

# Directed Weight Neural Networks for Protein Structure Representation Learning

Jiahua Li<sup>\*1,2</sup> Shitong Luo<sup>\*1</sup> Congyue Deng<sup>\*3</sup> Chaoran Cheng<sup>4</sup> Jiaqi Guan<sup>4</sup> Leonidas Guibas<sup>3</sup>  
Jian Peng<sup>1,4,5</sup> Jianzhu Ma<sup>2</sup>

## Abstract

A protein performs biological functions by folding to a particular 3D structure. To accurately model the protein structures, both the overall geometric topology and local fine-grained relations between amino acids (e.g. side-chain torsion angles and inter-amino-acid orientations) should be carefully considered. In this work, we propose the Directed Weight Neural Network for better capturing geometric relations among different amino acids. Extending a single weight from a scalar to a 3D directed vector, our new framework supports a rich set of geometric operations on both classical and  $SO(3)$ -representation features, on top of which we construct a perceptron unit for processing amino-acid information. In addition, we introduce an equivariant message passing paradigm on proteins for plugging the directed weight perceptrons into existing Graph Neural Networks, showing superior versatility in maintaining  $SO(3)$ -equivariance at the global scale. Experiments show that our network has remarkably better expressiveness in representing geometric relations in comparison to classical neural networks and the (globally) equivariant networks. It also achieves state-of-the-art performance on various computational biology applications related to protein 3D structures. All codes and models will be published upon acceptance.

(Rohl et al., 2004; Källberg et al., 2012; Baek et al., 2021; Jumper et al., 2021), another thread of tasks with protein 3D structures as input starts to draw a great interest, such as function prediction (Hermosilla et al., 2020; Gligorić et al., 2021), decoy ranking (Lundström et al., 2001; Kwon et al., 2021; Wang et al., 2021), protein docking (Duhovny et al., 2002; Shulman-Peleg et al., 2004; Gainza et al., 2020; Sverrisson et al., 2021), and driver mutation identification (Lefèvre et al., 1997; Antikainen & Martin, 2005; Li et al., 2020; Jankauskaitė et al., 2019).

Most existing works in modelling protein structures directly borrow neural network designs from other application areas such as computer vision and data mining (e.g. 3D-CNNs (Ji et al., 2012), GNNs (Kipf & Welling, 2016), Transformers (Vaswani et al., 2017)). Although compatible with general 3D object data, these models mainly focus on the overall protein topology but overlook the subtleties in the fine-grained geometries, which could be essential in many scenarios. For instance, given an amino acid characterized by its backbone atoms (carbon, nitrogen, and oxygen) and a side-chain, as shown in Figure 1, the locations of the backbone atoms determine the protein skeleton locally and its local frame orientation affects how it interacts with other amino acids, either of which can have important impacts on the entire protein structure (Nelson et al., 2008). It is also well-known that a small change to the side chain angles could cause a significant change to the protein functionality, especially when it interacts with other biological molecules (Jacobson et al., 2002; Misiura et al., 2021).

Recent attempts in building geometric-aware neural networks mainly focus on baking 3D *rigid transformations* into network operations, leading to the area of  $SO(3)$ -invariant and equivariant networks. One of representative works is the Vector Neuron Network (VNN) (Deng et al., 2021), which achieves  $SO(3)$ -equivariance on point clouds by generalizing scalar neurons to vectors. Another follow-up work is the GVP-GNN (Jing et al., 2021) that similarly vectorizes hidden neurons on GNN and demonstrates better prediction accuracy on protein design and quality evaluation tasks. These models adopt *linear combinations* of input vector features, which significantly limits their modeling capability.

## 1. Introduction

Built from a sequence of amino acid residues, a protein performs its biological functions by folding to a particular 3D conformation. Therefore, processing such 3D structures is the key for protein function analysis. While we have witnessed remarkable progress in protein structure predictions

<sup>\*</sup>Equal contribution <sup>1</sup>HeliXon Limited, China <sup>2</sup>Institute for Artificial Intelligence, Peking University <sup>3</sup>Stanford University <sup>4</sup>University of Illinois at Urbana-Champaign <sup>5</sup>Institute for Industry AI Research, Tsinghua University. Correspondence to: Jianzhu Ma <majianzhu@pku.edu.cn>.

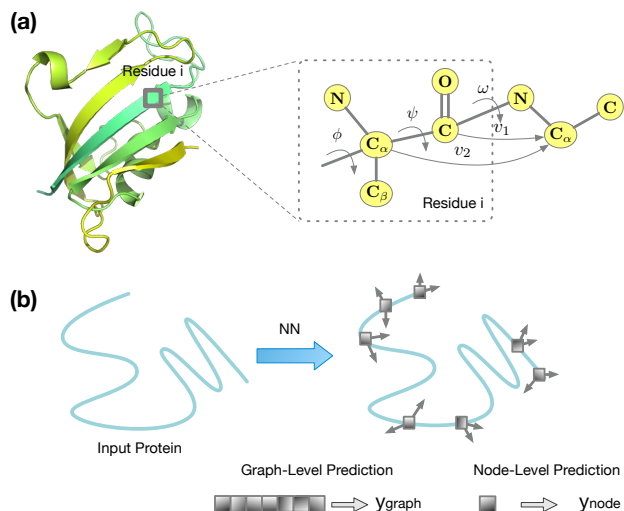


Figure 1. Overview. a). Each amino acid has its own **rigid backbone** with four heavy atoms, and can be represented by both lists of scalar and vector features. b). Protein representation tasks associated with the 3D structure. **Graph-level** tasks evaluate the whole protein structures, and **Node-level** tasks operate on each residues

A simple example is that, given two input vector features  $v_1$  and  $v_2$ , the outputs  $w_1 v_1 + w_2 v_2$  through one linear layer is constrained in the 2D plane spanned by  $v_1, v_2$  even after applying their vector version of non-linearities. A more concrete application in the experiment section demonstrates how VNN-based models are limited in perceiving angles (e.g. side-chain torsion angles), which have been proven crucial for proteins to interact with other partners through non-covalent bonds (Nelson et al., 2008).

To achieve more sensitive geometric awareness in both global transformations and local relations, we propose a *Directed Weight ( $\vec{W}$ ) perceptrons* by extending not only the hidden neurons but the weights from scalars to 2D/3D vectors, naturally saturating the entire network with 3D structures in the Euclidean space. Directed weights support a set of geometric operations on both the vector neurons (vector-list features) and the classical (scalar-list) latent features, and perform flexible integration of the hybrid scalar-vector features. As protein structures are naturally attributed proximity graphs, we introduce a new *Equivariant Message Passing Paradigm* on protein graphs, to connect the  $\vec{W}$ -perceptrons with the graph learning models by using rigid backbone transformations for each amino acid, which provides a versatile framework for bringing the biological suitability and flexibility of the GNN architecture. This paradigm frees the model architecture itself from introducing additional equivariance constraints (Cohen & Welling, 2016), making more complicated model designs possible.

To summarize, our key contributions include:

- We propose a new learning framework based on the

*Directed Weights* for capturing fine-grained geometric relations, especially for the subtle structure details in proteins.

- We construct an *Equivariant Message Passing* paradigm based on protein structures.
- Our overall framework is versatile in terms of compatibility with existing deep graph learning models, making them biologically suitable with minimal modifications of existing GNN models.
- We demonstrate our network expressivity over classical scalar-weighted networks on a synthetic dataset. Our models also achieve state-of-the-art performance on multiple protein learning tasks.

## 2. Related Work

**Representation learning on protein 3D structure.** Using machine learning models to understand protein 3D structures could be traced back to a decade ago. Early approaches mainly rely on hand-crafted features extracted from protein structures and simple statistical methods are adopted to predict multiple functional annotations (Schaap et al., 2001; Zhang & Zhang, 2010). Later, deep learning models have been found to achieve tremendous success in this area. 3D CNNs are first proposed to process protein 3D structures by scanning atom-level features relying on multiple 3D voxels. One of the representative works by Derevyanko et al. (2018) adopts a 3D CNN-based model for assessing the quality of the predicted structures and achieves state-of-the-art performance. Amidi et al. (2018) employs a similar idea to classify enzymes classes by using 3D CNN. 3D CNNs also shed light on other tasks such as interface prediction (Townshend et al., 2019) and protein fold recognition (Hermosilla et al., 2020). Gainza et al. (2020); Sverrisson et al. (2021) extend 3D CNNs to spherical convolutions for operating on radius regions, which can also be naturally applied to the Fourier space (Zhemchuzhnikov et al., 2021) and the 3D Voronoi Tessellation space (Igashov et al., 2021). Graph Convolutional Networks have also been adopted to capture geometric relationships and biochemical interactions between amino acids in the protein structures (Ying et al., 2018; Gao & Ji, 2019; Fout, 2017), and have been shown to achieve tremendous success in protein function prediction (Li et al., 2021), protein design (Strokach et al., 2020) and binding affinity/site prediction (Vecchio et al., 2021). More recently, transformer-based methods have drawn a lot of interest in extracting representations for protein structures and have a trend to replace conventional 3D CNN-based methods in other bioinformatics tasks (Ingraham et al., 2019; Baek et al., 2021; Jumper et al., 2021; Cao et al., 2021).

**Equivariant neural networks.** Equivariance is an important property for geometric learning models to generalize to unseen conditions (Cohen & Welling, 2016; Weiler et al.,

2018; Köhler et al., 2020; Satorras et al., 2021; Thomas et al., 2018; Fuchs et al., 2020; Anderson et al., 2019; Klicpera et al., 2020; Batzner et al., 2021; Eismann et al., 2021). While data augmentation is not an appropriate solution, neural network models such as Tensor Filed Network (Thomas et al., 2018) and Cormorant (Anderson et al., 2019), have been developed to generate irreducible representations for achieving rotation equivariant for 3D objects. In comparison to Tensor Filed Network and Cormorant, the vector neuron network (VNN) achieves rotation equivariance in a much simpler way by generalizing the values of hidden neurons from scalars to vectors defined in the 3D Euclidean space (Deng et al., 2021). Jing et al. (2020) later introduces GVP-GNN for learning protein structure representations by considering geometric factors such as orientations as vector representations and gains reasonable performance improvement on downstream tasks. Schütt et al. (2021) introduces equivariant message passing for vector representations using equivariant features. However, to guarantee rotation equivariance, these vector-based networks can only linearly combine 3D vectors, in essence, limiting their geometric expressiveness.

### 3. Directed Weight Perceptron

A protein is modeled as a KNN-graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  where each node  $u \in \mathcal{V}$  corresponds to one amino acid represented by a scalar-vector tuple  $h_u = (s_u, \mathbf{V}_u)$  with  $s_u \in \mathbb{R}^C$ ,  $\mathbf{V}_u \in \mathbb{R}^{C \times 3}$ , and the edges are constructed spatially by querying its  $k$ -nearest neighbours in the space. For notation simplicity, we set the channel numbers for scalar and vector features as the same but they can be different in practice. Details for the channel numbers can be found in the appendix Section B.1.

Classical neural networks only consider scalar features of the form  $s \in \mathbb{R}^C$ , with each layer transforming them with a weight matrix  $\mathbf{W} \in \mathbb{R}^{C' \times C}$  and a bias term  $\mathbf{b} \in \mathbb{R}^{C'}$ :

$$s' = \mathbf{W}s + \mathbf{b} \quad (1)$$

Although has been proved to be a universal approximator, such layer has intrinsically no geometric interpretation in the 3D shape space, limiting both expressiveness and robustness – even a simplest 3D rigid transformation on the input can result in drastic and unpredictable changes in the network outputs. Recent attempts studying equivariance (Deng et al., 2021; Jing et al., 2020) have lifted neuron features from scalar-lists to vector-lists  $\mathbf{V} \in \mathbb{R}^{C \times 3}$  with a layer operation:

$$\mathbf{V}' = \mathbf{W}\mathbf{V} \quad (2)$$

Under this construction, latent space  $\text{SO}(3)$ -actions are simply matrix (right) multiplications, which establishes a clear 3D Euclidean structure on the latent space and shows input-output consistency under rigid transformations. Despite the vectorization of features and their hidden representations,

their operations are still limited to classical linear combinations weighted by  $\mathbf{W}$ . To define *operations* that are more adaptive to the geometrically meaningful features, we introduce the *Directed Weights*, denoted by  $\vec{\mathbf{W}} \in \mathbb{R}^{C' \times C \times 3}$ , accompanied with a set of geometric operators  $\square$  (e.g.  $\cdot$  or  $\times$ ) on both scalar-list features  $s \in \mathbb{R}^C$  and vector-list features  $\mathbf{V} \in \mathbb{R}^{C \times 3}$

$$\vec{\mathbf{W}} \square s, \quad \vec{\mathbf{W}} \square \mathbf{V} \quad (3)$$

, allowing flexible architectural designs with rich expressiveness in geometric relations.

#### 3.1. $\vec{\mathbf{W}}$ -Operators

**Geometric Vector Operations.** Let  $\mathbf{W} \in \mathbb{R}^{C' \times C}$  be a conventional scalar weight matrix,  $\vec{\mathbf{W}}_1 \in \mathbb{R}^{C' \times C \times 3}$  and  $\vec{\mathbf{W}}_2 \in \mathbb{R}^{C \times 3}$  be directed weight tensors. We can define two operations that leverage geometric information:

$$s'_{\text{dot}}(\mathbf{V}; \vec{\mathbf{W}}_1) = \vec{\mathbf{W}}_1 \mathbf{V} \in \mathbb{R}^{C'} \quad (4)$$

$$\mathbf{V}'_{\text{crs}}(\mathbf{V}; \vec{\mathbf{W}}_1, \vec{\mathbf{W}}_2) = \vec{\mathbf{W}}_1 \cdot (\vec{\mathbf{W}}_2 \times \mathbf{V}) \in \mathbb{R}^{C' \times 3} \quad (5)$$

Here  $s'_{\text{dot}}$  transforms  $C$  vector features to  $C'$  scalars using dot-product with directed weights, which explicitly measures angles between vectors. In  $\mathbf{V}'_{\text{crs}}$ , a vector crosses a directed weight before being projected onto the hidden space. As the output of plain linear combinations between two vectors  $\mathbf{v}_1$  and  $\mathbf{v}_2$  can only lie in the plane  $w_1\mathbf{v}_1 + w_2\mathbf{v}_2$ , cross-product in Equation 5 creates a new direction outside the plane, which is crucial in 3D modelling. For instance, the side-chain angles of a given residue could be largely determined by its  $C_\alpha$  and  $C_\beta$  vectors, but may lie on the direction perpendicular to the space constructed by the two vectors.

**Scalar Lifting.** In addition, a directed weight  $\vec{\mathbf{W}} \in \mathbb{R}^{C \times 3}$  can lift scalars to vectors by adopting the following operation,

$$\mathbf{V}'_{\text{lift}}(s; \vec{\mathbf{W}}) = s\vec{\mathbf{W}} \in \mathbb{R}^{C \times 3} \quad (6)$$

This maps each scalar to a particular vector, enabling inverse transformations from  $\mathbb{R}$  to  $\mathbb{R}^3$ . For instance, we can map a scalar representing the distance between two amino acids to a vector pointing from one amino acid to the other, leading to more biological meaningful representations for the protein fragment.

**Linear Combinations.** In the end, we keep the linear combination operations with scalar weights for both scalar and vector features:

$$s'_{\text{lin}}(s; \mathbf{W}) = \mathbf{W}s \in \mathbb{R}^{C'} \quad (7)$$

$$\mathbf{V}'_{\text{lin}}(\mathbf{V}; \mathbf{W}) = \mathbf{W}\mathbf{V} \in \mathbb{R}^{C' \times 3} \quad (8)$$

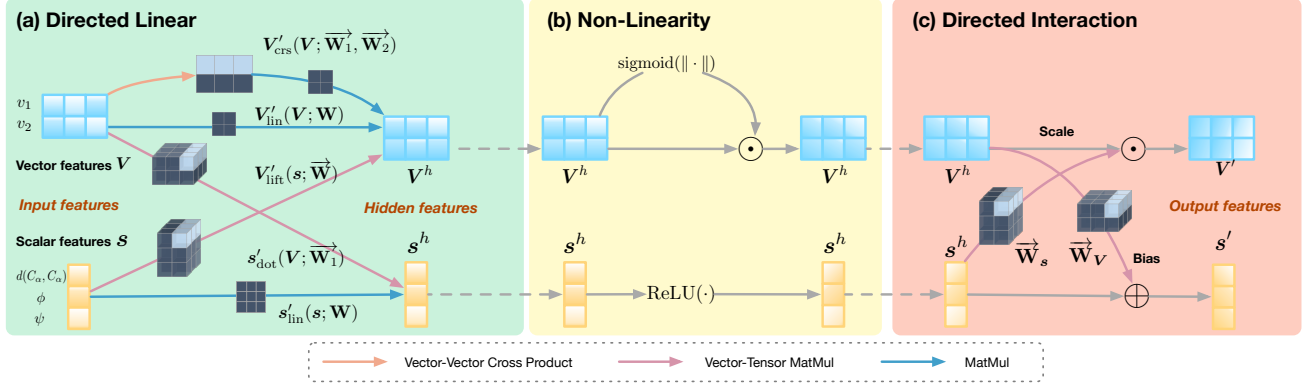


Figure 2. Model Details. a) **Directed Linear Module** applies multiple geometric operations to update scalar and vector features respectively. b) **Non-Linearity Modules** employs ReLU and sigmoid functions for scalar and vector representations. c) **Directed Interaction Module** updates the features by using one another as updating parameters.

### 3.2. $\vec{W}$ -Perceptron Unit

Now we combine all the  $\vec{W}$ -operators together, assembling them into what we call Directed Weight Perceptrons ( $\vec{W}$ -perceptrons) for processing the protein 3D structures. A  $\vec{W}$ -perceptron unit is a function mapping from  $u = (s, V)$  to another scalar-vector tuple  $u' = (s', V')$ , which can be stacked as network layers to form multi-layer perceptrons.

A single unit comprises three modules in a sequential way: the **Directed Linear** module, the **Non-Linearity** module, and the **Directed Interaction** module (Figure 2). We show that these modules can properly tackle the defects of prior models and encourage efficient scalar-vector feature interactions within the network.

**Directed Linear Module.** The output hidden representations derived from the set of operations introduced previously are concatenated and projected into another hidden space of scalars and vectors separately:

$$s^h = \mathbf{W}_s[s'_{\text{dot}}, s'_{\text{lin}}] \in \mathbb{R}^{C'} \quad (9)$$

$$V^h = \mathbf{W}_V[V'_{\text{crs}}, V'_{\text{lift}}, V'_{\text{lin}}] \in \mathbb{R}^{C' \times 3} \quad (10)$$

Here  $\mathbf{W}_s \in \mathbb{R}^{C' \times (C' + C')}$  and  $\mathbf{W}_V \in \mathbb{R}^{C' \times (C' + C + C')}$  are scalar weight matrices. In other word, the five separate updating functions allow transformation from scalar to scalar (Equation 7), scalar to vector (Equation 6), vector to scalar (Equation 4), and vector to vector (Equation 8, Equation 5), , boosting the model’s expressiveness to reason geometrically.

**Non-linearity Module.** We then apply the non-linearity module to the hidden representation ( $s^h, V^h$ ). Specifically, we apply standard ReLU non-linearity (Nair & Hinton, 2010) to the scalar components. For the vector representations, following (Weiler et al., 2018; Jing et al., 2020; Schütt et al., 2021), we compute a sigmoid activation on the

L2-norm of each vector and multiply it back to the vector entries accordingly:

$$s^h \leftarrow \text{ReLU}(s^h) \quad (11)$$

$$v_{ij}^h \leftarrow v_{ij}^h \cdot \text{sigmoid}(\|v_i^h\|_2) \quad (12)$$

where  $v_i^h \in \mathbb{R}^3$  are the vector columns in  $V^h$  and  $v_{ij}^h$  are their entries.

**Directed Interaction Module.** Finally we introduce the Directed Interaction module, integrating the hidden features  $s^h$  and  $V^h$  into the output tuple ( $s', V'$ )

$$s' = s^h + \vec{\mathbf{W}}_V \cdot V^h \in \mathbb{R}^{C'} \quad (13)$$

$$V' = (s^h \vec{\mathbf{W}}_s) \odot V^h \in \mathbb{R}^{C' \times 3} \quad (14)$$

Here  $\vec{\mathbf{W}}_V, \vec{\mathbf{W}}_s$  are directed weight matrices with sizes  $C' \times C' \times 3$  and  $C' \times 3$ , respectively,  $\odot$  denotes element-wise multiplication for two matrices. This module establishes a connection between the scalar and vector feature components, facilitating feature blending. Specifically, Equation 13 dynamically determines how much the output should rely on scalar and vector representations, and Equation 14 weights a list of vectors using the scalar features as attention scores.

## 4. $\vec{W}$ -Equivariant Graph Neural Networks

Conventional message-passing-based graph neural networks (Gilmer et al., 2017; Thomas et al., 2018) for molecular modeling do not allow scalar-vector communications and usually treat them as separate channels. While being versatile enough to be plugged into any existing message passing networks, our  $\vec{W}$ -Perceptron also provides a more flexible toolbox bridging scalar and vector channels with geometrical and biological interpretations. On the other hand, to achieve SO(3)-equivariance without the loss of expressivity,



we introduce an equivariant message passing paradigm for protein graphs, making network architecture designs free from the geometric constraints (Section 4.1). In the end, we demonstrate our adaptability by integrating this framework into multiple graph neural networks variants (Section 4.2).

#### 4.1. Equivariant Message Passing on Proteins

A function  $f : \mathbb{R}^3 \rightarrow \mathbb{R}^3$  is  $\text{SO}(3)$ -equivariant (rotation-equivariant) if any rotation matrix  $R \in \mathbb{R}^{3 \times 3}$  applied to the input vector  $x$  leads to the same transformation on the output  $f(x)$ :

$$Rf(x) = f(Rx) \quad (15)$$

Such equivariant property can be achieved on a protein graph with local orientations naturally defined from its biological structure. Specifically, each amino acid node  $u_i \in \mathcal{V}$  has four backbone heavy atom ( $C_\alpha^u, C_\beta^u, N^u, O^u$ ), defining a local frame  $O_u$  as:

$$x_u = N^u - C_\alpha^u \in \mathbb{R}^3 \quad (16)$$

$$y_u = C_\beta^u - C_\alpha^u \in \mathbb{R}^3 \quad (17)$$

$$O_u = [x_u, y_u, x_u \times y_u]^\top \in \mathbb{R}^{3 \times 3} \quad (18)$$

The local frame  $O_u$  is a rotation matrix that maps a 3D vector from the local to the global coordinate system. An equivariant message passing paradigm then emerges through transforming node features back and forth between adjacent local frames. Formally, give an amino acid  $u$  with hidden representation  $h = (s, V)$ , let  $f_l$  and  $f_g$  be transformations on the vector feature  $V$  from and to the global coordinate system:

$$f_l(h, O_u) := (s, VO_u^\top) \quad (19)$$

$$f_g(h, O_u) := (s, VO_u) \quad (20)$$

The message passing update for node  $u$  from layer  $l$  to layer  $l + 1$  is performed in the following steps:

1. Transform the neighbourhood vector representations of  $u$  into its local coordinate system  $O_u$ .
2. For each adjacent node  $v$ , compute the message  $m_u^{l+1}$  on edge  $(u, v) \in \mathcal{E}$  with an multi-layer  $\vec{W}$ -perceptron  $\mathcal{F}$  to the aggregated node and edge feature  $[h_u^l, h_v^l, e_{uv}^l]$  (Equation 21).
3. Update node feature  $h_u^l$  with another multi-layer  $\vec{W}$ -perceptron  $\mathcal{H}$ , and transform it back to the global coordinate system (Equation 22).

This paradigm achieves  $\text{SO}(3)$ -equivariance in a very general sense with no constraints on  $\mathcal{F}, \mathcal{H}$ . The formal proof of equivariance is presented in the appendix Section A.

$$m_u^{l+1} = \sum_{v \in \mathcal{N}_u} \mathcal{F}(f_l([h_u^l, h_v^l, e_{uv}^l], O_u)) \quad (21)$$

$$h_u^{l+1} = f_g(\mathcal{H}(h_u^l, m_u^{l+1}), O_u) \quad (22)$$

#### 4.2. $\vec{W}$ -Graph Neural Networks

Different variants of GNN models have been demonstrated to be helpful for different bioinformatics tasks involving protein 3D structures (Ingraham et al., 2019; Strokach et al., 2020; Baldassarre et al., 2021). Therefore, we integrate the  $\vec{W}$ -Perceptrons with our proposed equivariant message passing paradigm into multiple popular GNN frameworks.

**DW-GCN.** This is the one described and implemented in the previous subsection with Equation 21 and Equation 22.

**DW-GIN.** We adopt graph isomorphism operator (Xu et al., 2018) with learnable weighing parameter  $\varepsilon$  for tunable skip connections,

$$m_u^{l+1} = (1 + \varepsilon)f_l(h_u^l, O_u) + \sum_{v \in \mathcal{N}_u} \mathcal{F}(f_l(h_v^l, O_u)) \quad (23)$$

**DW-GAT.** In comparison to GCN and GIN, it is not trivial to incorporate with the Graph Attention Network (GAT) (Veličković et al., 2017). In particular, we define separate attentions for the scalar and vector representations

$$(s_u^l, V_u^l) = f_l(h_u^l, O_u) \quad (24)$$

$$s'_u = \sum_{v \in \mathcal{N}_u \cup \{u\}} \alpha_v^s s_v^l \quad (25)$$

$$V'_u = \sum_{v \in \mathcal{N}_u \cup \{u\}} \alpha_v^v V_v^l \quad (26)$$

$$h_u^{l+1} = f_g(\mathcal{H}((s', V')), O_u) \quad (27)$$

The attention scores  $\alpha^s, \alpha^v$  are the softmax values over the inner products of all neighboring source-target pairs defined as the follows:

$$\alpha_v^s = \frac{\exp(\langle s_u, s_v \rangle)}{\sum_{w \in \mathcal{N}_u \cup \{u\}} \exp(\langle s_u, s_w \rangle)} \quad (28)$$

$$\alpha_v^v = \frac{\exp(\text{tr}(V_u^\top V_v))}{\sum_{w \in \mathcal{N}_u \cup \{u\}} \exp(\text{tr}(V_u^\top V_w))} \quad (29)$$

The product for scalars and vectors are standard inner product (Equation 28) and Frobenius product (Equation 29), respectively. With the attention mechanism, an amino acid can determine the subset of neighbors which contribute more to obtain representations, as amino acids interact with each other unevenly.

## 5. Experiments

We first design a synthetic task (Section 5.1) to demonstrate the intuition and basic rationale why our  $\vec{W}$ -Perceptron is more powerful in perceiving angular features compared to other frameworks. Next, we conduct extensive experiments to evaluate the performance of  $\vec{W}$ -GNN variants on multiple benchmarks, including two node-level classification tasks:

Residue Identification (**RES**) (Section 5.2), Computational Protein Design (**CPD**) (Section 5.3), and three graph-level tasks: Model Quality Assessment (**MQA**) (Section 5.4), Fold Classification (**FOLD**), and Enzyme Reaction Classification (**REACT**) (Section 5.5). Ablation studies and other training details are documented in the appendix Section B and Section C.

### 5.1. Synthetic Task

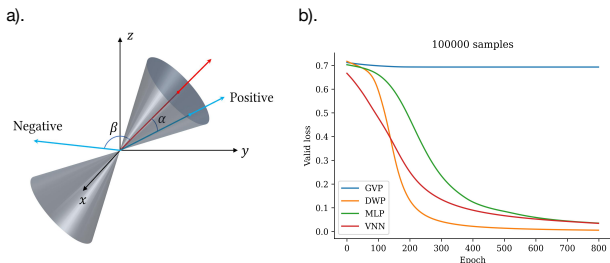


Figure 3. a). **The synthetic study.** The vector in red is the anchor used to calculate the ground truth labels. Vectors falling into the cone are **positive** samples, whereas laying outside the cone are **negative** samples. b). The curves are the values of loss functions of different methods to be compared.

**Datasets.** We test the performance of different approaches on a synthetic binary angular classification dataset. We sample a random vector  $v_m \in \mathbf{R}^3$  as the anchor vector. Then each vector  $v_i$  in the space is labeled positive if its angle between the anchor vector  $a_{mi} \leq \pi/4$ , and negative otherwise. We generate 100,000 random vectors with 50% positive samples and 50% negative samples. For a fair comparison, we restrict the number of parameters of different models to approximately the same.

**Results.** As shown in Figure 3, in general, our Directed Weight Perceptron (**DWP**) converges faster than other models and achieves the lowest validation loss. We also notice that the Geometric Vector Perceptron (**GVP**) (Jing et al., 2020) cannot converge given a sufficient number of epochs due to the poor capability of perceiving angular features by only using linear combinations of 3D vector features. In comparison to GVP, Vector Neural Network (**VNN**) (Deng et al., 2021) performs better due to its non-linear ReLU operation for vector features. Plain Multi-Layer Perceptron (**MLP**) can also capture that according to its universal approximation ability. The superior performance demonstrates that our  $\vec{W}$ -Perceptron has a better ability on perceiving potential geometric features in space.

### 5.2. Residue Identification

**Datasets.** Residue Identification (**RES**) aims to predict the identity of a particular masked amino acid in the protein 3D structure based on its local surrounding structure context

(Torng & Altman, 2017). We download 100,000 substructures from the ATOM3D project (Townshend et al., 2020), which are originally derived from 574 proteins from the PDB (Berman et al., 2000). We split the entire dataset into training, validation, and testing datasets with the ratio of 80%, 10%, 10% and make sure there are no two proteins with similar structures in the test and non-test datasets.

**Metrics.** RES is a classification task that assigns each masked position a certain type of amino acid. We use classification accuracy to evaluate all the methods to be compared.

**Baselines.** The **3D-CNN** model (Ji et al., 2012) encodes the positions of the protein’s atoms in a voxelized 3D volume. We also compare GNN variants such as **GCN** (Kipf & Welling, 2016), **GIN** (Xu et al., 2018) and **GAT** (Veličković et al., 2017) with our **DW-GCN**, **DW-GIN** and **DW-GAT** models. In addition, we include **GVP-GNN** (Jing et al., 2021) discussed above as well.

**Results** As shown in Table 1, **GCN**, **GIN** and **GAT** almost cannot accurately identify the type of amino acids based on the local protein structures, suggesting the lack of ability of conventional GNN models to capture 3D geometric information. **3D-CNN** performs better due to the power of the voxelization technique in the 3D space. Our models outperform other models, suggesting  $\vec{W}$ -GNNs could help represent protein substructure geometries with the equivariant property.

### 5.3. Computational Protein Design

**Datasets.** Computational Protein Design (**CPD**) predicts whether a protein sequence can fold into a given backbone structure, and is viewed as the inverse problem of protein folding. Specifically, we focus on two protein structure databases CATH and TS50. CATH v4.2 is collected by (Ingraham et al., 2019) and organizes all structures in a hierarchical structure (Orengo et al., 1997). We choose 18,024 chains as the training set, 608 as the validation set, and 1,120 as the test set. TS50 dataset (Li et al., 2014) is a relatively old benchmark consisting of 50 protein structures and is popular adopted in structural biology communities (Qi & Zhang, 2020; Zhang et al., 2020).<sup>1</sup>

**Metrics.** Native Sequence Recovery Rate is adopted to evaluate the prediction performance, which compares the predicted protein sequence with the ground truth sequence at each position and calculates the proportion of the correctly recovered amino acids (Li et al., 2014). Ingraham et al.

<sup>1</sup>As there is no canonical training set for TS50, we follow prior methods to collect protein sequences with less than 30% similarity with the TS50 test set from the CATH training set and use these new protein sequences for training (Jing et al., 2020; Qi & Zhang, 2020).

(2019) also introduces the perplexity score (Jelinek et al., 1977) which evaluates whether a model could assign high likelihood to the test sequences.

Table 1. Results of different RES methods on ATOM3D.

Model	3D-CNN	GCN	GIN	GAT
Acc %	45.1	8.2	9.1	12.4
Model	GVP-GNN	DW-GCN	DW-GIN	DW-GAT
Acc %	48.2	50.2	<b>50.8</b>	47.5

Table 2. Results of different CPD methods on the CATH 4.2.

Model	Perplexity ↓			Recovery % ↑		
	Short	Single	All	Short	Single	All
St-Transformer	8.54	9.03	6.85	28.3	27.6	36.4
St-GCN	8.31	8.88	6.55	28.4	28.1	37.3
St-GIN	8.03	8.52	6.15	27.7	28.4	38.1
St-GAT	10.86	10.67	9.89	26.2	26.8	35.2
GVP-GNN	7.10	7.44	5.29	32.1	32.0	40.2
DW-GCN	<b>5.42</b>	5.69	3.94	<b>39.6</b>	38.5	47.5
DW-GIN	5.84	<b>5.39</b>	<b>3.85</b>	38.8	<b>40.1</b>	<b>47.8</b>
DW-GAT	6.02	5.53	4.13	37.5	39.3	46.7

Table 3. Results of structure biology methods on the TS50.

Model	Rosetta	SPIN	ProteinSolver	Wang’s	SPIN2	SBROF	ProDCoNN	St-Trans
Recv %	30.0	30.3	30.8	33.0	33.6	39.2	40.7	42.3
Model	GCN	GIN	GAT	GVP-GNN	DW-GCN	DW-GAT	DW-GIN	
	50.7	31.2	32.4	32.7	44.9	53.8	<b>54.5</b>	52.7

**Baselines.** For the CATH v4.2 dataset, we compare our model with three models which have achieved the state-of-the-art performance on these sequence design task, including **GVP-GNN**, **Structured Transformer** and **Structured GNN** proposed by Ingraham et al. (2019). Structured Transformer employs transformer-based architecture and all of these three models generate the sequence in an auto-regressive way. We compare multiple machine learning approaches in the TS50 dataset, including 3D CNN-based methods (**ProDCoNN** (Zhang et al., 2020), **DenseCPD** (Qi & Zhang, 2020)), sequential models (**Wang’s model** (Wang et al., 2018), **SPIN** (Li et al., 2014), **SPIN2** (O’Connell et al., 2018)) and GNN-based method (Strokach et al., 2020). We also compare our performance with relatively simple methods based on classical statistics (Schaap et al., 2001; Cheng et al., 2019).

**Results.** As shown in Table 2, our models achieve significant improvement over the state-of-the-art methods on both the perplexity and recovery metrics on the CATH dataset. This result also suggests that the prediction tasks on the short-chain and single-chain subsets are more challenging, which is consistent with the result in (Ingraham et al., 2019).

As shown in Table 3, our models also gain superior performance over the previous state-of-the-art method DenseCPD at least 2.0%. These results provide strong evidence that adopting  $\vec{W}$ -GNNs is a more efficient way for designing valid protein sequences.

#### 5.4. Model Quality Assessment

**Datasets.** Model quality assessment (MQA) evaluates the quality of predicted protein structures by computational algorithms (Kwon et al., 2021). The goal is to fit a machine learning model approximating the numerical metric used to compare the predicted 3D structure with the native 3D structure (Lundström et al., 2001). Some methods predict global metrics GDT\_TS (Zemla, 2003) and TM-score (Zhang & Skolnick, 2004) for ranking structures. Others regress local quality scores such as LDDT (Mariani et al., 2013) and CAD (Olechnovič et al., 2013), which focus more on identifying incorrect substructures. For this task, we collect the data from CASP (Moult et al., 2014) with each target having multiple scored decoys. We randomly split the targets in CASP5-CASP10 and sample 50 decoys for each target for generating the training and validation sets, and use the CASP11 stage 2 proteins as the test set to ensure no similar structures are involved. In total, we collect 508 proteins for training, 56 for validation, and 85 targets with 150 decoys for the test.

**Metrics.** We choose the GDT\_TS (Zemla, 2003) as the metric to evaluate the model quality, then adopt the correlation between the predicted and real GDT\_TS values to measure the performance of different MQA methods. We employ three statistical correlation metrics: Pearson’s correlation  $r$ , Spearman’s  $\rho$ , and Kendall’s  $\tau$ . We calculate the correlation for each target and average all the correlations over all the targets. We also calculate the Global correlations by taking the union of all decoy sets without considering the targets (Pagès et al., 2019).

**Baselines.** **VoroMQA** (Olechnovič & Venclovas, 2017) leverages statistical potential model with protein contact maps, while **RWplus** (Zhang & Zhang, 2010) relies on physical energy terms. **SBROD** (Karasikov et al., 2019) uses hand-crafted features as prior knowledge to measure the model quality. **Proq3D** (Uziela et al., 2017) employs a fully connected neural network for regression. **3DCNN** (Derevyanko et al., 2018) and **Ornate** (Pagès et al., 2019) apply 3D CNNs which have been thoroughly discussed in the previous sections. **GraphQA** (Baldassarre et al., 2021), **DimeNet** (Klicpera et al., 2020), **GVP-GNN** are the most similar models as ours which adopted GNN-based methods on protein graphs.

**Results.** As shown in table 4, our models outperform other methods on both average and global metrics except for Spearman’s correlation  $\rho$  where we achieve the second-

best, demonstrating our models can not only evaluate the quality for the same protein but also works well across different proteins in a more general way. Notably, DW-GAT performs best in average metrics by capturing local geometric information using the elaboratively-designed attention module.

Table 4. Results of MQA models on CASP 11 stage 2.

Model	Average $\uparrow$			Global $\uparrow$		
	$r$	$\rho$	$\tau$	$r$	$\rho$	$\tau$
VoroMQA	0.42	0.41	0.29	0.65	0.69	0.51
RWplus	0.17	0.19	0.13	0.06	0.03	0.01
SBROD	0.43	0.41	0.29	0.55	0.57	0.39
Proq3D	0.44	<b>0.43</b>	0.30	0.77	<b>0.80</b>	0.59
3DCNN	0.49	0.39	0.27	0.64	0.67	0.48
Ornate	0.39	0.37	0.26	0.63	0.67	0.48
DimeNet	0.30	0.35	0.28	0.61	0.62	0.43
GraphQA	0.48	0.40	0.42	0.75	0.72	0.74
GVP-GNN	0.58	0.33	0.46	0.80	0.61	0.81
DW-GCN	0.62	0.37	0.51	0.84	0.65	0.85
DW-GIN	0.63	0.36	0.52	<b>0.86</b>	0.67	<b>0.88</b>
DW-GAT	<b>0.65</b>	0.42	<b>0.53</b>	0.83	0.69	0.86

## 5.5. Fold and Reaction Classification

**Datasets.** Fold Classification (FOLD) predicts the structure category of a given protein structure. We choose the SCOP v1.75 dataset collected by (Murzin et al., 1995) that organizes structures into 3-layer hierarchical classes. In total, there are 16,712 proteins covering 7 major structural types with 1,195 identified folds. We adopt a reduced dataset from Hou et al. (2018), and remove homologous sequences between test and training data sets at Family (**Fam**), Superfamily (**Sup**) or **Fold** levels, resulting in three different classification tasks.

The Enzyme-Catalyzed Reaction Classification (REACT) study is a very similar task as it requires to classify the EC number (Webb et al., 1992) of a catalyzed enzyme based on the protein structures, which is basically to classify protein folds for different enzymes. Therefore, we put these two tasks side-by-side to evaluate the performance of different methods. We collect 37,428 proteins from 384 types of Enzyme-Catalyzed classes and split them into training, validation set, test set and ensure no sequence or structure overlaps across different sets. In total, there are 29,215 structures for training, 2,562 for validation, and 5,651 for test.

**Metrics.** Following standard benchmarks (Diehl, 2019; Hermosilla Casajus et al., 2021), we use the classification accuracy for evaluating the prediction performance.

**Baselines.** We compare with **CNN-based** models and **GNN-based** models which learn the protein annotations using 3D structures from scratch. For fair comparison, we do not include LSTM-based or transformer-based methods, as they all pre-train their models using millions of protein sequences and only fine-tune their models on 3D structures (Bepler & Berger, 2019; Alley et al., 2019; Rao et al., 2019; Strodtthoff et al., 2020; Elnaggar et al., 2020).

**Results.** As shown in Table 5, our models demonstrate superior prediction performance in both tasks across different levels of the hierarchy in comparison to others. Classifying the fold and enzyme classes based on the protein structure is one of the key problems in structural biology. These strong improvements state our proposed models can be used as new tools to conduct function annotations for new proteins.

Table 5. Results on Fold and React classification

	Architecture	FOLD			REACT
		Fold	Sup	Fam	
Hou et al. (2018)	1D ResNet	17.0%	31.0%	77.9%	70.9%
Derevyanko et al. (2018)	3D CNN	30.2%	37.3%	89.5%	75.1%
Kipf & Welling (2016)	GCNN	16.8%	21.3%	82.8%	67.3%
Diehl (2019)	GCNN	12.9%	16.3%	72.5%	57.9%
Baldassarre et al. (2021)	GCNN	23.7%	32.5%	84.4%	60.8%
Hermosilla Casajus et al. (2021)	GCNN	31.0%	35.6%	<b>92.5%</b>	76.3%
Glgorijević et al. (2021)	LSTM+GCNN	15.3%	20.6%	73.2%	63.3%
Our method	DW-GCN	31.2%	<b>37.8%</b>	84.8%	76.0%
Our method	DW-GIN	<b>31.8%</b>	37.3%	85.2%	<b>76.7%</b>
Our method	DW-GAT	29.9%	36.6%	79.0%	57.2%

## 6. Conclusion

For the first time, we generalize the scalar weights in neural networks to 3D directed vectors for better perception of geometric features like orientations and angles, which are essential in learning representations from protein 3D structures. Based on the directed weights  $\vec{W}$ , we provide a toolbox of operators as well as a  $\vec{W}$ -Perceptron Unit encouraging efficient scalar-vector feature interactions. We also enforce global  $SO(3)$ -equivariance on a message passing paradigm using the local orientations naturally defined by the rigid property of each amino acid residue, which can be used for designing more powerful networks. Our  $\vec{W}$  versions of popular graph neural network architectures achieve comparably good or better performances on multiple biological tasks in comparison with existing state-of-the-art methods.

**Limitations and future work.** While our methods show great performance, there is still a huge design space of Non-Linearity functions for  $SO(3)$  vector representations. In the future, we will extend our framework to other 3D learning tasks like small molecules or point clouds by learning local rotation transformations and using directed vector weights.



## References

- Alley, E. C., Khimulya, G., Biswas, S., AlQuraishi, M., and Church, G. M. Unified rational protein engineering with sequence-based deep representation learning. *Nature methods*, 16(12):1315–1322, 2019.
- Amidi, A., Amidi, S., Vlachakis, D., Megalooikonomou, V., Paragios, N., and Zacharaki, E. I. Enzyenet: enzyme classification using 3d convolutional neural networks on spatial representation. *PeerJ*, 6:e4750, 2018.
- Anderson, B., Hy, T.-S., and Kondor, R. Cormorant: Covariant molecular neural networks. *arXiv preprint arXiv:1906.04015*, 2019.
- Antikainen, N. M. and Martin, S. F. Altering protein specificity: techniques and applications. *Bioorganic & medicinal chemistry*, 13(8):2701–2716, 2005.
- Baek, M., DiMaio, F., Anishchenko, I., Dauparas, J., Ovchinnikov, S., Lee, G. R., Wang, J., Cong, Q., Kinch, L. N., Schaeffer, R. D., et al. Accurate prediction of protein structures and interactions using a three-track neural network. *Science*, 373(6557):871–876, 2021.
- Baldassarre, F., Menéndez Hurtado, D., Elofsson, A., and Azizpour, H. Graphqa: protein model quality assessment using graph convolutional networks. *Bioinformatics*, 37(3):360–366, 2021.
- Batzner, S., Smidt, T. E., Sun, L., Mailoa, J. P., Kornbluth, M., Molinari, N., and Kozinsky, B. Se (3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. *arXiv preprint arXiv:2101.03164*, 2021.
- Bepler, T. and Berger, B. Learning protein sequence embeddings using information from structure. *arXiv preprint arXiv:1902.08661*, 2019.
- Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., Weissig, H., Shindyalov, I. N., and Bourne, P. E. The protein data bank. *Nucleic acids research*, 28(1): 235–242, 2000.
- Cao, Y., Das, P., Chenthamarakshan, V., Chen, P.-Y., Melnyk, I., and Shen, Y. Fold2seq: A joint sequence (1d)-fold (3d) embedding-based generative model for protein design. In *International Conference on Machine Learning*, pp. 1261–1271. PMLR, 2021.
- Cheng, J., Choe, M.-H., Elofsson, A., Han, K.-S., Hou, J., Maghrabi, A. H., McGuffin, L. J., Menéndez-Hurtado, D., Olechnovič, K., Schwede, T., et al. Estimation of model accuracy in casp13. *Proteins: Structure, Function, and Bioinformatics*, 87(12):1361–1377, 2019.
- Cohen, T. and Welling, M. Group equivariant convolutional networks. In *International conference on machine learning*, pp. 2990–2999. PMLR, 2016.
- Deng, C., Litany, O., Duan, Y., Poulenard, A., Tagliasacchi, A., and Guibas, L. Vector neurons: A general framework for so (3)-equivariant networks. *arXiv preprint arXiv:2104.12229*, 2021.
- Derevyanko, G., Grudinin, S., Bengio, Y., and Lamoureux, G. Deep convolutional networks for quality assessment of protein folds. *Bioinformatics*, 34(23):4046–4053, 2018.
- Diehl, F. Edge contraction pooling for graph neural networks. *arXiv preprint arXiv:1905.10990*, 2019.
- Duhovny, D., Nussinov, R., and Wolfson, H. J. Efficient unbound docking of rigid molecules. In *International workshop on algorithms in bioinformatics*, pp. 185–200. Springer, 2002.
- Eismann, S., Townshend, R. J., Thomas, N., Jagota, M., Jing, B., and Dror, R. O. Hierarchical, rotation-equivariant neural networks to select structural models of protein complexes. *Proteins: Structure, Function, and Bioinformatics*, 89(5):493–501, 2021.
- Elnaggar, A., Heinzinger, M., Dallago, C., Rihawi, G., Wang, Y., Jones, L., Gibbs, T., Feher, T., Angerer, C., Steinegger, M., et al. Prottrans: towards cracking the language of life’s code through self-supervised deep learning and high performance computing. *arXiv preprint arXiv:2007.06225*, 2020.
- Fout, A. M. *Protein interface prediction using graph convolutional networks*. PhD thesis, Colorado State University, 2017.
- Fuchs, F. B., Worrall, D. E., Fischer, V., and Welling, M. Se (3)-transformers: 3d roto-translation equivariant attention networks. *arXiv preprint arXiv:2006.10503*, 2020.
- Gainza, P., Sverrisson, F., Monti, F., Rodola, E., Boscaini, D., Bronstein, M., and Correia, B. Deciphering interaction fingerprints from protein molecular surfaces using geometric deep learning. *Nature Methods*, 17(2):184–192, 2020.
- Gao, H. and Ji, S. Graph u-nets. In *international conference on machine learning*, pp. 2083–2092. PMLR, 2019.
- Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., and Dahl, G. E. Neural message passing for quantum chemistry. In *International conference on machine learning*, pp. 1263–1272. PMLR, 2017.
- Gligorijević, V., Renfrew, P. D., Kosciółek, T., Leman, J. K., Berenberg, D., Vatanen, T., Chandler, C., Taylor, B. C.,

- Fisk, I. M., Vlamakis, H., et al. Structure-based protein function prediction using graph convolutional networks. *Nature communications*, 12(1):1–14, 2021.
- Hermosilla, P., Schäfer, M., Lang, M., Fackelmann, G., Vázquez, P. P., Kozlíková, B., Krone, M., Ritschel, T., and Ropinski, T. Intrinsic-extrinsic convolution and pooling for learning on 3d protein structures. *arXiv preprint arXiv:2007.06252*, 2020.
- Hermosilla Casajus, P., Schäfer, M., Lang, M., Fackelmann, G., Vázquez Alcocer, P. P., Kozlíková, B., Krone, M., Ritschel, T., and Ropinski, T. Intrinsic-extrinsic convolution and pooling for learning on 3d protein structures. In *International Conference on Learning Representations, ICLR 2021: Vienna, Austria, May 04 2021*, pp. 1–16. OpenReview. net, 2021.
- Hou, J., Adhikari, B., and Cheng, J. Deepsf: deep convolutional neural network for mapping protein sequences to folds. *Bioinformatics*, 34(8):1295–1303, 2018.
- Igashov, I., Olechnovič, K., Kadukova, M., Venclovas, Č., and Grudin, S. Vorocnn: deep convolutional neural network built on 3d voronoi tessellation of protein structures. *Bioinformatics*, 37(16):2332–2339, 2021.
- Ingraham, J., Garg, V. K., Barzilay, R., and Jaakkola, T. Generative models for graph-based protein design. 2019.
- Jacobson, M. P., Friesner, R. A., Xiang, Z., and Honig, B. On the role of the crystal environment in determining protein side-chain conformations. *Journal of molecular biology*, 320(3):597–608, 2002.
- Jankauskaitė, J., Jiménez-García, B., Dapkūnas, J., Fernández-Recio, J., and Moal, I. H. Skempi 2.0: an updated benchmark of changes in protein–protein binding energy, kinetics and thermodynamics upon mutation. *Bioinformatics*, 35(3):462–469, 2019.
- Jelinek, F., Mercer, R. L., Bahl, L. R., and Baker, J. K. Perplexity—a measure of the difficulty of speech recognition tasks. *The Journal of the Acoustical Society of America*, 62(S1):S63–S63, 1977.
- Ji, S., Xu, W., Yang, M., and Yu, K. 3d convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):221–231, 2012.
- Jing, B., Eismann, S., Suriana, P., Townshend, R. J., and Dror, R. Learning from protein structure with geometric vector perceptrons. *arXiv preprint arXiv:2009.01411*, 2020.
- Jing, B., Eismann, S., Soni, P. N., and Dror, R. O. Equivariant graph neural networks for 3d macromolecular structure. *arXiv preprint arXiv:2106.03843*, 2021.
- Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.
- Källberg, M., Wang, H., Wang, S., Peng, J., Wang, Z., Lu, H., and Xu, J. Template-based protein structure modeling using the raptorx web server. *Nature protocols*, 7(8):1511–1522, 2012.
- Karasikov, M., Pagès, G., and Grudin, S. Smooth orientation-dependent scoring function for coarse-grained protein quality assessment. *Bioinformatics*, 35(16):2801–2808, 2019.
- Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.
- Klicpera, J., Groß, J., and Günnemann, S. Directional message passing for molecular graphs. *arXiv preprint arXiv:2003.03123*, 2020.
- Köhler, J., Klein, L., and Noé, F. Equivariant flows: exact likelihood generative learning for symmetric densities. In *International Conference on Machine Learning*, pp. 5361–5370. PMLR, 2020.
- Kwon, S., Won, J., Kryshtafovych, A., and Seok, C. Assessment of protein model structure accuracy estimation in casp14: Old and new challenges. *Proteins: Structure, Function, and Bioinformatics*, 2021.
- Lefèvre, F., Rémy, M.-H., and Masson, J.-M. Alanine-stretch scanning mutagenesis: a simple and efficient method to probe protein structure and function. *Nucleic acids research*, 25(2):447–448, 1997.
- Li, B., Yang, Y. T., Capra, J. A., and Gerstein, M. B. Predicting changes in protein thermodynamic stability upon point mutation with deep 3d convolutional neural networks. *bioRxiv*, 2020. doi: 10.1101/2020.02.28.959874.
- Li, S., Zhou, J., Xu, T., Huang, L., Wang, F., Xiong, H., Huang, W., Dou, D., and Xiong, H. 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*, pp. 975–985, 2021.
- Li, Z., Yang, Y., Faraggi, E., Zhan, J., and Zhou, Y. Direct prediction of profiles of sequences compatible with a protein structure by neural networks with fragment-based local and energy-based nonlocal profiles. *Proteins: Structure, Function, and Bioinformatics*, 82(10):2565–2573, 2014.

- Lundström, J., Rychlewski, L., Bujnicki, J., and Elofsson, A. Pcons: A neural-network-based consensus predictor that improves fold recognition. *Protein science*, 10(11): 2354–2362, 2001.
- Mariani, V., Biasini, M., Barbato, A., and Schwede, T. lddt: a local superposition-free score for comparing protein structures and models using distance difference tests. *Bioinformatics*, 29(21):2722–2728, 2013.
- Misiura, M., Shroff, R., Thyer, R., and Kolomeisky, A. Dlpacker: Deep learning for prediction of amino acid side chain conformations in proteins. *bioRxiv*, 2021.
- Moult, J., Fidelis, K., Kryshtafovych, A., Schwede, T., and Tramontano, A. Critical assessment of methods of protein structure prediction (caspp)—round x. *Proteins: Structure, Function, and Bioinformatics*, 82:1–6, 2014.
- Murzin, A. G., Brenner, S. E., Hubbard, T., and Chothia, C. Scop: a structural classification of proteins database for the investigation of sequences and structures. *Journal of molecular biology*, 247(4):536–540, 1995.
- Nair, V. and Hinton, G. E. Rectified linear units improve restricted boltzmann machines. In *ICML*, 2010.
- Nelson, D. L., Lehninger, A. L., and Cox, M. M. *Lehninger principles of biochemistry*. Macmillan, 2008.
- O’Connell, J., Li, Z., Hanson, J., Heffernan, R., Lyons, J., Paliwal, K., Dehzangi, A., Yang, Y., and Zhou, Y. Spin2: Predicting sequence profiles from protein structures using deep neural networks. *Proteins: Structure, Function, and Bioinformatics*, 86(6):629–633, 2018.
- Olechnovič, K. and Venclovas, Č. Voromqa: Assessment of protein structure quality using interatomic contact areas. *Proteins: Structure, Function, and Bioinformatics*, 85(6): 1131–1145, 2017.
- Olechnovič, K., Kulberkytė, E., and Venclovas, Č. Cad-score: a new contact area difference-based function for evaluation of protein structural models. *Proteins: Structure, Function, and Bioinformatics*, 81(1):149–162, 2013.
- Orengo, C. A., Michie, A. D., Jones, S., Jones, D. T., Swindells, M. B., and Thornton, J. M. Cath—a hierarchical classification of protein domain structures. *Structure*, 5(8):1093–1109, 1997.
- Pagès, G., Charmettant, B., and Grudin, S. Protein model quality assessment using 3d oriented convolutional neural networks. *Bioinformatics*, 35(18):3313–3319, 2019.
- Qi, Y. and Zhang, J. Z. Densecpd: improving the accuracy of neural-network-based computational protein sequence design with densenet. *Journal of chemical information and modeling*, 60(3):1245–1252, 2020.
- Rao, R., Bhattacharya, N., Thomas, N., Duan, Y., Chen, X., Canny, J., Abbeel, P., and Song, Y. S. Evaluating protein transfer learning with tape. *Advances in neural information processing systems*, 32:9689, 2019.
- Rohl, C. A., Strauss, C. E., Misura, K. M., and Baker, D. Protein structure prediction using rosetta. *Methods in enzymology*, 383:66–93, 2004.
- Satorras, V. G., Hoogetboom, E., and Welling, M. E(n) equivariant graph neural networks. *arXiv preprint arXiv:2102.09844*, 2021.
- Schaap, M. G., Leij, F. J., and Van Genuchten, M. T. Rosetta: A computer program for estimating soil hydraulic parameters with hierarchical pedotransfer functions. *Journal of hydrology*, 251(3-4):163–176, 2001.
- Schütt, K. T., Unke, O. T., and Gastegger, M. Equivariant message passing for the prediction of tensorial properties and molecular spectra. *arXiv preprint arXiv:2102.03150*, 2021.
- Shulman-Peleg, A., Nussinov, R., and Wolfson, H. J. Recognition of functional sites in protein structures. *Journal of molecular biology*, 339(3):607–633, 2004.
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.
- Strodthoff, N., Wagner, P., Wenzel, M., and Samek, W. Udsmpot: universal deep sequence models for protein classification. *Bioinformatics*, 36(8):2401–2409, 2020.
- Strokach, A., Becerra, D., Corbi-Verge, C., Perez-Riba, A., and Kim, P. M. Fast and flexible protein design using deep graph neural networks. *Cell Systems*, 11(4):402–411, 2020.
- Sverrisson, F., Feydy, J., Correia, B. E., and Bronstein, M. M. Fast end-to-end learning on protein surfaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15272–15281, 2021.
- Thomas, N., Smidt, T., Kearnes, S., Yang, L., Li, L., Kohlhoff, K., and Riley, P. Tensor field networks: Rotation-and translation-equivariant neural networks for 3d point clouds. *arXiv preprint arXiv:1802.08219*, 2018.
- Torng, W. and Altman, R. B. 3d deep convolutional neural networks for amino acid environment similarity analysis. *BMC bioinformatics*, 18(1):1–23, 2017.
- Townshend, R., Bedi, R., Suriana, P., and Dror, R. End-to-end learning on 3d protein structure for interface prediction. *Advances in Neural Information Processing Systems*, 32:15642–15651, 2019.

- Townshend, R. J., Vögele, M., Suriana, P., Derry, A., Powers, A., Laloudakis, Y., Balachandar, S., Anderson, B., Eismann, S., Kondor, R., et al. Atom3d: Tasks on molecules in three dimensions. *arXiv preprint arXiv:2012.04035*, 2020.
- Uziela, K., Menendez Hurtado, D., Shu, N., Wallner, B., and Elofsson, A. Proq3d: improved model quality assessments using deep learning. *Bioinformatics*, 33(10): 1578–1580, 2017.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008, 2017.
- Vecchio, A., Deac, A., Liò, P., and Veličković, P. Neural message passing for joint paratope-epitope prediction. 2021.
- Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., and Bengio, Y. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.
- Wang, J., Cao, H., Zhang, J. Z., and Qi, Y. Computational protein design with deep learning neural networks. *Scientific reports*, 8(1):1–9, 2018.
- Wang, X., Flannery, S. T., and Kihara, D. Protein docking model evaluation by graph neural networks. *Frontiers in Molecular Biosciences*, 8:402, 2021.
- Webb, E. C. et al. *Enzyme nomenclature 1992. Recommendations of the Nomenclature Committee of the International Union of Biochemistry and Molecular Biology on the Nomenclature and Classification of Enzymes*. Number Ed. 6. Academic Press, 1992.
- Weiler, M., Geiger, M., Welling, M., Boomsma, W., and Cohen, T. 3d steerable cnns: Learning rotationally equivariant features in volumetric data. *arXiv preprint arXiv:1807.02547*, 2018.
- Xu, K., Hu, W., Leskovec, J., and Jegelka, S. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018.
- Ying, R., You, J., Morris, C., Ren, X., Hamilton, W. L., and Leskovec, J. Hierarchical graph representation learning with differentiable pooling. *arXiv preprint arXiv:1806.08804*, 2018.
- Zemla, A. Lga: a method for finding 3d similarities in protein structures. *Nucleic acids research*, 31(13):3370–3374, 2003.
- Zhang, J. and Zhang, Y. A novel side-chain orientation dependent potential derived from random-walk reference state for protein fold selection and structure prediction. *PloS one*, 5(10):e15386, 2010.
- Zhang, Y. and Skolnick, J. Scoring function for automated assessment of protein structure template quality. *Proteins: Structure, Function, and Bioinformatics*, 57(4):702–710, 2004.
- Zhang, Y., Chen, Y., Wang, C., Lo, C.-C., Liu, X., Wu, W., and Zhang, J. Prodcnn: Protein design using a convolutional neural network. *Proteins: Structure, Function, and Bioinformatics*, 88(7):819–829, 2020.
- Zhemchuzhnikov, D., Igashov, I., and Grudin, S. 6dcnn with roto-translational convolution filters for volumetric data processing. *arXiv preprint arXiv:2107.12078*, 2021.



## A. Proof of Rotation Equivariance

A function  $f$  taking a 3D vector  $x \in \mathbb{R}^3$  as input is rotation equivariance, if applying ant rotation matrix  $R \in \mathbb{R}^{3 \times 3}$  on  $x$  leads to the same transformations of the output  $f(x)$ . Formally,  $f : \mathbb{R}^3 \rightarrow \mathbb{R}^3$  is rotation equivariance by fulfilling:

$$R(f(x)) = f(Rx) \quad (30)$$

For notation consistency, we consider each row of the matrix is an individual 3D vector and use right-hand matrix multiplication here. When performing equivariant message passing on protein graph for node  $i$  with local rotation matrix  $O_i$ , we first transform the vector representations of its neighbors from global reference to its local reference, that is, for a particular neighbor node  $j$  with vector feature  $V_j \in \mathbb{R}^{C \times 3}$ , apply our DWNN layers  $f$ , and transform the updated features back to the global reference. We set only one neighbor node  $u_j$  for  $u_i$  for simplicity. The updated vector representation  $V_i$  for node  $i$  is

$$V_i = [f(V_j O_i^T)] O_i \quad (31)$$

If we apply a global rotation matrix  $R$  to all vectors in the global frame, the local rotation matrix  $O_i$  will be transformed to  $O'_i = O_i R$ , and  $V_j$  to be  $V'_j = V_j R$ , now the output  $V'_i$  is

$$V'_i = [f(V'_j O_i'^T)] O'_i \quad (32)$$

$$= [f(V_j R R^T O_i^T)] O_i R \quad (33)$$

$$= [f(V_j O_i^T)] O_i R \quad (34)$$

And if instead, we directly apply the rotation matrix to original output  $V_i$ , we got

$$V'_i = [f(V_j O_i^T)] O_i R \quad (35)$$

$$= V_i R \quad (36)$$

In other words, rotating the input leads to the same transformations to output, so we can preserve rotation equivariance for vector representations in this way. Note that scalar features remain invariant when applying global rotation, so our message-passing paradigm with global and local coordinate transformations is rotation equivariance.

## B. Experiment Details

We represent a protein 3D structure as an attributed graph, with each node and edge attached with scalar and vector features that are geometric-aware. We implement our DWNN in the equivariant message passing manner, with 3 layer Directed Weight Perceptrons for all tasks.

### B.1. Protein Features

In this paper, we use an attributed graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  to represent the protein structure, where each node corresponds to

one particular amino acid in the protein with edges connecting its  $k$ -nearest neighbors. Here we set  $k = 30$ . The node features  $\mathcal{V} = \{v_1, \dots, v_N\}$  and edge features  $\mathcal{E} = \{e_{ij}\}_{i \neq j}$  are both multi-channel scalar-vector tuples with scalar features like distances and dihedral angles and vector features like unit vectors representing particular orientations.

A node  $v_i$  represent the  $i$ -th residue in the protein with scalar and vector features describing its geometric and chemical properties if available. Therefore, a node in this graph may have multi-channel scalar-vector tuples  $(s_i, V_i)$ ,  $s_i \in \mathbb{R}^6$  or  $\mathbb{R}^{26}$ ,  $V_i \in \mathbb{R}^{3 \times 3}$  as its initial features.

- **Scalar Feature.** The  $\{\sin, \cos\} \circ \{\psi, \omega, \phi\}$ . Here  $\{\psi, \omega, \phi\}$  are dihedral angles computed from its four backbone atom positions,  $C\alpha_{i-1}, N_i, C\alpha_i, N_{i+1}$ .
- **Scalar Feature.** A one-hot representation of residue if the identity is available.
- **Vector Feature.** The unit vectors in the directions of  $C\alpha_{i+1} - C\alpha_i$  and  $C\alpha_{i-1} - C\alpha_i$ .
- **Vector Feature.** The unit vector in the direction of  $C\beta_i - C\alpha_i$  corresponds to the side-chain directions.

The edge  $e_{ij}$  connecting the  $i$ -th residue and the  $j$ -th residue also has multi-channel scalar-vector tuples as its feature  $(s_{ij}, V_{ij})$ ,  $s_{ij} \in \mathbb{R}^{34}$ ,  $V_{ij} \in \mathbb{R}^{1 \times 3}$

- **Scalar Feature.** The encoding of the distance  $\|C\alpha_j - C\alpha_i\|$  using 16 Gaussian radial basis functions with centers spaced between 0 to 20 angstroms.
- **Scalar Feature.** The positional encoding of  $j - i$  corresponding the relative position in the protein sequence.
- **Scalar Feature:** The contact signal describes if the two residues contact in the space, 1 if  $\|C\alpha_j - C\alpha_i\| \leq 8$  and 0 otherwise.
- **Scalar Feature.** The H-bond signal describes if there may be a H-bond between the two nodes calculated by backbone distance.
- **Vector Feature.** The unit vector in the direction of  $C\alpha_j - C\alpha_i$ .

### B.2. Model Details

**Synthetic Task.** We consider the length of each vector as one channel scalar feature  $\in \mathbb{R}$  and the vector itself as one channel vector feature  $\in \mathbb{R}^{1 \times 3}$ . The VNN and GVP models in this experiment are be set to 3 layers. And our DWP is only 1 layer. We also train a 3-layer MLP by concatenating the scalar and vector feature as a four-channel scalar feature

$\in \mathbb{R}^4$  as input. Specifically, the parameter counts of VNN, GVP, MLP, DWP is 24,28,24 and 24.

**CPD.** We train our model in a BERT-style recovery target (Strokach et al., 2020), more specifically, we dynamically mask 85% residues of the whole protein and let the model recover them using structure information from its neighbors using CrossEntropy loss. During testing, we mask all the residues and recover the whole protein sequence in one forward pass, we also try to iteratively predict one residue per forwarding pass.

**MQA.** Because MQA is a regression task, we train our model using MSE loss. When testing, we compute the different types of correlations between predicted scores and ground truth.

**RES.** Residue identity requires the model to classify the center residue from a local substructure of the protein. We construct a subgraph of each local substructure according to Section B.1. To guarantee equitable comparison, we retrain all the methods based on our protein graph from scratch.

**Fold and React.** For these two graph-level classification tasks, we construct graph for each protein as above and use CrossEntropy loss.

### B.3. Training Details

For the synthetic task, we train each model for 1000 epochs with a learning rate of  $1e-3$  and plot the training loss for evaluation.

For all models we trained, we use 128 scalar channels and 32 vector channels for each node’s hidden representations, 64 scalar channels, and 16 vector channels for each edge’s hidden representations. We train our models with learning rate  $3e-4$  for CPD and  $2e-4$  for MQA,  $5e-4$  for RES and  $1e-3$  for REACT/FOLD, a dropout rate of 10% (Srivastava et al., 2014), and Layernorm paradigm. We train all the models on NVIDIA Tesla V100 for 100 epochs for each task.

## C. Ablation Studies

In the end, we conduct ablation experiments to study the contribution of different modules of our model on the tasks of CPD (table 6) and RES (table 7). More specifically, we replace the Directed Linear Module with the regular linear layer with scalar weights (No Directed Linear), remove the Interaction Module (No Interaction), and also break the equivariance by canceling coordinate transformations in the step of message passing (No Equivariance). As shown in Table 6, in general, all three components of our model are important since removing each of them results in a significant performance drop. According to the performance gain, the directed weights are the most important module for the CPD task. For Residue Identity, however, equivariance mes-

sage passing is the most important component, indicating that the necessity of rotation equivariance in learning the representation of local protein structures.

Table 6. Performance of different ablated CPD models on the CATH 4.2 and TS50 dataset.

Model	Perplexity ↓			Recovery % ↑			
	Short	Single-chain	All	Short	Single-chain	All	TS50
No DirectedLinear	6.02	6.67	4.85	36.7	35.0	42.9	46.8
No Interaction	5.87	6.36	4.27	37.5	36.3	44.9	48.7
No Equivariance	5.47	6.04	4.13	38.1	37.1	45.8	49.5
DW-GNN	<b>5.42</b>	<b>5.69</b>	<b>3.94</b>	<b>39.6</b>	<b>38.5</b>	<b>47.5</b>	<b>54.5</b>

Table 7. Performance of different ablated RES models on the ATOM3D dataset.

Model	No DirectedLinear	No Interaction	No Equivariance	DW-GNN
Acc %	47.3	47.7	33.0	<b>50.2</b>