INDEPENDENT SE(3)-EQUIVARIANT MODELS FOR END-TO-END RIGID PROTEIN DOCKING

Anonymous authors
Paper under double-blind review

ABSTRACT

Protein complex formation is a central problem in biology, being involved in most of the cell's processes, and essential for applications such as drug design or protein engineering. We tackle *rigid body protein-protein docking*, i.e., computationally predicting the 3D structure of a protein-protein complex from the individual unbound structures, assuming no three-dimensional flexibility during binding. We design a novel pairwise-independent SE(3)-equivariant graph matching network to predict the rotation and translation to place one of the proteins at the right location and the right orientation relative to the second protein. We mathematically guarantee that the predicted complex is always identical regardless of the initial placements of the two structures, avoiding expensive data augmentation. Our model approximates the binding pocket and predicts the docking pose using keypoint matching and alignment through optimal transport and a differentiable Kabsch algorithm. Empirically, we achieve significant running time improvements over existing protein docking software and predict qualitatively plausible protein complex structures despite not using heavy sampling, structure refinement, or templates.

1 INTRODUCTION

In a recent breakthrough, ALPHAFOLD 2 (Jumper et al., 2021; Senior et al., 2020) has provided a solution to a grand challenge in biology—inferring a protein's three-dimensional structure from its amino acid sequence (Baek et al., 2021), following the dogma *sequence determines structure*.

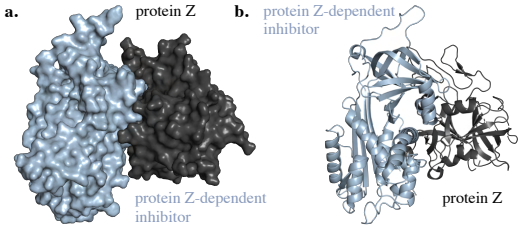

Figure 1: **Different views of the 3D structure of a protein complex. a.** Surface and **b.** cartoon view of protein Z and its inhibitor.

Besides their complex three-dimensional nature, proteins dynamically alter their function and structure in response to cellular signals, changes in the environment, or upon *molecular docking*. In particular, protein-protein or protein-ligand interactions are involved in various biological processes including signal transduction, protein synthesis, DNA replication and repair. Understanding how proteins behave in molecular docking is crucial to understanding their interaction mechanism and subsequently to developing therapeutic interventions that depend on drugs targeting the interactions. While we are able to accurately model a protein's structure, capturing the dynamic nature of proteins in biological interactions remains an open problem.

We here address the problem of *rigid body protein-protein docking* which refers to computationally predicting the 3D structure of a protein-protein complex given the 3D structures of the two proteins in unbound state. *Rigid body* means no deformations occur within a protein during binding with another protein which is a realistic assumption in many biological reactions.

Popular docking softwares (Chen et al., 2003; Venkatraman et al., 2009; De Vries et al., 2010; Torchala et al., 2013; Schindler et al., 2017; Sunny and Jayaraj, 2021) are typically computationally expensive, taking between minutes and hours to solve a single example pair, while not being guaranteed to find accurate complex structures. These methods largely follow the following steps: i.) randomly sample a large number (e.g., millions) of candidate initial complex structures, ii.) employ a scoring

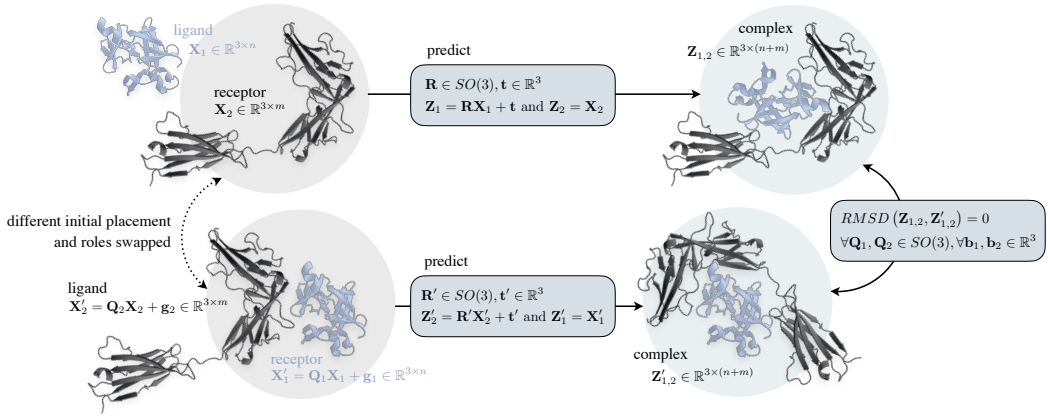

Figure 2: **Intuition of Independent SE(3)-Invariance.** Our method predicts a rotation and a translation transformation to place the ligand in the binding location. We mathematically guarantee to always predict the same final complex structure —up to an SE(3) transformation, i.e., zero RMSD— independent of the initial unbound positions, rotations, or identities of the ligand and receptor.

function to rank the candidates, iii.) adjust and refine the top complex structures based on an energy model (e.g., force field). We here take a first step towards tackling these issues by using deep learning models for direct prediction of protein complex structures.

Contributions. We design a fast, end-to-end method for rigid body docking that directly predicts the SE(3) transformation to place one of the proteins (ligand) at the right location and the right orientation with respect to the second protein (receptor). Our method explicitly incorporates the inductive biases of SE(3)-invariance of the final docking position and commutativity, i.e., the same complex structure is predicted irrespective of the initial 3D placements and roles of both the ligand and receptor (see Fig. 2). We achieve this desiderata by a combination of i.) a novel type of pairwise independent SE(3)-equivariant graph matching networks, ii.) an attention-based keypoint selection algorithm that discovers and aligns representative points with the binding pocket location using optimal transport, iii.) a differentiable superimposition model to recover the optimal rotation matrix and translation vector. We do not use heavy candidate sampling or ranking, templates, task-specific geometric or chemical hand-crafted features, or pre-computed meshes as done in prior work. This enables us to achieve plausible structures with a speed-up of 80-500x compared to popular docking software, offering a promising alternative to current solutions for this problem.

2 Related Work

Geometric Deep Learning. Graph Neural Networks (GNNs) are becoming the de facto choice for learning methods that deal with graph structured data (Bruna et al., 2013; Defferrard et al., 2016; Kipf and Welling, 2016; Gilmer et al., 2017; Li et al., 2019). Inspired by various priors about symmetries occurring naturally in different data types, recent architectures are taylored to explicitly incorporate such specific equivariance properties (Cohen and Welling, 2016a;b; Thomas et al., 2018; Fuchs et al., 2020; Finzi et al., 2020; Eismann et al., 2020; Satorras et al., 2021). These have demonstrated successes in a variety of tasks such as particle system dynamics or conformation-based energy estimation (Weiler and Cesa, 2019; Rezende et al., 2019; Romero and Cordonnier, 2020). An important property is the SE(3)-equivariance, being crucial for models operating on the structural data, like molecular structures (Fuchs et al., 2020; Hutchinson et al., 2020) and 3D point clouds (Thomas et al., 2018), due to their inherent rotational and translational symmetries. SE(3)-equivariance is also a necessary property for the problem of rigid protein docking, being, however, not sufficient to model the bi-molecular interaction processes. The goal of our work is to bridge this gap.

Protein Folding. Deep neural networks have been used to predict inter-residue contacts, distance and/or orientations (Adhikari and Cheng, 2018; Yang et al., 2020; Senior et al., 2020; Ju et al., 2021; Wu et al., 2021), that are subsequently transformed into additional constraints or differentiable energy terms for protein structure optimization. AlphaFold 2 (Jumper et al., 2021; Senior et al., 2020) and Rosetta Fold (Baek et al., 2021) are the class of state-of-the-art end-to-end approaches to

directly predict protein structures from co-evolution information embedded in homologous sequences. As a sub-task evaluated in protein folding, protein quality assessment estimates the quality of computational protein models in terms of divergence from their native structure, for which GNN architectures have also been applied (Sanyal et al., 2020; Baldassarre et al., 2021).

Protein-Protein Docking and Interaction. Experimentally determining structures of protein complexes is often expensive and time-consuming, rendering a premium on computational methods (Vakser, 2014). Protein docking methods (Chen et al., 2003; Venkatraman et al., 2009; De Vries et al., 2010; Biesiada et al., 2011; Torchala et al., 2013; Schindler et al., 2017; Weng et al., 2019; Sunny and Jayaraj, 2021; Christoffer et al., 2021; Yan et al., 2020) typically run several steps: first, they sample thousands or millions of candidates from the configurational space of the interacting proteins; second, they rank these candidate structures using a scoring function (Moal et al., 2013; Basu and Wallner, 2016; Launay et al., 2020; Eismann et al., 2020); finally, top-ranked candidates undergo a structure refinement process using energy or geometric models (Verburgt and Kihara, 2021). Peptide-protein docking is a subproblem of protein-protein docking, its performance has been refined recently using protein folding networks (Tsaban et al., 2021; Ko and Lee, 2021). Khramushin et al. (2021) conduct peptide-protein docking using surface motifs on the protein. As another subproblem in protein-protein interaction (PPI), protein interface prediction (Fout et al., 2017; Townshend et al., 2019; Liu et al., 2020; Xie and Xu, 2021; Dai and Bailey-Kellogg, 2021) aims to locate protein interfaces by identifying all contacted amino acid pairs. Very recently, ALPHAFOLD 2 and ROSETTAFOLD have been utilized as subroutines (Humphreys et al., 2021; Pei et al., 2021; Jovine) to improve PPIs from different aspects, e.g., combining physics-based docking method CLUSPRO (Kozakov et al., 2017) to improve protein-protein docking (Ghani et al., 2021), combining extended multiple-sequence alignments to predict the structure of heterodimeric protein complexes using only sequence information (Bryant et al., 2021), and in parallel with the development of this work, Evans et al. (2021) extend ALPHAFOLD 2 to multiple chains during both training and inference.

Drug-Target Interaction (DTI). DTI, aiming at elucidating the drug target binding poses and affinity, plays an essential role in drug discovery. Prior methods (Wallach et al., 2015; Li et al., 2021) try to predict binding affinity from protein-ligand co-crystal structure, which is highly costly to obtain from wet lab experiments. As a traditional approach for DTI, protein ligand docking provides a computational method to obtain the protein ligand complex structure. It is usually composed of two steps: sampling candidates from the vast conformational space of the molecules being docked; scoring each of the sampled docking pose (Bao et al., 2021). Various protein ligand docking software have been developed (Trott and Olson, 2010; Koes et al., 2013; McNutt et al., 2021), however, they inevitably suffer from the high computational cost. We note that these methods can not be applied to the protein rigid docking problem we study here, since they are especially designed for small molecule ligand (compounds), for which the binding pocket is usually known apriori. In contrast, our rigid docking approach could potentially be extended to accelerate the protein ligand docking process, which we leave as future work.

3 SE(3)-INVARIANT AND COMMUTATIVE RIGID BODY DOCKING

We start by introducing the rigid body docking problem and derive the necessary conditions for enforcing the useful inductive biases of SE(3)-invariance and commutativity.

Problem Setup of Rigid Protein-Protein Docking. We are given as input a pair of proteins forming a complex. They are (arbitrarily) denoted as ligand and receptor and consist of n and m residues, respectively. We represent the two proteins in their bound (docked) state as 3D point clouds $\mathbf{X}_1^* \in \mathbb{R}^{3\times n}, \mathbf{X}_2^* \in \mathbb{R}^{3\times m}$, where each residue's location is given by the coordinates of its corresponding α-carbon atom. In the unbound state, the docked ligand is randomly rotated and translated in space, resulting in a modified point cloud $\mathbf{X}_1 \in \mathbb{R}^{3\times n}$. For simplicity and w.l.o.g., the receptor remains in its bound location $\mathbf{X}_2 = \mathbf{X}_2^*$.

The task is to predict a rotation $\mathbf{R} \in SO(3)$ and a translation $\mathbf{t} \in \mathbb{R}^3$ such that $\mathbf{R}\mathbf{X}_1 + \mathbf{t} = \mathbf{X}_1^*$, based on the input proteins and their unbound positions $\mathbf{X}_1$ and $\mathbf{X}_2$.

Thus, $\mathbf{R} = \mathbf{R}(\mathbf{X}_1|\mathbf{X}_2)$ and $\mathbf{t} = \mathbf{t}(\mathbf{X}_1|\mathbf{X}_2)$ are functions of the two proteins, where we omit residue identity or other protein information in this notation, for brevity.

Note that we assume rigid backbone and side-chains for both proteins. We therefore do not tackle the more challenging problem of *flexible docking*, but our approach offers an important step towards it.

We desire that the predicted complex structure is independent of the initial locations and orientations of the two proteins, as well as of their roles – see Fig. 2. Formally, we wish to guarantee that:

$$\begin{aligned}
&(\mathbf{R}(\mathbf{Z}_1|\mathbf{Z}_2)\mathbf{Z}_1 + \mathbf{t}(\mathbf{Z}_1|\mathbf{Z}_2)) \oplus \mathbf{Z}_2 \equiv (\mathbf{R}(\mathbf{X}_1|\mathbf{X}_2)\mathbf{X}_1 + \mathbf{t}(\mathbf{X}_1|\mathbf{X}_2)) \oplus \mathbf{X}_2, \quad \text{(SE(3)-invariance)} \\
&(\mathbf{R}(\mathbf{X}_1|\mathbf{X}_2)\mathbf{X}_1 + \mathbf{t}(\mathbf{X}_1|\mathbf{X}_2)) \oplus \mathbf{X}_2 \equiv \mathbf{X}_1 \oplus (\mathbf{R}(\mathbf{X}_2|\mathbf{X}_1)\mathbf{X}_2 + \mathbf{t}(\mathbf{X}_2|\mathbf{X}_1)), \quad \text{(commutativity)} \\
&\forall \mathbf{Q}_1, \mathbf{Q}_2 \in SO(3), \forall \mathbf{g}_1, \mathbf{g}_2 \in \mathbb{R}^3, \forall \mathbf{X}_1 \in \mathbb{R}^{3\times n}, \mathbf{X}_2 \in \mathbb{R}^{3\times m}, \text{ and } \mathbf{Z}_l = \mathbf{Q}_l\mathbf{X}_l + \mathbf{g}_l, l \in \{1, 2\}.
\end{aligned} \tag{1}$$

for any rotations $\mathbf{Q}_1, \mathbf{Q}_2$ and translations $\mathbf{g}_1, \mathbf{g}_2$, where $\oplus$ is concatenation along columns, and $\equiv$ denotes identity after superimposition, formally defined as zero Root-Mean-Square Deviation (RMSD) between two 3D point sets after superimposing them using the Kabsch algorithm (Kabsch, 1976). An immediate question arises:

How do the constraints in Eq. (1) translate into constraints for $\mathbf{R}(\cdot|\cdot)$ and $\mathbf{t}(\cdot|\cdot)$?

The rotation matrix $\mathbf{R}$ and translation vector $\mathbf{t}$ change in a systematic way when we apply $SE(3)$ transformations on the two proteins, or when we swap their role. These properties restrict our class of functions as derived below.

SE(3)-Invariance Constraints. If we apply any distinct $SE(3)$ transformations on the unbound ligand $\mathbf{X}_1$ and receptor $\mathbf{X}_2$, i.e. we aim to dock $\mathbf{Q}_1\mathbf{X}_1 + \mathbf{g}_1$ onto $\mathbf{Q}_2\mathbf{X}_2 + \mathbf{g}_2$, then the rotation matrix $\mathbf{R}(\mathbf{Q}_1\mathbf{X}_1 + \mathbf{g}_1|\mathbf{Q}_2\mathbf{X}_2 + \mathbf{g}_2)$ and translation vector $\mathbf{t}(\mathbf{Q}_1\mathbf{X}_1 + \mathbf{g}_1|\mathbf{Q}_2\mathbf{X}_2 + \mathbf{g}_2)$ can be derived from $\mathbf{R}(\mathbf{X}_1|\mathbf{X}_2)$ and $\mathbf{t}(\mathbf{X}_1|\mathbf{X}_2)$ assuming that we always do rotations first. In this case, $\mathbf{R}(\mathbf{Q}_1\mathbf{X}_1 + \mathbf{g}_1|\mathbf{Q}_2\mathbf{X}_2 + \mathbf{g}_2)$ can be decomposed into three consecutive rotations: i.) apply $\mathbf{Q}_1^\top$ to undo the rotation $\mathbf{Q}_1$ applied on $\mathbf{X}_1$, ii.) apply $\mathbf{R}(\mathbf{X}_1|\mathbf{X}_2)$, iii.) apply $\mathbf{Q}_2$ to rotate the docked ligand together with the receptor. This gives $\mathbf{R}(\mathbf{Q}_1\mathbf{X}_1 + \mathbf{g}_1|\mathbf{Q}_2\mathbf{X}_2 + \mathbf{g}_2) = \mathbf{Q}_2\mathbf{R}(\mathbf{X}_1|\mathbf{X}_2)\mathbf{Q}_1^\top$, which in turn constraints the translation vector. We provide a formal statement (proven in Appendix B.1):

Proposition 1. *For any $\mathbf{Q}_1, \mathbf{Q}_2 \in SO(3), \mathbf{g}_1, \mathbf{g}_2 \in \mathbb{R}^3$, SE(3)-invariance of the predicted docked complex defined in Eq. (1) is guaranteed iff*

$$\begin{aligned}
\mathbf{R}(\mathbf{Q}_1\mathbf{X}_1 + \mathbf{g}_1|\mathbf{Q}_2\mathbf{X}_2 + \mathbf{g}_2) &= \mathbf{Q}_2\mathbf{R}(\mathbf{X}_1|\mathbf{X}_2)\mathbf{Q}_1^\top \\
\mathbf{t}(\mathbf{Q}_1\mathbf{X}_1 + \mathbf{g}_1|\mathbf{Q}_2\mathbf{X}_2 + \mathbf{g}_2) &= \mathbf{Q}_2\mathbf{t}(\mathbf{X}_1|\mathbf{X}_2) - \mathbf{Q}_2\mathbf{R}(\mathbf{X}_1|\mathbf{X}_2)\mathbf{Q}_1^\top\mathbf{g}_1 + \mathbf{g}_2
\end{aligned} \tag{2}$$

As a direct consequence of this proposition, we have the following statement.

Proposition 2. *Any model that satisfies Proposition 1 achieves invariance of the predicted complex w.r.t. any random SE(3) transformation on $\mathbf{X}_1$ conditioning on a fixed $\mathbf{X}_2$, as well as equivariance w.r.t. SE(3) transformations on $\mathbf{X}_2$:*

$$\begin{aligned}
&\mathbf{R}(\mathbf{Z}_1|\mathbf{X}_2)\mathbf{Z}_1 + \mathbf{t}(\mathbf{Z}_1|\mathbf{X}_2) = \mathbf{R}(\mathbf{X}_1|\mathbf{X}_2)\mathbf{X}_1 + \mathbf{t}(\mathbf{X}_1|\mathbf{X}_2), \quad \textit{where } \mathbf{Z}_1 = \mathbf{Q}_1\mathbf{X}_1 + \mathbf{g}_1 \\
&\mathbf{R}(\mathbf{X}_1|\mathbf{Z}_2)\mathbf{X}_1 + \mathbf{t}(\mathbf{X}_1|\mathbf{Z}_2) = \mathbf{Q}_2\left[\mathbf{R}(\mathbf{X}_1|\mathbf{X}_2)\mathbf{X}_1 + \mathbf{t}(\mathbf{X}_1|\mathbf{X}_2)\right] + \mathbf{g}_2, \quad \textit{where } \mathbf{Z}_2 = \mathbf{Q}_2\mathbf{X}_2 + \mathbf{g}_2 \\
&\forall \mathbf{Q}_1, \mathbf{Q}_2 \in SO(3), \forall \mathbf{g}_1, \mathbf{g}_2 \in \mathbb{R}^3, \forall \mathbf{X}_1 \in \mathbb{R}^{3\times n}, \forall \mathbf{X}_2 \in \mathbb{R}^{3\times m}.
\end{aligned}$$

Commutativity. Instead of docking $\mathbf{X}_1$ with respect to $\mathbf{X}_2$, we can also dock $\mathbf{X}_2$ with respect to $\mathbf{X}_1$. In this case, we require the final complex structures to be identical after superimposition, i.e., zero RMSD, which we denote as *commutativity*. This is satisfied as follows.

Proposition 3. *Commutativity as defined in Eq. (1) is guaranteed iff*

$$\mathbf{R}(\mathbf{X}_2|\mathbf{X}_1) = \mathbf{R}^\top(\mathbf{X}_1|\mathbf{X}_2); \quad \mathbf{t}(\mathbf{X}_2|\mathbf{X}_1) = -\mathbf{R}^\top(\mathbf{X}_1|\mathbf{X}_2)\mathbf{t}(\mathbf{X}_1|\mathbf{X}_2), \tag{3}$$

where we used the notation $\mathbf{R}^\top(\mathbf{X}_1|\mathbf{X}_2) := (\mathbf{R}(\mathbf{X}_1|\mathbf{X}_2))^\top$. The proof is provided in Appendix B.2.

Point Permutation Invariance. We also require permutation invariance w.r.t. all vectors of $\mathbf{X}_1$ and, respectively, of $\mathbf{X}_2$. Formally, both $\mathbf{R}(\mathbf{X}_1|\mathbf{X}_2)$ and $\mathbf{t}(\mathbf{X}_1|\mathbf{X}_2)$ should not depend on the order or columns of $\mathbf{X}_1$ and, respectively, of $\mathbf{X}_2$.

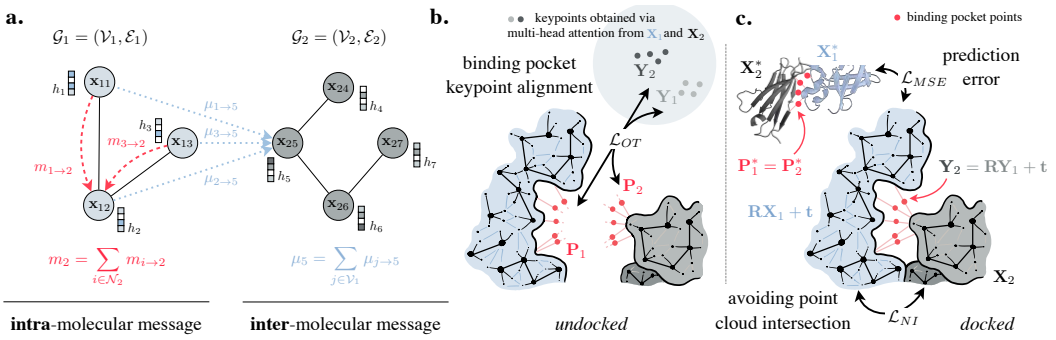

Figure 3: **Details on Model Architecture and Losses.** **a.** The message passing operations in IEGMN guarantee pairwise independent SE(3)-equivariance as in Eq. (4), **b.** We predict keypoints for each protein that are aligned with the binding pocket location using an additional optimal transport (OT) loss, **c.** After predicting the docked position, we compute an MSE loss on the ligand, as well as a loss to discourage body intersections.

4 MODEL

Proteins and their k-NN Graphs. A protein is a sequence of amino acid residues that folds in a 3D structure. Each residue has a general structure with a side-chain specifying its type, allowing us to define a *local frame* and derive SE(3)-invariant features for any pair of residues —see Appendix A.

We represent a protein as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, similar to Fout et al. (2017); Townshend et al. (2019); Liu et al. (2020). Each node $i \in \mathcal{V}$ represents one residue and has 3D coordinates $\mathbf{x}_i \in \mathbb{R}^3$ corresponding to the α-carbon atom's location. Edges are given by a k-nearest-neighbor (k-NN) graph using Euclidean distance of the 3D node coordinates.

Overview of our Approach. Our general strategy is summarized as follows. We first build k-NN protein graphs as described above, $\mathcal{G}_1 = (\mathcal{V}_1, \mathcal{E}_1)$ and $\mathcal{G}_2 = (\mathcal{V}_2, \mathcal{E}_2)$. We design SE(3)-invariant node features $\mathbf{F}_1 \in \mathbb{R}^{d \times n}, \mathbf{F}_2 \in \mathbb{R}^{d \times m}$ and edge features $\{\mathbf{f}_{j \to i} : \forall (i,j) \in \mathcal{E}_1 \cup \mathcal{E}_2\}$ (see Appendix A).

We then apply several layers of a function Φ that guarantees, by design, the so called *pairwise independent SE(3)-equivariance* for two sets of 3D coordinates, as well as invariance for the node latent embeddings. Formally, this is defined as:

$$\begin{aligned} &\text{Given } \mathbf{Z}_1, \mathbf{H}_1, \mathbf{Z}_2, \mathbf{H}_2 = \Phi(\mathbf{X}_1, \mathbf{F}_1, \mathbf{X}_2, \mathbf{F}_2) \\ &\text{we have } \mathbf{Q}_1\mathbf{Z}_1 + \mathbf{g}_1, \mathbf{H}_1, \mathbf{Q}_2\mathbf{Z}_2 + \mathbf{g}_2, \mathbf{H}_2 = \Phi(\mathbf{Q}_1\mathbf{X}_1 + \mathbf{g}_1, \mathbf{F}_1, \mathbf{Q}_2\mathbf{X}_2 + \mathbf{g}_2, \mathbf{F}_2), \\ &\quad \forall \mathbf{Q}_1, \mathbf{Q}_2 \in SO(3), \forall \mathbf{g}_1, \mathbf{g}_2 \in \mathbb{R}^3. \end{aligned} \tag{4}$$

We implement Φ as a special type of message-passing neural network (MPNN), as described below. We further use the output of this MPNN to compute parametric functions $\mathbf{R}(\mathbf{X}_1|\mathbf{X}_2;\theta)$ and $\mathbf{t}(\mathbf{X}_1|\mathbf{X}_2;\theta)$. These functions should depend on pairwise interactions between the two proteins modeled as cross-messages, but also incorporate the 3D structure in a pairwise-independent SE(3)-equivariant way to satisfy Eq. (1), Proposition 1 and Proposition 3. We achieve this by discovering keypoints from each protein based on a neural attention mechanism, train them to softly represent the respective binding pocket locations via an optimal transport based auxiliary loss and, finally, obtain the SE(3) transformation by superimposing them via a differentiable version of the Kabsch algorithm. An additional soft-constraint is designed to discourage point cloud intersections.

We now detail each of these model components.

Independent E(3)-Equivariant Graph Matching Networks (IEGMNs). We design the function Φ to satisfy pairwise independent coordinate SE(3)-equivariance in Eq. (4). Our architecture, called *Independent E(3)-Equivariant Graph Matching Network* (IEGMN), extends both Graph Matching Networks (GMN) (Li et al., 2019) and E(3)-Equivariant Graph Neural Networks (E(3)-GNN) (Satorras et al., 2021). IEGMNs process pairs of graphs $\mathcal{G}_1 = (\mathcal{V}_1, \mathcal{E}_1)$, $\mathcal{G}_2 = (\mathcal{V}_2, \mathcal{E}_2)$ with inter-graph messages, as well as E(3)-equivariant coordinate updates. The l-th layer of IEGMNs transforms node latent embeddings $\{\mathbf{h}_i^{(l)}\}_{i \in \mathcal{V}_1 \cup \mathcal{V}_2}$ and node coordinates $\{\mathbf{x}_i^{(l)}\}_{i \in \mathcal{V}_1 \cup \mathcal{V}_2}$ as

$$\mathbf{m}_{j\to i} = \varphi^e(\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)}, \exp(-\|\mathbf{x}_i^{(l)} - \mathbf{x}_j^{(l)}\|^2/\sigma), \mathbf{f}_{j\to i}), \forall e_{j\to i} \in \mathcal{E}_1 \cup \mathcal{E}_2 \tag{5}$$

$$\mu_{j\to i} = a_{j\to i}\mathbf{W}\mathbf{h}_j^{(l)}, \forall i \in \mathcal{V}_1, j \in \mathcal{V}_2 \text{ or } i \in \mathcal{V}_2, j \in \mathcal{V}_1 \tag{6}$$

$$\mathbf{m}_i = \frac{1}{|\mathcal{N}(i)|}\sum_{j\in\mathcal{N}(i)} \mathbf{m}_{j\to i}, \forall i \in \mathcal{V}_1 \cup \mathcal{V}_2 \tag{7}$$

$$\mu_i = \sum_{j\in\mathcal{V}_2} \mu_{j\to i}, \forall i \in \mathcal{V}_1 \tag{8}$$

$$\mu_i = \sum_{j\in\mathcal{V}_1} \mu_{j\to i}, \forall i \in \mathcal{V}_2 \tag{9}$$

$$\mathbf{x}_i^{(l+1)} = \eta\mathbf{x}_i^{(0)} + (1-\eta)\mathbf{x}_i^{(l)} + \sum_{j\in\mathcal{N}(i)} (\mathbf{x}_i^{(l)} - \mathbf{x}_j^{(l)})\varphi^x(\mathbf{m}_{j\to i}), \forall i \in \mathcal{V}_1 \cup \mathcal{V}_2 \tag{10}$$

$$\mathbf{h}_i^{(l+1)} = LN\left((1-\beta)\cdot\mathbf{h}_i^{(l)} + \beta\cdot\varphi^h(\mathbf{h}_i^{(l)}, \mathbf{m}_i, \mu_i, \mathbf{f}_i)\right), \forall i \in \mathcal{V}_1 \cup \mathcal{V}_2, \tag{11}$$

where $\mathcal{N}(i)$ are the graph neighbors of node i, φ^x is a real multivariate (scalar) function, φ^h, φ^e are functions that output a vector $\mathbb{R}^d$, $\mathbf{f}_{j\to i}$ and $\mathbf{f}_i$ are original edge and node features (extracted SE(3)-invariantly from the residues), LN is Layer or Batch Normalization, and $a_{j\to i}$ is an attention based coefficient:

$$a_{j\to i} = \frac{\exp(\langle \mathbf{V}\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)}\rangle)}{\sum_{j'}\exp(\langle \mathbf{V}\mathbf{h}_i^{(l)}, \mathbf{h}_{j'}^{(l)}\rangle)}, \tag{12}$$

where $\mathbf{V}$ is a trainable weight matrix.

The output of several IEGMN layers is then denoted as:

$$\mathbf{Z}_1 \in \mathbb{R}^{3\times n}, \mathbf{H}_1 \in \mathbb{R}^{d\times n}, \mathbf{Z}_2 \in \mathbb{R}^{3\times m}, \mathbf{H}_2 \in \mathbb{R}^{d\times m} = IEGMN(\mathbf{X}_1, \mathbf{F}_1, \mathbf{X}_2, \mathbf{F}_2). \tag{13}$$

It is then straightforward to prove the following (see Appendix B.3):

Proposition 4. *IEGMNs satisfy the pairwise independent SE(3)-equivariance property in Eq. (4).*

Keypoints for Differentiable Protein Superimposition. We use multi-head attention to obtain K points for each protein, $\mathbf{Y}_1, \mathbf{Y}_2 \in \mathbb{R}^{3\times K}$, which we name **keypoints**. We desire that these are trained to become representative points for the binding pocket of the respective protein pair (encouraged by an additional loss described later). If this holds perfectly, then the superimposition of $\mathbf{Y}_1$ and $\mathbf{Y}_2$ will give the corresponding desired superimposition of $\mathbf{X}_1$ and $\mathbf{X}_2$

$$\mathbf{y}_{1k} := \sum_{i=1}^{n} \alpha_i^k \mathbf{z}_{1i}; \quad \mathbf{y}_{2k} := \sum_{j=1}^{m} \beta_j^k \mathbf{z}_{2j},$$

where $\mathbf{z}_{1i}$ denotes the i-th column of matrix $\mathbf{Z}_1$, and $\alpha_i^k = softmax_i(\frac{1}{\sqrt{d}}\mathbf{h}_{1i}^\top \mathbf{W}\mu(\varphi(\mathbf{H}_2)))$ are attention scores (similarly defined for β_j^k), with $\mathbf{W} \in \mathbb{R}^{d\times d}$ a parametric matrix, φ a linear layer plus a LeakyReLU non-linearity, and $\mu(\cdot)$ being the mean vector.

Differentiable Kabsch Model. We design the rotation and translation to map protein 1 to protein 2 to be the same transformation to superimpose $\mathbf{Y}_1$ and $\mathbf{Y}_2$. For this, we compute a differentiable version of the Kabsch algorithm. Let $\mathbf{A} = \overline{\mathbf{Y}}_2\overline{\mathbf{Y}}_1^\top \in \mathbb{R}^{3\times 3}$ computed using normalized keypoints. The corresponding singular value decomposition (SVD) decomposition is $\mathbf{A} = \mathbf{U}_2\mathbf{S}\mathbf{U}_1^\top$, where $\mathbf{U}_2, \mathbf{U}_1 \in O(3)$. Finally, we derive the differentiable functions

$$\begin{aligned} \mathbf{R}(\mathbf{X}_1|\mathbf{X}_2;\theta) &= \mathbf{U}_2 \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & d \end{pmatrix} \mathbf{U}_1^\top, \quad \text{where } d = \text{sign}(\det(\mathbf{U}_2\mathbf{U}_1^\top)) \\ \mathbf{t}(\mathbf{X}_1|\mathbf{X}_2;\theta) &= \mu(\mathbf{Y}_2) - \mathbf{R}(\mathbf{X}_1|\mathbf{X}_2;\theta)\mu(\mathbf{Y}_1), \end{aligned} \tag{14}$$

where $\mu(\cdot)$ is the mean vector of a point cloud. It is straightforward to prove that this model satisfies all the equivariance properties in Eqs. (1) to (3). From a practical perspective, the gradient and backpropagation through the SVD operation was analyzed by (Ionescu et al., 2015; Papadopoulo and Lourakis, 2000) and implemented in the automatic differentiation frameworks such as PyTorch.

MSE Loss. During training, we randomly decide which protein is the receptor (say protein 2), keep it in the docked position (i.e., $\mathbf{X}_2 = \mathbf{X}_2^*$), predict the SE(3) transformation using Eq. (14) and use it to compute the final position of the ligand as $\tilde{\mathbf{X}}_1 = \mathbf{R}(\mathbf{X}_1|\mathbf{X}_2)\mathbf{X}_1 + \mathbf{t}(\mathbf{X}_1|\mathbf{X}_2)$. The mean squared error (MSE) loss is then $\mathcal{L}_{MSE} = \frac{1}{n}\sum_{i=1}^{n} \|\mathbf{x}_i^* - \tilde{\mathbf{x}}_i\|^2$.

Optimal Transport and Binding Pocket Keypoint Alignment. As mentioned before, we desire that $\mathbf{Y}_1$ and $\mathbf{Y}_2$ are representative points for the binding pocket location of the respective protein pair. However, this needs to be encouraged explicitly, which we achieve using an additional loss.

We first define the binding pocket point sets, inspiring from previous PPI work (Section 2). Given the residues' α-carbon locations of the bound (docked) structures, $\mathbf{X}_1^*$ and $\mathbf{X}_2^*$, we select all pairs of residues at less than τ Euclidean distance ($\tau = 8$Å in our experiments). We assume these are interacting residues. Denote these pairs as $\{(\mathbf{x}_{1s}^*, \mathbf{x}_{2s}^*), s \in 1, \dots, S\}$, where S is a variable size across data pairs. We compute the midpoints of these segments denoted as $\mathbf{P}_1^*, \mathbf{P}_2^* \in \mathbb{R}^{3\times S}$, where $\mathbf{p}_{1s}^* = \mathbf{p}_{2s}^* = 0.5 \cdot (\mathbf{x}_{1s}^* + \mathbf{x}_{2s}^*)$. We view $\mathbf{P}_1^*$ and $\mathbf{P}_2^*$ as contact or binding pocket points. However, in the unbound state, these sets are randomly moved in space together with the respective protein residue coordinates $\mathbf{X}_1$ and $\mathbf{X}_2$. We denote them as $\mathbf{P}_1, \mathbf{P}_2 \in \mathbb{R}^{3\times S}$. For clarity, if $\mathbf{X}_1 = \mathbf{Q}\mathbf{X}_1^* + \mathbf{g}$, then $\mathbf{P}_1 = \mathbf{Q}\mathbf{P}_1^* + \mathbf{g}$.

We desire that $\mathbf{Y}_1$ is a representative set for the 3D set $\mathbf{P}_1$ (and, similarly, $\mathbf{Y}_2$ for $\mathbf{P}_2$). However, while at training time we know that every point $\mathbf{p}_{1s}$ corresponds to the point $\mathbf{p}_{2s}$ (and, similarly, $\mathbf{y}_{1k}$ aligns with $\mathbf{y}_{2k}$, by assumption), we unfortunately do not know the actual alignment between points in $\mathbf{Y}_l$ and $\mathbf{P}_l$, for every $l \in \{1, 2\}$. This can be recovered using an additional optimal transport loss:

$$\mathcal{L}_{\text{OT}} = \min_{\mathbf{T}\in\mathcal{U}(S,K)} \langle \mathbf{T}, \mathbf{C}\rangle, \quad \text{where } \mathbf{C}_{s,k} = \|\mathbf{y}_{1k} - \mathbf{p}_{1s}\|^2 + \|\mathbf{y}_{2k} - \mathbf{p}_{2s}\|^2, \tag{15}$$

where $\mathcal{U}(S, K)$ is the set of $S \times K$ transport plans with uniform marginals. The optimal transport plan is computed using an Earth Mover's Distance and the POT library (Flamary et al., 2021), while being kept fixed during back-propagation and optimization when only the cost matrix is trained.

Note that our approach assumes that $\mathbf{y}_{1k}$ corresponds to $\mathbf{y}_{2k}$, for every $k \in \{1, \dots, K\}$. Intuitively, each attention head k will identify a specific geometric/chemical local surface feature of protein 1 by $\mathbf{y}_{1k}$, and match its complementary feature of protein 2 by $\mathbf{y}_{2k}$.

Avoiding Point Cloud Intersection. In practice, our model does not enforce a useful inductive bias, namely that proteins forming complexes are not "intersecting" with each other. To properly define this problem, we first need a notion of the "interior" of a protein point cloud. Following previous work (Sverrisson et al., 2021; Venkatraman et al., 2009), we define the surface of a protein point cloud $\mathbf{X} \in \mathbb{R}^{3\times n}$ as $\{\mathbf{x} \in \mathbb{R}^3 : G(\mathbf{x}) = \gamma\}$, where $G(\mathbf{x}) = -\sigma \ln(\sum_{i=1}^{n} \exp(-||\mathbf{x} - \mathbf{x}_i||^2/\sigma))$. The parameters σ and γ are chosen such that there exist no "holes" inside a protein (we found $\gamma = 10, \sigma = 25$ to work well). As a consequence, the interior of the protein is given by $\{\mathbf{x} \in \mathbb{R}^3 : G(\mathbf{x}) < \gamma\}$. Then, the condition for non-intersecting ligand and receptor can be written as $G_1(\mathbf{x}_{2j}) > \gamma, \forall j \in 1, \dots, m$ and $G_2(\mathbf{x}_{1i}) > \gamma, \forall i \in 1, \dots, n$. As a loss function, this becomes

$$\mathcal{L}_{\text{NI}} = \frac{1}{n}\sum_{i=1}^{n} \max(0, \gamma - G_2(\mathbf{x}_{1i})) + \frac{1}{m}\sum_{j=1}^{m} \max(0, \gamma - G_1(\mathbf{x}_{2j})). \tag{16}$$

Surface Aware Node Features. We know that surface contact modeling is very important for protein docking. As a consequence, we incorporate a novel surface feature type that differentiates residues closer to the surface of the protein from those in the interior. We wish to do that without pre-computing meshes, to prioritize efficiency (similar to (Sverrisson et al., 2021)). Concretely, for each node i in the k-NN protein graph of the ligand, we pre-compute the norm of the weighted average of its graph neighbor forces, which can be interpreted as the normalized gradient of the $G(\mathbf{x})$ surface function defined above. This feature is SE(3)-invariant and is defined as

$$\rho_i(\mathbf{x}_i; \lambda) = \frac{\|\sum_{i'\in\mathcal{N}_i} w_{i,i',\lambda}(\mathbf{x}_i - \mathbf{x}_{i'})\|}{\sum_{i'\in\mathcal{N}_i} w_{i,i',\lambda}\|\mathbf{x}_i - \mathbf{x}_{i'}\|}, \quad \text{where } w_{i,i',\lambda} = \frac{\exp(-||\mathbf{x}_i - \mathbf{x}_{i'}||^2/\lambda)}{\sum_{j\in\mathcal{N}_i} \exp(-||\mathbf{x}_i - \mathbf{x}_j||^2/\lambda)}. \tag{17}$$

Intuitively, nodes in the interior of the protein have very low values of this feature since they are surrounded by vectors from all directions that cancel out, while nodes near the surface have

contributions from forces coming from a small cone (for convex local surfaces) or wide cone (for concave areas). Values for points near the surface are in $(0, 1)$ depending on the local curvature of the surface, with $\frac{2}{\pi}$ being the value for flat surface points with infinite uniformly spread neighbors.

Table 1: **Complex Prediction Results.** We note that the proprietary ATTRACT and HDOCK software might internally use parts of the test sets (as templates, features or in other forms), thus their numbers might be optimistic.

	DIPS Test Set						**DB5.5 Test Set**					
	Complex RMSD			Interface RMSD			Complex RMSD			Interface RMSD		
Methods	Mean	Median	Std	Mean	Median	Std	Mean	Median	Std	Mean	Median	Std
ATTRACT	14.93	17.17	10.39	14.02	12.41	11.81	10.09	9.55	9.88	10.69	7.48	10.90
HDOCK	10.77	6.23	11.39	8.88	3.90	10.95	5.34	0.30	12.04	4.76	0.24	10.83
IEGMN	17.90	17.66	6.80	13.27	13.29	6.26	15.35	13.98	4.65	13.67	13.03	4.85

Table 2: Inference time comparison.

	Runtime on DIPS Test Set					**Runtime on DB5.5 Test Set**				
Methods	Mean	Median	Min	Max	Std	Mean	Median	Min	Max	Std
ATTRACT	1285s	793s	62s	8192s	793s	570s	524s	180s	1708s	373s
HDOCK	778s	635s	145s	3177s	570s	615s	461s	210s	2593s	459s
IEGMN	5s	3s	1s	22s	5s	5s	3s	1s	53s	10s

5 EXPERIMENTS

Datasets. We use the Docking Benchmark 5.5 (DB5.5) dataset[1] (Vreven et al., 2015) and Database of Interacting Protein Structures (DIPS)[2] (Townshend et al., 2019). DB5.5 is a gold standard dataset in terms of data quality, but with only 253 examples. DIPS is a larger protein complex structures dataset mined from the Protein Data Bank (Berman et al., 2000). Dataset statistics are in Appendix C.

We filter DIPS to only keep proteins with a maximum atom count of 10,000. Datasets are then randomly partitioned in train/validation/test splits of sizes 203/25/25 (DB5.5) and 39,937/974/965 (DIPS), but for DIPS we the split is based on protein families to ensure that no two similar proteins are in the same split. For the final evaluation in Table 1, we use the full DB5.5 test set, and randomly sample 100 pairs from different protein families from the DIPS test set.

Evaluation Metrics. For measuring quality of the predicted complex structures, we report Complex Root Mean Square Deviation (CRMSD) and Interface Root Mean Square Deviation (IRMSD) defined below. Given the ground truth complex structure $\mathbf{Z}^* \in \mathbb{R}^{3\times(n+m)}$ and the predicted complex structure $\mathbf{Z} \in \mathbb{R}^{3\times(n+m)}$, we first superimpose them using the Kabsch algorithm (Kabsch, 1976), and then compute the RMSE loss as: CRMSD $= \sqrt{\frac{1}{n+m}\|\mathbf{Z}^* - \mathbf{Z}\|_F^2}$. IRMSD is computed similarly, but using only the coordinates of the interface residues with distance less than 8Å to the other protein.

Training Details. We train our models on the train part of DIPS first, using Adam (Kingma and Ba, 2014) with learning rate 1e-4 and early stopping with patience of 20 epochs. We update the best validation model only when it achieves a score of less than 98% of the previous best validation score, where the score is the median of Ligand RMSD on the full DIPS validation set. The best DIPS validated model is then tested on the DIPS test set. For DB5.5, we fine tune DIPS pre-trained models on the DB5.5 training set and use early stopping with patience of 50 epochs. The best DB5.5 validated model is finally tested on DB5.5 test. During training, we randomly assign the roles of

[1]DB5.5 is downloaded from https://zlab.umassmed.edu/benchmark/
[2]DIPS is downloaded from https://github.com/drorlab/DIPS

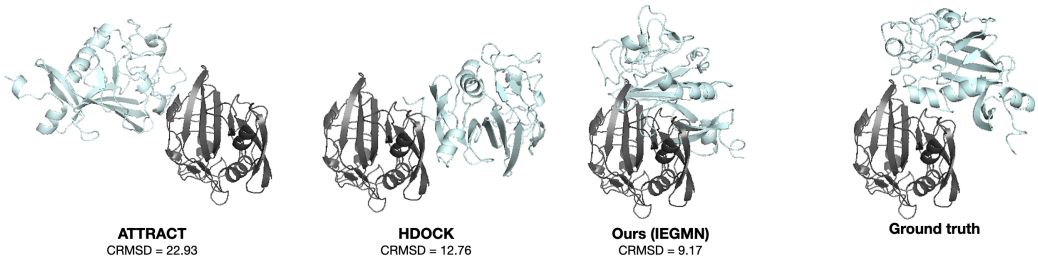

Figure 4: Visualization of a protein complex (2j3g) successfully predicted by IEGMN.

ligand and receptor. Also, during both training and testing, we randomly rotate and translate the ligand in space (even though our model is invariant to this operation) for all baselines.

Baselines. We compare our IEGMN method with the docking software ATTRACT[3] (Schindler et al., 2017) and HDOCK[4] (Yan et al., 2020; 2017b;a; Huang and Zou, 2014; 2008). We note that most of the other docking methods provide only web interfaces for manual job submission, while ATTRACT and HDOCK provide user-friendly local packages suitable for automatic experiments.

On testing sets, ATTRACT fails for '1N2C' in DB5.5, 'oi_4oip.pdb1_8', 'oi_4oip.pdb1_3' and 'p7_4p7s.pdb1_2' in DIPS. For such failure cases, we use the unbound input structure as the prediction for metrics calculation.

Complex Prediction Results. Results are shown in Table 1. We note that our method is competitive, but does not always outperform the baselines. Possible explanations include the facts that we do not use heavy candidate sampling and re-ranking, we do not rely on task-specific hand-crafted features and we currently do not perform structure fine-tuning, aiming to predict the SE(3) ligand transformation in a direct shot. Moreover, we note that the baselines might have used part of our test set in their models, for example to learn surface templates, thus, their reported scores might be optimistic. Notably, HDOCK score function was validated on DB4 which overlaps with DB5.5. A more approprite comparison would require us to re-build these baselines by explicitly removing any information from our test sets, a task that is not possible given the absence of open-source implementations.

Computational efficiency results. We show inference times in Table 2. We note that our IEGMN method is between 80-500 times faster than the baselines. Hardware specifications are as follows: ATTRACT was run on a 6-Core Intel Core i7 2.2 GHz CPU; HDOCK was run on a single Intel Xeon Gold 6230 2.1 GHz CPU; IEGMN was run on a single Intel Core i9-9880H 2.3 GHz CPU.

Visualization. We show in Fig. 4 one successful example of a test DIPS protein pair for which our model outperforms all baselines.

6 Conclusion

We have presented a fast, end-to-end rigid protein docking approach that does not rely on candidate sampling, templates task-specific features or pre-computed meshes. Our method smartly incorporates useful rigid protein docking priors including commutativity and SE(3)-invariance, thus avoiding the computational burden of data augmentation. We look forward to extend our model for flexible docking, as well as integrating it into rigid body interaction in physical simulations.

[3]We use the VirtualBox: `www.attract.ph.tum.de/services/ATTRACT/ATTRACT.vdi.gz`
[4]HDOCK is downloaded from `http://huanglab.phys.hust.edu.cn/software/HDOCK/`

ETHICS STATEMENT

The protein-protein docking problem we target is very important for understanding cell mechanism, and the resulted protein complexes are essential for virtually all cellular processes. Such technologies are very powerful and have the potential to drastically change how people approach impactful areas such as drug discovery, but could also potentially be mistakenly misused. However, we note that our model, as well as existing protein docking software, have yet to achieve a more realistic qualitative level and proved to be more interpretable and reliable when transferred to different types of proteins.

REPRODUCIBILITY STATEMENT

In order to ensure reproducibility, we have made the efforts in the following respects: 1) Provide self-contained proofs of the main claims in Appendix B; 2) Provide details on experimental configurations in Appendix D. 3) We will fully release our training and evaluation code, as well as our train/validation/test splits, after the reviewing process.

Moreover, all the datasets are publicly available as described in the main text.

REFERENCES

B. Adhikari and J. Cheng. Confold2: Improved contact-driven ab initio protein structure modeling. *BMC bioinformatics*, 19(1):1–5, 2018. 2

M. Baek, F. DiMaio, I. Anishchenko, J. Dauparas, S. Ovchinnikov, G. R. Lee, J. Wang, Q. Cong, L. N. Kinch, R. D. Schaeffer, et al. Accurate prediction of protein structures and interactions using a three-track neural network. *Science*, 373(6557):871–876, 2021. 1, 2

F. Baldassarre, D. Menéndez Hurtado, A. Elofsson, and H. Azizpour. Graphqa: protein model quality assessment using graph convolutional networks. *Bioinformatics*, 37(3):360–366, 2021. 3

J. Bao, X. He, and J. Z. Zhang. Deepbsp—a machine learning method for accurate prediction of protein–ligand docking structures. *Journal of Chemical Information and Modeling*, 2021. 3

S. Basu and B. Wallner. Dockq: a quality measure for protein-protein docking models. *PloS one*, 11 (8):e0161879, 2016. 3

H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne. The protein data bank. *Nucleic acids research*, 28(1):235–242, 2000. 8

J. Biesiada, A. Porollo, P. Velayutham, M. Kouril, and J. Meller. Survey of public domain software for docking simulations and virtual screening. *Human genomics*, 5(5):1–9, 2011. 3

J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*, 2013. 2

P. Bryant, G. Pozzati, and A. Elofsson. Improved prediction of protein-protein interactions using alphafold2 and extended multiple-sequence alignments. *bioRxiv*, 2021. 3

R. Chen, L. Li, and Z. Weng. Zdock: an initial-stage protein-docking algorithm. *Proteins: Structure, Function, and Bioinformatics*, 52(1):80–87, 2003. 1, 3

C. Christoffer, S. Chen, V. Bharadwaj, T. Aderinwale, V. Kumar, M. Hormati, and D. Kihara. Lzerd webserver for pairwise and multiple protein–protein docking. *Nucleic Acids Research*, 2021. 3

T. Cohen and M. Welling. Group equivariant convolutional networks. In *International conference on machine learning*, pages 2990–2999. PMLR, 2016a. 2

T. S. Cohen and M. Welling. Steerable cnns. *arXiv preprint arXiv:1612.08498*, 2016b. 2

B. Dai and C. Bailey-Kellogg. Protein interaction interface region prediction by geometric deep learning. *Bioinformatics*, 2021. 3

S. J. De Vries, M. Van Dijk, and A. M. Bonvin. The haddock web server for data-driven biomolecular docking. *Nature protocols*, 5(5):883–897, 2010. 1, 3

M. Defferrard, X. Bresson, and P. Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. *arXiv preprint arXiv:1606.09375*, 2016. 2

S. Eismann, R. J. Townshend, N. Thomas, M. Jagota, B. Jing, and R. O. Dror. Hierarchical, rotation-equivariant neural networks to select structural models of protein complexes. *Proteins: Structure, Function, and Bioinformatics*, 2020. 2, 3

R. Evans, M. O'Neill, A. Pritzel, N. Antropova, A. W. Senior, T. Green, A. Žídek, R. Bates, S. Blackwell, J. Yim, O. Ronneberger, S. Bodenstein, M. Zielinski, A. Bridgland, A. Potapenko, A. Cowie, K. Tunyasuvunakool, R. Jain, E. Clancy, P. Kohli, J. Jumper, and D. Hassabis. Protein complex prediction with alphafold-multimer. *bioRxiv*, 2021. doi: 10.1101/2021.10.04.463034. 3

M. Finzi, S. Stanton, P. Izmailov, and A. G. Wilson. Generalizing convolutional neural networks for equivariance to lie groups on arbitrary continuous data. In *International Conference on Machine Learning*, pages 3165–3176. PMLR, 2020. 2

R. Flamary, N. Courty, A. Gramfort, M. Z. Alaya, A. Boisbunon, S. Chambon, L. Chapel, A. Corenflos, K. Fatras, N. Fournier, L. Gautheron, N. T. Gayraud, H. Janati, A. Rakotomamonjy, I. Redko, A. Rolet, A. Schutz, V. Seguy, D. J. Sutherland, R. Tavenard, A. Tong, and T. Vayer. Pot: Python optimal transport. *Journal of Machine Learning Research*, 22(78):1–8, 2021. URL http://jmlr.org/papers/v22/20-451.html. 7

A. Fout, J. Byrd, B. Shariat, and A. Ben-Hur. Protein interface prediction using graph convolutional networks. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6533–6542, 2017. 3, 5

F. B. Fuchs, D. E. Worrall, V. Fischer, and M. Welling. Se (3)-transformers: 3d roto-translation equivariant attention networks. *arXiv preprint arXiv:2006.10503*, 2020. 2

U. Ghani, I. Desta, A. Jindal, O. Khan, G. Jones, S. Kotelnikov, D. Padhorny, S. Vajda, and D. Kozakov. Improved docking of protein models by a combination of alphafold2 and cluspro. *bioRxiv*, 2021. 3

J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl. Neural message passing for quantum chemistry. In *International Conference on Machine Learning*, pages 1263–1272. PMLR, 2017. 2

S.-Y. Huang and X. Zou. An iterative knowledge-based scoring function for protein–protein recognition. *Proteins: Structure, Function, and Bioinformatics*, 72(2):557–579, 2008. 9

S.-Y. Huang and X. Zou. A knowledge-based scoring function for protein-rna interactions derived from a statistical mechanics-based iterative method. *Nucleic acids research*, 42(7):e55–e55, 2014. 9

I. R. Humphreys, J. Pei, M. Baek, A. Krishnakumar, I. Anishchenko, S. R. Ovchinnikov, J. Zheng, T. Ness, S. Banjade, S. R. Bagde, V. Stancheva, X. Li, K. Liu, Z. Zheng, D. Barerro, U. Roy, I. S. Fernandez, B. Szakal, D. Branzei, E. C. Greene, S. Biggins, S. Keeney, E. A. Miller, J. C. Fromme, T. Hendrickson, Q. Cong, and D. Baker. Structures of core eukaryotic protein complexes. 2021. doi: 10.1101/2021.09.30.462231. 3

M. Hutchinson, C. L. Lan, S. Zaidi, E. Dupont, Y. W. Teh, and H. Kim. Lietransformer: Equivariant self-attention for lie groups. *arXiv Preprint*, 2012.10885, 2020. 2

J. Ingraham, V. K. Garg, R. Barzilay, and T. Jaakkola. Generative models for graph-based protein design. 2019. 15

C. Ionescu, O. Vantzos, and C. Sminchisescu. Matrix backpropagation for deep networks with structured layers. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2965–2973, 2015. 6

L. Jovine. Using machine learning to study protein–protein interactions: From the uromodulin polymer to egg zona pellucida filaments. *Molecular Reproduction and Development*, n/a(n/a). doi: https://doi.org/10.1002/mrd.23538. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/mrd.23538. 3

F. Ju, J. Zhu, B. Shao, L. Kong, T.-Y. Liu, W.-M. Zheng, and D. Bu. Copulanet: Learning residue co-evolution directly from multiple sequence alignment for protein structure prediction. *Nature Communications*, 12(1):2535, May 2021. 2

J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021. 1, 2, 15

W. Kabsch. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography*, 32(5):922–923, 1976. 4, 8

A. Khramushin, T. Tsaban, J. K. Varga, O. Avraham, and O. Schueler-Furman. Patchman docking: Modeling peptide-protein interactions in the context of the receptor surface. *bioRxiv*, 2021. 3

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 8

T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016. 2

J. Ko and J. Lee. Can alphafold2 predict protein-peptide complex structures accurately? *bioRxiv*, 2021. 3

D. R. Koes, M. P. Baumgartner, and C. J. Camacho. Lessons learned in empirical scoring with smina from the csar 2011 benchmarking exercise. *Journal of chemical information and modeling*, 53(8): 1893–1904, 2013. 3

D. Kozakov, D. R. Hall, B. Xia, K. A. Porter, D. Padhorny, C. Yueh, D. Beglov, and S. Vajda. The cluspro web server for protein–protein docking. *Nature protocols*, 12(2):255–278, 2017. 3

G. Launay, M. Ohue, J. Prieto Santero, Y. Matsuzaki, C. Hilpert, N. Uchikoga, T. Hayashi, and J. Martin. Evaluation of consrank-like scoring functions for rescoring ensembles of protein–protein docking poses. *Frontiers in molecular biosciences*, 7:308, 2020. 3

S. Li, J. Zhou, T. Xu, L. Huang, F. Wang, H. Xiong, W. Huang, D. Dou, and H. Xiong. Structure-aware interactive graph neural networks for the prediction of protein-ligand binding affinity. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 975–985, 2021. 3

Y. Li, C. Gu, T. Dullien, O. Vinyals, and P. Kohli. Graph matching networks for learning the similarity of graph structured objects. In *International Conference on Machine Learning*, pages 3835–3845. PMLR, 2019. 2, 5

Y. Liu, H. Yuan, L. Cai, and S. Ji. Deep learning of high-order interactions for protein interface prediction. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 679–687, 2020. 3, 5

A. T. McNutt, P. Francoeur, R. Aggarwal, T. Masuda, R. Meli, M. Ragoza, J. Sunseri, and D. R. Koes. Gnina 1.0: molecular docking with deep learning. *Journal of cheminformatics*, 13(1):1–20, 2021. 3

I. H. Moal, M. Torchala, P. A. Bates, and J. Fernández-Recio. The scoring of poses in protein-protein docking: current capabilities and future directions. *BMC bioinformatics*, 14(1):1–15, 2013. 3

T. Papadopoulo and M. I. Lourakis. Estimating the jacobian of the singular value decomposition: Theory and applications. In *European Conference on Computer Vision*, pages 554–570. Springer, 2000. 6

J. Pei, J. Zhang, and Q. Cong. Human mitochondrial protein complexes revealed by large-scale coevolution analysis and deep learning-based structure modeling. *bioRxiv*, 2021. 3

D. J. Rezende, S. Racanière, I. Higgins, and P. Toth. Equivariant hamiltonian flows. *arXiv preprint arXiv:1909.13739*, 2019. 2

D. W. Romero and J.-B. Cordonnier. Group equivariant stand-alone self-attention for vision. *arXiv preprint arXiv:2010.00977*, 2020. 2

S. Sanyal, I. Anishchenko, A. Dagar, D. Baker, and P. Talukdar. Proteingcn: Protein model quality assessment using graph convolutional networks. *bioRxiv*, 2020. 3

V. G. Satorras, E. Hoogeboom, and M. Welling. E(n)-equivariant graph neural networks. *arXiv preprint arXiv:2102.09844*, 2021. 2, 5

C. E. Schindler, I. Chauvot de Beauchêne, S. J. de Vries, and M. Zacharias. Protein-protein and peptide-protein docking and refinement using attract in capri. *Proteins: Structure, Function, and Bioinformatics*, 85(3):391–398, 2017. 1, 3, 9

A. W. Senior, R. Evans, J. Jumper, J. Kirkpatrick, L. Sifre, T. Green, C. Qin, A. Žídek, A. W. Nelson, A. Bridgland, et al. Improved protein structure prediction using potentials from deep learning. *Nature*, 577(7792):706–710, 2020. 1, 2

S. Sunny and P. Jayaraj. Fpdock: Protein–protein docking using flower pollination algorithm. *Computational Biology and Chemistry*, 93:107518, 2021. 1, 3

F. Sverrisson, J. Feydy, B. E. Correia, and M. M. Bronstein. Fast end-to-end learning on protein surfaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15272–15281, 2021. 7

N. Thomas, T. Smidt, S. Kearnes, L. Yang, L. Li, K. Kohlhoff, and P. Riley. Tensor field networks: Rotation-and translation-equivariant neural networks for 3d point clouds. *arXiv preprint arXiv:1802.08219*, 2018. 2

M. Torchala, I. H. Moal, R. A. Chaleil, J. Fernandez-Recio, and P. A. Bates. Swarmdock: a server for flexible protein–protein docking. *Bioinformatics*, 29(6):807–809, 2013. 1, 3

R. Townshend, R. Bedi, P. Suriana, and R. Dror. End-to-end learning on 3d protein structure for interface prediction. *Advances in Neural Information Processing Systems*, 32:15642–15651, 2019. 3, 5, 8

O. Trott and A. J. Olson. Autodock vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *Journal of computational chemistry*, 31(2):455–461, 2010. 3

T. Tsaban, J. Varga, O. Avraham, Z. Ben-Aharon, A. Khramushin, and O. Schueler-Furman. Harnessing protein folding neural networks for peptide-protein docking. 2021. 3

I. A. Vakser. Protein-protein docking: From interaction to interactome. *Biophysical journal*, 107(8): 1785–1793, 2014. 3

V. Venkatraman, Y. D. Yang, L. Sael, and D. Kihara. Protein-protein docking using region-based 3d zernike descriptors. *BMC bioinformatics*, 10(1):1–21, 2009. 1, 3, 7

J. Verburgt and D. Kihara. Benchmarking of structure refinement methods for protein complex models. *Proteins: Structure, Function, and Bioinformatics*, 2021. 3

T. Vreven, I. H. Moal, A. Vangone, B. G. Pierce, P. L. Kastritis, M. Torchala, R. Chaleil, B. Jiménez-García, P. A. Bates, J. Fernandez-Recio, et al. Updates to the integrated protein–protein interaction benchmarks: docking benchmark version 5 and affinity benchmark version 2. *Journal of molecular biology*, 427(19):3031–3041, 2015. 8

I. Wallach, M. Dzamba, and A. Heifets. Atomnet: a deep convolutional neural network for bioactivity prediction in structure-based drug discovery. *arXiv preprint arXiv:1510.02855*, 2015. 3

M. Weiler and G. Cesa. General $e(2)$-equivariant steerable cnns. *arXiv preprint arXiv:1911.08251*, 2019. 2

G. Weng, E. Wang, Z. Wang, H. Liu, F. Zhu, D. Li, and T. Hou. Hawkdock: a web server to predict and analyze the protein–protein complex based on computational docking and mm/gbsa. *Nucleic acids research*, 47(W1):W322–W330, 2019. 3

J. Wu, T. Shen, H. Lan, Y. Bian, and J. Huang. Se (3)-equivariant energy-based models for end-to-end protein folding. *bioRxiv*, 2021. 2

Z. Xie and J. Xu. Deep graph learning of inter-protein contacts. *bioRxiv*, 2021. 3

Y. Yan, Z. Wen, X. Wang, and S.-Y. Huang. Addressing recent docking challenges: A hybrid strategy to integrate template-based and free protein-protein docking. *Proteins: Structure, Function, and Bioinformatics*, 85(3):497–512, 2017a. 9

Y. Yan, D. Zhang, P. Zhou, B. Li, and S.-Y. Huang. Hdock: a web server for protein–protein and protein–dna/rna docking based on a hybrid strategy. *Nucleic acids research*, 45(W1):W365–W373, 2017b. 9

Y. Yan, H. Tao, J. He, and S.-Y. Huang. The hdock server for integrated protein–protein docking. *Nature protocols*, 15(5):1829–1852, 2020. 3, 9

J. Yang, I. Anishchenko, H. Park, Z. Peng, S. Ovchinnikov, and D. Baker. Improved protein structure prediction using predicted interresidue orientations. *Proceedings of the National Academy of Sciences*, 117(3):1496–1503, 2020. 2

Appendix

CONTENTS

A Representing Proteins as Graphs **15**

B Proofs of the main propositions **16**

B.1 Proof of Proposition 1. 16

B.2 Proof of Proposition 3. 17

B.3 Proof of Proposition 4. 17

C Datasets **17**

D Hyperparameters **17**

A REPRESENTING PROTEINS AS GRAPHS

A protein is comprised of amino acid residues. The structure of an amino acid residue is shown in Figure Fig. 5. Generally, an amino acid residue contains amino (-NH-), α-carbon atom and carboxyl (-CO-), along with a side chain (R) connected with the α-carbon atom. The side chain (R) is specific to each type of amino acid residues.

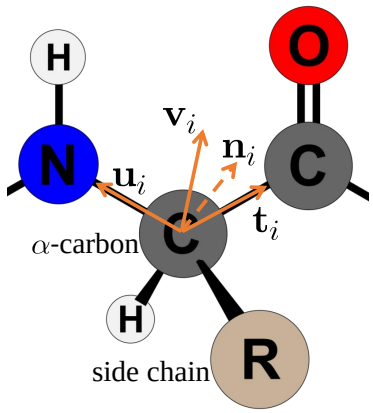

Figure 5: Representation of an amino acid residue.

We work on residue level (our approaches can be extended to atom level as well). A protein is represented by a set of nodes where each node is an amino acid residue in the protein. Each node i has a 3D coordinate $\mathbf{x}_i \in \mathbb{R}^3$ which is the 3D coordinate of α-carbon atom of the residue.

The neighborhood of a node is the set of k ($k = 10$ in our experiments) nearest nodes where the distance is the Euclidean distance between 3D coordinates.

Node feature is a one dimension indicator (one-hot encoding) of the type of amino acid residue. This one dimension indicator will be passed into an embedding layer.

Local Coordinate System. Similar to Ingraham et al. (2019) and Jumper et al. (2021), we introduce a local coordinate system for each residue which denotes the orientation of a residue. Based on this, we can further design SE(3)-invariant edge features. As shown in Figure 5, for a residue i, we

denote the unit vector pointing from α-carbon atom to nitrogen atom as $\mathbf{u}_i$. We denote the unit vector pointing from α-carbon atom to carbon atom of the carboxyl (-CO-) as $\mathbf{t}_i$. $\mathbf{u}_i$ and $\mathbf{t}_i$ together define a plane, and the normal of this plane is $\mathbf{n}_i = \frac{\mathbf{u}_i \times \mathbf{t}_i}{\|\mathbf{u}_i \times \mathbf{t}_i\|}$. Finally, we define $\mathbf{v}_i = \mathbf{n}_i \times \mathbf{u}_i$. Then $\mathbf{n}_i$, $\mathbf{u}_i$ and $\mathbf{v}_i$ together form the basis of residue i's local coordinate system. They denote the orientation of residue i.

Then we introduce the edge features of an edge $j \to i \in \mathcal{E}$. These features describe the *relative position* of j with respect to i, the *relative orientation* of j with respect to i and the *distance* between j and i.

Relative Position Edge Features First we introduce the edge features $\mathbf{p}_{j\to i}$ which describe relative position of j with respect to i:

$$\mathbf{p}_{j\to i} = \begin{bmatrix} \mathbf{n}_i^\top \\ \mathbf{u}_i^\top \\ \mathbf{v}_i^\top \end{bmatrix} [\mathbf{x}_j - \mathbf{x}_i]$$

Relative Orientation Edge Features As we mention above, each residue has orientation which carries information. Here we introduce the edge features $\mathbf{q}_{j\to i}$, $\mathbf{k}_{j\to i}$ and $\mathbf{t}_{j\to i}$ which describe relative orientation of j with respect to i:

$$\mathbf{q}_{j\to i} = \begin{bmatrix} \mathbf{n}_i^\top \\ \mathbf{u}_i^\top \\ \mathbf{v}_i^\top \end{bmatrix} [\mathbf{n}_j], \quad \mathbf{k}_{j\to i} = \begin{bmatrix} \mathbf{n}_i^\top \\ \mathbf{u}_i^\top \\ \mathbf{v}_i^\top \end{bmatrix} [\mathbf{u}_j], \quad \mathbf{t}_{j\to i} = \begin{bmatrix} \mathbf{n}_i^\top \\ \mathbf{u}_i^\top \\ \mathbf{v}_i^\top \end{bmatrix} [\mathbf{v}_j]$$

Distance-Based Edge Features Distance also carries information. Here we use radial basis function of distance as edge features:

$$\mathbf{f}_{j\to i,r} = e^{-\frac{(\|\mathbf{x}_j - \mathbf{x}_i\|)^2}{2\sigma_r^2}}, r = 1, 2, ..., R$$

Where R and scale parameters $\{\sigma_r\}_{1\le r\le R}$ are hyperparameters. In experiments, the set of scale parameters we used is $\{1.5^x | x = 0, 1, 2, ..., 14\}$. So for each edge, there are 15 distance-based edge features.

Surface Aware Node Features We additionally compute 5 surface aware node features defined in Eq. (17) using $\lambda \in \{1., 2., 5., 10., 30.\}$.

B PROOFS OF THE MAIN PROPOSITIONS

B.1 PROOF OF PROPOSITION 1.

Proof. Denote the predicted ligand position by $\mathbf{R}(\mathbf{X}_1|\mathbf{X}_2)\mathbf{X}_1 + \mathbf{t}(\mathbf{X}_1|\mathbf{X}_2) = \tilde{\mathbf{X}}_1$. Then the transformation to dock $\mathbf{Q}_1\mathbf{X}_1 + \mathbf{g}_1$ with respect to $\mathbf{Q}_2\mathbf{X}_2 + \mathbf{g}_2$ is the same as the transformation to change $\mathbf{Q}_1\mathbf{X}_1 + \mathbf{g}_1$ into $\mathbf{Q}_2\tilde{\mathbf{X}}_1 + \mathbf{g}_2$. We use the notation: $\mathbf{R}^\top(\mathbf{X}_1|\mathbf{X}_2) = (\mathbf{R}(\mathbf{X}_1|\mathbf{X}_2))^\top$. Then, we have the following derivation steps:

$$\mathbf{R}(\mathbf{X}_1|\mathbf{X}_2)\mathbf{X}_1 + \mathbf{t}(\mathbf{X}_1|\mathbf{X}_2) = \tilde{\mathbf{X}}_1$$
$$\mathbf{X}_1 + \mathbf{R}^\top(\mathbf{X}_1|\mathbf{X}_2)\mathbf{t}(\mathbf{X}_1|\mathbf{X}_2) = \mathbf{R}^\top(\mathbf{X}_1|\mathbf{X}_2)\tilde{\mathbf{X}}_1$$
$$\mathbf{X}_1 + \mathbf{R}^\top(\mathbf{X}_1|\mathbf{X}_2)\mathbf{t}(\mathbf{X}_1|\mathbf{X}_2) = \mathbf{R}^\top(\mathbf{X}_1|\mathbf{X}_2)\mathbf{Q}_2^\top(\mathbf{Q}_2\tilde{\mathbf{X}}_1 + \mathbf{g}_2 - \mathbf{g}_2)$$
$$\mathbf{X}_1 + \mathbf{R}^\top(\mathbf{X}_1|\mathbf{X}_2)\mathbf{t}(\mathbf{X}_1|\mathbf{X}_2) = \mathbf{R}^\top(\mathbf{X}_1|\mathbf{X}_2)\mathbf{Q}_2^\top(\mathbf{Q}_2\tilde{\mathbf{X}}_1 + \mathbf{g}_2) - \mathbf{R}^\top(\mathbf{X}_1|\mathbf{X}_2)\mathbf{Q}_2^\top\mathbf{g}_2$$
$$\mathbf{X}_1 + \mathbf{R}^\top(\mathbf{X}_1|\mathbf{X}_2)\mathbf{t}(\mathbf{X}_1|\mathbf{X}_2) + \mathbf{R}^\top(\mathbf{X}_1|\mathbf{X}_2)\mathbf{Q}_2^\top\mathbf{g}_2 = \mathbf{R}^\top(\mathbf{X}_1|\mathbf{X}_2)\mathbf{Q}_2^\top(\mathbf{Q}_2\tilde{\mathbf{X}}_1 + \mathbf{g}_2)$$
$$\mathbf{Q}_1^\top(\mathbf{Q}_1\mathbf{X}_1 + \mathbf{g}_1 - \mathbf{g}_1) + \mathbf{R}^\top(\mathbf{X}_1|\mathbf{X}_2)(\mathbf{t}(\mathbf{X}_1|\mathbf{X}_2) + \mathbf{Q}_2^\top\mathbf{g}_2) = \mathbf{R}^\top(\mathbf{X}_1|\mathbf{X}_2)\mathbf{Q}_2^\top(\mathbf{Q}_2\tilde{\mathbf{X}}_1 + \mathbf{g}_2)$$
$$\mathbf{Q}_1^\top(\mathbf{Q}_1\mathbf{X}_1 + \mathbf{g}_1) - \mathbf{Q}_1^\top\mathbf{g}_1 + \mathbf{R}^\top(\mathbf{X}_1|\mathbf{X}_2)(\mathbf{t}(\mathbf{X}_1|\mathbf{X}_2) + \mathbf{Q}_2^\top\mathbf{g}_2) = \mathbf{R}^\top(\mathbf{X}_1|\mathbf{X}_2)\mathbf{Q}_2^\top(\mathbf{Q}_2\tilde{\mathbf{X}}_1 + \mathbf{g}_2)$$
$$\mathbf{R}(\mathbf{X}_1|\mathbf{X}_2)\mathbf{Q}_1^\top(\mathbf{Q}_1\mathbf{X}_1 + \mathbf{g}_1) - \mathbf{R}(\mathbf{X}_1|\mathbf{X}_2)\mathbf{Q}_1^\top\mathbf{g}_1 + \mathbf{t}(\mathbf{X}_1|\mathbf{X}_2) + \mathbf{Q}_2^\top\mathbf{g}_2 = \mathbf{Q}_2^\top(\mathbf{Q}_2\tilde{\mathbf{X}}_1 + \mathbf{g}_2)$$
$$\mathbf{Q}_2\mathbf{R}(\mathbf{X}_1|\mathbf{X}_2)\mathbf{Q}_1^\top(\mathbf{Q}_1\mathbf{X}_1 + \mathbf{g}_1) - \mathbf{Q}_2\mathbf{R}(\mathbf{X}_1|\mathbf{X}_2)\mathbf{Q}_1^\top\mathbf{g}_1 + \mathbf{Q}_2\mathbf{t}(\mathbf{X}_1|\mathbf{X}_2) + \mathbf{g}_2 = \mathbf{Q}_2\tilde{\mathbf{X}}_1 + \mathbf{g}_2$$

From the last equation above, one derives the transformation of $\mathbf{Q}_1\mathbf{X}_1 + \mathbf{g}_1$ into $\mathbf{Q}_2\tilde{\mathbf{X}}_1 + \mathbf{g}_2$, which is, by definition of the functions $\mathbf{R}$ and $\mathbf{t}$, the same as the transformation to dock $\mathbf{Q}_1\mathbf{X}_1 + \mathbf{g}_1$ with respect to $\mathbf{Q}_2\mathbf{X}_2 + \mathbf{g}_2$. This transformation is

$$\mathbf{R}(\mathbf{Q}_1\mathbf{X}_1 + \mathbf{g}_1|\mathbf{Q}_2\mathbf{X}_2 + \mathbf{g}_2) = \mathbf{Q}_2\mathbf{R}(\mathbf{X}_1|\mathbf{X}_2)\mathbf{Q}_1^\top$$
$$\mathbf{t}(\mathbf{Q}_1\mathbf{X}_1 + \mathbf{g}_1|\mathbf{Q}_2\mathbf{X}_2 + \mathbf{g}_2) = \mathbf{Q}_2\mathbf{t}(\mathbf{X}_1|\mathbf{X}_2) - \mathbf{Q}_2\mathbf{R}(\mathbf{X}_1|\mathbf{X}_2)\mathbf{Q}_1^\top\mathbf{g}_1 + \mathbf{g}_2$$

which concludes the proof. □

B.2 Proof of Proposition 3.

Proof. As in Appendix B.1, we denote $\mathbf{R}(\mathbf{X}_1|\mathbf{X}_2)\mathbf{X}_1 + \mathbf{t}(\mathbf{X}_1|\mathbf{X}_2) = \tilde{\mathbf{X}}_1$. Then the transformation to dock $\mathbf{X}_2$ with respect to $\mathbf{X}_1$ is the same as the transformation to change $\tilde{\mathbf{X}}_1$ back to $\mathbf{X}_1$, which is

$$\mathbf{R}(\mathbf{X}_1|\mathbf{X}_2)\mathbf{X}_1 + \mathbf{t}(\mathbf{X}_1|\mathbf{X}_2) = \tilde{\mathbf{X}}_1$$
$$\mathbf{X}_1 + \mathbf{R}^\top(\mathbf{X}_1|\mathbf{X}_2)\mathbf{t}(\mathbf{X}_1|\mathbf{X}_2) = \mathbf{R}^\top(\mathbf{X}_1|\mathbf{X}_2)\tilde{\mathbf{X}}_1$$
$$\mathbf{X}_1 = \mathbf{R}^\top(\mathbf{X}_1|\mathbf{X}_2)\tilde{\mathbf{X}}_1 - \mathbf{R}^\top(\mathbf{X}_1|\mathbf{X}_2)\mathbf{t}(\mathbf{X}_1|\mathbf{X}_2)$$

From the last equation above, we derive the transformation to change $\tilde{\mathbf{X}}_1$ back to $\mathbf{X}_1$, which is the same as the transformation to dock $\mathbf{X}_2$ with respect to $\mathbf{X}_1$. □

B.3 Proof of Proposition 4.

Proof. Let $\mathbf{X}_1^{(l+1)}, \mathbf{H}_1^{(l+1)}, \mathbf{X}_2^{(l+1)}, \mathbf{H}_2^{(l+1)} = \text{IEGMN}(\mathbf{X}_1^{(l)}, \mathbf{H}_1^{(l)}, \mathbf{X}_2^{(l)}, \mathbf{H}_2^{(l)})$ be the output of an IEGMN layer. Then, for any matrices $\mathbf{Q}_1, \mathbf{Q}_2 \in SO(3)$ and any translation vectors $\mathbf{g}_1, \mathbf{g}_2 \in \mathbb{R}^3$, we want to prove that IEGMN satisfy the pairwise independent SE(3)-equivariance property:

$$\mathbf{Q}_1\mathbf{X}_1^{(l+1)} + \mathbf{g}_1, \mathbf{H}_1^{(l+1)}, \mathbf{Q}_2\mathbf{X}_2^{(l+1)} + \mathbf{g}_2, \mathbf{H}_2^{(l+1)} = \text{IEGMN}(\mathbf{Q}_1\mathbf{X}_1^{(l)} + \mathbf{g}_1, \mathbf{H}_1^{(l)}, \mathbf{Q}_2\mathbf{X}_2^{(l)} + \mathbf{g}_2, \mathbf{H}_2^{(l)})$$

where each column of $\mathbf{X}_1^{(l)} \in \mathbb{R}^{3\times n}$,$\mathbf{H}_1^{(l)} \in \mathbb{R}^{d\times n}$,$\mathbf{X}_2^{(l)} \in \mathbb{R}^{3\times m}$ and $\mathbf{H}_2^{(l)} \in \mathbb{R}^{d\times m}$ represent an individual node's coordinate embedding or feature embedding.

We first note that the equations of our proposed IEGMN layer that compute messages $\mathbf{m}_{j\to i}$, $\mu_{j\to i}$, $\mathbf{m}_i$ and μ_i are SE(3)-invariant. Indeed, they depend on the initial features which are SE(3)-invariant by design, the current latent node embeddings $\{\mathbf{h}_i^{(l)}\}_{i\in\mathcal{V}_1\cup\mathcal{V}_2}$, as well as the Euclidean distances on the current node coordinates $\{\mathbf{x}_i^{(l)}\}_{i\in\mathcal{V}_1\cup\mathcal{V}_2}$. Thus, we also derive that the equation that computes the new latent node embeddings $\mathbf{h}_i^{(l+1)}$ is SE(3)-invariant. Last, the equation that updates the coordinates $\mathbf{x}_i^{(l+1)}$ is SE(3)-equivariant with respect to the 3D coordinates of nodes from the same graph as i, but SE(3)-invariant with respect to the 3D coordinates of nodes from the other graph since it only uses invariant transformations of the latter.

□

C Datasets

The overview of datasets is in Table 3.

Table 3: **Overview of Datasets.**

Dataset	# Pairs of Proteins	# Residues per Protein	# Atoms per Protein
DIPS	41876	276 (±189)	2159 (±1495)
DB5.5	253	268 (±215)	2089 (±1694)

D Hyperparameters

We perform hyperparameter search over the choices listed in Table 4 and select the best hyperparameters for DB5.5 and DIPS respectively based on their corresponding validation sets.

Table 4: Hyperparameter choices. LN stands for layer normalization, BN stands for batch normalization.

Hyperparameter	Choice
Node degree (for k-NN)	10
Weight of the intersection loss	0.0, 1.0
Normalization for h_i of IEGMN layers	No, LN
Normalization for others	No, BN, LN
Number of attention heads (K)	25, 50, 100
Slope of leaky relu	0.1, 0.01
Dimension of h_i of IEGMN layers	32, 64
Dimension of residue type embedding	32, 64
Number of IEGMN layers	5, 8
If IEGMN layers except the first one share parameters	True, False
η of coordinates skip connection	0.0, 0.25
Weight decay	0, 1e-5, 1e-4, 1e-3