Team name	Deeplab
Team member(s) (firstname lastname; …)	Stavros Thedorakis, Vassilis Pitsikalis, Nikos Epitropakis, Agamemnon Krasoulis, Nikos Antonopoulos
Affiliation	deeplab.ai
Contact email	sth@deeplab.ai, nanton@deeplab.ai
Contact phone number (optional)	
Protein targets (for example: 3CLPro/Nsp5, BoAT1, Fc Receptor, Furin, IL6R, M protein, Nspx, OrfXx, N, E, etc…) | 3 required	- 3CLPro - Papain-like protease (PL - PRO) - Spike

Section 1: methods & metrics

Describe what methods you have used, how they are independent from one another, what your workflow was, how you performed the cross-correlation between your methods. If applicable, please report estimated performance metrics of your methods, such as accuracy, sensitivity, false-discovery rate, etc., and how those metrics were obtained (e.g. cross-validation). Please provide key references if available.

Methods:

We develop a geometric deep learning-based approach by learning descriptive representations for both ligand and protein structures and subsequently employing them in a graph representation to predict a binding affinity score. We are inspired by recent methods from the protein-ligand screening literature [1,2]. Our main contributions lie in using multiple protein representations at different levels and combining predictions across different models; in the way that the two ligand-protein representations are fused; as well as the pretraining of the ligand GNN adopted by [2]. Previous deep learning approaches, such as [7, 8] were based on 3D-convolutions completely disregarding the graph structure of both protein and ligand. Additionally, they only focused on using regression for binding affinity prediction which, as is shown in [5], does not guarantee good performance in the screening scenario, where most ligands do not bind. We observed that our proposed end-to-end approach substantially outperformed individual models when evaluated on the screening power test defined in [5], i.e., the PDBbind core set v.2019[5]. We observed an improvement in performance over competitive methods with a factor of 2.5 [5].

Next, we provide more details on the methods and the metrics themselves.

The models we employ comprise three main components: 1. A graph neural network (GNN) which learns an embedding for **ligands**, referred to as *ligand-gnn*; 2. A graph neural network which learns an embedding for **proteins**, referred to as *protein-gnn*; 3. A fully connected neural network which combines the two embeddings in a biologically meaningful way to estimate the final binding affinity score.

The ligand pre-processing and modelling pipeline that we employ was developed at the atomic-level and was inspired by the work of Hu et al. [2]. So as in [2] we employ pretraining of the ligand graph neural network (GNN) with unsupervised methods on 2 million molecules from Zinc15.

For protein modelling, we focused on two different approaches/models: an atom-level protein pocket representation, which was similar to ligand processing; a protein pocket surface representation which includes both geometric and chemical features on the nodes of the 3D mesh utilizing prior work of Gainza et al. [1]

Overall our contributions are as follows:

1. Application of geometric deep learning to binding affinity prediction.
2. Employing multiple protein representations at different levels (e.g. atom-level and 3D surface which includes both geometric and chemical features) while combining predictions across the different models.
3. Modelling ligands utilizing a graph neural networks.
4. Predicting binding affinity without the need of modelling the docking pose of the molecule while utilizing just the graph representation of the ligand suitable for large libraries screening.

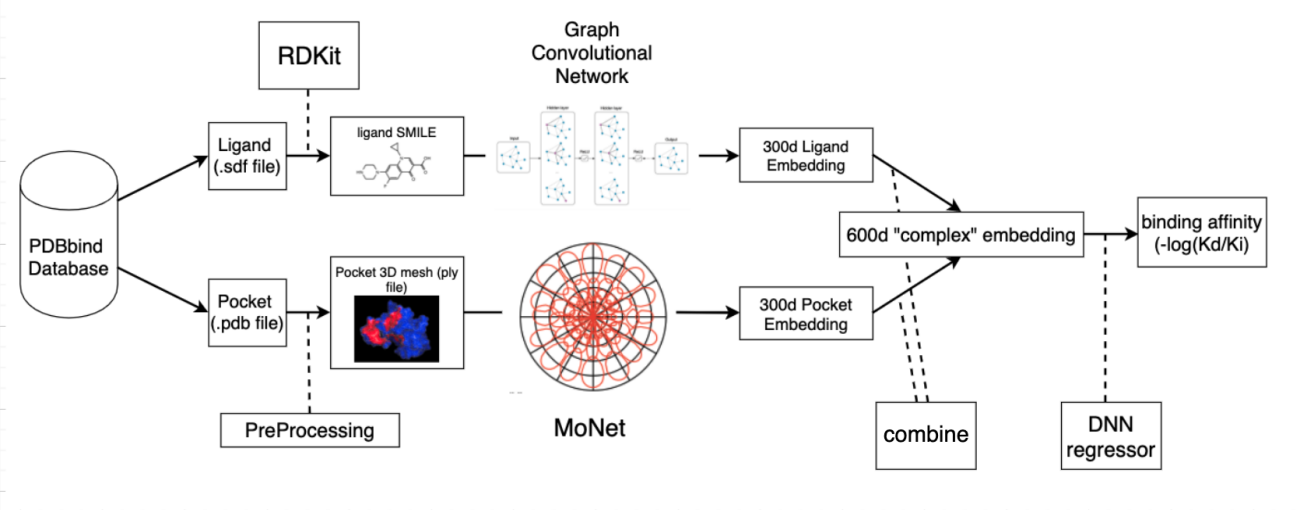


Figure 1: Architecture of GNN + MoNet model. The ligand and protein representations are fed through independent graph neural networks and their embeddings are combined at the end with a fully connected neural network regressor which predicts protein-ligand pair binding affinity.

Our final prediction was based on a fusion of the two models, using an ensemble voting scheme. Models were trained using the refined set of the PDBbind database [4,6]. System hyper-parameters, including architectural choices for the two models as well as parameters of the fusion mechanism, were tuned using a screening power evaluation metric on the PDBbind core set v.2019 *(validation set)* [5] . The set of proteins included in the validation set were excluded from model training.

We chose the enhancement factor (EF) as our main evaluation metric, which is defined as follows [5]:

$$EF_a = NTB_a \,/\, NTB_{total} \times \alpha,$$

where NTB_a is the number of true binders ranked among the top $\alpha\%$ranked candidates selected by a given scoring function, and NTB_{total} is the total number of true binders for a given target protein in the dataset (typically five in the PDBbind core set). We set $\alpha = 1$, thereby evaluating screening power of top *1%* (i.e., EF_1).

In the *PDBbind core set v.2019* set, our method achieved state-of-the-art screening performance with an $EF_1 = 39.16$. To our knowledge, this score is larger than the highest-reported one in the literature with absolute improvement of 300% [5].

For the final screening, we included four molecule libraries/databases, which are described in the next section. We extracted 15 pockets in total from the target proteins using the P2rank algorithm [3]: three for *6m0j* and six for *6y7m* and *6wuu*. We then ran inference for all molecules in the four libraries, separately for each extracted pocket. We finally selected those ligands that achieved the highest scores with our model fusion strategy, regardless of binding pocket.

Section 2: targets

Describe for each protein target: why you chose it, from which source you obtained it (e.g., insidecorona.net / covid.molssi.org / rcsb.org) and why this is the best quality structure, if any pre-processing (e.g., energy minimization, residue correction, alternative folding, …) was performed.

We selected the following three proteins, following suggestions provided during the Jedi Symposium sessions, as well as optimising in terms of resolution and the rest of the suggested values of importance. The structures of all proteins were obtained from rcsb.org. As a pre-processing step, we removed water molecules and ligands to match the representation we had in our training and evaluation sets. Multiple pockets extraction were applied to each protein using the P2rank algorithm [https://github.com/rdk/p2rank]. We additionally investigated other pocket extraction methods and decided to go forward with P2rank [3] , as it showed the best performance.

Target 1: ***6y7m*** *(3CLPro)*
link: ***https://www.rcsb.org/structure/6Y7M***
This was a useful structure as it contained a ligand bound to the protein. Interestingly, the pocket extraction tool ranked the pocket where the ligand binds as first as can be seen in Figure 2.

Target 2: ***6wuu (***Papain-like protease (PL - PRO))
link: ***https://www.rcsb.org/structure/6WUU***
We specifically chose the Biological Assembly 1 that contained one chain with the peptide inhibitor. Again here, the pocket extraction tool ranked the pocket where the peptide inhibitor binds as first.

Target 3: **6m0j** *(Spike)*
link: **https://www.rcsb.org/structure/6M0J**
We discarded ACE2 and kept only the spike receptor-binding domain.

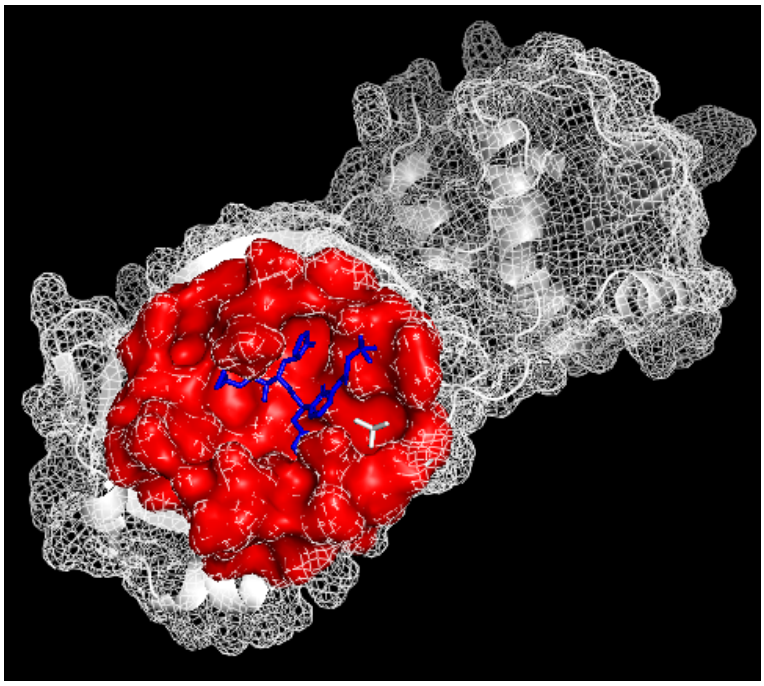


Figure 2: The 6y7m structure (white) along with its ligand (blue) and our predicted pocket (red) from p2rank. We expanded the size of the pocket to match the size of the pockets in the PDBbind database.

Section 3: libraries

Describe which libraries you have used, how they were combined, if any compounds were removed / added, why additions are relevant, any unique features of your library, etc. Please provide the sources you obtained the libraries from (if publicly available). Describe the procedure of data preparation (removal of duplicates, standardization, etc). Indicate if different libraries were used for different targets, and why. If possible, provide a download link to your version of the library.

Training And Evaluation:

Since our solution was based on machine learning, we used a strict cross-validation approach to assess the performance of our algorithm on sets of proteins excluded from the training set. More specifically, we used the refined set of the PDBbind (v2019) database [4], which contains 4,852 high protein-ligand pairs and their affinity score. For model evaluation we used the PDBbind core set (v2019) [4], which contains 285 protein-ligand pairs. All proteins and ligands included in the core validation set were excluded from model training.

Screening:

For the virtual screening procedure, we used the following libraries shared by the JEDI community. All four libraries were used for all targets. During screening, we removed duplicates entries from the ligand libraries.

Library	Number of ligands
ZINC15 (w. Ro5 filtering rule)	~ 1 B
MERC	~ 5 M
CAS	~ 50 K
SWEETLEAD	~ 4.5 K

Table 1: Libraries used for virtual screening procedure

Section 4: results

Briefly describe you key findings, any interesting trends in your data, a description of your top 5 compounds for each target. If possible, provide a link to a code and/or data repository. Please do not submit randomly selected compounds!

Results:

Our method achieved a score of $EF_1 = 39.16$ in the validation set (i.e., PDBbind core set (v2019)). To our knowledge, this is the highest-reported screening power score on this dataset.

METHOD	*EF1*
ChemPLP@Gold (best model on [5])	*12*
DOUBLE GNN (Ours)	*32.22*
GNN + MONET (Ours)	*33.25*
FUSION (Ours, our final selection)	*39.17*

Table 2: Performance comparison of state-of-the-art models in the literature [5], our individual models and our model fusion.

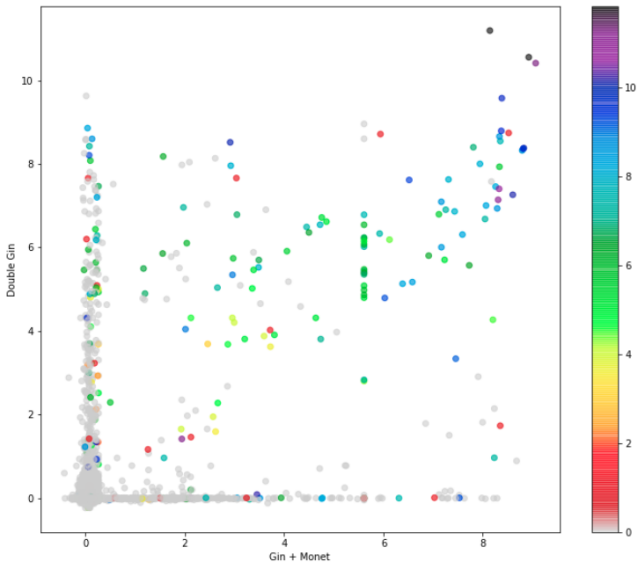


Figure 3: Predicted scores of the Double Gin and Gin + Monet methods on the PDBbind core set. The colors indicate the ground truth binding affinity, -logKd or -logKi (depending on which is available). Red points are cross-binders for which the value of binding affinity is unknown. We see that where the models agree there is a very high precision in identifying active ligands. The predicted score is correlated with binding affinity which explains our success on the screening power test. Independent methods have a higher false positive rate as there are many inactive ligands with a high score from just one model.

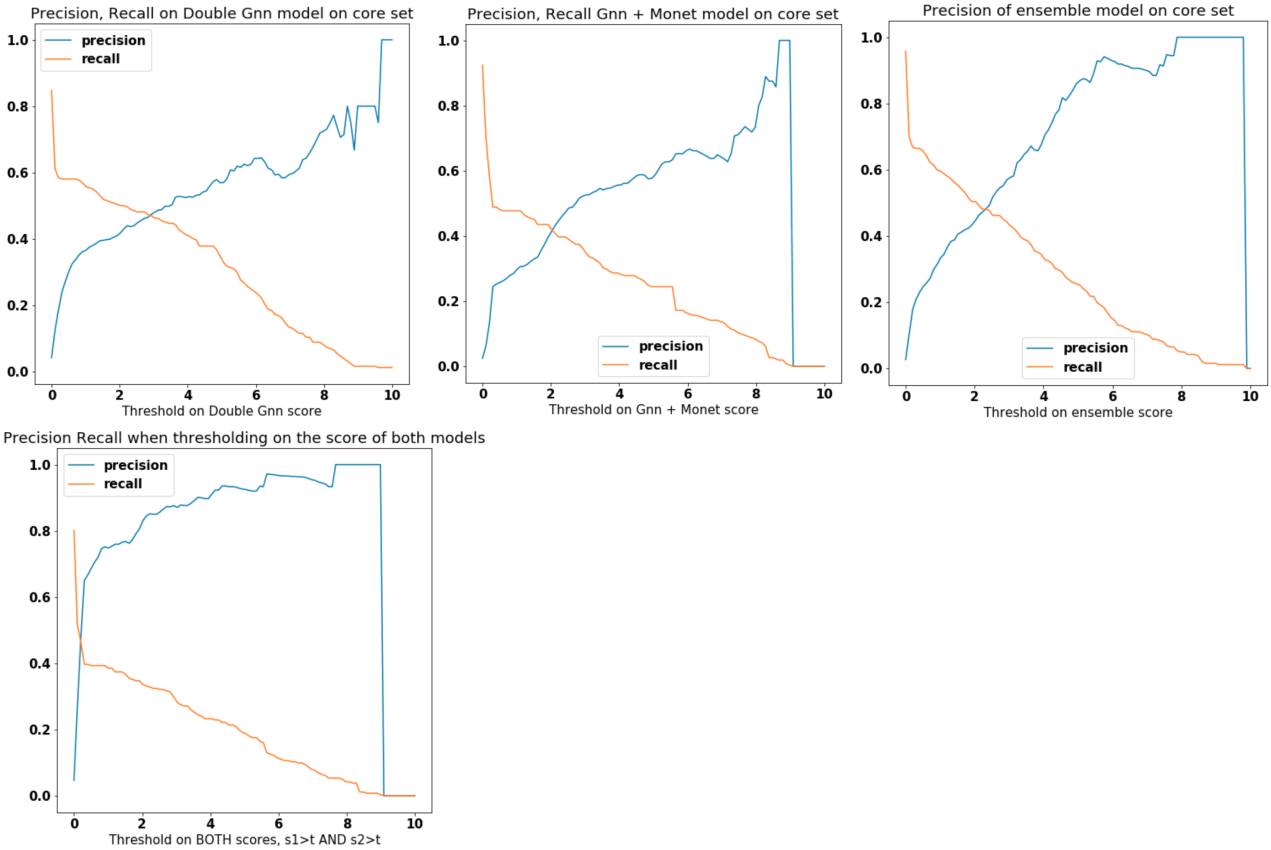
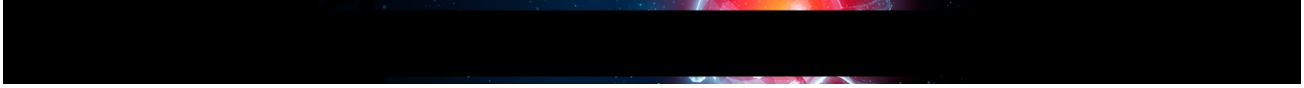


Figure 4: Precision and recall curves for different score thresholds. Combining the two models leads to higher precision. This is most evident when we threshold both scores. At a threshold of 3 the precision is already 87%.

Final selection:



Next we provide descriptions on how we have constructed the final top-ranked 10K list per protein. This has been built guided by our analysis and results on the core set. An important conclusion towards this was the following:

> *We found out that a combination of the predictions of both models was much more precise than any model alone.*

This can be seen in Figures 3,4. In particular we can observe that:

- In Figure 3 we see *many more true positives in areas of the graph where both models agree*.
- At the same time, this is further illustrated in Figure 4 where *very high precision is achieved when both models are above a score threshold*.

Having this in mind for our final ranking we have employed the following strategy:

1. *High-precision multi-agreeable items.* These are the top-of-the-list items. We place first molecules that both algorithms gave a high binding affinity score. We should note that the relative number of these items is indicatively 5.6% for Spike, 23% for PL-PRO and 32% for 3CLPro. We ranked these *top-of-the-list items* according to a weighted average score. We report this score in the lists. These items have been measured to have high precision.

2. *Rest-of-the-list top-ranked items.* Next right after (1) above, we have ranked the rest of the items as predicted by the maximum over the scores given by any of the models.

Recapping, given the above, our final list is of the following form: On top it contains a small percentage of the items on which all approaches coincided. Next, we append the rest of the items given their maximum score. This strategy has the following side effect. *The scores in the first part of the list may have scores that are lower than some of the ones that follow.* However, this is on purpose given our strategy. We expose in the submitted list both our ranking and the scores given the described side effect.

Ranking	*Molecule*	*Score*
1	*part_1: mol_1*	*score*
2	*part_1: mol_2*	*score*
3	*part_1: mol_3*	*score*
...	*...*	*...*
N	*part_1: mol_N*	*score*
N+1	*part_2: mol_1*	*score*
N+2	*part_2: mol_2*	*score*
...	*...*	*...*
N+K	*part_2: mol_K*	*score*

Table 3: This table shows schematically our strategy in the combination of multiple lists: First part contains the ranked list of ligands for which both approaches coincide, part_1 (mol_1, …, mol_N). Second part

contains the highest ranked rest of the ligands as ranked by the maximum score of any approach, part_2 (mol_1, …, ,mol_K). Note that part_1:mol_i is not the same as part_2:mol_i, since this is a remapping of the original list of all 10K ligands in which each of them is unique.

Conclusion:

We have built an innovative combination of approaches building on top of recent state-of-the-art ones [1,2] and contributing significant components; together with appropriate pretraining , preprocessing, pocket selection, we have managed to show state-of-the-art results in the PDBbind core set. Given the best evaluated models we followed a double metric strategy after screening 1B items to provide our final lists. These lists have been constructed by employing ranked items by two criteria, highly ranked agreeable items by multiple approaches, and maximum scored items. An upcoming publication is going to be published in the near future describing all research and technical details required for the reproduction of the results. At the same time, we are going to release the repository of the code that implements the main parts of our submission, contributing in this way to the interdisciplinary communities of drug discovery and specifically protein-ligands' interactions for the fight against covid-19 and not only.

References:

[1] Gainza, P., Sverrisson, F., Monti, F., Rodolà, E., Boscaini, D., Bronstein, M. M., & Correia, B. E. (2020). Deciphering interaction fingerprints from protein molecular surfaces using geometric deep learning. *Nature Methods*, *17*(2), 184–192. https://doi.org/10.1038/s41592-019-0666-6

[2] Hu, W., Liu, B., Gomes, J., Zitnik, M., Liang, P., Pande, V., & Leskovec, J. (2019). Strategies for Pre-training Graph Neural Networks. 1, 1–5. http://arxiv.org/abs/1905.01591

[3] Krivák, R., & Hoksza, D. (2018). P2Rank: machine learning based tool for rapid and accurate prediction of ligand binding sites from protein structure. Journal of Cheminformatics, 10(1), 1–12. https://doi.org/10.1186/s13321-018-0285-8

[4] Liu, Z., Su, M., Han, L., Liu, J., Yang, Q., Li, Y., & Wang, R. (2017). Forging the Basis for Developing Protein-Ligand Interaction Scoring Functions. Accounts of Chemical Research, 50(2), 302–309. https://doi.org/10.1021/acs.accounts.6b00491

[5] Su, M., Yang, Q., Du, Y., Feng, G., Liu, Z., Li, Y., & Wang, R. (2019). Comparative Assessment of Scoring Functions: The CASF-2016 Update. Journal of Chemical Information and Modeling, 59(2), 895–913. https://doi.org/10.1021/acs.jcim.8b00545

[6] Wang, R., Fang, X., Lu, Y., Yang, C. Y., & Wang, S. (2005). The PDBbind database: Methodologies and updates. Journal of Medicinal Chemistry, 48(12), 4111–4119. https://doi.org/10.1021/jm048957q

[7] Jiménez, J., Škalič, M., Martínez-Rosell, G., & De Fabritiis, G. (2018). KDEEP: Protein-Ligand Absolute Binding Affinity Prediction via 3D-Convolutional Neural Networks. *Journal of Chemical Information and Modeling*, *58*(2), 287–296. https://doi.org/10.1021/acs.jcim.7b00650

[8] Gomes, J., Ramsundar, B., Feinberg, E. N., & Pande, V. S. (2017). *Atomic Convolutional Networks for Predicting Protein-Ligand Binding Affinity*. 1–17. http://arxiv.org/abs/1703.10603