Supplementary Information

Unsupervisedly Prompting AlphaFold2 for Accurate Few-Shot Protein Structure Prediction

Jun Zhang[1,*], Sirui Liu[1], Mengyun Chen[2], Haotian Chu[2], Min Wang[2], Zidong Wang[2], Jialiang Yu[2], Ningxi Ni[2], Fan Yu[2], Dechin Chen[3], Yi Isaac Yang[3], Boxin Xue[4], Lijiang Yang[4], Yuan Liu[5] and Yi Qin Gao[1,3,4,6,*]

Affiliations:

[1] Changping Laboratory, Beijing 102200, China.

[2] Huawei Hangzhou Research Institute, Huawei Technologies Co. Ltd., Hangzhou 310051, China.

[3] Institute of Systems and Physical Biology, Shenzhen Bay Laboratory, Shenzhen 518055, China.

[4] Beijing National Laboratory for Molecular Sciences, College of Chemistry and Molecular Engineering, Peking University, Beijing 100871, China.

[5] Department of Chemical Biology, College of Chemistry and Molecular Engineering, Peking University, Beijing 100871, China.

[6] Biomedical Pioneering Innovation Center, Peking University, Beijing 100871, China.

* To whom correspondence should be addressed. Email: jzhang@cpl.ac.cn or gaoyq@pku.edu.cn

Table of contents

Datasets

1. Test sets

We prepared four independent test sets, i.e., CASP14, CAMEO, poor MSA and *de novo*, to benchmark performance of EvoGen. CASP14 test set contains 84 domain-divided single-chain targets in the official CASP14 name list with sequence length less or equal than 512. CAMEO test set contains all single-chain targets for CAMEO dating from 2021-08-21 to 2022-02-12. For ease of inference, we filtered out sequences longer than 512, resulting in 292 targets in total. Poor MSA dataset consists of single protein chains with known PDB structures but with less than 30 available MSA's. It is created by filtering all PDB entries in PSP Database (PSPD)[1] with a date truncation at 2020-05-14. Since AF2 is trained for single chain PSP, we further filtered this dataset to exclude any chains forming protein-protein interactions in heteromers. We also removed any sequences (with labeled structures) which are shorter than 15 amino acids. The resulting dataset contains 1074 targets, among which 382 targets do not have any MSA and are excluded for few-shot MSA augmentation experiments (Section III in the main text). These poor-MSA targets not only contain very few MSA information, but also show limited structural similarity to any structures in the PDB database. Besides, we reused the list of de novo targets for RaptorX[2] which contains 35 artificial designed proteins using the Rosetta energy function. Twenty-one targets in this set were benchmarked by RaptorX, and we plotted the results of RaptorX on these applicable targets in Fig. 6d. We also curated a non-redundant and representative CASP15 test set containing 16 single-chain targets with publicly known reference structures and named identically to CASP15 official website, including T1104, T1106-S1, T1106-S2, T1113, T1114-S1, T1114-S2, T1119, T1120, T1121, T1122, T1123, T1124, T1134-S1, T1134-S2, T1152, T1187. Benchmark of CASP15 is shown in Fig. S3.

2. MSA trimming

Because all experiments in this paper were designed for low-data regime, we performed *MSA trimming* for CASP14 and CAMEO targets whenever MSA is abundant. Given a maximum MSA depth $N_{\max}$, the MSA trimming follows the same procedure as

adopted for PSPD-Lite[1]. Specifically, for each target sequence whose MSA depth exceeding $N_{\max}$, we first filtered its MSA according to three primary rules: i) all MSA's with coverage less than 50% are removed; ii) all MSA's with >90% identity to target are removed; iii) all MSA's with <20% identity to target are removed. If MSA depth of the target still exceeds $N_{\max}$ after filtering, we further selected representative MSA's via a heuristic strategy as follows: We initialized an MSA pool using the target sequence alone, then added to this pool a new MSA given that this candidate is of no more than 90% identity to all MSA's already in the pool, and that this candidate is closest to the target in terms of the Hamming's distance. This iterative selection stops when no more candidates can be accepted or the MSA pool is full (up to $N_{\max}$). MSA trimming with $N_{\max}$=128 was performed for CASP14 and CAMEO test sets.

3. Training sets

We curated two training sets for EvoGen. The "labeled set" contains both sequences and structural labels, while the "unlabeled set" is composed merely of sequences (and MSA) without structural labels. The labeled set consists of 447K filtered PDB structures extracted from PSPD-Lite with a date truncation before 2020-05-14. CASP14, CASP15 and CAMEO test sets are naturally excluded from the training set. We did not explicitly exclude the same structure fold further in order to make a fair benchmark against AlphaFold2 which was trained on the same dataset. The unlabeled set further expands the labeled set by adding 648K filtered non-redundant sequences in UniRef50[3] extracted from PSPD-Lite[1], and only the sequence information (i.e. MSA) is preserved whereas the structure labels are deprecated. For both training sets, MSA trimming is performed with a $N_{\max} = 256$ following the strategy described above. Additional filtering was performed after trimming: i) All entries with MSA depth smaller than 128 are removed (the poor MSA test set is thus excluded from the training set); ii) Any sequences in the de novo test set are manually removed; iii) Sequences or structural labels with length shorter than 20 amino acids are also removed. EvoGen was trained on the unlabeled set for MSA calibration, and fine-tuned on the labeled set for MSA augmentation.

Training Settings

1. Training objective of EvoGen

As elaborated in the main text, we aim to optimize the deep neural network model in order to maximize the conditional log-likelihood in Eq. (S1),

$$LL = \mathbb{E}_{m\in\mathcal{D},\mathbf{S}_m\in\{\mathbf{S}_m\}_{\text{target}}} \log p_\theta\left(\mathbf{S}_m \mid \{\mathbf{S}_m\}_{\text{context}}\right) \tag{S1}$$

where we divide a full set of MSA m into two subsets: $\{\mathbf{S}_m\}_{\text{context}}$ serves as conditional information, while $\{\mathbf{S}_m\}_{\text{target}}$ is used as training targets. This likelihood is intractable, however, we can derive an evidence lower bound (ELBO) for it by means of variational inference. Simply speaking, log-likelihood in Eq. (S1) can be re-formulated as Eq. (S2),

$$\log p\left(\mathbf{S}_m \mid \{\mathbf{S}_m\}_{\text{context}}, \theta\right) = \log \int p\left(\mathbf{S}_m, \mathbf{z} \mid \{\mathbf{S}_m\}_{\text{context}}, \theta\right) d\mathbf{z} \tag{S2}$$

where $\mathbf{z}$ is a latent variable generated by a (potentially data-dependent) prior, and it has a lower bound according to Jensen's equality where we denote $\{\mathbf{S}_m\}_{\text{context}}$ as $\{\mathbf{S}_m\}$ for short,

$$\begin{aligned} \log p\left(\mathbf{S}_m \mid \{\mathbf{S}_m\}, \theta\right) &\geq \mathcal{L}\left(\theta, \phi\right) \\ \mathcal{L}\left(\theta, \phi\right) &= \mathbb{E}_{q_\phi}\left[\log p_\theta\left(\mathbf{S}_m, \mathbf{z} \mid \{\mathbf{S}_m\}\right) - \log q_\phi\left(\mathbf{z} \mid \mathbf{S}_m, \{\mathbf{S}_m\}\right)\right] \\ &= \mathbb{E}_{q_\phi}\left[\log p_\theta\left(\mathbf{S}_m \mid \mathbf{z}, \{\mathbf{S}_m\}\right) - D_{\text{KL}}\left(q_\phi\left(\mathbf{z} \mid \mathbf{S}_m, \{\mathbf{S}_m\}\right) \| \, p_\theta\left(\mathbf{z} \mid \{\mathbf{S}_m\}\right)\right)\right] \end{aligned} \tag{S3}$$

Eq. (S3) consists of two models: a generative or decoder model p_θ performing reconstruction according to the context and latent variable $\mathbf{z}$, whereas an inference or encoder model q_ϕ performing variational inference for the posterior. The tightness of ELBO is controlled by the variational inference model q_ϕ which aims to minimize the Kullback-Leibler (KL) divergence to the true posterior $p\left(\mathbf{z} \mid \mathbf{S}_m, \{\mathbf{S}_m\}\right)$. Therefore, it is natural to approximate both models with a deep neural network which is known for its expressivity as in Variational Auto-Encoders (VAE)[4]. During training, we optimized model parameters in order to maximize the ELBO in Eq. (S3).

In vanilla VAE, the prior for latent variable is usually a simple distribution like the standard normal. However in EvoGen, since we are dealing with contexts and sequences of varied lengths, we choose to learn a data-dependent prior for the latent variables.

Particularly, like denoising diffusion models[5], the dimension of latent variables is consistent with the length of target sequence, hence, making the model transferable to sequences of varied length. Besides, we introduced multi-scale priors which take the form of autoregressive Gaussians[6,7] to make ELBO tighter. Compared to a single Gaussian, autoregressive Gaussians can better approximate any complex distribution, meanwhile allow fast and straightforward sampling which is crucial to the selection of priors.

Note that Eq. (S3) consists of two terms, one for reconstruction loss as in an autoencoder, the other for KL divergence which can be considered as a regularizer. To stabilize training and avoid posterior collapse[8], we adopted a warm-up schedule as in NVAE[7] to gradually tune-up the strength of the KL divergence term. Besides, since both terms depend on the length of input sequence, we balanced the mini-batch gradient according to the sequence length as well. We scaled the loss of each MSA with a weight factor proportional to the square root of target length as recommended by AlphaFold2[9].
We trained the model using a batch size of 128 MSAs, each MSA was cropped to a maximum length of 256 and maximum depth of 128. We adopted ADAM optimizer[10] (with default beta, epsilon=1e-6) and clipped the gradient by norm bounded by 0.1. The learning rate was warmed up from 0 to 5e-4 during the first 3K steps, then decayed according to a cosine learning rate schedule to 1e-5 during 100K steps. In total 150K training iterations (or gradient steps) were executed for unsupervised pre-training using the unlabeled training dataset (see Datasets in SI) which aims to maximize Eq. (S3), and the resulting model is used for MSA calibration throughout the paper.

Another 50K training iterations were performed using the labeled training dataset (see Datasets in Supplementary Information) under the guidance of AF2 which aims to minimize the combined loss in Eq. (S4),

$$\mathcal{L}_{\text{finetune}} = 0.5\mathcal{L}_{\text{FAPE}} + 0.5\mathcal{L}_{\text{torsion}} + 0.01\mathcal{L}_{\text{viol}} + 0.01\mathcal{L}_{\text{conf}} - 0.1\mathcal{L}_{\text{EvoGen}} \quad \text{(S4)}$$

where $\mathcal{L}_{\text{FAPE}}$ stands for clamped frame-aligned point errors (FAPE) of both backbone and sidechains, $\mathcal{L}_{\text{torsion}}$ for the loss of sidechain torsional angles, $\mathcal{L}_{\text{viol}}$ for violation losses, $\mathcal{L}_{\text{conf}}$ for confidence loss, and $\mathcal{L}_{\text{EvoGen}}$ corresponds to Eq. (S3). All loss terms in Eq. (S4) except $\mathcal{L}_{\text{EvoGen}}$ take the same form as AF2[9]. Note that we also relaxed the parameters of Evoformer module and the confidence head of AF2 during fine-tuning since we observed

that virtual MSA generated by EvoGen may cause AF2 to overestimate the quality of predictions. The fine-tuned model was adopted for MSA augmentation throughout the paper.

We performed training over 128 Ascend-910 NPU cards using MindSpore[11] and adopted hybrid float precisions during training to reach the optimal performance.

2. Differentiate through AF2

For MSA augmentation, we trained EvoGen with respect to relevant structural losses feedbacked by AF2 according to Eq. (S4). However, to compute the supervised losses, we need to transform the softmax-valued output (see “Model Details” in SI) of EvoGen to be one-hot-coded MSA features then passed to AF2. Simply using ArgMax transform would stop the gradient and forbid backpropagation through EvoGen.

Therefore, we applied Gumbel-Softmax trick[12] to generate nearly one-hot samples according to the softmax logits, and adopted straight-through estimator[13,14] to allow backpropagation of EvoGen in joint with AF2. Let $\mathbf{S}_{\mathrm{gs}}$ denote Gumbel-Softmax samples which are differentiable with respect to EvoGen parameters, $\mathbf{S}_{\mathrm{hard}}$ denote one-hot MSA features after ArgMax transform of $\mathbf{S}_{\mathrm{gs}}$, and $f\left(\mathbf{S}_{\mathrm{gs}}\right)$ is an arbitrary function of EvoGen output, the straight-through estimator reads like Eq. (S5),

$$f(\mathbf{S}) = f\left(\mathbf{S}_{\mathrm{gs}}\right) + \mathrm{StopGrad}\left[f(\mathbf{S}_{\mathrm{hard}}) - f\left(\mathbf{S}_{\mathrm{gs}}\right)\right] \tag{S5}$$

where “StopGrad” stands for stop-gradient operation. During forward inference, Eq. (S5) computes the function value using the one-hot coded $\mathbf{S}_{\mathrm{hard}}$, while during backpropagation, the gradient with respect to the Gumbel-Softmax samples are computed.

Inference Settings

1. Inference settings of AF2

We conducted all the experiments without templates. AF2 model-3 released by DeepMind was chosen for inference and training in all experiments unless specified otherwise. AF2 model-3 slightly outperformed the other two template-free models (model-4 and model-5) on our benchmark dataset, and it is also recommended as default model by batch-mode ColabFold[15]. After MSA trimming, MSA subsampling is no longer performed during inference, except when we deliberately sub-sampled MSA for purposes (see Section I in Experiments & Results). We also turned off any other settings which could cause non-deterministic effects (e.g., BERT) during AF2 inference. Unless stated otherwise, AF2 inference was executed exclusively using a recommended number of three recycles.

2. Inference settings of EvoGen

One special hyper-parameter during EvoGen inference (for both MSA calibration and augmentation) is the context MSA ratio, which determines how much fraction of available MSA is used as contexts during inference. Let r_{ctx} denotes context MSA ratio range between 0 and 1, and N_{MSA} denote the available MSA number (possibly after MSA trimming) provided to EvoGen, then the number of context MSA is the integer part of $r_{\mathrm{ctx}}N_{\mathrm{MSA}}$. Note that regardless of r_{ctx}, the first sequence in MSA, i.e., the query sequence itself, is always included in the context.

By setting a large r_{ctx} (close to unity), the calibrated or generated MSA tend be more consistent. In contrast, a small r_{ctx} means more randomness in sub-sampled contexts, and usually leads to more noisy output. This hyper-parameter can help us strike balance between exploration (with smaller r_{ctx}) and exploitation (with larger r_{ctx}). In our experiments, we chose three values for $r_{\mathrm{ctx}} \in \{0.5,0.7,0.9\}$ for each all tasks where multiple MSA sequences are available unless specified otherwise.

For MSA augmentation, there is an additional hyper-parameter N_{aug} controlling the augmented MSA depth. For few-shot learning, we set $N_{\mathrm{aug}} = 128$ in order to make a fair

comparison to vanilla AF2 with trimmed MSA depth of 128. For single-sequence prediction, or zero-shot learning, we ran inference using three different values $N_{\mathrm{aug}} \in \{16,32,64\}$, to test the impact of this hyper-parameter, and did not observe significant change of performance as long as $N_{\mathrm{aug}} \geq 32$.

After finetuned under the guide of AF2, we found that directly fed Softmax output of EvoGen without any hardening transform to the downstream AF2 model yields slightly better performance for MSA augmentation. This might benefit from the "dark knowledge" in the Softmax output which turns a token of amino acid (one-hot code) at a position into a distribution of all possible amino acids at this position, hence, helps smooth the folding landscape of AF2.

Given a specific choice of r_{ctx} (and) or N_{aug}, we ran five independent inferences using different Gaussian random noises for MSA calibration and few-shot MSA augmentation experiments. In zero-shot MSA augmentation experiments, we reduced the number of random trials to two. Among all executed trials, we ranked all predictions according to the confidence score (i.e., residue-averaged plDDT) yielded by AF2, and reported the top-1 prediction as the "first prediction" in all experiments. We also reported the *de facto* "best prediction" with the ground truth label as reference.

We recorded all the "first" and "best" predictions in our experiments, which can be checked via the open-source link. We also kept records of the output of EvoGen (i.e., calibrated MSA features) which could be used to reproduce the reported structures using third-party implementation of AF2 like ColabFold[15].

3. Probing alternative conformations

When MSA is sufficiently deep, direct implementing AF2 inference will lead to limited variations in predicted structures as proved in this paper and related work[16]. Consequently, implementing the generative inference workflow presented in this paper to probe alternative may find wide applications in protein science beyond few-shot learning scenarios.

We summarized a brief protocol of how to increase the diversity of AF2 prediction with the help of EvoGen. First, select a reasonable N_{max} and perform MSA trimming

accordingly. Secondly, randomly sub-sample N_{sub} from the trimmed MSA pool and feed them to EvoGen. We remark here that previous research[16] also suggested implementing AF2 with a shallow MSA in order to get diverse structure predictions. Thirdly, choose a context MSA ratio r_{ctx} and perform MSA calibration accordingly with one or more random seeds. Finally, pass the reconstructed MSA features to AF2 and perform structure predictions, and cluster the confident predictions with proper similarity metrics like TMScore[17]. According to our experiments, we recommend $N_{\max} = 512$ or 1024, $N_{\mathrm{sub}} \in \{16,32,64\}$ and $r_{\mathrm{ctx}} \in \{0.25,0.5,0.75\}$ in practice for efficient probing of alternative protein conformations.

4. Calibrated MSA patterns

We investigated how calibrated MSA help improve AF2 performance. Figure S1 shows two exemplary cases where the raw MSA fails (Fig. S1a) but the calibrated MSA successfully restores the 3D structure (Fig. S1b) predicted by AF2. We performed DCA[18] with pyDCA[19] based on raw and calibrated MSA, respectively, and found that, calibrated MSA slightly improves the number of true positive prediction of inter-residue contacts (defined as CA atoms distance shorter than 8 Å; Fig. S1c). Besides, calibrated MSA also reduces false positive contact predictions to some extent (Fig. S1c). These findings suggest that calibrated MSA may help reduce noise for protein folding. Given that MSA depth in our tested cases is generally too shallow for a meaningful DCA, we additionally performed “deep DCA” using neural-network-based models as adopted in AlphaFold1[20] and RaptorX[2], etc. Here we simply regarded Evoformer block of AF2 as a good analyzer of coevolutionary information, and compared the quality of MSA based on the accuracy of the distogram prediction yielded by Evoformer (similar to what AlphaFold1 did). Our results showed that the raw MSA lead to much error-prone distogram compared to calibrated MSA (Fig. S1d), particularly in regions with potential contacts (pair distance shorter than 15 Å), further demonstrating that the calibrated MSA contain much less perturbative signals to correct folding.

We also inspected what the calibrated MSA look like. Since EvoGen is designed and trained following the “relativity” principle (Fig. 1a), it is not surprisingly that, the calibrated

MSA bear significant resemblance with their original counterparts, showing >95% overall sequence identity, which means the mutation rate is not significantly changed. However, we found there are some patterns in the calibration to which the improved performance may attribute (Fig. S2): For instance, during calibration, mutations in homologue sequences may be recovered to the conserved one according to the target sequence. A residue may be substituted with another residue with similar physical and chemical properties - a common phenomenon in protein design. Moreover, EvoGen could help filling some “gaps” – artifacts during alignment – meaningfully, thus increasing the effective “coverage” which is known to influence MSA quality, and removing unknown or rare residues (denoted by “X” in AF2) by making reasonable replacement. These patterns can be useful to denoise the raw MSA and smooth the folding landscape.

Model Details

1. Input and output of EvoGen

The input to EvoGen is a set of MSA sequences $m \equiv \{\mathbf{S}_m^i\}_{i=1,\ldots,N_m}$. The first sequence is always the query sequence ($S_m^1 \equiv Q_m$), the query sequence does not contain gaps or deletions. While the other sequences are aligned to the query, they may contain gaps or deletions due to alignment.

Each sequence $\mathbf{S}_m^i \in m$ is featurized by the type of amino acid and the number of deletions at each position along the sequence. The amino acid is categorized into a vocabulary of 22 tokens, including 20 for common amino acids, 1 for rare amino acids and 1 for gap token. The deletion number of each position in a sequence is transformed via arctan function as in AF2.

The output of EvoGen should correspond to the input in order to perform reconstruction. The amino acid type at each position along the sequence is predicted by a softmax function with 22 logits corresponding to vocabulary tokens. The arctan deletion number is first discretized into 6 bins ranging from 0.2 to 0.95, and a softmax function with 6 logits predicts the discretized values.

2. Hyperformer

Hyperformer inherits the overall architecture of Evoformer[9] but exhibits several key differences (Fig. 1c). First of all, the original biased attention is replaced with hyper-attention inspired by Molecular CT[21], and the attention coefficient between a d-dimensional Query vector $\mathbf{q}_i$ and Key vector $\mathbf{k}_j$ is computed as

$$\mathrm{Att}(i,j) = \mathrm{softmax}\left(\frac{\mathbf{q}_i^{\mathrm{T}}\mathbf{W}_{ij}\mathbf{k}_j}{\sqrt{d}} + b_{ij}\right) \tag{S6}$$

where $\mathbf{W}_{ij}$ and b_{ij} are learnable parameters or activations of neural networks which are both functions of the relative positions (or pair activations) between i-th and j-th tokens. Similar to hyper-networks[22], $\mathbf{W}_{ij}$ and b_{ij} here represent learnable affine transform of the space basis and the offset of the resulting inner product, respectively. Vanilla attention (or biased attention) is a special case of hyper-attention in Eq. (S6) given an identity $\mathbf{W}_{ij}$ and

zero (or non-zero) b_{ij}. In EvoGen we adopted rotary positional embedding (RoPE)[23] as $\mathbf{W}_{ij}$, so that $\mathbf{W}_{ij}$ can be decomposed into product of two position-dependent vectors and merged with the linear transform of Query and Key vectors. The second difference lies in the embedding of relative positions. We adopted an approach similar to T5 model[24] and grouped $|i-j|$ into discretized buckets according to log-scales[25]. This way of relative positional embedding not only expands the horizon of sequence models without inducing extra memory cost, but also equips the model with a hyperbolic view of distances as inductive biases. Thirdly, we added a new Query Conditioning Module into EvoGen encoder (Fig. 1c), which is a neural network that mixes context activations with query activations in order to help the model learn relative differences between MSA and the target sequence. Lastly, similar to AlphaFold-Multimer[26], we changed the order of the "outer product mean" operation to the beginning of each Hyperformer block (Fig. 1c), allowing the single update and pair update to be executed in parallel and separately.

3. Latent module

Latent module is designed to summarize the statistics of MSA features. The Statistics Module in the Latent modules in encoder (matching network 1 in Fig. 1d) is responsible for summarizing target sequences into deviations with respect to the data-dependent priors, then the posteriors are calculated as the addition of priors and the corresponding deviations. Such posterior formula reflects the principle of "relativity" in the model design and stabilizes training, which was first observed in NAVE[7]. On the other hand, latent modules in decoder (matching network 2 in Fig. 1d) are responsible for estimating the priors according to context sequences. Specifically, the Statistics module take a set of sequence representations (Fig. 1b) from single-track as input activations, and it first performs average pooling over these activations, i.e., aggregates multiple activation tensors of many MSA sequences into a single activation tensor, followed by a MLP which further projects this pooled activation into the latent space. Note that by this means, the dimension can be different from the representation space. Finally, the Statistics Module outputs the statistics of the latent prior or posterior for the "re-parametrization trick"[4], which correspond to the means and variances of diagonal multivariate Gaussian distributions in

our case.

Given the overall symmetry between encoder and decoder (Fig. 1a) and the principle of “relativity”, we employed twin (or Siamese) matching networks to learn the relevant statistics (Fig. 1d). Another neural network called Sampling Module, draws random samples according to learned posteriors with Gaussian noises via re-parametrization trick[4]. During generation, only context sequences are provided to EvoGen and the model predicts the priors, according to which we can sample new sequences.

4. Model hyperparameters

EvoGen is composed of a pair of encoder and decoder with relative symmetry similar to U-Net[27], each consisting of 12 Hyperformer blocks. Similar to AF2, in Hyperformer, we set the dimension of sequence representation to be 256 and the dimension of pair representation to be 128 (Fig. 1b in the main text). Therefore, the scaling parameter for hyper-attention $d = 256$ in Eq. (S6). According to the principle of “hierarchy”, we adopted 3 Latent Module blocks between the encoder and decoder (Fig. 1a), with increasing latent dimensions (64, 128, 256, respectively) during encoding (or equivalently, decreasing dimensions during decoding).

Supplementary Figures

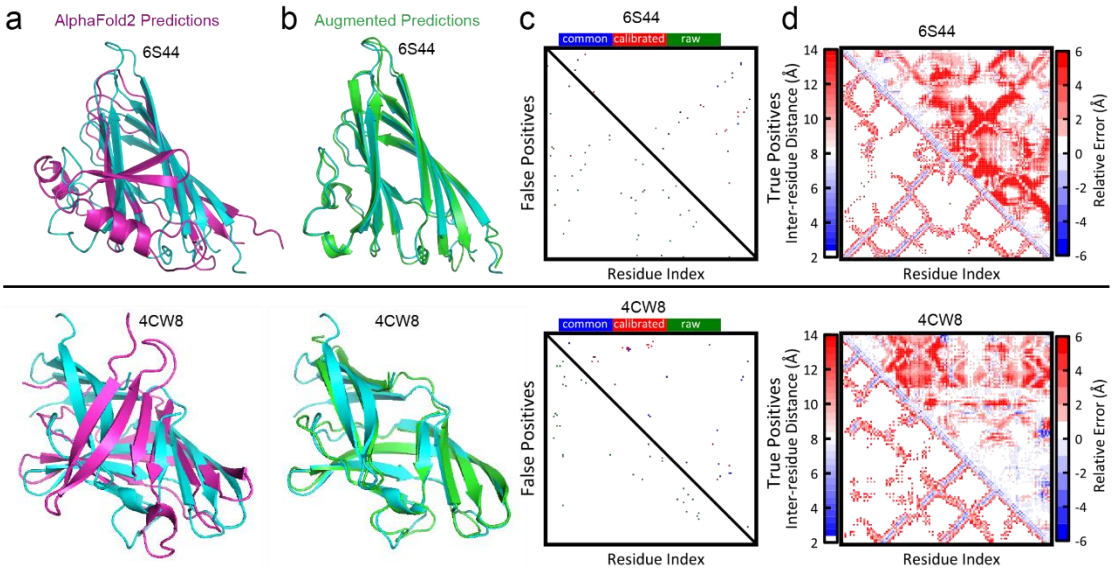

Figure S1. Calibrated MSA rescue structure predictions for two exemplary targets, 6S44 (PDB ID; upper panels) and 4CW8 (PDB ID; lower panels). **(a~b)** Reference PDB structure (cyan) is compared against 3D structures predicted by AlphaFold2 based on raw MSA (**a**; magenta) and calibrated MSA (**b**; green), respectively. **(c)** Contact maps predicted via DCA based on raw (green) and calibrated MSA (red). The upper-right matrix shows the true positive (TP) predictions, whereas the lower-left part shows false positive (FP) ones. TPs made commonly by the two approaches are colored in blue, while shared FPs are not shown. **(d)** Lower-left: Ground-true reference of the inter-residue distances with a cutoff of 14 A. Upper-right: The error of predicted Inter-residue distances based on raw MSA against predictions from calibrated MSA.

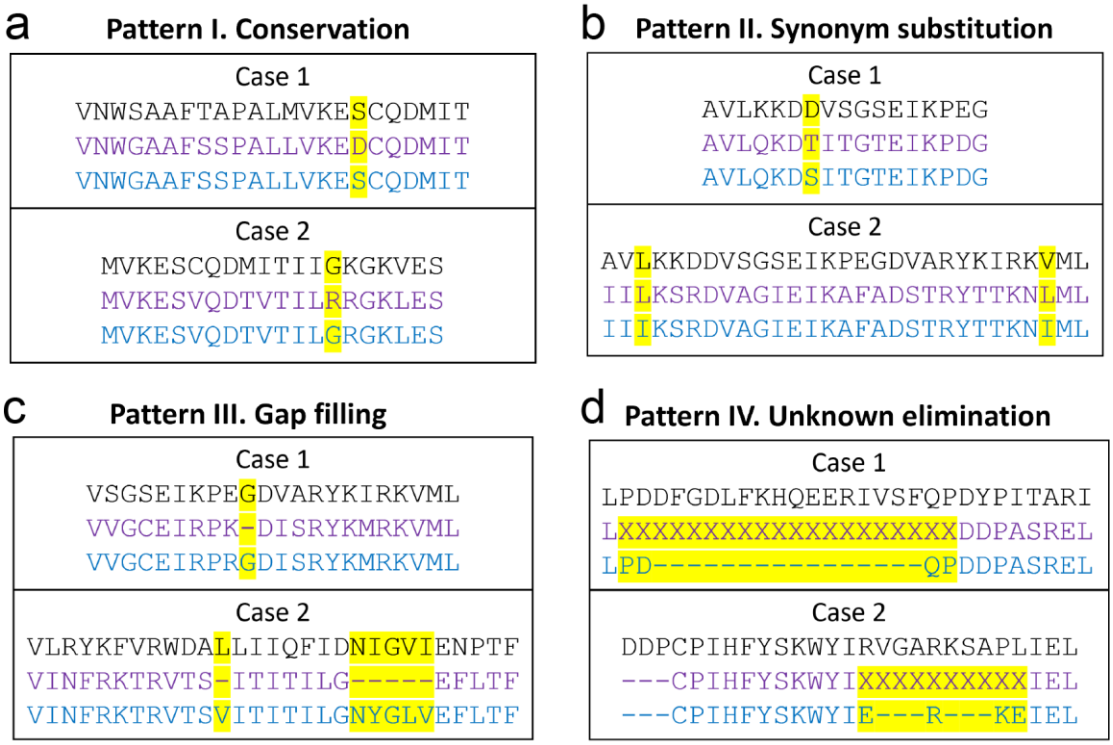

Figure S2. Patterns of calibrated MSA generated by EvoGen. Target sequences (black), original homologue sequence (purple) and calibrated homologue sequence (blue) are aligned as in MSA. **(a)** Recovering a mutation in a homolog sequence into the original amino acid type in the target sequence. **(b)** Replacing a residue with a synonymous residue with similar physical and chemical properties. **(c)** Filling alignment gaps with reasonable residues. **(d)** Reducing unknown residues (X) by meaningful residues and gaps.

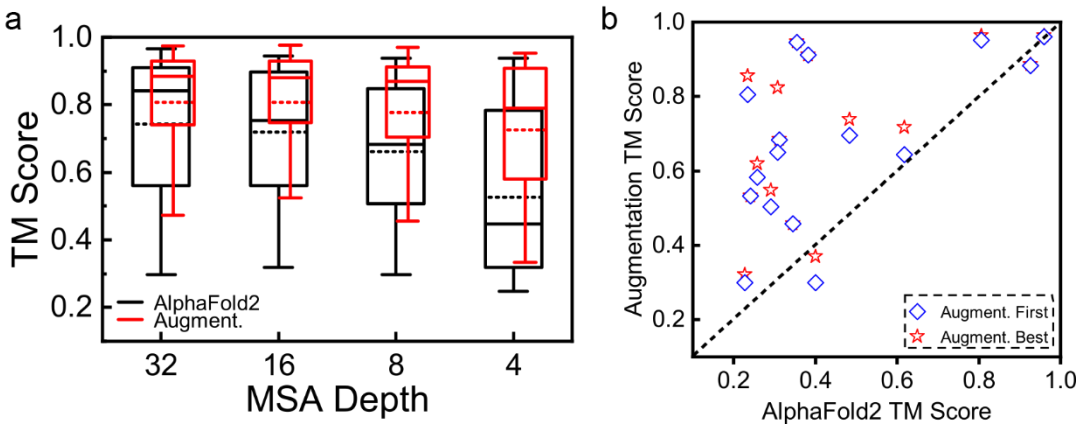

Figure S3. Benchmark MSA augmentation on CASP15 test set. **(a)** Performance of AF2 over CASP15 targets at varied MSA depths without (black boxes) and with (red boxes) MSA augmentation. **(b)** The quality of single-sequence AF2 predictions with (red stars) and without MSA augmentation over CASP15 test set targets. The predicted structures were ranked according to the predicted confidence (i.e., averaged per-residue plDDT) and the most confident structure (called “first”; symbolized by blue squares) was reported. Besides, the best scored structure (called “best”; symbolized by red stars) assuming the ground-true score is known.

References

1 Liu, S. *et al.* PSP: Million-level Protein Sequence Dataset for Protein Structure Prediction. *arXiv preprint arXiv:2206.12240* (2022).

2 Xu, J., Mcpartlon, M. & Li, J. Improved protein structure prediction by deep learning irrespective of co-evolution information. *Nature Machine Intelligence* **3**, 601-609 (2021).

3 Suzek, B. E. *et al.* UniRef clusters: a comprehensive and scalable alternative for improving sequence similarity searches. *Bioinformatics* **31**, 926-932 (2015).

4 Kingma, D. P. & Welling, M. (2014).

5 Ho, J., Jain, A. & Abbeel, P. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems* **33**, 6840-6851 (2020).

6 Gregor, K., Danihelka, I., Graves, A., Rezende, D. & Wierstra, D. in *International conference on machine learning.* 1462-1471 (PMLR).

7 Vahdat, A. & Kautz, J. NVAE: A deep hierarchical variational autoencoder. *Advances in Neural Information Processing Systems* **33**, 19667-19679 (2020).

8 Bowman, S. R. *et al.* Generating sentences from a continuous space. *arXiv preprint arXiv:1511.06349* (2015).

9 Jumper, J. *et al.* Highly accurate protein structure prediction with AlphaFold. *Nature* **596**, 583-589 (2021).

10 Kingma, D. P. & Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

11 MindSpore. MindSpore. *https://www.mindspore.cn/*, doi:https://www.mindspore.cn/ (2020).

12 Jang, E., Gu, S. & Poole, B. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144* (2016).

13 Bengio, Y., Léonard, N. & Courville, A. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432* (2013).

14 Van Den Oord, A. & Vinyals, O. Neural discrete representation learning. *Advances in neural information processing systems* **30** (2017).

15 Mirdita, M. *et al.* ColabFold: making protein folding accessible to all. *Nature Methods*, 1-4 (2022).

16 Del Alamo, D., Sala, D., Mchaourab, H. S. & Meiler, J. Sampling alternative conformational states of transporters and receptors with AlphaFold2. *Elife* **11**, e75751 (2022).

17 Zhang, Y. & Skolnick, J. TM-align: a protein structure alignment algorithm based on the TM-score. *Nucleic acids research* **33**, 2302-2309 (2005).

18 Morcos, F. *et al.* Direct-coupling analysis of residue coevolution captures native contacts across many protein families. *Proceedings of the National Academy of Sciences* **108**, E1293-E1301 (2011).

19 Zerihun, M. B., Pucci, F., Peter, E. K. & Schug, A. pydca v1. 0: a comprehensive software for direct coupling analysis of RNA and protein sequences. *Bioinformatics* **36**, 2264-2265 (2020).

20 Senior, A. W. *et al.* Improved protein structure prediction using potentials from deep learning. *Nature* **577**, 706-710 (2020).

21 Zhang, J., Zhou, Y., Lei, Y.-K., Yang, Y. I. & Gao, Y. Q. Molecular CT: Unifying Geometry

and Representation Learning for Molecules at Different Scales. *arXiv preprint arXiv:2012.11816* (2020).

22 Ha, D., Dai, A. & Le, Q. V. Hypernetworks. *arXiv preprint arXiv:1609.09106* (2016).

23 Su, J., Lu, Y., Pan, S., Wen, B. & Liu, Y. Roformer: Enhanced transformer with rotary position embedding. *arXiv preprint arXiv:2104.09864* (2021).

24 Raffel, C. *et al.* Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.* **21**, 1-67 (2020).

25 Wu, K., Peng, H., Chen, M., Fu, J. & Chao, H. in *Proceedings of the IEEE/CVF International Conference on Computer Vision.* 10033-10041.

26 Evans, R. *et al.* Protein complex prediction with AlphaFold-Multimer. *BioRxiv* (2021).

27 Ronneberger, O., Fischer, P. & Brox, T. in *International Conference on Medical image computing and computer-assisted intervention.* 234-241 (Springer).