SHREC 2025:
Protein Shape Classification

Tin Barisin, Eugen Rusakov, Udo Göbel

CONTACT AI Research, CONTACT Software GmbH

Wiener Straße 1–3, 28359 Bremen, Germany

Correspondence: tin.barisin@contact-software.com

In practice, proteins can be recorded in arbitrary poses and aligning them in a canonical pose within the same class is a tedious task. Furthermore, even determining canonical pose of protein over many classes is ambiguous. This is, in contrast to human-made objects such as furniture, which are found in a few standard poses. Hence, extracting and processing rotation invariant features is desirable. Here, we aim to explore rotation invariant deep learning in the context of protein shape classification.

Preprocessing:

We split training set into 7,428/1,816 (approx. 80:20) as training/validation split. For every *vtk* file we normalize the mesh to fit into a unit sphere and extract point cloud with 8196 points using *trimesh* library. Furthermore, for every mesh separately we normalize chemical features to be in [0,1] interval. We use nearest neighbour as an interpolation method to assign potential and normal potential features to the sampled point cloud.

Method:

We use a simplified version of RIConv++ network [ZHY22] with three simple rotation invariant features per layer. Original work extract eight RI features on the local neighbours and relies on the "local" ordering of the neighbours. By keeping only three features, we remove the need to induce the order in the neighbourhood and hence simplifying the feature extraction.

Rotation invariant (RI) layer of network consists of several steps:

1. FPS (furthest point sampling algorithm) to subsample point cloud,
$$P \in \mathbb{R}^{N_0 \times 6} \xrightarrow{FPS} \tilde{P} \in \mathbb{R}^{N \times 6}$$
2. K-nearest neighbour algorithm to create local neighbourhood,

CONTACT Software

$$\tilde{P} \in \mathbb{R}^{\frac{N}{2}\times 6} \xrightarrow{KNN} \widetilde{P_K} \in \mathbb{R}^{\frac{N}{2}\times K\times 6}$$

3. rotation invariant feature extraction for every point in the local neighbourhood,

$$\widetilde{P_K} \in \mathbb{R}^{\frac{N}{2}\times K\times 6} \xrightarrow{RI} \widetilde{P_{RI}} \in \mathbb{R}^{\frac{N}{2}\times K\times 3}$$

4. MLP projector of RI features to higher dimension,

$$\widetilde{P_{RI}} \in \mathbb{R}^{\frac{N}{2}\times K\times 3} \xrightarrow{MLP} \widetilde{P_{MLP}} \in \mathbb{R}^{\frac{N}{2}\times K\times C_0}$$

5. combination with the features from the previous layer (if any) with MLP projection,

$$[\widetilde{P_{MLP}}, F_{-1}] \in \mathbb{R}^{\frac{N}{2}\times K\times (C_0+C_1)} \xrightarrow{MLP} F_K \in \mathbb{R}^{\frac{N}{2}\times K\times C}$$

6. max pooling over local neighbourhood

$$F_K \in \mathbb{R}^{\frac{N}{2}\times K\times C} \xrightarrow{MaxPool} F \in \mathbb{R}^{\frac{N}{2}\times C}$$

Let (p, n_p) be reference point and (x, n_x) point in its local neighbourhood. Then following three rotation invariant features are extracted

$$d = ||x - p||, \qquad \alpha_1 = \angle(\overline{xp}, n_p), \qquad \alpha_2 = \angle(\overline{xp}, n_x).$$

Note that the chemical features given in the datasets: Potential and Normal Potential are both scalars and are hence considered to be rotation invariant, too. Optionally, (for RUN 2) they can be incorporated into the layer at the step 5. This enables efficiently combination of geometric and chemical information.

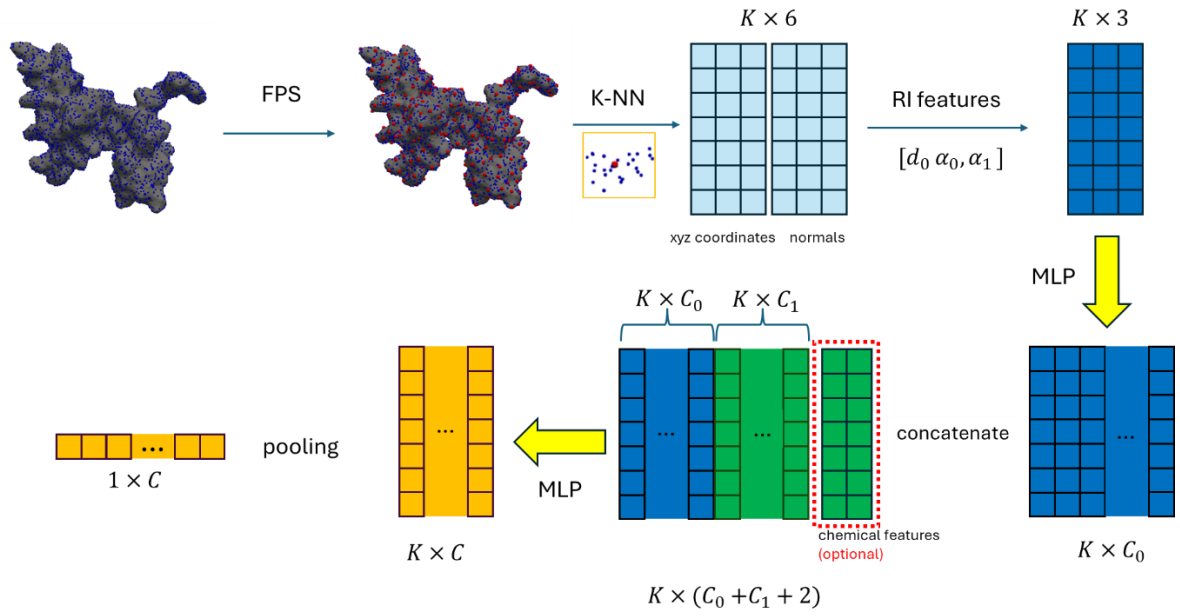


Figure 1: Rotation invariant layer of our network that consists of 6 steps, as described in text.

Network architecture

We stack for 4 RI layers to incorporate several scales of point clouds sampled by FPS algorithm. This includes 4 subsampled versions of point cloud with sizes of 1024, 512, 256 and 128 points. Final output of the fourth layer is max-pooled in a 1024-dim embedding vectors, which is processed with two-layer MLP (512,256) to obtain class probabilities. In total, network has 5.7 million parameters.

We trained all our models for 250 epochs using Adam optimizer with a default learning rate 0.001 and step learning rate decay every 20 epochs on a single GPU NVIDIA RTX 3500 Ada with Intel i7 - 13700HX CPU.

Besides application of rotation invariant deep learning to this problem, contributions of our approach are following: exploring the scaling of RIConv++ models to better capture protein structure (original model has only 0.6M) and incorporating non-geometrical features in the architecture and combining it with geometrical information on several scales.

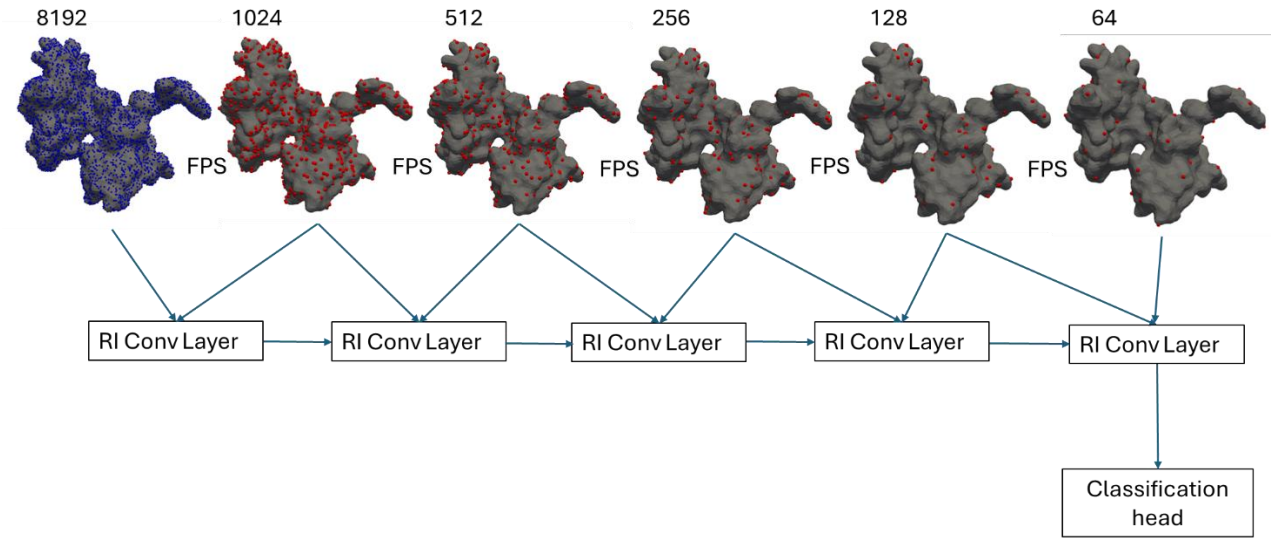


Figure 2: Rotation invariant network used for protein classification with 5 RI Conv layers and 5 subsampling steps on input point cloud.

Models:

Two models are trained/tested:

- RUN 1: Geometry-only model (uses only point cloud as an input),
- RUN 2: Geometry+Potential Model (point cloud + additional features for every point: potential and normal potential).

Validation accuracy:

- RUN 1: 92.93%
- RUN 2: 93.97%

Runtimes for 8192 pc:

	Point cloud extraction	Training	Testing
RUN 1	6 hours 20 min	23 hours	4min 15 seconds
RUN 2	6 hours 20 min	28 hours 15 min	3min 15 seconds

Point cloud extraction for test set: 1 hour and 30 min.

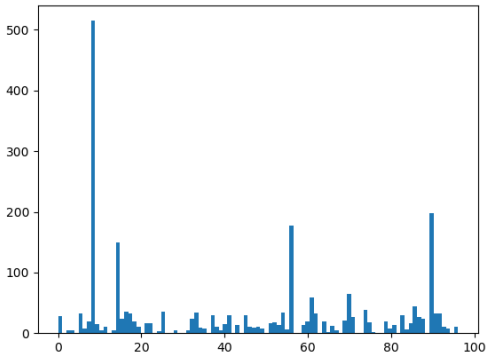
Evaluation:

Submitted files:

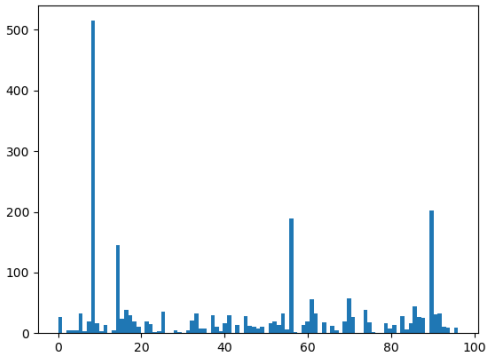
1. test_set_2_run1.csv - predicted class labels
2. test_set_2_run2.csv - predicted class labels

Histograms of predicted classes

RUN 1:



RUN 2:



Test accuracy:

- RUN 1: 91.90%
- RUN 2: 92.76%

All statistics on test set:

RUN 1:
Accuracy: 0.9190,
Precision: 0.8065,
Recall: 0.7825,
F1 Score: 0.7859,

RUN 2:
Accuracy: 0.9276
Precision: 0.8871,
Recall: 0.8472,
F1 Score: 0.8584

Suggestions for improvements:

We have explored both model scaling and point cloud scaling (from 1024 to 8192), see appendix for details. However, due to computational constraints we have not tested whether increasing network depth and adjusting the size of subsampling steps have the effect on the accuracy. Increasing the dataset size and improving class balance (e.g. by reweighting the loss function) are also left unexplored in this challenge.

References:

[ZHY22] ZHANG Z., HUA B.-S., YEUNG S.-K.: Riconv++: Effective rotation invariant convolutions for 3d point clouds deep learning. International Journal of Computer Vision 130, 5 (2022), 1228–1243.

GitHub - cszyzhang/riconv2: The official source codes for the riconv++ (IJCV2022) paper

Appendix:

Details on network configuration

First MLP layer in Fig. 1 has one hidden layer of dimension 32 (3->32->64). Second MLP layer is a one-layer MLP.

Marking “+2” in the following lines distinguishes between RUN 1 and RUN 2 architecture with geometry-only or geometry+potential information. “K” denotes number of nearest neighbours sampled.

RIConv Layer 1: $C = 128,\ C_1 = 0\ (+2),\ C_0 = 64,\ \mathrm{K} = 8$

RIConv Layer 2: $C = 256,\ C_1 = 128\ (+2),\ C_0 = 64,\ \mathrm{K} = 16$

RIConv Layer 3: $C = 512,\ C_1 = 256\ (+2),\ C_0 = 64,\ \mathrm{K} = 32$

RIConv Layer 4: $C = 1024,\ C_1 = 512\ (+2),\ C_0 = 64,\ \mathrm{K} = 64$

RIConv Layer 5: $C = 2048,\ C_1 = 1024\ (+2),\ C_0 = 64,\ \mathrm{K} = 128$

MLP (classification head): 2048 (+2) -> 1024 -> 512 -> 97

Additional experiments with point cloud sizes and models

Runtimes for 2048 pc and 1.5M model:

	Point cloud extraction	Training	Testing	Val acc
RUN 1	3 hours 10 min	9 hours 45 min	1.5-2min	89.58%
RUN 2	3 hours 10 min	9 hours 50 min	1.5-2min	91.71%

Point cloud extraction for test set: 40 min

Validation accuracy for 4096 pc and 1.5M model:

RUN 1: 91.56% RUN 2: 93.58%