
Re-analyzing Scoring Function Convolutional Neural Networks

Chuyan Zhou, Hanze Li, Xinyang Xu

School of Information Science and Technology

ShanghaiTech University

{zhouchy2022, lih2024, xuxy2024}@shanghaitech.edu.cn

Abstract

Protein-ligand binding affinity prediction is fundamental to drug discovery and computational biology. This study presents a comprehensive re-analysis of Scoring Function Convolutional Neural Networks (SFCNN) for predicting protein-ligand binding affinities, with particular emphasis on integration with AlphaFold3 predicted structures. We implement and evaluate SFCNN using 4-dimensional one-hot encoded voxel representations on the PDBbind v2019 refined dataset and CASF-2016 benchmark. Our analysis extends to predicted structures from Chai-1, enabling evaluation of model performance on computationally generated protein-ligand complexes. Results reproduce strong performance in prediction on CASF 2016 true data with anomalies in Chai-1 deleted for fair comparison (Pearson correlation 0.7297 & 0.7722, the latter with test-time augmentation, TTA) and expectedly reduced correlation on predicted structures. Through detailed case studies and error analysis, we assess factors affecting model performance including the failures of structure prediction models, hybridization challenges, rotational variance, and hyperparameter settings. This work attempts to provide insights into the applicability of deep learning scoring functions for both experimental and predicted protein-ligand complex structures in drug discovery. Our code is in https://github.com/itsuitsuki/SFCNN_PLA_torch for release.

1 Introduction

Protein-ligand binding affinity prediction represents a cornerstone challenge in computational drug discovery and structural bioinformatics. Accurate prediction of binding affinities is paid much attention to for rational drug design. Traditional approaches have relied on empirical scoring functions [Guedes et al., 2018], physics-based models [Huang et al., 2010, Morris et al., 2009], and knowledge-based potentials [Gohlke et al., 2000], each with inherent limitations in capturing the complex, non-linear nature of protein-ligand interactions [Shirali et al., 2025].

The advent of deep learning has revolutionized this field, offering data-driven approaches that can learn intricate patterns from structural data without explicit feature engineering, such as graph-based [Dandibhotla et al., 2025, Lim et al., 2019] or sequence-based methods [Öztürk et al., 2018]. Among these methods, Scoring Function Convolutional Neural Networks (SFCNN) have emerged as a promising approach, utilizing 3D convolutional architectures to process voxelized representations of protein-ligand complexes. This methodology directly leverages spatial information encoded in atomic coordinates, potentially capturing subtle geometric and chemical features that govern binding affinity [Wang et al., 2022].

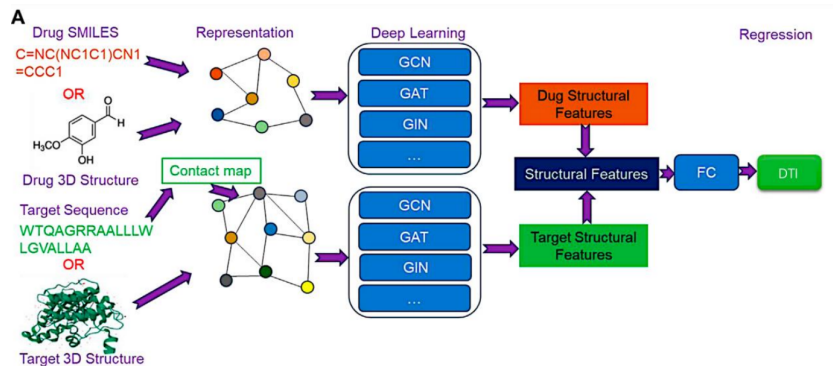


Figure 1: An example of existing non-CNN deep methods for PLA prediction, which calls for sequences and graphs frequently.

Recent advances in protein structure prediction, particularly AlphaFold3 and Chai-1, have opened new possibilities for structure-based drug design. These tools can generate high-quality protein-ligand complex structure predictions, potentially expanding the scope of computational drug discovery beyond experimentally determined structures since we can directly build a sequence (or graph)-to-prediction pipeline [Abramson et al., 2024, Boitreau et al., 2024]. However, the applicability of existing scoring functions to predicted structures remains an open question, requiring systematic evaluation and analysis.

In this work, we present a comprehensive re-analysis of SFCNN, focusing on both methodological improvements and evaluation on diverse structural datasets. Our contributions include a set of novel analysis steps: (1) detailed implementation and optimization of SFCNN architecture, (2) systematic evaluation on both experimental and predicted structures, (3) error analysis and case studies of outlier complexes and the variance of model checkpoints, and (4) insights into the transferability of deep learning scoring functions across different structural data sources.

2 Background

2.1 Protein-Ligand Binding Affinity Prediction

The accurate prediction of protein-ligand binding affinity, a measure of interaction strength between a protein and a ligand, is crucial for drug discovery [Abramson et al., 2024]. Computational methods that can reliably estimate affinity accelerate the identification of promising drug candidates.

Traditional approaches have largely relied on molecular docking combined with scoring functions to predict binding poses and their associated affinities [Huang et al., 2010, Shirali et al., 2025]. These scoring functions include force-field-based methods (e.g., in AutoDock [Morris et al., 2009]), empirical functions tuned with experimental data [Guedes et al., 2018], and knowledge-based potentials derived from structural databases [Gohlke et al., 2000]. While foundational, these methods often face limitations in accuracy and generalizability [Shirali et al., 2025].

The increasing availability of structural and affinity data has driven the rise of machine learning (particularly deep learning, DL) for this task. DL models can automatically learn salient features from molecular data. For instance, graph Neural Networks (GNNs) have shown significant promise by treating molecules as graphs, whether by incorporating 3D structural information [Lim et al., 2019] or by operating on sequence-based graph representations [Dandibhotla et al., 2025]. Other DL architectures, such as DeepDTA, predict affinity directly from sequence information (e.g., protein sequences and ligand SMILES strings) [Öztürk et al., 2018]. Also, 3D Convolutional Neural Networks (CNNs) like SFCNN learn from 3D representations of the protein-ligand complex [Wang et al., 2022], as our research subject.

2.2 Structure Prediction Models for Complex Generation

State-of-the-art AI models like AlphaFold 3 [Abramson et al., 2024] and Chai-1 [Boitreau et al., 2024] now enable highly accurate prediction of biomolecular complex structures, crucial for sub-

sequent analyses such as binding affinity assessment. AlphaFold 3 employs an Evoformer module (Pairformer) and a diffusion model to generate 3D coordinates [Abramson et al., 2024]. Chai-1 utilizes a unified transformer-based architecture for diverse inputs (including arbitrary ligands via SMILES) and directly predicts the 3D atomic coordinates of the complex [Boitreau et al., 2024], distinguishing its approach from AlphaFold 3’s diffusion mechanism.

Due to computational resource limitations, our study relies on public web servers. The AlphaFold 3 web server, while powerful, offers a restricted set of ligand choices for protein-ligand docking. Conversely, the Chai-1 web server supports arbitrary ligand inputs. Therefore, to accommodate our need for diverse ligand complex generation, we will use the Chai-1 web server for subsequent analysis.

2.3 Benchmark Datasets: PDBbind and CASF

For training and evaluation, we utilize the **PDBbind v2019 refined set**, which contains high-quality protein–ligand complexes with experimentally determined binding affinities [Wang et al., 2005] and also the dataset for **the CASF-2016 benchmark** derived from PDBbind, providing a standardized test set for assessing the scoring power of binding affinity prediction models [Su et al., 2018]. The labels in both datasets contain $pK_a = -\lg K_a$, $-\lg K_i$ or $-\lg K_d$ values. These datasets are widely adopted in the field and facilitate fair comparisons among different methods.

3 Methods

3.1 Feature Extraction

In each dataset (PDBbind v2019 and CASF 2016), for each protein–ligand complex, coordinates of atoms are first adjusted where the geometric center of the ligand molecule is moved to the origin, to let the model accurately capture more entire information of the **binding pocket**. A cubic grid of G shape ($20 \times 20 \times 20$) is created as the input feature so as to approximate the original sequence of coordinates i.e. the structure, whose granularity is one $\text{\AA}^3$ for one unit. The grid approximates the coordinates in the given structure and outer atoms are truncated outside the range of the grid as well, resulting in some loss in precision.

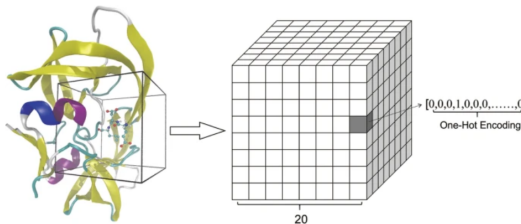


Figure 2: The encoding scheme. Atoms outside the grid are truncated.

The seen district is thus $8000 \text{\AA}^3$ around the ligand center. Every unit in this grid is a one-hot vector calculated by checking the existence of different types of atoms in this unit. This captures the spatial distribution of atom types within the binding site, resulting in a 4-dimensional tensor input of shape $(20 \times 20 \times 20 \times N_{\text{features}})$, where $N_{\text{features}} = 28$ is the total number of unique atom type features considered after distinguishing between protein and ligand atoms.

For feature engineering, specific atom types and their hybridization states are considered. These include Hydrogen (H), Carbon (C), Nitrogen (N), Oxygen (O), Fluorine (F), Phosphorus (P), Sulfur (S), Selenium (Se), and a general category for Halogens (encompassing F, Cl, Br, I). Carbon atoms are further distinguished based on their sp , sp^2 , and sp^3 hybridization states. Similarly, Nitrogen atoms are also categorized into sp , sp^2 , and sp^3 hybridization states. Sulfur atoms are classified based on sp^2 and sp^3 (or similar higher-order) hybridization states. The hybridization states, since PDB files as the data file format don’t record them, are deduced by a set of complicated rules in OpenBabel [O’Boyle et al., 2011] as in the original paper of SFCNN. All other atom types not explicitly falling into these defined categories are grouped into a single “others” feature type. Each of these defined atom/hybridization categories contributes to the feature vector for a given grid cell, with the atoms from the protein and those from the ligand counted in different indices, thus forming the final feature dimension of the input tensor. For the formula, for an atom type as aforementioned denoted as t ,

$$G[x, y, z, t] = \begin{cases} 1 & \text{if } t \text{ is in the structure,} \\ 0 & \text{otherwise.} \end{cases}$$

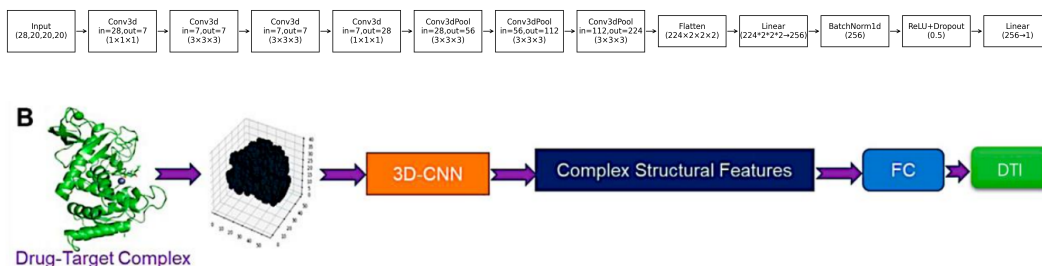


Figure 3: The model architecture and inference pipeline of SFCNN.

where each component contributes 14 types for this feature dimension, which sums up to 28 as N_{features} .

3.2 SFCNN Architecture

The SFCNN model architecture for Protein-Ligand Affinity Prediction is constructed as a deep convolutional neural network, followed by fully connected layers. The core of the model consists of seven 3D convolutional blocks. The first four convolutional blocks follow a sequence of 3D Convolution, 3D Batch Normalization, and Rectified Linear Unit (ReLU) activation. The subsequent three convolutional blocks extend this pattern by adding a 3D Max Pooling (MaxPool3D) layer after the ReLU activation.

Following the convolutional blocks, the output tensor is flattened and then processed by a multi-layer perceptron (MLP). This MLP consists of a dense (fully connected) layer, followed by 1D Batch Normalization (BatchNorm1D), ReLU activation, a dropout layer for regularization, and finally, another L2-regularized dense layer that outputs the predicted binding affinity. Figure 3 also shows the exact architecture as depicted and the inference pipeline, including the feature extraction part.

The whole architecture is re-implemented via PyTorch, instead of the original TensorFlow library as in the requirement, and also since the original version is neither much readable nor runnable.

3.3 Weight Initialization

We also employ a specific random weight initialization strategy for the layers of the SFCNN model. For 3D Convolutional layers, weights are initialized using the Kaiming Normal method (also known as He Normal initialization) with fan-out and ReLU nonlinearity. If these convolutional layers have a bias term, it is initialized to zeros. For Linear (fully connected) layers, weights are initialized using Kaiming Uniform initialization. An additional scaling step is performed where the weight data is multiplied by 0.1. Bias terms in linear layers, if present, are initialized to zeros. For both 1D Batch Normalization and 3D Batch Normalization layers, the learnable affine parameters (weight) are initialized to ones, and the bias terms are initialized to zeros.

4 Training

4.1 Data Preparation and Augmentation

Same as in the original paper, we apply the PDBbind v2019 refined dataset as the primitive set, also excluding complexes from the test set to prevent data leakage. The primitive set is then separated into the training set and the validation set, where the training set contains 4100 complex structure-label pairs and the validation set contains 486 pairs.

For the training set, augmentation is also performed by applying random rotations to each complex to enhance the rotation invariance property of the model. Besides the original complex structure, we apply 9 random rotations for each complex. On the other hand, CASF-2016 scoring power benchmark is specified as the testing dataset, while we create two versions of the set, one for original evaluation and another applied with test-time augmentation (TTA). There are 285 complexes in this dataset as the testing set, while 266 are overlapped with PDBbind v2019 refined set with a population of 4852.

As mentioned above, we de-duplicate the 266 in the training set, and yield the other 4586 pairs for training and validating as above [Wang et al., 2022].

We store the processed datasets into Apache Arrow format, which is a cross-language development platform for in-memory data, integrated with HuggingFace as our tool for storing. It specifies a standardized language-independent columnar memory format for flat and hierarchical data, organized for efficient analytic operations on modern hardware like CPUs and GPUs. The Hugging Face datasets library utilizes Apache Arrow to store datasets, enabling efficient data access and manipulation.

4.2 Training Objective and Testing Metric

The training follows mean squared error (MSE), a classical objective as the loss function:

$$L_{\text{MSE}} = \sum_{i \in \text{batch}} (y_i - \hat{y}_i)^2$$

where L_{MSE} is the MSE loss, y_i is the label PLA value and $\hat{y}_i$ is the predicted value; and the Pearson correlation R as the validation and testing metric to show the fitting and generalization performance of the specified hyperparameter set:

$$R = \frac{\sum_{i=1}^n (y_i - \bar{y})(\hat{y}_i - \bar{\hat{y}})}{\sqrt{\sum_{i=1}^n (y_i - \bar{y})^2} \sqrt{\sum_{i=1}^n (\hat{y}_i - \bar{\hat{y}})^2}}$$

where the barred symbol is the mean value in the list of values. The overall loss function can be simply set as the MSE loss, but theoretically, we can also apply a Pearson correlation penalty for further increasing it since the correlation is differentiable: $L = L_{\text{MSE}} - \lambda R$, but experiments show this will suffer oscillation in the Pearson correlation, not improving the performance. Hence we still set $\lambda = 0$. The Pearson Correlation R will be evaluated at the end of each epoch for the model, where we keep the best checkpoint with the largest R dynamically in the training process.

4.3 Further Training Configuration

As is addressed in 3.2, we apply a weight decay (i.e. L2 regularization) term to the last FC layer which is a hyperparameter; we also apply the RMSprop optimizer as is introduced in Ruder [2017] in detail with a default set of parameters. We also initialize the learning rate as 1e-05, and apply a linear warmup scheduler with 200 steps (batches), gradually and linearly increasing the learning rate to a set hyperparameter (denoted as the final LR), preventing the loss from drastically oscillating in the first several epochs (except the first epoch) after the initialization. Additionally, we set gradient clipping to a norm of 3, saving the training from gradient explosion. As in Wang et al. [2022], we train with 200 epochs. We have implemented the training script efficiently, including leveraging the mentioned dataset format, applying the data parallel scheme onto the model, and prefetching and multithreading in the data loading section.

4.4 Hyperparameter Optimization

Furthermore, we have built an Optuna-based hyperparameter search pipeline with 25 trials in each study. The TPESampler and the median pruner are applied for optimizing hyperparameter-related R metrics. This is proven effective in enhancing the testing performance of the model checkpoint as the searching result, which is higher than other human-tuned checkpoints both with and without TTA.

We have finally saved 3 checkpoints during the training and optimization process, where the best is the one optimized by the hyperparameter rounded from the searched result, (denoted as **SFCNN-HPO**), and the others are by hand, where the only difference between them is the dropout values: 0.1 and 0.5 respectively, hence we call the other two **SFCNN-D1** and **SFCNN-D5** by these values. The hyperparameters for the checkpoints mentioned above are shown in table 1.

Hyperparameter	SFCNN-HPO	SFCNN-D1	SFCNN-D5
Final LR	0.0016	0.004	0.004
Dropout	0.3	0.1	0.5
Weight Decay	0.015	0.01	0.01
Batch Size	16	32	32

Table 1: Hyperparameter settings for different model checkpoints.

Checkpoint	Without TTA				With TTA
	MSE	Pearson Corr.	Spearman Corr.	C-Index	Pearson Corr.
SFCNN-HPO	0.0101	0.7297	0.7306	0.767	0.7722
SFCNN-D5	0.012	0.7102	0.7018	0.7569	0.7214
SFCNN-D1	0.0105	0.7267	0.7083	0.7626	0.7397

Table 2: Model performance w/o and w/ TTA, on GSE.

5 Evaluation

We have conducted the evaluation process including (1) ordinary evaluation, named as **groundtruth structure evaluation** (GSE), with groundtruth structures in the CASF-16 testing set from wet lab data, (2) **bootstrapped evaluation** (BE) based on predicted structures by Chai-1 and (3) comparison of Pearson Corr. performance for GSE and BE. We also report the performance of each model checkpoint for (1) and (2). Since the bootstrapped evaluation involves sequence extraction, some anomalies in this process have occurred and therefore the *intersection* of complexes is evaluated, yielding finally 282 complexes except 1o0h, 2ymd and 2zcr.

5.1 Groundtruth Structure Evaluation

The performance in GSE is listed in table 2. As this table indicates, test-time augmentation (TTA), composed of multiple rotations, multi-prediction, and subsequent value averaging, significantly improves the performance without extra training or any modification of models. We also show other performance metrics such as MSE, Spearman correlation (see appendix A.1), and concordance index (C-index, see appendix A.2) as auxiliary information in GSE, but not in BE since we compare mainly on the Pearson correlation metric.

5.2 Bootstrapped Evaluation based on Chai-1

Since we predict the PLA values based on a previously predicted set of data, the prediction can be called as done on a bootstrapped dataset (testing set), and the evaluation is thus named as bootstrapped evaluation, which is in greater detail discussed in section 5.2. For convenience and analysis afterwards, we only keep the TTA on in the following evaluation, without the other choice. We follow the procedures introduced in section 5.2.1 and section 5.2.2 to collect and create the testing dataset for bootstrapped evaluation.

5.2.1 Sequence Extraction and Chai-1 Calling

By our written script, we first extract the SMILES sequence from the ligand MOL2 file using RDKit [Landrum et al., 2025], and extract the amino acid sequence(s) of the protein(s) by a dictionary-based rule. We then input every extracted sequence set into Chai-1, using MMSeq2 [Steinegger and Söding, 2017] as multi-sequence alignment to enhance the generation without preset templates. Among the results given by Chai-1, we pick the one with the highest confidence (pTM score, instead of RMSD as a metric) rank, i.e. the `pred.rank_0.cif`. The corresponding CIF files containing predicted structures are then fed into the parsing.

5.2.2 CIF Parsing and Bootstrapped Feature Extraction

For a CIF file, the hybridization is not given either. In the parsing of PDB files, we choose the same paradigm as the original paper in SFCNN by OpenBabel [O’Boyle et al., 2011] as said in section 3.1. However, for CIF files, we have no given libraries to effectively infer the hybridization states, so we apply a *heuristic* method to find the potential hybridization for carbon, nitrogen, and sulfur atoms.

This determination is based on the atom’s element type, its specific name within the residue, the type of its parent residue (e.g., amino acid or ligand), and whether it is classified as a HETATM (non-standard residue or ligand). For atoms within standard protein residues, the function applies specific rules derived from common chemical structures: for example, carbon and nitrogen atoms forming peptide bonds, as well as carbons in aromatic rings, are generally assigned sp^2 hybridization; most other carbon atoms such as alpha and beta carbons are typically considered sp^3 .

5.2.3 Performance Evaluation

The table 3 shows with TTA, the mutual Pearson correlation of each checkpoint, defined as the Pearson correlation on GSE and BE predicted values providing more information for comparison, and the ordinary BE Pearson correlation as well. Similarly, the automatically searched checkpoint (SFCNN-HPO) outperforms the other two in both types of Pearson correlation with TTA, and SFCNN-D5 is the best w/o TTA. Another result is that the performance of these models in BE is apparently lower than that in GSE, which is also expected.

Checkpoint w/ TTA	BE	Mutual
SFCNN-HPO	0.6147	0.7406
SFCNN-D1	0.5946	0.7349
SFCNN-D5	0.5669	0.6976
Checkpoint w/o TTA	BE	Mutual
SFCNN-HPO	0.4592	0.5035
SFCNN-D1	0.5244	0.5175
SFCNN-D5	0.5934	0.6800

Table 3: Comparison of BE and mutual Pearson correlations.

5.3 Visualization and Observation

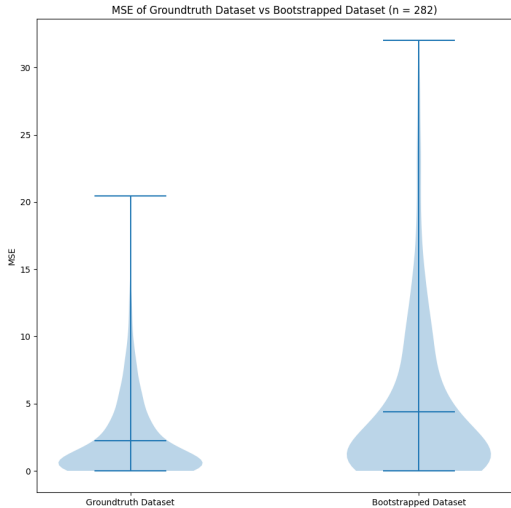


Figure 4: Violin graph for MSE (SFCNN-HPO)

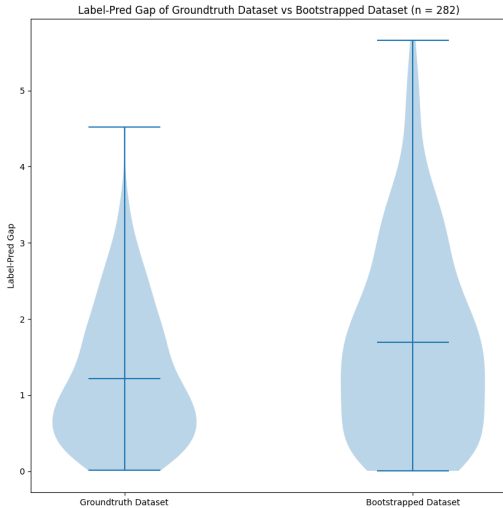


Figure 5: Violin graph for label gaps (SFCNN-HPO w/ TTA)

In all checkpoints with or without TTA, we observe that the distribution of MSE across the complexes in GSE is comprehensively and on average lower than in BE as defined in section 5, where plotted in fig. 5 is evaluated for a representative checkpoint.

We have also plotted the predicted values and the label values in fig. 10, from which we can observe a stronger linearity (identity) in GSE than BE, where BE shows a steeper slope (almost perpendicular

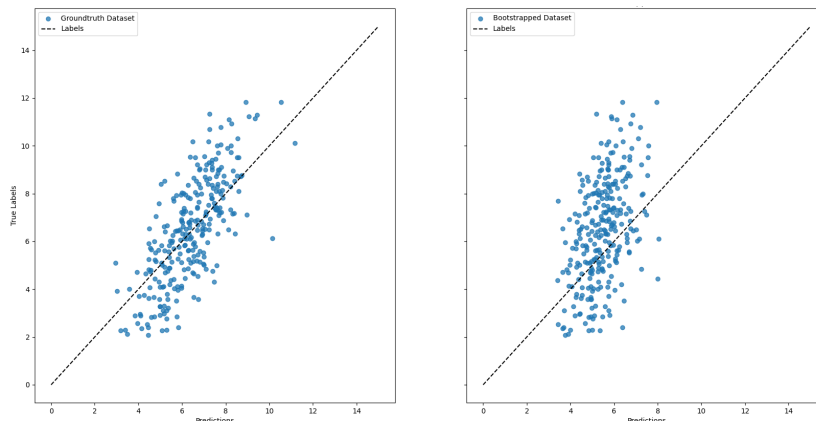


Figure 6: Predictions and labels on GSE (left) and BE (right) on CASF-16. (SFCNN-HPO w/ TTA)

to x -axis with an average at about $x = 6$) in this coordinate system where the x -axis is predictions and y -axis is for labels.

This phenomenon of under-performance attracts our attention to why the bootstrapped evaluation suffers such a salient performance gap, which can be partially ascribed to heuristic differences. corresponding to the following analysis via **categorization** of the failure modes of predicted complexes as in section 6.1 and the potential factors in SFCNN or the structure prediction models causing errors as in section 6.2.

6 Analysis

In the evaluation section, we find there are complexes with large MSE when predicted on the bootstrapped structures while performing relatively much better on the real structures, as plotted in fig. 7 for SFCNN-D1 without TTA. The red points are the complexes with MSE on GSE less than 2.5, MSE on BE more than 10 and the gap more than 10. We would like to find the cause for this performance gap behind this phenomenon, by case studies utilizing PyMol [DeLano, 2002] to visualize the complexes with the largest MSE gaps in most checkpoints.

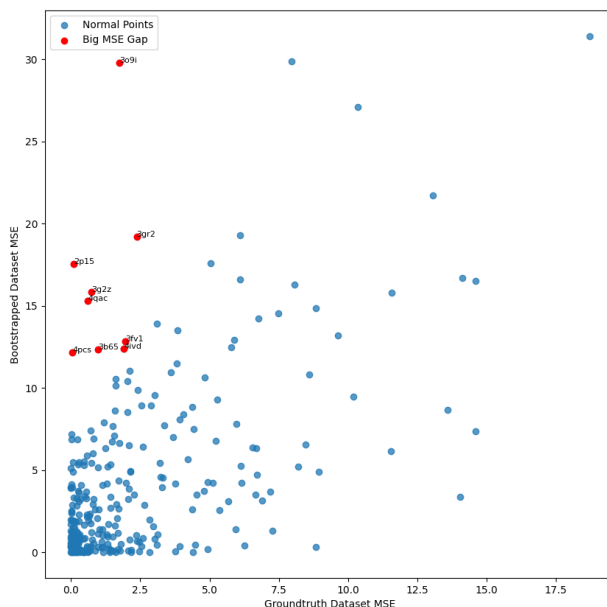


Figure 7: MSE and outliers in MSE gaps (SFCNN-D1, w/o TTA).

6.1 Structure Prediction Failure Modes

For visualization, the real protein structure is rendered as a green cartoon, while the protein portion of the predicted complex is displayed as a blue cartoon; the real ligand is shown in stick representation, colored orange, with the stick radius set to $0.25\text{\AA}$ for clarity, while the predicted ligand is also displayed in stick representation but colored purple, maintaining the same stick radius of $0.25\text{\AA}$.

Most proteins are well-folded aligning the groundtruth (much similar to the real structure) probably since Chai-1 or similar models have an abundant training set of protein data, where there are some failure modes involving the absence of ions, failure in ligand folding or locating, or both. There are still some small differences of carbon strands (as curved thin lines in PyMol) probably causing the errors in proteins. The following case studies are categorized into 3 categories, showing typical cases of failure in ligand folding with high MSE gaps.

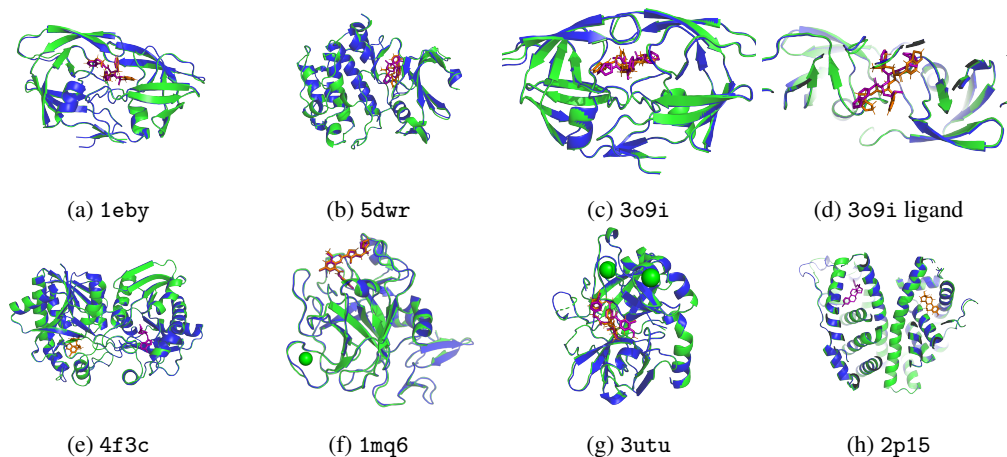


Figure 8

Failure in Ligand Folding. For only partially overlapped (in PyMol) cases that the ligand does not form symmetry, we call them ligand folding failures. For instance, in 1eby (fig. 8a), one end of the predicted ligand is not elongated as well as the groundtruth, and the 5dwr (fig. 8b) one also shows a similar failure of folding. Intuitively, a more similar (or homogeneous) structure results in a closer projected predicted value in a nonlinear model as long as the model represents a mapping whose image shapes a well-formed manifold, which is potentially a reason for the errors to occur that the structures do not truly align, although the locations are approximately the same.

Misplacement of Ligands. For the misplacement of ligands, in 3o9i (fig. 8c) the ligand shows a central symmetry around the ligand geometric center, between the predicted and the groundtruth which can be seen more clearly in fig. 8d (though the predicted ligand here seems also overlapped with the real ligand, but there appears no observable overlapping); and in 4f3c (fig. 8e) the ligand also shows a central symmetry around the protein geometric center. For each protein, the two protein parts or the complex parts are centrally symmetric enough to form a perfect alignment, but there are slight misalignments on two sides for each complex in this category, hence the *wrongly symmetrized* ligand in the prediction may cause significant misinformation.

For 2p15, though the protein has a perfect structure with central symmetry and there is also a perfect alignment for the groundtruth and bootstrapped proteins, the predicted ligand molecule shows an inconsistent axis-symmetric property (i.e. *chiral*) with the groundtruth one (while the proteins are achiral here) which leads to the failure, clearly seen in fig. 8h.

Failure in Extracting Ions We can observe in 1mq6 and 3utu (fig. 8f and fig. 8g), there are spheres in the green (groundtruth) protein structure, representing ions such as Na^+ in 1mq6. However, these ions cannot be extracted by our script which reads PDB files as sequences since the ions cannot be arranged in the sequence, but should be seen as another ligand. By this *absence of ions*, though the proteins are still predicted well, the predicted value will change a lot since the encoded feature grid loses an “other” term, which may influence the information from the grid significantly due to its

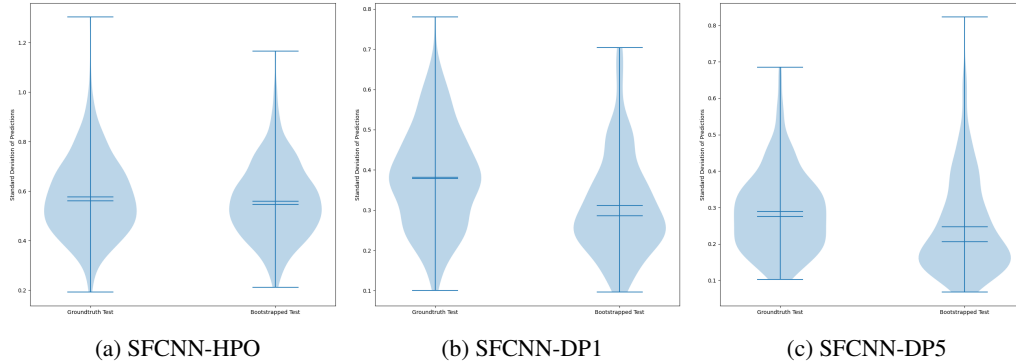


Figure 9: Violin graph for standard deviations of predictions in GSE and BE w/ TTA.

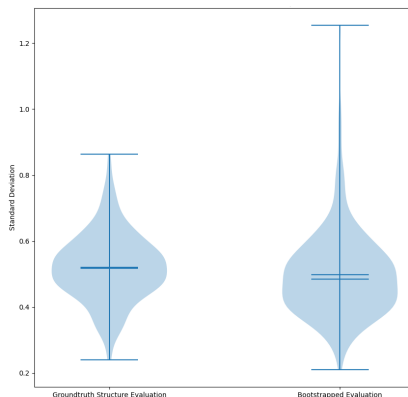


Figure 10: Violin graph for standard deviation of predictions across 3 checkpoints w/ TTA.

sparsity, which is also visualized via the absence of spheres in the blue proteins, though the ligands align perfectly.

Summary. From the case studies researching on 7 complexes from the CASF-16 dataset, for Chai-1 or this series of structure prediction neural models, the failure modes causing significant errors in *downstream tasks* such as PLA tasks here involve frequently the failure in ligand folding and locating, and the absence of extra information for hybridization inference as well as the absence of ion representations at least in this work.

Moreover, though predicting structures of complexes with outstanding performance, the objective of downstream tasks is not the same as the prediction, not similar to the same pretraining-finetuning paradigm in language modeling tasks. This can be explained as a reason for the failures when the structures are not too divergent, also concluded for AlphaFold 3 in Zheng et al. [2025].

6.2 Variance

We want to find the general weakness of the model. Therefore, we analyze the variances as possible reasons for low performance and instability.

6.2.1 Rotational Variance

Besides the errors brought by the structure prediction model, SFCNN by assumption can be also weak since its convolutional architecture cannot provide rotational invariance, but only shift and scaling equivariance. Hence, by TTA, before the averaging we record the standard deviation (root of variance, SD) of the predicted values of each complex, as in fig. 9. The BE predicted values show lower mean SD values than GSE, probably because of the overfitting of SFCNN-HPO on the manifold of real structures and their rotation configurations, which we call as **rotational variance**. In fact, by table 2, we can observe that TTA designed by us to reduce the rotational variance of the trained model (without modifying the weights) shows great effects on significantly improving the BE

Pearson correlation by at most about 0.2, much more than the improvement for GSE, which verifies the existence of rotation variance.

6.2.2 Training Variance

Since the shape of hidden states in the SFCNN model changes at each layer, it is impossible to directly add the residual connection for preserving identity, comparatively causing some instability, which may cause the checkpoint predicted values to have large variance when comparing different model checkpoints, called **training variance**. This assumption is general without comparing GSE and BE.

We attempt to verify this hypothesis by plotting the SD of the pre-averaging predicted values when TTA is generated by all models, but to our surprise, the violin graph of each side seems similar to each other and does not show higher standard deviation levels than all cases when evaluated at a single checkpoint, i.e. the concatenation of predicted values does not increase the variance of the values, which indicates this variance cannot be concluded as the potential cause for errors.

7 Further Discussion

While this study has provided insights into SFCNN performance and its application to predicted structures, several avenues for future research could further advance the field of protein-ligand binding affinity prediction.

Expansion of training datasets. The size and diversity of the training dataset are crucial for the generalization capabilities of deep learning models. While PDBbind is a valuable resource, exploring and integrating other extensive datasets such as BioLiP [Yang et al., 2012] could be beneficial. BioLiP is a comprehensive database of biologically relevant ligand-protein interactions, focusing on the binding sites. While PDBbind (and thus CASF-2016 which is derived from it) primarily curates binding affinity data alongside structures, BioLiP provides a large collection of ligand-bound protein structures which could be used for pre-training, learning richer structural features of interactions, or augmenting affinity datasets if binding data can be cross-referenced or reliably estimated. Augmenting the training set with diverse examples from BioLiP might help the model learn more robust features of protein-ligand interactions.

Architectural enhancements for SFCNN. The current SFCNN architecture, while effective, could potentially be improved. For instance, incorporating cross-layer connections with non-identity $1 \times 1 \times 1$ convolutions could act as projection shortcuts, similar to those in ResNet architectures [He et al., 2016]. Such modifications might allow for deeper models or better gradient flow, potentially leading to enhanced predictive performance by enabling the learning of more complex representations without succumbing to vanishing gradients.

Alternative architectures (ViT). The expressive power of purely convolutional architectures might have inherent limitations in capturing long-range dependencies within the 3D voxel space. Vision Transformers (ViTs) have demonstrated remarkable success in 2D image analysis and are increasingly being adapted for 3D data, while Chen et al. [2021] has provided a strongly performing 3D ViT architecture for medical imaging. Developing a Voxel Vision Transformer specifically tailored for protein-ligand complexes could offer a powerful alternative. Such models could potentially better capture global contextual information within the binding pocket and interactions between distant atom groups, potentially surpassing the capabilities of traditional SFCNN for this task.

8 Conclusion

In this work, we conducted a comprehensive re-analysis of Scoring Function Convolutional Neural Networks (SFCNN) for predicting protein-ligand binding affinities, evaluating their performance on both experimentally determined structures (CASF-2016 benchmark) and computationally predicted structures generated by Chai-1. Our optimized SFCNN-HPO model, utilizing 4-dimensional one-hot encoded voxel representations and enhanced by test-time augmentation (TTA), demonstrated strong predictive capability on the CASF-2016 groundtruth data, achieving a Pearson correlation coefficient (R) of 0.7722. As anticipated, when applied to the bootstrapped dataset of Chai-1 predicted structures,

the model's performance was reduced, yielding an R value of 0.6147 with TTA, highlighting the challenge of applying these models to computationally derived structures.

Our detailed analysis identified key factors contributing to the performance disparity on predicted structures. These primarily included inaccuracies in ligand pose prediction by Chai-1 (such as errors in folding, placement, and the omission of crucial ions) and the inherent rotational variance of the SFCNN architecture, the latter being effectively mitigated by TTA. This study underscores the significant impact of upstream structure prediction quality on downstream binding affinity calculations. It emphasizes the challenges and considerations essential for reliably integrating deep learning scoring functions with computationally generated structural data, which is increasingly prevalent in drug discovery pipelines.

9 Contribution

- Chuyan Zhou: All coding unless otherwise mentioned, including (not only) the re-implementation of the scripts and all running of model architecture, the feature extraction, the training process, the hyperparameter search, the evaluation on CASF-16 and Chai-1 predicted set, the visualization of all charts in this project, and Github repository maintenance (90% coding work). Provided all computational resources (2x A6000). Completed all slides of, and presented both inter-team presentations. Wrote all parts of the final report starting from the background, including the background, the method, the evaluation, the analysis, the conclusion, the reference and the appendix. Reviewed & refined the abstract and the introduction from an initial version. (98% report work) Worked as a main presenter for the slides of final presentation. 20% work of running Chai-1-based structure generation by web server.
- Hanze Li: Final Presentation slides major contributor. First version of introduction and abstract in the final report. Worked on a major part for the coding of the script that extracts SMILES sequences from MOL2 files, and extracts protein sequences from PDB files. 40% work of running Chai-1 web server.
- Xinyang Xu: Major contributor for the Case study part, including comprehensive visualization and cause attribution. The coding of parts of CIF feature extraction and also part of the coding of the script that extracts protein sequences from PDB files. 40% work of running Chai-1 web server.

References

- Isabella A. Guedes, Felipe S. S. Pereira, and Laurent E. Dardenne. Empirical scoring functions for structure-based virtual screening: Applications, critical aspects, and challenges. *Frontiers in Pharmacology*, 9, September 2018. ISSN 1663-9812. doi: 10.3389/fphar.2018.01089. URL <http://dx.doi.org/10.3389/fphar.2018.01089>.
- Sheng-You Huang, Sam Z. Grinter, and Xiaoqin Zou. Scoring functions and their evaluation methods for protein–ligand docking: recent advances and future directions. *Physical Chemistry Chemical Physics*, 12(40):12899, 2010. ISSN 1463-9084. doi: 10.1039/c0cp00151a. URL <http://dx.doi.org/10.1039/C0CP00151A>.
- Garrett M. Morris, Ruth Huey, William Lindstrom, Michel F. Sanner, Richard K. Belew, David S. Goodsell, and Arthur J. Olson. Autodock4 and autodocktools4: Automated docking with selective receptor flexibility. *Journal of Computational Chemistry*, 30(16):2785–2791, April 2009. ISSN 1096-987X. doi: 10.1002/jcc.21256. URL <http://dx.doi.org/10.1002/jcc.21256>.
- Holger Gohlke, Manfred Hendlich, and Gerhard Klebe. Knowledge-based scoring function to predict protein-ligand interactions. *Journal of Molecular Biology*, 295(2):337–356, January 2000. ISSN 0022-2836. doi: 10.1006/jmbi.1999.3371. URL <http://dx.doi.org/10.1006/jmbi.1999.3371>.
- Azam Shirali, Vitalii Stebliankin, Ukesh Karki, Jimeng Shi, Prem Chapagain, and Giri Narasimhan. A comprehensive survey of scoring functions for protein docking models. *BMC Bioinformatics*, 26(1), January 2025. ISSN 1471-2105. doi: 10.1186/s12859-024-05991-4. URL <http://dx.doi.org/10.1186/s12859-024-05991-4>.

- Somanath Dandibhotla, Madhav Samudrala, Arjun Kaneriy, and Sivanesan Dakshanamurthy. Gnnseq: A sequence-based graph neural network for predicting protein–ligand binding affinity. *Pharmaceuticals*, 18(3), 2025. ISSN 1424-8247. doi: 10.3390/ph18030329. URL <https://www.mdpi.com/1424-8247/18/3/329>.
- Jaechang Lim, Seongok Ryu, Kyubyong Park, Yo Joong Choe, Jiyeon Ham, and Woo Youn Kim. Predicting drug–target interaction using a novel graph neural network with 3d structure-embedded graph representation. *Journal of Chemical Information and Modeling*, 59(9):3981–3988, August 2019. ISSN 1549-960X. doi: 10.1021/acs.jcim.9b00387. URL <http://dx.doi.org/10.1021/acs.jcim.9b00387>.
- Hakime Öztürk, Arzucan Özgür, and Elif Ozkirimli. Deepdta: deep drug–target binding affinity prediction. *Bioinformatics*, 34(17):i821–i829, September 2018. ISSN 1367-4811. doi: 10.1093/bioinformatics/bty593. URL <http://dx.doi.org/10.1093/bioinformatics/bty593>.
- Yu Wang, Zhengxiao Wei, and Lei Xi. Sfcnn: a novel scoring function based on 3d convolutional neural network for accurate and stable protein–ligand affinity prediction. *BMC Bioinformatics*, 23(1), June 2022. ISSN 1471-2105. doi: 10.1186/s12859-022-04762-3. URL <http://dx.doi.org/10.1186/s12859-022-04762-3>.
- Josh Abramson, Jonas Adler, Jack Dunger, Richard Evans, Tim Green, Alexander Pritzel, Olaf Ronneberger, Lindsay Willmore, Andrew J. Ballard, Joshua Bambrick, Sebastian W. Bodenstein, David A. Evans, Chia-Chun Hung, Michael O’Neill, David Reiman, Kathryn Tunyasuvunakool, Zachary Wu, Akvilė Žemgulytė, Eirini Arvaniti, Charles Beattie, Ottavia Bertolli, Alex Bridgland, Alexey Cherepanov, Miles Congreve, Alexander I. Cowen-Rivers, Andrew Cowie, Michael Figurnov, Fabian B. Fuchs, Hannah Gladman, Rishub Jain, Yousuf A. Khan, Caroline M. R. Low, Kuba Perlin, Anna Potapenko, Pascal Savy, Sukhdeep Singh, Adrian Stecula, Ashok Thillaisundaram, Catherine Tong, Sergei Yakneen, Ellen D. Zhong, Michal Zielinski, Augustin Židek, Victor Bapst, Pushmeet Kohli, Max Jaderberg, Demis Hassabis, and John M. Jumper. Accurate structure prediction of biomolecular interactions with alphafold 3. *Nature*, 630(8016):493–500, May 2024. ISSN 1476-4687. doi: 10.1038/s41586-024-07487-w. URL <http://dx.doi.org/10.1038/s41586-024-07487-w>.
- Jacques Boitreaud, Jack Dent, Matthew McPartlon, Joshua Meier, Vinicius Reis, Alex Rogozhnikov, and Kevin Wu. Chai-1: Decoding the molecular interactions of life. October 2024. doi: 10.1101/2024.10.10.615955. URL <http://dx.doi.org/10.1101/2024.10.10.615955>.
- Renxiao Wang, Xuiliang Fang, Yipin Lu, Chao-Yie Yang, and Shaomeng Wang. The pdbname database: Methodologies and updates. *Journal of Medicinal Chemistry*, 48(12):4111–4119, May 2005. ISSN 1520-4804. doi: 10.1021/jm048957q. URL <http://dx.doi.org/10.1021/jm048957q>.
- Minyi Su, Qifan Yang, Yu Du, Guoqin Feng, Zhihai Liu, Yan Li, and Renxiao Wang. Comparative assessment of scoring functions: The casf-2016 update. *Journal of Chemical Information and Modeling*, 59(2):895–913, November 2018. ISSN 1549-960X. doi: 10.1021/acs.jcim.8b00545. URL <http://dx.doi.org/10.1021/acs.jcim.8b00545>.
- Noel M O’Boyle, Michael Banck, Craig A James, Chris Morley, Tim Vandermeersch, and Geoffrey R Hutchison. Open babel: An open chemical toolbox. *Journal of Cheminformatics*, 3(1), October 2011. ISSN 1758-2946. doi: 10.1186/1758-2946-3-33. URL <http://dx.doi.org/10.1186/1758-2946-3-33>.
- Sebastian Ruder. An overview of gradient descent optimization algorithms, 2017. URL <https://arxiv.org/abs/1609.04747>.
- Greg Landrum, Paolo Tosco, Brian Kelley, Ricardo Rodriguez, David Cosgrove, Riccardo Vianello, sriniker, Peter Gedeck, Gareth Jones, Eisuke Kawashima, NadineSchneider, Dan Nealschneider, Andrew Dalke, Matt Swain, Brian Cole, Samo Turk, Aleksandr Savelev, tadhurst cdd, Alain Vaucher, Maciej Wójcikowski, Ichiru Take, Rachel Walker, Vincent F. Scalfani, Hussein Faara, Kazuya Ujihara, Daniel Probst, Juuso Lehtivarjo, guillaume godin, Axel Pahl, and Niels Maeder. rdkit/rdkit: 2025_03_2 (q1 2025) release, 2025. URL <https://doi.org/10.5281/zenodo.591637>.

- Martin Steinegger and Johannes Söding. Mmseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nature Biotechnology*, 35(11):1026–1028, October 2017. ISSN 1546-1696. doi: 10.1038/nbt.3988. URL <http://dx.doi.org/10.1038/nbt.3988>.
- Warren L. DeLano. The PyMOL Molecular Graphics System. <https://pymol.org>, 2002. Schrödinger, LLC.
- Haiyang Zheng, Hanfeng Lin, Adebowale A. Alade, Jingjing Chen, Erika Y. Monroy, Min Zhang, and Jin Wang. Alphafold3 in drug discovery: A comprehensive assessment of capabilities, limitations, and applications. April 2025. doi: 10.1101/2025.04.07.647682. URL <http://dx.doi.org/10.1101/2025.04.07.647682>.
- Jianyi Yang, Ambrish Roy, and Yang Zhang. Biolip: a semi-manually curated database for biologically relevant ligand–protein interactions. *Nucleic Acids Research*, 41(D1):D1096–D1103, October 2012. ISSN 1362-4962. doi: 10.1093/nar/gks966. URL <http://dx.doi.org/10.1093/nar/gks966>.
- Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. doi: 10.1109/CVPR.2016.90.
- Junyu Chen, Yufan He, Eric C. Frey, Ye Li, and Yong Du. Vit-v-net: Vision transformer for unsupervised volumetric medical image registration. In *2021 Medical Imaging with Deep Learning Conference (MIDL)*, 2021. doi: 10.48550/ARXIV.2104.06468. URL <https://arxiv.org/abs/2104.06468>.

A Appendix for Metrics

A.1 Spearman Correlation

The Spearman rank correlation coefficient (SRCC or ρ) is a nonparametric measure of rank correlation (statistical dependence between the rankings of two variables). It assesses how well the relationship between two variables can be described using a monotonic function. The Spearman correlation between two variables is equal to the Pearson correlation between the rank values of those two variables. For a sample of size n , the n raw scores X_i, Y_i (as random variables) are converted to ranks $rg(X_i), rg(Y_i)$, and ρ is computed as:

$$\rho = \frac{\sum_i ((rg(X_i) - \overline{rg(X)})(rg(Y_i) - \overline{rg(Y)}))}{\sqrt{\sum_i (rg(X_i) - \overline{rg(X)})^2 \sum_i (rg(Y_i) - \overline{rg(Y)})^2}}$$

Alternatively, if all n ranks are distinct integers, it can be computed using the popular formula:

$$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)}$$

where $d_i = rg(X_i) - rg(Y_i)$ is the difference between the two ranks of each observation, and n is the number of observations. The random variables are respectively substituted into y_i and $\hat{y}_i$ respectively as defined above.

A.2 Concordance Index

The Concordance Index (C-index or CI) is a measure of the goodness of fit for models, most commonly used in survival analysis or to evaluate the performance of a ranking model. It quantifies the fraction of all pairs of individuals whose predicted outcomes are correctly ordered. For two samples i and j , with true outcomes y_i and y_j and predicted outcomes $\hat{y}_i$ and $\hat{y}_j$: A pair is concordant if $y_i > y_j$ and $\hat{y}_i > \hat{y}_j$, or if $y_i < y_j$ and $\hat{y}_i < \hat{y}_j$. A pair is discordant if $y_i > y_j$ and $\hat{y}_i < \hat{y}_j$, or if $y_i < y_j$ and $\hat{y}_i > \hat{y}_j$. A pair is tied if $y_i = y_j$ or $\hat{y}_i = \hat{y}_j$.

The C-index is calculated as:

$$C = \frac{\text{Number of concordant pairs} + 0.5 \times \text{Number of tied pairs (in prediction)}}{\text{Total number of comparable pairs}}$$

More formally, considering all comparable pairs (i, j) such that $y_i \neq y_j$:

$$C = \frac{\sum_{i,j:y_i > y_j} (\mathbf{1}_{\hat{y}_i > \hat{y}_j} + 0.5 \cdot \mathbf{1}_{\hat{y}_i = \hat{y}_j})}{\sum_{i,j:y_i > y_j} 1}$$

where $\mathbf{1}_{\text{condition}}$ is the indicator function, which is 1 if the condition is true and 0 otherwise. The denominator counts the total number of pairs where the true outcomes are different.