[RE] LEARNING FROM PROTEIN STRUCTURE WITH GEOMETRIC VECTOR PERCEPTRONS

Eugene Teoh, Afiqah Kamarudin & Yue Shuen Kew
Electronics and Computer Science
University of Southampton
Southampton, UK
{wct1c16,nabk1g17,ysk2a15}@soton.ac.uk

ABSTRACT

This paper attempts to reproduce baseline experiments on the Geometric Vector Perceptron (GVP), a novel method introduced as a replacement for Multi-layer Perceptrons to improve over existing methods on learning from protein structures. Traditionally, Convolutional Neural Networks have been shown to perform well on geometric tasks, whereas Graph Neural Networks perform well on relational tasks. GVP attempts to enhance the geometric reasoning ability of Graph Neural Networks. The baseline experiments use a synthetic dataset to emulate the behaviour of geometric and relational tasks.

1 INTRODUCTION

The Geometric Vector Perceptron (GVP) is a novel technique that leverages the geometric and relational aspects of a 3D protein structure to solve computational biology problems. The GVP augments graph neural networks (GNNs), replacing the multilayer perceptron layer to improve the geometric reasoning ability of GNNs. While computationally simple, the GVP layers are also equivariant under 3D rotations and reflections, making this method a lightweight alternative to existing equivariant approaches and can be applied to large molecular structures.

In protein structure learning, GVP is the first architecture designed to jointly learn on both relational and geometric representations of 3D macromolecular structures. The empirical evidence presented in the paper shows that GVP outperforms existing architectures, including state-of-the-art (SOTA) methods, in standard benchmark tasks such as model quality assessment (MQA) and computational protein design (CPD). Previous SOTA approaches in these tasks include convolutional neural networks (CNNs) and graph neural networks, which leverages geometric and relational aspects respectively. The GVP was proposed as an improvement over these methods by seeking to combine the strengths of both architectures.

In this reproducibility project, we attempt to reproduce baseline experiments described by Jing et al. (2021). The original paper used simple synthetic tasks to demonstrate the capability of GVP augmented GNN (GVP-GNN for short) in dual relational and geometric learning, and its performance was compared against CNNs and GNNs. Additionally, the important equivariance property of GVP layers was tested and the pre-trained model provided by the authors was tested for robustness on a new dataset.

2 EXPERIMENTAL DESIGN

Our reproducibility project focused on proving and explaining the theoretical concepts. To reproduce the original paper, we examined the OpenReview reviews on the paper and used this as the basis for our reproducibility project. Important evaluations from the reviewers were the novelty of the method and its conceptual contribution, the importance of its equivariance properties, the high performance on difficult tasks, and that it is broadly applicable.

Accordingly, we determined that this project could be split into four tasks: (a) proving important equivariant properties claimed in the paper, (b) proving the conceptual contribution claimed with the synthetic dataset used in the paper, (c) reproducing the original results on a specific notable problem using real datasets that were used in the paper, and (d) testing the model on another notable problem in the domain by finding new datasets that were not used in the paper. Due to lack of domain knowledge, and time and computation constraints, tasks (a) and (b) were decided as the main objectives of this reproducibility project.

The first claim of Jing et al. (2021) was that the vector and scalar outputs of the GVP are equivariant and invariant, respectively, under rotations and reflections in 3D Euclidean space. This is the desired property that would make the GVP generalise better, improve model accuracy, and reduce overfitting, and hence we decided to test this property. The second claim was that using GVP to augment graph neural networks improved the geometric reasoning abilities of GNNs. For this, we attempted to reproduce the synthetic task, examining the methodology and comparing the obtained results to the original paper's. We compared the performance of convolutional neural networks, graph neural networks, and GVP-GNN on the synthetic "off-center", "perimeter" and "combined" tasks, evaluating the models' geometric, relational and joint geometric-relational reasoning abilities respectively.

As our secondary goals were not feasible, in lieu of it, we decided to investigate the performance of the original authors' pre-trained model on a new dataset. For the CPD task, the authors provided a model trained on the CATH4.2 dataset. We found that a more recent dataset, CATH4.3, introduced new protein structures and investigated whether the pre-trained model was able to generalise to the new data with reasonable accuracy. Parts of the author's code were used in this project. The synthetic tests were written by us, however, we used the author's PyTorch GVP implementation and code for the equivariance test.

3 Results and discussion

In this section, we present the results of the equivariance theorem test, the comparison between three different models in the synthetic task, and the performance of the original authors' pretrained model on the CATH4.3 dataset.

In preliminary investigations, we attempted to retrain a CPD model on the CATH4.2 dataset, using the author's original code. The code was able to run on the university's Yann server, however each epoch took approximately 3 hours and 40 minutes on average. The original authors trained their model over 2 days for 100 max epochs using a single Titan X GPU. From this, we determined that training the CPD and MQA models from scratch required more time and computational resources than we had available.

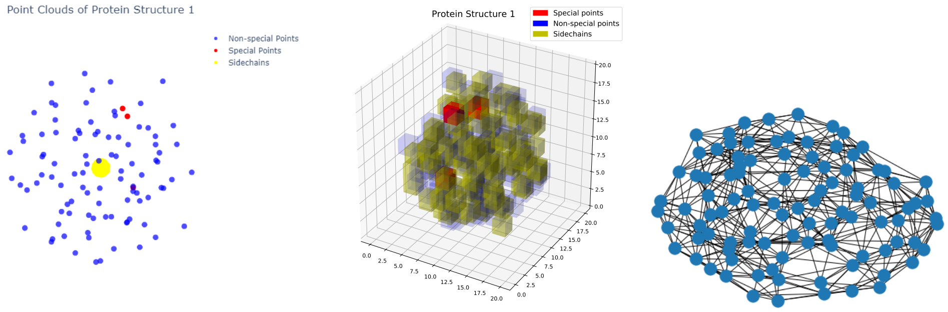
3.1 Equivariance and Invariance test

GVP proposed by the authors claimed equivariance and invariance properties of vector and scalar outputs of GVP respectively with respect to rotations and reflections, R. The transformation is described as $R = V \times U$, where $U \in \mathbb{R}^{3x3}$ is a unitary matrix. A unitary matrix is orthogonal. To ensure this condition, QR decomposition is used to compute R. The orthogonality property allows isometry of Euclidean space, proving vector equivariance. A test function is used to validate this relationship. Verifying the difference between transformed and non-transformed vector outputs of GVP, both vector output values are different but its magnitude is the same.

Scalar output of GVP is claimed to be invariant. This is verified by confirming that both the scalar outputs of GVP on the same data with or without rotation are the same. The difference between both scalar outputs was 2.86×10^{-6}, proving its invariant property.

3.2 Synthetic Task

The synthetic dataset was obtained from the GoogleDrive folder linked in the 24 Nov 2020 revision of the paper. The dataset consists of 20,000 data samples that mimic the qualities of protein structures. Each sample is a point cloud consisting of 100 points in $\mathbb{R}^3$, distributed uniformly in a sphere, and each point is associated with a "sidechain" unit vector which determines its orientation. The three first points of each sample are marked as special points. Figure 1 shows that each sample was transformed into a voxelised representation for the CNN model, and a 10 nearest neighbour graph representation for the GNN and GVP-GNN models.



(a) Original structure (b) Voxelised representation (c) Graph representation

Figure 1: Synthetic dataset, in (a) the original form, (b) the form used for CNN, (c) the form used for GNN and GVP-GNN

All three model architectures used in the paper was described as ”shallow 3-layer” models. From this, we approximated the CNN model as such: three convolutional layers with (3,3,3) kernels and ReLU activation, an adaptive max pooling layer, and a fully-connected linear layer for the regression output. The resulting model had 58k parameters. The GVP-GNN architecture includes: one linear layer each to transform node and edge dimensions to the stated hidden dimensions, three GVP message passing layers (described in Jing et al. (2021)), one GVP layer to with zero output vector dimension, and finally a one-layer feed-forward network with dropout. Layer Normalisation with affine transform parameters are also applied before every GVP layer. The GNN architecture is identical to GVP-GNN but replaces all GVP layers with linear layers.

Emulating the original paper, the models had the same intermediate dimensionality of 32 channels (GVP-GNN has 20 scalar and 4 vector channels). Some details not described in the paper were: max epoch of 100, batch size of 64, and Adam optimizer with default learning rate of 0.001. The same training procedure was used for all models. The Pytorch Lightning framework was used to ensure a streamlined process, and no architecture or hyperparameter tuning was performed. Table 1 presents the MSE losses for the synthetic tasks. The paper claims that CNN should perform well on the geometric tasks, GNN should perform well on relational tasks. GVP-GNN combines the strength both CNN and GNN, as such, it should perform relatively well on both geometric and relational, and excel at the combined task. Our results do not reflect the findings of the original paper. Additionally, GVP-GNN performed worse on geometric and relational tasks, and on par with CNN on the combined task.

Table 1: Performance of the three compared model architectures on the geometric, relational and joint geometric-relational tasks.

Model	Parameters	Off-center (geometric)	Perimeter (relational)	Combined
CNN	58k	**1.462**	**74.018**	0.0246
GNN	44k	2.956	74.482	**0.0192**
GVP-GNN	23k	3.542	193.444	0.0242

We encountered some difficulties while reproducing this task due to the ambiguous descriptions of the methodology. Firstly, the model architectures were not described in detail. Certain parts had to be estimated from the number of trainable parameters listed in the paper, hence our resulting models were not an exact reproduction of the originals. Secondly, Jing et al. (2021) stated that, ”The MSE losses are standardized such that predicting a constant value (i.e the mean) would result in unit loss”. No context was provided as to how or why this was done, and hence we decided to forgo this step in our reproduction.

The synthetic dataset was not published nor documented rigorously. Neither the experiment nor the dataset for the synthetic tasks were published in the author’s GitHub. We only found the synthetic dataset in previous revisions of the paper, rather than the final revision. The linked GoogleDrive folder also contained outdated data, which they had not updated despite changing one of the tasks from classification to regression. We only received the correct targets after contacting the author. Some parts of the dataset was unclear as well: the paper described the special points as ”randomly chosen and labelled as special”, but the author informed us that the first three points were the chosen special points.

Although Jing et al. (2021) outlined the main experiments in sufficient detail and presented strong empirical evidence, the baseline experiment was reported less rigorously. We were unable to reproduce the same exceptional performance of the GVP-GNN over the CNN and GNN that was showed in the original paper. The baseline experiment described in the paper was not easily reproducible due to ambiguities in methodology.

3.3 PRETRAINED MODEL

This part of the experiment considered only the Computational Protein Design (CPD) task. As claimed by the authors, GVP-GNN model achieved state-of-the-art performance on CATH 4.2. This motivated the investigation of performance of the model on new datasets, to verify whether such outstanding performance resulted from the strength of the architecture or hyperparameter tuning on the dataset used.

Two pretrained models on CATH 4.2 and TS50 datasets were provided by the authors[1]. The pretrained model on CATH 4.2 were used to evaluate on a new dataset – CATH 4.3. 65,984 new protein structures were introduced in the newer version and 603 protein strutures were removed. The model was tested on strictly new protein structures to evaluate its performance. The dataset was curated similarly to that of CATH 4.2 by Ingraham et al. (2019), whereby a protein structure was removed if the length of its protein sequence was more than 500.

Table 2 summarised the difference in performance on both datasets. Interestingly, CATH 4.3 evidenced in a slightly lower perplexity score eventhough protein structures tested on were dissimilar to the ones that the model were trained on. This may be reasoned by the amino acids that makes up a protein sequence are the same, though with different orders for different protein structures.

Table 2: Pretrained CATH model performance on CATH 4.2 and CATH 4.3.

Dataset	**Perplexity**	**Recovery**
CATH 4.2	5.293	0.402
CATH 4.3	4.799	0.413

4 CONCLUSION

In our attempt at reproducing experiments described by Jing et al. (2021) to demonstrate important conceptual contributions of the geometric vector perceptron, we found that the baseline experiments were not reported as rigorously as the benchmark experiments on MQA and CPD. This led to difficulties in producing an exact reproduction of the experiments, and our obtained results also differed from the original. However, the author was amenable to our efforts and provided some help in clarifying crucial parts of the methodology. Additionally, deep learning for the domain of computational biology is not as streamlined and well-documented as other more popular domains such as computer vision, which made this reproducibility project more challenging.

REFERENCES

John Ingraham, Vikas K Garg, Regina Barzilay, and Tommi Jaakkola. Generative models for graph-based protein design. In *Advances in Neural Information Processing Systems*, 2019.

Bowen Jing, Stephan Eismann, Patricia Suriana, Raphael John Lamarre Townshend, and Ron Dror. Learning from protein structure with geometric vector perceptrons. In *International Conference on Learning Representations*, 2021.

[1] Pretrained Models: `https://github.com/drorlab/gvp/tree/master/models`