Predicting Protein-Ligand Binding Using Deep Learning with Spatial Transformations

David Ban

North Allegheny Senior High School
10375 Perry Highway, Wexford PA, 15090

Mentor: Dr. David R. Koes
School of Medicine
University of Pittsburgh

Acknowledgements

I would like to thank Professor David R. Koes for his support and guidance, and the computer facilities at the University of Pittsburgh.

Table of Contents

1 Background

With the hundreds of different harmful viruses, bacteria, and life threatening diseases, there is a pressing need for new drugs. However, biopharmaceutical research and development is a long and laborious process. Out of the tens of thousands of initial compounds screened, only a small fraction of compounds are found to bind with the target protein, and only few get to the stage of clinical trials. Lately, computer simulation and artificial intelligence are beginning to make impact on biomedical research including drug discovery aiming to reduce the time and effort needed for bringing new drugs to market [1].

1.1 Drug Discovery

The current drug discovery process takes an average of 10-15 years and can cost up to 2.6 billion dollars to develop one new prescription drug from thousands of compounds. This is largely due to the number of initial early failures in the drug discovery process, as well as later issues of safety, efficacy, and selectivity [2]. This imposes the need to improve the efficacy and efficiency of drug discovery by pre-screening drug molecules and accurately selecting promising candidates for further testing and clinical trials.

1.2 Protein-Ligand Binding

Because proteins are responsible for many of the body functions and diseases, almost all drugs on the market act on proteins: over forty percent of the drugs act on receptors, nearly thirty percent on enzymes and fifteen percent on transporter proteins [3]. For example, one of the top selling prescriptions drugs, Rivaroxaban (brand name Xarelto) binds to Factor Xa, an enzyme of the coagulation cascade, to inhibit thrombin formation, treat and prevent blood clots [4] (Fig. 1).Thus, it is imperative to understand the binding of ligands to proteins during the process of finding new drugs.

Figure 1: Example protein-ligand binding conformation: Binding of a Rivaroxaban (Xarelto) ligand with Factor Xa protein, an important interaction that prevents blood clots. [5]

Proteins are large complex molecules that are essential to all life, whether it is providing mechanical support as structural protein, catalyzing chemical reactions as enzymes or communicating between cells as receptors or signaling factors. They bind to small molecules, nucleic acid or other proteins, and their unique sequences and 3D structures are highly deterministic of their functions. Ligands are small molecules or drugs that are capable of binding to their target proteins, which leads to a reshaping of the protein's structure and alters their functions, such as deactivating a diseased protein and stimulating or restoring certain protein activities [6].

The drug discovery procedure typically begins by identifying the target protein implicated in diseases followed by determination of compounds that bind to the protein with maximal wanted effect and minimal side effect [7]. However, because protein-ligand binding is highly dependent on the complex structure and orientation of both molecules and the intermolecular forces between them, it is nearly impossible to predict if a ligand and protein will bind, and is even more difficult to predict the binding sites. Although computational screening for novel therapeutic drugs has been widely employed in modern drug discovery, because of the lack of efficient tools to pre-screen drug molecules for promising candidates, most pharmaceutical companies still perform large amounts of laboratory testing which costs millions of dollars for developing one drug [2].

1.3 Computer Aided Drug Discovery (CADD)

Computer-aided drug discovery (CADD), for over 20 years, has played a vital role in drug discovery and development and has become an indispensable tool in the pharmaceutical industry, using different softwares to virtually screen, discover and optimize biological compounds [8]. When the structure of a target protein is available, CADD can perform high-throughput screening for large numbers of compounds [6]. CADD screening is able to significantly reduce the number of prominent candidate compounds before clinical trials, thus decreasing the time, cost, and number of compounds that must be synthesized and tested *in vitro*.

Although CADD has made a significant improvement for drug discovery by selecting a small number of prominent candidate compounds before clinical trials, it requires knowledge of the structural and activity relationships for the proteins and ligands, and, in general, is very computationally intensive,even for small time scales. It usually takes supercomputers to simulate even nanoseconds of real time, which makes it extremely dif[U+FB01]cult to determine drug binding ef[U+FB01]cacy [9].

1.4 Deep Learning Neural Networks

Deep learning has been widely used to develop artificial intelligence. It can learn and obtain high accuracy of identification from large amounts of data which contains multiple hidden layers of nonlinear progressing units for learning data representations providing powerful handling of big and complex data [10].

One of the most popular neural networks are Convolutional Neural Networks (CNNs). CNNs are particularly applicable to drug discovery because of its ability for image recognition, and can learn spatial relationships in data by learning feature maps. Each additional layer allows the network to learn higher level features. For example, the first layer might learn to identify lines and edges, while a later layer could identify shapes. The final layers will then be able to recognize high-level and complex features such as distinctions between animals, as shown in Fig. 2 [11]. This can be especially applicable to drug discovery as CNNs may be able to identify binding positions giving the image of the ligand and protein. Unlike previous methods of CADD, this does not require an underlying relationship and a pre-programmed understanding, and instead it learns the underlying relationships as long as there is sufficient training data [12].

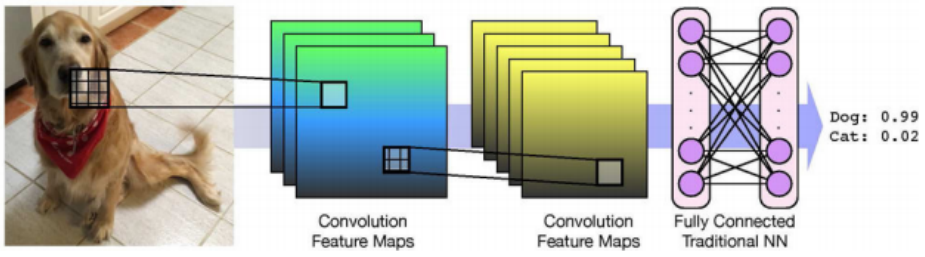

Figure 2: A CNN used for image recognition.[13]

.

The impressive performance of CNNs at image recognition suggest that they are well-suited for other types of data dependent on spatial relationships, such as protein−ligand structures. CNNs have previously been successfully trained on a coordinate grid to learn scoring functions for affinity prediction and pose discrimination [11] [14], while other studies have also trained CNNs to predict binding affinities or quantum mechanical energies and forces between atoms [15].

Although CNNs are extremely powerful, they are highly susceptible to errors caused by translation and rotation of the input. Because of the importance of the 3D structures and orientations of proteins and ligands for their proper interactions, it is critical to develop a tool that takes into account of these features when selecting compounds that can correctly bind to the protein of interest. One possible solution is to

use spatial transformer networks (STNs).

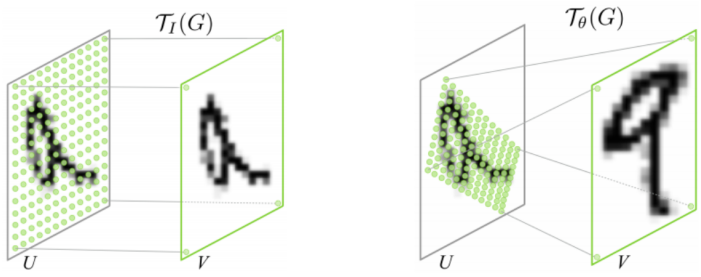

Figure 3: A STN modifying the MNIST handwriting dataset [16]

.

STNs are CNNs that have at least one spatial transformer module, which allows for the spatial manipulation of data within the network. This module can be inserted into existing CNN architectures, giving neural networks the ability to spatially transform inputs to an "ideal" version of the input as illustrated in figure 3. This allows networks to recognize the important structures in the input, even if the input is spatially transformed, and allows the model to generalize to a larger variety of inputs [16]. Despite the potential advantages of STNs in dealing with three-dimensional objects, the application of STNs for protein-ligand binding has yet to be attempted.

2 Research Goal

The goal of this research is to investigate if STNs can predict protein-ligand binding. In this project, the aim was to build a model leveraging STNs to predict the binding conformation of protein-ligand complexes to be used for computational drug screening. Specifically, it will determine if STNs are able to accurately reconstruct the original binding position after applying rigid-body transformations to known protein-ligand complexes. If proven to be successful, this technique can be further developed as an effective and efficient tool for predicting protein-ligand binding.

3 Methods

Deep learning STNs were created with the use of the PyTorch and Caffe backbone for the neural network and utilizing Gnina [13] to aid the protein-ligand processing. Various novel programs were written for the purpose of creating the network, analysis of data, and network optimization.

3.1 Models Using PyTorch and Caffe Framework

STN models were built using both PyTorch and Caffes frameworks against the same rigid-body perturbations. Both frameworks are open sourced and are free to use to the general public along with being very flexible in their applications. Caffe uses C++ with a Python interface and was developed in the University of California, Berkeley [17]. PyTorch is a machine learning library that is solely written in Python and is primarily developed by Facebook [18]. Gnina is a platform for researching structure-based deep learning developed by David R. Koes [13], and was used as a basis for each model.

3.2 Training and Testing Data

Models were trained and tested with the PDBbind refined set [19] which contains the binding structure of 4,463 known protein-ligand complexes, with no unusual features such as uncommon elements. The refined set is a subset of the PDBbind database that is representative of the entire set which consists of over 16,000 protein-ligand complexes.Training was performed on a randomly-sampled 80% of the refined set and cross-validated on the remaining 20%.

3.3 Ligand Translation and Rotation

After the known binded protein-ligand pair is mapped onto the grid, the ligand is moving away from it's target protein, and randomly perturbed by applying rigid-body translations (x, y, z) or rotations (roll, pitch, yaw) in both sine and cosine directions (Figure 4). These perturbations are to mimic the case of target proteins and ligands with unknown binding conformations. While these perturbations could be applied to the protein or to the entire structure, this project focused on perturbing the ligand only.

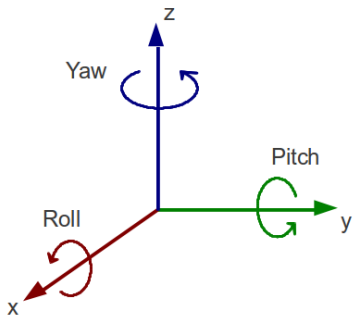

Figure 4: Visualization of rotation (roll, pitch, yaw) and translation (x, y, z) in the Cartesian plane

3.4 Data Input Format

After the ligand was moved away from the binding position of target protein by translation or rotation, the protein-ligand complex was voxelized. CNNs typically take 2-dimensional pixel images composed of red, green, and blue channels as inputs, and in order to feed atomic structures to CNNs, each atom of the protein-ligand complex was“voxel-ized” into a 48x48x48 voxel plane, with each voxel measuring 0.5 cubic Å. A visualization of the atomic representation is shown in figure 5. This allows the input data to fully express the spatial features of the protein-ligand complex while making the structure less complex.

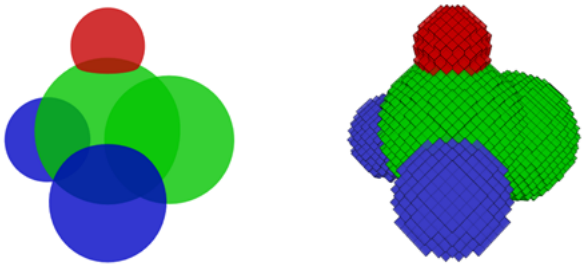

Figure 5: Molecule (left) and its "voxel-ized" representation (right)

3.5 Deep Learning Model Architecture

The model architecture was built with varying fundamental network structures, such as varying layers and hyperparameters, to gauge efficiencies of different models. However, each one consisted of a spatial transformer module preceding several 3-D convolution layers, max-pooling layers which follow the standard CNN architecture where each layer learns higher order representations of the data as spatial dimensionality decreases. After the convolutional layers, there are several fully connected layers with nine outputs. Six of which relate to rotational loss and three of which relate to translational loss. The CNN models were defined and trained in both Caffe and PyTorch [17] [18], and were trained by iteratively updating its weights using standard backpropagation [20].

3.6 Hyper-parameters

Hyperparameters are variables which determine different qualities of the network, such as how fast the learning rate is, or how many layers there are. They can be tuned before applying a learning algorithm to a dataset and are important because they directly control the behavior of the training algorithm. The parameters can then be modified in order to achieve the best result.

Model Depth: "Depth" refers to the number of convolutional layers within each model, and varies with

each model. The initial model contained 3 convolution layers, and models with more layers were also evaluated. This theoretically allowed for a more expressive model, but became much more computational expensive and increases the risk of suffering from vanishing gradients.

Pooling Type: Pooling layers reduce the size of their inputs by propagating a single value for each window (or kernel) of the input. The propagated value can either be the maximum value or the average value of the kernel and the kernel size can be varied. In each model, max pooling was used with a kernel size of 2.

Fully Connected Layer: After a series of convolution and pooling layers, fully connected layers reduces the final feature maps to nine outputs - 6 rotational outputs, three for sine and three for cosine, and 3 translational outputs. Each model run contained three fully connected layers.

Output: After being run through the CNN, the network would then predict the amount the ligand was perturbed by. To get smooth gradients for backpropagation, angular loss was instead replaced by sine and cosine loss. Thus, for each model, nine loss outputs were produced: three Euclidean losses in x, y, z, dimensions, and six sine and cosine losses for roll, pitch, and yaw. Euclidean loss is a type of loss which relates to how poorly the the network is doing, and is defined as $\frac{1}{2N}\sum_{i=1}^{N}\|x_i^1 - x_i^2\|_2^2$.

3.7 Metrics for Model Evaluation

Since the output of each model is a binding conformation, the results of the model will be reported as Pearson R coefficients, representing the correlation between the predicted conformation and the ground truth conformation, and Euclidean loss. Pearson R value, also referred to as Pearson correlation coefficient, is a statistical measure of the linear correlation between two variables - in this case, the predicted conformation versus the true conformation. The range of values is $[-1, 1]$, with -1 showing a total negative relationship, 0 showing no relationship, and 1 showing a total positive relationship. Figure 6 summarizes the process of the data input and output through STN models.

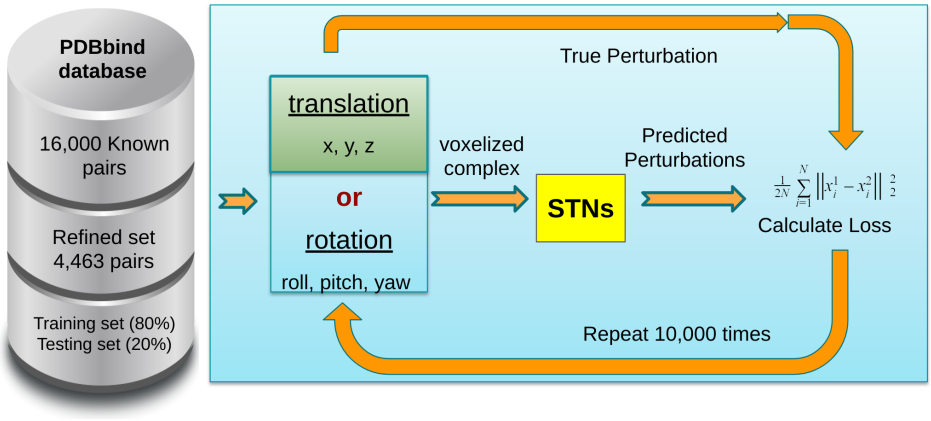

Figure 6: Overarching methods for each model. Initial inputs were taken from the PDBBind refined set and the ligand was translated or rotated away from the correct position. The entire complex would then be voxelized, run through the STN, and a predicted perturbation would be outputted. Loss would then finally be calculated based off the true perturbations, and the process would repeat 10,000 times

4 Results

Using the data from the PDBbind refined set with systemic optimization of the training parameters, STNs built in either PyTorch or Caffe were able to predict the correct binding positions following rigid body protein-ligand perturbations.

4.1 Convergence

Convergence is an important criterion for determining the success of a machine learning model and generally describes the process of a sequence or series of iterations approaching a limit. Ideally, the network should converge to zero Euclidean loss, which would be a perfect learning model with an error of 0 and a Pearson R coefficient of 1. In reality, the loss is rarely zero and the Pearson R coefficient is hardly 1.

Figure 7 shows the convergence for both PyTorch and Caffe models following translational perturbations. In PyTorch, the loss decreases from 1.5 to 0.3 after 10,000 iterations (Fig. 7, left). The learning rate, as indicated by the rate of decrease in loss, slows down after 6,000 iterations, and shows possibility of smaller errors with more than 10,000 iterations. In the Caffe model, the model converges at approximately 4,000 iterations with a euclidean loss changing from 2.0 of 0.2 (Fig. 7, right).

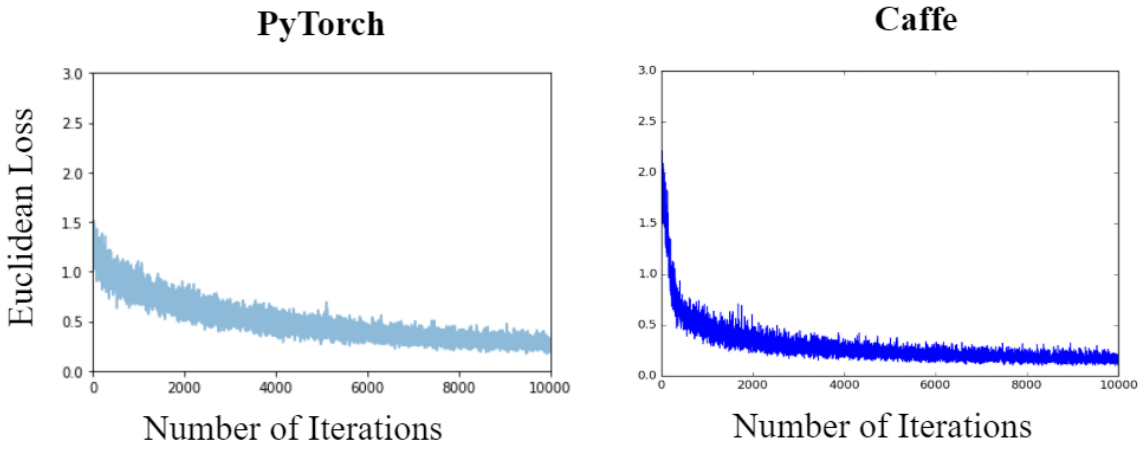

Figure 7: Convergence of loss functions for translation in PyTorch and Caffe.

Convergence for PyTorch and Caffe models following rotational perturbations are illustrated in Figure 8. The loss for PyTorch model changed from 0.8 to about 0.3 after 10,000 iterations, and seemed to be still decreasing at the end (Fig. 8, left). For the Caffe model, the loss dropped from 1.2 sharply to about 0.6 after 1000 iterations, then slowed down after 2000 iteration and reached to about 0.4 at the end of 10, 000 iterations (Fig. 8, right).

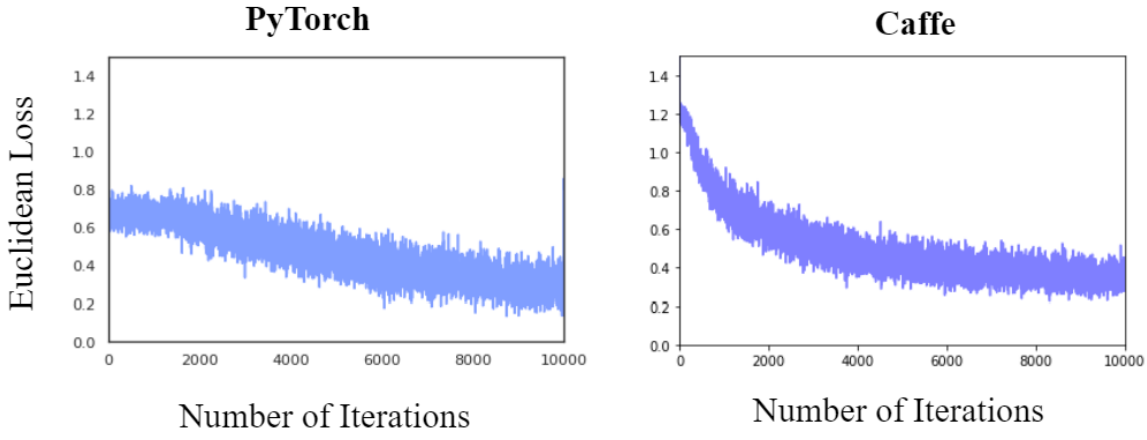

Figure 8: Convergence of loss functions for rotation in PyTorch and Caffe.

These results demonstrate that STNs in both PyTorch and Caffe models are able to learn the relationships from the training data and put the ligands back to their correct positions following translational or rotational perturbations with much smaller errors at the end of the test. Both models converge, which is evident by the reduction of loss initially and then asymptotically approaching a limit. The Caffe model shows a much faster learning rate than the PyTorch model, but the models are not perfect and there is still room to improve.

4.2 Prediction Accuracy: Translation

Error is calculated for the prediction of ligand binding position after translations of the ligand by finding the absolute difference between the prediction and true perturbations. As the ligand is translated in x, y, z directions, the error was measured for each iteration. Error for an ideal model would be zero for all iterations (with it only having one bar with an error of 0). Models that are learning would have a higher frequency of lower error when compared to the random (as shown in Figure 9. If the network was not learning and randomly guessing, the model would follow the blue trend for the error in figure 9 (dark blue for Caffe on the right and light blue for PyTorch on the left). These results demonstrate that the models are capable of learning as there is an inherent bias towards the lower ranges of error in all three translational directions (x, y and z) in both PyTorch (tan histogram) and Caffe (green histogram).

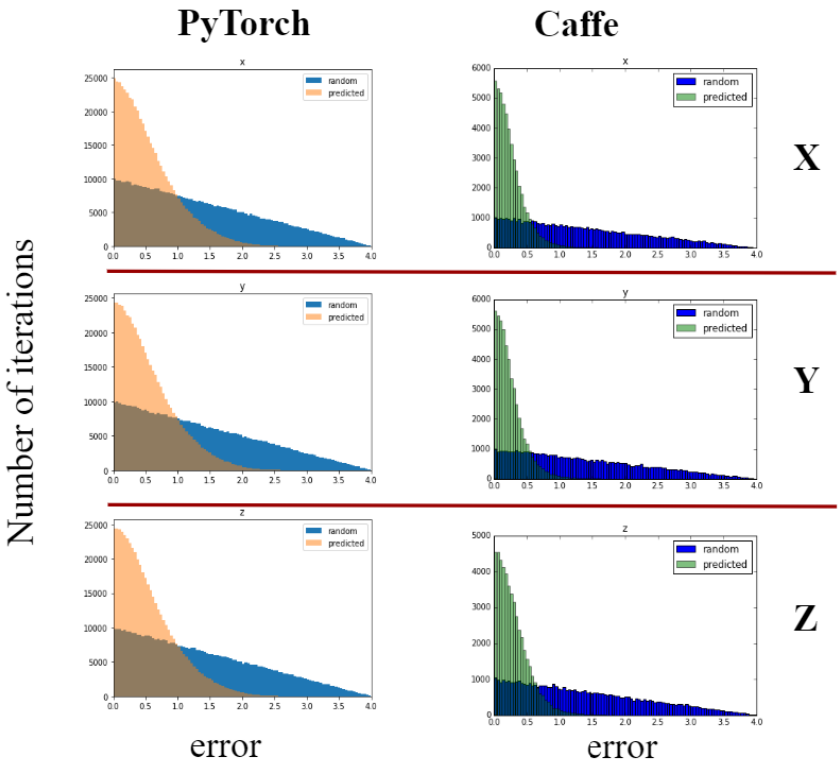

Figure 9: Errors in trained PyTorch and Caffe models following translational perturbations (green in Caffe, yellow for PyTorch) versus random guessing (dark blue in Caffe, light blue in PyTorch).

In order to determine the accuracy of prediction, the predicted value was correlated to the true value and Pearson R coefficients were measured. Fig. 10 demonstrates the learning capabilities for both the PyTorch and Caffe translational models. For PyTorch, with a Pearson R coefficient of 0.85 for the predicted value versus true value and in all three directions, it is clear that the model is capable of predicting configurations very close to the true binding configuration. Caffe models showed similar

results with slightly higher Pearson R coefficient of 0.96.

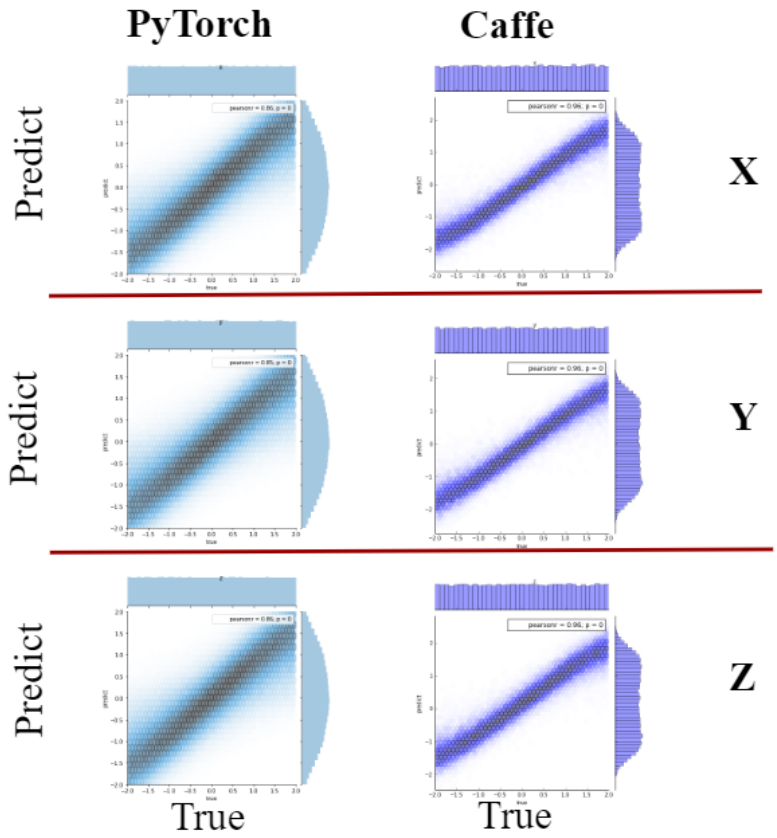

Figure 10: Correlations between predicted and true values for translational models in PyTorch and Caffe.

4.3 Prediction Accuracy: Rotation

As the ligand is rotated in roll, pitch and yaw directions, the sine and cosine of angular error compared to the true rotation was measured for each iteration. Figure 11 shows the error in relation with the number of iterations for PyTorch (left) and Caffe (right) models following roll, pitch and yaw rotation in the cosine direction. If the network was not learning, the error trend would follow the light blue (PyTorch) or dark blue (Caffe) histogram. In contrast, both models have a higher number of iterations with very low error as illustrated in the tan and green histograms which indicates that the models are learning. Both STNs showed better performance when compared to random. Similar results were found in all three rotational axis in the sine graphs (data not shown).

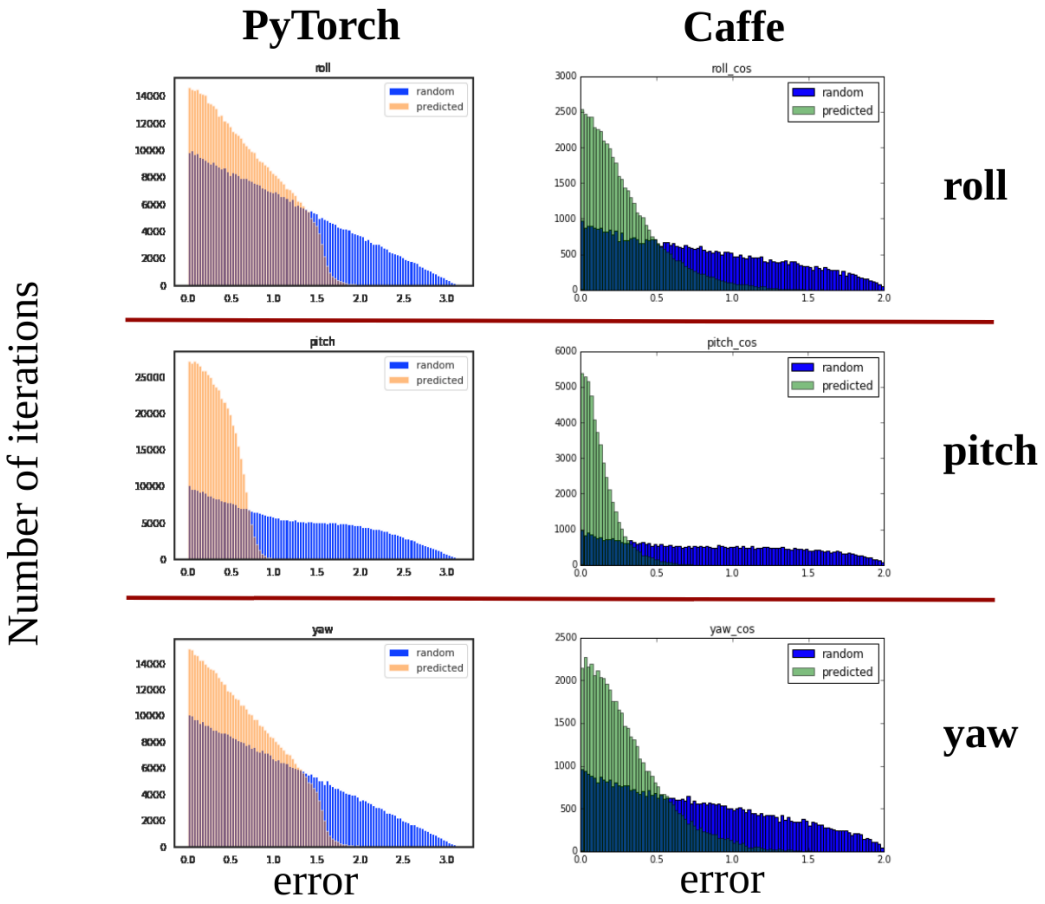

Figure 11: Errors in trained PyTorch and Caffe models (tan or green) versus random guessing (light and dark blue) following rotational perturbation for cosine prediction.

Correlations between predicted and true values for rotation models in PyTorch and Caffe were also determined. Although there are good correlations for the Caffe rotation model, with Pearson R values of 0.68 - 0.89, the PyTorch model only showed marginal correlations (Pearson R values 0.25-0.52).

5 Discussion

The STNs deep learning computational tools developed in this study to predict protein-ligand binding was synthesized through the PyTorch [18] or Caffe [17] framework implementing the Gnina [13] framework for molecular docking. Both PyTorch and Caffe models are able to learn the binding conformations of the protein-ligand pairs from the training dataset and converge following ligand rigid-body transformations in the three translational directions (x, y and z) and three rotational perturbations (roll, pitch and yaw). They also have the ability to predict the binding positions for unseen ligands to their target proteins after the ligand is moved away from its binding protein.

Although both PyTorch and Caffe converge after translational perturbations, the Caffe model has a much steeper drop in loss during the first 1000 iterations compared to PyTorch model, and has smaller error, higher Pearson R values than PyTorch model, indicating a faster learning and more accurate prediction for Caffe than PyTorch.

For rotational perturbations, both PyTorch and Caffe are also capable of learning from the known protein-ligand pairs. Similar to translation perturbations, Caffe model has a steeper decrease in loss in the beginning of run and smaller error/higher Pearson R values than that of PyTorch model. Caffe maybe a quicker learning and more accurate model to predict ligand binding to proteins compared to PyTorch.

However, Caffe model took much longer to run compared to PyTorch, 12 hours vs 2 hours for Caffe and PyTorch, respectively. This may be a tradeoff between accuracy versus speed.

When compared to translation, rotation converged at a much slower rate in both frameworks, which may indicate that the learning of ligand orientation is a more difficult problem than determining binding location of a correctly oriented ligand. Also note that Pearson R coefficients for rotational models in Caffe in the roll (0.68) and yaw (0.68) directions should have been greater because of the clusters that appear at the top left and bottom right corners of the graphs (see Fig. 12). These resulted from the inability of the model to distinguish between $+\pi$ and $-\pi$ which decreases the coefficient value.

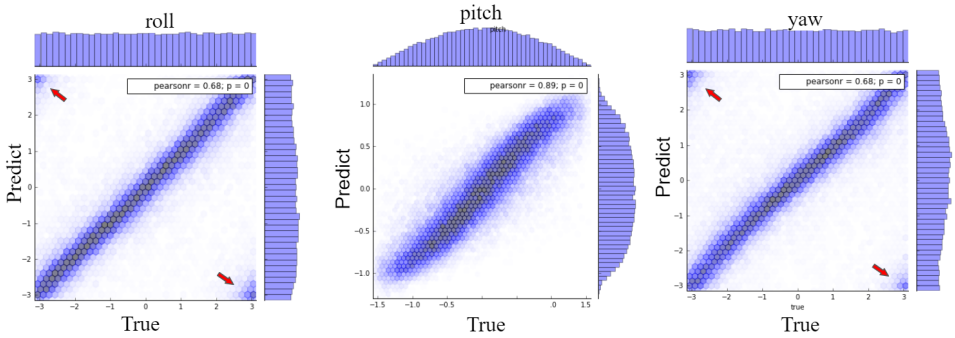

Figure 12: Correlation between predicted and true values for cosine rotation in roll, pitch and yaw direction in Caffe model. (arrows indicate clusters)

As an initial study of this kind, these results show that STNs are a promising tool for accurately predicting the conformations of a protein-ligand complex. Further modifications of the networks and examination of coding will be done in order to improve the capability and accuracy of protein-ligand binding prediction in both frameworks, especially for PyTorch model and the rotational ligand transformations. More testing will need to be done to expand the model to translations and rotations of the entire complex.

The model may also be improved through incorporating newer methods, such as dilated convolutional layers that allow the model to learn features that span more of the input space. It might also be worth exploring non-rigid transformations, which will increase the complexity of the problem, but may allow the model to generalize better for real-world applications.

6 Conclusion

This is the first study that uses STNs to predict protein-ligand binding in drug discovery. The network developed is able to predict the true binding sites of randomly translated and rotated ligands in both PyTorch and Caffe frameworks. The Caffe model demonstrates as a faster learning and more accurate model, whereas PyTorch as a more time-efficient model. Although further testing and more in-depth studies are needed before these models can be applied to pharmaceuticals, these results open a new direction (possibility) of developing computational tools for drug discovery and pave the way of using deep learning STNs to screen drug candidates before further experimental testing and clinical trials, accelerating the process of drug discovery.

References

[1] Michael Wainberg, Daniele Merico, Andrew Delong, and Brendan J Frey. Deep learning in biomedicine. *Nature Biotechnology*, 36(9):829–838, October 2018.

[2] Joseph A. DiMasi, Henry G. Grabowski, and Ronald W. Hansen. Innovation in the pharmaceutical industry: New estimates of R&D costs. *Journal of Health Economics*, 47:20–33, 2016.

[3] Mathias Rask-Andersen, Markus Sällman Almén, and Helgi B. Schiöth. Trends in the exploitation of novel drug targets. *Nature Reviews Drug Discovery*, 10:579–590, Aug 2011.

[4] Sosipatros Bratsos. Pharmacokinetic properties of rivaroxaban in healthy human subjects. *Cureus*, August 2019.

[5] S. Roehrig, A. Straub, J. Pohlmann, T. Lampe, J. Pernerstorfer, K. Schlemmer, P. Reinemer, E. Perzborn, and M. Schaefer. Factor xa in complex with BAY59-7939, November 2008.

[6] William R. Pitt, Mark D. Calmiano, Boris Kroeplien, Richard D. Taylor, James P. Turner, and Michael A. King. Structure-based virtual screening for novel ligands. In *Protein-Ligand Interactions*, pages 501–519. Humana Press, 2013.

[7] Jeff Settleman and Robert L. Cohen. Communication in drug development: "translating" scientific discovery. *Cell*, 164(6):1101–1104, Mar 2016.

[8] Douglas B. Kitchen. Computer-aided drug discovery research at a global contract research organization. *Journal of Computer-Aided Molecular Design*, 31(3):309–318, November 2016.

[9] A.R. Leach. 4.05 - ligand-based approaches: Core molecular modeling. In John B. Taylor and David J. Triggle, editors, *Comprehensive Medicinal Chemistry II*, pages 87 – 118. Elsevier, Oxford, 2007.

[10] Hongming Chen, Ola Engkvist, Yinhai Wang, Marcus Olivecrona, and Thomas Blaschke. The rise of deep learning in drug discovery. *Drug discovery today*, 23 6:1241–1250, 2018.

[11] David K Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alan Aspuru-Guzik, and Ryan P Adams. Convolutional networks on graphs for learning molecular fingerprints. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 2224–2232. 2015.

[12] Kevin Gurney. *An Introduction to Neural Networks*. Taylor & Francis, Inc., Bristol, PA, USA, 1997.

[13] Matthew Ragoza, Joshua Hochuli, Elisa Idrobo, Jocelyn Sunseri, and David Ryan Koes. Protein-ligand scoring with convolutional neural networks. *Journal of Chemical Information and Modeling*, 57:942–957, 2017.

[14] Jocelyn Sunseri, Jonathan E. King, Paul G. Francoeur, and David Ryan Koes. Convolutional neural network scoring and minimization in the d3r 2017 community challenge. *Journal of Computer-Aided Molecular Design*, 33(1):19–34, July 2018.

[15] Giuseppe Carleo and Matthias Troyer. Solving the quantum many-body problem with artificial neural networks. *Science*, 355(6325):602–606, 2017.

[16] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 2017–2025. 2015.

[17] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[18] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPS-W*, 2017.

[19] Renxiao Wang, Xueliang Fang, Yipin Lu, and Shaomeng Wang. The pdbbind database: Collection of binding affinities for protein-ligand complexes with known three-dimensional structures. *Journal of Medicinal Chemistry*, 47(12):2977–2980, Jun 2004.

[20] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.