

# Macromolecules Structural Classification With a 3D Dilated Dense Network in Cryo-Electron Tomography

Shan Gao , Renmin Han, Xiangrui Zeng, Zhiyong Liu, Min Xu , and Fa Zhang 

**Abstract**—Cryo-electron tomography, combined with subtomogram averaging (STA), can reveal three-dimensional (3D) macromolecule structures in the near-native state from cells and other biological samples. In STA, to get a high-resolution 3D view of macromolecule structures, diverse macromolecules captured by the cellular tomograms need to be accurately classified. However, due to the poor signal-to-noise-ratio (SNR) and severe ray artifacts in the tomogram, it remains a major challenge to classify macromolecules with high accuracy. In this paper, we propose a new convolutional neural network, named 3D-Dilated-DenseNet, to improve the performance of macromolecule classification. In 3D-Dilated-DenseNet, there are two key strategies to guarantee macromolecule classification accuracy: 1) Using dense connections to enhance feature map utilization (corresponding to the baseline 3D-C-DenseNet); 2) Adopting dilated convolution to enrich multi-level information in feature maps. We tested 3D-Dilated-DenseNet and 3D-C-DenseNet both on synthetic data and experimental data. The results show that, on synthetic data, compared with the state-of-the-art method in the SHREC contest (SHREC-CNN), both 3D-C-DenseNet and 3D-Dilated-DenseNet outperform SHREC-CNN. In particular, 3D-Dilated-DenseNet improves 0.393 of F1 metric on tiny-size macromolecules and 0.213 on small-size macromolecules. On experimental data, compared with 3D-C-DenseNet, 3D-Dilated-DenseNet can increase classification performance by 2.1 percent.

**Index Terms**—Cryo-electron tomography, image classification, convolution neural network

## 1 INTRODUCTION

THE cellular process is performed by the cooperative interaction of macromolecules [1], [2]. To get a better insight into the cellular process, native structures and spatial organizations of these macromolecules require to be understood. Some macromolecular complexes are composed of molecules with strong interactions, which leading to macromolecular complexes owns a stable structure. For these macromolecular complexes, traditional structural biology methods such as x-ray are amenable to reveal their structure [3]. However, many macromolecular complexes are not stable enough, and their ultrastructure in the native state remains to be solved. To address this issue, Cryo-Electron Tomography (cryo-ET), with the ability to reveal the structure of macromolecular complexes in a near-native

state at the sub-molecular resolution, is proposed [4]. Specifically, cryo-ET has been widely applied and discovered many important structures such as SARS-COV-2 [5], *S. pombe* mitotic chromosomes [6], and primary cilia [7].

In cryo-ET, the frozen-hydrated sample is placed on the transmission electron microscopy stage and progressively tilted to different angles from  $-60^\circ$  to  $+60^\circ$  with  $1^\circ$ ,  $2^\circ$ , or  $3^\circ$  increments (Fig. 1 A). Meanwhile, two-dimensional (2D) projection images (tilt series) are collected by a camera at each tilt angle. These tilted images are then aligned to reconstruct a three-dimensional (3D) cellular tomogram at a sub-molecular resolution [8] (Fig. 1 B). The reconstructed tomogram can contain tens of thousands of macromolecules in the crowded cellular environment. As high-energy electrons can damage biological samples, the total electron dose is limited, which leads to a low signal-to-noise ratio (SNR) of the tomogram. And, because of the absence of tilt angles from  $60^\circ$  to  $90^\circ$  and  $-60^\circ$  to  $-90^\circ$ , the reconstructed tomogram always has severe ray artifacts (missing wedge). To further obtain a 3D view with higher SNR and resolution, multiple copies of the identical macromolecule (subtomogram) need to be detected, classified, aligned, and averaged [9], [10], [11], [12], [13] (Figs. 1 C and 1 D), which is called subtomogram averaging (STA) [14]. However, due to the low SNR and severe ray artifacts caused by the missing wedge effect, it remains a significant challenge to classify macromolecules.

For macromolecules with known structures, the template matching method can help recognize macromolecules by calculating the cross-correlation of the template with the subtomogram [15]. However, because the template

- Shan Gao is with High Performance Computer Research Center, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China, and also with the University of Chinese Academy of Sciences, Beijing 101408, China. E-mail: gaoshan@ict.ac.cn.
- Renmin Han is with Research Center for Mathematics and Interdisciplinary Sciences, Shandong University, Qingdao 266237, China. E-mail: hanrenmin@gmail.com.
- Xiangrui Zeng and Min Xu are with Computational Biology Department, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213 USA. E-mail: {xiangrui, mxu1}@cs.cmu.edu.
- Zhiyong Liu and Fa Zhang are with High Performance Computer Research Center, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China. E-mail: {liuzhiyong, zhangfa}@ict.ac.cn.

Manuscript received 30 Nov. 2020; revised 26 Feb. 2021; accepted 9 Mar. 2021. Date of publication 17 Mar. 2021; date of current version 3 Feb. 2022.

(Corresponding author: Fa Zhang.)

Digital Object Identifier no. 10.1109/TCBB.2021.3065986

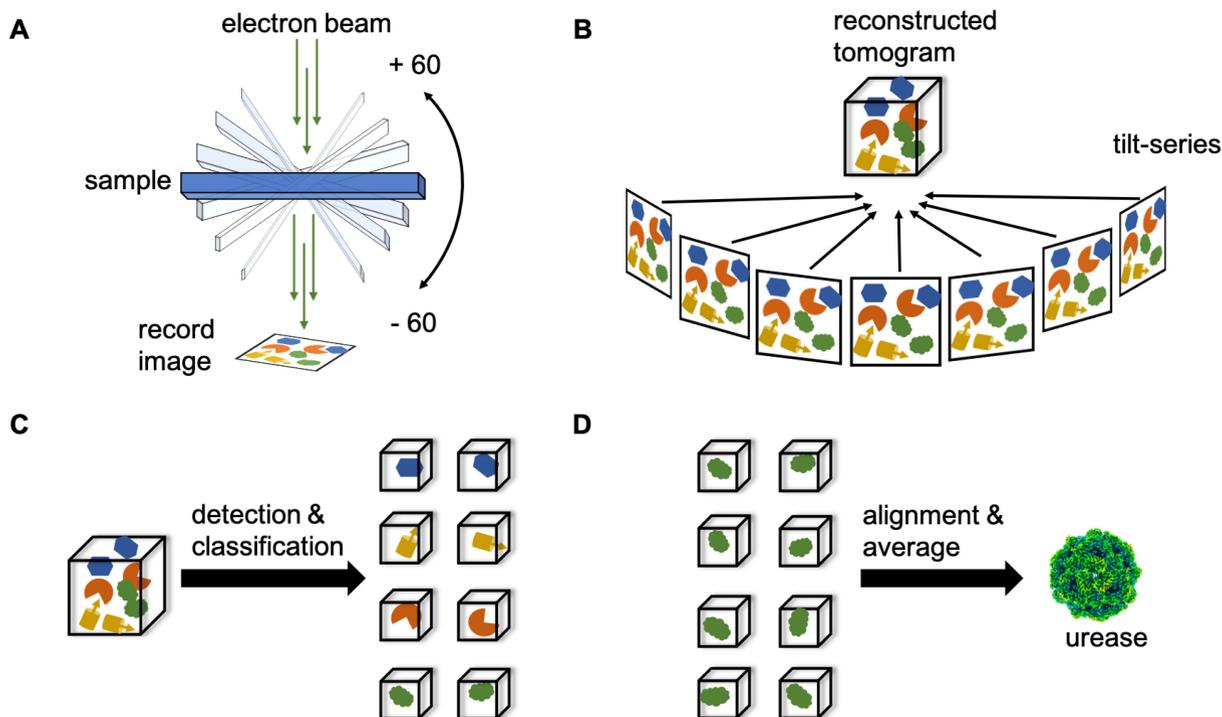


Fig. 1. The workflow of STA. (A) The collection of tilt-series images. (B) The reconstruction of the tomogram. (C) Macromolecule detection and classification in STA. (D) Macromolecule alignment and average in STA. The green high resolution structure represents urease from the pathogen *Yersinia* [18].

matching-based method severely depends on templates, this method has poor robustness. When the targets contain additional bound components or conformational changes, templates will misfit targets. Besides, targets and templates from different organisms can also affect matching accuracy. Thus, to overcome the above limitations, template-free classification methods are proposed [16], [17]. The common template-free framework recognizes macromolecules by iteratively clustering macromolecules to similar groups. However, clustering large quantities of subtomogram is very computationally intensive and time-consuming. Therefore, the clustering-based methods are only suitable for a small amount of data with few categories.

Recently, several machine learning methods have been proposed for macromolecule classification [19], [20], [21], [22]. One classical method is the support vector machine (SVM) combined with template matching [19]. The method uses two types of manually designed features to train the SVM. One is the correlation coefficient features computed from templates, and the other is the invariant rotation features. However, these manually designed features limit model generalization. With the blowout of cryo-ET data amount and computer performance breakthrough growth, supervised deep learning methods based on the convolutional neural network (CNN) model become popular [19], [20]. The CNN-based methods recognize macromolecules in two steps: feature extraction and feature classification. The CNN-based method first encodes the input subtomogram into high-dimensional feature maps. Based on these feature maps, the feature classifier outputs the category of input subtomogram. Thus, for the CNN-based method, discriminative features are the key to macromolecule classification. However, due to low SNR and severe ray artifacts, it is a

significant challenge to obtain accurate visual features of macromolecules. Although previous studies have proved that the CNN model with deep convolution layers can extract useful features, this kind of deep model is computing resource waste when subtomograms can be classified with shallow layers.

In this article, we proposed a novel convolution neural network, named 3D-Dilated-DenseNet, to improve classification performance. The main idea of 3D-Dilated-DenseNet is to enrich multi-level information of feature maps and increase feature map utilization. To enrich multi-level information of feature maps, we adopted the dilated convolutional layer [23]. By expanding kernels with zeros, the dilated convolution layer can enlarge the receptive field and convolve feature maps with gaps. To increase feature map utilization, we used dense shortcuts between all convolutional layers [24]. Each convolutional layer can use multi-level feature maps of all preceding convolutional layers as input. However, the increased amount of feature maps in dense connection can lead to more model parameters and extend training time. To address this issue, we designed a bottleneck component to reduce the model size and improve model training efficiency.

To prove the performance of our method, we tested 3D-Dilated-DenseNet on both synthetic data and experimental data. The synthetic data is published in the SHREC contest [25], which contains twelve categories of macromolecules. And SHREC classified these twelve category macromolecules into four groups: large, medium, small, and tiny according to molecular weight. The experimental data contains seven categories of macromolecules that are manually extracted from EMPIAR [26]. On synthetic data, the results show that, compared with the state-of-the-art CNN

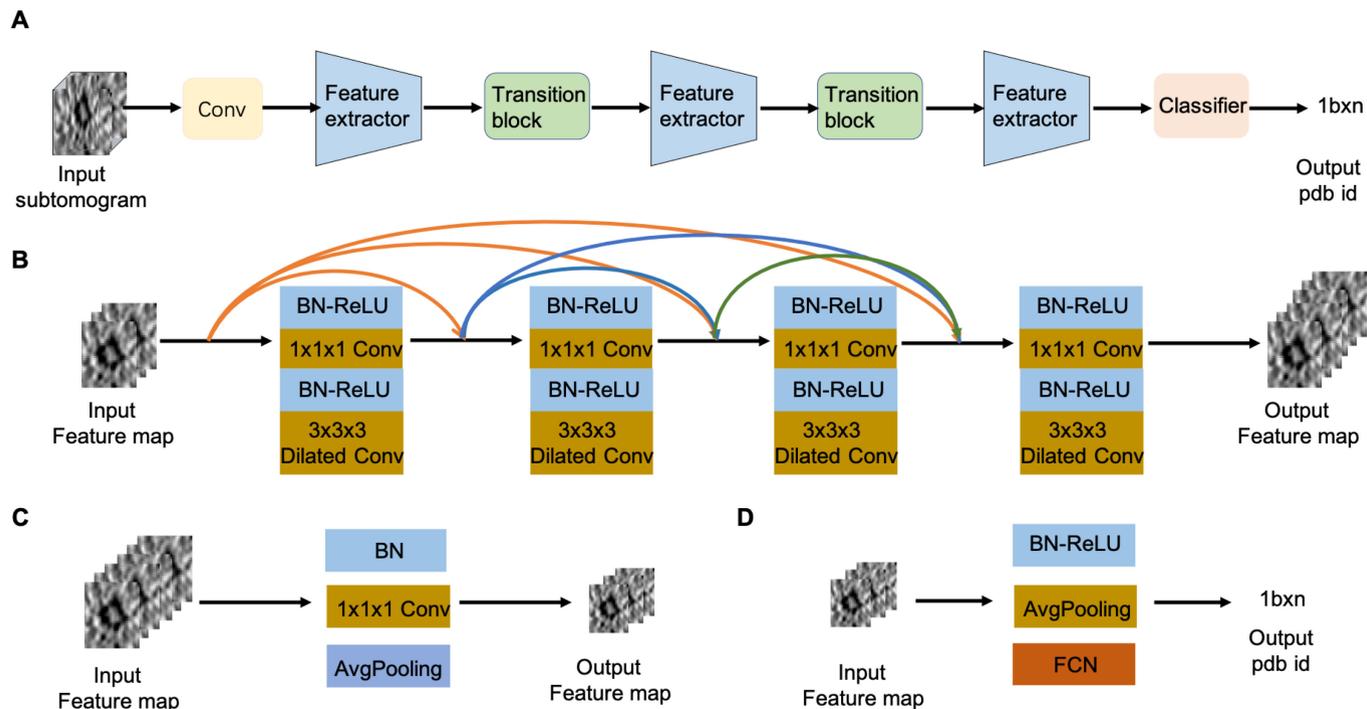


Fig. 2. The architecture of 3D-Dilated-DenseNet. (A) The model framework of 3D-Dilated-DenseNet. (B) The detailed information of the feature extractor module. (C) The detailed information of the transition block. (D) The detailed information of the classifier. Here, BN means batch normalization.

method in the SHREC contest (SHREC-CNN) [25], the 3D-Dilated-DenseNet improves 0.393 of F1 metric on tiny-size macromolecule and 0.213 on small-size macromolecules. To explore the contribution of the key components: dense connection, bottleneck component, and dilated convolutional layer, we provided an ablation study. Based on the simple 3D-CNN (3D-base-CNN), we gradually add each key component and obtained three more CNN models: 3D-DenseNet, 3D-C-DenseNet, and 3D-Dilated-DenseNet. The obtained results show that compared with 3D-base-CNN, the dense connection can increase classification performance by 5.30 percent. Compared with 3D-C-DenseNet, the dilated convolution can overall increase classification performance by 3.95 percent. Because 3D-C-DenseNet is well outperforming SHREC-CNN, we make 3D-C-DenseNet as a baseline model and tested both 3D-C-DenseNet and 3D-Dilated-DenseNet on experimental data. The results show that compared 3D-C-DenseNet, 3D-Dilated-DenseNet increases classification accuracy by an average of 2.1 percent. Overall, both dense connection and dilated convolutional layer can increase classification performance, which gives us guidance for designing a macromolecule classification CNN model.

## 2 METHODS

### 2.1 3D-Dilated-DenseNet Framework

In this section, we describe our macromolecule classification CNN model, 3D-Dilated-DenseNet, in detail. We first defined the input and output of the 3D-Dilated-DenseNet. The input is a 3D subtomogram data, represented as  $\mathbb{S}^{n \times n \times n}$ . One subtomogram contains only one complete macromolecule. The output is the category of the macromolecule that is a four digits ID number recorded in the protein data bank

(PDB) (noted as PDB ID) [27]. Next, we introduce our model framework. As shown in Fig. 2 A, our model contains three main modules: feature extractor, feature transition block, and feature classifier. For a given input subtomogram, after the first convolutional layer and a series of feature extractors and transition blocks, the input subtomogram can be encoded into highly-dimensional feature maps. Then, based on these feature maps, the classifier can output the category of the macromolecule. The brief introduction of each module is at the following.

The feature extractor (Fig. 2 B) (Section 2.2) can learn multi-level feature maps of the given subtomogram. As shown in Fig. 2 B, the feature extractor includes a cascade of composite layers with dense connections. The composite layer includes a  $1 \times 1 \times 1$  convolutional layer followed by a  $3 \times 3 \times 3$  dilated convolutional layer. Before each convolutional layer, there is a batch normalization (BN)-ReLU. As the dense connection and dilated convolution are two critical components in the feature extractor, we also name the feature extractor *dilated-dense block*. The detailed information of dense connection and dilated convolutional layer are shown separately in Sections 2.2.1 and 2.2.2.

The transition block can compress feature maps for further forward propagating. Due to dense connection in dilated-dense block, all feature maps in the block remains the same size. The input of the transition block is the concatenation of the previous dilated-dense block input and the output of each composite layer in the same dilated-dense block. However, such large-size input feature maps can cause a great increase of model parameter size. To address this issue, the transition block is designed with a bottleneck component: BN- $1 \times 1 \times 1$  convolutional layer (Section 2.3) followed by an average pooling layer.

The classifier classifies the extracted feature maps in two steps: 1) mapping the extracted high-dimensional feature maps into a 1D category vector; 2) taking the item with the highest probability in the category vector as output. When mapping high-dimensional features into a 1D vector, to increase the non-linearity of the classifier, the traditional classifiers usually include two fully connected (FC) layers. However, due to FC layers cover the most parameters of the model, the traditional classifier, with two FC layers can result in model size increment and making model easily overfit. To reduce the number of FC layers, we use Global Average Pool (GAP) (Section 2.4) module to compress high-dimensional feature maps before the FC layer. Therefore, our classifier is BN-ReLU-GAP, followed by the FC layer.

## 2.2 Dilated-Dense Block

### 2.2.1 Dense Connection in Dilated-Dense Block

In the macromolecule classification CNN model, stacking deep convolutional layers is a simple and straightforward way to obtain advanced feature maps. However, this kind of method has two disadvantages. First, for some macromolecules with large size and noticeable geometry features, a shallow CNN model is enough for classification. Thus, it is a waste of computing resources to adopt a deep CNN model. Second, deep CNN models are generally accompanied by the gradient disappearance problem, which is not good for model training. Dense connections, with the ability to enhance feature utilization, can help reduce the model depth and avoid gradient vanishing problems. Therefore, we adopted dense connections in the feature extractor. As shown in Fig. 2 B, each composite layer of the dilated-dense block has dense shortcuts with other composite layers, which means each layer receives the feature maps of all preceding layers as inputs and propagates its own feature-maps into all subsequent layers.

Here, we define the computation of dense connection in the dilated-dense. We denote the input of the  $k$ th dilated-dense block as  $x_k^0$  ( $k = 1, 2, 3$ ), the  $i$ th composite layer and its corresponding output of the  $k$ th dilated-dense block as  $F_k^i(\cdot)$  and  $x_k^i$  ( $i = 1, \dots, 4$ ). With dense shortcuts connecting each composite layer, the composite layer  $F_k^i$  receives the feature maps from the input of the  $k$ th dilated-dense block and the output of its all preceding composite layers ( $x_k^1, \dots, x_k^{i-1}$ ). Thus, the output of  $F_k^i$  is computed by Eq. (1)

$$x_k^i = F_k^i(x_k^0, x_k^1, \dots, x_k^{i-1}). \quad (1)$$

### 2.2.2 Dilated Convolution in Dilated-Dense Block

In the macromolecule classification task, representative feature maps are essential to guarantee classification performance. Therefore, we adopted dilated convolutional layers with different dilation rates [23] to extract multi-level features, which can enhance model feature representation and improve model robustness. Figs. 3 A, 3 B, and 3 C show a 2D dilated convolution example with different dilation rates (noted as  $r$ ). In Fig. 3 A, when  $r = 1$ , the dilated convolution is standard convolution. In Figs. 3 B and 3 C, dilated convolution is to insert  $r - 1$  zeros in every two adjacent weights of the standard convolution. The area covered by the dilated

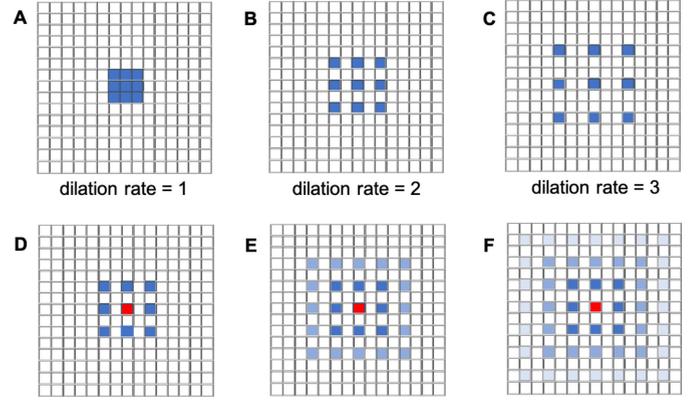


Fig. 3. A 2D example of dilated convolution layer with  $3 \times 3$  kernel. A-C are the dilated convolution layers with dilation rates: 1, 2, 3. D-F illustrate the reason for gridding artifacts.

convolutional kernel is named as receptive field. For the stacked dilated convolutional layers, the receptive field of the  $l$ th dilated convolutional layer can be computed by Eqs. (2) and (3). Here,  $R$  denoted the size of the receptive field,  $S_i$  is the stride of the  $i$ th dilated convolution and  $w$  is the kernel size. In our dilated-dense block, each composite function contains  $3 \times 3 \times 3$  dilated convolutional layer with stride 1. If the  $r$  of all  $3 \times 3 \times 3$  dilated convolutional layer is 2, the receptive field of the 3th and the 4th dilated convolutional layer is  $13 \times 13 \times 13$  and  $17 \times 17 \times 17$ . As the smallest macromolecule in our data just covers 8 pixels, such a large receptive field is ineffective to fine-grated structures. Thus, it is best for the receptive field of the stacked dilated convolutional layer not to exceed the size of the smallest macromolecule

$$R \times R = (2^{r+1} - 1) \times (2^{r+1} - 1) \quad (2)$$

$$R_l = R_{l-1} + \prod_{i=1}^{l-1} S_i * (w - 1). \quad (3)$$

Besides having an appropriate receptive field, the stacked dilated convolutional layers need to avoid gridding artifacts [28]. Here, we illustrate the reason for gridding artifacts. Figs. 3 D, 3 E, and 3 F represent three feature maps that are gradually convolved by three stacked dilated convolutional layers. All dilated convolutional kernels are  $3 \times 3$  with  $r = 2$ . First, we look into the first feature map (noted as  $F1$ ) (Fig. 3 D) that is convolved by only one dilated convolutional layer. In  $F1$ , the central pixel (marked as red) is computed from the nearby  $5 \times 5$  region (marked as blue) of the input image. With the  $F1$  further forward propagating, the followed dilated convolutional layers outputs feature map  $F2$  and feature map  $F3$  (Figs. 3 E and 3 F). In  $F2$  and  $F3$ , we can see that the computation of red pixel in each feature map is always at a checkboard pattern (darker blue pixels means more contribution while lighter blue pixels means less contribution). Thus, in the final output feature map, each pixel is computed from entirely separate sets of the input unit, which can lead to gridding artifacts. To address this issue, we designed the dilation rates of the dilated-dense block based on the hybrid dilated convolution rule (HDC) [28]. First, the common divisor of the dilation rates

should only be 1. Second, the dilation rates should be designed as a zigzag structure such as [1, 2, 3, 1, 2, 3]. After testing four dilated-dense blocks with different dilation rates (Section ), the final dilation rates of our dilated-dense block is [1, 2, 1, 1].

### 2.3 Bottleneck Component

In 3D-Dilated-DenseNet, both dilated-dense block and transition block adopt the bottleneck component. The key module of the bottleneck component is the  $1 \times 1 \times 1$  convolution, which can compress feature maps by channel-wise pooling and further improve model training efficiency. Here, we show quantitative analysis of adopting the  $1 \times 1 \times 1$  convolutional layer in dilated-dense block and transition block separately. First, we analyze dilated-dense block without bottleneck component. Let each composite layer outputs  $m$  feature maps, so the input feature map of the  $F_k^i$  is  $m_0 + m \times (i - 1)$  (here,  $m_0$  is the input feature map amount of the  $k$ th dilated-dense block), and the number of the  $3 \times 3 \times 3$  convolutional kernel parameters of the  $i$ th composite layer is  $3 \times 3 \times 3 \times (m_0 + m \times (i - 1)) \times$ . Here, we define  $m$  as 12,  $m_0$  as 24. Thus, the input feature map channel of  $F_k^4$  is  $24 + (4 - 1) \times 12 = 60$ , and the parameter number of the convolution layer is  $3 \times 3 \times 3 \times 60 \times 12 = 19440$ . When we adopted the bottleneck component in the composite layer and set the channel-wise pooling value as 0.5, for the same  $3 \times 3 \times 3$  convolutional layer, the input feature map channel and convolutional layer parameters can be reduced to 30 and  $1 \times 1 \times 1 \times 60 \times 30 + 3 \times 3 \times 3 \times 30 \times 12 = 11520$ . That means the parameter number of  $3 \times 3 \times 3$  convolutional layer is reduced by 22.2 percent.

Similarly, for transition block, we denote  $j$ th transition block as  $t_j$ , and  $t_j^0$  as the input of  $t_j$ . Because the input of the transition block includes the input of the previous dilated-dense block and the output of all composite layers of the same block. Thus,  $t_j^0 = x_i^0 + \dots + x_i^4$  ( $i = j$ ). For transition block without bottleneck component, the input feature map channel is  $24 + (5 - 1) \times 12 = 72$ . After adopted the bottleneck component, the channel of the input feature map can be reduced to half 36, though it adds convolution parameters. The number of additional parameters is  $t_j^0 \times (t_j^0/2) \times 1 \times 1 \times 1$ .

### 2.4 Global Average Pooling

In the classifier of our 3D-Dilated-DenseNet, we apply a global average pooling (GAP) followed by one FC layer to substitute the traditional classifier, which only contains FC layers. For the trained classifier, the input is  $K$  high-dimensional feature maps. After GAP, each  $k$ th input feature map ( $f_k(x, y, z)$ ) is compressed into a point by Eq. (4). Then, the FC layer, with learned weight matrix (a 2D array noted as  $w$ ), maps the output of GAP to a category vector, noted as  $c$ . Each item in the category vector shows the probability that the input subtomogram belongs to the corresponding category, and the item with the largest probability value is considered as the output. For our classifier, suppose there are 96 input feature maps with size  $4 \times 4 \times 4$ , and the output is a  $1 \times 12$  category vector. Compared with traditional classifier with only one FC layer, the parameters are reduced from  $96 \times 4 \times 4 \times 4 \times 1 \times 12 = 73728$  to  $96 \times 1 \times 12 = 1152$ .

Besides reducing model parameters, GAP can help visualize class activation mapping (CAM) [29]. Because the CAM clearly shows discriminative regions that have a contribution to category prediction, the visualization of CAM enhances the interpretability of CNN methods. Here, we illustrate the computation of CAM. The CAM is a weighted summation of all final extracted feature maps (Eq. (5)). The weight matrix, noted as  $w^c$ , is a 1D weight vector of the FC layer, which shows the contribution of each input feature map to the classification

$$F_k = \sum_{x,y,z} f_k(x, y, z) \quad (4)$$

$$\text{CAM}_c(x, y, z) = \sum_k w_k^c f_k(x, y, z). \quad (5)$$

Due to  $\text{CAM}_c$  is the weighted summation of the high-dimensional feature maps, the size of  $\text{CAM}_c(x, y, z)$  is smaller than the input subtomogram. However, when these two images are of different sizes, it is inexact to analyze the effective regions of the extracted feature maps with reference to the input subtomogram. Thus, for discussion conveniently,  $\text{CAM}_c(x, y, z)$  is upsampled to the size of the input image with bicubic interpolation.

## 3 EXPERIMENTS AND RESULTS

### 3.1 Data Preparation

Here, we introduce the synthetic data and experimental data in our work. The synthetic data is the public subtomogram data set released by SHREC[25], [30] (noted as SHREC-data). The SHREC-data contains ten reconstructed 3D tomograms and their corresponding ground truth table. Each tomogram, with the size of  $512 \times 512 \times 512$  (1 voxel equals 1 nanometer), contains macromolecules which are uniformly distributed at random Euler angles (Fig. 4 A). The ground truth table records the detailed information of each macromolecule: three-dimensional coordinates, Euler angle, and category. According to the ground truth table, we extracted 20785 subtomograms from the tomogram, covering 12 categories. Each subtomogram, with the size of  $32 \times 32 \times 32$ , contains only one complete macromolecule (Fig. 4 A). Each category contains  $\sim 1700$  macromolecules and is further classified into large, medium, small, and tiny macromolecule according to their molecular weight by SHREC (Table 1). The density map of each macromolecule is shown in Fig. 5

As shown in Fig. 4 A, the SNR of SHREC-data is extremely poor. For the convenience of observing the details in tomograms, we generated noise-free tomograms. First, we downloaded the structure information of 12 categories of macromolecules from PDB bank [27]. Then, with the reference of the tomogram resolution in SHREC-data, we generated density maps of 12 categories of macromolecules by IMOD [31] and an empty  $512 \times 512 \times 512$  volume. Finally, according to each macromolecule information in the ground truth table: location and Euler angle, we put the generated density map into the empty volume. The method of generating noise-free subtomograms is the same.

The experimental data consists of seven tomograms, which are reconstructed from 2D tilt series. These tilt series

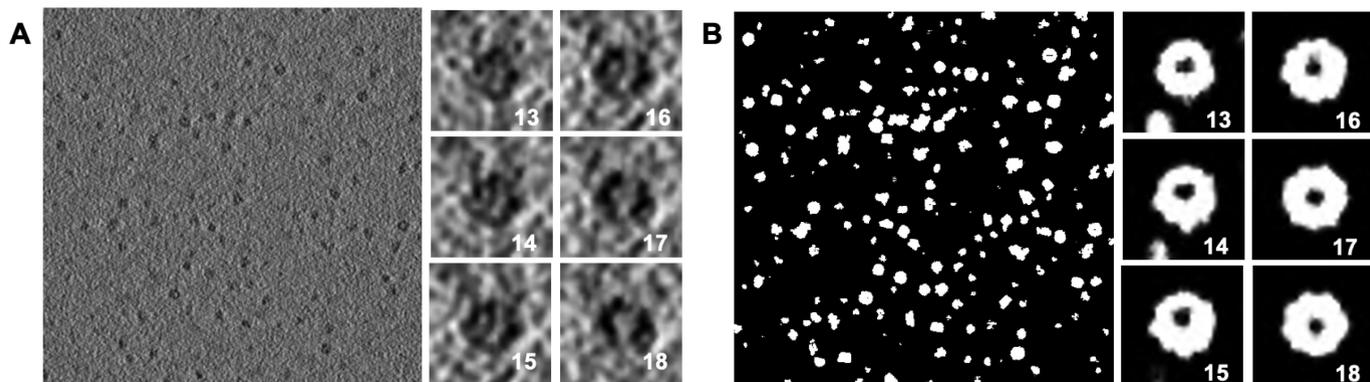


Fig. 4. The example of synthetic data. (A) The middle slice of one  $512 \times 512 \times 512$  tomogram. The right  $32 \times 32$  slices are the consecutive slices of a subtomogram with PDB ID 4d8q. The number of the right corner is their slice index. (B) Ground truth corresponding to Fig. A.

TABLE 1  
Macromolecular Complexes Information of SHREC-Data, Including PDB ID, Name and Molecular Weight

PDB	3gl1	3qm1	1s3x	3h84	2cg9	3d2f
Name	Ssb1, Hsp70	LJ0536 S106A	Hsp70 ATPase	GET3	Hsp90-Sba1	Sse1p, Hsp70
Mol. weight (kDa)	84.61	62.62	42.75	158.08	188.73	236.11
PDB	1u6g	3cf3	1bxn	1qvr	4d8q	4b4t
Name	Cand1-Cul1-Roc1	P97/vcp	Rubisco	ClpB	TRiC/CCT	26S proteasome
Mol. weight (kDa)	238.82	541.74	559.96	593.36	1309.28	1952.74

are published by EMPIAR [26] with the index of 10172, 10169, 10143, 10135, 10133, and 10131. Each tilt-series contains only one category of macromolecules. According to the sort of above EMPIAR index, the macromolecules categories are insulin-bound insulin receptor, hemagglutinin, apoferritin, T20S proteasome, DNAB helicase-helicase, glutamate dehydrogenase, and rabbit muscle aldolase. Here, we denote experimental data as EMPIAR-data. To obtain subtomograms, first, we applied the difference of Gaussians (DoG) [32] to denoise tomograms. Then we manually picked up 400 macromolecules from each tomogram with

size  $28 \times 28 \times 28$ . In Fig. 6, we present each slice of the 3D subtomogram, and we can see that the SNR of EMPIAR-data is higher than SHREC-data. As experimental data does not have ground-truth information, we can not generate noise-free EMPIAR-data.

### 3.2 Training Details

All CNN models in this work are implemented with Pytorch and trained and tested on 2 GTX 1080ti GPUs. In the training stage, for hyperparameters, the batch size is 64, the training epoch is 30, the optimizer is Adam, and the initial value of the learning rate is 0.1. After every ten epochs, the learning rate is linearly scaled to 0.1 times the original value. To better convergence, we applied the Xavier algorithm to initialize the networks [33]. For FC layers and convolution layers, the weights are random values that obey

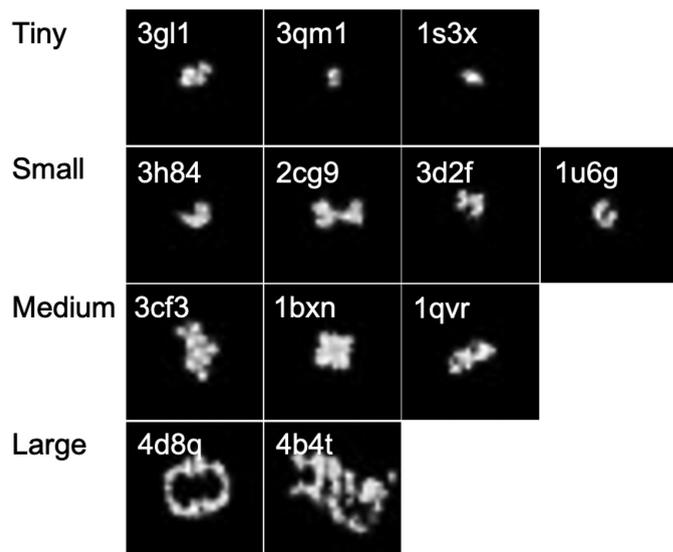


Fig. 5. The density map and its size of each macromolecule in SHREC-data.

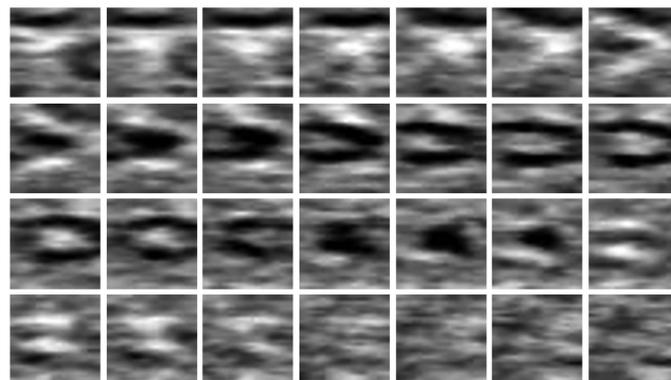


Fig. 6. The example of experimental data. Each slice corresponds to a depth of 3D subtomograms.

TABLE 2  
Each Macromolecule Classification F1 Score on SHREC-Data

Model	PDB ID											
	tiny			small				medium			large	
	1s3x	3qm1	3gl1	3d2f	1u6g	2cg9	3h84	1qvr	1bxn	3cf3	4b4t	4d8q
SHREC-CNN	0.154	0.193	0.318	0.584	0.522	0.343	0.332	0.8	0.904	0.784	0.907	0.951
3D-Dilated-DenseNet	<b>0.684</b>	<b>0.485</b>	<b>0.675</b>	<b>0.778</b>	<b>0.652</b>	<b>0.565</b>	<b>0.635</b>	<b>0.855</b>	<b>0.971</b>	<b>0.846</b>	<b>1</b>	<b>0.997</b>

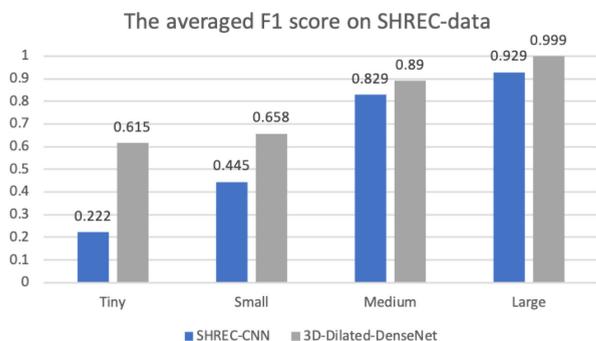


Fig. 7. The averaged F1 score of the SHREC-CNN model and 3D-Dilated-DenseNet on different size.

[-a, a] uniform distribution, where  $a = \sqrt{\frac{6}{n_{in} + n_{out}}}$ ,  $n_{in}$  and  $n_{out}$  is the channel size of input and output feature maps. For the batch normalization layer,  $\gamma$  is set as 1,  $\beta$  is set as 0, and all biases are set as 0.

### 3.3 The Experiments of 3D-Dilated-DenseNet on Synthetic Data

#### 3.3.1 The Classification Performance of 3D-Dilated-DenseNet

Here, we show the performance of 3D-Dilated-DenseNet on SHREC-data. First, we tested 3D-Dilated-DenseNet on SHREC-data and compared 3D-Dilated-DenseNet with the state-of-the-art method on the SHREC contest (SHREC-CNN) [25]. Second, we provided a confusion matrix to analyze the bottleneck of 3D-Dilated-DenseNet.

Because the F1 metric can show the balance of accuracy and recall, the F1 metric is commonly used to evaluate the performance of classification methods. Also due to the SHREC contest only provides the F1 metric of the state-of-the-art method, we adopted the F1 metric (Eq. (6)) to compare the performance of SHREC-CNN and 3D-Dilated-DenseNet (Table 2)

$$F1 = \frac{2 * \text{recall} * \text{precision}}{\text{recall} + \text{precision}} = \frac{2TP}{2TP + FN + FP}. \quad (6)$$

In Eq. (6). TP means true positive, FN means false negative and FP means false positive.

The results of Table 2 show that, first, compared with SHREC-CNN, 3D-Dilated-DenseNet highly improves the classification performance, especially for tiny- and small-size macromolecules. Second, the classification performance has a high relationship to macromolecule size. To further explore this relationship, we averaged the F1 score of macromolecules that belong to the same size (Fig. 7). From Fig. 7, we can see that for both 3D-Dilated-DenseNet and SHREC-CNN,

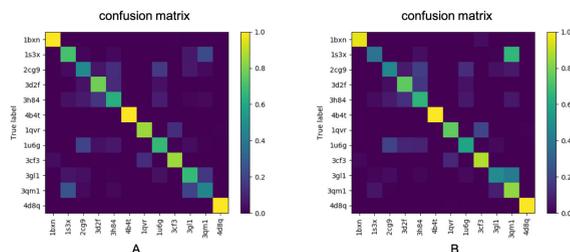


Fig. 8. The confusion matrix of the 3D-Dilated-DenseNet. (A) The confusion matrix of equal sampling weight at the training stage (B) The confusion matrix with increased 3qm1 sampling weight at the training stage. The  $y$ -axis represents truth label, and the  $x$ -axis represents predicted label. A brighter pixels means that the model is more likely to classify the macromolecule to the category of the  $x$ -axis.

the larger the macromolecule size, the better the model classification performance. For large-size macromolecule, the F1 score of both networks is higher than 0.9. And the F1 score of 3D-Dilated-DenseNet is almost close to 1. For small-size macromolecules, the model performance gets poorer. The classification performance on tiny-size macromolecules is the worst. This makes sense because, when extracting highly abstract features by a series of convolution and pooling processes, larger macromolecules can keep more structural information and is easier to be classified.

Here, to explore the performance bottleneck of 3D-Dilated-DenseNet, we provide a confusion matrix (Fig. 8). The confusion matrix can describe the accuracy of the classification model, especially to see if the model confuses two types of categories. The row of the confusion matrix represents the predicted category, while the column represents the truth category. From Fig. 8 A, we can see that 3D-Dilated-DenseNet easily confused 3qm1 with the other two tiny-size macromolecules: 1s3x and 3gl1. To improve classification accuracy on macromolecule 3qm1, we increased the sampling weight of 3qm1 at the training stage. The result (Fig. 8 B) shows that increasing the sampling weight of 3qm1 does help improve the classification accuracy of 3qm1. However, it makes the model confuse 1s3x and 3gl1 into 3qm1. The main reason for this is that, first, it is hard for the CNN model to extract representative structure features in tiny-size macromolecules that have low SNR. Second,  $\sim 1700$  samples of each category macromolecule are not enough. Much more training data is needed for improving tiny-size macromolecule classification.

#### 3.3.2 Ablation Studies

Our 3D-Dilated-DenseNet contains three key components: dense connection,  $1 \times 1$  convolution layer of bottleneck

TABLE 3  
The Ablation Study of Each Network Component

Model Name	dense connection	bottleneck component	dilated convolution	tiny	small	medium	large
3D-base-CNN				0.498	0.520	0.840	0.992
3D-DenseNet	✓			0.583	0.594	0.870	0.999
3D-C-DenseNet	✓	✓		0.581	0.586	0.853	0.999
3D-Dilated-DenseNet	✓	✓	✓	<b>0.615</b>	<b>0.658</b>	<b>0.890</b>	<b>0.999</b>

module, and dilated convolution layer. Here, we provide an ablation study to explore the contribution of each key component. There are four CNN models tested in the ablation study, and each model is updated by adding a new component. First, we designed a base CNN model and noted it as 3D-base-CNN. In 3D-base-CNN, the feature extractor module only contains stacking composite functions: BN-ReLU- $3 \times 3 \times 3$  convolutional layer, the transition module only has BN-average pooling layer, and the classifier is the same as the 3D-Dilated-DenseNet. Based on 3D-base-CNN, we connected each convolution layer with dense shortcuts and updated 3D-base-CNN to 3D-DenseNet. Then based on 3D-DenseNet, we applied a  $1 \times 1 \times 1$  convolutional layer in the transition module and composite functions of each feature extractor and got the new model 3D-C-DenseNet. Finally, we used a dilated convolution layer to substitute the standard convolution layer and obtained the 3D-Dilated-DenseNet. Table 3 shows the performance contribution of each key component to different size macromolecules. Here, the F1 metric is adopted to evaluate the model performance. We first computed the F1 score of each category of the macromolecule. Then averaged the F1 scores of the same size macromolecule. The results shown in Table 3 helped us verify the two following things.

*Dense connection and dilated convolution both have a significant contribution to macromolecule classification tasks.* From Table 3, for each model, because extracting discriminative features for large-size macromolecules is easy, the results show no difference in large-size macromolecules classification. The F1 scores on large-size macromolecules are all close to 0.99. Here, we pay more attention to the classification performance of tiny- and small- macromolecules. By comparing 3D-base-CNN with 3D-DenseNet, we can see that 3D-DenseNet averagely increased classification performance by 8.5 percent on tiny-size macromolecule, 7.3 percent on small-size macromolecule, and 2.7 percent on medium-size macromolecule. Due to the  $1 \times 1 \times 1$  convolutional layer is mainly adopted to improve model training efficiency, it shows no significant performance improvement. By comparing 3D-C-DenseNet with 3D-Dilated-DenseNet, the result shows that the dilated convolutional layer averagely increased classification performance by 5.3 percent on tiny-size macromolecule, 6.8 percent on small-size macromolecule, 3.7 percent on medium-size macromolecule.

*The  $1 \times 1 \times 1$  convolutional layer of the bottleneck module significantly improves model training efficiency.* To explore the efficiency of adopting  $1 \times 1 \times 1$  convolutional layer, we compared the parameter size, GPU memory, and training time of each model (Table 4). As the 3D-base-CNN does not contain dense shortcuts, the parameter size, GPU memory,

and training time of 3D-base-CNN are the least. For 3D-DenseNet, as we adopted dense shortcuts to enhance feature map utilization, the number of channels shows multiple increases. Further, the parameter size and the GPU memory of 3D-DenseNet are apparently growing, and the training time expands to 1.38h. For 3D-C-DenseNet, after applying  $1 \times 1 \times 1$  convolution, the parameter size of 3D-C-DenseNet has been reduced from 1.62MB to 0.98MB, which further leads to the training time of 3D-C-DenseNet reduces to 1.03h. For 3D-Dilated-DenseNet, as the difference between 3D-C-DenseNet and 3D-Dilated-DenseNet is using dilated convolution layer to substitute standard convolution layer. The parameter size, GPU memory as well as training time of 3D-C-DenseNet and 3D-Dilated-DenseNet are almost the same.

### 3.3.3 The Studies of Dilated-Dense Block With Different Dilation Rates

Here, to explore the effect of different dilation rates on the macromolecule classification performance, we designed four kinds of dilated-dense blocks. Then we tested four 3D-Dilated-DenseNets, each 3D-Dilated-DenseNet only contains one kind of dilated-dense block. Table 5 records the detailed dilation rates of each tested 3D-Dilated-DenseNet and their classification performance on the same size macromolecules, evaluated by the averaged F1 score. Each item of the dilation rate group separately represents the dilation rates of four

TABLE 4  
The Params Size, GPU Memory and Training Time of Each Model

	Params size(MB)	GPU memory (MiB)	Training Time(h)
3D-base-CNN	0.39	3904	0.58
3D-DenseNet	1.62	9270	1.38
3D-C-DenseNet	0.98	11642	1.03
3D-Dilated-DenseNet	0.98	11642	1.03

TABLE 5  
Dilated-Dense Block With Different Dilation Rates

Test ID	Dilation rates in the dilated-dense block	Tiny	Small	Medium	Large
Test 1	1-1-1-1	0.562	0.590	0.853	0.997
Test 2	1-2-1-1	<b>0.621</b>	<b>0.619</b>	0.851	0.997
Test 3	1-1-2-1	0.593	0.585	0.865	0.997
Test 4	2-2-2-2	0.561	0.577	<b>0.881</b>	<b>0.998</b>

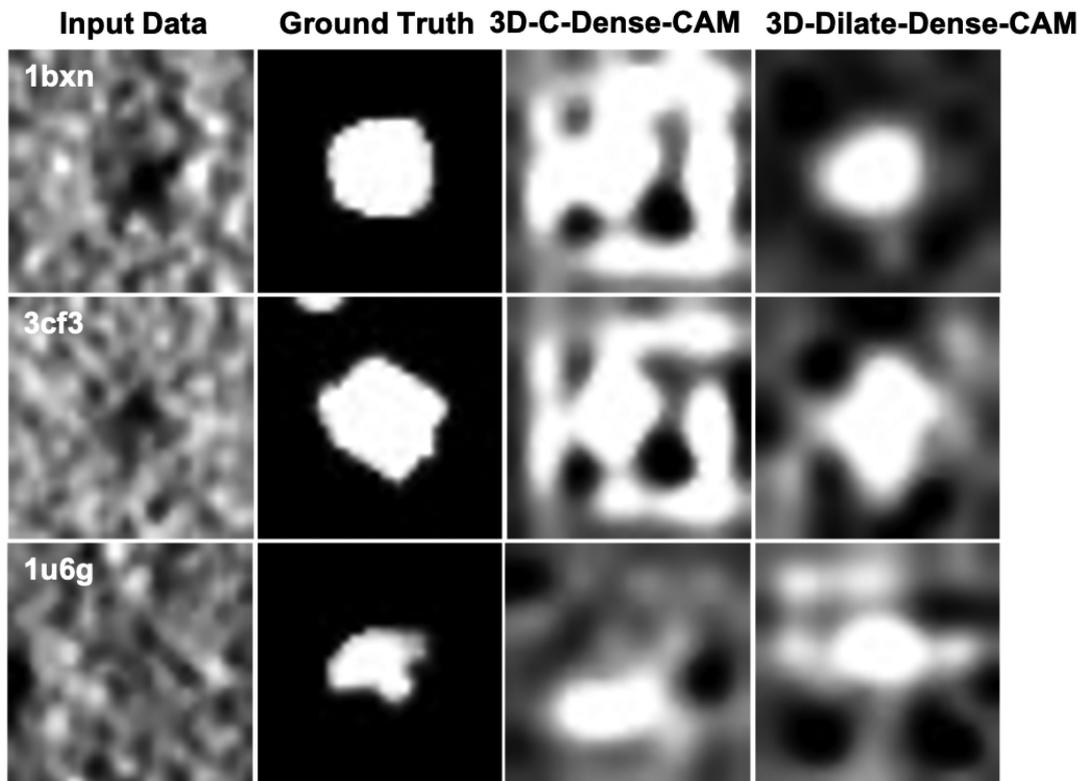


Fig. 9. Class activation mapping image of 3D-C-DenseNet and 3D-Dilated-DenseNet. Each row represents one category of macromolecule. And the column images are raw input data, noiseless ground truth, CAM images of 3D-C-DenseNet and 3D-Dilated-DenseNet

$3 \times 3 \times 3$  dilated convolutional layer in the dilated-dense block. For example,  $1 - 1 - 1 - 1$  represent that the dilation rates of all dilated convolutional layers in the block is 1.

From Table 5, first, by comparing the result of test 1 with the others, we verify that the dilated convolutional layer can increase the macromolecule classification performance. Second, test 2-4 shows that the dilation rate group:  $2 - 2 - 2 - 2$  has the worst performance on tiny- and small-size macromolecules and has the best performance on medium- and large-size macromolecules. This makes sense because, for this kind of block, the receptive field of the 3rd, and 4th dilated convolution layer is too large to extract structure features for small-size macromolecules. On the contrary, it can enrich multi-level features for large-size macromolecules. But overall, we find that the dilated-dense block with the dilated convolutional layer in the front position shows better performance.

### 3.3.4 The Class Activation Mapping of 3D-Dilated-DenseNet

In this experiment, we visualized the class activation mapping (CAM) of both 3D-C-DenseNet and 3D-Dilated-DenseNet to observe the features extracted by models. Good feature maps can generally show the region where the macromolecules are located and their representative structure information. Due to poor SNR of the input subtomograms, for visualizing convenience, we compare the CAM of 3D-C-DenseNet and 3D-Dilated-DenseNet with the noiseless ground truth image (Fig. 9). In Fig. 9, each row represents one kind of macromolecule. From left to right, the displayed images are the input subtomogram, the noiseless ground

truth, CAM of 3D-C-DenseNet, and 3D-Dilated-DenseNet. As all data in Fig. is 3D data, we only present the center slice of the 3D view. In the input subtomogram, the black cluster region represents the macromolecule, which is at the center. In the noiseless ground truth image, the black areas represent the background while the white areas represent macromolecules. In the CAM of 3D-C-DenseNet and 3D-Dilated-DenseNet, the bright white regions represent activation regions that have a contribution to classification. And, the area with higher brightness means more contribution during classification.

The results in Fig. show that the CAM of 3D-Dilated-DenseNet presents more representative features. First, comparing with CAM of 3D-C-DenseNet, the CAM of 3D-Dilated-DenseNet clearly shows the areas where contain a macromolecule. Second, the CAM of 3D-Dilated-DenseNet presents clear boundaries and has a similar shape feature to the macromolecules in noiseless ground truth image.

## 3.4 The Experiments of 3D-Dilated-DenseNet on Experimental Data

Here, we tested 3D-Dilated-DenseNet on experimental data. Due to SHREC does not provide reproducing details of SHREC-CNN, here, we make 3D-C-DenseNet as the baseline model to explore the effect of the dilated convolution on experimental data. Table 6 records the classification performance of these two models, which is evaluated by the F1 metric. Due to experimental data has high SNR and large-size macromolecules, the F1 scores on experimental data are obviously higher than synthetic data. Especially for DNAB helicase-helicase and apoferritin, the F1 score is 1, which means FP and FN is equal to 0. For classification

TABLE 6  
Macromolecule Classification F1 Score on Experimental Data

Model	macromolecule category						
	rabbit muscle aldolase	glutamate dehydrogenase	DNAB helicase-helicase	T20S proteasome	apoferritin	hemagglutinin	insulin-bound insulin receptor
3D-C-DenseNet	0.9231	0.9558	1.0	0.9339	1.0	0.9569	0.9958
3D-Dilated-DenseNet	<b>0.9915</b>	<b>0.9655</b>	1.0	<b>0.9917</b>	1.0	<b>0.9677</b>	1.0

performance on other categories, the result shows that compared with 3D-C-DenseNet, dilated convolution averagely increase classification performance by 2.1 percent.

#### 4 DISCUSSION AND CONCLUSION

In STA, macromolecule classification is essential for obtaining a macromolecular structure with sub-molecular resolution. In this work, we proposed a macromolecule classification CNN model (3D-Dilated-DenseNet) to improve macromolecule classification performance. Our 3D-Dilated-DenseNet, with dense connection and dilated convolution, can extract multi-level feature maps and fully utilize feature map to get better feature representation. Considering the large model size introduced by dense connection, we also design a bottleneck component. We tested 3D-Dilated-DenseNet on both synthetic data and experimental data to verify model performance. On synthetic data, the results show that our 3D-Dilated-DenseNet significantly outperforms the state-of-the-art model on the SHREC contest. To quantify the contribution of each key component, we provide an ablation study. The result shows that both dilated convolution and dense connection have a significant contribution to the classification performance. And the bottleneck component can reduce the model size and improve training efficiency. To further demonstrate the validity of dilated convolution in the macromolecule classification task, we explored four dilated-dense blocks that contain different dilation rates and visualized the feature map of 3D-C-DenseNet and 3D-Dilated-DenseNet. The results show that first, it is best that the receptive field of the stacked dilated convolution not exceed the macromolecule size. Second, the dilated convolution layer can help the model extract a much more representative feature map. On experimental data, compared with 3D-C-DenseNet, the 3D-Dilated-DenseNet also improved macromolecule classification by 2.1 percent. In our future work, we will focus on breaking through the classification performance of tiny- and small-size macromolecules and exploring model performance with more experimental data.

#### ACKNOWLEDGMENTS

This research was supported by the National Key Research and Development Program of China under Grants 2017YFA0504702 and 2017YFE0103900 NSFC projects under Grants 61932018, 62072441, 62072280, and 62072283, Beijing Municipal Natural Science Foundation under Grant No. L182053, And SG was supported by the Postgraduate Study Abroad Program of National Construction on High-level Universities funded by China Scholarship Council. Correspondence should be addressed to M. Xu and F. Zhang

#### REFERENCES

- [1] B. Alberts, "The cell as a collection of protein machines: Preparing the next generation of molecular biologists," *Cell*, vol. 92, no. 3, pp. 291–294, 1998.
- [2] L. H. Hartwell, J. J. Hopfield, S. Leibler, and A. W. Murray, "From molecular to modular cell biology," *Nature*, vol. 402, no. 6761, pp. C47–C52, 1999.
- [3] V. Lucić, A. Rigort, and W. Baumeister, "Cryo-electron tomography: The challenge of doing structural biology in situ," *J. Cell Biol.*, vol. 202, no. 3, pp. 407–419, 2013.
- [4] R. I. Koning and A. J. Koster, "Cryo-electron tomography in biology and medicine," *Ann. Anatomy-Anatomischer Anzeiger*, vol. 191, no. 5, pp. 427–445, 2009.
- [5] C. Liu *et al.*, "The architecture of inactivated SARS-CoV-2 with postfusion spikes revealed by Cryo-EM and Cryo-ET," *Structure*, vol. 28, no. 11, pp. 1218–1224, 2020.
- [6] S. Cai, C. Chen, Z. Y. Tan, Y. Huang, J. Shi, and L. Gan, "Cryo-ET reveals the macromolecular reorganization of *S. pombe* mitotic chromosomes in vivo," *Proc. Nat. Acad. Sci. USA*, vol. 115, no. 43, pp. 10 977–10 982, 2018.
- [7] P. Kiesel *et al.*, "The molecular structure of mammalian primary cilia revealed by cryo-electron tomography," *Nat. Structural Mol. Biol.*, vol. 27, no. 12, pp. 1115–1124, 2020.
- [8] R. Han *et al.*, "AuTom: A novel automatic platform for electron tomography reconstruction," *J. Structural Biol.*, vol. 199, no. 3, pp. 196–208, 2017.
- [9] R. Han, L. Wang, Z. Liu, F. Sun, and F. Zhang, "A novel fully automatic scheme for fiducial marker-based alignment in electron tomography," *J. Structural Biol.*, vol. 192, no. 3, pp. 403–417, 2015.
- [10] C. Che, Z. Xian, X. Zeng, X. Gao, and M. Xu, "Domain randomization for macromolecule structure classification and segmentation in electron cryo-tomograms," in *Proc. IEEE Int. Conf. Bioinf. Biomed.*, 2019, pp. 6–11.
- [11] R. Han *et al.*, "A joint method for marker-free alignment of tilt series in electron tomography," *Bioinformatics*, vol. 35, no. 14, pp. i249–i259, 2019.
- [12] Y. Lü *et al.*, "Spark-based parallel calculation of 3D fourier shell correlation for macromolecule structure local resolution estimation," *BMC Bioinf.*, vol. 21, no. 13, pp. 1–18, 2020.
- [13] R. Li *et al.*, "Few-shot learning for classification of novel macromolecular structures in cryo-electron tomograms," *PLoS Comput. Biol.*, vol. 16, no. 11, 2020, Art. no. e1008227.
- [14] W. Wan and J. Briggs, "Cryo-electron tomography and subtomogram averaging," in *Methods in Enzymology*. Amsterdam, The Netherlands: Elsevier, 2016, pp. 329–367.
- [15] J. O. Ortiz, F. Förster, J. Kürner, A. A. Linaroudis, and W. Baumeister, "Mapping 70S ribosomes in intact cells by cryoelectron tomography and pattern recognition," *J. Structural Biol.*, vol. 156, no. 2, pp. 334–341, 2006.
- [16] A. Bartesaghi, P. Sprechmann, J. Liu, G. Randall, G. Sapiro, and S. Subramaniam, "Classification and 3D averaging with missing wedge correction in biological electron tomography," *J. Structural Biol.*, vol. 162, no. 3, pp. 436–450, 2008.
- [17] M. Xu, M. Beck, and F. Alber, "High-throughput subtomogram alignment and classification by fourier space constrained fast volumetric matching," *J. Structural Biol.*, vol. 178, no. 2, pp. 152–164, 2012.
- [18] R. D. Righetto *et al.*, "High-resolution cryo-EM structure of urease from the pathogen *Yersinia enterocolitica*," *Nat. Commun.*, vol. 11, no. 1, pp. 1–10, 2020.
- [19] Y. Chen *et al.*, "Detection and identification of macromolecular complexes in cryo-electron tomograms using support vector machines," in *Proc. 9th IEEE Int. Symp. Biomed. Imag.*, 2012, pp. 1373–1376.

- [20] C. Che, R. Lin, X. Zeng, K. Elmaaroufi, J. Galeotti, and M. Xu, "Improved deep learning-based macromolecules structure classification from electron cryo-tomograms," *Mach. Vis. Appl.*, vol. 29, no. 8, pp. 1227–1236, 2018.
- [21] R. Lin, X. Zeng, K. Kitani, and M. Xu, "Adversarial domain adaptation for cross data source macromolecule in situ structural classification in cellular electron cryo-tomograms," *Bioinformatics*, vol. 35, no. 14, pp. i260–i268, 2019.
- [22] M. Xu *et al.*, "Deep learning-based subdivision approach for large scale macromolecules structure recovery from electron cryo tomograms," *Bioinformatics*, vol. 33, no. 14, pp. i13–i22, 2017.
- [23] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," in *Proc. Int. Conf. Learn. Representations*, 2016.
- [24] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4700–4708.
- [25] I. Gubins *et al.*, "Classification in cryo-electron tomograms," in *12th Eurographics Workshop on 3D Object Retrieval*, S. Biasotti, G. Lavoué, and R. C. Veltkamp, Eds., Eurographics Assoc., 2019, pp. 49–54. [Online]. Available: <https://doi.org/10.2312/3dor.20191061>
- [26] 2021. [Online]. Available: <https://www.ebi.ac.uk/pdbe/emdb/empiar/>
- [27] 2021. [Online]. Available: <https://www.rcsb.org/>
- [28] P. Wang *et al.*, "Understanding convolution for semantic segmentation," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, 2018, pp. 1451–1460.
- [29] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2921–2929.
- [30] I. Gubins *et al.*, "SHREC 2020: Classification in cryo-electron tomograms," *Comput. Graph.*, vol. 91, pp. 279–289, 2020.
- [31] J. R. Kremer, D. N. Mastronarde, and J. R. McIntosh, "Computer visualization of three-dimensional image data using IMOD," *J. Structural Biol.*, vol. 116, no. 1, pp. 71–76, 1996.
- [32] A. Bundy and L. Wallen, "Difference of Gaussians," in *Catalogue of Artificial Intelligence Tools*. Berlin, Germany: Springer, 1984, pp. 30–30.
- [33] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. 13th Int. Conf. Artif. Intell. Statist.*, 2010, pp. 249–256.



**Xiangrui Zeng** is currently working toward the PhD degree in computational biology at Carnegie Mellon University, Pittsburgh, Pennsylvania. His current research interests include computational biology and computer vision.



**Zhiyong Liu** received the PhD degree in computer science from the Institute of Computing Technology (ICT), Chinese Academy of Sciences (CAS), Beijing, China. He is the chair professor of ICT, CAS. His current research interests include high-performance algorithms and architecture, parallel processing, and bioinformatics.



**Min Xu** received the BE degree in computer science from Beihang University, Beijing, China, the MSc degree from the School of Computing, National University of Singapore, Singapore, the MA degree in applied mathematics from the University of Southern California (USC), Los Angeles, California, and the PhD degree in computational biology and bioinformatics from the USC, Los Angeles, California. He is an assistant professor with Computational Biology Department, School of Computer Science, Carnegie Mellon University. His career has centered on developing computational methods for the study of cellular systems using imaging and omics data.



**Shan Gao** received the BS degree in computer science from Wuhan University, Wuhan, China. She is currently working toward the PhD degree in the Institute of Computing Technology (ICT), Chinese Academy of Sciences (CAS), Beijing, China. Her current research interests include bioinformatics and biomedical image processing.



**Fa Zhang** received the PhD degree in computer science from the Institute of Computing Technology (ICT), Chinese Academy of Sciences (CAS), Beijing, China. He is a chair professor at ICT, CAS. His current research interests include bioinformatics, biomedical image processing, and high-performance computing.



**Renmin Han** received the PhD degree in computer science from the Institute of Computing Technology (ICT), Chinese Academy of Sciences (CAS), Beijing, China. He serves as a professor with the Research Center for Mathematics and Interdisciplinary Sciences, Shandong University. His current research interests include bioinformatics, biomedical image processing, and high performance computing.

▷ For more information on this or any other computing topic, please visit our Digital Library at [www.computer.org/csdl](http://www.computer.org/csdl).