# VN-EGNN: Equivariant Graph Neural Networks with Virtual Nodes Enhance Protein Binding Site Identification

**Florian Sestak**[1] **Lisa Schneckenreiter**[1] **Sepp Hochreiter**[1] **Andreas Mayr**[1]

**Günter Klambauer**[1]

[1]ELLIS Unit Linz & LIT AI Lab, Institute for Machine Learning,
Johannes Kepler University Linz, Austria
{sestak,schneckenreiter,hochreit,mayr,klambauer}@ml.jku.at

## Abstract

Being able to identify regions within or around proteins, to which ligands can potentially bind, is an essential step to develop new drugs. Binding site identification methods can now profit from the availability of large amounts of 3D structures in protein structure databases or from AlphaFold predictions. Current binding site identification methods rely on geometric deep learning, which takes geometric invariances and equivariances into account. Such methods turned out to be very beneficial for physics-related tasks like binding energy or motion trajectory prediction. However, their performance at binding site identification is still limited, which might be due to limited expressivity or oversquashing effects of E(n)-Equivariant Graph Neural Networks (EGNNs). Here, we extend EGNNs by adding virtual nodes and applying an extended message passing scheme. The virtual nodes in these graphs both improve the predictive performance and can also learn to represent binding sites. In our experiments, we show that VN-EGNN sets a new state of the art at binding site identification on three common benchmarks, COACH420, HOLO4K, and PDBbind2020.

## 1 Introduction

**Binding site identification remains a central computational problem in drug discovery.** With the advent of AlphaFold (Jumper et al., 2021), millions of 3D structures of proteins have been unlocked for further investigation by the scientific community (Tunyasuvunakool et al., 2021; Cheng et al., 2023). Information about the 3D structure of a protein can provide crucial information about its function. One of the most important fields that should profit from these 3D structures, is drug discovery (Ren et al., 2023; Sadybekov and Katritch, 2023). It has been envisioned that the availability of 3D structures will allow to purposefully design drugs that alter protein function in a desired way. However, to enable this structure-based drug design, further computational approaches have to be utilized/employed, concretely either *docking* or *binding site identification* methods (Lengauer and Rarey, 1996; Cheng et al., 2007; Halgren, 2009). While docking approaches predict the location of a specific small molecule, called ligand, within a protein's active site upon binding, binding site identification aims at finding regions on the protein likely to form a binding pocket and interact with unknown ligands (Schmidtke and Barril, 2010). For both approaches, deep learning methods, and specifically geometric deep learning have brought significant advances (Méndez-Lucio et al., 2021;

The NeurIPS 2023 Workshop on New Frontiers of AI for Drug Discovery and Development (AI4D3 2023), New Orleans, LA, USA, 2023.

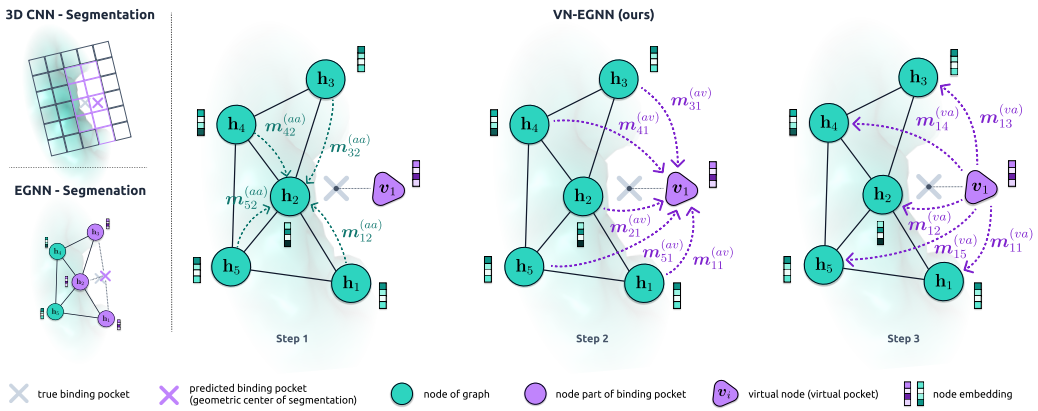


Figure 1: Overview of binding site identification methods. **Top Left**: Traditional methods, based on segmentation of a voxel grid, in which the pocket center is calculated as the geometric center of the positively labeled voxels. **Bottom Left:** Geometric Deep Learning approaches, such as EGNN, in which the pocket center is calculated as the geometric center of the positively labeled nodes. **Right:** VN-EGNN approach (ours): the predicted binding site center is the position of the virtual node after $L$ message passing layers.

Lu et al., 2022; Corso et al., 2023). In this work, we aim at improving binding site identification through geometric deep learning methods.

**Methods to identify binding sites.** The identification of binding sites relies on the successful combination of physical, chemical and geometric information. Initially, machine learning methods for binding site prediction were based on carefully designed input features due to their tabular processing structure. For instance, FPocket (Le Guilloux et al., 2009) relies on Voronoi tessellation and alpha spheres (Liang et al., 1998) and additionally takes an electronegativity criterion into account. A random forest based method, which makes use of the protein surface, is P2Rank (Kriv´ak and Hoksza, 2018). With the advent of end-to-end deep learning and especially with the breakthrough of convolutional (Lecun et al., 1998) and graph neural networks (GNNs) (Scarselli et al., 2009; Defferrard et al., 2016; Kipf and Welling, 2017; Gilmer et al., 2017; Satorras et al., 2021), the construction of input features can be learned which helped to advance predictive quality. For instance, DeepSite (Jiménez et al., 2017) is a voxel-based 3D convolutional neural network for binding site prediction. Convolutional operations on the 3D space are, however, computationally very demanding and so quickly other approaches to tackle binding site identification were developed, e.g., DeepSurf (Mylonas et al., 2021) or PointSite (Yan et al., 2022). DeepSurf operates on surface-based representations and places several voxelized grids on the protein's surface, while PointSite is based on a form of sparse convolutions to reduce the computational overhead and keep sparse regions in the 3D space to be sparse. Typical convolutional networks, however, do not perform well at binding site identification likely because of the irregularity of protein structures and due to the fact that proteins may be arbitrarily rotated and shifted in space (Zhang et al., 2023). To overcome these issues, EquiPocket (Zhang et al., 2023) uses E(n)-Equivariant Graph Neural Networks (EGNNs) (Satorras et al., 2021) on a protein surface graph. Here, we follow the approach of EquiPocket to use EGNNs for identifying binding pockets. Although EGNNs are prime candidates for this task, they exhibit poor performance at binding site identification (Zhang et al., 2023), which might be due to a) their limited expressivity (Joshi et al., 2023) or b) their inability to learn to represent binding sites. We aim at alleviating both problems by extending EGNNs with so-called virtual nodes.

**Virtual nodes have improved message-passing neural networks (MPNNs) w.r.t. expressive power.** Virtual nodes, sometimes called super-nodes or supersource-nodes, that are introduced into a message-passing scheme and connected to all other nodes, have been used and investigated in several works. In a benchmark setting Hu et al. (2020) showed that adding virtual nodes tends to increase the predictive performance. Hwang et al. (2022) provide a theoretical analysis of the benefits of virtual nodes in terms of expressiveness, demonstrate the increased expressiveness of GNNs with virtual nodes and also hint at the fact that such nodes can decrease oversmoothing. Cai et al. (2023) and Cai (2023) show that an MPNN with one virtual node, connected to all nodes, can approximate a

Transformer layer. Low rank global attention (Puny et al., 2020) can be seen as one virtual node, which improves expressiveness. Practically, such nodes have already been suggested in the original work by Gilmer et al. (2017) and they were even mentioned earlier in Scarselli et al. (2009) and used in application areas such as drug discovery (Li et al., 2017; Pham et al., 2017; Ishiguro et al., 2019). Geometric GNNs, such as EGNNs, might as well suffer from limited expressiveness, oversmoothing (Rusch et al., 2023) or similar effects as conventional GNNs. Consequently, Joshi et al. (2023) investigated the power of these networks in greater detail and argue for the case of EGNNs that the method might suffer from oversquashing (Alon and Yahav, 2021; Topping et al., 2022). Alon and Yahav (2021) themselves mention that virtual nodes might have been used as a technique to overcome oversquashing effects. In order to reduce oversquashing effects in the usage of EGNNs for binding site identification, we suggest to extend EGNNs with virtual nodes and introduce an adapted message passing scheme. To the best of our knowledge, this is the first application of virtual nodes to geometric graph networks.

**Virtual nodes in EGNNs can learn representations of unknown physical entities, such as binding sites.** Geometric graph networks, such as EGNNs, usually have coordinate features associated with their node features, but could also have higher order features (Brandstetter et al., 2021). We follow the same approach for the virtual nodes in our virtual node EGNN (VN-EGNN). Since protein graphs usually contain a large number of nodes, we introduce several virtual nodes, which we distribute evenly on a sphere around the protein. Although our network is trained w.r.t. correctly segmenting conventional nodes according to whether they represent a binding pocket atom or not, we empirically observed that many of the virtual nodes indeed seemed to converge towards the actual physical binding positions of ligands on the protein. This gives rise to the assumption that virtual node features might form an abstract representation of the binding site. It would furthermore allow to employ losses directly on the binding site representations, which could in turn lead to improved performance.

**Contributions.** In this work, we contribute the following: **a)** We propose a novel type of graph neural network geared towards the identification of binding sites of proteins. **b)** We demonstrate that the virtual nodes in the message-passing scheme learn useful representations of binding pockets. **c)** We assess the performance of other methods, baselines and our method on benchmarking datasets.

## 2 Equivariant Graph Neural Networks with Virtual Nodes

Since proteins are described by atom coordinate measurements in arbitrary coordinate systems and since the atoms might be quite unevenly distributed in space, it seems attractive to employ geometric deep networks, as these architectures allow predictions that are translation-, rotation-, and permutation-equivariant by design[1]. Especially, we decided to build upon EGNNs (Satorras et al., 2021) due to their simplicity and robustness.

**Notational preliminaries.** We give an overview on variable and symbol notation in Appendix A and a more detailed description and discussion on how we represent proteins and binding sites in Appendices B.1 and B.2 respectively. To quickly summarize, the coordinates of protein atom centers are denoted as $\mathbf{x} \in \mathbb{R}^3$, atom property features as $\boldsymbol{h} \in \mathbb{R}^D$. Proteins are characterized by a neighborhood graph $\mathcal{P}$, where atoms are represented as nodes and edges are drawn between spatially close atoms. We use $\mathcal{N}(i)$ to indicate neighbouring nodes to node $i$ within $\mathcal{P}$ and denote edge features between atoms as $a_{ij}$. Atoms close to binding site centers are assigned a label $y_n = 1$ and $y_n = 0$, otherwise. We assume there are $M$ known binding pockets, which are characterized by their center coordinates $\mathbf{y} \in \mathbb{R}^3$. We denote predicted atom labels by $\hat{y}_n \in [0, 1]$ and predicted binding pocket center points by $\hat{\mathbf{y}} \in \mathbb{R}^3$. Thereby, we predict a fixed number of $K$ center points.

**Application of EGNNs to binding site prediction.** EGNNs are straightforward to apply to our protein graph $\mathcal{P}$ to predict node labels $y_n$, i.e. we may use them to perform a *node-level classification task*. We state the layer-wise ($l$) message passing scheme of EGNNs as given by Satorras et al. (2021) in eqs. (1) to (4), where we initialize $\mathbf{x}_n^0 := \mathbf{x}_n$ (atom positions) and $\boldsymbol{h}_n^0 = \boldsymbol{h}_n$ (atom properties) for each atom $n$, which corresponds to a node in $\mathcal{P}$:

[1]For the purpose of being self-contained, we list these properties in Appendix C.

$$\boldsymbol{m}_{ij} = \boldsymbol{\phi}_e(\boldsymbol{h}_i^l, \boldsymbol{h}_j^l, \|\mathbf{x}_i^l - \mathbf{x}_j^l\|^2, a_{ij}) \quad (1) \qquad \boldsymbol{m}_i = \sum_{j \in \mathcal{N}(i)} \boldsymbol{m}_{ij} \quad (2)$$

$$\mathbf{x}_i^{l+1} = \mathbf{x}_i^l + \frac{1}{|\mathcal{N}(i)|} \sum_{j \in \mathcal{N}(i)} \frac{\mathbf{x}_i^l - \mathbf{x}_j^l}{\|\mathbf{x}_i^l - \mathbf{x}_j^l\|} \phi_x(\boldsymbol{m}_{ij}) \quad (3) \qquad \boldsymbol{h}_i^{l+1} = \boldsymbol{\phi}_h\left(\boldsymbol{h}_i^l, \boldsymbol{m}_i\right) \quad (4)$$

Here $\boldsymbol{\phi}_e$, $\phi_{\mathbf{x}}$ and $\boldsymbol{\phi}_h$ denote multilayer-perceptrons (MLPs). The message passing scheme can be abbreviated as:

$$\left((\mathbf{x}_1^{l+1}, \boldsymbol{h}_1^{l+1}), \ldots, (\mathbf{x}_N^{l+1}, \boldsymbol{h}_N^{l+1})\right) = \text{EGNN}\left((\mathbf{x}_1^l, \boldsymbol{h}_1^l), \ldots, (\mathbf{x}_N^l, \boldsymbol{h}_N^l)\right). \quad (5)$$

We can extract predictions $\hat{y}_n$ for each atom $n$ by a read-out function applied to the last message passing step $L$, i.e., we have: $\hat{y}_n = \sigma(\boldsymbol{w}^\top \boldsymbol{h}_n^L)$ with an activation function $\sigma$ and (shared) parameters $\boldsymbol{w}$.

**VN-EGNN: Extension of EGNN with virtual nodes.** To tackle oversquashing effects, we extend the neighborhood graph $\mathcal{P}$ of our protein with a set of $K$ virtual nodes, which exhibit edges to all other nodes. To be able to process this extended graph, we extend EGNNs by locating the virtual nodes at coordinates $\mathbf{z}_1, \ldots, \mathbf{z}_K \in \mathbb{R}^3$ and associating them with a set of properties $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_K \in \mathbb{R}^D$. The new message passing scheme

$$\begin{aligned}\left((\mathbf{x}_1^{l+1}, \boldsymbol{h}_1^{l+1}), \ldots, (\mathbf{x}_N^{l+1}, \boldsymbol{h}_N^{l+1}), (\mathbf{z}_1^{l+1}, \boldsymbol{v}_1^{l+1}), \ldots, (\mathbf{z}_K^{l+1}, \boldsymbol{v}_K^{l+1})\right) = \\ \text{VN-EGNN}\left((\mathbf{x}_1^l, \boldsymbol{h}_1^l), \ldots, (\mathbf{x}_N^l, \boldsymbol{h}_N^l), (\mathbf{z}_1^l, \boldsymbol{v}_1^l), \ldots, (\mathbf{z}_K^l, \boldsymbol{v}_K^l)\right), \quad (6)\end{aligned}$$

then consists of three phases, in which atom embeddings and atoms' coordinate embeddings are updated twice:

$$\boldsymbol{h}_n^l \to \boldsymbol{h}_n^{l+1/2} \to \boldsymbol{h}_n^{l+1} \qquad \mathbf{x}_n^l \to \mathbf{x}_n^{l+1/2} \to \mathbf{x}_n^{l+1} \qquad \forall n$$

while virtual node embeddings are only updated once per message passing step

$$\boldsymbol{v}_k^l \to \boldsymbol{v}_k^{l+1} \qquad \mathbf{z}_k^l \to \mathbf{z}_k^{l+1} \qquad \forall k.$$

**Message Passing Phase I** between **atoms** (analogous to EGNN):

$$\boldsymbol{m}_{ij}^{(aa)} = \boldsymbol{\phi}_{e^{(aa)}}(\boldsymbol{h}_i^l, \boldsymbol{h}_j^l, \|\mathbf{x}_i^l - \mathbf{x}_j^l\|, a_{ij}) \quad (7) \qquad \boldsymbol{m}_j^{(aa)} = \frac{1}{|\mathcal{N}(j)|} \sum_{i \in \mathcal{N}(j)} \boldsymbol{m}_{ij}^{(aa)} \quad (8)$$

$$\mathbf{x}_j^{l+1/2} = \mathbf{x}_j^l + \frac{1}{|\mathcal{N}(j)|} \sum_{i \in \mathcal{N}(j)} \frac{\mathbf{x}_i^l - \mathbf{x}_j^l}{\|\mathbf{x}_i^l - \mathbf{x}_j^l\|} \phi_{x^{aa}}(\boldsymbol{m}_{ij}^{(aa)}) \quad (9) \qquad \boldsymbol{h}_j^{l+1/2} = \boldsymbol{\phi}_{h^{(aa)}}\left(\boldsymbol{h}_j^l, \boldsymbol{m}_j^{(aa)}\right) \quad (10)$$

**Message Passing Phase II** from **atoms** to **virtual node**:

$$\boldsymbol{m}_{ij}^{(av)} = \boldsymbol{\phi}_{e^{(av)}}(\boldsymbol{h}_i^{l+1/2}, \boldsymbol{v}_j^l, \|\mathbf{x}_i^{l+1/2} - \mathbf{z}_j^l\|, d_{ij}) \quad (11) \qquad \boldsymbol{m}_j^{(av)} = \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{m}_{ij}^{(av)} \quad (12)$$

$$\mathbf{z}_j^{l+1} = \mathbf{z}_j^l + \frac{1}{N} \sum_{i=1}^{N} \frac{\mathbf{x}_i^{l+1/2} - \mathbf{z}_j^l}{\|\mathbf{x}_i^{l+1/2} - \mathbf{z}_j^l\|} \phi_{x^{av}}(\boldsymbol{m}_{ij}^{(av)}) \quad (13) \qquad \boldsymbol{v}_j^{l+1} = \boldsymbol{\phi}_{h^{(av)}}\left(\boldsymbol{v}_j^l, \boldsymbol{m}_j^{(av)}\right) \quad (14)$$

**Message Passing Phase III** from **virtual node** to **atoms**:

$$\boldsymbol{m}_{ij}^{(va)} = \boldsymbol{\phi}_{e^{(va)}}(\boldsymbol{v}_i^{l+1}, \boldsymbol{h}_j^{l+1/2}, \|\mathbf{z}_i^{l+1} - \mathbf{x}_j^{l+1/2}\|, d_{ij}) \quad (15) \qquad \boldsymbol{m}_j^{(va)} = \frac{1}{K} \sum_{i=1}^{K} \boldsymbol{m}_{ij}^{(va)} \quad (16)$$

$$\mathbf{x}_j^{l+1} = \mathbf{x}_j^{l+1/2} + \frac{1}{K} \sum_{i=1}^{K} \frac{\mathbf{z}_i^{l+1} - \mathbf{x}_j^{l+1/2}}{\|\mathbf{z}_i^{l+1} - \mathbf{x}_j^{l+1/2}\|} \phi_{x^{va}}(\boldsymbol{m}_{ij}^{(va)}) \quad (17) \qquad \boldsymbol{h}_j^{l+1} = \boldsymbol{\phi}_{h^{(va)}}\left(\boldsymbol{h}_j^{l+1/2}, \boldsymbol{m}_j^{(va)}\right) \quad (18)$$

Here, $\boldsymbol{\phi}_{e^{(aa)}}, \ldots, \boldsymbol{\phi}_{h^{(va)}}$ are again MLPs. The MLPs $\phi_{\cdot}$ are layer-specific, i.e. $\phi_{\cdot}^{l}$ and currently do not consider edge features $d_{ij}$ and $a_{ij}$. To keep the notation uncluttered, we skipped the layer index $l$ in above formulae.

**Initialization of virtual nodes in VN-EGNN.** The $K$ virtual nodes are initially evenly distributed across a sphere using a Fibonacci grid (see Appendix F), where the radius is defined as the distance between the protein center and its most distant atom. Virtual node property features $\boldsymbol{v}_k$ are initially set to a fixed, but learnable vector $\boldsymbol{c} \in \mathbb{R}^D$.

The following proposition shows, that analogously to EGNNs, VN-EGNNs are equivariant with respect to roto-translations by construction:

**Proposition 1.** *Equivariant graph neural networks with virtual nodes as defined in eqs. (7) to (18) are equivariant with respect to roto-translations of the input and virtual node coordinates.*

*Proof.* See Appendix D. □

**Objective.** As mentioned in Section 1, and as demonstrated by initial computational experiments in Section 3, we could observe, that virtual node locations $\mathbf{z}_1^L, \ldots, \mathbf{z}_K^L$ at the last layer tended to converge towards the observed binding site center region points $\mathbf{y}_1, \ldots, \mathbf{y}_M$, *although* VN-EGNN was trained to only predict the labels $y_n$ correctly (semantic segmentation, for further details see Appendix G). In this setting we used a Dice loss (see Appendix B.4) on the atom-level predictions $\hat{y}_1, \ldots, \hat{y}_N$.

These initial experiments and the thereof following interpretation of virtual node points as binding site region center points, motivated us to extend our method to directly predict the binding site region center points correctly, i.e. we aim for a minimum distance between predicted and experimentally found region centers. Consequently, for further experiments we used a *multi-modal* objective function summing a distance loss, which we denote as binding site center loss, and a semantic segmentation loss. This allows to directly tackle the much more challenging problem to predict binding site region center points and to extract predictions for these points as outputs of the last EGNN layer: $\hat{\mathbf{y}}_k := \mathbf{z}_k^L$ $(1 \leq k \leq K)$.

$$\mathcal{L} = \text{Dist}\left(\{\mathbf{y}_1, \ldots, \mathbf{y}_M\}, \{\hat{\mathbf{y}}_1, \ldots, \hat{\mathbf{y}}_K\}\right) + \alpha \text{Dice}\left((y_1, \ldots, y_N), (\hat{y}_1, \ldots, \hat{y}_N)\right) \tag{19}$$

where $\alpha > 0$ is a hyperparameter and which is further detailed in Appendix B.4.

## 3 Experiments

### 3.1 Datasets

**scPDB (Desaphy et al., 2015)** is a frequently utilized dataset for binding site prediction (Kandel et al., 2021; Stepniewska-Dziubinska et al., 2020), encompassing both protein and ligand structures. We employed the 2017 release of scPDB in the training and validation. This release comprises 17,594 structures, 16,034 entries, 4,782 proteins, and 6,326 ligands.

**PDBbind (Wang et al., 2004)** is a widely recognized dataset integral to the study of protein-ligand interactions. This dataset provides detailed 3D structural information of proteins, ligands, and their respective binding sites, complemented by rigorously determined binding affinity values derived from laboratory evaluations. For our work, we draw upon the v2020 edition, which is divided into two sets: the general set (comprising 14,127 complexes) and the refined set (containing 5,316 complexes). While the general set encompasses all protein-ligand interactions, only the refined set, curated for its superior quality from the general collection, is used in our experiments.

**COACH420** and **HOLO4K** are benchmark datasets utilized for the prediction of binding sites, as originally detailed by Krivák and Hoksza (2018). Following the methodologies of Krivák and Hoksza (2018); Mylonas et al. (2021); Aggarwal et al. (2022), we adopt the so-called `mlig` subsets from each of these datasets, which encompass the significant ligands pertinent to binding site prediction.

### 3.2 Evaluation Setup

**Dataset split.** In our experimental framework, we utilized the scPDB dataset for training, with 10 % reserved explicitly for validation and hyperparameter selection. Testing was subsequently conducted on the COACH420, HOLO4K and PDBbind datasets.

**Metrics.** In assessing our experimental outcomes, we leveraged the **DCC** and **DCA** metrics, which are well-established in the literature (see e.g., Chen et al., 2011). The DCC is the distance between the predicted and actual binding site centers, whereas the DCA is the shortest distance between the predicted binding site center and any atom of the ligand. Following the criteria used in Stepniewska-Dziubinska et al. (2020) and Zhang et al. (2023), predictions with a DCC/DCA below a certain threshold are considered successful, which is commonly referred to as the *DCC/DCA success rate*. Adhering to these works, we maintained a threshold of 4Å throughout our experiments.

**Methods compared.** We benchmark our framework against various models spanning over different categories: *Geometry-based*: Fpocket. *CNN-based*: DeepSite, Kalasanty, DeepSurf. *Topological graph-based*: GAT, GCN, GCN2. *Spatial graph-based*: SchNet, EGNN, EquiPocket. Figure 1 provides a comparative overview, illustrating the distinctions between previous methods and our approach.

### 3.3 Experimental Settings

**Pre-processing.** For the scPDB dataset, structures were clustered based on their Uniprot IDs. From each cluster, protein structures with the longest sequences were selected, in alignment to the strategies used in Krivák and Hoksza (2018) and Zhang et al. (2023). For comprehensive data preparation across all datasets, solvent atoms were excluded, and any absent hydrogen atoms were added back. Erroneous structures were removed.

**Graph Features.** We employed an atom graph representation, incorporating only those atoms that are within a 2Å distance to the protein solvent accessible surface. The surface calculations were conducted using the MSMS software (Sanner et al., 1996). For the nodal features of the atom nodes, we focused on three specific feature types: the atom type, the associated residue type of the atom, and the distance of the atom to the computed surface. For the characterization of atom and residue types, we utilized learned embeddings.

**Post-Processing.** Our final model architecture included 8 virtual nodes. However, upon observation, certain virtual nodes exhibited convergence to similar positions. To cluster these spatially similar virtual nodes, we applied the Mean-Shift algorithm (Comaniciu and Meer, 2002), yielding an average of 5 predictions per protein.

**Implementation Details.** We optimized our model using the AdamW optimizer (Loshchilov and Hutter, 2017) over 260 epochs, with a batch size of 64. The best-performing model was selected based on its performance on the validation set. Each multi-layer perceptron $\phi_{e^{(aa)}}, \ldots, \phi_{h^{(va)}}$ in our model has two linear layers, a 0.1 dropout rate (Hinton et al., 2012), and a SiLU activation function (Elfwing et al., 2018). $\phi_{x^{va}}$ is special because it contains a learnable scaling parameter $\lambda$ that is multiplied with the output node of the MLP. We set the node feature size to 30 and the message size to 50. Layernorm (Ba et al., 2016) was used to normalize the feature representations during message passing. The number of virtual nodes was set to 8. An MLP was employed on the learned embeddings, to align them with the node feature size. Subsequently, a read out function was applied to the atom feature nodes, producing the final predictions for the segmentation. All hyperparameters tested can be found in Table E1 (Appendix E).

### 3.4 Results

In Table 1, we present the *success rates* for the DCC and DCA metrics across the benchmark datasets COACH420, HOLO4K, and PDBbind. Overall, our method outperformed all previous methods. Our empirical findings highlight that for the **COACH420** and **HOLO4K** datasets, in terms of the DCC metric, our method significantly outperforms our main baselines DeepSurf and EquiPocket. Also for the DCA metric, our model shows a notable improvement for these datasets. For the **PDBbind** dataset, the advantage in the DCC metric is less pronounced. Yet, within the PDBbind context, our model achieves a notably improved score in the DCA metric compared to the EquiPocket baseline.

Table 1: Performance at binding site identification in terms of DCC and DCA success rates.[a]

| Methods | Param | COACH420 | | HOLO4K | | PDBbind2020 | |
|---|---|---|---|---|---|---|---|
| | (M) | DCC↑ | DCA↑ | DCC↑ | DCA↑ | DCC↑ | DCA↑ |
| Fpocket (Le Guilloux et al., 2009)[b] | \ | 0.228 | 0.444 | 0.192 | 0.457 | 0.253 | 0.371 |
| DeepSite (Jiménez et al., 2017)[b] | 1.00 | \ | 0.564 | \ | 0.456 | \ | \ |
| Kalasanty (Stepniewska-Dziubinska et al., 2020)[b] | 70.64 | 0.335 | 0.636 | 0.244 | 0.515 | 0.416 | 0.625 |
| DeepSurf (Mylonas et al., 2021)[b] | 33.06 | 0.386 | 0.658 | 0.289 | 0.635 | 0.510 | 0.708 |
| GAT (Veličković et al., 2017)[b] | **0.03** | 0.039(0.005) | 0.130(0.009) | 0.036(0.003) | 0.110(0.010) | 0.032(0.001) | 0.088(0.011) |
| GCN (Kipf and Welling, 2017)[b] | 0.06 | 0.049(0.001) | 0.139(0.010) | 0.044(0.003) | 0.174(0.003) | 0.018(0.001) | 0.070(0.002) |
| GAT + GCN[b] | 0.08 | 0.036(0.009) | 0.131(0.021) | 0.042(0.003) | 0.152(0.020) | 0.022(0.008) | 0.074(0.007) |
| GCN2 (Chen et al., 2020)[b] | 0.11 | 0.042(0.098) | 0.131(0.017) | 0.051(0.004) | 0.163(0.008) | 0.023(0.007) | 0.089(0.013) |
| SchNet (Schütt et al., 2017)[b] | 0.49 | 0.168(0.019) | 0.444(0.020) | 0.192(0.005) | 0.501(0.004) | 0.263(0.003) | 0.457(0.004) |
| EGNN (Satorras et al., 2021)[b] | 0.41 | 0.156(0.017) | 0.361(0.020) | 0.127(0.005) | 0.406(0.004) | 0.143(0.007) | 0.302(0.006) |
| EquiPocket (Zhang et al., 2023)[b] | 1.70 | 0.423(0.014) | 0.656(0.007) | 0.337(0.006) | 0.662(0.007) | 0.545(0.010) | 0.721(0.004) |
| VN-EGNN (ours) | 0.21 | **0.526(0.016)** | **0.677(0.011)** | **0.498(0.004)** | **0.713(0.006)** | **0.569(0.014)** | **0.795(0.011)** |

[a] The standard deviation across training re-runs is indicated in parentheses. [b] Results from Zhang et al. (2023).

This suggests our model's adeptness at identifying the ligand's approximate position, even if it does not pinpoint the exact central binding site, which is to be expected in this setting in which the ligand is unknown.

## 4 Discussion

In the application area of protein binding site identification, we have extended the EGNN approach with virtual nodes which alleviates the limited expressivity and the inability to learn representations of unknown physical entities. We evaluated its performance using three prominent datasets: COACH420, HOLO4K, and PDBbind, on which our method set a new state-of-the-art. To the best of our knowledge, prior methods have determined the binding site center solely based the geometric center, calculated from segmented parts of the protein surface. In contrast, our VN-EGNN approach directly represents the **binding site centers as virtual nodes**.

**Limitations.** While our model offers predictions of the binding pockets without a specific ranking, on average five binding pockets are predicted per protein. We believe this number maintains a balance between offering a reasonable amount of predictions and ensuring the metrics are still relevant. In future work we will equip VN-EGNN with more virtual nodes, but also a sophisticated ranking strategy on the virtual node representations.

## Acknowledgments

The ELLIS Unit Linz, the LIT AI Lab, the Institute for Machine Learning, are supported by the Federal State Upper Austria. We thank the projects AI-MOTION (LIT-2018-6-YOU-212), DeepFlood (LIT-2019-8-YOU-213), Medical Cognitive Computing Center (MC3), INCONTROL-RL (FFG-881064), PRIMAL (FFG-873979), S3AI (FFG-872172), DL for GranularFlow (FFG-871302), EPILEPSIA (FFG-892171), AIRI FG 9-N (FWF-36284, FWF-36235), AI4GreenHeatingGrids(FFG- 899943), INTEGRATE (FFG-892418), ELISE (H2020-ICT-2019-3 ID: 951847), Stars4Waters (HORIZON-CL6-2021-CLIMATE-01-01). We thank Audi.JKU Deep Learning Center, TGW LOGISTICS GROUP GMBH, Silicon Austria Labs (SAL), FILL Gesellschaft mbH, Anyline GmbH, Google, ZF Friedrichshafen AG, Robert Bosch GmbH, UCB Biopharma SRL, Merck Healthcare KGaA, Verbund AG, GLS (Univ. Waterloo) Software Competence Center Hagenberg GmbH, TÜV Austria, Frauscher Sensonic, TRUMPF and the NVIDIA Corporation.

## References


Aggarwal, R., Gupta, A., Chelur, V., Jawahar, C., and Priyakumar, U. D. (2022). DeepPocket: Ligand Binding Site Detection and Segmentation using 3D Convolutional Neural Networks. *Journal of Chemical Information and Modeling*, 62(21):5069–5079.

Alon, U. and Yahav, E. (2021). On the Bottleneck of Graph Neural Networks and its Practical Implications. In *International Conference on Learning Representations*.

Ba, J. L., Kiros, J. R., and Hinton, G. E. (2016). Layer Normalization. *arXiv preprint arXiv:1607.06450*.

Brandstetter, J., Hesselink, R., van der Pol, E., Bekkers, E. J., and Welling, M. (2021). Geometric and Physical Quantities improve E (3) Equivariant Message Passing. In *International Conference on Learning Representations*.

Cai, C. (2023). Local-to-global Perspectives on Graph Neural Networks. *arXiv preprint arXiv:2306.06547*.

Cai, C., Hy, T. S., Yu, R., and Wang, Y. (2023). On the Connection Between MPNN and Graph Transformer. *arXiv preprint arXiv:2301.11956*.

Chen, K., Mizianty, M. J., Gao, J., and Kurgan, L. (2011). A Critical Comparative Assessment of Predictions of Protein-Binding Sites for Biologically Relevant Organic Compounds. *Structure*, 19(5):613–621.

Chen, M., Wei, Z., Huang, Z., Ding, B., and Li, Y. (2020). Simple and Deep Graph Convolutional Networks. In *International Conference on Machine Learning*, volume 119, pages 1725–1735.

Cheng, A. C., Coleman, R. G., Smyth, K. T., Cao, Q., Soulard, P., Caffrey, D. R., Salzberg, A. C., and Huang, E. S. (2007). Structure-based maximal affinity model predicts small-molecule druggability. *Nature Biotechnology*, 25(1):71–75.

Cheng, J., Novati, G., Pan, J., Bycroft, C., Žemgulytė, A., Applebaum, T., Pritzel, A., Wong, L. H., Zielinski, M., Sargeant, T., Schneider, R. G., Senior, A. W., Jumper, J., Hassabis, D., Kohli, P., and Avsec, Ž. (2023). Accurate proteome-wide missense variant effect prediction with AlphaMissense. *Science*, 381(6664):eadg7492.

Comaniciu, D. and Meer, P. (2002). Mean shift: a robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619.

Corso, G., Stärk, H., Jing, B., Barzilay, R., and Jaakkola, T. S. (2023). DiffDock: Diffusion Steps, Twists, and Turns for Molecular Docking. In *International Conference on Learning Representations*.

Defferrard, M., Bresson, X., and Vandergheynst, P. (2016). Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering. In *Advances in Neural Information Processing Systems*, volume 29.

Desaphy, J., Bret, G., Rognan, D., and Kellenberger, E. (2015). sc-PDB: a 3D-database of ligandable binding sites—10 years on. *Nucleic Acids Research*, 43(D1):D399–D404.

Elfwing, S., Uchibe, E., and Doya, K. (2018). Sigmoid-Weighted Linear Units for Neural Network Function Approximation in Reinforcement Learning. *Neural Networks*, 107:3–11.

Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., and Dahl, G. E. (2017). Neural Message Passing for Quantum Chemistry. In *International Conference on Machine Learning*, volume 70, pages 1263–1272.

Halgren, T. A. (2009). Identifying and Characterizing Binding Sites and Assessing Druggability. *Journal of Chemical Information and Modeling*, 49(2):377–389.

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*.

Hu, W., Fey, M., Zitnik, M., Dong, Y., Ren, H., Liu, B., Catasta, M., and Leskovec, J. (2020). Open Graph Benchmark: Datasets for Machine Learning on Graphs. In *Advances in Neural Information Processing Systems*, volume 33, pages 22118–22133.

Hwang, E., Thost, V., Dasgupta, S. S., and Ma, T. (2022). An Analysis of Virtual Nodes in Graph Neural Networks for Link Prediction. In *Learning on Graphs Conference*.

Ishiguro, K., Maeda, S.-i., and Koyama, M. (2019). Graph Warp Module: an Auxiliary Module for Boosting the Power of Graph Neural Networks in Molecular Graph Analysis. *arXiv preprint arXiv:1902.01020*.

Jiménez, J., Doerr, S., Martínez-Rosell, G., Rose, A. S., and de Fabritiis, G. (2017). DeepSite: protein-binding site predictor using 3D-convolutional neural networks. *Bioinformatics*, 33(19):3036–3042.

Joshi, C. K., Bodnar, C., Mathis, S. V., Cohen, T., and Lio, P. (2023). On the Expressive Power of Geometric Graph Neural Networks. In *International Conference on Machine Learning*, volume 202, pages 15330–15355.

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., Bridgland, A., Meyer, C., Kohl, S. A. A., Ballard, A. J., Cowie, A., Romera-Paredes, B., Nikolov, S., Jain, R., Adler, J., Back, T., Petersen, S., Reiman, D., Clancy, E., Zielinski, M., Steinegger, M., Pacholska, M., Berghammer, T., Bodenstein, S., Silver, D., Vinyals, O., Senior, A. W., Kavukcuoglu, K., Kohli, P., and Hassabis, D. (2021). Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873):583–589.

Kandel, J., Tayara, H., and Chong, K. T. (2021). PUResNet: prediction of protein-ligand binding sites using deep residual neural network. *Journal of Cheminformatics*, 13(1):1–14.

Kipf, T. N. and Welling, M. (2017). Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations*.

Krivák, R. and Hoksza, D. (2018). P2Rank: machine learning based tool for rapid and accurate prediction of ligand binding sites from protein structure. *Journal of Cheminformatics*, 10(1):1–12.

Le Guilloux, V., Schmidtke, P., and Tuffery, P. (2009). Fpocket: An open source platform for ligand pocket detection. *BMC Bioinformatics*, 10:168.

Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-Based Learning Applied to Document Recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Lengauer, T. and Rarey, M. (1996). Computational methods for biomolecular docking. *Current Opinion in Structural Biology*, 6(3):402–406.

Li, J., Cai, D., and He, X. (2017). Learning Graph-Level Representation for Drug Discovery. *arXiv preprint arXiv:1709.03741*.

Liang, J., Edelsbrunner, H., and Woodward, C. (1998). Anatomy of protein pockets and cavities: Measurement of binding site geometry and implications for ligand design. *Protein Science*, 7(9):1884–1897.

Loshchilov, I. and Hutter, F. (2017). Decoupled Weight Decay Regularization. *arXiv preprint arXiv:1711.05101*.

Lu, W., Wu, Q., Zhang, J., Rao, J., Li, C., and Zheng, S. (2022). TANKBind: Trigonometry-Aware Neural NetworKs for Drug-Protein Binding Structure Prediction. In *Advances in Neural Information Processing Systems*, volume 35, pages 7236–7249.

Méndez-Lucio, O., Ahmad, M., del Rio-Chanona, E. A., and Wegner, J. K. (2021). A geometric deep learning approach to predict binding conformations of bioactive molecules. *Nature Machine Intelligence*, 3(12):1033–1039.

Mylonas, S. K., Axenopoulos, A., and Daras, P. (2021). DeepSurf: a surface-based deep learning approach for the prediction of ligand binding sites on proteins. *Bioinformatics*, 37(12):1681–1690.

Pham, T., Tran, T., Dam, H., and Venkatesh, S. (2017). Graph Classification via Deep Learning with Virtual Nodes. *arXiv preprint arXiv:1708.04357*.

Puny, O., Ben-Hamu, H., and Lipman, Y. (2020). Global Attention Improves Graph Networks Generalization. *arXiv preprint arXiv:2006.07846*.

Ren, F., Ding, X., Zheng, M., Korzinkin, M., Cai, X., Zhu, W., Mantsyzov, A., Aliper, A., Aladinskiy, V., Cao, Z., Kong, S., Long, X., Man Liu, B. H., Liu, Y., Naumov, V., Shneyderman, A., Ozerov, I. V., Wang, J., Pun, F. W., Polykovskiy, D. A., Sun, C., Levitt, M., Aspuru-Guzik, A., and Zhavoronkov, A. (2023). AlphaFold accelerates artificial intelligence powered drug discovery: efficient discovery of a novel CDK20 small molecule inhibitor. *Chemical Science*, 14(6):1443–1452.

Rusch, T. K., Bronstein, M. M., and Mishra, S. (2023). A Survey on Oversmoothing in Graph Neural Networks. *arXiv preprint arXiv:2303.10993*.

Sadybekov, A. V. and Katritch, V. (2023). Computational approaches streamlining drug discovery. *Nature*, 616(7958):673–685.

Sanner, M. F., Olson, A. J., and Spehner, J. C. (1996). Reduced surface: An efficient way to compute molecular surfaces. *Biopolymers*, 38(3):305–320.

Satorras, V. G., Hoogeboom, E., and Welling, M. (2021). E(n) Equivariant Graph Neural Networks. In *International Conference on Machine Learning*, volume 139, pages 9323–9332.

Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., and Monfardini, G. (2009). The Graph Neural Network Model. *IEEE Transactions on Neural Networks*, 20(1):61–80.

Schmidtke, P. and Barril, X. (2010). Understanding and Predicting Druggability. A High-Throughput Method for Detection of Drug Binding Sites. *Journal of Medicinal Chemistry*, 53(15):5858–5867.

Schütt, K., Kindermans, P.-J., Sauceda Felix, H. E., Chmiela, S., Tkatchenko, A., and Müller, K.-R. (2017). Schnet: A continuous-filter convolutional neural network for modeling quantum interactions. In *Advances in Neural Information Processing Systems*, volume 30.

Shamir, R. R., Duchin, Y., Kim, J., Sapiro, G., and Harel, N. (2019). Continuous Dice Coefficient: a Method for Evaluating Probabilistic Segmentations. *arXiv preprint arXiv:1906.11031*.

Stepniewska-Dziubinska, M. M., Zielenkiewicz, P., and Siedlecki, P. (2020). Improving detection of protein-ligand binding sites with 3D segmentation. *Scientific Reports*, 10(1):5035.

Swinbank, R. and James Purser, R. (2006). Fibonacci grids: A novel approach to global modelling. *Quarterly Journal of the Royal Meteorological Society*, 132(619):1769–1793.

Topping, J., Giovanni, F. D., Chamberlain, B. P., Dong, X., and Bronstein, M. M. (2022). Understanding over-squashing and bottlenecks on graphs via curvature. In *International Conference on Learning Representations*.

Tunyasuvunakool, K., Adler, J., Wu, Z., Green, T., Zielinski, M., Žídek, A., Bridgland, A., Cowie, A., Meyer, C., Laydon, A., Velankar, S., Kleywegt, G. J., Bateman, A., Evans, R., Pritzel, A., Figurnov, M., Ronneberger, O., Bates, R., Kohl, S. A. A., Potapenko, A., Ballard, A. J., Romera-Paredes, B., Nikolov, S., Jain, R., Clancy, E., Reiman, D., Petersen, S., Senior, A. W., Kavukcuoglu, K., Birney, E., Kohli, P., Jumper, J., and Hassabis, D. (2021). Highly accurate protein structure prediction for the human proteome. *Nature*, 596(7873):590–596.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., and Bengio, Y. (2017). Graph Attention Networks. *arXiv preprint arXiv:1710.10903*.

Wang, R., Fang, X., Lu, Y., and Wang, S. (2004). The PDBbind Database: Collection of Binding Affinities for Protein Ligand Complexes with Known Three-Dimensional Structures. *Journal of Medicinal Chemistry*, 47(12):2977–80.

Yan, X., Lu, Y., Li, Z., Wei, Q., Gao, X., Wang, S., Wu, S., and Cui, S. (2022). PointSite: A Point Cloud Segmentation Tool for Identification of Protein Ligand Binding Atoms. *Journal of Chemical Information and Modeling*, 62(11):2835–2845.

Zhang, Y., Huang, W., Wei, Z., Yuan, Y., and Ding, Z. (2023). EquiPocket: an E(3)-Equivariant Geometric Graph Neural Network for Ligand Binding Site Prediction. *arXiv preprint arXiv:2302.12177*.

# A Notation Overview

Table A1: Overview of used symbols and notations

| Definition | Symbol/Notation | Type |
|---|---|---|
| number of atom nodes | $N$ | $\mathbb{N}$ |
| number of virtual nodes | $K$ | $\mathbb{N}_0$ |
| number of known binding pockets | $M$ | $\mathbb{N}_0$ |
| dimension of node features | $D$ | $\mathbb{N}$ |
| dimension of messages | $E$ | $\mathbb{N}$ |
| number of message passing layers/steps | $L$ | $\mathbb{N}$ |
| node indices | $i, j, k, n$ | $\{1, ..., K\}$ or $\{1, ..., N\}$ |
| binding pocket index | $m$ | $\{1, ..., M\}$ |
| layer/step index | $l$ | $\{1, ..., L\}$ |
| index set of 10 nearest neighbor atoms | $\mathcal{N}(i)$ | $\{1, ..., N\}^{10}$ |
| atom node coordinates | $\mathbf{x}_i^l$ | $\mathbb{R}^3$ |
| virtual node coordinates | $\mathbf{z}_j^l$ | $\mathbb{R}^3$ |
| atom node feature representation | $\boldsymbol{h}_i^l$ | $\mathbb{R}^D$ |
| virtual node feature representation | $\boldsymbol{v}_j^l$ | $\mathbb{R}^D$ |
| edge feature between atoms | $a_{ij}$ | $\mathbb{R}$ |
| edge feature between atom and virtual node | $d_{ij}$ | $\mathbb{R}$ |
| ground-truth atom label | $y_n$ | {0,1} |
| predicted atom label | $\hat{y}_n$ | $[0, 1]$ |
| ground-truth binding site center | $\mathbf{y}_m$ | $\mathbb{R}^3$ |
| prediction of binding site center | $\hat{\mathbf{y}}_k$ | $\mathbb{R}^3$ |
| messages* | $\boldsymbol{m}_{ij}^{(aa)}, \boldsymbol{m}_{ij}^{(av)}, \boldsymbol{m}_{ij}^{(va)}$ | $\mathbb{R}^E$ |
| neural networks for message passing*: | | |
| message calculation | $\boldsymbol{\phi}_{e^{(aa)}}, \boldsymbol{\phi}_{e^{(av)}}, \boldsymbol{\phi}_{e^{(va)}}$ | $\mathbb{R}^D \times \mathbb{R}^D \times \mathbb{R} \times \mathbb{R} \to \mathbb{R}^E$ |
| coordinate update | $\phi_{\mathbf{x}^{(aa)}}, \phi_{\mathbf{x}^{(av)}}, \phi_{\mathbf{x}^{(va)}}$ | $\mathbb{R}^E \to \mathbb{R}$ |
| feature update | $\boldsymbol{\phi}_{h^{(aa)}}, \boldsymbol{\phi}_{h^{(av)}}, \boldsymbol{\phi}_{h^{(va)}}$ | $\mathbb{R}^D \times \mathbb{R}^E \to \mathbb{R}^D$ |
| segmentation loss | $\mathcal{L}_{\text{segm}}$ | $\mathbb{R}^N \times \mathbb{R}^N \to \mathbb{R}$ |
| binding site center loss | $\mathcal{L}_{\text{bsc}}$ | $\mathbb{R}^{3\times M} \times \mathbb{R}^{3\times K} \to \mathbb{R}$ |

* The superscripts (aa), (av) and (va) represent the message passing direction (**atom** to **atom**, **atom** to **virtual node**, **virtual node** to **atom**).

# B Problem Statement

## B.1 Representation of proteins.

The 3D structure of a protein is usually given by some measurement of its atoms that form the primary amino acid sequence of the protein and the absolute coordinates for the atoms are given as 3D points $\mathbf{x} \in \mathbb{R}^3$. The atoms themselves as well as the amino acids are characterized by their physical, chemical and biological properties. We assume that these properties are summarized by feature vectors $\boldsymbol{h} \in \mathbb{R}^D$, which are located at the atom centers (either of all the atoms or only the ones forming the protein backbone). We formally represent proteins by a neighborhood graph $\mathcal{P} = (\mathcal{P}_N, \mathcal{P}_E)$ with $N$ atom-property pairs, i.e. $\mathcal{P}_N = \{(\mathbf{x}_n, \boldsymbol{h}_n)\}_{n=1}^N$ with $\mathbf{x}_n \in \mathbb{R}^3$ and $\boldsymbol{h}_n \in \mathbb{R}^D$ and a set of directed edges $\mathcal{P}_E$ which consist of atom-property pairs $(i, j)$. Each node $i$ has incoming edges from the 10 nearest nodes $j$ that are closer than 30Å according to the Euclidean distance $\|\mathbf{x}_i - \mathbf{x}_j\|$.

## B.2 Representation of binding sites.

Binding sites are regions around or within proteins, to which ligands can potentially bind. Basically, one can either describe binding sites *explicitly* or *implicitly*. In their explicit representation binding

sites would be directly described by the location specifics of the regions, where ligands are located, especially by a region center point. In their implicit representation, binding sites would be described by the atoms of the protein, which surround the ligand. Atoms close to the ligand would be marked as binding site atoms. It might be worth mentioning, that several binding sites per protein are possible.

Formally, for the explicit representation, we describe the (experimentally observed) binding site center points of $M$ distinct binding sites by $\mathbf{y}_m \in \mathbb{R}^3$ with $1 \leq m \leq M$. For the implicit representation, we assign to each protein atom $n$ a label $y_n \in \{0, 1\}$, which is set to 1 if the atom center is within the threshold distance of observed binding ligands, and 0 otherwise.

### B.3 Objective.

From an abstract point of view, we want to learn a predictive machine learning model $\mathcal{F}$, parameterized by $\omega$, which maps proteins characterized by the positions of their atoms together with their properties to a binary prediction per atom, whether it might form a binding site, and to $K$ 3D coordinates representing binding site region center points:

$$\mathcal{F}_\omega : \bigtimes_{n=1}^{N} \left( \underbrace{\mathbb{R}^3 \times \mathbb{R}^D}_{\substack{\text{protein 3D atom} \\ \text{coords with} \\ D\text{-dim features}}} \right) \mapsto \underbrace{[0,1]^N}_{\substack{\text{sem. segm.} \\ \text{protein atoms}}} \times \underbrace{\bigtimes_{k=1}^{K} \mathbb{R}^3}_{\substack{\text{pos. pred.} \\ \text{virt. nodes}}} \tag{B.1}$$

$$:= \mathcal{F}_\omega^{\text{segm}} \qquad := \mathcal{F}_\omega^{\text{bsc}}$$

$$\mathcal{F}_\omega\left((\mathbf{x}_1, \boldsymbol{h}_1), \ldots, (\mathbf{x}_N, \boldsymbol{h}_N)\right) = \left((\hat{y}_1, \ldots, \hat{y}_N), (\hat{\mathbf{y}}_1, \ldots, \hat{\mathbf{y}}_K)\right)$$
$$\mathcal{F}_\omega^{\text{segm}}\left((\mathbf{x}_1, \boldsymbol{h}_1), \ldots, (\mathbf{x}_N, \boldsymbol{h}_N)\right) := \text{proj}_1 \mathcal{F}_\omega\left((\mathbf{x}_1, \boldsymbol{h}_1), \ldots, (\mathbf{x}_N, \boldsymbol{h}_N)\right)$$
$$\mathcal{F}_\omega^{\text{bsc}}\left((\mathbf{x}_1, \boldsymbol{h}_1), \ldots, (\mathbf{x}_N, \boldsymbol{h}_N)\right) := \text{proj}_2 \mathcal{F}_\omega\left((\mathbf{x}_1, \boldsymbol{h}_1), \ldots, (\mathbf{x}_N, \boldsymbol{h}_N)\right),$$

where $\text{proj}_i$ is a projection, that gives the $i$-th component (i.e., prediction of the semantic segmenation part or coordinate predicitons or virtual nodes). Note, that for our predictive model, we use a fixed number $K$ of binding point centers, while indeed $M$ might have been observed for a specific protein.

### B.4 Utilized Loss Functions.

**Segmentation loss.** For semantic segmentation (i.e., the prediction of $\mathcal{F}_\omega^{\text{segm}}$), we use a Dice loss, that is based on the continuous Dice coefficient (Shamir et al., 2019), with $\epsilon = 1$:

$$\mathcal{L}_{\text{segm}} = \text{Dice}\left((y_1, \ldots, y_N), (\hat{y}_1, \ldots, \hat{y}_N)\right) := 1 - \frac{2 \sum_{n=1}^{N} y_n \, \hat{y}_n + \epsilon}{\sum_{n=1}^{N} y_n + \sum_{n=1}^{N} \hat{y}_n + \epsilon} \tag{B.2}$$

Perfect predictions lead to a Dice loss of 0, while perfectly wrong predictions would lead to a Dice of 1 (in case $\epsilon = 0$ and the denominator is $> 0$).

**Binding site center loss.** For prediction of the binding site region center points (i.e., the prediction of $\mathcal{F}_\omega^{\text{bsc}}$), we use the (squared) Euclidean distance between the set of predicted points and the set of observed ones. More specifically, we assume to be given $M$ observed center points $\{\mathbf{y}_1, \ldots, \mathbf{y}_M\}$. Each of the binding site center points should be detected by at least one of the $K$ outputs from $\mathcal{F}_\omega^{\text{bsc}}$, which translates to using the minimum squared distance to any predicted center point for any of the observed center points:

$$\mathcal{L}_{\text{bsc}} = \text{Dist}\left(\{\mathbf{y}_1, \ldots, \mathbf{y}_M\}, \{\hat{\mathbf{y}}_1, \ldots, \hat{\mathbf{y}}_K\}\right) := \frac{1}{M} \sum_{m=1}^{M} \min_{k \in 1, \ldots, K} \|\mathbf{y}_m - \hat{\mathbf{y}}_k\|^2. \tag{B.3}$$

Our optimization objective is then the sum of the Dice and the Dist loss:

$$\alpha \text{Dice}\left((y_1, \ldots, y_N), (\hat{y}_1, \ldots, \hat{y}_N)\right) + \text{Dist}\left(\{\mathbf{y}_1, \ldots, \mathbf{y}_M\}, \{\hat{\mathbf{y}}_1, \ldots, \hat{\mathbf{y}}_K\}\right) \tag{B.4}$$

with the hyperparameter $\alpha = 1$.

## C Background on Group Theory and Equivariance

A group in the mathematical sense is a set $G$ along with a binary operation $\circ : G \times G \to G$ with the following properties:

- *Associativity*: The group operation is associative, i.e. $(g \circ h) \circ k = g \circ (h \circ k)$ for all $g, h, k \in G$.
- *Identity:* There exists a unique identity element $e \in G$, such that $e \circ g = g \circ e = g$ for all $g \in G$.
- *Inverse:* For each $g \in G$ there is a unique inverse element $g^{-1} \in G$, such that $g \circ g^{-1} = g^{-1} \circ g = e$.
- *Closure:* For each $g, h \in G$ their combination $g \circ h$ is also an element of $G$.

A group action of group $G$ on a set $X$ is defined as a set of mappings $T_g : X \to X$ which associate each element $g \in G$ with a transformation on $X$, whereby the identity element $e \in G$ leaves $X$ unchanged ($T_e(x) = x \quad \forall x \in X$).

An example is the group of translations $\mathbb{T}$ on $\mathbb{R}^n$ with group action $T_t(x) = \mathbf{x} + \mathbf{t} \quad \forall \mathbf{x}, \mathbf{t} \in \mathbb{R}^n$, which shifts points in $\mathbb{R}^n$ by a vector $\mathbf{t}$.

A function $f : X \to Y$ is equivariant to group $G$ with group action $T_g$ if there exists an equivalent group action $S_g : Y \to Y$ on $G$ such that

$$f(T_g(x)) = S_g(f(x)) \quad \forall x \in X, g \in G.$$

For example, a function $f : \mathbb{R}^n \to \mathbb{R}^n$ is translation-equivariant if a translation of an input vector $\mathbf{x} \in \mathbb{R}^n$ by $\mathbf{t} \in \mathbb{R}^n$ leads to the same transformation of the output vector $f(x) \in \mathbb{R}^n$, i.e. $f(\mathbf{x} + \mathbf{t}) = f(\mathbf{x}) + \mathbf{t}$.

Equivariant graph neural networks (EGNNs) $\psi$ as defined by Satorras et al. (2021) exhibit three types of equivariances:

1. *Translation equivariance:* EGNNs are equivariant to column-wise addition of a vector $\mathbf{t} \in \mathbb{R}^n$ to all points in a point cloud $\mathbf{X} \in \mathbb{R}^{n \times N}$: $\psi(\mathbf{X} + \mathbf{t}) = \psi(\mathbf{X}) + \mathbf{t}$.
2. *Rotation and reflection equivariance*: Rotation or reflection of all points in the point cloud by multiplication with an orthogonal matrix $\mathbf{R} \in \mathbb{R}^{n \times n}$ leads to an equivalent rotation of the output coordinates: $\psi(\mathbf{R}\mathbf{X}) = \mathbf{R}\psi(\mathbf{X})$.

   The group spanning all translations, rotations and reflections in $\mathbb{R}^n$ is called Euclidean group, denoted E($n$), as it preserves Euclidean distances. A proof for E(n)-equivariance of VN-EGNN can be found in Appendix D.
3. *Permutation equivariance:* The numbering of elements in a point cloud or graph nodes does not influence the output, i.e. multiplication with a permutation matrix $\boldsymbol{P} \in \mathbb{R}^{N \times N}$ leads to the same permutation of output nodes: $\psi(\mathbf{X}\boldsymbol{P}) = \psi(\mathbf{X})\boldsymbol{P}$. This property holds for message passing graph neural networks in general, as they aggregate and update node information based on local neighborhood structure, regardless of the order in which nodes are presented.

## D Equivariance of VN-EGNN

In this section we show that the equivariance property of EGNN (Satorras et al., 2021) extends to VN-EGNN, i.e., that rotation by an orthogonal matrix $\mathbf{R} \in \mathbb{R}^{3 \times 3}$ and translation by a vector $\mathbf{t} \in \mathbb{R}^3$ of atom and virtual node coordinates leads to an equivalent transformation of output coordinates while leaving node features invariant when applying the message passing steps of VN-EGNN.

**Proposition 1.** *(more formal) Equivariant graph neural networks with virtual nodes (*VN-EGNN*) as defined in eqs. (7) to (18) are equivariant with respect to roto-translations of the input and virtual node coordinates. With the definitions* $\boldsymbol{H}^l := (\boldsymbol{h}_1^l, \ldots, \boldsymbol{h}_N^l)$, $\mathbf{X}^l := (\mathbf{x}_1^l, \ldots, \mathbf{x}_N^l)$, $\boldsymbol{V}^l := (\boldsymbol{v}_1^l, \ldots, \boldsymbol{v}_K^l)$, $\mathbf{Z}^l := (\mathbf{z}_1^l, \ldots, \mathbf{z}_K^l)$, *and* $\left(\boldsymbol{H}^{l+1}, \mathbf{X}^{l+1}, \boldsymbol{V}^{l+1}, \mathbf{Z}^{l+1}\right) = \text{VN-EGNN}\left(\boldsymbol{H}^l, \mathbf{X}^l, \boldsymbol{V}^l, \mathbf{Z}^l\right)$[2]*, which is*

[2]For simplicity of notation, we re-order and re-group input and output data vectors of VN-EGNN compared to the main text here to be able to use the matrices $\boldsymbol{H}, \mathbf{X}, \boldsymbol{V}, \mathbf{Z}$ as inputs and outputs.

*analogous to eq. (6), the following holds (equivariance to roto-translations):*

$$\left(\boldsymbol{H}^{l+1}, \mathbf{R}\mathbf{X}^{l+1} + \mathbf{t}, \boldsymbol{V}^{l+1}, \mathbf{R}\mathbf{Z}^{l+1} + \mathbf{t}\right) = \text{VN-EGNN}\left(\boldsymbol{H}^{l}, \mathbf{R}\mathbf{X}^{l} + \mathbf{t}, \boldsymbol{V}^{l}, \mathbf{R}\mathbf{Z}^{l} + \mathbf{t}\right), \tag{D.1}$$

*where the addition $\mathbf{X}^l + \mathbf{t}$ is defined as column-wise addition of the vector $\mathbf{t}$ to the matrix $\mathbf{X}$.*

*Proof.* We will proceed by tracking the propagation of node roto-translations through the VN-EGNN network. First, we want to show invariance in eq. (7) in phase I of message passing, equivalently to Satorras et al. (2021), i.e.:

$$\boldsymbol{m}_{ij}^{(aa)} = \boldsymbol{\phi}_{e^{(aa)}}(\boldsymbol{h}_i^l, \boldsymbol{h}_j^l, \|\mathbf{R}\mathbf{x}_i^l + \mathbf{t} - [\mathbf{R}\mathbf{x}_j^l + \mathbf{t}]\|, a_{ij}) = \boldsymbol{\phi}_{e^{(aa)}}(\boldsymbol{h}_i^l, \boldsymbol{h}_j^l, \|\mathbf{x}_i^l - \mathbf{x}_j^l\|, a_{ij}) \tag{D.2}$$

Assuming the initial node features $\boldsymbol{h}_i^0$ and edge representations $a_{ij}$ do not encode information about the original coordinates $\mathbf{x}_i^0$, it remains to be shown that the Euclidean distance between two nodes is also invariant to translation and rotation:

$$\begin{aligned}
\|\mathbf{R}\mathbf{x}_i^l + \mathbf{t} - [\mathbf{R}\mathbf{x}_j^l + \mathbf{t}]\|^2 &= \|\mathbf{R}\mathbf{x}_i^l - \mathbf{R}\mathbf{x}_j^l\|^2 \\
&= (\mathbf{x}_i^l - \mathbf{x}_j^l)^\top \mathbf{R}^\top \mathbf{R}(\mathbf{x}_i^l - \mathbf{x}_j^l) \\
&= (\mathbf{x}_i^l - \mathbf{x}_j^l)^\top \mathbf{I}(\mathbf{x}_i^l - \mathbf{x}_j^l) \\
&= \|\mathbf{x}_i^l - \mathbf{x}_j^l\|^2 \\
\|\mathbf{R}\mathbf{x}_i^l + \mathbf{t} - [\mathbf{R}\mathbf{x}_j^l + \mathbf{t}]\| &= \|\mathbf{x}_i^l - \mathbf{x}_j^l\|
\end{aligned} \tag{D.3}$$

Consequently, the sum over messages (eq. (8)) and the feature update function (eq. (10)), which only uses the summed messages and previous node features as input, are invariant as well, leaving the intermediate output feature representations $\boldsymbol{h}_i^{l+1/2}$ independent of coordinate transformations.

For the remaining equation (eq. (9)) of phase I the equivariance property can be shown as follows, where eq. (D.3) is used in the first equality:

$$\begin{aligned}
&\mathbf{R}\mathbf{x}_j^l + \mathbf{t} + \frac{1}{|\mathcal{N}(j)|} \sum_{i \in \mathcal{N}(j)} \frac{\mathbf{R}\mathbf{x}_i^l + \mathbf{t} - [\mathbf{R}\mathbf{x}_j^l + \mathbf{t}]}{\|\mathbf{R}\mathbf{x}_i^l + \mathbf{t} - [\mathbf{R}\mathbf{x}_j^l + \mathbf{t}]\|} \phi_{x^{aa}}(\boldsymbol{m}_{ij}^{(aa)}) \\
&\quad = \mathbf{R}\mathbf{x}_j^l + \mathbf{t} + \frac{1}{|\mathcal{N}(j)|} \sum_{i \in \mathcal{N}(j)} \frac{\mathbf{R}\mathbf{x}_i^l + \mathbf{t} - [\mathbf{R}\mathbf{x}_j^l + \mathbf{t}]}{\|\mathbf{x}_i^l - \mathbf{x}_j^l\|} \phi_{x^{aa}}(\boldsymbol{m}_{ij}^{(aa)}) \\
&\quad = \mathbf{R}\mathbf{x}_j^l + \mathbf{t} + \frac{1}{|\mathcal{N}(j)|} \mathbf{R} \sum_{i \in \mathcal{N}(j)} \frac{\mathbf{x}_i^l - \mathbf{x}_j^l}{\|\mathbf{x}_i^l - \mathbf{x}_j^l\|} \phi_{x^{aa}}(\boldsymbol{m}_{ij}^{(aa)}) \\
&\quad = \mathbf{R}\left(\mathbf{x}_j^l + \frac{1}{|\mathcal{N}(j)|} \sum_{i \in \mathcal{N}(j)} \frac{\mathbf{x}_i^l - \mathbf{x}_j^l}{\|\mathbf{x}_i^l - \mathbf{x}_j^l\|} \phi_{x^{aa}}(\boldsymbol{m}_{ij}^{(aa)})\right) + \mathbf{t} \\
&\quad = \mathbf{R}\mathbf{x}_j^{l+1/2} + \mathbf{t}
\end{aligned} \tag{D.4}$$

In phase II of message passing, we input the updated atom node coordinates $\mathbf{R}\mathbf{x}_i^{l+1/2} + \mathbf{t}$ from eq. (D.4) together with virtual node coordinates $\mathbf{R}\mathbf{z}_j^l + \mathbf{t}$, both subjected to identical rotation and translation. Invariance of eqs. (11), (12) and (14) can be deduced similarly to above, using the invariance properties of node features $\boldsymbol{h}_i^{l+1/2}$ and $\boldsymbol{v}_i^l$, edge features $a_{ij}$ and $d_{ij}$, and Euclidean distance (eq. (D.3)):

$$\begin{aligned}
\boldsymbol{m}_{ij}^{(av)} &= \boldsymbol{\phi}_{e^{(av)}}(\boldsymbol{h}_i^{l+1/2}, \boldsymbol{v}_j^l, \|\mathbf{R}\mathbf{x}_i^{l+1/2} + \mathbf{t} - [\mathbf{R}\mathbf{z}_j^l + \mathbf{t}]\|, d_{ij}) \\
&= \boldsymbol{\phi}_{e^{(av)}}(\boldsymbol{h}_i^{l+1/2}, \boldsymbol{v}_j^l, \|\mathbf{x}_i^{l+1/2} - \mathbf{z}_j^l\|, d_{ij})
\end{aligned} \tag{D.5}$$

Thus, the output virtual node features $\boldsymbol{v}_j^{l+1}$ are invariant to roto-translations of node coordinates.

Equivariance of output virtual node coordinates $\mathbf{z}_j^{l+1}$ follows analogously to eq. (D.4):

$$\mathbf{R}\mathbf{z}_j^l + \mathbf{t} + \frac{1}{N}\sum_{i=1}^{N} \frac{\mathbf{R}\mathbf{x}_i^{l+1/2} + \mathbf{t} - [\mathbf{R}\mathbf{z}_j^l + \mathbf{t}]}{\|\mathbf{R}\mathbf{x}_i^{l+1/2} + \mathbf{t} - [\mathbf{R}\mathbf{z}_j^l + \mathbf{t}]\|} \phi_{x^{av}}(\boldsymbol{m}_{ij}^{(av)}) = \mathbf{R}\mathbf{z}_j^{l+1} + \mathbf{t} \tag{D.6}$$

The same derivations of message invariance

$$\begin{aligned} \boldsymbol{m}_{ij}^{(va)} &= \boldsymbol{\phi}_{e^{(va)}}(\boldsymbol{v}_i^{l+1}, \boldsymbol{h}_j^{l+1/2}, \|R\mathbf{z}_i^{l+1} + \mathbf{t} - [\mathbf{R}\mathbf{x}_j^{l+1/2} + \mathbf{t}]\|, d_{ij}) \\ &= \boldsymbol{\phi}_{e^{(va)}}(\boldsymbol{v}_i^{l+1}, \boldsymbol{h}_j^{l+1/2}, \|\mathbf{z}_i^{l+1} - \mathbf{x}_j^{l+1/2}\|, d_{ij}) \end{aligned} \tag{D.7}$$

and coordinate equivariance

$$\mathbf{R}\mathbf{x}_j^{l+1/2} + \mathbf{t} + \frac{1}{K}\sum_{i=1}^{K} \frac{\mathbf{R}\mathbf{z}_i^{l+1} + \mathbf{t} - [\mathbf{R}\mathbf{x}_j^{l+1/2} + \mathbf{t}]}{\|\mathbf{R}\mathbf{z}_i^{l+1} + \mathbf{t} - [\mathbf{R}\mathbf{x}_j^{l+1/2} + \mathbf{t}]\|} \phi_{x^{va}}(\boldsymbol{m}_{ij}^{(va)}) = \mathbf{R}\mathbf{x}_j^{l+1} + \mathbf{t} \tag{D.8}$$

can be applied to phase III (eqs. (15) to (18)), proving that invariance of feature representations $\boldsymbol{h}_j^{l+1}$ and equivariance of coordinates $\mathbf{x}_j^{l+1}$ holds true for atom nodes as well, thus, proving proposition 1.

□

# E Hyperparameters and hyperparameter selection

Table E1 shows the evaluated hyperparameters. Bold indicates the parameters used in final model.

| hyperparmater | considered and **selected** values |
|---|---|
| optimizer | {**AdamW**, Adam } |
| learning rate | {**0.001**, 0.0001} |
| activation function | { **SiLU**, ReLU } |
| dimension of node features $D$ | {20, **30**} |
| dimension of the messages $P$ | {40, **50**} |
| number of message passing layers/steps $L$ | {2, 3, 4, **5**} |
| number of virtual nodes $K$ | {4, **8**} |

Table E1: A list of considered and selected hyperparameters.

# F Fibonacci grid

The Fibonacci Sphere (Swinbank and James Purser, 2006) offers a solution for evenly distributing points on a sphere. We chose this method to obtain the starting coordinates of virtal nodes for its simplicity and efficiency.

# G Initial experiment

Figure G1 shows the training curves for a VN-EGNN during the development phase. The model was only trained to minimize the segmentation loss $\mathcal{L}_{\text{segm}}$. Even in the absence of a binding site center loss $\mathcal{L}_{\text{bsc}}$, the virtual nodes tend to converge towards the actual binding site center. This finding inspired us to further refine the positions of the virtual nodes by including $\mathcal{L}_{\text{bsc}}$ directly in the optimization objective, which further improved the results.

# H Visualizations

Figure H1, shows our model predictions visualized with Pymol.

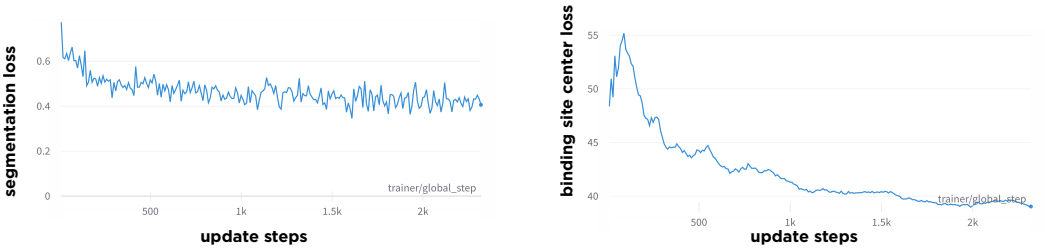


Figure G1: Training curves of a VN-EGNN during development. The model was trained only using segmentation loss, i.e. Dice loss. **Left**: Learning curve depicting the segmentation loss during training. **Right**: Learning curve depicting the positional binding site center point loss, which was not explicitly optimized for this experiment. Despite only being trained to minimize the segmentation loss, the virtual nodes converged towards the known binding sites centers.

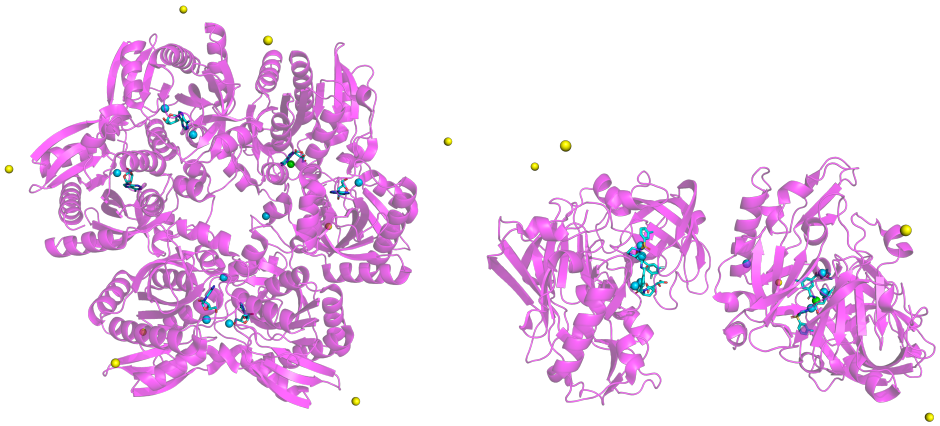

Figure H1: Examples of detected binding sites. Two different proteins visualized with Pymol, the yellow points represent the initial positions of the virtual nodes, the turqoise points represent the virtual nodes after $L$ message passing steps. The green dot represents the true positions as it is in the dataset. Because the dataset contains only one ligand also for symmetric proteins, we took the original proteins from the PDB database. As our plots shows our model distributes the virtual nodes among different ligands. To better see the different positions of the virtual nods we lowered the opacity of the cartoon visualization type. **Top**:1odi. **Bottom**: 3lpk.