DOMAIN-AWARE PROTEIN TOKENIZATION FOR STRUCTURE GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

De novo protein backbone generation is a central task in protein design, and discrete structural representations provide a multimodal generative paradigm for this task by enabling the unified modeling of protein structure, sequence, and function. However, existing methods focus primarily on residue-level local geometry and rarely use structural domains as training signals, limiting their ability to capture the modular organization of proteins. We introduce DPAR, a domain-aware discrete framework that encodes backbone geometry and domain segmentation as a sequence of discrete symbols. DPAR combines region-aware tokenization that captures both domain and global content with autoregressive generation of structure tokens and domain boundaries, using the resulting layout to guide segment-wise conditioning of a flow-matching decoder. We validate DPAR on reconstruction and generation tasks and demonstrate that it matches or outperforms existing models based on residue-level protein structure tokenizers. We show how domain layouts enable segment-wise conditioning during inference, which boosts backbone designability. Together, these properties establish DPAR as a promising framework for domain-aware discrete protein structure generation.

1 INTRODUCTION

Protein structure provides the physical basis for protein function and is therefore essential for understanding life processes (Anfinsen, 1973; Henzler-Wildman & Kern, 2007; Dill & MacCallum, 2012). This central role has made protein structure generation an important task in computational biology (Huang et al., 2016). Recent generative models learn the distribution of natural protein structures and use this knowledge to generate novel protein backbones (Yim et al., 2023; Watson et al., 2023; Hayes et al., 2025; Wang et al., 2025b). Among existing approaches, discrete token-based generation is particularly attractive because it models continuous three-dimensional structures as sequences of tokens. This paradigm enables scalable language modeling and provides a natural interface for unified multimodal generation (van den Oord et al., 2017; Lu et al., 2024; Chameleon Team, 2024). At the core of this paradigm is a structure tokenizer, which maps continuous protein structures into discrete tokens for subsequent generative modeling.

However, current approaches to discrete token-based generation largely overlook the modular organization of protein folds. Proteins naturally exhibit multiscale substructures: residues form local structural elements, such as helices and β-strands, which then assemble through long-range contacts into motifs, domains, and complete backbones (Orengo et al., 1997; Somnath et al., 2021; Wang et al., 2025a). Among these levels, domains provide a natural intermediate scale between local geometry and global topology. They typically comprise spatially interacting residues and form compact units that can fold independently, often serving as reusable modules of protein structure, function, and evolution (Ponting & Russell, 2002; Vogel et al., 2004; Han et al., 2007). In particular, explicit supervision of protein domains has been shown to improve function-related prediction and protein sequence design (Calef et al., 2026; Liu et al., 2025; Dai et al., 2024; Yuan et al., 2025a). This suggests that domain information can provide intermediate-scale structural signals beyond those captured by residue-level or global representations alone.

Without explicit domain-level information, existing methods face two key limitations. First, structure tokens are generated without intermediate-scale constraints, allowing local prediction errors to accumulate during decoding and degrade backbone reconstruction and designability (Hsieh et al.,

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

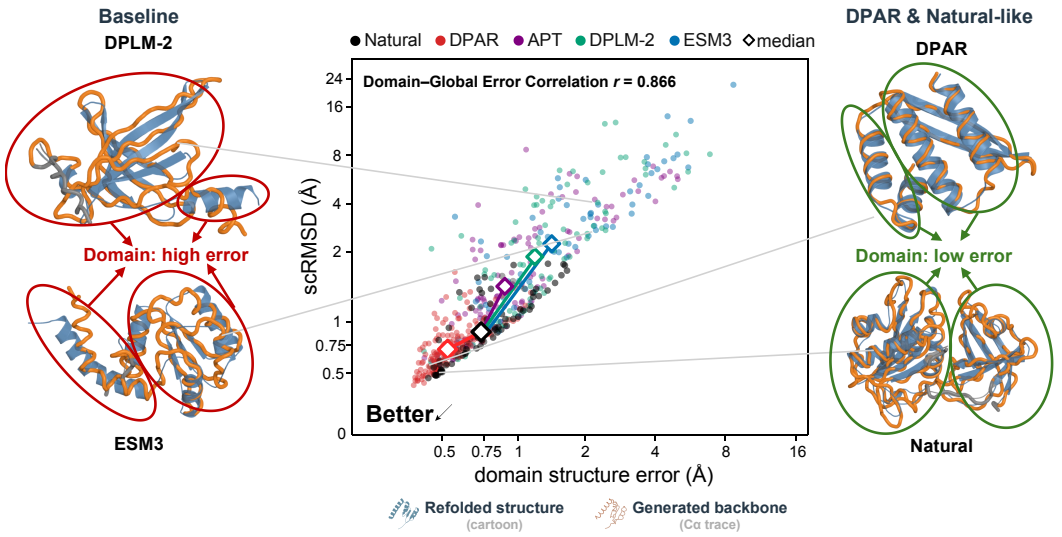


Figure 1: **The failure mode of existing methods.** Faithful generation of multidomain proteins requires preserving individual domain folds. Existing models exhibit a local–global inconsistency: domains can remain locally similar after refolding while undergoing substantial changes in their relative placement. Domain-structure error strongly correlates with scRMSD (Pearson's $r = 0.866$), supporting its use for evaluating multidomain consistency. Compared with existing models, DPAR achieves lower domain-structure error and scRMSD, narrowing the structural-consistency gap between generated and natural proteins. The details are defined in Appendix.

2025; Dilip et al., 2026a). Consistent with this limitation, current models for discrete token-based generation exhibit substantial domain-structure errors in multidomain proteins, affecting both individual domain folds and their relative arrangement (Vogel et al., 2004; Han et al., 2007). Second, prevailing models primarily operate on residue-level tokens or global structures (Dilip et al., 2026a; Qu et al., 2026; Sun et al., 2026), leaving domains underexplored as representational units and learning signals. As shown in Figure 1, these domain-structure errors correlate strongly with scRMSD (Pearson's $r = 0.866$), linking domain preservation to overall backbone quality. This motivates the question: *can domain information improve discrete token-based generation?*

To answer this question, we introduce DPAR, a domain-aware framework for discrete token-based generation that represents backbone geometry and domain segmentation within a unified sequence of discrete symbols. DPAR operates directly on the three-dimensional coordinates of protein backbones and incorporates domain grouping into the encoding process. Residues within each domain are encoded using domain-specific context, allowing the resulting representations to capture intra-domain geometry without discarding global information. We train a scalable Diffusion Transformer (DiT) decoder with a flow-matching loss to reconstruct complete protein backbones from structure tokens and their domain layouts, eliminating the need for multiple symmetry-invariant reconstruction losses typically used in protein structure tokenization (Peebles & Xie, 2023; Dilip et al., 2026b). Unlike prior approaches that leave domain organization implicit, DPAR adopts an autoregressive prior that jointly generates structure tokens and domain boundaries in a unified sequence and uses the resulting layout for domain-aware conditioning (Chen et al., 2022; Lu et al., 2024). Evaluated on reconstruction and generation tasks, DPAR substantially improves reconstruction and achieves state-of-the-art performance in discrete token-based generation, matching or outperforming prior methods across all evaluated generation metrics.

Finally, our model for discrete token-based generation matches or outperforms state-of-the-art methods across all evaluated metrics.

1. We propose a domain-aware protein structure tokenizer that combines intra-domain encoding with whole-chain context, incorporating domain information into discrete representation learning and enabling backbone decoding. To the best of our knowledge, this is the first tokenization model to explicitly incorporate domain information into discrete token-based generation.
2. We evaluate the model on reconstruction and generation tasks. In reconstruction tasks, we achieve significant improvements over existing models. The resulting model achieves state-of-the-art performance in discrete token-based generation and approaches that of leading continuous generative models.
3. We introduce a domain-guided strategy for selecting conditioning information at inference time. The strategy applies structured sparsification to mitigate inconsistencies between conditioning information used early in generation and the target structure.

2 Related Work

Protein Tokenization. Discrete generative models typically encode continuous data into tokens and learn generative distributions in the resulting discrete space (van den Oord et al., 2017; Esser et al., 2021; Austin et al., 2021; Sun et al., 2024). This paradigm has recently been extended to protein structure modeling, where structure tokenizers encode continuous 3D structures into discrete representations, enabling unified modeling of structure, sequence, and functional information in a shared token space (Gao et al., 2025; Hayes et al., 2025; Wang et al., 2025b). Early works typically construct structure tokens from local 3D environments around individual residues, relying on geometric modules or specialized reconstruction losses. More recent methods improve the scalability and geometric robustness of this pipeline by using data augmentation to induce stochastic invariance or equivariance and scalable architectures (Yuan et al., 2025b; Dilip et al., 2026a;b; Sun et al., 2026). Despite these advances, structure tokens remain largely residue-level or local-geometric, leaving domain organization to token context.

Flow Matching and Diffusion Autoencoders. Diffusion and Flow Matching offer generative alternatives to direct reconstruction in tokenizers. Diffusion reverses progressive noising through iterative sampling (Ho et al., 2020), whereas Flow Matching learns a transport velocity field and can reduce inference steps through straighter paths (Lipman et al., 2023; Liu et al., 2023). In protein modeling, geometry-aware diffusion and structure-space Flow Matching are widely used for backbone generation and conditional design (Yim et al., 2023; Watson et al., 2023; Bose et al., 2024; Yim et al., 2024; Geffner et al., 2025). Their use in autoencoders enables generative decoding from continuous latents or discrete tokens with less reliance on complex reconstruction losses and specialized geometric components (Preechakul et al., 2022; Li et al., 2025; Dilip et al., 2026b). DPAR uses this capacity to reconstruct backbones from partial token sequences with a Flow Matching decoder.

Domain-Aware and Hierarchical Protein Modeling. Proteins are naturally organized from residues and secondary-structure elements to domains and complete folds, linking local geometry to global topology (Orengo et al., 1997; Vogel et al., 2004). This hierarchical view has motivated models that incorporate substructure through multi-scale representations or substructure supervision. HoloProt jointly models protein sequences, residue-level structures, and molecular surfaces, while SSHG groups residues into secondary-structure motifs to capture higher-order geometric relationships (Somnath et al., 2021; Wang et al., 2025a). These methods organize protein representations across structural scales, but do not use domains as units for 3D backbone generation. Other approaches introduce substructures through supervision: SES-Adapter incorporates FoldSeek- and DSSP-derived structural descriptions into protein language models, while Magneton uses domain, motif, and functional-site annotations to improve protein encoders (Tan et al., 2024; Calef et al., 2026). In these models, substructures remain supervision or inputs rather than variables generated with 3D structures. ProDVa further introduces domains into generation through a dynamic vocabulary of functionally annotated natural fragments for protein sequence generation (Liu et al., 2025). However, these sequence fragments do not directly represent domain geometry or inter-domain spatial relationships. DPAR instead models domain boundaries alongside structure tokens to organize 3D backbone generation at the domain level.

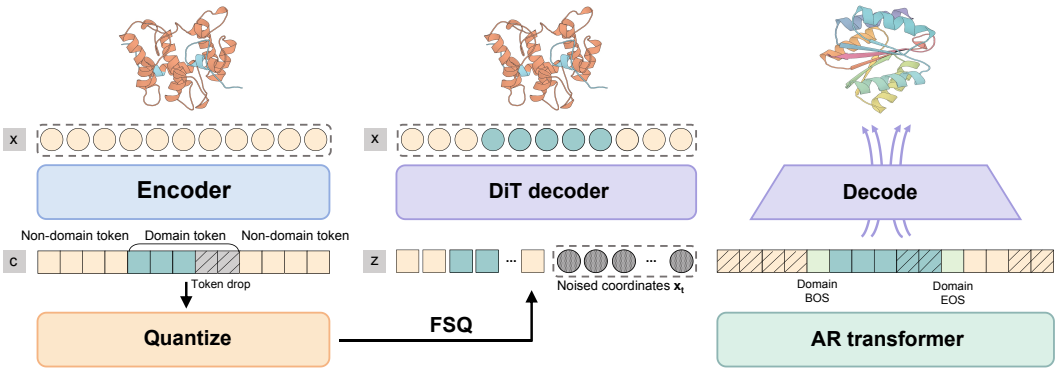


Figure 2: **Overview of DPAR.** DPAR is a domain-aware protein backbone generation model comprising a region-aware structure tokenizer and an autoregressive prior. During training, the tokenizer combines domain-specific and whole-chain features and quantizes them into one structure token per residue; a conditional flow decoder learns to reconstruct the complete Cα backbone from partial token conditions. The autoregressive Transformer then learns to generate structure tokens and domain boundary symbols jointly. At inference, DPAR selects conditioning tokens within each generated segment and uses the flow decoder to construct the complete backbone.

3 METHOD

DPAR consists of a region-aware structure tokenizer and an autoregressive prior. Within the tokenizer, the encoder and quantization module map a backbone to discrete tokens, while a diffusion decoder reconstructs the complete backbone from partial tokens. Training proceeds in two stages. First, we train an autoencoder, following flow-based protein models such as Proteina and AlphaFold 3 (Geffner et al., 2025; Abramson et al., 2024; Wang et al., 2025c). Second, we train a GPT-style autoregressive model over DPAR tokens to evaluate generative capabilities (Radford et al., 2018; 2019).

Feature Preparation. Given centered Cα coordinates $\mathbf{x}^{\text{data}} \in \mathbb{R}^{L\times 3}$ and region annotations from the TED dataset (Lau et al., 2024), we construct a region-aware representation for the residues. The normalized annotations specify up to three domain intervals $\mathcal{D}_s$, where $s \in \{0, \ldots, S-1\}$ and $S \leq 3$; we denote the complete collection by $\mathcal{D} = \{\mathcal{D}_s\}_{s=0}^{S-1}$. Residues in the s-th selected domain receive the group label, while all remaining residues are assigned to a shared residual group. We apply a random rotation during training to learn stochastic equivariance (Geffner et al., 2025).

Under the flow-autoencoder framework , we encode the full chain and each selected domain separately. The geometric Transformer takes the sum of coordinate, residue-position, and region-relative position embeddings, with geometric distances and sequence separations encoded as attention biases (Vaswani et al., 2017). Domain residues use their domain-specific representations, while residual residues use the full-chain representations. These features are restored to the original residue order and combined with a domain/residual type embedding and layer normalization to obtain the final region-aware representation.

The encoder outputs a latent conditioning sequence $e_\theta(\mathbf{x}^{\text{data}}) = \mathbf{c} \in \mathbb{R}^{L\times d}$. Finite scalar quantization (FSQ) quantizes the projected latent vectors using $(8, 8, 8, 8)$ levels, yielding conditioning tokens with an effective codebook size of 4096 (Mentzer et al., 2024) . These tokens condition a diffusion decoder trained with a flow-matching objective.

Reconstruction Loss. To enable partial structural conditioning, we independently retain a random prefix within each annotated domain while keeping non-domain token content visible. For chains without annotated domains, we retain a random whole-chain prefix. We additionally drop all structural token content with probability 0.05 while preserving residue positions and the region layout. These nested prefixes provide variable conditioning budgets(Rippel et al., 2014; Bachmann et al., 2025; Dilip et al., 2026a). The decoder reconstructs all valid residues using linear flow matching(Lipman et al., 2023). Appendix B specifies the sampling and training details. We interpolate

between Gaussian noise and the ground-truth coordinates and supervise the velocity over all valid residues:

$$\mathbf{x}_t = (1-t)\boldsymbol{\epsilon} + t\mathbf{x}^{\text{data}}, \qquad \mathcal{L}_{\text{full}} = \mathbb{E}\left[\left\| v_\phi(\mathbf{x}_t, t; \mathcal{C}) - \left(\mathbf{x}^{\text{data}} - \boldsymbol{\epsilon}\right)\right\|^2_{\text{valid}}\right]. \tag{1}$$

Here, $\mathcal{C}$ contains the partial token conditions and their layout, $\boldsymbol{\epsilon}$ is Gaussian noise, and t interpolates from noise at $t = 0$ to data at $t = 1$. The norm averages the squared coordinate error over all valid residues. To strengthen intra-domain supervision, we apply the same objective to valid domain crops, yielding $\mathcal{L}_{\text{domain}}$. We jointly train the tokenizer with

$$\mathcal{L}_{\text{AE}} = \mathcal{L}_{\text{full}} + \lambda_D \mathcal{L}_{\text{domain}}, \tag{2}$$

We set $\lambda_D = 0.05$ to balance whole-chain and domain-level supervision.

Autoregressive Generation Protein structure modeling involves both spatial dependencies among residues and organization at the domain level (Orengo et al., 1997; Lau et al., 2024; Qu et al., 2026). We therefore introduce domain boundaries explicitly into residue-level autoregressive prediction , encoding structural content and regional layout in the same symbolic sequence. The model jointly learns the distributions of structure tokens and boundaries; the generated layout is subsequently used for segment-wise condition selection. We employ a parseable boundary serialization by inserting the shared symbols `DOM_START` and `DOM_END` on either side of each domain (Vinyals et al., 2015). These symbols mark entering and leaving a domain, respectively, while individual domains are distinguished by their order of occurrence. After appending `EOS`, the conditioning tokens and boundary symbols form a serialized sequence $\mathbf{y}$ of length $T = L + 2S + 1$. Its joint probability factorizes autoregressively as

$$p_\psi(\mathbf{y}) = \prod_{j=0}^{T-1} p_\psi(y_j \mid \mathbf{y}_{<j}). \tag{3}$$

We combine chain-level and within-domain positional encodings with attention biases for domain membership and shared segments. Training uses weighted next-token cross-entropy and an auxiliary objective that predicts the distance to the next control symbol at non-domain structure-token positions:

$$\mathcal{L}_{\text{AR}} = \mathcal{L}_{\text{CE}} + \lambda_b \mathcal{L}_{\text{bdry}}. \tag{4}$$

Segment-wise Conditioning. We first sample a complete sequence of structure tokens and boundary symbols under target-length and grammar constraints, using temperature scaling followed by top-p and min-p filtering (Holtzman et al., 2020; Nguyen et al., 2025). Prior work shows that tail drop can reduce autoregressive sampling errors by delegating high-frequency structural details to the diffusion decoder (Dilip et al., 2026a). However, a whole-chain cutoff concentrates retained tokens near the beginning of the sequence, potentially leaving later domains without sufficient structural guidance. By generating structure tokens and boundary symbols, DPAR produces a parseable layout of domain and residual segments. This layout enables segment-specific conditioning by retaining only a structural prefix for each domain segment while masking all token features of the residual segment preceding the first domain. These operations reduce reliance on autoregressively generated details while maintaining structural guidance across domains.

To select the tail-drop cutoff, we use minimum-entropy sampling. Finite cutoff sampling cannot adapt the retained amount across regions, whereas a fixed entropy cutoff depends on a predefined absolute threshold. Minimum-entropy sampling instead determines cutoffs from changes in each segment's entropy curve, making it suitable for selecting conditioning prefixes independently for every domain or residual segment. Specifically, we compute predictive entropy at each position, smooth the entropy curve separately within each contiguous domain or residual block, and retain the content preceding the first local minimum. The cutoff rule and fallback cases are specified in Appendix B.2.

Diffusion Decoding. To reconstruct a complete backbone from the selected tokens, we fix the content and layout condition $\mathcal{C}$, integrate the learned velocity field from a centered Gaussian initialization, and solve the resulting ordinary differential equation with the Euler method:

$$d\mathbf{x}_t = v_\phi(\mathbf{x}_t, t; \mathcal{C})\, dt. \tag{5}$$

To exploit structural cues from the autoregressive model while reducing the influence of sampling bias during late-stage reconstruction, we employ conditioning annealing, based on classifier-free guidance (Kynkaanniemi et al., 2024; Chung et al., 2025; Ho & Salimans, 2022). The strategy progressively decreases the weight of the content-conditioned branch, so that late-stage reconstruction relies more heavily on the decoder prior:

$$\mathbf{v}_g(\mathbf{x}_t, t) = \mathbf{v}_u(\mathbf{x}_t, t) + (1 - t^\alpha)\left[\mathbf{v}_c(\mathbf{x}_t, t) - \mathbf{v}_u(\mathbf{x}_t, t)\right], \qquad \alpha \in (0, 1]. \tag{6}$$

Here, $\mathbf{v}_c$ and $\mathbf{v}_u$ are two evaluations of the same decoder under the same condition $\mathcal{C}$: $\mathbf{v}_c$ retains the token content, whereas $\mathbf{v}_u$ sets that content to zero while retaining the same layout. The exponent α controls how rapidly reconstruction shifts from conditioned guidance to the decoder prior. To control the trade-off between structural quality and diversity, we use the relationship between velocity and score in Gaussian flows (Albergo et al., 2025). The score, defined as the gradient of the log-density of the noisy distribution, can be estimated from the velocity:

$$\mathbf{s}_\phi(\mathbf{x}_t, t; \mathcal{C}) = \frac{t\, v_\phi(\mathbf{x}_t, t; \mathcal{C}) - \mathbf{x}_t}{1 - t}, \qquad t < 1. \tag{7}$$

During numerical integration, the score is evaluated only up to $t = 1 - \varepsilon_t$, where $\varepsilon_t > 0$ is a small numerical cutoff, equivalently clipping the denominator away from zero. The resulting guided score $\mathbf{s}_g$ enters the continuous-time dynamics in Eq. (8). To control density correction and stochastic exploration separately, we introduce a score-drift term and a stochastic perturbation, yielding the stochastic differential equation

$$d\mathbf{x}_t = \left[\mathbf{v}_g + \gamma\kappa(t)\mathbf{s}_g\right] dt + \sqrt{2\eta\kappa(t)}\, d\mathbf{W}_t. \tag{8}$$

Here, γ and η control the correction and noise strengths, respectively; $\kappa(t)$ modulates temporal weighting, and $\mathbf{W}_t$ is a Wiener process in the centered coordinate space.

4 Experiments

4.1 Structural Reconstruction

We evaluate the reconstruction quality of the discrete structural representation using RMSD and TM-score. RMSD measures the coordinate deviation between the reconstructed backbone and the ground-truth structure, whereas TM-score measures their consistency in overall topology. We conduct evaluations on CAMEO, CASP14, CASP15 and a CATH subset. During data splitting, we exclude from the training set any samples homologous to proteins in any downstream test set. For proteins without official domain annotations, DPAR treats them as single-domain samples in the reconstruction task. We compare DPAR with DPLM-2, ESM3, FoldToken4, Kanzi and IST, all of which represent protein structures using discrete tokens. Table 1 shows that DPAR maintains strong structural reconstruction performance across all four test sets, achieving the lowest RMSD on CAMEO and CASP15 datasets. These results validate the structural fidelity of the tokens and provide a basis for evaluating their generative capability.

We report two additional metrics on CATH to localize reconstruction errors. We consider these metrics only over CATH dataset because its domain annotations are based on experimentally determined PDB structures and define the regions needed for both calculations. Domain RMSD measures local geometry after independently aligning each annotated domain. Outside-span RMSD quantifies structural deviations in residues outside the annotated domain spans after whole-chain alignment, including connecting and other unannotated regions.

While the above metrics capture reconstruction fidelity, strong reconstruction performance alone does not guarantee high-quality backbone generation (Hsieh et al., 2025; Dilip et al., 2026b). Since DPAR targets unconditional backbone generation, we prioritize generation quality when selecting its tokenizer.

4.2 Downstream Structural Analysis

Unconditional Backbone Generation. Among discrete token-based generation methods, we compare DPAR with ESM3, DPLM-2, Kanzi, and APT, with results reported in Table 2. Inference settings and evaluation metrics are provided in Appendix B and Appendix D. Among the evaluated

Table 1: Structural reconstruction quality of different protein autoencoders. RMSD values are in Å. D-RMSD and O-RMSD denote Domain RMSD and Outside-domain RMSD, respectively. Lower RMSD and higher TM-score indicate better performance. **Bold** = best (per column); underline = second-best

Model	CAMEO		CASP14		CASP15		CATH			
	RMSD (↓)	TM (↑)	RMSD	TM	RMSD	TM	RMSD	TM	D-RMSD	O-RMSD
DPLM-2 (118M)	1.712	0.893	1.437	0.918	3.310	0.810	3.675	0.787	1.321	6.955
ESM3 (648M)	0.959	**0.961**	**0.540**	**0.983**	1.365	0.941	1.628	0.924	**0.560**	2.209
FoldToken4 (85M)	1.252	0.916	1.275	0.922	1.875	0.882	1.943	0.868	1.236	2.576
Kanzi (44M)	1.060	0.950	0.941	0.955	1.408	0.933	1.218	0.930	0.956	1.475
IST (11M)	1.152	0.948	1.478	0.924	1.261	0.938	1.964	0.885	1.019	2.609
DPAR (44M)	**0.919**	0.946	0.842	0.971	**1.110**	**0.956**	**1.054**	**0.947**	0.711	**1.447**

Table 2: Unconditional protein backbone generation results across models. **Bold** = best (per column); underline = second-best. The ESM3 records use 32 and 256 sampling steps. DPAR results are shown under two temperature and min-p configurations. Metric definitions are provided in Appendix.

Model	**QUALITY**		**DIVERSITY**	
	Designability (↑)	scRMSD (↓)	Diversity (↓)	Novelty (↓)
ESM3 (1.4B/32)	0.404	3.502	0.369	0.789
ESM3 (1.4B/256)	0.463	3.164	0.373	0.783
DPLM-2 (650M)	0.619	2.278	**0.326**	0.730
Kanzi (250M)	0.617	3.650	0.386	0.763
APT (252M)	0.871	1.352	0.395	0.774
DPAR(210M)				
$T = 0.70$, min-p $= 0.05$	**0.959**	**0.832**	0.487	**0.713**
$T = 0.75$, min-p $= 0$	0.938	0.946	0.417	0.819

discrete token-based generation methods, DPAR achieves the highest designability (0.9594) and the lowest scRMSD (0.832), while maintaining competitive diversity and novelty. We further examine the effects of autoregressive sampling parameters. Compared with $T = 0.75$ and min-p $= 0$, $T = 0.70$ and min-p $= 0.05$ improves designability by 2.3%, reduces scRMSD by 12.1%, and lowers the novelty score by 12.9%, at the cost of reduced diversity. Among samples with AR domains, generated domain residues and Merizo-assigned domains in refolded backbones achieve a mean IoU of 0.691, indicating coarse layout agreement (Lau et al., 2023); evaluation details are provided in Appendix F.

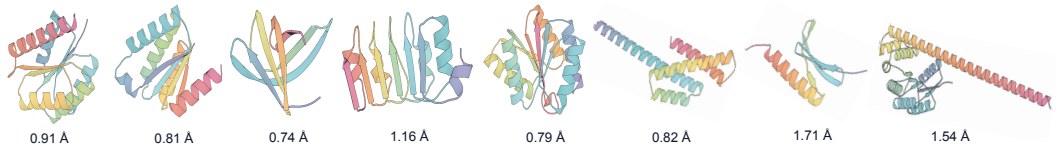


Figure 3: Designable samples generated from an autoregressive model trained on DPAR tokens. scRMSDs shown underneath each visualization.

CATH C-level class-conditional generation. Previous studies of protein backbone generation have observed that generated samples tend to be enriched in α-helices, with relatively fewer β-sheets (Geffner et al., 2025). For design tasks requiring a specific fold type, high designability cannot substitute for control over the target structural composition. To test whether the generator can cover different overall fold classes, we train it conditioned on CATH-C level labels, specifying the "mainly α," "mainly β," and "mixed α/β" classes. Table 3 summarizes the results by target class.

Table 3: Generation results conditioned on CATH structural classes.Results are reported for the "mainly α," "mainly β," and "mixed α/β" CATH-C classes.

CATH Class	GENERATION METRICS		Sec. Struct. (%)		
	Designability	Diversity	α	β	Coil
Mainly α	0.962	0.430	78.72	3.03	18.25
Mainly β	0.903	0.404	45.37	20.76	33.87
Mixed α/β	0.948	0.408	61.01	10.94	28.05

In addition to designability, we report the proportions of residues in α-helices, β-sheets, and coils to compare the structural composition produced under different class conditions. The proportion of β-sheet structure increases substantially under the mixed α/β" and, in particular, the mainly β" conditions compared with the "mainly α" condition, while the generated backbones maintain high designability. Further details on conditional generation are provided in Appendix B.5.

4.3 Ablation Studies

Segment-wise Selection. To examine how segment-wise selection shapes continuous decoding, we ablate the positions of autoregressive structure tokens retained as decoding conditions. DPAR applies spline smoothing separately to each segment's predicted-entropy curve and retains the prefix up to its first local minimum, extending prior entropy-based prefix selection to the segment level(Dilip et al., 2026a). Approximately 80% of proteins in the training set have zero or one annotated domain. We therefore conduct the conditioning-token dropout ablation on proteins with one annotated domain using position-preserving token dropout; the findings are specific to this single-domain setting. Pre-domain token dropout, post-domain token dropout, and within-domain token dropout remove conditioning information before, after, and within the domain span, respectively, while leaving the region layout unchanged. Table 4 reports only these results and shows that pre-domain token dropout simultaneously increases designability and decreases scRMSD. The generated domain layout defines the leading residual segment and per-domain prefixes used to condition the flow decoder. We therefore evaluate domain-aware representation, boundary prediction, and leading-residual masking as a coupled generation strategy, while the dropout ablation probes conditioning choices within this strategy. These results indicate that the domain span delimited by `DOM_START`/`DOM_END` defines the segmentation and provides a more effective conditioning region for protein generation.

Table 4: Results of conditional token dropout on single-domain proteins. Token dropout zeros the corresponding conditioning values while leaving token positions and the region layout unchanged. Random token dropout zeros the same number of conditions as within-domain token dropout, allowing us to determine whether the observed change arises from the domain positions themselves or merely from dropping an equal number of conditions.

Dropout setting	Avg. Tokens ($\mu \pm \sigma$)	Designability (↑)	scRMSD (↓)	scTM (↑)
Pre-domain	12.7 ± 4.8	0.971	0.803	0.951
Post-domain	18.9 ± 7.1	0.796	1.452	0.898
Within-domain	7.8 ± 6.5	0.839	2.065	0.885
Random	7.8 ± 6.5	0.810	2.048	0.881
Full condition	19.7 ± 7.6	0.796	1.472	0.897

Classifier Annealing. To assess the role of classifier annealing, we ablate the annealing exponent α to determine how the decay rate of autoregressive conditioning affects generation quality during continuous decoding. Table 5 shows classifier annealing progressively attenuates the influence of the conditions retained by segment-wise entropy selection, with a smaller α corresponding to faster attenuation. Across the tested values, lowering α from 1.00 to 0.25 consistently increases

Table 5: Effect of classifier annealing with segment-wise entropy selection held fixed. A smaller α produces faster decay of conditional guidance. Increasing classifier annealing strength improves designability and scTM while reducing scRMSD, whereas constant performs worst.

α	Designability (↑)	scRMSD (↓)	scTM (↑)
0.25	0.959	0.832	0.951
0.35	0.933	1.052	0.941
0.50	0.883	1.374	0.925
1.00	0.727	1.892	0.892
constant	0.583	2.699	0.850

designability and scTM while decreasing scRMSD. These results show that stronger conditioning annealing improves self-consistency metrics under the evaluated sampling configuration.

Codebook Size. We examine the effect of codebook capacity on reconstruction quality in Figure 4. Reducing the 4k codebook used in the base DPAR model to 1k increases CAMEO RMSD from 0.919 Å to 1.456 Å. This result indicates that compressing the codebook limits how much structural information the discrete representation can retain, and that the 4k codebook provides the representational capacity necessary for high-quality structural reconstruction in DPAR.

Scalability. We compare the Base encoder (2 layers, 256 dimensions) with the Large encoder (4 layers, 384 dimensions) using a 4k codebook. Increasing the encoder size raises RMSD from 0.919 Å to 1.190 Å, rather than producing the expected improvement in reconstruction. This observation suggests that the current DPAR model is not limited solely by encoder parameter capacity. There are a variety of possible confounders here. The benefits of a larger encoder may still depend on coordinated increases in decoder capacity, training data, and optimization settings.

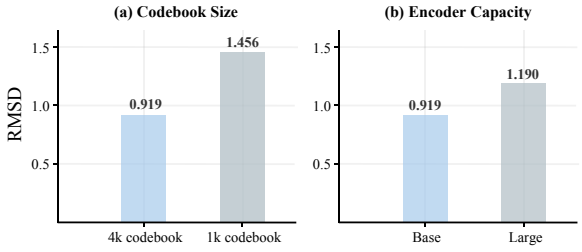


Figure 4: **Ablation of codebook size and encoder capacity.** (a) Comparison of 1k and 4k codebooks. (b) Comparison of the Base encoder (2 layers, 256 dimensions) and Large encoder (4 layers, 384 dimensions) with a 4k codebook. All results are evaluated on CAMEO using RMSD to assess reconstruction quality.

5 Conclusion

We present DPAR, a domain-aware framework that combines region-aware residue-level tokenization with autoregressive generation of structure tokens and domain boundaries. The generated region layout guides conditional flow decoding, linking structural content with mesoscale organization. To our knowledge, DPAR is the first protein structure tokenization framework to incorporate explicit domain information into discrete token-based protein backbone generation. Looking forward, DPAR provides a path toward modeling functional regions of varying lengths and scales, without requiring a fixed token count, together with their relationships to the full protein backbone. The present model focuses on single-chain Cα backbones, and its applicability to longer chains and more complex protein systems remains to be established. We plan to address these limitations and explore extensions of DPAR in future work.

AI USE STATEMENT

Generative AI was used during the preparation of this paper to assist with language and grammar refinement, as well as with the implementation and debugging of certain code. All AI-assisted text was carefully reviewed and revised by the authors. Similarly, any AI-assisted code was manually inspected, tested, and validated before use. The authors take full responsibility for the accuracy and integrity of the final manuscript and all reported results.

ETHICS STATEMENT

This study uses protein structures and domain annotations described in Appendix C and does not involve human participants or identifiable personal data.

REPRODUCIBILITY STATEMENT

Section 3 describes the model and training objectives, while Appendix B, Appendix C, and Appendix D provide hyperparameters, data processing, and evaluation protocols.

REFERENCES

Josh Abramson, Jonas Adler, Jack Dunger, Richard Evans, Tim Green, Alexander Pritzel, Olaf Ronneberger, Lindsay Willmore, Andrew J. Ballard, Sebastian W. Bodenstein, et al. Accurate structure prediction of biomolecular interactions with AlphaFold 3. *Nature*, 630:493–500, 2024.

Michael S. Albergo, Nicholas M. Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *Journal of Machine Learning Research*, 26(209):1–80, 2025.

Christian B. Anfinsen. Principles that govern the folding of protein chains. *Science*, 181(4096): 223–230, 1973.

Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. In *Advances in Neural Information Processing Systems*, volume 34, pp. 17981–17993, 2021.

Roman Bachmann, Jesse Allardice, David Mizrahi, Enrico Fini, Oguzhan Fatih Kar, Elmira Amirloo, Alaaeldin El-Nouby, Amir Zamir, and Afshin Dehghan. FlexTok: Resampling images into 1D token sequences of flexible length. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 2241–2292. PMLR, 2025.

Avishek Joey Bose, Tara Akhound-Sadegh, Guillaume Huguet, Kilian Fatras, Jarrid Rector-Brooks, Cheng-Hao Liu, Andrei Cristian Nica, Maksym Korablyov, Michael M. Bronstein, and Alexander Tong. SE(3)-stochastic flow matching for protein backbone generation. In *International Conference on Learning Representations*, 2024.

Robert Calef, Arthur Liang, Manolis Kellis, and Marinka Zitnik. Greater than the sum of its parts: Building substructure into protein encoding models. In *International Conference on Learning Representations*, 2026.

Chameleon Team. Chameleon: Mixed-modal early-fusion foundation models. *arXiv preprint arXiv:2405.09818*, 2024.

Ting Chen, Saurabh Saxena, Lala Li, David J. Fleet, and Geoffrey Hinton. Pix2Seq: A language modeling framework for object detection. In *International Conference on Learning Representations*, 2022.

Hyungjin Chung, Jeongsol Kim, Geon Yeong Park, Hyelin Nam, and Jong Chul Ye. CFG++: Manifold-constrained classifier free guidance for diffusion models. In *International Conference on Learning Representations*, 2025.

Crystallography: Protein Data Bank. Nature New Biology, 233:223, 1971.

Fengyuan Dai, Yuliang Fan, Jin Su, Chentong Wang, Chenchen Han, Xibin Zhou, Jianming Liu, Hui Qian, Shunzhi Wang, Anping Zeng, Yajie Wang, and Fajie Yuan. Toward de novo protein design from natural language. *bioRxiv*, 2024. Preprint, version 1.

J. Dauparas, I. Anishchenko, N. Bennett, et al. Robust deep learning–based protein sequence design using ProteinMPNN. *Science*, 378(6615):49–56, 2022. doi: 10.1126/science.add2187.

Rohit Dilip, Ayush Varshney, and David Van Valen. Adaptive protein tokenization. *arXiv preprint arXiv:2602.06418*, 2026a.

Rohit Dilip, Evan Zhang, Ayush Varshney, and David Van Valen. Flow autoencoders are effective protein tokenizers. In *International Conference on Learning Representations*, 2026b.

Ken A. Dill and Justin L. MacCallum. The protein-folding problem, 50 years on. *Science*, 338 (6110):1042–1046, 2012.

Patrick Esser, Robin Rombach, and Björn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12873–12883, 2021.

Zhangyang Gao, Cheng Tan, Jue Wang, Yufei Huang, Lirong Wu, and Stan Z. Li. FoldToken: Learning protein language via vector quantization and beyond. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(1):219–227, 2025.

Tomas Geffner, Kieran Didi, Zuobai Zhang, Danny Reidenbach, Zhonglin Cao, Jason Yim, Mario Geiger, Christian Dallago, Emine Kucukbenli, Arash Vahdat, et al. Proteina: Scaling flow-based protein structure generative models. In *International Conference on Learning Representations*, 2025.

Jung-Hoon Han, Sarah Batey, Adrian A. Nickson, Sarah A. Teichmann, and Jane Clarke. The folding and evolution of multidomain proteins. *Nature Reviews Molecular Cell Biology*, 8:319–330, 2007.

Thomas Hayes, Roshan Rao, Halil Akin, Nicholas J. Sofroniew, Deniz Oktay, Zeming Lin, Robert Verkuil, Vincent Q. Tran, Jonathan Deaton, Marius Wiggert, et al. Simulating 500 million years of evolution with a language model. *Science*, 387(6736):850–858, 2025.

Katherine Henzler-Wildman and Dorothee Kern. Dynamic personalities of proteins. *Nature*, 450: 964–972, 2007.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851, 2020.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. In *International Conference on Learning Representations*, 2020.

Cheng-Yen Hsieh, Xinyou Wang, Daiheng Zhang, Dongyu Xue, Fei Ye, Shujian Huang, Zaixiang Zheng, and Quanquan Gu. Elucidating the design space of multimodal protein language models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 24156–24175. PMLR, 2025.

Po-Ssu Huang, Scott E. Boyken, and David Baker. The coming of age of de novo protein design. *Nature*, 537(7620):320–327, 2016.

G. Huguet, J. Vuckovic, K. Fatras, et al. Sequence-augmented SE(3)-flow matching for conditional protein backbone generation. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

W. Kabsch. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A*, 32(5):922–923, 1976.

Tuomas Kynkaanniemi, Miika Aittala, Tero Karras, Samuli Laine, Timo Aila, and Jaakko Lehtinen. Applying guidance in a limited interval improves sample and distribution quality in diffusion models. In *Advances in Neural Information Processing Systems*, volume 37, pp. 122458–122483, 2024.

A. M. Lau, S. M. Kandathil, and D. T. Jones. Merizo: a rapid and accurate protein domain segmentation method using invariant point attention. *Nature Communications*, 14:8445, 2023. doi: 10.1038/s41467-023-43934-4.

Andy M. Lau, Nicola Bordin, Shaun M. Kandathil, Ian Sillitoe, Vaishali P. Waman, Jude Wells, Christine A. Orengo, and David T. Jones. Exploring structural diversity across the protein universe with the encyclopedia of domains. *Science*, 386(6721):eadq4946, 2024.

Shaoning Li, Le Zhuo, Yusong Wang, Mingyu Li, Xinheng He, Fandi Wu, Hongsheng Li, and Pheng-Ann Heng. ProteinAE: Protein diffusion autoencoders for structure encoding. *arXiv preprint arXiv:2510.10634*, 2025.

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.

Z. Lin, H. Akin, R. Rao, et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023. doi: 10.1126/science.ade2574.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023.

Nuowei Liu, Jiahao Kuang, Yanting Liu, Tao Ji, Changzhi Sun, Man Lan, and Yuanbin Wu. Protein design with dynamic protein vocabulary. In *Advances in Neural Information Processing Systems*, 2025.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations*, 2023.

Jiasen Lu, Christopher Clark, Sangho Lee, Zichen Zhang, Savya Khosla, Ryan Marten, Derek Hoiem, and Aniruddha Kembhavi. Unified-IO 2: Scaling autoregressive multimodal models with vision, language, audio, and action. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26439–26455, 2024.

Fabian Mentzer, David Minnen, Eirikur Agustsson, and Michael Tschannen. Finite scalar quantization: VQ-VAE made simple. In *International Conference on Learning Representations*, 2024.

Minh Nguyen, Andrew Baker, Clement Neo, Allen Roush, Andreas Kirsch, and Ravid Shwartz-Ziv. Turning up the heat: Min-p sampling for creative and coherent LLM outputs. In *International Conference on Learning Representations*, 2025.

C. A. Orengo, A. D. Michie, S. Jones, D. T. Jones, M. B. Swindells, and J. M. Thornton. CATH—a hierarchic classification of protein domain structures. *Structure*, 5(8):1093–1108, 1997.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4195–4205, 2023.

Chris P. Ponting and Robert R. Russell. The natural history of protein domains. *Annual Review of Biophysics and Biomolecular Structure*, 31:45–71, 2002.

Konpat Preechakul, Nattanat Chatthee, Suttisak Wizadwongsa, and Supasorn Suwajanakorn. Diffusion autoencoders: Toward a meaningful and decodable representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10619–10629, 2022.

Yanru Qu, Cheng-Yen Hsieh, Zaixiang Zheng, Ge Liu, and Quanquan Gu. Protein autoregressive modeling via multiscale structure generation. In *International Conference on Machine Learning*, 2026.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. Technical report, OpenAI, 2018.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. Technical report, OpenAI, 2019.

Oren Rippel, Michael Gelbart, and Ryan Adams. Learning ordered representations with nested dropout. In *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pp. 1746–1754. PMLR, 2014.

C. E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27 (3):379–423, 1948.

Vignesh Ram Somnath, Charlotte Bunne, and Andreas Krause. Multi-scale representation learning on proteins. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

M. Steinegger and J. Söding. MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nature Biotechnology*, 35:1026–1028, 2017. doi: 10.1038/nbt.3988.

Michael Sun, Weize Yuan, Gang Liu, Wojciech Matusik, and Marinka Zitnik. Protein structure tokenization via geometric byte pair encoding. In *International Conference on Learning Representations*, 2026.

Peize Sun, Yi Jiang, Shoufa Chen, Shilong Zhang, Bingyue Peng, Ping Luo, and Zehuan Yuan. Autoregressive model beats diffusion: Llama for scalable image generation. *arXiv preprint arXiv:2406.06525*, 2024.

Yang Tan, Mingchen Li, Bingxin Zhou, Bozitao Zhong, Lirong Zheng, Pan Tan, Ziyi Zhou, Huiqun Yu, Guisheng Fan, and Liang Hong. Simple, efficient, and scalable structure-aware adapter boosts protein language models. *Journal of Chemical Information and Modeling*, 64(16):6338–6349, 2024.

Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

M. van Kempen, S. S. Kim, C. Tumescheit, et al. Fast and accurate protein structure search with Foldseek. *Nature Biotechnology*, 42:243–246, 2024. doi: 10.1038/s41587-023-01773-0.

M. Varadi, S. Anyango, M. Deshpande, et al. AlphaFold protein structure database: massively expanding the structural coverage of protein-sequence space with high-accuracy models. *Nucleic Acids Research*, 50(D1):D439–D444, 2022. doi: 10.1093/nar/gkab1061.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, pp. 5998–6008, 2017.

Oriol Vinyals, Lukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, and Geoffrey Hinton. Grammar as a foreign language. In *Advances in Neural Information Processing Systems*, volume 28, 2015.

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, et al. SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature Methods*, 17:261–272, 2020.

Christine Vogel, Matthew Bashton, Nicola D. Kerrison, Cyrus Chothia, and Sarah A. Teichmann. Structure, function and evolution of multidomain proteins. *Current Opinion in Structural Biology*, 14(2):208–216, 2004.

Shih-Hsin Wang, Yuhao Huang, Taos Transue, Justin Baker, Jonathan Forstater, Thomas Strohmer, and Bao Wang. Towards multiscale graph-based protein learning with geometric secondary structural motifs. In *Advances in Neural Information Processing Systems*, 2025a.

Xinyou Wang, Zaixiang Zheng, Fei Ye, Dongyu Xue, Shujian Huang, and Quanquan Gu. DPLM-2: A multimodal diffusion protein language model. In *International Conference on Learning Representations*, 2025b.

Yuyang Wang, Jiarui Lu, Navdeep Jaitly, Josh Susskind, and Miguel Angel Bautista. SimpleFold: Folding proteins is simpler than you think. *arXiv preprint arXiv:2509.18480*, 2025c.

Joseph L. Watson, David Juergens, Nathaniel R. Bennett, Brian L. Trippe, Jason Yim, Helen E. Eisenach, Woody Ahern, Andrew J. Borst, Robert J. Ragotte, Lukas F. Milles, et al. De novo design of protein structure and function with RFdiffusion. *Nature*, 620:1089–1100, 2023.

D. B. Wetlaufer. Nucleation, rapid folding, and globular intrachain regions in proteins. *Proceedings of the National Academy of Sciences of the United States of America*, 70(3):697–701, 1973. doi: 10.1073/pnas.70.3.697.

Jason Yim, Brian L. Trippe, Valentin De Bortoli, Emile Mathieu, Arnaud Doucet, Regina Barzilay, and Tommi Jaakkola. SE(3) diffusion model with application to protein backbone generation. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 40001–40039. PMLR, 2023.

Jason Yim, Andrew Campbell, Emile Mathieu, Andrew Y. K. Foong, Michael Gastegger, José Jiménez-Luna, Sarah Lewis, Victor Garcia Satorras, Bastiaan S. Veeling, Frank Noé, et al. Improved motif-scaffolding with SE(3) flow matching. *Transactions on Machine Learning Research*, 2024.

Chaohao Yuan, Songyou Li, Geyan Ye, Yikun Zhang, Long-Kai Huang, Wenbing Huang, Wei Liu, Jianhua Yao, and Yu Rong. Annotation-guided protein design with multi-level domain alignment. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, volume 1, pp. 1855–1866. ACM, 2025a.

Xinyu Yuan, Zichen Wang, Marcus D. Collins, and Huzefa Rangwala. Protein structure tokenization: Benchmarking and new recipe. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 73645–73670, 2025b.

Yang Zhang and Jeffrey Skolnick. Scoring function for automated assessment of protein structure template quality. *Proteins: Structure, Function, and Bioinformatics*, 57(4):702–710, 2004. doi: 10.1002/prot.20264.

Yang Zhang and Jeffrey Skolnick. TM-align: a protein structure alignment algorithm based on the TM-score. *Nucleic Acids Research*, 33(7):2302–2309, 2005.

A. Overview of generative biology

Proteins are biological macromolecules made of amino acids linked in a sequence. A protein sequence draws on 20 common amino acids and, under suitable conditions, folds into a three-dimensional structure (Anfinsen, 1973). Function depends on both amino-acid identity and the spatial arrangement and interactions of residues. The sequence specifies the components; the structure shows how they are organized. Each residue contains the backbone atoms N, Cα, C, and O, together with an amino-acid-specific side chain. Because Cα connects the backbone to the side chain, its coordinates are commonly used to represent residue position. Many structure-generation methods therefore model a backbone as a sequence of Cα coordinates (Geffner et al., 2025; Dilip et al., 2026a). This representation captures the overall fold and backbone topology, but omits the complete backbone, side chains, and atom-level chemical interactions. Geometric plausibility alone does not establish stability or function.

Protein structure has several levels of organization. A domain lies between local secondary structure and the complete protein. It typically comprises residues that contact one another, form a relatively compact unit, and tend to fold somewhat independently (Wetlaufer, 1973). Domains can also serve as structural or functional modules reused during evolution. Multidomain proteins combine domains with connecting regions; their function also depends on interdomain interfaces and relative positions (Vogel et al., 2004). Domain boundaries depend on criteria such as structural compactness, residue contacts, and evolutionary relationships. CATH classifies domains by structure and homology (Orengo et al., 1997), while TED extends automated domain identification to the large collection of predicted structures in AFDB (Lau et al., 2024). CATH is named for its first four classification levels: Class describes secondary-structure composition; Architecture describes overall spatial arrangement without connectivity; Topology also accounts for how secondary structures connect; and Homologous superfamily groups domains thought to share a common ancestor. Classification codes follow these C–A–T–H levels. A CATH domain ID identifies a specific domain by a four-character PDB identifier, a chain identifier, and a two-digit domain number.

Protein generation can target sequences or structures; here we study structure generation. Folding predicts structure from sequence, inverse folding designs sequences that might realize a given structure, and structure generation samples new three-dimensional backbones from a learned protein distribution (Watson et al., 2023; Yim et al., 2023). Generation may be unconditional or guided by chain length, secondary structure, or functional requirements. The objective is a geometrically plausible new backbone that a real sequence might realize. Visual inspection can reveal chain breaks or abnormal extension, but it cannot establish whether any sequence would fold into the generated conformation. A common evaluation therefore uses ProteinMPNN to design sequences for a generated backbone and ESMFold to refold them (Dauparas et al., 2022; Lin et al., 2023; Yim et al., 2023). Geometric agreement between the refolded structure and generated backbone estimates designability. This computational proxy does not measure stability or function experimentally, and individual generated examples provide qualitative evidence.

B. Inference Pipeline and Hyperparameters

DPAR generates protein backbones in two stages. An autoregressive model produces structure tokens and region boundaries, then a conditional flow decoder recovers a continuous Cα backbone. During autoregressive sampling, a KV cache reuses the keys and values computed for the existing prefix. Each step computes only the new token state, avoiding repeated processing of the full prefix. Before flow decoding, prediction entropy selects which generated token content enters the decoder. The decoder integrates geometric information across residues through repeated full-chain updates.

B.1 Autoregressive Generation

Given a target residue length L_{target}, the autoregressive prior generates structure tokens and generic region-boundary symbols from N to C. A finite-state grammar rules out empty domains, unclosed domains, and incorrect sequence lengths. It masks EOS until the target number of structure tokens has been generated. If generation is still inside a domain at the target length, it forces a domain-end symbol before ending the sequence. Sampling applies grammar and length masks, temperature scaling, top-p, min-p, and multinomial sampling, in that order. Inference caches the computed

keys and values of each Transformer layer instead of recomputing the entire prefix at every step. With identical inputs and random states, caching and full recomputation produce identical token sequences for each sample.

B.2 Segment Conditions

After generating the full AR sequence, DPAR selects a conditional prefix for each contiguous domain or residual segment. It computes $H_j = -\sum_k p_j(k) \log p_j(k)$ from the predictive distribution at each generation position (Shannon, 1948), using temperature $\tau_H = 0.7$; p_j is defined over valid categories. Only structure-token entropy is mapped to the residue axis.

For a segment of length $n_s \geq 10$, DPAR fits a `UnivariateSpline` (Virtanen et al., 2020) with smoothing parameter $0.08n_s$. It uses the first local minimum of the smoothed entropy curve within $8 \leq i \leq n_s - 2$ as the cutoff and retains only the preceding token content. If no usable minimum exists, or if the segment has at most 9 residues, it retains the full segment. If spline fitting fails, it uses a five-point moving average with edge padding. Prefix selection leaves AR generation, token positions, chain length, and region boundaries intact. Section B.3 describes how the decoder uses the selected condition with annealed guidance.

B.3 Conditional Flow Decoding

Flow decoding maps the selected FSQ IDs to decoder conditions and starts from centered Gaussian coordinates. The main generation configuration uses guidance weight $w_0 = 1$ and annealing exponent $\alpha = 0.25$:

$$\mathbf{v}_g = \mathbf{v}_u + (1 - t^{0.25})(\mathbf{v}_c - \mathbf{v}_u).$$

The conditional and zero-content branches retain the same residue positions and region layout, differing only in structural token content. We obtain the guided score $\mathbf{s}_g$ from $\mathbf{v}_g$ using Eq. (7) and integrate the drift

$$\mathbf{b}_g = \mathbf{v}_g + 2\frac{1-t}{t+0.01}\mathbf{s}_g.$$

We use Euler updates on a uniform grid of 100 time points, with coordinate re-centering and self-conditioning along the trajectory. No additional per-step noise is injected.

B.4 Model Architecture

Tokenizer			
Architecture	**Value**	**Training**	**Value**
Encoder channels	256	Batch size	4×16 rotations
Encoder layers	2	Learning rate	2×10^{-5}
Encoder heads	8	Min LR	5×10^{-6}
Flow decoder channels	512	Warmup	500
Flow decoder layers	12	AdamW β_1	0.9
Flow decoder heads	8	AdamW β_2	0.999
FSQ levels	8, 8, 8, 8	Weight decay	0.01
Codebook size	4,096	Grad clip	1.0
Max tokens	256	Micro steps	1
Parameters	43,845,964	λ_{D}	0.05

Autoregressive Model			
Architecture	**Value**	**Training**	**Value**
Layers	16	Batch size	64
Hidden dim.	1024	Learning rate	3×10^{-4}
FFN dim.	4096	Min LR	3×10^{-5}
Attention heads	16	Warmup	2,000
Positional encoding	2D RoPE	AdamW β_1	0.9
Pair-bias channels	64	AdamW β_2	0.999
Max tokens	263	Weight decay	0.1
Parameters	210,469,988	Grad clip	1.0
		Micro steps	1

B.5 CATH-Conditioned Generation

Conditional and unconditional generation use the same tokenizer. For conditional training, we use 97,511 single-domain proteins whose domains are annotated in TED and assigned CATH IDs, retaining the source-level data split and restricting resampling to the training set. We train a separate autoregressive model that receives the target TED–CATH ID from the beginning of each sequence, allowing it to guide even residues generated before the domain begins. The same conditioning mechanism can in principle use CATH IDs at any classification level; the experiment in Table 3 uses C-level IDs. Generation follows the unconditional pipeline, except that the generic domain boundary symbols are replaced by `DOMAIN_BOS(CATH ID)` and `DOMAIN_EOS`. During sampling, grammar constraints require the domain-begin symbol to match the requested ID and the domain to close before the chain ends. The generated tokens are decoded with the existing decoder, without training a CATH-specific decoder.

C. Training Data

C.1 Data Sources and DFS Construction

Early protein-generation models drew mainly on approximately 30,000 nonredundant structural homology classes in the Protein Data Bank (PDB). Their training scale was constrained by the number of experimentally determined structures (Crystallography: Protein Data Bank, 1971). Large collections of AlphaFold2-predicted structures have since expanded the available training data. Proteina's data construction, for example, involved approximately 214 million predicted structures in AFDB, while AlphaFold3 used approximately 40 million predicted structures from MGnify (Abramson et al., 2024; Dilip et al., 2026b; Geffner et al., 2025). Training directly at that scale demands substantial storage and compute resources. We therefore use the Foldseek-clustered AlphaFold Database (DFS) as our base training set.

DFS contains nonredundant representatives from AFDB (Varadi et al., 2022). MMseqs2 first reduces sequence redundancy, Foldseek then clusters structures by three-dimensional similarity, and one representative is retained per cluster (Steinegger & Söding, 2017; van Kempen et al., 2024). Proteina and Genie2 also use DFS (Geffner et al., 2025; Lin et al., 2024). After further quality control and test-set decontamination, we retain 400,936 structures for training.

C.2 Data Preprocessing

We apply the following filters to protein chains in DFS:

1. Remove chains in which more than 70% of residues are coil, to reduce samples with an overly uniform secondary-structure composition;
2. Remove structures with a chain-level mean pLDDT below 80;
3. Retain only chains in which at least 80% of residues have pLDDT above 70, to exclude predicted structures with extensive local low-confidence regions;

4. Remove from the training set all samples structurally homologous to evaluation chains. The evaluation sets comprise an independently filtered CAMEO, CASP14, CASP15, and a subset of CATH.

The first two filters screen overall structural quality, while the third limits local low-confidence regions. The fourth reduces the risk of homology leakage between training and evaluation data. Earlier protein-generation work has applied similar quality filters and test-set decontamination principles (Bose et al., 2024; Dilip et al., 2026b).

C.3 Domain Annotation

The model takes Cα coordinates of valid residues as input, converts them to nm, and centers them at the geometric center of those residues. We obtain domain annotations from The Encyclopedia of Domains (TED) (Lau et al., 2024), which provides structural-domain boundaries for protein models in the AlphaFold Database and CATH classifications where available. Domain annotations are mapped to residue intervals in each chain. Highly overlapping intervals are merged, and the canonical label follows the priority order domain, repeat, family, then superfamily. Domains are ordered from N to C, with at most three selected per chain. All remaining valid residues, including linkers and termini, are assigned to the residual region.

The exported dataset contains 29,641, 284,758, 80,507, and 6,030 chains with zero, one, two, and three selected domains, respectively. Thus, 78.4% of the 400,936 chains have at most one selected domain.

D. Evaluation Metrics

For protein unconditional generation, we sample 10 proteins of each length between 60-128, resulting in 690 samples in total. (Huguet et al., 2024).

Designability Designability asks whether an amino-acid sequence designed for a backbone refolds into a consistent structure. Following FrameDiff, we use ProteinMPNN to generate eight sequences per backbone at sampling temperature 0.1, then predict their structures with ESMFold (Lin et al., 2023; Yim et al., 2023; Dauparas et al., 2022). We compute the root-mean-square deviation (RMSD) between each predicted structure and the original backbone. The lowest value for a backbone is its self-consistency RMSD (scRMSD). A sample is considered designable when scRMSD is $\leq$2 Å, and the designability score is the proportion of samples meeting this criterion. We report both designability and scRMSD. Under this computational criterion, only 60% of natural proteins are designable, although their scRMSD is approximately 0.90 Å; the two measures therefore convey different aspects of backbone quality.

Diversity Following the first of the two diversity measures used by Bose et al. (2024), we compute pairwise TM-scores among all designable samples at each specified protein length (Zhang & Skolnick, 2004). We average the pairwise scores within each length and then average the length-wise means to obtain the diversity value. TM-score ranges from 0 to 1, with higher scores indicating greater structural similarity. Lower mean TM-scores therefore indicate greater structural diversity.

Novelty We use Foldseek to search against the PDB dataset and return the maximum TM alignment. The novelty score is the mean of these maximum TM-scores over all designable samples. Lower scores are considered better, as they suggest the generated structures are less similar to known structures in the reference sets, implying higher novelty.

Domain RMSD For structural reconstruction on CATH, we compute Domain RMSD using the annotated domain spans. For each domain, we extract the corresponding Cα coordinates from the reconstructed and reference structures, align the two domain fragments independently by an optimal rigid-body transformation, and calculate the RMSD over their aligned residues. This per-domain alignment measures reconstruction error within the domain without including changes in its position or orientation relative to the full chain.

E. Sources and Use of Baseline Models

APT. The public APT repository currently provides only tokenizer and language model checkpoints trained on sequences of at most 128 tokens. Therefore, the APT performance in the unconditional generation table (Table 2) in the main text is taken from the original paper.

Kanzi. For the structure reconstruction comparison, we used the public Cα tokenizer implementation and pretrained weights, reconstructed the test backbones through their encoding–decoding procedure, and computed the reconstruction metrics. The authors have not released the generative model weights. For the unconditional generation comparison, we cite the Kanzi-AR results reported in the original paper, corresponding to a 250M autoregressive model, $\eta = 0.66$, and best-of-2 candidate selection.

DPLM-2. Our experiments used the official implementation and public weights released. For DPLM-2, we used a structure temperature annealing range from 2.0 to 0.1, a random unmasking probability of 1.0, a temperature of 1.0, and a maximum of 500 iterations for generation.

ESM3. Our experiments used its official open implementation and the open-weight ESM3-open model. For ESM3, we used `GenerationConfig(track="structure")` to report structures, with default settings of temperature 1.0, cosine scheduling, a random strategy, temperature annealing enabled, and top-p = 1.0. We did not observe a significant improvement in designability from increasing the number of steps. We report results at 32 and 256 steps; in the latter setting, one token is revealed at each step.

FoldToken4. We obtained FoldToken4 from the publicly available FoldToken_open repository and followed the described reconstruction procedure.

IST. We used the 4k-codebook tokenizer and followed exactly the encoding and decoding procedures described in the protein-structure-tokenizer repository.

F. Extended Results

Domain–Global Structural Consistency. The scatter points in Figure 1 are obtained from APT, DPAR, DPLM-2, and ESM3 (Dilip et al., 2026a; Wang et al., 2025b; Hayes et al., 2025) under their respective generation protocols, with all samples processed using the same domain segmentation and evaluability criteria. The domain structure error W is computed by independently aligning each domain between the input backbone and its best refold using Kabsch alignment (Kabsch, 1976), and then aggregating the squared alignment errors weighted by domain length:

$$W = \sqrt{\frac{\sum_k \sum_{i \in \text{domain } k} \|R_k x_i + t_k - y_i\|_2^2}{\sum_k n_k}}. \tag{9}$$

Here, n_k is the number of residues in domain k, and unassigned residues are excluded. Independent domain alignment removes relative translations and rotations between domains. The relative placement error P is obtained by comparing the error after jointly aligning all domain residues with the within-domain error:

$$P = \sqrt{\max\left(0,\ \text{MSE}_{\text{joint}} - W^2\right)}. \tag{10}$$

Thus, W captures structural errors within domains, whereas P captures errors arising from their relative arrangement. Pearson correlation was computed on the original W and scRMSD values pooled across all models. Figure 5 presents a quantitative comparison of the median W and P values across models.

Generated Layouts and Structural Segmentation. To test whether the domain routing intervals generated by the AR model correspond to distinct regions in three-dimensional structure, we compared AR intervals in DPAR-generated samples with post hoc domains assigned by Merizo (Lau et al., 2023). Merizo was given the ESMFold-refolded structure with the lowest scRMSD among

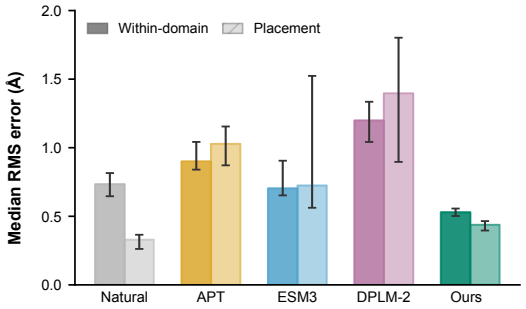


Figure 5: Quantitative comparison of median within-domain errors (W) and inter-domain placement errors (P) across APT, ESM3, DPLM-2, and DPAR.

eight designed sequences. Among samples with AR domains, the mean IoU of the unions of domain residues was 0.691, and 71.61% of AR domains had an IoU of at least 0.5 with at least one Merizo domain. These results indicate a coarse correspondence between generated routing intervals and post hoc structural partitions, but we emphasize that AR boundaries cannot be treated as precise biological domain boundaries; Merizo itself is also only a reference for structural segmentation.

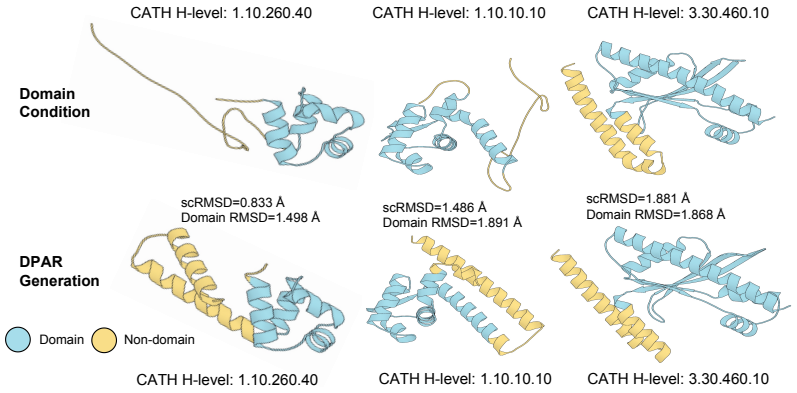


Figure 6: Domain-conditioned protein generation. The top row shows the natural protein domain conditions, and the bottom row shows the corresponding DPAR-generated proteins. In all three examples, the domain of the generated protein has the same CATH H-level classification as the corresponding natural protein. Whole-chain scRMSD and domain RMSD are annotated above the generated structures.

Scaffolding Conditioned on a Natural Domain. To demonstrate the use of domain tokens as local conditioning, we used experimentally determined single-domain proteins with official CATH annotations as input (Orengo et al., 1997). We retained the natural domain's complete discrete tokens, had the AR model generate a C-terminal region of a preset length after it, and then reconstructed the whole-chain backbone with the conditional flow decoder. This experiment used an annealing exponent of $\alpha = 0.5$ and is reported separately from the main unconditional-generation configuration. We report whole-chain scRMSD and the Cα RMSD of the natural domain after independent alignment with the original domain. Domain RMSD measures local geometric deviation, but our main interest is whether the refolded domain retrieves a reference with the same CATH classification as the conditioned domain. To assess this, we cropped ESMFold-refolded structures using the oracle boundaries of the conditioned CATH domains and searched them against the CATH v4.3.0 S40 reference set using Foldseek, followed by TM-align reranking based on query-normalized TM-score (van Kempen et al., 2024; Zhang & Skolnick, 2005). We then compared the CATH classification of the Top-1 reference structure with that of the conditioned domain at the C, C.A, C.A.T, and C.A.T.H levels. We emphasize classification agreement, particularly at the H level, over minimizing domain RMSD alone, because it reflects structural and evolutionary relationships relevant to domain identity

and potential function. Figure 6 shows the three protein examples in this exploratory experiment: 2l49_B, 2f2e_A, and 6fgj_A. These cases illustrate C-terminal extension conditioned on a natural domain while maintaining agreement with its CATH H-level classification after refolding.