Prediction of Molecular Field Points using SE(3)-Transformer Model

Florian B. Hinz
Department of Pharmaceutical Sciences
University of Basel
Klingelbergstrasse 50
4056 Basel
Switzerland
florian.hinz@unibas.ch

Amr H. Mahmoud
Department of Pharmaceutical Sciences
University of Basel
Klingelbergstrasse 50
4056 Basel
Switzerland
amr.abdallah@unibas.ch

Markus A. Lill
Department of Pharmaceutical Sciences
University of Basel
Klingelbergstrasse 50
4056 Basel
Switzerland
markus.lill@unibas.ch

Abstract

Due to their computational efficiency, 2D fingerprints are typically used in similarity-based high-content screening. The interaction of a ligand with its target protein, however, relies on its physicochemical interactions in 3D space. Thus, ligands with different 2D scaffolds can bind to the same protein if these ligands share similar interaction patterns. Molecular fields can represent those interaction profiles. For efficiency, the extrema of those molecular fields, named field points, are used to quantify the ligand similarity in 3D. The calculation of field points involves the evaluation of the interaction energy between the ligand and a small probe shifted on a fine grid representing the molecular surface. These calculations are computationally prohibitive for large datasets of ligands, making field point representations of molecules intractable for high-content screening. Here, we overcome this roadblock by one-shot prediction of field points using generative neural networks based on the molecular structure alone. Field points are predicted by training an SE(3)-Transformer, an equivariant, attention-based graph neural network architecture, on a large set of ligands with field point data. Initial data demonstrates the feasibility of this approach to precisely generate negative, positive and hydrophobic field points within 1 Å of the ground truth for a diverse set of drug-like molecules.

1 Introduction

Similarity based virtual screening often relies on 2d or 3d molecular structures. The interaction between ligand and a target protein, however, depends on the strength of physicochemical interactions between the two entities. Those interactions are best modeled by molecular interaction fields of a ligand with molecular probes characterizing the interacting protein. Consequently, ligands with different molecular structure but similar molecular interaction fields can bind at the same binding site. In Cheeseright et al. (2006) it was suggested that the molecular interaction fields are sufficiently well represented by their extrema points (see Fig. 1), named field points. In the same publication, a methodology for constructing molecular field points for electrostatic, van der Waals and hydrophobic interaction fields was described. The successful identification of alternative lead compounds with different molecular topology but similar binding properties based on field points was illustrated for a range of different ligands and targets (e.g. in Low and Vinter (2008),Cheeseright et al. (2008), Cheeseright et al. (2009)). In this paper, we train an equivariant, attention based, graph neural network for field point prediction. Equivariance is a mathematical property determining the effect of a transformation on a function's input to its output (see Definition 1). In the context of deep learning, equivariance properties allow for data efficiency by reflecting problem symmetries. Thus, incorporating equivariance into a model seems to be specifically promising in the area of computational biology and cheminformatics, as data is scarce and the bio-chemical processes occur irrespectively of rotational and translational coordinate transforms.

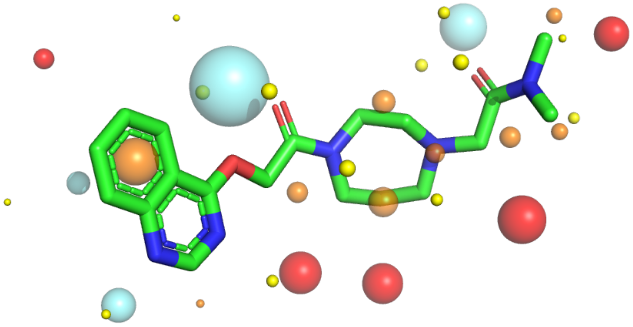

Figure 1: Molecule with associated field points (red: electrostatic positive, aquamarine: electrostatic negative, orange: hydrophobic, yellow: van der Waals). A larger sphere indicates a higher absolute value of the molecular force field at this position.

Convolutional neural networks were first applied in Lecun et al. (1998) and have since then proved to be impressively successful for a range of applications, such as the analysis of image, video and audio data. CNNs owe this effectiveness to weight sharing, constructed in a way that results in translation equivariance (see e.g. section 1.1 in Gerken et al. (2021) for a formal context). Loosely speaking, this equivariance property ensures that applying a convolutional layer to a translated image is equivalent to translating the result of the application of the convolutional layer to the original image. This translation equivariance property was shown to be useful in a range of applications, most prominently in image classification tasks (see e.g. Krizhevsky et al. (2012) for an early example).

Inspired by the success of CNNs, extensive research effort was devoted to the construction of neural networks satisfying equivariance properties in a more general, group theoretically formalized context (see e.g. Cohen and Welling (2016) for pioneering work). These architectures reduce the number of parameters while maintaining expressivity, by incorporating existing problem-inherent symmetries into the model. The resulting reduction of model complexity leads to increased training efficiency, specifically in higher dimensions. Further advantages of equivariance properties are a more understandable, interpretable and robust response of the network to transformations of the input data.

In the context of cheminformatics we are concerned with problem symmetries in the three-dimensional space: A molecule is still of the same type, no matter how it is shifted or rotated in the Euclidean space. The strength and dynamics of protein-ligand interactions do not depend on the point of observation. If we rotate our point of view onto the same molecule, our prediction of the molecular interaction fields should be rotated accordingly. Naturally, it seems to be a promising approach to introduce an inductive bias into a neural network that mathematically guarantees such properties. More specifically, we are interested in network architectures that are equivariant w.r.t. the group of rotations and translations in three dimensions, i.e. SE(3). One model architecture satisfying this property is the SE(3)-Transformer (introduced in Fuchs et al. (2020)). The SE(3)-Transformer is an attention based, graph neural network with tensorfield type building blocks (see Thomas et al. (2018)). We build on the model suggested in Fuchs et al. (2020) and its efficient implementation by NVIDIA (NVIDIA, 2022), and applied it to a large database consisting of small molecules and their field points.

In section 2 we provide details on the dataset and the construction of descriptors. In section 3 we describe the model architecture and introduce the loss function used for learning. For the purpose of quantifying the quality of the predictions of our model, we introduce evaluation functions in chapter 4 and discuss the results.

2 Dataset Preparation

The original data set consists of $\sim$ 24 million small molecules of sizes ranging from 6 to 100 atoms. For each molecule the data set contains up to 5 different conformations. The data is artificially generated by third-party software in accordance with the elaborations presented in Cheeseright et al. (2006). We first construct a diverse subset based on the Tanimoto distance of Morgan fingerprints (Morgan, 1965) using a greedy algorithm, resulting in a data set of 85000 molecules. Each data sample consists of a molecule and the associated field points (see Table 2, Table 3 and Figure 5 for an example). After constructing additional descriptors, each molecule per data sample is composed of the following information: Graph topology (coordinates and edges), node degree, atom type (defined by element and hybridization state), partial charge, atom size, and Wildman-Crippen logP.

Molecules are represented by graphs where information between neighboring nodes is exchanged along edges using the SE(3)-Trasnformer. Two different graph topologies have been tested. In one approach, the graph topology is defined by the covalent bonds. In the second approach, the graph topology is based on Euclidean distances, i.e. two nodes (e.g. atoms) are connected via an undirected edge, if their distance is less than 7 Å.
The atom type is one-hot encoded as a vector of dimension 24. The node degree is kept as natural number scalar. For the partial charge, atom size, and Wildman-Crippen logP value we apply a radial basis function expansion. In detail, let $x \in \mathbb{R}$ be a feature taking values in the range $[x_{min}, x_{max}]$ and $c_1, \dots, c_m \in \mathbb{R}$ with

$$x_{min} = c_1 < c_2 < \cdots < c_m = x_{max}$$

be $m \in \mathbb{N}$ equidistant support points in the range of values of the corresponding feature. Define

$$\sigma := \sqrt{|c_1 - c_2|} > 0.$$

Then the scalar feature $x \in [x_{min}, x_{max}]$ is expanded to an m dimensional vector as follows:

$$r(x) = \left(e^{-\frac{1}{\sigma^2}(x-c_1)^2}, e^{-\frac{1}{\sigma^2}(x-c_2)^2}, \dots, e^{-\frac{1}{\sigma^2}(x-c_m)^2}\right).$$

For the partial charge we choose $m = 15, x_{min} = -1.14, x_{max} = 1.70$, for the atom size $m = 10, x_{min} = 1.20, x_{max} = 2.10$ and for the Wildman-Crippen logP value $m = 20, x_{min} = -1.95, x_{max} = 0.886$.

In total, the features ("node degree", "atom type", "partial charge", "atom size", "Crippen logP value") add up to a vector of dimension $70 = 1 + 24 + 15 + 10 + 20$. That means each node in the graph is associated to a feature vector $f_{input} \in \mathbb{R}^{70}$ that will serve as input to the model described in the next section.

3 Model and Loss function

In the following we are using the same terminology of *rotation order* and *fiber structure* as in Fuchs et al. (2020) (see Terminology 1 in the Appendix). Our model is built on the NVIDIA implementation (NVIDIA, 2022) of the SE(3)-Transformer as described in Fuchs et al. (2020). In detail, we are using a neural network consisting of 7 "SE3 Attention Blocks" of the following form (see Figure 2 and Terminology 2, 3, 4):

- ConvSE3 (Tensor field network convolution transforming an arbitrary input fiber to an arbitrary output fiber. In this case used for computing attention key fiber and value fiber with specified number of degrees and channels)

- LinearSE3 (self-interaction of channels within degrees for computing the query fiber)
- AttentionSE3 (attention calculation over all neighboring atoms)
- LinearSE3 (self-interaction of channels within degrees to obtain output fiber)

We apply layer normalization in each ”ConvSE3” component. In total, the network contains about 3.97 million learnable parameters. In the hidden layers we allow rotation orders $0, 1, 2$ with 32 channels each, corresponding to the fiber structure ['0': 32; '1': 32; '2': 32]. Recall that scalars are represented by rotation order 0, vectors by 1, and rotation order 2 corresponds to a higher order geometric object of dimension 5. Hence, per node (e.g. atom) of the graph, the attention block calculates a vector of dimension $32 \cdot (1 + 3 + 5) = 288$. Subsequent to the 7 attention blocks, we apply a final ”ConvSE3” layer in order to transform to the output fiber structure ['0': 3 ; '1': 3] (network architecture is illustrated in Figure 2). Thus, the model predicts 3 scalars and vectors per node (e.g. atom) of the graph (see first image of Figure 3). These vectors represent the positions of field points relative to the coordinates of the corresponding atom. The associated scalar per vector corresponds to a weighting of this specific prediction. In order to train these predictions, a suitable loss function was developed. Note that we consider the field point prediction tasks as separate problems for each type of field point (positive, negative, hydrophobic, van der Waals). Thus, a separate model was trained for each type of field point.

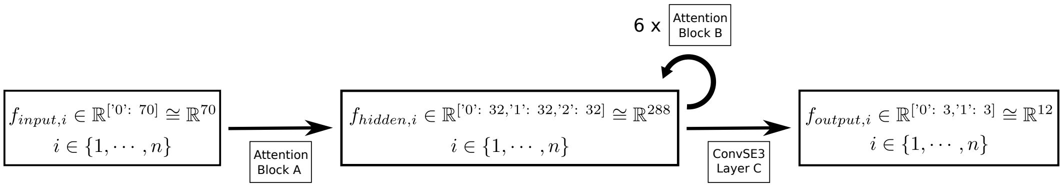
6 x
Attention Block B
$f_{input,i} \in \mathbb{R}^{['0':\ 70]} \cong \mathbb{R}^{70}$
$i \in \{1, \cdots, n\}$
Attention Block A
$f_{hidden,i} \in \mathbb{R}^{['0':\ 32,'1':\ 32,'2':\ 32]} \cong \mathbb{R}^{288}$
$i \in \{1, \cdots, n\}$
ConvSE3 Layer C
$f_{output,i} \in \mathbb{R}^{['0':\ 3,'1':\ 3]} \cong \mathbb{R}^{12}$
$i \in \{1, \cdots, n\}$

Figure 2: Model Architecture: For all nodes 1 to n, the input feature vector of dimension 70 is transformed via an ”Attention Block A” (see Terminology 2) to 32 scalars, 3-dimensional vectors and 5-dimensional vectors each. Hence, the dimension of the hidden feature vector sums up to $288 = 32 \cdot (1+3+5)$. Further six layers of ”Attention Block B” (see Terminology 3) transform to the same dimension. Finally the ”ConvSE3 Layer C” (see Terminology 4) transforms to the output feature vector of dimension 12 (three scalars and three 3-dimensional vectors each).

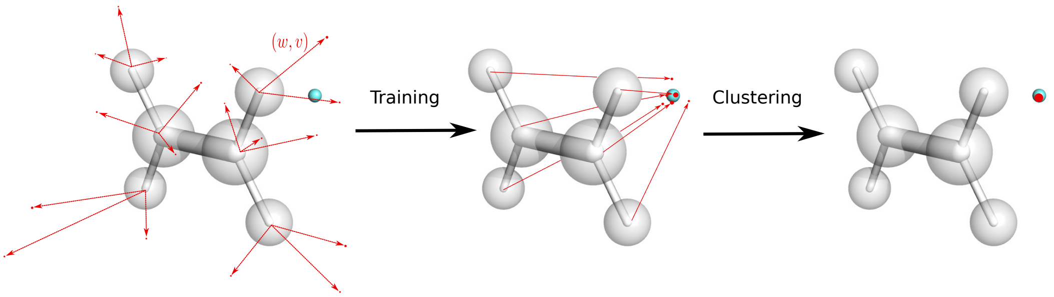

Figure 3: First Image: Per atom the model predicts three tupels (w, v) of weights and vectors. The size of the red dots indicates the value of the weights. The weights together with the endpoints of these vectors determine a Gaussian Mixture Model. Second Image: Using the symmetrized Kullback Leibler divergence as loss function, the model learns to form prediction point clouds around the true field points (only one vector per atom shown). Third Image: Finally, we apply a clustering algorithm to obtain a localized prediction.

Consider a molecule consisting of $n \in \mathbb{N}$ nodes. For node $i \in \{1, \ldots, n\}$, denote by $w_{i,1}, w_{i,2}, w_{i,3} \in \mathbb{R}$ the predicted scalars (used to determine the probability weights in the following) and by $v_{i,1}, v_{i,2}, v_{i,3} \in \mathbb{R}^3$ the predicted vectors (used to point to the field points relative to the atom position in the following), respectively. By applying a softmax function, the scalars $(w_{i,j})_{(i,j)\in\{1,\cdots,n\}\times\{1,2,3\}}$ determine a probability distribution as follows: For $i \in \{1, \cdots, n\}, j \in \{1, 2, 3\}$, define

$$q_{i,j} := \frac{\exp(w_{i,j})}{\sum\limits_{(k,l)\in\{1,\cdots,n\}\times\{1,2,3\}} \exp(w_{k,l})} \in [0, 1].$$

Denote by $a_i = (a_{i,1}, a_{i,2}, a_{i,3}) \in \mathbb{R}^3, i \in \{1, \cdots, n\}$ the coordinate position of the i'th node (i.e. atom). For the purpose of training, we interpret the predicted probabilities and vectors as determining weights and centers of a Gaussian Mixture Model as follows:

$$q(x) := \sum_{(i,j)\in\{1,\cdots,n\}\times\{1,2,3\}} q_{i,j} \cdot \phi_{a_i+v_{i,j},0.2}(x), \tag{1}$$

where we denote by

$$\phi_{\mu,\sigma}(x) = \frac{1}{\sqrt{(2\pi)^3\sigma^3}} \exp\left(-\frac{1}{2\sigma^2}||x-\mu||_2^2\right)$$

the isotropic Gaussian density function in 3 dimensions with mean $\mu \in \mathbb{R}^3$ and standard deviation $\sigma \in \mathbb{R}_+$. For a molecule with $m \in \mathbb{N}$ field points of a certain type (e.g. hydrophobic), let

$$(f_j^x, f_j^v) \in \mathbb{R}^3 \times \mathbb{R},\, j \in \{1, \cdots, m\}$$

denote the coordinate position and field value of the j'th field point. We define probabilities $p_j \in [0, 1], j \in \{1, \ldots, m\}$ in proportion to the field values:

$$\forall j \in \{1, \ldots, m\}: \qquad p_j := \frac{|f_j^v|}{\sum_{i=1}^m |f_i^v|} \in [0, 1]. \tag{2}$$

Analogously to (1), the probability weights $(p_j)_{j=1}^m$ and true field point locations $(f_j^x)_{j=1}^m$ determine a Gaussian Mixture Model as

$$p(x) := \sum_{j=1}^m p_j \cdot \phi_{f_j^x,0.2}(x). \tag{3}$$

Note that weighting by field values as in (2) takes into account that larger field values are more relevant in determining the binding properties of the molecule and thus more important to predict correctly.

By construction of q and p (see (1), (3)) we can expect that density q being approximately similar to p will result in reasonable field point predictions. A natural choice as a measure of divergence between two densities is the symmetrized Kullback-Leibler divergence:

$$\begin{aligned} L_1(p, q) &= KL(p|\, q) + KL(q|\, p) \\ &= \int_{\mathbb{R}^3} p(x) \cdot \log\left(\frac{p(x)}{q(x)}\right) dx + \int_{\mathbb{R}^3} q(x) \cdot \log\left(\frac{q(x)}{p(x)}\right) dx \end{aligned} \tag{4}$$

Let us denote by $\underline{q} := ((q_{i,j}, v_{i,j}))_{(i,j)\in\{1,\cdots,n\}\times\{1,2,3\}}$ and $\underline{p} := \left((p_j, f_j^x)\right)_{j=1}^m$ the probabilities and vectors determining the densities q and p respectively (compare (1) and (3)). Since calculating the quantity (4) is analytically intractable and computationally (e.g. via Monte Carlo simulations) demanding, we replace it by the following loss:

$$L_2(\underline{p}, \underline{q}) = \sum_{j=1}^m p_j \log\left(\frac{p_j}{q(f_j^x)}\right) + \sum_{\substack{i\in\{1,\cdots,n\} \\ j\in\{1,2,3\}}} q_{i,j} \log\left(\frac{q_{i,j}}{p(a_i + v_{i,j})}\right) \tag{5}$$

Note that intuitively, the equation (5) can be thought of as taking the discrete probability at support points $f_1^x, .., f_m^x$ and $a_i + v_{i,j}, (i,j) \in \{1, \cdots, n\} \times \{1,2,3\}$ respectively, as the first argument of the KL divergence in both terms of (4).

Each field point across different molecules should be equally important (proportionate by field value) to be predicted correctly. Hence, we need to scale the loss L_2 by the sum of field values to obtain:

$$L_3(\underline{p}, \underline{q}) = \left(\sum_{j=1}^{m} f_j^v \right) \cdot L_2(\underline{p}, \underline{q}).$$

During training we observed that a small penalization on the length of prediction vectors is essential for the preservation of locality and training convergence. Moreover, a penalization of large probability weights in the form of a quadratic sum of probability weights was beneficial to cause the model to predict clouds of predictions instead of few high probability vectors, leading to more robustness and better performance. Including both penalization terms, we define the final loss function as

$$L_4(\underline{p}, \underline{q}) = \left(\sum_{j=1}^{m} f_j^v \right) \cdot \left(L_2(\underline{p}, \underline{q}) + \alpha \cdot \sum_{\substack{i \in \{1, \cdots, n\} \\ j \in \{1,2,3\}}} q_{i,j} \cdot ||v_{i,j}||_2 + \beta \cdot \sum_{\substack{i \in \{1, \cdots, n\} \\ j \in \{1,2,3\}}} q_{i,j}^2 \right), \tag{6}$$

where $\alpha = \frac{1}{10}$ and $\beta = 10$.

The model described so far, results in predictions forming point clouds around the target field point (see second image of Fig. 3). In order to obtain more localized and precise predictions, we apply a clustering algorithm. We choose an agglomerative clustering as implemented in Pedregosa et al. (2011) with a linkage distance threshold of 1 Å to obtain the final field point predictions (third image of Fig. 3). The probability weights of all predictions contained in a cluster are summed up and a prediction is made at the weighted average position, if the sum exceeds a certain threshold $c \in [0,1]$. More specifically, for $k, M, k_1, \ldots, k_M \in \mathbb{N}$, let $\mathcal{C}_k$ be the k'th cluster found by the clustering algorithm, consisting of the probabilities and coordinates $(q_{k_1}, u_{k_1}), \ldots, (q_{k_M}, u_{k_M}) \in \mathbb{R} \times \mathbb{R}^3$:

$$\mathcal{C}_k = \{(q_{k_1}, u_{k_1}), \ldots, (q_{k_M}, u_{k_M})\}.$$

Then the k'th field point prediction suggested by the model is:

$$(q_{\mathcal{C}_k}, u_{\mathcal{C}_k}) = \left(\sum_{j=1}^{M} q_{k_j}, \frac{1}{\sum_{l=1}^{M} q_{k_l}} \sum_{j=1}^{M} q_{k_j} u_{k_j} \right).$$

We decide to make a prediction at the point $u_{\mathcal{C}_k}$ if the associated cluster probability $q_{\mathcal{C}_k}$ exceeds the threshold $c \in [0,1]$, meaning

$$q_{\mathcal{C}_k} > c. \tag{7}$$

Assuming that $l \in \mathbb{N}$ clusters were predicted, we denote by

$$\underline{u}_{\mathcal{C}} := (u_{\mathcal{C}_1}, \cdots, u_{\mathcal{C}_l}) \tag{8}$$

the tuple of predicted cluster locations.

4 Results

The data set is split randomly into training set (80%) and test set (20%) on molecule basis (all conformations of one molecule will be in the same set). We train on a GPU of 24 GB memory (GeForce RTX 3090) with a batch size of 50 (accumulated batch size is 25000) for about 5 days. In order to quantify the quality of our field point predictions, let us introduce the true positive rate and the weighted true positive rate (e.g. weighted sensitivity) as evaluation functions: For $r > 0$ and $\underline{u}_{\mathcal{C}}$ as in (8), define

$$L^r_{TPR}(\underline{p}, \underline{u}_{\mathcal{C}}) \quad := \quad \frac{1}{m} \sum_{j=1}^{m} \left(1 - \prod_{i=1}^{l} \mathbb{1}_{]r,\infty[}(||f^x_j - u_{\mathcal{C}_i}||_2) \right) \tag{9}$$

$$L^r_{WTPR}(\underline{p}, \underline{u}_{\mathcal{C}}) \quad := \quad \sum_{j=1}^{m} p_j \left(1 - \prod_{i=1}^{l} \mathbb{1}_{]r,\infty[}(||f^x_j - u_{\mathcal{C}_i}||_2) \right) \tag{10}$$

To put the quantity (10) into context, we also calculate the positive predictive value (precision):

$$L^r_{PPV}(\underline{p}, \underline{u}_{\mathcal{C}}) := \frac{1}{l} \sum_{i=1}^{l} \left(1 - \prod_{j=1}^{m} \mathbb{1}_{]r,\infty[}(||f^x_j - u_{\mathcal{C}_i}||_2) \right).$$

Note that $L^r_{TPR}(\underline{p}, \underline{u}_{\mathcal{C}})$ (equation (9)) corresponds to the proportion of ground truth field points that were predicted by the model (in the sense that at least one prediction is within the distance $r > 0$). The quantity $L^r_{WTPR}(\underline{p}, \underline{u}_{\mathcal{C}})$ (equation (10)) is defined similarly, however weights the correct prediction of each ground truth field point in proportion to its field value. The measure $L^r_{WTPR}(\underline{p}, \underline{u}_{\mathcal{C}})$ might be more relevant, as it is indeed more important to predict field points of high value correctly. The precision $L^r_{PPV}(\underline{p}, \underline{u}_{\mathcal{C}})$ corresponds to the proportion of predictions that are closer than distance $r > 0$ to a ground truth field point. In the following analysis, we chose the cluster probability threshold $c = 0.005$ for predicting a field point at a cluster center (compare (7)) for all field point types. Results for ranging cutoff values can be found in Figure 6.

The results for each field point type and maximal distance r from ground truth position are shown for the test set in Table 1 and for the training set in Table 4. Note that larger values for $\hat{L}^r_{WTPR}, \hat{L}^r_{TPR}$ and $\hat{L}^r_{PPV}$ are considered as higher prediction quality, with 1 being the optimum. Per construction, all three evaluation quantities are monotonically increasing with r. We observe that there is little difference in model performance between training and test set for $r \geq 1$. Slight overfitting can be noticed for $r = 0.5$. For all field points we achieve a precision of at least 80% for $r = 1$, meaning that only few predictions are far off from a ground truth field point.

The model performs slightly better for positive electrostatic field points than for negative electrostatic field points, specifically for $r = 0.5$ and $r = 1$ in terms of precision. However, with a weighted true positive rate of 0.865 (negative field point) and 0.858 (positive field point) at $r = 1$ on the test set, the model seems to capture the majority of high value field points for both field point types.

Figure 4 shows three examples for (a) negative and (b) positive field points highlighting the overall excellent performance of the model to reproduce the ground truth points. Not surprisingly, field points that are predominantly caused by single nearby polar atoms with highly negative or positive partial charge are precisely reproduced (e.g. field points n1, n9, n11, p3, p15, p16, etc.). However, the model also learns field points that originate from the electrostatic potential from multiple, sometimes topologically distant, polar atoms (e.g. n2, n4-6, p1, p2, p8, p20, etc.). Negative and positive field points originate not only from atoms that can undergo hydrogen bonding such as oxygen and nitrogen, but also halogen atoms (n3) or electropositive hydrogen atoms (e.g. p18-21). This demonstrates that the network model not only learns trivial projections from isolated atoms but the topological and spatial context of the molecules. The latter is modeled in our network by defining graph edges with a maximum Euclidean distance of 7 Å. Whereas the dominant field points (e.g. n2, n4-7, n9) are all well-reproduced, sometimes weak field points are not predicted (e.g. n8, n10) or

Results on test set								
	Negative field points				Positive field points			
$r =$	0.5	1	1.5	2	0.5	1	1.5	2
$\hat{L}^r_{WTPR}$	0.786	0.865	0.884	0.896	0.803	0.858	0.872	0.883
$\hat{L}^r_{TPR}$	0.706	0.786	0.805	0.818	0.738	0.793	0.807	0.819
$\hat{L}^r_{PPV}$	0.642	0.806	0.864	0.898	0.735	0.836	0.876	0.898
	Hydrophobic field points				van der Waals field points			
$r =$	0.5	1	1.5	2	0.5	1	1.5	2
$\hat{L}^r_{WTPR}$	0.953	0.970	0.982	0.990	0.574	0.637	0.667	0.701
$\hat{L}^r_{TPR}$	0.940	0.959	0.972	0.983	0.530	0.585	0.612	0.644
$\hat{L}^r_{PPV}$	0.862	0.940	0.968	0.985	0.768	0.893	0.939	0.964

Table 1: Field point prediction results on test set.

their position is shifted (e.g. p1-2). In the case of n10, the inherent flexibility of the nearby hydroxyl group strongly influences the position of this field point. Thus, field points related to such flexible hydroxyl groups (and similar functional groups) will have variable positions dependent on the generated rotation state of the functional group. This variability within the training set, makes it for the model very difficult to learn coherent rules of field point generation.

The best performance of the model was achieved for hydrophobic field points (Figure 4 c). For $r = 1$ on the test set, the precision ($\hat{L}^r_{PPV} = 0.940$) as well as the sensitivity ($\hat{L}^r_{WTPR} = 0.970$, $\hat{L}^r_{TPR} = 0.959$) indicate that almost all ground truth field points were predicted by the model, with very few far-off predictions. Many hydrophobic field points are located on hydrophobic atoms (e.g. h1, h9) or in aromatic rings (e.g. h2, h3, h5-7, etc.). Interestingly, the model is able to differentiate between homocyclic rings (e.g. h2, h3), where the field point is co-localized with the center of mass of the ring, and heterocyclic rings (e.g. h5-7), where the field point is shifted due to the presence of polar atoms within the ring structure. The right-most molecule in Figure 4 c, shows the challenging case of a long aliphatic chain. Most field points were well reproduced but some spatially nearby field points (e.g. h15-18) were compressed into single field points. This behavior is due to the applied clustering algorithm.

For the van der Waals field points, the model achieves a high precision of 0.893 at a distance of $r = 1$. However, compared to the other field point types the model predicts relatively few ground truth field points ($\hat{L}^r_{WTPR} = 0.637$). In Figure 4 d it can be observed that several ground truth field points were not predicted at all, though most existing predictions are accurate. Presumably, the prediction task is more difficult than for other field point types, since the attractive component of the van der Waals fields is weak and more homogenously distributed over the molecular surface. This results in a large number of field points representing shallow energy minima which are less well defined compared to the other field points.

To study the dependency of prediction performance on graph topology, the same network architecture was trained using a graph with edges defined by the covalent bonds. Table 5 and Figure 7 show the inferior performance of this model. The main factor for this performance drop is the lack of information flow between spatially close but topologically distant atoms. For example, field point 1 (in Figure 7, right) is the result of the negative partial charges of the carbonyl atom and the single aromatic ring. Thus, the graph defined by covalent bonds only is unable to capture this information as those functional groups have a relatively large topological distance, i.e. there is a lack of information flow in the SE(3)-Transformer model. The model instead predicts field point c1 based on the carbonyl atom and c2 based on the aromatic ring, both ignoring the correlative effect of those functional groups. The same can be observed for field point 3 which is based on the fields from the aromatic ring and secondary amine. The model predicts additional field points c1 and c3 which are not present in the ground truth data. Those field points do not exist in reality as the negative potential from the corresponding carbonyl atoms is largely cancelled by the positive methyl group

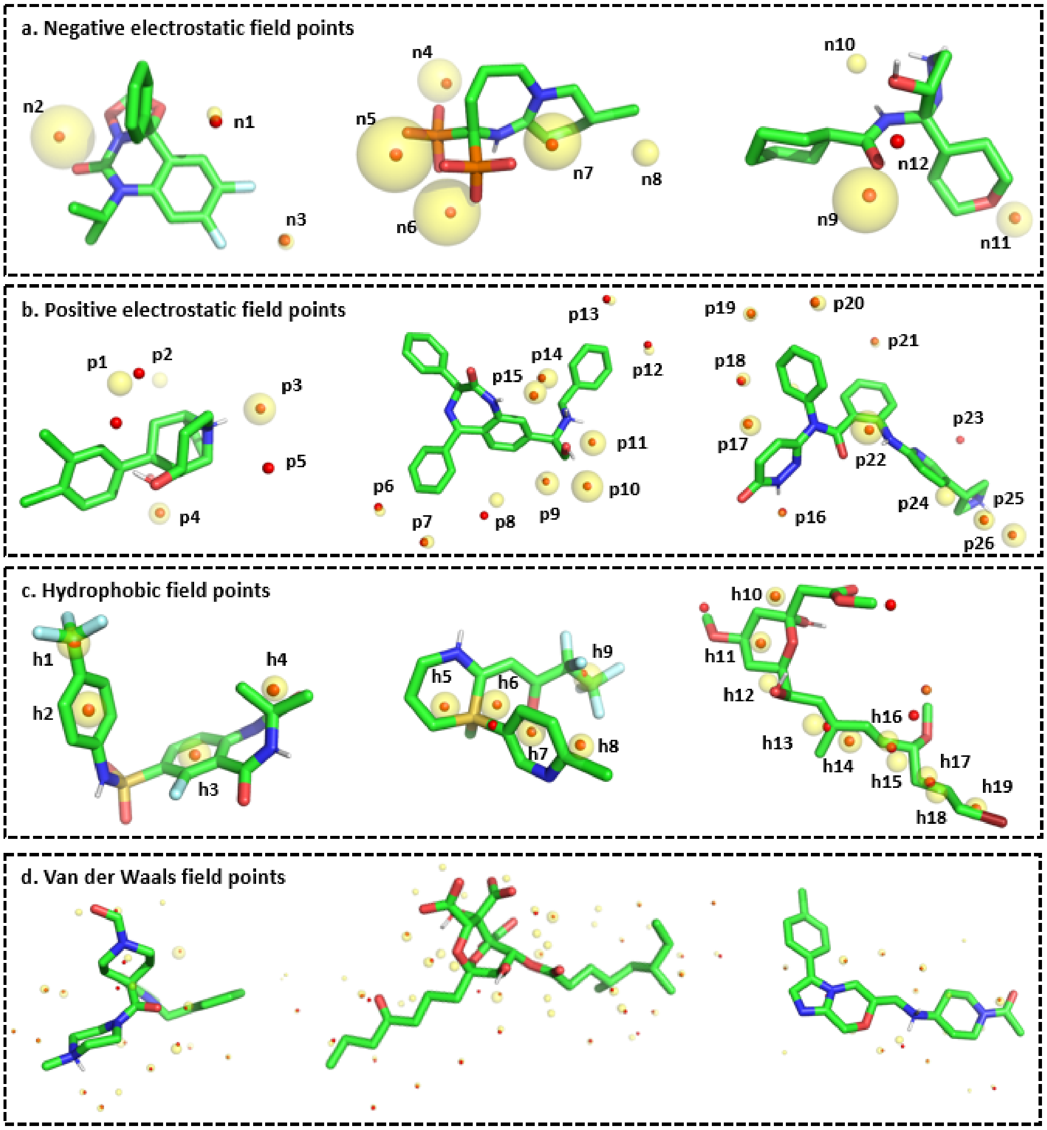

Figure 4: Examples of (a) negative, (b) positive, (c) hydrophobic and (d) van der Waals field points. Shown are ground truth field points as transparent yellow spheres and predicted field points as solid red spheres. Radii of spheres representing ground truth field points are scaled by field point value, i.e. strength of field point.

and positive amine, respectively for c1 and c3. The model that is based on information flow via covalent bonds is unable to correctly capture those physical effects as the atoms with opposite partial charge are not topologically adjacent to each other.

5 Conclusion

We demonstrated the benefits of an equivariant, attention based, graph neural network in the context of molecular field point prediction. A model based on the SE(3)-Transformer was trained using a large set of small molecules. Our model successfully predicts field points of different types of molecular interaction fields. In comparison to current methods of field point prediction that are based on computationally demanding calculations for the interaction energies between a probe and the molecule, our trained model allows for an efficient one-shot prediction of field points. We also demonstrated that field points that are spawned by topologically distant atoms can be reliably predicted if a graph structure was generated that is based on spatial rather than topological context of the molecule. The optimized model will allow in further research the use of physicochemical field point information for similarity based virtual screening on huge databases of compounds.

References

Tim Cheeseright, Mark Mackey, Sally Rose, and Andy Vinter. Molecular field extrema as descriptors of biological activity: Definition and validation. *Journal of Chemical Information and Modeling*, 46(2):665–676, 2006. doi: 10.1021/ci050357s. URL https://doi.org/10.1021/ci050357s. PMID: 16562997.

Timothy J. Cheeseright, Mark D. Mackey, James L. Melville, and Jeremy G. Vinter. Fieldscreen: Virtual screening using molecular fields. application to the dud data set. *Journal of Chemical Information and Modeling*, 48(11):2108–2117, 2008. doi: 10.1021/ci800110p. URL https://doi.org/10.1021/ci800110p. PMID: 18991371.

Timothy J. Cheeseright, Melanie Holm, Frank Lehmann, Sabine Luik, Marcia Göttert, James L. Melville, and Stefan Laufer. Novel lead structures for p38 map kinase via fieldscreen virtual screening. *Journal of Medicinal Chemistry*, 52(14):4200–4209, 2009. doi: 10.1021/jm801399r. URL https://doi.org/10.1021/jm801399r. PMID: 19489590.

Taco S. Cohen and Max Welling. Group equivariant convolutional networks. *CoRR*, abs/1602.07576, 2016. URL http://arxiv.org/abs/1602.07576.

Fabian B. Fuchs, Daniel E. Worrall, Volker Fischer, and Max Welling. Se(3)-transformers: 3d roto-translation equivariant attention networks, 2020.

Jan E. Gerken, Jimmy Aronsson, Oscar Carlsson, Hampus Linander, Fredrik Ohlsson, Christoffer Petersson, and Daniel Persson. Geometric deep learning and equivariant neural networks. *CoRR*, abs/2105.13926, 2021. URL https://arxiv.org/abs/2105.13926.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. URL https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791.

Caroline M. R. Low and J. G. Vinter. Rationalizing the activities of diverse cholecystokinin 2 receptor antagonists using molecular field points. *Journal of Medicinal Chemistry*, 51 (3):565–573, 2008. doi: 10.1021/jm070880t. URL https://doi.org/10.1021/jm070880t. PMID: 18201065.

H. L. Morgan. The generation of a unique machine description for chemical structures-a technique developed at chemical abstracts service. *Journal of Chemical Documentation*, 5(2):107–113, 1965. doi: 10.1021/c160017a018. URL https://doi.org/10.1021/c160017a018.

NVIDIA. Se(3)-transformers for pytorch. https://github.com/NVIDIA/DeepLearningExamples/tree/master/DGLPyTorch/DrugDiscovery/SE3Transformer, 2022.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011. URL https://scikit-learn.org/stable/modules/generated/sklearn.cluster.AgglomerativeClustering.html#sklearn.cluster.AgglomerativeClustering.

Nathaniel Thomas, Tess Smidt, Steven M. Kearnes, Lusann Yang, Li Li, Kai Kohlhoff, and Patrick Riley. Tensor field networks: Rotation- and translation-equivariant neural networks for 3d point clouds. *CoRR*, abs/1802.08219, 2018. URL http://arxiv.org/abs/1802.08219.

Appendix A: Terminology

In the following section, we denote by $\oplus$ the direct sum, $\otimes$ the Kronecker product and $I_3 \in \mathbb{R}^{3\times 3}, I_5 \in \mathbb{R}^{5\times 5}$ are the identity matrices in 3 and 5 dimensions respectively.

Definition 1 (Equivariance). *Let G be a group and X, Y be sets. Let $\triangleright_X$, $\triangleright_Y$ be group actions of G on X and Y respectively. A map $\Phi : X \longrightarrow Y$ is called $G-$equivariant if it satisfies:*

$$\forall g \in G, x \in X : \qquad \Phi(g \triangleright_X x) = g \triangleright_Y \Phi(x)$$

Moreover, in the special case of

$$\forall g \in G, x \in X : \qquad \Phi(g \triangleright_X x) = \Phi(x),$$

the map Φ is called $G-$invariant.

Terminology 1 (Rotation order, fiber structure). *A linear group representation of the group of rotations in 3 dimensions $SO(3)$ can be decomposed into irreducible representations of dimensions $2l+1$ for $l \in \mathbb{N} \cup \{0\}$. We refer to $l \in \mathbb{N} \cup \{0\}$ as the "rotation order". The rotation orders $l = 0, 1$ can be viewed as scalars and vectors in 3-dimensional space, respectively. If a feature vector v consists of $m_0, m_1, m_2 \in \mathbb{N}$ elements (also called "channels") of rotation orders $l = 0, 1, 2$, respectively, we say that its "fiber structure" is ['0': m_0; '1': m_1; '2': m_2]. Consequently the feature vector v is structured as follows:*

$$v \in \underbrace{\mathbb{R} \oplus \ldots \oplus \mathbb{R}}_{m_0 \ times} \oplus \underbrace{\mathbb{R}^3 \oplus \ldots \oplus \mathbb{R}^3}_{m_1 \ times} \oplus \underbrace{\mathbb{R}^5 \oplus \ldots \oplus \mathbb{R}^5}_{m_2 \ times} \cong \mathbb{R}^{m_0 + 3\cdot m_1 + 5\cdot m_2}.$$

Defining

$$\mathbb{R}^{[\text{'0': } m_0;\ \text{'1': } m_1;\ \text{'2': } m_2]} := \underbrace{\mathbb{R} \oplus \ldots \oplus \mathbb{R}}_{m_0 \ times} \oplus \underbrace{\mathbb{R}^3 \oplus \ldots \oplus \mathbb{R}^3}_{m_1 \ times} \oplus \underbrace{\mathbb{R}^5 \oplus \ldots \oplus \mathbb{R}^5}_{m_2 \ times},$$

we denote

$$v \in \mathbb{R}^{[\text{'0': } m_0;\ \text{'1': } m_1;\ \text{'2': } m_2]} \cong \mathbb{R}^{m_0 + 3\cdot m_1 + 5\cdot m_2}.$$

Moreover, for $\tilde{m}_0 \le m_0, \tilde{m}_1 \le m_1, \tilde{m}_2 \le m_2$ with $\tilde{m}_0, \tilde{m}_1, \tilde{m}_2 \in \mathbb{N}$ we agree to the notation

$$\begin{aligned} v^{[\text{'0': } \tilde{m}_0;\ \text{'1': } \tilde{m}_1;\ \text{'2': } \tilde{m}_2]} &:= \big(v_1, \cdots, v_{\tilde{m}_0}, v_{m_0+1}, \cdots, v_{m_0+3\cdot\tilde{m}_1}, \\ &\qquad v_{m_0+3\cdot m_1+1}, \cdots, v_{m_0+3\cdot m_1+5\cdot\tilde{m}_2}\big) \\ &\in \mathbb{R}^{[\text{'0': } \tilde{m}_0;\ \text{'1': } \tilde{m}_1;\ \text{'2': } \tilde{m}_2]} \cong \mathbb{R}^{\tilde{m}_0 + 3\cdot\tilde{m}_1 + 5\cdot\tilde{m}_2}, \end{aligned}$$

being a subvector of v consisting only of the first $\tilde{m}_0, \tilde{m}_1, \tilde{m}_2$ channels of rotation orders $0, 1$ and 2 respectively.

Terminology 2 (Attention Block A). *Consider a molecule consisting of $n \in \mathbb{N}$ atoms with coordinates $a_1, \ldots, a_n \in \mathbb{R}^3$. Per construction, for each atom $j \in \{1, \ldots, n\}$ we associate an initial feature vector $f_j \in \mathbb{R}^{[\text{'0': } 70]} \cong \mathbb{R}^{70}$. Thus, the molecule can be represented as a graph $G = (\mathcal{V}, E)$, with $\mathcal{V} = \{(1, a_1, f_1), (2, a_2, f_2), \cdots, (n, a_n, f_n)\}$ the set of nodes and E the set of edges connecting nodes. The attention block consists of the following four layers:*

1. *ConvSE3 Layer: For all $i, j \in \{1, \ldots, n\}$, calculate tensor field convolutions to obtain key and value vectors*

   $$\begin{aligned} k_{i,j} &= W_K(a_j - a_i) f_j \in \mathbb{R}^{[\text{'0': } 16]} \cong \mathbb{R}^{16} \\ v_{i,j} &= W_V(a_j - a_i) f_j \in \mathbb{R}^{[\text{'0': } 16, \text{'1': } 16,\ \text{'2': } 16]} \cong \mathbb{R}^{144} \end{aligned}$$

   where $W_K(a_j - a_i) \in \mathbb{R}^{16\times 70}$ and $W_V(a_j - a_i) \in \mathbb{R}^{144\times 70}$ are tensor field network type embedding matrices.

2. *LinearSE3 Layer: Calculate self-interaction to obtain the query vector*

   $$q_i = W_Q f_i \in \mathbb{R}^{[\text{'0': } 16]} \cong \mathbb{R}^{16},$$

   where $W_Q \in \mathbb{R}^{16\times 70}$.

3. *AttentionSE3 Layer: Calculate attention per node. For* $(i, a_i, f_i) \in \mathcal{V}$*, let* $N(i) \subset \{1, \cdots, n\}$ *denote the set of indices of neighbors of the i'th node. For* $i, j \in \{1, \ldots, n\}$*, define*

$$\begin{aligned} \alpha_{i,j} &= \frac{\exp\left(\frac{q_i^T k_{i,j}}{\sqrt{16}}\right)}{\sum\limits_{j' \in N(i)} \exp\left(\frac{q_i^T k_{i,j'}}{\sqrt{16}}\right)} \in [0,1] \\ f_{\text{att},i} &= \sum_{j \in N(i)} \alpha_{i,j} \cdot v_{i,j} \in \mathbb{R}^{['0'\text{: } 16,'1'\text{: } 16,\ '2'\text{: } 16]} \cong \mathbb{R}^{144} \end{aligned}$$

4. *LinearSE3 Layer: For* $i \in \{1, \cdots, n\}$*, concatenate the fibers of* f_i *and* $f_{\text{att},i}$*:*

$$\tilde{f}_i = \left(f_i^{['0'\text{: } 16]}, f_{\text{att,i}}\right) \in \mathbb{R}^{['0'\text{: } 32,'1'\text{: } 16,\ '2'\text{: } 16]} \cong \mathbb{R}^{160}.$$

Calculate the output vector

$$f_{out,i} = W_{out} \cdot \tilde{f}_i \in \mathbb{R}^{['0'\text{: } 32,'1'\text{: } 32,\ '2'\text{: } 32]} \cong \mathbb{R}^{288},$$

where $W_{out} \in \mathbb{R}^{288 \times 160}$ *is a block matrix, consisting of self-interaction submatrices of dimensions* 32×32*,* 96×48*,* 160×80*. More specifically*

$$W_{out} = W_{out}^0 \oplus \left(W_{out}^1 \otimes I_3\right) \oplus \left(W_{out}^2 \otimes I_5\right),$$

where $W_{out}^0 \in \mathbb{R}^{32 \times 32}, W_{out}^1 \in \mathbb{R}^{32 \times 16}, W_{out}^2 \in \mathbb{R}^{32 \times 16}$*. Note that the multiplication with* W_{out} *corresponds to a* 1×1 *convolution of channels within each rotation order.*

Terminology 3 (**Attention Block B**)**.** *Using the same notation as in 2, let* $a_i \in \mathbb{R}^3$ *denote the coordinates of the i'th node. The output of "Attention Block A" results in a feature vector* $f_i \in \mathbb{R}^{['0'\text{: } 32,'1'\text{: } 32,\ '2'\text{: } 32]} \cong \mathbb{R}^{288}$ *per node* $i \in \{1, \cdots, n\}$*. "Attention Block B" transforms this feature vector to a feature vector of the same fiber structure by applying the following four layers:*

1. *ConvSE3 Layer: For all* $i, j \in \{1 \ldots, n\}$*, calculate tensor field convolutions to obtain key and value vectors*

$$\begin{aligned} k_{i,j} &= W_K(a_j - a_i) f_j \in \mathbb{R}^{['0'\text{: } 16,'1'\text{: } 16,\ '2'\text{: } 16]} \cong \mathbb{R}^{144} \\ v_{i,j} &= W_V(a_j - a_i) f_j \in \mathbb{R}^{['0'\text{: } 16,'1'\text{: } 16,\ '2'\text{: } 16]} \cong \mathbb{R}^{144}, \end{aligned}$$

where $W_K(a_j - a_i) \in \mathbb{R}^{144 \times 288}$ *and* $W_V(a_j - a_i) \in \mathbb{R}^{144 \times 288}$ *are tensor field network type embedding matrices.*

2. *LinearSE3 Layer: Calculate self-interaction to obtain the query vector*

$$q_i = W_Q f_i \in \mathbb{R}^{['0'\text{: } 16,'1'\text{: } 16,\ '2'\text{: } 16]} \cong \mathbb{R}^{144},$$

where $W_Q \in \mathbb{R}^{144 \times 288}$ *is a block matrix, consisting of self-interaction submatrices of dimensions* 16×32*,* 48×96*,* 80×160*. More specifically*

$$W_Q = W_Q^0 \oplus \left(W_Q^1 \otimes I_3\right) \oplus \left(W_Q^2 \otimes I_5\right),$$

where $W_Q^0 \in \mathbb{R}^{32 \times 32}, W_Q^1 \in \mathbb{R}^{16 \times 32}, W_Q^2 \in \mathbb{R}^{16 \times 32}$*. Note that the multiplication with* W_Q *corresponds to a* 1×1 *convolution of channels within each rotation order.*

3. *AttentionSE3 Layer: Calculate attention per node. For* $(i, a_i, f_i) \in \mathcal{V}$*, let* $N(i) \subset \{1, \cdots, n\}$ *denote the set of indices of neighbors of the i'th node. For* $i, j \in \{1, \ldots, n\}$ *define*

$$\begin{aligned} \alpha_{i,j} &= \frac{\exp\left(\frac{q_i^T k_{i,j}}{\sqrt{144}}\right)}{\sum\limits_{j' \in N(i)} \exp\left(\frac{q_i^T k_{i,j'}}{\sqrt{144}}\right)} \in [0,1] \\ f_{\text{att},i} &= \sum_{j \in N(i)} \alpha_{i,j} \cdot v_{i,j} \in \mathbb{R}^{['0'\text{: } 16,'1'\text{: } 16,\ '2'\text{: } 16]} \cong \mathbb{R}^{144}. \end{aligned}$$

4. *LinearSE3 Layer: For $i \in \{1, \cdots, n\}$, concatenate the fibers of f_i and $f_{\text{att},i}$*

$$\begin{aligned}\tilde{f}_i &= \left(f_i^{[\text{'0': 32}]}, f_{\text{att},i}^{[\text{'0': 16}]}, f_i^{[\text{'1': 32}]}, f_{\text{att},i}^{[\text{'1': 16}]}, f_i^{[\text{'2': 32}]}, f_{\text{att},i}^{[\text{'2': 16}]}\right)\\ &\in \mathbb{R}^{[\text{'0': 48,'1': 48, '2': 48}]} \cong \mathbb{R}^{432}.\end{aligned}$$

Calculate the output vector

$$f_{out,i} = W_{out} \cdot \tilde{f}_i \in \mathbb{R}^{[\text{'0': 32,'1': 32, '2': 32}]} \cong \mathbb{R}^{288},$$

where $W_{out} \in \mathbb{R}^{288\times 432}$ is a block matrix, consisting of self-interaction submatrices of dimensions 32×48, 96×144, 160×240. More specifically

$$W_{out} = W_{out}^0 \oplus \left(W_{out}^1 \otimes I_3\right) \oplus \left(W_{out}^2 \otimes I_5\right),$$

where $W_{out}^0, W_{out}^1, W_{out}^2 \in \mathbb{R}^{32\times 48}$. Note that the multiplication with W_{out} corresponds to a 1×1 convolution of channels within each rotation order.

Terminology 4 (**ConvSE3 Layer C**)**.** *The output of ”Attention Block B” results in a feature vector $f_i \in \mathbb{R}^{288}$ (fiber structure [’0’: 32, ’1’: 32,’2’: 32]) per node $i \in \{1, \cdots, n\}$. The final ”ConvSE3 Layer” transforms f_i to the output feature vector $f_{out,i} \in \mathbb{R}^{12}$ of fiber structure [’0’: 3, ’1’: 3] as follows: For $i, j \in \{1, \ldots, n\}$ define*

$$\begin{aligned}h_{i,j} &= W_{out}(a_j - a_i) f_j \in \mathbb{R}^{[\text{'0': 3,'1': 3}]} \cong \mathbb{R}^{12}\\ f_{self,j} &= W_{self} \cdot f_j^{[\text{'0': 32,'1': 32}]} \in \mathbb{R}^{[\text{'0': 3,'1': 3}]} \cong \mathbb{R}^{12}\\ \tilde{f}_{i,j} &= h_{i,j} + f_{self,j} \in \mathbb{R}^{[\text{'0': 3,'1': 3}]} \cong \mathbb{R}^{12}\\ f_{out,i} &= \sum_{j \in N(i)} \tilde{f}_{i,j} \in \mathbb{R}^{[\text{'0': 3,'1': 3}]} \cong \mathbb{R}^{12}\end{aligned}$$

where $W_{out}(a_j - a_i) \in \mathbb{R}^{12\times 128}$ is a tensor field network type embedding matrix and $W_{self} \in \mathbb{R}^{12\times 288}$ is block matrix, consisting of self-interaction submatrices of dimensions 3×32 and 9×96. More specifically

$$W_{self} = W_{self}^0 \oplus \left(W_{self}^1 \otimes I_3\right),$$

where $W_{self}^0, W_{self}^1 \in \mathbb{R}^{3\times 32}$. Note that the multiplication with W_{self} corresponds to a 1×1 convolution of channels within each rotation order.

Appendix B: Tables

Molecule information						
Coordinate	N	A	P	S	W	D
$(-0.571, -0.329, 0.007)$	C	2	-0.1340	1.7	0.08129	3
$(0.571, 0.329, -0.007)$	C	2	-0.2480	1.7	0.08129	3
$(-0.574, -1.321, 0.027)$	H	15	0.0460	1.2	0	1
$(-1.431, 0.164, -0.003)$	H	15	0.0460	1.2	0	1
$(1.431, -0.164, 0.003)$	H	15	0.0830	1.2	0	1
$(0.574, 1.321, -0.027)$	H	15	0.0830	1.2	0	1

Table 2: Features per atom for a sample molecule. Abbreviations: N: atom name, A: atom type, P: partial charge, S: atom size, W: Wildman-Crippen logP, D: node degree

Field point information		
Coordinate	Field value	Field point type
$(0.028, 0.069, 2.488)$	-6.5370	F-
$(0.028, -0.032, -2.489)$	-6.5370	F-
$(-3.251, -1.860, -0.082)$	-0.6560	FO
$(1.078, -1.915, -2.090)$	-0.9210	FO
$(-1.083, 1.828, -2.165)$	-0.9210	FO
$(1.077, -1.816, 2.175)$	-0.9210	FO
$(-0.024, 0.108, 2.960)$	-0.8500	FO
$(-1.079, 1.911, 2.092)$	-0.9210	FO
$(-0.001, 0.001, -0.004)$	4.2050	FI

Table 3: Field point information: F-: electrostatic negative, F+: electrostatic positive, FI: van der Vaals, FO: hydrophobic

Results on training set								
	Negative field points				Positive field points			
$r =$	0.5	1	1.5	2	0.5	1	1.5	2
$\hat{L}^r_{WTPR}$	0.801	0.868	0.883	0.894	0.816	0.861	0.872	0.882
$\hat{L}^r_{TPR}$	0.713	0.782	0.799	0.811	0.747	0.793	0.805	0.816
$\hat{L}^r_{PPV}$	0.658	0.813	0.868	0.901	0.753	0.845	0.881	0.901

	Hydrophobic field points				Van der Waals field points			
$r =$	0.5	1	1.5	2	0.5	1	1.5	2
$\hat{L}^r_{WTPR}$	0.958	0.973	0.984	0.991	0.575	0.634	0.664	0.698
$\hat{L}^r_{WTPR}$	0.946	0.961	0.973	0.983	0.531	0.583	0.609	0.641
$\hat{L}^r_{PPV}$	0.869	0.944	0.970	0.986	0.776	0.895	0.940	0.964

Table 4: Field point prediction results on training set

Results for negative field points								
	Distance-based graph topology				Molecule graph topology			
$r =$	0.5	1	1.5	2	0.5	1	1.5	2
$\hat{L}^r_{WTPR}$	0.786	0.865	0.884	0.896	0.464	0.653	0.720	0.759
$\hat{L}^r_{TPR}$	0.706	0.786	0.805	0.818	0.523	0.723	0.793	0.834
$\hat{L}^r_{PPV}$	0.642	0.806	0.864	0.898	0.344	0.516	0.620	0.708

Table 5: Comparison between distance-based graph topology (7 Å) and molecule graph topology (defined by covalent bonds) on negative electrostatic field points.

Appendix C: Graphics

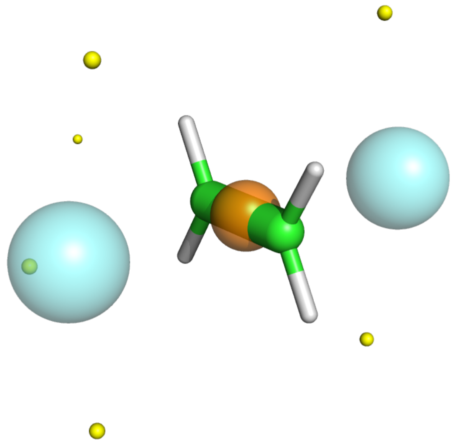

Figure 5: Visualization of a simple molecule (consisting only of hydrogen and carbon atoms) and its field points (aquamarine: electrostatic negative, orange: hydrophobic, yellow: van der Waals).

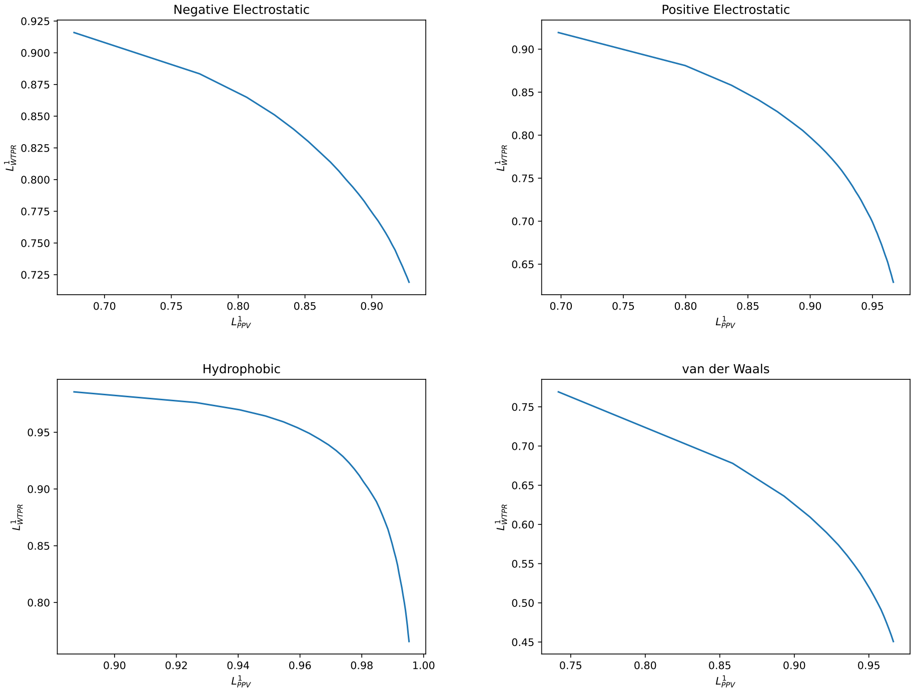

Figure 6: Weighted sensitivity (L^1_{WTPR}) vs precision (L^1_{PPV}) for the four field point types. The distance $r = 1$ was fixed (compare Tables 4 to 1) and the quantities L^1_{WTPR}, L^1_{PPV} calculated for 50 equidistant cutoff values between $c \in [0.001, 0.1]$ (compare 7).

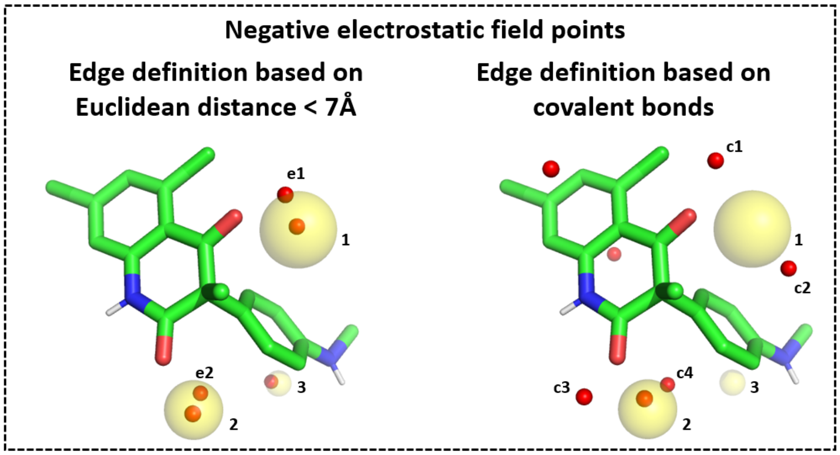
Negative electrostatic field points
Edge definition based on Euclidean distance < 7Å
Edge definition based on covalent bonds
e1
1
e2
2
3
c1
1
c2
c3
c4
2
3

Figure 7: Example of comparison of different graph topologies used to predict negative field points. Left: Edges of graph defined by Euclidean distance between atoms being smaller than 7 Å. Right: Edge of graph defined based on covalent bonds. Shown are ground truth field points as transparent yellow spheres and predicted field points as solid red spheres. Radii of spheres representing ground truth field points are scaled by field point value, i.e. strength of field point.