Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

A MATLAB toolbox for 3D chemical field data classification using PointNet++

Haoyu Zhu
Tongren University
Rongling Gu
Zhejiang Hisun Pharmaceutical (China)
Shijie Du
Tongren University
Lu Xu
lxchemo@163.com

Tongren University
Qin Yang
Yangtze University

Research Article

Keywords: PointNet++, 3D molecular field data, MATLAB toolbox, Classification, Point cloud analysis

Posted Date: May 11th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-9055250/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations:
No competing interests reported.

Tables 1 to 4 are available in the Supplementary Files section.

A MATLAB toolbox for 3D chemical field data classification using PointNet++

Haoyu Zhu[a,b], Rongling Gu[c], Shijie Du[b], Lu Xu[a,b*], Qin Yang[d*]

[a] School of Sports and Health Science, Tongren University, Tongren, 554300, Guizhou, PR China

[b] College of Material and Chemical Engineering, Tongren University, Tongren, 554300, Guizhou, PR China

[c] Zhejiang Hisun Pharmaceutical Co., Ltd., Taizhou, 318000, Zhejiang, PR China

[d] School of Physics and Optoelectronic Engineering, Yangtze University, Jingzhou 434023, Hubei, PR China

∗ Corresponding authors.

Tel.: +86 856 62782039; fax: +86 856 62782039.

E-mail addresses: lxchemo@163.com (L. Xu), yangqin00055@hnu.edu.cn (Q. Yang).

Abstract PointNet++ is a widely used deep learning architecture for processing three-dimensional (3D) point cloud data. By hierarchically extracting local features and subsequently aggregating them into global representations, it enables effective learning from unordered point clouds. In chemistry, PointNet++ can be applied to the analysis of 3D molecular structures and molecular field data, because such data, including three-dimensional electron cloud density (ECD) and other molecular fields, can be naturally represented as point clouds. Accordingly, PointNet++ has potential utility in drug design, chemical reaction prediction, and materials science. At present, PointNet++ is implemented predominantly in Python-based environments. To facilitate its use in 3D chemical data analysis, a free MATLAB toolbox was developed in the present study. The toolbox provides two principal functions: (1) conversion of molecular field data into 3D point cloud representations and (2) PointNet++-based classification modeling of 3D point cloud data. The practical utility of the toolbox was demonstrated using real datasets. In a representative application involving a molecular ECD dataset related to p53 activation, the toolbox achieved a classification accuracy of 0.8333 on an external test set, indicating that discriminative local and global features in complex 3D chemical field data can be learned effectively.

Keywords: PointNet++; 3D molecular field data; MATLAB toolbox; Classification; Point cloud analysis

1. Introduction

With the rapid development of artificial intelligence (AI) and machine learning technologies, the processing and analysis of 3D point cloud data has become increasingly important [1-3]. In fields such as autonomous driving [4], robot navigation [5], augmented reality [6], and 3D modeling [7], there is a growing demand for the recognition, classification, and segmentation of 3D objects. Chemists also pose new challenges to existing machine learning methods when analyzing various 3D chemical field data [8-12]. In order to overcome these problems, researchers have developed various algorithms to handle point cloud data [13-16]. PointNet++ is an innovative neural network architecture that has made significant progress in processing point cloud data [17].

The development of PointNet++ can be traced back to 2017, when PointNet was proposed as a deep learning network that directly processes point cloud data. PointNet learns the global features of point clouds using symmetric functions and multi-layer perceptrons (MLPs), enabling classification and segmentation of 3D objects [18]. Although PointNet achieved some success in processing point cloud data, it had limitations in handling local structures and hierarchical features. To overcome these limitations, PointNet++ was proposed in 2018 [17]. PointNet++ added a hierarchical structure to PointNet, enhancing the model's representation ability through multi-scale grouping (MSG) and local feature extraction. This improvement made PointNet++ more effective in handling complex 3D scenes. PointNet++ can more accurately capture local and global features in point cloud data. By using a hierarchical grouping strategy, PointNet++ can simulate the way humans perceive 3D objects, leading to significant performance improvements in classification, segmentation, and detection tasks [17].

In the field of chemistry, molecular field data generally includes 3D spatial coordinates and one or more physical/chemical quantity features, such as ECD [19,20], electrostatic potential field (ESPF) [21-24], van der Waals surface (vdWs) [24,25], etc., which constitute typical 3D point cloud data. The 3D structure of a molecule determines its

reactivity, stability, biological activity, and the way it interacts with other molecules [26,27]. Analysis of 3D molecular structures is crucial for understanding the physical and chemical properties of molecules. Especially, according to the Cohn-Kornberg theorem, the ECD determines all properties of a ground-state molecule. The various 3D molecular field data obtained based on strict quantum mechanics methods (such as density functional theory) are accurate data for characterizing the 3D structure of molecules [28]. Based on accurate ECD data or its derived descriptors, it was anticipated that general quantitative structure–activity relationship (QSAR) models could be established [9,29,30].

Analysis of 3D molecular field data has always been a huge challenge [31,32]. Among the difficulties, the problem of molecular alignment was considered the primary obstacle. Due to the lack of a unified standard, different alignment methods often resulted in significantly different predictive performances [33]. In addition, an ideal 3D descriptor was required to remain invariant under molecular rotation and translation, while still accurately capturing molecular shape—an objective that was difficult to achieve in practice [9,34,35]. On the other hand, the construction of effective 3D descriptors typically required the discretization of molecular fields or shape information in high-dimensional space. This process was not only computationally intensive but also failed to balance short-range and long-range interactions effectively [9,33]. Traditional 3D-QSAR methods, such as comparative molecular field analysis (CoMFA), were widely adopted, but they exhibited significant limitations due to their dependence on field calculations based on fixed grids and probe placements. For example, grid division was arbitrary, probe positions were fixed, and molecular conformational alignment was often inconsistent [23,33,36]. These limitations encouraged researchers to explore new 3D representation strategies that did not rely on explicit alignment—for instance, by utilizing electron density distributions that were inherently invariant to rotation and translation [9,19]. In doing so, the dependence on rigid alignment was reduced, allowing molecular conformations and interaction fields to be described more

112 comprehensively [37]. Therefore, it is necessary to introduce new feature
113 extraction/modeling methods based on 3D molecular ECD from the fields of
114 mathematics or computer science for molecular design and QSAR analysis.
115 As a milestone method in point cloud analysis, PointNet++ offers a powerful tool for
116 molecular structure analysis in chemistry. It not only enables a better understanding of
117 the physical and chemical properties of molecules, but also plays an important role in
118 drug design, QSAR modeling, and related fields. Shortly after its introduction, PointNet
119 was applied to crystal structure research [38-41], where atoms were initially treated as
120 individual points to form point clouds with atomic coordinates and features. In
121 subsequent studies, 3D electron density and electrostatic potential fields were also
122 employed as input features for point cloud models. In the field of drug discovery,
123 PointNet has been used to predict protein–ligand binding affinity from 3D molecular
124 complexes [42], demonstrating strong potential for structural biochemical tasks. In the
125 conceptual framework proposed in this work, a complete system could eventually be
126 developed to predict a broad range of molecular properties, including physical,
127 chemical, and biological activities. As a preliminary step, promoting a point cloud
128 learning approach based on the intrinsic molecular feature of ECD appears particularly
129 important.
130 The original PointNet++ was implemented based on the PyTorch framework using the
131 Python language. For many chemistry researchers who are not very familiar with
132 Python, using MATLAB to implement PointNet++ might be a more user-friendly
133 choice [43]. Therefore, this article aims to provide a complimentary MATLAB toolkit
134 that utilizes PointNet++. The toolkit will offer the following functions: (1) converting
135 relevant chemical molecular field data into a frequently-used 3D point cloud format;
136 and (2) executing classification tasks using PointNet++. The usage of the toolbox will
137 be demonstrated by analyzing two real dataset.

138 **2. Methods**

139 *2.1. PointNet*

PointNet is a deep learning network designed for processing point cloud data. It can learn features directly from raw point clouds without any preprocessing or assumptions. This type of network is particularly suitable for tasks such as classification and segmentation of 3D shapes. The following is a brief introduction to the feature extraction and classification principles of PointNet. Rotational alignment typically refers to converting point cloud data to a uniform coordinate system before processing it, in order to facilitate subsequent processing and analysis. Since molecular field data is usually measured in the same 3D Euclidean space, this rotation is not necessary. Therefore，rotational alignment is omitted in this article, and a more detailed description of PointNet can be found in the original references.

PointNet takes a point cloud as input. In the shape classification task, the point cloud is composed of a set of 3D coordinate points, and each point may also include additional information such as color, intensity, and normal as features. Suppose a point cloud data includes *n* points, for each point, PointNet uses a shared MLP to extract local features:

$$f(\boldsymbol{x}_i) = \mathrm{MLP}(\boldsymbol{x}_i) \quad (i=1, 2, 3, \dots, n)$$

These local features are then processed by global symmetric functions to obtain global features:

$$F(f_1, f_2, \dots, f_n) = \max\{f_1, f_2, \dots, f_n\}$$

In this context, $\boldsymbol{x}_i$ represents the *i*th point in the point cloud, f_i is the local feature extraction function, and *F* represents the global feature. PointNet utilizes symmetric functions (such as maximum, average, etc.) to handle the unordered nature of point clouds. This implies that the network is not sensitive to the arrangement of input points. The global feature is further processed by another MLP to integrate features from different points. In the language of chemometrics, all points are randomly projected, and then the maximum value of the projection is taken as a feature. Multiple features can be extracted by multiple projections.

Finally, PointNet uses a fully connected layer to output classification results. For multi-class classification problems, this is typically a softmax layer. The key advantage of

PointNet lies in its ability to directly process point cloud data without complex data preprocessing steps, making it very effective in handling 3D shapes.

2.2. PointNet++

PointNet++ inherits the concept of PointNet and enhances the model's ability to capture local structures by introducing a MSG strategy. PointNet++ applies PointNet recursively on a nested partitioning of the input point set. PointNet++ feature extraction can be performed in the following 4 steps as:

(1) It first partition the input set of points into overlapping local regions and the region centers are chosen by furthest point sampling (FPS). By FPS, it selects a given number of scattered points from the point cloud to ensure that the distance between the sampled points is as far as possible.

(2) With the selected centers, ball query is performed to define a set of overlapping local regions. Given a query center point and a set of points (point cloud), a ball query finds the points that are within a certain radius of the center point. This radius can be fixed or dynamically adjusted to accommodate different data and tasks. The original literature also introduces some sampling methods designed to deal with variations in spatial point densities. These methods are not discussed in this paper, because chemical molecular field data is generally uniformly gridded.

(3) PointNet is performed on each local region to extract local features. Each local region is represented by the coordinates of its center and the extracted features. By this step, the point cloud is reduced to a set of centers with coordinates and extracted local features.

(4) Repeat steps (1)-(3) on the current center points for several times; and

(5) Subsequently, perform PointNet on all the current center points to obtain the global features of the whole point cloud.

Finally, PointNet++ processes the extracted global-local features using one or more fully connected layers for the final classification. The overall architecture of the classification model was shown in Figure 1. Of course, researchers can also choose

their preferred traditional machine learning or deep learning models for classification/regression using the extracted features.

[Please insert Fig. 1 here.]

3. Data used for demonstration

3.1. Dataset 1

A set of three-dimensional ECD data was used to demonstrate the application of classification algorithms. 130 small molecules were selected from approximately 470 items retrieved from ChEMBL (https://www.ebi.ac.uk/chembl/) [44] in December 2024, from the subset "Bioactivity data for target CHEMBL4096 (Cellular tumor antigen p53) - Activity", based on their reported relevance to p53 activation. This collection, referred to as Dataset 1, was stratified by activity class and randomly divided into training, validation, and test sets in a 8:2:3 ratio. The training set consisted of 40 active and 40 inactive compounds, the validation set contained 10 active and 10 inactive compounds, and the independent external test set comprised 30 labeled molecules (15 active and 15 inactive). The geometric structure of each molecule was optimized using Gaussian 16 software (Gaussian, Inc., Wallingford, USA), employing the B3LYP method with 6-311G(d, p) basis sets. The ECD data were calculated using the Multiwfn software (version 3.6) [45]. For each molecule, a $500 \times 500 \times 500$ cubic grid array was generated, with a grid spacing of 0.2 Bohr to balance data volume and sampling accuracy.

3.2. Dataset 2

In addition, a commonly used benchmark dataset in point cloud learning was employed, consisting of a three-class classification task involving "Chair," "Display," and "Microwave" categories [18]. The training set contained 140, 170, and 90 samples for "Chair," "Display," and "Microwave," respectively; the validation set included 40 samples for each of the three classes; and the test set consisted of 30, 40, and 20 samples for the corresponding categories. The standard data processing procedures were consistent with those used for the dataset 1.

4. MATLAB functions and an example of analysis

The MATLAB codes in the toolbox are written and tested with MATLAB R2024a (Mathworks, Sherborn, MA). The toolkit consists of 2 file folders ('Files01' and 'Files02'), each of which is designed for a specific task as described below.

4.1. Transforming 3D chemical field data into commonly used point cloud format

The codes in the first file folder 'Files01' transform the data into a commonly used point cloud format (*.ply). A brief introduction to the codes in 'Files01' is shown in Table 1 and more detailed descriptions are included in the comments next to the codes.

For MATLAB 3D array data, the function 'd3tom4' transforms it into an $n\times4$ matrix, where n is the number of points in a data cloud. In order to reduce memory and increase computing speed, it allows users to set a threshold, and points with intensity values below this threshold will be discarded and will not be used to form the output point cloud data. In addition, 'd3tom4' allows real coordinate values to be reflected in the generated point cloud, which is realized by considering the interval (grid width) of sampling points.

Two MATLAB scripts are used for data transformation. The script "example01" transforms a MATLAB 3D array data to a point cloud data (*.ply) with the same name and plots the output data as shown in Figure 2. The script "example02" transforms a 4-column *.txt data to a point cloud data (*.ply).

[Please insert Table 1 here.]

[Please insert Fig. 2 here.]

4.2. Performing PointNet++ classification

The second directory "Files 02" comprised code designed to extract global features from point cloud data (.ply) derived from molecular ECD. These features were subsequently used to train and evaluate a classification model. Dataset 1 consisted of 80 molecules for training (40 active and 40 inactive) and 20 molecules for validation (10 active and 10 inactive), along with an independent test set comprising 30 labeled but not folder-separated molecules. Prior to training, each ECD sample was converted into a spatial field containing over 100,000 points, uniformly distributed within a

252 500 × 500 × 500 voxel grid. Each point encoded three-dimensional coordinates and an
253 associated intensity value. A representative example of the ECD point cloud structure
254 is shown in Figure 3.
255 [Please insert Fig. 3 here.]
256 The script “Runme1_train” was employed to define global variables and set training
257 parameters. Based on the structure of the raw data, the input dimensionality was set to
258 4, and the number of points sampled per point cloud was set to 1,000. The training
259 process was conducted for 40 epochs, with a learning rate of 0.00012, an L2
260 regularization coefficient of 0.3, a learning rate decay step of 20, a decay factor of 0.1,
261 a gradient decay factor of 0.7, and a squared gradient decay factor of 0.999.
262 Additionally, the script “Runme1_train” invoked several MATLAB functions from the
263 “func” folder to perform feature transformation and extraction on the input point cloud
264 data (*.ply), which were subsequently used for classification. A brief description of the
265 functions in the “func” folder was provided in Table 2, and more detailed explanations
266 were included in the code comments.
267 [Please insert Table 2 here.]
268 The model training and validation followed the stages below:
269 (1) Data Preparation: Point cloud data were loaded, preprocessed, augmented, and
270 then split into training and validation sets.
271 (2) Model Definition: A PointNet++ classification model was defined, including the
272 implementation of the forward propagation process, which comprised feature
273 extraction, global pooling, and classification layers.
274 (3) Loss Function and Optimizer: The loss function, consisting of cross-entropy loss
275 and an L2 regularization term, was defined. The Adam optimizer was employed to
276 update model parameters during training.
277 (4) Training Phase: During each training iteration, point cloud data were passed
278 through the model to generate predictions, the loss was computed by comparing
279 predicted and true labels, and model parameters were updated via backpropagation.

(5) Validation Phase: During validation, the model was evaluated on a separate dataset with fixed parameters, and its performance was assessed based on validation loss and classification accuracy.

(6) Model Saving: After completing all training epochs, the model was saved as PNET.MAT, which contained the final network parameters and state variables.

It should be emphasized that, compared to the official PointNet toolbox provided by MATLAB, the implemented PointNet++ model introduced a hierarchical local feature extraction mechanism. Modular "Set Abstraction" layers—consisting of FPS, Ball Query for neighborhood construction, and MLPs for local feature encoding—were applied to progressively downsample and aggregate multi-scale local features. The layer parameters were explicitly defined in the code (e.g., KKK = 10 controlled the number of neighboring points; the function 'initializeTransform' aligned feature spaces). Additionally, dynamic learning rate decay (learnRateDropFactor = 0.07) and strong regularization (l2Regularization = 0.3) were adopted to enhance training stability. This allowed the model to capture fine-grained local structures (e.g., locally active regions of molecular ECD) while also achieving robust global abstraction through hierarchical feature integration, significantly improving classification accuracy in complex scenarios. The outputs of the script "Runme1_train" included:

>>dataSizes, a vector containing the number of training and validation samples respectively;

>>labelCountsTrain, a vector containing the counts of each category in the training labels;

>>labelCountsVal, a vector containing the counts of each category in the validation labels;

>>Viewdat, a matrix containing the XYZ coordinates of the point cloud loaded from filenames{1} by pcread, used for plotting the 3D point cloud;

>>gradients, a structure containing the gradients of the model parameters computed by dlfeval2, and subsequently updated by adding the L2 regularization term;

>>loss, a scalar value representing the loss for the current mini-batch (output of dlfeval2);

>>state, a structure containing the updated network state after the forward pass;

>>acc, a scalar value representing the accuracy, computed first on the mini-batch predictions and later updated to the overall accuracy after confusion matrix aggregation;

>>cmat, a sparse matrix of size [numClasses × numClasses], initialized to all zeros and then updated by aggregating validation results using aggreateConfusionMetric;

>>YVal, a 2×20 dlarray containing the validation labels in one-hot encoded format, where each column represents a sample and each row corresponds to a class;

>>YPred, a 2×20 dlarray containing the predicted scores or probabilities for each validation sample, where each column represents the model's output for one sample;

>>YValLabel, a row vector of true class labels for the validation samples, obtained by taking the index of the maximum value in YVal along the first dimension;

>>YPredLabel, a row vector of predicted class labels for the validation samples, obtained by taking the index of the maximum value in YPred along the first dimension. After the script completed, the loss, training accuracy, and validation accuracy of the model on Dataset 1 were obtained. The performance of the classification model was then visualized in plots (Figure 4).

[Please insert Fig. 4 here.]

To further evaluate the applicability of the toolbox, a standard three-class object classification task was conducted, involving objects with distinct morphological characteristics (Figure 5). Compared to Dataset 1, several training parameters were substantially modified. Specifically, the input data dimension was set to 3, the number of sampled points was reduced to 100, the number of training epochs was decreased to 15, and the learning rate was increased to 0.0002. The L2 regularization coefficient and the learning rate decay period were lowered to 0.2 and 5 respectively, while the decay factor and the gradient decay factor were increased to 0.5 and 0.8 respectively. All other settings remained unchanged. These adjustments were made to adapt the PointNet++

classification model to the relatively smaller 3D datasets. The classification results are shown in Figure 6.

[Please insert Fig. 5 here.]

[Please insert Fig. 6 here.]

Subsequently, the script “Runme2_postClassEval” was used to further evaluate the performance of the classification model. This script reloaded the pretrained model weights (PNET.MAT) and reused the feature extraction backbone of PointNet++. The data loading and preprocessing steps were identical to those used in the original classification task. At the final stage, an additional line of code was included to compare the predicted labels with the ground truth, thereby enabling the calculation of classification accuracy based on discretized regression outputs. During execution, the function “pointnetClassifier” output the class probability distribution for each validation sample, stored in the variable “YPred”. Taking Dataset 1 as an example, the first row of the matrix represented the probability for the “active” class, while the second row corresponded to the “inactive” class. The classification decision module then converted the continuous probabilities into discrete class labels using the following code:

```
[~, YValLabel] = max (YVal, [], 1);
[~, YPredLabel] = max (YPred, [], 1);
```

The resulting labels were stored in “YValLabel” and “YPredLabel” respectively. Upon completion, the script produced the following outputs:

>>filenames4_short, a cell array of strings containing the file names of the validation samples, obtained by removing a common path prefix from filenames4 using the erase function; the order of filenames4_short corresponds exactly to the order of validation samples processed by the confusion matrix, enabling easy retrieval of prediction results for each sample;

>>cmat, a confusion matrix updated by calling aggreateConfusionMetric with the previous cmat, YValLabel, and YPredLabel; it accumulates the classification results of

the validation samples processed so far;

>>acc, a scalar representing the classification accuracy, computed as the sum of the diagonal elements of cmat divided by the total number of predictions in cmat.

The evaluation of the classification model revealed accuracy rates of 0.85 and 0.8917 for Dataset 1 and Dataset 2 respectively (Table 3). In addition to accuracy, other classification metrics were calculated using the function “evaluateClassificationFull” and were displayed in the command window, including the per-sample prediction results and detailed evaluation metrics such as Precision, Recall, F1-score, Specificity, and AUC. In addition, the script generated a confusion matrix and a multi-class ROC curve, visualizing the model’s performance on the validation set (Figure 7). Based on the comparison of the results, the classification model was found to perform better on Dataset 2 than on Dataset 1.

[Please insert Table 3 here.]

[Please insert Fig. 7 here.]

Finally, the script “Runme3_test” was employed to further evaluate the generalization capability of the model on an independent external dataset. This script directly applied the PointNet++ classifier from the trained model along with the saved model weights (PNET.MAT) to predict the class labels of molecules contained in the “test_external” subdirectory. During execution, the command-line interface prompted the user to manually input the corresponding ground-truth labels. The predicted labels were then compared with the user-provided true labels, and a set of evaluation metrics consistent with those used during validation—such as accuracy, precision, recall, and F1-score—were computed and reported accordingly. These outputs facilitated a comprehensive analysis of the model’s predictive capability on previously unseen data. The test results showed that the classification accuracies for Dataset 1 and Dataset 2 reached 0.8333 and 0.8889, respectively (Table 4), and the other performance metrics were also largely consistent with those obtained on the validation sets (Figure 8).

[Please insert Table 4 here.]

[Please insert Fig. 8 here.]

5. Discussions and remarks

It is demonstrated that the toolbox is very easy to use. For the PointNet++-based toolbox, only a few key functions are required to perform data preprocessing, network training, and prediction on point cloud classification tasks. Some remarks on the toolbox are as follows.

(1) For point cloud data with 4 dimensions (i.e., 3D coordinates plus an intensity value) sampled from a 500×500×500 space, it is critical to preserve the original spatial distribution. The toolbox avoids centering the coordinates to zero; instead, only normalization and data augmentation (e.g., random rotations, translations, and fixed-point sampling) are applied. This approach ensures that the intrinsic geometric and intensity information is retained for subsequent hierarchical feature extraction.

(2) The toolbox initializes several network modules (input transform, shared MLP layers, feature transform, and classification MLP) with carefully selected channel sizes (e.g., 64, 128, 256, and 512) to capture both local and global features. When tuning the parameters—such as the neighborhood size in local feature extraction or the number of points sampled per cloud—users are encouraged to examine training loss curves and validation accuracy plots. These observations help in avoiding parameter values that are too large or too small, which could otherwise lead to overfitting or underfitting.

(3) Employing the Adam optimizer with L2 regularization and a dynamic learning rate decay schedule, the toolbox effectively stabilizes the training process and accelerates convergence. Real-time monitoring of the loss and accuracy curves, along with the accumulation of confusion matrices on the validation set, provides a comprehensive evaluation framework. Although the current classification labels did not incorporate continuous activity values, which limited the model's performance on borderline samples (as reflected by the RMSE metric), the toolbox

still demonstrated strong classification stability and generalization capability.

(4) For binary tasks where molecules are labeled only as active or inactive, performance metrics such as sensitivity and specificity (derived from the confusion matrix) indicate that the model can reliably distinguish between classes. In contrast, on a smaller dataset (approximately 5000 points per sample) with a three-class problem, the model exhibits significantly improved classification accuracy. This suggests that the toolbox is capable of adapting to different data scales and label granularities, thereby offering robust performance across varying scenarios.

6. Conclusions

In conclusion, a MATLAB-based PointNet++ toolbox was developed for the classification of three-dimensional (3D) molecular field data. The toolbox enables the conversion of molecular field data into point cloud representations and supports the training and evaluation of PointNet++ classification models. Its functionality was first validated in a representative chemical application using a molecular electron cloud density (ECD) dataset associated with p53 activation (Dataset 1), and was further assessed through a supplementary generalization test using a standard three-class point cloud object dataset (Dataset 2). The results showed that the hierarchical PointNet++ model, incorporating local feature abstraction layers and flexible parameter settings, achieved good classification performance, with external independent test accuracies of 0.8333 and 0.8889 for Datasets 1 and 2, respectively. These findings indicate that the toolbox can effectively capture both local and global structural information in complex 3D chemical data while demonstrating adaptability across different tasks. Overall, this toolbox provides a convenient and practical MATLAB implementation for applying PointNet++ to 3D chemical data analysis. Future work may extend the toolbox to additional types of molecular field data and further refine its functionality to support broader applications in chemical data modeling.

Ethics approval and consent to participate

Not applicable

Consent for publication

Not applicable

Availability of data and materials

The toolbox and two test datasets are freely available from the GitHub website https://github.com/zhuhaoyu98/PointNetPlusPlus--toolbox.

Competing interests

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

Funding

This study was financially supported by the National Natural Science Foundation of China (Grant No. 82260896).

Authors' contributions

H.Z.: Writing - original draft, Data curation, Formal analysis, Visualization. R.G.: Data curation. S.D.: Data curation. L.X.: Conceptualization, Methodology, Supervision, Writing - review & editing. Q.Y.: Writing - review & editing. All authors have read and agreed to published version of the manuscript.

Acknowledgments

We sincerely thank Prof. Alin Xia and Prof. Lijuan Tang for their valuable assistance in testing the program. The authors are responsible for any possible errors.

References

[1] J. Döllner, Geospatial artificial intelligence: potentials of machine learning for 3D point clouds and geospatial digital twins, PFG-J. Photogramm. Remote Sens. Geoinf. Sci. 88 (2020) 15-24.

[2] Y. Guo, H. Wang, Q. Hu, H. Liu, L. Liu, M. Bennamoun, Deep learning for 3D point clouds: a survey, IEEE Trans. Pattern Anal. Mach. Intell. 43 (2021) 4338-4364.

[3] B. Haznedar, R. Bayraktar, A. E. Ozturk, Y. Arayici, Implementing PointNet for point cloud segmentation in the heritage context, Herit. Sci. 11 (2023).

[4] A. Asvadi, C. Premebida, P. Peixoto, U. Nunes, 3D Lidar-based static and moving

obstacle detection in driving environments: An approach based on voxels and multi-region ground planes, Robot. Auton. Syst. 83 (2016) 299-311.

[5] X. Wang, Y. Mizukami, M. Tada, F. Matsuno, Navigation of a mobile robot in a dynamic environment using a point cloud map, Artif. Life Robot. 26 (2021) 10-20.

[6] L. Han, T. Zheng, Y. Zhu, L. Xu, L. Fang, Live semantic 3D perception for immersive augmented reality, IEEE Trans. Vis. Comput. Graph. 26 (2020) 2012-2022.

[7] P. Chen, H. Yang, K. Chen, Y. Chen, MVSNet++: learning depth-based attention pyramid features for multi-view stereo, IEEE Trans. Image Process. 29 (2020) 7261-7273.

[8] B. K. Alsberg, N. Marchand-Geneste, R. D. King, A new 3D molecular structure representation using quantum topology with application to structure-property relationships, Chemometr. Intell. Lab. Syst. 54 (2000) 75-91.

[9] L. Xu, Q. Yang, Deep electron cloud-activity and field-activity relationships, J. Chemom. 37 (2023) 3503.

[10] B. Hu, Z. K. Kuang, S. Y. Feng, D. Wang, S. B. He, D. X. Kong, Three-dimensional biologically relevant spectrum (BRS-3D): shape similarity profile based on PDB ligands as molecular descriptors, Molecules. 21 (2016) 1554.

[11] H. Zhang, C. Shen, H. R. Zhang, W. X. Chen, Q. Q. Luo, L. Ding, Discovery of novel DGAT1 inhibitors by combination of machine learning methods, pharmacophore model and 3D-QSAR model, Mol. Divers. 25 (2021) 1481-1495.

[12] J. K. Daré, M. P. Freitas, Different approaches to encode and model 3D information in a MIA-QSAR perspective, Chemometr. Intell. Lab. Syst. 212 (2021) 104286.

[13] T. Hackel, J. D. Wegner, K. Schindler, Joint classification and contour extraction of large 3D point clouds, ISPRS J. Photogramm. Remote Sens. 130 (2017) 231-245.

[14] Z. Liu, X. W. Xiao, S. S. Zhong, W. N. Wang, Y. L. Li, L. Zhang, Z. Xie, A feature-

preserving framework for point cloud denoising, Comput.-Aided Des. 127 (2020) 102857.

[15] Y. C. Liu, J. W. Guo, B. Benes, O. Deussen, X. P. Zhang, H. Huang, TreePartNet: neural decomposition of point clouds for 3D tree reconstruction, ACM Trans. Graph. 40 (2021) 1-12.

[16] A. Olarini, M. Cocchi, V. Motto-Ros, L. Duponchel, C. Ruckebusch, Exploratory analysis of hyperspectral imaging data, Chemometr. Intell. Lab. Syst. 252 (2024) 105174.

[17] C. R. Qi, L. Yi, H. Su, L. J. Guibas, Pointnet++: deep hierarchical feature learning on point sets in a metric space, Adv. Neural Inf. Process. Syst. 30 (2017).

[18] C. R. Qi, H. Su, K. Mo, L. J. Guibas. Pointnet: deep learning on point sets for 3d classification and segmentation, in: IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 652-660.

[19] P. G. Mezey, QSAR and the ultimate molecular descriptor: the shape of electron density clouds, J. Math. Chem. 45 (2009) 544-549.

[20] Z. Xi, Z. H. Yu, Z. W. Niu, S. R. Ban, G. F. Yang, Development of a general quantum-chemical descriptor for steric effects: density functional theory based QSAR study of herbicidal sulfonylurea analogues, J. Comput. Chem. 27 (2006) 1571-1576.

[21] R. T. Kroemer, P. Hecht, K. R. Liedl, Different electrostatic descriptors in comparative molecular field analysis: a comparison of molecular electrostatic and coulomb potentials, J. Comput. Chem. 17 (1996) 1296-1308.

[22] K. C. Tsai, Y. C. Chen, N. W. Hsiao, C. L. Wang, C. L. Lin, Y. C. Lee, M. Y. Li, B. H. Wang, A comparison of different electrostatic potentials on prediction accuracy in CoMFA and CoMSIA studies, Eur. J. Med. Chem. 45 (2010) 1544-1551.

[23] G. E. Kellogg, S. Phatak, A. Nicholls and J. A. Grant, Validation of poisson-boltzmann electrostatic potential fields in 3D QSAR: a CoMFA study on multiple datasets, QSAR Comb. Sci. 22 (2004) 959-964.

[24] R. Concu, G. Podda, E. Uriarte and H. González-Díaz, Computational chemistry study of 3D-structure-function relationships for enzymes based on markov models for protein electrostatic, HINT, and van der waals potentials, J. Comput. Chem. 30 (2009) 1510-1520.

[25] D. Douguet, F. Payan, Sensaas: shape-based alignment by registration of colored point-based surfaces, Mol. Inform. 39(2020).

[26] T. Sato, H. Yuki, D. Takaya, S. Sasaki, A. Tanaka, T. Honma, Application of support vector machine to three-dimensional shape-based virtual screening using comprehensive three-dimensional molecular shape overlay with known inhibitors, J. Chem. Inf. Model. 52 (2012) 1015-1026.

[27] G. Yalcin-Ozkat, Molecular modeling strategies of cancer multidrug resistance, Drug Resist. Updates. 59 (2021) 100789.

[28] S. Axelrod, R. Gómez-Bombarelli, GEOM, energy-annotated molecular conformations for property prediction and molecular generation, Sci. Data. 9 (2022) 185.

[29] F. Ozgencil, G. Eren, Field-based 3D-QSAR studies on amide- and urea-containing NAMPT inhibitors, Chemometr. Intell. Lab. Syst. 242 (2023) 104984.

[30] J. M. Li, T. Zhao, Q. Yang, S. J. Du, L. Xu, A review of quantitative structure-activity relationship: the development and current status of data sets, molecular descriptors and mathematical model, Chemometr. Intell. Lab. Syst. 256(2025): 105278.

[31] G. Klebe, U. Abraham, On the prediction of binding-properties of drug molecules by comparative molecular-field analysis, J. Med. Chem. 36 (1993) 70-80.

[32] C. Cruciani, P. Crivori, P. A. Carrupt, B. Testa, Molecular fields in quantitative structure-permeation relationships: the VolSurf approach, J. Mol. Struct. 503 (2000) 17-30.

[33] J. G. Wilkes, I. B. Stoyanova-Slavova, D. A. Buzatu, Alignment-independent technique for 3D QSAR analysis, J. Comput.-Aided Mol. Des. 30 (2016) 331-345.

[34] A. Pop, V. Domsa, L. Tamas, Rotation invariant graph neural network for 3D point clouds, Remote Sens. 15 (2023).

[35] J. Cremer, L. M. Sandonas, A. Tkatchenko, D. A. Clevert, G. De Fabritiis, Equivariant graph neural networks for toxicity prediction, Chem. Res. Toxicol. 36 (2023) 1561-1573.

[36] M. Xu, A. Q. Zhang, S. K. Han, L. S. Wang, Studies of 3D-quantitative structure-activity relationships on a set of nitroaromatic compounds: CoMFA, advanced CoMFA and CoMSIA, Chemosphere. 48 (2002) 707-715.

[37] S. Chtita, A. Belhassan, M. Bakhouch, A. I. Taourati, A. Aouidate, S. Belaidi, M. Moutaabbid, S. Belaaouad, M. Bouachrine, T. Lakhlifi, QSAR study of unsymmetrical aromatic disulfides as potent avian SARS-CoV main protease inhibitors using quantum chemical descriptors and statistical methods, Chemometr. Intell. Lab. Syst. 210 (2021) 104266.

[38] R. S. DeFever, C. Targonski, S. W. Hall, M. C. Smith, S. Sarupria, A generalized deep learning approach for local structure identification in molecular simulations, Chem. Sci. 10 (2019) 7503-7515.

[39] Z. Y. Shen, Y. Z. S. Sun, T. P. Lodge, J. I. Siepmann, Development of a pointnet for detecting morphologies of self-assembled block oligomers in atomistic simulations, J. Phys. Chem. B. 125 (2021) 5275-5284.

[40] S. Kim, B. D. Lee, M. Y. Cho, M. Pyo, Y. K. Lee, W. B. Park, K. S. Sohn, Deep learning for symmetry classification using sparse 3D electron density data for inorganic compounds, NPJ Comput. Mater. 10 (2024).

[41] S. Ahmadi, M. A. Ghanavati, S. Rohani, Machine learning-guided prediction of cocrystals using point cloud-based molecular representation, Chem. Mater. 36 (2024) 1153-1161.

[42] Y. J. Wang, S. Wu, Y. W. Duan, Y. Huang, A point cloud-based deep learning strategy for protein-ligand binding affinity prediction, Brief. Bioinform. 23 (2022).

[43] V. Consonni, G. Baccolo, F. Gosetti, R. Todeschini, D. Ballabio, A MATLAB

toolbox for multivariate regression coupled with variable selection, Chemometr. Intell. Lab. Syst. 213 (2021) 104313.

[44] D. Mendez, A. Gaulton, A. P. Bento, J. Chambers, M. De Veij, E. Félix, M. P. Magariños, J. F. Mosquera, P. Mutowo, M. Nowotka, M. Gordillo-Marañón, F. Hunter, L. Junco, G. Mugumbate, M. Rodriguez-Lopez, F. Atkinson, N. Bosc, C. J. Radoux, A. Segura-Cabrera, A. Hersey, A. R. Leach, ChEMBL: towards direct deposition of bioassay data, Nucleic Acids Res. 47 (2019) 930-940.

[45] T. Lu, F. W. Chen, Multiwfn: A multifunctional wavefunction analyzer, J. Comput. Chem. 33 (2012) 580-592.

Captions of figures and tables:

Fig. 1. Schematic of the PointNet++ classification model architecture.

Fig. 2. Electron cloud density data of a molecule plotted as a point cloud.

Fig. 3. Point cloud structure of (a) active and (b) inactive molecules in the transformed dataset 1.

Fig. 4. Performance of the PointNet++ classification model on the dataset 1.

Fig. 5. Point cloud structures of (a) chair, (b) display, and (c) microwave objects in the transformed dataset 2.

Fig. 6. Performance of the PointNet++ classification model on the dataset 2.

Fig. 7. Visualization outputs from the evaluation phase, including the confusion matrices of (a) Dataset 1 and (b) Dataset 2, as well as the ROC curves of (c) Dataset 1 and (d) Dataset 2.

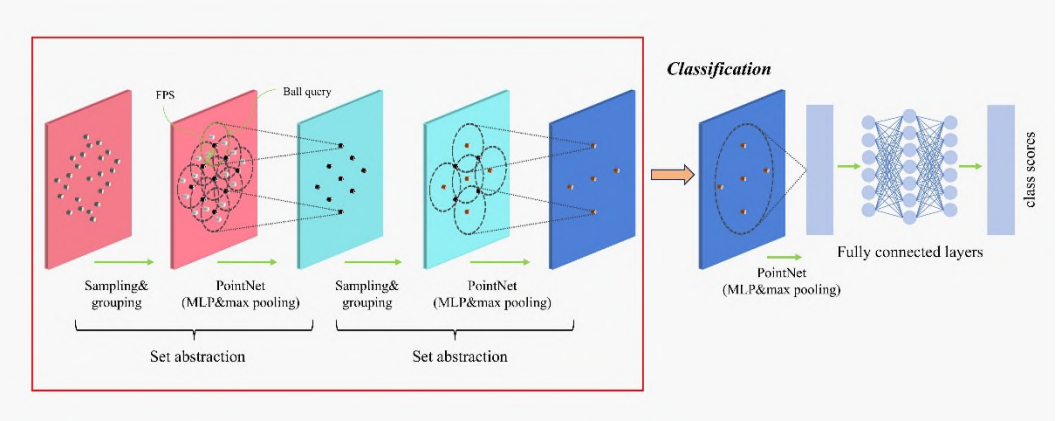
Fig. 8. Visualization outputs from the external testing phase, including the confusion matrices of (a) Dataset 1 and (b) Dataset 2, as well as the ROC curves of (c) Dataset 1 and (d) Dataset 2.

Table 1 MATLAB Functions and scripts used in the file folder 'Files01'.

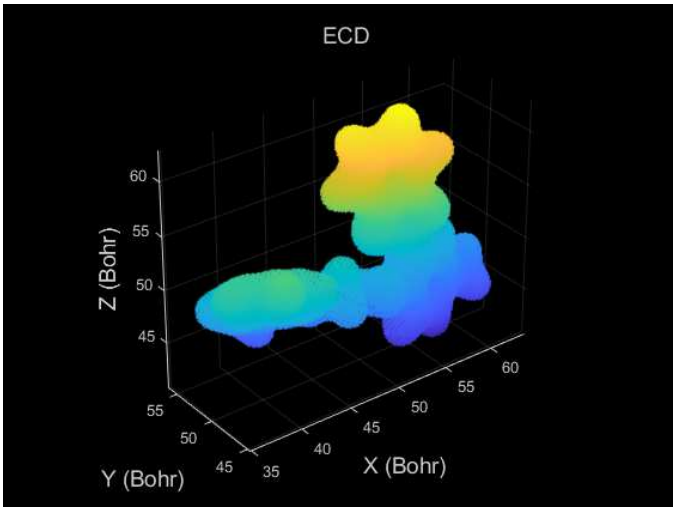
Table 2 MATLAB Functions used in the file folder 'func'.

Table 3 Evaluation results of the classification model.

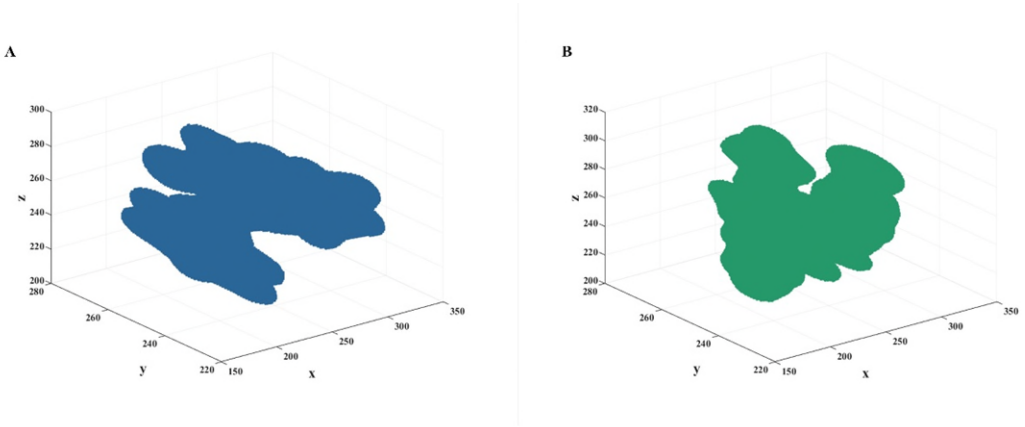
Table 4 External test results of the classification model.



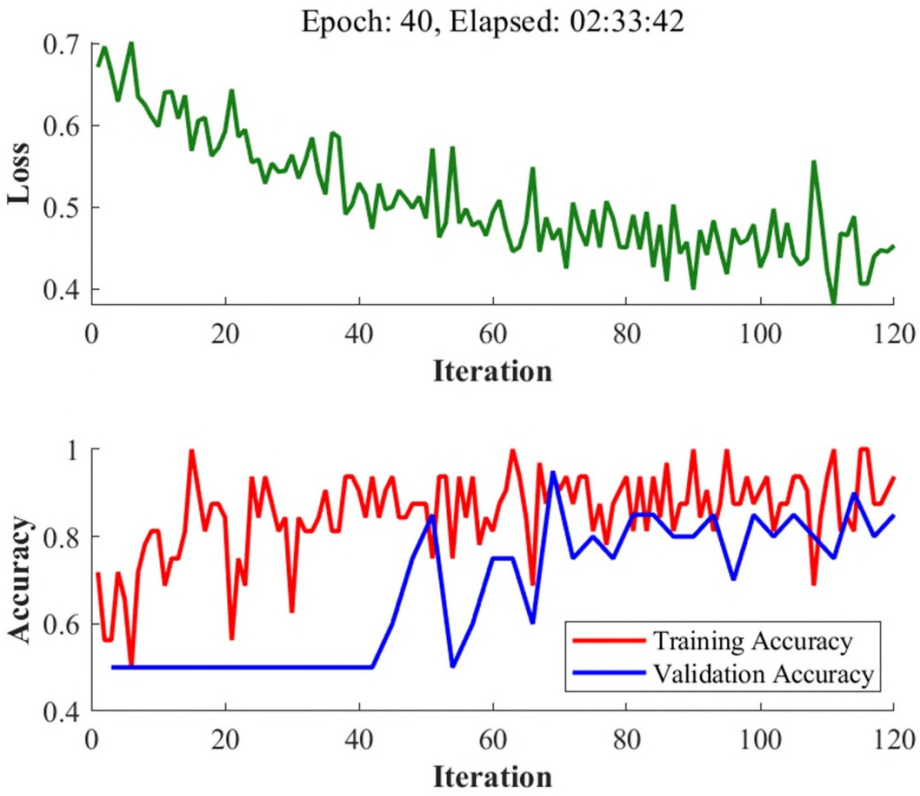
[Figure 1] Schematic of the PointNet++ classification model architecture



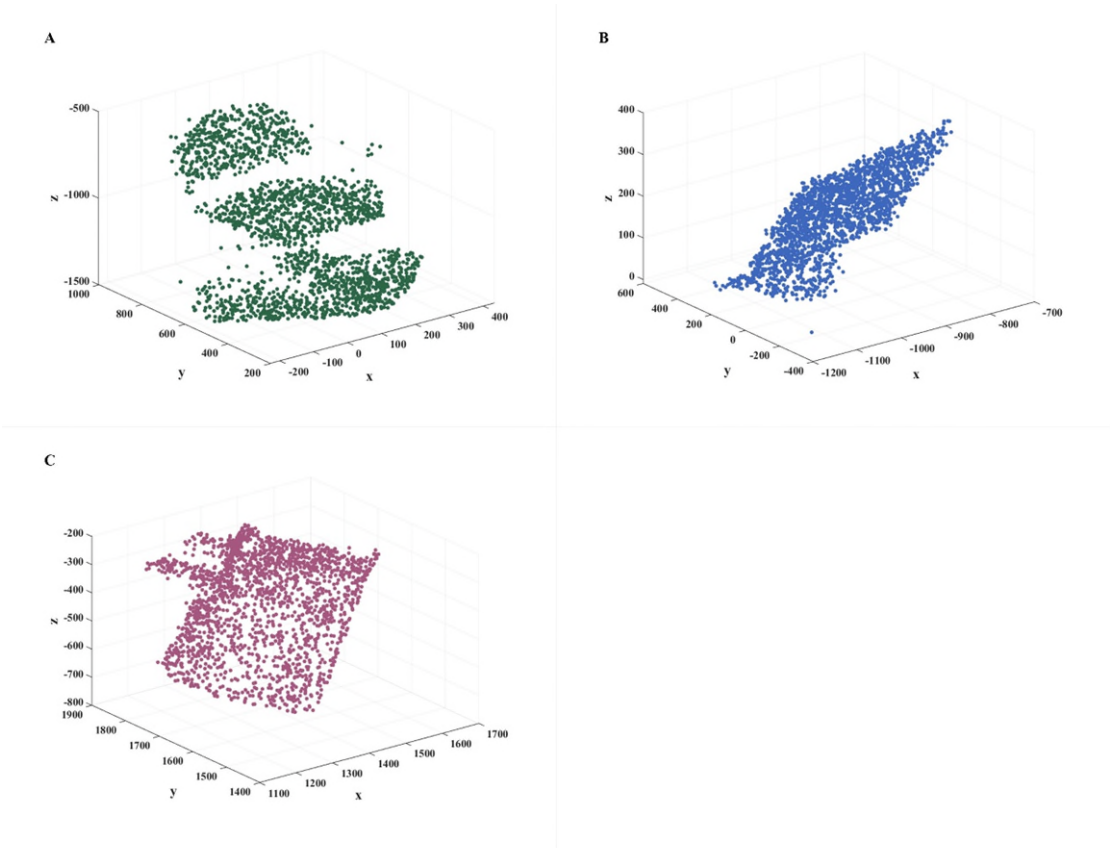
[Figure 2] Electron cloud density data of a molecule plotted as a point cloud



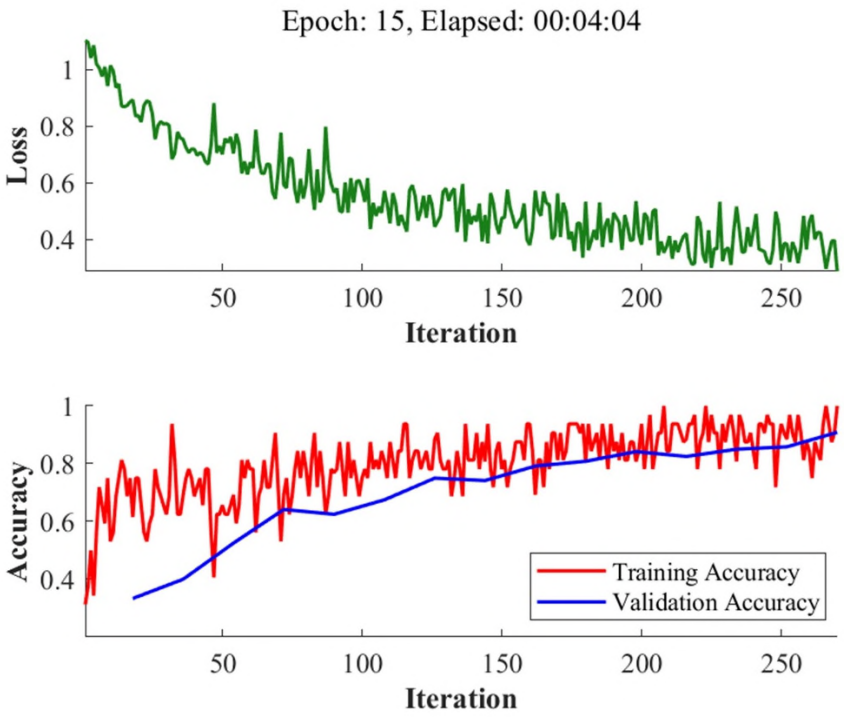
[Figure 3] Point cloud structure of (a) active and (b) inactive molecules in the transformed dataset 1



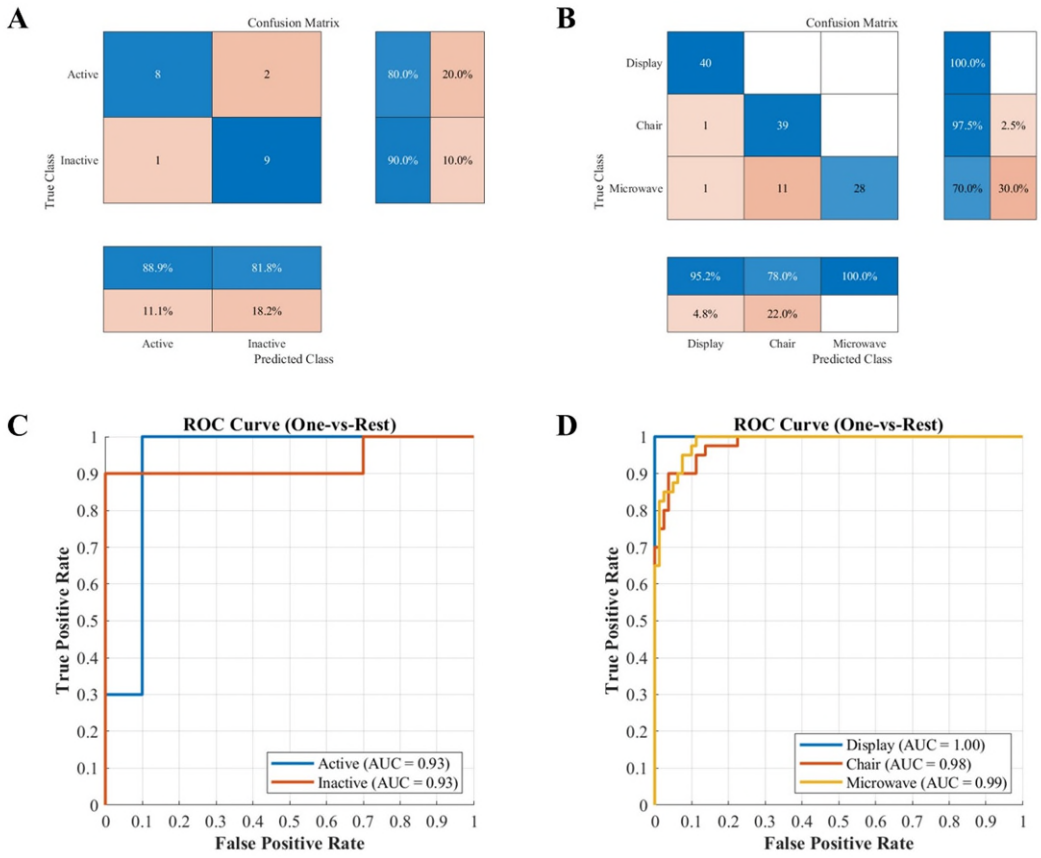
[Figure 4] Performance of the PointNet++ classification model on the dataset 1



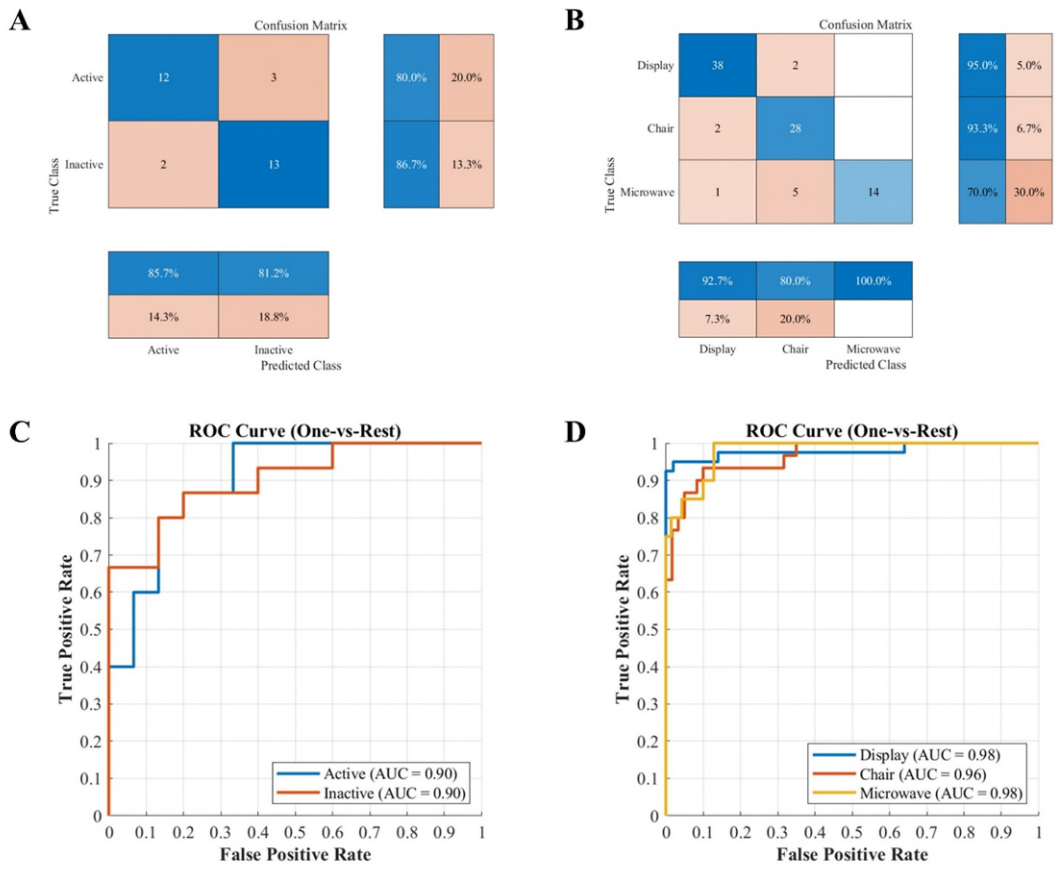
[Figure 5] Point cloud structures of (a) chair, (b) display, and (c) microwave objects in the transformed dataset 2



[Figure 6] Performance of the PointNet++ classification model on the dataset 2



[Figure 7] Visualization outputs from the evaluation phase, including the confusion matrices of (a) Dataset 1 and (b) Dataset 2, as well as the ROC curves of (c) Dataset 1 and (d) Dataset 2.



[Figure 8] Visualization outputs from the external testing phase, including the confusion matrices of (a) Dataset 1 and (b) Dataset 2, as well as the ROC curves of (c) Dataset 1 and (d) Dataset 2.

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- Table1.docx
- Table2.docx
- Table3.docx
- Table4.docx