representation for molecular generation with specific properties, which is protein-context-dependent. In terms of machine learning, it includes continuous feature embedding encoder and molecular generation decoder. Our key contribution is to propose an efficient geometric embedding method, including the spatial structure representations of drug molecule (converting the 3-D coordinates into image) and the geometric graph representations of protein target (modeling the protein surface as a mesh). The 3-D geometric information is vital to successful molecular generation, which is different from previous molecular generative methods based on 1-D or 2-D. Our model framework generates specific molecules in two phases, by first generating special image with molecular 3-D information to learn latent representations and generating molecules with constrained condition based on geometric graph convolution for specific protein and then inputting the generated structural molecules into a parser network for obtaining Simplified Molecular Input Line

Manuscript received April 14, 2021; revised July 31, 2021 and December 14, 2021; accepted January 24, 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 62072388, Grant 62122025, Grant 61872309, Grant 61972138, and Grant 62102140; in part by the Collaborative Project Fund of Fuzhou-Xiamen-Quanzhou Innovation Zone under Grant 3502ZCQXT202001; in part by the Industry Guidance Project Foundation of Science Technology Bureau of Fujian Province under Grant 2020H0047; in part by the Natural Science Foundation of Science Technology Bureau of Fujian Province under Grant 2019J01601; in part by the Creation Fund Project of Science Technology Bureau of Fujian Province under Grant 2019C0021; in part by the Hunan Provincial Natural Science Foundation under Grant 2020JJ4215 and Grant 2021JJ10020; in part by the Fundamental Research Project of Yunnan Province under Grant 202001BB050052; in part by the Scientific Research Fund of Yunnan Provincial Department of Education under Grant 2022J0450; and in part by the Fujian Sunshine Charity Foundation. (*Corresponding authors:* Junfeng Yao; Wei Wei; Xiangxiang Zeng.)

Chunyan Li is with the School of Informatics, Xiamen University, Xiamen 361005, China, and also with the School of Mathematics and Computer Sciences, Yunnan Minzu University, Kunming 650500, China.

Junfeng Yao is with the Institute of Artificial Intelligence and the School of Film, Xiamen University, Xiamen 361005, China (e-mail: yao0010@xmu.edu.cn).

Wei Wei is with the School of Computer Science and Engineering, Xi'an University of Technology, Xi'an 710048, China (e-mail: weiwei@xaut.edu.cn).

Zhangming Niu is with MindRank AI Ltd., Hangzhou, Zhejiang 311113, China.

Xiangxiang Zeng is with the College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China (e-mail: xzeng@hnu.edu.cn).

Jin Li is with the School of Software, Yunnan University, Kunming 650091, China.

Jianmin Wang is with the Integrative Biotechnology & Translational Medicine, Yonsei University, Incheon 21983, Republic of Korea.

Color versions of one or more figures in this article are available at <https://doi.org/10.1109/TNNLS.2022.3147790>.

Digital Object Identifier 10.1109/TNNLS.2022.3147790

Entry System (SMILES) strings. Our model achieves competitive performance that implies its potential effectiveness to enable the exploration of the vast chemical space for drug discovery.

Index Terms—Coordinate, geometry, graph convolutional network, mesh, molecular generation, variational autoencoder (VAE).

I. INTRODUCTION

IT IS a crucial challenge to find candidate drug molecules with desired chemical properties in the field of drug discovery. Traditionally, it is a long and costly process with a very high failure rate for finding a new drug [1]. Generating drug molecules with some specific properties by neural-network-inspired models has shown strong potentials for accelerating drug discovery process that tries to explore large chemical space in a data-driven way [2]. The process of generating molecules is generally divided into two steps: how to effectively characterize molecules, learning the potential spatial distribution of molecules; and how to map the potential representations of molecules back to molecular structures with special properties [2], [3]. At present, the representation of molecules has been extensively studied [4], [5], and there are various molecular generation models. However, it is still a huge challenge to efficiently represent molecules and generate effective molecules with certain properties.

The previous works on molecular generation are mainly based on 1-D Simplified Molecular Input Line Entry System (SMILES) [6] or 2-D molecular graph [7], [8] to represent molecules and then learn the latent space for sampling by using various deep learning-based frameworks. SMILES are in the form of single-line text made up of molecular notations for representing molecules. One-dimensional SMILES cannot capture long-distance dependence, and it cannot learn smooth molecular embedding due to disregard of molecular structural characteristics. Two-dimensional molecular graphs are easier and more intuitive to express molecules, but for similar molecular graphs, they often have very different 3-D structures and molecular properties. There have been many researches on molecular representation, but most of which are based on 1-D and 2-D structure studies, ignoring the 3-D spatial topology of molecules. The molecular 3-D structure features play an important role in biochemical function and activity prediction. Existing studies have shown that it can successfully predict new active molecules and binding sites by using molecular encoding based on 3-D structure, which cannot be predicted by molecular 1-D or 2-D embedding [9]. Since it is very time-consuming and labor-intensive to

simulate molecular conformation with dynamic methods, our aim is to use machine learning to learn 3-D representations of molecules from 1-D or 2-D structures. Similarly, the structure of protein is very complicated due to the existence of protein folding, where its 3-D structure determines its properties. The effective characterization of the 3-D protein structure is currently a common challenge. The interactions between molecules and proteins occur in 3-D space, and therefore, the 3-D information is essential for successful molecular generation and compound design. To our best knowledge, there have been two 3-D-based representation models proposed for molecular generation: LigDream [10] and DeLinker [11]. LigDream can design novel compounds with desirable characteristics such as drug-likeness. DeLinker incorporates 3-D structural information directly in the molecular design process and can be applied to fragment linking and scaffold hopping. However, they have a high computational cost and complex substructure decomposition.

In this article, we propose a novel constrained variational autoencoder (VAE) to generate molecules targeting severe acute respiratory syndrome (SARS) associated coronavirus (SARS-CoV) 3C-like Protease (3CLPro), which is one of the Coronavirus Disease-19 (COVID-19) targets. Our primary contribution is a molecular generative model based on geometric embedding, including the spatial structure representation for drug molecule and the geometric graph representation for the protein target. Specifically, in terms of geometric embedding of molecules, inspired by the great success of traditional convolutional neural networks (CNNs) in the field of computer vision, we convert 3-D drug molecules into special images with 3-D coordinate information and then use CNNs to extract 3-D features of molecules in the images. Second, in terms of protein geometric embedding, we propose an efficient geometric graph convolution method (curvature-based quadratic error metrics and mesh sampling for convolutional operation) to obtain comprehensive protein representation by using graph CNNs (GCNs) [12]–[14]. We evaluate our model through a wide range of experiments from molecular generation, reconstruction, and visualization to chemical properties evaluation. Our model achieved competitive performance comparing the state-of-the-art VAE-based model as baselines. All in all, the main contributions of this article are as follows.

- 1) A 3-D structure-based molecular visual representation algorithm by converting 3-D coordinates into images for drug molecules to implement CNNs.
- 2) A geometry-based protein graph representation algorithm by modeling the protein surface as a mesh and using curvature-based quadratic error metrics and mesh sampling mechanism to implement GCN.
- 3) A constrained VAE framework a Constrained Variational AutoEncoder based on GEOMETRIC representation for molecular generation (GEOM-CVAE): through the design of the constrained VAE model, not only the target protein is added to the model as a restrictive condition for molecular generation, but also the distribution of latent variables is further restricted. The high-quality molecules targeting known proteins are generated at last.

The rest of the article is organized as follows. Section II reviews the related work. Section III presents our approaches.

Section IV demonstrates and analyzes the experimental results. The conclusion is shown in Section V.

II. RELATED WORK

A. Molecular Generation

There are many deep generative models for generating molecules, mainly including VAEs-based models [2], [10], [15]–[18], generative adversarial networks (GANs)-based models [19], autoregressive (AR)-based models [20], reinforcement learning-based models [21], and flow-based models [22]–[24]. GAN is an implicit generative model, while VAE is a likelihood-based generative model. Since GAN is susceptible to mode collapse, it is always more difficult to train than VAE. The VAE-based molecular generation models can be divided into two main categories, namely 1-D-SMILES-based molecular generation [15], [16] and 2-D-graph-based molecular generation [2], [7], [8], [25]. The grammar VAE (GVAE) [16] builds a parse tree based on SMILES grammar and introduces context free grammar (CFG) constrained decoder for generating effective SMILES. Similarly, the syntax-directed VAE (SD-VAE) [15] proposed a syntax-directed method, which includes CFG parsing for syntax and attribute grammar for semantics. The GraphVAE [25] is a molecular graph generation model to output a probabilistic fully-connected graph by using a graph matching algorithm. The junction tree VAE (JT-VAE) [2] generates a tree-structured scaffold and then uses a graph message passing network to generate molecules.

B. Geometric-Based Generative Models

LigDream [10] and DeLinker [11] generated molecules considering 3-D molecular structures. The LigDream [10] is a voxel-based generative model by using molecular volumetric representation, and then 3-D CNN is used to learn molecular continuous embedding for screening novel molecular scaffolds. The LigDream is computationally inefficient due to convolving over large regions of empty space. Therefore, large memory overhead and computational complexity must be endured, which is the disadvantage of the voxel-based methods. The DeLinker [11] takes two partial structures and generates molecules by utilizing 3-D information such as the relative distance and orientation. It requires additional decomposition of molecular structures to obtain substructures, which increases the computational complexity. However, the methods did not pay more attention to the 3-D representation of molecules. Differently, our approach is closely related to 3-D geometric information extraction for capturing spatial topological features. We start from the perspective of learning continuous embedding, that is, learning the 3-D representation of different objects in the model. The proposed geometric graph representation algorithm for protein further constraints the output latent space to ensure the effective and novel molecular generation. GEOM-CVAE extracts features by using CNNs and GCNs, which requires less computational overhead than the voxel-based methods.

III. GEOMETRY-BASED CONSTRAINED VAE

Our approach extends the VAE [26] to constrained VAE by proposing an efficient geometric embedding method for

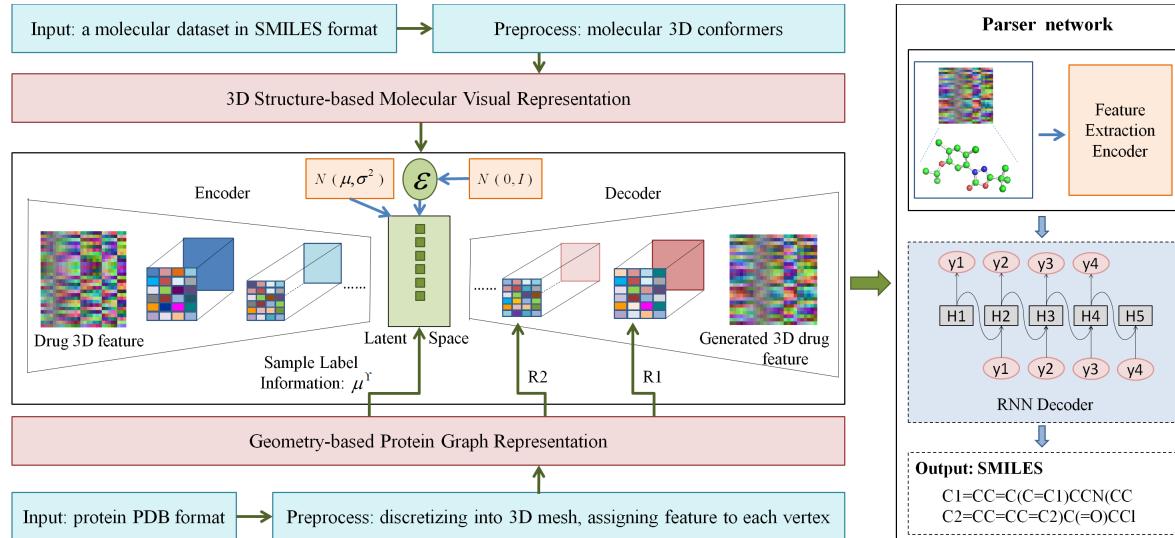


Fig. 1. Overview of the geometry-based molecular generation by using constrained VAE, including four modules: 3-D structure-based molecular visual representation module, geometry-based protein graph representation module, constrained VAE module, and parser network module. The constrained VAE uses sample labels to limit the latent space distribution and uses geometric graph representation for protein to limit the decoder. GEOM-CVAE generates molecules with the specific properties that target the protein. The parser network is used to obtain molecular SMILES.

drug molecule and protein representation. By restricting the potential spatial distribution and constrained decoder, new molecules with special properties are generated, which are in the form of images with 3-D spatial information. Then they are further sent to the parser network to obtain molecular SMILES, as shown in Fig. 1, which mainly consists of four modules: 3-D structure-based molecular visual representation module, geometry-based protein graph representation module, constrained VAE module, and parser network module. The constrained VAE uses sample labels and geometric protein representation to limit the latent space distribution and the decoder, respectively. Then the parser network is used to output generative SMILES sequences.

A. Three-Dimensional Structure-Based Molecular Visual Representation

Encouraged by the great success of CNNs in computer vision, we encode molecular spatial structures into images using the coordinates of each atom in the molecules. Specifically, a molecule is defined as $M = (V, P)$, where V is the atom set and P is the coordinate set. $|V| = N$ and each atom v has a position coordinates $P_v = (x_v, y_v, z_v)$, $P \in \mathcal{R}^{N \times 3}$. The main idea is that the coordinate (X, Y, Z) of each atom corresponds to the red, green, and blue (RGB) of the pixel in the image, which is convenient to use CNNs to extract the spatial structure features of the molecules. Since the number of atoms in each molecule is different, it is a key to generate a fixed-size image for each molecule to be input into the neural network. Here, we use a trick. The process of visual representation is sensitive to molecular rotation and translation. In order to ensure the invariance of rotation and further increase the robustness of the model, each molecule is randomly rotated, and the number of rotations is different for different molecule, which is exactly the number of atoms in each molecule. In this way, a batch of feature vectors $F = (N_{\text{mol}_1}, N_{\text{mol}_1}, 3), (N_{\text{mol}_2}, N_{\text{mol}_2}, 3), \dots, (N_{\text{mol}_n}, N_{\text{mol}_n}, 3)$

with molecular 3-D structure information are obtained. N_{mol_n} is denoted as the number of atoms in a molecule, which changes for different molecules. Then, convert each feature vector of F into an image with a fixed size $60 \times 60 \times 3$. In this way, the 3-D spatial feature expression of the molecules is finally obtained. Fig. 2 shows the process of converting the 3-D conformation of molecule into a visual representation. Furthermore, we use CNNs to extract image features, which is based on the 3-D spatial structure of molecules. The molecular visual algorithm is summarized as Algorithm 1.

Algorithm 1 Three-Dimensional Structure-Based Molecular Visual Algorithm

Input: SMILES dataset \mathcal{S} , $|\mathcal{S}| = t$. Empty sets H and X .
Preprocess: Molecular conformers with 3-D coordinates vector set $C = \{C_1, \dots, C_i\}_{i=1}^{i=t}$, the set of atomic numbers $N = \{n_1, \dots, n_i\}_{i=1}^{i=t}$.
Output: molecular images with 3-D information.

```

1: initial  $i = 1$ ;
2: while  $t > 0$  do
3:    $m_i \leftarrow n_i$ 
4:   repeat
5:      $\forall$  rotational matrix  $R$ ,  $C'_i \leftarrow (RC_i)$ ,  $C_i \in \mathcal{R}^{n_i \times 3}$ 
6:     Add  $C'_i$  into set  $H$ 
7:    $m_i \leftarrow m_i - 1$ 
8:   until ( $m_i \leq 0$ )
9:   Add  $H$  into set  $X$ ,  $H \in \mathcal{R}^{n_i \times n_i \times 3}$ 
10:   $H \leftarrow null$ 
11:   $i \leftarrow i + 1$ 
12:   $t \leftarrow t - 1$ 
13: end while
14: the set of images  $M \leftarrow X$ 
15: return  $M$ .

```

B. Geometry-Based Protein Graph Representation

A high-quality description of protein structure is essential for a variety of downstream tasks. The protein structure prediction [27]–[30] remains an important challenging problem in computational structural biology. Most of the previous work focused on 1-D sequence-based manner [31] and 2-D protein

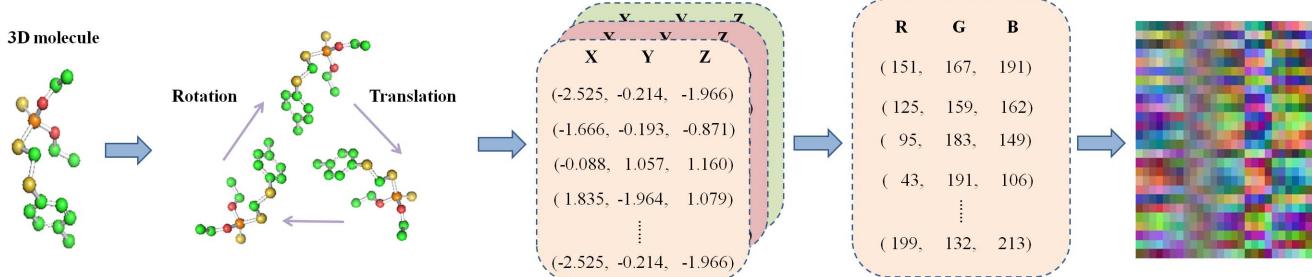


Fig. 2. Three-dimensional structure-based molecular visual representation method. It shows the process of converting molecular 3-D spatial information into visual image after the 3-D molecule is rotated and translated.

graph [32] with nodes denoting the non-hydrogen atoms and edges denoting the connections between K nearest neighbors for each atom using predefined threshold. Then, the atom embeddings are pooled to generate residue-level embeddings. However, the protein graph can only obtain a simple protein representation and cannot capture the complex 3-D structural information. Geometric deep learning (GDL) [33] is a nascent field extending deep neural network architectures based on images, such as CNNs, to geometric data such as surfaces and curvatures [34], which have been shown to outperform handcrafted feature extraction [35]. In this section, we exploit GDL techniques and present a novel geometric graph learning algorithm to capture the geometric features of 3-D protein, which is a general method to obtain the 3-D protein graph by learning protein surfaces and can be applicable to any protein structure.

1) Protein Surface Learning and Discretizing Into Mesh: Learn protein surface and convert the surface into a 3-D mesh structure according to the solvent excluded surface (SES). The protein 3-D mesh, consisting of many vertices and triangles, can be obtained using the MSMS program [36]. The SES can be triangulated with a given vertex density and probe radius. Each vertex in the mesh is assigned geometric and chemical features. Fig. 3(a) shows the composition of geometric features and chemical features, which is computed directly on the protein 3-D mesh. The geometric features include mean curvatures, Gaussian curvatures, shape indexes, coordinate of vertices, and normal of vertices. Specifically, the shape index describes the shape around each vertex on the surface [34], [37]. The values range from -1 (highly concave) to $+1$ (high convex). The shape index can be defined using vertex mean curvature H and Gaussian curvature K . The principal curvatures G_1, G_2 can be calculated by using H and K as: $G_1 = H + (H^2 - K)^{1/2}$ and $G_2 = H - (H^2 - K)^{1/2}$. The shape index Si can be computed as follows:

$$Si = \frac{2}{\pi} \arctan \frac{G_1 + G_2}{G_1 - G_2}. \quad (1)$$

On the other hand, the chemical features include hydrophobicity, vertex charges based on hydrogen bond potential, Poisson–Boltzmann continuum electrostatics, and type attribute of vertices. PDB2PQR [38] package automates the task of preparing structures for continuum electrostatics calculations, and Adaptive Poisson–Boltzmann Solver (APBS) [39] is used to compute Poisson–Boltzmann electrostatics. Multivalue, provided in the APBS, can be used to assign each

vertex of the mesh the corresponding electrostatics, which is normalized to a value between -1 and 1 . The type attributes of vertices are used to vectorize the different residues and atom types in protein structure. Following the previous method [34], the original generated protein mesh is further regularized by using pymesh [40] to remove degenerated triangles, obtuse triangles, isolated vertices, and duplicated faces and vertices. The geometric and chemical features previously assigned to each vertex in the original mesh must be redistributed to the new mesh vertex through methods such as K -nearest neighbor feature interpolation. In the end, we obtain a regularized protein 3-D mesh with 10736 vertices, 21492 faces for SARS-CoV 3CLPro, and mark it as original mesh M_0 .

2) Geometry-Based Protein Graph Convolutional (GGC) Algorithm: In order to capture both local and global context, a hierarchical multi-scale representation of the mesh is essential. We seek a relatively simplified model to replace the original model. Preserving key features by reducing the vertices and faces of the model is the essence of surface simplification algorithms. We propose a novel curvature-based quadratic error metrics mesh sampling algorithm, which combines the quadratic error metric [41] with curvature to simplify the original mesh structure and retain the key features in the mesh, that we are interested in. Garland and Heckbert's simplification algorithm [41] is based on edge collapse to minimize the vertices and faces of the original mesh, maintains surface error approximations, and retains the original features of the model using quadric matrices. The order of edge collapse is determined by the initial error matrix of each vertex. Each vertex can be considered as the intersection of the planes around its surrounding triangles. The vertex error is defined as the squares sum of the distance from the vertex to these planes. Fig. 4 shows the contractions of vertex pairs, including edge contraction and non-edge contraction [41].

Curvature describes the degree of bending of a geometric body, such as the degree to which a curved surface deviates from a plane or the degree to which a curve deviates from a straight line. Curvature indicates the features and the details of the curve, which reflects the surface characteristics of the mesh. Regarding the 3-D structure of a protein, no matter how it is folded, we are always focus on the pocket area or ligand binding site area, which is often a hollow area with a relatively large curvature change. Therefore, when the mesh is simplified, we hope to retain as much as possible the edges formed by the points with large curvature, so as to better preserve the geometric and detailed features of the 3-D mesh.

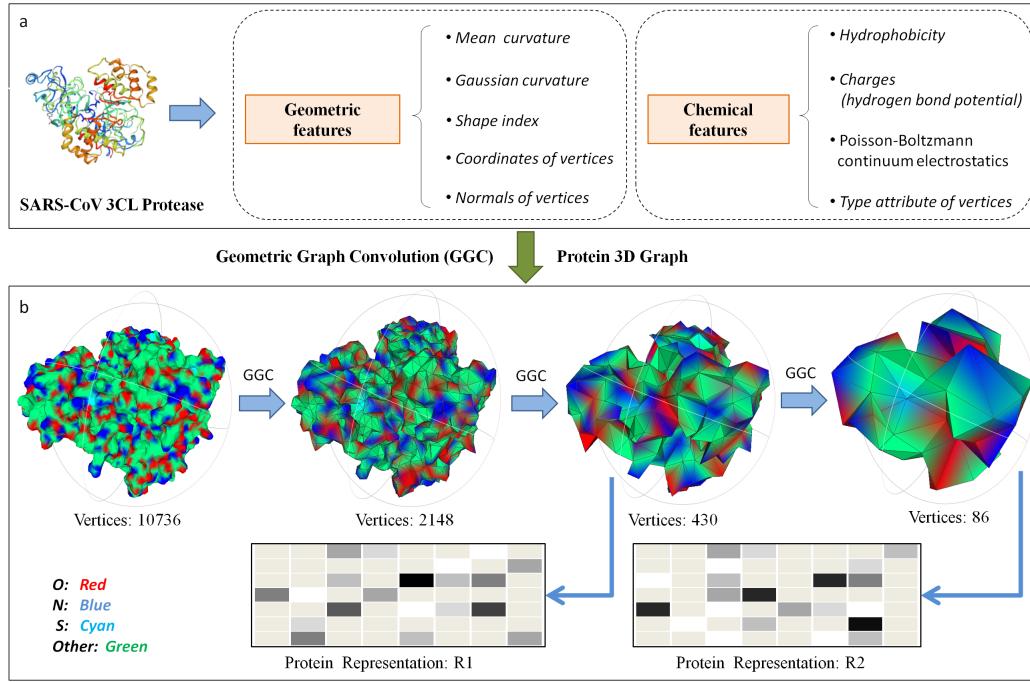


Fig. 3. Geometry-based protein graph representation. (a) Geometric and chemical features of 3-D protein structure can be obtained by learning the protein surface. (b) Proposed 3-D protein representing pipeline by using the geometry-based protein graph convolutional (GGC) algorithm. The structure of the protein is expressed in the form of a 3-D mesh, and the mesh structure is simplified by the proposed curvature-based quadratic error metrics algorithm, so that the vertices of the new mesh are 1/5 times the number of vertices of the previous mesh. Each time the simplification of protein structure is accompanied by the operation of Chebyshev graph convolution. By simplifying M_0 with the original 10736 vertices, we obtain M_1 with 2148 vertices, iteratively simplify M_1 to get M_2 with 430 vertices and simplify M_2 to get M_3 with 86 vertices. The simplification of mesh can also be regarded as the graph sampling and graph pooling in graph neural network. The output of the last two layers in GGC network, namely $R1$ and $R2$, are input to the decoder of GEOM-CVAE as restrictive conditions.

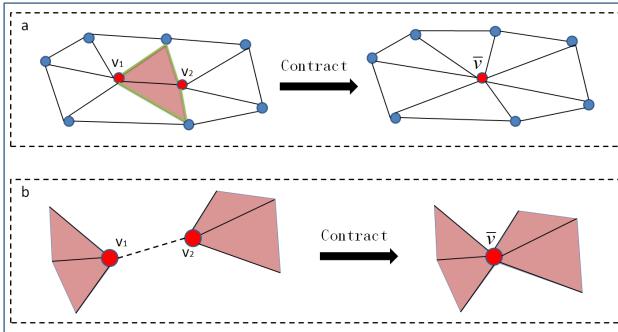


Fig. 4. Vertex pair contractions. (a) Edge contraction. The edge v_1v_2 is contracted into a single-point \bar{v} . The shaded triangles become degenerate. (b) Non-edge contraction. When non-edge pairs v_1v_2 are collapsed, v_1v_2 are joined and contracted together to \bar{v} .

From the original surface simplification algorithm [41], the vertex error is defined as the distance from the vertex to the planes where the vertex is located. Let $v = [v_x, v_y, v_z, 1]^T$ be a vertex in a 3-D mesh. Let $q = [a, b, c, d]^T$ represent the coefficient of the plane equation $ax + by + cz + d = 0(a^2 + b^2 + c^2 = 1)$. The vertex v error is defined as

$$\begin{aligned}\Delta(v) &= \Delta([v_x, v_y, v_z, 1]^T) = \sum_{q \in \text{planes}(v)} (q^T v)^2 \\ &= \sum_{q \in \text{planes}(v)} (v^T q)(q^T v) = \sum_{q \in \text{planes}(v)} v^T (qq^T)v \\ &= v^T \left(\sum_{q \in \text{planes}(v)} Q_q \right) v\end{aligned}\quad (2)$$

where $Q_q = qq^T = [a, b, c, d]^T [a, b, c, d]$ is quadratic error metrics and $\text{planes}(v)$ denotes all the triangles that meet at the vertex v . When the contraction $(v_1, v_2) \rightarrow \bar{v}$ happens, the new position \bar{v} needs to be determined to maintain the original mesh's topology. The position for \bar{v} can be found by minimizing $\Delta(\bar{v})$, which can be solved using $\partial\Delta(\bar{v})/\partial x = \partial\Delta(\bar{v})/\partial y = \partial\Delta(\bar{v})/\partial z = 0$. Considering the specific 3-D structure of a protein and focusing on part area, for a given contraction $(v_m, v_n) \rightarrow \bar{v}$, the collapse cost is marked as coll_cost , we propose a new curvature-based quadratic error metrics which approximates the error $\Delta^{(\text{new})}$ at \bar{v} based on quadratic error metrics, similar to previous work [42]. The difference between our method and [42] is that we additionally consider the mean curvature, use different coefficients to limit the Gaussian curvature and the mean curvature, and impose different degrees of penalty on the collapsed edges based on different curvatures. It is defined with respect to the vertex mean curvature H and vertex Gaussian curvature K

$$\text{coll_cost}(v_m, v_n) = 1 - e^{-t \left(\sum_{j \in \{m, n\}} (\alpha |K_{v_j}| + \beta |H_{v_j}|) \right)} \quad (3)$$

where $K_{v_j} = G_1^{(v_j)}G_2^{(v_j)}$, and $H_{v_j} = G_1^{(v_j)} + G_2^{(v_j)}/2$, α and β are predefined hyperparameters, and t is a parameter for tuning the effect of curvature. coll_cost is further defined by the principal curvatures G_1, G_2 , ($G_1 \geq G_2$) as: $\text{coll_cost}(v_m, v_n) = 1 - e^{-t \left(\sum_{j \in \{m, n\}} (\alpha |G_1^{(v_j)}G_2^{(v_j)}| + \beta |G_1^{(v_j)} + G_2^{(v_j)}/2|) \right)}$. Therefore, curvature-based quadratic error metrics can be defined as

$$\Delta^{(\text{new})}(\bar{v}) = (\Delta(v_m) + \Delta(v_n)) \times \text{coll_cost}(v_m, v_n). \quad (4)$$

Algorithm 2 Geometry-Based Protein Graph Convolutional (GGC) Algorithm

Input: protein structure \mathcal{P} , step size *stepsize*, heap \mathcal{H} , down sampling factor f and transformation *down_trans*, Chebyshev graph convolution *ChebyGCN*.

Preprocess: Discretizing protein surface into mesh M_0 , assigning geometric features and chemical features as the original feature vector X_{ori} to each vertex in mesh.

Output: 3-D protein graph representation.

```

1: function CURVATURE-BASEDERRORMETRICS( $M_0, f$ )
2:    $\forall (v_m, v_n) \in M_0$ 
3:      $coll\_cost(v_m, v_n) \leftarrow 1 - e^{-t(\sum_{j \in [m,n]} (\alpha|K_{v_j}| + \beta|H_{v_j}|))}$ 
4:      $\mathcal{COST} \leftarrow (\Delta(v_m) + \Delta(v_n)) \times coll\_cost(v_m, v_n)$ .
5:     Push the  $\mathcal{COST}$  into the heap  $\mathcal{H}$ .
6:   while  $f > 0$  do
7:      $'_M \leftarrow M_0$  remove  $(v_m, v_n)$  with the least cost.
8:     update costs of left valid pairs in  $\mathcal{H}$ .
9:      $M_0 \leftarrow '_M, f \leftarrow f - 1$ 
10:    end while
11:     $\mathcal{D}' \leftarrow down\_trans('_M, M_0)$ 
12:    return Simplified mesh  $'_M, \mathcal{D}'$ 
13:  end function
14: initial  $t = 0$ ;
15:  $G_0 \leftarrow$  Convert  $M_0$  to graph
16:  $L_0 \leftarrow$  Convert  $G_0$  to Laplacian Matrix
17:  $X_0 \leftarrow ChebyGCN(L_0, X_{ori})$             $\triangleright$  Eq. (5)
18: repeat
19:    $M_t, \mathcal{D}_t \leftarrow$  CURVATURE-BASEDERRORMETRICS( $M_0, f$ )
20:    $L_t \leftarrow L_{t-1}$  multiply  $\mathcal{D}_t$ 
21:    $X_{pooling} \leftarrow X_{t-1}$  multiply  $\mathcal{D}_t$ 
22:    $X_t \leftarrow ChebyGCN(L_t, X_{pooling})$ 
23:    $t \leftarrow t + 1$ 
24: until ( $t \geq stepsize$ )
25: return  $X_t$ .
```

The order of edge collapses is defined by $\Delta^{(new)}$. When collapses happen, the edge with the smaller error is always selected for collapse. The greater the curvature, the greater the collapse error, and the later the edge collapse. It maintains the geometric characteristics of the 3-D mesh, which is what we pay more attention to. The process of edge collapses is the process of selecting and deleting points in the graph, as well as the process of graph sampling and graph pooling. Fig. 3(b) shows the 3-D protein representing pipeline by using the geometric graph convolutional network, where each step of the convolution process is accompanied by the process of graph sampling and graph pooling with the output of the graph Laplacian matrix and transformed node embedding. After each graph is pooled, the 3-D protein graph nodes are reduced to a preset ratio, while retaining more geometric features and details. We use Chebyshev graph convolution [12], [13] to capture the geometric feature of protein. It, with K -order approximation, can effectively avoid the high computational overhead caused by the eigendecomposition and obtain neighbor node information with K -hop, which is important to capture complex relationships, especially long-range dependence. The graph convolution with K -order

polynomial form is defined as $g_{\theta'} * x \approx \sum_{k=0}^K \theta'_k (\phi T_k(\hat{\Lambda}) \phi^T) x$, $\hat{\Lambda} = 2/\lambda_{\max} \Lambda - I_n$ is the eigenvalue matrix scaled using the largest eigenvalue of Laplacian matrix, and I_n is the identity matrix. $\theta' \in \mathbb{R}^K$ represents a Chebyshev vector. θ'_k is the k th-dimensional component. $T_k(x)$ is defined recursively as $T_k(x) = 2x T_{k-1}(x) - T_{k-2}(x)$ with $T_0(x) = 1$ and $T_1(x) = x$. $\hat{\mathcal{L}} = 2/\lambda_{\max} \mathcal{L} - I_n$, and $\mathcal{L} = I_n - \mathbf{D}^{-1/2} \mathbf{A} \mathbf{D}^{-1/2}$. The Chebyshev GCN in the form of matrix is formulated as

$$Y(X) = \sum_{k=0}^{K-1} \theta'_k T_k(\hat{\mathcal{L}}) X \quad (5)$$

where $T_k(\hat{\mathcal{L}}) = 2\hat{\mathcal{L}} T_{k-1}(\hat{\mathcal{L}}) - T_{k-2}(\hat{\mathcal{L}})$, $T_0(\hat{\mathcal{L}}) = 1$, $T_1(\hat{\mathcal{L}}) = \hat{\mathcal{L}}$, $\hat{\mathcal{L}} = 2/\lambda_{\max} \mathcal{L} - I_n$, and \mathcal{L} is the normalized graph Laplacian. $T_k(\hat{\mathcal{L}})$ is a set of Chebyshev basis. $T_k(\hat{\mathcal{L}})X$ is a projection of the input features to the Chebyshev basis. $\theta' = [\theta'_0, \theta'_1, \dots, \theta'_{K-1}]$ are learnable weights shared across nodes. The geometric protein graph convolution using curvature-based quadratic error metrics is summarized as Algorithm 2. The curvature-based quadratic error metrics method has an overall complexity of $O(N_0 \log N_0)$, and N_0 is the number of triangles in original mesh M_0 .

C. Molecular Generation With Geometric Representations

Fig. 1 shows the deep constrained VAE by using sample labels to limit the latent space and using 3-D protein structure characterization to limit the decoder, which is the output of the last two GCNs, namely $R1$ and $R2$ in the process of geometry-based protein graph representation. Then $R1$ and $R2$ are the input to the decoder of the constrained VAE as restrictive conditions. The constrained VAE generates drug molecular images with the specific properties. Then the generated molecular images are sent to the parser network, which is composed of a feature extraction encoder and a recurrent neural network (RNN) decoder. The feature extraction encoder obtains 3-D molecular representations by extracting generated images based on the spatial structure of the molecules, and the RNN decoder parses the 3-D drug representations into SMILES.

We wish to learn both an encoder and a decoder that map X to a continuous embedding Z and Z to X , respectively. The VAE [26] defines Z as a latent variable and prior distribution $p(Z)$ imposed on the latent code representation. The encoder in VAE is a variational posterior $q_\varphi(Z|X)$ parameterized by φ and a probabilistic decoder is defined by a likelihood function $p_\theta(X|Z)$, which is a generative distribution parameterized by θ . When sampling the latent space, we use the reparameterization trick to make the whole model trainable. The VAE loss consists of reconstruction loss \mathcal{L}_{re} and Kullback–Leibler Divergence (KL) \mathcal{L}_{KL} , which is defined by minimizing the upper bound on negative log-likelihood $-\log(p_\theta(X))$

$$\begin{aligned} \mathcal{L}(X; \varphi, \theta) &= \mathcal{L}_{re} + \mathcal{L}_{KL} \\ &= \mathbb{E}_{q_\varphi(Z|X)} [-\log p_\theta(X|Z)] + KL[q_\varphi(Z|X) \| p(Z)]. \end{aligned} \quad (6)$$

If $p_\theta(X|Z)$ is Bernoulli distribution with parameter ρ ($0 \leq \rho \leq 1$), let D be the dimension of X , then $-\log p_\theta(X|Z) = \sum_{j=1}^D [-x_j \log \rho_j(z) - (1 - x_j) \log(1 - \rho_j(z))]$. If $p_\theta(X|Z)$

is normal distribution, $\overline{\mu}(z)$ and $\overline{\sigma}^2(z)$ are two neural networks for mean μ and variance σ with input z , respectively, then $-\log p_\theta(X|Z) = 1/2 \|x - \overline{\mu}(z)/\overline{\sigma}(z)\|^2 + 1/2 \sum_{t=1}^D \log \overline{\sigma}_t^2(z) + D/2 \log 2\pi$. Due to our constrained VAE using sample labels to limit the latent space, KL-divergence is different from VAE's KL-divergence and can be defined as $KL[q_\phi(Z|X) \parallel p(Z)] = 1/2 \sum_{t=1}^D \{[\mu_t(x) - \mu_t^Y(x)]^2 + \sigma_t^2(x) - \log \sigma_t^2(x) - 1\}$, where Y is the embedding vector of the sample label.

IV. EXPERIMENTS

Following previous works [2], [22], we validate our model by answering following questions.

- 1) *Molecule reconstruction and validity:* Can our model reconstruct input molecules from the latent embedding? Can our model generate valid and novel molecules as many as possible?
- 2) *Molecular 3-D structure similarity verification:* Can our model generate similarity molecules based on 3-D structures? How to verify the similarity based on 3-D topological structure? It is different from the 2-D structure-based Morgan fingerprint similarity (Tanimoto coefficient).
- 3) *Molecular chemical properties evaluation:* Do the molecules generated by our model have good chemical properties, such as drug-likeness, synthetic accessibility (SA), and liposolubility (logP)?

A. Experimental Setups

1) *Dataset:* We use the AID1706 Bioassay data for COVID-19 in PubChem database (<https://pubchem.ncbi.nlm.nih.gov/bioassay/1706>), that is, a high-throughput screening assay to identify inhibitors of the SARS coronavirus 3CLPro. The AID1706 record belongs to the assay project "Summary of probe development efforts to identify inhibitors of 3CLPro" from the Scripps Research Institute Molecular Screening Center (<https://hts.florida.scripps.edu/>). It contains about 290 K active and inactive drug molecules for the SARS-CoV 3CLPro. For the activity score of the dataset, we set a threshold as 15. Any drug molecule with an activity score greater than or equal to this threshold and less than 100 is considered to be an active molecule to 3CLPro and that is a positive sample in model training. Cleaning data to exclude invalid molecules and distinguish isomeric SMILES, which is the same compound but appears as different SMILES sequence. We also deal with molecules with more than one covalently bonded unit for obtaining the largest covalent unit in a molecule. In the process of generating geometric graph for 3CLPro, the dimension of original protein embedding is set to 29, which is the sum of the dimensions of geometric features and chemical features. The 3CLPro is triangulated using the MSMS program [36] with a probe radius of 1.5 Å and a density of 3.0. Protein mesh structure is regularized using pymesh [40] with a resolution of 1.0, which is consistent with the setting of Masif [34].

2) *Baselines:* GEOM-CVAE is a VAE-based molecular generation model and focus on 3-D geometric representation. Therefore, we choose the following models as baselines:

1) five VAE-based models: JT-VAE [2], Graph-VAE [25], Character-VAE [3], Grammar-VAE [16], SD-VAE [15]; 2) an atom-by-atom AR long short-term memory (LSTM) model: AR-LSTM [43]; and 3) two flow-base models: GraphNVP [24] and graph residual flow (GRF) [23].

3) *Model Configuration:* The latent space dimension in our model is set as 512. The 3-D molecular spatial structure is converted into a $60 \times 60 \times 3$ image. The hidden dimension and output dimension of geometric graph convolution in Chebyshev network are set as 128, layer and Chebyshev order are both set to 3. In the parser network, the vocabulary list contains 33 tokens belonging to the following set: $\{C, c, N, n, S, s, P, O, o, B, F, I, Cl, [nH], Br, 1, 2, 3, 4, 5, 6, 7, @, \#, =, ?, +, /, \backslash\backslash, (,), [,]\}$. The embedding size and hidden size of the RNN decoder in the parser network is set 512 and 1024, respectively. Following the above setup, we implemented geometric feature-based molecular generation using Pytorch framework and Adam optimizer with learning rate 0.001.

B. Experimental Results

Below, we show the effectiveness of our proposed model by answering the three questions previously raised.

1) Molecule Reconstruction and Validity:

a) *Setup:* We evaluate the capabilities of our model through reconstruction, validity, uniqueness, and novelty, which is a widely used metrics for evaluating whether our model can generate valid and novel molecules as many as possible from the latent space. The reconstruction accuracy is the percentage of molecules in the input dataset that can be reconstructed from latent representations. The validity refers to the percentage of chemically valid molecules (checked by RDKit) in all the generated molecules. The uniqueness refers to the ratio of the unique valid molecules to all generated valid molecules, and the novelty is also a ratio between the generated valid molecules not in the training dataset and all generated valid molecules. We report the results of this evaluation by randomly selecting 100 active drug molecules out of all 444 active molecules, where each molecule is encoded ten times and each encoding is decoded ten times. When RNN decoding is performed through the parser network, the long- and short-distance dependencies are captured as much as possible and controlled by using a different sampling length threshold. We propose a new overall metric Randomly overall metric of Reconstruction, Validity, Uniqueness and Novelty (R-RVUN) which is the average of reconstruction, validity, uniqueness, and novelty for a set of randomly selected samples.

b) *Results:* Table I shows that our model outperforms the other baselines in reconstruction and uniqueness. Surprisingly, the proposed model obtains 100% reconstruction accuracy, which may be due to the randomness of the selected molecules and the accuracy of learning the latent space about the molecular 3-D structure. In terms of validity, we obtained 81.8% accuracy, which is lower than the validity accuracy of JT-VAE. That is because we are concerned about the geometric structure representation for 3-D molecule in the conformational space and 3-D protein surface and have not additionally controlled

TABLE I
PERFORMANCE COMPARISON. BASELINE RESULTS ARE TAKEN
FROM [2] AND [22]

Model Type	Model	Rec.	Val.	Uni.	Nov.	R-RVUN
VAE-based	JT-VAE	76.7%	100%	-	-	88.35%
	Character-VAE	44.6%	0.7%	-	-	22.65%
	Grammar-VAE	53.7%	7.2%	-	-	30.45%
	SD-VAE	76.2%	43.5%	-	-	59.85%
	GraphVAE	-	13.5%	-	-	13.50%
AR-based	AR-LSTM	-	89.2%	-	-	89.20%
Flow-based	GraphNVP	100%	42.6%	94.8%	100%	84.35%
	GRF	100%	73.4%	53.7%	100%	81.78%
Geometry-based	GEOM-CVAE	100%	81.8%	100%	94.11%	93.98%

the validity of the generated molecule. In addition, we obtained 94.11% accuracy in terms of novelty, which is lower than the accuracy of flow-based models GraphNVP and GRF. To further control the validity and novelty of generated molecules is the goal of our next research. However, in the end, we obtained an R-RVUN accuracy of 93.98%, which is the overall metric for molecular generation.

2) Molecular 3-D Structure Similarity Verification:

a) *Setup.*: The task is to perform molecular similarity verification considering the 3-D topological structure and to visualize the 3-D conformers, which have the greatest similarity to the topological structure of the seed molecules. Given a molecule a , the task is to find a different molecule b in a set of generated molecules S from a seed molecule. The molecule b has the greatest similarity $\sim(a, b) \geq \max \sim(a, s_i)$, where \sim is denoted as similarity score and $s_i \in S$. The previous method is to calculate Tanimoto similarity with Morgan fingerprint, which is a 2-D metric and is not suitable for the similarity measurement of 3-D structures. Since our model is based on the 3-D structural representation of molecules, it is crucial to find an effective measure of similarity for 3-D structures. We use extended three-dimensional fingerprint (E3FP) [9] as the similarity metric to find the greatest similarity molecule. E3FP is 3-D representation of molecular conformers, extended-connectivity fingerprint (ECFP) [44] to 3-D topological space. E3FP encodes small-molecule 3-D substructures and aggregates neighbor information of nearby atoms, even if they do not exist connected bonds between each other. Most importantly, E3FP can predict novel interactions and properties that is unpredictable by 2-D fingerprints. We generate conformers follow the parameter configuration [9], and the number of rotatable bonds determined the number of molecular conformers. Minimize conformers with the universal force field (UFF) in RDKit and sort them with predicted energy, then for selecting candidate conformer from potential conformers pool we calculate root-mean-square deviation (RMSD) with some threshold sigma ($\sigma = 0.5$) in order to maximize the difference between different conformers. Finally, three conformers with the lowest energy are accepted for each molecule to calculate the 3-D fingerprint.

b) *Results.*: In Fig. 5, we visualize the generated molecules from two different random seed molecules. The molecules in the red dashed box are seed molecules. Furthermore, Fig. 6 shows the conformers of random five different seed molecules and the de novo molecular conformations with the closest E3FP similarity from the seed molecules. It verifies that our model can effectively capture the 3-D structural features

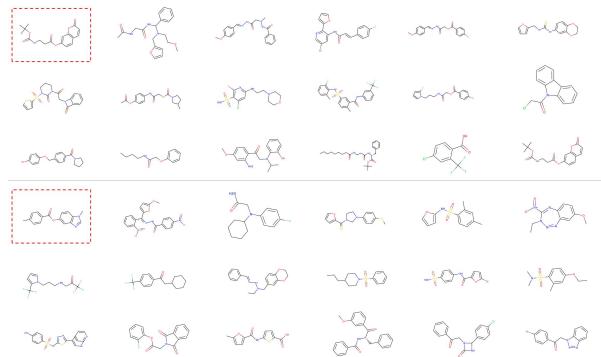


Fig. 5. Visualization of the generated molecules from two different seed molecules highlighted in red dashed box.

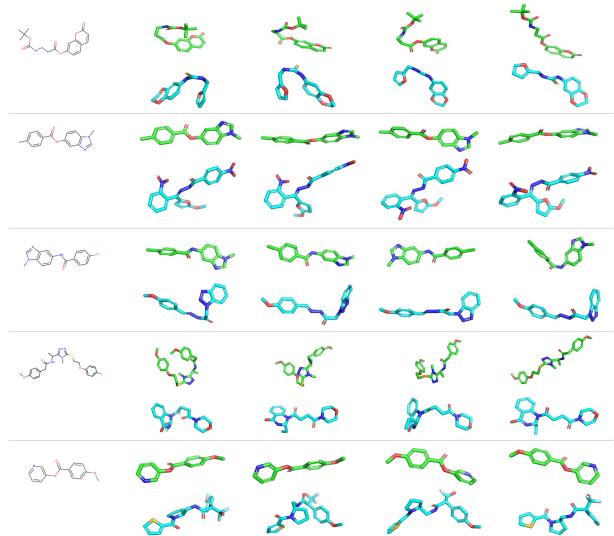


Fig. 6. Visualization of the conformers of random five different seed molecules and the de novo molecular conformations with the closest similarity from the seed molecules.

of molecules and generate effective and novel drug molecules through latent variable sampling and geometric embedding.

3) Molecular Chemical Properties Evaluation:

a) *Setup.*: It is an indispensable step to perform chemical drug attribute evaluation on the generated molecules. The chemical properties of the generated molecules are an important indicator for the quality evaluation of the molecular generation. If the drug has good properties, the generated molecule will be of high quality; otherwise, the quality will be poor or invalid. It is related to whether the generated drug molecules have drug-related attributes and can be considered as drugs for synthesis and use. First, evaluate whether the generated molecule has druggability. If it has druggability, then whether it is synthesizable, and if it is not synthesizable, then the molecule is ineffective; otherwise, further evaluate whether the generated molecule has the function of being absorbed.

b) *Results.*: We randomly sample 1000 generated drug molecules for quantitative estimation of drug-likeness (QED), SA, and logP. We finally obtained the evaluation results with the average values of QED, SA, and logP being 0.6700, 2.3761, and 2.7940, with the maximum values of which are 0.9443, 5.2723 and 5.892, respectively. In particular, we obtained the maximum QED value of 0.9443 compared to

TABLE II
GENERATIVE NOVEL MOLECULES WITH THE BEST QED SCORES.
BASELINE RESULTS ARE TAKEN FROM [22]

Model	1st	2nd	3rd	4th	5th
JT-VAE	0.925	0.911	0.910	-	-
GEOM-CVAE	0.9443	0.9425	0.9120	0.9111	0.9089

	SMILES	Image
1st	<chem>CCc1ccc(C(=O)Nc2ccc(S(C)(=O)=O)cc2)cc1</chem>	
2nd	<chem>Oc1cccc(OCc2ccc(C(=O)N3CCCC3)cc2)cc1</chem>	
3rd	<chem>Cc1ccc(S(=O)(=O)Nc2ccc2)c(C)c1</chem>	
4th	<chem>Cc1ccc(Cl)cc1C(=O)Nc1ccc(C(N)=O)cc1</chem>	
5th	<chem>CCC(C)CSC(=O)NN2C(=O)c1cccc21</chem>	

Fig. 7. Novel best molecules discovered by our method.

JT-VAE's maximum QED value of 0.925, and the molecules with the QED value of 0.9120 and 0.9111 are the most active molecules for the SARS-CoV 3CLPro. We report the discovered novel molecules sorted by QED scores in Table II and show the novel molecules in Fig. 7. Therefore, our model outperforms baselines by a large margin, implying the superiority of capturing 3-D molecular structures for molecular representation learning.

V. CONCLUSION

In this work, we aimed to construct deep constrained VAE based on geometric molecular structure. We propose a novel geometric molecular representation method for drug molecules and protein molecules. Although our method does not achieve the best performance on all metrics, our model outperforms previous 1-D- or 2-D-based molecule generation models on most metrics. We open a window for GDL in chemistry. We have taken an important step in the research of geometric molecule structure. For future work, we will continue to focus on the research of molecular generation and further improve our model to identify significant 3-D features and implement a Haar graph neural network for related molecular tasks [45], [46].

REFERENCES

- [1] S. M. Paul *et al.*, "How to improve R&D productivity: The pharmaceutical industry's grand challenge," *Nature Rev. Drug Discovery*, vol. 9, no. 3, pp. 203–214, Mar. 2010.
- [2] W. Jin, R. Barzilay, and T. Jaakkola, "Junction tree variational autoencoder for molecular graph generation," in *Proc. ICML*, 2018, pp. 2323–2332.
- [3] R. Gómez-Bombarelli *et al.*, "Automatic chemical design using a data-driven continuous representation of molecules," *ACS Central Sci.*, vol. 4, no. 2, pp. 268–276, 2018.
- [4] C. Li, W. Wei, J. Li, J. Yao, X. Zeng, and Z. Lv, "3DMol-Net: Learn 3D molecular representation using adaptive graph convolutional network based on rotation invariance," *IEEE J. Biomed. Health Informat.*, early access, Jun. 14, 2021, doi: [10.1109/JBHI.2021.3089162](https://doi.org/10.1109/JBHI.2021.3089162).
- [5] C. Li, J. Wang, Z. Niu, J. Yao, and X. Zeng, "A spatial-temporal gated attention module for molecular property prediction based on molecular geometry," *Briefings Bioinf.*, vol. 22, no. 5, p. bbab078, Sep. 2021, doi: [10.1093/bib/bbab078](https://doi.org/10.1093/bib/bbab078).
- [6] D. Weininger, "SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules," *J. Chem. Inf. Comput. Sci.*, vol. 28, no. 1, pp. 31–36, 1988.
- [7] J. You, B. Liu, R. Ying, V. Pande, and J. Leskovec, "Graph convolutional policy network for goal-directed molecular graph generation," in *Proc. NIPS*, 2018, pp. 1–12.
- [8] J. Liu, A. Kumar, J. Ba, and K. Swersky, "Towards permutation-invariant graph generation," in *Proc. Workshop Learn. Reasoning With Graph Struct. Data (ICML)*, 2019, pp. 1–8.
- [9] S. D. Axen, X.-P. Huang, E. L. Cáceres, L. Gendelev, B. L. Roth, and M. J. Keiser, "A simple representation of three-dimensional molecular structure," *J. Med. Chem.*, vol. 60, no. 17, pp. 7393–7409, Sep. 2017.
- [10] M. Skalic, J. Jiménez, D. Sabbadin, and G. De Fabritiis, "Shape-based generative modeling for de novo drug design," *J. Chem. Inf. Model.*, vol. 59, no. 3, pp. 1205–1214, Mar. 2019.
- [11] F. Imrie, A. R. Bradley, M. van der Schaar, and C. M. Deane, "Deep generative models for 3D linker design," *J. Chem. Inf. Model.*, vol. 60, no. 4, pp. 1983–1995, Mar. 2020.
- [12] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Proc. NIPS*, 2016, pp. 1–9.
- [13] J. Bruna, W. Zaremba, A. Szlam, and Y. Lecun, "Spectral networks and locally connected networks on graphs," in *Proc. ICLR*, 2014, pp. 1–14.
- [14] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, [arXiv:1609.02907](https://arxiv.org/abs/1609.02907).
- [15] H. Dai, Y. Tian, B. Dai, S. Skiena, and L. Song, "Syntax-directed variational autoencoder for structured data," 2018, [arXiv:1802.08786](https://arxiv.org/abs/1802.08786).
- [16] M. J. Kusner, B. Paige, and J. M. Hernández-Lobato, "Grammar variational autoencoder," 2017, [arXiv:1703.01925](https://arxiv.org/abs/1703.01925).
- [17] T. Ma, J. Chen, and C. Xiao, "Constrained generation of semantically valid graphs via regularizing variational autoencoders," 2018, [arXiv:1809.02630](https://arxiv.org/abs/1809.02630).
- [18] K. Märtens and C. Yau, "BasisVAE: Translation-invariant feature-level clustering with variational autoencoders," 2020, [arXiv:2003.03462](https://arxiv.org/abs/2003.03462).
- [19] N. De Cao and T. Kipf, "MolGAN: An implicit generative model for small molecular graphs," 2018, [arXiv:1805.11973](https://arxiv.org/abs/1805.11973).
- [20] M. Popova, M. Shvets, J. Oliva, and O. Isayev, "MolecularRNN: Generating realistic molecular graphs with optimized properties," 2019, [arXiv:1905.13372](https://arxiv.org/abs/1905.13372).
- [21] M. Olivecrona, T. Blaschke, O. Engkvist, and H. Chen, "Molecular de-novo design through deep reinforcement learning," *J. Cheminformatics*, vol. 9, no. 1, p. 48, Dec. 2017.
- [22] C. Zang and F. Wang, "MoFlow: An invertible flow model for generating molecular graphs," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2020, pp. 1–10.
- [23] S. Honda, H. Akita, K. Ishiguro, T. Nakanishi, and K. Oono, "Graph residual flow for molecular graph generation," 2019, [arXiv:1909.13521](https://arxiv.org/abs/1909.13521).
- [24] K. Madhawa, K. Ishiguro, K. Nakago, and M. Abe, "GraphNVP: An invertible flow model for generating molecular graphs," 2019, [arXiv:1905.11600](https://arxiv.org/abs/1905.11600).
- [25] M. Simonovsky and N. Komodakis, "GraphVAE: Towards generation of small graphs using variational autoencoders," in *Proc. ICANN*, 2018, pp. 412–422.
- [26] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," 2013, [arXiv:1312.6114](https://arxiv.org/abs/1312.6114).
- [27] S. E. Biehn and S. Lindert, "Accurate protein structure prediction with hydroxyl radical protein footprinting data," *Nature Commun.*, vol. 12, no. 1, pp. 1–10, Dec. 2021, doi: [10.1038/s41467-020-20549-7](https://doi.org/10.1038/s41467-020-20549-7).
- [28] F. F. Alam and A. Shehu, "Variational autoencoders for protein structure prediction," in *Proc. 11th ACM Int. Conf. Bioinf., Comput. Biol. Health Informat.*, Sep. 2020, pp. 1–10.
- [29] D. Varela and J. Santos, "Protein structure prediction in an atomic model with differential evolution integrated with the crowding niching method," *Natural Comput.*, Sep. 2020, doi: [10.1007/s11047-020-09801-7](https://doi.org/10.1007/s11047-020-09801-7).
- [30] M. Torrisi, G. Pollastri, and Q. Le, "Deep learning methods in protein structure prediction," *Comput. Struct. Biotechnol. J.*, vol. 18, pp. 1301–1310, Jan. 2020.
- [31] T. Bepler and B. Berger, "Learning protein sequence embeddings using information from structure," in *Proc. ICLR*, 2019, pp. 1–17.
- [32] S. Sanyal, I. Anishchenko, A. Dagar, D. Baker, and P. Talukdar, "ProteinGCN: Protein model quality assessment using graph convolutional networks," *bioRxiv*, Apr. 2020, doi: [10.1101/2020.04.06.028266](https://doi.org/10.1101/2020.04.06.028266).

- [33] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, "Geometric deep learning: Going beyond Euclidean data," *IEEE Signal Process. Mag.*, vol. 34, no. 4, pp. 18–42, Jul. 2017.
- [34] P. Gainza *et al.*, "Deciphering interaction fingerprints from protein molecular surfaces using geometric deep learning," *Nature Methods*, vol. 17, no. 2, pp. 184–192, Feb. 2020.
- [35] F. Monti, D. Boscaini, J. Masci, E. Rodolà, J. Svoboda, and M. M. Bronstein, "Geometric deep learning on graphs and manifolds using mixture model CNNs," 2016, *arXiv:1611.08402*.
- [36] M. F. Sanner, A. J. Olson, and J.-C. Spehner, "Reduced surface: An efficient way to compute molecular surfaces," *Biopolymers*, vol. 38, no. 3, pp. 305–320, Mar. 1996.
- [37] S. Yin, E. A. Proctor, A. A. Lugovskoy, and N. V. Dokholyan, "Fast screening of protein surfaces using geometric invariant fingerprints," *Proc. Nat. Acad. Sci. USA*, vol. 106, no. 39, pp. 16622–16626, Sep. 2009.
- [38] T. J. Dolinsky *et al.*, "PDB2PQR: Expanding and upgrading automated preparation of biomolecular structures for molecular simulations," *Nucleic Acids Res.*, vol. 35, pp. 522–525, Jul. 2007.
- [39] N. A. Baker, D. Sep., S. Joseph, M. J. Holst, and J. A. McCammon, "Electrostatics of nanosystems: Application to microtubules and the ribosome," *Proc. Nat. Acad. Sci. USA*, vol. 98, no. 18, pp. 10037–10041, Aug. 2001.
- [40] Q. Zhou. (2018). *PyMesh—Geometry Processing Library for Python*. [Online]. Available: <https://pymesh.readthedocs.io/en/latest/>
- [41] M. Garland and P. S. Heckbert, "Surface simplification using quadric error metrics," in *Proc. 24th Annu. Conf. Comput. Graph. Interact. Techn. (SIGGRAPH)*, 1997, pp. 209–216.
- [42] P. Pellizzoni and G. Savio, "Mesh simplification by curvature-enhanced quadratic error metrics," *J. Comput. Sci.*, vol. 16, no. 8, pp. 1195–1202, Aug. 2020.
- [43] Y. Li, O. Vinyals, C. Dyer, R. Pascanu, and P. Battaglia, "Learning deep generative models of graphs," 2018, *arXiv:1803.03324*.
- [44] D. Rogers and M. Hahn, "Extended-connectivity fingerprints," *J. Chem. Inf. Model.*, vol. 50, no. 5, pp. 742–754, Apr. 2010.
- [45] M. Li, Z. Ma, Y. G. Wang, and X. Zhuang, "Fast Haar transforms for graph neural networks," *Neural Netw.*, vol. 128, pp. 188–198, Aug. 2020.
- [46] Y. Wang, M. Li, Z. Ma, G. Montufar, X. Zhuang, and Y. Fan, "Haar graph pooling," in *Proc. ICML*, 2020, pp. 9952–9962.



Chunyan Li is currently pursuing the Ph.D. degree with the School of Informatics, Xiamen University, Xiamen, China.

She is also an Assistant Researcher with Yunnan Minzu University, Kunming, China. Her current research interests include graph neural network, generative model, and self-supervised learning in the field of bioinformatics.



Junfeng Yao received the B.S. and Ph.D. degrees from Central South University, Changsha, China, in 1995 and 2001, respectively.

He is currently a Professor with the School of Film, Xiamen University, China. His research interests include computer graphics, virtual reality, intelligent algorithm research, and industrial process simulation (for more information, visit <https://cdmc.xmu.edu.cn/info/1010/1062.htm>).



Wei Wei (Senior Member, IEEE) received the Ph.D. degree from Xi'an Jiaotong University, Xi'an, China, in 2010.

He is currently an Associate Professor with the School of Computer Science and Engineering, Xi'an University of Technology, Xi'an. He has over 100 papers published or accepted by international conferences and journals. His research interests include wireless networks, sensor networks, image processing, mobile computing, distributed computing, pervasive computing, the Internet of Things, sensor data, and cloud computing.

Dr. Wei is a Senior Member of the China Computer Federation (CCF).



Zhangming Niu is currently a Recognized Leader in artificial intelligence (AI) development with ten years of experience in building novel products for blue chip companies, governments, and institutions. He is also the Chief Executive Officer (CEO) of MindRank AI Ltd., Hangzhou, China, an AI drug discovery-based company.

Dr. Niu is ranked in the top ten of multiple global AI competitions for healthcare and industry applications. He held the Ninth Place in the Association for Computing Machinery Knowledge Discovery and Data Mining (ACM KDD) Cup in 2019 and the Sixth Place in the IEEE International Conference on Multimedia and Expo (ICME) Grand Challenge AI Competition in 2019 and was the Winner of the Tencent Advertising Algorithm AI Competition in 2019.



Xiangxiang Zeng (Senior Member, IEEE) received the Ph.D. degree in system engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2011.

He is currently an Yuelu Distinguished Professor with the College of Information Science and Engineering, Hunan University, Changsha, China. Before joining Hunan University in 2019, he was with the Department of Computer Science, Xiamen University, Xiamen, China. His main research interests include computational intelligence, graph neural networks, and bioinformatics.



Jin Li received the B.Sc. degree in computer science, the M.Sc. degree in computational mathematics, and the Ph.D. degree in telecommunication and information system from Yunnan University, Kunming, China, in 1998, 2004, and 2012, respectively.

He is currently with the National Pilot School of Software, Yunnan University, as a Professor of machine learning and bioinformatics. His research interests include machine learning and bioinformatics.



Jianmin Wang is currently pursuing the Ph.D. degree in integrative biotechnology and translational medicine with Yonsei University, Incheon, Republic of Korea.

Before joining Yonsei University in 2021, he was a Research Assistant with Hunan University, Changsha, China. His current research interests include deep learning for computational biology and drug discovery.