Consistent Synthetic Sequences Unlock Structural Diversity in Fully Atomistic De Novo Protein Design

Anonymous Author(s)
Affiliation
Address
email

Abstract

High-quality training datasets are crucial for the development of effective protein design models, but existing synthetic datasets often include unfavorable sequence-structure pairs, impairing generative model performance. We leverage ProteinMPNN, whose sequences are experimentally favorable as well as amenable to folding, together with structure prediction models to align high-quality synthetic structures with recoverable synthetic sequences. In that way, we create a new dataset designed specifically for training expressive, fully atomistic protein generators. By retraining La-Proteína, which models discrete residue type and side chain structure in a continuous latent space, on this dataset, we achieve new state-of-the-art results, with improvements of +54% in structural diversity and +27% in co-designability. To validate the broad utility of our approach, we further introduce *Proteína-Atomística*, a unified flow-based framework that jointly learns the distribution of protein backbone structure, discrete sequences, and atomistic side chains without latent variables. We again find that training on our new sequence-structure data dramatically boosts benchmark performance, improving Proteína-Atomística's structural diversity by +73% and co-designability by +5%. Our work highlights the critical importance of aligned sequence-structure data for training high-performance de novo protein design models. All data will be publicly released.

1 Introduction

De novo protein design aims to generate functional proteins from scratch, making it a central challenge in molecular biology [39, 20, 26, 25]. Recent generative models have made impressive progress to design protein backbones using diffusion and flow-based approaches [22, 47, 50, 6, 27]. Several methods have begun to move beyond backbone-only modeling to enable all-atom generation [15, 9, 37]. Since the sequence serves as the actual design specification for synthesis, and side chains are pivotal in biochemical interactions, generating complete atomistic structures is crucial for structure-guided protein design. As models must reason about the generated sequence and structure to ensure cross consistency, fully atomistic training data plays a crucial role in fully atomistic de novo design.

We identify a critical limitation in commonly used training datasets derived from the AlphaFold Database (AFDB) [43]. Specifically, the (real sequence, synthetic structure) pairs in the AFDB are largely not co-designable by ESMFold [29] (see Fig. 1), AlphaFold2 [23], or Boltz-1 [48],[1] meaning the sequences do not likely fold into their given structures to the best of available in silico approximations. This is surprising, given that the AFDB was created through computational structure prediction. Hence, this data is not well-suited for training joint sequence-structure models, as the data pairs are not consistently reproducible via common folding models. This motivated us to construct

[1] We used single-sequence mode as well as multiple sequence alignments (MSAs) with different databases, but we were not able to reliably reproduce the AFDB structures.

Submitted to AI for Science workshop (NeurIPS 2025). Do not distribute.

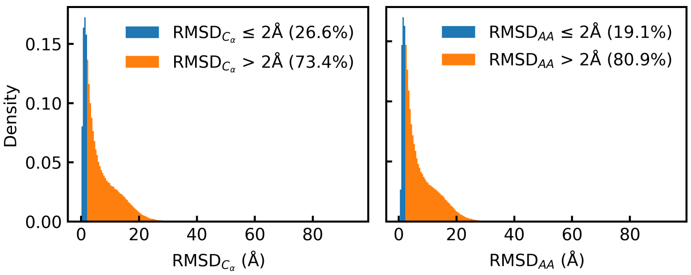

Figure 1: **Co-designability of $\mathcal{D}_{\mathrm{AFDB-clstr}}$.** Histograms of C_α and all-atom RMSD between AFDB and ESMFold structures show only 26.6% of protein backbones and 19.1% of the all-atom structures are considered designable. As a result, most AFDB synthetic structures are not recoverable.

a high-quality dataset from scratch: We leverage ProteinMPNN [10], which is known for strong in silico success and wetlab validation [47, 34], and generate several sequences for each Foldseek AFDB cluster representative structure [28, 5]. We then re-folded all new synthetic sequences to obtain corresponding fully atomistic structures for the new synthetic sequences. By generating fully atomistic sequence-structure pairs in this manner, we construct a more aligned dataset ideally suited for the training of expressive atomistic protein generators. We will publicly release the dataset.

Next, we used our new dataset to train fully atomistic protein generative models that need to capture the intricate relationship between atomistic structures and amino acid identity. Side-chain coordinates cannot be determined without knowledge of the sequence, either explicitly or implicitly, and co-generating diverse and consistent sequences and atomistic structures is challenging. Therefore, many methods rely on a multistage process: generating the backbone, predicting the sequence, and optionally packing side-chains using rotamers [19, 4, 18, 8]. Recent efforts have made progress toward full-atom co-design by incorporating all-atom representations during structure generation [9, 37, 31, 8]. However, these methods still do not explicitly model the joint distribution of sequences and atomistic structures in a unified framework. Recently, La-Proteína [15] jointly learned sequences and side-chain structures via a continuous latent space, achieving strong performance in de novo design and atomistic motif scaffolding. Training La-Proteína on our new data significantly improves the model samples' structural diversity (+54%) and co-designability (+27%), highlighting the importance of well-aligned training data to accurately model the complex sequence-structure relationship.

To validate the generality of our approach, we further propose a multi-modal framework that operates in the explicit observable space, providing a complementary approach to La-Proteína's latent space method. Specifically, we introduce *Proteína-Atomística*, a unified flow-based framework that jointly learns the distribution over fully atomistic protein structure and sequence. We treat this as a joint multi-modal generation task with three co-dependent modalities (Fig. 2): *(i)* C_α atom positions capture large-scale backbone structure. *(ii)* categorical amino acid identities define the protein sequence. *(iii)* non-C_α backbone and side-chain atoms represent local details. We again observe that training on our new aligned sequence-structure data dramatically boosts the model's performance—structural diversity by 73% and co-designability by 5%. This confirms the broad utility of our newly created, aligned data for training different types of fully atomistic protein generative models.

Our experiments emphasize that consistent synthetic sequences play a significant role in enhancing structural diversity. We also show in ablation studies that simply replacing AFDB structures with those from ESMFold to create a "100% designable" dataset degrades both the ESMFold-based designability and structural diversity of generated proteins. This observation served as a key motivation to leverage ProteinMPNN for predicting new sequences, thereby creating a fundamentally new training dataset that consists of both synthetic structures and synthetic sequences, in contrast to the AFDB. Since our models are directly trained on ProteinMPNN sequences and are the first to surpass ProteinMPNN in co-designability, they remove the need for ProteinMPNN-based re-design at the end of generation—a common step in existing pipelines that requires subsequent side-chain redesign to accommodate changes in sequence space.

Contributions: *(i)* We find that AFDB structures are not recoverable with common structure prediction models and argue that the low consistency of AFDB-derived datasets is a critical limiting factor for atomistic structure and sequence co-generation. *(ii)* To overcome this limitation, we introduce a new high-quality dataset consisting of aligned synthetic sequences and structures, ideally suited for the training of high-performance fully atomistic protein generators. *(iii)* We introduce Proteína-Atomística, a novel unified multi-modal flow-based generative framework that jointly and explicitly models the distribution over fully atomistic protein structures and sequences. *(iv)* We show that when trained on our new data Proteína-Atomística outperforms all prior non-unified methods and La-Proteína achieves new state-of-the-art performance in fully atomistic protein generation.

2 Related Work

Protein design has witnessed significant progress through generative models focusing on either sequence or structure. Sequence generation often relies on autoregressive models [32, 14] or discrete diffusion [3, 45], trained on large datasets. For protein backbones, diffusion models have shown remarkable success, with seminal works like Chroma [22] and RFDiffusion [47]. Subsequent works employ diffusion or flow matching on frame-based representations [50, 6, 49, 44, 21], while other works apply diffusion to C_α coordinates [27, 41]. Scaling data and model size in Genie2 [28] and Proteína [16] has led to near-perfect backbone designability metrics. These methods showcase diverse parameterizations and architectures within the broader diffusion/flow matching framework.

However, these single-modality generation methods typically decouple sequence and structure. They either generate a sequence first and then fold it with ESMFold [29] or AlphaFold2 [23], or generate a structure and then infer a sequence with ProteinMPNN [10]. In contrast, recent efforts have focused on co-design methods that aim to jointly model sequence and backbone structure distributions within a single generative framework, such as diffusion/flow-based ProteinGenerator [30], MultiFlow [7] and DPLM-2 [46], energy-based CarbonNovo [38], and language model-based ESM3 [17]. MultiFlow [7] also distills synthetic training sequences and structures to boost co-generation performance, similarly to us leveraging ProteinMPNN, but at a smaller scale and without analyses of the AFDB.

Despite progress in protein co-design, achieving accurate atomistic detail remains challenging. Early all-atom diffusion attempts like Protpardelle [9] yield poor results. Pallatom's [37] use of Atom14 representations could lead to atom-type ambiguities, hindering performance or downstream tasks [37]. Other methods explore latent spaces [31], modular design [8], or specific tasks [2].

2.1 La-Proteína

More recently, La-Proteína [15] introduced a partially latent protein representation that combines explicit and implicit modeling. In this approach, the coarse C_α-backbone structure is modeled explicitly as in Proteína, while sequence and atomistic (non-C_α) details are captured through per-residue latent variables of fixed dimensionality. This hybrid representation sidesteps the challenges associated with explicit side-chain representations, through the training of an initial autoencoder. By applying flow matching in this partially latent space, La-Proteína effectively models the joint distribution over sequences and full-atom structures. See paper for details [15]. We use both Proteína-Atomística and La-Proteína to explore the impact of synthetic data on all-atom protein generation.

3 Aligning Synthetic Protein Sequence and Structure

Our investigation into constructing a new training dataset for explicit all-atom protein generation was motivated by the limitations of the Foldseek-clustered AFDB dataset [42, 5], which was used for instance by Genie2 [28] ($\mathcal{D}_{\mathrm{AFDB-clstr}}$ $\sim$0.6M). We assessed the in-silico co-designability of $\mathcal{D}_{\mathrm{AFDB-clstr}}$ by folding its sequences (length$\in$[32,512]) with ESMFold and computing the C_α and all-atom RMSD between the folded and original AFDB structures. Surprisingly, only 19.1% of the dataset met the standard 2Å co-designability threshold based on all-atom RMSD (Fig. 1). Further analysis using other public structure prediction models on a random subset of $\mathcal{D}_{\mathrm{AFDB-clstr}}$ revealed that even the best co-designability achieved, $\sim$65% with ColabFold using MSAs, fell short of the expected 100% designability under AlphaFold2 (AF2) [23, 33]. This significant sequence-structure misalignment poses a substantial challenge to scaling fully atomistic protein generative models with existing sources of large-scale high-quality synthetic sequence-structure data. Furthermore, Boltz-1 obtains scores roughly the same as ESMFold when using MSAs. Without MSAs it exhibited the lowest consistent recovery. Although we do not expect ESMFold and Boltz-1 to be highly consistent with the AFDB, it is crucial to understand the limitations of relying on the AFDB for training protein design models due to the severe disagreement with other popular structure prediction models.

To address this, we create a novel dataset ($\mathcal{D}_{\mathrm{SYN-ours}}$) that targets the joint alignment between synthetic sequence and synthetic structure, as follows: (i) For each cluster representative in $\mathcal{D}_{\mathrm{AFDB-clstr}}$ with an average pLDDT$_{\mathrm{AF2}} \geq 0.8$, (ii) we produce four sequences with ProteinMPNN, (iii) refold each recording the C_α-RMSD between the AFDB- and ESMFold-generated structures (using C_α, as different sequences have different side chains), (iv) select the sequence with the lowest RMSD, and (v) filter the structures to include those with pLDDT$_{\mathrm{ESMFold}} \geq 0.8$. This results in

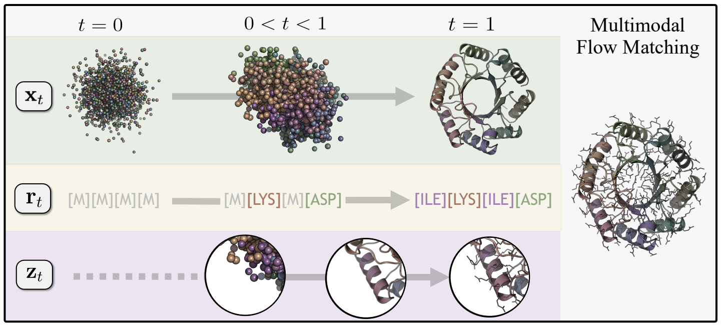

Figure 2: **Proteína-Atomística.** We use a multimodal flow matching framework to learn a mapping from noise distributions of C_α atoms ($\mathbf{x}_t$), amino acid sequences ($\mathbf{r}_t$), and non-C_α atoms ($\mathbf{z}_t$) to realistic atomistic structures. We prevent leakage by initiating the generation of non-C_α atoms $\mathbf{z}_t$ only after their corresponding residues in the sequence $\mathbf{r}_t$ are unmasked.

$\sim$0.43M high-quality samples. Consequently, $\mathcal{D}_{\text{SYN}-\text{ours}}$ identifies confident regions of overlap between folding and inverse folding models, to enabling the modeling of a better recoverable joint sequence-to-structure relationship. In contrast to MultiFlow [7], which replaces PDB sequences with ProteinMPNN ones, we start from a large structurally diverse dataset and refold to recover side chains.

4 Proteína-Atomística

On the one hand, we use our new data to retrain La-Proteína [15], see Sec. 5. To make general conclusions and to also see the data's effect when training a model without a special latent framework, we additionally develop a novel, "data-space" fully-atomistic protein generator without latent variables, called *Proteína-Atomística*, which we now introduce.

4.1 Explicit Multi-Modal Flow Matching Framework

Atomistic protein modeling can be decomposed into explicit modeling of the protein backbone, amino acid sequence, and side-chain atoms. A significant challenge within this breakdown lies in the modeling of side chains, primarily due to the fact that an amino acid residue and its side-chain structure encode the same underlying information in discrete and continuous forms, respectively. Specifically, during a generation process that involves discrete residue tokens, the set of side-chain atoms associated with a residue dynamically changes whenever the residue type is altered or unmasked. Therefore, a robust atomistic modeling framework must effectively handle this variable number of atoms and also provide a good initialization strategy for these newly generated side-chain atoms (as detailed in Sec. 4.2). This inherent complexity makes extending existing backbone or backbone-sequence design methods to joint fully atomistic modeling non-trivial.

To tackle the challenge posed by the variable number of atoms, we adopt the Atom37 representation for protein structures [23]. In this representation, each potential heavy atom of a residue is assigned a unique position within a 37-dimensional array. This choice offers an advantage over the Atom14 representation used by Pallatom [37], as Atom37 avoids interpretation ambiguities where a single position can correspond to multiple atom types. For any non-existent atoms of a given residue, their corresponding positions in the Atom37 array are set to zero and they are subsequently masked out in the model's sequence track (see Sec. 4.2).

Proteína-Atomística achieves fully atomistic protein generation through multi-modal flow matching over C_α coordinates $\mathbf{x} \in \mathbb{R}^{L\times 3}$, amino acid sequence $\mathbf{r} \in \{0, .., 19\}^L$, and non-$C_\alpha$ atom coordinates $\mathbf{z} \in \mathbb{R}^{L\times 36\times 3}$, as illustrated in Fig. 2. In addition, while both C_α and non-C_α atoms are in Euclidean space, their roles differ: C_α define the global structure and non-C_α specify local residue details. This functional difference, coupled with the variable number of non-C_α atoms, presents a significant challenge in extending backbone and backbone-sequence models to full atomistic generation, a challenge that our multi-modal approach effectively addresses. We now present the details of the *Proteína-Atomística* modeling framework:

1. Flow Matching for C_α Atoms. Following Proteína [16], we define a flow ψ_t that pushes an easy-to-sample noise distribution p_0 to a data distribution p_1 through intermediate densities $p_t = [\psi]_t * p_0$, where "$*$" denotes push-forward and $t \in [0, 1]$ is a time variable. This flow is parameterized by an ODE $d\mathbf{x}_t = \mathbf{v}^\theta(\mathbf{x}_t, t)dt$, defined through a learnable vector field $\mathbf{v}^\theta(\mathbf{x}_t, t)$ with parameters θ, with $\mathbf{x}_0 \sim p_0$ and $\mathbf{x}_1 \sim p_1$. By the continuity equation, the true vector field $\mathbf{u}_t$ satisfies $\partial p_t/\partial t = -\nabla_{\mathbf{x}_t} \cdot \big(p_t\, \mathbf{u}_t\big)$, but $\mathbf{u}_t$ is intractable. To address this, conditional flow matching (CFM) constructs for each data sample $\mathbf{x}_1$ a tractable conditional path $p_t(\mathbf{x}_t|\mathbf{x}_1)$. We draw $\mathbf{x}_0 \sim p_0$ and

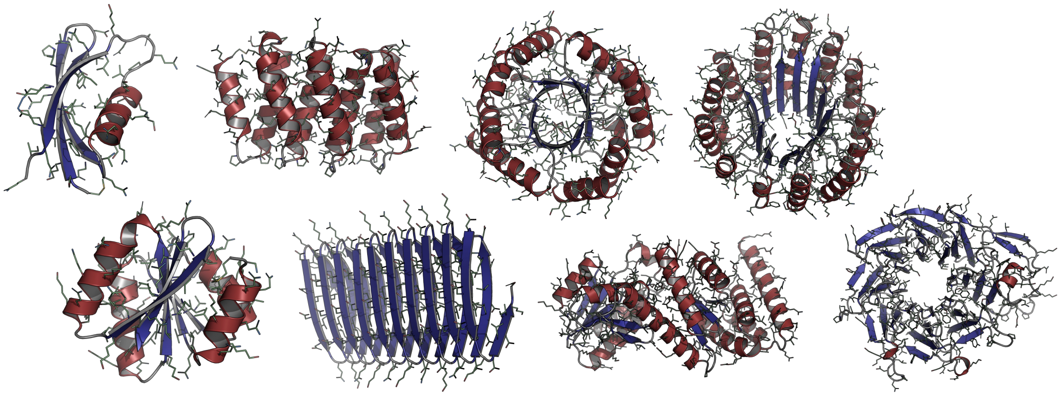

Figure 3: **Proteína-Atomística samples**, ranging from 100 to 400 residues. All shown samples co-designable.

interpolate linearly $\mathbf{x}_t = t\mathbf{x}_1 + (1-t)\mathbf{x}_0$, so that the exact velocity $\mathbf{x}_1 - \mathbf{x}_0$ is known. The CFM objective then regresses the learnable field $\mathbf{v}^\theta(\mathbf{x}_t, t)$ onto this target across random t, $\mathbf{x}_0$, and $\mathbf{x}_1$. At convergence, $\mathbf{v}_t^\theta$ approximates the true $\mathbf{u}_t$, enabling generation of C_α coordinates.

2. Flow Matching for Amino-Acid Sequence. The flow matching framework for amino acid sequences operates in the discrete space of residue types $\{0, .., 19\}$. Following MultiFlow [7], we introduce a mask token M and define the flow to push an all-mask prior $p_0 = \delta\{\mathrm{M}\}$ toward the target sequence distribution $p_1 = \delta\{\mathbf{r}_1\}$, where $\delta\{i\}$ denotes the Kronecker delta (*i.e.*, a one-hot distribution centered at token i). To learn the "velocity", *i.e.* the rate matrix in probability space, we define a conditional path $p_t(\mathbf{r}_t|\mathbf{r}_1) = t\,\delta\{\mathbf{r}_1\} + (1-t)\,\delta\{\mathrm{M}\}$. This path interpolates between the masked and target sequences. In practice, it corresponds to a simple stochastic masking scheme: each residue is independently masked with probability $1-t$ and kept with probability t.

3. Flow Matching for Non-C_α Atoms. We adopt the same flow matching formulation used for C_α atoms. Specifically, we define a linear interpolant $\mathbf{z}_t = t\mathbf{z}_1 + (1-t)\mathbf{z}_0$, and train the parameterized velocity field $\mathbf{v}^\theta(\mathbf{z}_t, t)$ to match the exact velocity $\mathbf{z}_1 - \mathbf{z}_0$. There are two key differences with the C_α case. First, as each residue contains only a subset of the 36 possible non-C_α atoms determined by its residue type, we mask out non-existent atoms during interpolation. Second, revealing the presence or absence of specific atoms may leak residue type information for masked positions in the sequence, making the sequence denoising task trivial. To prevent this, we remove all non-C_α atoms for residues masked in $\mathbf{r}_t$ during training. During generation, to align with training, we only denoise non-C_α atoms once its residues are unmasked. Therefore, it is crucial to provide a good initialization for the non-C_α coordinates when a residue is unmasked—an issue we discuss in the following sections.

Local Coordinate Modeling for Non-C_α Atoms. Non-C_α atoms are structurally organized around their corresponding C_α atoms. To leverage this property, we offer two local coordinate modeling strategies, simplifying the learning task by predicting offsets rather than global coordinates and facilitating better initialization of non-C_α atoms. The first approach calculates the relative position of non-C_α atoms directly with respect to their corresponding C_α atom: $\mathbf{z}_i^{\text{local}} = \mathbf{z}_i - \mathbf{x}_i$. The second strategy, inspired by related work [28], constructs a residue-centric local coordinate frame $(\mathbf{t}_i, \mathbf{R}_i)$ with frame translations $\mathbf{t}_i$ and frame rotations $\mathbf{R}_i$ using the C_α coordinates of three neighboring residues $(\mathbf{x}_{i-1}, \mathbf{x}_i, \mathbf{x}_{i+1})$ via the Gram-Schmidt process. Non-C_α coordinates $\mathbf{z}_i$ are then transformed to local ones via $\mathbf{z}_i^{\text{local\_frame}} = \mathbf{R}_i^{-1}(\mathbf{z}_i - \mathbf{t}_i)$. Notably, while local coordinate transformations are a common technique in structure prediction models [23, 29], their application in atomistic structure generation remains underexplored [13].

4.2 Proteína-Atomística Architecture

The Proteína-Atomística architecture consists of two primary components: a core residue-level Transformer trunk and an atom-level Transformer encoder-decoder (Fig. 7). The residue-level trunk is responsible for the global backbone processing and is a high-capacity, non-equivariant architecture that leverages a stack of biased self-attention layers to predict the vector field for flow-based generation from noisy inputs. To address the complexities of atomistic modeling, our atom-level Transformer modules are designed to tackle three key challenges: handling the variable number of atoms, generalizing to atomistic representations, and initializing the fully masked non-C_α atoms of masked residues. We elaborate on our approach to these challenges in the subsequent paragraphs.

Atom Sequence Expansion. Each residue does not possess all 36 possible non-C_α atoms, resulting in empty dimensions in $\mathbf{z} \in \mathbb{R}^{L\times 36\times 3}$ that cannot be directly featurized. To address this, we expand the Atom37 representations into an atom sequence containing only existing atoms, following a default atom order. For masked residues, where all non-C_α atoms are absent, we represent them with a pseudo-atom token [M] in the atom sequence as a special atom type and set its coordinate to zero, and residue type to a mask token. For instance:

$$\underbrace{\text{[ALA]-[M]-[CYS]}}_{\text{amino-acid sequence}} \Longrightarrow \underbrace{\text{[N-CA-C-O-CB]-[CA-M]-[N-CA-C-O-CB-SG]}}_{\text{atom sequence}}$$

We then expand all associated residue-level features to match the atom sequence, allowing us to treat the atom sequence similarly to the residue sequence and reuse architectural modules. The Transformer's ability to handle variable-length inputs resolves the varying atom number problem.

Atom-Level Encoding and Decoding. Inspired by AlphaFold3 [1], we encode atom-level information using atom encoders followed by a cross-attention layer that integrates residue and atom features before the main backbone processing trunk. We note unlike prior methods our models do not use any triangle update layers. After this trunk, another cross-attention layer updates both representations, followed by atom decoders for further atom-level refinement [1]. At the output stage, residue-level representations are used to predict C_α vector field $\mathbf{v}^\theta_{\mathbf{x},t}$ and the residue type probability logits $\mathbf{c}^\theta_{1|t}$, while atom-level representations are used to predict the non-C_α vector field $\mathbf{v}^\theta_{\mathbf{z},t}$.

Initialization Prediction for Masked Residues. To predict the structure of non-C_α atoms of masked residues, we introduce a prediction head that leverages the pseudo-atom token's learned representation and context from neighboring atoms and residues. Our initial experiments revealed that, as expected, directly predicting the clean coordinates $\mathbf{z}$ is challenging as the number of atoms to predict is unknown. To address this, we propose learning an initialization $\mathbf{z}^\theta_{\text{init},t}$ through an augmented objective. We refer to this as an initialization as it is used only when the residue transitions from a masked to a non-mask state (see Alg. 2). During training, this initialization head is regressed towards $\mathbf{z}-\boldsymbol{\epsilon}_\mathbf{z}$, where $\boldsymbol{\epsilon}_\mathbf{z}$ is a randomly sampled Gaussian noise vector. Notably, the standard conditional flow matching objective relies on learning a vector field conditioned on noisy inputs; however, for side-chain initialization, there is no noisy input available, as the residue type is unknown. As a result, the model effectively learns to predict the expected clean state $\mathbf{z}$, representing an average side-chain structure across the 20 possible residue types. This initialization is refined into a realistic atomistic structure in the remaining denoising iterations during inference. Note that the initialization becomes easier to learn as the denoising process progresses, as more context is available and the remaining structure is less noisy, aligning with our choice of schedules for each explicit modality (Fig. 9). This approach also aligns the magnitude of the training target with that of vector fields, facilitating the training process. At generation time, this initialization serves as a reasonable approximation for the initial structure of non-C_α atoms in initially masked residues. See Appendix Sec. D.4 for more details.

5 Experiments

We trained two 200M parameter unconditional Proteína-Atomística models, for lengths (i) 32-400 and (ii) 32-256 using local coordinates without frames to align with prior baselines [37] (alternative coordinate modeling schemes are ablated in Table 4). For La-Proteína we train an autoencoder from scratch and a subsequent flow matching model according to the procedure described in La-Proteína [15] for lengths 32-500. The only difference between the original and our La-Proteína is

Table 1: **Proteína-Atomística and La-Proteína de novo fully atomistic protein generation performance** when trained on $\mathcal{D}_{\text{SYN}-\text{ours}}$ compared to baselines. All models generate 100 proteins for lengths $\in [50, 400]$ with step size 50. We report multimodal sampling configurations that generate the (i) most all-atom co-designable (codes), (ii) most diverse samples (div), and (iii) an optimal trade-off (opt). The best values are bolded.

Method	CODES-AA (%)↑	DES-M1 (%)↑	DIV-AA↑	NOV-PDB-AA↓	NOV-AFDB-AA↓
ProteinGenerator	10.0	57.1	28	0.75	0.78
Protpardelle	13.6	62.8	25	0.74	0.76
PLAID	22.3	34.9	63	0.85	0.88
Pallatom	51.6	62.5	282	**0.66**	**0.71**
La-Proteína ($\mathcal{D}_{\text{AFDB}-\text{clstr}}$)	70.6	85.5	314	0.77	0.84
Proteína-Atomística $_{\text{codes}}$	87.8	88.1	263	0.77	0.81
Proteína-Atomística $_{\text{opt}}$	83.1	85.8	321	0.76	0.80
Proteína-Atomística $_{\text{div}}$	71.6	72.0	333	0.75	0.80
La-Proteína $_{\text{codes},\mathcal{D}_{\text{SYN}-\text{ours}}}$	**92.6**	**92.5**	418	0.75	0.83
La-Proteína $_{\text{div},\mathcal{D}_{\text{SYN}-\text{ours}}}$	87.8	87.4	**475**	0.74	0.82

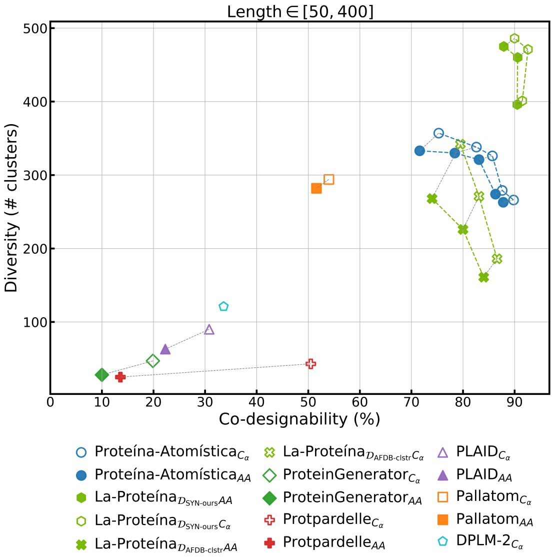

Figure 4: **Pareto frontier of the co-designability-diversity trade-off of Proteína-Atomística and La-Proteína** for proteins with length $\in [50, 400]$. Solid and hollow markers represent metrics calculated on all-atom and C_α basis, respectively. For atomistic models, the all-atom and C_α scores for the same generated proteins are connected by gray dashed line.

the change of training data. We emphasize that data is a critical hyperparameter in all prior de novo protein design methods. While we show Proteína-Atomística and La-Proteína to be state-of-the-art performers in Sec. 5.1, we also analyze the impact of $\mathcal{D}_{\text{SYN}-\text{ours}}$ specifically, in Sec. 5.2. As all included baselines leverage different datasets or combinations of AFDB/PDB/UniRef/etc., we intend for the public release of $\mathcal{D}_{\text{SYN}-\text{ours}}$ to offer another alternative that can be leveraged for its synthetic consistency. We further ablate our new explicit data-space method by comparing against recent backbone-only and backbone-sequence (no side chain) models in Appendix Tables 5-6, including a no-side-chain version of Proteína-Atomística itself (see Appendix Sec. C.1).

We evaluate our models using standard de novo protein design metrics, extending them to backbone-sequence co-design and all-atom (AA) contexts, following prior work [16, 37, 7]. De novo success metrics include **Designability (DES)**, the ability to inverse fold the generated protein backbone with ProteinMPNN and refold the generated sequences [47], with variants DES-M1 (single-shot) and DES-M8 (standard for backbone-only; best of 8 sequences); **Co-designability (CODES)**, similar to DES-M1 but using the model's output sequence; and **All-Atom Co-designability (CODES-AA)**, an extension of CODES using all-atom scRMSD. CODES and CODES-AA are reported for models that produce backbone and sequence, and atomistic side-chain structures, respectively. We also report structural **Diversity** and **Novelty** of the (co-)designable samples, for C_α design (M8 and M1), backbone-sequence co-design, and all-atom contexts. For metric details see Geffner et al. [16].

5.1 De Novo All-Atom Protein Generation

In Table 1, we compare Proteína-Atomística and La-Proteína trained on $\mathcal{D}_{\text{SYN}-\text{ours}}$ to recent fully atomistic generative models. Using multimodal low temperature sampling, both Proteína-Atomística and La-Proteína leverage the known trade-off [16, 28] between designability and diversity. We also plot the Pareto frontier for both all-atom and backbone-only co-designability in Fig. 4

Notably, Proteína-Atomística generates highly designable and diverse structures (Fig. 3) while achieving competitive novelty scores, indicating that our model does not overfit to PDB or AFDB. These improvements are further surpassed by La-Proteína when trained on our $\mathcal{D}_{\text{SYN}-\text{ours}}$, which obtains state-of-the-art performance with all-atom co-designability of 87.8% and 475 clusters when steered towards structural diversity via low temperature sampling. Furthermore, both Proteína-based models on average generate 66-70% α-helices and 6-10% β-sheets. We further demonstrate that both Proteína-Atomística and La-Proteína obtain comparable geometric side chain accuracy metrics in Appendix Fig. 11. The impact of synthetic consistency up to length 400 is evident in the comparison of La-Proteína with $\mathcal{D}_{\text{AFDB}-\text{clstr}}$ and $\mathcal{D}_{\text{SYN}-\text{ours}}$, where we observe a best-case improvement in all-atom co-designability of 31% and diversity of 51%, respectively, establish new state-of-the-art results. We further discuss the generalization of performance gains due to synthetic consistent data in Sec. 5.2.

5.2 Understanding the Impact of Synthetic Data

To demonstrate that $\mathcal{D}_{\text{AFDB}-\text{clstr}}$ is challenging for facilitating joint learning of sequence and structure, we further investigated the impact of synthetic data. To this end, we constructed two further synthetic datasets based on $\mathcal{D}_{\text{AFDB}-\text{clstr}}$: (1) $\mathcal{D}_{\text{ESMFold}}$ and (2) $\mathcal{D}_{\text{des}}$. In $\mathcal{D}_{\text{ESMFold}}$, samples have

Table 2: **Impact of Synthetic Data.** All models generate 100 proteins for lengths ∈ [50, 100, 150, 200, 250]. Training on ESMFold structures or filtering for ESMFold designability hurts performance unless those synthetic ESMFold structures are coupled with recoverable sequences.

Model	CODES-AA (%)↑	DES-M1 (%)↑	DIV-AA↑
Proteína-Atomística $\mathcal{D}_{\text{AFDB-clstr}}$	76.8	87.6	154
Proteína-Atomística $\mathcal{D}_{\text{ESMFold}}$	71.0	86.0	132
Proteína-Atomística $\mathcal{D}_{\text{Des}}$	72.2	87.2	120
La-Proteína $\mathcal{D}_{\text{AFDB-clstr}}$	81.0	89.8	213
Proteína-Atomística $\mathcal{D}_{\text{SYN-ours}}$	81.2	82.4	267
La-Proteína $\mathcal{D}_{\text{SYN-ours}}$	**92.2**	**93.2**	**283**

the same sequences as in $\mathcal{D}_{\text{AFDB-clstr}}$ but the structures are computed by ESMFold with a filter of pLDDT ≥ 0.8. $\mathcal{D}_{\text{Des}}$ is a subset of $\mathcal{D}_{\text{AFDB-clstr}}$ (uses direct AFDB structures) with all structures passing the DES-M8 filter. Both $\mathcal{D}_{\text{ESMFold}}$ and $\mathcal{D}_{\text{Des}}$ contain $\sim$0.16M samples.

Table 2 demonstrates that, counterintuitively, neither using 100% designable structures $\mathcal{D}_{\text{ESMFold}}$ for training nor leveraging the designable subset $\mathcal{D}_{\text{Des}}$ improves the performance of the model, even when the goal is to generate designable and diverse structures. As a side, Table 2 also confirms that the new Proteína-Atomística architecture trained on $\mathcal{D}_{\text{SYN-ours}}$, which combines AFDB's structural diversity with ProteinMPNN sequences (subsequently refolded with ESMFold to recover consistent full atomistic detail), achieves highly accurate and diverse fully atomistic generation (see Sec. 3 for $\mathcal{D}_{\text{SYN-ours}}$ procedure). This highlights the importance of utilizing better-aligned synthetic sequences *and* structures to facilitate scalable co-design over both modalities. Furthermore by training on $\mathcal{D}_{\text{SYN-ours}}$ La-Proteína sees co-designability and diversity improvements of 13.8% and 32.9%.

5.3 Latent vs. Explicit Modeling of Protein Sequences

Table 2 shows that La-Proteína's latent approach better learns aligned sequence-structure co-generation compared to Proteína-Atomística in particular when trained on $\mathcal{D}_{\text{AFDB-clstr}}$. We found that this is due to lower co-designability at longer lengths, also implying lower diversity scores (diversity is calculated among designable samples only). La-Proteína's autoencoder bypasses the challenge of aligning explicit, discrete, and continuous modalities, generating more diverse and co-designable samples. The latent variable framework avoids minimizing a complex joint continuous and discrete objective during the generative model training. Moreover, La-Proteína's autoencoder component effectively learns to tie together consistent sequences and structures rather than trying to learn how to explicitly match them through separate modality-based objectives. Although learning the structure-to-sequence mapping in the explicit data space is more challenging, Proteína-Atomística establishes a strong alternative for future work that relies on direct access to explicit observables.

Switching to $\mathcal{D}_{\text{SYN-ours}}$ significantly improves Proteína-Atomística's co-designability, dramatically boosts diversity, and yields competitive results with La-Proteína. The model now learns a more empirically recoverable sequence distribution from its structures (Fig. 6). Notably, both La-Proteína and Proteína-Atomística show significant improvements on $\mathcal{D}_{\text{SYN-ours}}$, generating more co-designable and diverse samples when compared to training on $\mathcal{D}_{\text{AFDB-clstr}}$. Furthermore, sequences generated from our models trained on $\mathcal{D}_{\text{SYN-ours}}$ fold better into their co-generated structures than those from ProteinMPNN (Appendix Table 7; see $\geq$DES-M1). This alleviates the need for ProteinMPNN re-design of generated backbones, a common component in design pipelines.

Please see our Appendix for ablation studies, experiment, dataset, and model architecture details.

6 Conclusions

Our study finds that AFDB structures are not recoverable with publicly available protein structure predictions models, which motivated us to create a carefully curated, yet diverse dataset of aligned sequences and structures. We also introduce and successfully validate Proteína-Atomística, a new unified multi-modal flow-based framework for de novo atomistic protein design that represents sequence, backbone, and side chains explicitly, without latent variables. Training both Proteína-Atomística and La-Proteína on $\mathcal{D}_{\text{SYN-ours}}$ dramatically improves their performance, achieving new state-of-the-art results. This demonstrates the critical importance of consistent and recoverable sequence-structure training data for atomistic protein design. Future work could address the consistent generation of longer atomistic proteins and analyze the importance of aligned sequence-structure data in the context of conditional tasks such as motif scaffolding and binder design.

References

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

[2] W. Ahern, J. Yim, D. Tischer, S. Salike, S. Woodbury, D. Kim, I. Kalvet, Y. Kipnis, B. Coventry, H. Altae-Tran, et al. Atom level enzyme active site scaffolding using rfdiffusion2. *bioRxiv*, pages 2025–04, 2025.

[3] S. Alamdari, N. Thakkar, R. van den Berg, N. Tenenholtz, B. Strome, A. Moses, A. X. Lu, N. Fusi, A. P. Amini, and K. K. Yang. Protein generation with evolutionary diffusion: sequence is all you need. *BioRxiv*, pages 2023–09, 2023.

[4] I. Anishchenko, S. J. Pellock, T. M. Chidyausiku, T. A. Ramelot, S. Ovchinnikov, J. Hao, K. Bafna, C. Norn, A. Kang, A. K. Bera, et al. De novo protein design by deep network hallucination. *Nature*, 600(7889):547–552, 2021.

[5] I. Barrio-Hernandez, J. Yeo, J. Jänes, M. Mirdita, C. L. M. Gilchrist, T. Wein, M. Varadi, S. Velankar, P. Beltrao, and M. Steinegger. Clustering predicted structures at the scale of the known protein universe. *Nature*, 622:637–645, 2023.

[6] J. Bose, T. Akhound-Sadegh, G. Huguet, K. Fatras, J. Rector-Brooks, C.-H. Liu, A. C. Nica, M. Korablyov, M. M. Bronstein, and A. Tong. SE(3)-stochastic flow matching for protein backbone generation. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024.

[7] A. Campbell, J. Yim, R. Barzilay, T. Rainforth, and T. Jaakkola. Generative flows on discrete state-spaces: Enabling multimodal flows with applications to protein co-design. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2024.

[8] R. Chen, D. Xue, X. Zhou, Z. Zheng, X. Zeng, and Q. Gu. An all-atom generative model for designing protein complexes, 2025.

[9] A. E. Chu, J. Kim, L. Cheng, G. E. Nesr, M. Xu, R. W. Shuai, and P.-S. Huang. An all-atom protein generative model. *Proceedings of the National Academy of Sciences*, 121(27):e2311500121, 2024.

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

[11] I. W. Davis, A. Leaver-Fay, V. B. Chen, J. N. Block, G. J. Kapral, X. Wang, L. W. Murray, W. B. Arendall III, J. Snoeyink, J. S. Richardson, et al. Molprobity: all-atom contacts and structure validation for proteins and nucleic acids. *Nucleic acids research*, 35(suppl_2):W375–W383, 2007.

[12] D. del Alamo, R. Frick, D. Truan, and J. Karpiak. Adapting proteinmpnn for antibody design without retraining. *bioRxiv*, 2025.

[13] A. dos Santos Costa, I. Mitnikov, M. Geiger, M. Ponnapati, T. Smidt, and J. Jacobson. Ophiuchus: Scalable modeling of protein structures through hierarchical coarse-graining so(3)-equivariant autoencoders, 2023.

[14] N. Ferruz, S. Schmidt, and B. Höcker. Protgpt2 is a deep unsupervised language model for protein design. *Nature communications*, 13(1):4348, 2022.

[15] T. Geffner, K. Didi, Z. Cao, D. Reidenbach, Z. Zhang, C. Dallago, E. Kucukbenli, K. Kreis, and A. Vahdat. La-proteina: Atomistic protein generation via partially latent flow matching. *arXiv preprint arXiv:2507.09466*, 2025.

[16] T. Geffner, K. Didi, Z. Zhang, D. Reidenbach, Z. Cao, J. Yim, M. Geiger, C. Dallago, E. Kucukbenli, A. Vahdat, and K. Kreis. Proteina: Scaling flow-based protein structure generative models. In *International Conference on Learning Representations (ICLR)*, 2025.

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

[18] B. Huang, Y. Xu, X. Hu, Y. Liu, S. Liao, J. Zhang, C. Huang, J. Hong, Q. Chen, and H. Liu. A backbone-centred energy function of neural networks for protein design. *Nature*, 602(7897):523–528, 2022.

[19] P.-S. Huang, Y.-E. A. Ban, F. Richter, I. Andre, R. Vernon, W. R. Schief, and D. Baker. Rosettaremodel: a generalized framework for flexible backbone protein design. *PloS one*, 6(8):e24109, 2011.

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

[21] G. Huguet, J. Vuckovic, K. Fatras, E. Thibodeau-Laufer, P. Lemos, R. Islam, C.-H. Liu, J. Rector-Brooks, T. Akhound-Sadegh, M. Bronstein, A. Tong, and A. J. Bose. Sequence-augmented se(3)-flow matching for conditional protein backbone generation. *arXiv preprint arXiv:2405.20313*, 2024.

[22] J. Ingraham, M. Baranov, Z. Costello, V. Frappier, A. Ismail, S. Tie, W. Wang, V. Xue, F. Obermeyer, A. Beam, and G. Grigoryan. Illuminating protein space with a programmable generative model. *Nature*, 623:1070–1078, 2023.

[23] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Zidek, A. Potapenko, A. Bridgland, C. Meyer, S. A. A. Kohl, A. J. Ballard, A. Cowie, B. Romera-Paredes, S. Nikolov, R. Jain, J. Adler, T. Back, S. Petersen, D. Reiman, E. Clancy, M. Zielinski, M. Steinegger, M. Pacholska, T. Berghammer, S. Bodenstein, D. Silver, O. Vinyals, A. W. Senior, K. Kavukcuoglu, P. Kohli, and D. Hassabis. Highly accurate protein structure prediction with alphafold. *Nature*, 596:583–589, 2021.

[24] H. Kim, M. Mirdita, and M. Steinegger. Foldcomp: a library and format for compressing and indexing large protein structure sets. *Bioinformatics*, 39(4):btad153, 03 2023.

[25] I. V. Korendovych and W. F. DeGrado. De novo protein design, a retrospective. *Quarterly reviews of biophysics*, 53:e3, 2020.

[26] B. Kuhlman and P. Bradley. Advances in protein structure prediction and design. *Nat. Rev. Mol. Cell Biol.*, 20:681–697, 2019.

[27] Y. Lin and M. Alquraishi. Generating novel, designable, and diverse protein structures by equivariantly diffusing oriented residue clouds. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, 2023.

[28] Y. Lin, M. Lee, Z. Zhang, and M. AlQuraishi. Out of many, one: Designing and scaffolding proteins at the scale of the structural universe with genie 2. *arXiv preprint arXiv:2405.15489*, 2024.

[29] Z. Lin, H. Akin, R. Rao, B. Hie, Z. Zhu, W. Lu, N. Smetanin, R. Verkuil, O. Kabeli, Y. Shmueli, A. dos Santos Costa, M. Fazel-Zarandi, T. Sercu, S. Candido, and A. Rives. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023.

[30] S. L. Lisanza, J. M. Gershon, S. W. Tipps, J. N. Sims, L. Arnoldt, S. J. Hendel, M. K. Simma, G. Liu, M. Yase, H. Wu, et al. Multistate and functional protein design using rosettafold sequence space diffusion. *Nature biotechnology*, pages 1–11, 2024.

[31] A. X. Lu, W. Yan, S. A. Robinson, S. Kelow, K. K. Yang, V. Gligorijevic, K. Cho, R. Bonneau, P. Abbeel, and N. C. Frey. All-atom protein generation with latent diffusion. In *ICLR 2025 Workshop on Generative and Experimental Perspectives for Biomolecular Design*, 2025.

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

[33] M. Mirdita, K. Schütze, Y. Moriwaki, L. Heo, S. Ovchinnikov, and M. Steinegger. Colabfold: making protein folding accessible to all. *Nature Methods*, 19(6):679–682, 2022.

[34] M. Pacesa, L. Nickel, C. Schellhaas, J. Schmidt, E. Pyatova, L. Kissling, P. Barendse, J. Choudhury, S. Kapoor, A. Alcaraz-Serna, et al. Bindcraft: one-shot design of functional protein binders. *bioRxiv*, pages 2024–09, 2024.

[35] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[36] F. Z. Peng, Z. Bezemek, S. Patel, J. Rector-Brooks, S. Yao, A. Tong, and P. Chatterjee. Path planning for masked diffusion models with applications to biological sequence generation. In *ICLR 2025 Workshop on Deep Generative Model in Machine Learning: Theory, Principle and Efficacy*, 2025.

[37] W. Qu, J. Guan, R. Ma, K. Zhai, W. Wu, and H. Wang. P (all-atom) is unlocking new path for protein design. *bioRxiv*, pages 2024–08, 2024.

[38] M. Ren, T. Zhu, and H. Zhang. Carbonnovo: Joint design of protein structure and sequence using a unified energy-based model. In *Forty-first International Conference on Machine Learning*, 2024.

[39] J. S. Richardson and D. C. Richardson. The de novo design of protein structures. *Trends in Biochemical Sciences*, 14(7):304–309, 1989.

[40] Z. Tang, S. Gu, J. Bao, D. Chen, and F. Wen. Improved vector quantized diffusion models, 2022.

[41] B. L. Trippe, J. Yim, D. Tischer, D. Baker, T. Broderick, R. Barzilay, and T. S. Jaakkola. Diffusion probabilistic modeling of protein backbones in 3d for the motif-scaffolding problem. In *The Eleventh International Conference on Learning Representations (ICLR)*, 2023.

[42] M. van Kempen, S. S. Kim, C. Tumescheit, M. Mirdita, J. Lee, C. L. M. Gilchrist, J. Söding, and M. Steinegger. Fast and accurate protein structure search with foldseek. *Nat Biotechnol.*, 42:243–246, 2024.

[43] M. Varadi, S. Anyango, M. Deshpande, S. Nair, C. Natassia, G. Yordanova, D. Yuan, O. Stroe, G. Wood, A. Laydon, A. Žídek, T. Green, K. Tunyasuvunakool, S. Petersen, J. Jumper, E. Clancy, R. Green, A. Vora, M. Lutfi, and S. Velankar. Alphafold protein structure database: Massively expanding the structural coverage of protein-sequence space with high-accuracy models. *Nucleic Acids Research*, 50:D439–D444, 2021.

[44] C. Wang, Y. Qu, Z. Peng, Y. Wang, H. Zhu, D. Chen, and L. Cao. Proteus: Exploring protein structure generation for enhanced designability and efficiency. *bioRxiv*, 2024.

[45] X. Wang, Z. Zheng, F. Ye, D. Xue, S. Huang, and Q. Gu. Diffusion language models are versatile protein learners. In *International Conference on Machine Learning*, 2024.

[46] X. Wang, Z. Zheng, F. Ye, D. Xue, S. Huang, and Q. Gu. Dplm-2: A multimodal diffusion protein language model. In *International Conference on Learning Representations (ICLR)*, 2025.

[47] J. L. Watson, D. Juergens, N. R. Bennett, B. L. Trippe, J. Yim, H. E. Eisenach, W. Ahern, A. J. Borst, R. J. Ragotte, L. F. Milles, B. I. M. Wicky, N. Hanikel, S. J. Pellock, A. Courbet, W. Sheffler, J. Wang, P. Venkatesh, I. Sappington, S. V. Torres, A. Lauko, V. D. Bortoli, E. Mathieu, R. Barzilay, T. S. Jaakkola, F. DiMaio, M. Baek, and D. Baker. De novo design of protein structure and function with rfdiffusion. *Nature*, 620:1089–1100, 2023.

[48] J. Wohlwend, G. Corso, S. Passaro, N. Getz, M. Reveiz, K. Leidal, W. Swiderski, L. Atkinson, T. Portnoi, I. Chinn, J. Silterra, T. Jaakkola, and R. Barzilay. Boltz-1: Democratizing biomolecular interaction modeling. *bioRxiv*, 2024.

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

[50] J. Yim, B. L. Trippe, V. D. Bortoli, E. Mathieu, A. Doucet, R. Barzilay, and T. Jaakkola. SE(3) diffusion model with application to protein backbone generation. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, 2023.

Appendix

A Additional Proteína-Atomística Sample Visualizations

In Fig. 5, we show additional fully atomistic proteins generated by Proteína-Atomística. Our model outputs diverse (co-)designable samples, including realistic side chain structures.

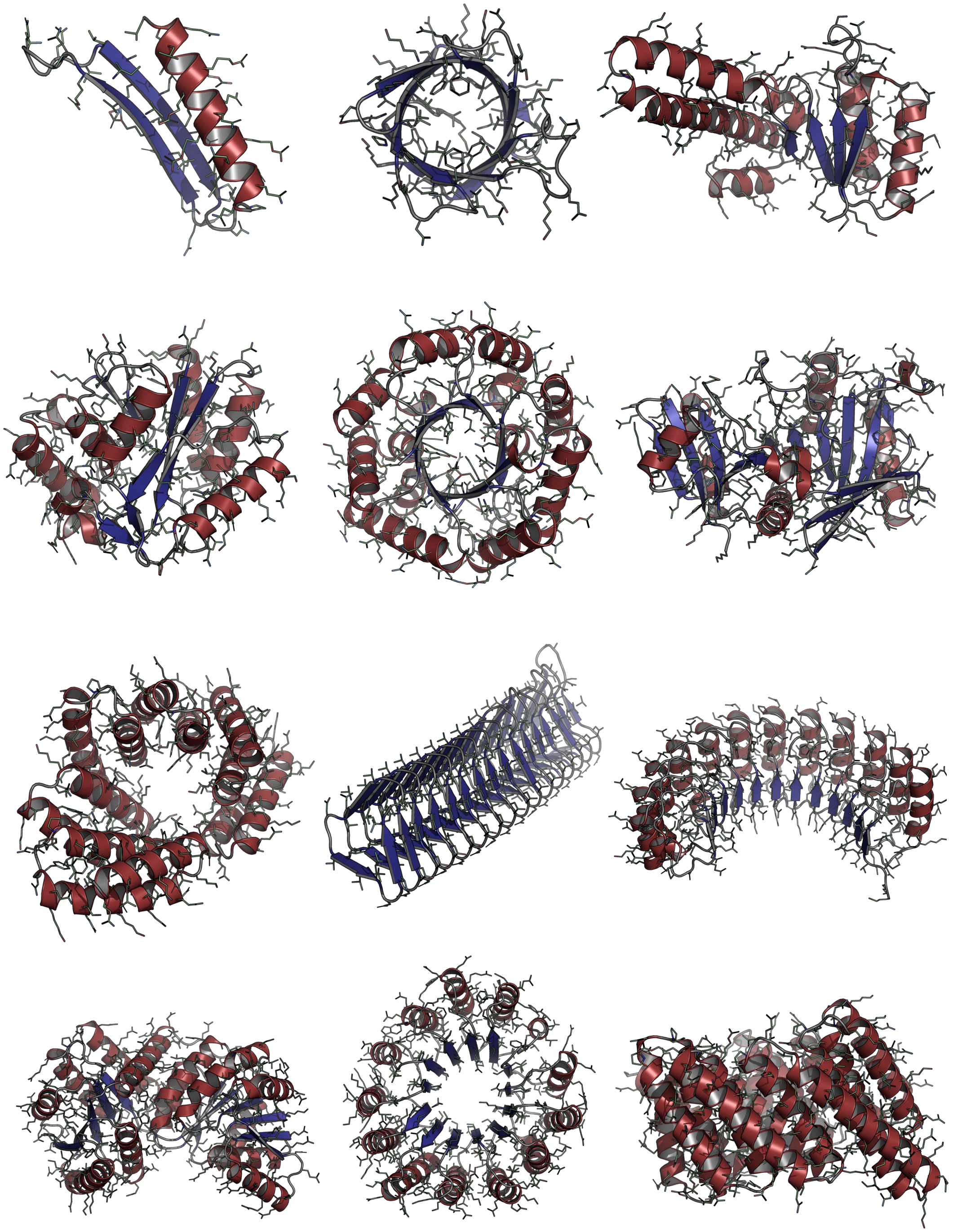

Figure 5: **Proteína-Atomística Samples**. Additional fully atomistic proteins generated by our model, ranging from 100 to 400 residues, including side chains. All shown samples are co-designable.

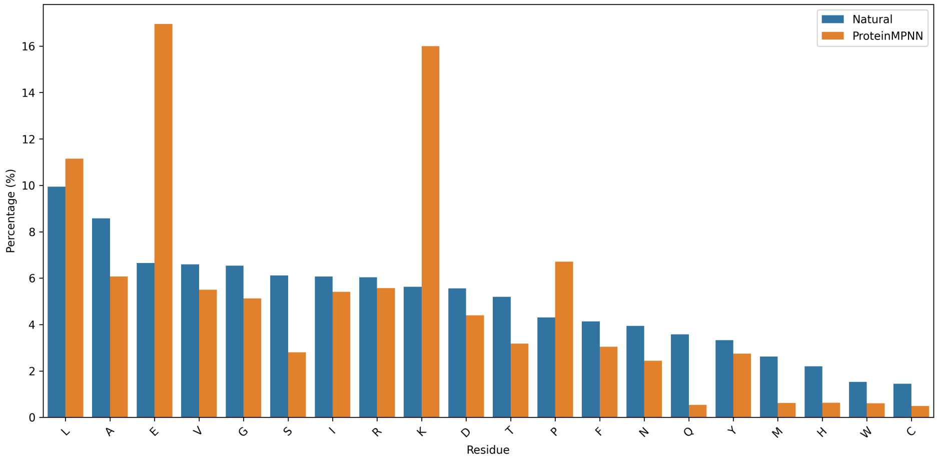

Figure 6: Amino acid sequence distribution for $\mathcal{D}_{\text{AFDB}-\text{clstr}}$ (natural) and $\mathcal{D}_{\text{SYN}-\text{ours}}$ (synthetic) for sequence length $\in [32, 256]$.

B Dataset Details

B.1 Dealing with the inconsistency between structure and sequence data of AFDB

$\mathcal{D}_{\text{AFDB}-\text{clstr}}$ is the dataset used in Genie2 [28] for training a protein backbone generative model. It is a subset of the AlphaFold Database (AFDB) [43] containing proteins clustered with both MMseqs2 [5] based on sequence similarity and Foldseek [42] based on structure similarity. $\mathcal{D}_{\text{AFDB}-\text{clstr}}$ only contains one structure per cluster (the cluster representative). After filtering with $N_{\text{residue}} \in [32, 256]$ and pLDDT≥80, $\mathcal{D}_{\text{AFDB}-\text{clstr}}$ contains 588,318 structures.

We analyzed the co-designability of structure-sequence pairs in $\mathcal{D}_{\text{AFDB}-\text{clstr}}$ by folding the sequences with ESMFold and checking if the lowest RMSD between folded structure and the AFDB structure is less than 2Å. We discovered that only 26.6% of $\mathcal{D}_{\text{AFDB}-\text{clstr}}$ are co-designable by C_α RMSD and even less by all-atom RMSD (Fig. 1). The low co-designability of $\mathcal{D}_{\text{AFDB}-\text{clstr}}$ poses a significant challenge to multimodal protein generation model training: even if the model fit both sequence and structure distributions of the dataset very well, the generated protein structure will not be consistent with the generated sequence. To address this challenge, we explored three other synthetic datasets based on downstream augmentations of $\mathcal{D}_{\text{AFDB}-\text{clstr}}$.

B.2 Datasets with designable structures

Our initial explorations began with targeting the structure-sequence inconsistency at the structural level. We hypothesized that refolding $\mathcal{D}_{\text{AFDB}-\text{clstr}}$ with ESMFold would, by definition, yield 100% co-designable samples, and that training on these designable samples would improve model performance. To test this hypothesis, we created two datasets:

1. D_{ESMFold}: We took sequences in the original $\mathcal{D}_{\text{AFDB}-\text{clstr}}$ and folded them with ESMFold. We applied a filter of $\text{pLDDT}_{\text{ESMFold}} \geq 80$ on the folded structures. As a result, all remaining structures should be co-designable with a high confidence score. D_{ESMFold} contained 163,552 samples.

2. D_{Des}: We took sequences in the original $\mathcal{D}_{\text{AFDB}-\text{clstr}}$ and folded them with ESMFold. We computed the all-atom RMSD between the ESMFold-folded structure and the original AFDB structure. We then filtered $\mathcal{D}_{\text{AFDB}-\text{clstr}}$ based on the all-atom RMSD with 2Å cutoff to create the D_{des} dataset containing 155,957 samples. Hence, in contrast to D_{ESMFold}, here we are relying on the original $\mathcal{D}_{\text{AFDB}-\text{clstr}}$ structures and filtering them to a designable subset.

We trained *Proteína-Atomística* on both D_{ESMFold} and D_{des}. Both datasets reduced the model's performance on all metrics (Table 2) and caused sequence overfitting (structure losses remained unaffected) despite the sequences for both datasets remaining unchanged. The results suggest that jointly modeling natural sequences and synthetic structures (predicted by either ESMFold or AF2) remains challenging when those structures cannot be easily recovered with both ESMFold and AF2[2]. While $\mathcal{D}_{\text{AFDB}-\text{clstr}}$ provides a diverse sequence and structure space, realigning the structures to the

[2] ColabFold with MSAs yields $\sim$65% co-designability on a random subset of 100 samples from $\mathcal{D}_{\text{AFDB}-\text{clstr}}$.

sequences using ESMFold and filtering by its confidence score may inadvertently reduce both data diversity and volume. This process may also contribute to the observed overfitting.

B.3 Details of Our Synthetic Data $\mathcal{D}_{\mathrm{SYN-ours}}$

Given that a sequence must reasonably fold into its given structure to be co-designable, and enforcing co-designability at the structural level worsened all models, we shifted our focus to the sequences. We observed that ProteinMPNN-based sequence resampling can significantly improve designability (see Sec. F.2.2 for detailed discussion), prompting us to create a dataset with synthetic sequences to target the central issue of inconsistent sequence-structure pairs. This choice is further motivated by the fact that ProteinMPNN is widely used for the “inverse folding” step in the standard multi-step “backbone generation”-“inverse folding”-“forward folding” pipeline employed by most de novo protein generative models, owing to its validated performance in wet-lab experiments [47, 10, 12].

It is crucial to note that, since we are modeling fully atomistic protein structures, we cannot utilize the given AFDB structures if there are any residue changes in the predicted synthetic sequences. This is because a change in sequence implies a different side-chain structure, potentially with a different number of atoms. Consequently, to use synthetic sequences, we refold the new sequences to ensure all-atom compatibility. We visualize both the natural and synthetic sequence distributions in Fig. 6.

To address the problems discussed in Sec. B.2 while preserving scale and diversity, we created $\mathcal{D}_{\mathrm{codes}}$ through the following steps: (i) generating four ProteinMPNN sequences for each $\mathcal{D}_{\mathrm{AFDB-clstr}}$ structure with lengths between 32 and 400, (ii) folding the ProteinMPNN sequences with ESMFold due to its computational efficiency, being $\sim 60\times$ faster than AF2, and (iii) selecting the sequence-structure pair with the lowest C_α RMSD to the original AFDB structure to preserve the structural diversity, as the original AFDB structures are cluster representatives. After filtering out samples with an average ESMFold pLDDT below 80, our curated dataset, which combines knowledge from ProteinMPNN and the confident predictions of both ESMFold and AF2, results in 429,965 high-quality samples. Furthermore, rather than relying on redesigning the sequence after structure-based generation and regenerating the side chains each time, $\mathcal{D}_{\mathrm{codes}}$ enables learning a consistent sequence-structure distribution, facilitating accurate single-step, fully atomistic design. It is worth mentioning that FoldComp [24] was used to store and access all datasets we prepared efficiently.

We present the amino acid residue distribution of all training samples, ranging in length from 32 to 256, in Fig. 6 for both the natural $\mathcal{D}_{\mathrm{AFDB-clstr}}$ sequences and those generated using ProteinMPNN in $\mathcal{D}_{\mathrm{SYN-ours}}$. We chose ProteinMPNN for its robust wetlab validation [10, 12]. However, it does overrepresent certain residue types, particularly charged species (E, K). While this overrepresentation is not inherently problematic for de novo design, as it allows the model to generate fully atomistic structures with high fidelity without redesign, it is still an important consideration for downstream usage.

C Architecture Details

Here we introduce the model versions in order of complexity. Starting with Proteína we add discrete sequence co-generation to create Proteína-Co-Design. We then extend this with side-chain co-generation to yield the full Proteína-Atomística framework.

C.1 Proteína-Co-Design

For the co-design setting, we start from the $\sim 60M$ Proteína architecture configuration that shows an optimal balance of accuracy and speed in the backbone-only setting (See Appendix C.2 of Geffner et al. [16] for Proteína speed analysis). To enable joint backbone-sequence modeling from a pure backbone model, we add three features:

(i) residue type index embeddings

(ii) argmax residue type index predictions for self-conditioning,

(iii) the independent residue type time variable, which dictates how much noise or, in this case, the percentage of tokens to be replaced with MASK tokens

Table 3: Hyperparameters for Proteína-Atomística model training. Rows highlighed in grey are specfic to the all-atom architecture. We denote two versions of Proteína-Atomística the one trained on shorter lengths up to 256 and the standard model trained to max length 400.

Model	**Proteína-**				
	Co-design	**Atomística (256)**	**Atomística (400)**	**Atomística Motif**	**Atomística-tri**
Architecture Component					
initialization	random	random	random	random	random
sequence repr dim	512	768	768	512	768
# registers	10	10	10	10	10
sequence cond dim	128	512	512	128	512
t sinusoidal enc dim	196	256	512	196	512
idx. sinusoidal enc dim	196	128	256	196	256
pair repr dim	196	512	256	196	256
seq separation dim	128	128	128	128	128
pair distances dim ($\mathbf{x}_t$)	64	64	64	64	64
pair distances dim ($\hat{\mathbf{x}}(\mathbf{x}_t)$)	128	128	128	128	128
pair distances min (Å)	1	1	1	1	1
pair distances max (Å)	30	30	30	30	30
residue type embedding dim	196	512	512	196	512
# attention heads	12	12	12	12	12
# transformer layers	12	15	15	12	15
# triangle layers	0	0	0	0	3
# number of atom layers	0	5	5	5	5
atom cond dim	0	128	128	128	128
atom dim	0	128	128	128	128
atom type embedding dim	0	128	128	128	128
# atom attention heads dim	0	8	8	8	8
# atom cross attention heads	0	8	8	8	8
side chain coords	N/A	local trans	local frame	local trans	local frame
# trainable parameters	59.3M	221M	222M	73.6M	226M
Training Details					
# train steps (length∈[32, 256])	100K	190k	210k	100K	145k
# finetune steps (length∈[32, 400])	N/A	N/A	100k	N/A	N/A
train batch size per GPU	28	8	12	8	4
finetune batch size per GPU	N/A	N/A	1	N/A	N/A
# GPUs	96	96	96	96	96
# grad. acc. steps	1	1	1	1	1
% forward folding	10	5	10	5	10
% inverse folding	10	5	10	5	10
% side chain packing	0	0	5	0	5

We note that both the C_α coordinates and residue types leverage self-conditioning, where in 50% of the training iterations, we run a first model forward pass to obtain predictions of the current structure and sequence and use those as additional inputs to the model during a second forward pass. This is a common technique for improving diffusion models and can be viewed as a form of recycling employed by AlphaFold2 [16, 23].

For the Co-design task only, we sample the sequence time from $\mathcal{B}(1.0, 2.5)$, where $\mathcal{B}(\cdot, \cdot)$ is the Beta distribution. This is a severely left-skewed distribution, which gives more weight to noisy times (sequences with a higher masking rate). For reference, we found that this did not make an impact in the all-atom task. Instead, we used the standard uniform distribution, given that we were directly modeling the structure-sequence duality with residue types and their structures. For co-design training, 10% of the batch iterations are used for forward and inverse folding, respectively. This was done to pin the two independent schedules so that when both structure and sequence time reach one, the structure and sequences are trained to align. Please see Table 3 for complete model configurations and compute resources used.

C.2 Proteína-Atomística

More architectural components are illustrated in Fig. 8.

C.3 Optional Triangle Multiplicative Updates

In addition to the highly scalable *Proteína-Atomística* demonstrated by Fig. 7, we trained another variant, *Proteína-Atomística-tri*, with triangle multiplicative layers, which were used to update the pair

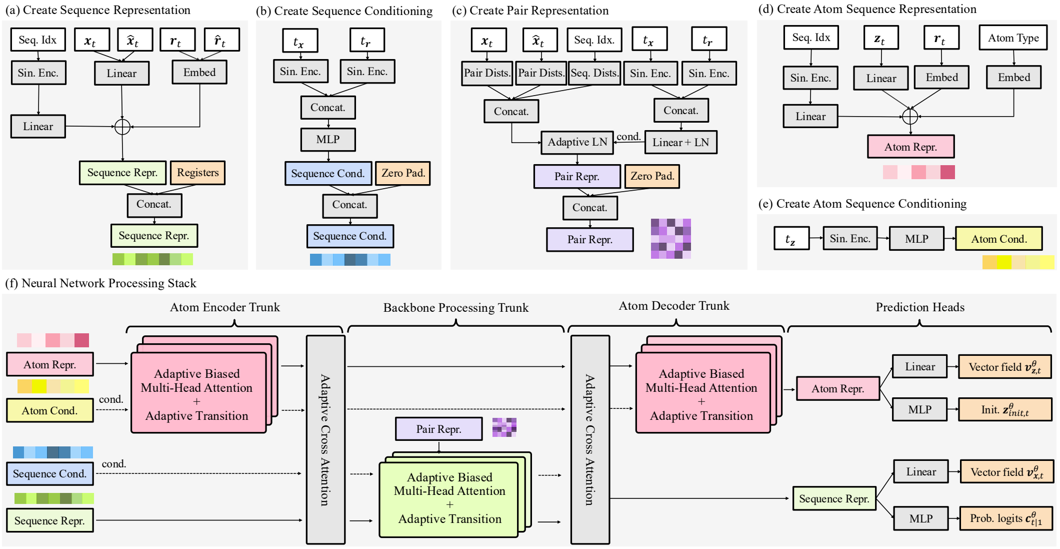

Figure 7: **Proteína-Atomística's transformer architecture.** *(a)-(c)* First generate an initial sequence representation, sequence conditioning features, and a pair representation. *(d)-(e)* Create atom representations and atom conditioning features for the expanded atom sequence. *(f)* Process these representations iteratively through trunks, moving from atom-level to sequence-level and back to atom-level. Each trunk incorporates conditioned multi-head attention layers, biased by the pair representation. Adaptive cross-attention is employed between trunks to update atom and sequence representations (see Appendix).

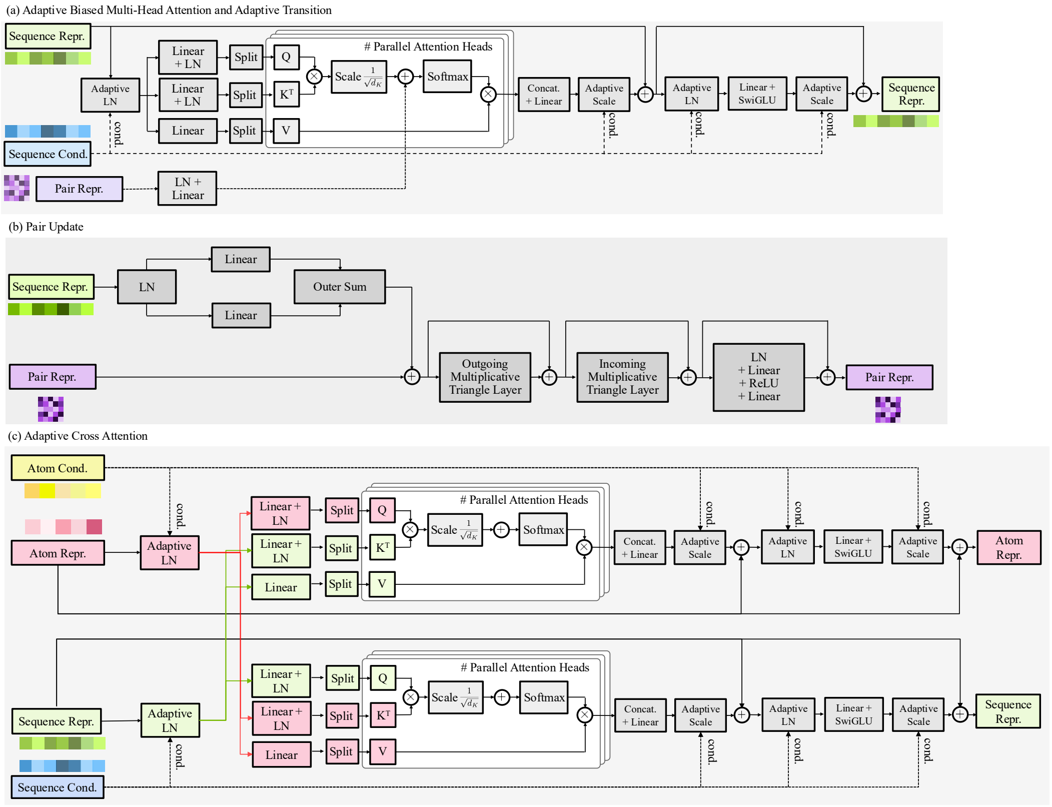

Figure 8: **Additional modules of Proteína-Atomística transformer architecture.** (a) Adaptive attention and transition. (b) Optional pair representation update with triangle multiplicative layers. (c) Adaptive cross attention.

representation. Fig. 8(b) shows how triangle multiplicative layers are used in the *Proteína-Atomística* architecture. During training, the pair representation was updated every 5 backbone processing layers, where the backbone processing layers are the core transformer layers shown in Fig. 8(a), resulting

Algorithm 1 Proteína-Atomística Training

```
while not converged do
    Sample protein $(\mathbf{x}, \mathbf{r}, \mathbf{z})$ from dataset
    Sample time steps $t_{\mathbf{x}}, t_{\mathbf{r}}, t_{\mathbf{z}}$ for each modality
    Convert global $\mathbf{z}$ to local coordinates using $\mathbf{x}$
    Sample noisy input $\mathbf{x}_t, \mathbf{r}_t, \mathbf{z}_t$ for each modality
    Zero out $\mathbf{z}_t$ for masked residues in $\mathbf{r}_t$

    Predict $\mathbf{v}^{\theta}_{\mathbf{x},t}$, $\mathbf{v}^{\theta}_{\mathbf{z},t}$, $\mathbf{z}^{\theta}_{\text{init},t}$ and $\mathbf{c}^{\theta}_{1|t}$
    Compute loss across modalities
    $\mathcal{L}_{\mathbf{x}} \leftarrow \|\mathbf{v}^{\theta}_{\mathbf{x},t} - (\mathbf{x} - \boldsymbol{\epsilon}_{\mathbf{x}})\|_2^2$
    $\mathcal{L}_{\mathbf{r}} \leftarrow \text{CrossEntropy}(\mathbf{c}^{\theta}_{1|t}, \mathbf{r})$
    for each residue $i$ do
        $\mathcal{L}_{\mathbf{z},i} \leftarrow \|\mathbf{v}^{\theta}_{\mathbf{z},t,i} - (\mathbf{z}_i - \boldsymbol{\epsilon}_{\mathbf{z},i})\|_2^2$, if $\mathbf{r}_{t,i} \neq \text{M}$
        $\mathcal{L}_{\mathbf{z},i} \leftarrow \|\mathbf{z}^{\theta}_{\text{init},t,i} - (\mathbf{z}_i - \boldsymbol{\epsilon}_{\mathbf{z},i})\|_2^2$, if $\mathbf{r}_{t,i} = \text{M}$
    end for
    $\mathcal{L} \leftarrow \frac{1}{L}(\mathcal{L}_{\mathbf{x}} + \mathcal{L}_{\mathbf{r}} + \mathcal{L}_{\mathbf{z}})$
    Calculate gradient and update model parameters
end while
```

Algorithm 2 Proteína-Atomística Sampling

```
Initialize $\mathbf{x}, \mathbf{r}, \mathbf{z}$ from noise distribution
for $i = 0$ to $N - 1$ do
    Predict $\mathbf{v}^{\theta}_{\mathbf{x},t}$, $\mathbf{v}^{\theta}_{\mathbf{z},t}$, $\mathbf{z}^{\theta}_{\text{init},t}$, $\mathbf{c}^{\theta}_{1|t}$
    if $\mathrm{d}t_{\mathbf{x}} > 0$ then
        Update $\mathbf{x}$ with Eq. (1)
    end if
    if $\mathrm{d}t_{\mathbf{r}} > 0$ then
        Unmask $\mathbf{r}$ with prob. $\mathrm{d}t_{\mathbf{r}} \cdot \frac{1+\eta t_{\mathbf{r}}}{1-t_{\mathbf{r}}}$
        Remask $\mathbf{r}$ with prob. $\mathrm{d}t_{\mathbf{r}} \cdot \eta$
    end if
    if $\mathrm{d}t_{\mathbf{z}} > 0$ then
        for each residue $j$ do
            If unmasked: update $\mathbf{z}_j$ with Eq. (2)
            If newly unmasked: set $\mathbf{z}_j \leftarrow \mathbf{z}^{\theta}_{\text{init},t}$
            If masked: set $\mathbf{z}_j \leftarrow 0$
        end for
    end if
end for
```

in 3 updates in total and ~ 4M parameters in triangle multiplicative layers. Table 9 demonstrates that *Proteína-Atomística-tri* exhibits improved performance on all metrics, especially the all-atom diversity. Considering that the triangle multiplicative layers are highly memory-intensive, we keep them as an optional and sparse add-on to our model architecture.

D Proteína-Atomística Training and Inference Details

D.1 Proteína-Atomística Training

The training process is outlined in Alg. 1. We start by sampling time steps to create noisy inputs for each modality (Sec. 4.1) and feeding them into the model. Both C_α and non-C_α sample time from the mixed uniform-beta distribution from Proteína [16] and the sequence time is sampled from $\mathcal{U}(0, 1)$. The training objectives are as follows: for C_α atoms, we use the standard conditional flow matching objective, while for amino acid sequences, we use a standard cross-entropy loss. For non-C_α atoms (i) for unmasked residues, we apply the flow matching loss to existing atoms, similar to C_α atoms; (ii) for masked residues, we regress the predicted pseudo-velocity $\mathbf{z}^{\theta}_{\text{init},t}$ towards an augmented objective as discussed in Sec. 4.2. See Appendix for further training details.

D.2 Proteína-Atomística Sampling

We sample C_α atoms by simulating the learned flow via an SDE. Since our flow is Gaussian, it relates to the score function as: $\mathbf{s}^{\theta}_{\mathbf{x},t} = (t\mathbf{v}^{\theta}_{\mathbf{x},t} - \mathbf{x}_t)/(1-t)$. This allows us to define an SDE for sampling

$$\mathrm{d}\mathbf{x}_t = \mathbf{v}^{\theta}_{\mathbf{x},t}\,\mathrm{d}t + g_{\mathbf{x}}(t)\mathbf{s}^{\theta}_{\mathbf{x},t}\,\mathrm{d}t + \sqrt{2g_{\mathbf{x}}(t)\gamma_{\mathbf{x}}}\,\mathrm{d}\mathcal{W}_t, \tag{1}$$

with noise scale $\gamma_{\mathbf{x}}$ and Wiener process $\mathcal{W}_t$. Setting $\gamma_{\mathbf{x}}$=1 produces the model's marginal distribution, while reducing $\gamma_{\mathbf{x}}$ can boost designability by lowering noise during generation, at the cost of diversity.

Following MultiFlow [7], for sequence sampling, we effectively perform iterative unmasking and remasking. Starting with a fully masked sequence $[\text{M}]^L$, at each timestep t, we predict residue type logits $\mathbf{c}^{\theta}_{1|t}$ and sharpen the distribution using a temperature τ to obtain probabilities $p_{1|r}(\mathbf{r}) = \text{softmax}(\mathbf{c}^{\theta}_{1|t}/\tau)$. Each masked residue is then unmasked with probability $\mathrm{d}t \cdot (1 + \eta t)/(1 - t)$, where η controls sampling stochasticity, and its type is sampled from $p_{1|r}(\mathbf{r})$. To maintain balance, each unmasked residue is subsequently remasked with probability $\mathrm{d}t \cdot \eta$. We also explore recent advances in discrete diffusion sampling algorithms [40, 36].

The generation of non-C_α atoms depends on the sequence generation process. Following the flow matching framework in Sec. 4.1, we begin generating non-C_α atoms for a residue only after it is unmasked. Accordingly, the generation process falls into three cases: (1) If the residue is already unmasked, we update its non-C_α coordinates using the same SDE as for C_α atoms:

$$\mathrm{d}\mathbf{z}_t = \mathbf{v}^{\theta}_{\mathbf{z},t}\,\mathrm{d}t + g_{\mathbf{z}}(t)\mathbf{s}^{\theta}_{\mathbf{z},t}\,\mathrm{d}t + \sqrt{2g_{\mathbf{z}}(t)\gamma_{\mathbf{z}}}\,\mathrm{d}\mathcal{W}_t; \tag{2}$$

(2) if the residue is newly unmasked at the current step, we initialize its atom coordinates using a single step Euler integration using the the predicted initialization $\mathbf{z}^{\theta}_{\text{init},t}$; (3) if the residue is still masked or has been remasked, we set $\mathbf{z}$ to zero. This framework enables the concurrent generation of side chains alongside backbones and sequences, contrasting with methods that generate side chains after backbone and sequence generation. This simultaneous approach allows for an increased influence of side chains on the local structure while retaining the flexibility to alter sequence identities.

The sampling process is detailed in Alg. 2. As we use distinct time schedules for the three modalities, we denote their respective timesteps with corresponding subscripts and use N to denote the number of timesteps. Our flexible and general framework, in principle, allows for sampling modalities in any order by adjusting these time schedules.

D.3 Defining Noise Schedules via the Time Distribution

Training Time Sampling Distribution. A key design choice in diffusion and flow matching models is the time sampling distribution $p(t)$, which effectively controls how the training objective is weighted across different stages of the generative process. Here, since we consider three distinct modalities, we sample time steps independently for each. Proteína-Atomística proposes to bias sampling toward later timesteps ($t \approx 1$) to encourage the model to allocate more capacity to generating fine-grained local structure. Specifically, for flow matching in Euclidean space—i.e., for $\mathbf{x}$ and $\mathbf{z}$—we use a mixed Beta distribution [16] for $t_{\mathbf{x}}$ and $t_{\mathbf{z}}$.

$$p(t) = 0.02\,\mathcal{U}(0,1) + 0.98\,\mathcal{B}(1.9, 1.0),$$

where $\mathcal{B}(\cdot,\cdot)$ is the Beta distribution. For the discrete modality $\mathbf{r}$, we sample $t_{\mathbf{r}}$ from $\mathcal{U}(0,1)$. Additionally, following [7], we give the options to allocate a small percentage of each training batch to forward folding ($t_{\mathbf{r}} = 1$), inverse folding ($t_{\mathbf{x}} = 1$) and also extend to side chain packing ($t_{\mathbf{x}} = 1$ and $t_{\mathbf{r}} = 1$). Please see Table 4 for specific ratios for each model configuration.

D.4 Side Chain Initialization

In Fig. 7, the initialization $\mathbf{z}^{\theta}_{\text{init},t}$ is predicted from atom representations that are also used for the vector field $\mathbf{v}^{\theta}_{\mathbf{z},t}$. Notably, the model does not have access to $\mathbf{z}_t$ for masked residues due to the structures being undefined for unknown residue types. This differs from the standard flow matching objective, which predicts a vector field conditioned on the noisy input. We have found that separating the initialization from the standard structure-to-structure vector field works best in practice (Table 11).

In Table 11 we empirically observed that directly predicting clean coordinates is challenging due to their high variance and our model's non-equivariant nature. To address this, we introduce an auxiliary objective that predicts $\mathbf{z} - \boldsymbol{\epsilon}_{\mathbf{z}}$, where $\boldsymbol{\epsilon}_{\mathbf{z}}$ is standard Gaussian noise not visible to the model. This formulation is effective for two reasons: (1) it aligns with the vector field objective for existing atoms, and (2) since $\boldsymbol{\epsilon}_{\mathbf{z}}$ is not known to the model, the optimal prediction converges to $\mathbb{E}[\mathbf{z} - \boldsymbol{\epsilon}_{\mathbf{z}}] = \mathbb{E}[\mathbf{z}] - \mathbb{E}[\boldsymbol{\epsilon}_{\mathbf{z}}] = \mathbb{E}[\mathbf{z}]$, ensuring that the prediction converges to the average clean coordinates. This, in turn, properly initializes newly unmasked side chain atoms.

An alternative interpretation of this augmented objective is that we aim to learn an augmented vector field that transforms a random starting point with average $\mathbb{E}[\boldsymbol{\epsilon}_{\mathbf{z}}] = 0$ to the average clean data $\mathbb{E}[\mathbf{z}]$. During generation, we can then obtain an initialization by performing a single-step Euler integration from noise ($t_{\text{pre-init}} = 0$) towards "clean data" ($t_{\text{init}} = 1$) using the learned vector field. We assume the side-chain structures for masked residues originate from zero. This conceptually means the side-chain coordinates are initially hidden behind the C_α atoms (in local coordinates) before being unmasked.

D.5 Two Stage Training

We used a training + finetuning strategy to train *Proteína-Atomística* on $\mathcal{D}_{\text{SYN}-\text{ours}}$. The model was first trained on a subset of $\mathcal{D}_{\text{SYN}-\text{ours}}$ containing proteins with lengths ranging from 32 to 256. The model is then finetuned on the full $\mathcal{D}_{\text{SYN}-\text{ours}}$ with protein lengths ranging from 32 to 400. The model with triangle multiplicative layers (*Proteína-Atomística-tri*) was only trained on protein lengths ranging from 32 to 256. We recorded the number of steps and learning rate in both training and finetuning stages for each variant of the model in Table 4.

D.6 Details in Multimodal Flow Matching

We present the detailed version of our training algorithm in Alg. 3. Here, SampleTimestep() is the function to sample timesteps for each modality based on the training time distributions in Sec. D.3 and Global2Local() is the function to transform global coordinates to local coordinates, where the transformation scheme (local translations or local frames) is chosen as a hyperparameter.

Global2Local: Non-C_α atoms are structurally organized around their corresponding C_α atoms. We offer two local coordinate modeling strategies to leverage this property, simplifying the learning task by predicting offsets rather than global coordinates and facilitating better initialization of non-C_α atoms. The first approach calculates the relative position of non-C_α atoms directly with respect to their corresponding C_α atom: $\mathbf{z}_i^{\text{local}} = \mathbf{z}_i - \mathbf{x}_i$. The second strategy, inspired by related work [28], constructs a residue-centric local coordinate frame $(\mathbf{t}_i, \mathbf{R}_i)$ using the C_α coordinates of three neighboring residues ($\mathbf{x}_{i-1}$, $\mathbf{x}_i$, $\mathbf{x}_{i+1}$) via the Gram-Schmidt process. Non-C_α coordinates $\mathbf{z}_i$ are then transformed to local ones via $\mathbf{z}_i^{\text{local\_frame}} = \mathbf{R}_i^{-1}(\mathbf{z}_i - \mathbf{t}_i)$. In the following sections, models denoted by *local trans* employ the local translation parameterization, while those denoted by *local frame* utilize the frame-based parameterization.

Algorithm 3 Proteína-Atomística Training

```
**Input:** $C_\alpha$ atom $\mathbf{x} \in \mathbb{R}^{L\times 3}$, amino-acid sequence $\mathbf{r} \in \{0, ..., 19\}^L$, non-$C_\alpha$ atom $\mathbf{z} \in \mathbb{R}^{L\times 36\times 3}$

**while** not converged **do**
    # **Step 1: Noising Process**
    $t_{\mathbf{x}}, t_{\mathbf{r}}, t_{\mathbf{z}} \leftarrow$ SampleTimestep()
    $\mathbf{r}_t \sim t_{\mathbf{r}}\delta\{\mathbf{r}\} + 1 - t_{\mathbf{r}}\delta(\mathrm{M})$
    $\boldsymbol{\epsilon}_{\mathbf{x}} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}) \in \mathbb{R}^{L\times 3}, \boldsymbol{\epsilon}_{\mathbf{z}} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}) \in \mathbb{R}^{L\times 36\times 3}$
    $\mathbf{x}_t \leftarrow t_{\mathbf{x}}\mathbf{x} + (1 - t_{\mathbf{x}})\boldsymbol{\epsilon}_{\mathbf{x}}$
    $\mathbf{z} \leftarrow \text{Global2Local}(\mathbf{z}, \mathbf{x})$        # if using local coordinates
    $\mathbf{z}_t \leftarrow t_{\mathbf{z}}\mathbf{z} + (1 - t_{\mathbf{z}})\boldsymbol{\epsilon}_{\mathbf{z}}$
    Zero out non-existing atoms in $\mathbf{z}_t$ based on $\mathbf{r}_t$

    # **Step 2: Neural Network**
    $\mathbf{v}^\theta_{\mathbf{x},t}, \mathbf{v}^\theta_{\mathbf{z},t}, \mathbf{z}^\theta_{\text{init},t}, \mathbf{c}^\theta_{1|t} \leftarrow \text{Transformer}(\mathbf{x}_t, \mathbf{r}_t, \mathbf{z}_t, \emptyset, \emptyset, \emptyset, t_{\mathbf{x}}, t_{\mathbf{r}}, t_{\mathbf{z}})$
    **if** rand(0, 1) $> 0.5$ **then**
        $\bar{\mathbf{r}} \leftarrow \arg\max \mathbf{c}^\theta_{1|t}$
        $\bar{\mathbf{x}} \leftarrow \mathbf{x}_t + (1 - t_{\mathbf{x}})\mathbf{v}^\theta_{\mathbf{x},t}$
        $\bar{\mathbf{z}} \leftarrow \mathbf{z}_t + (1 - t_{\mathbf{z}})\mathbf{v}^\theta_{\mathbf{z},t}$
        $\mathbf{v}^\theta_{\mathbf{x},t}, \mathbf{v}^\theta_{\mathbf{z},t}, \mathbf{z}^\theta_{\text{init},t}, \mathbf{c}^\theta_{1|t} \leftarrow \text{Transformer}(\mathbf{x}_t, \mathbf{r}_t, \mathbf{z}_t, \text{sg}(\bar{\mathbf{x}}), \text{sg}(\bar{\mathbf{r}}), \text{sg}(\bar{\mathbf{z}}), t_{\mathbf{x}}, t_{\mathbf{r}}, t_{\mathbf{z}})$
    **end if**

    # **Step 3: Loss Calculation**
    $\mathcal{L}_{\mathbf{r}} \leftarrow \text{CrossEntropy}(\mathbf{c}^\theta_{1|t}, \mathbf{r})$
    $\mathcal{L}_{\mathbf{x}} \leftarrow \frac{1}{L}||\mathbf{v}^\theta_{\mathbf{x},t} - (\mathbf{x} - \boldsymbol{\epsilon}_{\mathbf{x}})||_2^2$
    **for** each residue $i$ **do**
        $\mathcal{L}_{\mathbf{z},i} \leftarrow \|\mathbf{v}^\theta_{\mathbf{z},t,i} - (\mathbf{z}_i - \boldsymbol{\epsilon}_{\mathbf{z},i})\|_2^2$, if $\mathbf{r}_{t,i} \neq \mathrm{M}$
        $\mathcal{L}_{\mathbf{z},i} \leftarrow \|\mathbf{z}^\theta_{\text{init},t,i} - (\mathbf{z}_i - \boldsymbol{\epsilon}_{\mathbf{z},i})\|_2^2$, if $\mathbf{r}_{t,i} = \mathrm{M}$
    **end for**
    $\mathcal{L} \leftarrow \frac{1}{L}\left(\mathcal{L}_{\mathbf{x}} + \mathcal{L}_{\mathbf{r}} + \mathcal{L}_{\mathbf{z}}\right)$

    Calculate gradient and update model parameters
**end while**
```

E Inference Details and Hyperparameters

We present a detailed version of the sampling algorithm in Alg. 4.

Algorithm 4 Proteína-Atomística Multimodal Sampling

```
**Input:** discretized timesteps for three modalities $\{t_{\mathbf{x},i}\}_{0..N}$, $\{t_{\mathbf{r},i}\}_{0..N}$, and $\{t_{\mathbf{z},i}\}_{0..N}$, stochasticity schedules $g_{\mathbf{x}}(t)$ and $g_{\mathbf{z}}(t)$, noise scales $\gamma_{\mathbf{x}}, \gamma_{\mathbf{z}}, \eta$, sequence temperature $\tau$
**Output:** generated proteins $(\mathbf{x}, \mathbf{r}, \mathbf{z})$
$\mathbf{x} \sim \boldsymbol{\mathcal{N}}(\mathbf{0}, \boldsymbol{I}) \in \mathbb{R}^{L\times 3}$
$\mathbf{z} \sim \boldsymbol{\mathcal{N}}(\mathbf{0}, \boldsymbol{I}) \in \mathbb{R}^{L\times 36\times 3}$
$\mathbf{r} \leftarrow [\mathrm{M}]^L$
**for** $i = 0$ **to** $N - 1$ **do**
    $\mathbf{v}^\theta_{\mathbf{x},t}, \mathbf{v}^\theta_{\mathbf{z},t}, \mathbf{z}^\theta_{\text{init},t}, \mathbf{c}^\theta_{1|t} \leftarrow \text{Transformer}(\mathbf{x}, \mathbf{r}, \mathbf{z}, \emptyset, \emptyset, \emptyset, t_{\mathbf{x},i}, t_{\mathbf{r},i}, t_{\mathbf{z},i})$
    **if** self-condition **then**
        $\bar{\mathbf{r}} \leftarrow \arg\max \mathbf{c}^\theta_{1|t}$
        $\bar{\mathbf{x}} \leftarrow \mathbf{x}_t + (1 - t_{\mathbf{x},i})\mathbf{v}^\theta_{\mathbf{x},t}$
        $\bar{\mathbf{z}} \leftarrow \mathbf{z}_t + (1 - t_{\mathbf{z},i})\mathbf{v}^\theta_{\mathbf{z},t}$
        $\mathbf{v}^\theta_{\mathbf{x},t}, \mathbf{v}^\theta_{\mathbf{z},t}, \mathbf{z}^\theta_{\text{init},t}, \mathbf{c}^\theta_{1|t} \leftarrow \text{Transformer}(\mathbf{x}_t, \mathbf{r}_t, \mathbf{z}_t, \bar{\mathbf{x}}, \bar{\mathbf{r}}, \bar{\mathbf{z}}, t_{\mathbf{x},i}, t_{\mathbf{r},i}, t_{\mathbf{z},i})$
    **end if**

    # Update CA Atoms
    $\mathrm{d}t_{\mathbf{x}} = t_{\mathbf{x},i+1} - t_{\mathbf{x},i}$
    **if** $\mathrm{d}t_{\mathbf{x}} > 0$ **then**
        $\hat{\mathbf{x}} \leftarrow \mathbf{x} + \mathbf{v}^\theta_{\mathbf{x},t}\mathrm{d}t_{\mathbf{x}} + g_{\mathbf{x}}(t_{\mathbf{x},i})\mathbf{s}^\theta_{\mathbf{x},t}\mathrm{d}t_{\mathbf{x}} + \sqrt{2g_{\mathbf{x}}(t_{\mathbf{x},i})\gamma_{\mathbf{x}}\,\mathrm{d}\mathcal{W}_t}$
    **end if**

    # Update Amino-Acid Sequence
    $\mathrm{d}t_{\mathbf{r}} = t_{\mathbf{r},i+1} - t_{\mathbf{r},i}$
    **if** $\mathrm{d}t_{\mathbf{r}} > 0$ **then**
        **if** sampling_alg = PURITY **then**
            $\hat{\mathbf{r}} \leftarrow$ purity_sample($\mathbf{c}^\theta_{1|t}, \mathrm{d}t_{\mathbf{r}}, \eta, \tau, \hat{\mathbf{r}}$) (Algorithm 5)
        **else if** sampling_alg = P2 **then**
            $\hat{\mathbf{r}} \leftarrow$ p2_sample($\mathbf{c}^\theta_{1|t}, \mathrm{d}t_{\mathbf{r}}, \eta, \tau, \hat{\mathbf{r}}$) (Algorithm 6)
        **else**
            $\hat{\mathbf{r}}_1 \sim \text{Softmax}(\mathbf{c}^\theta_{1|t}/\tau)$
            $p_{\text{unmask}} \leftarrow \mathrm{d}t_{\mathbf{r}} \cdot (1 + \eta t_{\mathbf{r},i})/(1 - t_{\mathbf{r},i})$
            $p_{\text{remask}} \leftarrow \mathrm{d}t_{\mathbf{r}} \cdot \eta$
            **for** $j = 1$ **to** $L$ **do**
                **if** $\mathbf{r}_j = \mathrm{M}$ **then**
                    $\hat{\mathbf{r}}_j \sim (1 - p_{\text{unmask}})\delta\{\mathrm{M}\} + p_{\text{unmask}}\delta\{\hat{\mathbf{r}}_{1,j}\}$
                **else**
                    $\hat{\mathbf{r}}_j \sim (1 - p_{\text{remask}})\delta\{\mathbf{r}_j\} + p_{\text{remask}}\delta\{\mathrm{M}\}$
                **end if**
            **end for**
        **end if**
    **end if**

    # Update Non-CA Atoms
    $\mathrm{d}t_{\mathbf{z}} = t_{\mathbf{z},i+1} - t_{\mathbf{z},i}$
    **if** $\mathrm{d}t_{\mathbf{z}} > 0$ **then**
        **for** $j = 1$ **to** $L$ **do**
            **if** $\mathbf{r}_j \neq \mathrm{M}$ and $\hat{\mathbf{r}}_j \neq \mathrm{M}$ **then**
                $\hat{\mathbf{z}}_j \leftarrow \mathbf{z}_j + \mathbf{v}^\theta_{\mathbf{z},t,j}\mathrm{d}t_{\mathbf{z}} + g_{\mathbf{z}}(t_{\mathbf{z},i})\mathbf{s}^\theta_{\mathbf{z},t,j}\mathrm{d}t_{\mathbf{z}} + \sqrt{2g_{\mathbf{z}}(t_{\mathbf{z},i})\gamma_{\mathbf{z}}\,\mathrm{d}\mathcal{W}_t}$
            **else if** $\mathbf{r}_j = \mathrm{M}$ and $\hat{\mathbf{r}}_j \neq \mathrm{M}$ **then**
                $\mathbf{z}_j \leftarrow \mathbf{z}^\theta_{\text{init},t,j}$
            **else**
                $\mathbf{z}_j = 0$
            **end if**
        **end for**
    **end if**
    $\mathbf{x}, \mathbf{r}, \mathbf{z} \leftarrow \hat{\mathbf{x}}, \hat{\mathbf{r}}, \hat{\mathbf{z}}$
**end for**
**Return** $(\mathbf{x}, \mathbf{r}, \mathbf{z})$
```

E.1 Inference Time Schedules

We sample from Proteína-Atomística following Alg. 2 for the C_α coordinates, residue types, and non-C_α backbone and side chain atoms, integrating from $t = 0$ to $t = 1$. For the coordinates of C_α ($\mathbf{x}$) and non-C_α atoms ($\mathbf{z}$), we simulate the SDE (Eq. 1 and Eq. 2) with the following definition for $g(t)$:

$$\begin{cases} g(t) = 1/(t + 0.01), & t \in [0, 0.99] \\ g(t) = 0, & t \in (0.99, 1) \end{cases}$$

We use $N_\mathbf{x} = 500$ and $N_\mathbf{z} = 600$ steps to discretize the unit interval into logarithmically spaced points. The PyTorch [35] code snippet to generate the logarithmic discretization is as follows:

```
t = 1.0 - torch.logspace(-2, 0, nsteps + 1).flip(0)
t = t - torch.min(t)
t = t / torch.max(t)
```

which ensures that $t \in [0, 1]$. For the sampling of residue types ($\mathbf{r}$), we use the $N_\mathbf{r} = 500$ steps to discretize the unit interval into quadratically spaced points. $\mathbf{x}_t$ and $\mathbf{r}_t$ are padded with ones for extra 100 steps to match the total number of 600 steps of the simulation. Fig. 9 visualizes the discretized t-schedule of different modalities during sampling.

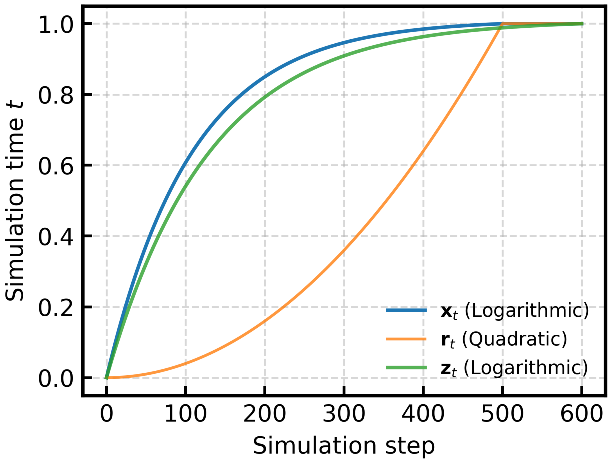

Figure 9: **Discretized t schedule during sampling** for backbone C_α ($\mathbf{x}_t$), amino acid residue types ($\mathbf{r}_t$), and all non-C_α atoms ($\mathbf{z}_t$). The last 100 steps of $\mathbf{x}_t$ and $\mathbf{r}_t$ are padded with 1.

E.2 Backbone-Sequence Co-Design

For both the backbone-sequence co-design and all-atom generation tasks, our models perform best with a combination of the offset schedules defined above and various low-temperature settings across the explicit data modalities that the model has access to (backbone C_α, residue types, other backbone and side chain atoms). Here we detail the specific parameters and how they impact the generated results.

E.2.1 Low Temperature Sampling for Backbone C_α

The stochasticity of backbone sampling is handled in the same way as done in Proteína [16]. This means we have a single noise scale to decrease the impact of the noise in relation to the score and vector field contributions. This can be seen in Eq. 1 where $\gamma_\mathbf{x}$ refers to the backbone noise scale shown in Table 4.

E.2.2 Discrete Diffusion Sampling for Residue Types

As detailed in Algorithm 4, we investigate two discrete diffusion sampling algorithms to refine the incorporation of stochasticity in the masking and unmasking processes. We adopt purity sampling [40] (Algorithm 5), which prioritizes unmasking tokens with high confidence, and self-path planning (P2) sampling [36] (Algorithm 6), which instead encourages remasking tokens with low confidence.

Algorithm 5 Purity Sampling

```
Input: predicted logits $\mathbf{c}^{\theta}_{1|t}$, time step $t$, time delta d$t$, sampling temperature $\tau$, stochasticity $\eta$, current sequence $\mathbf{r}_t$
Output: updated sequence $\hat{\mathbf{r}}$
probs $\leftarrow$ Softmax$(\mathbf{c}^{\theta}_{1|t}/\tau)$
MaxLogProb $\leftarrow \max(\log(\text{probs}), \text{dim} = -1)$
MaxLogProb $\leftarrow$ MaxLogProb $- (\mathbf{r}_t \neq \text{M}) \cdot \infty$
SortedIndices $\leftarrow$ ArgSort(MaxLogProb, descending $=$ True)
$p_{\text{unmask}} \leftarrow \min(1, \text{d}t \cdot \frac{1+\eta t}{1-t})$
ToUnMask $\leftarrow$ (Uniform$(0,1) \leq p_{\text{unmask}}) \wedge (\mathbf{r}_t = \text{M})$
NumToUnmask $\leftarrow \sum$(ToUnMask)
$\hat{\mathbf{r}}_{\text{samples}} \sim$ Categorical(probs)
$\hat{\mathbf{r}} \leftarrow \mathbf{r}_t$
for $i = 1$ to NumToUnmask do
    idx $\leftarrow$ SortedIndices$[i]$
    $\hat{\mathbf{r}}_{\text{idx}} \leftarrow \hat{\mathbf{r}}_{\text{samples,idx}}$
end for
$p_{\text{remask}} \leftarrow \text{d}t \cdot \eta$
ToReMask $\leftarrow$ (Uniform$(0,1) < p_{\text{remask}}) \wedge (t + \text{d}t < 1)$
$\hat{\mathbf{r}} \leftarrow \hat{\mathbf{r}} \cdot (1 -$ ToReMask$) + \text{M} \cdot$ ToReMask
return $\hat{\mathbf{r}}$
```

Algorithm 6 P2 Sampling

```
Input: predicted logits $\mathbf{c}^{\theta}_{1|t}$, sampling temperature $\tau$, stochasticity $\eta$, current sequence $\mathbf{r}_t$
Ouput: updated sequence $\hat{\mathbf{r}}$
$\kappa(t) = 1 - t$
$\epsilon \sim$ Gumbel$(0,1)$
logprob, $\hat{\mathbf{r}_1}$ = LogSoftmax$(\mathbf{c}^{\theta}_{1|t}/\tau + \epsilon)$.max(dim=-1)
score $\leftarrow$ logprob
score$[\mathbf{r}_t \neq M] \leftarrow$ score$[\mathbf{r}_t \neq M] * \eta$
ToMask $\leftarrow$ Top-K-Lowest$_{\kappa(t)}$(score)
$\hat{\mathbf{r}} \leftarrow \hat{\mathbf{r}}_1$
for $j \in$ ToMask do
    if $[\mathbf{r}]_j \neq$ M then
        $[\mathbf{r}]_j \leftarrow$ M
    end if
end for
for $j \notin$ ToMask do
    if $[\mathbf{r}]_j =$ M then
        $[\mathbf{r}]_j \leftarrow [\hat{\mathbf{r}}_1]_j$
    end if
end for
```

In total, three hyperparameters define the discrete flow matching sampling: (1) the sampling algorithm type, either purity or P2; (2) the temperature τ, which directly controls the predicted logits, with $\tau < 1$ emphasizing likely residue types and $\tau > 1$ acting as a distribution smoothing parameter; and (3) the stochasticity (η), which governs the remasking ratio.

We test both purity and P2 sampling for all models and select the optimal settings based on the empirical results.

E.3 Stochastic Side Chain and Non-C_α Sampling

Once a residue is unmasked, side chains and non-C_α atoms are updated with the same stochastic SDE as used for backbone C_α generation. This can be seen in Eq. 2 where $\gamma_{\mathbf{z}}$ refers to the side chain and non-C_α noise scale shown in Table 4. When the residue is masked, we do not update the side chain structure as there is not yet a residue to predict the structure of. In the event of a residue being unmasked we follow the steps described in Sec. D.4

Our flow matching design is tailored to accommodate dynamic changes in residue types during generation, enabling flexible atom sequence lengths, *i.e.* a variable number of side chain atoms as a function of time. For example, consider a single residue during generation where $0 < t_\alpha < t_\beta < t_\omega < 1$. At t_α, the current residue is alanine with associated structures $\mathbf{x}_{t_\alpha}, \mathbf{z}_{t_\alpha}$. At t_β, the alanine is replaced with a mask token, and the side chain and non-C_α backbone structure $\mathbf{z}_{t_\alpha}$ is zeroed out since the structure of an unknown residue is undefined. Then, at t_ω, the residue token is unmasked to lysine, requiring the initialization of the side chain structure with a different number of atoms than previously used at t_α.

F Ablation Studies

Table 4: Inference Parameters for Proteína-Atomística, Proteína-Co-design, and La-Proteína for backbone C_α, sequence, and side chain/non-C_α modalities. For La-Proteína we report the local latent temperature as side chain noise scale.

				Backbone		Sequence		Side Chain
Model	Where	Train Length	Local Coord	Noise Scale	Algo.	Temperature	Noise Scale	Noise Scale
Proteína-Atomística $_{\text{div}}$	Table 7, 8	256	trans	0.62	P2	0.20	5.0	0.45
Proteína-Atomística $_{\text{codes}}$	Table 7, 8	256	trans	0.20	P2	0.45	5.0	0.60
Proteína-Atomística $_{\text{div}}$	Table 7, 8	256	frame	0.60	Purity	0.20	0	0.60
Proteína-Atomística $_{\text{codes}}$	Table 7, 8	256	frame	0.30	Purity	0.20	0	0.45
Proteína-Atomística $_{\text{div}}$	Table 9	400	frame	0.60	Purity	0.45	0	0.1
Proteína-Atomística $_{\text{opt}}$	Table 9	400	frame	0.45	Purity	0.30	0	0.1
Proteína-Atomística $_{\text{codes}}$	Table 9	400	frame	0.35	Purity	0.30	0	0.1
Proteína-Atomística-tri$_{\text{opt}}$	Table 9	256	frame	0.5	Purity	0.45	0	0.3
Proteína-Co-design $_{\text{div}}$	Table 7	256	N/A	0.60	Purity	0.20	5.0	N/A
Proteína-Co-design $_{\text{codes}}$	Table 7	256	N/A	0.30	Purity	0.20	5.0	N/A
La-Proteína $_{\text{div}}$	Table 7, 8	512	N/A	0.30	N/A	N/A	N/A	0.1
La-Proteína $_{\text{codes}}$	Table 7, 8	512	N/A	0.10	N/A	N/A	N/A	0.1

Model Specifications. Table 4 details the specific hyperparameters for the Proteína-Atomística and *Proteína-Co-design* variants used in Table 1, Fig. 10, Table 7, and Table 8. All models follow a similar paradigm: the lower the noise scale, the lower the diversity and the higher the designability. The choices between the various discrete flow matching parameters were based on settings that yielded optimal results, focusing on a strong trade-off between diversity and designability. Please see the end of Sec. 4.1 for precise definitions of (i) local translational, and (ii) local frame-based coordinates.

Ablation Strategy. Since Proteína-Atomística generates backbone C_α atoms, amino acid residues, and atomistic side chains with non-C_α atoms simultaneously, our ablation study follows a hierarchical structure, incrementally integrating data modalities, starting from the simplest representation: (1) C_α only (Sec. F.1), (2) backbone-sequence co-design (Sec. F.2), and (3) fully atomistic proteins (Sec. F.3).

F.1 Backbone C_α Only Design

We start with comparing architectures designed for backbone C_α generation, and analyzing the role synthetic sequences play on structure-based benchmarks. Table 5 presents the primary subset of de novo backbone design benchmarks for recent models, following Geffner et al. [16]. We include both M1 (single-shot) and M8 (standard best of 8 ProteinMPNN samples) variants to quantify the impact of ProteinMPNN and its learned sequence distribution on structure-based protein design.

The overall ranking of models in Table 5 changes depending on whether M1 or M8 is used to evaluate designability and diversity. Notably, a large portion of undesignable samples can be made designable by resampling the possible sequence, as shown by the percentage of designable samples, which increases by 10-28% when moving from M1 to M8.

We also introduce Genie2-Flow in Table 5, a variant of the Genie2 model that replaces the backbone diffusion process with the flow matching training and inference procedure of Proteína. Genie2-Flow achieves the best balance between designability and diversity at both the M1 and M8 levels.

Table 5: **Performance of de novo backbone generation for C_α only models.** All models generate 100 proteins for lengths $\in$ [50, 100, 150, 200, 250]. Genie2-Flow uses the Genie2 architecture with the conditional flow matching training of Proteína. Here M8 refers results gain through best of 8 ProteinMPNN sequences. M1 denotes using the first ProteinMPNN sequence.

Method	Dataset	DES-M8 (%) ↑	DES-M1 (%) ↑	DIV-M8 ↑	DIV-M1 ↑	NOV-PDB ↓
FrameFlow	PDB	88.6	61.2	236 (0.53)	160 (0.52)	0.69
RFDiffusion	PDB	94.4	77.8	217 (0.46)	158 (0.34)	0.71
Genie2	AFDB	95.2	74.3	281 (0.59)	233 (0.49)	0.63
FoldFlow-2	PDB	97.4	83.2	239 (0.49)	200 (0.48)	0.68
FoldFlow-2 (reft)	PDB	81.6	53.2	218 (0.53)	131 (0.49)	0.65
Proteína $^{\text{60M no tri}}_{\gamma=0.25}$	Genie2	**98.4**	**87.8**	139 (0.28)	127 (0.29)	0.75
Proteína $^{\text{60M no tri}}_{\gamma=0.45}$	Genie2	95.8	79.2	250 (0.52)	203 (0.51)	0.70
Genie2-Flow$_{\gamma=0.25}$	Genie2	96.6	78.2	**359 (0.74)**	**284 (0.73)**	**0.62**

F.2 Backbone-Sequence Co-Design

Now, we examine the backbone-sequence co-design task to gain a deeper understanding of how sequences influence the generated structures during explicit joint learning.

F.2.1 Extended Discussion of Explicit Co-Design

First, we see that Proteína-Co-design and Proteína-Atomística generate more consistent designable proteins compared to having a separate ProteinMPNN step. This is shown by the $\geq$ DES-M1 column of Table 7, where the sequences generated by our models yield higher designability than a separate ProteinMPNN call. This is important because it demonstrates that we have an accurate model that can operate without the need for always trying to redesign a more fitting sequence (and side chain structure by definition) to the already generated structure, as done in standard multi-stage design pipelines. DES-M8 is always higher than both CODES and DES-M1, signifying that many sequences can fold into similar structures, which we know to be true fundamentally. While our model does not eliminate the potential need for inverse folding-based post-optimization to maximize M8 scores, it achieves high single-shot accuracy with superior side chain structures (Fig. 11), setting a strong foundation for further optimization. Although DES-M8 is higher than M1, finding the best of eight different sequences would require redesigning the side chain structures afterwards. In contrast, Proteína-Atomística generates accurate fully atomistic structures with an aligned sequence in one go.

Second, the success of Proteína-Atomística and Proteína-Co-design is not just due to solving the consistency issues present in using AFDB for fully atomistic training. In Table 6, we see that when we take three prominent model architectures (Proteína, MultiFlow/FrameFlow, Genie2) and train them on the same data $\mathcal{D}_{\mathrm{AFDB-clstr}}$, our Proteína-Co-design outperforms them significantly. Furthermore, we observe that when MultiFlow is trained with its distilled data (comprising PDB and model-generated structures, all with ProteinMPNN sequences), Proteína-Co-design trained on $\mathcal{D}_{\mathrm{AFDB-clstr}}$ achieves competitive performance. Additionally, we find that removing the adversarial

Table 6: **Ablation of popular architectures for codesign on AFDB**. Results for Multiflow base without distillation are taken from their original paper. We trained Multiflow and Genie2-flow-codesign, and evaluated all models by generating 100 proteins for lengths $\in$ [50, 100, 150, 200, 250].

Method	CODES-CA (%) ↑	DIV-CA ↑
MultiFlow (PDB)	42.0	72
MultiFlow (PDB & distilled)	86.7	160
MultiFlow ($\mathcal{D}_{\mathrm{AFDB-clstr}}$)	40.0	52
Genie2-Flow-Co-design ($\mathcal{D}_{\mathrm{AFDB-clstr}}$)	83.0	79
Proteína-Co-design ($\mathcal{D}_{\mathrm{AFDB-clstr}}$)	86.4	153
Proteína-Co-design ($\mathcal{D}_{\mathrm{SYN-ours}}$)	87.0	226

or inconsistent structure-sequence pairs and replacing them with in-silico consistent ones (i.e. training on $\mathcal{D}_{\text{SYN}-\text{ours}}$) increases the accuracy for both co-designability and diversity. As a result, we demonstrate that both architecture and framework as well as data make non-trivial contributions. This result mirrors the behavior observed in Table 5, where instead of relying on noisy best-of-8 ProteinMPNN sampling, here we can learn a diverse and consistent structure-sequence distribution.

F.2.2 What led us to build a more consistent dataset?

We observed that ProteinMPNN-based sequence resampling can significantly improve designability, as evident from the disparity in designability between M1 and M8 in Table 5. Notably, up to 28% of the generated backbones can transition from undesignable to designable simply by resampling the sequence and selecting the best of 8. This suggests that suitable sequences exist for these novel de novo structures, but generating them in a single shot is non-trivial. Moreover, even with ProteinMPNN, the most likely sequence is not guaranteed to be the best, highlighting the need for low-temperature sampling in many of its applications [49, 12].

The observed disparity, combined with the fact that the clustered AFDB is only 19.1% co-designable-all-atom (Fig. 1), led us to investigate the role of ProteinMPNN in enabling consistency in modeling the joint distribution of protein structure and sequence. Given that finding the proper sequence significantly affects sequence-free model performance (Table 5), training on largely non-co-designable data seemed problematic.

We emphasize that simply aligning the structures to known sequences (*i.e.*, training on ESMFold structures) is insufficient and even hurts performance (Fig. 2). To clarify, although we aim to push our models to generate the best designability possible, training on a large amount of diverse and 100% designable structures hurts performance compared to a largely non-designable dataset. To gain a deeper understanding, we investigated the effects of architecture and data on explicitly learning the joint backbone-sequence distribution in the de novo co-design setting (Table 6).

Also see related discussions in Sec. B.

F.2.3 Backbone success does not always translate to multi-modal tasks

Table 6 shows that while Genie2-Flow sets new state-of-the-art results for backbone design, it performs poorly when extended to backbone-sequence co-design. Specifically, Genie2-Flow exhibits a 3.6x diversity drop when comparing ProteinMPNN single-shot (M1) diversity to that of the model-generated sequences (CA). We note that Proteína, Genie2, Genie2-Flow, and Proteína-Co-design were trained on identical datasets, with Proteína-Co-design being identical to the 60M Proteína but with sequence features and discrete flow matching training.

Furthermore, we found that Proteína-Co-design, trained on the unaltered clustered AFDB, matches MultiFlow's performance when trained on PDB and model-generated structures with distilled ProteinMPNN sequences. In contrast, training MultiFlow on the same Genie2 data resulted in co-designability and diversity collapse compared to its distilled form. This highlights the core Proteína transformer's accurate and robust usage for both backbone and backbone-sequence co-design, across natural and synthetic sequence datasets.

F.2.4 Extended Co-Design Results

Table 7 presents the full benchmark performance of the models captured in Fig. 10. Overall, Proteína-Co-design outperforms all prior baselines. Furthermore, how we model the side chains and non-C_α atoms with respect to their central C_α (local vs. frame) greatly impacts the diversity metric. Lastly, by comparing our backbone C_α-sequence co-design model, Proteína-Co-design, to Proteína-Atomística, we observe that significant backbone diversity can be achieved through the incorporation of all-atom modeling (non-C_α backbone atoms and side chains). Here both models are trained on $\mathcal{D}_{\text{SYN}-\text{ours}}$ for fair comparisons.

F.3 Fully Atomistic De Novo Protein Generation

Building on the findings from our backbone C_α-sequence co-design model, Proteína-Co-design, we investigate key aspects of Proteína-Atomística model, including its architecture, stochastic multi-

Table 7: **Backbone-Sequence Co-design performance** compared to baselines. All models generate 100 proteins for lengths ∈ [50, 100, 150, 200, 250]. We report the two multi-modal sampling configurations that generate the (i) most co-designable (codes) and (ii) most diverse samples (div). The best model for co-designability and diversity is emphasized. For parameterization definitions see Table 4. All Proteína-Atomística and Proteína-Co-design are trained with $\mathcal{D}_{\text{SYN-ours}}$. The ≥ DES-M1 column refers to models in which the co-generated sequences offer higher co-designability than ProteinMPNN redesign (1 sample).

Method	**Backbone-Sequence Co-design**			**Backbone-Only Design**			
	≥ DES-M1	CODES (%)↑	DIV-CA↑	DES (%)		DIV	
				M8↑	M1↑	M8↑	M1↑
ProteinGenerator	✗	32.0	48	86.2	73.0	85	82
Protpardelle	✗	65.8	41	95.8	75.0	59	51
PLAID	✗	34.0	79	49.2	36.4	117	81
DPLM-2 (650M, co-generation)	✗	40.6	90	59.0	42.8	133	100
Multiflow	✗	86.7	160	99.6	92.0	191	173
CarbonNovo	✓	76.0	161	89.6	70.4	201	148
P(all-atom)	✗	80.0	263	98.2	95.4	349	299
Proteína-Co-design $_{\text{codes}}$	✓	97.0	156	99.2	97.0	158	155
Proteína-Co-design $_{\text{div}}$	✓	87.0	226	96.2	85.2	256	223
Proteína-Atomística $_{\text{codes,local frame}}$	✓	96.2	162	99.4	93.8	166	162
Proteína-Atomística $_{\text{div,local frame}}$	✓	82.4	230	94.8	81.8	260	227
Proteína-Atomística $_{\text{codes,local trans}}$	✓	97.2	136	99.2	96.0	134	130
Proteína-Atomística $_{\text{div,local trans}}$	✓	84.2	274	96.4	82.4	320	268
Proteína-Atomística-tri $_{\text{local frame}}$	✓	88.6	236	97.0	87.8	257	227
La-Proteína $_{\text{codes},\mathcal{D}_{\text{AFDB-clstr}}}$	✗	88.6	221	99.0	95.0	249	235
La-Proteína $_{\text{div},\mathcal{D}_{\text{AFDB-clstr}}}$	✗	84.6	221	98.6	89.8	259	233
La-Proteína $_{\text{codes},\mathcal{D}_{\text{SYN-ours}}}$	✓	96.8	244	99.6	96.6	249	242
La-Proteína $_{\text{div},\mathcal{D}_{\text{SYN-ours}}}$	✓	93.6	285	99.2	93.2	298	278

Table 8: **All Atom max length 250 performance** compared to baselines. All models generate 100 proteins for lengths ∈ [50, 100, 150, 200, 250]. We report the two multimodal sampling configurations that generate the (i) most all atom codesignable (codes) and (ii) most diverse samples (div). For parameterization definitions see Table 4.

Method	CODES-AA (%)↑	DES-M1 (%)↑	DIV-AA↑	NOV-PDB-AA↓	NOV-AFDB-AA↓
ProteinGenerator	16.0	73.0	24	0.75	0.78
Protpardelle	19.2	75.0	22	0.74	0.77
PLAID	25.4	36.4	56	0.83	0.87
P(all-atom)	76.8	87.2	251	**0.67**	**0.73**
Proteína-Atomística $_{\text{codes,local frame}}$	95.4	94.0	163	0.76	0.81
Proteína-Atomística $_{\text{div,local frame}}$	77.0	81.8	215	0.74	0.80
Proteína-Atomística $_{\text{codes,local trans}}$	96.2	96.0	135	0.78	0.81
Proteína-Atomística $_{\text{div,local trans}}$	81.4	82.4	267	0.73	0.79
Proteína-Atomística-tri $_{\text{local frame}}$	86.4	87.8	235	0.75	0.80
La-Proteína $_{\text{codes},\mathcal{D}_{\text{AFDB-clstr}}}$	84.4	95.0	208	0.80	0.87
La-Proteína $_{\text{div},\mathcal{D}_{\text{AFDB-clstr}}}$	81.0	89.8	213	0.79	0.86
La-Proteína-tri $_{\text{codes},\mathcal{D}_{\text{AFDB-clstr}}}$	89.2	95.0	124	0.81	0.87
La-Proteína-tri $_{\text{div},\mathcal{D}_{\text{AFDB-clstr}}}$	83.6	90.2	176	0.78	0.85
La-Proteína $_{\text{codes},\mathcal{D}_{\text{SYN-ours}}}$	**96.2**	**96.6**	242	0.78	0.85
La-Proteína $_{\text{div},\mathcal{D}_{\text{SYN-ours}}}$	92.2	93.2	**283**	0.78	0.85

modal sampling procedure, and side chain initialization method, and assess their individual impacts on model performance.

F.3.1 Extended Atomistic Benchmarks and Side Chain Representations

Tables 8 and 9 demonstrate the impact of varying noise scale parameters on the trade-off between designability and diversity (*codes* vs. *div* vs. *opt* settings, see Table 4).

Moreover, we find that both local translation and frame-based side chain parameterizations are useful (*local frame* vs. *local trans*), but their relative effectiveness depends on the specific goals of the task. In particular, the local frame is advantageous in high co-designability settings, where it achieves better diversity with comparable co-designability. In contrast, local translation is more effective in high diversity settings, where it yields better co-designability and diversity. See Sec. D.6 for definitions of local translation and frame-based parameterizations.

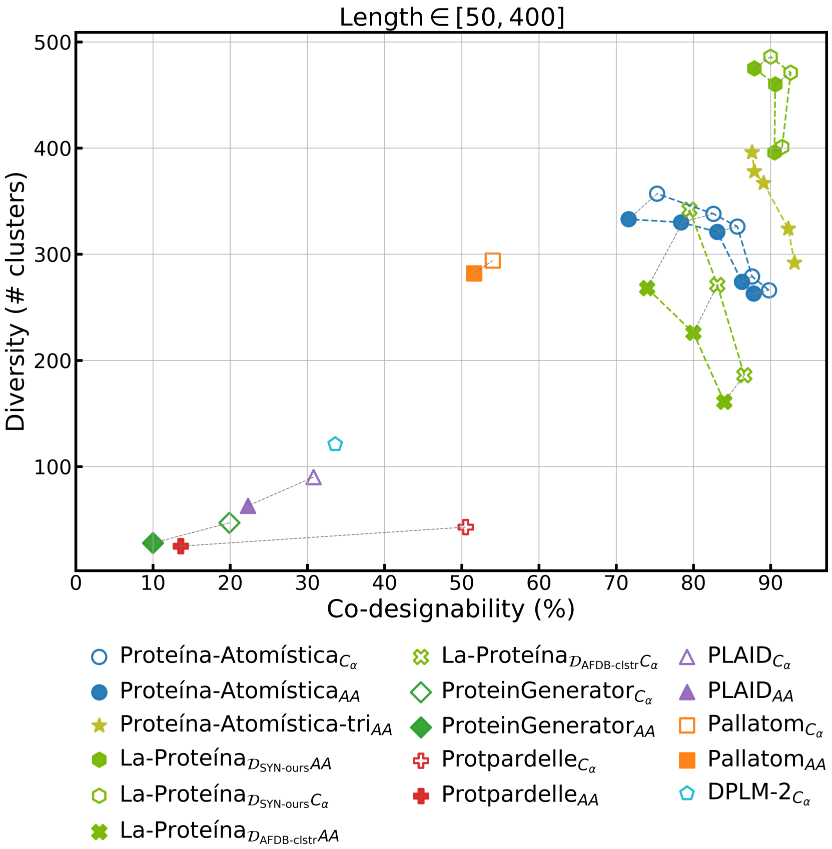

Figure 10: **Pareto frontier of the co-designability-diversity trade-off**. We show metrics of proteins with length $\in [50, 400]$. Solid and hollow markers represent metrics calculated on all-atom and C_α basis, respectively. For atomistic models, the all-atom and C_α scores for the same generated proteins are connected by gray dashed line, and obtained from the same model.

Tables 8 and 9 also illustrate the effect of incorporating triangle updates, which demonstrate improved performance up to a length of 400, despite being trained only up to 256. This is notable, especially when compared to the other Proteína-Atomística variants, which were finetuned to a length of 400. Further details on the triangle update layers can be found in Appendix C.3.

Table 10 further demonstrates the impact of our introduced consistent synthetic data on even longer lengths that the original La-Proteína was trained and evaluated on.

Table 9: **Max length 400 performance of Proteína-Atomística on de novo all atom generation** compared to baselines. All models generate 100 proteins for lengths $\in$ [50, 400] with step size 50. We report the three multimodal sampling configurations that generate the (i) most all-atom co-designable (codes), (ii) most diverse samples (div), and (iii) an optimal trade-off (opt). The best values are bolded. All instances of Proteína-Atomística here use local frames for the side chains. For parameterization definitions see Table 4.

Method	CODES-AA (%)↑	DES-M1 (%)↑	DIV-AA↑	NOV-PDB-AA↓	NOV-AFDB-AA↓
ProteinGenerator	10.0	57.1	28	0.75	0.78
Protpardelle	13.6	62.8	25	0.74	0.76
PLAID	22.3	34.9	63	0.85	0.88
Pallatom	51.6	62.5	282	**0.66**	**0.71**
Proteína-Atomística $_{\text{codes}}$	87.8	88.1	263	0.77	0.81
Proteína-Atomística $_{\text{opt}}$	83.1	85.8	321	0.76	0.80
Proteína-Atomística $_{\text{div}}$	71.6	72.0	333	0.75	0.80
Proteína-Atomística-tri $_{\text{opt}}$	87.6	88.3	396	0.73	0.77
La-Proteína $_{\text{codes},\mathcal{D}_{\text{AFDB-clstr}}}$	76.0	90.1	308	0.77	0.85
La-Proteína $_{\text{div},\mathcal{D}_{\text{AFDB-clstr}}}$	70.6	85.5	314	0.77	0.84
La-Proteína-tri $_{\text{codes},\mathcal{D}_{\text{AFDB-clstr}}}$	84.8	90.1	161	0.81	0.87
La-Proteína-tri $_{\text{div},\mathcal{D}_{\text{AFDB-clstr}}}$	75.0	84.3	268	0.78	0.84
La-Proteína $_{\text{codes},\mathcal{D}_{\text{SYN-ours}}}$	**90.6**	**91.2**	460	0.75	0.83
La-Proteína $_{\text{div},\mathcal{D}_{\text{SYN-ours}}}$	87.9	87.4	**475**	0.74	0.82

Table 10: Impact of consistent synthetic data on La-Proteína. All models generate 100 proteins for lengths $\in$ [100, 500] with step size 100. Baselines taken directly from [15].

Model	CODES-AA (%) ↑	DIV-AA ↑
P(all-atom)	36.7	134
La-Proteína ($\mathcal{D}_{\mathrm{AFDB-clstr}}$)	68.4	206
La-Proteína ($\mathcal{D}_{\mathrm{SYN-ours}}$)	86.8	318

Table 11: **Ablation of the side chain initialization.** All results here use the same model weights and sampling hyperparameters for a Proteína-Atomística model with "local trans" non-C_α coordinates.

Method	CODES-AA (%) ↑	DIV-AA ↑
Gaussian Initialization	56.8	177
Zero Initialization	60.8	196
Learned clean data objective	38.2	76
Learned vector field (default)	81.4	262

F.3.2 Pareto Frontier

We include Fig. 10, an updated pareto frontier to include Proteína-Atomística trained with additional triangle multiplicative updates. Adding 4M worth of triangle multiplicative updates to our 222M Proteína-Atomística further pushes the Pareto frontier. We emphasize that *Proteína-Atomística-tri* is only trained up to length 256 but shows the ability to generalize to longer proteins. Given the increased time and memory costs, we leave further improvements of the Proteína and Proteína-Atomística transformers to future work. These triangle operations are typically seen as required for protein modeling success. In contrast, we are able to take advantage of scaling our data and simpler transformer architectures to yield strong performance.

F.3.3 Atomistic Side Chain Initialization

The side chain structures of proteins generated by Proteína-Atomística are of variable atom sequence length as a function of generation time, because the residue types may change during the generation process (through a series of remasking and unmasking operations). As a result Proteína-Atomística must be able to handle the resetting and regeneration of accurate side chain structures subject to the discrete sampling process of the discrete flow matcher (see Sec. D.4). This dynamic coupling requires careful handling of the initialization point as seen in Table 11, demonstrating the importance of learning a meaningful side chain initialization, instead of using naive zero or random Gaussian initialization. Furthermore, we see that if using a "clean data prediction objective", where we try to predict the side chain structure from the mask token directly rather than using our introduced vector field-like augmentation (c.f. Sec. D.4), the model struggles to generate accurate side chains.

F.4 Atomistic Side Chain Evaluation

To evaluate the generated atomistic protein structures, we compute: (1) MolProbity score [11], (2) clash scores, (3) bond length outliers, and (4) angle outliers, as shown in Fig. 11. MolProbity (MP) score is a composite evaluation metric of macromolecular structures. It measures geometric and stereochemical quality, including steric clashes, backbone dihedral angles, and side-chain conformations. Lower MP score indicates higher structure quality. The clash, bond and angle metrics focus on measuring the physical correctness of the atomistic details of the generated side-chains.

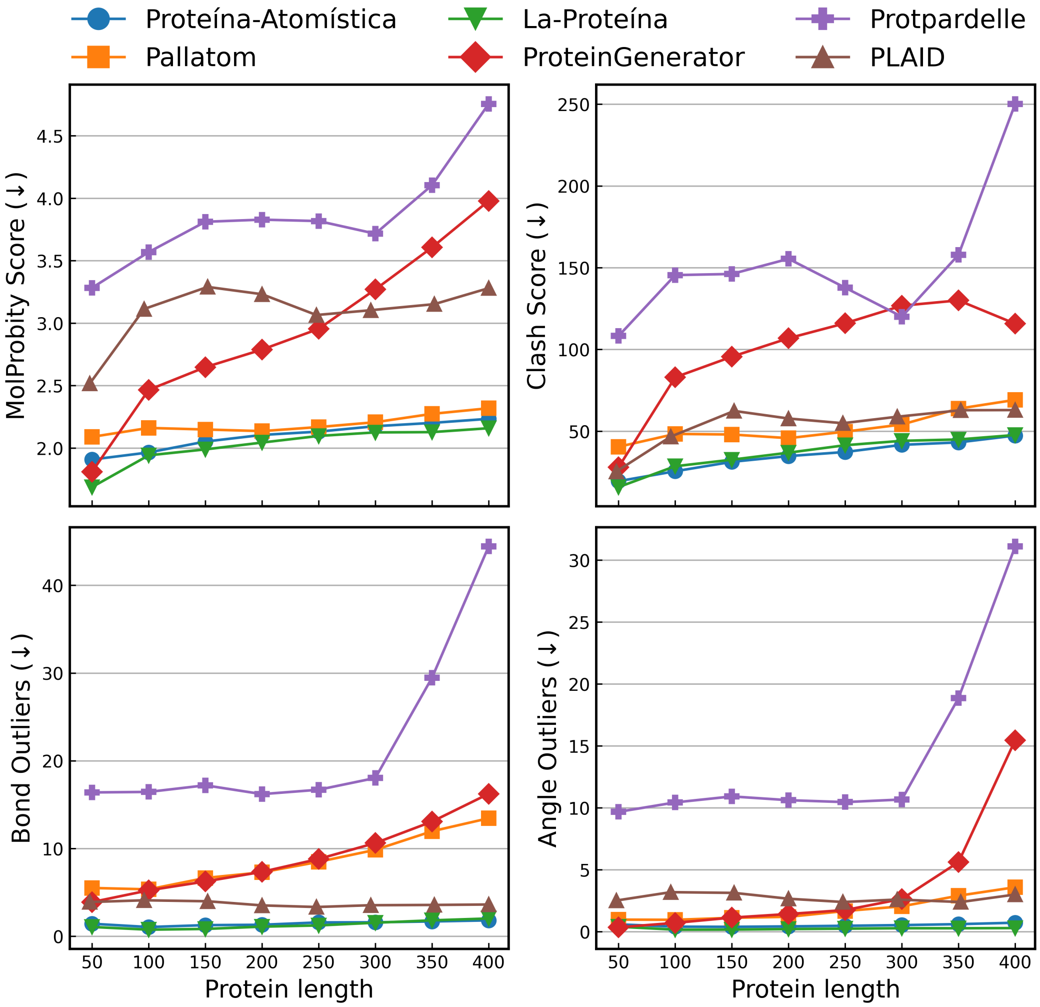

Figure 11: **Side chain structure evaluations.** Lower scores indicate higher side chain quality.

G Metric Definitions and Baselines

G.1 De Novo Design Metrics

To assess the performance of our models, we employ standardized metrics [16, 47, 37] for de novo protein design, adapting them for backbone-sequence co-design and all-atom contexts. The metrics used include:

1. **Designability (DES)**: This measures the ability to inverse fold a generated protein backbone using ProteinMPNN [10] and refold the generated sequences. We report two variants: **DES-M1** (single shot) and **DES-M8** (best of 8 sequences), where DES-M1 evaluates the designability of a single sequence generated by ProteinMPNN, and DES-M8 evaluates the designability of the best sequence out of 8 generated sequences.
2. **Co-designability (CODES)**: Similar to DES-M1, but using the model's output sequence instead of ProteinMPNN-generated sequences. We also report **All-Atom Co-designability (CODES-AA)**, an extension of CODES that uses all-atom scRMSD.
3. **Diversity**: We evaluate the structural diversity of samples by counting the number of Foldseek [42] clusters formed by the filtered subset of backbones, using a TM-score threshold of 0.5. Higher cluster counts indicate greater diversity.
    - **DIV-AA**: Diversity metric filtered for All-Atom Co-designable samples (CODES-AA)
    - **DIV-CA**: Diversity metric filtered for Co-designable samples (CODES)

- **DIV-M8**: Diversity metric filtered for Designable samples (DES-M8)
- **DIV-M1**: Diversity metric filtered for Designable samples (DES-M1)

4. **Novelty**: This metric evaluates a model's ability to generate structures that are distinct from those in predefined reference sets (PDB and $\mathcal{D}_{\text{Genie2}}$). We report the average maximum TM-score between designable structures and the reference sets, with lower scores indicating greater novelty. Specifically, we report PDB novelty for co-designable samples (**NOV-PDB**) and all-atom co-designable samples (**NOV-PDB-AA**), as well as their counterparts with respect to Genie2 (**NOV-AFDB** and **NOV-AFDB-AA**).

Our evaluation protocol involves generating samples across a range of lengths, from 50 to 250 to align with prior work such as P(all-atom) as well as 50 to 400 to evaluate a more difficult spectrum. We then compute the aforementioned metrics across these samples. For designability, we use ProteinMPNN to generate sequences for each backbone and ESMFold [29] to predict structures, calculating the self-consistency RMSD (scRMSD) between predicted and original structures. A sample is considered designable if its scRMSD is under 2Å.

G.2 Side Chain Accuracy Metrics

To evaluate the atomistic protein structures generated by *Proteína-Atomística*, we compute several metrics that assess the accuracy and physical correctness of the generated side chains. These metrics include:

1. **MolProbity Score**: The MolProbity (MP) score is a composite evaluation metric that assesses the geometric and stereochemical quality of macromolecular structures [11]. It is a combination of several individual metrics, including:

   - **Clashscore**: Measures the number of steric clashes between atoms in the protein structure.
   - **Ramachandran outliers**: Refers to the percentage of residues with dihedral angles (ϕ and ψ) that fall outside the allowed regions of the Ramachandran plot.
   - **Rotamer outliers**: Refers to the percentage of residues with side-chain conformations that are inconsistent with the expected rotameric states.

   A lower MolProbity (MP) score indicates higher structure quality. Notably, a score of ≥ 3 indicates significant stereochemical issues, highlighting potential problems with the structure's accuracy. The MP score is a widely used and reliable metric for evaluating protein structure quality.
2. **Clash Scores**: Clash scores measure the number of steric clashes between atoms in the protein structure. Steric clashes occur when two or more atoms are too close to each other, resulting in unfavorable interactions. A lower clash score indicates fewer steric clashes and a more physically realistic structure.
3. **Bond Length Outliers**: Bond length outliers refer to the percentage of bonds in the protein structure that deviate significantly from their expected lengths. A lower percentage of bond length outliers indicates a more accurate structure.
4. **Angle Outliers**: Angle outliers refer to the percentage of bond angles in the protein structure that deviate significantly from their expected values. A lower percentage of angle outliers indicates a more accurate structure.

These metrics provide a comprehensive evaluation of the accuracy and physical correctness of the generated side chains. By assessing the MolProbity score, clash scores, bond length outliers, and angle outliers, we can better understand the strengths and weaknesses of *Proteína-Atomística* and prior atomistic models in generating accurate atomistic protein structures.

Across all lengths, *Proteína-Atomística* generates more accurate side chains compared to prior methods (Fig. 11). Proteína-Atomística achieves a length-averaged MP score of 2.097 compared to 4.307 of P(all-atom) (the next closest performing model from Table 1). The next closest is ProteinGenerator, which has an average MP score of 2.940 but has the lowest all-atom co-designability.

G.3 Baselines

In this section, we discuss the sampling configurations of the baselines we compared to in this paper. For backbone design methods not included below nor introduced by us, the results were taken from Geffner et al. [16].

Pallatom: We used the code and checkpoint from the public Pallatom repository. We used the default configuration suggested: `t_min=0.01`, `t_max=1`, `gamma=0.2`, `step_scale=2.25`, and `T=200`. No training code is provided at this time.

Protpardelle: We used the code and checkpoint from the public Protpardelle repository. We used the default `uncond_sampling.yml` file provided in the repository for unconditional sampling. For motif scaffolding, we prepared the `.pdb` files of each task based on the corresponding contigs and then used the provided `cond_sampling.yml` configuration for sampling.

ProteinGenerator: We used the code from the public ProteinGenerator repository. We used the base checkpoint set in the repository. We followed the default configuration for unconditional sampling except for the number of sampling steps. Since we sampled proteins with length up to 400 residues, we increased the number of sampling steps from the default 25 to 100 for better generation quality, as recommended in the repository.

PLAID: We used the code from the public PLAID repository. We used the 100M parameter checkpoint hosted on the PLAID HuggingFace repo as it is the only loadable option. Since PLAID only supports sampling proteins with length divisible by 4, the actual length we sampled are $[48, 96, 152, 200, 248, 296, 352, 400]$. We used the default unconditional sampling configuration in the repository.

DPLM-2: We use the code from the DPLM Repository specifically the pull request from DPLM-2 branch. We follow the instructions in the README.md to generate proteins from their 650M co-generation model using the indicated inference configuration and settings.

MultiFlow: We use the code from the MultiFlow Repository. We use the provided `inference_unconditional` config provided adjusted for the appropriate length intervals.

CarbonNovo: We use the code from the CarbonNovo Repository. We use the provided predict.py to generate proteins of the desired lengths.

FoldFlow-2: We use the code from the FoldFlow Repository. We use the `runner/inference.py` script with both provided FoldFlow-2 weights with `model=ff2`. We use the default sampling parameters provided in inference.yaml.

H Limitations

While Proteína-Atomística performs well, it faces challenges in balancing natural sequence distribution learning with generated sample diversity. Key limitations include: increased computational cost and decreased speed associated with full-atom modeling compared to backbone-only approaches; the inability to capture protein dynamics; and the lack of guarantees for desired function or binding affinity. These limitations highlight exciting directions for future research. Future work can additionally explore similar techniques for conditional tasks such as motif scaffolding and binder design, as well as the generation of even longer protein sequences as done in La-Proteína [15].

I Broader Impact

Our method advances the field of de novo protein design by enabling joint generation of sequences and all-atom structures, with potential applications in drug discovery, enzyme engineering, and biomaterials. While this capability could accelerate the development of novel therapeutics and sustainable biocatalysts, it raises ethical considerations, such as the risk of misuse for harmful purposes. Additionally, the model's performance depends on the quality of training data, which may inherit biases from structure prediction tools like AlphaFold2 and ESMFold. We emphasize responsible use and encourage further research into safety measures and bias mitigation to ensure positive societal impact.