PAIRED READOUTS FOR PROTEIN–LIGAND BINDING AFFINITY

Anonymous authors
Paper under double-blind review

ABSTRACT

Structure-based models of protein–ligand binding affinity have long been benchmarked by training on PDBbind and testing on CASF-2016, a 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 pretrained embeddings. We ask the complementary question of which inductive biases are worth building into an equivariant message passing network (MPNN) trained from scratch, and focus on how the network can use the unbound state of a complex, obtained by removing intermolecular edges. In a basic equivariant MPNN trained on PDBbind CleanSplit, adding the unbound state as a second pooled input does not improve accuracy. Because the two states contain the same atoms, their embeddings can be combined at each atom, at each intermolecular edge and over the pocket, and we compare equivariant operations for these paired readouts. The resulting model, PEAR (Paired Embeddings of Atoms and Residues), is the most accurate of the compared models on CASF-2016, including those with pretrained embeddings, and removing any of its paired readouts lowers its accuracy. On ATOM3D LBA at 30% sequence identity, a variant of PEAR matches the best reported accuracy. On seven held-out pocket clusters, a residue-scale variant has the highest average correlation of any single model after finetuning on 25 complexes of each target, and finetuning improves each of our models on every cluster.

1 INTRODUCTION

For most of the past decade, the dominant evaluation protocol for structure-based scoring functions has been to train a neural network on PDBbind's general or refined datasets and test on the CASF-2016 test set (Su et al., 2019). Unfortunately, this train/test split suffers from data leakage of close structural, chemical and protein-family analogs (Graber et al., 2025; Li et al., 2024). As a result, determining whether models have low RMSE because they memorize motifs from training examples has been a major concern. Graber et al. (2025) constructed PDBbind CleanSplit, a PDBbind split in which complexes similar to any CASF test example, measured by TM-score, ligand RMSD and Tanimoto similarity, are removed. Under these criteria, nearly half of CASF-2016 turned out to have close analogs in the original training data. When the 3D CNN model Pafnucy (Stepniewska-Dziubinska et al., 2018), 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. Li et al. (2024) reached the same conclusion through LP-PDBBind, an alternative decontaminated reorganization of PDBbind—all scoring functions they retrained had benefited from leakage on the original split. The ATOM3D benchmark (Townshend et al., 2021) 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 models that perform best 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 (Graber et al., 2025), the model proposed alongside CleanSplit, augments a graph network with protein and ligand language model embeddings, while GatorAffinity (Wei et al., 2025), an SE(3)-equivariant network, is pretrained on roughly 1.5 million synthetic complexes. While this showcases the potential of pretraining and self-supervised learning, the advantages it provides may obscure the importance of developing architectures that can better

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

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 building into an architecture 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 (Batzner et al., 2022; Batatia et al., 2022). Following their success on small molecules, there has been considerable effort to apply equivariant architectures to binding affinity prediction (Zhang et al., 2024; Nguyen & Kang, 2025; Li et al., 2026a; Wei et al., 2025). 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 (Öztürk et al., 2018; Yang et al., 2022), voxel-based convolutional networks (Stepniewska-Dziubinska et al., 2018; Jiménez et al., 2018; Kyro et al., 2023), or handcrafted interaction descriptors (Sánchez-Cruz et al., 2021). Moreover, models that incorporate message passing as a component, equivariant or not, are frequently *hybrid* architectures that combine it with convolutional networks (Kyro et al., 2023; Jones et al., 2021), with attention (Sim & Lee, 2026; Lim et al., 2025), or with language model embeddings (Graber et al., 2025).

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 (Behler & Parrinello, 2007). Baking this decomposition into machine-learned interatomic potentials (MLIPs) accordingly allows one network to be applied to molecules of different sizes and compositions (Batatia et al., 2022). 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 (Batzner et al., 2022). On the other hand, binding free energy is an average over an ensemble of configurations (Gilson & Zhou, 2007), 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. Traditionally, scoring functions address this problem by fitting a linear combination of structural proxy terms, some standing in for enthalpic contributions and others for entropic ones. For example, the empirical scoring function of Böhm (1994) takes the form

$$\Delta G_{\mathrm{bind}} = \Delta G_0 + \Delta G_{\mathrm{hb}} \sum_{\text{h-bonds}} f(\Delta R, \Delta\alpha) + \Delta G_{\mathrm{ionic}} \sum_{\text{ionic}} f(\Delta R, \Delta\alpha) + \Delta G_{\mathrm{lipo}} A_{\mathrm{lipo}} + \Delta G_{\mathrm{rot}} N_{\mathrm{rot}}, \tag{1}$$

where each coefficient is fitted to measured binding constants and $f(\Delta R, \Delta\alpha)$ penalises deviation from ideal distance and angle. Expectedly, such functions come in many forms, differing in which terms are included and how each is parametrised (Wang et al., 2004; Cheng et al., 2009).

With the advent of neural network scoring functions, these terms no longer need to be chosen by hand, resulting in many possible architectures on the same set of input features. Similar to their handcrafted counterparts, many modeling choices have become empirically interchangeable in the context of 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 (Gomes et al., 2017; Zhang et al., 2023; Li et al., 2026b), or combine the two with a more general MLP (Qu et al., 2024). 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. Equivariant encoders are increasingly used as components of larger pipelines (Wei et al., 2025; Nguyen & Kang, 2025), and are adopted in pursuit of models that generalize to unseen targets (Li et al., 2026b). Having outlined the reason why there is a wide variety of proposed architectures, we believe this represents an opportunity to better adapt equivariant neural networks to the binding affinity problem. Improving the equivariant encoder is already a rich subject in its own right (Batzner et al., 2022; Batatia et al., 2022). Independent of that, one can also add learnable modules that use the encoder's output as an abstract input representation of the structure, as was done in CheapNet, for example (Lim et al., 2025). If the unbound state of the complex is available, these modules can additionally 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. The prediction head can then use these to influence its predictions, similar to the way an empirical scoring function uses additional terms to improve its regression. To our knowledge this axis has not been explored (Figure 1).

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. 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, which motivates the present study.
- We discuss which operations are available for combining node, edge and graph level features between the two states in an equivariant manner, and measure their effects 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.

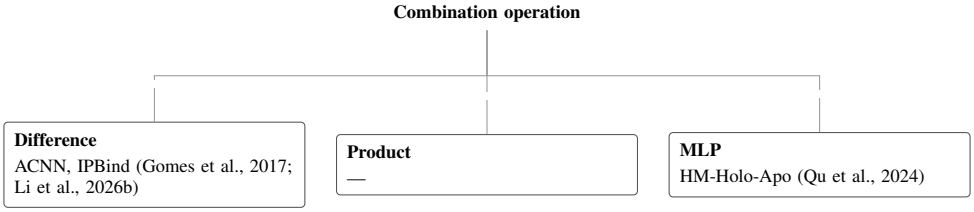
2 TWO-STATE MODELS

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 (Figure A1). 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 (Jiang et al., 2021; Moon et al., 2022; Sim & Lee, 2026), or explicitly maintain parallel representations of the bound and unbound states, in some cases through weight-shared networks (Gomes et al., 2017; Zhang et al., 2023; Qu et al., 2024; Li et al., 2026b), 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 (Figure 1). In adjacent fields, pairs of embeddings have been combined with more care, including in equivariant settings, which introduces additional computational considerations.

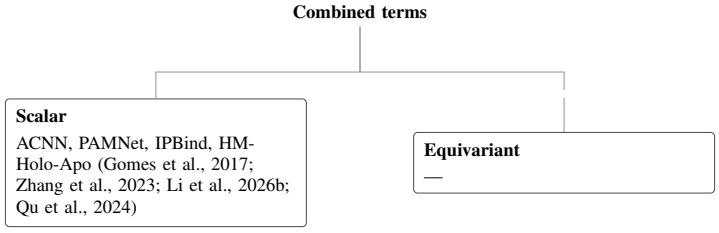
Node-wise comparisons. Comparing two structures node by node has been done in both graph similarity learning and kernel methods. Li et al. (2019) compare the two graphs node by node at every message passing layer, using attention to match each node against the nodes of the other graph. They report that this outperforms embedding each graph separately and concatenating the two graph vectors. Writing h_i for the features of node i, each node is assigned the difference between h_i and its closest match in the other graph,

$$\sum_j \mu_{j\to i} \;=\; h_i \;-\; \sum_j a_{j\to i}\, h_j, \tag{2}$$

where j runs over the nodes of the other graph, $a_{j\to i}$ is an attention weight. De et al. (2016) compare two molecules by first pairing their atomic environments by some measure of best match,



(a) Operation combining the two representations.



(b) Terms combined, by angular order.

Figure 1: Specific operations employed to combine bound and unbound states.

then summing the resulting similarities,

$$\hat{K}(A,B) = \frac{1}{N} \max_{\pi} \sum_{i} k\big(\mathcal{X}_i^A,\, \mathcal{X}_{\pi_i}^B\big), \tag{3}$$

where k is the similarity between environments $\mathcal{X}$. 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 both 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 (Lim et al., 2025).

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 (Katchalski-Katzir et al., 1992; Ritchie & Kemp, 2000; Neveu et al., 2016). The SOAP kernel (Bartók et al., 2013; De et al., 2016) 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 (Lin et al., 2015; Fukui et al., 2016). As mentioned earlier, these are seldom used in neural network scoring functions (Figure 1).

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, it 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. (2019) 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 $L = 0$, which means that only the $(\ell, \ell) \to 0$ paths from the previous layer can be kept and the $\ell_1 \neq \ell_2$ paths are

discarded. In a network such as Anderson et al. (2019) or Batatia et al. (2022) this is not an issue, because the product serves as a nonlinearity in the *internal* layers, and as such all paths below a given $\ell_{\max}$ serve as inputs to the next layer. In a terminal or *readout* module, however, we must contract the output of the product against a third factor to keep the paths $(\ell_1, \ell_2) \to L$ with $L \leq \ell_{\max}$. This way of extracting invariants is described in (Bartók et al., 2013) as the *bispectrum*, whereas the direct reduction is the *power spectrum*.

In Table 1 we summarise such 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. 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

$$\tilde{h}_i^{(c,\ell,m)} \;=\; \sum_{j \in N(i)} \big(W h_j\big)_c \, R_\ell(r_{ij}) \, Y_\ell^m(\hat{\mathbf{r}}_{ij}) \tag{4}$$

for a node embedding expanded about a centre. Equation 4 is a weighted sum of contributions from the neighbours of i. In SOAP and in the atomic cluster expansion the same object is called a neighbour *density*, with fixed weights in place of $W h_j$. As alluded to earlier, this allows features to be combined in small local groups defined by the environment of an atom. A feature of the spherical harmonic expansion is that the same atom contributes differently to two overlapping neighbourhoods, since its direction from each centre is different. In MACE (Batatia et al., 2022), an equivariant message passing interatomic potential, the same density is built in each layer from the previous layer's node features.

Table 1: Operations that combine two representations. Below the rule we suggest candidate operations that combine bound and unbound counterparts, with the symbols used in Table 3 for the ones measured there. Rows sharing an operation differ only in how it is reduced. The reduction step takes the equivariant output to scalars, where $[\,\cdot\,]_0$ denotes the $L = 0$ part of a product. $\to$ indicates that the output is equivariant for downstream purposes, and — that the output is already scalar.

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

3 METHOD

Problem definition. We predict the binding affinity of a protein–ligand complex from its crystal structure. The complex is a graph $G = (V, E) = (V_l \cup V_p,\ E_l \cup E_p \cup E_{\mathrm{inter}})$, 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. The loss is the mean squared error on $\mathrm{p}K$.

Graph construction and features. We build the atom-scale graph following Yang et al. (2023). The residue-scale graph has one pocket node per residue, placed at the residue centroid, whose features are the mean of the residue's atom descriptors, an amino-acid embedding, and a B-factor. Full featurization details are in Appendix E. In both cases the unbound copy is the same graph with E_{inter} removed.

Backbone and readouts. A shared encoder f_θ runs on the complex graph G and on the same graph with intermolecular edges removed, so there are two embeddings at every node and every edge. Let $h_i^{(t)}$ be the state at node i after layer t, and $e_{ij}^{(t)}$ the edge state. Each layer is

$$h_i^{(t+1)} \;=\; \mathrm{U}\Big(h_i^{(t)},\; \bigoplus\nolimits_{j\in N(i)} \mathrm{M}\big(h_i^{(t)},\, h_j^{(t)},\, e_{ij}^{(t)}\big)\Big), \tag{5}$$

with M and U equivariant maps on features of angular order up to $\ell_{\max}$. The message M is the Clebsch–Gordan product of $h_j^{(t)}$ with the spherical harmonics of the bond direction $\hat{\mathbf{r}}_{ij}$, with path weights given by a radial MLP of the interatomic distance, the edge type and the scalar features of i and j (Appendix G). Each intermolecular edge has a hidden state $e_{ij}^{(t)}$, initialized from its two endpoints by a symmetric encoder and updated residually after every layer from the new node states. The updated edge states are added to their endpoint nodes through a learned projection, which we call edge feedback (Appendix D). The atom-scale model has five layers and the residue-scale model two, both at $\ell_{\max}$=0 (Appendix E), and Appendix D gives the full forward pass. Let v index a *site*, which may be a node, an intermolecular edge, or the pocket as a whole, and let z_v be that site's embedding after the final layer. A paired readout combines the two copies at v, reduces the result to scalars, and aggregates over the sites of its level,

$$s \;=\; \mathcal{A}\Big(\big[\,\Phi\big(z_v^{b},\, z_v^{ub}\big)\,\big]_{\text{red}}\Big), \qquad \hat{y} \;=\; g_\theta\big([\,s_1;\ldots;s_K\,]\big), \tag{6}$$

for K readouts, whose scalars are concatenated to form the input of the head g_θ. Φ and $[\,\cdot\,]_{\text{red}}$ can be instantiated as any of the operations and reductions suggested in Table 1.

PEAR uses four paired readouts. At each atom the operation is

$$\otimes^{\rho}_{\text{CG}} \;:\; \big[\,\Phi(z_v^{b}, z_v^{ub})\,\big]_{\text{red}} \;=\; \Big[\big[\,\tilde{z}_v^{b} \otimes_{\text{CG}} \tilde{z}_v^{ub}\,\big]_{L} \otimes \tilde{z}_v^{ub}\Big]_{0}, \tag{7}$$

where $\tilde{z}_v$ is a density expanded about the site v. At each intermolecular edge the operation is a concatenation of the difference operation and an MLP. The same product of Equation 7 is taken at the pocket, once over atoms and once over residue centroids. Writing Φ^{ρ} for Equation 7 and $\Delta_{ij} = \big[[\,z_{ij}^{b} - z_{ij}^{ub}\,]_{L} \otimes z_{ij}^{ub}\big]_{0}$, the full model is

$$\begin{aligned}\hat{y} \;=\; g_\theta\Big[\; &\underbrace{\mathcal{A}\big(z^{b}\big)}_{\text{bound}} \;;\; \underbrace{\textstyle\sum_i \Phi^{\rho}\big(\tilde{z}_i^{b}, \tilde{z}_i^{ub}\big)}_{\text{atom}} \;;\; \underbrace{\textstyle\sum_{ij\in E_{\text{inter}}} \big[\,\Delta_{ij}\,;\,\psi\big(z_{ij}^{b}, z_{ij}^{ub}\big)\,\big]}_{\text{edge}} \;; \\ &\underbrace{\Phi^{\rho}\big(\tilde{Z}^{b}, \tilde{Z}^{ub}\big)}_{\text{pocket, atoms}} \;;\; \underbrace{\Phi^{\rho}\big(\tilde{Z}_{\text{res}}^{b}, \tilde{Z}_{\text{res}}^{ub}\big)}_{\text{pocket, residues}} \;\Big],\end{aligned} \tag{8}$$

where $\mathcal{A}(z^b)$ is the pooled output of the encoder on the bound graph. It is the only input to the head that does not involve the unbound copy. Every other input to the head is a paired readout computed from both copies. The atom density $\tilde{z}_v$ is that of Equation 4, and $\tilde{Z}$ is the same expansion taken about the ligand centroid over the pocket,

$$\tilde{Z}^{(c,\ell,m)} \;=\; \sum_{j\in V_p} \big(W' h_j\big)_c\, R_\ell(r_j)\, Y_\ell^{m}(\hat{\mathbf{r}}_j), \tag{9}$$

with $\mathbf{r}_j$ measured from the centroid and W' an independent projection. $\tilde{Z}_{\text{res}}$ sums over residue centroids using the same method. Each centroid is placed at the mean position of its residue's heavy atoms, and the scalar features of those atoms are averaged to form the centroid's own h_j.

We refer to the atom-scale model as PEAR and to the residue-scale model as $\text{PEAR}_{\text{residue}}$, and write $\text{PEAR}_{\text{backbone}}$ and $\text{PEAR}_{\text{residue,backbone}}$ for the same models without paired readouts. In the results we hold the edge and pocket readouts fixed and vary the atom readout across the operations of Table 1.

Table 2: PDBbind CleanSplit, 5-fold CV ensembles. PEAR is most accurate on CASF-2016, and $\text{PEAR}_{\text{residue}}$ has the highest correlation on the independent subset. Baseline numbers come from Kopko et al. (2025) and Sim & Lee (2026), and the GEMS variants use frozen language-model and ATOMICA embeddings. Best in **bold**. Depth and readout variants are in Appendix I.

		CASF-2016		Independent	
Method	Params	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
$\text{PEAR}_{\text{residue}}$	831K	1.282	0.822	1.383	**0.833**
+ joint ensemble	—	**1.233**	—	—	—

4 EXPERIMENTS

Evaluation protocols. We evaluate our architecture under three protocols, each of which removes a different kind of train–test similarity. *PDBbind CleanSplit* (Graber et al., 2025) removes from PDBbind v2020 every training complex that is similar to a CASF-2016 test complex in protein structure, ligand, binding pose and affinity, or that shares its ligand at a similar affinity, together with redundant complexes within the training set (Appendix F). Test targets and their families can still appear in training with other ligands, so CleanSplit tests generalization to new complexes of largely familiar targets. *ATOM3D LBA* (Townshend et al., 2021) partitions the PDBbind v2019 refined set so that test proteins share at most 30% (or 60%) sequence identity with training proteins. Following Lim et al. (2025), each LBA experiment is run three times with different random seeds. The *held-out pocket clusters* of Kopko et al. (2025) remove from training the PLINDER pocket-lDDT community of each of seven CASF-2016 targets (1NVQ, 1SQA, 2P15, 2VW5, 3DD0, 3F3E and 3O9I) and apply CleanSplit filtering to the remaining complexes, which tests generalization to new binding-site geometry. For the pocket clusters, we additionally finetune each model on 25 labeled complexes of the held-out target and evaluate it on the remaining cluster members, as in Kopko et al. (2025) (Appendix F). In particular, we ablate our architecture in detail on CleanSplit, and make small adjustments to the model on other tasks to reach competitive performance.

Preliminary experiment on a basic MPNN. We first test a basic equivariant MPNN, which only passes messages between atoms as described in Equation 5, in two settings under the CleanSplit protocol, one where the bound and unbound graphs are pooled and then concatenated as inputs to an MLP (the two-state model), and another where the unbound graph is omitted (the one-state model). At two and five layers and at $\ell_{\max}$ of 0, 1 and 2, we find that 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 3, upper block). This motivated us to devise the operations outlined above to allow the bound and unbound states to interact in more elaborate ways.

Atom paired readout ablations. We test different atom paired readouts across $\ell_{\max}$ of the backbone and find that the $\ell_{\max}$=0 model with $\otimes^{\rho}_{\text{CG}}$ has the lowest mean RMSE, 1.293 (Table 3, middle block). The $\ell_{\max}$=2 model with Δ has almost the same RMSE, 1.297, and we use the former because its $\ell_{\max}$=0 backbone has fewer Clebsch–Gordan paths in each message-passing layer, which makes the model less expensive from a training time perspective.

Experiments on CleanSplit. Table 2 reports 5-fold cross-validated ensembles on CASF-2016 and on its independent subset, the 144 complexes that fall below the CleanSplit similarity thresholds against all of unfiltered PDBbind (Graber et al., 2025). We find that PEAR is the most accurate method on CASF-2016, and $\text{PEAR}_{\text{residue}}$ has the highest correlation on the independent subset. We additionally find that removing any paired readout module, and, in addition, the edge feedback or edge update mechanisms (Appendix D) in the backbone, always results in poorer performance (Table 3, lower block).

Table 3: CleanSplit test RMSE, mean ± std over five seeds. The upper block is the basic MPNN of the preliminary experiment, without edge states, which pools the bound and unbound copies separately, the middle block varies the atom paired readout, and the lower block removes components of PEAR. Models are parameter-matched across $\ell_{\max}$ (Appendix E).

Configuration	$\ell_{\max}$=0	$\ell_{\max}$=1	$\ell_{\max}$=2
2 layers, one state	1.441 ± 0.007	1.453 ± 0.016	1.405 ± 0.023
2 layers, two states	1.460 ± 0.012	1.443 ± 0.035	1.489 ± 0.021
5 layers, one state	1.386 ± 0.034	1.389 ± 0.057	1.392 ± 0.038
5 layers, two states	1.423 ± 0.038	1.399 ± 0.027	1.384 ± 0.024
Δ	1.326 ± 0.020	1.313 ± 0.032	$\mathbf{1.297} \pm 0.012$
Δ_{sym}	1.342 ± 0.037	1.326 ± 0.018	1.338 ± 0.014
Δ+MLP	1.366 ± 0.022	1.302 ± 0.020	1.316 ± 0.019
Δ_{sym}+MLP	1.334 ± 0.019	1.325 ± 0.018	1.326 ± 0.023
$\|\Delta\|^2$	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
$\otimes_{\mathrm{CG}}$	1.352 ± 0.012	1.303 ± 0.014	1.315 ± 0.011
$\otimes^{\rho}_{\mathrm{CG}}$	$\mathbf{1.293} \pm 0.015$	1.344 ± 0.021	1.316 ± 0.012
Components removed from PEAR			
Edge feedback	1.323 ± 0.014	—	—
Pocket paired readout (residues)	1.326 ± 0.033	—	—
Pocket paired readout (atoms)	1.343 ± 0.037	—	—
Edge updates	1.344 ± 0.039	—	—
Atom paired readout	1.359 ± 0.037	—	—
Edge paired readout	1.359 ± 0.051	—	—
All paired readouts	1.386 ± 0.023	1.391 ± 0.013	1.386 ± 0.019

Experiments on held-out pocket clusters. Table 4 reports the mean and worst-case Pearson r over the seven held-out clusters, before and after finetuning. We find that before finetuning the average correlation of our models is below that of GEMS. Part of this gap may come from the protein language model embeddings that GEMS uses as residue features, which Graber et al. (2025) found to improve its accuracy on CleanSplit and which are trained on sequence databases that include proteins from the held-out families. Finetuning improves our models more than it improves GEMS. Interestingly, finetuning also raises the correlation of every model we trained on each of the seven clusters, whereas it lowers the correlation of GEMS on four of them (Kopko et al., 2025). After finetuning, $\mathrm{PEAR}_{\mathrm{residue}}$ at ℓ=2 has the highest average correlation of any single model, and $\mathrm{PEAR}_{\mathrm{residue}}$ at ℓ=1 has the highest worst-case correlation. We additionally find that after finetuning every PEAR and $\mathrm{PEAR}_{\mathrm{residue}}$ setting has a higher average correlation than its backbone model, and that $\mathrm{PEAR}_{\mathrm{residue}}$ is more accurate than PEAR at every ℓ.

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, $\mathrm{PEAR}_{\mathrm{backbone}}$, and $\mathrm{PEAR}_{\mathrm{LBA}}$, a variant of PEAR developed specifically for this task. In particular in $\mathrm{PEAR}_{\mathrm{LBA}}$, four settings are adjusted—the input features, backbone, and general structure of the model remain the same (Appendix E). On LBA 30%, $\mathrm{PEAR}_{\mathrm{LBA}}$ achieves the lowest RMSE and highest Pearson correlation coefficient as well as the second highest Spearman correlation coefficient out of the models considered (Table 5). Its differences from CheapNet are within one standard deviation on all three metrics (Appendix L), so we do not regard it 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 $\mathrm{PEAR}_{\mathrm{backbone}}$, once again showcasing the utility of paired readouts for generalization tasks.

Ensembles of the two scales. We stratify the CASF-2016 test set into quartiles along six per-complex axes and find that PEAR is more accurate on small ligands, rigid pockets and loose interfaces, while $\mathrm{PEAR}_{\mathrm{residue}}$ is more accurate on large ligands, flexible pockets and tight interfaces (Appendix K). Accordingly, ensembling PEAR and $\mathrm{PEAR}_{\mathrm{residue}}$ lowers the CASF-2016 RMSE to 1.233 (Table 2) and averages $r = 0.599$ on the held-out clusters after finetuning, above either model (Appendix J).

Table 4: Held-out pocket clusters, mean and worst-case Pearson r before (Pre-FT) and after fine-tuning 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 and the full readout grid are in Appendix J. Baselines from Kopko et al. (2025). Among single models, the best per column is in **bold** and the second-best underlined. The ensemble is also in bold where it exceeds every single model.

	Pre-FT		FT-25	
Method	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
$\text{PEAR}_{\text{backbone}}$	0.434	0.219	0.518	0.305
$\text{PEAR}_{\text{residue,backbone}}$	0.430	0.218	0.555	0.379
PEAR	0.452	0.197	0.541	0.340
PEAR, ℓ=1	0.416	0.214	0.542	0.339
PEAR, ℓ=2	0.427	0.194	0.541	0.320
$\text{PEAR}_{\text{residue}}$	0.451	0.207	0.570	0.363
$\text{PEAR}_{\text{residue}}$, ℓ=1	0.447	0.235	0.575	**0.396**
$\text{PEAR}_{\text{residue}}$, ℓ=2	0.461	0.242	**0.580**	0.367
PEAR + $\text{PEAR}_{\text{residue}}$, ℓ=2	0.479	0.242	**0.599**	0.388

Table 5: ATOM3D LBA at 30% and 60% sequence identity. All values are means over three runs, and standard deviations are given in Appendix L. Every baseline is quoted from Lim et al. (2025), which reports the same three-run protocol and collects the earlier entries from Townshend et al. (2021), Somnath et al. (2021) and Kong et al. (2024). A dash marks a split on which a baseline was not reported. Best per column in **bold**, second-best underlined.

		LBA 30%			LBA 60%		
Method	Params	RMSE ↓	Pearson ↑	Spearman ↑	RMSE ↓	Pearson ↑	Spearman ↑
Atom3D-GNN	0.38M	1.601	0.545	0.533	1.408	0.743	0.743
Atom3D-3DCNN	2.22M	1.416	0.550	0.553	1.621	0.608	0.615
ProNet	1.39M	1.463	0.551	0.551	1.343	0.765	0.761
HoloProt	1.44M	1.464	0.509	0.500	1.365	0.749	0.742
MaSIF	0.62M	1.484	0.467	0.455	1.426	0.709	0.701
SchNet	0.21M	1.370	0.590	0.571	—	—	—
MACE	3.91M	1.372	0.612	0.592	—	—	—
LEFTNet	0.85M	1.366	0.592	0.580	—	—	—
GET	0.69M	1.327	0.620	0.611	—	—	—
CheapNet	1.39M	1.311	0.642	**0.639**	1.238	**0.794**	0.789
pretrained							
Uni-Mol	47.6M	1.434	0.565	0.540	1.357	0.753	0.750
ProFSA	>47.6M	1.377	0.628	0.620	1.377	0.764	0.762
BindNet	>47.6M	1.340	0.632	0.620	**1.230**	0.793	0.788
$\text{PEAR}_{\text{backbone}}$	1.35M	1.386	0.621	0.613	1.328	0.773	0.775
PEAR	1.68M	1.340	0.629	0.615	1.261	0.791	**0.793**
PEAR_{LBA}	1.60M	**1.308**	**0.647**	0.634	1.291	0.782	0.782

5 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. To this end, we benchmarked PEAR and its variants on CleanSplit, ATOM3D LBA and held-out pocket clusters. We contribute parameterizations of these models that match or exceed the performance of known models in the literature. Furthermore, 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, or to improve the backbone independently of the readouts.

REFERENCES

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*, 2021. arXiv:2108.09926.

Brandon Anderson, Truong Son Hy, and Risi Kondor. Cormorant: Covariant Molecular Neural Networks. In *Advances in Neural Information Processing Systems*, 2019.

Albert P. Bartók, Risi Kondor, and Gábor Csányi. On representing chemical environments. *Physical Review B*, 87:184115, 2013.

Ilyes Batatia, Dávid P. 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 *Advances in Neural Information Processing Systems*, pp. 11423–11436, 2022.

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. *Nature Communications*, 13:2453, 2022.

Jörg Behler and Michele Parrinello. Generalized Neural-Network Representation of High-Dimensional Potential-Energy Surfaces. *Physical Review Letters*, 98:146401, 2007.

Hans-Joachim Böhm. The development of a simple empirical scoring function to estimate the binding constant for a protein-ligand complex of known three-dimensional structure. *Journal of Computer-Aided Molecular Design*, 8(3):243–256, 1994. doi: 10.1007/BF00126743.

Tiejun Cheng, Xun Li, Yan Li, Zhihai Liu, and Renxiao Wang. Comparative Assessment of Scoring Functions on a Diverse Test Set. *Journal of Chemical Information and Modeling*, 49(4):1079–1093, 2009.

Sandip De, Albert P. Bartók, Gábor Csányi, and Michele Ceriotti. Comparing molecules and solids across structural and alchemical space. *Physical Chemistry Chemical Physics*, 18:13754–13769, 2016.

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.

Johannes Gasteiger, Janek Groß, and Stephan Günnemann. Directional Message Passing for Molecular Graphs. In *International Conference on Learning Representations*, 2020.

Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural Message Passing for Quantum Chemistry. In *International Conference on Machine Learning*, 2017.

Michael K. Gilson and Huan-Xiang Zhou. Calculation of Protein-Ligand Binding Affinities. *Annual Review of Biophysics and Biomolecular Structure*, 36:21–42, 2007.

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.

Rohan Gorantla, Aryo Pradipta Gema, Ifigeneia Xanthi Yang, Álvaro Serrano-Morrás, Benjamin Suutari, Jordi Juárez-Jiménez, and Antonia S. J. S. Mey. Learning binding affinities via fine-tuning of protein and ligand language models. *Journal of Chemical Information and Modeling*, 65:12279–12291, 2025.

David Graber, Peter Stockinger, Fabian Meyer, Denis Werber, Claus Horn, Rebecca Buller, and Sarah Mishra. Resolving data bias improves generalization in binding affinity prediction. *Nature Machine Intelligence*, 7:1713–1725, 2025.

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.

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.

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.

E. Katchalski-Katzir, I. Shariv, M. Eisenstein, A. A. Aflalo, and I. A. Vakser. Molecular surface recognition: determination of geometric fit between proteins and their ligands by correlation techniques. *Proceedings of the National Academy of Sciences*, 89:2195–2199, 1992.

Risi Kondor. The principles behind equivariant neural networks for physics and chemistry. *Proceedings of the National Academy of Sciences*, 122(41):e2415656122, 2025. doi: 10.1073/pnas.2415656122.

Risi Kondor, Zhen Lin, and Shubhendu Trivedi. Clebsch-Gordan Nets: a Fully Fourier Space Spherical Convolutional Neural Network. In *Advances in Neural Information Processing Systems*, 2018.

Xiangzhe Kong, Wenbing Huang, and Yang Liu. Generalist Equivariant Transformer Towards 3D Molecular Interaction Learning. In *International Conference on Machine Learning*, 2024.

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.

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.

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, 2026a. doi: 10.1007/s12539-025-00745-z.

Jie Li, Xingyi Guan, Oufan Zhang, Kunyang Sun, Yingze Wang, Dorian Bagni, and Teresa Head-Gordon. Leak Proof PDBBind: A Reorganized Dataset of Protein-Ligand Complexes for More Generalizable Binding Affinity Prediction. *ArXiv*, 2024. arXiv:2308.09639.

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, 2026b. doi: 10.1109/OJEMB.2026.3667030. arXiv:2504.16261.

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, pp. 3835–3845, 2019.

Hyukjun Lim, Sun Kim, and Sangseon Lee. CheapNet: Cross-attention on Hierarchical representations for Efficient protein-ligand binding Affinity Prediction. In *The Thirteenth International Conference on Learning Representations*, 2025. URL `https://openreview.net/forum?id=A1HhtITVEi`.

Tsung-Yu Lin, Aruni RoyChowdhury, and Subhransu Maji. Bilinear CNN Models for Fine-Grained Visual Recognition. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 1449–1457, 2015.

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*, 13:3661–3673, 2022. doi: 10.1039/D1SC06946B.

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.

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*, 2025. doi: 10.1021/acs.jcim.5c00773. arXiv:2504.04654.

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.

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.

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.

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.

Kristof T. Schütt, Oliver T. Unke, and Michael Gastegger. Equivariant Message Passing for the Prediction of Tensorial Properties and Molecular Spectra. In *International Conference on Machine Learning*, 2021.

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.

Vignesh Ram Somnath, Charlotte Bunne, and Andreas Krause. Multi-Scale Representation Learning on Proteins. In *Advances in Neural Information Processing Systems*, 2021.

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. doi: 10.1093/bioinformatics/bty374.

Minyi Su, Qifan Yang, Yu Du, Guoqin Feng, Zhihai Liu, Yan Li, and Renxiao Wang. Comparative Assessment of Scoring Functions: The CASF-2016 Update. *Journal of Chemical Information and Modeling*, 59(2):895–913, 2019.

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 *NeurIPS Datasets and Benchmarks Track*, 2021.

Renxiao Wang, Yipin Lu, Xueliang Fang, and Shaomeng Wang. An Extensive Test of 14 Scoring Functions Using the PDBbind Refined Set of 800 Protein-Ligand Complexes. *Journal of Chemical Information and Computer Sciences*, 44(6):2114–2125, 2004.

Jinyuan Wei, Yao Zhang, Paul A. Ramdhan, Zhe Huang, Gustavo Seabra, Zhangyang Jiang, Chenglong Li, and Yanjun Li. GatorAffinity: Boosting Protein–Ligand Binding Affinity Prediction with Large-Scale Synthetic Structural Data. *bioRxiv*, 2025. 2025.09.29.679384.

Maurice Weiler, Mario Geiger, Max Welling, Wouter Boomsma, and Taco Cohen. 3D Steerable CNNs: Learning Rotationally Equivariant Features in Volumetric Data. In *Advances in Neural Information Processing Systems*, 2018.

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.

Ziduo Yang, Weihe Zhong, Qiujie 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. doi: 10.1021/acs.jpclett.2c03906.

Shuo Zhang, Yang Liu, and Lei Xie. A Universal Framework for Accurate and Efficient Geometric Deep Learning of Molecular Systems. *Scientific Reports*, 2023. arXiv:2311.11228.

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.

A LLM USAGE

In line with the ICLR policy on large language models (LLMs), we disclose how LLMs were used in preparing this submission.

Writing and editing. We used an LLM-based assistant to polish the prose of drafts written by the authors, tightening wording and fixing grammar. The scientific content, including the research questions, the analysis of operations on paired embeddings, the design of PEAR, and the interpretation of results, was conceived and written by the authors.

Typesetting. An LLM assisted with LaTeX formatting, including table layout and the taxonomy figure. All tables report numbers taken directly from our experiment logs or from the cited papers; no reported value was produced by an LLM.

LLMs were not used for research ideation, and they are not authors of this work. The authors reviewed all LLM-assisted text and take full responsibility for the content of the paper, including any errors.

B INPUT AND REPRESENTATION CHOICES

The first of the two surveys of Section 2.

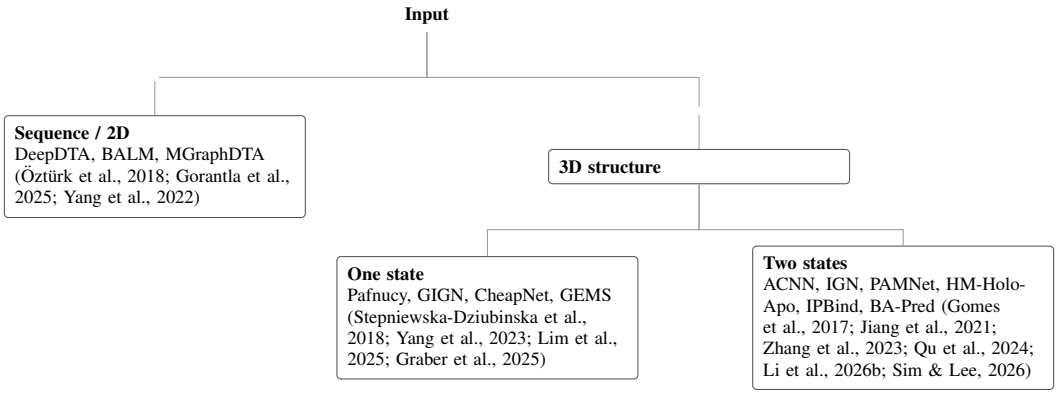
C EQUIVARIANT OPERATIONS ON PAIRS OF EMBEDDINGS

We give the equivariant paired readout operations available at each atom on the pair (h_i^b, h_i^{ub}), organized by tensor degree. Each operation below produces an equivariant output, which is then reduced to scalars before the non-equivariant MLP head. The reduction is a channel-wise inner product against $h_{ub}^{(\ell)}$ at each ℓ, the standard $(\ell, \ell) \to 0$ rotation-invariant contraction $\sum_m f_m \cdot h_{ub,m}$ used in the SOAP kernel (Bartók et al., 2013; De et al., 2016). The Δ_{sym} variant instead contracts the $\ell > 0$ features against $h_b^{(\ell)} + h_{ub}^{(\ell)}$, giving $\|h_b^{(\ell)}\|^2 - \|h_{ub}^{(\ell)}\|^2$ per channel. The operations differ in how they combine bound and unbound features upstream of this reduction. For the $\ell_{\max}$ sweep of Table 3, the hidden multiplicities are set to hold the parameter count fixed across angular orders: 384×0^e at $\ell_{\max}$=0, $224 \times 0^e + 224 \times 1^o$ at $\ell_{\max}$=1, and $160 \times 0^e + 160 \times 1^o + 160 \times 2^e$ at $\ell_{\max}$=2.

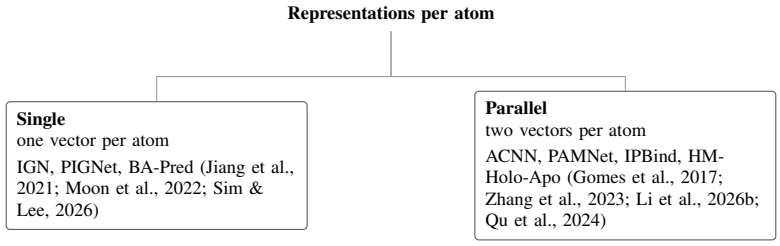
C.1 LINEAR

The most general equivariant linear map from two sets of irreps to a third acts independently within each angular order ℓ (Kondor, 2025). Within each ℓ, the map can mix the C multiplicity channels freely, so the space of equivariant linear maps is $2C^2$-dimensional per ℓ:

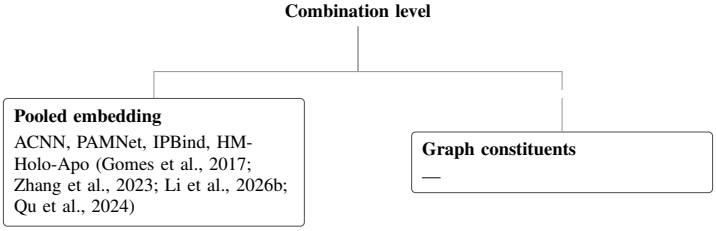
$$\delta h_i^{(\ell)} = A^{(\ell)} h_i^{b,(\ell)} + B^{(\ell)} h_i^{ub,(\ell)}, \tag{10}$$



(a) Input data.



(b) Representations per atom, for models given two states as input.



(c) Level at which the two representations are combined, for models holding parallel representations.

Figure A1: How models take in and combine two states of a complex.

where $A^{(\ell)}, B^{(\ell)} \in \mathbb{R}^{C\times C}$ are learnable matrices acting on the multiplicity index. The diagonal choice $A^{(\ell)} = \mathrm{diag}(\alpha_1, \ldots, \alpha_C)$, $B^{(\ell)} = \mathrm{diag}(\beta_1, \ldots, \beta_C)$ recovers the per-channel form $\alpha_c\, h_i^{b,(\ell,c)} + \beta_c\, h_i^{ub,(\ell,c)}$, a $2C$-dimensional subspace. A linear equivariant map acts within each ℓ and cannot mix different ℓ's. The difference $h_i^{b,(\ell)} - h_i^{ub,(\ell)}$ ($A = I$, $B = -I$) is the natural choice, since it is zero when $h_b = h_{ub}$ and is maximal when they disagree; but other linear combinations are equally valid and could be learned. To reduce the difference to a scalar for the MLP head, we take the per-channel inner product against $h_{ub}^{(\ell)}$ at each ℓ, giving $h_b^{(\ell)} \cdot h_{ub}^{(\ell)} - \|h_{ub}^{(\ell)}\|^2$ per channel (Δ). Contracting the $\ell > 0$ features against $h_b^{(\ell)} + h_{ub}^{(\ell)}$ instead gives $\|h_b^{(\ell)}\|^2 - \|h_{ub}^{(\ell)}\|^2$ (Δ_{sym}), antisymmetric under bound/unbound swap and zero when $h_b = h_{ub}$.

C.2 Bilinear

The Clebsch–Gordan product is the unique bilinear equivariant operation on irreducible representations (Kondor et al., 2018). Given $h_i^{b,(\ell_1)}$ and $h_i^{ub,(\ell_2)}$, it produces outputs transforming as $\ell = |\ell_1 - \ell_2|, \ldots, \ell_1 + \ell_2$. Unlike the linear map, it couples different angular orders—for instance, it can detect whether a shift in the dipolar environment ($\ell = 1$) correlates with a change in the quadrupolar environment ($\ell = 2$).

Coupling same-order channels $(\ell, \ell) \to 0$ produces invariant outputs,

$$\left(h_i^{b,(\ell)} \otimes h_i^{ub,(\ell)}\right)^{(0)} \propto \sum_m h_{i,m}^{b,(\ell)}\, h_{i,m}^{ub,(\ell)}, \tag{11}$$

the $(1,1) \to 0$ component being a dot product measuring alignment between the bound and unbound vector environments. This contraction over m is the same rotation-invariant form used in the SOAP kernel (Bartók et al., 2013; De et al., 2016), with the shared encoder in place of SOAP's shared basis. The full CG product is more expensive, scaling as C^2 bilinear terms per (ℓ_1, ℓ_2) pair per atom, but several standard strategies reduce this cost. These include projecting to fewer channels before applying the product, keeping only diagonal pairs (ℓ, ℓ), and truncating high-ℓ outputs. Another option is gating (Weiler et al., 2018), which uses only the $(0, \ell) \to \ell$ channel of the product so that bound scalars elementwise-modulate the unbound features.

C.3 Beyond bilinear

For the scalar ($\ell = 0$) components, equivariance does not constrain the function, and any smooth function is admissible. In particular, concatenation followed by a nonlinear map,

$$f_\theta\Big(\Big[h_i^{b,(0)};\ h_i^{ub,(0)}\Big]\Big), \tag{12}$$

subsumes both the linear and bilinear operations on the scalars.

D Full architecture

This appendix gives one forward pass of PEAR from inputs to prediction. Algorithm 1 lists the steps in order, and the equations below define each of them. Unless stated otherwise, the equations apply to both scales. Sizes are those of PEAR (Table 6).

Inputs. The complex is the graph $G = (V_l \cup V_p,\ E_l \cup E_p \cup E_{\text{inter}})$ of Section 3, and the unbound copy is $G^{ub} = (V_l \cup V_p,\ E_l \cup E_p)$. Each node has a 35-dimensional descriptor whose first 10 entries are the element one-hot x_i^{el} and whose remaining 25 entries x_i^{sc} are used as scalars (Table 7). The initial node state is

$$h_i^{(0)} \ =\ W_{\text{in}}\big[\, E_{\text{el}}\, x_i^{\text{el}};\ x_i^{\text{sc}}\,\big], \tag{13}$$

with $E_{\text{el}} \in \mathbb{R}^{16\times 10}$ a learned element embedding and W_{in} a linear map to 384 scalar channels ; W_{in} is a single linear map, with no nonlinearity before the first layer. Each edge (i, j) has the distance r_{ij}, the direction $\hat{\mathbf{r}}_{ij}$, a 14-dimensional radial basis $b(r_{ij})$ on 0–5 Å, and a learned 16-dimensional edge-type embedding τ_{ij}.

Message passing. For $t = 0, \ldots, T-1$, with $T = 5$ for PEAR and $T = 2$ for $\text{PEAR}_{\text{residue}}$, let $s_i^{(t)} = [h_i^{(t)}]_0$ denote the scalar part of the node state. The radial MLP gives the weights of the Clebsch–Gordan product on each edge,

$$w_{ij}^{(t)} \ =\ \text{MLP}_R\big(\,[\,b(r_{ij});\ \tau_{ij};\ s_i^{(t)};\ s_j^{(t)}\,]\,\big), \tag{14}$$

with two hidden layers of width 64 and one output per tensor-product weight, 384 at $\ell_{\max}$=0, without an output bias (Appendix G) The activations are SiLU and the final layer is initialized uniformly on $[-10^{-3}, 10^{-3}]$, so the residual path dominates early in training. The node scalars entering Equation 14 are those of the current layer, $s^{(t)}$, recomputed at every t. The message and its aggregation are

$$m_{ij}^{(t)} \ =\ h_j^{(t)} \otimes_{\text{CG}}^{w_{ij}^{(t)}} Y\big(\hat{\mathbf{r}}_{ij}\big), \qquad a_i^{(t)} \ =\ \frac{1}{Z} \sum_{j \in N(i)} m_{ij}^{(t)}, \tag{15}$$

where $Y = (Y_\ell^m)_{\ell \le \ell_{\max}}$ are the real spherical harmonics, $\otimes_{\text{CG}}^{w}$ is the Clebsch–Gordan product in `uvu` mode with path weights w, and $N(i)$ contains the neighbours of i in the graph being encoded, so intermolecular edges enter only the bound copy The aggregation is a plain sum, $Z = 1$. The node update is

$$\tilde{h}_i^{(t+1)} \ =\ h_i^{(t)} \ +\ \sigma\big(\, W_U\, a_i^{(t)}\,\big) \tag{16}$$

where the aggregated message passes through an equivariant batch normalization, a linear map, and an $e3nn$ gate, and is then added to $h_i^{(t)}$; a linear skip replaces the identity when the input and output irreps differ. The self-interaction term is disabled in the reported models (`self_cg: none`)..

Edge states and feedback. In standard message passing, edge features are computed once, used as arguments to the message function (Gilmer et al., 2017), and discarded after aggregation. In PEAR each intermolecular edge instead has its own hidden state, updated alongside the nodes. It is initialized from its endpoints by a symmetric encoder,

$$e_{ij}^{(0)} \;=\; \phi_{\text{init}}\big(\, s_i^{(0)} + s_j^{(0)},\; |s_i^{(0)} - s_j^{(0)}|\,\big), \tag{17}$$

so that $e_{ij}^{(0)} = e_{ji}^{(0)}$. After each node update the edge state is updated residually from the new node scalars,

$$e_{ij}^{(t+1)} \;=\; e_{ij}^{(t)} \;+\; \phi_{\text{upd}}\big(\, e_{ij}^{(t)},\; \hat{s}_i + \hat{s}_j,\; |\hat{s}_i - \hat{s}_j|\,\big), \qquad \hat{s}_i = W_e\,[\tilde{h}_i^{(t+1)}]_0, \tag{18}$$

and added back to its endpoints, which we call edge feedback,

$$h_i^{(t+1)} \;=\; \tilde{h}_i^{(t+1)} \;+\; \sum_{j:\,(i,j)\in E_{\text{inter}}} W_{\text{fb}}\, e_{ij}^{(t+1)}. \tag{19}$$

ϕ_{init} and ϕ_{upd} are MLPs with one hidden layer and SiLU: ϕ_{init} maps 2×384 to 64 through a width-64 layer, and ϕ_{upd} maps 3×64 to 64 through a width-128 layer, weight-tied across the T layers. Node scalars are projected to the edge width by a linear bottleneck W_e before entering ϕ_{upd}. The edge state does not enter MLP_R or the message, so it reaches the nodes only through Equation 19.. The final state $e_{ij}^{(T)}$ therefore depends on the node states of every layer. Edge-to-node feedback has been used in other molecular architectures (Gasteiger et al., 2020; Schütt et al., 2021). A simpler alternative computes the edge features once from the final node scalars, $e_{ij} = \phi_{\text{init}}(s_i^{(T)} + s_j^{(T)},\, |s_i^{(T)} - s_j^{(T)}|)$, without persistent updating. PEAR uses the persistent variant.

The unbound copy. The same encoder, with the same parameters, is applied to G^{ub}. We write $z_i^b = h_i^{(T)}$ and z_i^{ub} for the final node states of the two copies, and $z_{ij}^b = e_{ij}^{(T)}$ for the final intermolecular edge states. G^{ub} passes no messages along the intermolecular edges, but it carries edge states over the same index pairs: z_{ij}^{ub} is initialized by Equation 17 from the unbound node scalars and updated by Equation 18 at every layer, with the same parameters, and the feedback of Equation 19 is applied to both copies. The two edge states therefore differ only through the node states that feed them.

Readouts and head. The prediction combines the pooled bound embedding with the four paired readouts of Equation 8,

$$s_{\text{bound}} = W_{\text{pool}} \textstyle\sum_i z_i^b, \tag{20}$$

$$s_{\text{atom}} = \textstyle\sum_i \Phi^\rho\big(\tilde{z}_i^b,\, \tilde{z}_i^{ub}\big), \tag{21}$$

$$s_{\text{edge}} = \textstyle\sum_{(i,j)\in E_{\text{inter}}} \big[\, \Delta_{ij}\,;\; \psi\big(z_{ij}^b,\, z_{ij}^{ub}\big)\,\big], \tag{22}$$

$$s_{\text{pocket}} = \big[\, \Phi^\rho\big(\tilde{Z}^b,\, \tilde{Z}^{ub}\big)\,;\; \Phi^\rho\big(\tilde{Z}_{\text{res}}^b,\, \tilde{Z}_{\text{res}}^{ub}\big)\,\big], \tag{23}$$

$$\hat{y} = g_\theta\big(\,[\, s_{\text{bound}}\,;\; s_{\text{atom}}\,;\; s_{\text{edge}}\,;\; s_{\text{pocket}}\,]\,\big), \tag{24}$$

where Φ^ρ is Equation 7, $\tilde{z}_i$ is the atom density of Equation 4 expanded to $\ell{=}1$, $\tilde{Z}$ and $\tilde{Z}_{\text{res}}$ are the pocket densities of Equation 9 expanded to $\ell{=}2$ with 16 channels, Δ_{ij} is defined above Equation 8, W_{pool} projects to 192 dimensions, and ψ is an MLP All pooling is an unnormalised sum. ψ is the `delta,mlp` comparison: a two-layer SiLU MLP of width 128 on $z_{ij}^b - z_{ij}^{ub}$, concatenated with a second such MLP of the pair. The head g_θ is two hidden layers of width 128 with SiLU and no dropout, ending in a scalar. The model is trained with the mean squared error on pK.

Algorithm 1 PEAR forward pass.

Require: complex graph G, unbound copy G^{ub}
for each copy $c \in \{b, ub\}$ **do**
  $h_i^{c,(0)} \leftarrow$ Eq. 13 for every node i
  $e_{ij}^{c,(0)} \leftarrow$ Eq. 17 for every $(i,j) \in E_{\text{inter}}$
end for
for $t = 0, \ldots, T-1$ **do**
  for each copy $c \in \{b, ub\}$ **do**
    $w_{ij}^{(t)}$, $m_{ij}^{(t)}$, $a_i^{(t)} \leftarrow$ Eqs. 14–15 ▷ $N(i)$ excludes E_{inter} when $c = ub$
    $\tilde{h}_i^{c,(t+1)} \leftarrow$ Eq. 16
    $e_{ij}^{c,(t+1)} \leftarrow$ Eq. 18
    $h_i^{c,(t+1)} \leftarrow$ Eq. 19
  end for
end for
$z_i^c \leftarrow h_i^{c,(T)}$ and $z_{ij}^c \leftarrow e_{ij}^{c,(T)}$ for $c \in \{b, ub\}$
s_{bound}, s_{atom}, s_{edge}, $s_{\text{pocket}} \leftarrow$ Eqs. 20–24
return $\hat{y} \leftarrow g_\theta([\, s_{\text{bound}}; s_{\text{atom}}; s_{\text{edge}}; s_{\text{pocket}} \,])$

E TRAINING SCHEDULES AND ARCHITECTURAL SPECIFICATIONS

Both models use the backbone and paired readouts of Section 3 at two scales, the atom model on per-atom pocket nodes and the residue model on one centroid node per residue. 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 (Yang et al., 2023), where the pocket comprises 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.

	PEAR	$\text{PEAR}_{\text{residue}}$
Architecture		
Message-passing layers	5	2
Scalar channels	384	320
Pooled projection	192	160
Backbone $\ell_{\max}$	0	0
Atom paired readout	$\otimes^{\rho}_{\text{CG}}$, atom density to ℓ=1	
Edge paired readout	Δ+MLP	
Pocket paired readouts	$\otimes^{\rho}_{\text{CG}}$, pocket density to ℓ=2, 16 channels	
Radial MLP	2 hidden layers of width 64	
Interaction cutoff	5 Å	
Pocket–pocket edges	covalent bonds	centroids within 10 Å
Node features	35-dim atom	residue mean of atom features, amino acid, B-factor (Table 7)
Parameters	1.68M	831K
Optimization		
Optimizer	AdamW	
Loss	MSE (pK)	
Batch size	64	
Peak / min LR	2×10^{-4} / 10^{-5}	10^{-4} / 10^{-5}
Warmup	5 epochs (linear)	
Weight decay	10^{-4}	
Gradient-norm clip	10.0	
Epochs	1200	800
Checkpoint	lowest validation RMSE	

Table 6: Architecture and optimization settings for PEAR and $\text{PEAR}_{\text{residue}}$. The atom density is that of Equation 4, expanded about each node up to the stated angular order ℓ. The pocket density is that of Equation 9, expanded about the ligand centroid. The radial MLP computes the weights of each message-passing product from the distance basis, the edge type and the scalar features of both endpoint nodes (Appendix G).

Feature	Encoding	Dims
Atom node		
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^3d, SP^3d^2	5
Aromatic	boolean	1
Hydrogen count	0–4	5
Edge		
Distance	smooth-finite RBF (0–5 Å)	14
Edge type	pocket–pocket / ligand–ligand / ligand–pocket, learned	16
Residue additions		
B-factor	z-scored per structure	1
Amino acid	learned embedding, 22 classes	8

Table 7: Input features. Atom nodes have a 35-dim descriptor. The residue model uses these mean-pooled over each residue's atoms, with a B-factor and an amino-acid embedding added. The 10-dim element one-hot is passed through a learned 16-dim embedding, and the remaining dimensions are used as scalars.

The atom model is trained for 1200 epochs. The residue model is shallower and smaller, with validation RMSE converging by 800 epochs, so we use that budget throughout. For the CleanSplit benchmark we also trained it to 1200 epochs to match the atom model, which slightly improved

CleanSplit accuracy, so Table 2 reports the 1200-epoch ensemble. The atom model retrained at 800 epochs has a 5-fold mean held-out R of 0.429, below the 800-epoch residue model at 0.451. At matched compute the residue model is higher by $+0.022$.

PEAR_{LBA} differs from PEAR in four settings and is otherwise identical to it. (i) The atom density $\tilde{z}_i$ of Equation 4 is expanded only to $\ell{=}0$, where PEAR expands it to $\ell{=}1$, so it keeps the radial term of each neighbour and drops the directional $\ell{=}1$ terms. (ii) The Clebsch–Gordan product in Φ^{ρ} of the atom paired readout (Equation 7) uses $\texttt{uvu}$ mode, where PEAR uses $\texttt{uuu}$ mode. In $\texttt{uuu}$ mode, output channel c couples channel c of both inputs, with one weight per channel. In $\texttt{uvu}$ 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) (Appendix G). (iii) The pocket paired readout over atoms, the first term of s_{pocket} in Equation 23, 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 W_{in} in Equation 13 directly, without the learned embedding E_{el}. 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.

F Evaluation protocols in detail

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 (Graber et al., 2025). 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.

ATOM3D LBA. The source data are the refined set of PDBbind v2019, filtered by Townshend et al. (2021) 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. (2025), results are the mean of three runs and are reported as RMSE and Pearson and Spearman correlation.

Held-out pocket clusters. The held-out pocket clusters are constructed as described by Kopko et al. (2025). Complexes of PDBbind v2020 are clustered with PLINDER (release 2024-06/v2) by pocket lDDT community at a threshold of 50. For each of the seven CASF-2016 targets (1NVQ, 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. (2025) 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. (2025).

G Parameter cost

A Clebsch–Gordan product in `e3nn` is parameterised by a mode that fixes how the two inputs' multiplicity channels are coupled to the output's, and the mode determines the parameter count. The three mode indices refer to input$_1$ (u), input$_2$ (v), and output (w). In `uvw` mode the weight tensor has shape $(\mathrm{mul}_u, \mathrm{mul}_v, \mathrm{mul}_w)$, giving $\mathcal{O}(n_c^3)$ parameters when the three multiplicities are of order n_c. This mode arises whenever the input and output multiplicities differ along a Clebsch–Gordan path. Coupling $16 \times 1^o \otimes 16 \times 1^o \to 64 \times 0^e$ then requires $16 \times 16 \times 64 = 16{,}384$ weights for that path. In `uvu` mode the weight tensor has shape $(\mathrm{mul}_u, \mathrm{mul}_v)$ and the output inherits the multiplicity of input$_1$, reducing the cost to $\mathcal{O}(n_c^2)$. This mode applies when the input and output multiplicities are equal, as in $64 \times 0^e \otimes 64 \times 0^e \to 64 \times 0^e$, which requires $64 \times 64 = 4{,}096$ weights. In `uuu` mode the product is diagonal, coupling channel c of the two inputs to channel c of the output, at $\mathcal{O}(n_c)$.

Our message-passing product couples the node features with the spherical harmonics of the bond direction. The spherical harmonics have multiplicity one, one copy of each order, and mixing across the node channels is done by the linear layers between message-passing steps. We use `uvu` mode for this product, with one weight per channel along each path. `uvw` mode would instead use a separate weight for each input and output channel pair. The paired readouts combine the bound and unbound copies, which come from the same backbone and share a channel layout, so we compute each paired readout channel by channel, in `uuu` mode. Because the backbone runs at $\ell_{\max} = 0$, each message-passing product outputs only scalars, and there are few Clebsch–Gordan paths.

The total number of tensor-product weights is the sum over Clebsch–Gordan paths of their individual weight counts. The radial MLP computes these weights for each edge from the edge features, so its output layer has one unit for each weight, 384 for the scalar backbone, without a bias. Its input concatenates the edge features (a 14-dimensional distance basis and a 16-dimensional edge-type embedding) with the scalar ($\ell = 0$) features of the two endpoint nodes. This increases the input from 30 to $30 + 2 \times 384 = 798$, so the first linear layer, at 798×64, contains most of each layer's parameters. With two hidden layers of width 64, each radial MLP has about 80K parameters, and the five message-passing layers together account for close to a quarter of the model.

H Preliminary experiment on a basic MPNN

Table 8 reports in full the results summarised in the upper block of Table 3, adding Pearson R to the RMSE shown there. The encoder is a basic equivariant MPNN, which is not the backbone of PEAR. It has the message passing of Appendix D but without edge states, and therefore without edge updates or edge feedback. It is applied to the complex graph and to the same graph with intermolecular edges removed, and the pooled embeddings are combined by an MLP.

Layers	Copies	$\ell_{\max}$	Test RMSE	Pearson R
2	1	0	1.441 ± 0.007	0.752
2	1	1	1.453 ± 0.016	0.746
2	1	2	1.405 ± 0.023	0.767
2	2	0	1.460 ± 0.012	0.741
2	2	1	1.443 ± 0.035	0.754
2	2	2	1.489 ± 0.021	0.733
5	1	0	1.386 ± 0.034	0.786
5	1	1	1.389 ± 0.057	0.783
5	1	2	1.392 ± 0.038	0.787
5	2	0	1.423 ± 0.038	0.768
5	2	1	1.399 ± 0.027	0.770
5	2	2	1.384 ± 0.024	0.784

Table 8: Rows compare one-state and two-state message passing on CleanSplit across network depth and $\ell_{\max}$. All rows are over five seeds.

I RESIDUE-SCALE READOUT ACCURACY ON CLEANSPLIT

The residue readout variants evaluated on the held-out clusters (Table 4) have nearly the same CleanSplit accuracy (Table 9). Raising the $\ell_{\max}$ of the $\otimes^{\rho}_{\mathrm{CG}}$ density expansion affects the CASF-2016 ensemble RMSE by less than one fold standard deviation. As a minimal ablation of the readout itself, we tested a Δ_{sym} variant and found its accuracy to be similar.

Residue readout	CASF-2016 RMSE	r
$\otimes^{\rho}_{\mathrm{CG}}$ (ℓ=0)	1.308	0.813
$\otimes^{\rho}_{\mathrm{CG}}$ (ℓ=1)	1.318	0.808
$\otimes^{\rho}_{\mathrm{CG}}$ (ℓ=2)	1.308	0.810
Δ_{sym} (ℓ=0)	1.304	0.811

Table 9: Residue-scale readout variants on CleanSplit (CASF-2016, five-fold CV ensembles, 800 epochs). The four variants fall within 0.014 RMSE.

J FULL OUT-OF-DISTRIBUTION READOUT GRIDS

Method	1NVQ	1SQA	2P15	2VW5	3DD0	3F3E	3O9I	Avg.	Min.
$\text{PEAR}_{\text{backbone}}$	0.523	0.640	0.410	0.380	0.352	0.219	0.518	0.434	0.219
$\text{PEAR}_{\text{residue,backbone}}$	0.539	0.695	0.395	0.453	0.347	0.218	0.359	0.430	0.218
PEAR, node paired readout varied									
$\otimes^{\rho}_{\text{CG}}$ (ℓ=0)	0.506	0.669	0.454	0.402	0.457	0.197	0.483	0.452	0.197
$\otimes^{\rho}_{\text{CG}}$ (ℓ=1)	0.507	0.667	0.435	0.393	0.312	0.214	0.384	0.416	0.214
$\otimes^{\rho}_{\text{CG}}$ (ℓ=2)	0.510	0.657	0.450	0.373	0.432	0.194	0.378	0.427	0.194
$\otimes_{\text{CG}}$ (ℓ=1)	0.504	0.669	0.444	0.374	0.436	0.178	0.328	0.419	0.178
Δ_{sym} (ℓ=0)	0.497	0.653	0.438	0.369	0.416	0.212	0.447	0.433	0.212
Δ_{sym} (ℓ=1)	0.508	0.660	0.410	0.392	0.469	0.212	0.458	0.444	0.212
Δ_{sym} (ℓ=2)	0.500	0.643	0.401	0.372	0.489	0.205	0.431	0.434	0.205
Δ_{asym} (ℓ=2)	0.475	0.648	0.410	0.399	0.464	0.216	0.353	0.424	0.216
$\text{PEAR}_{\text{residue}}$, node paired readout varied									
$\otimes^{\rho}_{\text{CG}}$ (ℓ=0)	0.517	0.703	0.413	0.493	0.407	0.207	0.414	0.451	0.207
$\otimes^{\rho}_{\text{CG}}$ (ℓ=1)	0.534	0.709	0.386	0.478	0.406	0.235	0.382	0.447	0.235
$\otimes^{\rho}_{\text{CG}}$ (ℓ=2)	0.521	0.693	0.476	0.470	0.425	0.242	0.404	0.461	0.242
Δ_{sym} (ℓ=0)	0.521	0.715	0.398	0.470	0.432	0.263	0.408	0.458	0.263
Joint 10-fold ensemble									
PEAR + $\text{PEAR}_{\text{residue}}$, ℓ=2	0.552	0.709	0.465	0.451	0.464	0.242	0.475	0.479	0.242

Table 10: Expanded grid for the Pre-FT columns of Table 4. The $\text{PEAR}_{\text{residue}}$ variants average slightly higher than the PEAR variants before finetuning. The ($\otimes_{\text{CG}}$, ℓ=1) row takes the Clebsch–Gordan product of the bound and unbound node embeddings directly, without the atom density of Equation 4, matched to the $\otimes^{\rho}_{\text{CG}}$ (ℓ=0) model on CleanSplit RMSE.

Method	1NVQ	1SQA	2P15	2VW5	3DD0	3F3E	3O9I	Avg.	Min.
$\text{PEAR}_{\text{backbone}}$	0.548	0.660	0.455	0.593	0.541	0.305	0.523	0.518	0.305
$\text{PEAR}_{\text{residue,backbone}}$	0.561	0.730	0.551	0.602	0.511	0.379	0.552	0.555	0.379
PEAR, node paired readout varied									
$\otimes^{\rho}_{\text{CG}}$ (ℓ=0)	0.526	0.687	0.510	0.603	0.557	0.340	0.562	0.541	0.340
$\otimes^{\rho}_{\text{CG}}$ (ℓ=1)	0.546	0.689	0.498	0.597	0.569	0.339	0.557	0.542	0.339
$\otimes^{\rho}_{\text{CG}}$ (ℓ=2)	0.542	0.677	0.518	0.619	0.537	0.320	0.573	0.541	0.320
$\otimes_{\text{CG}}$ (ℓ=1)	0.538	0.675	0.519	0.618	0.563	0.306	0.556	0.539	0.306
Δ_{sym} (ℓ=0)	0.523	0.688	0.485	0.560	0.543	0.328	0.567	0.528	0.328
Δ_{sym} (ℓ=1)	0.541	0.695	0.488	0.610	0.573	0.316	0.561	0.541	0.316
Δ_{sym} (ℓ=2)	0.547	0.696	0.498	0.608	0.571	0.324	0.529	0.539	0.324
Δ_{asym} (ℓ=2)	0.514	0.678	0.487	0.614	0.568	0.281	0.595	0.534	0.281
$\text{PEAR}_{\text{residue}}$, node paired readout varied									
$\otimes^{\rho}_{\text{CG}}$ (ℓ=0)	0.560	0.705	0.535	0.645	0.569	0.363	0.611	0.570	0.363
$\otimes^{\rho}_{\text{CG}}$ (ℓ=1)	0.573	0.739	0.537	0.634	0.560	0.396	0.587	0.575	0.396
$\otimes^{\rho}_{\text{CG}}$ (ℓ=2)	0.563	0.732	0.572	0.646	0.558	0.367	0.624	0.580	0.367
Δ_{sym} (ℓ=0)	0.570	0.727	0.522	0.646	0.574	0.375	0.605	0.574	0.375
Joint 10-fold ensembles									
PEAR + $\text{PEAR}_{\text{residue}}$	0.582	0.724	0.555	0.680	0.592	0.385	0.626	0.592	0.385
PEAR + $\text{PEAR}_{\text{residue}}$, ℓ=1	0.590	0.740	0.559	0.671	0.589	0.400	0.615	0.595	0.400
PEAR + $\text{PEAR}_{\text{residue}}$, ℓ=2	0.584	0.736	0.577	0.681	0.591	0.388	0.634	0.599	0.388
PEAR + $\text{PEAR}_{\text{residue}}$, Δ_{sym}	0.590	0.733	0.548	0.679	0.596	0.395	0.621	0.595	0.395
PEAR + PEAR, $\otimes_{\text{CG}}$ (ℓ=1)	0.547	0.690	0.526	0.620	0.566	0.330	0.570	0.550	0.330

Table 11: Expanded grid for the FT-25 columns of Table 4. After finetuning, the four $\text{PEAR}_{\text{residue}}$ variants span 0.570 to 0.580 in average r, and among the node paired readouts of PEAR, Δ_{sym} at ℓ=0 has the lowest average r, 0.528.

K QUARTILE-STRATIFIED ANALYSIS ON CASF-2016

We bin CASF-2016 into quartiles along six per-complex features: ligand heavy-atom count, pocket heavy-atom count, T1 interface size, inter-edge count, ligand/pocket atom ratio, and apo–holo backbone Cα RMSD from ApoBind (Aggarwal et al., 2021). 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 $\text{PEAR}_{\text{residue}}$ models and their unweighted mean ensemble.

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.

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 12: The ensemble has the lowest RMSE in 20 of 24 quartile bins. Of the four exceptions, the single atom model is lowest in three small/rigid/loose bins (ligand Q1, lig/pkt Q1, apo–holo Q1, all within 0.053 RMSE), and the single residue model in apo–holo Q3. Test RMSE by quartile for single-fold PEAR and $\text{PEAR}_{\text{residue}}$ models and their unweighted mean ensemble across six stratification axes.

L ATOM3D LBA WITH STANDARD DEVIATIONS

Table 13 repeats Table 5 with the standard deviation over three runs. Baseline values are quoted from Lim et al. (2025). Uni-Mol 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
pretrained							
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
$\text{PEAR}_{\text{backbone}}$	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 13: ATOM3D LBA, mean ± standard deviation over three runs. Best per column in **bold**, second-best <u>underlined</u>, by mean.