DS-ProGen: A Dual-Structure Deep Language Model for Functional Protein Design

Yanting Li[1,2,*], Jiyue Jiang[1,*,†], Zikang Wang[3,*], Ziqian Lin[1], Dongchen He[1], Yuheng Shan[4], Yanruisheng Shao[1], Jiayi Li[1], Xiangyu Shi[1], Jiuming Wang[1], Yanyu Chen[1], Yimin Fan[1], Han Li[1], Yu Li[1,†]

[1]The Chinese University of Hong Kong
[2]Hong Kong University of Science and Technology (Guangzhou)
[3]The Hong Kong Polytechnic University
[4]National University of Singapore
yli106@connect.hkust-gz.edu.cn, jiangjy@link.cuhk.edu.hk, zikang.wang@connect.polyu.hk, liyu@cse.cuhk.edu.hk

Abstract

Inverse Protein Folding (IPF) is a critical subtask in the field of protein design, aiming to engineer amino acid sequences capable of folding correctly into a specified three-dimensional (3D) conformation. Although substantial progress has been achieved in recent years, existing methods generally rely on either backbone coordinates or molecular surface features alone, which restricts their ability to fully capture the complex chemical and geometric constraints necessary for precise sequence prediction. To address this limitation, we present **DS-ProGen**, a dual-structure deep language model for functional protein design, which integrates both backbone geometry and surface-level representations. By incorporating backbone coordinates as well as surface chemical and geometric descriptors into a next-amino-acid prediction paradigm, DS-ProGen is able to generate functionally relevant and structurally stable sequences while satisfying both global and local conformational constraints. On the PRIDE dataset, DS-ProGen attains the current state-of-the-art recovery rate of **61.47%**, demonstrating the synergistic advantage of multi-modal structural encoding in protein design. Furthermore, DS-ProGen excels in predicting interactions with a variety of biological partners, including ligands, ions, and RNA, confirming its robust functional retention capabilities.

Introduction

Inverse Protein Folding (IPF) remains a pivotal topic in molecular biology and protein engineering, with the aim of designing protein sequences with targeted functionalities for given three-dimensional structures (Yue and Dill 1992; Zhou et al. 2023; Gao et al. 2023; Jiang et al. 2025b). Recently, significant strides have been made in text processing through advancements in Natural Language Processing (NLP) and Large Language Models (LLMs) (Gu et al. 2021; Achiam et al. 2023; Jiang et al. 2023; Liu et al. 2024). These methodologies have been successfully applied to the "language" of biology (Ferruz, Schmidt, and Höcker 2022; Madani et al. 2023; Jiang et al. 2025c; Nijkamp et al. 2023; Wang et al. 2025; Jiang et al. 2025a), such as sequences and structural bioinformatics, offering fresh perspectives and technological foundations for prediction of sequence-structure relationships (Xiao et al. 2024; Heinzinger et al. 2024).

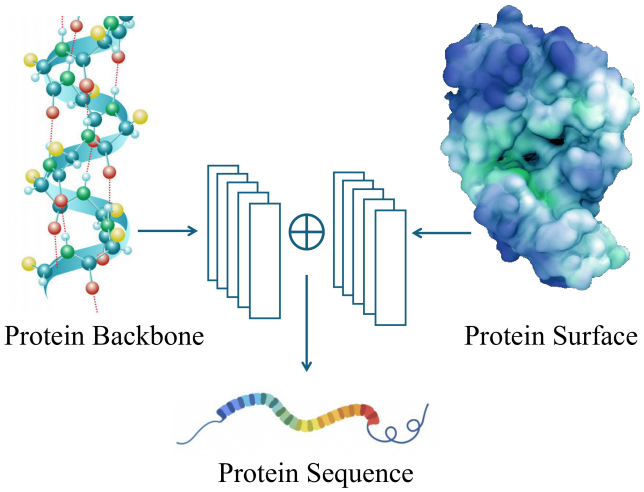

Figure 1: Protein sequence design conditioned on structural inputs: backbone geometry and solvent-accessible surface area (SASA) characteristics.

Despite notable progress in applying language model approaches to protein science (Ferruz, Schmidt, and Höcker 2022; Madani et al. 2023; Nijkamp et al. 2023), most existing models for inverse folding rely heavily on singular structural representations either utilizing backbone geometric information, like skeletal coordinates (Dauparas et al. 2022; Jendrusch, Korbel, and Sadiq 2021), or focusing solely on molecular surfaces (Song et al. 2024). While backbone coordinates can reflect the intrinsic topology of the entire protein structure, they fail to capture crucial chemical features on the exterior surface that interact with the environment or ligands (Emonts and Buyel 2023; Cao et al. 2022). Conversely, surface information tends to overlook the influence of backbone conformations on the overall stability and functionality

*These authors contributed equally to this work.
†Corresponding authors.

of proteins (Song et al. 2024). Therefore, a significant challenge remains in achieving comprehensive modeling of both the "internal-external" structures in inverse folding tasks.

Our study introduces **DS-ProGen** (Dual-Structure Protein Generator) model, which uniquely captures both the protein backbone and its SASA characteristics (Figure 1). Unlike unimodal models, DS-ProGen uses a backbone encoder to capture main chain coordinates, dihedral angles, and local geometric vectors, crucial for understanding internal structures. The surface encoder processes atomic types, curvature distributions, and chemical environments, focusing on micro-features at active sites and binding interfaces. These dual streams of information feed into a multimodal space and a multi-layer decoder that applies autoregressive techniques in sequence modeling. This approach allows for a holistic consideration of both geometric layout and surface chemical features when predicting amino acids. DS-ProGen outperforms traditional methods that rely solely on backbone coordinates or surface features, offering a richer structural and chemical analysis during inverse folding. Tests confirm its enhanced sequence generation capabilities and high functional accuracy in scenarios like ligand binding and complex molecular interactions.

The main contributions of our paper are: (1) Introducing the first multimodal protein language model for inverse folding, capturing backbone and surface chemical details. (2) Using geometric vectors with autoregressive Transformer decoding to improve sequence accuracy and structural fidelity. (3) Achieving higher sequence recovery rates on benchmarks and producing proteins viable for ligand and ion interactions. (4) Demonstrating enhancements in structural reconstruction and functional retention, with significant implications for drug discovery and synthetic biology.

Related Works

Language Models for Scientific Applications

Transformer-based language models (Devlin et al. 2019; Radford et al. 2018) have been adapted to scientific domains, which often require structured representations and domain-specific vocabularies. Recent works like NatureLM (Xia et al. 2025) and UniGenX (Zhang et al. 2025) show that combining language models with scientific data accelerates tasks such as molecule synthesis and discovery.

Protein Language Models

Protein sequences mirror natural language properties, making them suitable for language modeling. Masked models like ESM (Lin et al. 2023; Hayes et al. 2025) capture structural and functional features through large-scale sequence pretraining, enabling tasks like function prediction. Autoregressive models such as ProGen2 (Nijkamp et al. 2023) and ProtGPT2 (Ferruz, Schmidt, and Höcker 2022) generate realistic proteins and support zero-shot fitness prediction, demonstrating strong design potential.

Inverse Protein Folding

Inverse protein folding (Yue and Dill 1992) aims to generate amino acid sequences for a desired target structure. Classical approaches like Rosetta (Das and Baker 2008) can be computationally intensive, while recent deep learning methods learn structure-conditioned sequence distributions. ProteinMPNN (Dauparas et al. 2022) and ESM-IF (Hsu et al. 2022) primarily rely on backbone geometry for strong sequence recovery. Other methods, such as PiFold (Gao et al. 2022) and SurfPro (Song et al. 2024), encode local geometry or surface annotations. These approaches reveal a trade-off between internal structure capture and external interface modeling, and most still underutilize natural sequence evolutionary priors.

Methods

To address these limitations, we propose **DS-ProGen**, a dual-structure deep language model for functional protein design that integrates both backbone and surface information to comprehensively characterize protein structures. DS-ProGen consists of a dual-branch encoder that extracts geometric and biochemical features from backbone coordinates and molecular surfaces, respectively, and a dual-structure fusion design decoder. An overview of the architecture is illustrated in Figure 2.

Backbone Geometric Encoder

We propose a backbone geometric encoder to extract the structural features of a protein based on the 3D coordinates of its backbone atoms.

To construct the input embeddings, we first extract the backbone atoms, namely, nitrogen (N), carbon (C), and alpha carbon (C_α), from each residue. Their 3D coordinates are concatenated into a tensor $c_i \in \mathbb{R}^{L\times 3\times 3}$, where L is the sequence length.

Each residue is represented by a combination of scalar and vector features: the scalar feature $\mathbf{s} \in \mathbb{R}^{d_s}$ includes dihedral angles and pairwise distances, while the vector feature $\mathbf{V} \in \mathbb{R}^{d_v\times 3}$ encodes local spatial orientations. These features are updated using Geometric Vector Perceptron (GVP) layers (Jing et al. 2020), which are designed to be rotation-equivariant for vector channels and rotation-invariant for scalar channels. In our implementation, we use 4-layer GVP, each with dimensionality $d_s = 128$ and $d_v = 16$.

The update rules within each GVP layer are given by:

$$\begin{aligned} \mathbf{V}' &= \sigma^+(\mathbf{v}_\mu) \odot \mathbf{V}_\mu, \\ \mathbf{s}' &= \sigma\left(W_m[\mathbf{s}; \mathbf{s}_h] + \mathbf{b}\right), \\ \mathbf{V}_h &= W_h\mathbf{V}, \quad \mathbf{V}_\mu = W_\mu\mathbf{V}_h, \\ \mathbf{s}_h &= \|\mathbf{V}_h\|_2, \quad \mathbf{v}_\mu = \|\mathbf{V}_\mu\|_2, \end{aligned} \tag{1}$$

where W_h, W_μ, and W_m are trainable weight matrices, and σ, σ^+ denote activation functions for scalar and vector channels, respectively.

After the GVP layers, a Transformer encoder (Vaswani et al. 2017) with 8 layers is applied to capture relational dependencies among residues via multi-head self-attention. The final output is a sequence of embeddings $B \in \mathbb{R}^{L\times h_s}$, where $h_s = 1024$ matches the hidden state dimension of the sequence decoder.

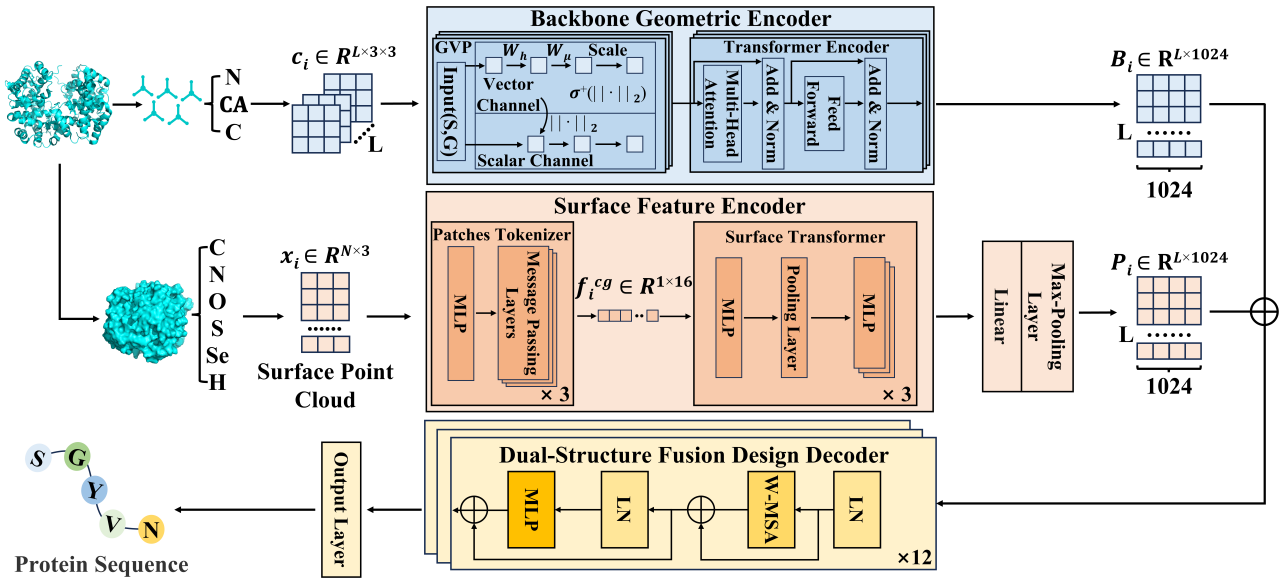

Figure 2: DS-ProGen integrates backbone and surface structural information to enable functional protein design. The backbone geometric encoder extracts geometric features from the N, C, and C_α atoms of protein structures. The surface feature encoder processes atomic types, surface points, and curvature features. The backbone and surface embeddings are projected into a unified hidden space and combined to form a single representation with the target sequence length. Then, a dual-structure fusion design decoder auto-regressively generates/designs the protein sequence conditioned on the structural context.

Surface Feature Encoder

In addition to backbone structural features, the surface information of a protein provides complementary insights into its three-dimensional geometry. To leverage this additional structural perspective, we design a dedicated surface encoder capable of extracting and encoding surface information.

Surface Features Our representation of the protein surface incorporates three major components: atomic features, chemical and geometric properties of the surface points, and local curvature information.

We extract both the atomic coordinates and atom types. Formally, for each protein chain, we denote the atom coordinates as $\{a_i\}_{i=1}^{M}$ and atom types as one-hot encodings $\{t_i\}_{i=1}^{M}$, where:

$$a_i \in \mathbb{R}^{1\times 3},\ t_i \in \mathbb{R}^{1\times 6}. \tag{2}$$

Here, M is the total number of atoms. To simplify modeling and align with existing atom embeddings, we restrict our focus to six common element types (C, N, O, S, Se, H) and discard proteins containing other atom types.

To describe the external geometry of proteins, we construct surface point clouds. We apply a smooth distance function (Blinn 1982) and van der Waals radii (Batsanov 2001) to generate accurate surface representations. The resulting surface points are denoted as:

$$\{x_i\}_{i=1}^{N},\ x_i \in \mathbb{R}^{1\times 3}, \tag{3}$$

where N is the total number of points sampled from the surface. To make the batched training available, as previous literature(Yuan et al. 2023) had done, we set the limitation of the point as 8,192. If the surface of the protein has more than 8,192 points, we will randomly choose 8,192 points as the input of surface information. If the surface data includes fewer than 8,192 points, we would randomly choose the points in the surface data above and then use them to pad the data to 8,192 points.

To enrich the surface points with local geometric and chemical context, we further compute detailed features around each surface point. For geometry, we derive normal vectors $n_i \in \mathbb{R}^{1\times 3}$ and multiscale mean and Gaussian curvatures u_i across five radii (1Å, 2Å, 3Å, 5Å, 10Å):

$$\{u_i\}_{i=1}^{N},\ u_i \in \mathbb{R}^{1\times 10}. \tag{4}$$

For chemical context, we consider the 16 nearest neighboring atoms around each surface point, following previous work (Yuan et al. 2023; Sverrisson et al. 2021), which provides a good trade-off between capturing sufficient local chemical information and maintaining computational efficiency. The distance between the i-th surface point and its j-th neighboring atom is:

$$d_{ij} = \|a_j - x_i\|, \tag{5}$$

And the complete neighborhood feature is:

$$\{(t_j, d_{ij}) \mid a_j \in N_i\} \in \mathbb{R}^{16\times 7}, \tag{6}$$

where N_i denotes the set of 16 nearest atoms to x_i, and t_j is the one-hot atom type encoding.

Patches Tokenizer Handling full surface point clouds is computationally prohibitive. Inspired by vision transformer (ViT) models (Tolstikhin et al. 2021), we segment the surface into local patches. We first select g center points via

farthest point sampling (FPS) (Eldar et al. 1997). For each selected center c_i, we gather a local patch using K-nearest neighbors (KNN) (Zhang 2016):

$$\{P_i\}_{i=1}^{g} = \text{KNN}(\{c_i\}_{i=1}^{g}, \{x_j\}_{j=1}^{N}) \in \mathbb{R}^{g\times K\times 3}, \quad (7)$$

where each patch P_i contains K surface points, centered around c_i.

Each patch is further enriched by embedding the one-hot atom types. An MLP followed by three message-passing layers encodes the atom features, producing a feature vector:

$$f_i^c \in \mathbb{R}^{1\times 6}. \quad (8)$$

To integrate geometric context, we concatenate the chemical embedding f_i^c with the curvature vector u_i, yielding:

$$f_i^{c_g} = [f_i^c; u_i] \in \mathbb{R}^{1\times 16}. \quad (9)$$

This combined feature is updated through an MLP:

$$f_i^{c_g(l+1)} = f_i^{c_g(l)} + \text{MLP}(f_i^{c_g(l)}), \quad (10)$$

starting from $f_i^{c_g(0)} = f_i^{c_g}$, $f_i \in \mathbb{R}$.

Surface Transformer The final features of all g patches are passed through a Transformer block (Vaswani et al. 2017) to model global interactions across the surface. Following feature aggregation, we project the $g \times d$ feature matrix into an $h_g \times h_s$ space via two consecutive linear layers, where h_g is the dimension of surface atomic features and h_s is the hidden state dimension of the sequence decoder layers. A max-pooling operation along the surface points reduces the dimensionality, yielding $S \in \mathbb{R}^{L\times h_s}$, where L denotes the target sequence length.

Dual-Structure Fusion Design Decoder

Our model adopts a Transformer decoder architecture to model the conditional generation of amino acid sequences (Vaswani et al. 2017; Nijkamp et al. 2023). In this framework, the sequence modeling task is formulated in an auto-regressive manner, where the joint probability of a sequence $\mathbf{x} = \{x_1, x_2, \ldots, x_T\}$ is factorized as a product of conditional probabilities:

$$P(\mathbf{x}) = \prod_{t=1}^{T} P(x_t \mid x_1, x_2, \ldots, x_{t-1}). \quad (11)$$

At each time step, the decoder predicts the next token based solely on the preceding tokens, ensuring causality in generation.

To provide the model with structural context, we first integrate the backbone and surface representations. Specifically, we compute the combined structural embedding as:

$$R = B + S, \quad (12)$$

where B and S are the backbone and surface feature embeddings, both projected into the same hidden dimension h_s.

The sequence decoder input is then constructed by prepending the combined structural embeddings R as condition to the beginning of the sequence input embeddings. Formally, the decoder input matrix is:

$$\mathbf{X} = [R; \mathbf{E_t}], \quad (13)$$

where $R \in \mathbb{R}^{L\times h_s}$ are the structural embeddings and $\mathbf{E_t} \in \mathbb{R}^{t\times h_s}$ are the sequence token embeddings.

For an input embedding matrix $\mathbf{X} \in \mathbb{R}^{t\times h_s}$, the Transformer decoder processes the sequence through stacked self-attention and feed-forward layers. The output hidden states $\mathbf{H} \in \mathbb{R}^{t\times h_s}$ are then projected by a linear layer to obtain the logits over the amino acid vocabulary:

$$\mathbf{Z} = \mathbf{H}\mathbf{W}_{\text{out}} + \mathbf{b}_{\text{out}}, \quad (14)$$

where $\mathbf{W}_{\text{out}} \in \mathbb{R}^{h_s\times V}$ and V is the vocabulary size. At each position, the next amino acid is predicted based on the logits.

Our implementation follows the GPT-2 (Radford et al. 2019; Nijkamp et al. 2023) architecture, using stacked Transformer decoder layers to progressively refine the sequence representations and predict the next amino acid at each step.

Experiments

Experimental Setup

Datasets Our training procedure is divided into a pretraining stage and a fine-tuning stage. Pretraining is motivated by the proven scaling laws of auto-regressive models, where performance systematically improves with larger model sizes and larger training datasets. Furthermore, prior work (Lin et al. 2023) has demonstrated that pretraining significantly enhances the generalization ability of protein sequence models.

For the pretraining phase, we construct two datasets: For backbone-only models, we gather over 40 million structure-sequence pairs from the FoldComp-compressed (Kim, Mirdita, and Steinegger 2023) AlphaFoldDB Cluster Representatives (Jumper et al. 2021) and the high-quality ESMAtlas datasets (Lin et al. 2023), including only proteins with fewer than 512 residues. For models learning from both backbone and surface, we select roughly 80,000 proteins from the same collection, focusing on those with valid surface data. This number balances the complexity of feature extraction, computational efficiency, and model performance. Our results confirm that using these 80,000 surface-annotated proteins achieves strong outcomes with manageable computational demands (Table 4).

For fine-tuning and evaluation, We use the PRIDE benchmark (chq1155 2024) with 32,389 training proteins from CATH4.3 and 504 test proteins from CAMEO, ensuring structural diversity and difficulty.

During fine-tuning and evaluation, the model with the surface encoder was trained only on samples with available surface information, or it just used the backbone information. The backbone-only model and baselines were trained on the full dataset without such restrictions.

During inference, we perform sampling to generate sequences. We use a temperature of 0.1 and apply top-k sampling with $k = 10$ to control the diversity of the generated

Model	0<len<100 ↑	100≤len<300 ↑	300≤len<500 ↑	Overall ↑
ProteinMPNN	41.63	48.61	52.07	48.44
PiFold	43.75	52.24	55.33	51.74
ESM-IF	39.73	52.72	58.64	51.85
GRADE-IF	44.06	51.64	57.29	51.90
InstructPLM	45.07	53.47	56.83	53.41
DS-ProGen (backbone-only)	43.14	55.18	60.27	52.61
DS-ProGen	**63.50**	**64.46**	**63.15**	**61.47**

Table 1: The average recovery rate (%) of DS-ProGen and baseline models on the PRIDE test set, grouped by sequence length. **Bold** indicates the best result in each column.

sequences while maintaining high fidelity to the predicted distribution.

Implementation Details Before training, we initialize the model parameters as follows: for the sequence decoder, we load the pretrained weights from ProGen2-small (Nijkamp et al. 2023); for the backbone encoder, we adopt the GVP and Transformer encoder parameters from ESM-IF (Hsu et al. 2022). During training, the backbone encoder is frozen to preserve its learned structural representations.

The overall model contains approximately 300M parameters. For the pretraining stage, we use a dataset of either 40 million or 80 thousand structure-sequence pairs. The model is trained for 1 epoch with a batch size of 16 using a single NVIDIA A100 GPU (80 GB memory) under FP32 precision. Training takes approximately 120 GPU-hours for the 40M dataset and 7 GPU-hours for the 80K dataset. The learning rate is set to 1×10^{-4} with the Adam optimizer, using a 5% warm-up phase followed by cosine annealing decay to zero.

For the fine-tuning stage, the model is trained for 2 epochs with a batch size of 8. The learning rate during fine-tuning is reduced to 2×10^{-5}, also using the Adam optimizer with cosine decay.

Evaluation Metrics

To evaluate the accuracy of sequence prediction, following previous research of inverse folding (Jing et al. 2020; Gao et al. 2024), we use the **recovery rate** as the primary metric. The recovery rate measures the proportion of residues where the predicted amino acid matches the ground-truth amino acid at the same position.

Formally, given a ground-truth sequence $\mathbf{y} = (y_1, y_2, \ldots, y_L)$ and a predicted sequence $\hat{\mathbf{y}} = (\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_L)$ of length L, the recovery rate R is defined as:

$$R = \frac{1}{L} \sum_{i=1}^{L} \mathbb{I}(y_i = \hat{y}_i), \tag{15}$$

where $\mathbb{I}(\cdot)$ is the indicator function, which equals 1 if the condition inside is true and 0 otherwise.

In addition to the recovery rate, to assess whether the generated sequences can correctly refold into the desired target structures, we further evaluate the predicted structures using AlphaFold3 (Abramson et al. 2024). Specifically, for each generated sequence, we predict its three-dimensional structure and compare it with the native target structure using two metrics: TM-Score (Zhang and Skolnick 2004) and Root-Mean-Square Deviation (RMSD). Higher TM-Score and lower RMSD indicate better structural quality.

Baselines

We compare our method against five models: ProteinMPNN (Dauparas et al. 2022), PiFold (Gao et al. 2022), ESM-IF (Hsu et al. 2022), GRADE-IF (Yi et al. 2023), and InstructPLM (Qiu et al. 2024). ProteinMPNN, PiFold, GRADE-IF, and InstructPLM are retrained on the PRIDE benchmark training set to ensure a fair comparison. For ESM-IF, we directly evaluate the released pretrained models without further fine-tuning. All baselines are based on their official implementations, with only minor modifications for environment compatibility.

Main Results

As shown in Table 1, DS-ProGen achieves the highest overall recovery rate of **61.47%** on the PRIDE test set, consistently outperforming all baselines across sequence length ranges. The improvement is particularly notable on short sequences (len<100), where DS-ProGen reaches **63.50%**, nearly 20 percentage points higher than the second-best model InstructPLM (45.07%). This highlights the value of surface features in low-resolution backbone graphs, where node sparsity limits the capacity of graph-based encoders to extract representations.

Moreover, as shown in Table 1, DS-ProGen not only achieves a higher median recovery rate but also exhibits a narrower distribution across test samples, suggesting greater robustness and reduced variance compared to the baselines.

Regarding structural fidelity, Table 2 shows that while DS-ProGen achieves the lowest RMSD among all models, its TM-Score is marginally lower than those of PiFold and ESM-IF. We note that the structures used for evaluation are predicted by AlphaFold3, which, despite its high accuracy, still introduces prediction noise. Therefore, differences in TM-Score or RMSD do not definitively indicate superiority or inferiority in structural realism or design quality of the generated sequences.

Despite achieving a high recovery rate and structural similarity, the designed sequences exhibit only 61.9% sequence

Model	TM-Score (%) ↑		RMSD (Å) ↓	
	Mean	Median	Mean	Median
ProteinMPNN	87.58	94.72	1.935	0.7477
PiFold	**88.43**	95.11	1.859	0.6448
ESM-IF	88.38	**95.58**	1.652	0.5811
DS-ProGen (backbone-only)	86.11	94.03	1.780	0.6087
DS-ProGen	87.30	94.88	**1.401**	**0.5575**

Table 2: TM-Score and RMSD measure between AlphaFold3-predicted structures from model-generated sequences and target structures, PRIDE test set. **Bold** indicates the best result in each column.

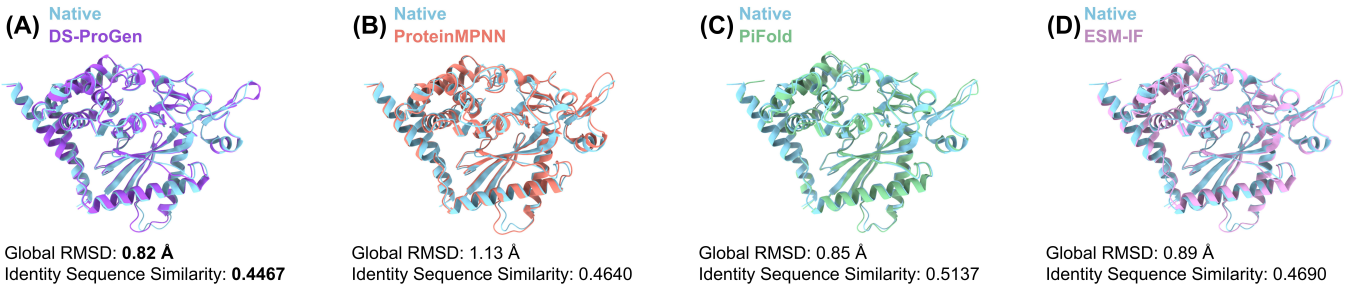

Figure 3: Design cases for DS-ProGen and baselines (Length is 403). A lower **RMSD** indicates greater structural similarity, while a lower **Identity Sequence Similarity** suggests stronger diversity in the designed sequences.

similarity (Figure 5), indicating significant diversity. The overall trends are encouraging. DS-ProGen's performance aligns closely with the top-performing baselines, and the uniformly low RMSD values across the test set suggest our model reliably generates sequences that fold into physically plausible structures (as illustrated in Figure 3). This underscores the benefit of incorporating surface information, which not only boosts sequence recovery but also maintains or enhances structural compatibility, especially regarding local precision and foldability.

Ablation Study

To systematically assess the contributions of different model components, we conduct two sets of ablation studies within the DS-ProGen framework: (1) encoder design, and (2) pretraining strategy.

Encoder Design. We evaluate the impact of using the backbone encoder (b_enc) and surface encoder (s_enc). The results are summarized in Table 3.

Methods	Recovery Rate (%)
DS-ProGen	61.47
w/o s_enc	50.71
w/o b_enc	31.06

Table 3: Ablation study on the encoder design of DS-ProGen. All models are pretrained on the same 80k dataset with available surface information. Removing either the backbone encoder (b_enc) or the surface encoder (s_enc) results in performance degradation, highlighting their complementary contributions.

As shown in Table 3, jointly using both the backbone and surface encoders yields the highest recovery rate (61.47%). Removing the surface encoder (w/o s_enc) leads to a moderate performance drop, while removing the backbone encoder (w/o b_enc) causes a significant degradation. This indicates that backbone geometry provides strong global structure cues, while surface captures important local features, and the combination of both is critical for sequence design.

Pretraining Strategy. We further examine the importance of pretraining by comparing models trained with/ and w/o initialization from large structure-sequence pairs. The results are presented in Table 4.

Methods	Recovery Rate (%)
DS-ProGen (backbone-only)	52.61
w/o pretraining	33.62

Table 4: Ablation study on the effect of pretraining in DS-ProGen. Pretraining on 4M structure-sequence pairs significantly improves model performance compared to training from scratch, demonstrating its critical role in enhancing generalization and convergence.

As shown in Table 4, pretraining brings a substantial gain of nearly 19% in recovery rate (52.61% vs. 33.62%). This highlights the necessity of pretraining for establishing strong structure-aware sequence priors, which significantly accelerates convergence and boosts generalization.

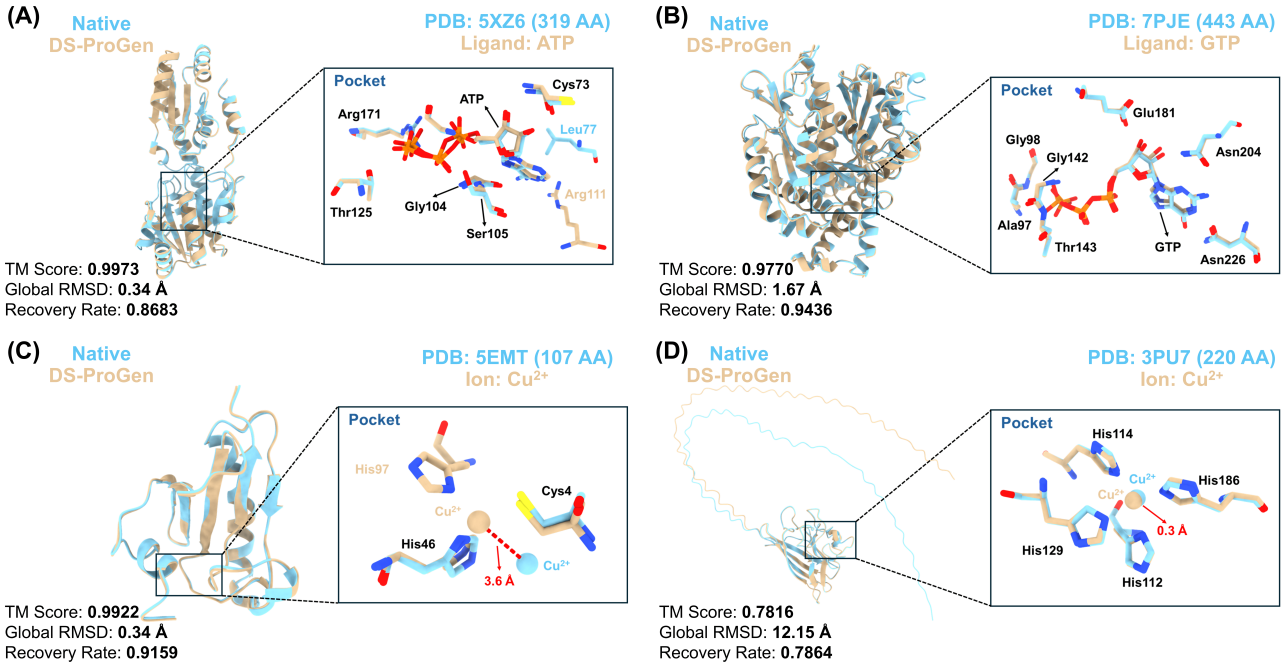

Figure 4: Structural alignment between ground truth structures (blue) and predicted structures (light brown) folded from protein sequences generated by DS-ProGen, visualized across four representative cases. Each panel highlights the global fold alignment and a zoomed-in view of the functional binding pocket, showing key residues together with the corresponding ligand or ion (e.g., ATP, GTP, Cu^{2+}). The predicted sequences lead to highly accurate structural models, with high TM-scores, low RMSDs, and strong recovery rates, demonstrating DS-ProGen's ability to preserve both global topology and local biochemical specificity.

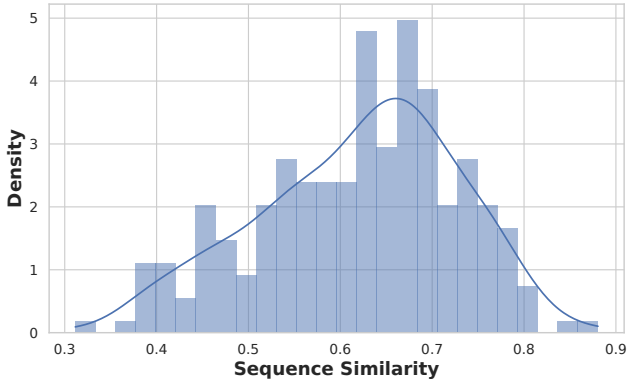

Figure 5: Histogram and kernel density estimation (KDE) plot of sequence similarity scores between generated sequence and groundtruth on PRIDE test set.

Downstream Tasks: Ligand and Ion-Protein Interaction Design

Beyond the standard inverse folding task, our further expectation is to design functional proteins with practical utility. Specifically, given the structure of a protein known to bind with a particular ligand or ion, we used DS-ProGen to generate compatible amino acid sequences. Molecular recognition between proteins and ligands drives many biological processes (Janin, Bahadur, and Chakrabarti 2008), while ion-protein interactions are always playing critical roles in biochemical processes (Roberts et al. 2015). To evaluate DS-ProGen in these contexts, we applied it to design sequences for both ligand- and ion-binding proteins, then predicted their 3D structures using AlphaFold3 and employed UCSF ChimeraX (Pettersen et al. 2021) for visualizing structural similarity between the ground truth and the predicted structures. In both tasks, as illustrated in Figure 4, the generated sequences consistently yield highly accurate structural models, reflected by high TM-scores, low RMSDs and strong recovery rates, demonstrating that DS-ProGen can effectively design proteins with both strong binding capabilities and high structural fidelity to their targets.

Conclusion and Outlook

We introduce DS-ProGen, a dual-structure generative model for protein design that utilizes both backbone and surface features. By leveraging geometric and chemical cues, it surpasses single-modality baselines with a 61.47% recovery rate on the PRIDE benchmark and excels in tasks like ligand and ion interaction design.

Looking ahead, future work will focus on scaling the training data and applying DS-ProGen to more complex design challenges, such as engineering enzymes and antibodies. We also plan to integrate protein dynamics to better model function. Ultimately, experimental validation of the designed proteins will be a critical next step to confirm their real-world efficacy and therapeutic potential.

Acknowledgments

We want to thank our anonymous AC, SPC and reviewers for their feedback. This work was supported by the Chinese University of Hong Kong (CUHK; award numbers 4937025, 4937026, 5501517, 5501329, 8601603, 8601663, and SHIAE BME-p1-24 to Y.L.); the Research Grants Council of the Hong Kong Special Administrative Region, China (Hong Kong SAR; project no. CUHK 24204023 and 14208525 to Y.L.); and the Innovation and Technology Commission of the Hong Kong SAR, China (project numbers GHP/065/21SZ, ITS/247/23FP and PRP/033/24FX to Y.L.).

References

Abramson, J.; Adler, J.; Dunger, J.; Evans, R.; Green, T.; Pritzel, A.; Ronneberger, O.; Willmore, L.; Ballard, A. J.; Bambrick, J.; Bodenstein, S. W.; Evans, D. A.; Hung, C.-C.; O'Neill, M.; Reiman, D.; Tunyasuvunakool, K.; Wu, Z.; Žemgulytė, A.; Arvaniti, E.; Beattie, C.; Bertolli, O.; Bridgland, A.; Cherepanov, A.; Congreve, M.; Cowen-Rivers, A. I.; Cowie, A.; Figurnov, M.; Fuchs, F. B.; Gladman, H.; Jain, R.; Khan, Y. A.; Low, C. M. R.; Perlin, K.; Potapenko, A.; Savy, P.; Singh, S.; Stecula, A.; Thillaisundaram, A.; Tong, C.; Yakneen, S.; Zhong, E. D.; Zielinski, M.; Žídek, A.; Bapst, V.; Kohli, P.; Jaderberg, M.; Hassabis, D.; and Jumper, J. M. 2024. Accurate structure prediction of biomolecular interactions with AlphaFold 3. *Nature*, 630(8016): 493—-500.

Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Batsanov, S. S. 2001. Van der Waals radii of elements. *Inorganic materials*, 37(9): 871–885.

Blinn, J. F. 1982. A generalization of algebraic surface drawing. *ACM transactions on graphics (TOG)*, 1(3): 235–256.

Cao, L.; Coventry, B.; Goreshnik, I.; Huang, B.; Sheffler, W.; Park, J. S.; Jude, K. M.; Marković, I.; Kadam, R. U.; Verschueren, K. H.; et al. 2022. Design of protein-binding proteins from the target structure alone. *Nature*, 605(7910): 551–560.

chq1155. 2024. PRIDE_Benchmark_ProteinDesign. github.com/chq1155/PRIDE_Benchmark_ProteinDesign. Accessed: 2025-04-26.

Das, R.; and Baker, D. 2008. Macromolecular modeling with rosetta. *Annu. Rev. Biochem.*, 77(1): 363–382.

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. 2022. Robust deep learning–based protein sequence design using ProteinMPNN. *Science*, 378(6615): 49–56.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, 4171–4186.

Eldar, Y.; Lindenbaum, M.; Porat, M.; and Zeevi, Y. Y. 1997. The farthest point strategy for progressive image sampling. *IEEE transactions on image processing*, 6(9): 1305–1315.

Emonts, J.; and Buyel, J. F. 2023. An overview of descriptors to capture protein properties–Tools and perspectives in the context of QSAR modeling. *Computational and structural biotechnology journal*, 21: 3234–3247.

Ferruz, N.; Schmidt, S.; and Höcker, B. 2022. ProtGPT2 is a deep unsupervised language model for protein design. *Nature communications*, 13(1): 4348.

Gao, Z.; Tan, C.; Chacón, P.; and Li, S. Z. 2022. PiFold: Toward effective and efficient protein inverse folding. *arXiv preprint arXiv:2209.12643*.

Gao, Z.; Tan, C.; Zhang, Y.; Chen, X.; Wu, L.; and Li, S. Z. 2023. Proteininvbench: Benchmarking protein inverse folding on diverse tasks, models, and metrics. *Advances in Neural Information Processing Systems*, 36: 68207–68220.

Gao, Z.; Tan, C.; Zhang, Y.; Chen, X.; Wu, L.; and Li, S. Z. 2024. Proteininvbench: Benchmarking protein inverse folding on diverse tasks, models, and metrics. *Advances in Neural Information Processing Systems*, 36.

Gu, Y.; Tinn, R.; Cheng, H.; Lucas, M.; Usuyama, N.; Liu, X.; Naumann, T.; Gao, J.; and Poon, H. 2021. Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare (HEALTH)*, 3(1): 1–23.

Hayes, T.; Rao, R.; Akin, H.; Sofroniew, N. J.; Oktay, D.; Lin, Z.; Verkuil, R.; Tran, V. Q.; Deaton, J.; Wiggert, M.; et al. 2025. Simulating 500 million years of evolution with a language model. *Science*, eads0018.

Heinzinger, M.; Weissenow, K.; Sanchez, J. G.; Henkel, A.; Mirdita, M.; Steinegger, M.; and Rost, B. 2024. Bilingual language model for protein sequence and structure. *NAR Genomics and Bioinformatics*, 6(4): lqae150.

Hsu, C.; Verkuil, R.; Liu, J.; Lin, Z.; Hie, B.; Sercu, T.; Lerer, A.; and Rives, A. 2022. Learning inverse folding from millions of predicted structures. In *International conference on machine learning*, 8946–8970. PMLR.

Janin, J.; Bahadur, R. P.; and Chakrabarti, P. 2008. Protein–protein interaction and quaternary structure. *Quarterly reviews of biophysics*, 41(2): 133–180.

Jendrusch, M.; Korbel, J. O.; and Sadiq, S. K. 2021. AlphaDesign: A de novo protein design framework based on AlphaFold. *Biorxiv*, 2021–10.

Jiang, J.; Chen, P.; Wang, J.; He, D.; Wei, Z.; Hong, L.; Zong, L.; Wang, S.; Yu, Q.; Ma, Z.; et al. 2025a. Benchmarking Large Language Models on Multiple Tasks in Bioinformatics NLP with Prompting. *arXiv preprint arXiv:2503.04013*.

Jiang, J.; Li, Y.; Cao, S.; Shan, Y.; Liu, Y.; Fei, T.; Yu, Y.; Feng, Y.; Li, Y.; Li, Y.; and Yuan, J. 2025b. Artificial intelligence in bioinformatics: a survey. *Briefings in Bioinformatics*, 26(6): bbaf576.

Jiang, J.; Wang, S.; Li, Q.; Kong, L.; and Wu, C. 2023. A Cognitive Stimulation Dialogue System with Multi-source Knowledge Fusion for Elders with Cognitive Impairment.

In Rogers, A.; Boyd-Graber, J.; and Okazaki, N., eds., *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 10628–10640. Toronto, Canada: Association for Computational Linguistics.

Jiang, J.; Wang, Z.; Shan, Y.; Chai, H.; Li, J.; Ma, Z.; Zhang, X.; and Li, Y. 2025c. Biological Sequence with Language Model Prompting: A Survey. *arXiv preprint arXiv:2503.04135*.

Jing, B.; Eismann, S.; Suriana, P.; Townshend, R. J.; and Dror, R. 2020. Learning from protein structure with geometric vector perceptrons. *arXiv preprint arXiv:2009.01411*.

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

Kim, H.; Mirdita, M.; and Steinegger, M. 2023. Foldcomp: a library and format for compressing and indexing large protein structure sets. *Bioinformatics*, 39(4): btad153.

Lin, Z.; Akin, H.; Rao, R.; Hie, B.; Zhu, Z.; Lu, W.; Smetanin, N.; Verkuil, R.; Kabeli, O.; Shmueli, Y.; et al. 2023. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637): 1123–1130.

Liu, A.; Feng, B.; Xue, B.; Wang, B.; Wu, B.; Lu, C.; Zhao, C.; Deng, C.; Zhang, C.; Ruan, C.; et al. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Madani, A.; Krause, B.; Greene, E. R.; Subramanian, S.; Mohr, B. P.; Holton, J. M.; Olmos, J. L.; Xiong, C.; Sun, Z. Z.; Socher, R.; et al. 2023. Large language models generate functional protein sequences across diverse families. *Nature Biotechnology*, 41(8): 1099–1106.

Nijkamp, E.; Ruffolo, J. A.; Weinstein, E. N.; Naik, N.; and Madani, A. 2023. Progen2: exploring the boundaries of protein language models. *Cell systems*, 14(11): 968–978.

Pettersen, E. F.; Goddard, T. D.; Huang, C. C.; Meng, E. C.; Couch, G. S.; Croll, T. I.; Morris, J. H.; and Ferrin, T. E. 2021. UCSF ChimeraX: Structure visualization for researchers, educators, and developers. *Protein Science*, 30(1): 70–82.

Qiu, J.; Xu, J.; Hu, J.; Cao, H.; Hou, L.; Gao, Z.; Zhou, X.; Li, A.; Li, X.; Cui, B.; et al. 2024. Instructplm: Aligning protein language models to follow protein structure instructions. *bioRxiv*, 2024–04.

Radford, A.; Narasimhan, K.; Salimans, T.; Sutskever, I.; et al. 2018. Improving language understanding by generative pre-training.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I.; et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8): 9.

Roberts, D.; Keeling, R.; Tracka, M.; Van Der Walle, C.; Uddin, S.; Warwicker, J.; and Curtis, R. 2015. Specific ion and buffer effects on protein–protein interactions of a monoclonal antibody. *Molecular pharmaceutics*, 12(1): 179–193.

Song, Z.; Huang, T.; Li, L.; and Jin, W. 2024. Surfpro: Functional protein design based on continuous surface. *arXiv preprint arXiv:2405.06693*.

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

Tolstikhin, I. O.; Houlsby, N.; Kolesnikov, A.; Beyer, L.; Zhai, X.; Unterthiner, T.; Yung, J.; Steiner, A.; Keysers, D.; Uszkoreit, J.; et al. 2021. Mlp-mixer: An all-mlp architecture for vision. *Advances in neural information processing systems*, 34: 24261–24272.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Wang, Z.; Wang, Z.; Jiang, J.; Chen, P.; Shi, X.; and Li, Y. 2025. Large Language Models in Bioinformatics: A Survey. *arXiv preprint arXiv:2503.04490*.

Xia, Y.; Jin, P.; Xie, S.; He, L.; Cao, C.; Luo, R.; Liu, G.; Wang, Y.; Liu, Z.; Chen, Y.-J.; et al. 2025. NatureLM: Deciphering the Language of Nature for Scientific Discovery. *arXiv preprint arXiv:2502.07527*.

Xiao, Y.; Sun, E.; Jin, Y.; Wang, Q.; and Wang, W. 2024. Proteingpt: Multimodal llm for protein property prediction and structure understanding. *arXiv preprint arXiv:2408.11363*.

Yi, K.; Zhou, B.; Shen, Y.; Liò, P.; and Wang, Y. 2023. Graph denoising diffusion for inverse protein folding. *Advances in Neural Information Processing Systems*, 36: 10238–10257.

Yuan, M.; Shen, A.; Fu, K.; Guan, J.; Ma, Y.; Qiao, Q.; and Wang, M. 2023. ProteinMAE: masked autoencoder for protein surface self-supervised learning. *Bioinformatics*, 39(12): btad724.

Yue, K.; and Dill, K. A. 1992. Inverse protein folding problem: designing polymer sequences. *Proceedings of the National Academy of Sciences*, 89(9): 4163–4167.

Zhang, G.; Li, Y.; Luo, R.; Hu, P.; Zhao, Z.; Li, L.; Liu, G.; Wang, Z.; Bi, R.; Gao, K.; et al. 2025. UniGenX: Unified Generation of Sequence and Structure with Autoregressive Diffusion. *arXiv preprint arXiv:2503.06687*.

Zhang, Y.; and Skolnick, J. 2004. Scoring function for automated assessment of protein structure template quality. *Proteins: Structure, Function, and Bioinformatics*, 57(4): 702–710.

Zhang, Z. 2016. Introduction to machine learning: k-nearest neighbors. *Annals of translational medicine*, 4(11).

Zhou, X.; Chen, G.; Ye, J.; Wang, E.; Zhang, J.; Mao, C.; Li, Z.; Hao, J.; Huang, X.; Tang, J.; et al. 2023. ProRefiner: an entropy-based refining strategy for inverse protein folding with global graph attention. *Nature Communications*, 14(1): 7434.