Exploring $\Delta\Delta G$ prediction with Siamese Networks

Andrew T. McNutt
Comp. and Systems Biology
University of Pittsburgh
Pittsburgh, PA
and.mcnutt@pitt.edu

David Koes
Comp. and Systems Biology
University of Pittsburgh
Pittsburgh, PA
dkoes@pitt.edu

Abstract

During lead optimization, lead molecules are refined for potency via slight modifications of their chemical structure. Relative binding free energy (RBFE) methods allow comparisons of molecular potency during this optimization. We utilize a Siamese Convolutional Neural Network (CNN) to directly estimate the RBFE with higher throughput than simulation based methods. Our models show improved performance over a previously published Siamese RBFE predictor. We observe decreased performance on out-of-domain RBFE predictions.

1 Introduction

Lead optimization is a phase of the drug discovery process that simultaneously optimizes a lead molecule for potency, solubility, and other pharmaceutical properties. Small modifications are made to the chemical scaffold of the lead molecule and tested for their affect on the properties of interest. Collections of molecules produced by this process are termed congeneric series. The synthesis and testing of each chemical modification takes considerable amounts of time and money. Relative binding free energy (RBFE) methods provide an *in silico* alternative to labor-intensive synthesis and testing by predicting the change in binding free energy, $\Delta\Delta G$, between congeneric series members. Typical methods for RBFE use molecular dynamics with alchemical perturbations or a thorough sampling of the endpoints of the transformation [1,2]. However, these methods suffer from both a lack of applicability to large changes between ligands and a low throughput of RBFE predictions [3,4].

As a faster alternative to simulation methods, ML-based scoring functions provide low error and high throughput for absolute binding affinity predictions [5–10]. Using absolute binding affinity methods as inspiration, Jiménez-Luna et al. [11] utilize a Siamese Convolutional Neural Network (CNN) [12,13] architecture to predict the RBFE between two bound protein-ligand complexes, removing the compounding error from subtracting predicted absolute binding free energies. Here we expand their work on Siamese Network RBFE predictors by introducing novel loss components and examining the impact of model architecture. Generalizability of the trained Siamese Network is evaluated.

2 Methods

2.1 Data

We use the BindingDB 3D Structure Series dataset [14] as it provides congeneric series with experimentally determined binding affinities for structurally enabled targets. The full dataset encompasses 1038 congeneric series with an average of 9.61 ligands per congeneric series. We filter the dataset as described in the appendix (A.2) to ensure the binding affinity measurements are high quality and that we only compare ligands with identical measures of potency: IC_{50}, K_d, or K_i. Our final filtered BindingDB dataset has 943 unique receptor structures, encompassing 1082 congeneric series with

35th Conference on Neural Information Processing Systems (NeurIPS 2021), Sydney, Australia.

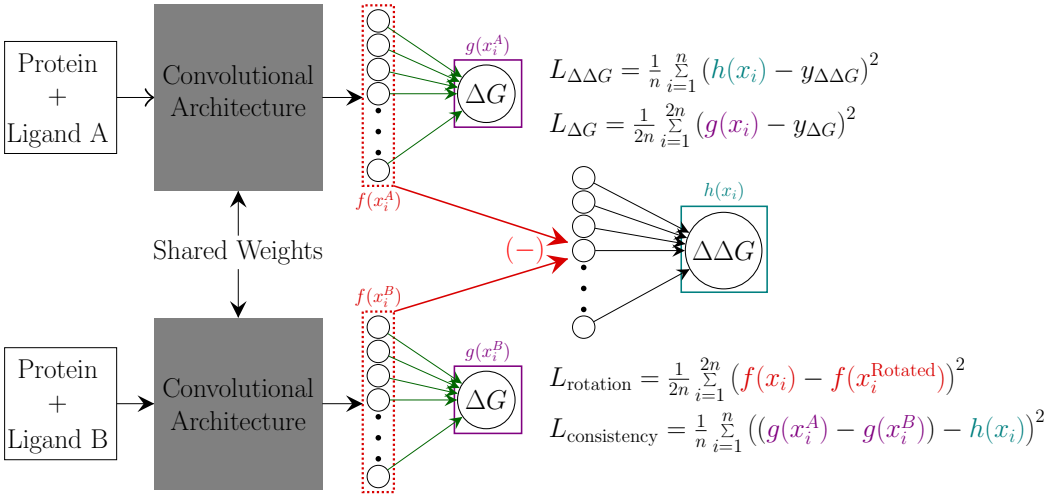


Figure 1: Siamese Network simultaneously predicts both $\Delta\Delta$G and ΔG using the latent vectors of each protein-ligand complex as determined by the shared convolutional architecture.

an average of 7.995 ligands per congeneric series (Figure A2). The average affinity range of each congeneric series is 2.023 pK (Figure A3).

2.2 Model Architecture

Similar to Jiménez-Luna et al.[11] we use a Siamese Network[12]. Siamese Networks use two arms with shared weights to determine the distance between a pair of inputs, and are used in object matching and object tracking[15–17]. We use 3D CNNs as the arms of our Siamese Network to learn directly from the 3D structure of protein-ligand complexes (Figure 1). We utilize the libmolgrid python library[18] with default settings to voxelize the complexes (further defined in A.1). The inputs are then passed through one of the main convolutional architectures employed by GNINA,[19] Default2018 or Dense, as defined in Francoeur et al.[6] (Figure A1). These convolutional architectures can predict absolute binding affinity directly from the 3D bound structure of a protein-ligand complex. We implement the Siamese Network by training a linear layer on the difference between the final latent vectors (27,648 and 224 for Default2018 and Dense, respectively) of the convolutional architectures of the two input complexes to predict the $\Delta\Delta G$. The latent vector of each complex is also linearly mapped to its absolute binding affinity (Figure 1).

We train our model using a linear combination of losses:

$$\mathcal{L}_{\text{Total}} = \alpha\mathcal{L}_{\Delta\Delta\text{G}} + \beta\mathcal{L}_{\Delta\text{G}} + \gamma\mathcal{L}_{\text{rotation}} + \delta\mathcal{L}_{\text{consistency}} \tag{1}$$

where $\alpha, \beta, \gamma, \delta \in \mathbb{R}^+$. $\mathcal{L}_{\Delta\Delta\text{G}}$ and $\mathcal{L}_{\Delta\text{G}}$ are the mean square error (MSE) of the RBFE and absolute binding affinity predictions, respectively. $\mathcal{L}_{\text{rotation}}$ is the MSE of the latent vectors of two randomly rotated versions of each protein-ligand pair (Figure 1). This component encourages the latent vectors to be rotationally invariant. $\mathcal{L}_{\text{consistency}}$ is the MSE of the difference between the predicted absolute binding affinities and the predicted RBFE, to ensure that the model is providing consistent predictions. Hyperparameters and other training information are provided in the appendix (A.3).

2.3 Additional Ligands Comparison

In order to directly compare our trained model to Jiménez-Luna et al.[11], we utilize the additional ligands training set as described in their manuscript. We train our models on a reference ligand, as described in A.2, and a given number of additional ligands for each congeneric series. In the one additional ligand training set, we train on the two ordered pairs of the reference ligand and one additional ligand. Then testing is carried out on the two-permutations between the ligands in the training set and ligands that the model has not seen. We construct 25 versions of the training and testing datasets for each number of additional ligands. Each version of the additional ligand set uses the same reference ligands and randomly chooses additional ligands to add to the training set for each

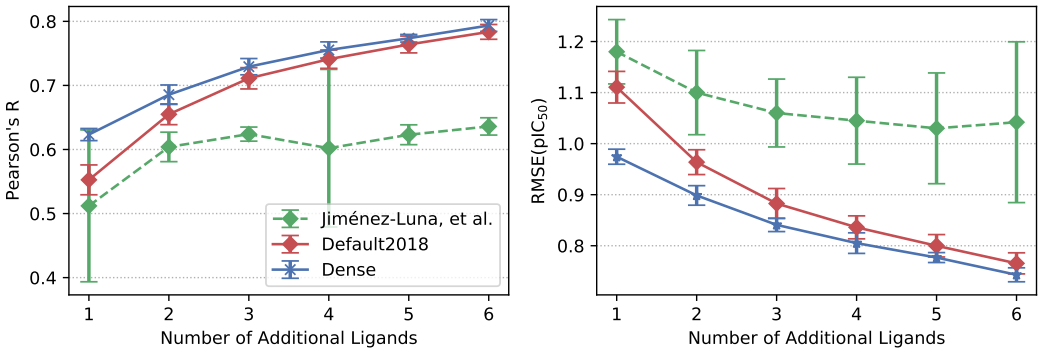


Figure 2: Comparison of our models to Jiménez-Luna et al. [11] on the additional ligands dataset. Error bars indicate ± 1 standard deviation of 25 individual models (only 5 for Dense).

congeneric series. In the case of the Dense convolutional architecture, we only use 5 versions of the training and testing datasets due to the architecture's high computational cost.

2.4 Ablation Study

Using the one additional ligand training and testing sets, we investigate the average performance of 25 models as we disable aspects of the model. We remove components of the loss function to determine their impact on performance. The contributions of the architectural components to the performance of the model is also explored. We evaluate the utility of the latent space subtraction by concatenating rather than subtracting the latent spaces of the convolutional arms of the network, requiring the RBFE prediction layer to double in input size. The importance of the Siamese network is evaluated by training a CNN that takes in one protein-ligand pair and predicts the absolute binding free energy. RBFE is the difference of the predicted absolute binding free energies of two ligands. When utilizing this architecture, we can no longer enforce $\mathcal{L}_{\text{consistency}}$.

2.5 Generalization to new Protein Families

RBFE predictions are most useful if they perform well on novel proteins and ligands outside of the training domain. We evaluate the worst case scenario for generalization via a leave-one-out protein family cross-validation. Using the protein family database [20] we annotate the proteins in the BindingDB dataset with all of its associated families, finding 73 protein families. We then construct a leave-one-out cross-validation set for each protein family, where the training set is composed of the whole BindingDB dataset without any congeneric series in a given protein family and the testing set is composed of only the congeneric series from the left-out protein family.

3 Results

3.1 Additional Ligands Comparison

Both of our model's predictions show higher correlation with the experimental RBFE ($\Delta\Delta G$) and lower root mean square error (RMSE) on the RBFE predictions in comparison to the model developed by Jiménez-Luna et al. [11] (Figure 2). The mean absolute error (MAE) of our models predictions show the same trend as the RMSE (Figure A4). Additionally, our models demonstrate a decreased variance across the 25 versions of the training and test splits. The models demonstrate a continual increase in performance as they are given more training information about the congeneric series. We find that the high parameter Dense model does better with lower amounts of congeneric series comparisons than the lower parameter, Default2018, model. The difference between the performance of the two CNN architectures decreases as more information is added to the training set of the models.

Table 1: Ablating different components of the network on the 1 Additional Ligands set to determine their utility in the full network. Parentheses indicate the ± 1 standard deviation of the 25 train/test versions. **Bold** indicates results are not significantly different from Standard ($p > 0.05$).

Ablation	Pearson's R	RMSE (pK)	MAE (pK)
Standard	$\mathbf{0.553(\pm 0.0233)}$	$\mathbf{1.11(\pm 0.0309)}$	$\mathbf{0.82(\pm 0.0187)}$
No $L_{\Delta\Delta G}$	$\mathbf{0.551(\pm 0.0202)}$	$\mathbf{1.12(\pm 0.0248)}$	$\mathbf{0.829(\pm 0.0179)}$
No $L_{\Delta G}$	$0.459(\pm 0.0238)$	$1.27(\pm 0.0289)$	$0.945(\pm 0.0182)$
No L_{rotation}	$\mathbf{0.556(\pm 0.0188)}$	$\mathbf{1.11(\pm 0.0233)}$	$\mathbf{0.819(\pm 0.0162)}$
No $L_{\text{consistency}}$	$\mathbf{0.536(\pm 0.021)}$	$1.14(\pm 0.0356)$	$0.842(\pm 0.0186)$
No $L_{\Delta\Delta G}$, $L_{\text{consistency}}$	$-0.0576(\pm 0.136)$	$1.24(\pm 0.0143)$	$0.908(\pm 0.0144)$
No $L_{\Delta G}$, $L_{\text{consistency}}$	$0.456(\pm 0.0231)$	$1.28(\pm 0.0319)$	$0.95(\pm 0.0233)$
Concatenation	$\mathbf{0.554(\pm 0.0134)}$	$\mathbf{1.11(\pm 0.0223)}$	$\mathbf{0.821(\pm 0.0174)}$
No Siamese Network	$0.5(\pm 0.0347)$	$1.15(\pm 0.0362)$	$0.854(\pm 0.021)$

3.2 Ablation Study

Removing $\mathcal{L}_{\Delta\Delta G}$ does not significantly decrease the performance of the RBFE predictions (Table 1), but increases absolute affinity prediction (Table A3). However, removing $\mathcal{L}_{\Delta G}$ or $\mathcal{L}_{\text{consistency}}$ drops the performance of the RBFE predictions by a considerable margin. The removal of $\mathcal{L}_{rotation}$ increases the performance of the Siamese Network, indicating that data augmentation is all that is required to provide the necessary rotational invariance. When we remove $\mathcal{L}_{\Delta\Delta G}$ and $\mathcal{L}_{\text{consistency}}$, the Siamese Network no longer provides predictions that are correlated with the experimental affinity values, however, the errors of the predictions are only slightly increased from the baseline.

Altering the Siamese Network architecture does not affect performance as much as removing components of the loss function. If we exchange the latent space subtraction of the Siamese Network for a concatenation, we do not see any change in performance of the model. If we train a convolutional architecture to predict the absolute affinity values using the same training set, we see the correlation to the experimental RBFE drops slightly and the error of predictions increases slightly as well.

3.3 Generalization to new Protein Families

When the Default2018 Siamese Network is trained on all of the BindingDB dataset and evaluated on a left out protein family, we find that the RBFE prediction varies widely across the protein families (Figure A5). The protein families that have high correlations tend to have low error (Figure A6 and A7). The average Pearson's R correlation coefficient across every protein family is 0.24; lower than the correlation on the additional ligands dataset with the least information for each congeneric series.

4 Discussion and Conclusion

Our models show higher correlation with experimental RBFE and lower errors of prediction than the model developed by Jiménez-Luna et al. [11]. We see an increase in model performance as the amount of information about each congeneric series is increased. Our highest parameter CNN architecture, Dense, was able to outperform the lower parameter Default2018 architecture on the smallest training set. However, the Dense model is initialized with weights from a absolute binding affinity prediction task (A.3) which provides the model with much greater initial knowledge of the problem than a randomly initialized network. When using the Dense architecture with random initialization, the model had lower RBFE prediction performance than the randomly intitialized Default2018 model (results not shown).

Only some components of the loss function are contributing to the models RBFE prediction performance. The removal of $\mathcal{L}_{\Delta\Delta G}$ does not have a large impact on model performance indicating $\mathcal{L}_{\Delta G}$ and $\mathcal{L}_{\text{consistency}}$ contribute significantly to the RBFE prediction performance. The removal of the $\mathcal{L}_{\text{rotation}}$ component increased the performance of the model, which may indicate some isotropic properties of the network architecture. The latent space structure imposed by the subtraction operation did not result in improved performance when using 2-permutations for training. The Siamese

architecture enables understanding of ordering within congeneric series, which is ignored when only training for absolute affinity prediction.

Despite good intra-congeneric series performance, our Siamese Network does not generalize well to new protein families suggesting the approach is best used later in the lead optimization process. Work still needs to be done on both absolute and relative binding affinity predictors to ensure that they are learning robust models of the intermolecular interactions.

Acknowledgments and Disclosure of Funding

The authors thank Paul Francoeur and Monica Dayao for their comments during the preparation of the manuscript.

This work is supported by R35GM140753 from the National Institute of General Medical Sciences and CHE-2102474 from the National Science Foundation.

References

[1] Lingle Wang, Jennifer Chambers, and Robert Abel. Protein–ligand binding free energy calculations with fep+. In *Biomolecular Simulations*, pages 201–232. Springer, 2019.

[2] Ercheng Wang, Huiyong Sun, Junmei Wang, Zhe Wang, Hui Liu, John ZH Zhang, and Tingjun Hou. End-point binding free energy calculation with mm/pbsa and mm/gbsa: strategies and applications in drug design. *Chemical reviews*, 119(16):9478–9508, 2019.

[3] Robert Abel, Lingle Wang, Edward D Harder, BJ Berne, and Richard A Friesner. Advancing drug discovery through enhanced free energy calculations. *Accounts of chemical research*, 50 (7):1625–1632, 2017.

[4] Lingle Wang, Yujie Wu, Yuqing Deng, Byungchan Kim, Levi Pierce, Goran Krilov, Dmitry Lupyan, Shaughnessy Robinson, Markus K Dahlgren, Jeremy Greenwood, et al. Accurate and reliable prediction of relative ligand binding potency in prospective drug discovery by way of a modern free-energy calculation protocol and force field. *Journal of the American Chemical Society*, 137(7):2695–2703, 2015.

[5] Oleg Trott and Arthur J Olson. Autodock vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *Journal of computational chemistry*, 31(2):455–461, 2010.

[6] Paul G Francoeur, Tomohide Masuda, Jocelyn Sunseri, Andrew Jia, Richard B Iovanisci, Ian Snyder, and David R Koes. Three-dimensional convolutional neural networks and a cross-docked data set for structure-based drug design. *Journal of Chemical Information and Modeling*, 60(9):4200–4215, 2020.

[7] Evan N Feinberg, Debnil Sur, Zhenqin Wu, Brooke E Husic, Huanghao Mai, Yang Li, Saisai Sun, Jianyi Yang, Bharath Ramsundar, and Vijay S Pande. Potentialnet for molecular property prediction. *ACS central science*, 4(11):1520–1530, 2018.

[8] José Jiménez, Miha Skalic, Gerard Martinez-Rosell, and Gianni De Fabritiis. K deep: protein–ligand absolute binding affinity prediction via 3d-convolutional neural networks. *Journal of chemical information and modeling*, 58(2):287–296, 2018.

[9] Marta M Stepniewska-Dziubinska, Piotr Zielenkiewicz, and Pawel Siedlecki. Development and evaluation of a deep learning model for protein–ligand binding affinity prediction. *Bioinformatics*, 34(21):3666–3674, 2018.

[10] Pedro J Ballester and John BO Mitchell. A machine learning approach to predicting protein–ligand binding affinity with applications to molecular docking. *Bioinformatics*, 26(9):1169–1175, 2010.

[11] José Jiménez-Luna, Laura Pérez-Benito, Gerard Martínez-Rosell, Simone Sciabola, Rubben Torella, Gary Tresadern, and Gianni De Fabritiis. Deltadelta neural networks for lead optimization of small molecule potency. *Chemical science*, 10(47):10911–10918, 2019.

[12] Jane Bromley, James W Bentz, Léon Bottou, Isabelle Guyon, Yann LeCun, Cliff Moore, Eduard Säckinger, and Roopak Shah. Signature verification using a "siamese" time delay neural network. *International Journal of Pattern Recognition and Artificial Intelligence*, 7(04):669–688, 1993.

[13] Yann LeCun, Bernhard Boser, John Denker, Donnie Henderson, Richard Howard, Wayne Hubbard, and Lawrence Jackel. Handwritten digit recognition with a back-propagation network. *Advances in neural information processing systems*, 2, 1989.

[14] Tiqing Liu, Yuhmei Lin, Xin Wen, Robert N Jorissen, and Michael K Gilson. Bindingdb: a web-accessible database of experimentally determined protein–ligand binding affinities. *Nucleic acids research*, 35(suppl_1):D198–D201, 2007.

[15] Lingxue Song, Dihong Gong, Zhifeng Li, Changsong Liu, and Wei Liu. Occlusion robust face recognition based on mask learning with pairwise differential siamese network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 773–782, 2019.

[16] Martin Simonovsky and Joshua Meyers. Deeplytough: learning structural comparison of protein binding sites. *Journal of chemical information and modeling*, 60(4):2356–2366, 2020.

[17] Gregory Koch, Richard Zemel, Ruslan Salakhutdinov, et al. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, volume 2. Lille, 2015.

[18] Jocelyn Sunseri and David R Koes. libmolgrid: Graphics processing unit accelerated molecular gridding for deep learning applications. *Journal of Chemical Information and Modeling*, 60(3): 1079–1084, 2020.

[19] Andrew T McNutt, Paul Francoeur, Rishal Aggarwal, Tomohide Masuda, Rocco Meli, Matthew Ragoza, Jocelyn Sunseri, and David Ryan Koes. Gnina 1.0: molecular docking with deep learning. *Journal of cheminformatics*, 13(1):1–20, 2021.

[20] Jaina Mistry, Sara Chuguransky, Lowri Williams, Matloob Qureshi, Gustavo A Salazar, Erik LL Sonnhammer, Silvio CE Tosatto, Lisanna Paladin, Shriya Raj, Lorna J Richardson, et al. Pfam: The protein families database in 2021. *Nucleic Acids Research*, 49(D1):D412–D419, 2021.

[21] Russell Spitzer and Ajay N Jain. Surflex-dock: Docking benchmarks and real-world application. *Journal of computer-aided molecular design*, 26(6):687–699, 2012.

[22] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings, 2010.

[23] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

A Appendix

A.1 Model Architecture

We use the default values of the libmolgrid python library to voxelize our receptor and ligand data. This creates 14 channels for both the receptor and ligand with the layout of the channels provided in Tables A1 and A2. A 23.5 Å cubic grid is constructed around the center of the ligand with a grid resolution of 0.5 Å.

The Default2018 and Dense architecture are shown in Figure A1.

A.2 Data Filtering

First the dataset is split into three different groups, one for each of the measures of potency (IC_{50}, K_d,K_i). A ligand can be in multiple groups if it has binding affinity measurements for multiple measures of potency. For each split, we strip any greater than (>) or less than(<) symbols from the binding affinity measurements of every ligand and use the remaining string as the exact binding affinity value. If a ligand has multiple measurements for a given measure of potency, we delete the

Channel Number	Type
0	Aliphatic CarbonXS Hydrophobe
1	Aliphatic CarbonXS Non-Hydrophobe
2	Aromatic CarbonXS Hydrophobe
3	Aromatic Carbon Non-Hydrophobe
4	Bromine, Iodine, Chlorine, Fluorine
5	Nitrogen, Nitrogen XS Acceptor
6	Nitrogen XS Donor, Nitrogen XS Donor/Acceptor
7	Oxygen, Oxygen XS Acceptor
8	Oxygen XS Donor/Acceptor, Oxygen XS Donor
9	Sulfur, Sulfur Acceptor
10	Phosphorus
11	Calcium
12	Zinc
13	GenericMetal, Boron, Manganese, Magnesium, Iron

Table A1: Receptor channel atom types

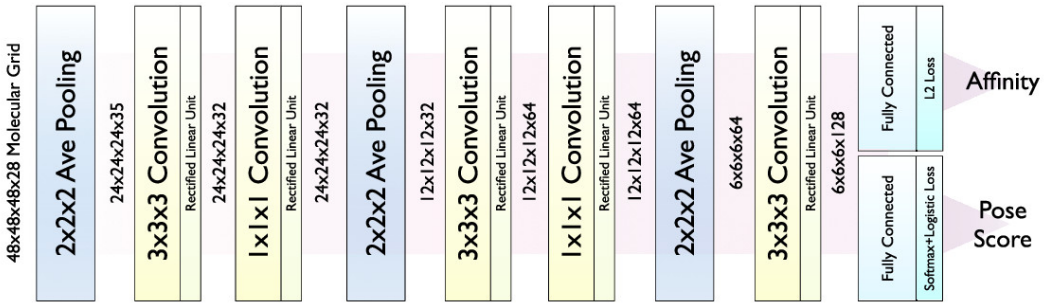
Channel Number	Type
0	Aliphatic Carbon XS Hydrophobe
1	Aliphatic Carbon XS Non-Hydrophobe
2	Aromatic Carbon XS Hydrophobe
3	Aromatic Carbon XS Non-Hydrophobe
4	Bromine, Iodine
5	Chlorine
6	Fluorine
7	Nitrogen, Nitrogen XS Acceptor
8	Nitrogen XS Donor, Nitrogen XS Donor/Acceptor
9	Oxygen, Oxygen XS Acceptor
10	Oxygen XS Donor/Acceptor, Oxygen XS Donor
11	Sulfur, Sulfur Acceptor
12	Phosphorus
13	GenericMetal, Boron, Manganese,Magnesium, Zinc, Calcium, Iron

Table A2: Ligand channel atom types

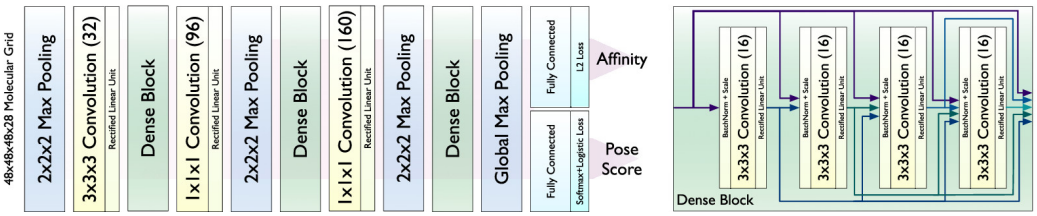
ligand from that measure of potency split if the range of the measurements is greater than one log unit. Otherwise, we take the median of the multiple measurements. After this filtering, we remove any ligands that have binding affinity information for a PDB ID that has no other ligands with binding affinity measurements. We then construct congeneric series by creating ordered pairs of ligands that have the same receptor and the same measure of potency (IC_{50}, K_d,K_i), we utilize the log-converted measurements ($-\log_{10}(\text{value})$), referred to as "pK", for each measure of potency. We next define a reference ligand. Each ligand in has a bound structure that is either the crystal pose or a pose determined via computational template docking to the protein using the Suflex docking software and the crystal ligand.[21] The reference ligand is assigned as the ligand with the highest Tanimoto similarity to the ligand used for the template docking, usually the ligand in the crystal that was used for template docking.

A.3 Hyperparameters and Training Information

During the training of our model we set $\alpha = 10$ and $\beta, \gamma, \delta = 1$. The Default2018 architecture's weights are initialized with the Xavier uniform method[22] and the biases are initialized to 0. The Dense model is initialized with weights and biases learned by training for ΔG prediction and pose selection.[6] We found that randomly initialized Dense models performed worse than Default2018 models and pretrained Default2018 models performed worse than randomly initialized Default2018 models (results not shown). All models are trained using the Adam stochastic gradient descent optimizer[23] with the default parameters ($\beta_1 = 0.9$,$\beta_2 = 0.999$,$\epsilon = 1 \times 10^{-8}$). Models are trained



(a) Default2018



(b) Dense

Figure A1: Architectures of the Default2018 and Dense models. The numbers in parenthesis indicate the output number of features from the convolutions.

Table A3: Ablation study of the Siamese Network with respect to absolute affinity prediction. Parentheses indicate the ± 1 standard deviation of the 25 train/test versions. **Bold** indicates results are not significantly different from Standard ($p > 0.05$).

Ablation	Pearson's R	RMSE (pK)	MAE (pK)
Standard	$\mathbf{0.864(\pm 0.0124)}$	$\mathbf{0.841(\pm 0.0377)}$	$\mathbf{0.556(\pm 0.0303)}$
No $L_{\Delta\Delta G}$	$0.873(\pm 0.0057)$	$0.814(\pm 0.018)$	$0.517(\pm 0.0136)$
No $L_{\Delta G}$	$0.0278(\pm 0.107)$	$101(\pm 34.4)$	$101(\pm 34.5)$
No L_{rotation}	$\mathbf{0.866(\pm 0.0059)}$	$\mathbf{0.833(\pm 0.0169)}$	$\mathbf{0.553(\pm 0.0141)}$
No $L_{\text{consistency}}$	$\mathbf{0.866(\pm 0.00559)}$	$\mathbf{0.833(\pm 0.0171)}$	$\mathbf{0.559(\pm 0.0158)}$
No $L_{\Delta\Delta G}$, $L_{\text{consistency}}$	$0.873(\pm 0.00425)$	$0.814(\pm 0.0132)$	$0.514(\pm 0.0121)$
No $L_{\Delta G}$, $L_{\text{consistency}}$	$-0.0118(\pm 0.0494)$	$6.65(\pm 0.465)$	$6.44(\pm 0.48)$
Concatenation	$\mathbf{0.87(\pm 0.00477)}$	$\mathbf{0.824(\pm 0.0155)}$	$0.531(\pm 0.0158)$
No Siamese Network	$0.846(\pm 0.0175)$	$0.883(\pm 0.0479)$	$0.616(\pm 0.0442)$

for 1000 epochs with a learning rate of 0.000367 and a scheduler that reduces the learning rate by a factor of 0.7 whenever the loss plateaus for more than 20 epochs. Data augmentation is achieved by randomly rotating and translating the inputs with a maximum translation of 2 Å from the center of mass of the ligand.

A.4 Additional Results

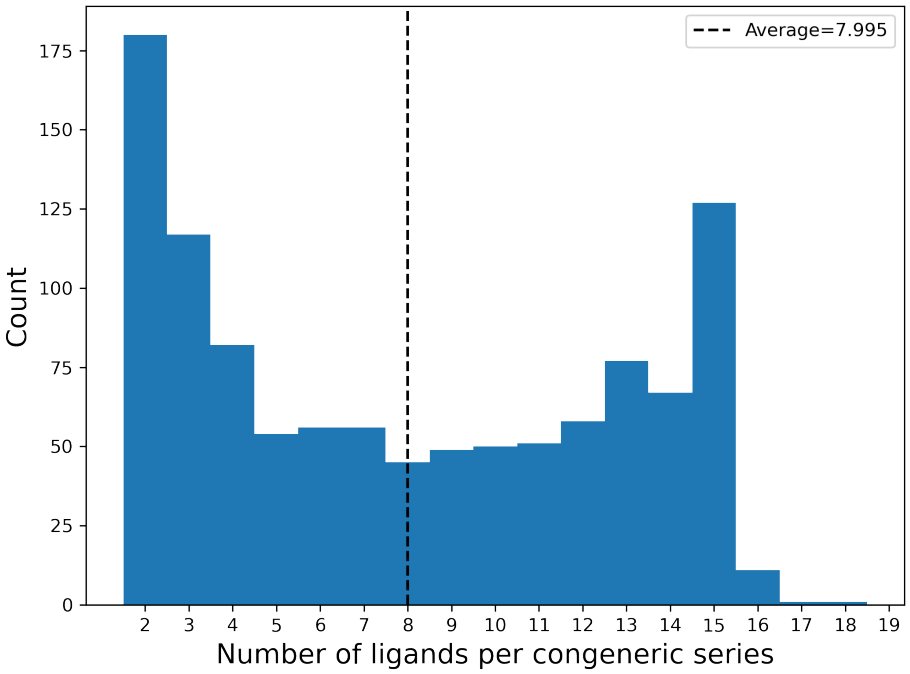


Figure A2: Number of ligands per congeneric series

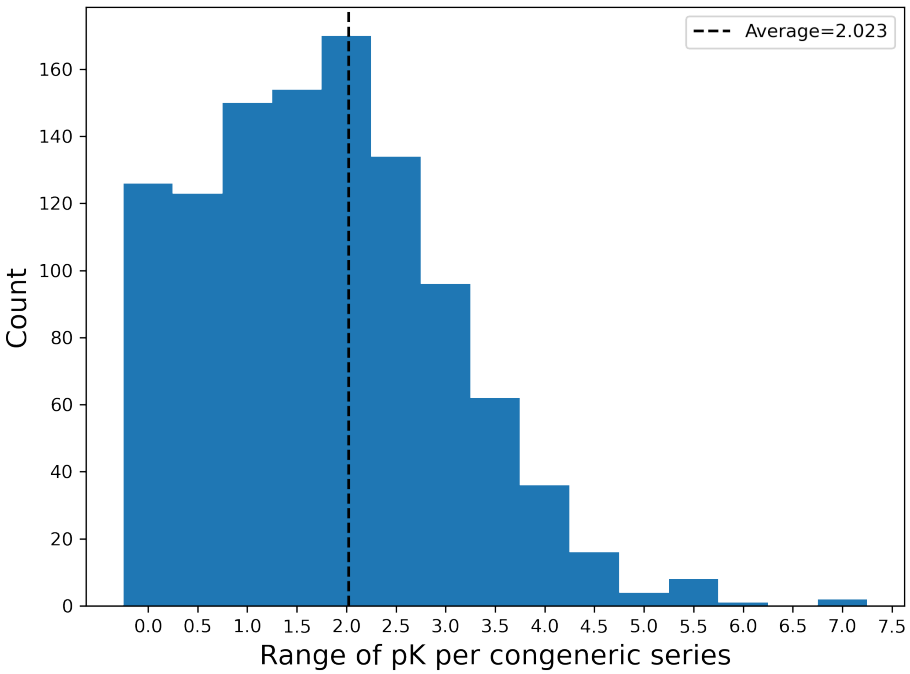


Figure A3: Range of absolute binding affinity per congeneric series in pIC_{50}

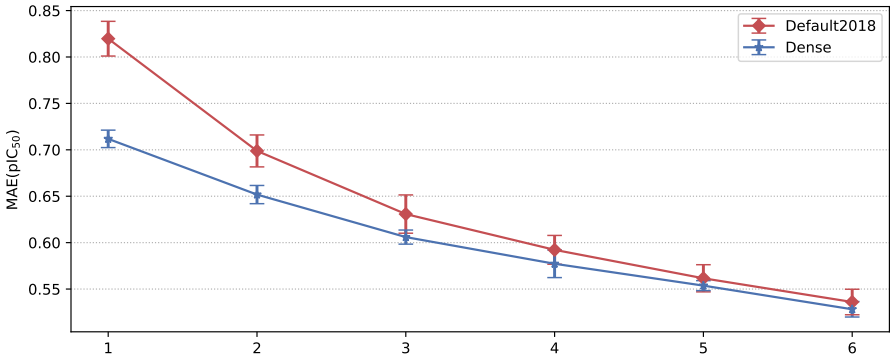


Figure A4: Mean absolute error (MAE) of $\Delta\Delta G$ for the additional ligands dataset for our two CNN architectures. Error bars indicate ± 1 standard deviation of 25 versions (only 5 for Dense) of the train-test split.

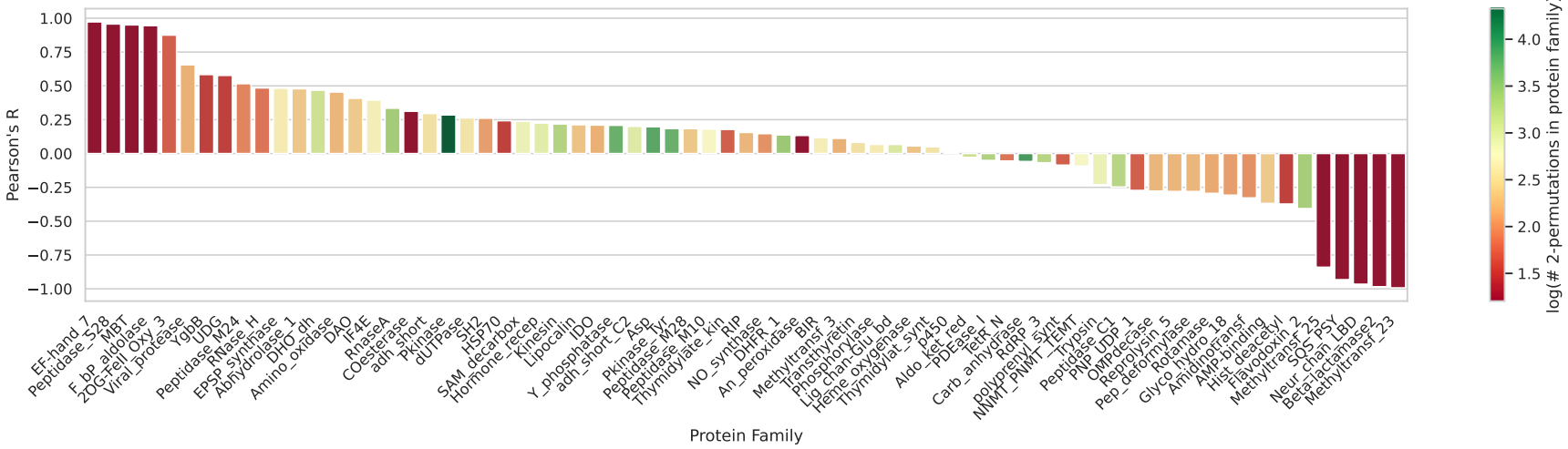


Figure A5: Pearson's R of $\Delta\Delta G$ for each protein family in the BindingDB dataset when we evaluate on the remaining data with the Default2018 architecture. Bars are colored by the log of the number of two-permutations in the left out protein family

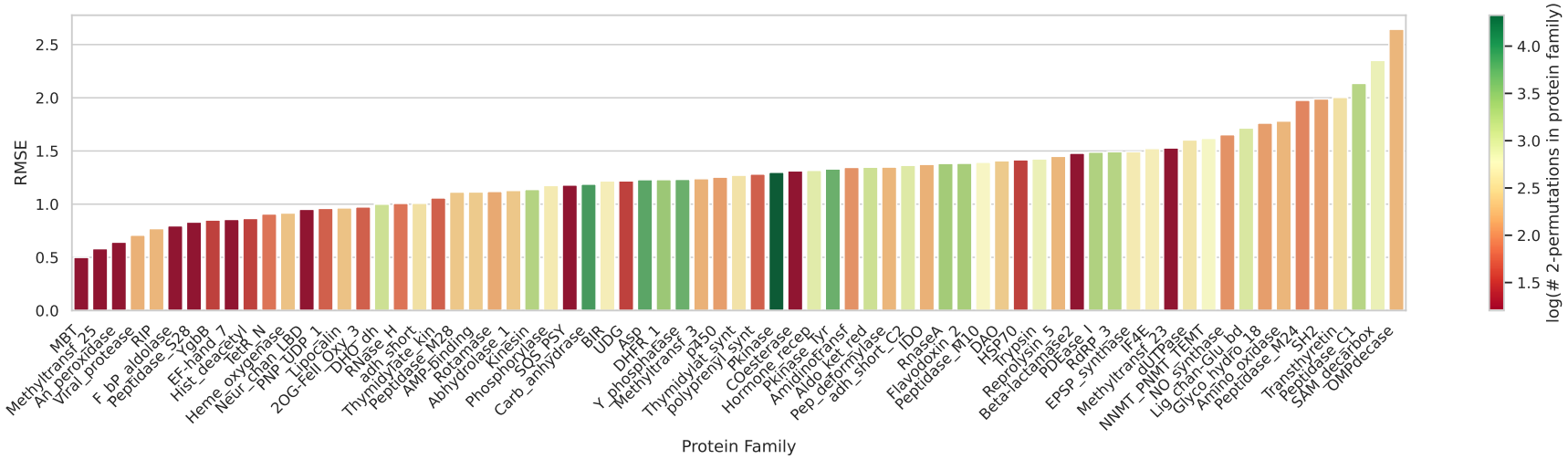


Figure A6: RMSE of $\Delta\Delta G$ for each protein family in the BindingDB dataset when we evaluate on the remaining data with the Default2018 architecture. Protein families with high correlations generally have low error. Bars are colored by the log of the number of two-permutations in the left out protein family

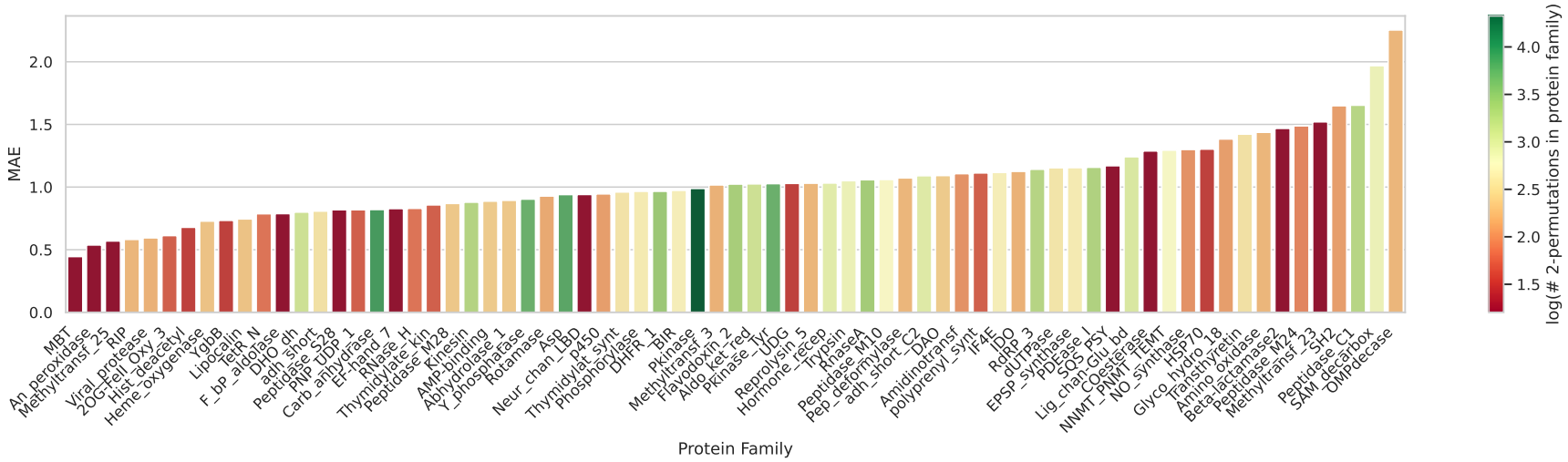


Figure A7: MAE of $\Delta\Delta G$ for each protein family in the BindingDB dataset when we evaluate on the remaining data with the Default2018 architecture. Protein families with high correlations generally have low error. Bars are colored by the log of the number of two-permutations in the left out protein family