Alternate Geometric and Semantic Denoising Diffusion for Protein Inverse Folding

Chenglin Wang[1]*, Yucheng Zhou[2]*, Zhe Wang[1], Zijie Zhai[1], Jianbing Shen[2], and Kai Zhang[1]✉

[1] East China Normal University, Shanghai, China,
[2] SKL-IOTSC, CIS, University of Macau, China
52275901013@stu.ecnu.edu.cn, yucheng.zhou@connect.um.edu.mo

Abstract. Protein inverse folding is a fundamental problem in bioinformatics, aiming to recover the amino acid sequences from a given protein backbone structure. Despite the success of existing methods, they still have two limitations: (1) widely used topological modeling via GNNs may not effectively integrate geometric context of the entire protein 3D structure by focusing on only local residue message passing, and (2) current denoising processes primarily rely on geometric relations to update residue representations, while neglecting the semantic and functional correlations between different amino acid types. In this work, we propose an Alternate Geometric and Semantic Denoising Diffusion (**AGSDD**) that performs two types of denoising, i.e., geometric denoising and semantic denoising in turn, in the joint Geo-semantic residue representation space: (1) the geometric denoising module uses a geometric contextual aggregator to encode global contextual information from the entire protein structure and selectively distributes information to each residue; and (2) the semantic denoising module uses a learnable key-value dictionary of residue-types to facilitate communication between them so that learned residue features can be more accurately aligned to proper residue types. In experiments, we conduct extensive evaluations on the CATH4.2, TS50 and TS500 datasets, and observe that even without using any pre-trained protein language models, **AGSDD** still outperforms leading methods, achieving state-of-the-art performance and exhibiting strong generalization capabilities.

Keywords: Protein Inverse Folding · Diffusion Model · Alternate Denoising.

1 Introduction

Protein inverse folding, a crucial task in bioinformatics and computational biology, aims to reversely explore possible amino acid (AA) sequences from a given protein 3D structure [24, 15, 43]. These predicted sequences can autonomously fold into functional proteins, enabling the design of novel proteins with desired structural and functional properties. Moreover, some of these designed proteins,

which may not occur naturally, have significant applications in biological research, including drug design and antibody engineering [18, 3, 40].

Numerous studies have revealed the effective application of neural networks in analyzing protein [27, 36, 2, 47]. Predicting AA sequences based on protein backbone structures is a 3D structure-to-sequence mapping problem. Numerous studies have utilized GNNs [30, 33] to extract protein structural features (i.e., residue features and their connections) [15, 3, 16], followed by the Transformer to generate protein sequences in an autoregressive manner [38, 4].

Recently, diffusion models have been extensively applied for generating meaningful contents in both vision and language [11, 26, 29, 44, 1, 41, 9], due to their ability to produce highly diverse yet faithful data from the desired distribution. Notably, diffusion models have shown promise in analyzing and interpreting protein structures. For instance, DPLM [39] adopted a discrete diffusion framework to train protein sequences, exhibiting the potential of the diffusion model for protein representation learning. Similarly, Grade-IF [42] proposed a graph diffusion model for protein inverse folding, effectively learning latent protein representations by capturing inter-residue interactions, which encapsulate various reasonable sequences for a given backbone structure.

Despite the wide application of diffusion models to proteins, current diffusion-based inverse folding has two challenges. First, existing methods typically employ GNNs to establish inter-residue interaction through geometry-driven denoising. However, the locality of GNN-based message passing fails to effectively integrate the contextual information across the entire protein chain, thereby limiting comprehensive residue representation learning. From a biological standpoint, the state of a protein chain is intrinsically linked to the collective contributions of its residues [28, 32, 31]. Viewing a protein chain as a steady-state system, each residue is vital for maintaining overall stability. Therefore, effective communication among residues is essential for protein representation learning. Furthermore, existing diffusion models predominantly rely on single-pattern geometry denoising that focuses on connection relationships between residues of the chain, while overlooking the impact of semantic correlations between different residue types on residue representation. In protein sequences, AA types are not merely discrete tokens but embody functional, biological, and evolutionary relationships between residues. Considering the semantic communication of the type of residue with all AA types can better update the residue representation and assign the residue to the most appropriate AA type.

To tackle these drawbacks, we propose an alternate geometric and semantic denoising process to perform more effective residue representation learning:
(1) **Geometric Denoising**: while preserving high-fidelity local structure modeling through GNNs, we introduce a Contextual Aggregator (CA) module. This module dynamically aggregates contextual information across the entire protein chain and distributes it selectively to each residue, enabling each residue to be aware of whole-chain geometric context and update. We also call this protein-specific denoising because it depends on the chain-level structural specificity.
(2) **Semantic Denoising**: we construct a learnable residue key-value dictio-

nary containing predefined semantic embeddings for all AA types and introduce a Semantic Alignment (SA) module. This module allows residue to dynamically aggregate type-specific semantic features through attention-based cross-type communication during denoising, facilitating flexible transformations between residue types and enhancing residue representation. We also call this protein-agnostic denoising because it operates on semantic features of AA and is independent of specific protein instances. Therefore, the alternate denoising process incorporates both a geometry-based learning channel that is protein-specific and a semantics-based learning channel that is protein-agnostic.

To evaluate the performance and generalization capacity of our method, we conduct experiments on the CATH4.2, TS50 and TS500 datasets [15, 20]. Extensive experiments demonstrate that our method significantly outperforms baseline methods and achieves state-of-the-art performance. Finally, we provide detailed visualization and analysis to illustrate the effectiveness of our method.

The contributions of this work are summarized below:

- We introduce a diffusion model with an alternate geometric and semantic denoising strategy to achieve optimization based on both geometric context and semantic relationships for residue representation learning;
- We design a contextual aggregator module and a semantic alignment module, which enhance residue representations by using the context of the entire chain and facilitating communication between residue types;
- Our method demonstrates strong generalization and surpasses state-of-the-art approaches on the CATH4.2, TS50 and TS500 datasets.

2 Related Work

Protein inverse folding can be formulated as a structure-based conditional generation, where 3D structure can be encoded to a knn-graph. Node and edge features represent residues and their relationships. Previous work like GraphTrans [15] extracted protein backbone features (e.g., angles and distances) for autoregressive sequence decoding. Recent works enhanced structural representation: GVP-GNN [16] introduced geometric vector perceptions to jointly model geometric and relational features, ProteinMPNN [3] incorporated virtual $\boldsymbol{C}_\beta$ atoms as additional input features for improved performance, PiFold [8] and VFN [22] leveraged virtual atoms to capture hidden structural patterns. Besides, to fully consider sequence information in the mapping process, ESM-IF [13] augmented training data and used this additional data to train, resulting in significant improvements. LM-Design [46] and KWDesign [6] employed pre-trained language models to refine amino acid sequences iteratively. These methods have achieved significant success in sequence recovery.

In recent years, generative models have garnered significant attention [19, 10, 35]. DDPM [11] utilized the Diffusion paradigm by progressively introducing Gaussian noise into images and learning its reverse process, which has achieved remarkable success in image generation. Furthermore, Latent Diffusion [29] and ControlNet [44] enhanced controllability by incorporating text as a condition

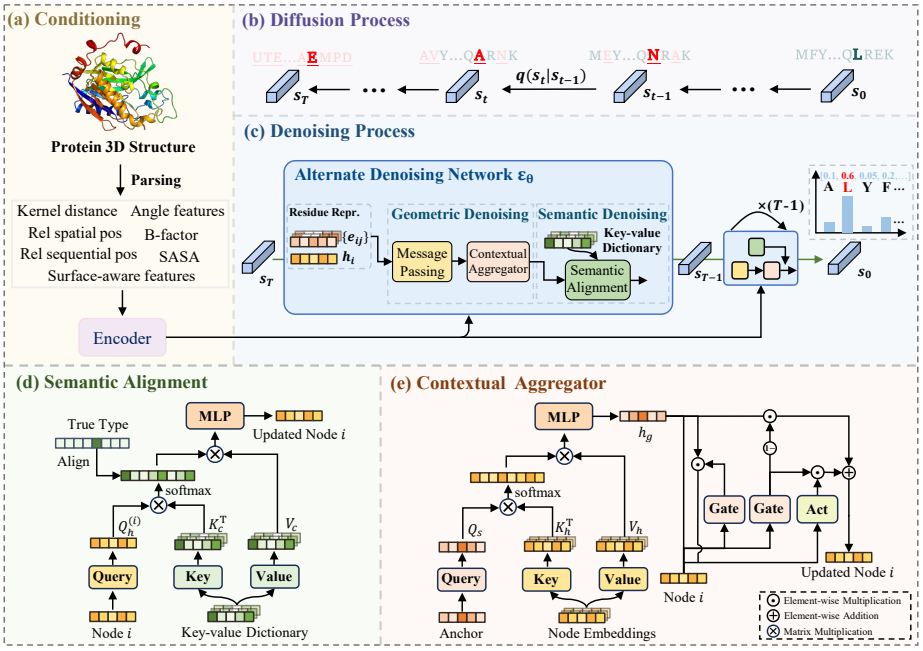


Fig. 1. Overview of **AGSDD** model, illustrated in (**a**), (**b**), (**c**). We take a certain residue type (highlighted in the figure) as an example. (**b**) The correct type (green) transforms to the incorrect noise type (underlined red). (**d**) We show the semantic alignment module in semantic denoising. The Key-value dictionary initializes the semantic features of all AA types. The true type is used to enforce the residue alignment to the proper type. (**e**) It is the contextual aggregator module in geometric denoising, where the anchor is initialized randomly to gather context of the protein-specific chain.

for image generation. In Addition, D3PM [1] has extended the multinomial diffusion model by [12] to handle discrete data. DPLM [39] applied diffusion for unconditional protein sequence generation, leading to a better understanding of proteins. DiffPreT [45] pre-trained a protein encoder by sequence-structure joint diffusion modeling and enhanced by SiamDiff, a method to capture the correlation between different conformers of a protein. CPDiffusion-SS [14] is a latent graph diffusion model that generates protein sequences based on coarse-grained secondary structure, enhancing the reliability and diversity of the generated proteins. GRADE-IF [42] proposed an innovative graph denoising diffusion model for structure-based protein sequence design, demonstrating significant potential in generating diverse protein sequences.

3 Method

In this section, we introduce our novel method **AGSDD**, using the diffusion model for protein inverse folding. As shown in Fig. 1, our approach starts with

feature extraction from input protein structure and diffusion modeling. We then delve into our alternate denoising network, comprising geometric denoising and semantic denoising.

3.1 Feature Extraction from Protein Structure

To obtain protein 3D structure features, we parse the backbone structure and construct a K-nearest neighbor graph $\mathcal{G}(\boldsymbol{X}, \boldsymbol{E})$ based on the coordinates of $\boldsymbol{C}_\alpha$ atoms, where K is 30 at default. The $\mathcal{G}(\boldsymbol{X}, \boldsymbol{E})$ comprises node features $\boldsymbol{X} \in \mathbb{R}^{N \times d_n}$ and edge features $\boldsymbol{E} \in \mathbb{R}^{M \times d_e}$, where these features are used to represent residues and their relationships, and N and M denote the numbers of nodes and edges, respectively. Following [42], node and edge features are defined as follows:

$$\boldsymbol{X} = \text{Encoder}_{\text{Node}}\left(\boldsymbol{X}_b; \boldsymbol{X}_{sasa}; \boldsymbol{X}_a; \boldsymbol{X}_s\right), \tag{1}$$

$$\boldsymbol{E} = \text{Encoder}_{\text{Edge}}\left(\boldsymbol{E}_k; \boldsymbol{E}_{sp}; \boldsymbol{E}_{se}\right), \tag{2}$$

where B-Factor $\boldsymbol{X}_b \in \mathbb{R}^{N \times 1}$ and solvent-accessible surface area (SASA) $\boldsymbol{X}_{sasa} \in \mathbb{R}^{N \times 1}$ are derived from the scalar values of $\boldsymbol{C}_\alpha$ atoms. B-Factor reflects the static stability of the protein, while SASA provides insights into protein folding and hydrophobicity. Angle features $\boldsymbol{X}_a \in \mathbb{R}^{N \times 4}$ contain the sine and cosine of backbone dihedral angles ψ and ϕ, i.e., local geometry of residues. Surface-aware features $\boldsymbol{X}_s \in \mathbb{R}^{N \times 5}$ are encoded as vectors according to a set of hyperparameters λ, representing the normalized distances between the central amino acid and its one-hop neighbors [5]. For edge features, kernel-based distances $\boldsymbol{E}_k \in \mathbb{R}^{M \times 15}$ are described using Gaussian radial basis functions (RBF) with varying bandwidths to capture distance information between connected residues at different scales, totaling 15 different distance features. $\boldsymbol{E}_{sp} \in \mathbb{R}^{M \times 12}$ are derived from the heavy atom positions of the corresponding residues, totaling 12 relative position features [42]. The relative sequence distances $\boldsymbol{E}_{se} \in \mathbb{R}^{M \times 66}$ use 65-dimensional one-hot vectors as bins to encode the relative sequence distance of two residues in the protein chain, along with a binary feature indicating whether the Euclidean distance between two connected residues is less than a specified threshold.

3.2 Diffusion Modeling

Our method is based on a diffusion modeling framework [1] for protein inverse folding, which includes both diffusion and denoising processes.

Diffusion Process. In the diffusion process, noise is introduced to the clean AA type $\boldsymbol{S}_0 \in \mathbb{R}^{N \times 20}$ of nodes. Specifically, at timestep t, each node's AA type $\boldsymbol{s}_0 \in \mathbb{R}^{20}$ in the sequence transforms to other amino acid types using a probability transfer matrix $\boldsymbol{Q}_t = \alpha_t \boldsymbol{I} + (1 - \alpha_t)\, \mathbf{1}_k \mathbf{1}_k^\top / k, \boldsymbol{Q}_t \in \mathbb{R}^{20 \times 20}$ with $\boldsymbol{I}$ being the identity matrix and k being the number of AA types and $\mathbf{1}$ being the one vector of dimension k, i.e.,

$$p\left(\boldsymbol{s}_t | \boldsymbol{s}_{t-1}\right) = \boldsymbol{Q}_t \cdot \boldsymbol{s}_{t-1}, \tag{3}$$

where $\boldsymbol{s}_t$ and $\boldsymbol{s}_{t-1}$ represent node's noise AA type in step t and $t-1$, respectively. Similar to DDPM [11], we can compute node's noise AA type in step t from initial step, denoted as follows:

$$p\left(\boldsymbol{s}_t|\boldsymbol{s}_0\right) = \bar{\boldsymbol{Q}}_t \cdot \boldsymbol{s}_0. \tag{4}$$

$\bar{\boldsymbol{Q}}_t$ denotes transition probability from initial step to t directly, and $\boldsymbol{s}_0$ represents node's original AA type.

Denoising Process. In the denoising process, each node's noise AA type is sampled from the uniformly prior distribution and iterated back to the initial distribution. The transformation between distributions is sketched as follows:

$$p_\theta\left(\boldsymbol{s}_{t-1}|\boldsymbol{s}_t, \mathcal{G}\right) = \sum_{\hat{\boldsymbol{s}}_0} q\left(\boldsymbol{s}_{t-1}|\hat{\boldsymbol{s}}_0, \boldsymbol{s}_t, \mathcal{G}\right) p_\theta\left(\hat{\boldsymbol{s}}_0|\boldsymbol{s}_t, \mathcal{G}\right), \tag{5}$$

where $\hat{\boldsymbol{s}}_0$ is predicted AA type and $q(\boldsymbol{s}_{t-1}|\hat{\boldsymbol{s}}_0, \boldsymbol{s}_t, \mathcal{G})$ represent posterior that can be computed as follows:

$$q\left(\boldsymbol{s}_{t-1}|\hat{\boldsymbol{s}}_0, \boldsymbol{s}_t, \mathcal{G}\right) = \mathbf{DIST}\left(\boldsymbol{s}_{t-1}|\frac{\boldsymbol{Q}_t^T \boldsymbol{s}_t \odot \bar{\boldsymbol{Q}}_{t-1}^T \hat{\boldsymbol{s}}_0}{\boldsymbol{s}_t^T \bar{\boldsymbol{Q}}_t \hat{\boldsymbol{s}}_0}\right), \tag{6}$$

where **DIST** is a categorical distribution over 20 AA types with probabilities computed by the posterior distribution [42].

3.3 AGSDD Denoising Network

As shown in Fig. 1, we propose **AGSDD**, including an alternate denoising network $\epsilon_\theta(\boldsymbol{S}_t, t, \mathcal{G})$ to predict the distribution $p_\theta(\hat{\boldsymbol{s}}_0|\boldsymbol{s}_t, \mathcal{G})$ of each node. The network includes two denoising phases: (1) geometric denoising consists of message passing and the contextual aggregator module, and (2) semantic denoising includes the semantic alignment module. We concatenate the corresponding $\boldsymbol{S}_t$ and $\boldsymbol{X}$ to form the initial node representation $\boldsymbol{H} = \{\boldsymbol{h}_1, ...\boldsymbol{h}_i, ...\boldsymbol{h}_N\}, \boldsymbol{H} \in \mathbb{R}^{N \times d}$.

Message Passing. The Message Passing module updates node representations using information from neighboring nodes and their relationships. Firstly, given a node $\boldsymbol{h}_i$ as an example, a gating mechanism within the **Cell** in Eq.(7-9), dynamically adjusts both node and edge features, producing the message $\boldsymbol{m}'_{ij}$. Specifically, the node $\boldsymbol{h}_i$ is concatenated with its neighboring node $\boldsymbol{h}_j$ as the message $\boldsymbol{m}_{ij}$, which is then merged with the edge features $\boldsymbol{e}_{ij}$ as gates, i.e.,

$$\boldsymbol{g}_{ij}^{(1)} = \sigma\left(\text{Linear}\left([\boldsymbol{e}_{ij}; \boldsymbol{m}_{ij}]\right)\right), \ \boldsymbol{g}_{ij}^{(2)} = \sigma\left(\text{Linear}\left([\boldsymbol{e}_{ij}; \boldsymbol{m}_{ij}]\right)\right), \tag{7}$$

where σ is the sigmoid function. $\boldsymbol{g}_{ij}^{(1)}$ and $\boldsymbol{g}_{ij}^{(2)}$ are two gates, which are used to update message $\boldsymbol{m}_{ij}$, i.e,

$$\boldsymbol{n}_{ij} = \mathbf{Act}\left(\text{Linear}(\boldsymbol{e}_{ij}) + \boldsymbol{g}_{ij}^{(1)} \odot \text{Linear}\left(\boldsymbol{m}_{ij}\right)\right), \tag{8}$$

$$\boldsymbol{m}'_{ij} = \boldsymbol{g}_{ij}^{(2)} \odot \boldsymbol{m}_{ij} + (1 - \boldsymbol{g}_{ij}^{(2)}) \odot \boldsymbol{n}_{ij}, \tag{9}$$

where $\mathbf{Act}(.)$ is the activation function. Subsequently, messages from all neighboring nodes are aggregated to update the central node's representation, i.e.,

$$\boldsymbol{h}_i' = \mathbf{MLP}\left(\boldsymbol{h}_i, \sum_{j \in \mathcal{N}_i} \boldsymbol{m}_{ij}'\right), \tag{10}$$

where $\boldsymbol{h}_i'$ is the updated feature of node i, and $\mathcal{N}_i$ represents the set of neighbors of node i.

Contextual Aggregator. To effectively enhance representations for residues, we propose the contextual aggregator module, as shown in Fig. 1e, which integrates the contextual information of the entire protein chain and selectively distributes it to each residue for access. Specifically, a learnable virtual anchor $\boldsymbol{h}_s \in \mathbb{R}^d$ is initialized at first. Subsequently, it is transformed to the query space, while the node representations updated from the Message Passing are transformed to the key and value spaces, i.e,

$$\boldsymbol{Q}_s = \boldsymbol{W}_q^s \cdot \boldsymbol{h}_s, \quad \boldsymbol{K}_h = \boldsymbol{W}_k^h \cdot \boldsymbol{H}', \quad \boldsymbol{V}_h = \boldsymbol{W}_v^h \cdot \boldsymbol{H}', \tag{11}$$

where $\boldsymbol{W}_q^s \in \mathbb{R}^{d \times d}$, $\boldsymbol{W}_k^h \in \mathbb{R}^{d \times d}$, $\boldsymbol{W}_v^h \in \mathbb{R}^{d \times d}$ are the projection matrices. $\boldsymbol{h}_s$ is randomly initialized, and $\boldsymbol{H}' = \{\boldsymbol{h}_1', ...\boldsymbol{h}_i', ...\boldsymbol{h}_n'\}$ represents all nodes in the protein. We compute the attention score between them and use the score to adaptively aggregate information from the entire protein chain, i.e,

$$\boldsymbol{h}_g = \mathbf{Softmax}\left(\frac{\boldsymbol{Q}_s \cdot \boldsymbol{K}_h^{\mathrm{T}}}{\sqrt{\boldsymbol{d}}}\right)\boldsymbol{V}_h, \tag{12}$$

where the output $\boldsymbol{h}_g \in \mathbb{R}^d$, encapsulates the contextual information of the entire protein-specific structure, which is then provided to each residue for access. We employ the **Cell** module with a gating mechanism shown in Eq.(13-15), which selectively receives the quantity of information based on the current node to enhance residue representation, i.e.,

$$\boldsymbol{g}_i^{(1)} = \sigma(\mathrm{Linear}([\boldsymbol{h}_i';\ \boldsymbol{h}_g]),\ \boldsymbol{g}_i^{(2)} = \sigma(\mathrm{Linear}([\boldsymbol{h}_i';\ \boldsymbol{h}_g])), \tag{13}$$

where σ is the sigmoid function. $\boldsymbol{g}_i^{(1)}$ and $\boldsymbol{g}_i^{(2)}$ are two gates, which are used to receive information from the protein-specific contextual feature $\boldsymbol{h}_g$ according to node $\boldsymbol{h}_i'$, i.e,

$$\boldsymbol{c}_i = \mathbf{Act}\left(\mathrm{Linear}(\boldsymbol{h}_i') + \boldsymbol{g}_i^{(1)} \odot \mathrm{Linear}(\boldsymbol{h}_g)\right), \tag{14}$$

$$\widetilde{\boldsymbol{h}}_i = \boldsymbol{g}_i^{(2)} \odot \boldsymbol{h}_g + (1 - \boldsymbol{g}_i^{(2)}) \odot \boldsymbol{c}_i, \tag{15}$$

where $\mathbf{Act}(.)$ is the activation function, $\widetilde{\boldsymbol{h}}_i$ is the updated feature of node i.

Semantic Alignment. To make residue align to proper types more accurately during the denoising process, we introduce the semantic alignment module, in Fig. 1d. It adaptively integrates type-specific semantic features to enable cross-type communication. Firstly, all amino acid types are initialized as a learnable residue key-value dictionary $\boldsymbol{H}_c \in \mathbb{R}^{20\times d}$. They are then mapped to the key and value spaces served as a reference, the node feature $\widetilde{\boldsymbol{h}}_i$ is mapped to the query space served as a request, i.e.,

$$\boldsymbol{Q}_h^{(i)} = \boldsymbol{W}_q \cdot \widetilde{\boldsymbol{h}}_i, \quad \boldsymbol{K}_c = \boldsymbol{W}_k \cdot \boldsymbol{H}_c, \quad \boldsymbol{V}_c = \boldsymbol{W}_v \cdot \boldsymbol{H}_c, \tag{16}$$

where $\boldsymbol{W}_q \in \mathbb{R}^{d\times d}$, $\boldsymbol{W}_k \in \mathbb{R}^{d\times d}$, and $\boldsymbol{W}_v \in \mathbb{R}^{d\times d}$ are the projection matrices. We calculate the correlation between the node and the 20 AA types using scaled dot-product attention, i.e.,

$$\boldsymbol{p}(\boldsymbol{h}_i) = \mathbf{Softmax}\left(\frac{\boldsymbol{Q}_h^{(i)} \cdot \boldsymbol{K}_c^{\mathrm{T}}}{\sqrt{\boldsymbol{d}}}\right), \tag{17}$$

where $\boldsymbol{p}(\boldsymbol{h}_i) \in \mathbb{R}^{20}$ represents the correlation between the i-th node and the 20 AA types. The type semantic embeddings are then weighted to the node, i.e.,

$$\boldsymbol{h}_i^l = \boldsymbol{p}(\boldsymbol{h}_i) \cdot \boldsymbol{V}_c. \tag{18}$$

To ensure each node aligns with the corresponding AA type more accurately, we apply cross-entropy loss to constrain the correlation matrix, i.e.,

$$\mathcal{L}_{attn} = -\frac{1}{N}\sum_{i=1}^{N} \boldsymbol{p}(\boldsymbol{h}_{true}^{(i)}) \log(\boldsymbol{p}(\boldsymbol{h}_i)), \tag{19}$$

where N is the number of nodes, and $\boldsymbol{p}(\boldsymbol{h}_{true}^{(i)}) \in \mathbb{R}^{20}$ is the ground truth AA types of node i. Finally, after two types of alternate denoising, the node representations are enhanced in each layer.

Following the diffusion framework [26], the time step t is mapped to γ and β to dynamically adjust the scale of features after computing the representation at each layer:

$$\boldsymbol{h}_i^l = \boldsymbol{h}_i^l * (\gamma + 1) + \beta, \tag{20}$$

γ and β denote scale and shift respectively. The dimensions of them are consistent with the node representation $\boldsymbol{h}_i^l$. The final node representation in layer m is mapped to the 20 AA types, which are associated with the secondary structure embedding ss [42]:

$$\boldsymbol{p}_i = \mathbf{MLP}\left(\boldsymbol{h}_i^m + \mathbf{Linear}(ss)\right), \tag{21}$$

where $\boldsymbol{p}_i \in \mathbb{R}^{20}$ represents the predicted amino acid type of the i-th node.

3.4 Training Objective

For the model training, we employed cross-entropy loss to optimize the model's final predictions for each node type, i.e.,

$$\mathcal{L}_{pred} = -\frac{1}{N}\sum_{i=1}^{N} \boldsymbol{p}(\boldsymbol{h}_{true}^{(i)}) \log(\boldsymbol{p}_i), \tag{22}$$

$$\mathcal{L} = \alpha \cdot \mathcal{L}_{pred} + \lambda \cdot \mathcal{L}_{attn}, \tag{23}$$

where α and λ are weight coefficients, and $\boldsymbol{p}(\boldsymbol{h}_{true}^{(i)})$ represents the true type of the i-th residue, $\boldsymbol{p}_i$ represents the predicted type of the i-th residue and $\mathcal{L}$ represents the final loss, which includes the prediction cross-entropy loss $\mathcal{L}_{pred}$ and the constraint loss $\mathcal{L}_{attn}$ in semantic denoising.

4 Experiments

4.1 Dataset and Evaluation Metrics

In our experiments, we compare our method against other approaches on the CATH4.2 dataset, a widely-used benchmark categorized based on the CATH topology classification [25]. Following the data-splitting in previous works, e.g., GraphTrans [15], PiFold [8], and GRADE-IF [42], we divide the dataset into 18,024 proteins for training, 608 proteins for validation, and 1,120 proteins for testing. In addition, we extend our evaluation to the TS50 and TS500 datasets to validate the generalization capability of our model. We employ two evaluation metrics for assessing generated AA sequences, i.e., **Recovery** rate and **Perplexity**. The recovery rate quantifies the accuracy of the generated sequences compared to the ground truth, providing insight into the model's precision. Perplexity measures the uncertainty in the model's predictions, reflecting its confidence and ability to generalize to unseen data.

4.2 Experimental Setting

To comprehensively evaluate the model's performance to recover sequences, we divide the test data into three categories: "short", "single", and "all", as shown in Table 1. The "short" comprises proteins with AA sequence lengths fewer than 100, and "single" includes proteins composed of a single chain; "all" encompasses the entire test dataset. The denoising network consists of six stacked layers, and the timestep for the diffusion model is set to 500. The model is trained for a total of 70,000 steps with a batch size of 32 and gradient accumulation over two steps on an NVIDIA A6000 GPU. We employ the Adam optimizer with a learning rate of 0.0005 and a weight decay of 0.00001. The weight of α and λ are both 0.5. In the inference process, we utilize accelerated inference methods based on [42, 34], with a skip interval of 500 and a single denoising step, striking a balance between recovery rate and perplexity.

Table 1. Experiment result on the CATH4.2 dataset.

Method	Perplexity↓			Recovery(%)↑		
	Short	**Single**	**All**	**Short**	**Single**	**All**
StructGNN [15]	8.29	8.74	6.40	29.44	28.26	35.91
GraphTrans [15]	8.39	8.83	6.63	28.14	28.46	35.82
GCA [37]	7.09	7.49	6.05	32.62	31.10	37.64
GVP [16]	7.23	7.84	5.36	30.60	28.95	39.47
AlphaDesign [7]	7.32	7.63	6.30	34.16	32.66	41.31
ProteinMPNN [3]	6.21	6.68	4.61	36.35	34.43	45.96
PiFold [8]	6.04	6.31	4.55	39.84	38.53	51.66
GRADE-IF [42]	5.49	6.21	4.35	45.27	42.77	52.21
VFN-IF [22]	5.70	5.86	4.17	41.34	40.98	54.74
AGSDD (ours)	**4.06**	**4.76**	**2.93**	**53.57**	**48.95**	**64.07**
w/ External Knowledge						
LM-Design [46]	6.77	6.46	4.52	37.88	42.47	55.65
KW-Design [6]	5.48	5.16	3.46	44.66	45.45	60.77

4.3 Main Results

To validate the effectiveness of our method, we compared it with other strong competitors using the CATH4.2 benchmark, and the results are shown in Table 1. Experimental results demonstrate that our model achieves state-of-the-art performance in AA sequence recovery and perplexity. To the best of our knowledge, our method is the first to achieve 60% recovery without external knowledge of pre-trained language models. In addition, compared to the VFN-IF model, our approach improves the recovery rate by 9.33%, confirming its superior performance. Compared with GRADE-IF, our method increases the recovery rate by 11.86%, indicating the effectiveness of semantic denoising in the denoising network for sequence recovery. Furthermore, while the LM-Design and KW-Design models utilize external knowledge from the pre-trained ESM [21], achieving recovery rates of 55.65% and 60.77%, respectively, our model improves the recovery rates by 8.42% and 3.30%. This demonstrates that our model delivers strong sequence recovery capabilities without external knowledge, thereby reducing computational complexity during the inference stage.

4.4 Generalization Capability Analysis

To verify the generalization capability of our model, we directly evaluated the trained model on the TS50 and TS500 datasets. The TS50 and TS500 datasets consist of 50 and 500 test proteins, respectively. As shown in Table 2, our model achieves state-of-the-art performance on both datasets, significantly outperforming existing methods. Specifically, our model achieves a perplexity (PPL) of 2.67 on TS50 and 2.31 on TS500, substantially outperforming existing approaches such as GRADE-IF and VFN-IF. For recovery rate (Rec), our model achieves

Table 2. Results of experiments on the TS50 and TS500 datasets. PPL refers to Perplexity, and Rec indicates Recovery (%).

Method	TS50		TS500	
	PPL	**Rec**	**PPL**	**Rec**
StructGNN [15]	5.40	43.89	4.98	45.69
GraphTrans [15]	5.60	42.20	5.16	44.66
GVP [16]	4.71	44.14	4.20	49.14
GCA [37]	5.09	47.02	4.72	47.74
AlphaDesign [7]	5.25	48.36	4.93	49.23
ProteinMPNN [3]	3.93	54.43	3.53	58.08
PiFold [8]	3.86	58.72	3.44	60.42
GRADE-IF [42]	3.71	56.32	3.23	61.22
VFN-IF [22]	3.58	59.54	3.19	63.65
AGSDD (ours)	**2.67**	**67.03**	**2.31**	**71.61**
w/ External Knowledge				
LM-Design [46]	3.50	57.89	3.19	67.78
KW-Design [6]	3.10	62.79	2.86	69.19

67.03% on TS50 and 71.61% on TS500. Notably, it is the first model, to our knowledge, that exceeds a recovery rate of 70% on the TS500 and over 65% on the TS50 without leveraging external knowledge in training. These results underscore the robustness and generalization capability of our approach. While models incorporating external knowledge in training, such as LM-Design [46] and KW-Design [6], also achieve competitive results, our model demonstrates that better performance can be reached purely through alternate denoising, thus reducing the reliance on external domain-specific information.

4.5 Ablation Study

To evaluate the impact of each module within the alternate denoising network, we conduct an ablation study, and the results are shown in Table 3. The performance metrics are evaluated across three datasets: CATH, TS50, and TS500. Firstly, removing the semantic alignment ("w/o SA") disrupts the model's understanding of various residue types, leading to a decline in performance across the CATH, TS50, and TS500 datasets. It demonstrates the necessity of the model's understanding of various residue types by aligning their representation during the denoising process. Similarly, excluding the contextual aggregator module ("w/o CA") also leads to a marked decline in performance. Without this module, the model is restricted to relying purely on graph neural network (GNN-based) neighbor inter-residue interactions, without leveraging holistic information from the entire protein chain. This limitation hinders the model's ability to contextualize residue interactions, as evidenced by decreased recall and increased perplexity across all datasets. These results confirm the effectiveness of integrating global

Table 3. Ablation study. "w/o SA" indicates the model without semantic alignment, "w/o CA" refers to the model without contextual aggregator in the geometric denoising, "w/o ALL" denotes the model without SA, CA and the cell module in message passing.

Model	CATH		TS50		TS500	
	Rec	PPL	Rec	PPL	Rec	PPL
AGSDD	**64.07**	**2.93**	**67.03**	**2.67**	**71.61**	**2.31**
w/o SA	63.13	3.00	64.46	2.80	70.32	2.39
w/o CA	61.60	3.16	64.24	2.89	68.74	2.52
w/o SA & CA	61.48	3.17	63.68	2.92	68.73	2.51
w/o ALL	60.96	3.21	63.61	2.95	68.36	2.54

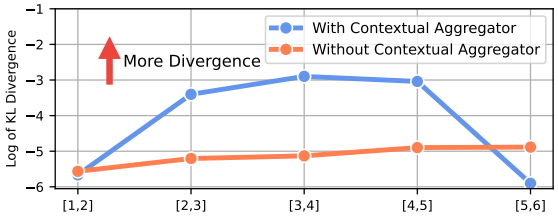


Fig. 2. Nonlinear features analysis of contextual aggregator module on layer output. The x-axis shows two adjacent layers, and the y-axis represents the logarithm of KL divergence, which measures changes in node feature distributions. The polyline depicts nonlinear divergences in node representations for 50 randomly selected protein cases from CATH4.2 test set with or without the CA module. Arrows indicate the direction towards nodes with stronger nonlinear features.

chain-level information to enrich residue representations and improve predictive accuracy. When both the semantic alignment and contextual aggregator module are simultaneously removed ("w/o SA & CA"), the model suffers further performance degradation. This reinforces the complementary contributions of these two components, highlighting that both are indispensable for capturing complex residue dependencies within the denoising network. Lastly, we explore the role of the Cell module within the Message Passing part, which integrates node and edge representations from neighboring nodes. When the Cell module is replaced with a MLP, model performance declines, indicating the crucial role of the Cell module in effectively integrating neighboring node and edge representations.

4.6 Nonlinear Analysis in Contextual Aggregator

To better understand the effectiveness of the CA module, we examine its influence on the nonlinear characteristics of layer outputs. In Fig. 2, we compare two scenarios: one where the CA module is used and another where it is not. The y-axis represents the logarithm of the KL divergence, which quantifies the changes in feature distributions between adjacent layers. A higher value indicates

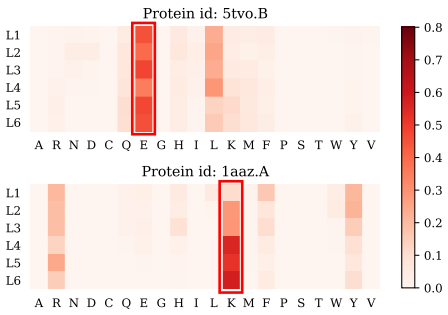


Fig. 3. How a specific position in the sequence attends to all AA types across layers in semantic alignment module. (**vertical**: different layers, **horizontal**: 20 AA types). The specific residue can incorporate the semantic information of the correct type as the layer goes deeper.

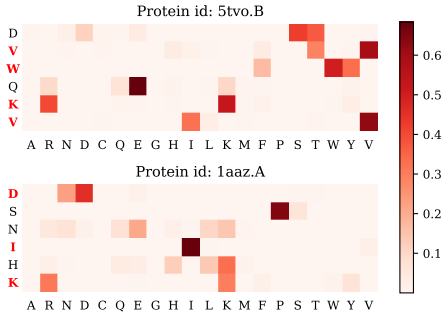
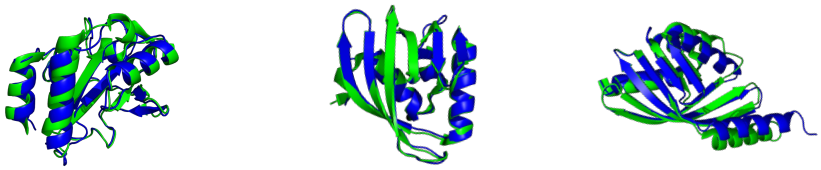


Fig. 4. How each position in the sequence attends to the correct type at the final layer in the semantic alignment module. (**Vertical**: prediction types of a segment, **horizontal**: 20 AA types). Residues with accurately predicted types (red) align to corresponding type semantic features.

a greater divergence, suggesting more pronounced nonlinear transformations. The results reveal that incorporating the CA module significantly increases the KL divergence across layers, especially in the earlier stages. This indicates that CA enhances the model's ability to capture and propagate complex nonlinear patterns, thereby improving its representation of inter-residue relationships. In contrast, without the CA module, the divergence remains consistently lower, suggesting limited capacity for nonlinear feature extraction. Thus, the CA module's impact is particularly beneficial for tasks that require nuanced representation of protein structures.

4.7 Visualization of Semantic Alignment

To investigate the impact of the semantic alignment module, we present visualization results of attention between residues and semantic features of all type. Specifically, we analyze the AA types to which individual residues attend across different layers of the denoising network, as shown in Fig. 3. We also show the AA types attended to by multiple residues in a continuous segment at the final layer, in Fig.4. In Fig. 3, the vertical axis (L1 to L6) represents the network layers, while the horizontal axis represents the 20 AA types. The values indicate the attention weights computed in the semantic denoising phase for the specific residue and each of the 20 AA types. For the 5tvo.B protein, the true type of the randomly chosen residue is glutamic acid (E), and for the 1aaz.A protein, it is lysine (K). The results demonstrate that as the number of layers increases, the attention weight for the node corresponding to the true AA type features of each residue gradually rises. By the final layer, the residue aligns with its true AA type, suggesting that the model effectively aligns residues with the semantic information of their true AA types and incorporates this information into



Recovery: 0.679 (PDB ID: 3fkf) RMSD: 0.917; avg.pLDDT: 90.39

Recovery: 0.663 (PDB ID: 4u13) RMSD: 0.565; avg.pLDDT: 95.39

Recovery: 0.654 (PDB ID: 2bng) RMSD: 0.923; avg.pLDDT: 89.63

Fig. 5. Comparison of the folding between predicted (Blue) and native (Green) structures, where the predicted structures are generated using AlphaFold2 based on **AGSDD**-designed AA sequences.

the residue to facilitate flexible transition of types to enhance representation. In addition, Fig. 4 visualizes the attention of multiple residues in the model's final layer. These residues are from a randomly selected continuous segment. The horizontal axis represents the 20 AA types, while the vertical axis represents the predicted amino acid types of these residues, where the red represents accurate prediction. The visualization shows that the correctly predicted residues have the highest attention weights for their true types in the semantic alignment module, which indicates that injecting type semantic information into the residue representations is beneficial for prediction.

4.8 Folding Ability

We further explore the folding ability of the generated amino acid sequences to verify its rationality. Specifically, we randomly select test proteins 3fkf, 4u13 and 2bng from the CATH4.2 test set and utilize the protein structure prediction method ColabFold [23], which offers user-friendly access to AlphaFold2 [17] for predicting the 3D structures of the generated amino acid sequences. These predicted structures are then aligned with the corresponding PDB structures. As shown in Fig. 5, the recovery rate of the generated sequences 3fkf is 0.679, and secondary structure elements such as α-helices and β-sheets are effectively formed. The average pLDDT score is 90.39, and the RMSD is 0.917, where the average pLDDT score assesses confidence in the predicted structure, and the RMSD measures the deviation between the predicted and fixed structures. These results demonstrate the validity and rationality of our model in generating new sequences based on fixed backbone structures.

5 Conclusion

In this paper, we propose an alternate geometric and semantic denoising diffusion **AGSDD** that performs protein-specific geometric denoising and protein-agnostic semantic denoising for protein inverse folding. Firstly, after local structure modeling through GNNs, our method integrates contextual information

from the entire 3D structure and assigns it selectively to each residue to maintain inter-residue communication, enhancing the residue representation. Moreover, we introduce a semantic denoising that use a learnable key-value dictionary of residue-types to facilitate communication between them in the denoising process. In addition, our cell module effectively decouples and computes the relevance of adjacent node and edge information. In experiments, our method surpasses existing leading approaches on the CATH4.2, TS50 and TS500 datasets.

6 Acknowledgements

This work is supported by the national key research and development program 2022YFC3400501, and national natural science foundation of China 62276099.

References

1. Austin, J., Johnson, D.D., Ho, J., Tarlow, D., Van Den Berg, R.: Structured denoising diffusion models in discrete state-spaces. NeurIPS (2021)
2. Chen, W., Wang, X., Wang, Y.: Fff: Fragment-guided flexible fitting for building complete protein structures. In: CVPR (2023)
3. Dauparas, J., Anishchenko, I., Bennett, N., Bai, H., Ragotte, R.J., Milles, L.F., Wicky, B.I., Courbet, A., de Haas, R.J., Bethel, N., et al.: Robust deep learning–based protein sequence design using proteinmpnn. Science (2022)
4. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. NAACL (2019)
5. Ganea, O., Huang, X., Bunne, C., Bian, Y., Barzilay, R., Jaakkola, T.S., Krause, A.: Independent se(3)-equivariant models for end-to-end rigid protein docking. In: ICLR (2022)
6. Gao, Z., Tan, C., Chen, X., Zhang, Y., Xia, J., Li, S., Li, S.Z.: KW-design: Pushing the limit of protein design via knowledge refinement. In: ICLR (2024)
7. Gao, Z., Tan, C., Li, S.Z.: Alphadesign: A graph protein design method and benchmark on alphafold DB (2023)
8. Gao, Z., Tan, C., Li, S.Z.: Pifold: Toward effective and efficient protein inverse folding. In: ICLR. OpenReview.net (2023)
9. Gong, S., Li, M., Feng, J., Wu, Z., Kong, L.: Diffuseq: Sequence to sequence text generation with diffusion models. In: ICLR (2023)
10. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. NeurIPS **27** (2014)
11. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. NeurIPS (2020)
12. Hoogeboom, E., Nielsen, D., Jaini, P., Forré, P., Welling, M.: Argmax flows and multinomial diffusion: Learning categorical distributions. NeurIPS (2021)
13. Hsu, C., Verkuil, R., Liu, J., Lin, Z., Hie, B., Sercu, T., Lerer, A., Rives, A.: Learning inverse folding from millions of predicted structures. In: ICML (2022)
14. Hu, Y., Tan, Y., Han, A., Zheng, L., Hong, L., Zhou, B.: Secondary structure-guided novel protein sequence generation with latent graph diffusion. CoRR (2024)
15. Ingraham, J., Garg, V., Barzilay, R., Jaakkola, T.: Generative models for graph-based protein design. NeurIPS (2019)

16. Jing, B., Eismann, S., Suriana, P., Townshend, R.J.L., Dror, R.O.: Learning from protein structure with geometric vector perceptrons. In: ICLR (2021)
17. 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 (2021)
18. Khoury, G.A., Smadbeck, J., Kieslich, C.A., Floudas, C.A.: Protein folding and de novo protein design for biotechnological applications. Trends in biotechnology (2014)
19. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. In: ICLR (2014)
20. Li, Z., Yang, Y., Faraggi, E., Zhan, J., 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 (2014)
21. Lin, Z., Akin, H., Rao, R., Hie, B., Zhu, Z., Lu, W., Smetanin, N., Verkuil, R., Kabeli, O., Shmueli, Y., et al.: Evolutionary-scale prediction of atomic-level protein structure with a language model. Science (2023)
22. Mao, W., Zhu, M., Sun, Z., Shen, S., Wu, L.Y., Chen, H., Shen, C.: De novo protein design using geometric vector field networks. In: ICLR. OpenReview.net (2024)
23. Mirdita, M., Schütze, K., Moriwaki, Y., Heo, L., Ovchinnikov, S., Steinegger, M.: Colabfold: making protein folding accessible to all. Nature methods (2022)
24. O'Connell, J., Li, Z., Hanson, J., Heffernan, R., Lyons, J., Paliwal, K., Dehzangi, A., Yang, Y., Zhou, Y.: Spin2: Predicting sequence profiles from protein structures using deep neural networks. Proteins: Structure, Function, and Bioinformatics (2018)
25. Orengo, C.A., Michie, A.D., Jones, S., Jones, D.T., Swindells, M.B., Thornton, J.M.: Cath–a hierarchic classification of protein domain structures. Structure (1997)
26. Peebles, W., Xie, S.: Scalable diffusion models with transformers. In: ICCV (2023)
27. Quan, R., Wang, W., Ma, F., Fan, H., Yang, Y.: Clustering for protein representation learning. In: CVPR (2024)
28. Rackovsky, S.: Global characteristics of protein sequences and their implications. Proceedings of the National Academy of Sciences (2010)
29. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: CVPR (2022)
30. Satorras, V.G., Hoogeboom, E., Welling, M.: E (n) equivariant graph neural networks. In: ICML. PMLR (2021)
31. Scheraga, H.A., Rackovsky, S.: Homolog detection using global sequence properties suggests an alternate view of structural encoding in protein sequences. Proceedings of the National Academy of Sciences (2014)
32. Scheraga, H.A., Rackovsky, S.: Global informatics and physical property selection in protein sequences. Proceedings of the National Academy of Sciences (2016)
33. Schütt, K., Unke, O., Gastegger, M.: Equivariant message passing for the prediction of tensorial properties and molecular spectra. In: ICML. PMLR (2021)
34. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. arXiv preprint arXiv:2010.02502 (2020)
35. Song, Y., Sohl-Dickstein, J., Kingma, D.P., Kumar, A., Ermon, S., Poole, B.: Score-based generative modeling through stochastic differential equations. In: ICLR. OpenReview.net (2021)
36. Sverrisson, F., Feydy, J., Correia, B.E., Bronstein, M.M.: Fast end-to-end learning on protein surfaces. In: CVPR (2021)
37. Tan, C., Gao, Z., Xia, J., Hu, B., Li, S.Z.: Global-context aware generative protein design. In: ICASSP (2023)

38. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. NeurIPS (2017)
39. Wang, X., Zheng, Z., Ye, F., Xue, D., Huang, S., Gu, Q.: Diffusion language models are versatile protein learners. arXiv preprint arXiv:2402.18567 (2024)
40. Watson, J.L., Juergens, D., Bennett, N.R., Trippe, B.L., Yim, J., Eisenach, H.E., Ahern, W., Borst, A.J., Ragotte, R.J., Milles, L.F., et al.: De novo design of protein structure and function with rfdiffusion. Nature **620** (2023)
41. Wu, T., Fan, Z., Liu, X., Zheng, H.T., Gong, Y., Jiao, J., Li, J., Guo, J., Duan, N., Chen, W., et al.: Ar-diffusion: Auto-regressive diffusion model for text generation. NeurIPS **36** (2023)
42. Yi, K., Zhou, B., Shen, Y., Lió, P., Wang, Y.: Graph denoising diffusion for inverse protein folding. In: NeurIPS (2023)
43. Yue, K., Dill, K.A.: Inverse protein folding problem: designing polymer sequences. Proceedings of the National Academy of Sciences (1992)
44. Zhang, L., Rao, A., Agrawala, M.: Adding conditional control to text-to-image diffusion models. In: ICCV (2023)
45. Zhang, Z., Xu, M., Lozano, A.C., Chenthamarakshan, V., Das, P., Tang, J.: Pre-training protein encoder via siamese sequence-structure diffusion trajectory prediction. In: NeurIPS 2023 (2023)
46. Zheng, Z., Deng, Y., Xue, D., Zhou, Y., Ye, F., Gu, Q.: Structure-informed language models are protein designers. In: ICML (2023)
47. Zhong, Z., Mottin, D.: Efficiently predicting mutational effect on homologous proteins by evolution encoding. In: ECML PKDD (2024)