Clustering for Protein Representation Learning

Ruijie Quan, Wenguan Wang*, Fan Ma, Hehe Fan, Yi Yang
ReLER, CCAI, Zhejiang University
https://github.com/QUANRJ/ClusteringPRL

Abstract

Protein representation learning is a challenging task that aims to capture the structure and function of proteins from their amino acid sequences. Previous methods largely ignored the fact that not all amino acids are equally important for protein folding and activity. In this article, we propose a neural clustering framework that can automatically discover the critical components of a protein by considering both its primary and tertiary structure information. Our framework treats a protein as a graph, where each node represents an amino acid and each edge represents a spatial or sequential connection between amino acids. We then apply an iterative clustering strategy to group the nodes into clusters based on their 1D and 3D positions and assign scores to each cluster. We select the highest-scoring clusters and use their medoid nodes for the next iteration of clustering, until we obtain a hierarchical and informative representation of the protein. We evaluate on four protein-related tasks: protein fold classification, enzyme reaction classification, gene ontology term prediction, and enzyme commission number prediction. Experimental results demonstrate that our method achieves state-of-the-art performance.

1. Introduction

Proteins are one of the most fundamental elements in living organisms and make significant contributions to nearly all fundamental biological processes in the cell. Composed of one or several chains of amino acids [63, 70], proteins fold into specific conformations to enable various biological functionalities [31]. A multi-level structure of proteins begins with the *primary structure*, which is defined by the sequence of amino acids forming the protein backbone [69]. The *secondary structure* is determined by hydrogen bonds between distant amino acids in the chain, resulting in substructures such as α-helices and β-sheets [75]. *Tertiary structure* arises from folding of secondary structures, determined by interactions between side chains of amino acids [9]. Lastly, the *quarternary structure* describes the arrangement of polypeptide chains in a multi-subunit arrangement [13]. Understanding the structure and function of proteins [22, 23, 30, 38, 80, 81, 94] is crucial in elucidating their role in biological processes and developing new therapies and treatments for a variety of diseases.

While a protein's conformation and function are primarily determined by its amino acid sequence, it is important to recognize that not all amino acids contribute equally to these aspects. In fact, certain amino acids, known as the *critical components*, play the primary role in determining a protein's shape and function [8, 18, 26, 41, 60, 61, 86]. Sometimes, even a single amino acid substitution can significantly impact a protein's overall structure and function [41, 60]. For example, sickle cell anemia results from a single amino acid change in hemoglobin, causing it to form abnormal fibers that distort red blood cell shape. Besides, the critical components of a protein's primary structure are essential for its biological activity. For instance, any of the first 24 amino acids of the adrenocorticotropic hormone (ACTH) molecule is necessary for its biological activity, whereas removing all amino acids between 25-39 has no impact [61, 86]. Also, proteins with the same critical components perform the same function, *e.g.*, the insulin A and B chains in various mammals contain 24 invariant amino acid residues necessary for insulin function, while differences in the remaining amino acid residues do not impact insulin function [8, 26]. In addition, proteins from the same family often have long stretches of similar amino acid sequences within their primary structure [18], suggesting that only a small portion of amino acids that differentiate these proteins are the critical components.

Motivated by the fact that certain amino acids play a more critical role in determining a protein's structure and function than the others [8, 18, 26, 41, 60, 61, 86], we devise a neural clustering framework for protein representation learning. Concretely, it progressively groups amino acids so as to find the most representative ones for protein classification. During each iteration, our algorithm proceeds three steps: spherical cluster initialization (SCI), cluster representation extraction (CRE), and cluster nomination

*Corresponding author: *Wenguan Wang*.

(CN). The iterative procedure starts by treating a protein as a graph where each node represents an amino acid, and each edge represents a spatial or sequential connection between amino acids. In SCI step (△), all the nodes are grouped into clusters based on their sequential and spatial positions. Subsequently, in CRE step (□), the information of neighboring nodes within the same cluster are aggregated into a single representative node, namely medoid node. This step effectively creates an informative and compact representation for each cluster. Lastly, in CN step (▽), a graph convolutional network (GCN) [72] is applied to score all the clusters, and a few top-scoring ones are selected and their medoid nodes are used as the input for the next iteration. By iterating these steps ↻(△□▽), we explore the protein's structure and discover the representative amino acids for protein representation, leading to a powerful, neural clustering based protein representation learning framework.

By embracing the powerful idea of clustering, our approach favorably outperforms advanced competitors. We observe notable improvements of **5.6**% and **2.9**% in $F_{\max}$ for enzyme commission number prediction [30] (§4.1) and gene ontology term prediction [30] (§4.2). Our method also yields remarkable enhancements of **3.3**% and **1.1**% in classification accuracy for protein fold classification [39] (§4.3) and enzyme reaction classification [38] (§4.4). We also provide comprehensive diagnostic analyses (§ 4.5) and visual results (§ 4.6), verifying the efficacy of our essential algorithm designs, showing strong empirical evidence for our core motivation, and confirming the capability of our algorithm in identifying functional motifs of proteins.

2. Related Work

Protein Representation Learning. It has been a topic of interest in the field of bioinformatics and computational biology in recent years. Existing methods for this topic can be broadly categorized into two types: *sequence-based* and *structure-based*. Early works on sequence-based protein representation learning typically apply word embedding algorithms [5, 89] and 1D convolutional neural networks [39, 48, 49, 77]. Though straightforward, these methods neglect the spatial information in protein structures. To address this limitation, structure-based methods explore the use of 3D convolutional neural networks [4, 15, 76] and GCNs [7, 30, 33, 44, 45, 81, 94] for this task. Recently, some approaches [44, 81] focus on atom-level representations, treating each atom as a node. The state-of-the-art performance has been achieved by learning at the amino acid-level [22, 94], indicating that protein representation is more closely related to amino acids rather than individual atoms.

Despite significant progress made by these existing methods, they treat all amino acids equally. In sharp contrast, we propose to learn the protein representation by a neural clustering framework. This allows us to capture the inherent variability and significance of different amino acids, leading to a more comprehensive and accurate representation of proteins. We are of the opinion that our method has several potential applications and extensions in the field of protein science. For instance, our neural clustering approach can benefit protein design by utilizing the learned crucial components to direct the design of novel protein sequences [10, 73] that possess specific properties or functions. This, in turn, can facilitate the development of new therapies and treatments for a diverse range of illnesses.

Clustering. Clustering is a fundamental data analysis task that aims to group similar samples together based on similarity, density, intervals or particular statistical distribution measures of the data space [34, 43, 59]. It helps to identify representative patterns in data which is meaningful for exploratory data analysis. Traditional clustering methods [54, 67] heavily rely on the original data representations. As a result, they often prove to be ineffective when confronted with data residing in high-dimensional spaces, such as images and text documents. Recently, deep learning-based clustering methods have attracted increased attention, and been successfully applied to various real-world applications, such as image segmentation [17, 51, 52, 82, 87, 95], unsupervised representation learning [11, 12, 24, 25, 55, 56, 65, 83, 90, 92], financial analysis [14, 32, 42], and text analysis [1, 2, 74].

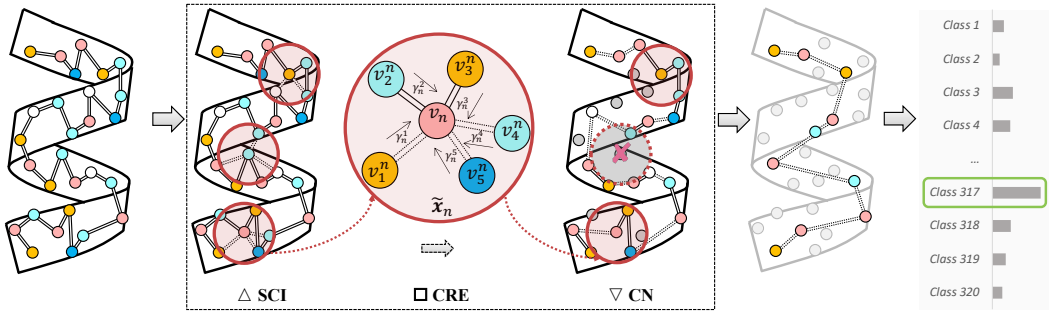
Drawing inspiration from the biological fact that the significance of various amino acids varies, we propose a neural clustering framework for end-to-end protein representation learning. Our objective is to leverage the inherent benefits of clustering to identify the representative amino acids. In experiments, we demonstrate the feasibility of our clustering-based method through numerical evaluations and provide visual evidence to reinforce the underlying motivation behind our approach.

3. Methodology

We have stated that the structure and function of a protein are represented by certain critical amino acids in § 1. This motivates us to regard the protein representation learning task as an amino acid clustering process, so as to automatically explore the critical components of the protein. Firstly, some basic concepts used throughout the paper and the task setup are introduced in § 3.1. Then, we elaborate on the neural clustering framework in § 3.2. Finally, § 3.3 presents our implementation details.

3.1. Notation and Task Setup

A protein is denoted as a triplet $\mathcal{P} = (\mathcal{V}, \mathcal{E}, \mathcal{Y})$, where $\mathcal{V} = \{v_1, \cdots, v_N\}$ is the set of nodes representing N amino acids, $\mathcal{E}$ the set of edges representing spatial or sequential connections between amino acids, and $\mathcal{Y}$ the set of labels. The target goal of protein classification is to learn a map-



(a) Input Protein (b) Iterative Clustering (c) Output Amino Acids (d) Classification

Figure 1. Overview of our iterative neural clustering pipeline for protein representation learning: (a) input protein with amino acids, (b) iterative clustering algorithm which repeatedly stacks three steps ↻(△□▽), (c) output can be seen as the critical amino acids of the protein, (d) output amino acids used for classification. The details of our iterative neural clustering method can be seen in §3.2.

ping $\mathcal{V} \rightarrow \mathcal{Y}$. Specifically, in single-label classification, *e.g.*, protein fold classification and enzyme reaction classification, the focus is on learning from a collection of examples that are linked to a *single* label from $\mathcal{Y}$. While in multi-class classification, *e.g.*, enzyme commission number prediction and gene ontology term prediction, each examples are associated with *multiple* labels from $\mathcal{Y}$. In what follows, we use $\{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N\}$ to denote the features of $\mathcal{V}$, where $\boldsymbol{x}_n \in \mathbb{R}^{256}$ is the feature vector of amino acid v_n. The feature vector can be derived from various sources, such as the one-hot encoding of amino acid types, the orientations of amino acids, and the sequential and spatial positions of amino acids. We use $A \in \{0,1\}^{N \times N}$ to denote the adjacency matrix of $\mathcal{V}$, where $A_{n,m} = 1$ if there exists an edge between amino acid nodes v_n and v_m, *i.e.*, $e_{n,m} \in \mathcal{E}$.

3.2. Iterative Clustering

In our neural clustering framework, we perform iterative clustering on amino acids of the input protein. Each iteration encompasses three steps, spherical Cluster Initialization (SCI), Cluster Representation Extraction (CRE), and Cluster Nomination (CN), as illustrated in Figure 1.

Spherical Cluster Initialization (SCI). For each amino acid, we initialize a cluster by considering other locally neighboring amino acids within a fixed receptive field, which draws inspiration from previous work in 3D geometry [64]. This approach enables the examination and comprehension of the spatial and sequential associations among amino acids, which hold paramount significance in determining the structure and functioning of proteins [22]. Specifically, for each amino acid v_n, we define a cluster as a set of amino acids within a fixed radius r, where v_n is regarded as the *medoid* node of the cluster. The fixed radius r is a hyperparameter which determines the extent to which the local amino acid nodes are considered for cluster initialization, and its impact on the final performance of protein analysis is studied in §4.5. For v_n, we denote the set of its neighbor amino acid nodes as $\mathcal{H}^n = \{v_1^n, \cdots, v_K^n\}$. Note that $v_n \in \mathcal{H}^n$. In the first iteration ($t = 1$), we conduct SCI process based on all the input N amino acids to form the clusters. In subsequent iterations ($t > 1$), we use the nominated N_{t-1} amino acids from the previous $t-1$-th iteration to initialize the clusters. This allows to focus on exploring the critical components of the protein graph and avoid redundantly exploring the same areas. The adjacency matrix A is regenerated in each SCI process with considering the connectivity among amino acids.

Cluster Representation Extraction (CRE). The second step aims to learn the overall representation of each cluster $\mathcal{H}^n$ by considering all the amino acids within it. Specifically, we construct the feature representation $\boldsymbol{x}_k^n$ of the amino acid node v_k^n in the cluster $\mathcal{H}^n$ by:

$$\boldsymbol{x}_k^n = f(\boldsymbol{g}_k^n, \boldsymbol{o}_k^n, d_k^n, s_k, \boldsymbol{e}_k), \tag{1}$$

where $\boldsymbol{g}_k^n = (\boldsymbol{z}_k - \boldsymbol{z}_n) \in \mathbb{R}^3$ denotes relative geometric coordinates, and $\boldsymbol{z}$ indicates the spatial coordinate; $\boldsymbol{o}_k^n \in \mathbb{R}^3$ is the 3D orientation vector; $d_k^n = (||\boldsymbol{z}_k - \boldsymbol{z}_n||_2) \in \mathbb{R}^3$ indicates the spatial distance between v_n and v_k^n; s_k is the sequential order on the amino acid chain that is relative to the beginning of the protein sequence; $\boldsymbol{e}_k \in \mathbb{R}^{128}$ denotes amino acid embedding (*i.e.*, one-hot encoding of the amino acid type) of v_k^n in the cluster; and f denotes an encoder which is implemented by a multilayer perceptron. In this way, our neural clustering framework considers both primary (1D sequence-based distance) and tertiary (3D coordinates and orientations) structure information of proteins. Then, we utilize the cross-attention mechanism [78] to calculate the attention score $\gamma_k^n \in (0,1)$ between the medoid amino acid node feature $\boldsymbol{x}_n$ and all the constituent ones $\boldsymbol{x}_k^n$ in the cluster $\mathcal{H}^n$:

$$\gamma_k^n = \frac{\exp(\boldsymbol{w}[\boldsymbol{x}_n, \boldsymbol{x}_k^n])}{\sum_{k=1}^K \exp(\boldsymbol{w}[\boldsymbol{x}_n, \boldsymbol{x}_k^n])}, \tag{2}$$

where $\boldsymbol{w}$ is a learnable vector, and $[\cdot]$ refers to the concatenation operation. The attention score γ_k^n denotes the level of focus of the medoid on other constituent amino acids. Then, the overall cluster representation $\tilde{\boldsymbol{x}}_n$ for the cluster $\mathcal{H}^n$ of node v_n is given as:

$$\tilde{\boldsymbol{x}}_n = \sum\nolimits_{k=1}^K \gamma_k^n \boldsymbol{x}_k^n. \tag{3}$$

Cluster Nomination (CN). To identify the most representative amino acids (*i.e.*, critical components) in the protein,

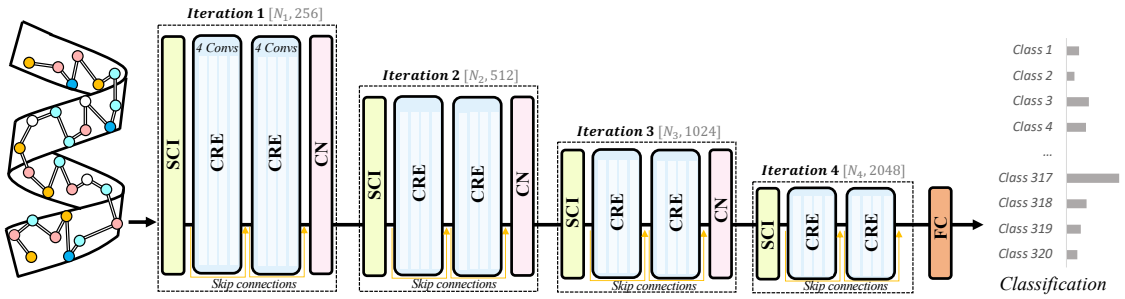


Figure 2. Our neural clustering framework architecture with four iterations. Given a protein, a set of 1D and 3D amino acids, our method adopts an iterative clustering algorithm to explore the most representative amino acids. At each iteration, B cluster representation extraction blocks are utilized to extract cluster features. The clustering nomination operation selects the fraction ω of amino acids for the next iteration, that $N_t = \lfloor \omega \cdot N_{t-1} \rfloor$. Details of the framework can be seen in §3.3.

we propose a cluster nomination process that learns to automatically select these critical components based on the cluster representations $\{\tilde{\boldsymbol{x}}_n\}_{n=1}^{N}$. Specifically, such cluster nomination process is achieved by a GCN, which takes each cluster representation $\tilde{\boldsymbol{x}}_n$ as input and calculates a *nomination* score $\varphi_n \in (0, 1)$:

$$\varphi_n = \sigma(\boldsymbol{W}_1\tilde{\boldsymbol{x}}_n + \sum\nolimits_{m=1}^{N_t} A_{n,m}(\boldsymbol{W}_2\tilde{\boldsymbol{x}}_n - \boldsymbol{W}_3\tilde{\boldsymbol{x}}_m)), \quad (4)$$

where $\boldsymbol{W}_{1,2,3}$ are learnable parameters and σ is ReLU function. By utilizing self-loops and the capability to learn functions of local extremas, we are able to score clusters based on both their global and local significance. The cluster feature is then weighted by the calculated scores:

$$\hat{X}^c = \Phi \odot X^c, \quad (5)$$

where $\Phi = [\varphi_1, \cdots, \varphi_{N_t}]^\top$, $X^c = [\tilde{\boldsymbol{x}}_1, \cdots, \tilde{\boldsymbol{x}}_{N_t}]$, and $\odot$ is broadcasted hadamard product. Based on the weighted cluster features $\hat{X}^c$ and the calculated nomination scores Φ, we select top-N_t clusters at the t-th iteration. Then, the top-N_t amino acids are utilized to form a new graph, viewed as the input for the next t+1-th iteration. Here N_t is the number of selected clusters at t-th iteration and is determined by the cluster nomination fraction ω, denoted as $N_t = \lfloor \omega \cdot N_{t-1} \rfloor$, that $t \in \{1, ..., T\}$ and $N_0 = N$. We will illustrate the impact of different values of ω in §4.5. Once the clusters are nominated in the CN phase, the medoids are then linked following the original sequential orders of the protein chain.

At each clustering iteration, we estimate cluster membership and centers by considering the sequential and spatial information of amino acids. Thus it explicitly probes the structures of the portion and captures the complex relationships among amino acids. Then, the representative amino acids (*i.e.*, cluster medoids) are identified and only those most informative ones (which can be viewed as critical components) are selected for next-iteration clustering, and eventually used for functionality prediction. After T iterations, a fully connected layer is used to project the feature representations of the nominated N_T amino acids to a $|\mathcal{Y}|$-dimension score vector for classification. Note that the attention-based selection is implicitly learnt from the supervision signals of the protein classification task. In a nutshell, our method utilizes an iterative clustering algorithm that repeatedly stacks three steps: SCI, CRE, and CN.

3.3. Implementation Details

Network Architecture. We progressively nominate N_T from N amino acids of the protein by T iterations (see Figure 2). We empirically set $T = 4$, suggesting the neural clustering framework consists of four iterations. At t-th iteration, we stack $B = 2$ CRE blocks to learn the representation of the selected N_t amino acids. In this way, our method is also a hierarchical framework that downsamples amino acids as the iteration proceeds. In order to ensure a large enough receptive field at late itera- tions, we increase the value of the cluster radius r applied in SCI step as the number of iterations increases. Concretely, the radii for the four iterations are r, $2r$, $3r$, and $4r$, respectively. We adopt skip connection [35] per CRE block to facilitate information flow and ease network training. Inspired by [22, 37, 40], we adopt rotation invariance (detailed in the appendix).

Training Objective. We follow the common protocol in this field [22, 37, 85, 94] to append a fully connected neural network as the classification head at the tail of network. Softmax activation and cross-entropy loss are used for single-label classification, while Sigmoid function and binary cross-entropy loss for multi-label classification.

Reproducibility. We implement our method using PyTorch-Geometric library. For all our experiments, the training and testing are conducted on a single Nvidia RTX 3090 GPU with 24 GB memory.

4. Experiments

We evaluate our method on four tasks following previous studies [22, 38, 94]: enzyme commission (EC) number prediction (§4.1), gene ontology (GO) term prediction (§4.2), protein fold classification (§4.3), and enzyme reaction classification (§4.4). Then, in §4.5, we present a series of diagnostic studies. Finally, we provide a set of visual results in §4.6 for in-depth analysis.

Table 1. F_{max} on EC and GO prediction and Accuracy (%) on fold and reaction classification. [†] denotes results taken from [85] and [*] denotes results taken from [38] and [36] (§4.1-§4.4).

Method	Publication	EC	GO			Fold Classification				Reaction
			BP	MF	CC	Fold	Super.	Fam.	Avg.	
ResNet [66]	*NeurIPS 2019*	0.605	0.280	0.405	0.304	10.1	7.21	23.5	13.6	24.1
LSTM [66]	*NeurIPS 2019*	0.425	0.225	0.321	0.283	6.41	4.33	18.1	9.61	11.0
Transformer [66]	*NeurIPS 2019*	0.238	0.264	0.211	0.405	9.22	8.81	40.4	19.4	26.6
GCN [46]	*ICLR 2017*	0.320	0.252	0.195	0.329	16.8*	21.3*	82.8*	40.3*	67.3*
GAT [79]	*ICLR 2018*	0.368	0.284†	0.317†	0.385†	12.4	16.5	72.7	33.8	55.6
GVP [44]	*ICLR 2021*	0.489	0.326†	0.426†	0.420†	16.0	22.5	83.8	40.7	65.5
3DCNN [15]	*Bioinform 2018*	0.077	0.240	0.147	0.305	31.6*	45.4*	92.5*	56.5*	72.2*
GraphQA [7]	*Bioinform 2021*	0.509	0.308	0.329	0.413	23.7*	32.5*	84.4*	46.9*	60.8*
New IEConv [36]	*ICLR 2022*	0.735	0.374	0.544	0.444	47.6*	70.2*	99.2*	72.3*	87.2*
GearNet [94]	*ICLR 2023*	0.810	0.400	0.581	0.430	48.3	70.3	99.5	72.7	85.3
ProNet [81]	*ICLR 2023*	-	-	-	-	52.7	70.3	99.3	74.1	86.4
CDConv [22]	*ICLR 2023*	0.820	0.453	0.654	0.479	56.7	77.7	99.6	78.0	88.5
Ours	-	**0.866**	**0.474**	**0.675**	**0.483**	**63.1**	**81.2**	99.6	**81.3**	**89.6**

4.1. Experiment on EC Number Prediction

Task and Dataset. Enzyme Commission (EC) number prediction seeks to anticipate the EC numbers of diverse proteins that elucidate their role in catalyzing biochemical reactions. The EC numbers are chosen from the third and fourth levels of the EC tree, resulting in 538 distinct binary classification tasks. As in [30], the dataset of this task consists of 15,550/1,729/1,919 proteins in `train`/`val`/`test` set, respectively. For GO term and EC number prediction, we follow the multi-cutoff splits in [30] to ensure that the `test` set only contains PDB chains with a sequence identity of no more than 95% to the proteins in the `train` set.

Training Setup and Evaluation Metric. EC number prediction can be regarded as a multi-label classification task. The performance is evaluated by the protein-centric maximum F-score F_{max}, which is based on the precision and recall of the predictions for each protein.

Performance Comparison. We compare our neural clustering method with 11 top-leading methods in Table 1. As seen, our method establishes a new state-of-the-art on EC number prediction task. It surpasses CDConv [22] by **5.6**% (0.820→**0.866**) and GearNet [94] by **6.9**% (0.810→**0.866**) in terms of F_{max}. This indicates that our method can learn informative representations of proteins that reflect their functional roles in catalyzing biochemical reactions.

4.2. Experiment on GO Term Prediction

Task and Dataset. GO term prediction aims to forecast whether a protein belongs to certain GO terms. These terms categorize proteins into functional classes that are hierarchically related and organized into three sub-tasks [30]: molecular function (MF) term prediction consisting of 489 classes, biological process (BP) term prediction including 1,943 classes, cellular component (CC) term prediction with 320 classes. The dataset contains 29,898/3,322/3,415 proteins for `train`/`val`/`test`, respectively.

Training Setup and Evaluation Metric. GO term prediction is also a multi-label classification task. The protein-centric maximum F-score F_{max} is reported.

Performance Comparison. We compare our method with 12 existing state-of-the-art methods for protein representation learning on the task of predicting the GO term of proteins, where most of them are CNN or GNN-based methods. The results are shown in Table 1, where our framework achieves competitive F_{max} scores on all three sub-tasks, especially on MF (**0.675** *vs* 0.654) and BP (**0.474** *vs* 0.453) terms, compared to CDConv [22]. Also, our method is clearly ahead of the second-best method, GearNet [94], by large margins, *i.e.*, **0.474** *vs* 0.400 BP, **0.675** *vs* 0.581 MF, **0.483** *vs* 0.430 CC. Our new records across three sub-tasks show that our neural clustering method can learn rich representations of proteins that capture their functional diversity.

4.3. Experiment on Protein Fold Classification

Task and Dataset. Protein fold classification, firstly introduced in [39], aims to predict the fold class label of a protein. It contains three different evaluation scenarios: 1) Fold, where proteins be- longing to the same superfamily are excluded during training, 12,312/736/718 proteins for train/val/ test, 2) Superfamily, where proteins from the same family are not included during training, 12,312/736/1,254 proteins for `train`/`val`/`test`, 3) Family, where proteins from the same family are used during training, 12,312/736/1,272 proteins for `train`/`val`/`test`.

Training Setup and Evaluation Metric. Protein fold classification can be seen as a single-label classification task. Mean accuracy is used as the evaluation metric.

Performance Comparison. In Table 1, we continue to compare our framework with these state-of-the-art methods on the task of classifying proteins into different fold classes. The fold class describes the overall shape and topology of a protein. Our framework yields superior performance. For example, it yields superior results as compared to CDConv [22] by **6.4**%, ProNet [85] by **10.4**% and GearNet [94] by **14.8**% on the Fold evaluation scenario. Considering that protein fold classification is challenging,

Table 2. Ablative experiments for the neural clustering algorithm. (a) An off-the-shelf clustering algorithm; (b) A simple average pooling method; (c) Randomly generate attention score γ_k^n. See §4.5 for details.

Method	EC	GO			Fold Classification				Reaction
		BP	MF	CC	Fold	Super.	Fam.	Avg.	
(a)	0.792	0.385	0.579	0.429	43.1	67.1	99.1	69.8	86.8
(b)	0.817	0.452	0.641	0.453	57.2	78.7	99.3	78.4	88.1
(c)	0.765	0.342	0.567	0.415	44.6	69.5	99.2	71.1	86.4
Ours	**0.866**	**0.474**	**0.675**	**0.483**	**63.1**	**81.2**	**99.6**	**81.3**	**89.6**

Table 3. Efficiency comparison to SOTA competitors on enzyme reaction classification. See §4.5 for details

Method	Acc.	Runing Time
New IEConv [36]	87.2%	75.3 ms
GearNet [94]	85.3%	*OOM*
ProNet [81]	86.4%	27.5 ms
CDConV [22]	88.5%	10.5 ms
Ours	89.6%	10.9 ms

such improvements are particularly impressive. Across the board, our neural clustering framework surpasses all other methods of protein fold classification, demonstrating that our framework can learn robust representations of proteins that reflect their structural similarity.

4.4. Experiment on Enzyme Reaction Classification

Task and Dataset. Enzyme reaction classification endeavors to predict the enzyme-catalyzed reaction class of a protein, utilizing all four levels of the EC number to portray reaction class. We utilize the dataset processed by [38], which consists of 384 four-level EC classes and $29,215/2,562/5,651$ proteins for `train`/`val`/`test`, where proteins have less than 50% sequence similarity in-between splits.

Training Setup and Evaluation Metric. Enzyme reaction classification is regarded as a single-label classification task. We adopt Mean accuracy as the evaluation metric.

Performance Comparison. Table 1 presents comparison results of classifying proteins into different enzyme reaction classes. In terms of classification accuracy, our neural clustering framework outperforms the classic GCN-based method by a margin, *e.g.*, GCN [46] by **22.3**%, GAT [79] by **34**%, and GrahQA [7] by **28.8**%. In addition, it surpasses recent three competitors, *i.e.*, CDConv [22] (**+1.1**%), ProNet [85] (**+3.2**%), and GearNet [94] (**+4.3**%). In summary, the proposed neural clustering framework achieves outstanding performance against state-of-the-art methods, suggesting that our method learns informative representations of proteins that reflect their catalytic activity.

4.5. Diagnose Analysis

Neural Clustering. To demonstrate the effectiveness of neural clustering, we compare it against three baseline approaches that employ naive methods as replacements. Firstly, we use an off-the-shelf clustering algorithm, GRACLUS [16], as a baseline (a). Secondly, we replace it with a simple average pooling method used in CDConv [22] as a baseline (b). Lastly, we replace the attention score γ_k^n with a random value as a baseline (c). As shown in Table 2, our method significantly outperforms all three baselines. Specifically, it surpasses baseline (a) by an impressive **11.5**%, baseline (b) by **2.5**%, and baseline (c) by **10.2**%. The superior performance compared to baseline (a) highlights the importance of using a learnable clustering approach for effective representation learning. This demonstrates that our neural clustering is able to capture meaningful patterns and structures in the protein that are not captured by the off-the-shelf clustering algorithm. Furthermore, comparison with baseline (c) supports the notion that learned assignment is more effective than random assignment, suggesting that neural clustering can leverage the inherent structure and relationships of the protein to make informed assignments, leading to improved performance.

Efficiency. We conduct an investigation into the efficiency of our neural clustering-based framework, focusing on its running time for enzyme reaction classification. The mean running time per prediction was measured using a single Nvidia RTX 3090 GPU, and the results are summarized in Table 3. In particular, GearNet [94], a competing method known for its high complexity, cannot be trained using the same GPU due to its resource requirements. Notably, our method achieves state-of-the-art performance while maintaining a comparable running time to existing approaches, suggesting the efficiency of our method.

Initial Clustering Radius. The initial clustering radius r determines the cluster size formed in SCI step. A larger radius leads to more amino acids in each cluster, potentially capturing a greater amount of spatial structural information. However, this also introduces the risk of increased noise within the clusters. Conversely, a smaller radius results in fewer amino acids being included in each cluster, which can reduce noise but may also lead to the loss of some critical information. Therefore, we conducted experiments ranging from 2.0 to 7.0 and assessed the performance on protein fold classification and enzyme reaction classification. The experimental results, presented in Figure 3, indicate that the optimal performance is achieved when $r = 4.0$, suggesting a suitable balance between capturing sufficient structural information and mitigating the detrimental effects of noise.

Cluster Nomination Fraction. In CN step, a cluster nomination fraction ω determines the proportion of clusters selected as the medoid nodes for the next iteration. A larger ω means that more clusters are retained, which may preserve more information but also increase redundancy and complexity. While a smaller ω means that fewer clusters are retained, which may reduce redundancy and complexity but also lose some information. We experiment with different values of ω from 20% to 70% and report results on protein fold classification and enzyme reaction classification. As

Table 4. Analysis of a different number of iterations. See details in §4.5.

T	EC	GO			Fold Classification				Reaction
		BP	MF	CC	Fold	Super.	Fam.	Avg.	
1	0.717	0.402	0.593	0.432	55.7	73.2	97.4	75.4	84.7
2	0.824	0.438	0.642	0.453	60.0	79.2	98.0	79.1	88.1
3	0.855	0.469	0.677	0.480	62.2	80.8	99.3	80.8	89.0
4	**0.866**	**0.474**	**0.675**	**0.483**	**63.1**	**81.2**	**99.6**	**81.3**	**89.6**
5	0.809	0.423	0.605	0.455	58.1	75.7	98.5	77.4	86.3

Table 5. $u\%$ missing coordinates (§4.5).

$u\%$	Fold	Super.	Fam.
0%	**63.1**	**81.2**	**99.6**
5%	61.9	79.8	99.5
10%	60.1	78.7	99.5
20%	56.7	76.9	99.3
30%	50.2	73.6	99.2
40%	47.8	71.3	99.0

Table 6. Comparison results with existing protein language models. See details in §4.5.

Method	Pretraining Dataset		EC	GO			Fold Classification				Reaction
				BP	MF	CC	Fold	Super.	Fam.	Avg.	
DeepFRI [62]	Pfam	10M	0.631	0.399	0.465	0.460	15.3	20.6	73.2	36.4	63.3
ESM-1b [68]	UniRef50	24M	0.864	0.470	0.657	0.488	26.8	60.1	97.8	61.6	83.1
ProtBERT-BFD [20]	BFD	2.1B	0.838	0.279	0.456	0.408	26.6	55.8	97.6	60.0	72.2
IEConv (amino level) [37]	PDB	476K	-	0.468	0.661	0.516	50.3	80.6	99.7	76.9	88.1
LM-GVP [85]	UniRef100	0.21B	0.664	0.417	0.545	0.527	-	-	-	-	-
GearNet-Edge-IEConv [94]	AlphaFoldDB	805K	0.874	0.490	0.654	0.488	54.1	80.5	99.9	78.2	87.5
Ours	-		0.866	0.474	**0.675**	0.483	**63.1**	**81.2**	99.6	**81.3**	**89.6**

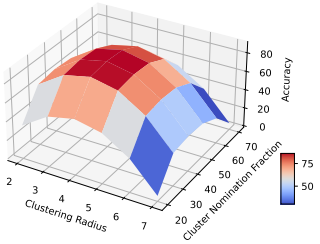


Figure 3. Performance change curve with different combinations of ω and r for enzyme reaction classification. See §4.5 for details.

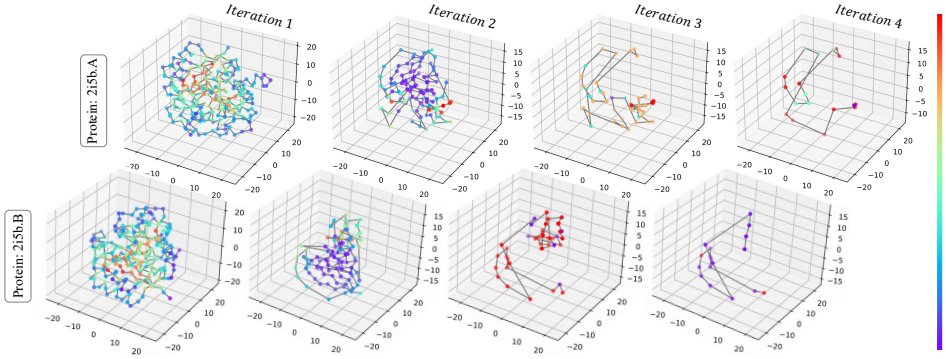


Figure 4. Visualization results of the protein structure at each iteration. The color of the node denotes the score calculated in CN step. See related analysis in §4.6.

shown in Figure 3, the best performance is achieved when $\omega = 40\%$, suggesting a suitable trade-off between preserving information and reducing redundancy and complexity.

Number of Iterations. Table 4 also studies the impact of the number of iterations. For enzyme reaction classification, increasing T from 1 to 4 leads to better performance (*i.e.*, 84.7%→89.6%). However, the accuracy drops significantly from 89.6% to 86.3% when T is set as 5. This may be because over-clustering finds insufficient and insignificant amino acids, which are harmful to representation learning. Similar trends can be observed in the results of other tasks.

Percentages of Missing Coordinates. In some cases, the protein structures may have missing coordinates due to experimental errors or incomplete data. To test the robustness of our framework to handle such cases, we randomly remove a certain percentage $u\%$ of coordinates from the protein structures and evaluate our framework on protein fold classification. The results are shown in Table 5, where we can find that our framework still achieves competitive performance when some of the coordinates are missing. For instance, in Superfamily evaluation scenario, our framework achieves an average accuracy of 78.7% when 10% of the coordinates are missing, which is only slightly lower than the accuracy of 81.2% when no coordinates are missing. This indicates that our framework learns robust representations of proteins that are not sensitive to missing coordinates.

Comparison to Existing Protein Language Models. To further showcase the effectiveness of our method, we compare our algorithm with some recent protein pretraining language models on fold classification [20, 37, 68, 85, 94], which are typically pre-trained with a large scale of data. As shown in Table 6, our method still yields better results without any pre-training or self-supervised learning. It may be that one or a few self-supervised tasks are insufficient to learn effective representations, as mentioned in [22]. This sheds light on the direction of our future efforts: it is interesting to embrace our algorithm with existing protein language models since our core idea is principled.

4.6. Visualization

We visualize the protein structure at each iteration in Figure 4. The color of the node corresponds to the score calculated in CN step. By using such an iterative clustering algorithm, this method is supposed to explore the critical amino

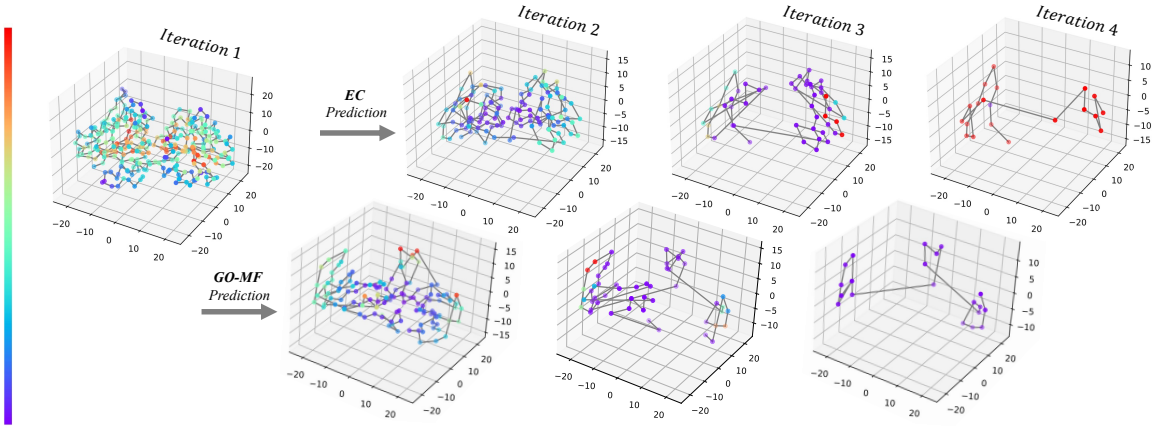


Figure 5. Clustering results for a protein exhibit variations across EC and GO-MF predictions. See related analysis in §4.6.

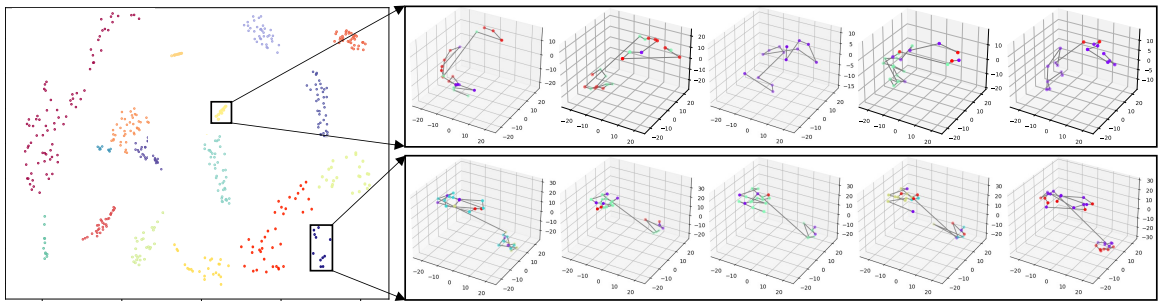


Figure 6. UMAP projection [58] of the learned representation. See related analysis in §4.6.

acids of the protein. For example, we examine the protein '2i5b.A', characterized by a complex structure consisting of numerous loops and helices. After the first iteration of clustering, our method selects some amino acids that are located at the ends or bends of these loops and helices as the medoid nodes for the next iteration. Subsequently, after the second clustering iteration, our method further narrows down the number of amino acids by selecting those that have high scores. Ultimately, our method identifies a small subset of amino acids with the highest scores, which are seen as the representative ones for the protein. When comparing the visualization results of protein chain pairs stemming from the same family or protein, *e.g.*, '2i5b.A' *vs* '2i5b.B', we observe remarkable similarity in their clustering outcomes, suggesting that they possess critical amino acids fundamentally determining their respective structures and functionalities. This further validates that our method is effective in identifying these critical amino acids.

In Figure 5, we present the clustering results of the same protein for different tasks: EC and GO-MF. Interestingly, we observe variations in the results for the same protein across different tasks, indicating that different critical components are identified for different functions. Moreover, some parts are highlighted for both two tasks. This is probably because these parts are informative across tasks. To further prove that our method can indeed discover some functional motifs of proteins, following LM-GV [85], we apply UMAP [58] to analyze the learned representation at the penultimate layer on Enzyme reaction classification and use DBSCAN32 [21] to extract protein clusters. 20 of 384 classes are shown in Figure 6, where two clusters are selected for detailed analysis. It is clear that proteins originating from the same cluster exhibit similar structural motifs, as generated by our method. This compelling observation underscores the efficacy of our clustering approach in identifying proteins possessing analogous structural motifs that are related to their enzyme reaction functions.

5. Conclusion

In this work, our epistemology is centered on protein representation learning by a neural clustering paradigm, which coins a compact and powerful framework to unify the community of protein science and respect the distinctive characteristics of each sub-task. The clustering insight leads us to introduce new approaches for spherical cluster initialization, cluster representation extraction, and cluster nomination based on both 1D and 3D information of amino acids. Empirical results suggest that our framework achieves superior performance in all four sub-tasks. Our research may potentially benefit the broader domain of bioinformatics and computational biology as a whole.

Acknowledgment. This work was supported in part by the Fundamental Research Funds for the Central Universities (No. 226-2022-00051), and Natural Science Foundation of Zhejiang Province (DT23F020008).

References

[1] Laith Abualigah, Khaled H Almotairi, Mohammed AA Al-qaness, Ahmed A Ewees, Dalia Yousri, Mohamed Abd Elaziz, and Mohammad H Nadimi-Shahraki. Efficient text document clustering approach using multi-search arithmetic optimization algorithm. *Knowledge-Based Systems*, 248:108833, 2022. 2

[2] Charu C Aggarwal and ChengXiang Zhai. A survey of text clustering algorithms. *Mining text data*, pages 77–128, 2012. 2

[3] Waqar Ali, Sebastiano Vascon, Thilo Stadelmann, and Marcello Pelillo. Quasi-cliquepool: Hierarchical graph pooling for graph classification. In *SAC*, 2023. 2

[4] Afshine Amidi, Shervine Amidi, Dimitrios Vlachakis, Vasileios Megalooikonomou, Nikos Paragios, and Evangelia I Zacharaki. Enzynet: enzyme classification using 3d convolutional neural networks on spatial representation. *PeerJ*, 6:e4750, 2018. 2

[5] Ehsaneddin Asgari and Mohammad RK Mofrad. Continuous distributed representation of biological sequences for deep proteomics and genomics. *PloS ONE*, 10(11):e0141287, 2015. 2

[6] Jinheon Baek, Minki Kang, and Sung Ju Hwang. Accurate learning of graph representations with graph multiset pooling. In *ICLR*, 2021. 2

[7] Federico Baldassarre, David Menéndez Hurtado, Arne Elofsson, and Hossein Azizpour. Graphqa: protein model quality assessment using graph convolutional networks. *Bioinformatics*, 37(3):360–366, 2021. 2, 5, 6

[8] TL Blundell and SP Wood. Is the evolution of insulin darwinian or due to selectively neutral mutation? *Nature*, 257 (5523):197–203, 1975. 1

[9] Carl Ivar Branden and John Tooze. *Introduction to protein structure*. Garland Science, 2012. 1

[10] Yue Cao, Payel Das, Vijil Chenthamarakshan, Pin-Yu Chen, Igor Melnyk, and Yang Shen. Fold2seq: A joint sequence (1d)-fold (3d) embedding-based generative model for protein design. In *ICML*, 2021. 2

[11] Guikun Chen, Xia Li, Yi Yang, and Wenguan Wang. Neural clustering based visual representation learning. In *CVPR*, 2024. 2

[12] Pengguang Chen, Shu Liu, and Jiaya Jia. Jigsaw clustering for unsupervised visual representation learning. In *CVPR*, 2021. 2

[13] Kuo-Chen Chou and Yu-Dong Cai. Predicting protein quaternary structure by pseudo amino acid composition. *Proteins: Structure, Function, and Bioinformatics*, 53(2):282–289, 2003. 1

[14] Liam Close and Rasha Kashef. Combining artificial immune system and clustering analysis: A stock market anomaly detection model. *Journal of Intelligent Learning Systems and Applications*, 12(04):83–108, 2020. 2

[15] Georgy Derevyanko, Sergei Grudinin, Yoshua Bengio, and Guillaume Lamoureux. Deep convolutional networks for quality assessment of protein folds. *Bioinformatics*, 34(23): 4046–4053, 2018. 2, 5

[16] Inderjit S Dhillon, Yuqiang Guan, and Brian Kulis. Weighted graph cuts without eigenvectors a multilevel approach. *IEEE TPAMI*, 29(11):1944–1957, 2007. 6

[17] Yuhang Ding, Liulei Li, Wenguan Wang, and Yi Yang. Clustering propagation for universal medical image segmentation. In *CVPR*, 2024. 2

[18] Russell F Doolittle. Similar amino acid sequences: chance or common ancestry? *Science*, 214(4517):149–159, 1981. 1

[19] David K Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P Adams. Convolutional networks on graphs for learning molecular fingerprints. In *NeurIPS*, 2015. 2

[20] Ahmed Elnaggar, Michael Heinzinger, Christian Dallago, Ghalia Rehawi, Yu Wang, Llion Jones, Tom Gibbs, Tamas Feher, Christoph Angerer, Martin Steinegger, et al. Prottrans: Towards cracking the language of lifes code through self-supervised deep learning and high performance computing. *IEEE TPAMI*, 2021. 7

[21] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD*, 1996. 8

[22] Hehe Fan, Zhangyang Wang, Yi Yang, and Mohan Kankanhalli. Continuous-discrete convolution for geometry-sequence modeling in proteins. In *ICLR*, 2023. 1, 2, 3, 4, 5, 6, 7

[23] Hehe Fan, Linchao Zhu, Yi Yang, and Mohan Kankanhalli. Pointlistnet: Deep learning on 3d point lists. In *CVPR*, 2023. 1

[24] Tuo Feng, Wenguan Wang, Xiaohan Wang, Yi Yang, and Qinghua Zheng. Clustering based point cloud representation learning for 3d analysis. In *ICCV*, 2023. 2

[25] Tuo Feng, Ruijie Quan, Xiaohan Wang, Wenguan Wang, and Yi Yang. Interpretable3d: an ad-hoc interpretable classifier for 3d point clouds. In *AAAI*, 2024. 2

[26] William A Frazier, Ruth Hogue Angeletti, and Ralph A Bradshaw. Nerve growth factor and insulin: Structural similarities indicate an evolutionary relationship reflected by physiological action. *Science*, 176(4034):482–488, 1972. 1

[27] Hongyang Gao and Shuiwang Ji. Graph u-nets. In *ICML*, 2019. 2

[28] Hongyang Gao, Yi Liu, and Shuiwang Ji. Topology-aware graph pooling networks. *IEEE TPAMI*, 43(12):4512–4518, 2021.

[29] Xing Gao, Wenrui Dai, Chenglin Li, Hongkai Xiong, and Pascal Frossard. ipool-information-based pooling in hierarchical graph neural networks. *IEEE TNNLS*, 33(9):5032–5044, 2021. 2

[30] Vladimir Gligorijević, P Douglas Renfrew, Tomasz Kosciolek, Julia Koehler Leman, Daniel Berenberg, Tommi Vatanen, Chris Chandler, Bryn C Taylor, Ian M Fisk, Hera Vlamakis, et al. Structure-based protein function prediction using graph convolutional networks. *Nature Communications*, 12(1):3168, 2021. 1, 2, 5

[31] Sridhar Govindarajan, Ruben Recabarren, and Richard A Goldstein. Estimating the total number of protein folds. *Proteins: Structure, Function, and Bioinformatics*, 35(4):408–414, 1999. 1

[32] Ramu Govindasamy, Surendran Arumugam, Jingkun Zhuang, Kathleen M Kelley, and Isaac Vellangany. Cluster analysis of wine market segmentation-a consumer based study in the mid-atlantic usa. *Economic Affairs*, 63(1): 151–157, 2018. 2

[33] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *NeurIPS*, 2017. 2

[34] John A Hartigan and Manchek A Wong. Algorithm as 136: A k-means clustering algorithm. *Journal of the royal statistical society. series c (applied statistics)*, 28(1):100–108, 1979. 2

[35] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 4

[36] Pedro Hermosilla and Timo Ropinski. Contrastive representation learning for 3d protein structures. In *ICLR*, 2022. 5, 6

[37] Pedro Hermosilla and Timo Ropinski. Contrastive representation learning for 3d protein structures. *arXiv preprint arXiv:2205.15675*, 2022. 4, 7

[38] Pedro Hermosilla, Marco Schäfer, Matěj Lang, Gloria Fackelmann, Pere Pau Vázquez, Barbora Kozlíková, Michael Krone, Tobias Ritschel, and Timo Ropinski. Intrinsic-extrinsic convolution and pooling for learning on 3d protein structures. In *ICLR*, 2021. 1, 2, 4, 5, 6

[39] Jie Hou, Badri Adhikari, and Jianlin Cheng. Deepsf: deep convolutional neural network for mapping protein sequences to folds. *Bioinformatics*, 34(8):1295–1303, 2018. 2, 5

[40] John Ingraham, Vikas Garg, Regina Barzilay, and Tommi Jaakkola. Generative models for graph-based protein design. In *NeurIPS*, 2019. 4, 1

[41] Vernon M Ingram. Sickle-cell anemia hemoglobin: the molecular biology of the first "molecular disease"-the crucial importance of serendipity. *Genetics*, 167(1):1–7, 2004. 1

[42] Deepak Jaiswal, Vikrant Kaushal, Pankaj Kumar Singh, and Abhijeet Biswas. Green market segmentation and consumer profiling: a cluster approach to an emerging consumer market. *Benchmarking: An International Journal*, 28(3):792–812, 2020. 2

[43] R Janani and S Vijayarani. Text document clustering using spectral clustering algorithm with particle swarm optimization. *Expert Systems with Applications*, 134:192–200, 2019. 2

[44] Bowen Jing, Stephan Eismann, Pratham N. Soni, and Ron O. Dror. Learning from protein structure with geometric vector perceptrons. In *ICLR*, 2021. 2, 5

[45] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICML*, 2017. 2

[46] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017. 5, 6

[47] Boris Knyazev, Graham W Taylor, and Mohamed Amer. Understanding attention and generalization in graph neural networks. In *NeurIPS*, 2019. 2

[48] Maxat Kulmanov and Robert Hoehndorf. Deepgoplus: improved protein function prediction from sequence. *Bioinformatics*, 36(2):422–429, 2020. 2

[49] Maxat Kulmanov, Mohammed Asif Khan, and Robert Hoehndorf. Deepgo: predicting protein functions from sequence and interactions using a deep ontology-aware classifier. *Bioinformatics*, 34(4):660–668, 2018. 2

[50] Junhyun Lee, Inyeop Lee, and Jaewoo Kang. Self-attention graph pooling. In *ICML*, 2019. 2

[51] Chen Liang, Wenguan Wang, Jiaxu Miao, and Yi Yang. Gmmseg: Gaussian mixture based generative semantic segmentation models. In *NeurIPS*, 2022. 2

[52] James Liang, Tianfei Zhou, Dongfang Liu, and Wenguan Wang. Clustseg: Clustering for universal segmentation. In *ICML*, 2023. 2

[53] Chuang Liu, Yibing Zhan, Jia Wu, Chang Li, Bo Du, Wenbin Hu, Tongliang Liu, and Dacheng Tao. Graph pooling for graph neural networks: Progress, challenges, and opportunities. In *IJCAI*, 2023. 2

[54] Stuart Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982. 2

[55] Yu Lu, Ruijie Quan, Linchao Zhu, and Yi Yang. Zero-shot video grounding with pseudo query lookup and verification. *IEEE TIP*, 33:1643–1654, 2024. 2

[56] Xiao Luo, Wei Ju, Meng Qu, Yiyang Gu, Chong Chen, Minghua Deng, Xian-Sheng Hua, and Ming Zhang. Clear: Cluster-enhanced contrast for self-supervised graph representation learning. *IEEE TNNLS*, 2022. 2

[57] Enxhell Luzhnica, Ben Day, and Pietro Lio. Clique pooling for graph classification. In *ICLR Workshop*, 2019. 2

[58] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018. 8

[59] Andrew Ng, Michael Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *NeurIPS*, 2001. 2

[60] Constance Tom Noguchi and Alan N Schechter. The intracellular polymerization of sickle hemoglobin and its relevance to sickle cell disease. *Blood*, 58(6):1057–1068, 1981. 1

[61] DAVID N Orth. Adrenocorticotropic hormone (acth). *Methods of hormone radioimmunoassay*, 2:245–278, 1979. 1

[62] Hakime Öztürk, Arzucan Özgür, and Elif Ozkirimli. DeepDTA: deep drug–target binding affinity prediction. *Bioinformatics*, 34(17):i821–i829, 2018. 7

[63] Linus Pauling, Robert B Corey, and Herman R Branson. The structure of proteins: two hydrogen-bonded helical configurations of the polypeptide chain. *Proceedings of the National Academy of Sciences*, 37(4):205–211, 1951. 1

[64] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *NeurIPS*, 2017. 3

[65] Ruijie Quan, Yu Wu, Xin Yu, and Yi Yang. Progressive transfer learning for face anti-spoofing. *IEEE TIP*, 30:3946–3955, 2021. 2

[66] Roshan Rao, Nicholas Bhattacharya, Neil Thomas, Yan Duan, Xi Chen, John Canny, Pieter Abbeel, and Yun S Song. Evaluating protein transfer learning with tape. In *NeurIPS*, 2019. 5

[67] Douglas A Reynolds et al. Gaussian mixture models. *Encyclopedia of biometrics*, 741(659-663), 2009. 2

[68] Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Jason Liu, Demi Guo, Myle Ott, C Lawrence Zitnick, Jerry Ma, et al. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. In *National Academy of Sciences*, 2021. 7

[69] Frederick Sanger. The arrangement of amino acids in proteins. In *Advances in Protein Chemistry*, pages 1–67, 1952. 1

[70] Frederick Sanger and Hans Tuppy. The amino-acid sequence in the phenylalanyl chain of insulin. 1. the identification of lower peptides from partial hydrolysates. *Biochemical journal*, 49(4):463, 1951. 1

[71] Vıctor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E (n) equivariant graph neural networks. In *ICML*, 2021. 1

[72] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1): 61–80, 2008. 2

[73] Alexey Strokach, David Becerra, Carles Corbi-Verge, Albert Perez-Riba, and Philip M Kim. Fast and flexible protein design using deep graph neural networks. *Cell Systems*, 11(4): 402–411, 2020. 2

[74] Alvin Subakti, Hendri Murfi, and Nora Hariadi. The performance of bert as data representation of text clustering. *Journal of Big Data*, 9(1):1–21, 2022. 2

[75] Peter D Sun, Christine E Foster, and Jeffrey C Boyington. Overview of protein structural and functional folds. *Current Protocols in Protein Science*, 35(1):17–1, 2004. 1

[76] Raphael Townshend, Rishi Bedi, Patricia Suriana, and Ron Dror. End-to-end learning on 3d protein structure for interface prediction. In *NeurIPS*, 2019. 2

[77] Masashi Tsubaki, Kentaro Tomii, and Jun Sese. Compound–protein interaction prediction with end-to-end learning of neural networks for graphs and sequences. *Bioinformatics*, 35(2):309–318, 2019. 2

[78] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 3

[79] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *ICLR*, 2018. 5, 6

[80] Chao Wang, Hehe Fan, Ruijie Quan, and Yi Yang. Protchatgpt: Towards understanding proteins with large language models. *arXiv preprint arXiv:2402.09649*, 2024. 1

[81] Limei Wang, Haoran Liu, Yi Liu, Jerry Kurtin, and Shuiwang Ji. Learning hierarchical protein representations via complete 3d graph networks. In *ICLR*, 2023. 1, 2, 5, 6

[82] Wenguan Wang, Jianbing Shen, Fatih Porikli, and Ruigang Yang. Semi-supervised video object segmentation with super-trajectories. *IEEE TPAMI*, 41(4):985–998, 2018. 2

[83] Wenguan Wang, Cheng Han, Tianfei Zhou, and Dongfang Liu. Visual recognition with deep nearest centroids. In *ICLR*, 2023. 2

[84] Zhengyang Wang and Shuiwang Ji. Second-order pooling for graph neural networks. *IEEE TPAMI*, 2020. 2

[85] Zichen Wang, Steven A Combs, Ryan Brand, Miguel Romero Calvo, Panpan Xu, George Price, Nataliya Golovach, Emmanuel O Salawu, Colby J Wise, Sri Priya Ponnapalli, et al. Lm-gvp: an extensible sequence and structure informed deep learning framework for protein property prediction. *Scientific Reports*, 12(1):6832, 2022. 4, 5, 6, 7, 8

[86] R Wheatland. Molecular mimicry of acth in sars–implications for corticosteroid treatment and prophylaxis. *Medical Hypotheses*, 63(5):855–862, 2004. 1

[87] Jiarui Xu, Shalini De Mello, Sifei Liu, Wonmin Byeon, Thomas Breuel, Jan Kautz, and Xiaolong Wang. Groupvit: Semantic segmentation emerges from text supervision. In *CVPR*, 2022. 2

[88] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *ICLR*, 2019. 2

[89] Kevin K Yang, Zachary Wu, Claire N Bedbrook, and Frances H Arnold. Learned protein embeddings for machine learning. *Bioinformatics*, 34(15):2642–2648, 2018. 2

[90] Junbo Yin, Dingfu Zhou, Liangjun Zhang, Jin Fang, Cheng-Zhong Xu, Jianbing Shen, and Wenguan Wang. Proposalcontrast: Unsupervised pre-training for lidar-based 3d object detection. In *ECCV*, 2022. 2

[91] Zhitao Ying, Jiaxuan You, Christopher Morris, Xiang Ren, Will Hamilton, and Jure Leskovec. Hierarchical graph representation learning with differentiable pooling. In *NeurIPS*, 2018. 2

[92] Xiaohang Zhan, Jiahao Xie, Ziwei Liu, Yew-Soon Ong, and Chen Change Loy. Online deep clustering for unsupervised representation learning. In *CVPR*, 2020. 2

[93] Muhan Zhang, Zhicheng Cui, Marion Neumann, and Yixin Chen. An end-to-end deep learning architecture for graph classification. In *AAAI*, 2018. 2

[94] Zuobai Zhang, Minghao Xu, Arian Jamasb, Vijil Chenthamarakshan, Aurelie Lozano, Payel Das, and Jian Tang. Protein representation learning by geometric structure pretraining. In *ICLR*, 2023. 1, 2, 4, 5, 6, 7

[95] Tianfei Zhou, Wenguan Wang, Ender Konukoglu, and Luc Van Gool. Rethinking semantic segmentation: A prototype view. In *CVPR*, 2022. 2