Supporting DARPin binder selection through deep learning

Sebastian Salzmann *Center of Artificial Intelligence*
ZHAW School of Engineering Zuerich, Switzerland
salzmseb@students.zhaw.ch

***Abstract*—With the rise of accurate protein structure prediction models, such as AlphaFold and RosettaTR, a new area for protein design and engineering has begun. Due to the ability to generate fast and accurate hypothesis of spatial protein properties by means of a 1D sequence of amino acids alone the space of putative protein structures has witnessed a cambrian explosion. One hope is that by understanding the structure of a protein, scientists can design drugs that bind to pre-selected regions, allowing them to target specific biological pathways and reduce the likelihood of off-target effects. Finding binding regions and motifs is therefore of high importance in drug and biological assay design, which is why we explore DARPins, a class of designable protein-based antibody mimetics, and predict whether two complementary regions of a protein complex bind to each other or not. To investigate this, we create our own dataset of protein complexes and decoys and transfer them into feature augmented graph structures. Subsequently we extract the binding region of the complex by a distance threshold and use geometric deep learning for binding site classification on the subgraphs which contain the protein-protein interaction space. We find that 2-hop-subgraphs, using the defined interaction atoms as seed, together with atom element labels and bond information manage to represent the binding region of the protein complex sufficiently to achieve 88% accuracy on the training set using a Graph Convolutional Network with a sum operation aggregation and a binary classification output. On our test set, consisting of 12 DARPin complexes and 12 decoys, we achieve 87.5% accuracy. We further find that our protein structure graph representations seem to particularly benefit from shallow graph isomorphic convolution layers which also employ sum operations by default and train a Graph Isomorphic Network which exhibits 98.7% on the training and 91.7% accuracy on the test set.**

***Index Terms*—protein structure modelling, binding affinity, GNN, molecular fingerprint, DARPin**

I. Introduction

Proteins are one of the four main molecular building blocks of organic life, alongside nucleic acids (such as DNA, the building block of the genetic code), lipids (also known as fats), and carbohydrates (also known as sugars). Each of these molecule classes plays a unique and important role in biology. Proteins and DNA in particular are closely related since the latter provides the one dimensional blueprint encoding the sequence of individual amino acids (=residues) which chained together make up the primary structure of proteins.

In proteins the secondary structure refers to the local folding of the chain, which can take the form of alpha helices or beta sheets. The tertiary structure is the overall three-dimensional shape of the protein, and is determined by the folding of the secondary structure elements and the interactions between the amino acid side chains. The quarternary structure refers to the arrangement of multiple protein subunits that make up a protein complex.

In 2020, the scientific community studying protein structures was amazed when DeepMind introduced a computational method for predicting the folding of chained 1D amino acid sequences to 3D protein structures with unprecedented accuracy, providing a solution for the long-standing protein folding problem [1], [2].

This remarkable achievement is due to the combined efforts of the scientific community, who collected atomic resolution structures through extensive experimental procedures, computational advancements in the field of deep learning such as transformers and the work of DeepMind [3]. Since 2020 the number of predicted protein structures has increased to around 220 million with AlphaFold2 and to more than 600 million with Meta‘s newer language model based approach [4]. As, admittedly unfair, comparison it took the scientific community 60 years to collect the first 180,000 structures.

While there is always room for optimization of the current structural predictions as they are not perfect and still lack accuracy depending on the sequence input, the most exciting avenue for exploration is expected to be in questions arising from the life sciences. Form and function are believed to be closely related in biology, and given the importance of proteins for all organic life, determining function given the form is one of the most promising and useful avenues to explore. A central hypothesis here is that in proteins a similar function displays similar surface or interaction patterns [5], [6].

Protein functions however are difficult to characterize, especially in their quarternary structure. They are often organized in various unit and subunits (=chains) forming a so called complex together and also display dynamic behavior such as conformational changes upon interaction with other molecules. Their function is governed by interacting sites which are regions on a protein that can bind to other target molecules such as proteins, nucleic acids or small molecules.

Drug design and discovery in particular are interested in the functional aspects of proteins since many drug targets are proteins [9]. Investigating possible interactions therefore is already part of the rationale design of drugs [10]. Current drug development remains time and cost consuming [11]–[13]. To bring a new drug to the market the estimated costs range from 314 million to 2.8 billion, a timeline of up to 15 years and around 90% failure rate from Phase 1 clinical trial to drug admission [14], [15].

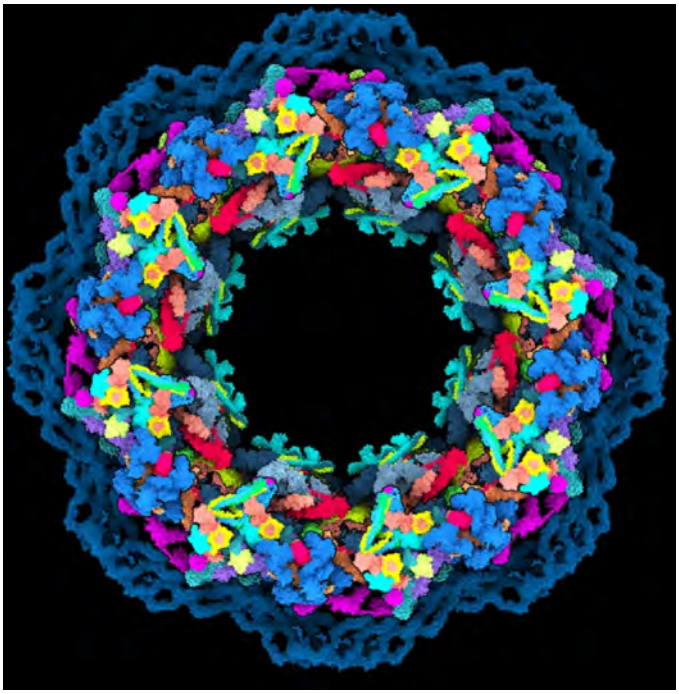


Fig. 1: The nuclear core complex, a protein jigsaw puzzle. It is the largest molecular machine in the human body consisting of more than 30 different protein subunits and 1000 protein units in total. AlphaFold2 helped in completing the puzzle to two thirds, twice as complete as before. Credit: Agnieszka Obarska-Kosinska [7], [8]

Structure-based approaches in drug selection are increasingly being used with Nirmatrelvir as a recent example, an antiviral Covid 19 drug developed by Pfizer, which was strongly supported by structural data of the Covid spike protein [10].

3D structures are thereby providing valuable insight into drug and target mechanisms and may shorten the design-make-test-analyse cycle (DMTA) in drug discovery [12], [16].

The DMTA uses data and hypothesis based approaches for designing and developing molecular candidates for subsequent hypothesis testing.

DARPins(=designed ankyrin repeat proteins) are a class of protein antibody mimetics [17]. They possess self compatible repeats with variable surface residues which can be designed and adapted. Furthermore they have favorable structural properties such as being relatively rigid in their conformation,aggregation resistant, which reduces toxicity to organisms and target binding affinities that compare and even surpass monoclonal antibodies [18]. Due to their modular nature, as they possess variable regions which can be designed through amino acid sequence variations, different DARPin variants can bind to entirely different targets and even be used for multispecfic binding. For example a recently developed dimeric DARPin (consisting of two different darpin units), selected via directed evolution and rational design, was reported to effectively eliminate Shiga toxin. Upon binding to two different of the toxin‘s subunits via the interface region, the conformational changes induced neutralizes the compounds toxicity [19].

The experimental process to select a DARPin for an appropriate target can be done via multiple in vitro assays. Ribosome display for example uses a DNA library of various DARPin sequences to transcribe the DNA to mRNA and translate the mRNA into a folded protein [20]. In the display assay, in contrast to a Ribosome‘s normal mode of action, protein and mRNA are hindered from release and stay attached to each other. This favors stability, hinders degradation and couples the folded protein structurally to its mRNA [20]. The produced complexes of mRNA-Ribosome-DARPin are subjected to binding assays where complexes displaying the desired properties are selected and through mutagenesis of the mRNA new variants of the found binder can be used to undergo another essay cycle. The whole process is termed directed evolution as it mimics an evolutionary process where proteins are undergoing a user defined selective pressure.

The process is time and money consuming since it takes trained laboratory personal to perform the assays and usually consists of many cycles before an appropriate candidate can be selected.

Shortening the selection process therefore targets a part of the DMTA cyle and lowers costs. With 3D structures now so readily available it only makes sense to explore this new wealth of data in order to gain better predictions and understanding of binder-target interactions which are instrumental in protein based therapeutics supporting rationale design in the search of an appropriate target epitope [21].

A type of computational modelling approach which is often named in one breath with molecules are graphs and graph neural networks (GNNs). GNNs are a type of machine learning model that use data represented as graphs, which consist of nodes and connecting edges, and learn from their relationship by passing messages between nodes in order to extract structural information that is deemed relevant.

Since chemical structures of molecules can be represented as graphs with atoms as nodes and chemical bonds as edges, the similarity in representation makes them easier to reason with. Additionally molecular properties such as electrostatic forces or hydrophobicity, representing non-euclidean attributes, can be attached to the nodes as features, while edges can carry node-node distance information and thereby contain positional information. Due to these favorable characteristics GNNs have been used to predict a wide range of molecular properties, such as drug-likeness [14], toxicity [22], and binding affinities [23] and have been proposed as a possible method for molecule epitope scoring in molecular docking problems. [21]

In recent years, graph neural networks (GNNs) have gained increasing attention and new methods have been developed for various fields such as chemistry, physics and neuroscience [24]–[26]. While the first GNNs aimed to expand convolutional neural networks [27] more recent developments are the addition of gating and attention mechanisms [28] as well as the introduction of new architectures which are able to deal with isomorphic graphs or operate on pointclouds [29], [30]. Graph isomorphism intuitively describes two graphs that are identical except for the labels. They therefore have the same number

of nodes, edges and edge connectivity [31] yet GNNs fail to distinguish simple isomorphism which becomes particularly important in graph classification tasks [29].

Graph classification in conjunction with proteins or biomolecules in general is an important task to accomplish since we often want to predict a particular whole graph/molecule property such as binding affinity or toxicity.

A. Related Work

Various computational approaches have been proposed for protein binding predictions.

They are varied in their nature though usually produce scores receiving different binder-target conformations as input, sampled from the space of possible conformations [32], [33].

Radom et al. [21] describes a computational approach for DARPin-target docking modelling which works in a semi-automated fashion and mainly uses available filtering functions in the Rosetta [34] and ClusPro software [35]. This approach however, while successful for the selected DARPins, is time consuming due to its semi-automated nature, relies on hand-crafted experimental settings and omits difficult targets knowingly. Neural net approaches have therefore also been proposed for ameliorating time lines, due to their fast inference, as well as their ability to deal with high dimensional data [21].

In the domain of neural networks McNutt [36] and Ahmet et al. [37] proposed two different Convolutional Neural Network (=CNN) approaches for predicting binding affinity though focus on protein-ligand predictions and not protein-protein ones.

In the domain of GNNs Nguyen et al. [23] proposed a model for predicting protein-ligand binding affinity and find that their model predictions worked better than non-neural approaches.

Wang et al. [38] however are the closest to our proposed research since they also focus on protein-protein interactions using GNNs. They manage to surpass their own, previous CNN approach by making the GNN model focus on the interface region of the protein complex.

As we explore a very specific class of proteins and want full control over the data input we therefore build a dataset of proteins using publicly available data from the Protein Data Bank, transform it into feature augmented interface region graph data and apply Graph Neural Network variants to perform a classification task, discriminating between binding and non-binding protein regions. Our performance on DARPin binding predictions is always directly evaluated since our test set is made up of DARPin-Target structures only.

II. Experimental design and outcomes

A. Experimental outline

The procedure for dataset generation can be described as follows:

1) Train and test set definition
2) PDB/Structure File download and preprocessing
3) Decoy Generation
4) Feature augmentation of nodes (atoms) and edges(bonds)
5) Subgraph and 2-hop-subgraph generation
6) Transfer of subgraph and 2-hop-subgraph into Pytorch-Geometric format

B. Training and Validation set

The decision which proteins to include into the training and validation set was performed by querying the Protein Data Bank (=PDB) with the following parameter:

- Refinement atomic resolution of maximally two Angstrom
- Protein as chain entity type (as oposed to DNA or metabolites)
- Consisting of two chains

The resulting IDs were used for downloading and preprocessing of the PDB files.

Since we rely on the PDB ID or PDB File alone the input of the workflow can therefore be expanded to all artificially generated structures as well as all structures in the Protein Data Bank.

C. Test set

As the test set 6 PDB IDS (one did not generate a decoy) of existing DARPin complexes from the publication of Radom et al. [21] were taken as well as 6 additional PDB IDs that were found in a PDB query. The test set therefore serves as direct assessment of a prediction for a DARPin binding site. The IDs are listed in appendix G.

D. PDB file preprocessing

PDB Files were loaded and ligands and water molecules removed from the structure. Hydrogen atoms were corrected using reduce and singular chains of the protein complexes were separated. Coordinates and atom elements were extracted from the given structure using Biotite [39] and saved as python objects.

E. Decoy generation

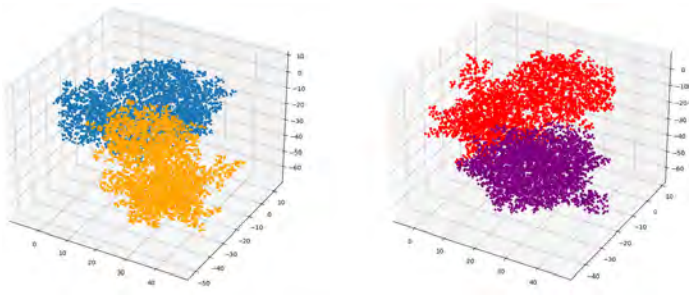
Decoys, non-existent protein structures, serve the important role of a negative dataset in protein structure assays. For investigating binding we therefore generated decoys of the original structures under the assumption that if the protein chains of the protein complex are rotated randomly a non-binding protein structure is constructed.

The two protein chains were randomly rotated as regards rotation angle and rotation axis (x,y or z). Additionally for a decoy to be accepted the number of interaction edges had to match $\pm$ 20% of the number of real interaction edges defined by the distance of 4 Angstrom of each Atom of one chain to any atom of the other chain.

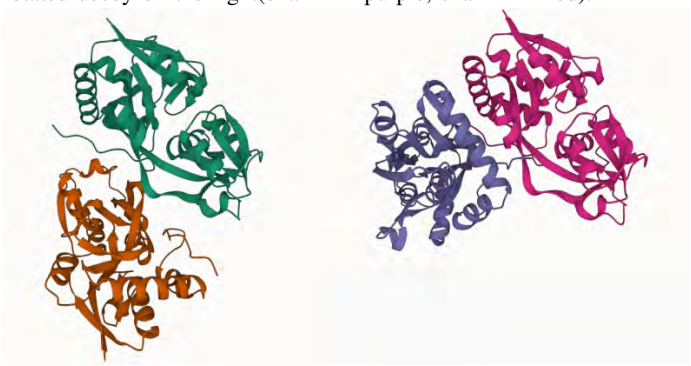
In case no appropriate decoy was found in 2000 random rotation iterations, chains did not display any interaction edges at the given distance or less than two chains were present the structures were discarded.

The chosen parameters were selected in order to create an appropriate, non-binding protein dataset which displays roughly the same number of (non-binding) interaction sites

with similar distance. Sterical clashes are less frequent since the number of interactions indirectly influences the relative positions (eg. if one protein rotates into the complementary one it produces many more interaction sites while violating the natural laws of physics) though cannot be excluded since not all data was checked visually. A display of decoy generation is shown in Figure 2.



(a) Original protein on the left (chain1 in yellow, chain2 in blue) and rotated decoy on the right(chain1 in purple, chain 2 in red).



(b) Original protein (chain 1 in turquoise, chain 2 in orange) and rotated decoy (chain 1 in red, chain 2 in blue). Chain 1 is kept in the same rotational conformation to make the relative change in position to chain 2 visible.

Fig. 2: Display of decoy generation by rotation. The displayed protein has the PDB ID 6A80, a transporter protein.

1) Decoy RMSD: The root mean squared distance (=RMSD) is a common measure of protein structure similarity.

The table I and II display the statistical description of the train and test dataset as regards the RMSD.

Measure	Value
Count	667
Mean	23.3
Std	8.1
Min	0.5
Max	66.3

TABLE I: Statistical RMSD description of the training set

Measure	Value
Count	14
Mean	18.6
Std	6.4
Min	9.5
Max	26.8

TABLE II: Statistical RMSD description of the test set

Decoy generation was not considering the RMSD which represents also a good parameter to measure similarity and when included as threshold value for decoy acceptance can help in appropriate dataset generation such as for example generating near native solution structures or very dissimilar structures.

F. Node and edge features

In order to bridge communication between the graph and protein world atoms from now on will be referred to as nodes and the bonds between them as edges. Edges were modelled as undirected. The following node and egde attributes were collected.

Name	Edge or Node	Source
Hydrophobicity	Node	Kyte Doolitle scale
Chemical Embeddings	Node	dMasif
Binding-Factor	Node	dMasif
Atom radius	Node	Wikipedia (Atomic radius)
Atom element	Node	PDB File
Residue indices	Node	PDB File
Atom coordinates	Node	PDB File
Covalent Edge	Edge	PDB File
Interaction Edge	Edge	Computed
Distance	Edge	Computed

TABLE III: Overview of node and edge features

Features were normalized across all samples where necessary.

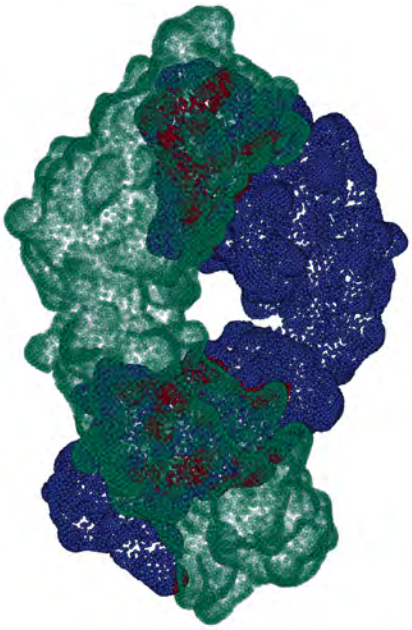
G. dMasif node feature augmentation

Dmasif is a geometric deep learning model that generates chemical and abstract feature embeddings as well as a binding factor prediction of protein point clouds and thereby can provide useful fingerprints in a fast and automated way. The original code was dockerized and features were generated for all proteins. The method produces a pointcloud with various features assigned to each point and therefore does not represent atoms directly. In order to assign the features to our atom nodes we computed for each point the closest atom and summed up the feature properties. For detailed parameters for running the model, some visual examples and the methods‘s mode of action please refer to the appendix C and the original masif and dmasif publications [5], [6].

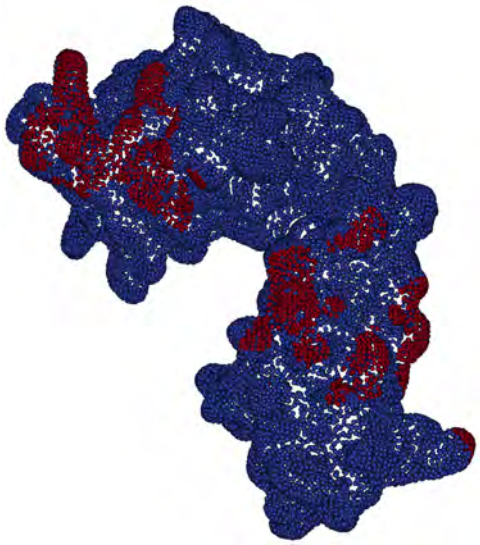
H. Interaction site definition

Interaction sites (=binding region or interface region) were defined as two chains that are within a threshold of 4 Angstrom distance of each other. All atoms of one chain within 4 Angstrom of the other chain were therefore labelled as interaction nodes/atoms.

Covalent bindings, representing edges between nodes, were defined as one binding class and saved as edge attributes. Interaction edges between interaction nodes were labelled as a second class.



(a) Bound complex of the 1A4K protein. The green, transparent protein chain is overlayed above its interaction sites (red) with the second chain (blue).



(b) One protein chain of the 1A4K protein with its interaction sites in red.

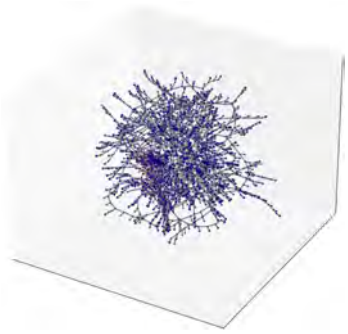
Fig. 3: Interaction sites visible in red in (a) the bound complex and (b) the single chain of the 1A4K protein complex.

I. Subgraphing of protein interfaces

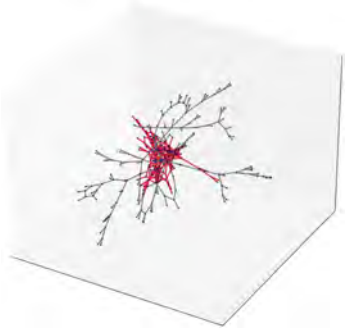
To focus the models attention on the interface regions of the binding subgraphs of the interaction site interface were extracted in two different ways.

For this the pytorch geometric subgraph and k-hop-subgraph using two hops was used.

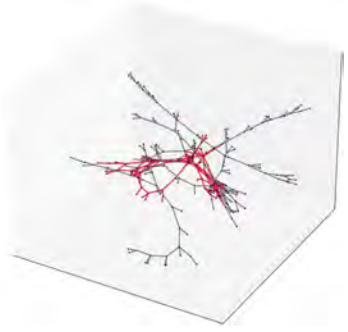
The subgraph functions returns a subgraph containing only the interaction atoms, while the k-hop-subgraph functions returns the interaction atoms and all atoms reachable via edges in two hops. The result of this operation is visible in Figure 4 and intuitively shows what a standard graph convolutional neural network receives as input when only considering nodes and edges.



(a) Display of the full graph with its nodes ($n_n = 3444$) and edges ($n_e = 7410$, grey=covalent, red=interaction) in the 4HRN DARPin-protein complex.



(b) Display of the 2-hop-subgraph of the 4HRN DARPin-protein complex with its nodes ($n_n = 350$) and edges ($n_e = 1160$, grey=covalent, red=interaction).



(c) Display of subgraph edges of the 4HRN DARPin-protein complex with its nodes ($n_n = 260$) and edges ($n_e = 724$, grey=covalent, red=interaction).

Fig. 4: Display of different graph structures without 3D positional information and n_n as the number of nodes and n_e as the number of edges. Interaction edges are displayed in red, covalent edges in grey.

The generated subgraphs and 2-hop-subgraphs were combined with the node and edge feature additions for all original structures and decoys and the two datasets were saved on the disk as PyTorch Geometric Data object.

A more detailed description of the Data objects and the parameter assignments can be found in section B of the appendix as well as the node and edge feature overview in table III.

J. Deep Learning Approach

All deep learning models used stem from the Pytorch Geometric library. As modelling framework we used PyTorch Lightning and Wandb for metric monitoring and model selection. For training and validation the two generated datasets (subgraph and 2_hop_graphs) were each split in a 80-20 split ratio and the test set was defined as the set of DARPin-protein complexes as described in section II-C. Random seed was set to 42 and batchsize was tuned for each dataset according to the maximum fit into memory. For each training run the top two checkpoints based on maximum validation accuracy were saved for evaluating the test set.

1) Graph convolutional Network: The first model tried was a standard Graph convolutional network (GCN) with a binary cross entropy loss distinguishing between binding (real protein) and non-binding (decoy protein) protein partners/chains. For a display of the detailed architecture and hyperparameters refer to the Appendix section D1. For the full graph dataset the graph convolution operations would result in a Cuda memory error which is why the were not used for the Graph Neural Networks.

We first compared the difference in performance on both the subgraph and 2-hop-graph visible in Table IV. When varying the node features some resulted in a decrease, some in increased training stability (data not shown). Further ablation experiments should be conducted though to evaluate the usefulness of each of the features. Th information that did matter nevertheless was giving node labels in terms of atom element labels. The following experiments were therefore only conducted with the atom element labelling.

Dataset	Dataset part	Accuracy
Subgraph	train	83%
Subgraph	test	74%
2-Hop	train	**89**%
2-Hop	test	**79**%

TABLE IV: Comparison of subgraph and 2-hop-subgraph with mean aggregations

2-hop-subgraph showed better performance than the subgraph methods. An additional benefit of using k-hop-subgraphs is that the hop parameter provides a tunable parameter which can be used to extract variably sized regions around the interaction site. The variable graph size together with the batch size also determines the memory usage of the training which can thereby also be influenced. Further experiments using higher hop numbers under consideration of memory usage should therefore be conducted in the future.

Due to the performance increase we conducted all following experiments with the 2-hop-subgraph dataset only.

The most prominent difference in performance as regards the architectural choices with GCNs was achieved with the addition of the sum-pooling method. The global-add-pool operation, in contrast to mean or maximum, is used to generate graph-sized embeddings which are then forwarded to the classification output.

For a comparison of mean, max and sum aggreations please consult the Table V.

Method	Dataset part	Accuracy
Mean	train	**89**%
Mean	test	79%
Max	train	85%
Max	test	71%
Sum	train	88%
Sum	test	**87**%

TABLE V: Comparison Mean Max and Sum Aggreation Operations in a GCN

This is seen as one of the strengths of the summation since we expect this to perform better on graph classification. For a more detailed explanation see this excellent explanation on injectivity and graph isomorphism.

While the final performance improved with the summation the learning and loss metrics were much more unstable and without selecting the checkpoints on the two best validation accuracy performances this result might have been missed. The curves can be inspected in the appendix E.

2) Graph Isomorphic Networks: As noted in the article (see the link II-J1) about injectivity standard GCN architectures are not able to distinguish simple graph structures. Xu et al. [29] therefore proposed the graph isomorphic architecture (=GIN) to overcome this limitation. Technically it works by using the same summation method as above with addition of concatenating the embeddings of different layers and an adapted message passing system. For an overview of the architecture please have a look at the appendix D2.

Interestingly only a combination of the more shallow hidden layer embedding size of 50 is able to surpass GCN performance (see Table V) with the summation method. The results are displayed in Table VI. The training exhibited more stable curves when compared to the GCN as visible in the appendix F.

Dataset part	Hidden layer Size	Accuracy
train/val	100	94/91%
test	100	79 %
train/val	50	**97/94**%
test	50	**92**%

TABLE VI: GIN Results when using different layer sizes.

The lower hidden layer sizes helped in the test dataset performance while the higher layer size model possibly exhibits overfitting which could be counteracted with more regularization.

III. Discussion

The results of this study outline a complete, automatable process going from Protein Databank IDs to protein interface graph structures while adding and computing various node and edge features. We welcomed very much the flexibility that this approach offers since the modularity of the preprocessing allows tuning parameters such as decoy RMSD cutoffs, interaction distance threshold, subgraph size and thereby adaption to memory restrictions in resource constrained settings as well as creation of balanced or unbalanced datasets at wish.

Our results indicate that Graph Neural Networks (GCN and GIN) can learn subgraph embeddings of protein interface sites

and distinguish them well from our artificially constructed decoys, which are the same protein with alternative interaction sites yet similar number of interactions. Applied on a whole different set of proteins, our DARPin test set, we achieve comparable results exhibited in training and validation accuracy. This is not entirely natural since apart from atomic resolution and that the structural entities are proteins consisting of two chain units no similarity measure with the DARPin set influenced our PDB query. The GNNs during training therefore seem to learn either sub or whole structural protein embeddings useful for discriminating between the classes and applicable to proteins in general. This by itself would need more investigation but might lead to more sophisticated initial dataset construction. If protein structures in the interface exhibit certain pattern it could allow selection of protein interfaces which possess certain structural motifs. This would permit fine tuning of what the graph model receives as input which is essential for the data-centric approach that neural networks are and might influence down-stream performance. Tailoring the dataset in general is a nice options to have since it allows posing different research questions without changing much of the boilerplate architecture and thereby allows quick adaptation to specific protein classes such as our DARPins.

The approach outlined by Wang et al. [38] is close to our work. Nevertheless they used a different modelling approach by using a siamese graph neural network which they hypothesize, through graph embedding substraction, can focus on the interface region. Our approach of focus on the other hand lies with our subgraphing technique. By initializing the atoms defined as interaction nodes as seed nodes together with building interaction edges we allow the graph structure to grow step wise in size with the interface regions as starting point (see Figure 4). Furthermore they use different model layer mechanisms (Attention and Gate Augmentation) as well as different prediction targets since they are focusing on docking scores, a regressional task. An interesting comparison therefore would be to use the datasets (DOCKGROUND and CAPRI Score dataset) described in the publication and change our classification output to a regressional one while keeping the rest of the architecture the same. Another interesting observation is that Wang et al. also chose a summation operation in their GAT layer which is in accordance with our findings that sum operations enhance performance on whole graph classification. Theory agrees that aggregation functions play an important role in the network‘s representational power and performance [29]. According to Xu mean aggregation captures the distribution of elements, max aggregation proves to be advantageous to identify representative elements, and sum aggregation enables the learning of structural graph properties [29]. Trying out the proposed GAT layer by Wang et al. or even combining it with our shallow GIN model could be an avenue to explore.

Another recent research study on protein interfaces that is related to our work was performed by Jha et al. [40] who, similar to Wang, used GAT and GCN networks. Nevertheless they perform prediction on amino acid residue level (in contrast to atom level as we do) and use language models to create per residue embeddings which they attach to their nodes as features. They therefore provide an interesting approach as regards the node feature augmentation which could serve as a valuable avenue to explore. Furthermore they use a classification output for binding and non-binding where they achieve 98% on the human PPI dataset which is above the reported performance with our dataset. A direct comparison in performance would have to be made though using the same dataset and using the residues as nodes. Furthermore we have to keep in mind that our test dataset might stem from a different distribution of proteins since training and validation set come from the PDB query while our test set contains the DARPins and served as a direct measure to tackle our research question, to ameliorate rational design in the context of DARPin binding prediction. Whether the named datasets are controlled as regards certain data distribution aspects (as for example protein structure variety) should also be looked at. In case of absence we suggest to just use the excellent Protein Databank Query to narrow down the search to proteins with the desired properties and simply provide them as PDB IDs to our workflow while decoys can either be generated or used from other sources.

In contrast to our findings as regards pooling method Jha et al. choose a mean pooling aggregation though lack to explain why or show performance data. Nguyen et al. [23], who also model drug-target interactions as graphs and then predicts binding affinities, in contrast to us reports the best performance with a max pooling method. They also tried out the GIN layer but reported a less good performance though it remains to mention that only the right combination of hidden layer size resulted in our best performant GIN model. As stated above theory agrees with our findings in favor of summation yet in practice different methods seem to achieve the best performance.

All in all our performance seems comparable yet is difficult to evaluate. without a direct comparison on the same dataset.

A. Limitations

There are various limitations in this work. Many parameters, like interaction distance thresholds, decoy acceptance or the hop size play an important role in the resulting final subgraph structure and therefore should be more thoroughly looked at.

Decoy generation also should be evaluated critically and thought should be put into a good set of metrics that then governs the artificial structure generation. It would need to be excluded that the generated decoys exhibit for example very obvious non-binding characteristics which made it easy for the model to distinguish them. At the same time can any decoy produced without not entirely be excluded from exhibiting binding properties. Decoy examples were visually looked at to exclude strong sterical clashes but most of the files remain uninvestigated. When looking at the RMSD statistics in Table II and I the minimum value shows 0.5 Amstrong for the training set which means that at least one decoy has basically the same structure as the target molecule which might make it very difficult, if not impossible to distinguish for a discriminative model. Most structures are around 25 Armstrong difference which is a considerable shift from the

original position. The RMSD at the same time provides a good control mechanism on how different the decoy should be and can serve to test where the model‘s limits are. For example an interesting approach would be to generate a variety of RMSD similarity range datasets (eg. from 0-10 Amstrong, 20-30,... etc.), test the model‘s limits as regards its discriminatory power and investigate which architectural changes would need to be implemented work in that regime or whether other additions are needed.

Another limitation not touched in this work are protein dynamics and conformational changes that can happen upon binding. Proteins in reality behave like living structure that move and twist around and are also subject to weak and indirect interactions which are hard to measure in dynamic settings. DARPins are in that sense favorable since they are quite rigid and their paratope (binding region) is known. As regards their target however we can only assume behavior by for example looking at the chain sequence statistics which can have parameters like occupancy, describing the flexibility of a certain region. In docking problems the usual approach is to generate many structures sampled from the space of possibilities (homology modelling) defined by the flexible parts [21]. This creates the real life scenario of having many decoys and hopefully some near native structures (1-2 Armstrong) which usually do not score best [21]. Testing the discriminatory limits of our subregion approach in high decoy number setting we therefore regard as essential for benefitting in DARPin-target binding assessments.

B. Considerations for future work and conclusion

Protein docking remains a difficult problem despite the newly available wealth on structural data. A recent survey [32] found that current molecular docking predictions when applied to protein-ligand interactions exhibit weak performance yet can be improved with machine learning rescoring approaches. While their findings cannot be directly transferred for protein-protein interactions it is evident that protein dynamics and physics remains a topic where active research is needed. We still think that the future shines bright in the space of protein structures and the proximal fields of research. Similar to how advances in sequencing technology, in particular in cost reduction, amplified and fueled genetic research, accurate computational protein structure predictions might do the same for this second, highly important bio-molecule.

As regards future improvements on our work the possibilities are manyfold though some are already in close reach.

Our ”from scratch” approach as regards the dataset construction permits changing the base dataset completely with modification of the PDB query/PDB IDs input alone. Thanks is also owed here to the excellent Pytorch Lightning framework that due to its modular nature allows simply writing another datamodule which can be plugged interchangeably into the existing code. With addition of some minor extra modifications (such as a quality control module, which for example sets a confidence cutoff) it would also allow computational structures from for example AlphaFold2 as input. At the same time the incorporation of other benchmark datasets used in other publications is not far from reach.

Furthermore our current approach only uses atom element labels and their connectivity without any further information such as positional atom information or the different edge labels (covalent and interaction) defined. While it is on one hand fascinating that a neural network can possibly distinguish these structural differences, the wealth of information present as regards node and edge features is far from explored and leaves room for a wide range of further experimentation and feature addition.

Another easy addition would be to use the GINE layer, a GIN layer which permits edge attributes. The edge attributes in our case contain interaction edge labels which would give the model information about the type of binding that occurs between atoms yet can be adapted to any other information suitable.

Besides edge attributes there is also the whole domain of positional information that has not been used but is present in the data. dMasif [6] for example is a pointcloud based protein feature prediction model which provides ample possibilities of ablation experiments to find out whether certain of its feature embeddings can aid in our prediction tasks. While we found the features we used arbitrariliy to stabilize training in certain cases (data not shown) we did not see an increase in performance. Though since the predictions by the model were quite impressive when inspected visually (see Appendix C) we believe that there still many things to try out.

An interesting exploration would also be in terms of going back into biology and try to find an explanation of why certain structures fail to be classified correctly. Are there structural motifs that are special to them or do they possess properties which make them different? Why do the models fail and if the structural motifs are present in the initial dataset how can we make sure that predictions take these more difficult cases under consideration? Attention mechanism come to mind since it essentially provide a weighing mechanism assigning importance.

Our initial research question of whether DARPin selection can be improved by deep learning might be too early to answer since theoretical results will always have to be looked at in practice. A next step could be to design an even more DARPin centric approach. Models could be trained on a variety of proteins, computationally produced or not, and then fine tuned to a large DARPin library, consisting of many DARPin variants. Here data from the Ribosome display could be a valuable source of input. The assay outputs approximate binding properties as well as mutated sequence variations of the same initial DARPins. Therefore differences in binding properties could possibly be correlated to structural differences. Tricky is that the variant structures can not be so easily produced since for example AlphaFold2 relies on evolutionary conservation methods and therefore is not expected to work well with single mutational changes introducing strong structural changes which is also listed as an official limitation. There are however efforts to take this into account and Akdel et al. [41] for example couple the structure predictions with additional mutation-analysis algorithms for increased accuracy of the mutated structures. Weissenow et al. [42] on the other hand propose a more protein-specific

Convolutional Neural Network based approach rather than AlphaFold's family averaged prediction which fare better in mutational experiments. For DARPins certainly an exciting thing to try.

Experimental binding affinity data in general can be beneficial since it allows to change the classification task into a regression one and using only minor changes converting our classification model into a docking scorer.

The ultimate system of DARPin selection would probably act in many dimensions. Taking in the target information alone it would preselect and determine the best possible binders for it or even dream up new possible ones. Network hallucination, a term coined for protein networks powered by the same embeddings learned from the folding task, are already generating entirely new structures that were confirmed to be functional [43]. Realizing that these trained neural network embeddings contain a broad amount of information about protein structures we can only guess of what else could be done with them if steered into the right direction.

In the end we believe that the complexity and dynamics of these living protein structures and the questions we are posing are so complex that it might be hard to cover them by one model or technique alone. In reality multiple modular sub networks, heuristics and (classical) algorithms specialized for a particular subtask will need to work in ensemble. Large deep learning architectures like AlphaFold2 exhibit these characteristics which to create require interdisciplinary teams, research infrastructure in terms of curated databases, detailed experiment monitoring, computational resources, as well as extensive experimentation and engineering efforts. Deep learning shines in high data regimes and with new combinatorial architectures like stable diffusion already making their way into structural molecular biology generating molecules [44] like they were images we believe that this is just the tip of the iceberg of what can be accomplished in the coming decades in structural protein research.

REFERENCES

[1] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, A. Bridgland, C. Meyer, S. A. Kohl, A. J. Ballard, A. Cowie, B. Romera-Paredes, S. Nikolov, R. Jain, J. Adler, T. Back, S. Petersen, D. Reiman, E. Clancy, M. Zielinski, M. Steinegger, M. Pacholska, T. Berghammer, S. Bodenstein, D. Silver, O. Vinyals, A. W. Senior, K. Kavukcuoglu, P. Kohli, and D. Hassabis, "Highly accurate protein structure prediction with alphafold," *Nature 2021 596:7873*, vol. 596, pp. 583–589, 7 2021. [Online]. Available: https://www.nature.com/articles/s41586-021-03819-2

[2] R. Evans, M. O'Neill, A. Pritzel, N. Antropova, A. Senior, T. Green, A. Žídek, R. Bates, S. Blackwell, J. Yim, O. Ronneberger, S. Bodenstein, M. Zielinski, A. Bridgland, A. Potapenko, A. Cowie, K. Tunyasuvunakool, R. Jain, E. Clancy, P. Kohli, J. Jumper, and D. Hassabis, "Protein complex prediction with alphafold-multimer," *bioRxiv*, p. 2021.10.04.463034, 3 2022. [Online]. Available: https://www.biorxiv.org/content/10.1101/2021.10.04.463034v2https://www.biorxiv.org/content/10.1101/2021.10.04.463034v2.abstract

[3] S. K. Burley and S. O. Burley@rcsb, "Impact of structural biologists and the protein data bank on small-molecule drug discovery and development," *Journal of Biological Chemistry*, vol. 296, p. 100559, 1 2021. [Online]. Available: http://www.jbc.org/article/S0021925821003379/fulltexthttp://www.jbc.org/article/S0021925821003379/abstracthttps://www.jbc.org/article/S0021-9258(21)00337-9/abstract

[4] E. Callaway, "Alphafold's new rival? meta ai predicts shape of 600 million proteins," *Nature*, vol. 611, pp. 211–212, 11 2022.

[5] P. Gainza, F. Sverrisson, F. Monti, E. Rodolà, D. Boscaini, M. Bronstein, and B. Correia, "Deciphering interaction fingerprints from protein molecular surfaces using geometric deep learning," *Nature Methods*, vol. 17, no. 2, pp. 184–192, 2020.

[6] F. Sverrisson, J. Feydy, B. E. Correia, and M. M. Bronstein, "Fast end-to-end learning on protein surfaces," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 15 267–15 276, 2021.

[7] P. Fontana, Y. Dong, X. Pi, A. B. Tong, C. W. Hecksel, L. Wang, T. M. Fu, C. Bustamante, and H. Wu, "Structure of cytoplasmic ring of nuclear pore complex by integrative cryo-em and alphafold," *Science*, vol. 376, 6 2022.

[8] T. U. Schwartz, "Solving the nuclear pore puzzle:using a battery of tools, the architecture of the nuclear pore complex is revealed," *Science*, vol. 376, pp. 1158–1159, 6 2022.

[9] R. Santos, O. Ursu, A. Gaulton, A. P. Bento, R. S. Donadi, C. G. Bologa, A. Karlsson, B. Al-Lazikani, A. Hersey, T. I. Oprea, and J. P. Overington, "A comprehensive map of molecular drug targets," *Nature Reviews Drug Discovery 2016 16:1*, vol. 16, pp. 19–34, 12 2016. [Online]. Available: https://www.nature.com/articles/nrd.2016.230https://www.nature.com/articles/nrd.2016.230/

[10] D. R. Owen, C. M. Allerton, A. S. Anderson, L. Aschenbrenner, M. Avery, S. Berritt, B. Boras, R. D. Cardin, A. Carlo, K. J. Coffman, A. Dantonio, L. Di, H. Eng, R. A. Ferre, K. S. Gajiwala, S. A. Gibson, S. E. Greasley, B. L. Hurst, E. P. Kadar, A. S. Kalgutkar, J. C. Lee, J. Lee, W. Liu, S. W. Mason, S. Noell, J. J. Novak, R. S. Obach, K. Ogilvie, N. C. Patel, M. Pettersson, D. K. Rai, M. R. Reese, M. F. Sammons, J. G. Sathish, R. S. P. Singh, C. M. Steppan, A. E. Stewart, J. B. Tuttle, L. Updyke, P. R. Verhoest, L. Wei, Q. Yang, and Y. Zhu, "An oral sars-cov-2 mpro inhibitor clinical candidate for the treatment of covid-19," *Science*, vol. 374, pp. 1586–1593, 12 2021. [Online]. Available: https://www.science.org/doi/10.1126/science.abl4784

[11] K. Smietana, M. Siatkowski, and M. Møller, "Trends in clinical success rates," *Nature Reviews Drug Discovery*, vol. 15, pp. 379–380, 6 2016.

[12] "Rethinking drug design in the artificial intelligence era," *Nature reviews. Drug discovery*, vol. 19, pp. 353–364, 5 2020. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/31801986/

[13] A. Mullard, "2020 fda drug approvals," *Nature reviews. Drug discovery*, vol. 20, pp. 85–90, 2 2021.

[14] D. Sun, W. Gao, H. Hu, and S. Zhou, "Why 90fails and how to improve it?" *Acta Pharmaceutica Sinica B*, vol. 12, pp. 3049–3062, 7 2022.

[15] O. J. Wouters, M. McKee, and J. Luyten, "Estimated research and development investment needed to bring a new medicine to market, 2009-2018," *JAMA*, vol. 323, pp. 844–853, 3 2020. [Online]. Available: https://jamanetwork.com/journals/jama/fullarticle/2762311

[16] "Chemical predictive modelling to improve compound quality," *Nature Reviews Drug Discovery 2013 12:12*, vol. 12, pp. 948–962, 11 2013. [Online]. Available: https://www.nature.com/articles/nrd4128

[17] P. Forrer, M. T. Stumpp, H. K. Binz, and A. Plückthun, "A novel strategy to design binding molecules harnessing the modular nature of repeat proteins," *FEBS Letters*, vol. 539, pp. 2–6, 3 2003. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/12650916/

[18] A. Plückthun, "Designed ankyrin repeat proteins (darpins): binding proteins for research, diagnostics, and therapy," *Annual review of pharmacology and toxicology*, vol. 55, pp. 489–511, 1 2015. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/25562645/

[19] Y. Zeng, M. Jiang, S. Robinson, Z. Peng, V. Chonira, R. Simeon, S. Tzipori, J. Zhang, and Z. Chen, "A multi-specific darpin potently neutralizes shiga toxin 2 via simultaneous modulation of both toxin subunits," *Bioengineering*, vol. 9, p. 511, 10 2022. [Online]. Available: https://www.mdpi.com/2306-5354/9/10/511/htmhttps://www.mdpi.com/2306-5354/9/10/511

[20] C. Zahnd, E. Wyler, J. M. Schwenk, D. Steiner, M. C. Lawrence, N. M. McKern, F. Pecorari, C. W. Ward, T. O. Joos, and A. Plückthun, "A designed ankyrin repeat protein evolved to picomolar affinity to her2," *Journal of molecular biology*, vol. 369, pp. 1015–1028, 6 2007. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/17466328/

[21] F. Radom, E. Paci, and A. Plückthun, "Computational modeling of designed ankyrin repeat protein complexes with their targets," *Journal of Molecular Biology*, vol. 431, pp. 2852–2868, 7 2019.

[22] J. Chen, Y. W. Si, C. W. Un, and S. W. Siu, "Chemical toxicity prediction based on semi-supervised learning and graph convolutional neural network," *Journal of cheminformatics*, vol. 13, 12 2021. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/34838140/

[23] T. Nguyen, H. Le, T. P. Quinn, T. Nguyen, T. D. Le, and S. Venkatesh, "Graphdta: predicting drug-target binding affinity with graph neural networks," *Bioinformatics (Oxford, England)*, vol. 37, pp. 1140–1147, 4 2021. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/33119053/

[24] D. Duvenaud, D. Maclaurin, J. Aguilera-Iparraguirre, R. Gómez-Bombarelli, T. Hirzel, A. Aspuru-Guzik, and R. P. Adams, "Convolutional networks on graphs for learning molecular fingerprints," 9 2015. [Online]. Available: https://arxiv.org/abs/1509.09292

[25] A. Sanchez-Gonzalez, J. Godwin, T. Pfaff, R. Ying, J. Leskovec, and P. W. Battaglia, "Learning to simulate complex physics with graph networks," pp. 8459–8468, 11 2020. [Online]. Available: https://proceedings.mlr.press/v119/sanchez-gonzalez20a.html

[26] J. Vohryzek, A. Griffa, E. Mullier, C. Friedrichs-Maeder, C. Sandini, M. Schaer, S. Eliez, and P. Hagmann, "Dynamic spatiotemporal patterns of brain connectivity reorganize across development," *Network Neuroscience*, vol. 4, pp. 115–133, 2 2020. [Online]. Available: https://direct.mit.edu/netn/article/4/1/115/95802/Dynamic-spatiotemporal-patterns-of-brain

[27] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *Faculty of Informatics - Papers (Archive)*, vol. 20, pp. 61–80, 1 2009. [Online]. Available: https://ro.uow.edu.au/infopapers/3165

[28] S. Brody, U. Alon, and E. Yahav, "How attentive are graph attention networks?" 5 2021. [Online]. Available: https://arxiv.org/abs/2105.14491v3

[29] K. Xu, S. Jegelka, W. Hu, and J. Leskovec, "How powerful are graph neural networks?" *7th International Conference on Learning Representations, ICLR 2019*, 10 2018. [Online]. Available: https://arxiv.org/abs/1810.00826v3

[30] "Dynamic graph cnn for learning on point clouds," *ACM Transactions on Graphics*, vol. 38, p. 146, 10 2019. [Online]. Available: https://doi.org/10.1145/3326362

[31] A. Njanko and D. B. Rawat, "On the identification of isomorphic graphs for graph neural network using multi-graph approach," *2022 IEEE 13th Annual Information Technology, Electronics and Mobile Communication Conference, IEMCON 2022*, pp. 61–66, 2022.

[32] F. Wong, A. Krishnan, E. J. Zheng, H. Stärk, A. L. Manson, A. M. Earl, T. Jaakkola, and J. J. Collins, "Benchmarking ¡scp¿alphafold¡/scp¿ -enabled molecular docking predictions for antibiotic discovery," *Molecular Systems Biology*, vol. 18, 9 2022. [Online]. Available: https://onlinelibrary.wiley.com/doi/10.15252/msb.202211081

[33] S. Y. Huang, "Exploring the potential of global protein-protein docking: an overview and critical assessment of current programs for automatic ab initio docking," *Drug discovery today*, vol. 20, pp. 969–977, 8 2015. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/25801181/

[34] A. Leaver-Fay, M. Tyka, S. M. Lewis, O. F. Lange, J. Thompson, R. Jacak, K. Kaufman, P. D. Renfrew, C. A. Smith, W. Sheffler, I. W. Davis, S. Cooper, A. Treuille, D. J. Mandell, F. Richter, Y. E. A. Ban, S. J. Fleishman, J. E. Corn, D. E. Kim, S. Lyskov, M. Berrondo, S. Mentzer, Z. Popović, J. J. Havranek, J. Karanicolas, R. Das, J. Meiler, T. Kortemme, J. J. Gray, B. Kuhlman, D. Baker, and P. Bradley, "Rosetta3: an object-oriented software suite for the simulation and design of macromolecules," *Methods in enzymology*, vol. 487, pp. 545–574, 2011. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/21187238/

[35] D. Kozakov, D. R. Hall, B. Xia, K. A. Porter, D. Padhorny, C. Yueh, D. Beglov, and S. Vajda, "The cluspro web server for protein-protein docking," *Nature protocols*, vol. 12, pp. 255–278, 2 2017. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/28079879/

[36]

[37] A. Ahmed, B. Mam, and R. Sowdhamini, "Deelig: A deep learning approach to predict protein-ligand binding affinity," *https://doi.org/10.1177/11779322211030364*, vol. 15, 7 2021. [Online]. Available: https://journals.sagepub.com/doi/full/10.1177/11779322211030364

[38] X. Wang, S. T. Flannery, and D. Kihara, "Protein docking model evaluation by graph neural networks," *Frontiers in Molecular Biosciences*, vol. 8, p. 402, 5 2021.

[39] P. Kunzmann and K. Hamacher, "Biotite: A unifying open source computational biology framework in python," *BMC Bioinformatics*, vol. 19, pp. 1–8, 10 2018. [Online]. Available: https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-018-2367-z

[40] K. Jha, S. Saha, and H. Singh, "Prediction of protein-protein interaction using graph neural networks," *Scientific reports*, vol. 12, 12 2022. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/35589837/

[41] M. Akdel, D. E. V. Pires, E. P. Pardo, J. Jänes, A. O. Zalevsky, B. Mészáros, P. Bryant, L. L. Good, R. A. Laskowski, G. Pozzati, A. Shenoy, W. Zhu, P. Kundrotas, V. R. Serra, C. H. M. Rodrigues, A. S. Dunham, D. Burke, N. Borkakoti, S. Velankar, A. Frost, K. Lindorff-Larsen, A. Valencia, S. Ovchinnikov, J. Durairaj, D. B. Ascher, J. M. Thornton, N. E. Davey, A. Stein, A. Elofsson, T. I. Croll, and P. Beltrao, "A structural biology community assessment of alphafold 2 applications," *bioRxiv*, p. 2021.09.26.461876, 9 2021. [Online]. Available: https://www.biorxiv.org/content/10.1101/2021.09.26.461876v1https://www.biorxiv.org/content/10.1101/2021.09.26.461876v1.abstract

[42] K. Weissenow, M. Heinzinger, and B. Rost, "Protein language-model embeddings for fast, accurate, and alignment-free protein structure prediction," *Structure (London, England : 1993)*, vol. 30, pp. 1169–1177.e4, 8 2022. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/35609601/

[43] I. Anishchenko, S. J. Pellock, T. M. Chidyausiku, T. A. Ramelot, S. Ovchinnikov, J. Hao, K. Bafna, C. Norn, A. Kang, A. K. Bera, F. DiMaio, L. Carter, C. M. Chow, G. T. Montelione, and D. Baker, "De novo protein design by deep network hallucination," *Nature*, vol. 600, pp. 547–552, 12 2021. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/34853475/

[44] L. Wu, C. Gong, X. Liu, M. Ye, and Q. Liu, "Diffusion-based molecule generation with informative prior bridges," 9 2022. [Online]. Available: https://arxiv.org/abs/2209.00865v1

APPENDIX

A. Glossary of important concepts

Protein Data Bank: Comprehensive database of experimental and computational 3D protein structures.

Proteins: Made up of amino acids determined by their amino acid sequence. Fold to a functioning protein often as protein complexes.

Primary protein structure: Amino acid sequence of the protein.

Secondary protein structure: Local structures such as beta sheets and alpha helices.

Tertiary protein structure: 3D structure after the folding process.

Quartnerary structure: Arangement of multiple protein structures to a protein complex.

Protein complexes: Can be arranged as monomers (one binder, one target) or oligomers (one binder, multiple target components).

Hetero/homo: Used in conjunction with complex description. Hetero $\rightarrow$ consist of components which differ from each other. Homo $\rightarrow$ consists of components which are the same.

Homology: Similarity possibly due to shared ancestry eg. in when talking of similar genes that are related -¿ Produce possibly similar proteins that are related.

DARPin: Designed Ankyrin Repeat Proteins are a class of antibody mimetics that have proven useful in clinics, diagnostics and research. Limited conformational flexibility reduces the sampling space, simplifying homology modelling.Introduction to DARPins.

Ligand: Synonym for Binder/DARPin.

Receptor: Synonym for Target.

Epitope: Target substructure to which the DARPin binds. Typically determined by Xray Cristallography of the complex since computationally still many possible binding geometries exist-¿ Uncertain time lines.

Paratope: DARPin (or protein in general) substructure which binds the target.

Decoy Artificially generated protein which competes with the real protein in evaluation metrics of a variety of problems for example such as docking scores.

Rigid body Docking: No conformational changes considered are considered for docking.

Flexible body docking: Conformational flexible parts of the proteins changes are considered for docking.

Binding affinity: Describes the amount of binding of a DARPin to a target structure.

Dissociation constant K_d: Describes the dissociation from the target.

Enzyme: Protein which catalyzes a chemical reaction.

Allosteric binding: Does not bind at the active site of an enzyme but rather at another site and eg. through conformational changes leads to altered enzyme activity.

SMILES: Molecular representation of molecules

Molecular Fingerprinting: Computational representation of molecular structures. various approaches, SMILES based, Machine learning based

Translation: process which converts mRNA into a chain of amino acids which then folds into a protein. Cool Youtube Video

Ribsome: Molecular protein factory

Ribosome display: In vitro method which links mRNA, which encode the DARPin to the translated DARPin protein via the Ribosome. The produced DARPin-Ribosome-mRNA complexes are then subjected to the target/ligand of interest upon which additional bio essays are performed to determine the best binder (eg. ELISA)

ELISA: Enzyme-linked immunosorbent assay = immunological assay which produces a colorimetric response and measures which binder is the best

B. Pytorch Geometric Data objects

In order to break down complexity the Data objects with their data are displayed here. Subgraphs and graphs only differ in their used atoms(=nodes) and the according binding property edge_index and edge_attribute reduction.

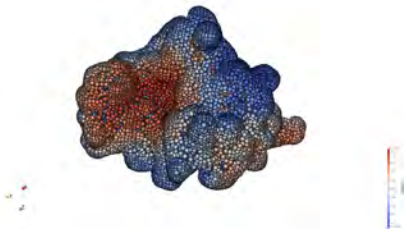
```
# Input description
pdb_id=protein_identifier
atom_features = [atom_element,
    hydrophobicity, dmasif_features]
pos = 3D coordinates
edge_index=edge_connections # In COO Format
edge_attr=[binding_type[covalent,
    interaction]]
edge_distance = distance of edges
y=[binding(1) or not_binding(0)]


# Pytorch Geometric DataType definition
GRAPH = Data(
         pdb_id=protein_identifier,
         x=atom_features_protein,
         pos=protein_coords,
         edge_index=protein_edge_index,
         edge_attr=edge_distance,
         y=torch.ones(1).long(),
      )
```

C. DMasif Model

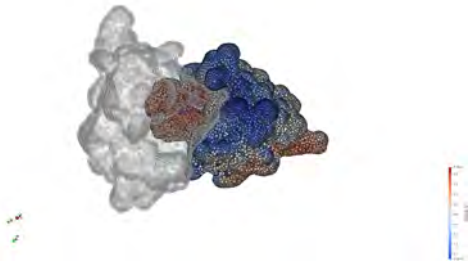
1) Parameters: Below are the parameters as taken by argparse when running the prediction.py file of dMasif in python.

```
Namespace(atom_dims=6, batch_size=1,
    curvature_scales=[1.0, 2.0, 3.0, 5.0,
    10.0], device='cuda', distance=1.05,
    dropout=0.0, emb_dims=16,
    embedding_layer='dMaSIF',
    experiment_name='model',
    in_channels=16, k=40, n_epochs=50,
    n_layers=3, n_rocauc_samples=100,
    no_chem=False, no_geom=False,
    orientation_units=16,
    pdb_list='pdb_ids.txt', post_units=8,
    profile=False, radius=12,
    random_rotation=False, resolution=0.7,
    restart_training='', search=False,
    seed=42, single_pdb='',
    single_protein=False, site=True,
    sup_sampling=100,
    unet_hidden_channels=8,
    use_mesh=False,
    validation_fraction=0.1, variance=0.1)
```

2) Binding site prediction examples: An example where dMasif predicted the correct binding site factor (red). While there are other regions that are colored in red they are less pronounced so the prediction serves as a good estimation which can be performed fast and automatically. The model however is not perfect and there are other examples where the prediction did not work well (data not shown).



(a) 1A7X single protein chain. The example shows a good binding site prediction of the model which can be confirmed when overlaying it with the transparent second chain structure as visible in (b).



(b) 1A7X protein complex with the binding site prediction chain (visible also in (a)) in color and the complementary chain in transparent solid color.

Fig. 5: Correct binding site prediction as performed by dMasif

D. Model architectures

1) Graph Convolutional Network Architecture: Below the baseline model summary as given out by PyTorch.

```
  | Name        | Type              | Params
-------------------------------------------------
0 | loss_module | BCEWithLogitsLoss | 0
1 | conv1       | GCNConv           | 300
2 | conv2       | GCNConv           | 10.1 K
3 | fc_1        | Linear            | 1.6 K
4 | out_layer   | Linear            | 17
-------------------------------------------------
```

2) Graph Isomorphic Network Architecture: Below the GIN model summary as given out by PyTorch.

```
  | Name        | Type              | Params
-------------------------------------------------
0 | loss_module | CrossEntropyLoss  | 0
1 | conv1       | GINConv           | 10.5 K
2 | conv2       | GINConv           | 20.4 K
3 | conv3       | GINConv           | 20.4 K
4 | lin1        | Linear            | 90.3 K
5 | lin2        | Linear            | 602
-------------------------------------------------
```

E. GCN: Mean Max and Sum Operation Comparison

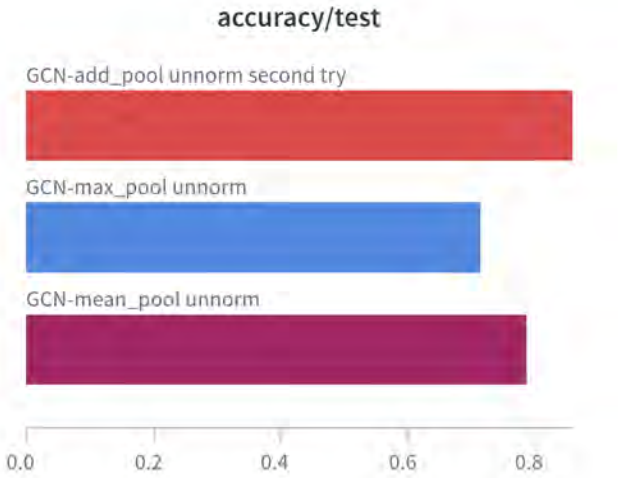


Fig. 6: Test Accuracy

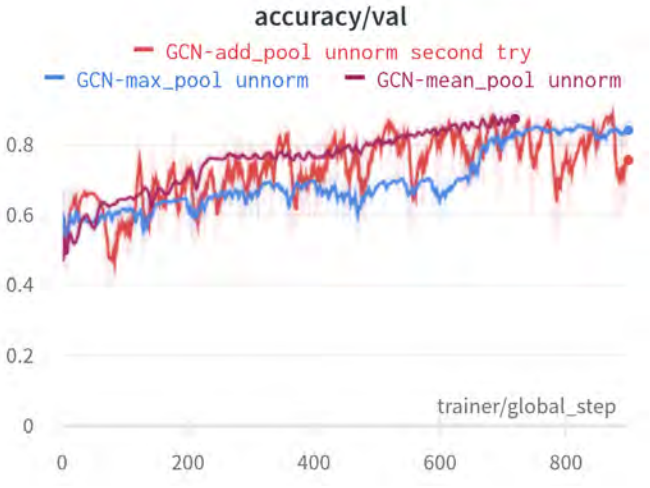


Fig. 7: Validation Accuracy

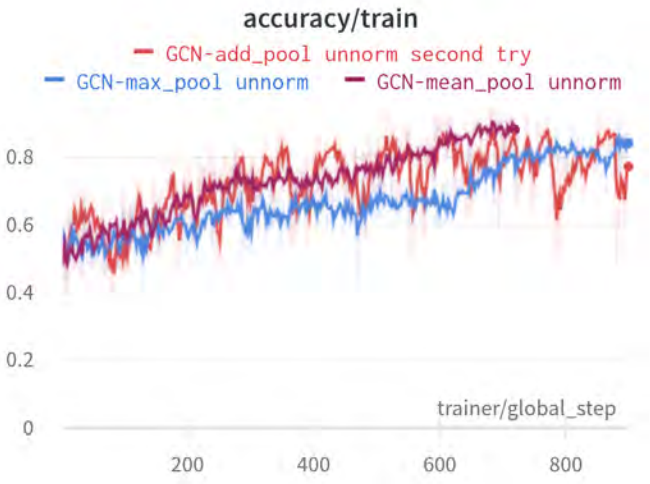


Fig. 8: Training Accuracy

F. Graph Isomorphic Training

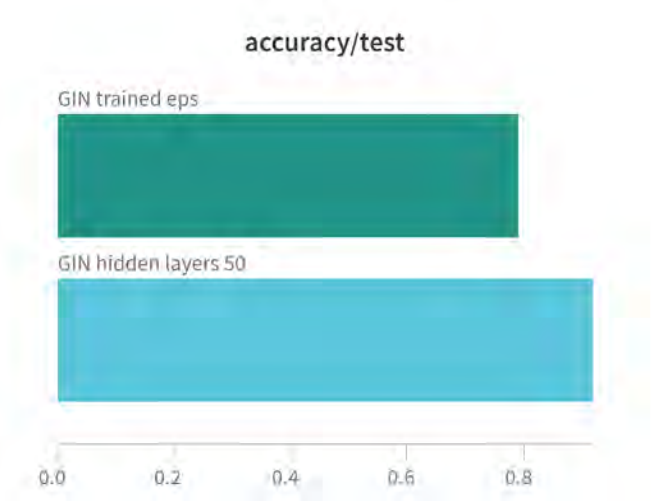


Fig. 9: Test Accuracy

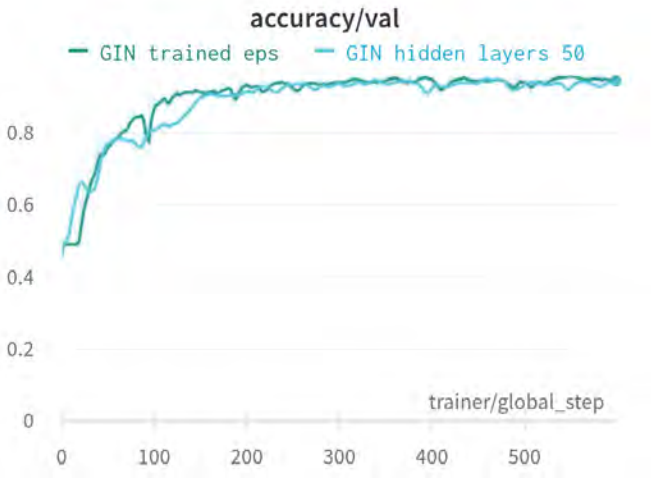


Fig. 10: Validation Accuracy

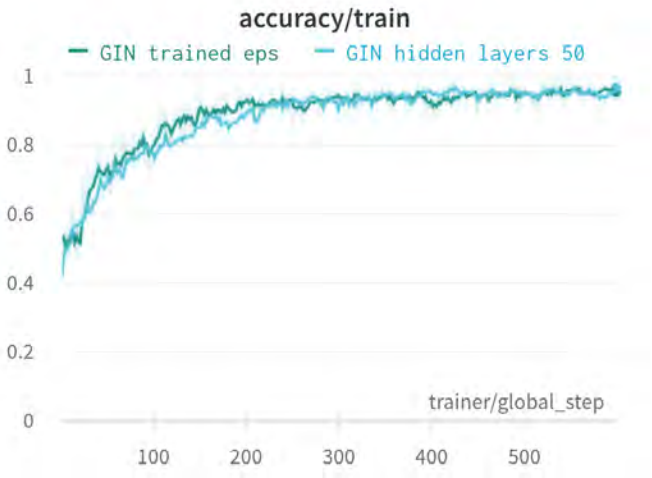


Fig. 11: Training Accuracy

G. DARPin Test PDB IDs

- 5MA6
- 4HRN
- 7TZ0
- 4HRL
- 5O2S
- 2V5Q
- 5KNH
- 5MBL
- 5FIO
- 5OOY
- 5OP1
- 6H47