CarbonNovo: Joint Design of Protein Structure and Sequence Using a Unified Energy-based Model

Milong Ren [1 2] **Tian Zhu** [1 2] **Haicang Zhang** [# 1 2]

Abstract

De novo protein design aims to create novel protein structures and sequences unseen in nature. Recent structure-oriented design methods typically employ a two-stage strategy, where structure design and sequence design modules are trained separately, and the backbone structures and sequences are generated sequentially in inference. While diffusion-based generative models like RFdiffusion show great promise in structure design, they face inherent limitations within the two-stage framework. First, the sequence design module risks overfitting as the accuracy of the generated structures may not align with that of the crystal structures used for training. Second, the sequence design module lacks interaction with the structure design module to further optimize the generated structures. To address these challenges, we propose CarbonNovo, a unified energy-based model for jointly generating protein structure and sequence. Specifically, we leverage a score-based generative model and Markov Random Fields for describing the energy landscape of protein structure and sequence. In CarbonNovo, the structure and sequence design module communicates at each diffusion step, encouraging the generation of more coherent structure-sequence pairs. Moreover, the unified framework allows for incorporating the protein language models as evolutionary constraints for generated proteins. The rigorous evaluation demonstrates that CarbonNovo outperforms two-stage methods across various metrics, including designability, novelty, sequence plausibility, and Rosetta energy.

[*]Equal contribution [1]Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China. [2]University of Chinese Academy of Sciences, Beijing, China. Correspondence to: Haicang Zhang <zhanghaicang@ict.ac.cn>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

1. Introduction

Proteins perform most of the biological functions fundamental for life. The task of *de novo* protein design is to create new proteins and has broad applications in drug development (Khoury et al., 2014; Cao et al., 2020; Vorobieva et al., 2021) and enzyme engineering (Dou et al., 2018). Computational methods traditionally focus on directed evolution (Dougherty & Arnold, 2009; Arnold, 2015) and rational design of novel proteins guided by geometric principles (Polizzi & DeGrado, 2020) and existing energy function (O'Meara et al., 2015; Park et al., 2016).

Recent diffusion-based methods for *de novo* protein design methods have made great progress in generating novel and functional proteins (Trippe et al., 2022; Shi et al., 2022; Watson et al., 2023; Yim et al., 2023b; Ingraham et al., 2023; Lin & Alquraishi, 2023). For example, RFdiffusion (Watson et al., 2023) employs a Denoising Diffusion Probabilistic Model (DDPM) in structure space, while Chroma and FrameDiff apply a score-based diffusion model. Additionally, flow-based generative models have been proposed for protein structure design (Bose et al., 2023). All these methods follow a two-stage framework where protein structures are generated first, and optimal sequences are subsequently generated through sequence design models like ProteinMPNN (Dauparas et al., 2022) and ESM-IF (Hsu et al., 2022).

There are two primary issues in the two-stage framework. First, the structures generated by the structure design models, such as RFDiffusion, contain inherent noise and are not as accurate as the experimentally solved crystal structures on which the sequence design models are trained. Consequently, the sequence design models are susceptible to overfitting. Second, as the structure and sequence modules are trained separately and the two modules interact only after the structure is generated from the last diffusion step, the errors in the sequence design stage cannot provide feedback to further optimize the generated structures. While structure and sequence *co-design* methods have been proposed, primarily focusing on some special protein families like antibody design (Luo et al., 2022; Martinkus et al., 2023), these methods (Lisanza et al., 2023) have not yet outperformed the *two-stage* methods in designing general proteins (Ingra-

ham et al., 2023; Watson et al., 2023). Efficient training and inference of a joint model in discrete sequence space and SE(3) invariant structure space pose challenges.

We introduce CarbonNovo, a novel approach that simultaneously generates protein structure and sequence using a unified energy-based model applicable across all protein families. Specifically, we utilize a score-based diffusion model and Markov Random Fields (MRF) to characterize the energy landscape of protein structure and sequence. CarbonNovo addresses the main limitations of the two-stage approaches. First, the structure and sequence design modules are jointly trained, with sequence design modules directly leveraging generated structures instead of crystal structures. Second, in both the training and inference stages, the two modules communicate at each diffusion step, enabling generated sequences and structures to have chances to refine each other. Notably, our framework allows the integration of large-scale pre-trained language models like ESM-2 using the network recycling technique. The language models act as evolutionary constraints for the generated proteins, which have proven useful in protein-related tasks such as protein structure prediction (Lin et al.) and variant interpretation (Meier et al., 2021).

Our main contributions are summarized as follows:

- We develop CarbonNovo, a unified framework capable of simultaneously generating sequences and structures for general protein families.
- We are the first to integrate a protein language model to enhance the generation of both protein structure and sequences.
- We explore various techniques for efficient training and inference of the joint model, such as a multi-stage training strategy and the discrete version of M-H Langevin algorithm for sequence sampling.
- CarbonNovo demonstrates superior performance compared to two-stage approaches across various metrics, including designability, novelty, Rosetta energy, and sequence plausibility.

2. Related work

2.1. Diffusion-based Models for Protein Design

Diffusion-based generative models have demonstrated remarkable success in generating various data types, including images (Song et al., 2020b) and videos (Ho et al.; Harvey et al., 2022), and discrete data like text (Li et al., 2022). In the case of *de novo* protein structure design, diffusion-based models have been effectively applied to model the generative process of structures in $\mathbb{R}^3$ space or SE(3) space. Subsequently, these structures are employed in sequence design to generate the final complete proteins (Dauparas et al., 2022; Ingraham et al., 2023). Co-design methods for structure and sequence have also emerged, utilizing two diffusion processes to model the structure and sequence, respectively (Luo et al., 2022; Martinkus et al., 2023; Lisanza et al., 2023). While these methods have demonstrated outstanding results in designing specific families such as antibodies (Luo et al., 2022; Martinkus et al., 2023), their performance across all general protein families falls short of the current gold standard methods like RFdiffusion and Chroma in the two-stage approach.

The most related to our work is ProteinGenerator (Lisanza et al., 2023), Chroma (Ingraham et al., 2023), and CarbonDesign (Ren et al., 2024). In PtoteinGenerator, they only train a DDPM for sequence design and utilize RosettaFold to predict the structures of these sequences. While both CarbonNovo and Chroma utilize MRF for sequence design, the two methods differ in terms of model architecture, training, and inference strategies. The key difference is that Chroma falls into the category of a two-stage approach, whereas CarbonNovo jointly generates sequences and structures both during training and inference. Moreover, CarbonNovo constructs the diffusion process in $\mathrm{SE}(3)$ space, while Chroma models in coordinate $\mathbb{R}^3$ space. An additional feature of CarbonNovo is the integration of a pre-trained language model as prior knowledge. CarbonDesign focuses solely on protein sequence generation, using an Inverseformer for encoding backbone structures and an MRF module for decoding the sequence.

2.2. Energy-based Models

Energy-based models (EBMs) are a broad class of generative models that are grounded in Gibbs distributions and become more powerful with modern neural networks (LeCun et al., 2006; Song & Kingma, 2021). EBMs have been widely used in various domains, including image generation(Xie et al., 2016; Du & Mordatch, 2019), video generation (Xie et al., 2017), voxel generation (Xie et al., 2018), point cloud generation (Xie et al., 2021), text generation(Deng et al., 2019), and protein structure prediction (Levada et al., 2008; Ren et al., 2024). The utility of EBMs relies on various efficient learning and sampling algorithms (Song & Kingma, 2021), such as gradient-based MCMC sampling, score matching (Hyvärinen & Dayan, 2005; Song & Ermon, 2019; Song et al., 2020a;b), noise contrastive learning (Gutmann & Hyvärinen, 2010), and pseudo-likelihood (Levada et al., 2008) and composite likelihood approximations (Zhang et al., 2019).

Recent works unify diffusion-based generative models in the framework of EBMs in terms of both training objectives and inference strategies. Building on these perspectives

(Salimans & Ho, 2021; Liu et al., 2022a; Du et al., 2023), we integrate the structure design and sequence design modules into a unified energy-based framework, utilizing a score-based diffusion model for continuous structure space and a MRF model for discrete sequence space.

3. Methods

In Section 3.1, we present the preliminary concepts for protein design. In Section 3.2, we develop the generative process for protein sequence and structure using a unified EBM. In Section 3.3, we describe the model architecture of CarbonNovo. Sections 3.4 and 3.5 details the sampling and training algorithms, respectively.

3.1. Preliminaries on *de novo* Protein Struture and Sequence Design

CarbonNovo aims to jointly design protein backbone structure and sequence. We use $\mathbf{s} \in \mathbb{R}^{N\times 20}$ to represent the amino acid sequence, where N is the number of amino acids in the protein, and each amino acid has 20 types. $\mathbf{x} \in \mathbb{R}^{N\times 4\times 3}$ is used to denote the 3D coordinates of the protein backbone atoms, emphasizing that the backbone is composed of four atoms $\{C, C_\alpha, N, O\}$.

We adopt the backbone frame parameterization used in AlphaFold2 (Jumper et al., 2021) and FrameDiff (Yim et al., 2023b). Each *frame* $\mathbf{T} = (\mathbf{R}, \mathbf{t})$ comprises a rotation $\mathbf{R} \in \mathrm{SO}(3)$ and a translation $\mathbf{t} \in \mathbb{R}^3$. The rotation $\mathbf{R}$ is determined by the relative positions of the C_α, N, and C atoms, while the translation $\mathbf{t}$ corresponds to the coordinate of the C_α atom. The backbone oxygen atom is parameterized by an additional torsion angle ϕ, describing the rotation around the bond between C_α and C.

3.2. Jointly Modeling Structure and Sequence on EBMs

The density given by an EBM can be written as (Song & Kingma, 2021):

$$p_\theta(\mathbf{a}) = \frac{1}{Z_\theta}\exp[-E_\theta(\mathbf{a})]. \tag{1}$$

Here, $\mathbf{a}$, $-E_\theta(\mathbf{a})$, Z_θ represent a single variable, a learnable neural network, and a normalization factor, respectively.

To generate more coherent structure-sequence pairs, we construct a joint energy framework for protein structure and sequence. The joint distribution of protein structure $\mathbf{T}^{(0)}$ and sequence $\mathbf{s}$ is defined as:

$$p_\theta(\mathbf{T}^{(0)}, \mathbf{s}) = p_\theta(\mathbf{s}|\mathbf{T}^{(0)})\ \ p_\theta(\mathbf{T}^{(0)}) \tag{2}$$

Consequently, the joint energy of structure and sequence is represented as:

$$E(\mathbf{T}^{(0)}, \mathbf{s}) = E_{\text{str}}(\mathbf{T}^{(0)}) + E_{\text{seq}}(\mathbf{s}; \mathbf{T}^{(0)}) \tag{3}$$

Structure energy The score-based diffusion model has been recognized as an energy-based model (Du et al., 2023). We utilize the SE(3) score-based diffusion model to characterize the energy of the structure $E_{\text{str}}(\mathbf{T}^{(0)})$. The structure energy can be expressed as (Salimans & Ho, 2021; Liu et al., 2022a; Du et al., 2023):

$$E_{\text{str}}(\mathbf{T}^{(0)}) = -\int \mathcal{S}_\theta^{\mathrm{SE}(3)}(\mathbf{T}, t)dt \tag{4}$$

Here, $\mathcal{S}_\theta^{\mathrm{SE}(3)}(\mathbf{T}, t) = \{\mathcal{S}_\theta^{\mathbf{R}}(\mathbf{R}, t), \mathcal{S}_\theta^{\mathbf{t}}(\mathbf{t}, t)\}$ denotes the score of the corresponding distribution.

Sequence energy For the sequence energy $E_{\text{seq}}(\mathbf{s})$, we employ an MRF model, a widely used energy-based model in both protein structure prediction (Ekeberg et al., 2013; Zhang et al., 2019) and protein design (Ingraham et al., 2023; Ren et al., 2024).

The sequence energy under an MRF model is defined as:

$$E_{\text{seq}}(\mathbf{s}; \mathbf{T}^{(0)}) = -\Big[\sum_i \psi_s(\mathbf{s}_i|\mathbf{T}^{(0)}) + \sum_{i,j}\psi_p(\mathbf{s}_i, \mathbf{s}_j|\mathbf{T}^{(0)})\Big]. \tag{5}$$

Here, ψ_s and ψ_p represent the conservation bias and pairwise coupling terms from the MRF model, respectively. $\mathbf{T}_\theta^{(0)}$ is the final structure predicted by score network.

3.3. Model Architecture

The CarbonNovo network consists of two main components: the structure design module and the sequence design module (Figure 1). At each time step t, the network takes the noisy backbone structure $\mathbf{T}^{(t)}$ as inputs and outputs the refined backbone structure $\mathbf{T}^{(t-\Delta t)}$ along with the optimal sequence for this step.

Structure design module Unlike previous work such as FrameDiff (Yim et al., 2023b), which relies solely on an Invariant Point Attention (IPA) network for the structure module, CarbonNovo additionally incorporates Triangle Attention Networks from Evoformer (Jumper et al., 2021) into the structure design module. More details can be found in Appendix algorithm 1.

The input features of the structure design module include the timestep embedding, the distogram and frame representation $(\mathbf{R}^{(t)}, \mathbf{t}^{(t)})$ of the noisy structure $\mathbf{T}^{(t)}$, and recycling features.

Instead of directly predicting the score and the refined structure $\mathbf{T}^{(t-\Delta t)}$, we predict the final structure $\hat{\mathbf{T}}^{(0)}$ from which both the score and $\mathbf{T}^{(t-\Delta t)}$ can be derived (Appendix B). This approach has two advantages: First, it allows us to impose more constraints on the final structure using auxiliary losses during training (Yim et al., 2023b; Watson et al., 2023). Second, it enables the prediction of the optimal sequence from the final structure at this step.

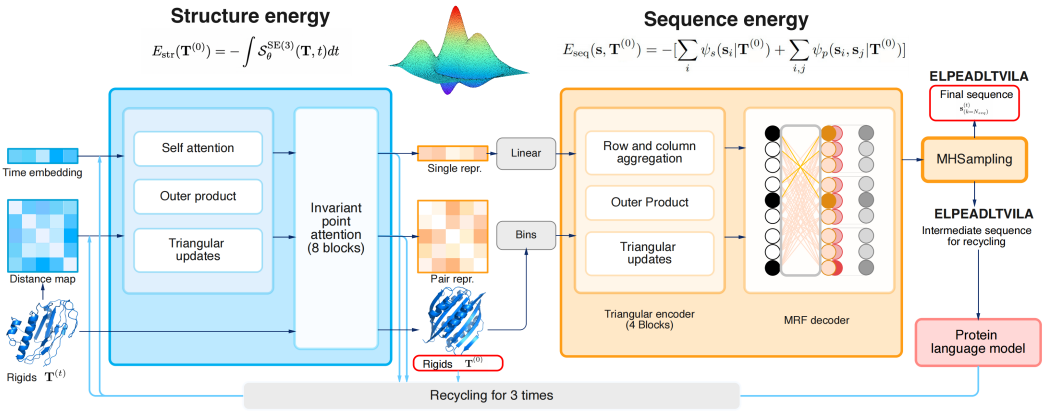

Figure 1. CarbonNovo architecture which jointly generates protein backbone structure and sequence.

Sequence design module For the sequence design module, we also adopt a Triangle Attention Network (Appendix Algorithm 2) which is adapted from Evoformer (Ren et al., 2024; Jumper et al., 2021).

The input features of the sequence design module include single and pair representations from the structure design module, the histogram of the predicted backbone structure $\hat{\mathbf{T}}^{(0)}$, and the recycling features.

The conservation bias terms and pairwise coupling terms in the MRF model (Equation 5) are then parameterized using the updated single and pair representations from the sequence design module. During training, we adopt a composite likelihood approximation to optimize the MRF model (Ren et al., 2024; Zhang et al., 2019; Ingraham et al., 2023). During inference, we use a discrete Langevin sampling method to generate sequences from the MRF model (Zhang et al., 2022).

Network recycling and the pre-trained language model

Drawing inspiration from the network recycling mechanism employed in AlphaFold for structure prediction (Jumper et al., 2021) and CarbonDesign for protein sequence design (Ren et al., 2024), we applied the network recycling mechanism for protein structure and sequence co-design. This approach has two main advantages: First, it enhances model capacity without increasing model size. Second, it allows for the extraction of additional features from intermediate predictions and provides error feedback for subsequent iterations.

In CarbonNovo, we extract additional features from the intermediate predictions of both the structure design and sequence design modules. Specifically, for the structure design module, we extract the distogram of the predicted $\hat{\mathbf{T}}^{(0)}$ and updated pair representations as additional features for the subsequent recycling stage. For the sequence design module, we extract the language model embeddings for the intermediate sequence sampled from the MRF model. These recycling features are utilized to update the input single and pair representations of the structure design and sequence design modules as follows:

$$\begin{aligned}\mathbf{r}^s &= \mathbf{r}^s + \text{Linear}\Big(\text{pLMEmbedding}(\mathbf{s})\Big) + \mathbf{r}^s_{\text{prev}},\\ \mathbf{r}^p &= \mathbf{r}^p + \text{Linear}\Big(\text{DistanceMap}(\mathbf{T}^{(0)}_\theta)\Big) + \mathbf{r}^p_{\text{prev}}.\end{aligned} \tag{6}$$

3.4. Sampling

In structure sampling, we employ the standard Langevin sampling algorithm, a widely utilized method in score-based diffusion models (Song et al., 2020b; Yim et al., 2023b). For sequence sampling, we investigate the Discrete Metropolis-Hastings Langevin algorithm (Zhang et al., 2022).

3.4.1. Structure Sampling

Following FrameDiff (Yim et al., 2023b), we employ the Langevin dynamic to sample the backbone structures.

First, the initial structure $\mathbf{T}^{(T_F)} = (\mathbf{R}^{(T_F)}, \mathbf{t}^{(T_F)})$ is sampled as follows:

$$p^{\text{SE}(3)}_{\text{inv}}(\mathbf{T}^{(T_F)}) = P_{\#}(\mathcal{N}(0, \text{Id}_3)^{\otimes N}) \otimes (\mathcal{IG}_{\text{SO}(3)}(0, \text{Id}))^{\otimes N}. \tag{7}$$

Then, during the Langevin sampling process, we utilize the structure module $\mathcal{S}^{\text{SE}(3)}_\theta(\mathbf{T}, t) = \{\mathcal{S}^{\mathbf{R}}_\theta(\mathbf{R}, t), \mathcal{S}^{\mathbf{t}}_\theta(\mathbf{t}, t)\}$ to compute ∇E_{str}. The structure proposal distribution can be defined as:

$$\begin{aligned}&q_{\text{str}}(\mathbf{T}^{(t-\Delta t)}|\mathbf{T}^{(t)}) = q_{\text{str}}(\mathbf{R}^{(t-\Delta t)}|\mathbf{R}^{(t)}) q_{\text{str}}(\mathbf{t}^{(t-\Delta t)}|\mathbf{t}^{(t)}),\\ &q_{\text{str}}(\mathbf{R}^{(t-\Delta t)}|\mathbf{R}^{(t)}) \sim \mathcal{IG}_{SO(3)}(\Delta t \mathcal{S}^{\mathbf{R}}_\theta(\mathbf{R}^{(t)}), \Delta t \text{Id})^{\otimes N},\\ &q_{\text{str}}(\mathbf{t}^{(t-\Delta t)}|\mathbf{t}^{(t)}) \sim P\mathcal{N}(\mu_\theta, \Delta t \text{Id}_3)^{\otimes N},\\ &\mu_\theta = \frac{1}{2}\Delta t \cdot \mathbf{t}^{(t)} + \Delta t \cdot \mathcal{S}^{\mathbf{t}}_\theta(\mathbf{t}^{(t)}).\end{aligned} \tag{8}$$

Here, $P \in \mathbb{R}^{3N\times 3N}$ is the projection matrix removing the center of mass $\frac{1}{N}\sum_{i=1}^{N}\mathbf{t}_i$, and N is the length of designed proteins.

3.4.2. Sequence Sampling

We employ a discrete Metropolis-Hastings Langevin sampling method for sequence sampling (Zhang et al., 2022). Here, we use a superscript t to denote the step of the diffusion iterations and a subscript k to denote the number of steps in the M-H sampling process.

We obtain the initial sequence $\mathbf{s}_{(0)}^{(t)}$ only from the single representation $\mathbf{r}^s$. The sequence proposal distribution $q_{\text{seq}}(\mathbf{s}_{(k+1)}^{(t)}|\mathbf{s}_{(k)}^{(t)}, \mathbf{T}_\theta^{(0)})$ is as follows:

$$\begin{aligned} q_{\text{seq}}(\mathbf{s}_{(k+1)}^{(t)}|\mathbf{s}_{(k)}^{(t)}, \mathbf{T}_\theta^{(0)}) &\sim \text{Categorical}\Big(\mathbf{M}^{\text{seq}}\Big), \\ \mathbf{M}^{\text{seq}} &= \text{Softmax}\Big(\frac{1}{2}\nabla E_{\text{seq}}(\mathbf{s}_{(k)}^{(t)}, \mathbf{T}_\theta^{(0)})\Delta\mathbf{s} - \frac{\Delta\mathbf{s}^2}{2\gamma}\Big), \\ \Delta\mathbf{s} &= \mathbf{s}_{(k+1)}^{(t)} - \mathbf{s}_{(k)}^{(t)}. \end{aligned} \tag{9}$$

3.5. Training

One of key distinctions of CarbonNovo from previous two-stage approaches like RFDiffusion and FrameDiff is the joint training of structure and sequence design modules. This joint training enables error feedback from the sequence design module to the structure design module, enhancing the overall design process.

3.5.1. Training Losses

Loss for structure design During the training of the structure design module, we adopted FrameDiff's method (Yim et al., 2023b) to obtain the noisy structure:

$$\mathrm{d}\mathbf{T}^{(t)} = \begin{bmatrix} 0 \\ -\frac{1}{2}P\mathbf{t}^{(t)} \end{bmatrix}\mathrm{d}t + \begin{bmatrix} \mathrm{d}\mathbf{B}_{\text{SO}(3)^N}^{(t)} \\ P\mathrm{d}\mathbf{B}_{\mathbb{R}^{3N}}^{(t)} \end{bmatrix}. \tag{10}$$

Here, $\mathbf{B}_{\text{SO}(3)^N}^{(t)}$ and $\mathbf{B}_{\mathbb{R}^{3N}}^{(t)}$ denotethe Brownian motion on $\text{SO}(3)^N$ and $\mathbb{R}^{3N}$ space, respectively.

The primary training objective for structure design module is the denoising score matching (DSM) loss (Song et al., 2020b; Yim et al., 2023b). The DSM loss, $\mathcal{L}_{\text{dsm}}$ (Equation 19), is divided into two components: the rotation loss $\mathcal{L}_{\text{rot}}$ in $\text{SO}(3)$ space and the translation loss $\mathcal{L}_{\text{trans}}$ in $\mathbb{R}^3$ space. We also employ the auxiliary losses in FrameDiff, including the backbone error loss, $\mathcal{L}_{\text{bb}}$ (Equation 24), and the loss for pairwise atomic distance within a local environment of $6\mathring{A}$, denoted as $\mathcal{L}_{\text{2D}}$ (Equation 23).

Additionally, we utilize the FAPE loss, $\mathcal{L}_{\text{FAPE}}$ (Equation 22), to directly supervise the *frames* of backbone structures, a loss proven effective in the protein structure prediction task (Jumper et al., 2021) . We also incorporate a distogram loss, $\mathcal{L}_{\text{dist}}$ (Equation 21), to directly supervise the pair representation $\mathbf{r}_{ij}^p$. Further details of training losses can be found in Appendix D.2.

Loss for sequence design We use single cross-entropy loss $\mathcal{L}_{\text{single}}$ and pair cross-entropy loss $\mathcal{L}_{\text{pair}}$ to supervise the conservation bias term $\psi_s(s_i|\mathbf{r}_i^s)$ and pairwise coupling term $\psi_p(s_i, s_j|\mathbf{r}_{ij}^p)$ in the MRF model (Equation 5), respectively.

Specifically, for $\mathcal{L}_{\text{single}}$, we first compute the logits from the single representation $\mathbf{r}_i^s$ and then compute the cross-entropy loss with the native sequence as labels.

For $\mathcal{L}_{\text{pair}}$, we use a composite likelihood to approximate the full likelihood of the sequence under the MRF model (Zhang et al., 2019; Ren et al., 2024). For each amino acid pair (s_i, s_j) in the sequence, the composite likelihood conditioned on all other amino acids is defined as:

$$\mathcal{P}(s_i, s_j|s_{ij}; \mathbf{r}_i^s, \mathbf{r}_{ij}^p) \tag{11}$$

$$\begin{aligned} &= \log P(S_i = \mathbf{s}_i, S_i = \mathbf{s}_j|S_{\neg\{i,j\}} = \mathbf{s}_{\neg\{i,j\}}; \mathbf{r}_i^s, \mathbf{r}_{ij}^p) \\ &= \log\Big\{\frac{1}{Z_{ij}}\exp\Big[\psi_s(s_i|\mathbf{r}_i^s) + \psi_s(s_j|\mathbf{r}_j^s) + \psi_p(s_i, s_j|\mathbf{r}_{ij}^p) \\ &\quad + \sum_{k\notin\{i,j\}}[\psi_p(s_i, s_k|\mathbf{r}_{ik}^p) + \psi_p(s_j, s_k|\mathbf{r}_{jk}^p)]\Big]\Big\}. \end{aligned} \tag{12}$$

Here, Z_{ij} represents the normalization factor. We compute the distribution of amino acid pairs with the composite likelihood from the pair representations $\mathbf{r}_{jk}^p$ and then compute the cross entropy loss with native amino acid pairs as labels.

3.5.2. Training Strategies

To improve training efficiency, we first pre-train the structure and sequence design modules separately. Subsequently, the two modules are jointly trained in an end-to-end manner within the CarbonDesign framework.

Pre-training stage We train the structure design module using the following loss functions:

$$\begin{cases} \mathcal{L}_{\text{str}} = \mathcal{L}_{\text{dsm}} + \mathcal{L}_{\text{aux}}\mathbb{I}(t < 0.25), \\ \mathcal{L}_{\text{aux}_1} = 0.5\mathcal{L}_{\text{dist}} + 1.0\mathcal{L}_{\text{bb}} + 1.0\mathcal{L}_{\text{2D}} + 2.0\mathcal{L}_{\text{FAPE}}, \\ \mathcal{L}_{\text{dsm}} = 1.0\mathcal{L}_{\text{trans}} + 0.5\mathcal{L}_{\text{rot}}. \end{cases}$$

We employ the auxiliary loss function only for samples where t is less than 0.25 (Yim et al., 2023b).

We train the sequence design module using the following loss functions:

$$\mathcal{L}_{\text{seq}} = 1.0\mathcal{L}_{\text{single}} + 1.0\mathcal{L}_{\text{pair}} + 0.01\mathcal{L}_1 + 0.02\mathcal{L}_2.$$

Here, $\mathcal{L}_1$ and $\mathcal{L}_2$ denote L1 and L2 regularization terms for both single and pair representations, which are used to parameterize the MRF model (Equation 5).

During training for the sequence design module, we add noises to crystal structures to mitigate overfitting. Specifically, we employ the forward process of diffusion on the crystal backbone structures, where the time step t for adding noises follows a uniform distribution $t \sim \mathrm{Uniform}([0, 0.1])$ (Equation 10).

This stage involves 10k training steps for the structure design module and 9k training steps for the sequence design module.

Co-training stage For this stage, we initialize the model weights from the pre-training stage. We note that, during the pre-training of the sequence design module, the input single representations are set $\mathbf{0}$, while in the co-training stage, they are set as the output single representations of the structure design module.

There are two sub-stages for the co-training stage. In the first stage, we add a side chain prediction objective as an auxiliary loss to optimize both the protein sequences and the backbone structures (Jumper et al., 2021; Ren et al., 2024). This loss is only applied for training samples with $t < 0.05$. In the second stage, we incorporate a clash loss to remove steric clashes in local structure conformations. Additionally, we enlarge the crop size in the second stage from 256 to 320.

$$\begin{cases} \mathcal{L}_{\text{stage}_1} = \mathcal{L}_{\text{dsm}} + \mathcal{L}_{\text{aux}}\mathbb{I}(t < 0.25) + 0.01\mathcal{L}_{\text{seq}}\mathbb{I}(t < 0.1) \\ \qquad\qquad + 0.1\mathcal{L}_{\text{side}}\mathbb{I}(t < 0.05), \\ \mathcal{L}_{\text{stage}_2} = \mathcal{L}_{\text{dsm}} + \mathcal{L}_{\text{aux}}\mathbb{I}(t < 0.25) + 0.1\mathcal{L}_{\text{seq}}\mathbb{I}(t < 0.1) \\ \qquad\qquad + 0.1\mathcal{L}_{\text{side}}\mathbb{I}(t < 0.05) + 0.1\mathcal{L}_{\text{viol}}\mathbb{I}(t < 0.05). \end{cases}$$

The first and second sub-stages involves 100k and 10k training steps.

3.6. Datasets

For the co-training stage and pre-training of the structure design module, CarbonNovo utilizes the same structure dataset as FrameDiff (Yim et al., 2023b). For pre-training the sequence design module, we utilize PDB data collected before August 1, 2021, aligning with the dataset used for ProteinMPNN (Dauparas et al., 2022). We also filter out the low-quality samples from the training dataset (see Appendix D.3).

We set a crop size of 256. Instead of random cropping, we employ the Protein Domain Parser (Alexandrov & Shindyalov, 2003) to bias sampling towards regions within protein domains (see Appendix D.4).

4. Experiments

In Section 4.1, we evaluate CarbonNovo in protein structure and sequence co-design using various metrics. In Section 4.2, we conducted ablation experiments to evaluate the contribution of CarbonNovo's key components. Then in Section 4.3, we perform a case study to illustrate protein interpolation in latent space.

4.1. *De novo* Protein Design

4.1.1. Baseline Models

We compare CarbonNovo to the representative *de novo* methods including RFdiffusion (Watson et al., 2023), Chroma (Ingraham et al., 2023), FrameDiff (FrameDiff-ICML) and its improved version (FrameDiff-Improved), FrameFlow (Yim et al., 2023a), and Genie (Lin & Alquraishi, 2023). More running details are in Appendix J.5.

4.1.2. Evaluation Metrics

Following previous work (Lin & Alquraishi, 2023; Yim et al., 2023b;a; Watson et al., 2023), we evaluate designed proteins using various metrics including designability, novelty, and diversity. Furthermore, we introduce two additional evaluation metrics: Rosetta energy and sequence plausibility. Here, for each method, we generated 64 sequences of lengths $\{100, 200, 300, 400, 500\}$ for assessment.

Designability To assess the designability of the generated proteins, we perform the *self-consistency* evaluation pipeline (Yim et al., 2023b; Bose et al., 2023), where the structures of generated sequences are predicted using both ESMFold (Lin et al.) and OmegaFold (Wu et al., 2022b). The evaluation involves three metrics: average scTMscore, average scRMSD, and Fraction, the proportion of scRMSD less than 2Å, and scTMscore greater than 0.5. Further details can be found in Appendix I.1.

Novelty To assess the novelty of designed proteins, we utilize Foldseek (van Kempen et al., 2023) to measure the similarity of our designed proteins to known proteins in the Protein Data Bank (PDB), using a threshold TM-score of less than 0.5. Undesignable proteins often exhibit high randomness in their structure, such as collapsing or having clashes (Appendix 5), resulting in very low similarity with natural proteins in the PDB. Following previous works (Lin & Alquraishi, 2023; Bose et al., 2023), we filter out these undesignable proteins. Further details can be found in Appendix I.3.

Diversity To quantify the diversity of the designed proteins, we calculated the average pairwise TM-score among the generated structures. A lower TM-score indicates higher diversity. Following previous work (Bose et al., 2023), we

Table 1. Evaluation of Designability, Diversity, and Novelty. As for the designability and novelty metrics, the results are presented with structures predicted by ESMFold on the left and OmegaFold on the right of the slash line.

	Designability			Diversity (↓)	Novelty (↑)
	scRMSD (↓)	scTMscore (↑)	Fraction (↑)		
RFdiffusion	3.494 / 3.891	0.897 / 0.753	69.81% / 60.00%	0.221	36.50% / 32.65%
Genie	7.581 / 8.929	0.672 / 0.589	31.43% / 28.75%	0.229	16.50% / 13.35%
FrameDiff-ICML	8.197 / 9.768	0.656 / 0.498	19.96% / 15.63%	0.239	5.31% / 4.65%
FrameDiff-Improve	6.524 / 6.793	0.755 / 0.629	27.81% / 28.75%	0.279	6.31% / 5.60%
FrameFlow	18.827 / 26.02	0.320 / 0.285	11.88% / 11.25%	0.275	5.60% / 4.65%
Chroma v1 (GitHub)	3.209 / 3.620	0.868 / 0.742	45.70% / 40.10%	**0.204**	25.62% / 23.92%
CarbonNovo+MPNN	2.431 / 2.541	0.917 / 0.834	73.16% / 70.15%	0.217	39.75% / 36.94%
CarbonNovo (default)	**1.943 / 1.990**	**0.924 / 0.859**	**81.38% / 77.38%**	0.217	**43.15% / 40.92%**

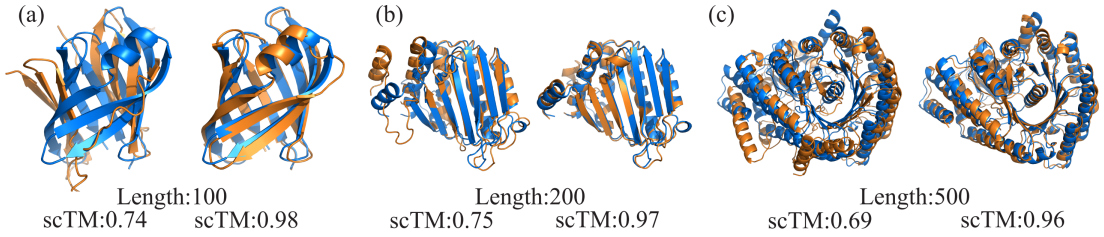

Figure 2. Comparison between sequences and structures jointly produced by CarbonNovo (right) and those designed using ProteinMPNN (left) at various lengths. The structures generated by CarbonNovo are in blue, while the structures predicted by ESMFold are in orange.

only evaluate the diversity for these *designable* proteins.

Rosetta energy Following previous works for sequence design (Anand et al., 2022; Liu et al., 2022b), we employ Rosetta energy (Alford et al., 2017) to measure the stability of designed proteins and the compatibility between their sequences and structures. More details on this metric can be found in Appendix I.4. Further details can be found in Appendix I.2.

Sequence plausibility Experimental validation in wet-lab settings demonstrates that designed sequences with lower likelihood exhibit higher solubility and foldability (Nijkamp et al., 2023). In this context, we evaluate sequence plausibility based on log-likelihood using another independent protein language model, ProGen2 (Nijkamp et al., 2023).

4.1.3. Evaluation Results

As shown in Table 1, CarbonNovo demonstrates superior performance compared to all two-stage methods in both designability and novelty. Regarding the diversity metric, CarbonNovo achieves performance comparable to Chroma and RFDiffusion, outperforming other methods.

To investigate the contribution of the co-design strategy in CarbonNovo compared to the two-stage strategy, we conducted an ablation study utilizing ProteinMPNN to design sequences for the generated backbone structures by CarbonNovo (referred to as CarbonNovo-MPNN). We observe that sequences and structures jointly generated by CarbonNovo are more compatible than those designed using CarbonNovo-MPNN (Table 1), highlighting the superiority of the co-design strategy over the two-stage strategy. We also present several designed proteins across different lengths in Figure 2 and in Appendix J.3.

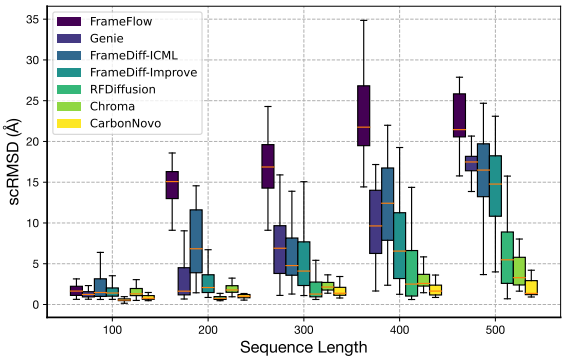

Figure 3. scRMSD of designed proteins vs. predicted proteins under various length.

We next evaluate the performance in designing proteins of varying lengths (Figure 3). First, the performance of all methods in designability drops as the length increases, consistent with previous studies (Watson et al., 2023; Lin &

Alquraishi, 2023; Bose et al., 2023). Second, CarbonNovo exhibits strong robustness to the designed length, significantly outperforming all other methods in designing long proteins.

Additionally, we evaluate CarbonNovo in terms of Rossta energy and sequence plausibility (Table 2). We observe that CarbonNovo outperforms all other methods in these two metrics.

Table 2. Comparison of Rosetta energy function and sequence log likelihood.

Methods	Rosetta energy (↓)	Log-likelihood (↓)
RFdiffusion	-2.64	-2.51
Genie	-	-2.60
FrameDiff-ICML	-2.41	-2.56
FrameDiff-Improve	-2.50	-2.55
FrameFlow	1.75	-2.95
Chroma	-2.79	-2.49
CarbonNovo	**-2.83**	**-2.44**

4.2. Ablation Studies

We trained several ablation models to evaluate the relative contributions of the key components to CarbonNovo's performance. Detailed model settings can be found in Appendix E.

As shown in Table 3, the language model, sequence design module, and auxiliary training loss all contribute to CarbonNovo's performance. Among these, the incorporation of language models demonstrates the most substantial contribution. Moreover, utilizing an energy-based sequence design module enhances the quality of designed sequences compared to the auto-regressive model.

Table 3. Ablation studies evaluating the contribution of the key CarbonNovo components to designability.

designability in Fraction (↑)	PLM	MRF	Pre-train Seq-model	Aux Loss
81.38%	✓	✓	✓	✓
51.75%	✗	✓	✓	✓
74.55%	✓	✗	✓	✓
74.68%	✓	✓	✗	✓
45.16%	✗	✗	✗	✓
35.55%	✗	✗	✗	✗

4.3. Structure Interpolation

We present an illustrative case of protein structure interpolation in latent representation space. We take the initially sampled structure as the latent representation (Equation 7).

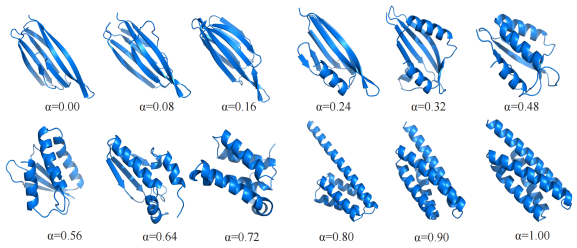

Figure 4. An example of structure morphing: starting from the top left, a protein consisting solely of beta sheet secondary structures gradually transitions to a protein with only alpha-helices in the bottom right.

For a pair of latent points, we perform a linear interpolation and map the interpolation points to the protein structure space through the generative process of CarbonNovo. Here, we denote the interpolation ratio of the starting and end points as $1-\alpha$ and α, respectively.

We select two latent points that evolve into protein structures of All-Beta and All-Alpha topologies of secondary structures, respectively. Protein secondary structure refers to the local three-dimensional arrangement of the backbone structures, primarily characterized by alpha-helices and beta-sheets. Our key observations include: First, the structure latent space is smooth, and almost all intermediate samples look like realistic protein structures (Figure 4). Second, as expected, secondary structures of interpolated structures exhibit a higher proportion of alpha-helices as α increases. Third, interpolated structures demonstrate greater similarity with the endpoint structure as α increases (refer to Figure 7 in the Appendix).

5. Conclusions

We present CarbonNovo, a novel approach for the joint generation of protein structure and sequence within a unified energy-based framework. Our methods also leverage a language model to improve the quality of the designed proteins. Our experiments demonstrate that CarbonNovo achieves state-of-the-art performance in various metrics compared to the two-stage methods.

While CarbonNovo primarily focuses on protein monomer design, it can be readily extended to protein complex design and conditional design like binder design (Ingraham et al., 2023; Watson et al., 2023). Our work is limited in focusing solely on the *in silico* metrics. Wet-lab experimental validation is crucial for a comprehensive evaluation and is left as our future work.

6. Code Avilibility

The CarbonNovo software is available on Github (https://github.com/zhanghaicang/carbonmatrix_public)

Impact Statement

This paper presents work whose goal is to advance the field of AI for Science. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

Acknowledgements

We acknowledge the financial support from the National Natural Science Foundation of China (grant no. 32370657) and the Project of Youth Innovation Promotion Association CAS to H.Z. We also acknowledge the financial support from the Development Program of China (grant no. 2020YFA0907000) and the National Natural Science Foundation of China (grant nos. 32271297 and 62072435). We thank the ICT Computing-X Center, Chinese Academy of Sciences, for providing computational resources.

We thank Xiaoyang Hou and Zaikai He for the useful discussions.

References

Alamdari, S., Thakkar, N., van den Berg, R., Lu, A. X., Fusi, N., Amini, A. P., and Yang, K. K. Protein generation with evolutionary diffusion: sequence is all you need. *bioRxiv*, pp. 2023–09, 2023.

Alexandrov, N. and Shindyalov, I. Pdp: protein domain parser. *Bioinformatics*, 19(3):429–430, 2003.

Alford, R. F., Leaver-Fay, A., Jeliazkov, J. R., O'Meara, M. J., DiMaio, F. P., Park, H., Shapovalov, M. V., Renfrew, P. D., Mulligan, V. K., Kappel, K., et al. The Rosetta all-atom energy function for macromolecular modeling and design. *Journal of chemical theory and computation*, 13(6):3031–3048, 2017.

Anand, N., Eguchi, R., Mathews, I. I., Perez, C. P., Derry, A., Altman, R. B., and Huang, P.-S. Protein sequence design with a learned potential. *Nature communications*, 13(1):746, 2022.

Arnold, F. H. The nature of chemical innovation: new enzymes by evolution. *Quarterly reviews of biophysics*, 48(4):404–410, 2015.

Betz, S. F., Raleigh, D. P., and DeGrado, W. F. De novo protein design: from molten globules to native-like states: Current opinion in structural biology 1993, 3: 601–610. *Current opinion in structural biology*, 3(4):601–610, 1993.

Bose, J., Akhound-Sadegh, T., FATRAS, K., Huguet, G., Rector-Brooks, J., Liu, C.-H., Nica, A. C., Korablyov, M., Bronstein, M. M., and Tong, A. Se (3)-stochastic flow matching for protein backbone generation. In *The Twelfth International Conference on Learning Representations*, 2023.

Cao, L., Goreshnik, I., Coventry, B., Case, J. B., Miller, L., Kozodoy, L., Chen, R. E., Carter, L., Walls, A. C., Park, Y.-J., et al. De novo design of picomolar sars-cov-2 miniprotein inhibitors. *Science*, 370(6515):426–431, 2020.

Dauparas, J., Anishchenko, I., Bennett, N., Bai, H., Ragotte, R. J., Milles, L. F., Wicky, B. I., Courbet, A., de Haas, R. J., Bethel, N., et al. Robust deep learning–based protein sequence design using proteinmpnn. *Science*, 378 (6615):49–56, 2022.

Deng, Y., Bakhtin, A., Ott, M., Szlam, A., and Ranzato, M. Residual energy-based models for text generation. In *International Conference on Learning Representations*, 2019.

Dou, J., Vorobieva, A. A., Sheffler, W., Doyle, L. A., Park, H., Bick, M. J., Mao, B., Foight, G. W., Lee, M. Y.,

Gagnon, L. A., et al. De novo design of a fluorescence-activating β-barrel. *Nature*, 561(7724):485–491, 2018.

Dougherty, M. J. and Arnold, F. H. Directed evolution: new parts and optimized function. *Current opinion in biotechnology*, 20(4):486–491, 2009.

Du, Y. and Mordatch, I. Implicit generation and modeling with energy based models. *Advances in Neural Information Processing Systems*, 32, 2019.

Du, Y., Durkan, C., Strudel, R., Tenenbaum, J. B., Dieleman, S., Fergus, R., Sohl-Dickstein, J., Doucet, A., and Grathwohl, W. S. Reduce, reuse, recycle: Compositional generation with energy-based diffusion models and mcmc. In *International Conference on Machine Learning*, pp. 8489–8510. PMLR, 2023.

Ekeberg, M., Lövkvist, C., Lan, Y., Weigt, M., and Aurell, E. Improved contact prediction in proteins: using pseudolikelihoods to infer potts models. *Physical Review E*, 87(1):012707, 2013.

Gutmann, M. and Hyvärinen, A. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 297–304. JMLR Workshop and Conference Proceedings, 2010.

Harvey, W., Naderiparizi, S., Masrani, V., Weilbach, C., and Wood, F. Flexible diffusion modeling of long videos. *Advances in Neural Information Processing Systems*, 35: 27953–27965, 2022.

Ho, J., Chan, W., Saharia, C., Whang, J., Gao, R., Gritsenko, A., Kingma, D. P., Poole, B., Norouzi, M., Fleet, D. J., et al. Imagen video: High definition video generation with diffusion models.

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

Huang, P.-S., Boyken, S. E., and Baker, D. The coming of age of de novo protein design. *Nature*, 537(7620): 320–327, 2016.

Hyvärinen, A. and Dayan, P. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(4), 2005.

Ingraham, J. B., Baranov, M., Costello, Z., Barber, K. W., Wang, W., Ismail, A., Frappier, V., Lord, D. M., Ng-Thow-Hing, C., Van Vlack, E. R., et al. Illuminating protein space with a programmable generative model. *Nature*, pp. 1–9, 2023.

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

Khoury, G. A., Smadbeck, J., Kieslich, C. A., and Floudas, C. A. Protein folding and de novo protein design for biotechnological applications. *Trends in biotechnology*, 32(2):99–109, 2014.

LeCun, Y., Chopra, S., Hadsell, R., Ranzato, M., and Huang, F. A tutorial on energy-based learning. *Predicting structured data*, 1(0), 2006.

Levada, A. L., Mascarenhas, N. D., and Tannús, A. A novel pseudo-likelihood equation for Potts MRF model parameter estimation in image analysis. In *2008 15th IEEE International Conference on Image Processing*, pp. 1828–1831. IEEE, 2008.

Li, X., Thickstun, J., Gulrajani, I., Liang, P. S., and Hashimoto, T. B. Diffusion-lm improves controllable text generation. *Advances in Neural Information Processing Systems*, 35:4328–4343, 2022.

Lin, Y. and Alquraishi, M. Generating novel, designable, and diverse protein structures by equivariantly diffusing oriented residue clouds. In *International Conference on Machine Learning*, pp. 20978–21002. PMLR, 2023.

Lin, Z., Akin, H., Rao, R., Hie, B., Zhu, Z., Lu, W., dos Santos, A., Costa, M. F.-Z., Sercu, T., Candido, S., et al. Language models of protein sequences at the scale of evolution enable accurate structure prediction.

Lisanza, S., Gershon, J., Tipps, S., Arnoldt, L., Hendel, S., Sims, J., Li, X., and Baker, D. Joint generation of protein sequence and structure with rosettafold sequence space diffusion. 2023.

Liu, N., Li, S., Du, Y., Torralba, A., and Tenenbaum, J. B. Compositional visual generation with composable diffusion models. In *European Conference on Computer Vision*, pp. 423–439. Springer, 2022a.

Liu, Y., Zhang, L., Wang, W., Zhu, M., Wang, C., Li, F., Zhang, J., Li, H., Chen, Q., and Liu, H. Rotamer-free protein sequence design based on deep learning and self-consistency. *Nature Computational Science*, 2(7):451–462, 2022b.

Luo, S., Su, Y., Peng, X., Wang, S., Peng, J., and Ma, J. Antigen-specific antibody design and optimization with diffusion-based generative models for protein structures. *Advances in Neural Information Processing Systems*, 35: 9754–9767, 2022.

Marchand, A., Van Hall-Beauvais, A. K., and Correia, B. E. Computational design of novel protein–protein interactions–an overview on methodological approaches and applications. *Current Opinion in Structural Biology*, 74:102370, 2022.

Martinkus, K., Ludwiczak, J., LIANG, W.-C., Lafrance-Vanasse, J., Hotzel, I., Rajpal, A., Wu, Y., Cho, K., Bonneau, R., Gligorijevic, V., et al. Abdiffuser: full-atom generation of in-vitro functioning antibodies. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Meier, J., Rao, R., Verkuil, R., Liu, J., Sercu, T., and Rives, A. Language models enable zero-shot prediction of the effects of mutations on protein function. *Advances in Neural Information Processing Systems*, 34:29287–29303, 2021.

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

O'Meara, M. J., Leaver-Fay, A., Tyka, M. D., Stein, A., Houlihan, K., DiMaio, F., Bradley, P., Kortemme, T., Baker, D., Snoeyink, J., et al. Combined covalent-electrostatic model of hydrogen bonding improves structure prediction with rosetta. *Journal of chemical theory and computation*, 11(2):609–622, 2015.

Park, H., Bradley, P., Greisen Jr, P., Liu, Y., Mulligan, V. K., Kim, D. E., Baker, D., and DiMaio, F. Simultaneous optimization of biomolecular energy functions on features from small molecules and macromolecules. *Journal of chemical theory and computation*, 12(12):6201–6212, 2016.

Polizzi, N. F. and DeGrado, W. F. A defined structural unit enables de novo design of small-molecule–binding proteins. *Science*, 369(6508):1227–1233, 2020.

Ren, M., Yu, C., Bu, D., and Zhang, H. Accurate and robust protein sequence design with carbondesign. *Nature Machine Intelligence*, 6(5):536–547, 2024.

Salimans, T. and Ho, J. Should ebms model the energy or the score? In *Energy Based Models Workshop-ICLR 2021*, 2021.

Shi, C., Wang, C., Lu, J., Zhong, B., and Tang, J. Protein sequence and structure co-design with equivariant translation. In *The Eleventh International Conference on Learning Representations*, 2022.

Song, Y. and Ermon, S. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

Song, Y. and Kingma, D. P. How to train your energy-based models. *arXiv preprint arXiv:2101.03288*, 2021.

Song, Y., Garg, S., Shi, J., and Ermon, S. Sliced score matching: A scalable approach to density and score estimation. In *Uncertainty in Artificial Intelligence*, pp. 574–584. PMLR, 2020a.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2020b.

Steinegger, M. and Söding, J. Mmseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nature biotechnology*, 35(11):1026–1028, 2017.

Trippe, B. L., Yim, J., Tischer, D., Baker, D., Broderick, T., Barzilay, R., and Jaakkola, T. S. Diffusion probabilistic modeling of protein backbones in 3d for the motif-scaffolding problem. In *The Eleventh International Conference on Learning Representations*, 2022.

van Kempen, M., Kim, S. S., Tumescheit, C., Mirdita, M., Lee, J., Gilchrist, C. L., Söding, J., and Steinegger, M. Fast and accurate protein structure search with foldseek. *Nature Biotechnology*, pp. 1–4, 2023.

Vorobieva, A. A., White, P., Liang, B., Horne, J. E., Bera, A. K., Chow, C. M., Gerben, S., Marx, S., Kang, A., Stiving, A. Q., et al. De novo design of transmembrane β barrels. *Science*, 371(6531):eabc8182, 2021.

Watson, J. L., Juergens, D., Bennett, N. R., Trippe, B. L., Yim, J., Eisenach, H. E., Ahern, W., Borst, A. J., Ragotte, R. J., Milles, L. F., et al. De novo design of protein structure and function with rfdiffusion. *Nature*, 620(7976): 1089–1100, 2023.

Wu, K. E., Yang, K. K., van den Berg, R., Zou, J., Lu, A. X., and Amini, A. P. Protein structure generation via folding diffusion. 2022a.

Wu, R., Ding, F., Wang, R., Shen, R., Zhang, X., Luo, S., Su, C., Wu, Z., Xie, Q., Berger, B., et al. High-resolution de novo structure prediction from primary sequence. *BioRxiv*, pp. 2022–07, 2022b.

Xie, J., Lu, Y., Zhu, S.-C., and Wu, Y. A theory of generative convnet. In *International Conference on Machine Learning*, pp. 2635–2644. PMLR, 2016.

Xie, J., Zhu, S.-C., and Nian Wu, Y. Synthesizing dynamic patterns by spatial-temporal generative convnet. In *Proceedings of the ieee conference on computer vision and pattern recognition*, pp. 7093–7101, 2017.

Xie, J., Zheng, Z., Gao, R., Wang, W., Zhu, S.-C., and Wu, Y. N. Learning descriptor networks for 3d shape synthesis and analysis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8629–8638, 2018.

Xie, J., Xu, Y., Zheng, Z., Zhu, S.-C., and Wu, Y. N. Generative pointnet: Deep energy-based learning on unordered point sets for 3d generation, reconstruction and classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14976–14985, 2021.

Yim, J., Campbell, A., Foong, A. Y., Gastegger, M., Jiménez-Luna, J., Lewis, S., Satorras, V. G., Veeling, B. S., Barzilay, R., Jaakkola, T., et al. Fast protein backbone generation with se (3) flow matching. *arXiv preprint arXiv:2310.05297*, 2023a.

Yim, J., Trippe, B. L., De Bortoli, V., Mathieu, E., Doucet, A., Barzilay, R., and Jaakkola, T. Se (3) diffusion model with application to protein backbone generation. In *International Conference on Machine Learning*, pp. 40001–40039. PMLR, 2023b.

Zhang, H., Zhang, Q., Ju, F., Zhu, J., Gao, Y., Xie, Z., Deng, M., Sun, S., Zheng, W.-M., and Bu, D. Predicting protein inter-residue contacts using composite likelihood maximization and deep learning. *BMC bioinformatics*, 20:1–11, 2019.

Zhang, R., Liu, X., and Liu, Q. A langevin-like sampler for discrete distributions. In *International Conference on Machine Learning*, pp. 26375–26396. PMLR, 2022.

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

A. The Relation between Diffusion Model and EBM

EBMs are a class of probabilistic models that parameterize a distribution as $p_\theta(x) = \exp(-E(x))/Z_\theta$. Here, Z_θ is the normalizing factor. Due to the complexity of calculating the normalization factor, it is difficult to compute likelihood or draw samples from the model. One popular method for EBM training is denoising score matching (Song & Kingma, 2021). The training objective is to minimize the Fisher Divergence D_F between the model and the approximate data distribution p_{data}:

$$D_F = \mathbb{E}_{p_{\text{data}}} \mathbb{E}_{z\sim\mathcal{N}(0,I)} \Big[\frac{1}{2} || \frac{\mathbf{z}}{\sigma} - \nabla_{\mathbf{x}} E(\mathbf{x} + \sigma \mathbf{z}) ||_2^2 \Big]. \tag{13}$$

Here, $\{\mathbf{x}^{(i)}\}_{i=1}^N \sim p_{\text{data}}(\mathbf{x})$ and $\{\mathbf{z}^{(i)}\}_{i=1}^N \sim \mathcal{N}(0, I)$. When minimizing the Fisher Divergence, this ensures that $\exp(-E(\mathbf{x})) \propto p_{\text{data}}(\mathbf{x})$ and therefore $\nabla_{\mathbf{x}} E(\mathbf{x}) = \nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x})$. As for the diffusion model, the training objective ($\mathcal{L}_{\text{dsm}}$) is identical to the denoising score matching (DSM) objective when training EBMs (Du et al., 2023):

$$\sigma_t^2 D_F = \mathbb{E}_{p_{\text{data}}} \mathbb{E}_{z\sim\mathcal{N}(0,I)} \Big[\frac{1}{2} || \mathbf{z} - \sigma_t \nabla_{\mathbf{x}} E(\mathbf{x} + \sigma \mathbf{z}) ||_2^2 \Big] = \mathcal{L}_{\text{dsm}}. \tag{14}$$

For diffusion model, $\mathcal{S}_\theta(\mathbf{x}, t)$ is used to predict the $-\sigma_t \nabla_{\mathbf{x}} E(\mathbf{x} + \sigma \mathbf{z})$. By training score network, the diffused data distribution could be recovered by the $\nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x}) \approx -\frac{\mathcal{S}_\theta(\mathbf{x},t)}{\sigma_t}$. Since diffusion and EBM share the same optimization objectives and sampling strategies, diffusion can be interpreted as an energy model. Therefore, in CarbonNovo, the structure design module and the sequence design module are jointly modeled as an energy model, allowing for joint optimization.

B. Calculate the Score Function

In CarbonNovo, we follow the approach outlined by FrameDiff (Yim et al., 2023b) to calculate the score function as energy on $\text{SE}(3)$ space at each timestep t, using the final structure $\mathbf{T}_\theta^{(0)} = (\mathbf{R}_\theta^{(0)}, \mathbf{t}_\theta^{(0)})$.

$$\nabla_{\mathbf{R}} \log p_{t|0}(\mathbf{R}^t | \mathbf{R}_\theta^0) = \frac{\mathbf{R}^t}{\mathbf{W}^t} \log\{\mathbf{R}_\theta^{(0,t)}\} \frac{\partial_{\mathbf{W}} f(\mathbf{W}^t, t)}{f(\mathbf{W}^t, t)}, \tag{15}$$

$$\nabla_{\mathbf{t}} \log p_{t|0}(\mathbf{t}^{(t)} | \mathbf{t}_\theta^{(0)}) = (1 - e^{-t})^{-1} (e^{-\frac{t}{2}} \mathbf{t}_\theta^{(0)} - \mathbf{t}^{(t)}), \tag{16}$$

where f represents the Brownian motion on $\text{SO}(3)$, $\mathbf{W}(\mathbf{R})$ denotes the rotation angle in radians for any $\mathbf{R} \in \text{SO}(3)$, $\mathbf{R}^{(0,t)}$is defined as $(\mathbf{R}^0)^{\text{T}}\mathbf{R}^t$, and $\mathbf{W}^t = \mathbf{W}(\mathbf{R}^{(0,t)})$. Here, log is the inverse of the exponential map on $\text{SO}(3)$.

C. Model Architectures

In our structure design module, we adopt the revised evoformer (Jumper et al., 2021) network architecture, similar to the main trunk used in ESMFold (Lin et al.). The method of triangular updates in our score network is demonstrated as shown in Algorithm 1. Additionally, we have implemented triangular update modules in the sequence design module as well. The process of these updates is illustrated in Algorithm 2.

Algorithm 1 Triangular update in structure design module.

Input: $\mathbf{r}_i^s, \mathbf{r}_{ij}^p$
$\mathbf{r}_i^s \leftarrow \mathbf{r}_i^s$ + Dropout(SelfAttention($\mathbf{r}_i^s$))
$\mathbf{r}_i^s \leftarrow \mathbf{r}_i^s$ + Dropout(SingleTransition($\mathbf{r}_i^s$))
$\mathbf{r}_{ij}^p \leftarrow \mathbf{r}_{ij}^p$ + OuterProductMean($\mathbf{r}_i^s$)
$\mathbf{r}_{ij}^p \leftarrow \mathbf{r}_{ij}^p$ + Dropout(TriangularMultiplicativeOutgoing($\mathbf{r}_{ij}^p$))
$\mathbf{r}_{ij}^p \leftarrow \mathbf{r}_{ij}^p$ + Dropout(TriangularMultiplicativeIncoming($\mathbf{r}_{ij}^p$))
$\mathbf{r}_{ij}^p \leftarrow \mathbf{r}_{ij}^p$ + Dropout(TriangleAttentionStartingNode($\mathbf{r}_{ij}^p$))
$\mathbf{r}_{ij}^p \leftarrow \mathbf{r}_{ij}^p$ + Dropout(TriangleAttentionEndingNode($\mathbf{r}_{ij}^p$))
$\mathbf{r}_{ij}^p \leftarrow \mathbf{r}_{ij}^p$ + Dropout(PairTransition($\mathbf{r}_{ij}^p$))
Output: $\mathbf{r}_i^s, \mathbf{r}_{ij}^p$

Algorithm 2 Triangular update in sequence design module.

Input: $\mathbf{r}_i^s, \mathbf{r}_{ij}^p$
$\mathbf{r}_i^s \leftarrow \mathbf{r}_i^s$ + Dropout(RowSum($\mathbf{r}_{ij}^p$))
$\mathbf{r}_i^s \leftarrow \mathbf{r}_i^s$ + Dropout(ColumnSum($\mathbf{r}_{ij}^p$))
$\mathbf{r}_{ij}^p \leftarrow \mathbf{r}_{ij}^p$ + OuterProductMean($\mathbf{r}_i^s$)
$\mathbf{r}_{ij}^p \leftarrow \mathbf{r}_{ij}^p$ + Dropout(TriangularMultiplicativeOutgoing($\mathbf{r}_{ij}^p$))
$\mathbf{r}_{ij}^p \leftarrow \mathbf{r}_{ij}^p$ + Dropout(TriangularMultiplicativeIncoming($\mathbf{r}_{ij}^p$))
$\mathbf{r}_{ij}^p \leftarrow \mathbf{r}_{ij}^p$ + Dropout(TriangleAttentionStartingNode($\mathbf{r}_{ij}^p$))
$\mathbf{r}_{ij}^p \leftarrow \mathbf{r}_{ij}^p$ + Dropout(TriangleAttentionEndingNode($\mathbf{r}_{ij}^p$))
$\mathbf{r}_{ij}^p \leftarrow \mathbf{r}_{ij}^p$ + Dropout(PairTransition($\mathbf{r}_{ij}^p$))
Output: $\mathbf{r}_i^s, \mathbf{r}_{ij}^p$

D. Training Details

D.1. Pre-training

To train CarbonNovo, we pre-trained a structure design module and a sequence design module for structure design and sequence design, respectively.

For the structure design module, we opted for a modified version of Evoformer as the network. The input for the network's single representation is the encoding information at time t, encoded in the same way as FrameDiff (Yim et al., 2023b). For the initialization of pair features, we calculated the protein's residue-residue map using a frame structure and divided the continuous values into 32 bins.

We utilized DSM loss function and auxiliary loss function to train the diffusion-based structure design module. The form of the DSM loss is as follows (Song et al., 2020b):

$$\mathcal{L}(\theta) = \mathbb{E}[\lambda||\nabla_{\mathbf{T}} \log p_{t|0}(\mathbf{T}^{(t)}|\mathbf{T}^{(0)}) - \mathcal{S}_\theta^{\mathrm{SE}(3)}(\mathbf{T}^{(t)}, t)||^2]. \tag{17}$$

Additionally, we employed auxiliary loss functions to further optimize the generated protein's backbone and prevent physical violations. We trained our network using four loss functions: the FAPE loss function, distogram loss function (Jumper et al., 2021), MSE on backbone loss function, and a 2D pair (Yim et al., 2023b) loss function for optimization. Compared to FrameDiff, we additionally employed $\mathcal{L}_{\mathrm{dist}}$ to supervise the coordinates of the pseudo$-C_\beta$ atoms. The pseudo$-C_\beta$ coordinates can be calculated using ideal angle and bond length defination: $b = C_\alpha - N$, $c = C - C_\alpha$, $a = \mathrm{cross}(b, c)$, pseudo$-C_\beta$ = $-0.58273431 \times a + 0.56802827 \times b - 0.54067466 \times c + C_\alpha$ (Dauparas et al., 2022). Similarly, we used the $\mathcal{L}_{\mathrm{FAPE}}$ function specific to the backbone to supervise the frame. The total loss is:

$$\mathcal{L}_{\mathrm{str}} = 1.0\mathcal{L}_{\mathrm{dsm}}^{\mathbf{t}} + 0.5\mathcal{L}_{\mathrm{dsm}}^{\mathbf{r}} + (0.5\mathcal{L}_{\mathrm{dist}} + 1.0\mathcal{L}_{\mathrm{bb}} + 1.0\mathcal{L}_{\mathrm{2D}} + 2.0\mathcal{L}_{\mathrm{FAPE}})\mathbb{I}(t < 0.25). \tag{18}$$

For the training of the structure design module, we chose the same training set as FrameDiff (`https://github.com/jasonkyuyim/se3_diffusion/tree/master`). Here, we used MMseq2 (Steinegger & Söding, 2017) to cluster all structures in the training set with a similarity of 40%. During training, we selected a batch size of 48.

D.2. Loss Function

The loss functions we employed primarily comprise three parts: (1) loss function for backbone generation ($\mathcal{L}_{\mathrm{trans}}$, $\mathcal{L}_{\mathrm{rot}}$), (2) loss function for sequence design ($\mathcal{L}_{\mathrm{seq}}$, $\mathcal{L}_{\mathrm{pair}}$), and (3) auxiliary loss functions ($\mathcal{L}_{\mathrm{dist}}$, $\mathcal{L}_{\mathrm{FAPE}}$, $\mathcal{L}_{\mathrm{bb}}$, $\mathcal{L}_{\mathrm{2D}}$, $\mathcal{L}_{\mathrm{side}}$, $\mathcal{L}_{\mathrm{clash}}$).

D.2.1. Backbone Generation Loss

Following the loss used in FrameDiff-Improve, we use the DSM loss when training CarbonNovo. The total SE(3) loss is:

$$\mathcal{L}_{\mathrm{dsm}} = 0.5\mathcal{L}_{\mathrm{rot}}^{\mathbf{r}} + \mathcal{L}_{\mathrm{trans}}^{\mathbf{t}}.$$

Here, the DSM rotation loss in $\mathrm{SO}(3)$ is defined as:

$$\mathcal{L}_{\mathrm{dsm}}^{\mathbf{r}} = \frac{1}{N}\sum_{n=0}^{N} \frac{1}{\mathbb{E}[||\nabla_{\mathbf{R}} \log p_{t|0}(\mathbf{R}_n^{(t)}|\mathbf{R}^{(0)})||_{\mathrm{SO}(3)}^2]} ||\nabla_{\mathbf{R}} \log p_{t|0}(\mathbf{R}_n^{(t)}|\mathbf{R}^{(0)}) - \mathcal{S}_\theta^{\mathbf{r}}(t, \mathbf{R}^{(t)})||^2. \tag{19}$$

The DSM translation loss in $\mathbb{R}^3$ space is:

$$\mathcal{L}_{\text{dsm}}^{\mathbf{t}} = \frac{1}{N}\sum_{n=0}^{N} ||\hat{\mathbf{t}}^{(0)} - \mathbf{t}^{(0)}||^2. \tag{20}$$

D.2.2. AUXILIARY LOSS

Distogram loss Similar to AlphaFold2 (Jumper et al., 2021), we used distogram prediction head and distogram loss to supervise the pair representation. We linearly project the symmertrized pair representations ($\mathbf{r}_{ij}^p + \mathbf{r}_{ji}^p$) into 64 distance bins and obtain the bin probabilities p_{ij}^b with a softmax. The label y_{ij}^b is a one-hot computed from the ground truth position of C_β for all amino acids. The cross-entropy loss averaged over all residue pairs:

$$\mathcal{L}_{\text{dist}} = -\frac{1}{N}\sum_{i,j}\sum_{b=1}^{64} y_{ij}^b \log p_{ij}^b. \tag{21}$$

FAPE loss We employed the FAPE loss function, firstly designed for structure prediction, to supervise the distances between frames in proteins. We use $\mathbf{x}_j$ and $\mathbf{x}_j^{\text{GT}}$ to represent the predicted atomic coordinates and the actual coordinates, while T_i and T_i^{GT} denotes the predicted local frame and the ground truth local frame, respectively. The FAPE loss is defined as:

$$\mathcal{L}_{\text{FAPE}} = \frac{1}{Z}\frac{1}{N^2}\min\Big(\sqrt{||T_i^{-1}\circ\mathbf{x}_j - T_i^{\text{GT}^{-1}}\circ\mathbf{x}_j^{\text{GT}}||^2 + \epsilon}, d_{\text{clamp}}\Big). \tag{22}$$

Here, Z is the length scale, d_{clamp} is the clamp size, and ϵ is a small constant.

2D loss We supervised the pairwise distances of all atoms in the backbone $\Omega \in \{C_\alpha, C, N, O\}$. The predicted distance d_{ij}^{ab} is from the atom a of the i-th amino acid to the atom b of the j-th amino acid, whereas the ground truth distance $\hat{d_{ij}^{ab}}$ is also from the atom a of the i-th amino acid to the atom b of the j-th amino acid. The 2D loss is:

$$\mathcal{L}_{\text{2D}} = \min\Big(\frac{1}{\sum_{i,j=1}^{N}\sum_{a,b\in\Omega}\mathbb{I}(d_{ij}^{ab} < 6\text{Å}) - N}(\sum_{i,j=1}^{N}\sum_{a,b\in\Omega}\mathbb{I}(d_{ij}^{ab} < 6\text{Å}))||d_{ij}^{ab} - \hat{d_{ij}^{ab}}||^2), d_{\text{clamp}}\Big). \tag{23}$$

We clamp values greater than d_{clamp} to stabilize training.

Backbone loss In order to avoid chain breaks or steric clashes, we introduced an auxiliary loss function from FrameDiff (Yim et al., 2023b). We use the MSE function to supervise the distance of the generated structure for the four atoms: $\Omega \in \{C_\alpha, C, N, O\}$. The form of the backbone loss can be represented as follows:

$$\mathcal{L}_{\text{bb}} = \frac{1}{4N}\min\Big(\sum_{i=1}^{N}\sum_{a\in\Omega}||\mathbf{x}_i^a - \hat{\mathbf{x}}_i^a||^2, d_{\text{clamp}}\Big). \tag{24}$$

We clamp values greater than d_{clamp} to stabilize training.

D.3. Datasets

For the training set, we filtered out low-quality data according to the following criteria: (1) Proteins with a length less than 50. (2) Proteins with more than 50% of atoms missing, and (3) Proteins with a resolution better than 5.0Å.

D.4. Protein Domain Parser

We adopted a domain-based crop strategy for long sequences to make the diffusion training more stable and generate more reasonable monomers. Random cropping often cuts through the domain splitting point or disordered regions of the cropped protein, which means the data can be biased compared to the ideal monomer structure. Previous work (Lin & Alquraishi, 2023; Wu et al., 2022a; Yim et al., 2023a) has trained using the CATH dataset or removed protein data that exceeded the crop size (Yim et al., 2023b). For this reason, we sample training data based on the protein domains. Unlike the random

cropping method, we use a Boolean value C_{ij} to record whether the distance between amino acid i and amino acid j is less than $8\mathring{\text{A}}$. We select the center point of the domain based on the following defined:

$$\omega_k = \frac{\sum_{i,j=0,k}^{k,n} C_{ij}}{(i * (L-i))^{\alpha}}. \tag{25}$$

Here, we use ω_k to denote the probability of position k being the center point of the domain. α is an empirical parameter, and we set it to 0.43 during training.

E. Ablation Studies

E.1. Protein Language Model

We conducted ablation experiments on these models to validate the benefits of language models. During the co-training stage, all outputs from the language model were masked. We evaluated this model using the same parameter settings. We discovered that the use of language models significantly enhanced CarbonNovo, with the designability in Fraction improving by nearly 30%.

E.2. MRF Decoder vs Auto-regressive Decoder

To test the superiority of MRF, an energy-based sequence design model, over auto-regressive models like ProteinMPNN (Dauparas et al., 2022). We did an ablation study that eliminated the pair energy and energy loss function. Sequence design was conducted solely through the single representation using the auto-regressive approach implemented in ProteinMPNN. During the training of this model, all other settings remained identical, including the number of training steps. We use the same evaluation pipeline. The designability decreased by 6.83% compared to CarbonNovo (default).

E.3. Pre-training and Co-training

In our ablation study, we initiated training without using a pre-trained sequence design module while keeping all other parameter settings unchanged and trained for the same number of steps as CarbonNovo (default). We observed that using pre-trained sequence and structure modules results in a 6.7% higher Designability in Fraction than not using one. Pre-training can provide a more stable and efficient training process, especially in the early stages, leading to faster convergence and improved performance metrics like designability.

We explain the superiority of using pre-trained weights as initialization from two perspectives:

Efficiency in training sequence modules Utilizing pre-trained sequence design models can significantly enhance the efficiency of training sequence modules. During joint training, only samples with a time step t less than 0.1 are used for training sequence design, and only 10% of samples are utilized for training. This leads to low training efficiency for the sequence design module, which in turn adversely affects the training efficiency of the structure module.

Stability and convergence in early training Given that the architecture of CarbonNovo integrates sequence and structure modeling, not using a pre-trained sequence design module can lead to instability in the sequence module during the initial stages of training.

E.4. Auxiliary Loss

We verified the effectiveness of the auxiliary functions during the training process. In the ablation studies, we set the weights of all auxiliary losses to 0 while keeping all other parameters unchanged and trained for the same number of steps. We found that the use of auxiliary losses resulted in a 9.61% improvement in Designability in Fraction.

F. Training Cost and Computational Complexity

We trained our models using only two Nvidia A100 GPUs. The pretraining stage comprised 200k steps and took about 7 days. Subsequently, the co-training stage consisted of approximately 100k additional steps and also took about 7 days.

We would like to highlight that CarbonNovo outperforms the majority of comparative methods in terms of inference speed,

Methods	time	inference steps
RFdiffusion+ProteinMPNN	125s + 41s	50
FrameDiff+ProteinMPNN	37s + 41s	500
Chroma + ChromaDesign	16s + 50s	500
Genie+ProteinMPNN	238s + 41s	1000
FrameFlow + ProteinMPNN	26s + 41s	500
CarbonNovo	108s	100

Table 4. Computational Complexity

as demonstrated in Table 4.

G. Hyperparameters

In parameterizing the diffusion process, we followed the configurations established in FrameDiff-Improve (Yim et al., 2023b). For the structure design network, the parameter settings were aligned with the main trunk of ESMFold (Lin et al.). Additionally, modifications were made to better tailor these parameters to our network, as detailed in Table 5. We show critical hyperparameters for training phases, including crop size, learning rate, steps, optimizer, and batch size in Table 6.

Table 5. Hyperparameters about CarbonNovo.

Category	Description	Value
Structure design module	Layer of blocks	1
	Single representation dimension	256
	Pair representation dimension	128
	Dropout	0.1
	Number of bins	15
Sequence design module	Layer of blocks	4
	Single representation dimension	384
	Pair representation dimension	128
	Dropout	0.1
	Number of bins	20
	Mask threshold	12 Å
Sampling	Timestep Δt	0.01
	Initial inference time t_F	1
Language model	Model	ESM-3B

Table 6. Hyperparameters about CarbonNovo.

Stage	Crop size	Learning rate	Optimizer	Training steps	Batch size	Crop strategy
Structure module pre-train	256	1e-4	Adam	100k	48	PDP
Sequence module pre-train	400	1e-3	Adam	90k	64	Random
Co-training stage1	256	1e-4	Adam	100k	48	PDP
Co-training stage2	320	1e-4	Adam	10k	48	PDP

Table 7. Model size and model details

	Model Size	Networks
CarbonNovo (Structure Module)	2.1M	1 Block of revised evoformer + IPA
CarbonNovo (Sequence Module)	4.8M	4 Blocks of revised evoformer + 2 layer MLP
pLM	3B	Transformer (ESM-3B)
RFdiffusion	59.8M	1,2,3D track
FrameDiff	17.4	MLP +IPA
Chroma	18.5M	16 Blocks GNN
Genie	4.1M	Triangular Blocks

H. Model Size and Model Details

I. Evaluation Metrics

I.1. designability

We employed three metrics to evaluate the designability of designed proteins: scTMscore, scRMSD, and Fraction.

scTMscore The scTMscore is calculated as the TMscore between the designed structure and the structure predicted by structure prediction methods (such as ESMFold and OmegaFold). TMscore is a tool used to assess the similarity between two protein structures independently of their sequences. The specific formula for calculating TMscore is as follows (Zhang & Skolnick, 2004):

$$\text{TMscore}(\mathbf{x}_{\text{design}}, \mathbf{x}_{\text{pred}}) = \max \left(\frac{1}{N} \sum_{i=1}^{L_{\text{align}}} \frac{1}{1 + (\frac{d_i}{d_0(N)})^2} \right).$$

Here, N is the length of the designed protein. L_{align} is the length of the aligned protein sequence, d_i is the distance between the i-th pair of aligned residues, and $d_0(N)$ is a scale factor dependent of N, which is a normalization factor typically based on the length of the proteins being compared. TMscore is a value ranging between 0 and 1, where a TMscore less than 0.5 generally indicates that the two structures do not share the same topological architecture (Zhang & Skolnick, 2004).

scRMSD The scRMSD is calculated as the root mean square deviation between the designed structure and the structure predicted by structure prediction methods (such as ESMFold, OmegaFold). scRMSD could be calculated as follows:

$$\text{RMSD}(\mathbf{x}_{\text{design}}, \mathbf{x}_{\text{pred}}) = \sqrt{\sum_{i=1}^{N} \frac{d_i^2}{N}}.$$

Here, d_i is the aligned distance between the i-th residues. N is the length of the designed protein. RMSD quantifies the average distance between the atoms (the backbone atoms $\{C_\alpha, C, O, N\}$ of the amino acids) of two proteins. A lower RMSD value indicates a higher similarity between the two structures.

Fraction We followed the thresholds established in previous work (Lin & Alquraishi, 2023; Yim et al., 2023b;a; Bose et al., 2023) and calculated the proportion of designed proteins with the scRMSD less than 2Å and the scTMscore greater than 0.5, higher proportion indicates the stronger foldability of the designed proteins.

I.2. Diversity

This metric quantifies the diversity in the structural landscape of the designed proteins. A higher Diversity metric indicates a greater range of structural variations among the proteins that can be designed, showcasing the method's capacity to produce a variety of protein structures. This is vital for numerous applications in protein engineering and functional studies.

I.3. Novelty

Exploring unseen protein conformations in natural proteins can contribute new insights to structural biology (Betz et al., 1993; Huang et al., 2016). In other real-world applications, such as drug design, the objective is to create drugs that match or surpass the functions of natural proteins, necessitating the design of novel and *designable* proteins (Marchand et al., 2022).

Methods	designability
RFdiffusion	64.50%
Chroma	40.50%
CarbonNovo	**74.50%**

Table 8. Evaluating CarbonNovo with AlphaFold

We employed the novelty metric to assess the novelty of the designed proteins, defined as their similarity to existing natural proteins. We used Foldseek (van Kempen et al., 2023) to search for similar protein conformations within the PDB100 database for the proteins we designed. Following the threshold set by previous work (Lin & Alquraishi, 2023; Yim et al., 2023b;a; Bose et al., 2023), we counted all *designable* proteins (scRMSD less than 2Å and scTMscore greater than 0.5) for which Foldseek returned a maximum TMscore of less than 0.5.

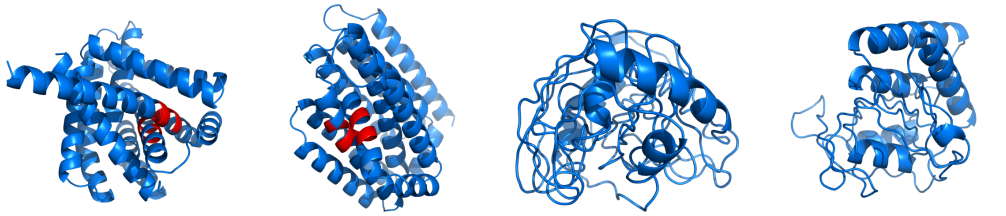

Figure 5. Undesignable but highly novel proteins. The scRMSD is the Root Mean Square Deviation between structures predicted by ESMFold and the designed structures. The pdbTM refers to the TMscore with the most similar natural protein found by Foldseek, compared to the target protein.

I.4. Rosetta Energy

We use the Rosetta energy function (available at `https://new.rosettacommons.org/`) to evaluate the compatibility of sequences and structures. For different *de novo* protein methods, we use the designed structures and sequences as inputs for Rosetta software. For each protein, we calculate the total energy value returned by Rosetta after 200 steps of relaxation. Here, we have analyzed 64 proteins each of lengths $\{100, 200, 300, 400, 500\}$ for comparison. It is noteworthy that Genie, which only outputs the position of the C_α atoms and lacks information on other atoms' positions, cannot be evaluated for Rosetta energy. FrameFlow shows positive energy values due to its poor performance on longer proteins resulting in numerous clashes and overlaps.

J. Additional Results

J.1. Evaluating CarbonNovo with AlphaFold

To address concerns about potential bias towards the PLM, we also used AlphaFold2 predictions for evaluations.

J.2. Performance with Different Inference Steps

We listed the default inference steps and methods of the approaches we compared and analyzed the impact of different step lengths in CarbonNovo on designability.

J.3. Compared to CarbonNovo+ProteinMPNN

We utilized ProteinMPNN to design structures that were generated by CarbonNovo, a process we refer to as CarbonNovo-MPNN, and carried out evaluations accordingly. We utilized the sequences and structures generated by CarbonNovo and calculated their scRMSD. Subsequently, for the designed structures, we designed sequences using ProteinMPNN and

Table 9. Performance under different inference steps.

Methods	Diffusion types	Inference steps	designability
RFdiffusion	DDPM	50	69.81%
Genie	DDPM	1000	31.43%
FrameDiff-ICML	SDE	500	19.96%
FrameDiff-Improve	SDE	500	27.81%
Chroma	SDE	100	45.70%
FrameFlow	ODE	500	11.88%
CarbonNovo	SDE	50	76.56%
CarbonNovo (default)	SDE	100	81.38%
CarbonNovo	SDE	200	83.42%

calculated their scRMSD. Figure 6 illustrates that the end-to-end simultaneous generation of sequences and structures by CarbonNovo outperforms the results obtained from the two-stage process.

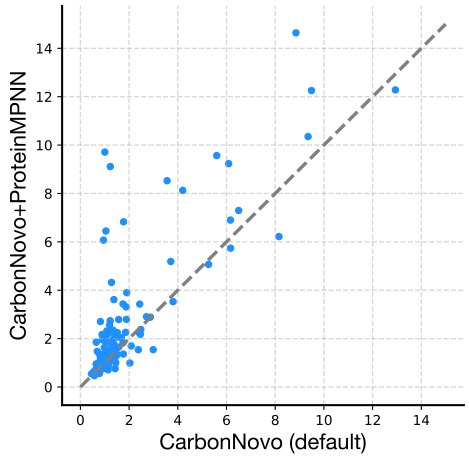

Figure 6. Head-to-head comparisons with CarbonNovo+ProteinMPNN on scRMSD metrics.

J.4. Case Study

We show the TMscore between interpolation structures with different interpolation ratio (α) and the start and endpoints structures.

J.5. Baseline Methods to Compare

RFdiffusion (Alamdari et al., 2023) We utilize the test script provided in the RFdiffusion GitHub repository (`https://github.com/RosettaCommons/RFdiffusion/tree/main`), with the model *Base_ckpt.pt* and all other default settings.

Chroma (Ingraham et al., 2023) Chroma offers multiple models (Chroma v0 and v1). For a fairer comparison, we use the best version (Chroma v1 Improved) offered on GitHub (`https://github.com/generatebio/chroma/tree/main`). Considering that Chroma can perform sequence design through its module, Chroma Design, we used it to generate the designed protein sequence eight times and then used these sequences for evaluation. All parameter settings employ the default options provided by GitHub.

FrameDiff (Yim et al., 2023b) We evaluated two versions provided by FrameDiff: FrameDiff-ICML and FrameDiff-Improve. These two versions correspond respectively to FrameDiff's performance as reported in the paper and the model

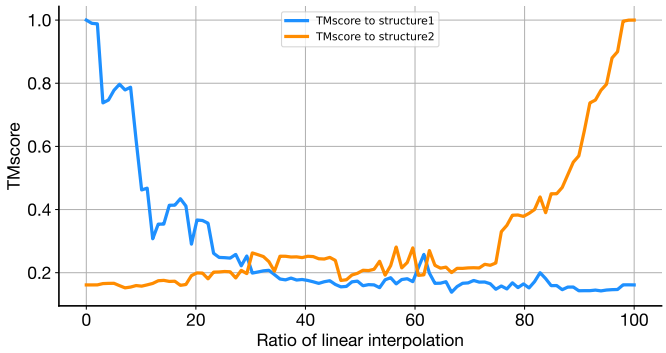

Figure 7. The TMscore between interpolation structures with different interpolation ratio (α) and start and end points structures.

trained with improved strategies available on GitHub. For these two versions, we utilized the test script provided in GitHub (`https://github.com/jasonkyuyim/se3_diffusion`) with *paper_weights.pth* and *best_weights.pth* models. All parameters were selected from the default options provided in the config file.

FrameFlow (Yim et al., 2023a) We used the test script provided in the GitHub repository (`https://github.com/microsoft/protein-frame-flow`). All the hyper-parameters are default.

Genie (Lin & Alquraishi, 2023) Genie offers multiple models based on different training dataset. Using a larger training dataset is one of its core contributions, for a fairer comparison, we used the best performance model (*swissprot_l_256*) reported in the Genie manuscript. All parameters were selected according to the default options provided in the config file in GitHub (`https://github.com/aqlaboratory/genie/tree/main`).

J.6. Tools used in calculating evaluation metrics

ProteinMPNN (Dauparas et al., 2022) We use ProteinMPNN for sequence design. Following previous works (Yim et al., 2023a;b; Lin & Alquraishi, 2023; Watson et al., 2023), we use the evaluated script in the ProteinMPNN GitHub repository with default model and parameters.

ESMFold (Lin et al.) We use ESMFold to predict the structures of designed sequences. We use the evaluated script in the esm GitHub repository (`https://github.com/facebookresearch/esm/tree/main`). We use ESMFold-v1 provided in GitHub. All parameters were selected from the default options.

OmegaFold (Wu et al., 2022b) We use OmegaFold to predict the structures of designed sequences. We utilize the test script provided in the OmegaFold GitHub repository (`https://github.com/HeliXonProtein/OmegaFold/tree/main`). We employed OmegaFold's Model 1 for the evaluation process. All parameter settings employ the default options provided in GitHub.

Foldseek (van Kempen et al., 2023) We used Foldseek to evaluate the novelty of the designed proteins. The input for Foldseek is the protein structure, and its output is the similarity rank between the input target protein and existing natural proteins in the PDB database. Here, we used the TMscore of the protein with the highest similarity returned by Foldseek for our evaluation. Considering that the amino acids in the PDB file of the designed proteins are filled with either glycine or alanine, we employed the TMalign mode provided by Foldseek (alignment-type 1). Unlike previous work (Lin & Alquraishi, 2023; Bose et al., 2023), our objective with *de novo* protein design is to create proteins that are different from natural proteins. Therefore, we searched across all proteins in the PDB database rather than limiting it to a train set.

Rosetta(Alford et al., 2017) We employed the Rosetta software for energy calculations. Rosetta evaluates the total energy of a structure by computing interactions such as van der Waals forces and hydrogen bonds. Its input is the designed protein structure and sequence, with the output being the calculated energy. We used Rosetta's official tools (`https://www.rosettacommons.org/software`) for these calculations. We selected the *ref2015* energy function during the calculation process and performed a relaxation step.

ProGen2 (Nijkamp et al., 2023) We use the test scripts provided in the ProGen2 GitHub repository (`https://github.`

com/salesforce/progen). All the hyper-parameters are default.