EQUISOLV: O(3)-EQUIVARIANT NEURAL NETWORKS FOR PREDICTING MOLECULAR SOLVATION ENERGY FIELDS

Anonymous authors
Paper under double-blind review

ABSTRACT

Drug design is an arduous process that depends on the discovery of functional protein-molecule interactions, whose energetic favourability depends on displacing solvent from the binding pocket. In practice, Grid Inhomogeneous Solvation Theory (GIST) is a commonly used method that provides accurate, computationally costly solvation thermodynamics through molecular dynamics (MD) simulations, rendering it infeasible for large-scale drug screenings. We propose an O(3)-equivariant graph neural network architecture, named EquiSolv, built on Euclidean neural network (e3nn) primitives that predicts GIST-derived scaled enthalpy (eGIST) fields. The field is defined at any coordinate and is evaluated on the 1.0 Å MD grid. On peptides, EquiSolv fuses ESM-C embeddings with geometric graph features; on DUD-E proteins EquiSolv omits ESM-C, which leaves MSE essentially unchanged in ablations. We predict per-atom parametric functions constructed as linear combinations, with learnable coefficients, of products of radial basis functions and spherical harmonics. We bound the approximation error of the population-optimal network, assuming a smooth interpolant. The bound does not cover finite-sample error. Of 1100 DBAASP peptide systems we train on 880 and test on 110 peptides; of the DUD-E proteins we train on 69 and test on 10 proteins. On the held-out peptides, we achieve an RMSE of 0.176 ± 0.026 kcal/mol/Å^3.

1 INTRODUCTION

Molecules exhibit 3-dimensional structure and electrostatic properties arising from their constituent atoms. These atoms collectively generate a continuous electrostatic potential field that affects the local free energy of interaction with other molecules. Solvent molecules such as water interact with this energy field, increasing the energy required for interaction since the solvent needs to be displaced. This can be relevant in interactions with proteins, whose function or inhibition may depend on the efficacy of such interactions (Gilson & Zhou, 2007). Thermodynamic properties of solvents around proteins are essential to understanding and modelling inter-protein interactions as well as protein-ligand interactions. Spatial distribution and energetics of solvent can greatly affect the formation and stability of molecular complexes and are an important subject for drug discovery and computational biology. Binding energy can even be used as a strong indicator for the potential binding location of a ligand to a protein. Grid Inhomogeneous Solvation Theory (GIST) (Nguyen et al., 2012) is a solvation thermodynamic analysis framework that functions through discretisation of space around proteins into a grid, then computing energy contributions at each grid point through molecular dynamics (MD) simulations. Addition of GIST information demonstrably benefits prediction of ligand binding, and in that setting scaled enthalpy (eGIST $= E_{SW} + 2E_{WW}$) outperformed grids that include the full GIST free-energy terms (Balius et al., 2017).

MD simulations are however computationally intense; on high-performance computing systems they can cost a full day of processing for an average protein (Phillips et al., 2020). For example, in this work we generate MD data for 50 ns of simulated water on a protein after over a full day of computation. While this may appear feasible on a per-protein basis, the effective cost is typically the protein conformation rather than the protein itself. Each GIST/MD simulation samples solvent dynamics around a specific, restrained protein conformation. In drug discovery, proteins are represented by an ensemble of conformations, often exceeding 100, to capture intrinsic flexibility.

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.

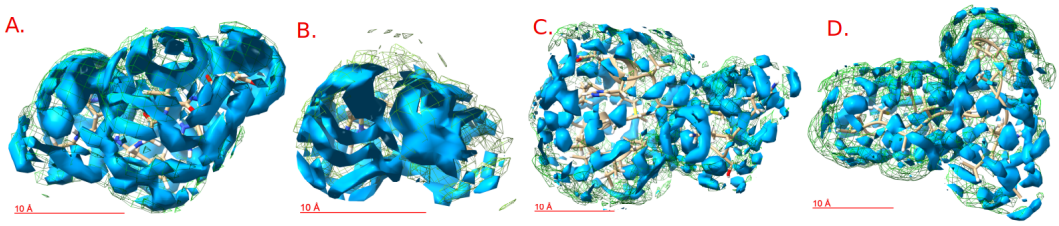


Figure 1: Visualisation of molecules: the two worst and two best sets of EquiSolv-predicted grids shown as a green mesh over MD-simulation water as a blue surface. Visible points are chosen by thresholding a p_voxel value ranging from 0 to 1 arbitrarily. Examples are selected by RMSE: **A**. `15534_hsp`, 0.162064; **B**. `17024_hsp`, 0.147764; **C**. `1506_hse`, 0.293269; **D**. `2219`, 0.296595.

This distinction is critical because ligand binding frequently follows a conformational selection mechanism, whereby a ligand preferentially binds to a specific protein conformation (out of myriad possible conformations) that maximises favourable interactions. For novel ligands, the relevant binding-competent conformation is generally unknown *a priori*, motivating ensemble docking against a large set of protein conformations. Under these conditions, the computational cost scales with the number of conformations, and performing explicit-solvent MD simulations (e.g., 50 ns each) over large conformational ensembles rapidly becomes prohibitively expensive, even with modern GPU resources.

Geometric deep learning approaches accelerate obtaining these solvation maps by prediction rather than simulation of solvent energetics. Graph neural networks (GNNs) can capture interactions between atoms while preserving rotational and translational symmetries through equivariant design. In order to improve on current GIST applications, we present EquiSolv, a novel learning framework for continuous solvent energy field reconstruction preserving $O(3)$-equivariance from molecule inputs.

EquiSolv builds each atom's initial embedding from structure-level features on a geometric graph annotated by $Graphein$ (Jamasb et al., 2022). When ESM-C is enabled, per-residue sequence embeddings are concatenated with these graph features (Evolutionary Scale Team, 2024). From these embeddings the network predicts coefficients of a radial-spherical basis, evaluating that basis around an atom gives its local field expansion. A query point's prediction is then the distance-weighted sum of the local expansions of nearby atoms. To summarise, our work contributes (1) an $O(3)$-equivariant learning framework predicting continuous scaled enthalpy energy fields from molecular graphs (illustrated in Figure 1), the equivariance property enhances data efficiency and the continuous output broadens the use case. We (2) provide a theoretical analysis of EquiSolv's approximation error. We (3) conduct experiments on different datasets curated by us for this purpose. We will (4) release these datasets upon publication. A common benchmark dataset like the one we contribute is required for this budding literature to move forward. Currently, the targets and datasets of the published models are different, which makes it very difficult to compare models.

2 Related Work

Molecular Dynamics. Explicit solvent MD combined with inhomogeneous solvation theory (IST) (Lazaridis, 1998) forms the basis of methods like GIST (Ramsey et al., 2016) which decompose local water structure into entropic and enthalpic contributions on a 3D grid around molecules in solvent. Similar IST-based tools like WATsite (Hu & Lill, 2014) and WaterMap (Abel et al., 2008) cluster water positions to hydration sites and assign an energy to each site. These can be used to rationalise the relationships between structure and activity, and to identify high-energy waters that can easily be displaced by ligands such as a drug, which helps to find binding pockets. While such MD methods are highly accurate and yield detailed chemically interpretable solvation maps, they are relatively slow and compute-heavy, which limits their usefulness for large-scale drug discovery operations that involve the screening of thousands of candidates. To deal with this, various approximate models have been developed that obtain average solvent information rather than sampling trajectories over time. 3D-RISM (Kovalenko & Hirata, 1998) solves Ornstein-Zernike equations on a grid and returns a

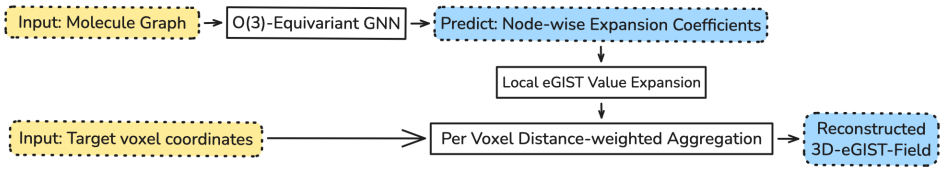


Figure 2: An overview of our framework. An input molecular graph constructed as in 3.1 is passed into a GNN with O(3)-equivariant backbone, which outputs per-node local expansion coefficients, described in the section 3.3. These coefficients are used to build local expansion functions, detailed in 3.4 that are further aggregated to reconstruct the continuous 3D field, explained in 3.5.

solvent number density around a fixed molecule, giving a more limited scope of information and a reduced accuracy in exchange for significantly lower computational cost. We do not report RMSE against 3D-RISM: its density output is not the GIST energy density, and putting the two on one scale would require the development of a non-standard mapping. Methods such as Goodford's GRID (Goodford, 1985), WaterScore (Garcia-Sosa et al., 2003), AcquaAlta (Rossato et al., 2011), SZMAP (Bayden et al., 2015) and other grid-based probe/scoring approaches evaluate interaction energy or heuristic stability of a single water molecule at a time in locations around the protein, resulting in energy or score fields at a fractional cost compared to MD simulations. Such methods however, lack the full decomposition that IST-based approaches use to give solute–water and water–water information and do not explicitly incorporate the interaction effects of multiple waters locally.

Deep Learning. More recent methods involve the exploration of deep learning as a tool for physically derived hydration maps. Early work by Ghanbarpour et al. (2020a) introduced a deep learning model that predicts hydration occupancy and desolvation energy around binding pockets derived from WATsite data. This work operates on binding pocket centered 3D grids and focuses on discrete hydration sites that may be relevant to ligand binding. In a related study, Ghanbarpour et al. (2020b) developed a method for on-the-fly prediction of protein hydration densities and energies using deep learning. Concurrently, Mahmoud et al. (2020) explored the multiple roles of hydration for accurate protein-ligand binding prediction, demonstrating how DL can model complex hydration-mediated interactions. Building on these foundations, Park & Seok (2022) developed GalaxyWater-CNN, a 3D convolutional neural network that predicts water positions on protein structures by generating water score maps. This method operates on fixed grids around molecules and outputs probability scores for water presence. In the same year, Kawama et al. (2022) introduced "gr Predictor", a 3D CNN that reproduces 3D-RISM oxygen density fields around proteins at a significantly reduced runtime compared to traditional 3D-RISM calculations. More recently, Zamanos et al. (2024) presented HydraProt, a deep learning tool for fast and accurate prediction of water molecule positions for protein structures, which typically trains on crystallographic waters rather than MD simulation data and predicts discrete hydration sites. Concurrently, Fukushima & Yoshidome (2026) introduce DeepGist, a 3D U-Net-based model that predicts a hydration grid from a protein input, achieving high correlation with GIST MD simulations. In contrast, EquiSolv predicts O(3)-equivariant atom-centred expansion coefficients, allowing the reconstructed field to be evaluated directly at arbitrary coordinates within the solvation shell.

In summation, these works show that neural networks can approximate complex hydration patterns and site-resolved thermodynamics. They differ from our approach in three ways. First, most methods create either 3D-RISM density fields or WATsite/IST site thermodynamics, whereas we learn to reconstruct a continuous GIST energy-density field derived directly from MD. Second, while previous deep learning methods typically target binding pockets or discrete hydration sites, we create a full solvation shell as a continuous scalar field around the protein. Third, existing models operate in voxel space or as site predictors, while EquiSolv predicts equivariant per-atom coefficients that can be used with a defined basis expansion to reconstruct GIST at any coordinate in space.

3 METHODS

We present a framework composed of three main components (Figure 2) that together enable learning and prediction of the properties of solvent around molecules. First, a graph construction module

that converts molecule structures into geometric representations that capture relationships between atoms. Second, a sequence of $e3nn$ (Geiger & Smidt, 2022) layers processes the node information of the molecular graph, and produces per-atom coefficients for the basis functions. Finally, the atom-specific local expansions, each being the linear combination of the basis functions using the predicted coefficients, are aggregated to evaluate a voxel's eGIST value.

Appendix A introduces notions from group and representation theory that are used in this work.

Symmetry. The solute is placed in a fixed frame, so translations are removed by centring and are not a symmetry we enforce. Rotations and reflections of that frame must rotate the scalar field with the molecule, which is $O(3)$ equivariance rather than $SE(3)$ or $E(3)$. The centre-of-mass term in Section 3.3 keeps that action exact, and the radial-spherical basis is the usual local expansion for such a field (Brockherde et al., 2017; Thomas et al., 2018).

Notation. Radial basis function (RBF) are denoted by $R(\cdot)$ and $Y_\ell^m(\cdot)$ denotes spherical harmonics of degree ℓ of order $m \in \{-\ell, \dots, \ell\}$. Let r_v be a voxel's 3D coordinates and V be the set of voxels of a given grid. The notation $\|\cdot\|$ denotes the Euclidean norm if not otherwise specified.

3.1 Graph Construction from Molecule Structures

We model the molecular graph G at the atomic level, where nodes correspond to atoms. Each atom a is characterised by its coordinates $r_a \in \mathbb{R}^3$ and a feature vector $h_a \in \mathbb{R}^k$. The neighbourhood of an atom a is defined based on spatial distance, namely, the set of all atoms within distance d $\mathcal{N}_a := \{b \in G \mid \|r_a - r_b\| \le d\}$. Using the $Graphein$ library metadata functions, molecular graphs are built using VDW-distance based edges and the nodes and edges are then enriched with features listed in Appendix B (Jamasb et al., 2022). Edge features are embedded first via learnable MLP transformations, then node features are embedded to the same dimension. Furthermore, we enrich our graphs with ESM-C embeddings (Evolutionary Scale Team, 2024). Molecular sequences are tokenised using the ESM-C tokeniser, a forward pass through the ESM-C model then generates per-residue token embeddings which are mapped to atoms via residue indices. For atom cases where tokens are unavailable, we use sequence-level mean embeddings. During training, ESM-C embeddings h_{ESM} are concatenated with the embedded graph features, f_{graph}: $h_a^{(0)} = [\text{MLP}(f_{\text{graph}}); h_{\text{ESM}}]$.

3.2 Per-Point Energy Densities

For each molecule, GIST grids hold coordinate information within the same space as the molecule they have been generated on along with various information including the solute-water and water-water energy densities, E_{SW} and E_{WW}, respectively. These raw GIST data values were used to compute the per-grid-point energy densities (eGIST) used for training using the formula $eGIST = E_{SW} + 2E_{WW}$ (Egger-Hoerschinger et al., 2025), the GIST-derived scaled enthalpy. Prior to training, eGIST values are transformed to log-space for numeric stability using the following formula,

$$\text{eGIST}_{\text{logspace}} = \text{sign}(\text{eGIST}_{\text{raw}}) \ln(1 + |\text{eGIST}_{\text{raw}}|).$$

Inference predictions are reverse-transformed to real-space,

$$\text{eGIST}_{\text{pred}} = \text{sign}(\text{eGIST}_{\text{logspace}})(e^{|\text{eGIST}_{\text{logspace}}|} - 1).$$

3.3 Layers: Feature Update

For a molecular graph, node features are updated following the core idea of the Tensor Field Network (TFN) (Thomas et al., 2018) with minor modifications. Appendix A.3 provides a detailed overview fo the used models and below we mainly introduce the minor modifications we propose. At layer i,

$$h_a^{(i+1)} = \gamma s(h_a^{(i)}) + (1-\gamma) \sum_{b \in \mathcal{N}_a} Y_{\ell_i}\left(\frac{r_{ab}}{\|r_{ab}\|}\right) \otimes_\Theta h_b^{(i)}, \tag{1}$$

where the hyperparameter γ is a weight factor in $[0, 1]$. In this work, we replace linear channel mixing in the update of each node a with a global positional awareness mechanism:

$$s(h_a^{(i)}) := Y_{\ell_i}\left(\frac{r_{aC}}{\|r_{aC}\|}\right) \otimes_\Theta h_a^{(i)},$$

where $r_{aC} := r_a - r_{CoM}$ denotes the displacement from the protein's centre of mass to node a. This term encodes each atom's global orientation within the protein while retaining exact $O(3)$-equivariance.

Gated Convolution Network. The update in Eq. (1) is linear. A gated convolution then applies the $e3nn$ gate, an $O(3)$-equivariant non-linearity (Geiger & Smidt, 2022), to that update.

$SE(3)$-Transformer. Besides the weight factor γ, the attention logits are scaled by $1/\sqrt{q}$ before the softmax, with q the dimension of k_{ab} (Eq. 2):

$$\alpha_{ab} = \frac{\exp\big((q_a \otimes_{\text{dot}} k_{ab})/\sqrt{q}\big)}{\sum_{b' \in \mathcal{N}_a} \exp\big((q_a \otimes_{\text{dot}} k_{ab'})/\sqrt{q}\big)} \,.$$

3.4 Local Expansion

An intuitive and direct strategy to address the limitations of GIST is to devise a predictive model that takes in a molecule G and outputs the scalar quantity of interest on the solvent-accessible surface. Rather than predicting values on a mesh grid, a widely adopted approach in quantum physics is to approximate the unknown (and potentially intractable) target functional as a linear combination of simpler basis functions. In this setting, similarly to the TFN's convolution kernels, the basis functions are typically constructed as the product of a radial basis function and a spherical harmonic (Brockherde et al., 2017; Rackers et al., 2022).

A direct application of this strategy to our setting must account for the locality of the structured solvation signal, since that signal decays as the distance between a voxel and an atom increases. We therefore model the value at each voxel as the combined contribution of atom-specific functions for nearby atoms, rather than one global function on G. We do not claim that electrostatics are short-ranged: our MD labels use particle-mesh Ewald (Appendix D). The message passing steps in our architecture allow our representations to capture interactions with further away atoms and ablations on our generated data show that at 10 Å, an atom's average contribution is about 4%. So, we opt for the more memory efficient option of not explicitly modelling extremely long-range effects. Distance- and charge-stratified errors are in Appendix N.

Following this idea, we approximate the field using a fixed family of basis functions. Analogous to how polynomials as a set of basis functions can be linearly combined to approximate a smooth function, we establish a predetermined family of basis functions $\mathcal{B}$ structured as follows,

$$\mathcal{B} := \big\{ R_k(\|r\|)\, Y_\ell^m\big(\tfrac{r}{\|r\|}\big) \;\big|\; \ell = 0, \dots, L_{\max};\; k = 1, \dots, M_\ell;\;\; m = -\ell, \dots, \ell \big\}.$$

The number of basis functions is decided by the final irreps' highest degree $L_{\max}$ and the multiplicity per degree M_ℓ. We choose the radial basis function $R_k : \mathbb{R}^+ \to \mathbb{R}$ to be a Gaussian shell radial function parametrised by α_k and β_k: $R_k(\|r\|) = \exp\big(-\alpha_k(\|r\| - \beta_k)^2\big)$, where r is the vector pointing from an atom a to a voxel v. The target local field is then approximated as a linear combination of these basis functions, called local expansions:

$$\hat{f}_a(r) = \sum_{\ell=0}^{L_{\max}} \sum_{k=1}^{M_\ell} \sum_{m=-\ell}^{\ell} c_{\ell,k,m}^{(a)} R_k(\|r\|) Y_\ell^m \left(\frac{r}{\|r\|}\right).$$

EquiSolv's task is therefore to predict the atom-specific coefficients $c_{\ell,k,m}^{(a)}$, for all atoms $a \in G$.

3.5 Aggregation and Loss

We now describe how multiple atom-specific contributions are aggregated into a single value associated with a given voxel v. To this end, we employ a weighted and normalised aggregation scheme. Conceptually, this corresponds to a weighted average of the atom-specific values $\hat{f}_a(r_v - r_a)$, where the weights are determined by a Gaussian shell radial function parametrised by α and β. These parameters control the centre and spread of the Gaussian, thereby modulating the influence of each atom according to its relative distance to the voxel. Hence, the aggregated value at voxel v equals

$$\hat{f}(r_v) = \sum_{a \in \mathcal{N}_v} \frac{R_{\alpha,\beta}(\|r_v - r_a\|)}{\sum_{a' \in \mathcal{N}_v} R_{\alpha,\beta}(\|r_v - r_{a'}\|)} \hat{f}_a(r_v - r_a),$$

where $\mathcal{N}_v := \{a \in G \mid \|r_a - r_v\| < d\}$ is the set of atom neighbours of a voxel v within some distance d and $R_{\alpha,\beta}(\|r\|) = \exp\big(-\alpha(\|r\| - \beta)^2\big)$.

The training objective is a weighted mean squared error over the effective set of voxels $V := \bigcup_{a\in G}\{\, v \mid \|r_v - r_a\| < d\,\}$, with $\mathcal{T}[v]$ the log-space eGIST value at voxel v. The bulk threshold τ and the non-bulk weight w_{nb} are hyperparameters of this loss. The reported model uses $\tau = 0.5$ and $w_{\mathrm{nb}} = 100$; ablations that remove the weighting use ordinary MSE ($w_v = 1$ for all v). Voxels with $|\mathcal{T}[v]| \le \tau$ receive weight 1; voxels with $|\mathcal{T}[v]| > \tau$ receive weight w_{nb}. The weights are then rescaled to sum to $|V|$:

$$\tilde{w}_v = \frac{w_v}{\sum_{u\in V} w_u}|V|, \qquad L = \frac{1}{|V|}\sum_{v\in V}\tilde{w}_v\,(\hat{f}(r_v) - \mathcal{T}[v])^2.$$

3.6 Time and Memory Complexity

Each voxel's value is computed based on the local expansion function of atoms within a fixed distance. We denote the average number of these quantities by N_a and the total number of voxels by N_v. Thus the spatial aggregation step has a time complexity of $\mathcal{O}(N_a N_v)$, i.e., $\mathcal{O}(N_v)$. We now also provide the time and memory complexity of a layer of our proposed EquiSolv. Let N, E, B denote respectively the number of nodes, edges of the molecule graph and the number of radial embeddings. Let P be the number of paths in the tensor product, then $P = \mathcal{O}(m^2L^2)$, where L, m are the maximum degree and multiplicity of the irreps. Finally, denote the input and output dimensions by d_1 and d_2. Then, a layer of our model has a memory complexity of $O(N(d_1 + d_2) + E(B + (L+1)^2) + P)$, and time complexity of $\mathcal{O}(Em^2L^6)$, dominated by tensor product operations (Passaro & Zitnick, 2023). In practice, we set $L \le 2$, which keeps the computational cost of our model realistic.

4 Error Analysis

We now provide an asymptotic error analysis of our method. The theoretical $f := E_{SW} + 2E_{WW}$ admits no known closed-form expression. EquiSolv aims to learn a parametric function with a given expansion setup characterised by the highest degree of spherical harmonics $L_{\max}$ and the multiplicities $\{M_\ell \mid \ell = 0, ..., L_{\max}\}$ of each irrep degree. The population-optimal approximator, denoted by $\hat{f}$, has coefficients given by the minimiser of the population risk over the described network class. The following proposition estimates the asymptotic behaviour of $\|f - \hat{f}\|_{L^2}$, the error between the unknown ground-truth field and the network-realisable optimal model.

Proposition 4.1. *Assume the grid interpolant is C^∞. Let d be the final aggregation distance described in the Section 3.5, ν be the maximum local density of the input molecule, i.e., $\nu = \max_{v\in V}|\mathcal{N}_v(d)|$, and $M_{\min} = \min\{M_\ell \mid \ell = 1, \ldots, L_{\max}\}$. The population-optimal model $\hat{f}$ composed of T layers of tensor field networks as in Thomas et al. (2018) and Fuchs et al. (2020) can approximate the unknown ground-truth IST function f. The L_2 error behaves asymptotically as*

$$\epsilon_{GIST} + \mathcal{O}\left(\nu^{1/2}(L_{\max}^{-s} + M_{\min}^{-s}) + M_{\min}^{-1/2}T^{-s}\right), \forall s \in \mathbb{N}.$$

Note that finite-sample error is not included in Proposition 4.1. We now give a sketch proof, see the Appendix C for the complete proof. The first source of error ϵ_{GIST} comes from the simulation and/or approximation by GIST, which is independent of our proposed method. We denote an MD simulation of f on a given grid as $\mathcal{T}$, since the grid has finite size and has a finite number $|V|$ of voxels, then by Whitney's extension theorem, there always exists an interpolant $\bar{f}$ such that i) $\forall v \in V,\ \mathcal{T}[v] = \bar{f}(v)$ and ii) $\bar{f} \in C^s, 1 \le s \le \infty$. Our proposed framework thus allows in theory an ideal approximator $\tilde{f}$, whose coefficients are unknown and optimal, which at best approximates the interpolant $\bar{f}$ with super-algebraically decaying error due to truncation of infinite series expansion. EquiSolv's training target is to predict those optimal coefficients of $\tilde{f}$, which brings an additional component: learning loss. In this work, we are interested in the population-optimal model, i.e., the ideal version of EquiSolv trained to global optimality over the population. We deliberately do not elaborate on the statistical learning error analysis as this topic on its own can be a rich study and requires detailed architectural analysis on the TFN-type models, thus is out of scope of this work. To summarise, the

error is decomposed into three sources:

$$\|f - \hat{f}\|_{L^2} = \|f - \bar{f} + \bar{f} - \tilde{f} + \tilde{f} - \hat{f}\|_{L^2} \leq \underbrace{\|f - \bar{f}\|_{L^2}}_{\epsilon_{GIST}\text{: Numerical error}} + \underbrace{\|\bar{f} - \tilde{f}\|_{L^2}}_{E_1\text{: Truncation error}} + \underbrace{\|\tilde{f} - \hat{f}\|_{L^2}}_{E_2\text{: Learning loss}}.$$

5 EXPERIMENTS

We prepare our dataset through MD simulations, evaluate EquiSolv's performance on GIST prediction tasks, and compare it against baseline architectures.

5.1 DATASETS

Peptide Models. Peptide structures were obtained from the Database of Antimicrobial Activity and Structure of Peptides (DBAASP) (Pirtskhalava et al., 2021). Only peptides composed of canonical amino acids were considered. Peptides containing nonstandard residues, chemical modifications, or oligomeric assemblies were excluded. All peptides were simulated as monomers. In total 1100 peptides were used. Disulfide bonds between cysteine residues were introduced when supported by structural features compatible with disulfide bridge formation. This dataset was split into 880 training, 110 validation, 110 testing.

Protein Structures. A set of 94 protein targets was selected from the DUD-E dataset (Mysinger et al., 2012) for molecular dynamics simulations. Corresponding protein structures were obtained from the Protein Data Bank (PDB). Crystallographic water molecules present in the experimental structures were retained, while co-factors and ions were removed prior to simulation setup. This dataset was split into 69 training, 15 validation, 10 testing.

Molecular Dynamics Simulations. All peptide and protein systems were simulated using the CHARMM36m force field (Huang et al., 2017; Best et al., 2012) for biomolecular residues, while water molecules were modelled using the modified CHARMM TIP3P water model (Jorgensen et al., 1983). Each system was centred in a cubic periodic box of explicit solvent such that all solute atoms were at least 10 Å from the box boundaries. Molecular dynamics simulations were performed using the GPU-accelerated (AMD Instinct MI250 on Adastra MI250) NAMD program (version 3.0) (Phillips et al., 2005; 2020). Each system was subjected to energy minimisation, equilibration, and production simulations of 50 ns. The generated GIST data will be made publicly available.

Grid Inhomogeneous Solvation Theory Calculations. GIST calculations (Nguyen et al., 2014) were performed on molecular dynamics trajectories to characterise the spatially resolved thermodynamic properties of water molecules surrounding peptides and proteins. For each grid voxel, GIST computed water occupancy, solute–water interaction energies, water–water interaction energies, and solvation thermodynamic quantities. The resulting three-dimensional maps of water density and thermodynamic properties provided a detailed representation of solvent structure and energetics around each solute. The GIST grids generated from these calculations were used as targets for subsequent machine learning analyses.

Further details on peptide models, protein structures, MD simulations, and GIST are in Appendix D.

5.2 TRAINING PROCEDURE

EquiSolv is trained using the Adam optimiser with an initial learning rate of 1e-2. We employ a cosine annealing learning rate scheduler with a warm-up phase: the learning rate linearly increases from 10% of the initial rate over 25 epochs, then follows a cosine decay schedule with a minimum learning rate of 3e-4. Training proceeds with a batch size of 8, chosen after larger sizes showed a detriment to learning, processing up to 6000 neighbouring grid points per atom due to memory constraints. All model parameters are initialised using Kaiming initialisation. Full hyper-parameter specifications are provided in Appendix E. Training is performed using distributed data parallel (DDP) training using 8 H100 GPUs (80 GB), with synchronised batch normalisation across devices (1 protein per GPU, effective batch size of 8). We note that the largest protein handled ($dpp4$) is 9,243 atoms large, and memory usage during training for our strongest model peaked at 28.8 GB, see Appendix F.

Table 1: Regression performance of EquiSolv for eGIST grid-value prediction. Predictions and ground truth are both at the training coarsening of 1.0 Å. All metrics are in real space (kcal/mol/Å^3). RMSE is the per-protein root-mean-square error; accuracy columns are the percentage of voxels with $|\text{pred} - \text{truth}|$ below the stated threshold. Values are per-protein mean $\pm$ std.

Test set	**RMSE**	**Accuracy $<$ threshold (%)**			
		0.1	0.5	1.0	1.5
DBAASP ($n = 110$)	0.176 ± 0.026	70.9 ± 6.0	97.6 ± 0.8	99.8 ± 0.1	99.9 ± 0.0
DUD-E ($n = 10$)	0.302 ± 0.013	58.6 ± 1.8	92.8 ± 0.5	99.0 ± 0.1	99.7 ± 0.0

Per-atom local expansion using predicted coefficients reconstructs eGIST at each grid-point. The loss is computed as the MSE between predicted and target eGIST values at grid points. Validation is performed every 5 epochs to monitor generalisation performance. The loss function evaluates predicted radial basis function coefficients by reconstructing eGIST values at grid coordinates using the importance-weighted averaging scheme described in Section 3.5, then comparing against ground truth. Gradient clipping is applied with a maximum norm of 1.0 to stabilise training.

5.3 Layer Implementations

Our final architecture, which yields the lowest MSE, consists of one layer of Gated Convolution (1gc) followed by three $SE(3)$-Transformers (3tr). Gated convolutions use SiLU for scalar activations and sigmoid for gate activations, and transformer layers use SiLU for their MLP activations. All layers share the same $\gamma = 0.5$, i.e., equal weight for self position awareness and message passing contribution from neighbours. In this study, we evaluate the atom-specific local eGIST field on grids within 10 $\mathring{A}$ to the atom. The output irreps are $11 \times 0e + 11 \times 1o + 11 \times 2e$, meaning that we model the local expansion using basis functions of angular degree 2 with 11 degrees of freedom for the radial profile. The latter is set such that the evaluation range (10 $\mathring{A}$) divided by the irreps' multiplicity (11) is strictly less than the grid size, which is 1$\mathring{A}$.

5.4 Ablations

We systematically ablate architectural components (GatedConvolution vs Transformer composition and layer count), training hyperparameters (learning rate, scheduler configuration, and input features such as ESM embeddings, $Graphein$ feature sets, grid coarsening methods). Key findings: (1) ESM-C is helpful on DBAASP and nearly neutral on DUD-E. On DBAASP, removing ESM-C from the four-transformer stack raises validation MSE (Appendix G). On DUD-E the same stack without ESM-C is marginally better than with it ($+0.3\%$ in log-space validation MSE). We therefore do not claim that ESM-C is required for protein targets. (2) Architecture choices impact training: with the tuned kernel and loss stack, a mixed GatedConv-Transformer stack (1xGC + 3xTr) achieves the best validation MSE, improving past pure 4xTransformer and pure 4xGatedConv architectures. Moreover, 5xGatedConv and other mixed orderings remain competitive but slightly worse than 1xGC + 3xTR. (3) Atom to grid aggregation: importance-weighted reduction strongly outperforms mean pooling and is vastly more stable than sum aggregation. (4) Ablated training choices, learned α/β in the interaction kernel, and overly fine/coarse grids hurt validation MSE, while distance-scaled "halves" α, learned β offset, weighted non-bulk loss with a stricter bulk threshold, and slightly coarser training spacing help. Complete ablation results and analysis are provided in Appendix G.

6 Results

We evaluate eGIST fields on DBAASP peptides and DUD-E proteins. Regression is compared with a 3D U-Net and with DeepGIST (DG) (Fukushima & Yoshidome, 2026). Strongly bound water recovery against GalaxyWater-CNN (GW-CNN) (Park & Seok, 2022) is in Appendix I. The primary metric is field error. All comparisons use GIST values from the molecular dynamics simulations. Metric definitions are in Appendix H.

Table 2: EquiSolv vs. DeepGIST vs. 3D-CNN on 110 DBAASP test peptides, real space, on the shared 1.0 Å enthalpic grid (voxels where all three predictions exist). Wins are EquiSolv / DeepGIST / 3D-CNN.

Metric	**EquiSolv** (1.0 Å)	**DeepGIST** (1.0 Å)	**3D-CNN** (1.0 Å)	**Wins** (Eq / DG / CNN)
All region (every voxel in the 4-way intersection)				
Pearson ↑	$\mathbf{0.551 \pm 0.047}$	0.522 ± 0.126	0.144 ± 0.030	61 / 49 / 0
RMSE ↓	$\mathbf{0.185 \pm 0.027}$	0.717 ± 0.019	0.331 ± 0.024	110 / 0 / 0
Non-bulk region ($\lvert truth - bulk \rvert > 0.1$)				
Pearson ↑	$\mathbf{0.600 \pm 0.045}$	0.556 ± 0.122	0.212 ± 0.055	66 / 44 / 0
RMSE ↓	$\mathbf{0.322 \pm 0.026}$	0.832 ± 0.021	0.431 ± 0.028	110 / 0 / 0

Regression Performance and Baselines. On 110 held-out peptides, EquiSolv's real-space RMSE is 0.176 ± 0.026 kcal/mol/Å^3, and 97.6% of voxels fall within 0.5 (Table 1). On the ten held-out proteins, RMSE is 0.302 ± 0.013. On the shared 1.0 Å grid it wins RMSE on all 110 peptides (0.185 vs. 0.717 vs. 0.331), including the non-bulk voxels (0.322, again 110/110), and leads Pearson correlation (0.551 vs. 0.522 vs. 0.144; non-bulk 0.600, wins 66/44) (Table 2). The 0.185 is the three-method voxel intersection, which is why it differs from 0.176. The 3D-CNN is a 3D U-Net trained on the same 1.0 Å labels. Strongly bound water recovery gives an AUC-ROC of 0.966 against 0.883 and 0.876 (Table 7). Since no published model predicts the enthalpy alone, we compare to the closest available baseline. DeepGIST's published target includes an entropy term our labels do not, and it was not retrained on eGIST, so the RMSE gap is not a matched-target comparison (Table 11). Native-grid numbers and the DUD-E per-protein results are in Appendices H, M, and I.2.

Strongly Bound Waters. Ranking voxels by predicted eGIST recovers sites below -1.0 kcal/mol/Å^3 more precisely than GW-CNN at every listed operating point (Appendix I, Table 8). At $K = 100$, precision is 0.917 against 0.448, with F1 wins on 109 of 109 peptides. The lead holds at $K = 500$ and $K = 1000$. It also holds in every size tercile (Appendix J). On the ten DUD-E proteins the same ranking wins F1 at every K from 100 to 5000 (Appendix I.2).

Additional Results. A stage-wise timing breakdown is in Appendix K.

7 Conclusion and Limitations

We propose EquiSolv, an $O(3)$-equivariant model that learns coefficients of a local radial-spherical basis and reconstructs a continuous solvent energy field, rather than a value on a fixed voxel grid. On 110 held-out peptides the real-space RMSE is 0.176 kcal/mol/Å^3. On the shared 1.0 Å grid EquiSolv wins RMSE on all 110 peptides against DeepGIST (transferred from targets including entropy) and a 3D U-Net, and strongly bound sites are recovered with F1 wins on 109 of 109 peptides at $K \leq 1000$. On DBAASP, end-to-end inference averages 9.6 s per peptide (Figure 3); after the model is loaded once, the remaining stages average about 2.9 s, of which the network forward pass is 0.23 s. By contrast, the 50 ns explicit-solvent MD used for each GIST label takes over a full day on an AMD Instinct MI250 (Section 5). That gap is what makes the field usable for large sets of structures.

Limitations. We find that scaled enthalpy fields around bigger molecules are more complex and therefore harder to predict. One way to improve on this is to flesh out the protein dataset further, as the set of 94 proteins may not be adequate for comprehensive learning. We further found that layer architectures of 4 or more Transformer layers use up our 80 GB GPU space when handling the large protein dataset. One way to improve the model and reduce memory usage with layer types that exponentially increase in complexity with number of edges would be to experiment with residue-level granularity for the protein graphs. Docking validation is a comprehensive in its own right and unnecessarily limits the scope of our work and is therefore left for future work. Another struggle in our work is finding compatible baselines, as our work tackles solvation prediction in a novel way that makes comparison to existing methods difficult. We plan to publicly release our simulation data to enable greater comparability of models moving forward.

AI USE STATEMENT

A language model was used to help with formatting, keep track of changes and create summaries for collaboration within our group, as an additional search tool for literature, to write boilerplate code, and help speed up use of run-commands for code execution by plugging args into a template command. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Dataset construction, MD and GIST settings, hyperparameters, features, ablations, metrics, and timing are in Appendices D-K. The GIST grids will be released with the camera-ready version. Code will be released with the camera-ready version.

REFERENCES

Robert Abel, Tom Young, Ramy Farid, Bruce J. Berne, and Richard A. Friesner. Role of the active-site solvent in the thermodynamics of factor Xa ligand binding. *Journal of the American Chemical Society*, 130(9):2817–2831, 2008.

Trent E. Balius, Marcus Fischer, Reed M. Stein, Thomas B. Adler, Crystal N. Nguyen, Anthony Cruz, Michael K. Gilson, Tom Kurtzman, and Brian K. Shoichet. Testing inhomogeneous solvation theory in structure-based ligand discovery. *Proceedings of the National Academy of Sciences*, 114 (33):E6839–E6846, 2017.

Alexander S Bayden, Demetri T Moustakas, Diane Joseph-McCarthy, and Michelle L Lamb. Evaluating free energies of binding and conservation of crystallographic waters using szmap. *Journal of chemical information and modeling*, 55(8):1552–1565, 2015.

Herman JC Berendsen, JPM van Postma, Wilfred F van Gunsteren, A DiNola, and JR Haak. Molecular dynamics with coupling to an external bath. *The Journal of Chemical Physics*, 81(8):3684–3690, 1984.

Robert B Best, Xiao Zhu, Jihyun Shim, Pedro EM Lopes, Jeetain Mittal, Michael Feig, and Alexander D MacKerell. Optimization of the additive charmm all-atom protein force field targeting improved sampling of the backbone ϕ, ψ and side-chain χ_1 and χ_2 dihedral angles. *Journal of Chemical Theory and Computation*, 8(9):3257–3273, 2012.

Felix Brockherde, Leslie Vogt, Li Li, Mark E. Tuckerman, Kieron Burke, and Klaus-Robert Müller. Bypassing the kohn-sham equations with machine learning. *Nature Communications*, 8(1), October 2017. ISSN 2041-1723.

Bernard R Brooks, Charles L Brooks, Alexander D Mackerell, Lennart Nilsson, Robert J Petrella, Benoît Roux, Youngdo Won, Georgios Archontis, Christian Bartels, Stefan Boresch, et al. Charmm: the biomolecular simulation program. *Journal of Computational Chemistry*, 30(10):1545–1614, 2009.

Martin D. Buhmann. *Radial Basis Functions: Theory and Implementations*. Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, 2003.

Tom Darden, Darrin York, and Lee Pedersen. Particle mesh ewald: An n·log(n) method for ewald sums in large systems. *The Journal of Chemical Physics*, 98(12):10089–10092, 1993.

Nadav Dym and Haggai Maron. On the universality of rotation equivariant point cloud networks. *International Conference on Learning Representations (ICLR)*, 2021.

Valentin J. Egger-Hoerschinger, Franz Waibl, Vjay Molino, Helmut Carter, Monica Fernández-Quintero, Steven Ramsey, Daniel R. Roe, Klaus R. Liedl, Michael K. Gilson, and Tom Kurtzman. Quantifying spatially resolved hydration thermodynamics using grid inhomogeneous solvation theory [v1.0]. *Living Journal of Computational Molecular Science*, 6(1):3059, Aug 2025.

Evolutionary Scale Team. Esm cambrian: Revealing the mysteries of proteins with unsupervised learning., 2024. URL https://www.evolutionaryscale.ai/blog/esm-cambrian.

Ben Finkelshtein, Chaim Baskin, Haggai Maron, and Nadav Dym. A simple and universal rotation equivariant point-cloud network. In *Topological, Algebraic and Geometric Learning Workshops 2022*, pp. 107–115. PMLR, 2022.

Bengt Fornberg, Elisabeth Larsson, and Natasha Flyer. Stable computations with gaussian radial basis functions. *SIAM Journal on Scientific Computing*, 33(2):869–892, 2011.

Fabian Fuchs, Daniel Worrall, Volker Fischer, and Max Welling. Se(3)-transformers: 3d roto-translation equivariant attention networks. *Advances in neural information processing systems*, 33: 1970–1981, 2020.

Yusaku Fukushima and Takashi Yoshidome. Deep gist: Deep learning models for predicting the distribution of hydration thermodynamics around proteins. *Journal of Chemical Information and Modeling*, 2026.

Alfonso T. Garcia-Sosa, Ricardo L. Mancera, and Philip M. Dean. Waterscore: a novel method for distinguishing between bound and displaceable water molecules in the crystal structures of the binding site of protein-ligand complexes. *Journal of Molecular Modeling*, 9(3):172–182, 2003.

Mario Geiger and Tess Smidt. e3nn: Euclidean neural networks. *arXiv:2207.09453*, 2022.

A. Ghanbarpour, A. H. Mahmoud, and M. A. Lill. Instantaneous generation of protein hydration properties from static structures. *Communications Chemistry*, 3(1):1–10, 2020a.

A. Ghanbarpour, A. H. Mahmoud, and M. A. Lill. On-the-fly prediction of protein hydration densities and free energies using deep learning. *arXiv preprint arXiv:2001.02201*, 2020b.

Michael K Gilson and Huan-Xiang Zhou. Calculation of protein-ligand binding affinities. *Annual Review of Biophysics and Biomolecular Structure*, 36:21–42, 2007.

Peter J. Goodford. A computational procedure for determining energetically favorable binding sites on biologically important macromolecules. *Journal of Medicinal Chemistry*, 28(7):849–857, 1985.

Thomas Hangelbroek, Wolodymyr Madych, Francis Narcowich, and Joseph D Ward. Cardinal interpolation with gaussian kernels. *Journal of Fourier Analysis and Applications*, 18(1):67–86, 2012.

Bingjie Hu and Markus A. Lill. Watsite: hydration site prediction program with pymol interface. *Journal of Computational Chemistry*, 35(16):1255–1263, 2014.

Jing Huang, Sarah Rauscher, Grzegorz Nawrocki, Ting Ran, Michael Feig, Bert L de Groot, Helmut Grubmüller, and Alexander D MacKerell. Charmm36m: an improved force field for folded and intrinsically disordered proteins. *Nature Methods*, 14(1):71–73, 2017.

Arian R. Jamasb, Ramon Viñas, Eric J. Ma, Charlie Harris, Kexin Huang, Dominic Hall, Pietro Lió, and Tom L. Blundell. Graphein - a python library for geometric deep learning and network analysis on biomolecular structures and interaction networks. In *Advances in Neural Information Processing Systems*, volume 35, pp. 27153–27167, 2022.

William L Jorgensen, Jayaraman Chandrasekhar, Jeffry D Madura, Roger W Impey, and Michael L Klein. Comparison of simple potential functions for simulating liquid water. *The Journal of Chemical Physics*, 79(2):926–935, 1983.

Kosuke Kawama, Yusaku Fukushima, Mitsunori Ikeguchi, Masateru Ohta, and Takashi Yoshidome. gr predictor: A deep learning model for predicting the hydration structures around proteins. *Journal of Chemical Information and Modeling*, 62(18):4460–4473, 2022.

Andriy Kovalenko and Fumio Hirata. Three-dimensional density profiles of water in contact with a solute of arbitrary shape: a RISM approach. *Chemical Physics Letters*, 290(1–3):237–244, 1998.

Johannes Kraml, Anna S Kamenik, Franz Waibl, Michael Schauperl, and Klaus R Liedl. Solvation free energy as a measure of hydrophobicity: Application to serine protease binding interfaces. *Journal of Chemical Theory and Computation*, 15(11):5872–5882, 2019.

Themis Lazaridis. Inhomogeneous fluid approach to solvation thermodynamics. 1. theory. *The Journal of Physical Chemistry B*, 102(18):3531–3541, 1998.

A. H. Mahmoud, M. R. Masters, Y. Yang, and M. A. Lill. Elucidating the multiple roles of hydration for accurate protein-ligand binding prediction via deep learning. *Communications Chemistry*, 3(1): 1–12, 2020.

Michael M. Mysinger, Michael Carchia, John J. Irwin, and Brian K. Shoichet. Directory of useful decoys, enhanced (dud-e): Better ligands and decoys for better benchmarking. *Journal of Medicinal Chemistry*, 55(14):6582–6594, 2012.

Crystal N Nguyen, Tom Kurtzman Young, and Michael K Gilson. Grid inhomogeneous solvation theory: hydration structure and thermodynamics of the miniature receptor cucurbit [7] uril. *The Journal of chemical physics*, 137(4), 2012.

Crystal N Nguyen, Anthony Cruz, Michael K Gilson, and Tom Kurtzman. Thermodynamics of water in an enzyme active site: Grid-based hydration analysis of coagulation factor xa. *Journal of Chemical Theory and Computation*, 10(7):2769–2780, 2014.

Mounir Nina, Dmitri Beglov, and Benoît Roux. Atomic radii for continuum electrostatics calculations based on molecular dynamics free energy simulations. *The Journal of Physical Chemistry B*, 101 (26):5239–5248, 1997.

Mats HM Olsson, Chresten R Søndergaard, Michal Rostkowski, and Jan H Jensen. Propka3: Consistent treatment of internal and surface residues in empirical pka predictions. *Journal of Chemical Theory and Computation*, 7(2):525–537, 2011.

S. Park and C. Seok. Galaxywater-cnn: Prediction of water positions on the protein structure by a 3d-convolutional neural network. *Journal of Chemical Information and Modeling*, 62(13): 3157–3168, 2022.

Saro Passaro and C Lawrence Zitnick. Reducing so (3) convolutions to so (2) for efficient equivariant gnns. In *International conference on machine learning*, pp. 27420–27438. PMLR, 2023.

James C Phillips, Rosemary Braun, Wei Wang, James Gumbart, Emad Tajkhorshid, Elizabeth Villa, Christophe Chipot, Robert D Skeel, Laxmikant Kale, and Klaus Schulten. Scalable molecular dynamics with namd. *Journal of Computational Chemistry*, 26(16):1781–1802, 2005.

James C Phillips, David J Hardy, Julio DC Maia, John E Stone, João V Ribeiro, Rafael C Bernardi, Ronak Buch, Giacomo Fiorin, Jérôme Hénin, Wei Jiang, et al. Scalable molecular dynamics on cpu and gpu architectures with namd. *The Journal of Chemical Physics*, 153(4), 2020.

Malak Pirtskhalava, Anthony A Amstrong, Maia Grigolava, Mindia Chubinidze, Evgenia Alimbarashvili, Boris Vishnepolsky, Andrei Gabrielian, Alex Rosenthal, Darrell E Hurt, and Michael Tartakovsky. Dbaasp v3: database of antimicrobial/cytotoxic activity and structure of peptides as a resource for development of new therapeutics. *Nucleic Acids Research*, 49(D1):D288–D297, 01 2021.

Joshua A. Rackers, Lucas Tecot, Mario Geiger, and Tess E. Smidt. Cracking the quantum scaling limit with machine learned electron densities. *arXiv:2201.03726*, 2022.

Steven Ramsey, Crystal Nguyen, Romelia Salomon-Ferrer, Ross C. Walker, Michael K. Gilson, and Tom Kurtzman. Solvation thermodynamic mapping of molecular surfaces in ambertools: Gist. *Journal of Computational Chemistry*, 37(21):2029–2037, 2016.

Gianluca Rossato, Beat Ernst, Angelo Vedani, and Marián Smieško. Acquaalta: A directional approach to the solvation of ligand-protein complexes. *Journal of Chemical Information and Modeling*, 51(8):1863–1873, 2011.

Nathaniel Thomas, Tess Smidt, Steven Kearnes, Lusann Yang, Li Li, Kai Kohlhoff, and Patrick Riley. Tensor field networks: Rotation-and translation-equivariant neural networks for 3d point clouds. *arXiv*, 2018.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

A. Zamanos, G. Ioannakis, and I. Z. Emiris. Hydraprot: a new deep learning tool for fast and accurate prediction of water molecule positions for protein structures. *Journal of Chemical Information and Modeling*, 64(3):1024–1035, 2024.

A PRELIMINARIES

We introduce the notions from group and representation theory that are used in this work.

A.1 GROUP, REPRESENTATION AND EQUIVARIANCE

The group $O(3)$ has broad real-world relevance and consists of 3D rotations and inversion. We briefly recall the definition of a group before introducing further mathematical details.

Definition A.1 (Group)**.** A group $(\mathcal{G}, \circ)$ is a set $\mathcal{G}$ with an associative operation $\circ$, such that i) the set is closed under the operation, i.e., $\forall x_1, x_2 \in \mathcal{G},\ x_1 \circ x_2 \in \mathcal{G}$; ii) the set contains an identity element e under the operation, i.e., $\forall x \in \mathcal{G},\ e \circ x = x \circ e = x$; and iii) each element in the set has an inverse, i.e., $\forall x \in \mathcal{G},\ \exists x^{-1} \in \mathcal{G},\ x^{-1} \circ x = x \circ x^{-1} = e$.

Definition A.2 (Representation)**.** A representation (π, W) of a group $(\mathcal{G}, \circ)$ on the vector space W is a map $\pi : \mathcal{G} \to GL(W)$, where $GL(W)$ is the group of invertible linear maps on W and the map π is a homomorphism, i.e.,

$$\forall x_1, x_2 \in \mathcal{G},\ \pi(x_1 \circ x_2) = \pi(x_1)\pi(x_2).$$

An irreducible representation (irrep) is a representation allowing no proper non-zero subspace of W such that the homomorphism restricted to the subspace is a representation, i.e., there does not exist $E \subset W$ such that $(\pi|_E, E)$ with $\pi|_E : \mathcal{G} \to GL(E)$ is a representation.

We illustrate these notions in the context of this work. A 3D vector admits rotations and inversion, represented by matrices. The set of rotation matrices together with the inversion matrix defines the natural representation of $O(3)$ on $\mathbb{R}^3$, which is irreducible.

New representations can be constructed from two given representations by tensor product:

$$\begin{aligned} \pi_{W_1 \otimes W_2} : \mathcal{G} &\to W_1 \otimes W_2 \\ x &\mapsto \pi_{W_1}(x) \otimes \pi_{W_2}(x). \end{aligned}$$

The term representation may sometimes refer either to the homomorphism or to the space on which the matrices act. In this work, we mainly use the term representation to refer to the space rather than the map. Finally, we give the definition of a function being equivariant under a group action.

Definition A.3 (Equivariance)**.** Given a group $\mathcal{G}$, a function $f : W_1 \to W_2$ is said to be $\mathcal{G}$-equivariant if

$$\forall x \in \mathcal{G}, \forall \omega \in W_1,\ f(\pi_{W_1}(x)\omega) = \pi_{W_2}(x) f(\omega).$$

In this context, rotating the input of an $O(3)$-equivariant function results in a corresponding rotation of the output in the co-domain.

A.2 SPHERICAL HARMONIC EXPANSION THEOREM

Spherical harmonics $\{Y_\ell^m\}_{\ell \in \mathbb{N}}$ form an orthonormal basis of square-integrable functions defined on the unit sphere $\mathcal{S}^2$, they are characterised by two indices, the degree ℓ, corresponding to the angular frequency; and the order $m \in \{-\ell, \dots, \ell\}$, indicating the orientations. A key property of the spherical harmonics is the equivariance under 3D rotations, making them a natural choice when working with geometric constraints like rotation-equivariance. The expansion theorem further indicates how the coefficients are computed when decomposing any square-integrable function on the unit sphere using spherical harmonics. Precisely,

Theorem A.4 (Spherical Harmonic Expansion)**.** *Let $f \in L^2(S^2)$ be a square-integrable function on the unit sphere $S^2 = \{(\theta, \phi) \mid 0 \le \theta \le \pi,\ 0 \le \phi < 2\pi\}$ with respect to the surface measure $d\Omega = \sin\theta\, d\theta\, d\phi$. Then f admits an expansion in terms of spherical harmonics:*

$$f(\theta, \phi) = \sum_{\ell=0}^{\infty} \sum_{m=-\ell}^{\ell} a_\ell^m\, Y_\ell^m(\theta, \phi),$$

with $a_\ell^m = \int_{S^2} f(\theta, \phi)\, \overline{Y_\ell^m(\theta, \phi)}\, d\Omega$. The series converges to f in the L^2-sense, i.e.,

$$\lim_{L \to \infty} \int_{S^2} \left| f(\theta, \phi) - \sum_{\ell=0}^{L} \sum_{m=-\ell}^{\ell} a_\ell^m\, Y_\ell^m(\theta, \phi) \right|^2 d\Omega = 0.$$

A.3 $O(3)$-EQUIVARIANT MODELS

We now introduce the $O(3)$-equivariant architectures that we build upon in this work. Recent models exploiting rotational symmetry share the central idea of constructing equivariant convolution kernels as separable products of a learnable radial function and spherical harmonics capturing the angular dependence. The Tensor Field Network (TFN) of Thomas et al. (2018) provides the foundational design of convolution kernels expressed as linear combinations of spherical harmonics up to finite degree multiplied by arbitrary learnable radial functions (convolution layer), followed by linear mixing across feature channels (self-interaction layer). Going in the same direction, the SE(3)-Transformer of Fuchs et al. (2020) specifies the radial profiles as MLPs and introduces a self-attention mechanism, adapting the Transformer framework (Vaswani et al., 2017) to equivariant message passing. The $e3nn$ library by Geiger & Smidt (2022) further optimises these constructions, implementing equivariant tensor products with compact MLP-based radial dependencies in a flexible and efficient way. Importantly, works by Dym & Maron (2021); Finkelshtein et al. (2022) show that networks of this form are universal approximators of continuous $O(3)$-equivariant functions. Mathematically, the kernel defined in the TFN is

$$K_R(r_{ab}) = \sum_{\ell,m} R_{\ell,m}(\|r_{ab}\|) Y_\ell^m \left(\frac{r_{ab}}{\|r_{ab}\|} \right).$$

At layer i, let the radial function be an MLP, denoted as $\phi^{(i)}$, the convolution with the learnable kernel can be written as

$$\begin{aligned} h_a^{(i+1)} &= \sum_{b \in \mathcal{N}_a} K_{\phi^{(i)}}(r_{ab}) h_b^{(i)} \\ &= \sum_{b \in \mathcal{N}_a} \left(\sum_{\ell,m} Y_\ell^m \left(\frac{r_{ab}}{\|r_{ab}\|} \right) \phi_{\ell,m}^{(i)}(r_{ab}) \right) h_b^{(i)} \\ &=: \sum_{b \in \mathcal{N}_a} Y_{\ell_{\text{in}}} \left(\frac{r_{ab}}{\|r_{ab}\|} \right) \otimes_\Theta h_b^{(i)}, \end{aligned}$$

where ℓ_{in} is the representation of h_b^i. We follow the generalised definition of tensor product from Geiger & Smidt (2022) and group the corresponding Clebsch-Gordan coefficients and the learnable radial scalar value function $R(r)$ into $\phi(r)$. The $\otimes_\Theta$ notation in the last equality packs the MLP $\phi^{(i)}$'s parameters in Θ.

$SE(3)$-Transformer further enhances the radial profile by the attention mechanism. Denote learnable self-interaction layers as $s_\bullet$, the equivariant message-passing layer in SE(3)-Transformer is

$$h_a^{(i+1)} = s_V(h_a^{(i)}) + \sum_{b \in \mathcal{N}_a} \alpha_{ab} v_{ab},$$

where α is the attention between a and b is defined as

$$\alpha_{ab} = \frac{\exp\left(q_a \otimes_{\text{dot}} k_{ab}\right)}{\sum_{b' \in \mathcal{N}_a \setminus \{a\}} \exp\left(q_a \otimes_{\text{dot}} k_{ab'}\right)}, \tag{2}$$

where $\otimes_{\text{dot}}$ is the weighted tensor product returning a scalar value. The per-node query vector q_a, the per-edge key vector k_{ab} and the per-edge value vector v_{ab} are defined as

$$\begin{aligned} q_a &= s_Q(h_a^{(i)}), \\ k_{ab} &= Y_{\ell_i}(\hat{r}_{ab}) \otimes_{\phi_K} h_b^{(i)}, \\ v_{ab} &= Y_{\ell_i}(\hat{r}_{ab}) \otimes_{\phi_V} h_b^{(i)}. \end{aligned}$$

B GRAPH FEATURES

Molecular graphs are constructed using the $Graphein$ library with atom-level granularity. Edges are created based on Van der Waals (VDW) distance interactions using `add_vdw_interactions`, along with additional biologically-relevant interaction types. Nodes and edges are enriched with features as detailed below.

B.1 Node Features

Node features are computed at the atom level and can be categorised into three groups based on the feature preset mode.

B.1.1 Core Node Features (feat_preset minimal)

The minimal preset, which is the standard preset unless otherwise stated, includes only explicitly defined feature blocks:

1. **Meiler embeddings**: Vector embeddings (typically 7-10 dimensions) encoding amino acid properties, added via the `meiler_embedding` node metadata function.
2. **Expasy protein scale properties**: Multiple per-residue amino acid property scalars added to nodes. Includes:
   - `expasy_hydropathy`, `hydropathy`, `hydrophobicity`, `hydrophobicity_KD`
   - `polarity`, `bulkiness`, `flexibility`
   - `isoelectric_point`, `instability_index`
   - `helix_propensity`, `sheet_propensity`, `turn_propensity`, `coil_propensity`
3. **Additional scalar features**:
   - `rsa`, `asa`: Relative and absolute solvent accessible surface area (from SASA/DSSP calculations, if available)
   - `hydrogen_bond_donor`, `hydrogen_bond_acceptor`: Binary indicators from respective node metadata functions
   - `molecular_weight`, `partial_charge`: Molecular properties (if available in Graphein installation)
4. **Distance threshold**: Maximum radius (r_{max}) distance threshold to count grid-points per atom from `atom_dist.csv` lookup table, added via `custom_distance_function`. To define this list we performed 50 ns MD simulations of the side chain of interest in a cubic simulation box and computed GIST data from the resulting trajectories. The distance cutoff was defined based on energetic deviations in the GIST grid (Esw+Eww). Specifically, we first computed the average grid energy and identified grid points whose energy deviated by more than 10% from this average. The distance was then defined as the maximum distance between any atom of the side chain and the furthest GIST grid point exhibiting an energy deviation exceeding this 10% threshold.
5. **Amino acid one-hot encoding**: Processed separately as `res_oh` (not included in feature blocks).

B.1.2 Extended Node Features (feat_preset full)

The full preset includes all features from the minimal preset, plus:

- **Auto-harvested features**: Any additional numeric node attributes (up to 32 dimensions per feature) that are transferred from the NetworkX graph but not explicitly listed in the known feature blocks. These are automatically collected from Graphein node metadata functions that may vary by installation version or configuration.

B.2 Edge Features

Edge features encode both interaction type and geometric information:

1. **Edge kind (one-hot encoded)**: Interaction type labels including:
   - `peptide_bond`: Peptide C-N bonds across residues
   - `covalent`: Covalent bonds from molecular topology

- `vdw_interaction`: Van der Waals distance-based interactions (primary edge construction method)
- `disulfide_interaction`, `hydrophobic_interaction`, `aromatic_interaction`
- `aromatic_sulphur_interaction`, `cation_pi_interaction`
- `hydrogen_bond_interaction`, `ionic_interaction`, `salt_bridge`
- `backbone_carbonyl_carbonyl_interaction`
- `vdw_clash`: Van der Waals clash edges (if available)

2. **Edge geometry**:
   - `edge_length`: Euclidean distance between connected atoms (scalar)
   - `edge_unit_vec`: Normalised 3D direction vector from source to target atom
3. **Additional edge attributes (full preset only)**: Any numeric edge attributes beyond `kind` that are present in the NetworkX graph edges, automatically collected from Graphein edge metadata functions.

B.3 Edge Construction

Edges are constructed using the following `Graphein` edge construction functions (in order of priority for kind assignment):

- `add_peptide_bonds`: Peptide C-N bonds across residues
- `add_atomic_edges`: Covalent bonds from molecular topology
- `add_vdw_interactions`: Van der Waals distance-based interactions (primary method)
- `add_disulfide_interactions`, `add_hydrophobic_interactions`, `add_aromatic_interactions`
- `add_aromatic_sulphur_interactions`, `add_cation_pi_interactions`
- `add_hydrogen_bond_interactions`, `add_ionic_interactions`, `add_salt_bridges`
- `add_backbone_carbonyl_carbonyl_interactions`
- `add_vdw_clashes`: Van der Waals clash detection

Note: VDW interactions were chosen to provide edge connectivity as they represent biologically accurate and more selective connections compared to other options such as distance, k-nearest neighbour, and triangulation based edges. The rest of the functions here provide featurisation of the edges provided by VDW interactions.

C Proof of Proposition 4.1

Proof. Let's start by defining the notation.

We denote a theoretical solution $\rho(r, \omega)$ to the Inhomogeneous Solvation Theory (IST) Euler-Lagrange equation, describing the solution density field, U_S the energy functions used in the system. We consider a compact, connected and convex 3D manifold $\mathcal{M} \subset \mathbb{R}^3$ surrounding a protein. Without loss of generality, let $\mathcal{M}$ be an open ball centred at the origin with large but finite radius R_0 such that it contains the whole system and that the boundary effect is negligible. For analysis purposes, we exclude a small neighbourhood of hard-core or atomic surface Γ where singularities occur, that is we are interested in

$$\mathcal{M}_\delta := \{r \in \mathcal{M} \mid \|r - r_a\| > \delta, \forall a \in \Gamma\}.$$

We also restrict the domain of the solvent density $\rho(r, \omega)$ and the energy functions $U(r, \omega)$ to $\mathcal{M}_\delta \times SO(3)$, so that they are analytic functions in r.

The theoretical ground-truth described by involved differential equations admits no closed-form expression, but GIST provides trusting realisation from simulations on discretised grids. We denote

the voxel set $V \subset \mathcal{M}_\delta$ and an output of eGIST result $\bar{g}$. This intrinsic numerical error thus lower-bounds the performance of any model learning from GIST outputs. In what follows, we admit this error and focus on analysis of asymptotic behaviour of other sources of error between the optimal approximator and f.

By Whitney's extension theorem, there exists an interpolant $\bar{f}$ defined on $\mathcal{M}_\delta$ such that i) $\bar{f}(r_v) = \mathcal{T}[v]\ \forall v \in V$ and ii) $\bar{f} \in C^s,\ 1 \leq s \leq \infty$. The function $\bar{f}$ can be non-parametric or without closed-form. This auxiliary function serves as a connection between the unknown ground-truth and the theoretical best realisation for our analysis.

We first define $\mathcal{E}_{L_{\max}}$ the linear space spanned by the basis functions defined in (3.4), precisely,

$$\begin{aligned}\mathcal{E}_{L_{\max}} &:= \text{span}\ \mathcal{B} \\ &= \left\{ \sum_{\ell,k,m} c_{\ell,k,m} R_k Y_\ell^m \ \mid c_\bullet \in \mathbb{R} \right\}.\end{aligned}$$

Let's introduce an unknown and ideal approximator $\tilde{f}$, under the same expansion setup and has optimal theoretical coefficients that minimise its L^2 distance to $\bar{f}$, i.e.,

$$\tilde{f} := \text{argmin}\big\{ \|\bar{f} - \phi\|_{L^2} \ \mid \ \phi \in \mathcal{E}_{L_{\max}} \big\}.$$

Then by the triangle inequality,

$$\begin{aligned}\|f - \hat{f}\|_{L^2} &= \|f - \bar{f} + \bar{f} - \tilde{f} + \tilde{f} - \hat{f}\|_{L^2} \\ &\leq \underbrace{\|f - \bar{f}\|_{L^2}}_{\epsilon_{GIST}\text{: Numerical error}} + \underbrace{\|\bar{f} - \tilde{f}\|_{L^2}}_{E_1\text{: Truncation error}} + \underbrace{\|\tilde{f} - \hat{f}\|_{L^2}}_{E_2\text{: Learning loss}}.\end{aligned}$$

The ideal approximator $\tilde{f}$:

For any voxel coordinate $r_v \in \mathcal{M}_\delta$, the exact form of $\tilde{f}(r_v)$ is

$$\tilde{f}(r_v) = \sum_{a \in \mathcal{N}_v} w_a^*(r) \tilde{f}_a(r_v),$$

where $r = r_v - r_a$, $\mathcal{N}_v$ contains all atoms contributing meaningfully to the voxel value, $w^*(r)$ is the ideal aggregation weight function evaluated at voxel v for atom a and $\tilde{f}_a$ is the ideal local expansion of a with optimal and unknown coefficients c^* such that the minimum error is achieved, such a function has the form

$$\tilde{f}_a(r_v) = \sum_{\ell=0}^{L_{\max}} \sum_{k=1}^{M_\ell} \sum_{m=-\ell}^{\ell} c_{\ell k m}^{*[a]} R_k(\|r\|_2) Y_\ell^m \left(\frac{r}{\|r\|_2} \right).$$

Thus the fully rolled out ideal approximator with a rearrangement of the summation order is

$$\tilde{f}(r_v) = \sum_{a \in \mathcal{N}_v} \sum_{\ell=0}^{L_{\max}} \sum_{m=-\ell}^{\ell} \left(\sum_{k=1}^{M_\ell} w_a^*(r) c_{\ell k m}^{*[a]} R_k(\|r\|_2) \right) Y_\ell^m \left(\frac{r}{\|r\|_2} \right).$$

The learned optimal approximator $\hat{f}$:

The learned optimal approximator has exactly the same form as the ideal approximator, but its coefficients are learned. A subtle difference between $\tilde{f}$ and $\hat{f}$ is that $\tilde{f}$ has theoretical optimal coefficients, these coefficients are themselves targets of EquiSolv, thus $\hat{f}$ is assumed optimal in the sense that its coefficients are output of the minimiser of the population risk over the network-induced hypothesis class and should not be confused with an estimator learned from a finite dataset.

$$\begin{aligned}\hat{f}(r_v) &:= \sum_{a \in \mathcal{N}_v} \hat{\omega}_a(r) \hat{f}_a(r_v) \\ &= \sum_{a \in \mathcal{N}_v} \frac{R(\|r\|_2)}{\sum_{a' \in \mathcal{N}_v} R(\|r'\|_2)} \hat{f}_a(r_v),\end{aligned}$$

where R is the Gaussian ring radial function with user-defined parameters α and β. Similarly $\hat{f}_a$ is

$$\hat{f}_a(r_v) = \sum_{\ell=0}^{L_{\max}} \sum_{k=1}^{M_\ell} \sum_{m=-\ell}^{\ell} c_{\ell km}^{[a]} R_k(\|r\|_2) Y_\ell^m \left(\frac{r}{\|r\|_2} \right),$$

with $c^{[a]}$ optimally estimated.

Next let's develop an upper bound for both error terms, respectively.

The truncation error $E_1 = \|\bar{f} - \tilde{f}\|_{L^2}$:

By Lemma C.1, the unknown ground-truth field f admits a physics-induced decomposition with an error term due to the truncation, $f = \sum_{a \in \mathcal{N}_v} f_a + \epsilon_{\mathcal{N}_v}$. For a smooth interpolant $\bar{f}$ of f such that $\bar{f}(v) = f(v), \forall v \in V$, the truncation error $\bar{\epsilon}_{\mathcal{N}_v}$ is identical to the original truncation error $\epsilon_{\mathcal{N}_v}$ at every voxel $v \in V$.

Again by Whitney's extension theorem, $\exists \tilde{\epsilon}(r) \in C^\infty$ a smooth residual function such that

$$\tilde{\epsilon}(r_v) = \epsilon_{\mathcal{N}_v}$$

A complete RBF-SH system therefore allows this residual to be represented

$$\tilde{\epsilon}(r_v) = \sum_{\ell=0}^{\infty} \sum_{m=-\ell}^{\ell} R_\epsilon(\|r_v\|) Y_\ell^m \left(\frac{r_v}{\|r_v\|} \right).$$

Let $\sum_{a \in \mathcal{N}_v} w_a(v) = 1$, one can then trivially distribute this residual into a finite linear combination having the form of

$$\tilde{\epsilon}(r_v) = \sum_{a \in \mathcal{N}_v} w_a(v) \tilde{\epsilon}(r_v).$$

Now define $\bar{f}_a(r_v) = f_a(r_v) + w_a \tilde{\epsilon}(r_v)$, then

$$\begin{aligned} \sum_{a \in \mathcal{N}_v} \bar{f}_a(r_v) &= \sum_{a \in \mathcal{N}_v} f_a(r_v) + \sum_{a \in \mathcal{N}_v} w_a \tilde{\epsilon}(r_v) \\ &= \sum_{a \in \mathcal{N}_v} f_a(r_v) + \tilde{\epsilon}(r_v) \\ &= f(r_v) \\ &= \bar{f}(r_v) \end{aligned}$$

In other words, the physics-induced truncation residual is not eliminated, but is redistributed among the sampled voxels and is mathematically absorbed into redefined local RBF-SH expansions:

$$\begin{aligned} \bar{f}(r_v) &= \sum_{a \in \mathcal{N}_v} \bar{f}_a(r_v) \\ &= \sum_{a \in \mathcal{N}_v} \sum_{\ell=0}^{\infty} \sum_{m=-\ell}^{\ell} \bar{R}_{\ell m}^{(a)}(\|r_v\|) Y_\ell^m \left(\frac{r_v}{\|r_v\|} \right) \end{aligned}$$

By orthogonality of the spherical harmonics,

$$\|E_1\|_{L^2}^2 = \sum_{\ell=0}^{L_{\max}} \sum_m \left\| \sum_a \left(\bar{R}_{\ell m}^{(a)} - \sum_{k=1}^{M_\ell} w_a^* c_{\ell km}^{*[a]} R_k \right) \right\|_{L^2}^2 + \sum_{\ell > L_{\max}} \sum_m \left\| \sum_a \bar{R}_{\ell m}^{(a)} \right\|_{L^2}^2$$

The missing angular part due to the finite highest spherical harmonic degree ($\ell > L_{\max}$) can be bounded by

$$\begin{aligned} E_{1a}^2 &:= \sum_{\ell > L_{\max}} \sum_m \left\| \sum_a \bar{R}_{\ell m}^{(a)} \right\|_{L^2}^2 \\ &\leq \sum_{\ell > L_{\max}} \sum_m \sum_a \left\| \bar{R}_{\ell m}^{(a)} \right\|_{L^2}^2 \quad \text{Cauchy-Schwarz Ineq.} \\ &\leq \max_{v \in V} |\mathcal{N}_v| (\mathrm{Cst}_a L_{\max}^{-s})^2. \end{aligned}$$

Denote the largest number of atom neighbours of a voxel, i.e., the maximum local density of the input graph, $\max_{v\in V}\mathcal{N}_v := \nu$, we have

$$E_{1a}^2 = \mathcal{O}(\nu L_{\max}^{-2s})$$
$$E_{1a} = \mathcal{O}(\nu^{1/2} L_{\max}^{-s})\ \forall s > 0,$$

the decay is super-algebraic given that $\bar{f} \in C^\infty$.

We next look at the radial approximation part error. $[\bar{R}_{\ell m}^{(a)}]_{a,\ell,m}$ being the radial profiles of a C^∞ function inherit the regularity. Rich theoretical results (Buhmann, 2003; Hangelbroek et al., 2012; Fornberg et al., 2011) show that the error between such target functions and Gaussian kernel approximants is well behaved.

Recall that our radial basis functions are shell-shaped Gaussian kernels characterised by two parameters α_k and β_k,

$$R_k(x) = \exp\left(-\alpha_k (x-\beta_k)^2\right).$$

In our study we define $[\beta_k]$ uniformly spaced over a user-defined range of $[\mathrm{r}_{\min}, \mathrm{r}_{\max}]$, and $\alpha_k = \frac{1}{2}\left(\frac{r_{\max}-r_{\min}}{M_\ell+1}\right)^{-2}$, constant for all k. The fill distance is defined as

$$h_\ell := \max_{r\in\mathcal{M}_\delta} \min_{k=1,\dots,M_\ell} |\|r\|_2 - \beta_k|$$

can be computed explicitly as

$$h_\ell = \frac{1}{2}\frac{\mathrm{r}_{\max} - \mathrm{r}_{\min}}{M_\ell}.$$

Under the hypothesis that $\tilde{f}$ is ideal, the minimum error (over the parameters' space) is achieved. The error between the unknown yet C^∞ target function $\bar{R}_{\ell m}^{(a)}$ and the approximant by Gaussian kernels is well studied, we apply the result of Hangelbroek et al. (2012) and thus

$$\begin{aligned}
E_{1b}^2 &:= \sum_{\ell\le L_{\max}} \sum_m \left\| \sum_a \left(\bar{R}_{\ell m}^{(a)} - \sum_{k=1}^{M_\ell} w_a^* c_{\ell k m}^{*[a]} R_k \right) \right\|_{L^2}^2 \\
&\le \sum_{\ell\le L_{\max}} \sum_m \left(\max_{v\in V} |\mathcal{N}_v| \right) \min \left\| \bar{R}_{\ell m}^{(a)}(\|r\|_2) - \left[\sum_{k=1}^{M_\ell} w_a^*(r) c_{\ell k m}^{*[a]} R_k(\|r\|_2) \right] \right\|_{L^2(\mathcal{M}_\delta)}^2 \\
&\le \mathrm{Cst}^{[a,\ell,m]} \nu h_{\max}^{2s} \\
&= \mathcal{O}(\nu M_{\min}^{-2s})\ \forall s > 0,
\end{aligned}$$

where $h_{\max} := \max\{h_\ell \mid \ell \le L_{\max}\}$ and $M_{\min} := \min\{M_\ell \mid \ell \le L_{\max}\}$. Thus $E_{1b} = \mathcal{O}(\nu^{1/2} M_{\min}^{-s})$

The learning loss $E_2 = \|\tilde{f} - \hat{f}\|_{L^2}$:

$$\begin{aligned}
E_2^2 &= \|\tilde{f} - \hat{f}\|_{L^2(\mathcal{M}_\delta)}^2 \\
&= \left\| \sum_{a\in\mathcal{N}_v} \sum_{\ell,m,k} w_a^*(r) c_{\ell k m}^{*[a]} R_k(\|r\|_2) Y_\ell^m\left(\frac{r}{\|r\|_2}\right) - \sum_{a\in\mathcal{N}_v} \sum_{\ell,m,k} \hat{\omega}_a(r) c_{\ell k m}^{[a]} R_k(\|r\|_2) Y_\ell^m\left(\frac{r}{\|r\|_2}\right) \right\|_{L^2(\mathcal{M}_\delta)}^2 \\
&= \left\| \sum_{a\in\mathcal{N}_v} \sum_{\ell=0}^{L_{\max}} \sum_{m=-\ell}^{\ell} \sum_{k=1}^{M_\ell} \left(\underbrace{w_a^*(r) c_{\ell k m}^{*[a]} - \hat{\omega}_a(r) c_{\ell k m}^{[a]}}_{\psi_{a,\ell,m,k}(r)} \right) R_k(\|r\|_2) Y_\ell^m\left(\frac{r}{\|r\|_2}\right) \right\|_{L^2(\mathcal{M}_\delta)}^2 \\
&= \sum_{a,a'\in\mathcal{N}_v} \sum_{\ell=0}^{L_{\max}} \sum_{m=-\ell}^{\ell} \sum_{\{k,k'\}=1}^{M_\ell} \int_{\mathcal{M}_\delta} \psi_{a',\ell,m,k'}(r) \psi_{a,\ell,m,k}(r) R_{k'}(\|r\|_2) R_k(\|r\|_2) dr \quad \text{(orthogonality of SH).}
\end{aligned}$$

Define the radial Gram matrix G for R_k as

$$\begin{aligned} G_{k,k'} &= \int_{\mathcal{M}_\delta} R_k(\|r\|_2) R_{k'}(\|r\|_2) dr \\ &= \int_0^{R_0} R_k(x) R_{k'}(x) x^2 dx \int_{S^2} d\Omega \quad \text{(spherical coordinates).} \\ &= 4\pi \exp\left(-\frac{\alpha_k}{2}(\beta_k - \beta_{k'})^2\right) \int_0^{R_0} \exp\left(-2\alpha_k \left(x - \frac{\beta_k + \beta_{k'}}{2} \right)^2 \right) x^2 dx \end{aligned} \tag{3}$$

Let $\lambda_{\max}(G)$ be the largest eigenvalue of G, then E_2^2 is upper bounded

$$E_2^2 \leq \lambda_{\max}(G) \sum_{a \in \mathcal{N}_v} \sum_{\ell=0}^{L_{\max}} \sum_{m=-\ell}^{\ell} \sum_{k=1}^{M_\ell} \|\psi_{a,\ell,m,k}(r)\|^2_{L^2(\mathcal{M}_\delta)}.$$

We next evaluate the asymptotic behaviour of $\lambda_{\max}(G)$.

Recall that

$$\alpha_k = \frac{1}{2}\left(\frac{r_{\max} - r_{\min}}{M_\ell + 1}\right)^{-2},$$

that is $\alpha_k \sim M_\ell^2$, meaning that the Gaussian bell is narrow enough such that we can approximate x^2 by $\left(\frac{\beta_k + \beta_{k'}}{2}\right)^2$ in the integral in the eq.(3), resulting

$$\begin{aligned} G_{k,k'} &\approx 4\pi \left(\frac{\beta_k - \beta_{k'}}{2}\right)^2 \sqrt{\frac{\pi}{2\alpha_k}} \exp\left(-\frac{\alpha}{2}(\beta_k + \beta_{k'})^2\right) \\ &= \text{Cst } M_\ell^{-1} \exp\left(-\frac{\alpha}{2}(\beta_k + \beta_{k'})^2\right) \\ &= \text{Cst } M_\ell^{-1} \exp\left(-\frac{1}{2}\frac{1}{2}\left(\frac{r_{\max} - r_{\min}}{M_\ell + 1}\right)^{-2} ((k - k')\frac{r_{\max} - r_{\min}}{M_\ell + 1})^2\right) \\ &= \text{Cst } M_\ell^{-1} \exp\left(-\frac{(k - k')^2}{4}\right) \\ &= \text{Cst}' \; M_\ell^{-1}. \end{aligned}$$

Again because of the high sharpness, the Gaussian curves hardly overlap, rendering the Gram matrix approximately banded with strict positive values concentrated around the diagonal. Thus

$$\sum_{k'} G_{kk'} = \mathcal{O}\left(M_\ell^{-1}\right).$$

Since the Gram matrix G is positive semi-definite with positive elements, $\lambda_{\max}(G) \leq \max_k \sum_{k'} G_{kk'} = \mathcal{O}(M_{\min}^{-1})$.

It remains to analyse $\sum_{\ell,m,k} \|\psi_{a,\ell,m,k}(r)\|^2_{L^2(\mathcal{M}_\delta)}$, the approximation error of a model minimising the population risk.

By the universality result of Dym & Maron (2021), Tensor Field Networks (Thomas et al., 2018) and $SE(3)$-Transformers (Fuchs et al., 2020) are capable of representing all $O(3)$-equivariant polynomial maps. Since such polynomials form a dense subspace of $\mathcal{C}_{O(3)}(\mathcal{M}, W_\ell)$, the space of continuous $O(3)$-equivariant functions from a compact domain $\mathcal{M} \subset \mathbb{R}^3$ to the representation space $W_\ell \simeq \mathbb{R}^{2\ell+1}$, these architectures are universal approximators on $\mathcal{M}$. In particular, Dym & Maron (2021) show constructively that any $O(3)$-equivariant polynomial of total degree $\leq D$ can be represented by a TFN of depth $T = 2D$ whose feature channels include all irreducible representations up to order $\ell \leq D$. By analogy with classical polynomial approximation theory, and assuming the target function admits bounded derivatives up to order s, the approximation bias of a degree-D (or equivalently depth-T) TFN can be bounded heuristically by $\mathcal{O}(T^{-s})$, following a Jackson-type argument.

Thus

$$\sum_{\ell,m,k} \|\psi_{a,\ell,m,k}(r)\|_{L^2(\mathcal{M}_\delta)} = \mathcal{O}(T^{-s}).$$

And

$$\begin{aligned} E_2 &\leq \sqrt{\lambda_{\max}(G) \sum_{a\in\mathcal{N}_v} \sum_{\ell=0}^{L_{\max}} \sum_{m=-\ell}^{\ell} \sum_{k=1}^{M_\ell} \|\psi_{a,\ell,m,k}(r)\|^2_{L^2(\mathcal{M}_\delta)}} \\ &\leq \sqrt{\lambda_{\max}(G)} \sum_{a\in\mathcal{N}_v} \sum_{\ell,m,k} \|\psi_{a,\ell,m,k}(r)\|_{L^2(\mathcal{M}_\delta)} \\ &= \mathcal{O}(M_{\min}^{-1/2}) \left(\mathcal{O}(T^{-s})\right) \\ &= \mathcal{O}\left(M_{\min}^{-1/2} T^{-s}\right). \end{aligned}$$

Assembling the error terms concludes the proof. □

Lemma C.1. *f can be algebraically decomposed atom-wise as*

$$f(r_v) = \sum_{a\in\mathcal{N}_v} \sum_{\ell=0}^{\infty} \sum_{m=-\ell}^{\ell} R_{\ell m}^{(a)}(r) Y_\ell^m \left(\frac{r}{\|r\|_2}\right) + \epsilon_{\mathcal{N}_v}(d),$$

where

$$\begin{aligned} r &:= r_v - r_a, \\ R_{\ell m}^{(a)} &:= \textit{Unknown radial expansion from atom } a, \\ \epsilon_{\mathcal{N}_v}(d) &:= \textit{neighbourhood truncation error depending on } \mathcal{N}_v. \end{aligned}$$

Proof. By IST,

$$\begin{aligned} E_{sw}(r) &:= \int_{\mathrm{SO}(3)} \rho(r,\omega) U_{\text{ext}}(r,\omega)\, d\omega \\ &= \int_{\mathrm{SO}(3)} \rho(r,\omega) \sum_{a\in P} U_a(r,\omega)\, d\omega \\ &= \sum_{a\in P} \int_{\mathrm{SO}(3)} \rho(r,\omega) U_a(r,\omega)\, d\omega \\ &=: \sum_{a\in P} E_{sw}^{(a)}(r) \end{aligned}.$$

For a given voxel v, if one only considers atoms within a distance d of the voxel, the term is decomposed as

$$E_{sw}(r_v) = \sum_{a\in\mathcal{N}_v} E_{sw}^{(a)}(r_v) + \underbrace{\sum_{a\notin\mathcal{N}_v} E_{sw}^{(a)}(r_v)}_{\epsilon_{\mathcal{N}_v}}$$

We now briefly discuss the asymptotic behaviour of the truncation error term. Asymptotically, $\rho(r,\omega) \to \rho_0(\omega)$ the bulk, and U_{ext} decays as $r \to \infty$. So

$$E_{sw} \sim \int_{SO(3)} \rho_0(\omega) U_{\text{ext}}(r,\omega) d\omega,$$

$$\int_{r>d} E_{sw}(r) dr \sim \int_r^\infty \int_{S^2} \rho_0(\omega) U_{\text{ext}}(r,\omega) r^2 d\omega\, dr,$$

Depending on different types of U_{ext}, one has:

Lennard-Jones: $E_{sw}(r) \sim C\, r^{-6}$, then $\int_{r>d} E_{sw}(r)dr \sim \int_{r>d} r^{-6} r^2 dr = O(d^{-3})$.

Screened electrostatics: $E_{sw}(r) \sim C\frac{\exp(-r/\lambda)}{r}$, then $\int_{r>d} E_{sw}(r)dr \sim \int_{r>d} \frac{\exp(-r/\lambda)}{r} r^2 dr = O((d+\lambda)e^{-d/\lambda})$.

Coulomb: $E_{sw}(r) \sim C\, r^{-1}$, then $\int_{r>d} E_{sw}(r)dr \sim \int_{r>d} r^{-1}r^2 dr = \infty$, diverge.

For the E_{ww} term,

$$E_{ww}(r) := \frac{1}{2}\int_{SO(3)}\int_{\mathcal{M}_\delta \times \mathrm{SO}(3)} \rho(r,\omega)\rho(r',\omega')c^{(2)}(r,\omega;\, r',\omega')\, dr'd\omega' d\omega$$

being independent of a. Let a set of weights $\mathbf{w}_r = [w_a]_{a\in\mathcal{N}_r}$ whose elements sum up to 1, then trivially

$$\begin{aligned} E_{ww}(r) &= \left(\sum_{a\in\mathcal{N}_r} w_a\right) E_{ww}(r) \\ &= \sum_{a\in\mathcal{N}_r} w_a E_{ww}(r) \\ &=: \sum_{a\in\mathcal{N}_r} E^{(a)}_{ww}(r). \end{aligned}$$

Therefore,

$$\begin{aligned} f(r_v) &= E_{sw}(r_v) + 2E_{ww}(r_v) \\ &= \sum_{a\in\mathcal{N}_v} E^{(a)}_{sw}(r_v) + \sum_{a\notin\mathcal{N}_v} E^{(a)}_{sw}(r_v) + 2\left(\sum_{a\in\mathcal{N}_v} w_a\right) E_{ww}(r_v) \\ &= \sum_{a\in\mathcal{N}_v} E^{(a)}_{sw}(r_v) + \epsilon_{\mathcal{N}_v} + \sum_{a\in\mathcal{N}_v} 2E^{(a)}_{ww}(r_v) \\ &=: \sum_{a\in\mathcal{N}_v} f_a(r_v) + \epsilon_{\mathcal{N}_v}. \end{aligned}$$

The last definition (or equality) comes from grouping the summation terms on a. Let's further rewrite

$$f_a(r_v) = g_a(r) \quad \text{with } r := r_v - r_a.$$

$g_a(r)$ is analytic on $\mathcal{M}_\delta$, thus it can be expanded in radial functions times spherical harmonics, precisely,

$$g_a(r) = \sum_{\ell=0}^{\infty}\sum_{m=-\ell}^{\ell} R^{(a)}_{\ell m}(r) Y^m_\ell\left(\frac{r}{\|r\|_2}\right)$$

concluding the proof. □

D Dataset Preparation Details

This appendix contains details on the creation of our GIST grid datasets using the molecule and protein datasets.

D.1 Peptide Models

For each peptide containing histidine residues, both fully protonated (positively charged) and neutral histidine states were considered. For neutral histidines, both tautomeric forms, protonated on the Nε (HSE) or Nδ (HSD) position, were included. A total of 130 peptides containing histidine residues were among the selected peptides. For each of these peptides, three protonation variants were generated (fully protonated, neutral HSE, and neutral HSD), resulting in 390 peptide systems. The N-terminus and C-terminus were modelled in their charged forms.

D.2 Protein Structures

For protein structures containing missing residues within polypeptide chains, the covalent connectivity across the gaps was removed. To maintain structural stability, harmonic positional restraints with a

force constant of 20 kcal·mol^{-1}·Å^{-2} were applied to the backbone atoms of residues adjacent to the missing segments.

Titratable residues were assigned standard protonation states appropriate for physiological conditions, with the exception of histidine residues. Histidine protonation states were determined using a multi-step procedure. First, PropKa3 (Olsson et al., 2011) calculations were used to predict whether each histidine residue was protonated or neutral. For histidines predicted to be neutral, the preferred tautomeric form was determined using Poisson-Boltzmann electrostatic calculations performed with the PBEQ module of the CHARMM program (Brooks et al., 2009). In the Poisson-Boltzmann calculations, molecular volumes were defined using atomic radii reported by Nina et al. (1997), and a dielectric constant of 20 was used for the protein interior and 80 for solvent. For each neutral histidine, calculations were performed for both tautomeric states (proton on Nε and proton on Nδ). The tautomeric form with the lower electrostatic energy was selected for subsequent simulations. Disulfide bridges between cysteine residues were introduced where appropriate, based on the protein structure and known biochemical connectivity.

D.3 Molecular Dynamics Simulation Parameters

Periodic boundary conditions were applied in all three dimensions. Long-range electrostatic interactions were treated using the particle mesh Ewald (PME) method (Darden et al., 1993). An appropriate number of potassium or chloride ions were added to neutralise the total system charge. Van der Waals interactions were smoothly truncated using a switching function starting at 9 Å and reaching zero at a cutoff distance of 11 Å. Short-range non-bonded interactions were evaluated at every integration step, while long-range electrostatic interactions were computed every four steps. A time step of 2 fs was used, with covalent bonds involving hydrogen atoms constrained. During molecular dynamics simulations, harmonic positional restraints with a force constant of 5 kcal·mol^{-1}·Å^{-2} were applied to the non-hydrogen atoms of the peptide or protein relative to their initial coordinates. These restraints were used to maintain the overall solute conformation while allowing full flexibility of solvent molecules. Each system was first subjected to energy minimisation, followed by equilibration for 1,000,000 integration steps at a temperature of 295 K and a pressure of 1.0 atm (101.325 kPa). Temperature and pressure were controlled using the Berendsen thermostat and barostat (Berendsen et al., 1984) with a relaxation time of 500 fs, with positional rescaling applied every four time steps. After initial equilibration, systems were further equilibrated for 20 ns. Production simulations were subsequently carried out for 50 ns, and atomic coordinates were saved every 1 ps for subsequent analysis and GIST calculations.

D.4 GIST Calculation Parameters

GIST analysis was carried out on a three-dimensional Cartesian grid with a uniform spacing of 0.5 Å, covering the region surrounding the solute. A GPU-accelerated implementation of GIST (Kraml et al., 2019) was employed to enable efficient analysis of large trajectory datasets. For each system, 50,000 frames were extracted from the 50 ns production simulations at uniform time intervals and used as input for the GIST calculations. Entropy terms were evaluated based on translational and orientational correlations of water molecules relative to bulk reference states. The reference structure used for these analyses was defined as the average structure computed over the production trajectory using the non-hydrogen atoms of the peptide or protein.

E Hyper-parameter Specifications

The configuration below corresponds to the run that produced our reported model, `dbaaspr_1gc3tr_stack_halves_beta1.0_wloss_thresh0.5_esm_attn`.

E.1 Training Hyper-parameters

EquiSolv is trained using the Adam optimiser with the following settings:

- **Initial learning rate**: 1×10^{-2}
- **Maximum epochs**: 4000

- **Batch size**: 1 per GPU on 8 GPUs (effective batch size of 8)
- **Gradient clipping**: enabled (`-gradclip`); maximum norm 1.0
- **Weight initialisation**: Kaiming initialisation for all parameters
- **Checkpoint interval**: every 1 epoch (`-save_interval 1`)

E.2 Learning Rate Schedule

We employ a cosine annealing scheduler with warmup:

- **Scheduler type**: CosineAnnealingLR with warmup (`-scheduler cosine`)
- **Warmup epochs**: 25
- **Warmup start factor**: 0.1 (learning rate linearly increases from 1×10^{-3} to 1×10^{-2} over the warmup window)
- **Minimum learning rate**: 3×10^{-4}
- **Schedule**: After warmup, follows cosine decay from initial rate to minimum rate over the remaining epochs

E.3 Architecture Specifications

The final architecture consists of 4 equivariant layers (`-layers GatedConv,Trs,Trs,Trs`):

Layer sequence: 1 GatedConv layer followed by 3 SE(3)-Transformer layers.

Layer irreps progression (output of each layer, from `-hidden_irreps`):

- Layer 1 output (GC): `4x0e + 4x1o + 4x2e`
- Layer 2 output (Tr): `6x0e + 6x1o + 6x2e`
- Layer 3 output (Tr): `9x0e + 9x1o + 9x2e`
- Layer 4 output (Tr) / final irreps: `11x0e + 11x1o + 11x2e`

Residual connections: Each layer uses a fixed mixing parameter $\gamma = 0.5$ balancing the self-interaction (global-position-aware) term and the message-passing contribution from neighbours.

E.4 Loss and Aggregation

- **Atom→voxel reduction**: importance-weighted (`-grid_reduce weighted`)
- **Distance-scaled α**: `-alpha_rule halves` (per-atom α scales with the local cutoff)
- **Learnable β offset**: enabled (`-learn_beta_offset`) with maximum offset 1.0 Å (`-beta_offset_max 1.0`)
- **Training loss**: weighted MSE as in Section 3.5 (`-use_weighted_loss`), with $\tau = 0.5$ (`-loss_bulk_water_threshold 0.5`) and $w_{\text{nb}} = 100$

E.5 Radial Basis Function Parameters

Per-atom RBF coefficients are parameterised as Gaussian shells with:

- **Radial range**: $r \in [1.0, 9.21]$ Å ($r_{\min}$ from `-r_min 1.0`; $r_{\max}$ per-atom from MD analysis, max 9.21 Å)
- **Multiplicity per irrep degree**: determined by the final irreps (11 radial functions per $\ell \in \{0, 1, 2\}$)
- **Sharpness parameter** α_k: computed as $\alpha_k = 0.5 \cdot \left(\frac{r_{\max} - r_{\min}}{M_\ell + 1}\right)^{-2}$ for each multiplicity M_ℓ, then optionally rescaled per-atom by the `halves` rule
- **Centre positions** β_k: uniformly spaced over $[r_{\min}, r_{\max}]$ per irrep degree, with a learnable additive offset bounded by 1.0 Å

E.6 Grid Processing Parameters

- **Original grid spacing**: 0.5 Å (from MD/GIST simulations)
- **Coarsened spacing**: 1.0 Å via mean aggregation (`-coarse_spacing 1.0 -coarse_method mean`)
- **Maximum neighbours per atom**: 6000 grid points (`-max_neighbours 6000`)
- **Neighbourhood strategy**: closest neighbours within the per-atom distance threshold
- **Distance filtering**: $r_{\min} = 1.0$ Å, $r_{\max} = 9.21$ Å (per-atom-specific from MD analysis; see Appendix B)

E.7 Early Stopping and Validation

- **Early stopping patience**: 50 epochs (`-ESpatience 50`)
- **Minimum training epochs**: 50 (`-min_epochs 50`; early stopping cannot trigger before this)
- **Validation frequency**: every 5 epochs (default `-val_freq 5`)
- **Checkpoint saving**: best validation-loss checkpoint automatically saved
- **Minimum improvement delta**: 0 (any improvement counts)

E.8 Feature Embedding Specifications

- **Feature preset**: `minimal` (`-feat_preset minimal`, see Appendix B)
- **Edge embedder**: learnable (`-learnable_edges`); edge output dimension 768 (`-nedgeout 768`)
- **Node hidden MLP**: two layers of width 512 (`-nhidden 512,512`)
- **Spherical harmonic degree**: automatically determined from layer irreps (up to $\ell = 2$ for the final irreps; intermediate kernels use up to $\ell = 2$)
- **ESM-C**: enabled (`-ESM`); ESM-C 600M weights (`esmc_600m_2024_12_v0`); embedding dimension 1280
- **ESM mode**: `both` (sequence- and residue-level embeddings concatenated, `-esm_mode both`)
- **Residue pooling**: `broadcast` (`-esm_residue_pool broadcast`)
- **ESM fusion**: attention-based (`-esm_attention_fusion`)

E.9 Distributed Training

Training is performed using Distributed Data Parallel (DDP) with:

- **GPUs**: 8 H100 GPUs on the Jean-Zay supercomputing unit
- **Batch normalisation**: synchronised across devices
- **Chunked loss computation**: chunk size of 100 atoms per batch for memory efficiency

F MEMORY USAGE

Table 3: Per-epoch CUDA memory footprint during training phase of strongest DUD-E model variant: 1gc3tr_stack_halves_beta1.0_esm_attn_dude (8 GPUs, 505 epochs, 4,040 epoch-rank samples). Values in GiB.

Metric	Min	Median	Mean	Max
Peak allocated (per epoch)	11.53	22.94	22.23	28.76
Allocated (end of epoch)	0.49	0.50	0.50	0.50
Reserved (end of epoch)	0.59	0.89	1.10	2.18

G ABLATION STUDIES

Table 4: DBAASP dataset solvation prediction ablation study. Best validation MSE (voxel-level) is reported in sign-preserving log space ($\text{egist}(x) = \text{sgn}(x)\log(1+|x|)$), the training objective; lower is better. The **paper model** is the best model found in DBAASP ablations. Each row names only the change(s) relative to the Paper model config. $\Delta\% = 100 \times (\text{MSE}_{\text{paper}} - \text{MSE})/\text{MSE}_{\text{paper}}$. Positive $\Delta\%$ means lower MSE than the paper model.

Variant (vs. paper model)	**val. MSE**	**best_epoch**	**$\Delta\%$**
Paper model (1 GC + 3 Tr)	**0.003743**	**110**	**0.0**
2 GC + 2 Tr	0.003813	115	-1.9
5 GC, no Tr	0.003819	120	-2.0
3 GC + 1 Tr	0.003827	95	-2.2
4 GC, +`grid_mean`, no ESM-attn	0.003855	130	-3.0
4 Tr, no ESM-attn	0.003859	90	-3.1
4 GC, no ESM-attn	0.003880	130	-3.7
2 Tr + 2 GC (Tr first)	0.003927	70	-4.9
2 GC + 3 Tr	0.003933	100	-5.1
4 Tr	0.003995	45	-6.7
4 Tr, +`grid_mean`, no ESM-attn	0.004001	90	-6.9
4 GC	0.004007	60	-7.1
4 Tr, `mn3000`, no ESM-attn	0.004012	90	-7.2
4 Tr, `mn9000`, no ESM-attn	0.004012	90	-7.2
2 GC + 2 Tr, +`grid_mean`, no ESM-attn	0.004072	85	-8.8
5 Tr, no GC	0.004115	80	-9.9
3 Tr + 1 GC (Tr first)	0.004207	40	-12.4
4 Tr, full feature preset, no ESM-attn	0.004272	35	-14.1
1 Tr + 3 GC (Tr first)	0.004298	170	-14.8
4 Tr, no ESM-C input	0.004428	30	-18.3
4 Tr, no wloss	0.005515	85	-47.3
4 Tr, no halves/β/wloss	0.005572	75	-48.9
4 Tr, no wloss, no ESM-attn	0.005749	50	-53.6
4 GC, no wloss, no ESM-attn	0.005920	125	-58.2
4 GC, no β/wloss, no ESM-attn	0.005994	80	-60.1
4 Tr, abs-weighted loss, no ESM-attn	0.007518	50	-100.9

All runs in Tables G and 5 used `ESpatience = 50`, `min_epochs = 50`, `scheduler = cosine`, `batch_size = 1` (x 8 GPUs), `max_neighbours (mn) = 6000`, `learnable_edges`, `nhidden = 512,512`, `nedgeout = 768`.

H EVALUATION METRICS

We evaluate EquiSolv using two complementary approaches: regression metrics for assessing the accuracy of eGIST predictions, and classification metrics for evaluating the identification of strongly bound water sites. Precision, recall, and F1 are computed per protein and then averaged across the test set; reported uncertainties are the standard deviation across proteins

Table 5: DUD-E solvation prediction ablation study. Best validation MSE (voxel-level) is reported in sign-preserving log space (egist$(x) = \mathrm{sgn}(x)\log(1+|x|)$), the training objective; lower is better. The **paper model** is the best model found in DBAASP ablations with a small improvement (+0.3%) from removing ESM-C input ($\mathrm{MSE}_{\mathrm{paper}} = 0.010067$ at epoch 410). $\Delta\% = 100 \times (\mathrm{MSE}_{\mathrm{paper}} - \mathrm{MSE})/\mathrm{MSE}_{\mathrm{paper}}$. Positive $\Delta\%$ means lower MSE than the paper model.

Variant (vs. paper model)	val. $\mathrm{MSE}_{\log}$	best_epoch	$\Delta\%$
Paper Model no ESM-C input	**0.010067**	**410**	+0.3
DBAASP config (1 GC + 3 Tr)	0.010093	435	0.0
full feature preset	0.010097	410	−0.04
2 GC + 2 Tr	0.010112	250	−0.2
+grid_mean	0.010208	245	−1.1
4 GC	0.010221	385	−1.3
no ESM-attn (concat)	0.010244	250	−1.5
5 GC, no Tr	0.010300	410	−2.1
+coars3	0.010972	105	−8.7
no wloss (halves, β, ESM-attn)	0.014373	455	−42.4

Regression Metrics. The primary task is to predict GIST energy values at grid points. We compute mean squared error (MSE) in log-transformed space as the training objective, and root mean squared error (RMSE) in real space (kcal/mol/Å^3) after inverse transformation. To assess prediction accuracy at different energy scales, we report the percentage of predictions within error thresholds of 0.1, 0.5, 1.0, and 1.5 kcal/mol/Å^3 of the true GIST values. These metrics are computed across all grid points for each protein, then averaged across the test set with standard deviations reported. The main results are the baseline comparisons shown in Table 2, we further show here the same comparison but with 0.5 Å truth grids used (native to DeepGIST training) (Table 6). Here, the two 1.0 Å models use KD-tree matching to find the nearest point on the truth grid within 0.51 Å distance (though this typically resolves to a distance of 0 as the lattices are the same).

Table 6: EquiSolv vs. DeepGIST vs. 3D-CNN on 110 DBAASP test proteins, real space. Each model uses native grid spacing (0.5 Å for DG, 1.0 Å for others). DeepGIST column reproduces the previous table to three decimals (pipeline unchanged); only the EquiSolv column and per-protein wins differ from the pre-refresh numbers. **Bold** = best per row. Wins list the number of proteins (out of 110) on which each model achieves the best value for that metric.

Metric	EquiSolv	DeepGIST	3D-CNN	Wins (Eq / DG / CNN)
All region (every KD-matched, in-range voxel)				
Pearson ↑	0.523 ± 0.053	$\mathbf{0.568 \pm 0.103}$	0.065 ± 0.013	41 / 69 / 0
NRMSE ↓	$\mathbf{0.888 \pm 0.038}$	2.069 ± 0.192	3.851 ± 1.209	110 / 0 / 0
RMSE ↓	$\mathbf{0.215 \pm 0.031}$	0.723 ± 0.018	0.684 ± 0.174	110 / 0 / 0
R^2 linfit ↑	0.276 ± 0.055	$\mathbf{0.333 \pm 0.121}$	0.004 ± 0.002	41 / 69 / 0
Non-bulk region ($\lvert truth - bulk \rvert > 0.1$)				
Pearson ↑	$\mathbf{0.582 \pm 0.049}$	0.579 ± 0.102	0.152 ± 0.031	59 / 51 / 0
NRMSE ↓	$\mathbf{0.851 \pm 0.037}$	1.240 ± 0.047	1.256 ± 0.164	110 / 0 / 0
RMSE ↓	$\mathbf{0.335 \pm 0.028}$	0.915 ± 0.055	0.631 ± 0.100	110 / 0 / 0
R^2 linfit ↑	0.341 ± 0.057	$\mathbf{0.345 \pm 0.123}$	0.024 ± 0.010	59 / 51 / 0

Table 7: Area Under the ROC Curve (GIST < -1.0 kcal/mol/Å^3).

Model	AUC-ROC
EquiSolv (1gc3tr)	$\mathbf{0.966 \pm 0.017}$
GW-CNN	0.883 ± 0.029
DeepGIST	0.876 ± 0.057

Classification Metrics. For the application task of identifying strongly bound water molecules, we evaluate classification performance using precision, recall, and F1-score. A water site is considered strongly bound if its GIST energy is below -1.0 kcal/mol/Å^3. Predictions are matched to ground truth sites using spatial matching with a distance tolerance of 1.5 Å, employing a greedy one-to-one matching algorithm to avoid double-counting. True positives (TP) are predicted strongly bound sites within 1.5 Å of a true strongly bound site; false positives (FP) are predicted sites without a nearby true site; and false negatives (FN) are true sites without a nearby prediction. Precision is defined as TP/(TP+FP), recall as TP/(TP+FN), and F1-score as the harmonic mean of precision and recall.

To ensure fair comparison between EquiSolv and GW-CNN, we employ a top-K evaluation strategy where both models are compared when making the same number of predictions (K = 100, 500, 1000, 2000, 5000). This controls for the number of predictions made and allows direct comparison of precision and recall at matched prediction counts. Both models are evaluated against the same ground truth threshold (GIST $<$ -1.0 kcal/mol/Å^3) to ensure consistency.

Detailed metric definitions and statistical analysis procedures are provided below.

H.1 Primary Metrics

All metrics are computed after inverse transformation from log-space to real-space:

$$\text{eGIST}_{\text{real}} = \text{sign}(\text{eGIST}_{\text{log}}) \cdot (e^{|\text{eGIST}_{\text{log}}|} - 1).$$

Mean Squared Error (MSE): Computed in log-space as the primary training objective:

$$\text{MSE}_{\text{log}} = \frac{1}{|\mathcal{V}|} \sum_{v \in \mathcal{V}} \left(\hat{f}_{\text{log}}(r_v) - \mathcal{T}_{\text{log}}[v] \right)^2,$$

where $\mathcal{V}$ is the set of valid grid points and $\hat{f}_{\text{log}}, \mathcal{T}_{\text{log}}$ are log-transformed predictions and targets.

H.2 Per-Atom Metrics

For each atom a, we compute local error statistics over its neighbourhood $\mathcal{N}_a$:

Root Mean Squared Error (RMSE) per atom:

$$\text{RMSE}_a = \sqrt{\frac{1}{|\mathcal{N}_a|} \sum_{v \in \mathcal{N}_a} \left(\hat{f}_{\text{real}}(r_v) - \mathcal{T}_{\text{real}}[v] \right)^2}.$$

Mean Absolute Error (MAE) per atom:

$$\text{MAE}_a = \frac{1}{|\mathcal{N}_a|} \sum_{v \in \mathcal{N}_a} \left| \hat{f}_{\text{real}}(r_v) - \mathcal{T}_{\text{real}}[v] \right|.$$

Percentage accuracy: Fraction of atoms with RMSE below threshold $\tau = 0.7$:

$$\text{Accuracy} = \frac{1}{|P|} \sum_{a \in P} \mathbf{1}[\text{RMSE}_a < \tau] \times 100\%,$$

where P is the set of all atoms.

H.3 Aggregation Procedures

- **Protein-level aggregation**: Metrics are first computed per-protein, then averaged across the test set
- **Grid point filtering**: Only grid points within $[r_{\min}, r_{\max}]$ distance from any atom are included
- **Invalid value handling**: NaN values from metric computations (e.g., division by zero) are excluded from aggregation
- **Per-atom statistics**: RMSE and MAE are computed per-atom, then statistics (mean, median, percentiles) are reported across all atoms

I CLASSIFICATION PERFORMANCE

I.1 TOP-K COMPARISON ON DBAASP

To fairly compare EquiSolv with GW-CNN for the task of identifying strongly bound water sites, we employ a top-K evaluation strategy that controls for the number of predictions made. Both models are evaluated against the same ground truth (GIST $<$ -1.0 kcal/mol/Å^3) at matched prediction counts. Results are presented in Table 8. EquiSolv (1gc3tr) beats GW-CNN at every operating point, with unanimous per-protein F1 wins at $K \leq 2000$ (108-109 wins out of 109). The advantage is largest at small K (F1 gap $\approx +0.14$ at K=100, $+0.34$ at K=500, $+0.27$ at K=1000), where EquiSolv's energy-ranked predictions are sharply more precise. At K=5000 both models have saturated recall (≥ 0.99) and performance converges.

Table 8: Classification performance comparison at matched prediction counts (top-K), using EquiSolv 1gc3tr model against GW-CNN. Models are evaluated against ground truth (GIST < -1.0 kcal/mol/Å^3), spatial match tolerance 1.5 Å. Per-protein wins by F1 score (out of 109 molecules with GW-CNN predictions).

K	**Model**	Precision	Recall	F1	Wins (F1)
100	EquiSolv	$\mathbf{0.917 \pm 0.047}$	$\mathbf{0.168 \pm 0.072}$	$\mathbf{0.278 \pm 0.080}$	109 / 0
	GW-CNN	0.448 ± 0.051	0.082 ± 0.031	0.135 ± 0.036	
500	EquiSolv	$\mathbf{0.723 \pm 0.103}$	$\mathbf{0.631 \pm 0.133}$	$\mathbf{0.655 \pm 0.068}$	109 / 0
	GW-CNN	0.353 ± 0.060	0.309 ± 0.072	0.319 ± 0.030	
1000	EquiSolv	$\mathbf{0.517 \pm 0.116}$	$\mathbf{0.871 \pm 0.092}$	$\mathbf{0.634 \pm 0.088}$	109 / 0
	GW-CNN	0.297 ± 0.054	0.515 ± 0.098	0.367 ± 0.038	
2000	EquiSolv	$\mathbf{0.296 \pm 0.087}$	$\mathbf{0.972 \pm 0.031}$	$\mathbf{0.445 \pm 0.101}$	108 / 0
	GW-CNN	0.234 ± 0.049	0.795 ± 0.100	0.355 ± 0.055	
5000	EquiSolv	$\mathbf{0.122 \pm 0.038}$	$\mathbf{0.996 \pm 0.007}$	$\mathbf{0.215 \pm 0.060}$	46 / 21
	GW-CNN	0.120 ± 0.036	0.991 ± 0.018	0.213 ± 0.057	

I.2 TOP-K COMPARISON ON DUD-E

We compare EquiSolv and GW-CNN on the DUD-E proteins.

Both models evaluated against ground truth (GIST < -1.0 kcal/mol/Å^3), spatial match tolerance 1.5 Å. Per-protein wins by F1 score (out of 10 DUD-E proteins).

Table 9: Classification performance comparison at matched prediction counts (top-K), EquiSolv (1gc3tr no-ESMC) vs GW-CNN on DUD-E. Models evaluated against ground truth (GIST < -1.0 kcal/mol/Å^3).

K	**Model**	Precision	Recall	F1	Wins (F1)
100	EquiSolv	$\mathbf{0.891 \pm 0.044}$	$\mathbf{0.022 \pm 0.007}$	$\mathbf{0.042 \pm 0.013}$	10 / 0
	GW-CNN	0.484 ± 0.088	0.012 ± 0.004	0.023 ± 0.007	
500	EquiSolv	$\mathbf{0.817 \pm 0.037}$	$\mathbf{0.098 \pm 0.029}$	$\mathbf{0.174 \pm 0.044}$	10 / 0
	GW-CNN	0.457 ± 0.034	0.056 ± 0.020	0.098 ± 0.030	
1000	EquiSolv	$\mathbf{0.772 \pm 0.042}$	$\mathbf{0.185 \pm 0.052}$	$\mathbf{0.294 \pm 0.059}$	10 / 0
	GW-CNN	0.440 ± 0.027	0.107 ± 0.036	0.169 ± 0.042	
2000	EquiSolv	$\mathbf{0.704 \pm 0.048}$	$\mathbf{0.336 \pm 0.084}$	$\mathbf{0.446 \pm 0.057}$	10 / 0
	GW-CNN	0.411 ± 0.020	0.198 ± 0.061	0.262 ± 0.045	
5000	EquiSolv	$\mathbf{0.556 \pm 0.070}$	$\mathbf{0.650 \pm 0.103}$	$\mathbf{0.589 \pm 0.030}$	10 / 0
	GW-CNN	0.360 ± 0.030	0.428 ± 0.101	0.384 ± 0.020	

J DBAASP MOLECULE SIZE CORRELATION

Molecules split into terciles by evaluated voxel count (proxy for molecule size). Performance at fixed K=1000. DeepGIST predictions are trilinear-interpolated from each molecule's location onto the 1gc3tr voxel sites and ranked by predicted GIST; same truth and threshold as the other two columns. For DUD-E, the number of test proteins was too small to obtain meaningful percentiles.

Table 10: Top-K (K=1000) performance by molecule size tercile.

			EquiSolv			**GW-CNN**			**DeepGIST**		
Tercile	$\bar{N}_{\text{vox}}$	$\bar{N}_{\text{true}}$	P	R	F1	P	R	F1	P	R	F1
Small (n=37)	8 063	542	**.471**	**.898**	**.597**	.285	.564	.365	.388	.752	.495
Medium (n=36)	10 153	603	**.519**	**.880**	**.643**	.295	.507	.367	.427	.731	.531
Large (n=36)	14 085	687	**.559**	**.835**	**.660**	.311	.471	.369	.465	.700	.551

EquiSolv (1gc3tr) outperforms both baselines across every size tercile at K=1000. F1 gap to GW-CNN: $+0.232$ (small), $+0.276$ (medium), $+0.290$ (large); F1 gap to DeepGIST: $+0.102$, $+0.111$, $+0.109$. DeepGIST sits between EquiSolv and GW-CNN in every tercile (e.g. medium: F1 $0.643 > 0.531 > 0.367$). Pearson correlation between voxel count and K=1000 precision: $r = 0.386$ ($p = 3.5 \times 10^{-5}$) for EquiSolv, $r = 0.399$ ($p = 1.7 \times 10^{-5}$) for DeepGIST, $r = 0.270$ ($p = 4.5 \times 10^{-3}$) for GW-CNN. All three models are slightly more precise on larger proteins (where K=1000 covers a smaller fraction of the grid), but EquiSolv's advantage is consistent across the full size distribution, not concentrated in any tercile.

K DBAASP TIMING

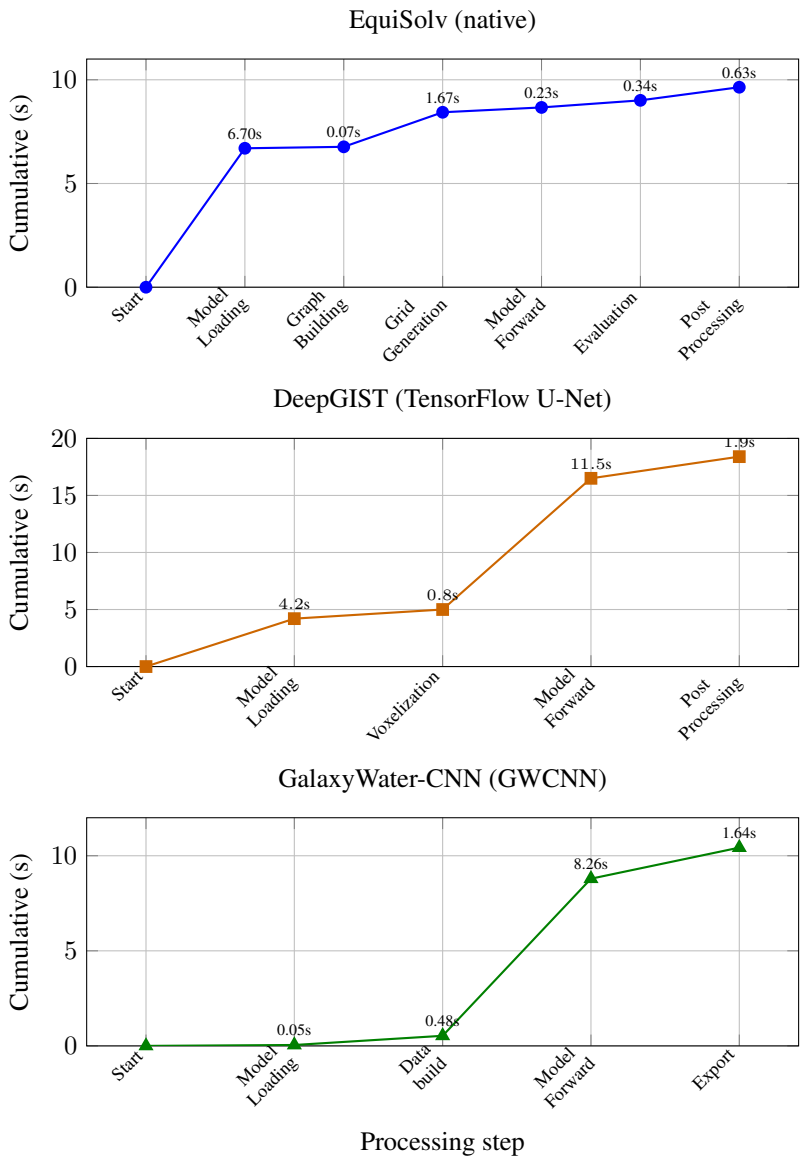


Figure 3: Mean cumulative inference time by pipeline stage (DBAASP dataset). **Top:** EquiSolv; **Middle:** DeepGIST; **Bottom:** GWCNN GPU-batch mode.

L BASELINE COMPARABILITY

Table 11: What each baseline predicts, and how it is placed on the common 1.0 Å eGIST grid. Head-to-head RMSE uses only the shared enthalpic quantity.

Method	**Target**	**Backbone / input**	**On the 1.0 Å grid**
EquiSolv	$E_{SW} + 2E_{WW}$	equivariant GNN, atom graph + ESM-C	identity (training grid)
DeepGIST	native ΔG_W; scored vs eGIST truth	3D U-Net, voxel grid	mean-aggregate from 0.5 Å
3D-CNN	$E_{SW} + 2E_{WW}$	3D U-Net, depth 4, 64 channels	identity
GW-CNN	water-binding score, not an energy	3D residual CNN	score sampled; AUC only

M HELD-OUT DUD-E PROTEINS

Table 12 lists the ten held-out DUD-E proteins. Sequence identity is the maximum identity to a training-set protein. RMSE does not follow that identity: SRC and ADRB1, the closest pairs, sit in the same RMSE band as the more distant targets. Per-protein RMSE spans 0.285-0.318. It is uncorrelated with residue count ($r = -0.13$, $p = 0.71$) and with voxel count ($r = +0.01$, $p = 0.97$).

Table 12: Held-out DUD-E proteins, sorted by maximum sequence identity to the training set. RMSE and Pearson are real-space eGIST errors.

Target	**Family**	**Res.**	**Voxels**	**ID (%)**	**RMSE**	**Pearson**	**Closest train**
src	kinase	233	53744	64.8	0.285	0.448	lck
adrb1	GPCR	276	58818	63.0	0.294	0.457	adrb2
igf1r	kinase (RTK)	256	57797	42.2	0.318	0.412	abl1
fabp4	lipid-binding	131	30919	39.7	0.306	0.395	pgh2
nos1	oxidoreductase	355	65698	38.2	0.307	0.442	xiap
pnph	other enzyme	244	48740	36.4	0.293	0.434	fkb1a
mp2k1	kinase	312	60771	36.4	0.313	0.420	fkb1a
tgfr1	kinase	260	60877	36.2	0.318	0.380	braf
tysy	other enzyme	311	57771	35.5	0.289	0.432	fkb1a
pygm	other enzyme	378	68918	32.7	0.290	0.447	fkb1a

N TAIL, DISTANCE, AND CHARGE ERRORS

The L_∞ column of Table 1 averages per-protein maxima of an energy density (kcal/mol/Å^3). It is not a binding affinity and does not enter $\Delta G_d = -RT \ln K_d$. The energy of a region is the integral of that density, so one voxel contributes $\text{error} \times \Delta V$. On the 1.0 Å grid, $\Delta V = 1$ Å^3 and only that cell is counted. The DUD-E mean L_∞ near 15.9 kcal/mol/Å^3 is the worst cell among about 3-7 $\times 10^4$ voxels (Appendix M). On DBAASP the pooled 99th-percentile absolute error is 0.654 kcal/mol/Å^3 (worst voxel 9.943); the DUD-E 99th percentile, averaged over the ten test proteins, is 1.000.

Table 13 pools absolute errors over the 110 DBAASP test molecules (1,183,966 voxels). Table 14 assigns each voxel to its distance to the nearest atom. Table 15 labels a voxel as charged when that nearest atom belongs to R, K, H, D, or E. Errors fall with distance. They are not higher next to ionisable residues (RMSE 0.164 charged vs. 0.186 neutral).

Table 13: Pooled absolute error on the DBAASP test voxels. Real-space units are kcal/mol/Å^3.

Space	RMSE	99th-pct AE	max AE
log	0.099	0.396	1.728
real	0.177	0.654	9.943

Table 14: Real-space absolute error by distance from the nearest atom (DBAASP test).

Distance	Voxels	RMSE	99th-pct AE	max AE
0-4 Å	240450	0.335	0.978	9.943
4-7 Å	590434	0.124	0.422	2.789
7-10 Å	353082	0.050	0.211	0.694

Table 15: Real-space absolute error by nearest-atom charge class (DBAASP test).

Environment	Voxels	RMSE	99th-pct AE	max AE
charged (R, K, H, D, E)	513500	0.164	0.631	9.943
neutral	670466	0.186	0.672	8.425