scientific reports

Check for updates

OPEN

# Geometry-aware lightweight convolutional network for efficient molecular property prediction

Huan Zhang[1], Guifei Zhou[1], Mingjing Tang[1], Jin Li[2], Hang Zhu[1] & Chunyan Li[1]✉

**Molecular representation learning (MRL) has demonstrated significant potential in various fields such as drug discovery, particularly in extracting molecular features under limited supervision. However, most existing approaches rely on one-dimensional sequences or two-dimensional topological structures, which fail to adequately capture the complexity of molecular three-dimensional (3D) geometry, thereby limiting their performance in complex property prediction tasks. To more effectively model spatial structural information, three-dimensional convolutional neural networks have recently gained attention in MRL research due to their ability to directly process voxelized 3D molecular data. Nevertheless, these methods often suffer from severe computational inefficiencies caused by the inherent sparsity of voxel data, resulting in a large number of redundant operations. In addition, the commonly used large convolutional kernels—though beneficial for increasing model capacity—introduce substantial computational overhead, which restricts scalability in practical applications. To address these challenges, we propose Prop3D, an efficient 3D molecular representation learning model. Prop3D adopts a kernel decomposition strategy that significantly reduces computational cost while maintaining high predictive accuracy. Experimental results on multiple public benchmark datasets demonstrate that Prop3D consistently outperforms several state-of-the-art methods in molecular property prediction. The source code is available at: https://github.com/zh-netizen/Prop3D**

Molecular representation learning (MRL)[1,2] has achieved remarkable progress in computational chemistry[3], drug design[4–6], and related fields, emerging as a pivotal research direction at the intersection of artificial intelligence and chemical sciences. The primary objective of MRL is to transform complex molecular structures into computer-processable vector representations or embeddings, thereby facilitating downstream tasks such as molecular property prediction[7] and drug screening.

Conventional MRL approaches[8] predominantly employ one-dimensional (1D)[9–13] continuous string representations (e.g., SMILES[9]) or two-dimensional (2D)[14–17] graph-based architectures for molecular modeling. For sequence-based representations, Smi2Vec[18] utilizes SMILES[9] sequences with LSTM/BiGRU networks for feature extraction, CheMixNet[10] integrates LSTM with multilayer perceptrons (MLPs) to incorporate molecular fingerprint features, while SMILES-BERT[19] adapts the BERT framework to enhance sequence modeling capabilities. Regarding graph-based representations, GC[20] employs graph convolutional operations for molecular embedding learning, and GraphCL[21] improves representation quality through graph contrastive learning. DFT-ANPD[22] employs 1D-CNN to extract molecular structural features and SMILES-BERT to obtain semantic features. These are fused using a two-sided attention mechanism, followed by a fully connected layer to predict the anticancer potential of compounds.

Although these methodologies have demonstrated substantial performance advantages in molecular property prediction tasks, their inherent neglect of three-dimensional (3D) structural information may constitute a critical limitation for further performance enhancement and application expansion. From a life sciences perspective, molecular physicochemical properties and drug bioactivities are fundamentally determined by their 3D conformations. Nevertheless, current MRL frameworks have not yet fully incorporated 3D molecular information, which consequently restricts their capacity to characterize complex molecular features and effectively adapt to downstream applications. Figure 1 illustrates both the 2D representation and 3D molecular structure of the compound with SMILES notation CN1C=NC2=C1C(=O)N (C(=O)N2C)C.

In recent years, significant progress has been made in the field of 3D molecular representation learning. Voxels, as a common three-dimensional molecular representation, effectively preserve the geometric information of molecules by mapping atoms onto one or multiple grid units within a 3D voxel space. Under such representations,

[1]School of Informatics, Yunnan Normal University, Kunming, China. [2]School of Software, Yunnan University, Kunming 650091, China. ✉email: lchy@ynnu.edu.cn

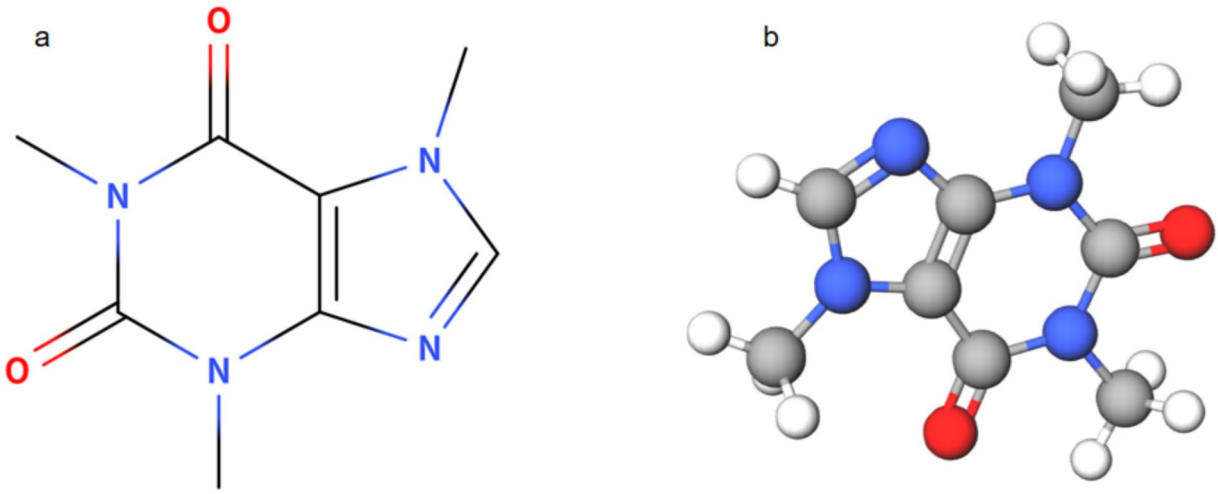


**Fig. 1**. **a** 2D structure of the compound along with its SMILES representation: CN1C=NC2=C1C(=O)N(C(=O)N2C)C. **b** 3D geometric structure of the compound, with its conformation constructed based on the same SMILES sequence.

three-dimensional convolutional neural networks (3D CNNs) have been widely adopted for spatial feature extraction, demonstrating strong modeling capabilities. Specifically, Casey et al.[23] constructed two specialized 3D grids based on electron charge density and electrostatic potential, employing rotation augmentation strategies to expand the dataset and enhance model generalization. Tran et al.[24] separately constructed 3D grids for six common elements—carbon, hydrogen, oxygen, nitrogen, sulfur, and chlorine—using CNN autoencoders for molecular feature learning. The Multi-resolution 3D-DenseNet[25] generates 3D grids based on Gaussian density models centered on atoms and enhances the model's hierarchical structural perception through multi-channel and multi-scale (4–14 Å) inputs. Meanwhile, Kuzminykh et al.[26] pointed out that the high sparsity of traditional 3D grids severely hampers the efficiency of convolution operations and thus proposed a smoothing strategy based on wavelet transforms to fill adjacent voxels and alleviate sparsity issues. Additionally, Drug3D-Net[27] combines convolutional neural networks with a spatiotemporal gated attention module to effectively exploit the 3D structural information of molecules, while ATOM3DCNN[28], as a baseline model, extracts spatial features of molecules via 3D convolution operations.

Despite the certain achievements of voxel-based and 3D CNN methods in modeling three-dimensional molecular structures, notable challenges remain. First, molecules exhibit highly sparse distributions in 3D space, resulting in voxel grids containing substantial invalid empty regions. This leads to significant redundant computations during convolution operations, causing considerable waste of computational resources and limiting the efficiency of feature extraction. Second, to enhance the expressive power of models and expand the receptive field, existing methods commonly rely on enlarging convolution kernel sizes. However, employing large-scale convolution kernels in 3D space markedly increases computational complexity and memory demands, severely restricting the applicability of these models in large-scale data processing and resource-constrained environments. Therefore, achieving a balance between improving model performance and effectively controlling computational resource consumption remains a critical problem that needs to be addressed in this domain.

To address the above issues, this study proposes a molecular representation learning (MRL) model for property prediction named Prop3D, which is based on convolutional neural networks (CNNs)[29–32]. Compared with traditional three-dimensional convolutional neural networks, Prop3D adopts a more lightweight convolution design that significantly reduces computational cost and the number of parameters through a kernel decomposition strategy, thereby improving computational efficiency. The model is built around three core modules to achieve efficient molecular feature learning. First, the model encodes molecular structures into regularized 3D grid data based on their 3D coordinate information, preserving spatial geometric features. Then, a standard 3D CNN is used to perform channel expansion and information fusion on the input 3D grid data to enhance interactions between different channels. Meanwhile, inspired by the InceptionNeXt[33] design, large convolution kernels are decomposed in 3D space to balance efficiency and computational resource consumption. Additionally, a channel and spatial attention mechanism (CBAM)[34] is integrated after each convolutional module to focus on key features and improve the generalization capability of the model. By combining 3D grid representation, channel information fusion, efficient kernel decomposition, and attention mechanisms, the Prop3D model provides an efficient and robust solution for molecular property prediction.

We systematically evaluated Prop3D on multiple publicly available datasets, including both structured datasets with atomic 3D coordinates and unstructured datasets without spatial information. Experimental results demonstrate that Prop3D shows significant performance advantages in molecular property prediction tasks, with prediction accuracy noticeably surpassing many state-of-the-art molecular representation learning models. As an MRL method based on three-dimensional spatial structure, Prop3D not only exhibits excellent

robustness but also achieves an optimal balance between model performance and computational efficiency, which endows it with significant practical value in real-world applications.

Our main contributions are as follows:

- We model the molecular datasets as 3D grids, which include both structured and unstructured datasets, in order to effectively preserve the 3D spatial information of the molecules, ensuring that their spatial geometric features are fully utilized during the model learning process.
- We propose Prop3D, an efficient molecular representation learning method based on convolutional neural networks. Notably, Prop3D is the first to introduce the large kernel decomposition strategy from InceptionNeXt into this domain. By optimizing the structure of large kernels, this design significantly reduces computational costs while maintaining a balance between model performance and resource consumption.
- We also conducted a systematic evaluation of Prop3D using multiple publicly available datasets. Experimental results show that, compared to existing state-of-the-art molecular representation learning methods, Prop3D achieves superior performance.

## Related work

### Large kernel convolutions and optimization

In recent years, the study of large-kernel convolutions has attracted increasing attention. Early representative models such as AlexNet[35] and Inception v1[36] employed large convolution kernels of 11×11 and 7×7, respectively, to enhance the perception of local spatial features. Subsequently, to improve computational efficiency, VGG[30] replaced large kernels with stacks of multiple 3×3 convolutions, while Inception v3[37] introduced a factorization approach that decomposed conventional k×k convolutions into sequential 1×k and k×1 convolutions, significantly reducing computational cost. ConvNeXt adopted a default configuration of 7×7 depthwise convolutions in its architecture, balancing performance and efficiency. To further explore the potential of large-kernel convolutions, RepLKNet[38] utilized structural re-parameterization techniques[39] to successfully scale the kernel size up to 31×31. VAN[40] achieved an effective receptive field of 21×21 by cascading large-kernel depthwise convolutions with dilated depthwise convolutions. More recently, SLaK[41] proposed decomposing the large k×k convolution into two asymmetric kernels (k×s and s×k), thereby improving computational efficiency. Meanwhile, InceptionNeXt[33] introduced a novel channel-wise decomposition strategy, splitting the large-kernel convolution into four parallel branches: a small square kernel, two orthogonal strip-shaped large kernels, and an identity mapping. This design effectively enhances throughput while maintaining strong performance.

Although InceptionNeXt has demonstrated superior performance in the field of image processing, its potential in 3D molecular representation learning has yet to be systematically explored. This paper aims to extend the design philosophy of large-kernel convolution decomposition in InceptionNeXt to 3D space, proposing a 3D large-kernel decomposition method that is computationally efficient, structurally simple, and speed-friendly. Under the premise of maintaining competitive performance, the proposed approach seeks to enhance the applicability and practical value of convolutional neural networks in 3D molecular representation learning tasks.

### Spatiotemporal attention mechanism

Since the introduction of the attention mechanism, it has made significant progress across various fields such as Natural Language Processing (NLP)[42] and Computer Vision (CV)[43], gradually becoming one of the core technologies. The additive attention mechanism, first proposed by Bahdanau[44], addressed the bottleneck problem of Seq2Seq models in handling long input sequences for neural machine translation tasks, significantly improving the modeling of long-range dependencies. Later, the Transformer model[45] introduced by Vaswani, incorporating self-attention and multi-head attention mechanisms, greatly advanced research in NLP. Furthermore, the application of Graph Attention Networks (GAT)[46] to graph-structured data extended the applicability of attention mechanisms. GAT assigns different weights to each node, avoiding expensive matrix operations and reliance on prior knowledge of graph structures, thus effectively enhancing the processing ability of graph data.

CBAM[34], an attention module designed for CNNs, aims to enhance the expressive power of feature maps. By introducing attention mechanisms in both the spatial and channel dimensions, CBAM has achieved significant performance improvements in image classification, object detection, and other image processing tasks. However, despite its widespread application in 2D image processing, research on its extension and application in 3D spaces, particularly in tasks like molecular representation learning, remains limited. Therefore, this study aims to effectively integrate the CBAM attention module into 3DCNNs and explore its potential in MRL tasks in 3D space.

## Materials and methods

### Problem definition

We consider a batch of training samples of size $N_{\text{batch}}$, where each sample input is a 3D molecular grid with dimensions $N_{\text{grid}} \times N_{\text{grid}} \times N_{\text{grid}}$ and $N_{\text{channel}}$ feature channels. Thus, the input data can be represented as:

$$X \in \mathbb{R}^{N_{\text{batch}} \times N_{\text{channel}} \times N_{\text{grid}} \times N_{\text{grid}} \times N_{\text{grid}}} \tag{1}$$

The output is a scalar representing the molecular property to be predicted. This can correspond to either a binary classification or a regression task. For classification, the model outputs a probability in the range $[0, 1]$, indicating the likelihood of a positive label; for regression, the output is a continuous scalar value. In both cases, the output has shape:

$$y_{\text{pred}} \in \mathbb{R}^{N_{\text{batch}} \times 1} \tag{2}$$

The goal is to learn a mapping function $f_{\text{pred}}$ parameterized by $\theta$, which transforms the input 3D molecular grid $X$ into the predicted molecular property $y_{\text{pred}}$:

$$f_{\text{pred}} : X \mapsto y_{\text{pred}}, \quad f_{\text{pred}}(X;\theta) \tag{3}$$

The training process optimizes parameters $\theta$ by minimizing a loss function $L$, measuring the discrepancy between the predicted output and ground truth $y$:

$$\theta^* = \arg\min_{\theta} L(f_{\text{pred}}(X;\theta), y) \tag{4}$$

where $y \in \mathbb{R}^{N_{\text{batch}} \times 1}$ is the ground truth label.

## Molecular 3D grid construction

In this study, we conduct modeling and evaluation using two types of molecular datasets: one that provides atomic-level 3D structural information and another that contains only SMILES representations. We adopt the preprocessing strategies proposed in ATOM3D[28] and Drug3D-Net[27] to uniformly convert molecular structures into 3D voxel grid representations. For datasets with 3D coordinates, voxel grids are constructed with the molecular geometric center as the origin, and random rotations are applied for conformational augmentation, followed by coordinate quantization based on a predefined resolution. For datasets containing only SMILES strings, low-energy 3D conformations are generated using RDKit and discretized into voxel grids. Atoms are assigned to separate channels according to their types, effectively encoding the spatial structure and elemental distribution of the molecule, and providing rich 3D geometric features for downstream modeling.

## Prop3D

Prop3D is an innovative 3D molecular representation learning method specifically designed for molecular property prediction tasks in 3D space. The method uses 3D Convolutional Neural Networks as its core framework. The core idea is to model the molecule as a 3D grid and enhance its features through operations such as rotation and translation. To better capture the spatial features of the molecule at different scales, Prop3D introduces large kernel convolutions as the primary feature extraction module and employs an efficient large kernel decomposition strategy. By stacking multiple such convolutional layers, Prop3D can progressively abstract higher-level spatial features. To further enhance the selectivity and importance of features, Prop3D incorporates the CBAM attention module after each convolutional neural network layer. Finally, Prop3D uses a Multi-Layer Perceptron (MLP) as the predictor for downstream tasks. Through an end-to-end training approach, Prop3D directly learns the mapping relationship from the molecular 3D structure to the target properties, providing powerful tool support for molecular property prediction. Figure 2 presents the 3D mesh generation process employed by Prop3D, as well as the architectural framework of its principal feature extraction module.

Recently, the large kernel decomposition concept proposed by InceptionNeXt[33] has achieved success in the field of computer vision. In order to apply large kernel convolutions to molecular representation learning, this study transfers the decomposition strategy to 3D space, achieving the decomposition of 3D convolutional kernels. Figure 3 illustrates the large kernel decomposition method employed in this study.

The following describes in detail the strategy for decomposing convolutional neural network kernels in this paper. The input data is represented as $X \in R^{B \times C \times D \times H \times W}$, where *B* denotes the batch size, *C* represents the number of channels, and *D*, *H*, and *W* correspond to the depth, height, and width of the input data, respectively.

Firstly, the input data is divided into several parts along the channel dimension, with each part undergoing independent convolution to perform different types of convolution operations. Specifically, the input tensor is divided into the following five parts:

$$Split(X) = X_{id}, X_{hwd}, X_{hw}, X_{wd}, X_{hd} \tag{5}$$

In order to partition the original data along the channel dimension, we use a channel ratio *i* to control the number of channels in each part. The parts $X_{hwd}$, $X_{hd}$, $X_{hw}$, and $X_{wd}$ have the same number of channels. The mathematical expression for this calculation is as follows:

$$C_{hwd} = \lfloor C \times i \rfloor \tag{6}$$

$$C_{id} = C - C_{hwd} - C_{hw} - C_{hd} - C_{wd} \tag{7}$$

where $C_n$ represents the number of channels in the $X_n$ part of the input data. For example, $C_{id}$ represents the number of channels in the $X_{id}$ part of the input data. *C* is the total number of channels in the input data.

For the input data of the $X_{id}$ part, no convolution operation is performed. Instead, the original values are preserved through an identity mapping. This strategy aims to avoid unnecessary computational overhead.The process of this operation can be represented by the following formula:

$$X_{id}^{'} = X_{id} \tag{8}$$

For the input data $X_{hwd}$, a depthwise convolution operation is applied with a cubic kernel of size $k_{\text{cube}} \times k_{\text{cube}} \times k_{\text{cube}}$. This operation performs 3D convolution independently on each input channel group

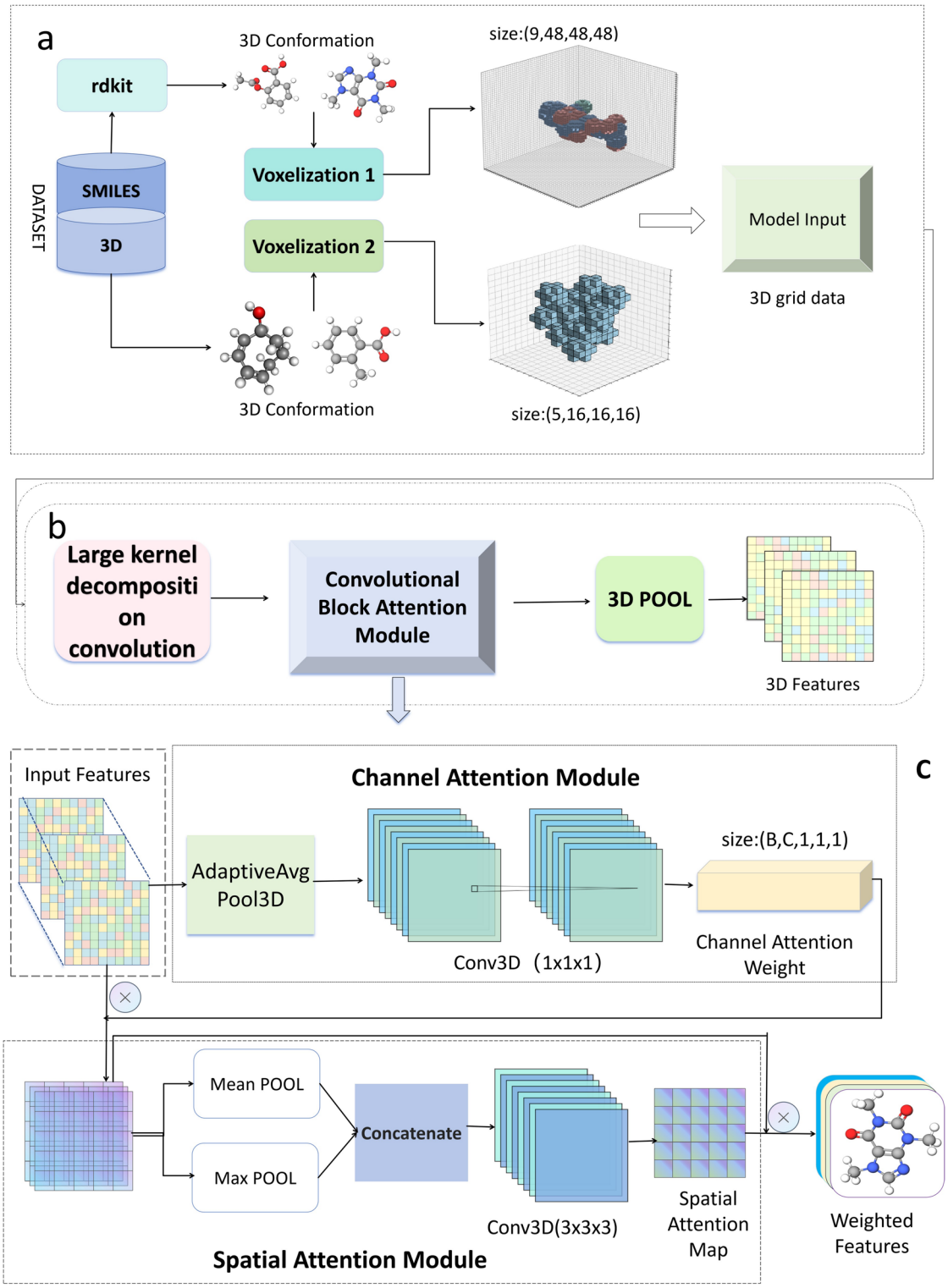


**Fig. 2**. This figure shows the core architecture of the proposed method: **a** voxelization to structure molecular data in 3D; **b** a feature extraction module using efficient kernel decomposition to capture rich 3D representations; and **c** the CBAM attention mechanism, with channel and spatial modules to emphasize important features.

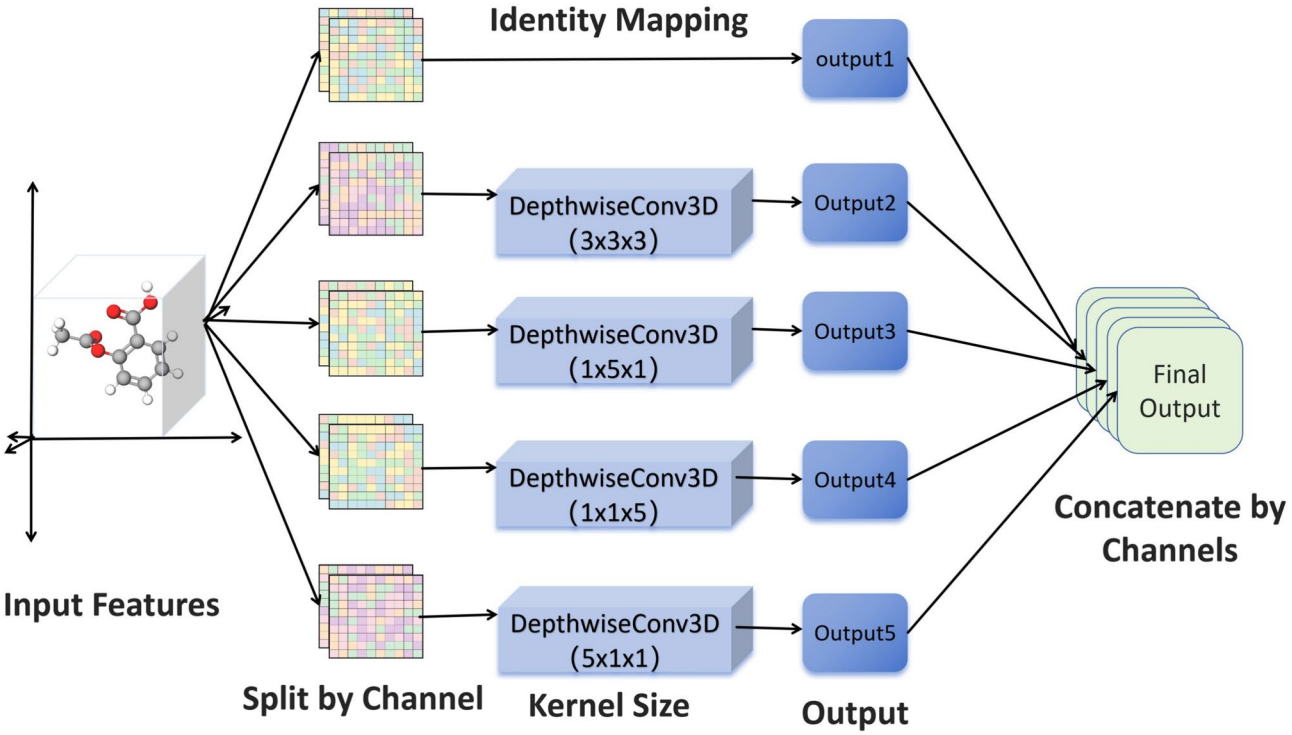


**Fig. 3**. This figure illustrates the large kernel convolution decomposition strategy in 3D space.

without mixing information across channels, thereby significantly reducing computational cost. To control the number of parameters and computation overhead, we set $k_{\text{cube}} = 3$ in this work. The operation can be expressed as:

$$X'_{hwd} = \text{DepthwiseConv3D}(X_{hwd}, k_{\text{cube}} \times k_{\text{cube}} \times k_{\text{cube}}) \tag{9}$$

For the input data $X_{wd}$, $X_{hd}$, and $X_{hw}$, we employ depthwise convolutions with striped kernels for feature extraction. Compared to traditional cuboid convolution kernels, these striped kernels decouple 3D spatial convolution into directional 1D convolutions along different axes. This design greatly reduces computational complexity while expanding the receptive field along three orthogonal directions, enabling efficient multi-scale spatial feature extraction. In our experiments, the kernel size $k$ is set to 5. The mathematical formulation is:

$$X'_{wd} = \text{DepthwiseConv3D}(X_{wd}, 1 \times 1 \times k) \tag{10}$$

$$X'_{hd} = \text{DepthwiseConv3D}(X_{hd}, 1 \times k \times 1) \tag{11}$$

$$X'_{hw} = \text{DepthwiseConv3D}(X_{hw}, k \times 1 \times 1) \tag{12}$$

Here, $X'_n$ denotes the output tensor obtained after applying depthwise convolution independently to the $X_n$ part of the input data. For example, $X'_{wd}$ denotes the output after depthwise convolution on the $X_{wd}$ part.

Finally, the outputs from all five branches—including the identity mapping branch $X'_{id}$—are concatenated along the channel dimension to obtain the final output tensor:

$$X' = \text{Concat}(X'_{id}, X'_{hwd}, X'_{wd}, X'_{hd}, X'_{hw}) \tag{13}$$

$$X' \in \mathbb{R}^{B \times C' \times D \times H \times W} \tag{14}$$

The concatenation operation integrates the feature maps processed independently by the five parallel branches along the channel dimension, forming the output feature representation of this convolutional layer. By appropriately allocating the number of channels in each branch, the output channel number $C'$ is kept the same as the input.

in this architecture, the concatenation operation (Concat) combines the feature maps processed independently by five parallel branches along the channel dimension, ultimately integrating them to form the output feature representation of this convolutional layer. $C'$ denotes the new number of channels.the number of channels in the output tensor is the same as the input, because the number of channels in the convolution operations is allocated and eventually concatenated back to the original channel count.

## Computational complexity analysis

Under the premise of consistent input data dimensions, this section systematically compares the performance differences in time and space complexity between the conventional direct large-kernel convolution approach in convolutional neural networks and the large-kernel decomposition strategy proposed herein. The time complexity analysis, grounded in computational complexity theory, rigorously derives and quantitatively compares the floating-point operations (FLOPs) of the two methods. Regarding space complexity, parameter count analytical expressions are constructed to comprehensively evaluate the differences in model parameter scale and memory consumption between the two approaches.

Let the input tensor be defined as

$$X \in \mathbb{R}^{C \times H \times W \times D} \tag{15}$$

In this work, $C$ denotes the number of channels, which is set to 64. The variables $H$, $W$, and $D$ represent the height, width, and depth of the input data, respectively. For example, at this layer of the neural network, when using voxelized data constructed from the QM9 dataset, the spatial dimensions are set to $H = W = D = 16$. The kernel sizes along the height, width, and depth dimensions are denoted by $K_H$, $K_W$, and $K_D$, respectively.

*Time complexity*

Time complexity of standard convolution In a standard 3D convolution operation, the input tensor is denoted as $X \in \mathbb{R}^{C \times H \times W \times D}$, and the resulting output tensor is $Y \in \mathbb{R}^{C \times H \times W \times D}$, where the number of input and output channels is the same and set to $C = 64$, and the spatial dimensions are $H = W = D = 16$. The kernel size is configured as $K_H = K_W = K_D = 5$, representing the kernel's size in the height, width, and depth dimensions, respectively.

At each spatial position of every output channel, the convolution operation iterates over all input channels and performs multiply-accumulate operations within a $K_H \times K_W \times K_D$ neighborhood. Therefore, the total computational complexity of this operation can be expressed as:

$$T_{\text{std}} = C^2 \cdot H \cdot W \cdot D \cdot K_H \cdot K_W \cdot K_D$$

By substituting the specific values, we obtain:

$$T_{\text{std}} = 64^2 \cdot 16^3 \cdot 5^3 = 4096 \cdot 110592 \cdot 125 = 262,144,000\text{FLOPs}$$

The time complexity of the convolution method used in this work We partition the input tensor *X* along the channel dimension into five sub-tensors, denoted as $\{X^{(i)}\}_{i=1}^{5}$, where the *i*-th sub-tensor has $C_i$ channels satisfying $\sum_{i=1}^{5} C_i = C$. Each sub-tensor corresponds to an independent transformation strategy, specified as follows:

- $X^{(1)} \in \mathbb{R}^{C_1 \times H \times W \times D}$ bypasses any convolutional transformation and is directly used as part of the output;
- For $X^{(i)} \in \mathbb{R}^{C_i \times H \times W \times D}$ where $i = 2, 3, 4, 5$, we apply 3D convolutional kernels with shape $K_H^{(i)} \times K_W^{(i)} \times K_D^{(i)}$, while maintaining identical input and output channel dimensions.

For the *i*-th sub-module ($i = 2, 3, 4, 5$), the theoretical computational complexity of its corresponding convolution operation can be expressed as:

$$T^{(i)} = \left( C_i \cdot H \cdot W \cdot D \cdot K_H^{(i)} \cdot K_W^{(i)} \cdot K_D^{(i)} \right), \quad i = 2, 3, 4, 5$$

The first sub-module, which introduces no convolution operations but only involves tensor copying or remapping, has computational complexity:

$$T^{(1)} = (C_1 \cdot H \cdot W \cdot D)$$

Therefore, the overall time complexity is the sum of computational costs across all sub-modules:

$$T_{\text{total}} = \sum_{i=1}^{5} T^{(i)} = \left( H \cdot W \cdot D \cdot \left[ C_1 + \sum_{i=2}^{5} \left( C_i \cdot K_H^{(i)} \cdot K_W^{(i)} \cdot K_D^{(i)} \right) \right] \right)$$

where $T_{\text{total}}$ represents the computational complexity of the large-kernel convolution with kernel decomposition strategy.

To provide a more concrete illustration of the computational cost, we substitute representative parameter values into the above formula. Specifically, the input channel number is set as $C = 64$, with each convolution branch adopting a channel ratio of 0.125. This results in $C_1 = 32$, and the remaining four convolution branches each having $C_2 = C_3 = C_4 = C_5 = 8$. The spatial dimensions of the input feature map are $H = W = D = 16$, where the HWD branch employs a cubic kernel of size $3 \times 3 \times 3$, and the other three directional branches use kernels of size 5 along a single axis.

Based on this configuration, the computational costs of each sub-module and the total complexity are:

$$\begin{aligned} T_{\text{total}} &= T^{(1)} + T^{(2)} + T^{(3)} + T^{(4)} + T^{(5)} \\ &= 131,072 + 884,736 + 163,840 + 163,840 + 163,840 \\ &= 1,507,328 \quad \text{FLOPs} \end{aligned}$$

In contrast, under the same input size and kernel dimensions, a standard 3D convolution requires approximately $262,144,000$ floating-point operations, demonstrating a significant efficiency gain of the proposed method.

*Space complexity*

Space complexity of standard convolution In a standard 3D convolution operation, the space complexity primarily consists of the memory required to store the output tensor and the convolutional weights. Given an input tensor $X \in \mathbb{R}^{C\times H\times W\times D}$, where $C$ is the number of input channels, and $H$, $W$, and $D$ are the spatial dimensions (height, width, and depth), the output tensor $Y$ has the same shape as the input, i.e., $C \times H \times W \times D$ . The convolutional weights have the shape $\mathbb{R}^{C\times C\times K_H\times K_W\times K_D}$, where $K_H$, $K_W$, and $K_D$ denote the kernel sizes along height, width, and depth respectively, because in a standard convolution each output channel connects to all input channels.

Therefore, the total space complexity, denoted as $S_{\text{std}}$, which represents the total memory footprint of the convolutional layer including both output activations and trainable parameters, can be expressed as:

$$S_{\text{std}} = (C \cdot H \cdot W \cdot D + C^2 \cdot K_H \cdot K_W \cdot K_D)$$

Here, the first term $C \cdot H \cdot W \cdot D$ corresponds to the memory required to store the output activations, and the second term $C^2 \cdot K_H \cdot K_W \cdot K_D$ corresponds to the memory required for storing the convolutional kernel weights.

By substituting the specific values $C = 64$, $H = W = D = 16$, and $K_H = K_W = K_D = 5$, we have:

$$S_{\text{std}} = 64 \times 16^3 + 64^2 \times 5^3 = 262,144 + 512,000 = 774,144 \quad \text{elements}$$

where "elements" refers to the total number of storage units needed for both the output tensor and the convolutional kernels, representing the overall memory cost of the standard 3D convolutional layer.

Space complexity of the convolution method used in this work In the proposed method, the input tensor is split into five channel-wise branches with a total of $C = 64$ channels, where $C_1 = 32$ channels are passed through an identity mapping (no parameters), and the remaining four convolutional branches each contain $C_i = 8$ channels ($i = 2, 3, 4, 5$), only these four branches have learnable parameters. Each convolutional branch employs depthwise convolutions with kernel weights of shape $C_i \times 1 \times K_H^{(i)} \times K_W^{(i)} \times K_D^{(i)}$, where $K_H^{(i)}, K_W^{(i)}, K_D^{(i)}$ represent the height, width, and depth of the kernel for the $i$-th branch. The overall space complexity is

$$S_{\text{ours}} = \left( C \cdot H \cdot W \cdot D + \sum_{i=2}^{5} C_i \cdot K_H^{(i)} \cdot K_W^{(i)} \cdot K_D^{(i)} \right)$$

where $H, W, D$ denote the spatial dimensions of the input. Substituting $C = 64$, $H = W = D = 16$, and kernel sizes $K^{(2)} = 3 \times 3 \times 3 = 27$, $K^{(3)} = K^{(4)} = K^{(5)} = 5$, we obtain

$$S_{\text{ours}} = 64 \cdot 16^3 + (8 \cdot 27 + 3 \cdot 8 \cdot 5) = 262,144 + 336 = 262,480 \quad \text{elements.}$$

The above analysis demonstrates the significant advantages of the proposed convolution strategy in both computational and memory efficiency. By decomposing a large 3D kernel into multiple lightweight, channel-wise submodules—including an identity mapping and depthwise convolutions—our method reduces the number of floating-point operations from over 262 million to just 1.5 million, and lowers the memory consumption by approximately 66.1%. More importantly, this design effectively mitigates the computational redundancy caused by the inherent sparsity of voxelized small-molecule data, ensuring that the model computation is concentrated on informative regions rather than wasted on empty space. This makes the method particularly suitable for efficient 3D representation learning in sparse molecular environments.

To more intuitively evaluate the effectiveness of the proposed convolutional kernel decomposition strategy in saving computational resources, this study takes the training process of the u298-atom task from the QM9 dataset as an example. We replaced all convolutional layers in the Prop3D model that use the kernel decomposition strategy with standard 3DCNN and measured the average training time over 50 epochs. The experimental results show that the Prop3D model requires approximately 235 seconds per epoch on average, whereas the standard 3DCNN model takes about 280 seconds. Regarding memory usage, the Prop3D model consumes approximately 3560 MB of GPU memory, while the standard 3DCNN model uses about 4480 MB. More importantly, as the convolutional kernel size and voxel resolution increase, the advantages of the proposed kernel decomposition strategy in reducing computational resources and memory consumption become increasingly significant. These results indicate that the proposed method can substantially reduce computational burden, improve training efficiency, and enhance model scalability when handling more complex 3D molecular data, thereby increasing its applicability and practical value in real-world scenarios.

# Experiments

## Datasets

To evaluate the effectiveness of Prop3D, we selected the QM9[47] dataset as a benchmark that provides 3D molecular information. For datasets that do not provide 3D molecular coordinates, we chose several commonly used molecular datasets recommended by MoleculeNet[48], including ESOL[49], FreeSolv[50], and Tox21[51]. Among these, ESOL and FreeSolv are regression task datasets, while Tox21 involves 12 classification tasks. Detailed information about each dataset is provided in Table 1.

ESOL: The ESOL dataset contains SMILES strings and aqueous solubility data (logarithmic solubility in mol/L) for 1128 drug molecules. The ESOL dataset is commonly used for regression analysis in neural networks. However, it does not include the geometric structure information of the molecules.

FreeSolv: The FreeSolv database provides computed and experimental hydration free energy data for 642 small molecules in water.

Tox21: The "Toxicology in the 21st Century" dataset is used to measure the toxicity of compounds and contains toxicity information for 7831 drug molecules across 12 different targets. Tox21 was used in the 2014 Tox21 Data Challenge .

QM9: The QM9 dataset is a database containing approximately 134,000 small organic molecules and their physicochemical properties. The molecules consist of 3 to 9 atoms (such as H, C, N, O, and F). The data comes from high-precision quantum chemical calculations and is primarily used in quantum chemistry, molecular modeling, and machine learning, especially for molecular property prediction and design. The task is to predict physicochemical properties (such as energy, polarizability, and rotational constants) based on the ground-state geometric structure of the molecules, with all tasks being regression tasks.

## Data preprocessing

For datasets providing atomic-level 3D coordinates, we follow the method proposed in ATOM3D[28] to convert molecules into 3D voxel grid representations. The grid is centered at the molecular geometric center, and random rotations are applied to the atomic coordinates for data augmentation. Coordinates are then quantized based on a predefined voxel resolution (1.0 Å) and mapped onto the spatial grid. Atoms of different types (H, C, O, N, F) are assigned to separate channels, resulting in a 5-channel 3D grid. With a grid radius of 7.5 Å, the final input tensor has a shape of 16×16×16×5, effectively preserving spatial structural information for downstream model learning.

For datasets that only provide SMILES representations, we follow the approach of Drug3D-Net[27], using RDKit[52] to generate the lowest-energy 3D conformations based on molecular topology. These conformations are discretized into voxel grids with a resolution of 0.5 Å and a size of 48×48×48. Atoms are assigned to 9 separate channels according to their types (C, N, O, F, P, S, Cl, Br, I), forming a 9-channel 3D representation. Random rotations and translations are applied to enhance spatial invariance. The ESOL[49], FreeSolv[50], and Tox21[51] datasets use preprocessed grid data provided by Drug3D-Net. Figure 4 demonstrates the modeling of the compound Triamcinolone Acetonide as 3D grid data with varying dimensions.

## Training and evaluation settings

*Pretraining setting*

For all datasets, the data was partitioned into training, validation, and test sets in a ratio of 8:1:1. During training on the QM9 dataset, the Adam optimizer was employed with an initial learning rate of 0.0001. The model was trained for 50 epochs with a batch size of 256 and consisted of three stacked large-kernel convolutional layers to enhance feature extraction. For the remaining datasets, the optimizer and learning rate settings remained unchanged. The model architecture included two stacked large-kernel convolutional layers, and training was conducted for 50 epochs with a batch size of 10.

*Implementation details*

Prop3D is based on the PyTorch[53] deep learning framework. We use RDKit[52], a cheminformatics software for molecular-related tasks. All training, validation, and testing processes are carried out on an NVIDIA 4090 GPU.

*Evaluation*

To comprehensively evaluate the performance of Prop3D across various molecular tasks, we conducted systematic experiments on four benchmark datasets: QM9, ESOL, FreeSolv, and Tox21. For the QM9 dataset, we adopted the same data splitting strategy as ATOM3D, dividing the dataset into training, validation, and test sets in a ratio of 8:1:1 to ensure comparability of the results. Mean Absolute Error (MAE) was used as both the training loss and the primary evaluation metric. We strictly followed the evaluation protocol established by

| DATASET | DATA TYPE | Number of task | Task type | Number | Metric |
|---|---|---|---|---|---|
| ESOL | SMILES | 1 | Regression | 1128 | RMSE |
| FreeSolv | SMILES | 1 | Regression | 642 | RMSE |
| Tox21 | SMILES | 12 | Classification | 7831 | ROC-AUC |
| QM9 | 3D STRUCTURE | 10 | Regression | 129433 | MAE |

**Table 1**. Dataset Description, including data type, number of tasks, task type, data size, and data split information for each dataset.

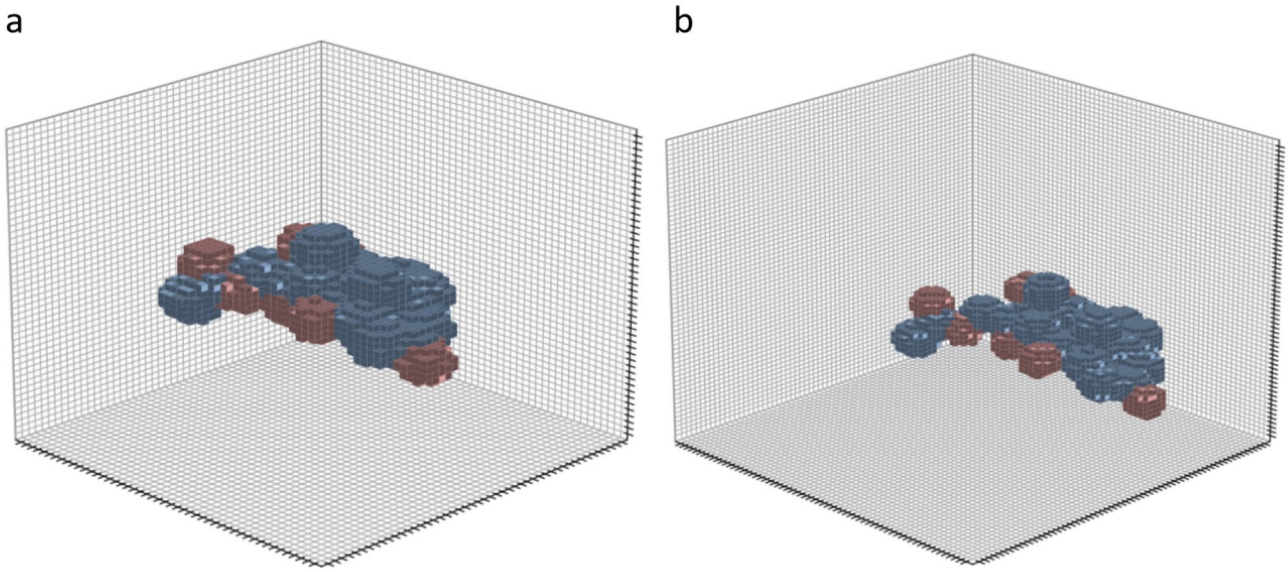


**Fig. 4**. The figure illustrates the compound of Triamcinolone Acetonide at different grid sizes. **a** The grid size is set to 48, presenting a coarser molecular structure representation. **b** The grid size is increased to 64, providing a denser and more detailed molecular structure representation compared to the grid size of 48.

ATOM3D, where each experiment was repeated three times using different random seeds, and the results were reported as the mean and standard deviation to ensure stability and reliability.

For the ESOL, FreeSolv, and Tox21 datasets, we also adopted an 8:1:1 split and ensured that all baseline models used for comparison followed the same data partitioning scheme, guaranteeing fairness in the comparisons. Specifically, for the ESOL and FreeSolv datasets, Mean Squared Error (MSE) was used as the training loss function and Root Mean Squared Error (RMSE) as the evaluation metric. For the classification task on the Tox21 dataset, we used cross-entropy as the loss function and the average Area Under the Receiver Operating Characteristic Curve (AUROC) as the evaluation metric. Each of these three datasets was evaluated over ten independent runs with different random seeds. For ESOL and FreeSolv, we selected the best result from each run and reported the mean and standard deviation across the ten best results. In contrast, for Tox21, since the baseline model Drug3D-Net did not report standard deviations, we followed a consistent strategy by reporting the average of the best results across the ten runs. The overall experimental design follows widely adopted practices in the literature, ensuring the rigor and credibility of the evaluation.

## Results

To comprehensively and rigorously evaluate the performance of the Prop3D model, we have designed a multi-dimensional experimental comparison framework, employing both semi-supervised learning and unsupervised learning baseline methods as reference standards.

*Baseline*

In the domain of supervised learning, we selected the following baseline methods: ATOM3D-3DCNN[28] proposes a molecular representation learning approach that employs molecular voxel data as input with a 3DCNN serving as the core architecture. Drug3D-Net[27] enhances the 3DCNN framework by incorporating a gated attention mechanism for processing molecular 3D grid data. DMPNN[54] represents an interactive message-passing methodology that accounts for intermolecular interactions. AttentiveFP[55] constitutes an attention-based graph neural network that emphasizes attention modeling of molecular features. HMGNN[56] integrates global molecular representations through attention mechanisms to further improve model performance.

For self-supervised learning benchmarks, N-Gram[57] generates molecular graph representations by constructing node embeddings through short-path analysis. MolCLR[58] adopts a 2D-2D view contrastive learning paradigm, enhancing representational capacity via atomic masking, bond deletion, and subgraph removal operations. GraphMVP[59] introduces a 2D-3D view contrastive learning framework that integrates multi-dimensional molecular information. GROVER[60] acquires deep molecular features through a motif-level prediction pretraining strategy. Both GEM[61] and Uni-Mol leverage 3D molecular information while designing predictive self-supervised learning schemes to further enhance the accuracy and generalizability of molecular representations. 3DGCL[62] represents a 3D-3D contrastive learning approach. These methodologies optimize molecular representation learning from diverse perspectives, demonstrating the diversity and innovation inherent in self-supervised learning paradigms.

*Performance evaluation*

Table 2 provides a detailed presentation of the performance evaluation results of Prop3D on the QM9 dataset, benchmarking against the 10 regression tasks reported in ATOM3D to verify the performance of Prop3D on

structured datasets. Concurrently, to validate the effectiveness of the large kernel decomposition strategy in 3D space, we removed the CBAM attention module from Prop3D, constructing a pure convolutional neural network architecture, referred to as Prop3D-CNN. Experimental results demonstrate that Prop3D-CNN significantly outperforms ATOM3D-3DCNN on 8 out of the 10 regression tasks, further substantiating the efficacy of the convolutional kernel decomposition strategy proposed in this study for enhancing molecular representation learning. Moreover, the complete Prop3D model surpasses Prop3D-CNN across all tasks, highlighting the additional contribution of the attention mechanism. The experimental data indicate that the pronounced superiority of Prop3D-CNN over ATOM3D-3DCNN fully validates the superiority of the convolutional kernel decomposition strategy. The relevant performance data are cited from ATOM3D, with the best-performing model results in each task prominently marked in bold.

Table 3 presents the performance evaluation of the Prop3D model on the ESOL and FreeSolv datasets, along with a comparative analysis against multiple baseline models. The performance metrics of the baseline models are sourced from the studies of 3DGCL and Drug3D-Net. The experimental results demonstrate that our proposed method significantly outperforms the existing baseline models on both datasets, thereby validating the efficacy and superiority of the approach introduced in this study. Figure 5 shows the performance of different models on the ESOL and Freesolv datasets.

Table 4 presents the performance of Prop3D on the 12 subtasks of the Tox21 dataset, aimed at assessing its effectiveness in classification tasks. The performance metrics of the baseline models are sourced from the Drug3D-Net study. In our experiments, Prop3D outperformed the baseline results reported by Drug3D-Net in 11 out of the 12 subtasks. The experimental findings demonstrate that the proposed Prop3D method maintains a significant advantage in handling classification tasks, thereby validating its broad applicability and superiority.

In summary, the experimental results demonstrate that Prop3D outperforms existing baseline models across multiple datasets and tasks, validating the effectiveness and advantages of the proposed method in molecular representation learning. In the regression tasks on the QM9 dataset, Prop3D significantly surpasses ATOM3D-3DCNN, proving the superiority of the convolutional kernel decomposition strategy. In the comparative experiments on the ESOL and FreeSolv datasets, Prop3D also significantly outperforms several baseline models. Furthermore, in the classification tasks on the Tox21 dataset, Prop3D excels, surpassing most of the sub-tasks reported by Drug3D-Net. Overall, Prop3D demonstrates its broad applicability and superior performance across different tasks and datasets.

## Conclusion and further works

In this study, we propose a novel molecular representation learning framework based on 3DCNN, termed Prop3D, which is designed to address the computational overhead caused by the inherent sparsity of voxelized small molecule data and the use of large convolutional kernels to expand the receptive field. Prop3D incorporates an innovative Large Kernel Convolution Decomposition Strategy, which significantly enhances the model's deep feature extraction capabilities while avoiding substantial increases in computational cost. This strategy markedly improves the model's ability to capture spatial and conformational features of complex molecular structures, thereby leading to more expressive molecular representations for downstream tasks such as molecular property prediction.

Furthermore, Prop3D achieves a favorable trade-off between model efficiency and computational resource consumption, effectively mitigating the tension between computational complexity and predictive performance. Through extensive comparative experiments across multiple benchmark datasets, Prop3D demonstrates superior performance. Compared with existing supervised learning models and self-supervised approaches (e.g., GraphCL, MolCLR), Prop3D achieves notably better results in molecular property prediction tasks. These

| Metric | MAE (lower is better) | | |
|---|---|---|---|
| TASK | ATOM3D-3DCNN | Prop3D-CNN | Prop3D |
| alpha | 3.045 ± 1.128 | 1.370 ± 0.015 | **1.273 ± 0.005** |
| cv | 1.418 ± 0.200 | 0.937 ± 0.022 | **0.880 ± 0.011** |
| mu | 0.754 ± 0.009 | 0.732 ± 0.001 | **0.631 ± 0.002** |
| r2 | 64.514 ± 1.524 | 49.180 ± 0.20 | **47.370 ± 0.011** |
| gap | **0.580 ± 0.004** | 0.684 ± 0.003 | 0.579 ± 0.012 |
| zpve | 88.219 ± 16.287 | 58.801 ± 0.920 | **42.522 ± 0.351** |
| homo | 0.303 ± 0.000 | 0.313 ± 0.015 | **0.285 ± 0.050** |
| lumo | **0.517 ± 0.011** | 0.633 ± 0.022 | 0.538 ± 0.022 |
| g298_atom | 4.369 ± 0.805 | 0.863 ± 0.017 | **0.810 ± 0.011** |
| h298_atom | 4.088 ± 0.229 | 0.923 ± 0.016 | **0.844 ± 0.052** |

**Table 2**. The performance evaluation of Prop3D on the QM9 dataset is presented. Compared to the 10 regression tasks in ATOM3D[28], the results validate the performance of Prop3D on structured datasets. Prop3D-CNN, with the CBAM module removed, significantly outperforms ATOM3D-3DCNN on 8 tasks, demonstrating the effectiveness of the convolutional kernel decomposition strategy. The complete Prop3D surpasses Prop3D-CNN across all tasks, highlighting the contribution of the attention mechanism. The best results are marked in bold. The data is sourced from ATOM3D.

| Metric | RMSE (lower is better) | |
|---|---|---|
| **Model** | **ESOL** | **Freesolv** |
| DMPNN | 1.050 (0.008) | 2.082 (0.082) |
| Attentive FP | 0.877 (0.029) | 2.073 (0.183) |
| HMGNN | 0.832 (0.010) | 1.857 (0.071) |
| N-GramRF | 1.074 (0.107) | 2.688 (0.085) |
| N-GramXGB | 1.083 (0.082) | 5.061 (0.744) |
| PretrainGNN | 1.100 (0.006) | 2.764 (0.002) |
| MolCLR | 1.271 (0.033) | 2.594 (0.249) |
| GraphMVP | 1.029 (0.033) | |
| GROVERbase | 0.983 (0.090) | 2.176 (0.052) |
| GROVERlarge | 0.895 (0.017) | 2.272 (0.051) |
| GEM | 0.798 (0.029) | 1.877 (0.094) |
| Drug3D-Net | 0.9683(0.090) | 1.4709(0.054) |
| Uni-Mol | 0.788 (0.029) | 1.620 (0.035) |
| 3DGCL | 0.778 (0.102) | 1.441 (0.19) |
| **Prop3D** | **0.762 (0.036)** | **1.2006 (0.016)** |

**Table 3**. The table presents the performance evaluation results of the Prop model on the ESOL and FreeSolv datasets, along with a comparison to multiple baseline models. The performance data of the baseline models are sourced from the studies of 3DGCL[62] and Drug3D-Net[27]. The experimental results indicate that the Prop model significantly outperforms the existing baseline models on both datasets.

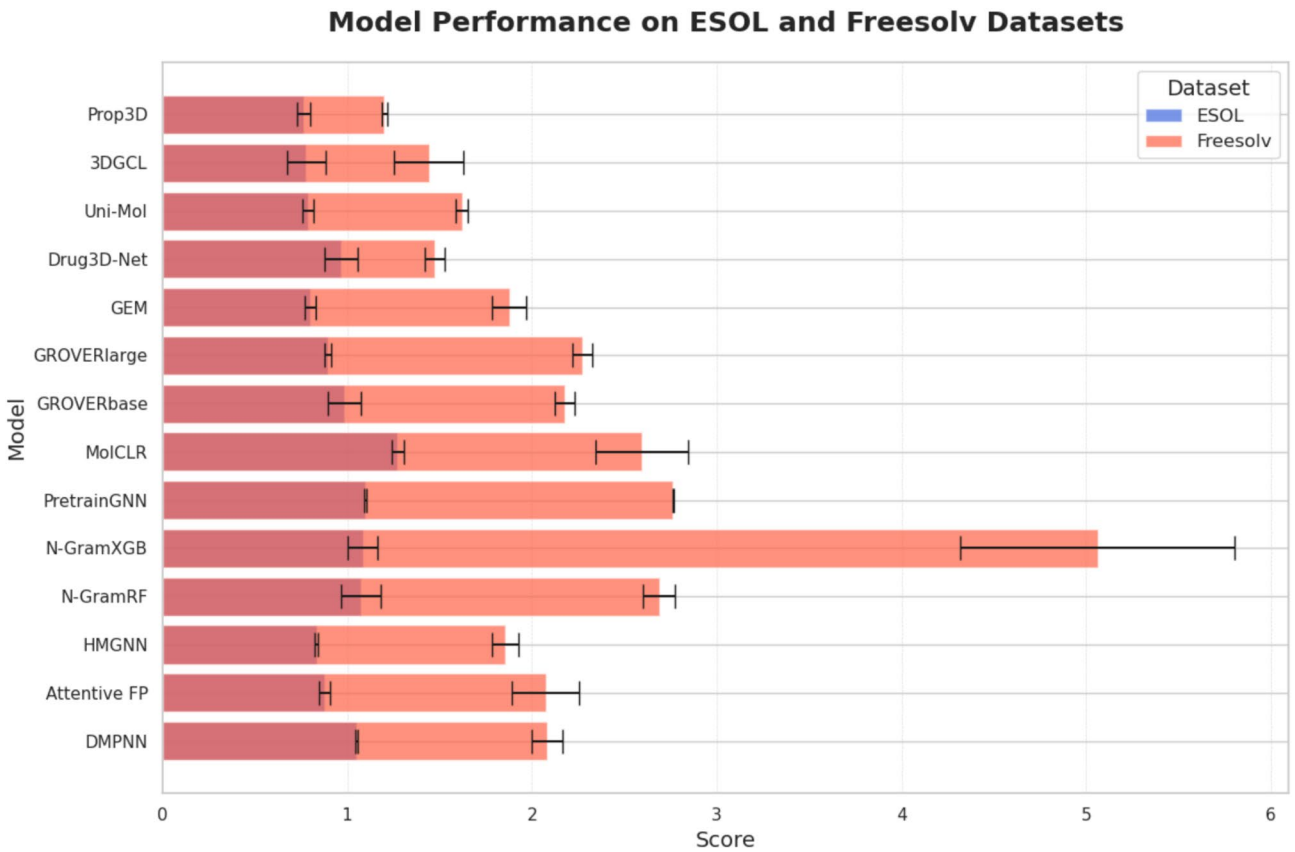


**Fig. 5**. This figure shows the performance of different models on the ESOL and Freesolv datasets. The error bars in the bar charts represent the standard deviation of the model scores, and the colors distinguish between the different datasets.

| Metric | ROC-AUC (higher is better) | | | |
|---|---|---|---|---|
| Model | Smi2Vec-LSTM | Smi2Vec-BiGRU | Drug3D-Net | Prop3D |
| NR-AR | 0.6914 | 0.7114 | 0.9708 | **0.9742** |
| NR-AR-LBD | 0.7477 | 0.8243 | 0.9689 | **0.9804** |
| NR-AhR | 0.6780 | 0.8793 | 0.9428 | **0.9470** |
| NR-Aromatase | 0.4964 | 0.6985 | 0.9647 | **0.9747** |
| NR-ER | 0.6231 | 0.7360 | 0.9234 | **0.9341** |
| NR-ER-LBD | 0.5308 | 0.8675 | **0.9619** | 0.9604 |
| NR-PPAR-gamma | 0.5659 | 0.7494 | 0.9704 | **0.9800** |
| SR-ARE | 0.6414 | 0.7611 | 0.9034 | **0.9230** |
| SR-ATAD5 | 0.5000 | 0.7632 | 0.9578 | **0.9745** |
| SR-HSE | 0.6120 | 0.7845 | 0.9601 | **0.9639** |
| SR-MMP | 0.7425 | 0.8599 | 0.9422 | **0.9536** |
| SR-p53 | 0.5180 | 0.7321 | 0.9638 | **0.9637** |

**Table 4**. The table presents the performance of Prop3D on the 12 subtasks of the Tox21 dataset, aimed at evaluating its effectiveness in classification tasks. The performance data of the baseline model is sourced from Drug3D-Net. The experimental results indicate that Prop3D outperforms the baseline results reported by Drug3D-Net in 11 out of the 12 subtasks.

experimental results strongly validate the effectiveness of the proposed large kernel convolution decomposition strategy in improving molecular representation learning and highlight its advantages in modeling complex molecular structures.

Although the proposed Prop3D framework enhances the performance of convolutional neural networks in the field of molecular representation learning to a certain extent, it still exhibits several limitations that warrant further exploration and refinement. Firstly, Prop3D is built upon three-dimensional convolutional neural networks (3D CNNs), whose dense convolution operations in 3D space incur substantial computational overhead. Compared with more lightweight architectures such as graph neural networks (GNNs), Prop3D may encounter efficiency bottlenecks when processing large-scale or high-resolution molecular datasets. Moreover, such computational intensity poses challenges for model deployment in resource-constrained environments, such as edge computing or high-throughput screening tasks. Secondly, the current model relies on a supervised learning paradigm, which requires a large amount of high-quality labeled data. However, in the vast molecular space, acquiring accurately labeled samples is often time-consuming and costly, resulting in limited training data availability. This significantly restricts the model's generalization and applicability in low-resource or unlabeled scenarios.

Future work will proceed along several directions. First, we plan to incorporate self-supervised or semi-supervised learning strategies to fully utilize large-scale unlabeled molecular data, thereby improving the model's representational ability and transferability under label-scarce conditions. Second, we aim to explore more lightweight network architectures to further reduce computational costs and enhance model deployability. Finally, we intend to extend the application of Prop3D to a broader range of structure-driven tasks, such as drug-drug interaction (DDI)[63,64] prediction and chemical-protein interaction (CPI)[65,66] prediction, in order to validate its generalization and scalability across diverse molecular learning scenarios.

We believe these directions will further advance Prop3D toward becoming a more efficient, generalizable, and practical molecular modeling framework, providing powerful computational tools and innovative paradigms for molecular science research.

## Data availability

All datasets used in this study are publicly available from the following sources. The QM9 dataset was obtained from the Atom3D project (https://www.atom3d.ai/smp.html), which provides quantum chemical properties of small organic molecules for machine learning and molecular modeling tasks. The ESOL, FreeSolv, and Tox21 datasets were obtained from the Drug3D-Net GitHub repository (https://github.com/anny0316/Drug3D-Net). These datasets include aqueous solubility data (ESOL), hydration free energies (FreeSolv), and toxicity information (Tox21), and are widely used in drug discovery and molecular property prediction research. All datasets were used in accordance with the terms and conditions specified by their respective original sources.

Received: 1 July 2025; Accepted: 25 September 2025
Published online: 31 October 2025

## References

1. Rong, Y. et al. Self-supervised graph transformer on large-scale molecular data. *Adv. Neural Inf. Process. Syst.* **33**, 12559–12571 (2020).
2. Wang, Y., Wang, J., Cao, Z. & Barati Farimani, A. Molecular contrastive learning of representations via graph neural networks. *Nat. Mach. Intell.* **4**, 279–287 (2022).
3. Goh, G. B., Hodas, N. O. & Vishnu, A. Deep learning for computational chemistry. *J. Comput. Chem.* **38**, 1291–1307 (2017).

4. Patani, G. A. & LaVoie, E. J. Bioisosterism: a rational approach in drug design. *Chem. Rev.* **96**, 3147–3176 (1996).
5. Silverman, R. B. & Holladay, M. W. *The Organic Chemistry of Drug Design and Drug Action* (Academic Press, 2014).
6. Xiong, J., Xiong, Z., Chen, K., Jiang, H. & Zheng, M. Graph neural networks for automated de novo drug design. *Drug Discov. Today* **26**, 1382–1393 (2021).
7. Shen, J. & Nicolaou, C. A. Molecular property prediction: recent trends in the era of artificial intelligence. *Drug Discov. Today: Technol.* **32**, 29–36 (2019).
8. Li, Z., Jiang, M., Wang, S. & Zhang, S. Deep learning methods for molecular representation and property prediction. *Drug Discov. Today* **27**, 103373 (2022).
9. Weininger, D. Smiles, a chemical language and information system: 1–introduction to methodology and encoding rules. *J. Chem. Inf. Comput. Sci.* **28**, 31–36 (1988).
10. Paul, A. *et al.* Chemixnet: Mixed dnn architectures for predicting chemical properties using multiple molecular representations. arXiv preprint arXiv:1811.08283 (2018).
11. Nguyen, D. D. & Wei, G.-W. Agl-score: algebraic graph learning score for protein-ligand binding scoring, ranking, docking, and screening. *J. Chem. Inf. Model.* **59**, 3291–3304 (2019).
12. Zang, Q. et al. In silico prediction of physicochemical properties of environmental chemicals using molecular fingerprints and machine learning. *J. Chem. Inf. Model.* **57**, 36–49 (2017).
13. Yang, M. et al. Machine learning models based on molecular fingerprints and an extreme gradient boosting method lead to the discovery of jak2 inhibitors. *J. Chem. Inf. Model.* **59**, 5002–5012 (2019).
14. Hu, W. *et al.* Strategies for pre-training graph neural networks. arXiv preprint arXiv:1905.12265 (2019).
15. Li, P. et al. An effective self-supervised framework for learning expressive molecular global representations to drug discovery. *Brief. Bioinform.* **22**, bbab109 (2021).
16. Ying, C. et al. Do transformers really perform badly for graph representation?. *Adv. Neural Inf. Process. Syst.* **34**, 28877–28888 (2021).
17. Sun, M. et al. Graph convolutional networks for computational drug development and discovery. *Brief. Bioinform.* **21**, 919–935 (2020).
18. Lin, X., Quan, Z., Wang, Z.-J., Huang, H. & Zeng, X. A novel molecular representation with bigru neural networks for learning atom. *Brief. Bioinform.* **21**, 2099–2111 (2020).
19. Wang, S., Guo, Y., Wang, Y., Sun, H. & Huang, J. Smiles-bert: large scale unsupervised pre-training for molecular property prediction. In *Proceedings of the 10th ACM international conference on bioinformatics, computational biology and health informatics*, 429–436 (2019).
20. Duvenaud, D. K. *et al.* Convolutional networks on graphs for learning molecular fingerprints. *Advances in neural information processing systems* **28** (2015).
21. You, Y. et al. Graph contrastive learning with augmentations. *Adv. Neural Inf. Process. Syst.* **33**, 5812–5823 (2020).
22. Norouzi, R., Norouzi, R., Abbasi, K., Norouzi, R. & Razzaghi, P. Dft_anpd: A dual-feature two-sided attention network for anticancer natural products detection. *Comput. Biol. Med.* **194**, 110442 (2025).
23. Casey, A. D., Son, S. F., Bilionis, I. & Barnes, B. C. Prediction of energetic material properties from electronic structure using 3d convolutional neural networks. *J. Chem. Inf. Model.* **60**, 4457–4473 (2020).
24. Tran, N., Kepple, D., Shuvaev, S. & Koulakov, A. Deepnose: Using artificial neural networks to represent the space of odorants. In *International Conference on Machine Learning*, 6305–6314 (PMLR, 2019).
25. Liu, S. et al. Multiresolution 3d-densenet for chemical shift prediction in nmr crystallography. *J. Phys. Chem. Lett.* **10**, 4558–4565 (2019).
26. Kuzminykh, D. et al. 3d molecular representations based on the wave transform for convolutional neural networks. *Mol. Pharm.* **15**, 4378–4385 (2018).
27. Li, C., Wang, J., Niu, Z., Yao, J. & Zeng, X. A spatial-temporal gated attention module for molecular property prediction based on molecular geometry. *Brief. Bioinform.* **22**, bbab078 (2021).
28. Townshend, R. J. *et al.* Atom3d: Tasks on molecules in three dimensions. arXiv preprint arXiv:2012.04035 (2020).
29. Chua, L. O. Cnn: A vision of complexity. *Int. J. Bifurc. Chaos* **7**, 2219–2425 (1997).
30. Simonyan, K. & Zisserman, A. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014).
31. Xie, S., Girshick, R., Dollár, P., Tu, Z. & He, K. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1492–1500 (2017).
32. Liu, X., Feng, H., Wu, J. & Xia, K. Persistent spectral hypergraph based machine learning (psh-ml) for protein-ligand binding affinity prediction. *Brief. Bioinform.* **22**, bbab127 (2021).
33. Yu, W., Zhou, P., Yan, S. & Wang, X. Inceptionnext: When inception meets convnext. In *Proceedings of the IEEE/cvf conference on computer vision and pattern recognition*, 5672–5683 (2024).
34. Woo, S., Park, J., Lee, J.-Y. & Kweon, I. S. Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)*, 3–19 (2018).
35. Krizhevsky, A., Sutskever, I. & Hinton, G. E. Imagenet classification with deep convolutional neural networks. *Commun. ACM* **60**, 84–90 (2017).
36. Szegedy, C. *et al.* Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1–9 (2015).
37. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. & Wojna, Z. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2818–2826 (2016).
38. Ding, X., Zhang, X., Han, J. & Ding, G. Scaling up your kernels to 31x31: Revisiting large kernel design in cnns. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 11963–11975 (2022).
39. Ding, X. *et al.* Repvgg: Making vgg-style convnets great again. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 13733–13742 (2021).
40. Guo, M.-H., Lu, C.-Z., Liu, Z.-N., Cheng, M.-M. & Hu, S.-M. Visual attention network. *Comput. Vis. Media* **9**, 733–752 (2023).
41. Liu, S. *et al.* More convnets in the 2020s: Scaling up kernels beyond 51x51 using sparsity. arXiv preprint arXiv:2207.03620 (2022).
42. Cambria, E. & White, B. Jumping nlp curves: A review of natural language processing research. *IEEE Comput. Intell. Mag.* **9**, 48–57 (2014).
43. Voulodimos, A., Doulamis, N., Doulamis, A. & Protopapadakis, E. Deep learning for computer vision: A brief review. *Comput. Intell. Neurosci.* **2018**, 7068349 (2018).
44. Bahdanau, D., Cho, K. & Bengio, Y. Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 (2014).
45. Vaswani, A. *et al.* Attention is all you need. *Advances in neural information processing systems* **30** (2017).
46. Veličković, P. *et al.* Graph attention networks. arXiv preprint arXiv:1710.10903 (2017).
47. Ruddigkeit, L., Deursen, R., Blum, L. C. & Reymond, J.-L. Enumeration of 166 billion organic small molecules in the chemical universe database gdb-17. *J. Chem. Inf. Model.* **52**, 2864–2875 (2012).
48. Wu, Z. et al. Moleculenet: a benchmark for molecular machine learning. *Chem. Sci.* **9**, 513–530 (2018).
49. Delaney, J. S. Esol: estimating aqueous solubility directly from molecular structure. *J. Chem. Inf. Comput. Sci.* **44**, 1000–1005 (2004).

50. Mobley, D. L. & Guthrie, J. P. Freesolv: a database of experimental and calculated hydration free energies, with input files. *J. Comput.-Aided Mol. Des.* **28**, 711–720 (2014).
51. Huang, R. et al. Tox21challenge to build predictive models of nuclear receptor and stress response pathways as mediated by exposure to environmental chemicals and drugs. *Front. Environ. Sci.* **3**, 85 (2016).
52. Landrum, G. Rdkit documentation. *Release* **1**, 4 (2013).
53. Imambi, S., Prakash, K. B. & Kanagachidambaresan, G. Pytorch. *Programming with TensorFlow: solution for edge computing applications*, 87–104 (2021).
54. Yang, K. et al. Analyzing learned molecular representations for property prediction. *J. Chem. Inf. Model.* **59**, 3370–3388 (2019).
55. Xiong, Z. et al. Pushing the boundaries of molecular representation for drug discovery with the graph attention mechanism. *J. Med. Chem.* **63**, 8749–8760 (2019).
56. Shui, Z. & Karypis, G. Heterogeneous molecular graph neural networks for predicting molecule properties. In *2020 IEEE International Conference on Data Mining (ICDM)*, 492–500 (IEEE, 2020).
57. Liu, S., Demirel, M. F. & Liang, Y. N-gram graph: Simple unsupervised representation for graphs, with applications to molecules. *Advances in neural information processing systems* **32** (2019).
58. Wang, Y., Wang, J., Cao, Z. & Barati Farimani, A. Molecular contrastive learning of representations via graph neural networks. *Nat. Mach. Intell.* **4**, 279–287 (2022).
59. Liu, S. *et al.* Pre-training molecular graph representation with 3d geometry. arXiv preprint arXiv:2110.07728 (2021).
60. Rong, Y. et al. Self-supervised graph transformer on large-scale molecular data. *Adv. Neural Inf. Process. Syst.* **33**, 12559–12571 (2020).
61. Fang, X. et al. Geometry-enhanced molecular representation learning for property prediction. *Nat. Mach. Intell.* **4**, 127–134 (2022).
62. Moon, K., Im, H.-J. & Kwon, S. 3d graph contrastive learning for molecular property prediction. *Bioinformatics* **39**, btad371 (2023).
63. Lin, X., Quan, Z., Wang, Z.-J., Ma, T. & Zeng, X. Kgnn: Knowledge graph neural network for drug-drug interaction prediction. *IJCAI* **380**, 2739–2745 (2020).
64. Feng, Y.-H., Zhang, S.-W. & Shi, J.-Y. Dpddi: a deep predictor for drug-drug interactions. *BMC Bioinform.* **21**, 419 (2020).
65. Cheng, F. et al. Prediction of chemical-protein interactions: multitarget-qsar versus computational chemogenomic methods. *Mol.r BioSyst.* **8**, 2373–2384 (2012).
66. Wang, J. & Ohsawa, Y. Interacting evolution of modeling and data exchange: A case for predicting chemical-protein interactive visualization. In *2024 IEEE International Conference on Big Data (BigData)*, 6877–6883 (IEEE, 2024).


## Author contributions

H.Z. implemented the core algorithm of the proposed model and wrote the initial draft of the manuscript. C.L. provided funding, data resources, equipment support, and revised the manuscript. G.Z., M.T., and J.L. contributed to funding acquisition, algorithm validation, and manuscript revision. H.Zh. assisted in algorithm verification. All authors participated in research discussions, provided critical feedback, and reviewed and approved the final version of the manuscript.

## Funding

This work was supported by the National Natural Science Foundation of China (Grant No. 62262072, 62472370 and 62362066), the Yunnan Provincial Philosophy and Social Science Planning Social Think Tank Project (Grant No.SHZK2024204), the Key Project of Basic Research in Yunnan Province (202501AS070007), the Major Science and Technology Special Plan Project of Yunnan Province (202302AE090022-1), the Basic Research Special Project of Yunnan Province Science and Technology Department (202401AU070051).

## Declarations

## Competing interests

The authors declare that they have no competing financial interests or personal relationships that could have influenced the work reported in this paper.

## Additional information

**Correspondence** and requests for materials should be addressed to C.L.