nature machine intelligence

Supplementary information https://doi.org/10.1038/s42256-024-00838-2

Accurate and robust protein sequence design with CarbonDesign

In the format provided by the authors and unedited

1 Supplementary Materials

Supplementary Notes

List of Supplementary Figures

List of Supplementary Tables

1.1 Details on running the compared methods

ESM-IF. We utlize the test script provided in the ESM GitHub repository (https://github.com/facebookresearch/esm/tree/main/examples/inverse_folding), with the model esm_if1_gvp4_t16_142M_UR50 and all other default settings.

ProteinMPNN. ProteinMPNN offers multiple models based on varying noise levels. For a more comprehensive comparison, we use the ProteinMPNN (default) model with 0.2Å noise and the ProteinMPNN (v_48_002) model with 0.02Å noise. We use the testing scripts of ProteinMPNN from the ProteinMPNN GitHub repository (https://github.com/dauparas/ProteinMPNN). Except for our selection of different models for testing, all parameter settings employ the default options provided by GitHub.

ProDESIGN-LE. We utilized all sequences designed by the ProDESIGN-LE provided server (http://falcon.ictbda.cn:89/serving2/submit/aFGjrWnGyA/?app=prodesign). All parameters were selected according to the default settings of this method.

ABACUS-R. We utilized the test script provided on the GitHub (https://github.com/liuyf020419/ABACUS-R/tree/main/demo) for protein sequence design through ABACUS-R. All parameters were selected from the default options provided in the config file on the website.

ESM-1v. In predicting functional effects of variants, we employed ESM-1v as the benchmark criterion. We use the testing script of ESM-1v in the ESM GitHub repository (https://github.com/facebookresearch/esm/tree/main/examples/variant-prediction). All the hyper-parameters are default.

ProGen2. We use the model ProGen2-xLarge (6.4B). The GitHub repository is (https://github.com/salesforce/progen/tree/main/progen2). All hyper-parameters are default.

1.2 Additional details on ablation models

The ablation models we trained include:

1. Based on CarbonDesign (default), we removed the network recycling, and this model will also disable the language model added during the recycling stages.
2. Based on CarbonDesign (default), we removed the pairwise amino acid head.
3. Based on CarbonDesign (default), we removed the side chain head during training.

	CAMEO	CASP15
CarbonDesign (default)	**0.60**	**0.54**
no network recycling and language model	0.52	0.48
no pairwise amino acid head	0.51	0.47
no side chain head	0.59	0.52

Table S1: Evaluation of ablation models on CAMEO and CASP15 testing sets.

1.3 Intuitive connections

During the encoding of backbone structures, the **direct** operation on nodes and edges plays a crucial role in determining the information flow and learning their representations.

ProteinMPNN and ESM-IF utilize different approaches for node and edge encoding. In ProteinMPNN, a graph neural network is used, while ESM-IF employs a Geometric Vector Perceptron (GVP) [62] for this task. Information on each edge in these models is updated based on the edge itself and its related edges (Figure S1**a**).

In contrast, CarbonDesign's Inverseformer uses triangular attention updates on edges, where the representation of each edge is updated by considering the representations of edges sharing a node (Figure S1**b**). This approach is inspired by AlphaFold's Evoformer, where triangular edge updates are motivated by the need to satisfy the triangle inequality constraints on residue-residue distances. In CarbonDesign, we establish an intuitive connection between triangular edge updates in sequence design and the Belief Propagation algorithm used in probabilistic graphical models.

In probabilistic models like Bayesian networks and Markov Random Fields, a graph $G = (V, E)$ is employed to describe the joint distribution of $P(X_1, X_2, ..., X_n)$ for n random variables (Figure S1**c**). Each variable x_i is represented as a node, and edges between variables represent direct correlations. The Belief Propagation algorithm aims to calculate the marginal distribution of a specific variable or a subset of variables by iteratively aggregating probability mass from neighboring nodes. Specifically, $m_{ji}(x_j)$ represents the "belief" of variable x_i based on variable x_j, and it is updated by aggregating information from all edges jk $(k \neq i)$ connected to node j (Equation 8).

$$m_{ji}(x_i) = \sum_{x_j} \left(\phi(x_j)\phi(x_i, x_j) \prod_{k \in N(j), k \neq i} m_{kj}(x_j) \right) \tag{8}$$

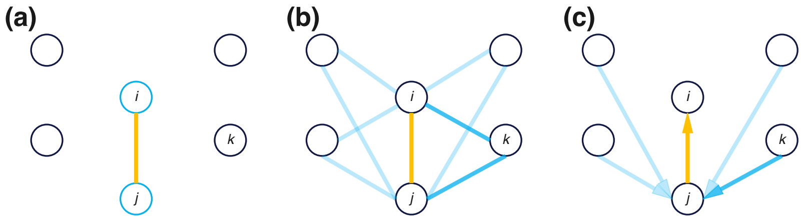

Fig. S1: **Edge update in ProteinMPNN, Inverseformer, and Belief Propagation algorithm. a,** ProteinMPNN updates representation of edge ij using the edge itself and the nodes i and j. **b,** Inverserformer updates representation of edge ij using the information from all related edges ik and jk $(i, j \neq k)$. **c,** BP algorithm updates the belief on edge ij using beliefs from all edges jk connected to j $(k \neq j)$.

1.4 Global inference mode for the amortized MRF model

In the MRF-sequence module, we leverage a *local inference mode* to generate intermediate sequences (see Methods in the main text) and a *global inference mode* to produce the final designed sequences (Algorithm 2), respectively. Since it is computationally infeasible to determine the sequences that exactly maximize the full likelihood under the MRF model (Equation 3), we use an efficient and straightforward greedy approach for approximation.

We initialize the sequence with the *local inference mode*, denoted as x^{intmd}. Subsequently, we update each amino acid by maximizing its conditional likelihood given the identities of other amino acids:

$$x_i^{\text{final}} = \arg\max_{x_i} \mathrm{P}(X_i = x_i | \, X_{\neg i} = x_{\neg i}^{\text{intmd}}; \, \mathbf{s}, \mathbf{z}), \quad i = 1, 2, 3, ..., L; \tag{9}$$

The conditional likelihood involves both the conservation bias term $h_i(x_i \mid \boldsymbol{s}_i)$ and the pairwise coupling term $e_{ij}(x_i, x_j \mid \boldsymbol{z}_{ij})$, and it can be calculated efficiently as follows:

$$\mathrm{P}(X_i = x_i | \, X_{\neg i} = x_{\neg i}^{\text{intmd}}; \, \mathbf{s}, \mathbf{z}) = \frac{1}{Z_i} \exp\{h_i(x_i \mid \mathbf{s}_i) + \sum_{i \neq j} e_{ij}(x_i, x_j^{\text{intmd}} \mid \mathbf{z}_{ij})\} \tag{10}$$

Here, Z_i is the local partition function that sums over all 20 possible amino acid types at position i. For both training and inference, we only include edges for neighboring residues within a $C_\beta - C_\beta$ distance of 8Å.

We introduce a temperature parameter, T, into CarbonDesign to regulate the diversity of the designed sequences. This parameter enables CarbonDesign to produce a set of sequences for a provided backbone structure. The conditional likelihood can be calculated as follows:

$$
\begin{aligned}
&\mathrm{P}(X_i = x_i |\ X_{\neg i} = x_{\neg i}^{\mathrm{intmd}};\ \mathbf{s}, \mathbf{z}, T) \\
&= \frac{1}{Z_i^T} \exp\{\frac{1}{T}[h_i(x_i \mid \mathbf{s}_i) \sum_{i \neq j} e_{ij}(x_i, x_j^{\mathrm{intmd}} \mid \mathbf{z}_{ij})]\}
\end{aligned} \tag{11}
$$

Here, T represents the sampling temperature. Lower sampling temperatures lead to a more concentrated distribution, tending to yield more accurate sequences, whereas higher temperatures lead to generating more diverse sequences.

We alternately update each amino acid, and after completing updates for the entire sequence, we proceed to the next round of updates until the sequence converges. Typically, sequences converge within 2 rounds of updating, and we set the maximum number of rounds as 3. We note that during the inference of the MRF model, both $\mathbf{s}_i$ and $\mathbf{z}_{ij}$ are held constant and treated as static inputs, and there is no need to run Inverformer to update them in the process.

Algorithm 2 Global inference mode of the MRFs model

```
function GlobalInference($\boldsymbol{x}^{\mathrm{intmd}}$, $\{h_i(x_i|\mathbf{s}_i)\}$, $\{e_{ij}(x_i, x_j|\mathbf{z}_{ij})\}$, $N_{\max}$=3, $T$)
    #$h_i(x_i|\ \mathbf{s}_i) \in \mathbb{R}^{20}$, $e_{ij}(x_i, x_j|\ \mathbf{z}_{ij}) \in \mathbb{R}^{20\times 20}$
    $\boldsymbol{x}_0^{\mathrm{intmd}} \leftarrow \boldsymbol{x}^{\mathrm{intmd}}$
    Indices $\leftarrow$ randomOrderIndices($\{0, 1, \cdots, L-1\}$)
    for m = 1, 2,..., $N_{\max}$ do
        for $i$ in Indices do
            #Update $x_i$ using equation 9 or 11
            if T=0 then
                $x_{i,m}^{\mathrm{intmd}} \leftarrow \arg\max_{x_i}\ \mathrm{P}(X_i = x_i|\ X_{\neg i} = x_{\neg i,m-1}^{\mathrm{intmd}};\ \mathbf{s}, \mathbf{z})$
            else
                $x_{i,m}^{\mathrm{intmd}} \leftarrow$ Sampling($\mathrm{P}(X_i = x_i|\ X_{\neg i} = x_{\neg i,m-1}^{\mathrm{intmd}};\ \mathbf{s}, \mathbf{z}, T)$)
            end if
        end for
        $\boldsymbol{x}^{\mathrm{final}} \leftarrow \boldsymbol{x}_{N_{\max}}^{\mathrm{intmd}}$
    end for
    return $\boldsymbol{x}^{\mathrm{final}}$
end function
```

1.5 Model inference

CarbonDesign consists of two main components: Inverseformer blocks and the MRF-Sequence Module. The Inverseformer blocks take input backbone features as initial representations to compute updated representations. Subsequently, the MRF-Sequence Module utilizes these representations to generate intermediate sequences, final designed sequences, and corresponding side chain structures.

For inference, the whole network is executed sequentially N_{cycle} times, where the output single and pair representations of the former execution are recycled as inputs for the next execution (Algorithm 3). During the recycling phase, the intermediate sequence is inferred using the MRF model, and additional recycling features are extracted from the protein language model ESM2 by obtaining embeddings of the sequence.

In the MRF-Sequence Module, we employ an efficient *local inference mode* and a more accurate *global inference mode* for generating intermediate and final designed sequences, respectively. The *local inference mode* utilizes only the *conservation bias* term to infer the intermediate designed sequence:

$$x_i^* = \arg\max_{x_i} \frac{1}{Z_i} \exp(h_i(x_i|\, \mathbf{s}_i)) \tag{12}$$

Here, Z_i represents the local partition function involving only the conservation bias terms at position i. In contrast, the global inference mode optimizes the sequence by maximizing the sequence probability under the MRF model, considering both the *conservation bias* term and the *pairwise coupling* term (Equation 12). The efficient local inference mode allows obtaining the embeddings of intermediate sequences in a computationally feasible manner. Since exact optimization is challenging for the global mode, we initialize the inference using the sequence from the local inference mode and update sequences using a fast greedy algorithm (Supplementary Note 2).

During the inference stage, when the types of amino acids are unknown, we first utilize the single presentation $\mathbf{s}_i$ to predict the side chain structures $\boldsymbol{x}_{i,a}^{\text{sidechain}} \in \mathbb{R}^{b\times 3}$ for all possible amino acids, where b represents the number of side chain atoms and a covers 20 amino acid types. The final side chain structures are materialized from $\boldsymbol{x}_{i,a}^{\text{sidechain}}$ once the final designed sequence is determined by the *global inference mode* of the MRFs model.

Algorithm 3 CarbonDesign Model Inference

```
function DesignSequence($\{\boldsymbol{x}_i^{\text{backbone}}\}$, $N_{\text{recycle}} = 3$, $N_{\text{blocks}} = 12$)
    #compute input node and edge features (see Input features in Methods)
    $\mathbf{d}_{ij} \leftarrow \|\boldsymbol{x}_i^{\text{backbone}} - \boldsymbol{x}_j^{\text{backbone}}\|_2$
    $\mathbf{d}_{ij} \leftarrow$ oneHotEncoding($\mathbf{d}_{ij}$, $\text{v}_{\text{bins}} = [\frac{3}{4}\text{Å}, \frac{3}{2}\text{Å}, ..., 15\text{Å}]$ )
    $\boldsymbol{t}_i \leftarrow$ computeLocalOrientations($\{\boldsymbol{x}_i^{\text{backbone}}\}$)

    #initialize recycling features as 0
    $\mathbf{s}_i^{\text{prev}}, \mathbf{z}_{ij}^{\text{prev}} = \mathbf{0}$

    for m = 1, 2, ..., $N_{\text{recycle}}$ do        # shared weights during recycling
        $\mathbf{s}_i \leftarrow$ Linear(Relu(Linear($\boldsymbol{t}_i$)))
        $\mathbf{z}_{ij} \leftarrow$ Linear(Relu(Linear($\boldsymbol{d}_{ij}$)))

        $\mathbf{z}_{ij} \leftarrow \mathbf{z}_{ij} +$ PariwiseRelativePositionEmbedding($i, j$)

        $\mathbf{s}_i \leftarrow \mathbf{s}_i +$ Linear(Relu(Linear($\mathbf{s}_i^{\text{prev}}$)))
        $\mathbf{z}_{ij} \leftarrow \mathbf{z}_{ij} +$ Linear(Relu(Linear($\mathbf{z}_{ij}^{\text{prev}}$)))

        for n = 1, 2, 3, ..., $N_{\text{blocks}}$ do
            $\mathbf{s}_i, \mathbf{z}_{ij} \leftarrow$ Inverseformer($\mathbf{s}_i$, $\mathbf{z}_{ij}$)
        end for

        #generate intermediate sequence using local inference mode (Equation 12)
        $\boldsymbol{x}^{\text{intermediate}} =$ MRFLocalInference($\mathbf{s}_i$)

        #extract embedding of intermediate sequence using ESM2
        $\{\boldsymbol{e}_i\} =$ EmbeddingFromESM2($\boldsymbol{x}^{\text{intermediate}}$)

      #update initial single and pair representations for next cycle
        $\mathbf{s}_i^{\text{prev}} \leftarrow \mathbf{s}_i +$ Linear($\boldsymbol{e}_i$)
        $\mathbf{z}_{ij}^{\text{prev}} \leftarrow \mathbf{z}_{ij}$
    end for

    #predict side chain angle $\chi_1$,$\chi_2$,$\chi_3$,$\chi_4$ for all possible amino acid types
    $\vec{\alpha}^{f}_{i,a}$ = Linear(ReLU($\mathbf{s}_i$))    # $\vec{\alpha}^{f}_{i,a} \in \mathbb{R}^2$, $f \in S_{\text{torsion names}}$, $a \in$ 20 amino acid types
     #calculate atom coordinates from torsion angles following AlphaFold
    $\boldsymbol{x}_{i,a}^{\text{sidechain}} =$ computeSC($\boldsymbol{x}_i^{\text{backbone}}$, $\vec{\alpha}^{f}_{i,a}$) # $\boldsymbol{x}_{i,a}^{\text{sidechain}} \in \mathbb{R}^{b\times 3}$

     #generate final sequence using global inference mode (Algorithm 2)
    $\boldsymbol{x}^{\text{final}} \leftarrow$ MRFGlobalInference($\boldsymbol{x}^{\text{intermediate}}, \{\mathbf{s}_i\}, \{\mathbf{z}_{ij}\}$)

    $\boldsymbol{x}_i^{\text{sidechain*}} \leftarrow$ extractSC($\boldsymbol{x}_{i,a}^{\text{sidechain}}, \boldsymbol{x}_i^{\text{final}}$)
    return $\boldsymbol{x}^{\text{final}}, \boldsymbol{x}^{\text{sidechain*}}$
end function
```

Supplementary Figures

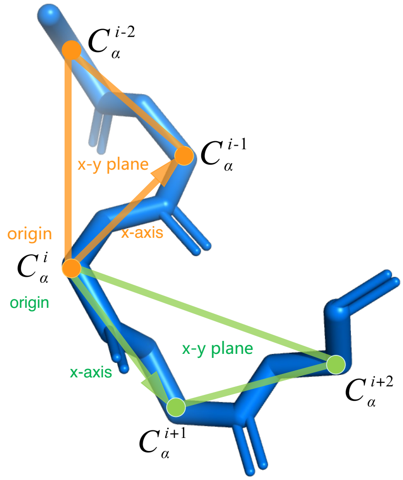

Fig. S2: **Computing local orientations of C_α atoms in backbone structures** . We utilize the Gram-Schmidt process to calculate the local frame formed by the C_α^{i-2}, C_α^{i-1}, and C_α^{i} atoms. Subsequently, we represent the local orientation of C_α^{i+1} as its local coordinate in the frame. Similarly, we calculate the local orientation of the C_α^{i-1} with respect to the C_α^{i}, C_α^{i+1}, and C_α^{i+2} atoms.

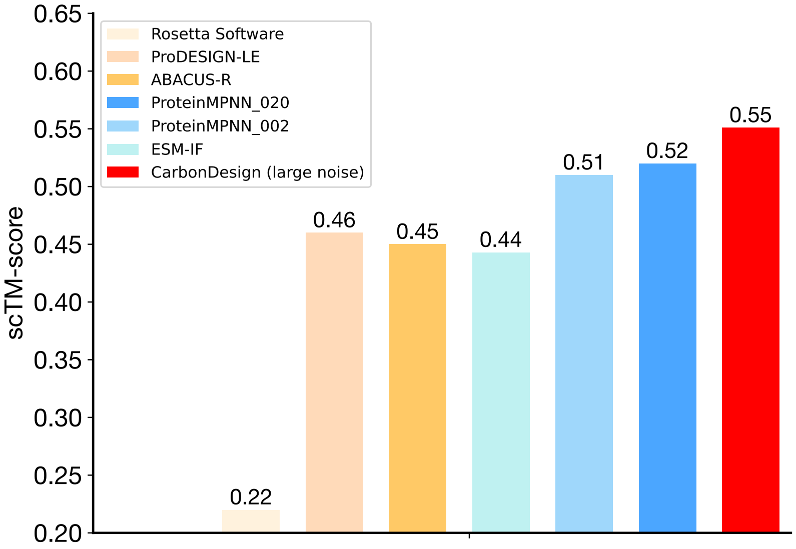

Fig. S3: **Average scTM-score on *de novo* backbone structures from FrameDiff across various methods.** We evaluate proteins of length 200, 300, and 400 generated by the improved version of FrameDiff. We generated 128 backbone structures for each length.

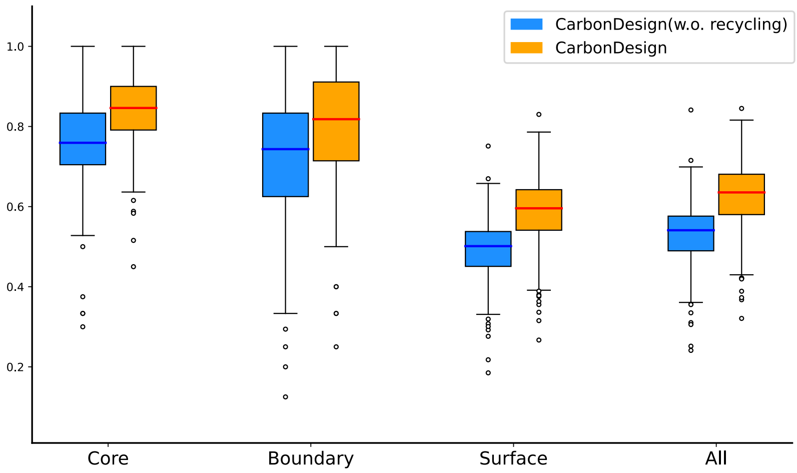

Fig. S4: **Evaluation of CarbonDesign on protein core and surface regions.** The relative solvent-accessible surface area (RSA) for each residue is calculated and categorized into Core (< 0.25), Boundary (0.25-0.75), and Surface (> 0.75) regions. Sequence recovery rates are evaluated for both the CarbonDesign default model and the model with the recycling and protein language model excluded.

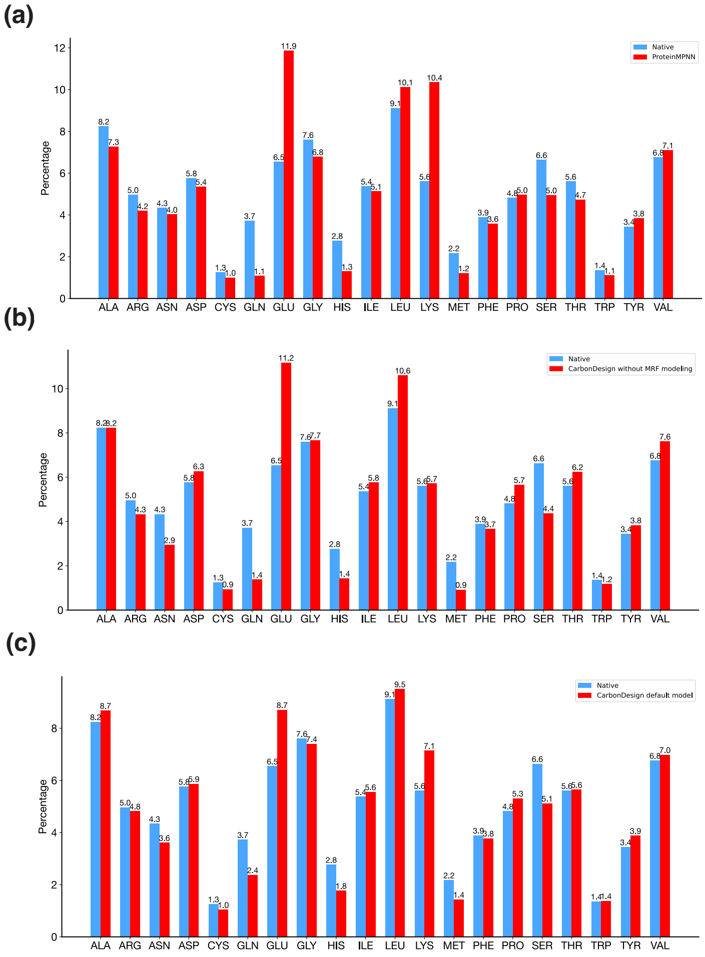

Fig. S5: **Distributions of amino acid types in designed and native sequences.a,** A comparison between the distributions of amino acid types in sequences designed by ProteinMPNN and native sequences. **b,** A comparison between the distributions of amino acid types in sequences designed by CarbonDesign without MRF modeling and native sequences. **c,** A comparison between the distributions of amino acid types in sequences designed by CarbonDesign and native sequences.

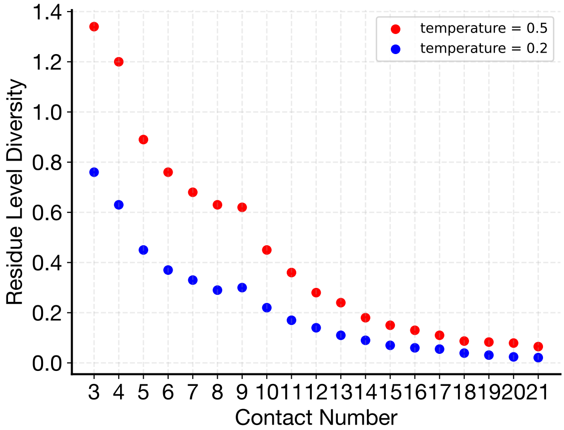

Fig. S6: **Correlation between the residue-level diversity of designed sequences and the contact numbers.** We measure residue-level diversity via the entropy of the amino acid distribution at specific sites and quantify structural context constraints by the number of residues within an 8 Å radius of each amino acid.

Supplementary Tables

T1104	T1106s2	T1109	T1110	T1112	T1113
T1114s1	T1114s2	T1114s3	T1120	T1121	T1122
T1123	T1124	T1125	T1127	T1129s2	T1130
T1131	T1132	T1133	T1134s1	T1134s2	T1137s1
T1137s2	T1137s3	T1137s4	T1137s5	T1137s6	T1137s7
T1137s8	T1137s9	T1139	T1145	T1146	T1147
T1150	T1151s2	T1153	T1154	T1155	T1157s1
T1157s2	T1158	T1159	T1162	T1163	T1169
T1170	T1173	T1174	T1175	T1176	T1177
T1178	T1179	T1180	T1181	T1182	T1183
T1184	T1186	T1187	T1188	T1194	

Table S2: List of protein names in the CASP15 testing set.

8g4u_A	7y4i_A	7rcw_A	7bi4_A	7v53_A	7pyv_B
7vyx_A	7nsn_A	T1125	T1157s1	T1154	T1158
T1169					

Table S3: List of protein names in the testing set of long proteins.

T1122	T1130	T1131	T1125	T1113	T1178
T1184	T1155	T1129s2			

Table S4: List of protein names in the testing set of orphan proteins.

		Protein	Protein		
ProDESIGN-LE	ABACUS-R	MPNN_002	MPNN_020	ESM-IF	CarbonDesign
36.4%	36.3%	46.9%	41.8%	32.6%	**55.1%**

Table S5: **Evaluation on the testing set of long proteins measured with sequence recovery rate.** The table presents the results for 13 proteins with more than 800 amino acids collected from both the CASP15 and CAMEO datasets. The average protein length in this set is 1239 amino acids.

ProDESIGN-LE	ABACUS-R	Protein MPNN_002	Protein MPNN_020	ESM-IF	CarbonDesign
38.9%	32.6%	38.5%	44.3%	46.2%	**49.1%**

Table S6: **Evaluation on the testing set of orphan proteins measured with sequence recovery rate.**

Methods	BRCA1	PTEN	TP53	MSH2	average
ESM-1v	0.896	**1.000**	**0.994**	0.812	0.926
ProGen2	0.876	**1.000**	0.952	**0.844**	0.918
CarbonDesign	**0.933**	0.986	0.984	0.822	**0.931**

Table S7: **Evaluation of CarbonDesign in predicting pathogenicity of variants with the testing set of clinically curated variants in ClinVar.**

Methods	Length 200	Length 300	Length 400	Length 500	Length 600
CarbonDesign(small noise)	0.84	0.69	0.58	0.58	0.48
CarbonDesign(high noise)	**0.89**	**0.80**	**0.74**	**0.64**	**0.54**

Table S8: **Evaluation on *de novo* backbone structures from RFDiffusion at varying noise levels, measured using scTM score.**

Single representation dimension	384
Pair representation dimension	128
Number of heads	8
Number of Inversformer blocks	12
Protein crop size during training	400
Dropout rate during training	0.1

Table S9: **Hyperparameters of CarbonDesgin architecture**

Temperature	0.0	0.1	0.2	0.3	0.4	0.5
Sequence recovery rate	**60.1%**	58.4%	58.3%	57.9%	57.4%	55.7%
Sequence Level Diversity	0.000	0.067	0.124	0.175	0.226	0.272

Table S10: **Sequence Recovery Rate on CAMEO dataset across various temperatures.** We generate 50 sequences per backbone structure and assess sequence-level diversity via average sequence similarity.

Temperature	0.0	0.1	0.2	0.3	0.4	0.5
scTM-score	0.801	**0.809**	0.799	0.783	0.773	0.716
Sequence Level Diversity	0.000	0.132	0.214	0.297	0.375	0.462

Table S11: **scTM-score on proteins with length 300 in *de novo* backbone structures across various temperatures.** We generate 50 sequences per backbone structure and assess sequence-level diversity via average sequence similarity.

Methods	Rosetta energy CAMEO	Rosetta energy CASP15
ProDESIGN-LE	-2.06	-1.51
ABACUS-R	-1.87	-1.25
ProteinMPNN_020	-2.25	-1.56
ProteinMPNN_002	-2.12	-1.47
ESM-IF	-2.11	-1.46
Rosetta Software	**-3.54**	**-2.01**
CarbonDesign	-2.32	-1.65

Table S12: **Evaluation on the testing set of CASP15 and CAMEO measured with Rosetta energy.**

Methods	Rosetta energy
ProDESIGN-LE	-3.29
ABACUS-R	-3.27
ProteinMPNN_020	-3.30
ProteinMPNN_002	-3.25
ESM-IF	-3.27
Rosetta Software	**-3.84**
CarbonDesign	-3.48

Table S13: **Evaluation on the testing set of *de novo* backbone structures measured with Rosetta energy.**

Categories	Methods	Spearman correlation
MSA free methods	Tranception-L	0.396
	RITA-XL	0.397
	ProGen2-XL	0.412
	ESM-1v	0.394
	CarbonDesign	**0.434**
MSA-based methods	Tranception-L (with retrieval on MSA)	0.444
	MSA-Transformer	0.428
	DeepSequence	0.429
	EVE	**0.457**
Ensemble methods	CarbonDesign+MSA-Transformer	0.485
	CarbonDesign+Tranception	0.489
	CarbonDesign+DeepSequence	0.489
	EVE+Tranception	0.479
	CarbonDesign+EVE	**0.501**

Table S14: **Evaluation on variants from 49 deep mutational scanning essays.** We conducted an assessment of methods based on MSA free methods, MSA-based methods, and ensemble methods. Spearman correlation between the prediction scores and experimental validated functional scores of the variants is utilized as a metric.

Methods	CarbonDesign (w.o. LM)	CarbonDesign
Spearman correlation	0.392	**0.435**

Table S15: **Evaluation of the ablation model of CarbonDesign without using the pre-trained protein language model on the DMS testing set.** Spearman correlation between the prediction scores and experimental validated functional scores of the variants is utilized as a metric.

Methods	CarbonDesign (w.o. LM)	CarbonDesign
auROC	0.912	**0.933**

Table S16: **Evaluation of the ablation model of CarbonDesign without using the pre-trained protein language model on the ClinVar testing set.** We used auROC as an evaluation metric with clinical labels as ground truth.

Methods	scTM-score
ProDESIGN-LE	0.38
ABACUS-R	0.36
ProteinMPNN_020	0.38
ProteinMPNN_002	0.37
ESM-IF	0.35
Rosetta Software	0.23
CarbonDesign	**0.40**

Table S17: **Evaluation of CarbonDesign on de novo backbone structure with length of 500 and 600 generated from FrameDiff.** 128 backbone structures were generated for each length.