Visual-like Template Diffusion: Boosting Single-Sequence Protein Structure Prediction by Adapting Image Diffusion Models

Xudong Wang[1†], Tong Zhang[1†], Zhen Cui[2*], Xu Guo[1], Fuyun Wang[1], Yuanzhi Wang[1], Xing Cai[1] and Wenming Zheng[3]

[1]School of Computer Science and Engineering, Nanjing University of Science and Technology, No. 200, Xiaolingwei Street, Nanjing, 210094, Jiangsu Province, China.
[2]School of Artificial Intelligence, Beijing Normal University, No. 19, Xinjiekou Outer Street, Beijing, 100875, China.
[3]School of Biological Science & Medical Engineering, Southeast University, No. 2, Sipailou, Nanjing, 210096, Jiangsu Province, China.

*Corresponding author(s). E-mail(s): zhen.cui@bnu.edu.cn;
Contributing authors: xd_wang@njust.edu.cn; tong.zhang@njust.edu.cn; xu.guo@njust.edu.cn; fyw271828@njust.edu.cn; yuanzhiwang@njust.edu.cn; caixing11@163.com; wenming_zheng@seu.edu.cn ;
†These authors contributed equally to this work.

Abstract

Single-sequence protein structure prediction has drawn increasing attention due to the high computational costs associated with obtaining homologous information. Here, we propose *a visual-like template diffusion method, named TDFold, to achieve accurate and highly efficient single-sequence 3D structure prediction for proteins.* Given a protein sequence, TDFold initially generates high-quality inter-residue geometries (distances and orientations) as templates from a probabilistic diffusion perspective. Since inter-residue geometries can be encoded as multi-channel feature matrices (each channel for either the inter-residue distance or orientation correlation), analogous to image feature

maps, we construct an image-level template diffusion module by adapting the stable diffusion (SD) model from text-vision generation to sequence-template diffusion for proteins. Subsequently, a lightweight sequence-template co-evolved learning (SCL) network is constructed to facilitate accurate and efficient protein structure prediction. As a result, TDFold possesses three highlights: (i) *better single-sequence prediction performance*: TDFold greatly outperforms existing protein language models (PLMs, e.g. ESMFold and OmegaFold) and homology-based methods (e.g. AlphaFold2, AlphaFold3 and RoseTTAFold) on homology-insufficient datasets such as Orphan and Orphan25, while also achieving promising results on the popular CASP14, CASP15 and CASP16 benchmarks; (ii) *low resource consumption*: By utilizing the lightweight SCL architecture, the GPU memory consumption of TDFold is generally lower than that of popular methods such as AlphaFold2 and ESMFold; (iii) *higher efficiency in training and inference*: TDFold can be trained within a week using a single NVIDIA 4090 GPU. Furthermore, the inference time of TDFold is significantly shorter (about 10x to 100x) than that of existing methods (ESMFold, AlphaFold2 and AlphaFold3) for long protein sequences. This work demonstrates the effectiveness of leveraging powerful vision diffusion models to enhance protein template generation, thereby establishing a new paradigm for single-sequence protein structure prediction. It also accelerates protein-related research, particularly for resource-limited universities and academic institutions. The code has been released to speed up biological research.

1 Introduction

In recent years, significant advancements have been achieved in AI-based protein structure prediction [1–9]. Notably, AlphaFold2 [1], AlphaFold3 [10] and RoseTTAFold [2] have demonstrated exceptional performance on the CASP14 dataset [11], marking a new milestone in the field. However, these deep models heavily rely on homologous information, including multiple sequence alignments (MSAs) [12] and templates [13], which are typically searched in biological databases such as UniRef [14] and PDB [15]. This reliance results in low accuracy of homology-based methods for proteins with few or limited homologous information. As illustrated in Fig. 1, templates are essential for defining the relative positions of residues, and the exclusion of template information leads to significant performance degradation for AlphaFold2 and RoseTTAFold. Additionally, obtaining MSAs and templates from biological databases is time-consuming and often hindered by scarce or absent homology information, further reducing the efficiency of protein structure prediction. To address these challenges, deep learning methods based on protein language models (PLMs) [16–20], such as ESMFold [16], trRosettaX-single [17], RGN2 [18], and OmegaFold [19], have recently emerged. These approaches leverage only the

textual context of amino acid sequences, eliminating the need for homologous information and significantly accelerating the protein feature extraction process.

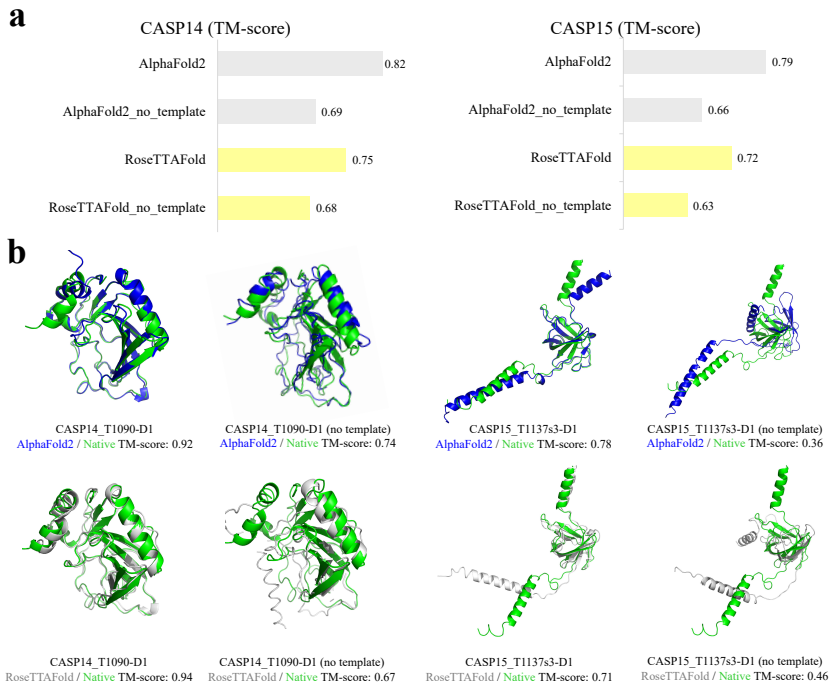

Fig. 1 The importance of template information to AlphaFold2's and RoseTTAFold's protein structure prediction performance on the CASP14 and CASP15 datasets. a, TM-score difference between two popular models (AlphaFold2, RoseTTAFold) and their "no template" versions on CASP14 and CASP15 datasets. **b,** The visualization results of 3D structure comparison between the original and "no template" versions. The AlphaFold2's predicted structures, the RoseTTAFold's predicted structures and the native structures are painted in blue, gray, and green, respectively.

Despite the significant advancements achieved by protein language model (PLM)-based methods, two critical challenges persist in enhancing the research and application of single-sequence protein structure prediction. First, the underutilization of template structural information during model training limits the structure modeling capabilities of current single-sequence prediction methods. Specifically, these methods either completely ignore the correlations between protein sequences and inter-residue geometries or only learn pseudo inter-residue geometries (e.g., inter-residue attention scores derived by ESMFold) without native structure supervision. This poses a significant challenge in inferring intricate three-dimensional structures solely from the sequential context of residues. Second, existing PLM-based methods adopt a large-scale architecture derived from language models, which consists of deeply stacked

transformer blocks [21]. Notably, the time complexity of the triangular attention mechanism used in ESMFold's evoformer is ($O(N^3)$, where N denotes the sequence length. This quadratic scaling relationship makes such methods resource-intensive and computationally expensive, leading to substantial memory requirements and computational overhead, particularly for long-sequence proteins.

In this study, we present TDFold, a novel single-sequence prediction model designed to accurately predict the three-dimensional structures of proteins. Given an amino acid sequence, TDFold infers its 3D structure through an end-to-end network architecture consisting of two distinct stages: template diffusion and Sequence-template Co-evolved Learning (SCL). Specifically, the template diffusion stage aims to derive high-quality inter-residue geometries (e.g., matrices of inter-residue distances and orientations) as templates to enhance single-sequence structure prediction. Considering that inter-residue geometries can be encoded as multi-channel feature matrices, analogous to image feature maps [22], with distinct channels capturing distance and orientation correlations, we adapted the Stable Diffusion (SD) framework [23], originally designed for text-to-image synthesis, to develop a computational method for sequence-to-template diffusion in protein structure modeling. Concretely, a protein template diffusion module is proposed by constructing two Low-Rank Adaptation (LoRA) [24] branches attached to the text and image (i.e., UNet [25]) encoders in the SD model. Leveraging the SD model's powerful ability to model correlations between textual sequences and multi-channel matrices, our template diffusion module enables reliable inter-residue geometry generation (i.e., "images" describing inter-residue distances and orientations) by treating given amino acid sequences as "text prompts". In the subsequent SCL stage, a lightweight co-evolved graph network [26] is constructed to predict protein structures by fusing a residue-level branch and another atom-level one. Specifically, the residue-level learning branch models sequence and inter-residue correlations in a co-evolved manner, while the atom-level graph learning branch captures the influence of side-chain atoms on backbone conformations within an amino acid sequence. These two branches work collaboratively to derive the final 3D coordinates of proteins. Compared to the PLMs like ESMFold, TDFold offers three key advantages: (i) Superior single-sequence prediction performance: By leveraging the powerful text-to-image generation capability of the SD model and using real inter-residue geometries as supervised information, TDFold generates high-quality inter-residue geometries as intermediate features. This eliminates dependence on homology data and significantly improves the performance of single-sequence protein structure prediction. (ii) Low resource consumption: By adopting the lightweight SCL architecture, TDFold's GPU memory consumption is generally lower than that of popular protein structure prediction methods such as AlphaFold2 and ESMFold. (iii) Higher efficiency in training and inference: TDFold can be trained within one week using a single NVIDIA 4090 GPU, including fine-tuning the

SD model and training the SCL network from scratch. Furthermore, the inference time of the stable diffusion model is significantly shorter (approximately 10x to 100x) than that of PLM-based methods, particularly for long-sequence proteins.

We report the performance on the homology-insufficient Orphan [18] and Orphan25 [17] datasets, as well as the popular CASP14, CASP15 and CASP16 datasets (Critical Assessment of Techniques for Protein Structure Prediction [11, 27, 28]). The experimental results demonstrate that our TDFold achieves state-of-the-art performance on the homology-insufficient datasets Orphan and Orphan25, as well as competitive performance (measured by TM-score [29], GDT_TS [30] and pLDDT [1, 31]) compared to ESMFold and OmegaFold on CASP14, CASP15 and CASP16. Furthermore, we provide an analysis of the computational time and GPU memory requirements for AlphaFold2, RoseTTAFold, ESMFold, and TDFold in protein structure prediction. TDFold requires approximately 10 seconds to predict the structure of a protein containing 500 residues, while ESMFold requires about 100 seconds, AlphaFold3 needs about 240 seconds, AlphaFold2 and RoseTTAFold require nearly 1000 seconds. In terms of GPU memory usage, TDFold occupies about 7 GB, whereas AlphaFold2, RoseTTAFold, and ESMFold require 12 GB, 16 GB, and 20 GB, respectively (due to the lack of open training weights for AlphaFold3, we used the AlphaFold3's server for testing and were unable to obtain its GPU memory usage data). Additionally, we visualize multiple cases of TDFold's predicted 3D structures and compare the generated inter-residue distances with those derived from homologous templates searched in biological databases. These experiments and model analyses collectively demonstrate that TDFold achieves the best protein structure prediction performance in a single-sequence-based mode while requiring the least memory and time for model inference.

2 Results

We firstly summarize the framework of the proposed TDFold. Then, we conduct comprehensive experiments on multiple popular protein datasets, including Orphan, Orphan25, CASP14, CASP15 and CASP16. Finally, we analyze the effectiveness of each module in TDFold in the ablation study.

Approach Summary The overall framework of the proposed TDFold is depicted in Fig. 2. This model employs a two-stage architecture for three-dimensional structure prediction. Initially, it generates inter-residue geometric information using a template diffusion module, followed by a lightweight sequence-template co-evolved Learning (SCL) network to predict the 3D structure. Specifically, for a given amino acid sequence, the template diffusion module denoises Gaussian distributed noises to generate the inter-residue geometry images (i.e. distance and orientation matrices) while using the protein sequence as the text prompt. To address the significant datatype difference

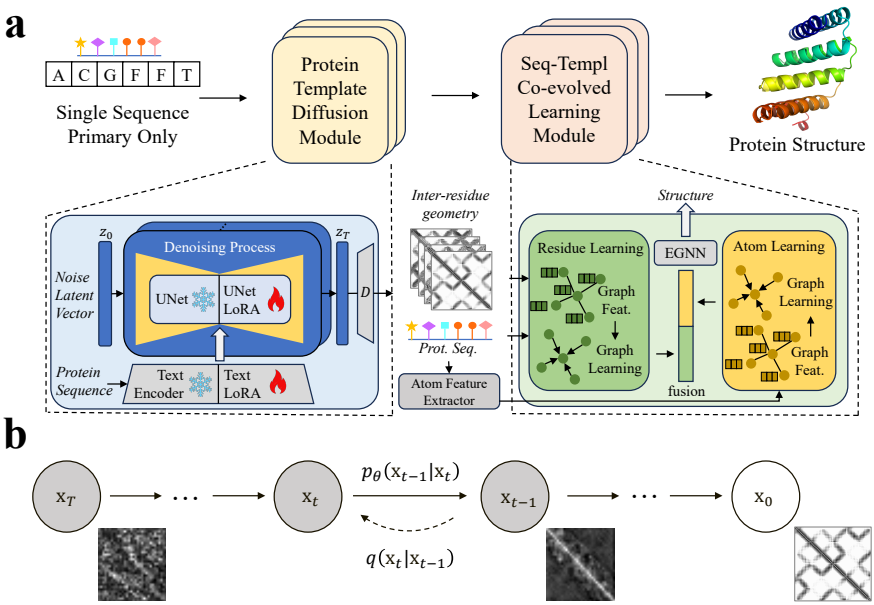

Fig. 2 The architecture and template diffusion process of TDFold. a, The architecture of TDFold for protein structure prediction. TDFold consists of two modules, the protein template diffusion module and the sequence-template co-evolved learning (SCL) module. For one given amino acid sequence, the protein template diffusion module generates inter-residue geometries (templates) guided by the sequence. Next, the sequence and generated inter-residue geometries are sent into the SCL module. The SCL learning module has a two-branch architecture to learn residue-level and atom-level features, respectively. Finally, the residue and atom features are fused and sent into the SE(3) equivariant graph neural network (EGNN) to predict the 3D structure. **b,** The forward and reverse process of the inter-residue geometry diffusion in TDFold. The forward process q of is a Markov chain that continuously adds Gaussian noises until the original image signal is covered by the noise signal. The reverse process p_θ learns to gradually denoise a normally distributed variable and restore the inter-residue geometry image.

(e.g. continuous geometries vs. discrete images) and semantic gap (e.g. protein geometry information vs. general images), we firstly discretize the value of the inter-residue geometric matrices and map them to the pixel value range (0-255) of the RGB image [32]. Then, we apply the Low-Rank Adaptation (LoRA) [24] fine-teuning technique to the stable diffusion (SD) model [23]. Specifically, the original training parameters of the SD model are frozen, and only the parameters of LoRA are trained in the fine-tuning process. For text encoder [33], we take a global LoRA to align sequence features with inter-residue geometric image features, mapping them into a shared latent space. For the UNet [25], we utilize four local LoRA focusing on learning the matrices of the distance of $C_\beta - C_\beta$, two dihedrals ω (C_α-C_β-C_β-C_α), θ (N-C_α-C_β-C_β) and a planar angle ϕ: C_α-C_β-C_β, to enhance the model's capacity to accurately learn the inter-residue geometries. With the help of LoRA fine-tuning technique, we transfer the SD model with powerful image generation capability to the inter-residue

geometries generation task. Through the denoising UNet, the SD model is able to capture the complex distribution of inter-residue geometric images and generate reliable samples as homologous templates in the prediction process.

Next, we establish a two-branch SCL network comprising residue-level and atom-level branches for the protein 3D structure prediction. The residue-level branch aims to effectively learn sequence-template interaction by integrating sequence and template (inter-residue geometric information) in a co-evolved manner, updating both the residue features and paired inter-residue relationships. The obtained residue features and inter-residue representations are then treated as nodes and edges, respectively, thereby constructing a residue-level graph. Subsequently, we employ the residue graph neural network [34] to learn geometric representations by aggregating residue features based on the inter-residue edges. The atom-level branch additionally incorporates sidechain effects, constructing an atom-level graph by treating atoms and bonds as nodes and edges. This representation is learned through the atomic graph neural network [35], which aggregates features from both backbone and sidechain atoms. The atom-level branch is then fused with the residue-level branch through variational learning, allowing for the integration of sidechain awareness into backbone generation. This fusion process enables adaptive refinement of the residue-level representation by considering the influence of atom-level variations. Following this two-branch fusion, the resulting features are input into the SE(3) equivariant graph neural network (EGNN) [36] for predicting the protein 3D structures.

Predicting structures of orphan proteins. To evaluate our TDFold to orphan proteins, we test the performance on Orphan and Orphan25 datasets which typically contain limited or no homologous information. Also, we compare the results with the three state-of-the-art methods named AlphaFold2 [1], AlphaFold3 [10] and RoseTTAFold [2], as well as four protein language model (PLM) based methods: ESMFold, OmegaFold, RGN2 and trRosettaX-single [16–19]. Specifically, only single sequence mode is tested for PLM-based methods and our TDFold, while both modes of using single sequence and full homology (including MSA and templates) are tested for AlphaFold2 and RoseTTAFold (only full homology mode for AlphaFold3). For performance measurement, the widely adopted TM-score [29], GDT_TS [30] and pLDDT [1, 31] are used as the metric for protein structure prediction.

As shown in Fig. 3a, our TDFold outperforms all these comparison methods on both Orphan and Orphan25 datasets. Specifically, for homology-based methods (AlphaFold2, AlphaFold3, RoseTTAFold), our TDFold outperforms them even though they use the full MSA and templates (full MT). This is because the homology-based methods highly rely on homologous information as input, while orphan proteins have rather limited homology (MSAs and templates). Among the PLM-based methods (ESMFold, OmegaFold, RGN2, and trRosettaX single), the structure prediction performance of ESMFold is better than the other three. Compared with them, TDFold achieves the best

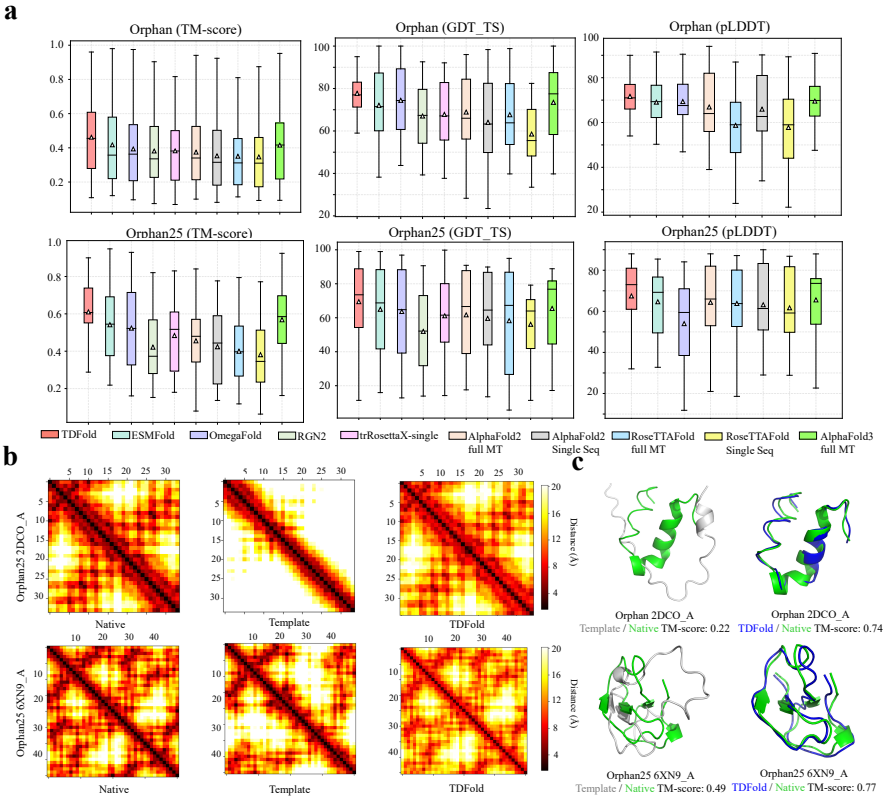

Fig. 3 The performance comparison on the Orphan and Orphan25 datasets and the visualization examples of the searched templates and TDFold's predictions. a, The performances (TM-score, GDT_TS and pLDDT) on the Orphan and Orphan25 datasets of TDFold, ESMFold, OmegaFold, RGN2, trRosettaX-single, AlphaFold2 (full MSA and Templates, full MT), AlphaFold2 (Single Seq), RoseTTAFold (full MT), RoseTTAFold (Single Seq) and AlphaFold3 (full MT) (since RGN2 and trRosettaX-single do not support the calculation of pLDDT, the pLDDT results shown in the figure exclude theirs.). For each box in the figure, the center line, bottom line, and top line represent the median, first quartile, and third quartile, respectively. The horizontal lines along the top and bottom edges represent the maximum and minimum observations. Besides, the white triangle represents the average value.**b,**The inter-residue distance matrices for 2DCO_A (Orphan) and 6XN9_A (Orphan25) generated from TDFold and searched template structures. Each data point in the distance image corresponds to a pair of residues. The distances are represented by a color gradient and the darker shades indicate closer proximity. **c,** The TDFold's prediction (in blue), template's structure (in gray) and the native structure (in green) of PDB 2DCO_A of Orphan and PDB 6XN9_A of Orphan25.

performances and outperforms ESMFold with the TM-score gains of 0.04 on Orphan and 0.07 on Orphan25. This may be attributed to the fact that TDFold well establishes a distribution mapping between one single protein sequence and the corresponding inter-residue geometries, which effectively boosts the protein structure prediction. Concretely, on the Orphan dataset, the average TM-score of TDFold is 0.46, while that of ESMFold is 0.42, OmegaFold is 0.39,

AlphaFold2 (full MT) is 0.37, AlphaFold3 (full MT) is 0.41, and RoseTTAFold (full MT) is 0.35. And for the Orphan25 dataset, the TM-score values are 0.61 of TDFold, 0.54 of ESMFold, 0.52 of OmegaFold, 0.44 of AlphaFold2 (full MT), 0.57 of AlphaFold3 (full MT), and 0.40 of RoseTTAFold (full MT).

For the another metric GDT_TS, on the Orphan dataset, the mean value of TDFold is 77.50, while that of ESMFold is 72.08, OmegaFold is 74.44, AlphaFold2 (full MT) is 68.91, AlphaFold3 (full MT) is 73.49, and RoseTTAFold (full MT) is 67.61. For the Orphan25 dataset, the GDT_TS values are 68.37 of TDFold, 64.93 of ESMFold, 63.65 of OmegaFold, 61.70 of AlphaFold2 (full MT), 65.46 of AlphaFold3 (full MT), and 58.25 of RoseTTAFold (full MT). Moreover, we also employ the pLDDT metric to assess the confidence of the model's predicted structures. The results of TDFold, ESMFold, OmegaFold, AlphaFold2 (full MT), AlphaFold3 (full MT) and RoseTTAFold (full MT) on Orphan dataset are 71.85, 69.52, 68.75, 67.23, 69.55, and 65.96, respectively. On the Orphan25 dataset, the pLDDT values are 67.48 of TDFold, 64.65 of ESMFold, 54.25 of OmegaFold, 64.87 of AlphaFold2 (full MT), 65.57 of AlphaFold3 (full MT), and 63.25 of RoseTTAFold (full MT). According to the results, the pLDDT confidence scores of TDFold provide a good indication of the agreement with native structures.

Additionally, we provide the visualizations of the inter-residue distance matrices (2DCO_A from Orphan and 6XN9_A from Orphan25) of the searched template and TDFold in Fig. 3b. Meanwhile, the predicted structures of TDFold that use the searched and TDFold-generated templates as SCL's inputs are depicted in Fig. 3c. For 2DCO_A and 6XN9_A, only a few homologous templates can be searched, and they present obvious differences from the native structure according to Fig. 3b. In contrast, the inter-residue distance images generated by TDFold are both more similar with the native ones. Correspondingly, using the generated inter-residue geometries for prediction obtains much better performances than using the searched templates. As shown in Fig. 3c, the TM-scores of using generated inter-residue geometries are 0.74 (2DCO_A) and 0.77 (6XN9_A), which are much higher than 0.22 and 0.49 of using the searched ones.

Predicting protein structures in CASP14, CASP15 and CASP16. To evaluate the performance of the TDFold for general protein structure prediction, we compare it with two homology-based methods named AlphaFold2 [1], RoseTTAFold [2], and one PLM-based method called ESMFold [16] on the popular CASP14 and CASP15 datasets.

The experimental results are shown in Fig. 4a. Overall, our TDFold achieves high prediction performances (average TM-score, GDT_TS, and pLDDT) on the CASP14, CASP15, and CASP16 datasets. For AlphaFold2 and RoseTTAFold, they obtain the TM-scores of 0.80, 0.75 on CASP14, 0.79, 0.68 on CASP15, and 0.78, 0.76 on CASP16 with full MSAs and templates as inputs. However, when using single sequences as inputs, their performances degrade to the TM-scores of 0.46, 0.43 on CASP14, 0.51, 0.44 on CASP15, and

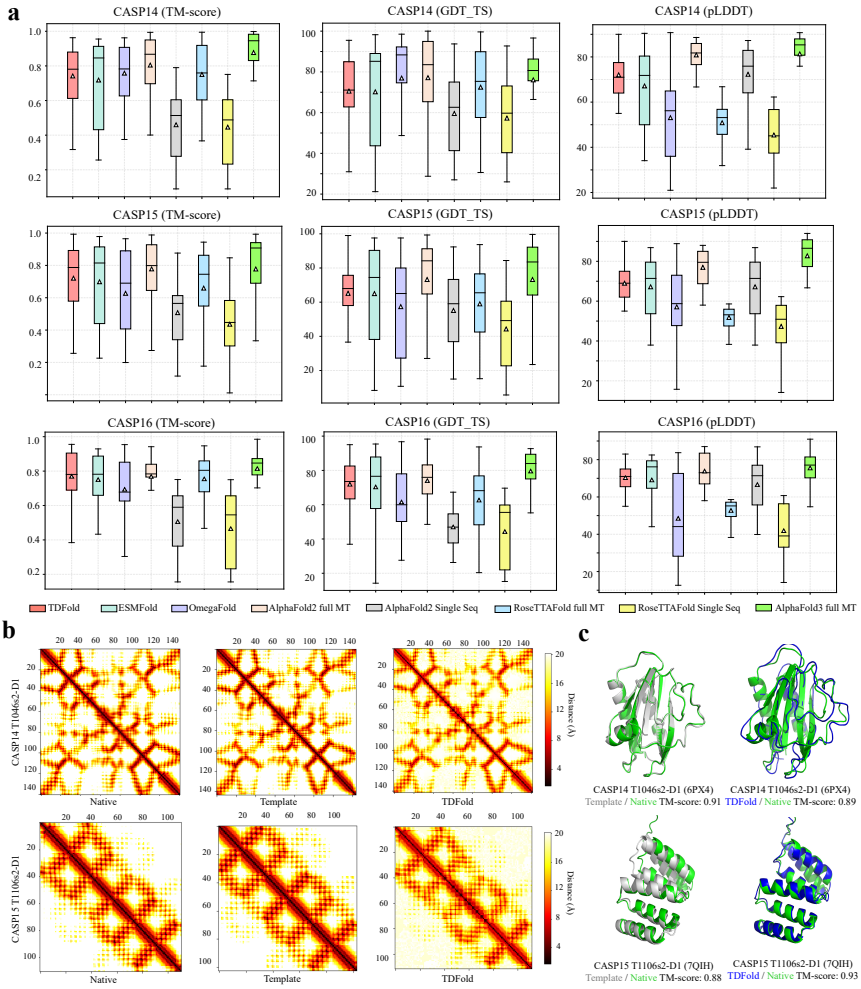

Fig. 4 The performance comparison on the CASP14, CASP15 and CASP16 datasets and the visualization examples of the searched templates and TDFold's predictions. a, The TM-score, GDT_TS, and pLDDT values of TDFold, ESMFold, OmegaFold, AlphaFold2 (full MT), AlphaFold2 (Single Seq), RoseTTAFold (full MT), RoseTTAFold (Single Seq) and AlphaFold3 (full MT) on the CASP14, CASP15 and CASP16 datasets. For each box in the figure, the center line, bottom line, and top line represent the median, first quartile, and third quartile, respectively. The horizontal lines along the top and bottom edges represent the maximum and minimum observations. Besides, the white triangle represents the average value. **b,** The inter-residue distance images generated by TDFold for T1046s2-D1 (PDB 6PX4) of CASP14 and T1106s2-D1 (PDB 7QIH) of CASP15 and searched template structures. Each data point in the distance image corresponds to a pair of residues. The distances are represented by a color gradient and the darker shades indicate closer proximity. **c,** The TDFold's prediction (in blue), template structure (in gray) and the native structure (in green) of T1046s2-D1 and T1106s2-D1.

0.50, 0.46 on CASP16, respectively. For PLM-based methods like ESMFold

and OmegaFold, our TDFold is better than ESMFold by obtaining 0.02 (0.73 of TDFold vs. 0.71 of ESMFold) TM-score performance gain on CASP14, 0.01 (0.7 of TDFold vs. 0.69 of ESMFold) gain on CASP15, and 0.02 (0.77 of TDFold vs. 0.75 of ESMFold) gain on CASP16. In addition, our method significantly outperforms OmegaFold by achieving 0.07 TM-score gain (0.7 of TDFold vs. 0.63 of OmegaFold) on the CASP15 dataset, 0.08 TM-score gain (0.77 of TDFold vs. 0.69 of OmegaFold) on the CASP16 dataset, while obtaining a performance that is 0.03 lower (0.73 of TDFold vs. 0.76 of OmegaFold) on the CASP14 dataset. It should be noticed that the training datasets of OmegaFold contain the protein sequences UniRef50 dataset (dated at 2021/04), and the structures from the Protein Databank PDB deposited before 2021, while the protein structures of CASP14 were released in the Protein Databank PDB from 2020/07. In contrast, the released training dataset dates of AlphaFold2, RoseTTAFold, ESMFold, and our TDFold are 2020/05.

To further comprehensively measure the performance, we also employ the GDT_TS and pLDDT metrics for evaluation. The GDT_TS results of TDFold, ESMFold, OmegaFold, AlphaFold2 (full MT), AlphaFold3 (full MT), and RoseTTAFold (full MT) on the CASP14 dataset are 70.86, 70.21, 75.35, 76.15, 77.08 and 72.07, respectively. On the CASP15 dataset, the GDT_TS values are 63.52 of TDFold, 62.99 of ESMFold, 57.37 of OmegaFold, 73.24 of AlphaFold2 (full MT), 73.26 of AlphaFold3 (full MT), and 60.35 of RoseTTAFold (full MT). And for the CASP16 dataset, the GDT_TS scores are 71.91 of TDFold, 70.33 of ESMFold, 61.55 of OmegaFold, 74.05 of AlphaFold2 (full MT), 79.59 of AlphaFold3 (full MT), and 62.68 of RoseTTAFold (full MT). Moreover, we also compare several methods with our TDFold using the pLDDT metric to assess the models' prediction confidence. On the CASP14 dataset, the mean value of TDFold is 72.06, while ESMFold is 67.14, OmegaFold is 53.25, AlphaFold2 (full MT) is 80.5, AlphaFold3 (full MT) is 81.35, and RoseTTAFold (full MT) is 72.3. For the CASP15 dataset, the pLDDT values are 69.6 of TDFold, 66.38 of ESMFold, 53.83 of OmegaFold, 79.25 of AlphaFold2 (full MT), 84.21 of AlphaFold3 (full MT), and 66.35 of RoseTTAFold (full MT). And on the CASP16 dataset, the pLDDT values are 70.33 of TDFold, 69.11 of ESMFold, 48.52 of OmegaFold, 73.86 of AlphaFold2 (full MT), 75.53 of AlphaFold3 (full MT), and 66.64 of RoseTTAFold (full MT). The approximate performances between the GDT_TS and pLDDT metrics further verify the good agreement with native structures of TDFold's predictions.

We visualize several inter-residue distance matrices for the proteins T1046s2-D1 (PDB code 6PX4) from CASP14 and T1106s2-D1 (PDB code 7QIH) from CASP15 in Fig. 4b. For T1046s2-D1 and T1106s2-D1, both the generated inter-residue distance images of TDFold and searched templates are closely aligned with the native ones. Accordingly, high TM-scores are obtained by using the searched templates (0.91 for T1046s2-D1 and 0.88 for T1106s2-D1) and generated ones (0.89 for T1046s2-D1 and 0.93 for T1106s2-D1). From the experimental results, it can be seen that our TDFold is able

Table 1 The TM-score of comparison methods and TDFold on virus-related proteins with a small amount of homologs

Protein	T1033-D1	T1039-D1	T1064-D1	T1082-D1	T1099-D1	T1123-D1
Type	crAs phage	crAs phage	SARS-CoV-2	T4 phage	hepatitis virus	astrovirus
Seq Len	100	161	75	71	178	214
MSA Num	3	3	14	11	10	9
AlphaFold2	0.41	0.58	0.41	0.4	0.8	0.62
AlphaFold3	0.44	0.47	0.71	0.73	0.85	0.28
ESMFold	0.33	0.28	0.44	0.38	0.47	0.31
TDFold	**0.94**	**0.89**	**0.73**	**0.76**	**0.87**	**0.79**

to generate high-quality inter-residue geometries comparable with those computed from the highly matched templates, which verifies the effectiveness of the designed paradigm of transferring the visual SD model to the protein template generation.

Promoting the structure prediction of virus-related proteins. Many viruses encode rapidly evolving proteins to evade host immune responses and enhance host-specific adaptation. Notable examples include coronaviruses' non structural proteins (NSPs) and accessory proteins involved in host immune escape and virus adaptive evolution, typically exhibiting less than 20% sequence homology across strains. These proteins play crucial roles in viral pathogenesis, their low sequence homology presents significant challenges for structural prediction. To evaluate existing methods' prediction performance on rapidly evolving viruses' structures, we test them on viral proteins from the CASP14-16 datasets. These included targets with limited homology (<20 homologous sequences), such as T1064 – structure of SARS-CoV-2 ORF8 accessory protein. As shown in Table 1, TDFold has superior performance compared to AlphaFold2, AlphaFold3 and ESMFold on these challenging targets. These results highlight TDFold's enhanced capability for virus structure prediction when homologous sequence information is scarce. TDFold plays a driving role in understanding the mechanisms by which these viruses evade immunity and developing corresponding drugs.

Comparison of inference time and GPU memory usage. For large-scale structural protein structure prediction tasks, inference time and memory usage are the two major factors determining inference costs. Therefore, we compare the inference computational costs of existing methods to evaluate their applicability for large-scale prediction tasks. Concretely, we provide a detailed comparison of time cost associated with predictions, as illustrated in Fig. 5a. AlphaFold2, AlphaFold3 [1] and RoseTTAFold, which rely on the search for homologous protein information, exhibit longer prediction times compared to language models such as ESMFold and our TDFold (e.g. for the protein with sequence length > 500, the inference time of AlphaFold2 and RoseTTAFold are both over 1000 seconds, the time of AlphaFold3 is about 240 seconds, the time of ESMFold is about 100 seconds, while the time of TDFold is only about

[1] Since DeepMind does not provide us with the model weights of AlphaFold3, we use the computation time of the AlphaFold3 server as the comparison result.

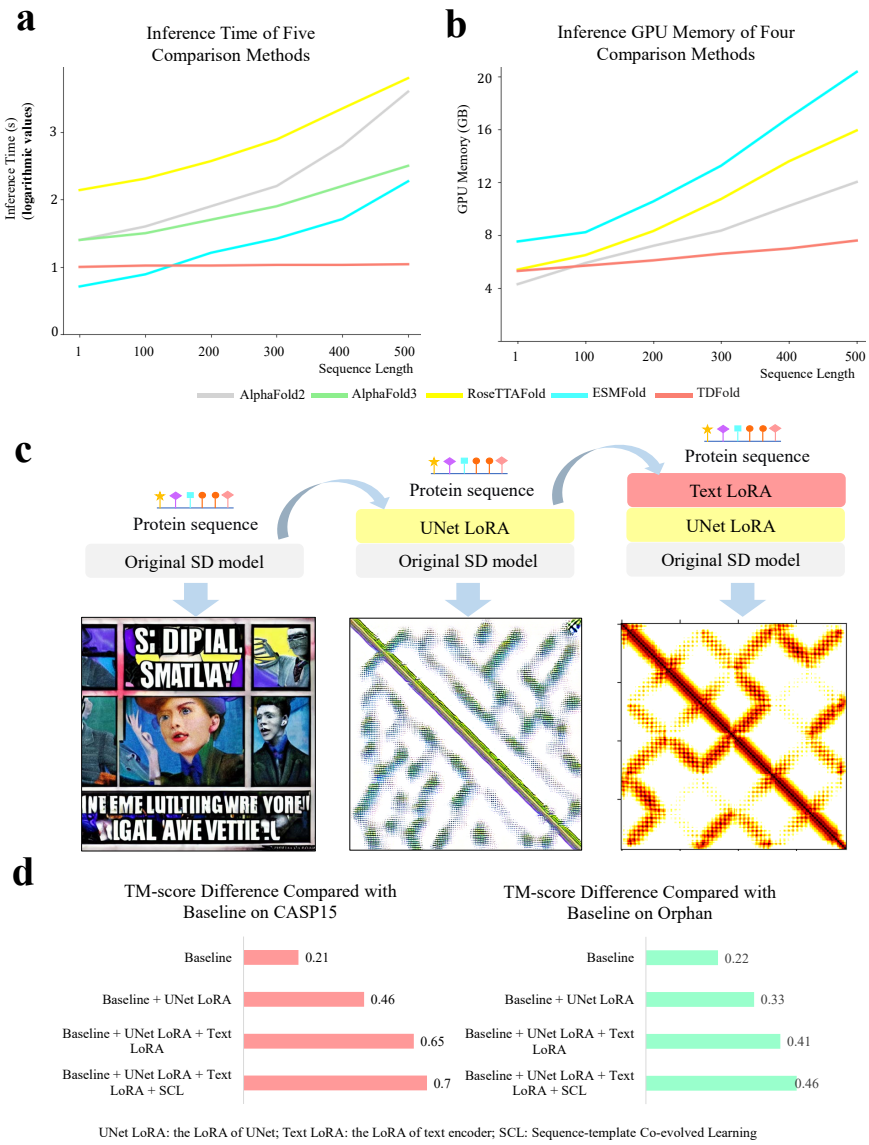

Fig. 5 Computation time-memory cost comparisons and model ablation analysis. a, Comparison of prediction times among AlphaFold2, AlphaFold3, RoseTTAFold, ESMFold and TDFold across 194 targets spanning our CASP14, CASP15, CASP16, Orphan and Orphan25 protein datasets. The prediction time of TDFold is almost lower than other methods, and only when the sequence length is less than 140, the prediction time is higher than ESMFold. Significantly, the inference time of all other methods increases with the length of the sequence, while the prediction time of TDFold basically remains stable. **b,** The GPU memory used for inference of AlphaFold2, RoseTTAFold, ESMFold, and TDFold. As the length of the amino acid sequence increases, the memory usage of the other three methods increase greatly (AlphaFold2: 4 GB to 12GB, RoseTTAFold: 5 GB to 16 GB, ESMFold: 8 GB to 20 GB). On the contrary, The memory usage of TDFold has slightly increased from 5 GB to 7 GB. **c,** The differences in the generated inter-residue distance images among the original SD model, the SD model enhanced with the UNet LoRA, and the SD model employing the "Text+UNet" LoRA are presented. Each ablation model is constructed by sequentially adding one or more components to the baseline model. **d,** TM-score difference between the SD model and several ablation models on the CASP15 and Orphan datasets.

10 seconds). While ESMFold achieves faster predictions for shorter sequences ($length \leq 140$), its prediction time escalates with increasing sequence length. This is because ESMFold adopts a triangular attention module with its $O(n^3)$ time complexity (where n is sequence length) in evoformer, which is the primary computational bottleneck governing the model's prediction time. In contrast, the prediction time of TDFold is primarily determined by the number of denoising steps and independent of the sequence length. And with the help of samplers including DPMsolver [37], UniPC [38] and DDIM [39], the number of denoising steps can be reduced from 1000 to just 25-50, dramatically speeding up the inference process.

Furthermore, the comparison of GPU memory occupation is shown in Fig. 5b. ESMFold exhibits the highest memory usage across all sequences, while AlphaFold2 and RoseTTAFold have moderate GPU memory consumption, and TDFold demonstrates the lowest memory usage on most sequences. Additionally, we analyzed the GPU memory usage during model inference for amino acid sequences of varying lengths. As the length of the amino acid sequence increases, the memory usage of AlphaFold2 and RoseTTAFold increases approximately 3 times (AlphaFold2: 4 GB to 12GB, RoseTTAFold: 5 GB to 16 GB), ESMFold's memory usage increases by 2.5 times (8 GB to 20 GB), while TDFold's memory usage only increases by 40% (5 GB to 7 GB). This is because TDFold adopts a lightweight SCL network, reducing the dimensionality and layers of the neural network, thereby reducing the use of GPU memory. Even as sequence length increases, TDFold's prediction time and GPU memory usage remain almost stable, highlighting the efficiency of TDFold in protein structure prediction.

Ablation Study. To assess the contribution of each component to TDFold's performance, we conduct an ablation study illustrated in Fig. 5. Specifically, the effectiveness of UNet LoRA and Text LoRA in promoting template diffusion is visualized in Fig. 5c, and the performance gains brought by each component on CASP15 and Orphan datasets are shown in Fig. 5d. On the left side of Fig. 5c, when prompted with a protein sequence, the generated image of the original stable diffusion model [23] presents mainly people and meaningless combinations of characters. This indicates that the model fails to comprehend the semantic context of inter-residue distance matrix and still regards the protein sequence as the natural language describing an image. In contrast, by introducing of the LoRA mechanism into the UNet, the generated image begins to present the diagonal structure, suggesting that the model start to learn the structural layout of inter-residue geometries. However, the sequence embedding and inter-residue distance image embedding are still not aligned, resulting in structural inconsistency in the generated image. Therefore, we add another LoRA for text encoder to finetune the semantics of the protein sequences to make them align with the inter-residue geometric image embeddings. On the right side, after incorporating both the text encoder LoRA and

UNet LoRA, the model effectively generates a coherent inter-residue distance image, well capturing the spatial geometric relationships between residues.

As illustrated in Fig.5d, adding each component (i.e. UNet LoRA, Text LoRA, SCL) effectively promote the prediction performance. For UNet LoRA, the considerable performance gains are obtained with 0.25 TM-score on CASP15 and 0.11 TM-score on Orphan. Meanwhile, Text LoRA also effectively improves the performances with 0.19 on CASP15 and 0.08 on Orphan. Furthermore, the SCL module also contributes to the prediction performances where the TM-scores increase approximately both 0.05 for CASP15 and Orphan. Moreover, we also compare the SD model (training-from-scratch + SCL) with the LoRA-finetuned version (UNet LoRA + Text LoRA + SCL) on both the CASP15 and Orphan datasets. The LoRA-finetuned model achieves superior performance, with TM-score improvements of 0.33 on CASP15 and 0.21 on Orphan. These results demonstrate the importance of fine-tuning – by leveraging the SD model's pre-trained semantic knowledge from 2 billion image-text pairs, LoRA effectively transfers the knowledge from the text-to-image generation to the inter-residue-geometry generation. Overall, in Figure 5, the ablation results verify the effectiveness of each component and the specific importance of UNet LoRA and Text LoRA to boost the accuracy of protein structure prediction. A detailed description of each ablation component is provided in Section 4.

3 Discussion

In this study, we introduce TDFold, a protein structure prediction model based on the proposed template diffusion model, and comprehensively evaluate its performance using both quantitative metrics and visualization results. TDFold effectively generates reliable inter-residue geometries (including distances and orientations) from a single protein sequence prompt. This approach allows for the utilization of sequence information and the generated inter-residue geometries to predict protein structures, eliminating the time-consuming process of searching for homologous information and achieving optimal predictive performance on orphan proteins. Moreover, TDFold demonstrates robustness in handling structure predictions for protein sequences with various levels of homologous information, especially those with a little or even no homologies (e.g. rapidly evolved viruses related proteins). The comprehensive experimental results on multiple CASP and Orphan datasets also verify their effectiveness in boosting single-sequence protein structure prediction. More importantly, with the help of LoRA fine-tuning technique, TDFold can be deployed on standard personal computers with a single NVIDIA 4090 24GB GPU, facilitating efficient training and inference. The huge advantage of TDFold in inference time also indicates that it is more suitable for large-scale prediction tasks that require high prediction speed.

Overall, TDFold demonstrates superior performance for orphan proteins or those with limited homologous information including orphan protein modeling where homologous sequences are absent, high-throughput applications due to reduced dependency on MSA generation and metagenomic proteins which often lack close homologs in databases. Additionally, TDFold is 10–100× faster than AlphaFold2 and requires substantially less GPU memory, making it highly efficient for large-scale protein structure predictions and scenarios with limited computational resources.

4 Methods

Datasets. In this study, the training samples are derived from protein data released in the PDB [15] prior to May 1, 2020, encompassing a total of 352,409 non-redundant protein domains. We evaluate our proposed model on five datasets: CASP14, CASP15, CASP16, Orphan, and Orphan25. The CASP14 dataset comprises available domain data, with target data correctly split by the domain definition index found in the download area of the CASP14 (Critical Assessment of Techniques for Protein Structure Prediction) website. For proteins with insufficient homologous information, we utilize two datasets from previous studies: Orphan [18] and Orphan25 [17]. The structural similarities (measured by TM-score) between proteins in the training dataset and in the benchmark datasets are all lower than 0.3. We further evaluated the sequence identity using BLAST, revealing that all benchmark datasets share the sequence identity that is less than 30% with the training set.

The datatype alignment between inter-residue geometries and images for the template diffusion. Due to the inconsistency between the continuous values of inter-residue geometries (distance and orientation) and the discrete image pixel values required by the stable diffusion model [23], we need to make feature datatype transformation for inter-residue geometric information. Specifically, the inter-residue distance and orientation, including d_{C_β} (distance between two "C_β" atoms), two dihedrals: ω (C_α-C_β-C_β-C_α), θ (N-C_α-C_β-C_β) and a planar angle ϕ (C_α-C_β-C_β), define the relative position of any two residues in a protein. For data consistency, we first divide the inter-residue C_β distance range into 36 intervals, i.e., (2.5 Å, 3.0 Å), (3.0 Å, 3.5 Å), $\cdots$, and (20.0 Å, 20.5 Å). For ω and θ in orientation, we also divide it into 36 bins from 0 to 360 degrees. But for planar angle ϕ, because it ranges from 0 to 180 degrees, it's divided into 18 parts. After discrete operations, the inter-residue distance and orientation can be represented as N*N (N is the length of protein sequence) discrete matrix with value range from 0 to 36 (0 to 18 for ϕ). In final, the geometric matrix are mapped into the range of RGB image pixel value ($0 \sim 255$) and can be processed by stable diffusion model.

The protein template diffusion module for inter-residue geometries generation. The protein template diffusion module is constructed through transfer learning based on the stable diffusion (SD) model [40]. The SD model is primarily trained on general images, which means it lacks the specific protein semantic information necessary for accurately representing inter-residue geometric images. To address this gap, we fine-tune the text encoder [33] and UNet [25] of the stable diffusion [23] using Low Rank Adaptation (LoRA) [24] (the specific process is introduced in the next section). This approach enables the model to effectively learn the semantic features associated with protein inter-residue geometries. Finally, the template diffusion module can generate inter-residue geometries with protein sequence prompts as conditions.

Specifically, the template diffusion module [40] is the probabilistic model designed to learn the inter-residue geometry (distances and orientations) distribution $p(x)$ (x is the inter-residue geometry) by gradually denoising a normally distributed variable, which corresponds to learning the reverse process of a fixed Markov Chain [41, 42] of length T. The probabilistic model can be interpreted as an equally weighted sequence of denoising autoencoders $e_\theta(x_t, t);\ t = 1...T$, which are trained to predict a denoised variant of input x_t, where x_t is a noisy version of the input x. Furthermore, the template (inter-residue geometries) generation process through the protein sequence prompt y can be regarded as a conditional distributions of the form $p(x \mid y)$ and controlled by a conditional denoising autoencoder [43] $e_\theta(x_t, t, y)$ with the cross-attention mechanism. To pre-process the protein sequence y, we introduce a protein sequence encoder τ_θ to project the y into an intermediate representation $\tau_\theta(y) \in \mathbb{R}^{N \times d^i_\tau}$. Based on inter-residue geometries and protein sequence condition, we then learn the template diffusion model objective L_{TDM} via

$$L_{TDM} = \mathbb{E}_{\varepsilon(x), y, \varepsilon \sim \mathcal{N}(0,1), t}[\varepsilon - e_\theta(x_t, t, \tau_\theta(y))^2_2] \tag{1}$$

The hierarchical LoRA model for the protein template diffusion module. In order to make the SD model [40] applicable to the template diffusion, we introduce the LoRA fine-tuning mechanism to the text encoder and UNet. Specifically, the original training parameters of the SD model are frozen, and only the parameters of LoRA are trained in the fine-tuning process. The integration of LoRA allows the text encoder model to map protein sequences and inter-residue geometries into a shared embedding space. This alignment is crucial for ensuring that the model understands the relationships between protein sequences and their corresponding geometric representations. And we apply LoRA to the UNet, enabling it to learn the distribution of inter-residue geometric data while being guided by protein sequence prompts.

For the text encoder, the incorporation of LoRA facilitates the mapping of protein sequences and inter-residue geometries into a unified embedding space. This integration is essential for enabling the model to effectively capture and interpret the intricate relationships between protein sequences and

their associated geometric structures. The LoRA model of text encoder can be represented as follows,

$$h_{geom} = W^I x_{geom}, \tag{2}$$
$$h_{seq} = W^T x_{seq} + \Delta W^G x_{seq}, \tag{3}$$
$$L_{CE} = cross_entropy((h_{seq} \cdot h_{geom}), y), \tag{4}$$

where W^T is the weight of original model weight of text encoder, ΔW^G is the global LoRA weight for learning the correlations between sequences and inter-residue geometries, and W^I is the weight of image encoder. The x_{seq} is the sequence embedding and the x_{geom} is the inter-residue geometry information. Moreover, the h_{seq} and h_{geom} are the hidden states for the x_{seq} and x_{geom}, respectively. Then, we calculate the cross entropy loss between the sequence features and geometric feature points multiplied by the label y. With the constraint of the loss function, the text features and inter-residue geometries features are gradually mapped into the same latent space.

Next, we apply LoRA to the UNet, enabling it to learn the distribution of inter-residue geometric data while being guided by protein sequence prompts. This process transfers the SD model from general image processing to the specific task of handling inter-residue geometric images. By fine-tuning these components, we enhance the model's ability to capture and generate protein inter-residue geometries effectively. The LoRA model of UNet can be represented as follows,

$$h_{geom} = W^U x_{geom} + \Delta W^L x_{geom}, \tag{5}$$
$$h'_{geom} = h_{geom} + cross_attn(h_{geom}, h_{seq}) \cdot h_{seq}, \tag{6}$$

where W^U is the weight of original UNet model weight, and ΔW^L is the local LoRA weights for predicting the value of inter-residue geometries $(d_{C_\beta}, \omega, \theta, \phi)$. After semantic alignment of sequence and geometric information through the text encoder LoRA, the sequence information is added as a guide to the geometric information generation process by the cross attention. After the optimizing the LoRA parameters, the reliable templates can be generated to promote the single-sequence protein structure prediction.

The long-text encoding capacity for protein sequence prompt embedding. The length of protein sequences varies from tens to thousands. However, the original CLIP text encoder used in stable diffusion only supports a maximum of 77 tokens (each token is a amino acid letter)—far fewer than the lengths of typical protein sequences. This token number limitation of the text encoder leads to a rigid constraint on the input protein sequence length.

To overcome the challenge of token numbers in the text encoder, position interpolation is introduced to the positional embedding of the text encoder to enable context window extensions. We leverage the property that position

encodings can be extended to non-integer positions by interpolating the encodings at adjacent integer positions [44]. The calculation for obtaining the new extended positional embedding $PE_{extend}(i)$ can be expressed as follows:

$$PE_{extend}(i) = (1-\lambda) \cdot PE(\frac{i}{d}) + \lambda \cdot PE(\frac{i}{d}+1), \lambda = \frac{i\%d}{d}, \tag{7}$$

where $PE(i)$ is the original positional embedding of i_{th} position, and d is the times number of interpolation. λ is a ratio between 0 and 1, determining whether the interpolated positional embedding for the i_{th} position is closer to its preceding or following position.

The lightweight sequence-template co-evolved learning module. The lightweight sequence-template co-evolved learning module consists two branches: the residue branch and the atom branch. The residue-level branch processes the protein sequence alongside generated inter-residue geometries $(d_{C_\beta}, \omega, \theta, \phi)$. It comprises two primary learning modules: a hybrid convolutional neural network (CNN) and a residue graph learning module. The hybrid CNN architecture combines both asymmetric and symmetric convolutional kernels to facilitate the effective learning of inter-residue geometric features. Specifically, asymmetric kernels are employed to capture the dihedral angle θ (N-C_α-C_β-C_β) and the planar angle ϕ (C_α-C_β-C_β), while symmetric kernels are utilized to model the distance d_{C_β} (C_β-C_β) and the dihedral angle ω (C_α-C_β-C_β-C_α). This dual-kernel design ensures that the inherent symmetries of inter-residue geometries—particularly in the case of d_{C_β} and ω—are accurately preserved and leveraged during the learning process. Subsequently, we construct a residue-level graph using the extracted residue features and their inter-residue representations. In this graph, residues are represented as nodes, while their interactions are encoded as edges. To perform residue representation learning, we employ a graph transformer [34]. Each layer is represented as $X^{(l+1)} = \text{LayerNorm}(X^{(l)} + \text{MultiHeadAttention}(X^{(l)}, A))$, where $X^{(l)}$ represents the feature matrix at the l-th layer and A denotes the adjacency matrix. This framework enables the learning of multi-head attention coefficients, which are used for node aggregation, thereby promoting effective message passing across the graph structure.

For the atom-level branch, we account for the unique interactions among atoms within residues by constructing an atom-level graph. In this graph, atoms are represented as nodes, and inter-atom bonds are modeled as edges. Leveraging the amino acid sequence, these graphs are assembled using the peptide bond "-C-N-" as a structural foundation. A Graph Neural Network (GNN) block [35] is then applied to learn atomic representations, enabling the efficient propagation of side-chain information to backbone atoms. Each GNN layer is formulated as $X^{(l+1)} = GNN(A, X^{(l)})$, where $X^{(l)}$ represents the feature matrix at the l-th layer and A denotes the adjacency matrix. To enhance the model's representational capacity, shortcut connections are

integrated between GNN layers, promoting more robust and effective learning of atomic features.

Fusing the graph learning features to predict the protein 3D coordinates. We design a variational learning process to learn weighting factors to fuse the residue-level and atom-level branches. Hereby, the residue features can be refined by mining local atom-level structures that affect inter-residue bonding, e.g. torsion and folding. Here, we use $\mathbf{X}^r = [\mathbf{x}_1^r, \cdots, \mathbf{x}_L^r]^\top \in \mathbb{R}^{M\times d}$ (M is the number of residues) to denote the residue feature matrix, and $\mathbf{X}^a = [\mathbf{x}_1^a, \cdots, \mathbf{x}_N^a]^\top \in \mathbb{R}^{N\times d}$ (N is the number of atoms) to denote the atom feature matrix. In the fusion process, the atom-residue similarity matrix $\mathbf{S} \in \mathbb{R}^{M\times N}$ is first calculated, where the element in the i-th row and j-th column is formulated as $r_{ij} = \frac{(\mathbf{x}_i^r)^\top \mathbf{x}_j^a}{\|\mathbf{x}_i^r\| \ \|\mathbf{x}_j^a\|}$. r_{ij} represents the relationship between the i-th residue and j-th atom in the protein.

Considering the complicated influence of atoms on the protein conformation, we resort to probabilistic reasoning to select salient atoms that may influence structural bonding, so as to refine residue features. Given the atom-residue relationship $\mathbf{R}$, we derive a set of random variables, denoted as $\mathcal{Z}$, by learning the posterior probability $p(\mathcal{Z}|\mathbf{X}^r, \mathbf{X}^a, \mathbf{R})$. Each element $z_{ij} \in \mathcal{Z}$ follows the Bernoulli distribution, denoted as $z_{ij} \sim B(p_{ij})$, where p_{ij} represents the probability that the j-th atom may influence the structural bonding of the i-th residue. However, the posterior probability $p(\mathcal{Z}|\mathbf{X}^r, \mathbf{X}^a, \mathbf{R})$ is usually intractable. For this problem, inspired by the reparameterization trick in [45], we develop the Bernoulli reparameterization to derive the posterior probability $p(\mathcal{Z}|\mathbf{X}^r, \mathbf{X}^a, \mathbf{R})$. Based on $\mathcal{Z}$, salient atom features are selected for each residue. These selected atom features are concatenated with the corresponding residue feature, and then fed into the SE(3) equivariant graph neural network (EGNN) [36] for projecting fusion features into the 3D coordinates.

References

[1] Jumper, J., Evans, R., Pritzel, A., et al: Highly accurate protein structure prediction with alphafold. Nature **596**, 583–589 (2021)

[2] Baek, M., DiMaio, F., Anishchenko, I., et al: Accurate prediction of protein structures and interactions using a three-track neural network. Science. **117**, 871–876 (2021)

[3] Zhang, J., Wang, Q., Barz, B., et al: Mufold: a new solution for protein 3d structure prediction. Proteins Struct. Funct. Bioinform. **78**, 1137–1152 (2010)

[4] Hong, Y., Lee, J., Ko, J.: A-prot: Protein structure modeling using msa transformer. BMC bioinformatics. **23**, 1–11 (2022)

[5] Zhang, C., Zheng, W., Mortuza, S., et al: Deepmsa: constructing deep multiple sequence alignment to improve contact prediction and fold-recognition for distant-homology proteins. Bioinformatics. **36**, 2105–2112 (2020)

[6] Marks, D.S., Hopf, T.A., Sander, C.: Protein structure prediction from sequence variation. Nat. Biotechnol. **30**, 1072 (2012)

[7] Ju, F., Zhu, J., Shao, B., et al: Copulanet: Learning residue co-evolution directly from multiple sequence alignment for protein structure prediction. Nat Commun. **12**, 2535 (2021)

[8] Yang, J., Anishchenko, I., Park, H., et al: Improved protein structure prediction using predicted interresidue orientations. Proc. Natl. Acad. Sci. U. S. A. **117**, 1496–1503 (2020)

[9] Xu, J., Mcpartlon, M., Li, J.: Improved protein structure prediction by deep learning irrespective of co-evolution information. Nature Machine Intelligence **3**(7), 601–609 (2021)

[10] Abramson, J., Adler, J., Dunger, J., et al: Accurate structure prediction of biomolecular interactions with alphafold 3. Nature **630**(8016), 493–500 (2024)

[11] Kryshtafovych, A., Schwede, T., Topf, M., et al: Critical assessment of methods of protein structure prediction (casp)—round xiv. Proteins: Structure, Function, and Bioinformatics **89**(12), 1607–1617 (2021)

[12] Jeanmougin, F., Thompson, J., Gouy, M., et al: Multiple sequence alignment with clustal x. Trends in biochemical sciences **23**(10), 403–405 (1998)

[13] Fiser, A.: Template-based protein structure modeling. Computational biology, 73–94 (2010)

[14] Suzek, B.E., Wang, Y., Huang, H., et al: Uniref clusters: a comprehensive and scalable alternative for improving sequence similarity searches. Bioinformatics. **31**, 926–932 (2015)

[15] wwPDB Consortium.: Protein data bank: the single global archive for 3d macromolecular structure data. Nucleic Acids Res. **47**, 520–528 (2018)

[16] Lin, Z., Akin, H., Rao, R., et al: Evolutionary-scale prediction of atomic-level protein structure with a language model. Science **379**(6637), 1123–1130 (2023)

[17] Wang, W., Peng, Z., Yang, J.: Single-sequence protein structure prediction using supervised transformer protein language models. Nature Computational Science **2**, 804–814 (2022)

[18] Chowdhury, R., Bouatta, N., Biswas, S., et al: Single-sequence protein structure prediction using a language model and deep learning. Nat Biotechnol **40**, 1617–1623 (2022)

[19] Wu, R., Ding, F., Wang, R., et al: High-resolution de novo structure prediction from primary sequence. BioRxiv, 2022–07 (2022)

[20] Fang, X., Wang, F., Liu, L., et al: Helixfold-single: Msa-free protein structure prediction by using protein language model as an alternative. arXiv preprint arXiv:2207.13921 (2022)

[21] Vaswani, A., Shazeer, N., Parmar, N., et al: Attention is all you need. (2017). Paper presented at the 31st International Conference on Neural Information Processing Systems, Long Beach Convention Center, 4–9 December 2017

[22] Peng, J., Xu, J.: Raptorx: exploiting structure information for protein alignment by statistical inference. Proteins: Structure, Function, and Bioinformatics **79**(S10), 161–171 (2011)

[23] Rombach, R., Blattmann, A., Lorenz, D., et al: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10684–10695 (2022)

[24] Hu, E., Shen, Y., Wallis, P., et al: Lora: Low-rank adaptation of large language models. arXiv preprint arXiv:2106.09685 (2021)

[25] Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks

for biomedical image segmentation. In: Medical Image Computing and Computer-assisted intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18, pp. 234–241 (2015). Springer

[26] Wang, X., Zhang, T., Liu, G., et al: Lightrosetta: High-efficient and accurate protein structure prediction using an ultra-lightweight deep graph model. bioRxiv, 2023–11 (2023)

[27] Kryshtafovych, A., Schwede, T., Topf, M., et al: Critical assessment of methods of protein structure prediction (casp)—round xv. Proteins: Structure, Function, and Bioinformatics **89**(12), 1607–1617 (2023)

[28] Kryshtafovych, A., Schwede, T., Topf, M., et al: Critical assessment of methods of protein structure prediction (casp)—round xvi. Proteins: Structure, Function, and Bioinformatics **89**(12), 1607–1617 (2025)

[29] Zhang, Y., Skolnick, J.: Scoring function for automated assessment of protein structure template quality. Proteins. **57**, 702–710 (2004)

[30] Zemla, A.: Lga: a method for finding 3d similarities in protein structures. Nucleic acids research **31**(13), 3370–3374 (2003)

[31] Mariani, V., Biasini, M., Barbato, A., et al: lddt: A local superposition-free score for comparing protein structures and models using distance difference tests. Bioinformatics **29**, 2722–2728 (2013)

[32] Gutub, A.: Pixel indicator technique for rgb image steganography. Journal of emerging technologies in web intelligence **2**(1), 56–64 (2010)

[33] Radford, A., Kim, J., Hallacy, C., et al: Learning transferable visual models from natural language supervision. In: International Conference on Machine Learning, pp. 8748–8763 (2021). PMLR

[34] Shi, Y., Huang, Z., Feng, S., et al: Masked label prediction: Unified message passing model for semi-supervised classification. (2021). Paper presented at the thirtieth International Joint Conference on Artificial Intelligence, virtual only, 19–26 August 2021

[35] Morris, C., Ritzert, M., Fey, M., et al: Weisfeiler and leman go neural: Higher-order graph neural networks (2019). Paper presented at the Conference of the Thirty-Third Conference on Artificial Intelligence, Hawaii, USA., 27 January – 1 February 2019

[36] Fuchs, F., Worrall, D., Fischer, V., et al: Se(3)-transformers: 3d rototranslation equivariant attention networks. (2020)

[37] Lu, C., Zhou, Y., Bao, F., et al: Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. Advances in Neural Information Processing Systems **35**, 5775–5787 (2022)

[38] Zhao, W., Bai, L., Rao, Y., et al: Unipc: A unified predictor-corrector framework for fast sampling of diffusion models. Advances in Neural Information Processing Systems **36**, 49842–49869 (2023)

[39] Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. arXiv preprint arXiv:2010.02502 (2020)

[40] Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. Advances in neural information processing systems **33**, 6840–6851 (2020)

[41] Chung, K.: Markov chains. Springer-Verlag, New York (1967)

[42] Norris, J.: Markov Chains vol. 2. Cambridge university press, London (1998)

[43] Zhai, J., Zhang, S., Chen, J., et al: Autoencoder and its various variants. In: 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 415–419 (2018). IEEE

[44] Chen, S., Wong, S., Chen, L., et al: Extending context window of large language models via positional interpolation. arXiv preprint arXiv:2306.15595 (2023)

[45] Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)