
Generalizing Equivariant Neural Networks for Protein–Ligand Binding by Comparing Atomic Environments

Ryan Wong

Department of Physics
University of Chicago
wongryan@uchicago.edu

Chetan Rupakheti

AbbVie Inc.
chetan.rupakheti@abbvie.com

Abhishek Pandey

AbbVie Inc.
abhishek.pandey@abbvie.com

Arvind Ramanathan

Argonne National Laboratory
ramanathana@anl.gov

Risi Kondor

Department of Computer Science
University of Chicago
risi@cs.uchicago.edu

Abstract

Structure-based models of protein–ligand binding affinity have long been benchmarked by training on PDBbind and testing on CASF-2016, a train/test split in which nearly half of the test complexes have close analogs in the training data. Many recent models developed on splits that remove these analogs rely on pre-trained embeddings. We ask what additional inductive biases should be built into an equivariant message passing network (MPNN) trained from scratch, and focus on how the network can use the unbound state of a complex. In a basic equivariant MPNN, adding the unbound state as a second pooled input does not improve accuracy. Because the two states contain the same atoms, their embeddings can instead be combined at each atom, at each intermolecular edge and over the pocket, through operations we refer to as paired readouts. The resulting model, PEAR, is to our knowledge the most accurate model on CASF-2016 when trained on CleanSplit, including models with pretrained embeddings. Furthermore, a variant of PEAR, PEAR_{LBA}, matches the best reported accuracy on ATOM3D LBA at 30% sequence identity. On seven held-out pocket clusters, a residue-scale variant, PEAR_{residue}, has the highest average correlation of any single model after finetuning on 25 complexes of each target. When developing PEAR, we were additionally inspired by the fact that binding affinity is an ensemble average over configurations, so that a model should not lean too heavily on the exact coordinates of its input structure. We find that PEAR is the most accurate atom-scale model under coordinate noise perturbation, and that PEAR_{residue} degrades about half as much under the same perturbation.

1 Introduction

For the past decade, the dominant evaluation protocol for structure-based scoring functions has been to train a neural network on PDBbind and test on the CASF-2016 test set. Unfortunately, this train/test split suffers from data leakage of close structural, chemical and protein-family analogs [1, 2]. Graber et al. [1] introduced PDBbind CleanSplit, a PDBbind split in which complexes similar to any CASF test example are removed (Appendix D). Under these criteria, nearly half of CASF-2016 turned out to have close analogs in the original training data. When Pafnucy [3], which had previously achieved state of the art on the CASF test set, was retrained on CleanSplit, its CASF

performance came close to that of a nearest-neighbour method that averages the affinities of the five most similar training complexes. The ATOM3D benchmark [4] addressed this problem by splitting its ligand binding affinity (LBA) task at 60% and 30% protein sequence identity thresholds, and on this benchmark model performance is lower on LBA30 than on LBA60. These works suggest that the best-performing models on the original benchmark all, to some extent, overfit to the training set.

Many recent models developed on decontaminated splits have, in one way or another, relied on pretrained embeddings. For example, GEMS [1], the model proposed alongside CleanSplit, augments a graph network with protein and ligand language model embeddings, while GatorAffinity [5], an SE(3)-equivariant network, is pretrained on 1.5 million synthetic complexes. While this showcases the potential of pretraining and self-supervised learning, their advantages may obscure the importance of developing architectures that can better generalize. Because we cannot guarantee that a pretraining task covers any given test example, we believe it is complementary to ask which inductive biases are worth utilizing to achieve good performance on decontaminated splits.

One inductive bias that has become a default in physics and chemistry is equivariance. Equivariant neural networks explicitly enforce symmetries such as rotation and translation invariance, and their most general form expresses its variables in the generalized Fourier space of the underlying group. In the Fourier-space model, each atom’s neighborhood is expanded in spherical harmonics of increasing angular order ℓ , and the resulting features are coupled through the Clebsch–Gordan tensor product to form messages. Networks of this form are among the most accurate methods for molecular property prediction and interatomic force fields [6, 7]. Following their success on small molecules, there has been considerable effort to apply equivariant architectures to binding affinity prediction [5, 8–10]. However, the Fourier-space formalism, combined with the message passing paradigm, has not separated itself from less physically rich models such as 2D graph and sequence-based networks [11, 12], voxel-based convolutional networks [3, 13, 14], or handcrafted interaction descriptors [15]. Moreover, models that incorporate message passing as a component, equivariant or not, are frequently *hybrid* architectures that combine it with convolutional networks [14, 16], with attention [17, 18], or with language model embeddings [1].

The reason behind this phenomenon is that interatomic potentials and binding affinity depend on the atomic structure in fundamentally different ways. The potential energy is a function of a single configuration, which makes it sufficient to include just one molecular structure per training or test example. It is furthermore well approximated by a sum of atomic contributions that depend only on neighbouring atoms within a cutoff [19]. Baking this decomposition into machine-learned interatomic potentials (MLIPs) accordingly allows one network to be applied to molecules of different sizes and compositions [7]. MLIPs therefore have a basis to generalize across system sizes, provided the training data covers the local environments they encounter. Equivariance and invariance can then be enforced on top of this to guarantee that forces and energies transform correctly under rotations and translations, removing predictions that are unphysical in this sense [6]. On the other hand, binding free energy is an average over an ensemble of configurations [20], whereas the input crystal structure is usually a single, averaged representation of the bound state. One way to picture the consequence of this mismatch is that, given just a single input structure, the network has to do some amount of *guesswork* about the parts of binding affinity that depend on the ensemble.

Practically, without the strict guidance that MLIPs have on the atomic structure, many modeling choices have become empirically interchangeable in the context of neural network scoring functions on benchmark datasets. For example, in networks that incorporate structure, the unbound state can be omitted, and the network then learns how the bound structure correlates with binding affinity. Models that encode both states either subtract one embedding from the other [21–23], or combine the two with a more general MLP [24]. The unbound state itself is usually obtained by removing intermolecular edges. Despite some choices deviating from the physical basis of binding affinity, all of these models work reasonably well relative to the literature at their time of publication. This is in contrast to MLIPs, which have converged on a single mathematical decomposition, and largely on a single type of architecture.

Equivariant message passing remains of interest to the binding affinity prediction community, both as a component of larger pipelines [5, 9] and in pursuit of models that generalize to unseen targets [23]. Independent of improving the equivariant encoder itself [6, 7], one can add learnable modules that use the encoder’s output as an abstract input representation of the structure, as was done in CheapNet [18]. If the unbound state of the complex is available, these modules can addi-

tionally combine bound and unbound embeddings at different levels of granularity, such as at the node or graph level. Doing so would produce “paired” embeddings using information that is sometimes irretrievably lost when the bound and unbound graphs are pooled. To our knowledge this axis has not been explored.

Following this line of thinking, we propose PEAR (Paired Embeddings of Atoms and Residues), an equivariant message passing network with a set of *paired readout* modules that combine the embeddings of the bound and unbound copies of a complex.

One additional intuition that guided us in designing PEAR is that because binding affinity is an ensemble average, a model should not lean too heavily on the exact coordinates of the input structure. In an equivariant network, higher- ℓ features are higher-degree functions of the interatomic directions, so a small displacement of an atom changes them more than it changes the scalar terms. This suggests two conflicting routes to designing a neural network for this task. In particular, networks can either represent relative orientations between atoms in great detail using higher- ℓ representations of their environment [3, 25–29], or discard details that overcommit to the exact configuration of the training example, for example by dispensing with relative orientations in favor of distances [1, 30, 31]. In this work, we account for both routes, hypothesizing that a network with higher- ℓ features that is robust to noise generalizes better than one with only one of the two properties. This manifested as two interventions on the neural network. First, we computed higher- ℓ terms from a separate spherical harmonics expansion of the local atomic environment *after* the message passing backbone, which are fed as inputs to the paired readouts of PEAR. Second, we coarse-grained the pocket into residues, with the B-factor and amino acid identity as additional features.

In summary, this work makes the following contributions.

- We show, in a basic equivariant message passing network (MPNN), that adding the unbound copy as a second pooled input does not improve accuracy.
- We measure the effect of combining node, edge and graph level features between the two states, on equivariant MPNNs trained on CleanSplit, through sweeps and ablations.
- We find parameterizations within this framework that match or exceed the best reported accuracy on CleanSplit, on ATOM3D LBA, and on a recently developed set of seven held-out pocket clusters, without pretraining.
- We measure how these interventions affect the coordinate noise robustness of models, to test its value as a heuristic for designing MPNNs for binding affinity prediction.

2 Method

Problem definition. We predict the binding affinity of a protein–ligand complex from its crystal structure. The complex is a graph $G = (V, E) = (V_l \cup V_p, E_l \cup E_p \cup E_{\text{inter}})$, where V_l and V_p are the heavy atoms of the ligand and of the pocket. Each node $i \in V$ has a feature vector $x_i \in \mathbb{R}^d$ and a coordinate $r_i \in \mathbb{R}^3$. The sets E_l and E_p are covalent bonds within the ligand and within the pocket, and E_{inter} the intermolecular pairs within a cutoff. The target $y \in \mathbb{R}$ is the measured affinity on a logarithmic scale, $-\log K_d$ or $-\log K_i$, where K_d and K_i are the dissociation and inhibition constants. We train f to minimise the squared error between $\hat{y} = f(G)$ and y , with f invariant to rotation and translation of $\{r_i\}$ and to permutation of V .

Bound and unbound states. We also form an unbound copy of each complex by removing E_{inter} from the graph. In a preliminary experiment on a basic equivariant MPNN, we find that pooling each copy separately and concatenating the two pooled embeddings as inputs to an MLP is no more accurate than a single copy across depth and ℓ_{max} (Appendix I). Although the unbound graph is a strict subgraph of the bound graph and therefore redundant from the perspective of a sufficiently expressive encoder, the bound state h_i^b and the unbound state h_i^{ub} differ if there are atoms belonging to the other molecule within the message-passing radius, and mathematical relationships between them contain information about how the local environment changes upon binding.

Paired readouts. Because the two copies contain the same atoms, each node and each intermolecular edge can be combined with its counterpart directly. A paired readout combines the two copies at a site, which may be a node, an intermolecular edge, or the pocket as a whole, reduces the result to

scalars, and aggregates over the sites of its level (Table 5, Appendix E). The scalars are concatenated to form the input of an MLP head g_θ ,

$$\hat{y} = g_\theta \left[\underbrace{\mathcal{A}(h^b)}_{\text{bound}}; \underbrace{\sum_i \Phi^\rho(\tilde{h}_i^b, \tilde{h}_i^{ub})}_{\text{atom}}; \underbrace{\sum_{ij \in E_{\text{inter}}} \psi(e_{ij}^b, e_{ij}^{ub})}_{\text{edge}}; \underbrace{[F^b \otimes F^{ub}]_0}_{\text{pocket}} \right], \quad (1)$$

where $\mathcal{A}(h^b)$ is the pooled output of the encoder on the bound graph, the only input to the head that does not involve the unbound copy. The atom readout $\Phi^\rho = [[\tilde{h}_i^b \otimes_{\text{CG}} \tilde{h}_i^{ub}]_L \otimes \tilde{h}_i^{ub}]_0$, which we denote $\otimes_{\text{CG}}^\rho$, acts on the expansions $\tilde{h}$ of Eq. 2. Its $(\ell, \ell) \rightarrow 0$ path is the SOAP overlap [32, 33] of the bound and unbound environments of atom i , which differs from the unbound self-overlap only when atoms of the partner molecule are within the cutoff. The edge readout ψ and the pocket term are defined in Appendix E.

Noise robustness. Both h^b and h^{ub} are produced by an equivariant message-passing backbone (Appendix B), in which each layer takes a tensor product of the node features with the spherical harmonics of the bond directions. Models with a higher ℓ_{max} in the backbone are less robust to coordinate noise (Appendix C, Figure 1). We therefore make two changes to the architecture to improve its noise robustness. First, we devise a module that produces learned higher- ℓ features without undergoing message passing. In general, this can be achieved by allowing the prediction head to depend on an additional spherical harmonics expansion. In our case, we use the scalar outputs $h_j^{(T,0)}$ of the final message-passing layer T as learned weights in an angular and radial expansion of each atom’s local environment $\mathcal{N}(i)$,

$$\tilde{h}_i^{(c,\ell,m)} = \sum_{j \in \mathcal{N}(i)} (Wh_j^{(T,0)})_c R_\ell(r_{ij}) Y_\ell^m(\hat{\mathbf{r}}_{ij}), \quad (2)$$

where W is a learnable projection with output channels c and R_ℓ is a radial basis. The pocket representation F is expanded the same way, centered at the ligand centroid over V_p . Second, we coarse-grain the pocket to one node per residue, mean-pooling each residue’s atomic features and adding a learned amino-acid embedding and a z -scored B-factor. Averaging the N atoms of a residue into one node reduces the centroid’s positional Gaussian noise by $\sqrt{N}$. We refer to this model as PEAR_{residue}, and to PEAR without paired readouts as PEAR_{backbone}.

3 Experiments

Evaluation protocols. We evaluate our architecture under three protocols, each of which removes a different kind of train–test similarity (Appendices D and J). *PDBbind CleanSplit* [1] removes training complexes similar to a CASF-2016 test complex, so it tests generalization to new complexes of largely familiar targets. *ATOM3D LBA* [4] partitions the PDBbind v2019 refined set so that test proteins share at most 30% (or 60%) sequence identity with training proteins. The *held-out pocket clusters* of Kopko et al. [34] remove the pocket community of each of seven CASF-2016 targets from training, which tests generalization to new binding-site geometry, and we additionally finetune each model on 25 labeled complexes of the held-out target.

Experiments on CleanSplit. We train our models on CleanSplit, which consists of 13,166 complexes. Table 1 reports 5-fold cross-validated ensembles, formed by averaging the predictions of the five fold models, on CASF-2016 and on its independent subset, the 144 complexes that fall below the CleanSplit similarity thresholds against all of PDBbind [1]. We find that PEAR is the most accurate method on CASF-2016, and PEAR_{residue} has the highest correlation on the independent subset. Any paired readout improves on PEAR_{backbone}, by 0.015–0.093 RMSE (Appendices F, G). Ensembling the two scales lowers the RMSE to 1.233.

ATOM3D LBA. PEAR_{LBA}, a variant of PEAR adjusted in four settings, has the lowest RMSE and highest Pearson correlation on LBA 30%, within one standard deviation of CheapNet. PEAR also displays the highest Spearman correlation on LBA 60% (Appendix J).

Robustness to coordinate noise. We re-score complexes in the test set after perturbing each atom with isotropic Gaussian noise of standard deviation σ (Appendix A). Models with a higher ℓ_{max} in

Table 1: PDBbind CleanSplit, 5-fold CV ensembles. Baseline numbers come from Kopko et al. [34] and Sim and Lee [17], and the GEMS variants include frozen language-model and ATOMICA embeddings. Best in **bold**. Depth and readout variants are in Appendix K.

Method	Params	CASF-2016		Independent	
		RMSE ↓	r ↑	RMSE ↓	r ↑
GenScore	1.06M	1.280	0.807	1.580	0.737
GEMS	1.03M	1.274	0.815	1.347	0.828
GEMS + ATOMICA	1.03M	1.292	0.808	1.361	0.822
BA-Pred	17.9M	1.290	0.818	—	—
PEAR	1.68M	1.258	0.834	1.414	0.822
PEAR _{residue}	831K	1.282	0.822	1.383	0.833
+ joint ensemble	—	1.233	—	—	—

Table 2: Held-out pocket clusters, mean and worst-case Pearson r before (Pre-FT) and after finetuning on 25 labeled complexes (FT-25). Avg. and Min. are the mean and the lowest per-cluster r across the seven targets. Per-cluster values are in Appendix M. Baselines from Kopko et al. [34].

Method	Pre-FT		FT-25	
	Avg. r	Min. r	Avg. r	Min. r
GEMS	0.470	0.236	0.537	0.386
GEMS + ATOMICA	0.498	0.311	0.550	0.385
PEAR _{backbone}	0.434	0.219	0.518	0.305
PEAR	0.452	0.197	0.541	0.340
PEAR _{residue} ($\ell=1$)	0.447	0.235	0.575	0.396
PEAR _{residue} ($\ell=2$)	0.461	0.242	0.580	0.367
PEAR + PEAR _{residue}	—	—	0.599	0.388

the backbone are more sensitive to noise (Appendix C). PEAR is, on average, the most accurate atom model up to $\sigma = 1 \text{ \AA}$ and maintains a high robustness to noise up to $\sigma = 2 \text{ \AA}$ (Figure 2). At $\ell_{\max} = 0$, PEAR_{residue} degrades by about half as much as PEAR. The fact that PEAR_{residue} has a higher error on CleanSplit but degrades less drastically with noise suggests that it may excel at a different set of complexes in comparison to PEAR. A stratification of the CASF-2016 test set shows that PEAR is disproportionately accurate when the complex is rigid, as measured by RMSD from ApoBind conformations [35], and when the ligand is small. This suggests that averaging each prediction between the atom and residue variants may predictably reduce error. Indeed, combining their 5-fold ensembles into a single ensemble reduces the RMSE to 1.233 (Appendix H).

Experiments on held-out pocket clusters. Table 2 reports the mean and worst-case Pearson r over the seven clusters. We find that before finetuning the average correlation of our models is below that of GEMS, possibly because the protein language model embeddings of GEMS are trained on sequence databases that include the held-out families. Our models improve more with finetuning than GEMS does. After finetuning, the correlation of every model we trained is higher on each of the seven clusters, whereas that of GEMS is lower on four of them [34]. Of the finetuned models, PEAR_{residue} at $\ell=2$ has the highest average correlation of any single model.

4 Conclusion

We have shown that the generalization capabilities of an equivariant MPNN can be enhanced through additional paired readout modules that combine learned representations of the bound and unbound states at varying levels of granularity. We demonstrated, using CleanSplit, that the unbound copy of the complex is not being exploited in a basic MPNN, highlighting the advantages of moving beyond simple concatenation of graph-level vectors. The simplicity and modularity of this approach allow practitioners to experiment with other operations for combining the two states, to improve the backbone independently of the readouts, with MACE [7], or to train on synthetic complexes [5].

References

- [1] David Graber, Peter Stockinger, Fabian Meyer, Siddhartha Mishra, Claus Horn, and Rebecca Buller. Resolving data bias improves generalization in binding affinity prediction. *Nature Machine Intelligence*, 7(10):1713–1725, 2025. doi: 10.1038/s42256-025-01124-5.
- [2] Jie Li, Xingyi Guan, Oufan Zhang, Kunyang Sun, Yingze Wang, Dorian Bagni, and Teresa Head-Gordon. Leak proof PDBBind: A reorganized data set of protein–ligand complexes for more generalizable binding affinity prediction. *The Journal of Physical Chemistry B*, 130(2):730–740, 2026. doi: 10.1021/acs.jpcc.5c08598.
- [3] Marta M. Stepniewska-Dziubińska, 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.
- [4] Raphael J. L. Townshend, Martin Vögele, Patricia Suriana, Alexander Derry, Alexander Powers, Yianni Laloudakis, Sidhika Balachandar, Bowen Jing, Brandon Anderson, Stephan Eismann, Risi Kondor, Russ B. Altman, and Ron O. Dror. ATOM3D: Tasks on molecules in three dimensions. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021.
- [5] Jinhang Wei, Yupu Zhang, Peter A. Ramdhan, Zihang Huang, Gustavo Seabra, Zhe Jiang, Chenglong Li, and Yanjun Li. GatorAffinity: Boosting protein–ligand binding affinity prediction with large-scale synthetic structural data. *bioRxiv*, page 2025.09.29.679384, 2025. doi: 10.1101/2025.09.29.679384.
- [6] Simon Batzner, Albert Musaelian, Lixin Sun, Mario Geiger, Jonathan P. Mailoa, Mordechai Kornbluth, Nicola Molinari, Tess E. Smidt, and Boris Kozinsky. E(3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. *Nat. Commun.*, 13:2453, 2022. doi: 10.1038/s41467-022-29939-5.
- [7] Ilyes Batatia, Dávid Péter Kovács, Gregor N. C. Simm, Christoph Ortner, and Gábor Csányi. MACE: Higher order equivariant message passing neural networks for fast and accurate force fields. In *Adv. Neural Inf. Process. Syst.*, volume 35, pages 11423–11436, 2022.
- [8] Xiangying Zhang, Haotian Gao, Haojie Wang, Zhihang Chen, Zhe Zhang, Xinchong Chen, Yan Li, Yifei Qi, and Renxiao Wang. PLANET: A Multi-objective Graph Neural Network Model for Protein–Ligand Binding Affinity Prediction. *Journal of Chemical Information and Modeling*, 64(7):2205–2220, 2024. doi: 10.1021/acs.jcim.3c00253.
- [9] Ngoc-Quang Nguyen and Jaewoo Kang. EquiCPI: SE(3)-Equivariant Geometric Deep Learning for Structure-Aware Prediction of Compound-Protein Interactions. *Journal of Chemical Information and Modeling*, 65(13), 2025. doi: 10.1021/acs.jcim.5c00773. arXiv:2504.04654.
- [10] Gaili Li, Yongna Yuan, and Ruisheng Zhang. Relevance of 3D Rotationally Equivariant Neural Networks for Predicting Protein-Ligand Binding Affinities. *Interdisciplinary Sciences: Computational Life Sciences*, 18(2):392–418, 2026. doi: 10.1007/s12539-025-00745-z.
- [11] Hakime Öztürk, Arzucan Özgür, and Elif Ozkirimli. DeepDTA: deep drug–target binding affinity prediction. *Bioinformatics*, 34(17):i821–i829, 2018. doi: 10.1093/bioinformatics/bty593.
- [12] Ziduo Yang, Weihe Zhong, Lu Zhao, and Calvin Yu-Chian Chen. MGraphDTA: deep multiscale graph neural network for explainable drug–target binding affinity prediction. *Chemical Science*, 13(3):816–833, 2022.
- [13] José Jiménez, Miha Škalič, Gerard Martínez-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. doi: 10.1021/acs.jcim.7b00650.
- [14] Gregory W. Kyro, Rafael I. Brent, and Victor S. Batista. HAC-Net: A Hybrid Attention-Based Convolutional Neural Network for Highly Accurate Protein–Ligand Binding Affinity Prediction. *Journal of Chemical Information and Modeling*, 63(7):1947–1960, 2023. doi: 10.1021/acs.jcim.3c00251.
- [15] Norberto Sánchez-Cruz, José L. Medina-Franco, Jordi Mestres, and Xavier Barril. Extended connectivity interaction features: improving binding affinity prediction through chemical description. *Bioinformatics*, 37(10):1376–1382, 2021. doi: 10.1093/bioinformatics/btaa982.
- [16] Derek Jones, Hyojin Kim, Xiaohua Zhang, Adam Zemla, Garrett Stevenson, W. F. Drew Bennett, Daniel Kirshner, Sergio E. Wong, Felice C. Lightstone, and Jonathan E. Allen. Improved Protein–Ligand Binding Affinity Prediction with Structure-Based Deep Fusion Inference. *Journal of Chemical Information and Modeling*, 61(4):1583–1592, 2021. doi: 10.1021/acs.jcim.0c01306.

- [17] Jaemin Sim and Juyong Lee. BA-Pred and RMSD-Pred: Integrated Graph Neural Network Models for Accurate Protein–Ligand Binding Affinity and Binding Pose Prediction. *Journal of Chemical Information and Modeling*, 66(7):3480–3495, 2026. doi: 10.1021/acs.jcim.5c02591.
- [18] Hyukjun Lim, Sun Kim, and Sangseon Lee. CheapNet: Cross-attention on hierarchical representations for efficient protein–ligand binding affinity prediction. In *International Conference on Learning Representations (ICLR)*, 2025.
- [19] Jörg Behler and Michele Parrinello. Generalized Neural-Network Representation of High-Dimensional Potential-Energy Surfaces. *Physical Review Letters*, 98:146401, 2007.
- [20] Michael K. Gilson and Huan-Xiang Zhou. Calculation of protein-ligand binding affinities. *Annual Review of Biophysics and Biomolecular Structure*, 36:21–42, 2007.
- [21] Joseph Gomes, Bharath Ramsundar, Evan N. Feinberg, and Vijay S. Pande. Atomic Convolutional Networks for Predicting Protein-Ligand Binding Affinity. *arXiv preprint arXiv:1703.10603*, 2017.
- [22] Shuo Zhang, Yang Liu, and Lei Xie. A Universal Framework for Accurate and Efficient Geometric Deep Learning of Molecular Systems. *Scientific Reports*, 13:19171, 2023. doi: 10.1038/s41598-023-46382-8.
- [23] Krinos Li, Xianglu Xiao, Zijun Zhong, and Guang Yang. Accurate and Generalizable Protein-Ligand Binding Affinity Prediction With Geometric Deep Learning. *IEEE Open Journal of Engineering in Medicine and Biology*, 7:86–93, 2026. doi: 10.1109/OJEMB.2026.3667030. arXiv:2504.16261.
- [24] Xiaoyang Qu, Lina Dong, Ding Luo, Yubing Si, and Binju Wang. Water Network-Augmented Two-State Model for Protein–Ligand Binding Affinity Prediction. *Journal of Chemical Information and Modeling*, 64(7):2263–2274, 2024. doi: 10.1021/acs.jcim.3c00567.
- [25] Pedro J. Ballester and John B. O. Mitchell. A machine learning approach to predicting protein–ligand binding affinity with applications to molecular docking. *Bioinformatics*, 26(9):1169–1175, 2010.
- [26] Liangzhen Zheng, Jingrong Fan, and Yuguang Mu. OnionNet: a multiple-layer intermolecular-contact-based convolutional neural network for protein–ligand binding affinity prediction. *ACS Omega*, 4(14):15956–15965, 2019.
- [27] Shuangli Li, Jingbo Zhou, Tong Xu, Liang Huang, Fan Wang, Haoyi Xiong, Weili Huang, Dejing Dou, and Hui Xiong. Structure-aware interactive graph neural networks for the prediction of protein–ligand binding affinity. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 975–985, 2021.
- [28] Ziduo Yang, Weihe Zhong, Qiuji Lv, Tiejun Dong, and Calvin Yu-Chian Chen. Geometric interaction graph neural network for predicting protein–ligand binding affinities from 3D structures (GIGN). *The Journal of Physical Chemistry Letters*, 14(8):2020–2033, 2023.
- [29] Seokhyun Moon, Wonho Zhung, Soojung Yang, Jaechang Lim, and Woo Youn Kim. PIGNet: a physics-informed deep learning model toward generalized drug–target interaction predictions. *Chemical Science*, 13(13):3661–3673, 2022.
- [30] Julia Rahman, M A Hakim Newton, Mohammed Eunus Ali, and Abdul Sattar. Distance plus attention for binding affinity prediction. *Journal of Cheminformatics*, 16(1):52, 2024.
- [31] Benjamin P. Brown. A generalizable deep learning framework for structure-based protein–ligand affinity ranking. *Proceedings of the National Academy of Sciences*, 122(42):e2508998122, 2025. doi: 10.1073/pnas.2508998122.
- [32] Albert P. Bartók, Risi Kondor, and Gábor Csányi. On representing chemical environments. *Physical Review B*, 87(18):184115, May 2013. doi: 10.1103/PhysRevB.87.184115.
- [33] Sandip De, Albert P. Bartók, Gábor Csányi, and Michele Ceriotti. Comparing molecules and solids across structural and alchemical space. *Physical Chemistry Chemical Physics*, 18(20):13754–13769, 2016.
- [34] Jakub Kopko, David Graber, Saltuk Mustafa Eyrilmez, Stanislav Mazurenko, David Bednar, Jiri Sedlar, and Josef Sivic. Generalization beyond benchmarks: Evaluating learnable protein-ligand scoring functions on unseen targets. *arXiv preprint arXiv:2512.05386*, 2025.
- [35] Rishal Aggarwal, Akash Gupta, and U. Deva Priyakumar. APObind: A dataset of ligand unbound protein conformations for machine learning applications in de novo drug design. *arXiv preprint arXiv:2108.09926*, 2021. ICML 2021 Workshop on Computational Biology.

- [36] Durward W. J. Cruickshank. Remarks about protein structure precision. *Acta Crystallographica Section D: Biological Crystallography*, 55(3):583–601, 1999. doi: 10.1107/S0907444998012645.
- [37] Antonija Kuzmanic and Bojan Zagrovic. Determination of ensemble-average pairwise root mean-square deviation from experimental b-factors. *Biophysical Journal*, 98(5):861–871, 2010.
- [38] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Židek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A. A. Kohl, Andrew J. Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislaw Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Michalina Pacholska, Tamas Berghammer, Sebastian Bodenstein, David Silver, Oriol Vinyals, Andrew W. Senior, Koray Kavukcuoglu, Pushmeet Kohli, and Demis Hassabis. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873):583–589, 2021.
- [39] Michael J. Hartshorn, Marcel L. Verdonk, Gianni Chessari, Suzanne C. Brewerton, Wijnand T. M. Mooij, Paul N. Mortenson, and Christopher W. Murray. Diverse, high-quality test set for the validation of protein–ligand docking performance. *Journal of Medicinal Chemistry*, 50(4):726–741, 2007. doi: 10.1021/jm061277y.
- [40] Dejun Jiang, Chang-Yu Hsieh, Zhenxing Wu, Yu Kang, Jike Wang, Ercheng Wang, Ben Liao, Chao Shen, Lei Xu, Jian Wu, Dongsheng Cao, and Tingjun Hou. InteractionGraphNet: A Novel and Efficient Deep Graph Representation Learning Framework for Accurate Protein–Ligand Interaction Predictions. *Journal of Medicinal Chemistry*, 64(24):18209–18232, 2021. doi: 10.1021/acs.jmedchem.1c01830.
- [41] Yujia Li, Chenjie Gu, Thomas Dullien, Oriol Vinyals, and Pushmeet Kohli. Graph Matching Networks for Learning the Similarity of Graph Structured Objects. In *International Conference on Machine Learning*, volume 97, pages 3835–3845, 2019.
- [42] Ephraim Katchalski-Katzir, Isaac Shariv, Miriam Eisenstein, Asher A. Friesem, Claude Aflalo, and Ilya A. Vakser. Molecular surface recognition: determination of geometric fit between proteins and their ligands by correlation techniques. *Proceedings of the National Academy of Sciences*, 89(6):2195–2199, 1992. doi: 10.1073/pnas.89.6.2195.
- [43] David W. Ritchie and Graham J. L. Kemp. Protein docking using spherical polar Fourier correlations. *Proteins: Structure, Function, and Genetics*, 39(2):178–194, 2000.
- [44] Emilie Neveu, David W. Ritchie, Petr Popov, and Sergei Grudinin. PEPSI-Dock: a detailed data-driven protein–protein interaction potential accelerated by polar Fourier correlation. *Bioinformatics*, 32(17):i693–i701, 2016.
- [45] Tsung-Yu Lin, Aruni RoyChowdhury, and Subhransu Maji. Bilinear CNN models for fine-grained visual recognition. In *IEEE Int. Conf. Comput. Vis.*, pages 1449–1457, 2015.
- [46] Akira Fukui, Dong Huk Park, Daylen Yang, Anna Rohrbach, Trevor Darrell, and Marcus Rohrbach. Multimodal Compact Bilinear Pooling for Visual Question Answering and Visual Grounding. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2016.
- [47] Brandon Anderson, Truong Son Hy, and Risi Kondor. Cormorant: Covariant Molecular Neural Networks. In *Advances in Neural Information Processing Systems*, 2019.

A Noise robustness

Each model is evaluated on CASF-2016 with every atom of the complex, in both the bound and the unbound copy, displaced by isotropic Gaussian noise of width $\sigma \in [0, 2]$ Å, and ligand–pocket edges are rebuilt at the 5 Å cutoff after perturbation. For each model and σ we take the median RMSE over 20–40 noise draws and report the mean across 5 seeds (Figure 1). Figure 2 shows every atom paired readout.

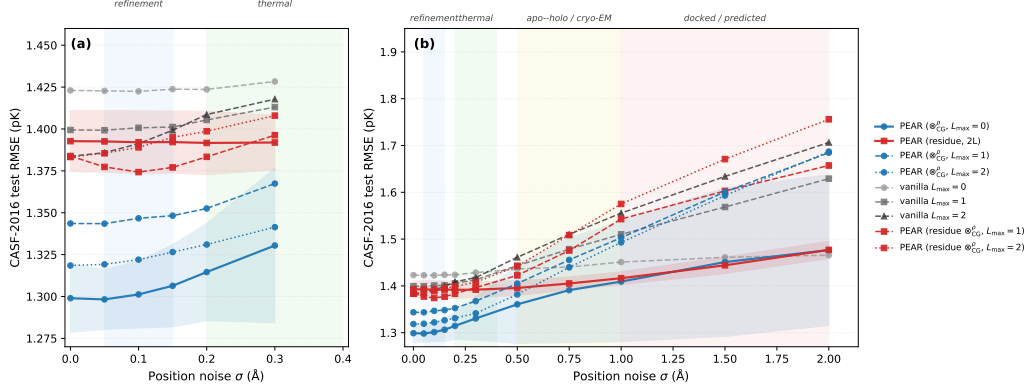


Figure 1: CASF-2016 test RMSE as every atom of the complex is perturbed by isotropic Gaussian noise of width σ , for PEAR and PEAR_{residue} ($\otimes_{CG}^{\rho}$ at $\ell_{max}=0, 1, 2$) and the no-paired-readout backbone (“vanilla”) at $\ell_{max}=0, 1, 2$. Panel (a) is an enlargement of $\sigma \leq 0.3$ Å. The slope is steeper at higher ℓ_{max} at both scales, and at $\ell_{max}=0$ the residue model is flat up to 0.5 Å. The range of σ includes X-ray refinement error (0.1 – 0.3 Å [36]) and thermal motion of ordered atoms (0.2 – 0.5 Å [37]), apo-holo backbone displacement (0.5 – 1.0 Å) and docked or predicted structures (1 – 2 Å [38, 39]).

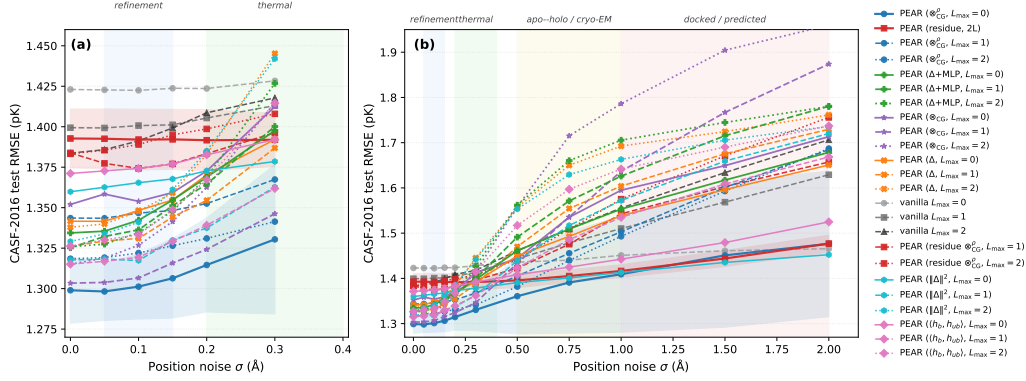


Figure 2: CASF-2016 test RMSE under the same perturbation for every atom paired readout at $\ell_{max}=0, 1, 2$, together with the residue $\otimes_{CG}^{\rho}$ models and the backbone-only (“vanilla”) models. Each colour corresponds to a readout and each line style to a backbone ℓ_{max} (solid 0, dashed 1, dotted 2). Panel (a) is an enlargement of $\sigma \leq 0.3$ Å.

B Equivariant message passing

In an $SO(3)$ -equivariant message-passing network, the output $h_i^{(t)}$ of node i in layer t decomposes by angular order as $h_i^{(t)} = \bigoplus_{\ell=0}^{\ell_{max}} h_i^{(t,\ell)}$. The message from node j to node i is the Clebsch–Gordan product of atom j ’s features with the spherical-harmonic embedding of the relative orientation of its neighbours,

$$m_{ij}^{(t)} = \bigoplus_{\ell_{out}} \sum_{\ell_1, \ell_2} W_{\ell_1 \ell_2 \ell_{out}}^{(t)}(r_{ij}) \left(h_j^{(t, \ell_1)} \otimes Y_{\ell_2}(\hat{\mathbf{r}}_{ij}) \right)^{(\ell_{out})}, \quad (3)$$

where the weights $W^{(t)}$ are produced by a radial MLP of the interatomic distance. Messages are aggregated and used to update the node state,

$$h_i^{(t+1)} = U \left(h_i^{(t)}, \sum_{j \in \mathcal{N}(i)} m_{ij}^{(t)} \right). \quad (4)$$

When $\ell_{\max} = 0$ the scheme reduces to a scalar MPNN; the canonical PEAR uses this scalar backbone, with higher- ℓ terms produced only by the atom expansion of Eq. 2 and the pocket expansion F .

C Noise sensitivity of higher- ℓ features

A higher- ℓ feature depends on the atomic positions through the bond directions. For a displacement $\delta_{\perp}$ perpendicular to a bond of length r , the bond direction $\hat{r}$ changes by an angle $\delta\theta \approx \delta_{\perp}/r$. The feature then changes by $|\delta Y_{\ell}^m| \approx |\nabla_{S^2} Y_{\ell}^m| \delta\theta$. For a degree- ℓ spherical harmonic this gradient has root-mean-square (RMS) value $\sqrt{\ell(\ell+1)}$ times the RMS of the harmonic,

$$\int_{S^2} |\nabla_{S^2} Y_{\ell}^m|^2 d\Omega = \ell(\ell+1) \int_{S^2} |Y_{\ell}^m|^2 d\Omega. \quad (5)$$

The relative change in a degree- ℓ feature is therefore $\sqrt{\ell(\ell+1)} \delta\theta$. The $\ell = 0$ features depend only on interatomic distances and atom types and are therefore unchanged by rotation of a bond direction. Each message-passing layer adds a tensor product of the node features with the spherical harmonics of the bond directions through a residual connection. In a backbone with higher- ℓ features, this sensitivity accumulates over the layers. Figure 1 reports test RMSE as a function of σ .

D Implementation details

PEAR and PEAR_{residue} use the same backbone and paired readouts (Section 2), but PEAR has one pocket node per atom and PEAR_{residue} one pocket node per residue centroid. They differ in depth and width, in the residue model’s amino-acid embedding and B-factor node features, and in how the pocket graph is built. At atom scale we use GIGN’s graph construction and node features [28], where the pocket is the protein heavy atoms within 5 Å of a ligand heavy atom, each ligand atom is joined to every pocket atom inside that radius, and pocket atoms are joined only to their covalently bonded neighbours. Because the residue graphs are much smaller, the cutoff range between centroids is 10 Å while keeping the radial basis cutoff at 5 Å for a comparable radial resolution. The residue model represents each residue by its mean-pooled atom features together with a learned amino-acid embedding and a z-scored B-factor. Table 3 lists the architecture and optimization settings and Table 4 the input features. All benchmark numbers are 5-fold cross-validation ensembles, formed by averaging the predictions of the five fold models. Figure 3 shows the full pipeline.

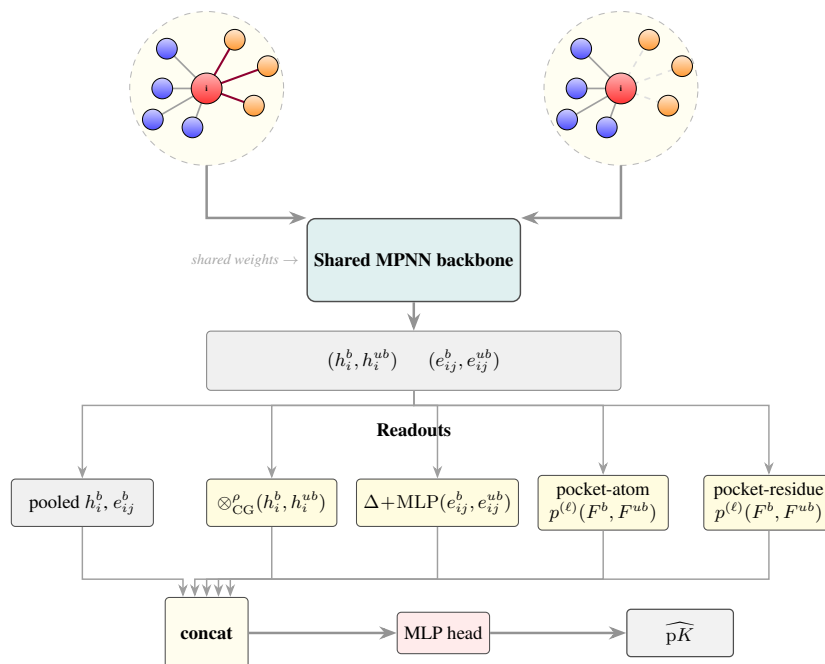


Figure 3: PEAR pipeline. The top row contains two copies of the protein–ligand complex. The *bound* graph has intra- and intermolecular edges (purple), and the *unbound* graph has intramolecular edges only. In the middle row, both copies are processed by a shared message-passing backbone ($\ell_{\max} = 0$), which outputs per-atom features (h_i^b, h_i^{ub}) and intermolecular edge states (e_{ij}^b, e_{ij}^{ub}) . In the bottom row, the inputs to the head are the pooled bound atom and edge features and four paired comparisons, per-atom ($\otimes_{\text{CG}}^\rho$), per-intermolecular-edge ($\Delta + \text{MLP}$), and two pocket-level descriptors (atom and residue scales) compared via the matched- ℓ cross-power $p_{cc'}^{(\ell)} = \sum_m F_{(\ell, c, m)}^b F_{(\ell, c', m)}^{ub}$. All readouts are concatenated and passed to an MLP head that outputs the scalar binding-affinity prediction.

Table 3: Architecture and optimization settings for the atom and residue models.

	Atom	Residue
<i>Architecture</i>		
Message-passing layers	5	2
Scalar channels	384	320
Pooled projection	192	160
Backbone $\ell_{\max}$	0	0
Node paired readout		$\otimes_{\text{CG}}^p$ (re-embed $\ell=1$)
Edge paired readout		Δ +MLP
Pocket descriptor		global SH, $\ell_{\max}=2$, 16 ch
Radial filter		node-conditioned MLP (64×2)
Interaction cutoff		5 Å
Pocket-pocket edges	covalent bonds	centroids within 10 Å
Node features	35-dim atom	35-dim (residue mean) + 8-d AA + B-factor
Parameters	1.68M	831K
<i>Optimization</i>		
Optimizer		AdamW
Loss		MSE (pK)
Batch size		64
Peak / min LR	$2 \times 10^{-4} / 10^{-5}$	$10^{-4} / 10^{-5}$
Warmup		5 epochs (linear)
Weight decay		10^{-4}
Gradient-norm clip		10.0
Epochs	1200	800 (1200 for CleanSplit)
Checkpoint		lowest validation RMSE

Table 4: Input features. Atom nodes have a 35-dim descriptor; the residue node features are the atom features mean-pooled per residue, plus a B-factor and an amino-acid embedding.

Feature	Encoding	Dims
<i>Atom node</i>		
Element	C, N, O, S, F, P, Cl, Br, I, other	10
Degree	0–6	7
Implicit valence	0–6	7
Hybridization	SP, SP ² , SP ³ , SP ³ d, SP ³ d ²	5
Aromatic	boolean	1
Hydrogen count	0–4	5
<i>Edge</i>		
Distance	smooth-finite RBF (0–5 Å)	14
Edge type	pocket-pocket / ligand-ligand / ligand-pocket, learned	16
<i>Residue additions</i>		
B-factor	z-scored per structure	1
Amino acid	learned embedding, 22 classes	8

PDBbind CleanSplit. The source data are the general and refined sets of PDBbind v2020, which together contain 19,443 complexes, with the CASF-2013 and CASF-2016 complexes excluded [1]. Preprocessing removes 450 complexes with inexact affinity labels or structures that cannot be parsed, leaving 18,623. Affinity labels are $-\log_{10}$ of K_i , K_d or IC_{50} , treated as interchangeable. Similarity between two complexes is measured by the TM-score of the proteins, the Tanimoto similarity T of the ligands, and the RMSD between the ligands after the proteins are superposed by TM-align.

The first filtering stage removes 711 training complexes. A training complex is removed if, for some CASF-2013 or CASF-2016 complex, the TM-score exceeds 0.8, $T + (1 - \text{RMSD})$ exceeds 0.8, and the affinities differ by at most 1 pK unit. A training complex is also removed if its ligand has $T > 0.9$

to a test ligand and the affinities differ by at most 1 pK unit. Pairs that are structurally similar but differ in affinity by more than 1 pK unit are kept, so that activity cliffs remain in the training data. The second stage removes redundancy within the training set. Two training complexes are connected if the TM-score exceeds 0.8, $T + (1 - \text{RMSD})$ exceeds 1.3, and the affinities differ by less than 0.5 pK units. The complex with the most connections is removed repeatedly until all connections are removed, which removes 1,451 complexes. Ties are broken in favour of removing general-set complexes and then lower-resolution structures. The independent subset of CASF-2016 consists of the 144 of its 285 complexes that fall below the first-stage thresholds against every complex of unfiltered PDBbind.

Held-out pocket clusters. The held-out pocket clusters are constructed as described by Kopko et al. [34]. Complexes of PDBbind v2020 are clustered with PLINDER (release 2024-06/v2) by pocket IDDT community at a threshold of 50. For each of the seven CASF-2016 targets (1INVQ, 1SQA, 2P15, 2VW5, 3DD0, 3F3E and 3O9I), every complex in the target’s community is assigned to the test set. The remaining complexes are filtered with CleanSplit and used for training and validation. Communities are defined on pocket geometry, so a protein can appear in the test set through one binding site and in the training set through another. Community detection also does not guarantee that every training pocket falls below the threshold against every test pocket. Kopko et al. [34] note that clusters usually preserve the protein fold, and that the procedure can in principle group similar pockets from different folds.

For few-shot adaptation, the 25 finetuning complexes of each target are the first 25 after shuffling its cluster with a fixed seed, and the remaining cluster members form the test set. We finetune all parameters with AdamW at a learning rate of 10^{-4} for 25 epochs and keep the epoch with the lowest RMSE on the 25 finetuning complexes. Each of the five cross-validation models is finetuned separately, and the five are ensembled by averaging their predictions for each test complex before computing Pearson r , as in Kopko et al. [34].

Code is available at <https://github.com/wonggrp/general-binding-mpnn>.

E Two-state models and paired readout operations

In the literature, input modalities to a neural network scoring function are usually either the protein sequence and a 2D graph of the ligand, or a 3D structure of the complex. Amongst those that use 3D structure, models that treat the complex as having two states either do so implicitly, by updating a single node state first with intramolecular and then with intermolecular edges [17, 29, 40], or explicitly maintain parallel representations of the bound and unbound states, in some cases through weight-shared networks [21–24], so that each atom in the complex has a pair of embeddings h_i^b and h_i^{ub} . We find that existing models taking the latter route perform a narrow range of downstream actions on these pairs. Namely, they first aggregate the embeddings within their respective graphs, then combine them by a difference or an MLP, usually on the scalar features. In adjacent fields, pairs of embeddings have been combined with more care, including in equivariant settings, which introduces additional computational considerations.

Node-wise comparisons. Comparing two structures node by node has been done in both graph similarity learning and kernel methods. Li et al. [41] compare the two graphs node by node at every message passing layer, using attention to match each node against the nodes of the other graph. They report that this outperforms embedding each graph separately and concatenating the two graph vectors. De et al. [33] compare two molecules by first pairing their atomic environments by some measure of best match, then summing the resulting similarities. They note that averaging over all pairs is equivalent to performing an aggregation step first, which loses resolving power in the sense that two structures composed of different environments can be identical under the kernel.

In both cases the correspondence between the nodes of the two graphs is unknown, so these methods evaluate a similarity between every pair of nodes before selecting or weighting among them. Contrastingly, for two states of the *same* complex the correspondence is known in advance, namely atom i in the bound state has a corresponding atom i in the unbound state, so each node and each edge can be combined with its counterpart directly. The correspondence also easily allows the two states to be matched and combined over a local region, which may be of interest since atoms tend to act in groups during binding [18].

Product-based comparisons. Comparing two structures by a product has a long history in structure-based methods. Several docking methods expand each molecule’s surface in spherical harmonics and search for the relative orientation that maximizes the product of the two expansions, which measures how well the shapes fit together [42–44]. The SOAP kernel [32, 33] applies a Clebsch–Gordan product within each atomic environment to obtain an invariant descriptor, then compares two environments by a dot product. More recently, in vision and multimodal learning, the outer product of two streams is argued to be more expressive than concatenation [45, 46]. These are seldom used in neural network scoring functions. In Table 5 we summarise these operations and suggest an analogous set of modules for combining bound and unbound embeddings. The operations there are Clebsch–Gordan products, differences, or concatenations, and differ in whether the combination is made at the node or graph level.

Combining equivariant features. Usually the combination step coincides with the last step of the pipeline, which is forced to be invariant throughout. If instead the outputs of the combination step are additional embeddings for the MLP, the combination can take place before the final concatenation, and its inputs can be equivariant. The encoder’s higher- ℓ features are then available to be combined, and the Clebsch–Gordan product becomes available as a nonlinearity for coupling two node features. In Anderson et al. [47] this appears in the internal layers as a one-body interaction, coupling an atom’s activation with itself. The Clebsch–Gordan product between two equivariant features returns output at every order L between $|\ell_1 - \ell_2|$ and $\ell_1 + \ell_2$, while the prediction requires that all variables be of type $L = 0$. In a terminal or *readout* module, we therefore contract the output of the product against a third factor to keep the paths $(\ell_1, \ell_2) \rightarrow L$ with $L \leq \ell_{\max}$. This way of extracting invariants is described in [32] as the *bispectrum*, whereas the direct reduction is the *power spectrum*.

Table 5: Operations that combine two representations. Below the rule we suggest candidate operations that combine bound and unbound counterparts, with the symbols used in Table 6 for the ones measured there. Rows sharing an operation differ only in how the output is reduced to scalars, where $[\cdot]_0$ denotes the $L = 0$ part of a product. An arrow ($\rightarrow$) means that the output is equivariant for downstream purposes, and — that the output is already scalar. The symbol c_ℓ is the order- ℓ coefficient of a fixed expansion, h_i a learned node embedding, $\mathcal{A}(\cdot)$ an aggregation over a graph, and $\tilde{h}_i$ the density of Eq. 2.

name	example source	level	ℓ from	operation	reduction step
shape correlation	Hex (docking program)	graph	fixed basis	$c_\ell(X) \otimes c_\ell(Y)$	$\rightarrow$
environment kernel	SOAP	node	fixed basis	$\langle [c_\ell(i)]^{\otimes 2}, [c_\ell(j)]^{\otimes 2} \rangle_0$	—
bispectrum	SOAP ($n=3$)	node	fixed basis	$c_\ell(i) \otimes c_\ell(i)$	$[[\cdot]_L \otimes c_\ell(i)]_0$
best-match sum	REMatch	node	fixed basis	$\max_\pi \sum_i \langle p(i), p(\pi_i) \rangle$	—
one-body interaction	Cormorant	node	backbone	$h_i \otimes h_i$	$\rightarrow$
dot product interaction	Cormorant	node	backbone	$h_i \otimes h_j$	$[\cdot]_0$
self-correlation	MACE	node	lift	$\tilde{h}_i \otimes \tilde{h}_i$	$\rightarrow$
endpoint subtraction	ACNN, PAMNet	graph	—	$\mathcal{A}(h^b) - \mathcal{A}(h^{ub})$	—
endpoint concatenation	HM-Holo-Apo	graph	—	$f_\theta([\mathcal{A}(h^b); \mathcal{A}(h^{ub})])$	—
cross-graph matching	GMN	node	—	$h_i^X - \sum_j a_{j \rightarrow i} h_j^Y$	—
symbol					
difference	Δ	node	backbone	$h_i^b - h_i^{ub}$	$[[\cdot]_L \otimes h_i^{ub}]_0$
symmetric difference	Δ_{sym}	node	backbone	$h_i^b - h_i^{ub}$	$[[\cdot]_L \otimes (h_i^b + h_i^{ub})]_0$
difference norm	$\ \Delta\ ^2$	node	backbone	$h_i^b - h_i^{ub}$	$[[\cdot]_L \otimes (h_i^b - h_i^{ub})]_0$
MLP	MLP	node	—	$f_\theta([h_i^b; h_i^{ub}])$	—
dot product	$\langle h^b, h^{ub} \rangle$	node	backbone	$h_i^b \otimes h_i^{ub}$	$[\cdot]_0$
Clebsch–Gordan product	$\otimes_{\text{CG}}$	node	backbone	$h_i^b \otimes h_i^{ub}$	$[[\cdot]_L \otimes h_i^{ub}]_0$
on densities	$\otimes_{\text{CG}}^\rho$	node	lift	$\tilde{h}_i^b \otimes \tilde{h}_i^{ub}$	$[[\cdot]_L \otimes \tilde{h}_i^{ub}]_0$
on pooled densities		graph	lift	$\mathcal{A}(\tilde{h}^b) \otimes \mathcal{A}(\tilde{h}^{ub})$	$[\cdot]_0$

An example of comparing two atom-centered descriptions is the SOAP kernel [32, 33] $\int d\hat{R} |\int \rho_A(\mathbf{r}) \rho_B(\hat{R}\mathbf{r}) d^3r|^2$. Expanded in spherical harmonics, the kernel can be written as a Clebsch–Gordan (CG) $(\ell, \ell) \rightarrow 0$ contraction of the two expansion coefficients,

$$p_\ell = (c^{(\ell)}(\rho_A) \otimes c^{(\ell)}(\rho_B))^{(0)} \propto \sum_m c_{\ell m}(\rho_A) c_{\ell m}(\rho_B). \quad (6)$$

Joint representations can also be formed by linear operations such as the difference $h_i^{b,(\ell)} - h_i^{ub,(\ell)}$ or in general $A^{(\ell)}h_i^{b,(\ell)} + B^{(\ell)}h_i^{ub,(\ell)}$ for learnable matrices $A^{(\ell)}, B^{(\ell)}$, or for scalar features by an MLP on the concatenated scalars. Constructing a paired readout involves two independent choices, namely how a joint higher- ℓ feature is constructed from the pair and how that feature is reduced to a scalar for the non-equivariant MLP head. We denote the individual operations as Δ ($h_b - h_{ub}$), $\Delta + \text{MLP}$, $\otimes_{\text{CG}}$ (a CG product $h_b \otimes_{\text{CG}} h_{ub}$), $\otimes_{\text{CG}}^\rho$ (a CG product on a density computed via Eq. 2), and $\langle h_{ub}, h_b \rangle$.

Each intermolecular edge has a hidden state e_{ij} , initialized from its two endpoints by a symmetric encoder and updated residually after every layer from the new node states.

The pocket descriptor is an angular and radial expansion of the protein pocket centered at the ligand centroid,

$$F^{(c,\ell,m)} = \sum_{j \in V_p} (W' h_j^{(T,0)})_c R_\ell(r_j) Y_\ell^m(\hat{\mathbf{r}}_j), \quad (7)$$

with an independent linear projection W' . A residue-scale expansion is constructed by first mean-pooling the atoms in each residue.

In Eq. 1, the edge readout ψ concatenates an MLP of the difference $e_{ij}^b - e_{ij}^{ub}$ with an MLP of the pair, and the pocket term is the cross power spectrum $\sum_m F_{\ell m}^b F_{\ell m}^{ub}$ at each order ℓ , computed both over pocket atoms and over residue centroids.

In full, the atom readout couples the bound and unbound expansions of Eq. 2 channel by channel and contracts each output order against the unbound expansion,

$$\otimes_{\text{CG}}^\rho : [\tilde{h}_i^b \otimes_{\text{CG}} \tilde{h}_i^{ub}]_L \otimes \tilde{h}_i^{ub}]_0, \quad (8)$$

where $[\cdot]_L$ denotes the order- L part of a product. The Clebsch–Gordan product is taken in uuu mode, in which output channel c couples channel c of both inputs, with one learnable weight per path $(\ell_1, \ell_2) \rightarrow L$.

The $(\ell, \ell) \rightarrow 0$ path of Eq. 8 is the overlap $\sum_m \tilde{h}_{b,m}^{(c,\ell)} \tilde{h}_{ub,m}^{(c,\ell)}$, which is Eq. 6 with the bound and unbound environments of the same atom in place of ρ_A and ρ_B . Every output order L of Eq. 8 is reduced to a scalar by an inner product against a reference equivariant feature (in our implementation, $\tilde{h}_{ub}$), with the $L = 0$ term reducing to the product of the two scalars,

$$s^{(c,L)} = \sum_M [\tilde{h}_b \otimes_{\text{CG}} \tilde{h}_{ub}]_{LM}^{(c)} \tilde{h}_{ub,M}^{(c,L)}. \quad (9)$$

Since every output order, including the overlap, is contracted against $\tilde{h}_{ub}$, each scalar passed to the head is a product of three activations. Contracting against $h_b + h_{ub}$ instead (Δ_{sym}) treats the two copies symmetrically, and we otherwise use the asymmetric contraction throughout. Figure 4 details the four readouts of the main text.

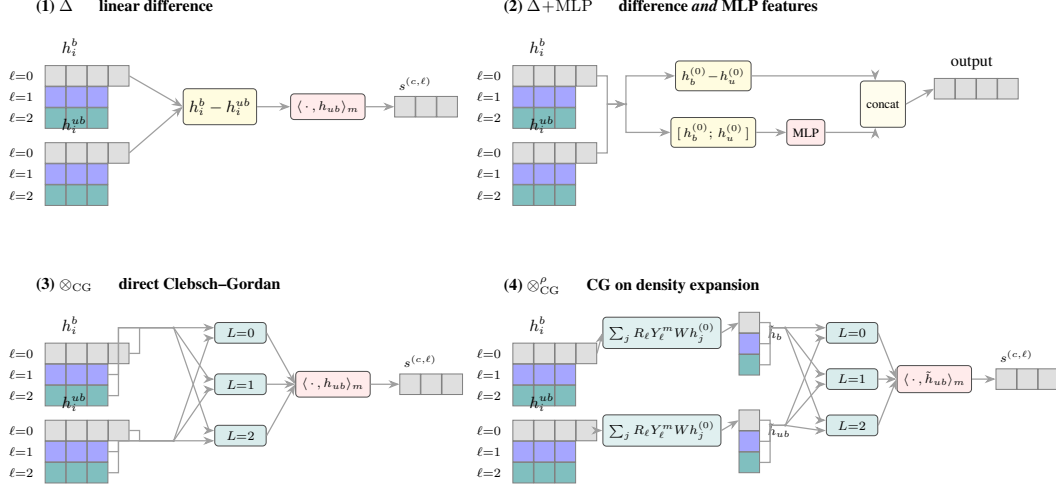


Figure 4: Each paired readout is a joint embedding of an atom’s bound and unbound environments, (h_i^b, h_i^{ub}) . **(1)** In Δ , the difference $h_i^b - h_i^{ub}$ is reduced to scalars by inner product against h_{ub} at every ℓ . **(2)** In $\Delta + \text{MLP}$, the scalar ($\ell=0$) channels of h_i^b and h_i^{ub} form two readouts, a difference and an MLP on their concatenation, which are concatenated. **(3)** $\otimes_{\text{CG}}$ takes channel-diagonal (uuu) Clebsch–Gordan products on the equivariant features, with all valid (ℓ_1, ℓ_2) pairs feeding each order- L output. Every order- L output is reduced to a scalar by inner product against h_{ub} . **(4)** In $\otimes_{\text{CG}}^\rho$, the scalars are first expanded into a local spherical-harmonic density $\tilde{h}_i^{(\ell)}$ (Eq. 2) and then coupled in the same way. $\otimes_{\text{CG}}^\rho$ is the only one of the four readouts that produces higher- ℓ paired features from a scalar backbone.

F Paired-readout ablation

Using any paired readout improves on the no-paired-readout MPNN, while the specific operation and ℓ_{max} matter much less (Table 6); $\otimes_{\text{CG}}^\rho$ is the most accurate at $\ell_{\text{max}}=0$. Models are parameter-matched across ℓ_{max} ($\sim 1.7\text{M}$ for PEAR, $\sim 1.35\text{M}$ without them), with the channel multiplicity reduced as ℓ_{max} grows.

Table 6: Atom-scale readout comparison on CleanSplit. Test RMSE, mean $\pm$ std over five seeds.

Atom paired readout	$\ell_{\text{max}}=0$	$\ell_{\text{max}}=1$	$\ell_{\text{max}}=2$
PEAR _{backbone}	1.386 ± 0.023	1.391 ± 0.013	1.386 ± 0.019
<i>Difference-based (linear in $h_b - h_{ub}$):</i>			
Δ	1.326 ± 0.020	1.313 ± 0.032	1.297 ± 0.012
Δ_{sym}	1.342 ± 0.037	1.326 ± 0.018	1.338 ± 0.014
$\Delta + \text{MLP}$	1.366 ± 0.022	1.302 ± 0.020	1.316 ± 0.019
$\Delta_{\text{sym}} + \text{MLP}$	1.334 ± 0.019	1.325 ± 0.018	1.326 ± 0.023
<i>Product-based (bilinear in h_b, h_{ub}):</i>			
$\ \Delta\ ^2$	1.360 ± 0.034	1.318 ± 0.020	1.329 ± 0.010
$\langle h_b, h_{ub} \rangle$	1.371 ± 0.025	1.324 ± 0.036	1.326 ± 0.012
$\otimes_{\text{CG}}$	1.352 ± 0.012	1.303 ± 0.014	1.315 ± 0.011
$\otimes_{\text{CG}}^\rho$	1.293 ± 0.015	1.344 ± 0.021	1.316 ± 0.012

G Component ablation

We ablate PEAR by removing each component in turn and rank by the impact on CleanSplit test RMSE (Table 7). Removing all paired comparisons increases the RMSE more than removing any single component. Removing the atom or the edge paired readout ($+0.066$ each) costs more than removing both together ($+0.047$), a difference within one standard deviation.

Table 7: Impact of removing message-passing components and readouts, ranked by mean CleanSplit RMSE over five seeds. In “Remove all comparisons” the unbound copy is pooled, and “1-copy” is the model without the unbound graph and descriptors.

Configuration	Params	Mean RMSE	Median	Δ
PEAR	1.68M	1.293 ± 0.015	1.294	—
Remove feedback	1.66M	1.323 ± 0.014	1.320	+0.030
Remove residue global comparison	1.61M	1.326 ± 0.033	1.324	+0.033
Remove local comparisons (atom + edge)	1.44M	1.340 ± 0.015	1.339	+0.047
Remove atom global comparison	1.61M	1.343 ± 0.037	1.338	+0.050
Remove edge updates	1.68M	1.344 ± 0.039	1.335	+0.051
Remove global comparisons (atom + residue)	1.54M	1.354 ± 0.026	1.346	+0.061
Remove atom comparison	1.63M	1.359 ± 0.037	1.337	+0.066
Remove edge comparison	1.49M	1.359 ± 0.051	1.340	+0.066
Remove all comparisons	1.35M	1.386 ± 0.023	1.374	+0.093
1-copy	1.42M	1.389 ± 0.016	1.394	+0.096

H Scale complementarity and joint ensembles

The atom and residue models are accurate on systematically different complexes, so ensembling one of each lowers error more than ensembling two comparable atom models (Table 8). Over all 285 CASF-2016 complexes, the signed residuals of the two models correlate at $r = 0.80$. Pairing PEAR with an atom-scale model of equal standalone RMSE raises the residual correlation to 0.93 and the ensemble RMSE from 1.243 to 1.270. Table 9 stratifies the test set along six axes. The ensemble is best in 20 of 24 quartile bins, and the atom and residue models each win some of the other four.

(a) Ensembles on CASF-2016.

Model	# models	RMSE	r
PEAR	5	1.258	0.834
PEAR _{residue}	5	1.282	0.822
Mixed (3 atom + 2 residue, mean over combinations)	5	1.240	0.840
Combined (atom 5-fold + residue 5-fold)	10	1.233	0.847
atom + residue (1-fold)	2	1.243	0.839
atom + atom, no residue pocket (1-fold)	2	1.270	0.825
Ideal, atom + residue	—	1.030	0.896
Ideal, atom + atom	—	1.131	0.873

Table 8: Joint ensembles of PEAR and PEAR_{residue} variants lower RMSE further because the two are accurate on systematically different complexes. “atom + atom” pairs the atom model with a PEAR ablation of equal standalone RMSE; “Ideal” is the per-sample-closest prediction among the constituent models.

We bin CASF-2016 into quartiles along six per-complex features, namely ligand heavy-atom count, pocket heavy-atom count, T1 interface size, inter-edge count, ligand/pocket atom ratio, and apo–holo backbone C α RMSD from ApoBind [35]. Quartile counts are approximately balanced ($n \approx 71$ –76 per bin; the apo–holo axis covers the 171 CASF complexes for which an apo structure is available). For each bin we report the test RMSE of single-fold PEAR and PEAR_{residue} models and their unweighted mean ensemble (Table 9).

Feature	Bin	atom	residue	ensemble	winner
Ligand heavy atoms	Q1: 9–17	1.292	1.447	1.306	atom
	Q2: 17–23	1.178	1.290	1.170	ens
	Q3: 23–30	1.172	1.133	1.062	ens
	Q4: 30–67	1.461	1.395	1.370	ens
Pocket heavy atoms	Q1: 50–124	1.135	1.155	1.074	ens
	Q2: 124–150	1.111	1.210	1.077	ens
	Q3: 150–175	1.502	1.480	1.424	ens
	Q4: 175–280	1.415	1.438	1.376	ens
T1 interface atoms	Q1: 0–58	1.109	1.164	1.062	ens
	Q2: 58–71	1.235	1.283	1.167	ens
	Q3: 71–85	1.367	1.407	1.335	ens
	Q4: 85–139	1.421	1.379	1.349	ens
Inter edges	Q1: 0–195	1.192	1.332	1.186	ens
	Q2: 195–257	1.204	1.224	1.140	ens
	Q3: 257–321	1.294	1.284	1.215	ens
	Q4: 321–642	1.466	1.438	1.405	ens
Lig/pkt atom ratio	Q1: .064–.124	1.438	1.640	1.489	atom
	Q2: .124–.155	1.098	1.129	1.058	ens
	Q3: .155–.196	1.372	1.311	1.236	ens
	Q4: .196–.472	1.249	1.168	1.160	ens
Apo–holo RMSD (Å)	Q1: 0.13–0.37	1.054	1.330	1.107	atom
	Q2: 0.37–0.48	1.224	1.280	1.170	ens
	Q3: 0.48–0.67	1.425	1.250	1.287	residue
	Q4: 0.67–1.13	1.393	1.427	1.367	ens

Table 9: The ensemble has the lowest RMSE in 20 of 24 quartile bins. Of the other four, the atom model is lowest in three (ligand Q1, lig/pkt Q1 and apo–holo Q1, all within 0.053 RMSE) and the residue model in apo–holo Q3. Test RMSE by quartile for the single-fold PEAR and PEAR_{residue} variants and their unweighted mean ensemble.

I Preliminary experiment on a basic MPNN

Table 10 compares one-state and two-state message passing on CleanSplit across network depth and $\ell_{\max}$. The encoder is a basic equivariant MPNN, which is not the backbone of PEAR. This encoder has the message passing of Appendix B but without edge states, and therefore without edge updates or edge feedback. The encoder is applied to the complex graph and to the same graph with intermolecular edges removed, and the pooled embeddings are combined by an MLP. At two and five layers and at $\ell_{\max}$ of 0, 1 and 2, the two-state model improves on the one-state model in only two of the six settings, in both cases by less than one standard deviation.

Table 10: One-state and two-state message passing on CleanSplit across network depth and $\ell_{\max}$. Test RMSE in pK units, mean $\pm$ std. All rows are over five seeds.

Layers	Copies	$\ell_{\max}$	RMSE
2	1	0	1.441 ± 0.007
2	1	1	1.453 ± 0.016
2	1	2	1.405 ± 0.023
2	2	0	1.460 ± 0.012
2	2	1	1.443 ± 0.035
2	2	2	1.489 ± 0.021
5	1	0	1.386 ± 0.034
5	1	1	1.389 ± 0.057
5	1	2	1.392 ± 0.038
5	2	0	1.423 ± 0.038
5	2	1	1.399 ± 0.027
5	2	2	1.384 ± 0.024

J ATOM3D LBA

The models in the literature evaluated on the LBA task are almost without exception atom-based. We therefore report the performance of a condensed set of models, namely PEAR, PEAR_{backbone}, and PEAR_{LBA}, a variant of PEAR developed specifically for this task. In PEAR_{LBA}, four settings are adjusted, while the input features, backbone, and general structure of the model remain the same (see below). On LBA 30%, PEAR_{LBA} achieves the lowest RMSE and highest Pearson correlation coefficient as well as the second highest Spearman correlation coefficient out of the models considered (Table 11). Its differences from CheapNet are within one standard deviation on all three metrics, so we do not regard this result as a decisive improvement over the state of the art. In addition, PEAR achieves the highest Spearman correlation coefficient on LBA 60%. Both models outperform PEAR_{backbone}, once again showcasing the utility of paired readouts for generalization tasks.

The source data are the refined set of PDBbind v2019, filtered by Townshend et al. [4] for a resolution below 2.5 Å, an R-factor below 0.25, and valid ligand bonding information. Labels are pK values. Sequence identity is computed with BLASTP. In the 30% split, training proteins share at most 30% sequence identity with validation and test proteins, and each sequence cluster makes up at most 20% of any partition. The 60% split applies the same procedure at a 60% threshold. We use the split files distributed with the benchmark, which contain 4,463 complexes in both splits. At 30% identity these are divided into 3,507 training, 466 validation and 490 test complexes, and at 60% into 3,563, 448 and 452. In each split two training complexes fail RDKit sanitization in both their mol2 and SDF files and are excluded, 1WC1 and 5DHU in the 60% split. This leaves 3,505 and 3,561 training complexes, and the validation and test sets are complete. We use the pocket files distributed with the benchmark, which contain the protein atoms within 6 Å of the ligand. Following Lim et al. [18], results are the mean of three runs and are reported as RMSE and Pearson and Spearman correlation.

PEAR_{LBA} differs from PEAR in four settings and is otherwise identical. (i) The atom density $\tilde{h}_i$ of Eq. 2 is expanded only to $\ell=0$, where PEAR expands the density to $\ell=1$. The $\ell=0$ expansion keeps the radial term of each neighbour and drops the directional $\ell=1$ terms. (ii) The Clebsch–Gordan product of the atom paired readout (Eq. 8) uses uvu mode, where PEAR uses uuu mode. In uuu mode, output channel c couples channel c of both inputs, with one weight per channel. In uvu mode, output channel u couples channel u of the first input with every channel v of the second, with one weight per pair (u, v) . (iii) The pocket paired readout over atoms is removed, and the pocket paired readout over residue centroids is kept at $\ell=2$. (iv) The 10-dimensional element one-hot is passed to the input projection directly, without a learned element embedding. The depth and width of the backbone and the edge paired readout are those of PEAR, and PEAR_{LBA} has 1.60M parameters, against 1.68M for PEAR. Table 11 reports the mean and standard deviation over three runs. UniMol and BindNet report means only.

Method	Params	LBA 30%			LBA 60%		
		RMSE ↓	Pearson ↑	Spearman ↑	RMSE ↓	Pearson ↑	Spearman ↑
Atom3D-GNN	0.38M	1.601 ± 0.048	0.545 ± 0.027	0.533 ± 0.033	1.408 ± 0.069	0.743 ± 0.022	0.743 ± 0.027
Atom3D-3DCNN	2.22M	1.416 ± 0.021	0.550 ± 0.021	0.553 ± 0.009	1.621 ± 0.025	0.608 ± 0.020	0.615 ± 0.028
ProNet	1.39M	1.463 ± 0.001	0.551 ± 0.005	0.551 ± 0.008	1.343 ± 0.025	0.765 ± 0.009	0.761 ± 0.003
HoloProt	1.44M	1.464 ± 0.006	0.509 ± 0.002	0.500 ± 0.005	1.365 ± 0.038	0.749 ± 0.014	0.742 ± 0.011
MaSIF	0.62M	1.484 ± 0.018	0.467 ± 0.020	0.455 ± 0.014	1.426 ± 0.017	0.709 ± 0.001	0.701 ± 0.001
SchNet	0.21M	1.370 ± 0.028	0.590 ± 0.017	0.571 ± 0.028	—	—	—
MACE	3.91M	1.372 ± 0.021	0.612 ± 0.010	0.592 ± 0.010	—	—	—
LEFTNet	0.85M	1.366 ± 0.016	0.592 ± 0.014	0.580 ± 0.011	—	—	—
GET	0.69M	1.327 ± 0.005	0.620 ± 0.004	0.611 ± 0.003	—	—	—
CheapNet	1.39M	<u>1.311</u> ± 0.003	<u>0.642</u> ± 0.001	0.639 ± 0.010	<u>1.238</u> ± 0.005	0.794 ± 0.002	<u>0.789</u> ± 0.001
<i>pretrained</i>							
Uni-Mol	47.6M	1.434	0.565	0.540	1.357	0.753	0.750
ProFSA	>47.6M	1.377 ± 0.010	0.628 ± 0.010	0.620 ± 0.010	1.377 ± 0.010	0.764 ± 0.000	0.762 ± 0.010
BindNet	>47.6M	1.340	0.632	0.620	1.230	<u>0.793</u>	0.788
PEAR _{backbone}	1.35M	1.386 ± 0.037	0.621 ± 0.021	0.613 ± 0.024	1.328 ± 0.012	0.773 ± 0.004	0.775 ± 0.003
PEAR	1.68M	1.340 ± 0.017	0.629 ± 0.011	0.615 ± 0.009	1.261 ± 0.022	0.791 ± 0.007	0.793 ± 0.006
PEAR _{LBA}	1.60M	1.308 ± 0.026	0.647 ± 0.017	<u>0.634</u> ± 0.023	1.291 ± 0.006	0.782 ± 0.001	0.782 ± 0.002

Table 11: ATOM3D LBA, mean ± standard deviation over three runs. Best per column in **bold**, second-best underlined, by mean. Baselines from Lim et al. [18].

K All CleanSplit variants

Table 12 lists all PEAR variants, namely three atom depths, two alternative readouts ($\otimes_{CG}$, Δ), and the residue model. The shallow atom models nearly match the five-layer one, so the paired readouts are effective without a deep backbone.

Table 12: PDBbind CleanSplit, 5-fold CV ensembles, full variant list.

Method	Params	CASF-2016		Independent	
		RMSE ↓	r ↑	RMSE ↓	r ↑
GenScore	1.06M	1.280	0.807	1.580	0.737
ATOMICA-MLP	—	1.965	0.583	1.960	0.704
GEMS	1.03M	1.274	0.815	1.347	0.828
GEMS + ATOMICA	1.03M	1.292	0.808	1.361	0.822
BA-Pred	17.9M	1.290	0.818	—	—
PEAR (5L)	1.68M	1.258	0.834	1.414	0.822
PEAR (3L)	721K	1.291	0.815	1.425	0.811
PEAR (2L)	749K	1.277	0.821	1.400	0.819
PEAR ($\otimes_{CG}$, $\ell=1$)	1.69M	1.299	0.819	1.452	0.810
PEAR (Δ , $\ell=2$)	1.70M	1.296	0.818	1.444	0.810
PEAR _{residue} (2L)	831K	1.282	0.822	1.383	0.833

L Held-out clusters before finetuning

Table 13 gives the per-cluster Pearson r before any finetuning, for all readout variants. The residue models average slightly higher than the atom models, and no variant of ours leads on every cluster. PEAR, our most accurate model on CASF-2016, leads on none of the seven. Comparison with Table 14 shows that models with higher $\ell_{\max}$ gain more from finetuning than those with $\ell_{\max}=0$, most clearly at the atom scale.

Table 13: Per-cluster Pearson r before finetuning, for all readout variants. The ($\otimes_{\text{CG}}, \ell=1$) row is the model with the direct-CG atom paired readout, matched to the $\otimes_{\text{CG}}^\rho (\ell=0)$ model on CleanSplit RMSE.

Method (Pre-FT)	INVQ	1SQA	2P15	2VW5	3DD0	3F3E	3O9I	Avg.	Min.
PEAR _{backbone}	0.523	0.640	0.410	0.380	0.352	0.219	0.518	0.434	0.219
<i>Atom scale:</i>									
$\otimes_{\text{CG}}^\rho (\ell=0)$	0.506	0.669	0.454	0.402	0.457	0.197	0.483	0.452	0.197
$\otimes_{\text{CG}}^\rho (\ell=1)$	0.507	0.667	0.435	0.393	0.312	0.214	0.384	0.416	0.214
$\otimes_{\text{CG}}^\rho (\ell=2)$	0.510	0.657	0.450	0.373	0.432	0.194	0.378	0.427	0.194
$\otimes_{\text{CG}} (\ell=1)$	0.504	0.669	0.444	0.374	0.436	0.178	0.328	0.419	0.178
$\Delta_{\text{sym}} (\ell=0)$	0.497	0.653	0.438	0.369	0.416	0.212	0.447	0.433	0.212
$\Delta_{\text{sym}} (\ell=1)$	0.508	0.660	0.410	0.392	0.469	0.212	0.458	0.444	0.212
$\Delta_{\text{sym}} (\ell=2)$	0.500	0.643	0.401	0.372	0.489	0.205	0.431	0.434	0.205
$\Delta (\ell=2)$	0.475	0.648	0.410	0.399	0.464	0.216	0.353	0.424	0.216
<i>Residue scale:</i>									
$\otimes_{\text{CG}}^\rho (\ell=0)$	0.517	0.703	0.413	0.493	0.407	0.207	0.414	0.451	0.207
$\otimes_{\text{CG}}^\rho (\ell=1)$	0.534	0.709	0.386	0.478	0.406	0.235	0.382	0.447	0.235
$\otimes_{\text{CG}}^\rho (\ell=2)$	0.521	0.693	0.476	0.470	0.425	0.242	0.404	0.461	0.242
$\Delta_{\text{sym}} (\ell=0)$	0.521	0.715	0.398	0.470	0.432	0.263	0.408	0.458	0.263

M Held-out clusters after finetuning

Table 14 gives the per-cluster Pearson r after FT-25 for all readout variants and the joint ensembles summarized in Table 2.

Table 14: Per-cluster Pearson r after finetuning on 25 labeled complexes (FT-25), for all readout variants and a joint ensemble. Best per column in **bold**.

Method (FT-25)	INVQ	1SQA	2P15	2VW5	3DD0	3F3E	3O9I	Avg.	Min.
GEMS	0.563	0.670	0.556	0.537	0.620	0.386	0.424	0.537	0.386
GEMS + ATOMICA	0.561	0.717	0.610	0.531	0.572	0.385	0.476	0.550	0.385
PEAR _{backbone}	0.548	0.660	0.455	0.593	0.541	0.305	0.523	0.518	0.305
Atom $\otimes_{\text{CG}}^\rho (\ell=0)$	0.526	0.687	0.510	0.603	0.557	0.340	0.562	0.541	0.340
Atom $\otimes_{\text{CG}}^\rho (\ell=1)$	0.546	0.689	0.498	0.597	0.569	0.339	0.557	0.542	0.339
Atom $\otimes_{\text{CG}}^\rho (\ell=2)$	0.542	0.677	0.518	0.619	0.537	0.320	0.573	0.541	0.320
Residue $\otimes_{\text{CG}}^\rho (\ell=0)$	0.560	0.705	0.535	0.645	0.569	0.363	0.611	0.570	0.363
Residue $\otimes_{\text{CG}}^\rho (\ell=1)$	0.573	0.739	0.537	0.634	0.560	0.396	0.587	0.575	0.396
Residue $\otimes_{\text{CG}}^\rho (\ell=2)$	0.563	0.732	0.572	0.646	0.558	0.367	0.624	0.580	0.367
Residue $\Delta_{\text{sym}} (\ell=0)$	0.570	0.727	0.522	0.646	0.574	0.375	0.605	0.574	0.375
Atom + Residue ($\ell=2$)	0.584	0.736	0.577	0.681	0.591	0.388	0.634	0.599	0.388