TOWARDS MULTISTATE PROTEIN DESIGN WITH ALL-ATOM GENERATIVE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Designing proteins that adopt defined conformations in different molecular contexts remains a central challenge in protein design. Here, we ask whether all-atom generative models can learn such context dependence and use it for multistate design. We first develop AtomDiff-X, a streamlined model for end-to-end all-atom protein design, and experimentally validate its designs. We then curate a large dataset of chameleon sequences, 6–11-residue segments that adopt different secondary structures in different protein contexts. To learn from these paired structural alternatives, we introduce a dual-track attention mechanism and fine-tune AtomDiff-X to reconstruct paired protein structures. The resulting model, AtomDiff-Chameleon, generates de novo chameleon sequences up to 48 residues long together with their structural contexts and achieves the best performance in an in silico benchmark. We further apply AtomDiff-Chameleon to design pairs of proteins with identical N-termini but different C-termini, yielding distinct structures that mimic alternative splicing to switch protein function. These results indicate that sequence–structure ambiguity in natural proteins provides scalable supervision for learning context-dependent conformations. Combined with end-to-end all-atom generation, this supervision offers a promising data-driven route towards multistate protein design.

1 INTRODUCTION

Computational protein design aims to create amino acid sequences that adopt specified structures and perform desired functions. Recent generative models have substantially increased the success rate of de novo protein design (Watson et al., 2023; Krishna et al., 2024). Most successful designs, particularly protein binders, are optimized for a single conformation in order to maximize their folding stability. Many natural proteins, however, populate multiple conformational states and shift among them in response to their changing molecular environment (Burmann et al., 2012; Porter & Looger, 2018). Designing sequences with controllable conformational equilibria would enable responsive systems for biosensing and signal transduction. Extending current generative models to support this form of multistate design remains a major challenge.

Under Anfinsen's thermodynamic hypothesis, a protein's native state is determined by its sequence in a given environment (Anfinsen, 1973). Single-state design can therefore be framed as generating an amino acid sequence with a conformational distribution concentrated around one target structure. Multistate design instead seeks a shared sequence that stabilizes each intended conformation in its corresponding context, maintains the desired balance between states, and disfavors unintended alternatives. In probabilistic terms, the model must learn both the joint distribution of protein sequence and structure and how molecular context shifts the conformational distribution of a shared sequence. This formulation reveals two coupled challenges of developing a generative model for multistate protein design: representing the totality of a protein's interatomic interactions without committing to a single state, and learning context–conformation relationships from data.

Current protein design workflows generate a protein backbone and then design sequences for that backbone. This decomposition has enabled highly successful single-state designs. We find it less suited to multistate design, where backbone and side-chain interactions jointly shape the alternative conformations. In order to address the first challenge on representation, we develop AtomDiff-X, a streamlined end-to-end all-atom generative model that does not separate backbone and side-chain

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.

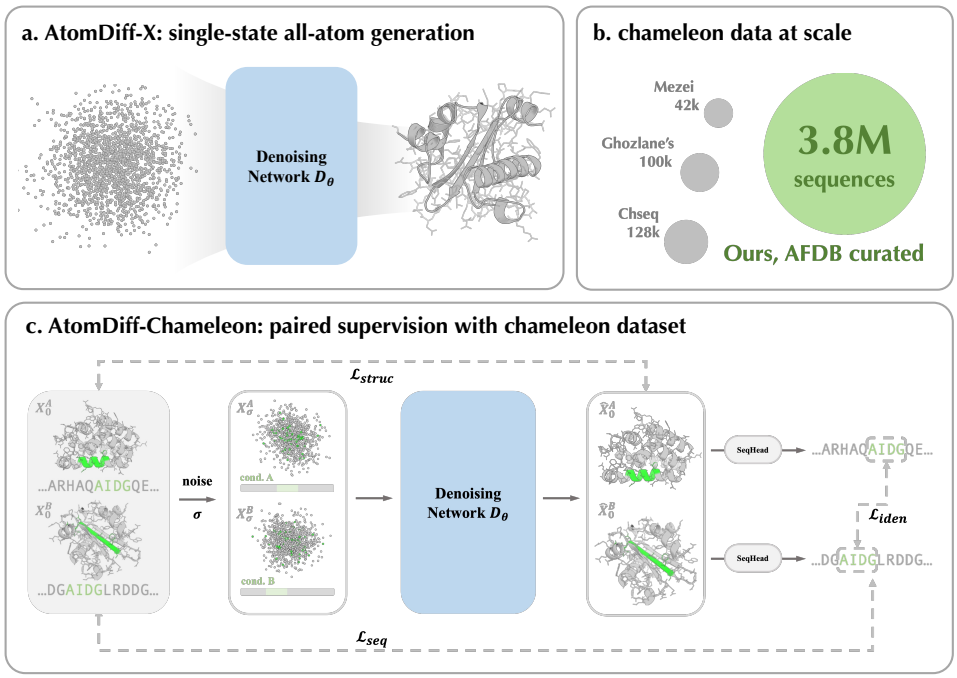


Figure 1: **From single-state all-atom generation to paired chameleon design. (a)** AtomDiff-X performs unconditional all-atom generation by denoising atom14 coordinates. **(b)** AFDB-derived chameleon data provide paired supervision, comprising 3.775 million distinct sequences and 11.76 million pairs. **(c)** AtomDiff-Chameleon extends pretrained AtomDiff-X through paired training, jointly denoising two structures under residue-correspondence and optional secondary-structure conditions. For structure and sequence depictions, green color marks the shared segments with the surrounding proteins in gray. Solid lines and arrows show information flow; dashed ones indicate training supervision on structure, sequence and cross-branch agreement with labeled loss terms.

generation (Figure 1a). We establish its ability to generate well-folded proteins and validate selected designs experimentally, providing a solid foundation for a multistate extension.

The critical bottleneck for the second challenge is a lack of paired training data (Jing et al., 2026). Learning context-dependent conformations requires examples that associate the same sequence with different structures in different environments. The key insight we start with is that natural chameleon sequences offer a scalable source of such supervision. These shared segments adopt different secondary structures when embedded in different protein scaffolds (Minor Jr & Kim, 1996; Li et al., 2015), so each scaffold defines a distinct structural context for the same local sequence. By matching these segments across structures, we convert otherwise single-state structural data into paired examples of context-dependent conformations (Figure 1b).

We fine-tune AtomDiff-X into AtomDiff-Chameleon using these paired examples (Figure 1c). The key innovation is a novel dual-track attention architecture that coordinates the shared segment across the two structural contexts provided by each pair. This enables AtomDiff-Chameleon to generate de novo chameleon sequences in distinct contexts, achieving the best performance in our in silico benchmark. To demonstrate its utility, we apply AtomDiff-Chameleon to design protein pairs that share an identical N-terminal functional motif but fold into different structures due to differing C-terminal sequences — mimicking alternative splicing to produce variants with switchable functions. Together, the all-atom generative model, the paired chameleon datasets, and the context-aware architecture form a practical framework for data-driven multi-state protein design.

2 Related Work

Generative modeling for protein design. Early structure-based generative models such as RFdiffusion (Watson et al., 2023) generate protein backbones, with compatible sequences subsequently

assigned by inverse-folding models (Dauparas et al., 2022). All-atom generative models attempt to design both protein backbones and sidechains, and are faced with the problem of representing different amino acids without knowing the protein sequence. Protpardelle (Chu et al., 2024) addresses this problem through a superposition of possible side chains, coupled with sequence design during sampling. Latent approaches offer another route: PLAID (Lu et al., 2026) generates compressed representations from a pretrained folding model, while La-Proteina (Geffner et al., 2026) combines explicit C_α coordinates with residue-level latents encoding sequence and the remaining atomic geometry. Pallatom (Qu et al., 2025) uses an `atom14` representation with virtual-atom padding, enabling generation in a fixed-dimensional coordinate space from which residue identities can be recovered. Our current work adopts the `atom14` representation and develops an all-atom diffusion architecture, AtomDiff-X. We further extend it to build AtomDiff-Chameleon.

Multistate protein design. Recent deep learning methods approach multistate protein design by coordinating sequence constraints across multiple structural states. Those methods differ in both their training strategy and sequence coordination mechanism. DynamicMPNN (Abrudan et al., 2026) extends the fixed-backbone sequence design model for multistate sequence design. It is trained on a limited set of paired structural data to learn the sequence distribution compatible with multiple input backbones. A second line of work adapts models trained on single-state structural data through inference-time sequence coordination. ProteinGenerator (Lisanza et al., 2025) performs diffusion in sequence space and coordinates states by averaging sequence logits across trajectories subject to different structural constraints. ProDiT (Jing et al., 2025) jointly models sequence and structure diffusion, coupling separately conditioned structure-denoising trajectories through a shared sequence-denoising trajectory. A third approach optimizes sequences through pretrained structure prediction models. SwitchCraft (Jing et al., 2026) backpropagates through Boltz-1 (Wohlwend et al., 2025) to optimize a shared sequence with structural constraints representing different molecular contexts. In contrast, our work finetunes an all-atom generative model directly on a large set of chameleon pairs to jointly generate the shared segments and the surrounding scaffolds. The sequence coordination is learned through a dual attention mechanism during training.

3 METHOD

3.1 ATOMDIFF-X: AN ALL-ATOM GENERATIVE MODEL

Representation and denoising setup. We model all the heavy atoms in a protein in 3D Cartesian space. An L-residue protein is represented as $X_0 \in \mathbb{R}^{L\times 14\times 3}$, following the `atom14` formulation from Pallatom (Qu et al., 2025) . Each protein residue is uniformly represented by 14 atoms, with the unused slots filled by virtual atoms placed at its C_α position. We adopt the EDM preconditioning framework (Karras et al., 2022) and predict clean coordinates as

$$\hat{X}_0 = c_{\text{skip}}(\sigma)X_\sigma + c_{\text{out}}(\sigma)F_\theta(c_{\text{in}}(\sigma)X_\sigma, e(\sigma)) \tag{1}$$

This full preconditioned denoiser is referred to as D_θ in Figure 1a and 1c. We maintain residue-level token representations and atom representations throughout the denoising process. The denoising network consists of a residue-level Token DiT and an atom-level Atom DiT described below.

Token DiT and Atom DiT. Inspired by PixelDiT (Yu et al., 2026), we adopt a dual-level architecture for scalable all-atom generation with unidirectional token-to-atom information flow. Token DiT aggregates structural information at the residue level: token features h_i are initialized by applying an atom-wise MLP to normalized noisy coordinates and then mean-pooling within each residue, while pair features z_{ij} encode residue-level distances and relative residue positions. Each block couples h_i and z_{ij} through outer-sum updates, followed by pair-biased attention and a feed-forward layer. The final symmetrized pair features $z_{ij} + z_{ji}$ feed an auxiliary head that predicts a clean residue-level distogram. Atom DiT operates on atom features a_i initialized directly from noisy coordinates. The final token representations are projected and combined with the noise embedding to form residue-specific conditioning. Figure 6 shows the detailed architecture.

Training and sequence readout. We train AtomDiff-X with a combination of coordinate denoising, local geometry, residue identity, and residue-pair distance objectives:

$$\mathcal{L} = \mathcal{L}_{\text{coord}} + \lambda_{\text{lddt}}\mathcal{L}_{\text{smooth-lDDT}} + \lambda_{\text{seq}}\mathcal{L}_{\text{seq}} + \lambda_{\text{dist}}\mathcal{L}_{\text{dist}}. \tag{2}$$

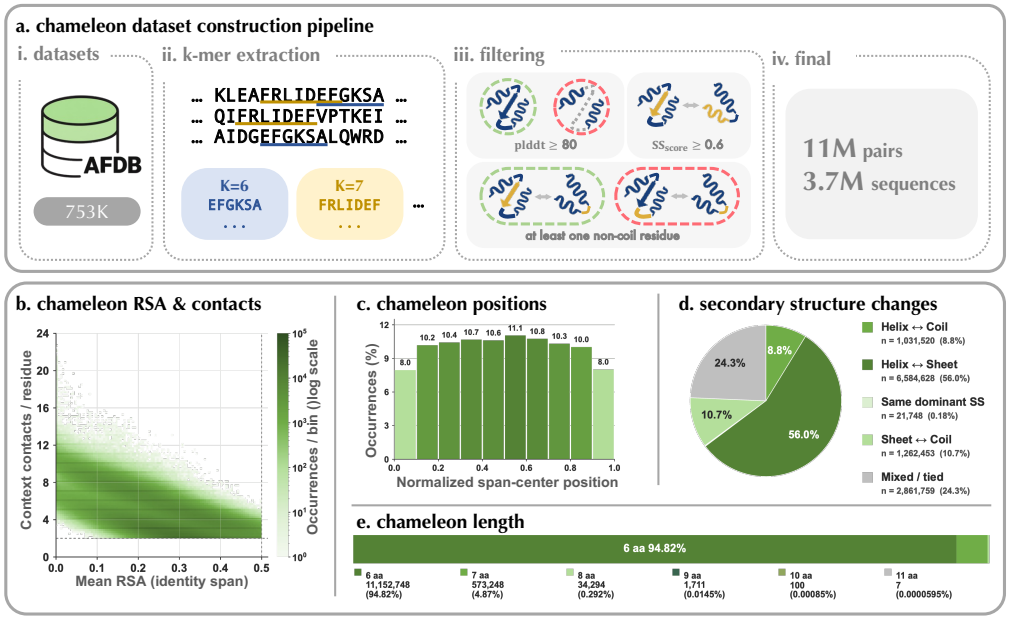


Figure 2: **Construction and statistics of the AFDB-derived chameleon dataset. (a)** Chameleon selection through exact k-mer matching and sequence/structure filtering. **(b)** Joint distribution of mean relative solvent accessibility (RSA) and nonlocal context contacts. **(c)** Distribution of normalized span-center position. **(d)** Paired secondary-structure categories. **(e)** Identity-span length distribution.

Here, $\mathcal{L}_{\text{coord}}$ is a weighted coordinate diffusion loss, $\mathcal{L}_{\text{smooth-lDDT}}$ is smoothed lddt loss, $\mathcal{L}_{\text{seq}}$ is sequence cross-entropy loss and $\mathcal{L}_{\text{dist}}$ is auxiliary distogram loss (Details in Appendix A.1.2).

A lightweight graph neural network predicts amino acid identities from `atom14` coordinates, effectively translating atom-wise positional information into chemical identity. This network is trained with both ground-truth and low-noise coordinates; the low-noise inputs enable gradients from sequence supervision to propagate into the coordinate denoising process.

3.2 Chameleon Pair Construction

While AtomDiff-X is trained on individual structures for simultaneous generation of protein backbones and sequences, for multistate protein design, we need paired supervision linking a shared sequence to distinct environments and its corresponding conformations. Our key idea is to define shared sequence at the fragment level and regard its surrounding protein scaffolds as the "environments". Natural chameleon segments share identical amino acid sequences but adopt distinct conformations across different protein environments. We set out to expand the current database of chameleon pairs to enable large-scale training.

Dataset construction. We start with the AlphaFold Protein Structure Database (AFDB), and search for exact matches of 6-, 7-, and 8-residue fragments. Each seed match is extended bidirectionally to a maximal contiguous identical span before subsequent filtering. Figure 2 summarizes the construction pipeline and the key statistics. Candidate pairs are filtered by their prediction confidence, secondary-structure dissimilarity, solvent accessibility, and whole-chain sequence similarity. Our final dataset contains 11,762,108 pairs spanning 3,774,934 distinct sequences (Figure 1b), which is approximately 29 times that reported by Li et al. (2015) (128,703 sequences). Details are provided in Appendix A.2.

3.3 AtomDiff-Chameleon: Paired Supervision and Generation

Using the curated chameleon dataset (Method 3.2), we extend the basic AtomDiff-X architecture (Method 3.1) to learn the context-dependent conformational changes. The new model, AtomDiff-Chameleon, is fine-tuned to jointly denoise two protein structures with chameleon labels and optional

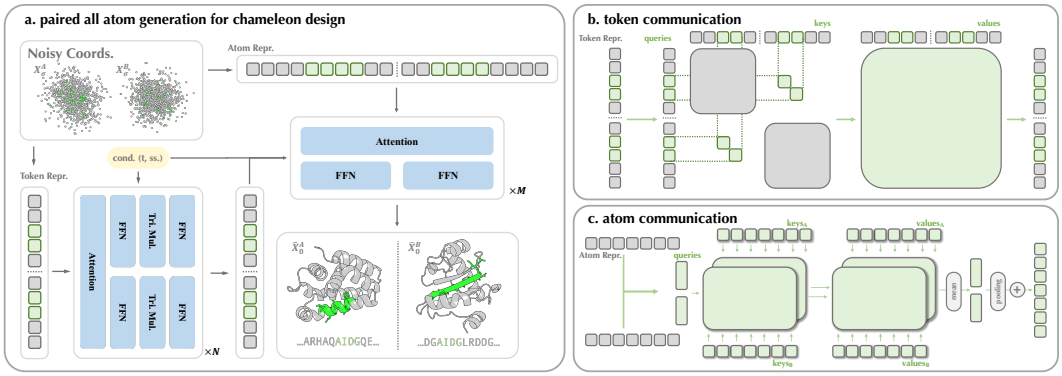


Figure 3: **AtomDiff-Chameleon for paired all-atom protein generation. (a)** Two structures are jointly denoised with specified residue correspondences and optional secondary-structure conditions. Token and atom representations are depicted as small squares and their matrix computation as big squares or rectangles; green color marking the shared regions and the corresponding information exchange. **(b)** Token attention connects all residues within each branch and corresponding shared residues across branches. **(c)** At atom-level, a shared query reads each residue's local atom features, and the resulting messages are averaged and projected into a common residual update for both residues.

secondary-structural annotations (Figure 1c). During inference, AtomDiff-Chameleon generates a chameleon segment and two distinct proteins as its contexts.

Cross-branch communication. AtomDiff-Chameleon jointly denoises two structures through coupled Token DiT and Atom DiT computations (Figure 3a). For a training pair X_0^A and X_0^B, let $\mathcal{P} = \{(i, \pi(i))\}$ denote the residue correspondences within the shared segments, their sequences satisfy $s_i^A = s_{\pi(i)}^B$ for all $(i, \pi(i)) \in \mathcal{P}$. Communication at both levels encourages agreement in residue identity while allowing distinct local conformations.

In Token DiT (Figure 3b), attention retains full connectivity within each diffusion branch and adds bidirectional connections between corresponding residues specified by $\mathcal{P}$. These connections allow corresponding tokens to exchange information from their respective protein contexts, while pair representations and triangle updates remain within each branch. In each Atom DiT block (Figure 3c), a symmetric residue-consensus pathway first constructs a query from each residue's pooled atom features and averages the queries of corresponding residues. Each branch uses this shared query to attend to its own residue's 14 atom features, producing a local message. The two messages are then averaged, and the resulting message is broadcast to all 14 atom slots and injected into the atom representations. This pathway operates alongside within-branch atom attention: the branches share the consensus update while retaining separate atomic representations throughout denoising. Details are provided in Appendix A.3.2.

Secondary-structure conditioning. To specify conformational preferences of the generated chameleon sequences, we provide per-residue labels $c_i^b \in \{H, E, C, \text{NULL}\}$ for each diffusion branch $b \in \{A, B\}$, where H, E, and C denote helix, strand, and coil, respectively, and NULL leaves the position unspecified. A binary indicator marking the chameleon positions (defined by $\mathcal{P}$), and the secondary-structural (SS) specifications (defined by c_i^b) are combined with diffusion-time embedding to modulate Token DiT and Atom DiT through AdaLN. As a strategy of diversifying the training samples, we randomly vary the positions and extent of SS specifications, ranging from the full specification to an all-NULL condition. The conditioning and label-sampling procedures are detailed in Appendix A.3.3.

Training objective. For fine-tuning AtomDiff-Chameleon, we add three sequence-identity loss terms to the AtomDiff-X single-state training objective (Method 3.1) denoted as $\mathcal{L}_{\text{base}}$ in Equation 3. $\mathcal{L}_{\text{message-ID}}$ supervises branch-specific atom features across Atom DiT blocks to encourage those representations to encode residue identity. $\mathcal{L}_{\text{shared-JS}}$ calculates the Jensen–Shannon divergence

between the corresponding chameleon positions, to encourage sequence agreement. $\mathcal{L}_{\text{shared-GT}}$ calculates the cross-entropy between predicted and the ground-truth chameleon sequences. The combined training objective is:

$$\mathcal{L}_{\text{paired}} = \mathcal{L}_{\text{base}} + \lambda_{\text{msg}}\mathcal{L}_{\text{message-ID}} + \lambda_{\text{cons}}\mathcal{L}_{\text{shared-JS}} + \lambda_{\text{gt}}\mathcal{L}_{\text{shared-GT}} \tag{3}$$

Loss definitions, masks, reductions, and noise-dependent weights are given in Appendix A.3.5.

Finetuning and chameleon generation. AtomDiff-Chameleon is finetuned on a mixture of individual proteins, identical pairs of individual protein structures, and chameleon pairs of different protein structures. For individual proteins and non-sharing residues in paired examples, the Token DiT and Atom DiT operate on single-branch features only. The cross-branch communication mechanism described above is applied to the identical sequences in the chameleon pairs. During training, we apply independent rigid-body augmentations to the two structures and corrupt them with Gaussian noise at a shared noise level σ, supervising each one with its clean structure. During generation, we initialize two diffusion branches with independent Gaussian coordinate noise and jointly denoise them with the conditioning control from chameleon positions and SS specifications. Detailed training and sampling settings are described in Appendix A.3.1, Appendix A.3.6 and Appendix A.4.1.

4 Experiments

We first evaluate AtomDiff-X's ability to design well-folded proteins by computational metrics and biochemical characterization (Experiment 4.1). We then examine AtomDiff-Chameleon on paired chameleon design and test its generalization to longer sequences (Experiment 4.2). Finally, we apply AtomDiff-Chameleon in motif-scaffolding tasks as a demonstration of switching protein structure and function (Experiment 4.3).

4.1 Unconditional all-atom generation and experimental characterization

Benchmark performance. Following the evaluation protocol in Geffner et al. (2026), we test AtomDiff-X's in silico performance by generating 100 proteins at each length in $\{100, 200, 300, 400, 500\}$, yielding 500 samples in total. We employ AtomDiff-X with two sampling strategies, Euler sampling and stochastic predictor–corrector (PC) sampling. We then fold the generated sequences with ESMFold (Lin et al., 2023) and compare the predicted structures with the generated samples. More details on sampler settings and metric definitions are given in Appendix A.4.

Table 1: **Unconditional all-atom generation.** Baselines are reported by Geffner et al. (2026). Diversity is measured by cluster counts; bold marks the largest values shown.

Method	Co-design (%) ↑	Clusters ↑ Str	Seq+Str	Seq
P(all-atom)	36.7	134	165	148
Protpardelle-1c	35.8	41	61	138
APM	19.0	32	59	64
PLAID	11.0	25	27	38
ProteinGenerator	9.8	12	24	28
Protpardelle	8.8	10	21	37
La-Proteina	**68.4**	**206**	**301**	**216**
Ours, Euler (step = 1.5)	54.0	52	91	144
Ours, stochastic PC	61.0	45	91	138

Table 2: **Protein Expression and FPLC-SEC.**

Outcome	All-atom designs	MPNN controls
Designs tested	19	5
Soluble, n (%)	14 (73.7)	3 (60.0)
No soluble expression	5	2
SEC: monomer	10	2
SEC: monomer + oligomer	2	0
SEC: oligomer	1	1
Sample lost	1	0

The results are reported in Table 1, with the baseline values taken from Geffner et al. (2026). AtomDiff-X ranks second in generation quality ("co-designability"), and the third in generation diversity ("clusters"). It achieves 54.0% all-atom co-designability with Euler sampling and 61.0% with stochastic PC sampling. The latter sampling strategy improves its co-designability by 7.0 percentage points, while its generation diversity drops slightly.

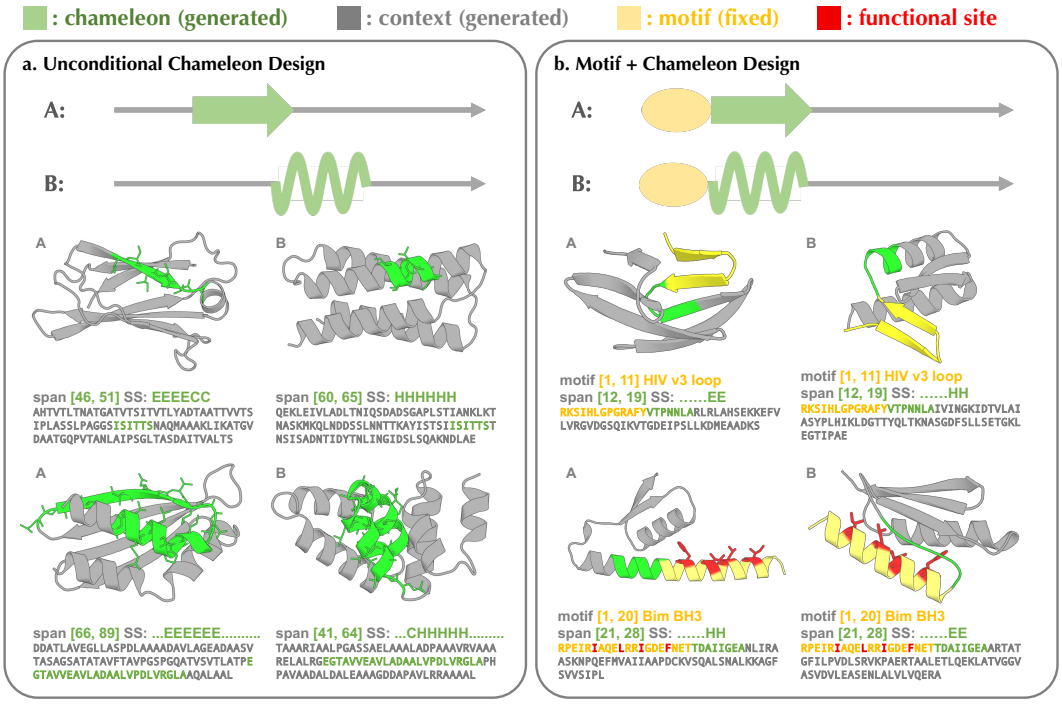


Figure 4: **Chameleon design examples.** For both structure and sequence depictions, chameleon segments are highlighted in green with the surrounding protein contexts in gray, motif in yellow and functional sites in red. Secondary-structure(SS) and motif conditions are listed under each structure. **(a)** Selected successful protein pairs with shared segment length of $k = 6$ (upper) and $k = 24$ (lower). **(b)** Motif-conditioned examples illustrating joint motif preservation and chameleon generation.

Experimental characterization. We selected 19 AtomDiff-X designs spanning 100–500 residues for biochemical characterization and included five additional ProteinMPNN-redesigned sequences on matched backbones for comparison (Dauparas et al., 2022). We performed protein purification and measured the designed proteins' oligomeric states in solution by fast protein liquid chromatography–size-exclusion chromatography (FPLC-SEC). Of the 19 all-atom designs, 14 (73.7%) showed soluble expression. Among these, 10 had SEC profiles consistent with monomeric species. Three of the five ProteinMPNN controls showed soluble expression, with two classified as monomeric (Table 2). Individual records and experimental protocol details are provided in Appendix A.6.

Both the benchmark and biochemical characterization support the conclusion that AtomDiff-X is capable of designing well-folded proteins without separating backbone and sequence generation.

4.2 De novo chameleon design and generalization

We set up a chameleon design benchmark to evaluate AtomDiff-Chameleon's in silico performance against ProteinGenerator (Lisanza et al., 2025). The benchmark assesses a model's ability to generate paired protein structures containing chameleon segments, with performance quantified through folding simulations.

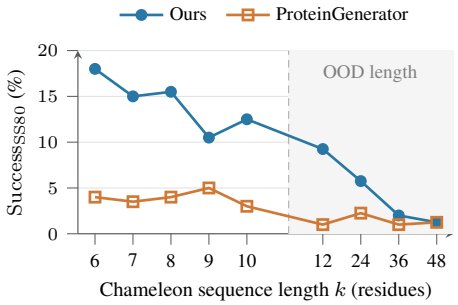
Benchmark setup. Each model generates 200 pairs of 96-residue proteins with chameleon length $k \in \{6, 7, 8, 9, 10\}$, yielding 1,000 pairs in total. For each length setting, we vary chameleon positions and its secondary-structure (SS) conditions. Chameleon positions are sampled independently in the two chains at random, leaving at least one flanking residue on either side. SS conditions are sampled from natural chameleon data, retaining mixed H/E/C patterns.

The individual sequences of the generated protein pairs are extracted and refolded by ESMFold. We define Exact as the event where the designed chameleon segments share the identical amino acid sequence in two generated structures, A and B. Let R_A and R_B denote the Cα self-consistency

Table 3: **Paired chameleon benchmark.** Benchmark results for $k = 6$–10, 1,000 pairs; all values are percentages. "w/o pairs": no paired supervision. †: identity enforced at inference.

Method	Exact	Success	$\text{Success}_{\text{SS80}}$
ProteinGenerator	100†	7.1	3.9
Ours, w/o pairs	0.0	0.0	0.0
Ours	75.1	**25.6**	**14.3**

Figure 5: **Success rate by chameleon sequence length.** Each method generates 200 pairs per length for $k = 6$–10 and 400 for the shaded OOD lengths.



RMSDs (scRMSDs) between the generated and refolded structures of the two sequences, a design **Success** is defined in Equation 4. For test cases where an SS condition is given, we calculate q_A and q_B, which are the fractions of conditioned positions where generated structure matches the input condition. This conditioned success is defined as $\textbf{Success}_{\text{SS80}}$ in Equation 5.

$$\text{Success} = \text{Exact} \wedge (R_A < 2\,\text{Å}) \wedge (R_B < 2\,\text{Å}), \tag{4}$$

$$\text{Success}_{\text{SS80}} = \text{Exact} \wedge (R_A < 2\,\text{Å}) \wedge (R_B < 2\,\text{Å}) \wedge \big[\min(q_A, q_B) \geq 0.8\big], \tag{5}$$

More details on sampling and metric definitions are provided in Appendix A.4.1 and A.4.2.

Joint design performance. In comparison with ProteinGenerator, AtomDiff-Chameleon achieves higher in silico success rates (25.6% and 14.3%) when tasked to generate chameleon sequences without and with SS-matching requirement respectively (Table 3). Under the SS-matching criterion, AtomDiff-Chameleon outperforms ProteinGenerator at every sequence length tested, from 6 to 10 residues (Figure 5). Figure 4a (upper example) shows a successful pair with $k = 6$, illustrating how an identical local sequence is accommodated in two different protein contexts.

Generalization to longer shared segments. To test whether AtomDiff-Chameleon generalizes beyond the short fragments represented in the curated dataset, we extend the benchmark to generating chameleon sequences of 12, 24, 36, and 48 residues. Following the same protocol setup, each model generates 400 pairs of 96-residue proteins per test length, totaling 1,600 pairs. Because these target lengths exceed the longest identity spans in our curated chameleon dataset, we randomly place the SS conditions from the original benchmark within the target segments and leave the remaining positions unconditioned. Details are provided in Appendix A.5.4.

As reported in Figure 5, AtomDiff-Chameleon achieves a higher overall success rate than ProteinGenerator – 4.56% (73/1,600) versus 1.38% (22/1,600). Notably, its $\text{Success}_{\text{SS80}}$ decreases as the target length increases. The lower example in Figure 4a illustrates a successful design at $k = 24$, in which an identical 24-residue sequence is embedded in two distinct generated protein contexts. At $k = 48$, successful pairs become rare, with only five successful designs each from AtomDiff-Chameleon and ProteinGenerator.

Contribution of chameleon paired supervision. Encouraged by the in silico benchmark results, we continue to investigate the individual contributions. We ablate the chameleon pairs from the training data (Section 3.3), while retaining the cross-branch communication architecture. This ablated variant ("Ours, w/o pairs" in Table 3) can still generate well-folded proteins (with a refolding success rate of 26.5%), but it yields zero Success or $\text{Success}_{\text{SS80}}$ cases (Table 3). This contrast highlights the critical role of paired supervision in learning cross-context sequence compatibility beyond single-state structure generation.

Together, the in silico benchmarks indicate that AtomDiff-Chameleon can generate de novo chameleon sequences. Its performance declines as the target length extends beyond the training distribution, consistent with the scarcity of long shared segments in the training data.

4.3 MOTIF-CONDITIONED CHAMELEON DESIGN

Chameleon sequences provide a starting point for multistate protein design but itself does not carry biological function. We next ask whether the local context sensitivity of chameleon segments can be combined with a functional motif to design switchable protein variants. Inspired by alternative splicing in eukaryotic cells, through which a single gene can produce protein variants with distinct structures and functions (Tauber et al., 2020), we task AtomDiff-Chameleon with designing pairs of motif-containing proteins that share an identical N-terminal sequence but adopt different structures when combined with different C-terminal sequences.

Design objective and evaluation. Let M denote a motif with a specified sequence and structure, X a chameleon segment, and A and B distinct C-terminal regions that provide alternative folding contexts. We aim to generate two constructs, M-X-A and M-X-B, that share the same N-terminal M-X sequence. Conditioned on M, AtomDiff-Chameleon jointly generates both constructs. The objective is to preserve the local structure of M while designing the shared X sequence to adopt different conformations in two contexts A and B. In addition to the Success metric defined in the Experiment 4.2, we evaluate motif preservation following Watson et al. (2023); Geffner et al. (2026). Detailed metric definitions and case-specific criteria are provided in Appendix A.7.

Joint motif and chameleon design. We choose two structurally distinct functional motifs for this case study: the V3 loop of HIV envelope glycoprotein gp120 (Jiang et al., 2010) and the BH3 motif of the pro-apoptotic protein Bim (Hinds et al., 2007). The V3 loop forms a β-hairpin and mediates binding to the HIV-1 co-receptor CXCR4. The upper example in Figure 4b shows a representative pair that preserves the β-hairpin while placing the shared chameleon segment in distinct structural contexts, resulting in different presentations of V3 epitope. The BH3 motif mediates interactions with pro-survival Bcl-2 family proteins. Its critical residues are highlighted in red in the lower example of Figure 4b. In context A, X extends the motif helix and leaves those functional residues solvent-exposed. In context B, the same X sequence adopts a different local conformation and shields these sites. In both cases, the context-dependent conformation of X alters the presentation and accessibility of the motif while preserving its local structure.[1]

5 CONCLUSION

We present a data-driven approach to context-dependent protein design that converts local sequence matches across distinct protein structures into paired supervision for all-atom generation. Computational and experimental results support the all-atom design capability of AtomDiff-X, providing a foundation for its multistate extension. With paired supervision, AtomDiff-Chameleon outperforms ProteinGenerator in joint generation of shared chameleon sequences within distinct structural contexts. Motif-conditioned designs further illustrate how a functional motif can be preserved while an adjacent chameleon segment alters its presentation or accessibility. More broadly, these findings highlight the value of deriving new supervision from existing structural data to support data-driven multistate protein design when direct multistate observations are scarce.

Our current designs share only local sequences while the remainder of each protein provides a different folding context. Therefore, they do not yet demonstrate conformational switching as a single complete protein sequence. The next challenge is to design a full-length protein that folds independently and adopts an alternative conformation in response to a defined molecular cue. Achieving this goal will require paired structural data of the same full-length sequence both in the absence and presence of a specific protein partner. Modeling these molecular cues as explicit external contexts could ultimately enable programmable multistate protein design.

[1]The exposed and occluded arrangements may respectively permit or restrict partner recognition, but their effects on binding and biological activity remain to be established experimentally.

AI USE STATEMENT

We used OpenAI Codex to assist with literature review, methodological discussions, experimental planning, code development and debugging, data analysis, figure preparation, and manuscript drafting and revision. AI-assisted outputs were checked against source publications, implementations, and experimental records. The authors take responsibility for the final methods, results, and claims.

ETHICS STATEMENT

This study uses publicly available protein structures and biochemical characterization of selected designs, without human participants or animal experiments. We recognize the potential dual-use implications of protein design. The motif-conditioned case studies are computational demonstrations and do not establish biological activity.

REPRODUCIBILITY STATEMENT

Model architecture and training objectives are detailed in Appendices A.1 and A.3, dataset construction in Appendix A.2, and generation and evaluation protocols in Appendix A.4. Additional computational results, biochemical procedures, and motif-case settings are provided in Appendices A.5–A.7. Our codebase and datasets are being prepared for public release. Once this preparation is complete, we will open-source the full codebase and release the datasets to support full reproducibility of the methods and results reported in this paper.

REFERENCES

Alex Abrudan, Sebastian Pujalte Ojeda, Chaitanya Joshi, Matthew Greenig, Felipe Engelberger, Alena Khmelinskaia, Jens Meiler, Michele Vendruscolo, and Tuomas Knowles. Multi-state protein sequence design with dynamicmpnn. In *International Conference on Learning Representations*, volume 2026, pp. 153620–153641, 2026.

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

Björn M Burmann, Stefan H Knauer, Anastasia Sevostyanova, Kristian Schweimer, Rachel A Mooney, Robert Landick, Irina Artsimovitch, and Paul Rösch. An α helix to β barrel domain switch transforms the transcription factor rfah into a translation factor. *Cell*, 150(2):291–303, 2012.

Dexiong Chen, Andrei Manolache, Mathias Niepert, and Karsten Borgwardt. Protein fold classification at scale: Benchmarking and pretraining. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=jPKqiaPTEd.

Alexander E Chu, Jinho Kim, Lucy Cheng, Gina El Nesr, Minkai Xu, Richard W Shuai, and Po-Ssu Huang. An all-atom protein generative model. *Proceedings of the National Academy of Sciences*, 121(27):e2311500121, 2024.

Justas Dauparas, Ivan Anishchenko, Nathaniel Bennett, Hua Bai, Robert J Ragotte, Lukas F Milles, Basile IM Wicky, Alexis Courbet, Rob J de Haas, Neville Bethel, et al. Robust deep learning–based protein sequence design using proteinmpnn. *Science*, 378(6615):49–56, 2022.

Tomas Geffner, Kieran Didi, Zhonglin Cao, Danny Reidenbach, Zuobai Zhang, Christian Dallago, Emine Kucukbenli, Karsten Kreis, and Arash Vahdat. La-proteina: Atomistic protein generation via partially latent flow matching. In *International Conference on Learning Representations*, volume 2026, pp. 67350–67395, 2026.

Amine Ghozlane, Agnel Praveen Joseph, Aurélie Bornot, and Alexandre G de Brevern. Analysis of protein chameleon sequence characteristics. *Bioinformation*, 3(9):367, 2009.

MG Hinds, C Smits, R Fredericks-Short, JM Risk, M Bailey, DCS Huang, and CL Day. Bim, bad and bmf: intrinsically unstructured bh3-only proteins that undergo a localized conformational change upon binding to prosurvival bcl-2 targets. *Cell Death & Differentiation*, 14(1):128–136, 2007.

Xunqing Jiang, Valicia Burke, Maxim Totrov, Constance Williams, Timothy Cardozo, Miroslaw K Gorny, Susan Zolla-Pazner, and Xiang-Peng Kong. Conserved structural elements in the v3 crown of hiv-1 gp120. *Nature structural & molecular biology*, 17(8):955–961, 2010.

Bowen Jing, Anna Sappington, Mihir Bafna, Ravi Shah, Adrina Tang, Rohith Krishna, Adam Klivans, Daniel J Diaz, and Bonnie Berger. Generating functional and multistate proteins with a multimodal diffusion transformer. *bioRxiv*, 2025.

Bowen Jing, Mihir Bafna, Anisha Parsan, Heyuan Michael Ni, David Kwabi-Addo, Bryan D. Bryson, Adam Klivans, and Bonnie Berger. Switchcraft: A programmatic framework for designing state-switching proteins. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=YtqaHnqv8c.

Wolfgang Kabsch and Christian Sander. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers: Original Research on Biomolecules*, 22(12):2577–2637, 1983.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems*, 35:26565–26577, 2022.

Rohith Krishna, Jue Wang, Woody Ahern, Pascal Sturmfels, Preetham Venkatesh, Indrek Kalvet, Gyu Rie Lee, Felix S Morey-Burrows, Ivan Anishchenko, Ian R Humphreys, et al. Generalized biomolecular modeling and design with rosettafold all-atom. *Science*, 384(6693):eadl2528, 2024.

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.

Wenlin Li, Lisa N Kinch, P Andrew Karplus, and Nick V Grishin. Chseq: A database of chameleon sequences. *Protein Science*, 24(7):1075–1086, 2015.

Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023.

Sidney Lyayuga Lisanza, Jacob Merle Gershon, Samuel WK Tipps, Jeremiah Nelson Sims, Lucas Arnoldt, Samuel J Hendel, Miriam K Simma, Ge Liu, Muna Yase, Hongwei Wu, et al. Multistate and functional protein design using rosettafold sequence space diffusion. *Nature biotechnology*, 43 (8):1288–1298, 2025.

Amy X. Lu, Jason Yang, Yueming Long, Theophile Lambert, Wilson Yan, Sarah A. Robinson, Kevin K. Yang, Vladimir Gligorijevic, Kyunghyun Cho, Richard Bonneau, Nathan C. Frey, Frances Arnold, and Pieter Abbeel. Controllable all-atom protein generation with latent diffusion. *bioRxiv*, 2026. doi: 10.1101/2024.12.02.626353. URL https://www.biorxiv.org/content/10.1101/2024.12.02.626353v3.

Mihaly Mezei. Revisiting chameleon sequences in the protein data bank. *Algorithms*, 11(8):114, 2018.

Daniel L Minor Jr and Peter S Kim. Context-dependent secondary structure formation of a designed protein sequence. *Nature*, 380(6576):730–734, 1996.

Lauren L Porter and Loren L Looger. Extant fold-switching proteins are widespread. *Proceedings of the National Academy of Sciences*, 115(23):5968–5973, 2018.

Wei Qu, Jiawei Guan, Rui Ma, Ke Zhai, Weikun Wu, and Haobo Wang. P(all-atom) is unlocking new path for protein design. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pp. 50786–50816. PMLR, 2025. URL https://proceedings.mlr.press/v267/qu25c.html.

Andrew Shrake and John A Rupley. Environment and exposure to solvent of protein atoms. lysozyme and insulin. *Journal of molecular biology*, 79(2):351–371, 1973.

Maria Tauber, Sarah Kreuz, Alexander Lemak, Papita Mandal, Zhadyra Yerkesh, Alaguraj Veluchamy, Bothayna Al-Gashgari, Abrar Aljahani, Lorena V Cortés-Medina, Dulat Azhibek, et al. Alternative splicing and allosteric regulation modulate the chromatin binding of uhrf1. *Nucleic acids research*, 48(14):7728–7747, 2020.

Matthew Z Tien, Austin G Meyer, Dariya K Sydykova, Stephanie J Spielman, and Claus O Wilke. Maximum allowed solvent accessibilites of residues in proteins. *PloS one*, 8(11):e80635, 2013.

Mihaly Varadi, Stephen Anyango, Mandar Deshpande, Sreenath Nair, Cindy Natassia, Galabina Yordanova, David Yuan, Oana Stroe, Gemma Wood, Agata Laydon, 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.

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(7976):1089–1100, 2023. doi: 10.1038/s41586-023-06415-8.

Jeremy Wohlwend, Gabriele Corso, Saro Passaro, Noah Getz, Mateo Reveiz, Ken Leidal, Wojtek Swiderski, Liam Atkinson, Tally Portnoi, Itamar Chinn, et al. Boltz-1 democratizing biomolecular interaction modeling. *BioRxiv*, pp. 2024–11, 2025.

Yongsheng Yu, Wei Xiong, Weili Nie, Yichen Sheng, Shiqiu Liu, and Jiebo Luo. Pixeldit: Pixel diffusion transformers for image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14273–14282, 2026.

A APPENDIX

CONTENTS

A.1 All-atom generative modeling details 14
A.1.1 Representation, architecture, and sequence readout 14
A.1.2 Diffusion formulation and training objectives 15
A.1.3 Pretraining data and optimization 17
A.2 Chameleon dataset construction 19
A.2.1 Source structures and shared-sequence discovery 19
A.2.2 Curation and structural annotations 19
A.2.3 Dataset composition and structural examples 20
A.3 Paired chameleon training details 20
A.3.1 Training examples and augmentation 20
A.3.2 Cross-branch communication 24
A.3.3 Secondary-structure conditioning 25
A.3.4 Motif conditioning 25
A.3.5 Paired training objectives 26
A.3.6 Fine-tuning protocol 27
A.4 Generation and evaluation protocols 27
A.4.1 Sampling, baseline implementation, and refolding 27
A.4.2 Evaluation metrics and reporting rules 29
A.5 Additional computational results 30
A.5.1 Unconditional generation and structure examples 30
A.5.2 Paired chameleon benchmark 32
A.5.3 Effect of paired supervision 32
A.5.4 Generalization to longer shared segments 34
A.6 Biochemical characterization 36
A.6.1 Individual outcomes and characterization data 36
A.7 Motif-conditioned design case studies 36
A.7.1 Case inputs and generation settings 36
A.7.2 Motif preservation and case-selection criteria 36

A.1 ALL-ATOM GENERATIVE MODELING DETAILS

This section specifies the unconditional model introduced in Section 3.1. The model evaluated in Section 4.1 is the same 190k-step pretrained model used to initialize paired chameleon training. Figure 6 illustrates the architecture; Algorithms 1–3 give its forward computation.

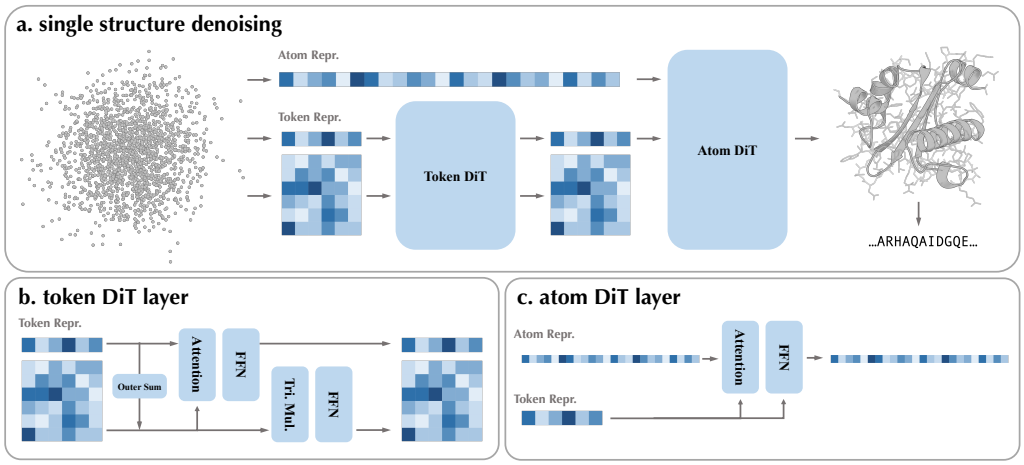


Figure 6: **Architecture of the all-atom generative model.** (a) Token DiT aggregates residue-level information to condition atom-level denoising; amino acid identities are read from the predicted coordinates. (b) Token blocks update single and pair representations. (c) Atom blocks use the final single representations and noise embedding for adaptive modulation. Algorithms 1–3 specify the computation, including the auxiliary distogram head and sequence readout.

A.1.1 REPRESENTATION, ARCHITECTURE, AND SEQUENCE READOUT

Coordinates and masks. For a protein of length L, we represent its structure by $X \in \mathbb{R}^{L\times 14\times 3}$ and its sequence by $s \in \{1,\dots,20\}^L$. Following the `atom14` representation (Qu et al., 2025), unused atom slots are placed at the residue's C_α position. These virtual slots participate in coordinate modeling and supervision, giving every residue the same 14-slot representation. They are distinct from batch-padding positions, which are excluded from attention and loss computation. We write m_{ir} for the model mask of slot r in residue i and v_i for the residue mask. Sequence identities are supervision targets and are not supplied to the unconditional denoiser.

Feature initialization. Let $U = c_{\text{in}}(\sigma)X_\sigma$ denote the normalized noisy coordinates. An atom-wise two-layer MLP, with widths $3\to 768\to 768$ and a SiLU activation, is applied before pooling:

$$h_i^{(0)} = \frac{\sum_{r=1}^{14} m_{ir}\,\phi(U_{ir})}{\sum_{r=1}^{14} m_{ir}}, \qquad a_{ir}^{(0)} = W_x U_{ir} + E_{\text{slot}}(r). \tag{6}$$

Here $h_i^{(0)}$ and $a_{ir}^{(0)}$ are the initial token and atom features. Initial pair features combine a binned residue-distance embedding with a learned embedding of the residue-index difference clipped to $[-32, 32]$. The distance embedding maps 64 one-hot bins to 256 channels; its 63 bin boundaries are evenly spaced from 2.3125 to 21.6875 and applied to the normalized input geometry. The noise embedding is $e(\sigma) = \cos(2\pi[w\log(\sigma)/4 + b])$, where $w, b \in \mathbb{R}^{256}$ are fixed Gaussian vectors. Separate linear projections provide token- and atom-level noise conditions.

Token-to-atom denoising. The network contains 16 Token DiT blocks followed by 16 Atom DiT blocks. Token blocks exchange information between the single and pair tracks through an outer-sum update and pair-biased attention, followed by single-feature transitions and outgoing/incoming triangular multiplication on the pair track. For each atom block, the conditioning vector is

$$c_{ir}^{\text{atom}} = W_h h_i^{(16)} + W_{\text{atom}} e(\sigma), \tag{7}$$

and is shared by all slots r of residue i. The final pair representation feeds the auxiliary distance head, while only the single representation conditions Atom DiT. Each block produces separate shift, scale, and residual-gating parameters for attention and feed-forward updates. We use RMSNorm for the feature normalization in this AdaLN-style modulation. The last modulation projection and the coordinate output head are initialized to zero.

Both attention stacks use query/key normalization and a learned sigmoid output gate. Token attention uses pair bias without rotary position encoding. Atom attention uses residue-index rotary position encoding, learned atom-slot embeddings, and a query scale of $\log(14L)/\log(1344)$. Four learned register tokens are added to each stack for each protein and removed before producing residue or atom outputs. When proteins are packed together, registers and attention remain specific to their source protein. Table 4 summarizes the dimensions.

Coordinate-to-sequence readout. The readout G_ϕ predicts 20 amino acid logits per residue from geometry. It first flattens each residue's 14×14 intra-residue distance matrix and maps its 196 entries to a 64-dimensional feature through a $196 \to 128 \to 64$ network with LayerNorm and ReLU. A single message-passing layer aggregates the 16 nearest residues in C_α distance, including the residue itself. For neighbors $j \in \mathcal{N}_i$, it applies a $129 \to 64 \to 64$ MLP with ReLU to the concatenation of the two node features and their C_α distance. The mean message is added to the local feature, and a linear classifier produces the logits:

$$g_i = \phi_{\text{local}}(\text{vec}\, D_i), \qquad \ell_i = W_{\text{seq}} \left[g_i + \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} \phi_{\text{msg}}(g_i, g_j, d_{ij}^{C_\alpha}) \right] + b_{\text{seq}}. \tag{8}$$

Here D_i is the intra-residue distance matrix, and invalid neighbors are masked. At generation time, the sequence is decoded by taking the highest-scoring amino acid at each position. During pretraining, the readout is optimized jointly with the denoiser, using the coordinate inputs specified below.

Table 4: All-atom model architecture.

Component	Configuration
Coordinate representation	$L \times 14 \times 3$; $L \leq 512$ in training
Token / Atom DiT depth	16 / 16 blocks
Token / atom / pair widths	768 / 384 / 256
Token / atom attention heads	12 / 12 (head widths 64 / 32)
Noise embedding width	256
Token / atom condition widths	768 / 384
Feed-forward expansion	$4\times$ input width
Learned registers	4 per protein in each stack
Token position encoding	Clipped relative index, cutoff 32
Atom position encoding	Residue RoPE + 14 learned slot embeddings
Query/key normalization	Enabled in both stacks
Atom query scaling	$\log(14L)/\log(1344)$
Auxiliary distance head	RMSNorm + linear map to 64 bins
Sequence readout	Width 64; 16 neighbors; 20 classes

A.1.2 Diffusion formulation and training objectives

Corruption and preconditioning. We use a log-normal noise distribution with independent Gaussian coordinate perturbations for each protein:

$$\begin{aligned} \sigma &= 16 \exp(-1.2 + 1.5z), & z &\sim \mathcal{N}(0,1), \\ X_\sigma &= X_0 + \sigma\epsilon, & \epsilon &\sim \mathcal{N}(0, I). \end{aligned} \tag{9}$$

For the EDM reconstruction in Eq. 1, the coefficients are

$$c_{\text{in}}(\sigma) = \frac{1}{\sqrt{\sigma^2 + \sigma_{\text{data}}^2}}, \qquad c_{\text{skip}}(\sigma) = \frac{\sigma_{\text{data}}^2}{\sigma^2 + \sigma_{\text{data}}^2}, \qquad c_{\text{out}}(\sigma) = \frac{\sigma\sigma_{\text{data}}}{\sqrt{\sigma^2 + \sigma_{\text{data}}^2}}, \tag{10}$$

with $\sigma_{\text{data}} = 16$ (Karras et al., 2022). Equation 9 specifies training corruption; inference schedules are described separately in Appendix A.4.1.

Algorithm 1 All-atom denoising and output heads

Require: Noisy coordinates $X_\sigma \in \mathbb{R}^{L\times 14\times 3}$, noise σ, residue indices, masks
Ensure: Clean-coordinate estimate $\hat{X}_0$, residue logits ℓ, distance logits B

$U \leftarrow c_{\text{in}}(\sigma) X_\sigma$; $\quad e \leftarrow \text{NoiseEmbed}(\log \sigma/4)$
$h_i \leftarrow \text{MaskedMean}_r\big[\text{MLP}_{\text{atom}\to\text{token}}(U_{ir})\big]$
$z_{ij} \leftarrow \text{DistanceEmbed}(U)_{ij} + \text{RelativePositionEmbed}(i,j)$
$a_{ir} \leftarrow W_x U_{ir} + E_{\text{slot}}(r)$; $\quad c_i^{\text{token}} \leftarrow W_{\text{token}} e$
Append protein-specific token registers; extend masks and conditions
for $b = 1, \ldots, 16$ **do**
  $(h, z) \leftarrow \text{TokenBlock}_b(h, z, c^{\text{token}}, \text{masks})$ ▷ Algorithm 2
end for
Remove token registers and their pair rows/columns
$B_{ij} \leftarrow W_{\text{dist}}\,\text{RMSNorm}(z_{ij} + z_{ji})$ ▷ 64-bin auxiliary head
$c_{ir}^{\text{atom}} \leftarrow W_h h_i + W_{\text{atom}} e$ for each of the 14 slots
Append protein-specific atom registers; extend masks and conditions
for $b = 1, \ldots, 16$ **do**
  $a \leftarrow \text{AtomBlock}_b(a, c^{\text{atom}}, \text{masks}, \text{positions}, L)$ ▷ Algorithm 3
end for
Remove atom registers; $\quad F_\theta \leftarrow W_{\text{out}} a + b_{\text{out}}$
$\hat{X}_0 \leftarrow c_{\text{skip}}(\sigma) X_\sigma + c_{\text{out}}(\sigma) F_\theta$
$\ell \leftarrow G_\phi(\hat{X}_0)$ ▷ Coordinate-to-sequence readout
return $\hat{X}_0, \ell, B$

Algorithm 2 Token DiT block

Require: Single features h, pair features z, noise condition c, attention/pair masks M, P

$(\alpha_1, \beta_1, \gamma_1, \alpha_2, \beta_2, \gamma_2) \leftarrow W_c\,\text{SiLU}(\text{LayerNorm}(c))$
$(u_i^{\text{L}}, u_i^{\text{R}}) \leftarrow \text{Split}(W_u\,\text{RMSNorm}(h_i))$
$z_{ij} \leftarrow P_{ij}\,[z_{ij} + W_z(u_i^{\text{L}} + u_j^{\text{R}})]$ ▷ Outer sum
$u \leftarrow (1+\gamma_1) \odot \text{RMSNorm}(h) + \beta_1$
$h \leftarrow h + \alpha_1 \odot \text{PairAttention}(u, z; M)$
$u \leftarrow (1+\gamma_2) \odot \text{RMSNorm}(h) + \beta_2$
$h \leftarrow h + \alpha_2 \odot \text{FFN}_h(u)$
$z \leftarrow P \odot [z + \text{TriangleOutgoing}(z; P)]$
$z \leftarrow P \odot [z + \text{TriangleIncoming}(z; P)]$
$z \leftarrow P \odot [z + \text{FFN}_z(\text{RMSNorm}(z))]$
return h, z

Algorithm 3 Atom DiT block

Require: Atom features a, residue-shared condition c, attention mask M, residue positions, length L

$(\alpha_1, \beta_1, \gamma_1, \alpha_2, \beta_2, \gamma_2) \leftarrow W_c\,\text{SiLU}(\text{LayerNorm}(c))$
$u \leftarrow (1+\gamma_1) \odot \text{RMSNorm}(a) + \beta_1$
$Q, K, V \leftarrow W_Q u, W_K u, W_V u$
$Q, K \leftarrow \text{RoPE}(\text{QKNorm}(Q, K); \text{residue positions})$
$Q \leftarrow [\log(14L)/\log(1344)]\,Q$
$o \leftarrow \text{Softmax}_M(QK^{\mathsf{T}}/\sqrt{d_{\text{head}}})V$
$a \leftarrow a + \alpha_1 \odot W_o[\text{ConcatHeads}(o) \odot \text{sigmoid}(W_g u)]$
$u \leftarrow (1+\gamma_2) \odot \text{RMSNorm}(a) + \beta_2$
$a \leftarrow a + \alpha_2 \odot \text{FFN}_a(u)$
return a

Coordinate objective. For this paragraph, flatten the residue/slot indices into atom-slot indices p, q. Let u_p denote the source pLDDT and define the confidence weight

$$t_p = \mathrm{clip}\left(\frac{u_p - 60}{20}, 0, 1\right), \qquad w_p^{\mathrm{conf}} = 3t_p^2 - 2t_p^3, \qquad w^{\mathrm{noise}}(\sigma) = \frac{1}{\sigma^2 + 1/5}. \tag{11}$$

Residue confidence is broadcast to all 14 slots; records without confidence annotations receive unit confidence weight. For the aligned training objective, a weighted Kabsch fit rigidly aligns X_0 to $\mathrm{stopgrad}(\hat{X}_0)$ using weights $m_p w_p^{\mathrm{conf}}$. Writing the aligned target as $\widetilde{X}_0$, the per-protein loss is

$$\mathcal{L}_{\mathrm{coord}} = \frac{w^{\mathrm{noise}}(\sigma)}{\sum_p m_p} \sum_p m_p w_p^{\mathrm{conf}} \, \|\hat{X}_{0,p} - \widetilde{X}_{0,p}\|_2^2. \tag{12}$$

The denominator counts modeled slots rather than confidence-weighted slots. If all confidence weights vanish, the alignment uses the binary mask as a fallback; the confidence-weighted coordinate loss remains zero.

Smooth distance objective. The smooth-lDDT term in Eq. 2 is implemented as an atom-pair distance agreement loss. Let $\Delta_{pq} = \left|\|X_{0,p} - X_{0,q}\|_2 - \|\hat{X}_{0,p} - \hat{X}_{0,q}\|_2\right|$, with distances expressed in Å, and $\omega_p = m_p w_p^{\mathrm{conf}}$. Then

$$\mathcal{L}_{\text{smooth-lDDT}} = 1 - \frac{\displaystyle\sum_{p,q} \omega_p \omega_q \left[\frac{1}{4} \sum_{\tau \in \{0.5,1,2,4\}} \mathrm{sigmoid}(\tau - \Delta_{pq})\right]}{\displaystyle\sum_{p,q} \omega_p \omega_q}. \tag{13}$$

The sum includes all modeled slot pairs within the protein, including diagonal entries; no distance-based neighborhood cutoff is applied.

Sequence and residue-distance supervision. The sequence readout is trained using ground-truth coordinates or denoised predictions. When $\sigma < 4$, the predicted coordinates are selected with probability 0.5; otherwise the readout receives ground-truth coordinates. This choice is made once per protein and applied to all its residues. For the resulting input X^{read},

$$\mathcal{L}_{\mathrm{seq}} = -\frac{1}{\sum_i v_i} \sum_i v_i \log \mathrm{softmax}\big(G_\phi(X^{\mathrm{read}})_i\big)_{s_i}. \tag{14}$$

Predicted coordinates are not detached, so this objective also updates the denoiser when predictions are selected. All amino acid classes use the same loss weight. For residue-distance supervision, the symmetrized pair features predict a 64-bin distribution, with 63 boundaries uniformly spaced between 2.3125 and 21.6875 Å. Let b_{ij} be the bin label of the clean residue-level distance and B_{ij} the predicted logits. We compute

$$\mathcal{L}_{\mathrm{dist}} = -\frac{\sum_{i,j} v_i v_j \log \mathrm{softmax}(B_{ij})_{b_{ij}}}{\sum_{i,j} v_i v_j}. \tag{15}$$

Both cross-entropy objectives use valid-position masks without additional pLDDT weighting.

Combined objective and reduction. The loss weights in Eq. 2 are $\lambda_{\mathrm{lddt}} = 0.25$, $\lambda_{\mathrm{seq}} = 0.05$, and $\lambda_{\mathrm{dist}} = 0.01$. Each term is first reduced within a protein. In the protein-averaged training stage, the objective is then averaged equally over all proteins in the global batch, across packed rows and devices. This reduction gives each protein equal weight regardless of its length or which other proteins share its packed row.

A.1.3 Pretraining data and optimization

Structure sources and pool sizes. The initial training collection consists of processed single-chain structures from AFDB v6, synchronized with UniProt 2025_03 (Varadi et al., 2022). Preprocessing

retains the first chain, selects structures with 32–512 residues and mean pLDDT at least 80, and trims low-confidence termini. At each end, residues outside the first run of three consecutive residues with pLDDT at least 70 are removed; chains without such a run or shorter than 32 residues after trimming are discarded. Retained residues are renumbered contiguously and converted to `atom14` records containing coordinates, atom masks, sequence identities, residue indices, chain indices, and per-residue pLDDT. This collection contains 753,902 structures, of which 575,730 have nonzero sampling probability after geometric and cluster-level exclusions.

From step 160k, each protein draw selects one of three streams before packing: short AFDB chains ($32 \leq L \leq 255$) with probability 0.70, TED-derived dominant-single-domain parents with probability 0.10, or TED-derived multi-domain parents with probability 0.20. The TED streams use complete AFDB-v4 parent coordinates from the full-atom archive corresponding to TEDBench (Chen et al., 2026). This TEDBench release comprises 462,175 representatives across 965 CATH topology classes; only its training split is used, yielding 332,007 converted parent chains of length 32–512 before long-pool selection. Domain boundaries come from TED v5's `ted_365m.domain_summary.cath.globularity.taxid.tsv.gz` (Lau et al., 2024). The effective sizes of the three continuation streams are 521,329, 7,564, and 29,423, respectively. Additional curation and sampling use AFDB compactness and secondary-structure filters with cluster-balanced sampling, TED domain-coverage and contact criteria with linear-helix filtering, and CATH-based reweighting within the dominant-single-domain stream.

Augmentation and packed batches. Each protein is centered using its observed coordinates, randomly rotated, and translated by an independent vector drawn from $\mathcal{N}(0, 9I)$ before coordinate corruption. We pack proteins into rows with a capacity of 512 residues, with two rows per device and at most 16 proteins per row. Token attention, atom attention, and pair updates are restricted to the same protein; padding is masked. Every protein retains its own noise level, rigid-body augmentation, and register tokens. With 32 devices, a batch has 64 packed rows and a maximum capacity of 32,768 residues; its actual protein count varies with sampled lengths and packing occupancy.

Optimization. We use Muon for the main matrix parameters and AdamW for the remaining parameter groups, including the sequence readout, output heads, embeddings, and modulation parameters. The configured peak learning rate is 5×10^{-4}, with a 5k-step linear warmup, a constant phase through step 120k, and cosine decay toward 10% of the peak at step 220k. The model used in this paper is selected at step 190k. Training uses mixed bfloat16 precision, activation checkpointing, and an exponential moving average with decay 0.999. Table 5 lists the configuration.

Table 5: Pretraining configuration.

Setting	Value
Selected training step	190,000
Devices / packed rows per device	32 / 2
Packed-row residue capacity	512
Maximum proteins per row	16
Precision / activation checkpointing	Mixed bfloat16 / enabled
Optimizer	Muon + AdamW parameter groups
Peak configured learning rate	5×10^{-4}
Warmup / constant phase endpoint	5,000 / 120,000 steps
Cosine endpoint / minimum LR ratio	220,000 steps / 0.1
Weight decay on decayed groups	0.01
Muon momentum / Newton–Schulz steps	0.95 / 5 (Nesterov enabled)
AdamW (β_1, β_2) / ϵ	$(0.9, 0.95)$ / 10^{-8}
EMA decay	0.999
Training σ_{data}	16
Log-noise location / scale	-1.2 / 1.5
Noise loss weighting	$1/(\sigma^2 + 1/5)$
Confidence transition interval	pLDDT 60–80, smoothstep
Auxiliary weights: lDDT / sequence / distance	0.25 / 0.05 / 0.01
Predicted-coordinate readout input	Probability 0.5 when $\sigma < 4$

Generation schedules, decoding, refolding, and evaluation metrics are described in Appendix A.4.

Table 6: **Curation criteria for chameleon pairs.** Endpoint criteria apply to both source structures. Sequence-similarity and SS-contrast criteria compare the paired occurrences.

Criterion	Retention rule
Mapping and atoms	Identical mapped sequences; chemically expected span atoms present and finite.
Span confidence	Minimum residue pLDDT ≥ 80.
Neighboring confidence	Window mean pLDDT ≥ 80; fraction below 70 ≤ 0.10.
Parent confidence	Population standard deviation of residue pLDDT < 15.
Whole-chain similarity	Aligned sequence identity < 0.30.
Local similarity	Exclude if identity ≥ 0.50, both coverages ≥ 0.70, and at least 16 residue–residue positions are aligned.
Structural contrast	Full-span weighted SS difference $S_{\mathrm{SS}} \geq 0.60$.
SS annotations	Valid full-span labels at both endpoints; at least one endpoint has a dominant H or E label with purity ≥ 0.60.
Structural environment	Mean span RSA ≤ 0.50 and mean nonlocal context contacts per residue ≥ 2.
Construct/tag motifs	No complete predefined motif match overlapping the shared interval by at least four residues.

A.2 Chameleon dataset construction

A.2.1 Source structures and shared-sequence discovery

We search the 753,902 preprocessed AFDB single-chain structures described in Appendix A.1.3, before the additional exclusions and sampling weights used for pretraining. Sequences are mapped to atom14 coordinates, preserving accession identifiers and residue positions. Each occurrence is a contiguous interval in a source protein; pairs link sequence-identical occurrences in different proteins, whose remaining structures provide their respective contexts.

We group exact seeds of lengths $k \in \{6, 7, 8\}$ by sequence after occurrence-level mapping, atom-completeness, and confidence checks. For each group, we enumerate all unordered occurrence pairs up to a limit of 4,096. Same-parent pairs are discarded without replenishment. Matches are extended bidirectionally while amino acids remain identical, producing maximal contiguous identity spans of length $\ell \geq k$. Curation evaluates the complete expanded span without an upper length bound.

A.2.2 Curation and structural annotations

Sequence consistency and confidence. Table 6 summarizes the retention criteria. Chemically expected span atoms must have valid masks and finite coordinates. Neighboring windows extend by at most 16 residues per side, shortened at chain ends. Global sequence alignments use match/mismatch scores $1/-1$ and gap-open/extension scores $-10/-1$. Identity divides exact matches by columns containing residues from both sequences. For local similarity, left and right neighboring sequences are aligned separately, then matches m and aligned-position counts a are summed. With available neighboring lengths n_A and n_B, identity is m/a and coverages are a/n_A and a/n_B; absent alignments contribute zero counts. Construct/tag screening searches for complete predefined motif strings within the shared sequence, including common protease sites and affinity tags.

Secondary-structure contrast and labels. We annotate complete source chains with DSSP 4.0.4, then extract the full identity spans (Kabsch & Sander, 1983). To quantify secondary-structure dissimilarity, we use a weighted mismatch score following Porter & Looger (2018):

$$S_{\mathrm{SS}} = \frac{1}{\ell}\sum_{i=1}^{\ell} w(u_i, v_i), \qquad w(u,v) = \begin{cases} 0, & u = v, \\ 1, & \{u,v\} \in \{\{\mathrm{H},\mathrm{E}\},\{\mathrm{T},\mathrm{E}\}\}, \\ 0.6, & \mathrm{C} \in \{u,v\},\ u \neq v, \\ 0.25, & \{u,v\} = \{\mathrm{H},\mathrm{T}\}. \end{cases} \tag{16}$$

Here, ℓ is the length of the complete identity span obtained by extending the seed match, and u_i and v_i denote the secondary-structure assignments at the corresponding position i in the two fragments, respectively. Each assignment takes one of four states: H (helix), E (β-strand), T (turn), or C (coil).

For Eq. 16, H/G/I map to H, E/B to E, T/S to T, and remaining coil states, including P, to C. The score averages all ℓ corresponding positions, with no additional RMSD gate. Three-state annotations additionally merge T into C. A dominant H/E/C label requires a unique most frequent state occupying at least 60% of positions; other spans are MIXED. At least one endpoint must be dominantly H or E. Per-residue strings retain mixed patterns for conditioning: dominant labels neither require uniform H/E differences nor exclude pairs sharing a dominant label.

Structural context. Parent-chain solvent-accessible areas are calculated with the Shrake–Rupley method in Biotite, using 200 surface points per atom, a 1.4 Å probe radius, and ProtOr radii (Shrake & Rupley, 1973). Residue areas are divided by the amino-acid-specific maxima of Tien et al. to obtain RSA, then averaged over the span (Tien et al., 2013). For each span residue i, a context contact is a residue j outside the entire identity span with $|i - j| > 4$ and representative-atom distance at most 8 Å. The representative atom is Cβ, falling back to Cα when Cβ is unavailable. Contact counts are averaged over span residues; these environment metrics use the complete source chain before training crops.

A.2.3 Dataset composition and structural examples

Canonical pair identity comprises both accessions, span positions, and the shared sequence. Repeated discoveries within and across seed searches are merged, retaining the representative from the largest discovery k and all discovery provenance. The final pool contains 11,762,108 pairs, 3,774,934 distinct shared sequences, 12,991,916 structural occurrences, and 666,702 parent proteins. Distinct shared sequences are counted by deduplicating complete identity-span amino acid strings, irrespective of their source proteins or number of paired occurrences. Occurrences are distinct accession/interval combinations and may have multiple partners, so pair counts do not imply statistical independence.

Figure 2 summarizes the dataset composition. Lengths 6–11 contain 11,152,748, 573,248, 34,294, 1,711, 100, and 7 pairs, respectively: 94.82% have length 6, and 11 is an observed maximum (Figure 2e). Merging A/B order, H/E, H/C, E/C, MIXED-involving, and same-dominant-label pairs constitute 55.98%, 8.77%, 10.73%, 24.33%, and 0.18% (Figure 2d). Length and SS distributions count pairs; environment and position distributions (Figure 2b,c) count distinct structural occurrences. Normalized position divides the zero-based span midpoint by $L - 1$, where L is the parent-chain length. Figures 7–9 illustrate 15 sequence-distinct pairs selected across predefined SS/length strata. Their allocation illustrates coverage, not population or training-sampler frequencies.

Comparison with earlier collections. Figure 1b compares sequence counts rather than pair or occurrence counts. For Mezei (2018), we sum the $N_L(\mathrm{cham})$ column for the 20-residue alphabet in that study's Table 1 over lengths 6–20, obtaining 42,975 sequences; this subtotal includes substrings of longer chameleons and excludes the reported 5-residue sequences. For Ghozlane et al. (2009), the total of 100,302 is the sum of 63,228, 34,408, 2,423, 179, and 64 sequences of lengths 4, 5, 6, 7, and 8, respectively. For ChSeq, we use the published loose-criterion count of 128,703 sequences of lengths 6–10 (Li et al., 2015). Our 3,774,934 distinct shared sequences have observed lengths of 6–11 residues. The source structures, length ranges, and curation criteria differ across collections, so this comparison describes resource sizes rather than discovery performance under a matched protocol.

A.3 Paired chameleon training details

This section specifies the training inputs, communication pathways, conditions, and objectives that extend the all-atom model in Appendix A.1. The construction of the underlying chameleon collection is described in Appendix A.2.

A.3.1 Training examples and augmentation

Logical training units. A training unit contains either one individual protein, two views of one protein (a self-copy pair), or two different proteins from the chameleon collection (a real pair). Let $\mathcal{B}_u$ index the one or two branches in unit u, with sequence and clean atom14 coordinates (s^b, X_0^b) for branch b. For a paired unit, $\mathcal{P}_u = \{(i, \pi(i))\}$ denotes residue correspondences within a designated contiguous shared segment, satisfying $s_i^A = s_{\pi(i)}^B$. The correspondence follows the N-to-C order in both branches. A self-copy pair shares coordinates in its source frame, whereas a real pair supplies

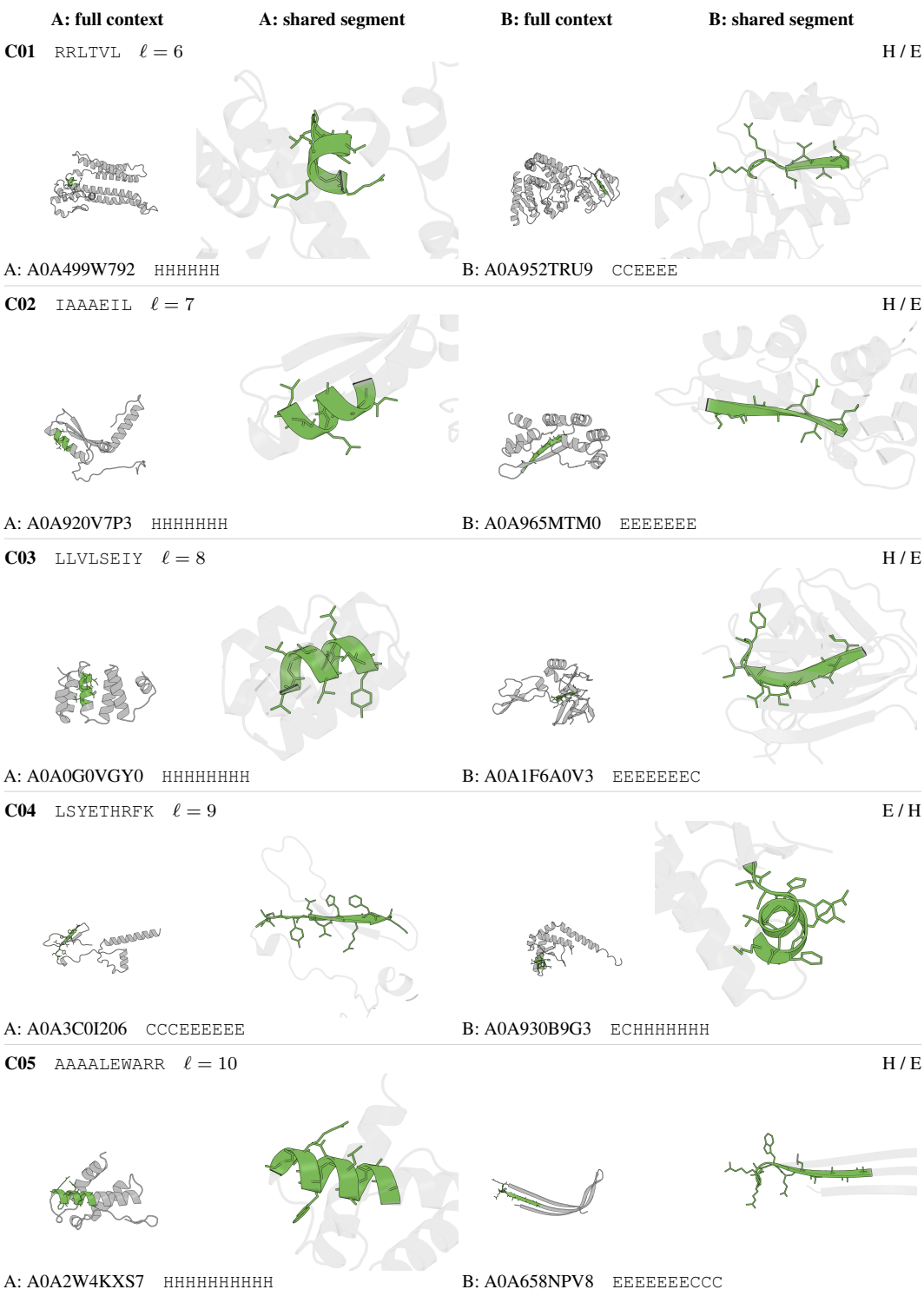


Figure 7: **Chameleon sequence pairs in distinct protein contexts (C01–C05).** Each row shows the complete parent structure and a local close-up for A and B. Green highlights the identical sequence; gray shows its protein context. Headers give the shared sequence, length, and dominant SS labels; per-residue SS strings appear below. The examples are selected across SS/length strata to illustrate dataset coverage, rather than its relative frequencies. Structures are AFDB predictions.

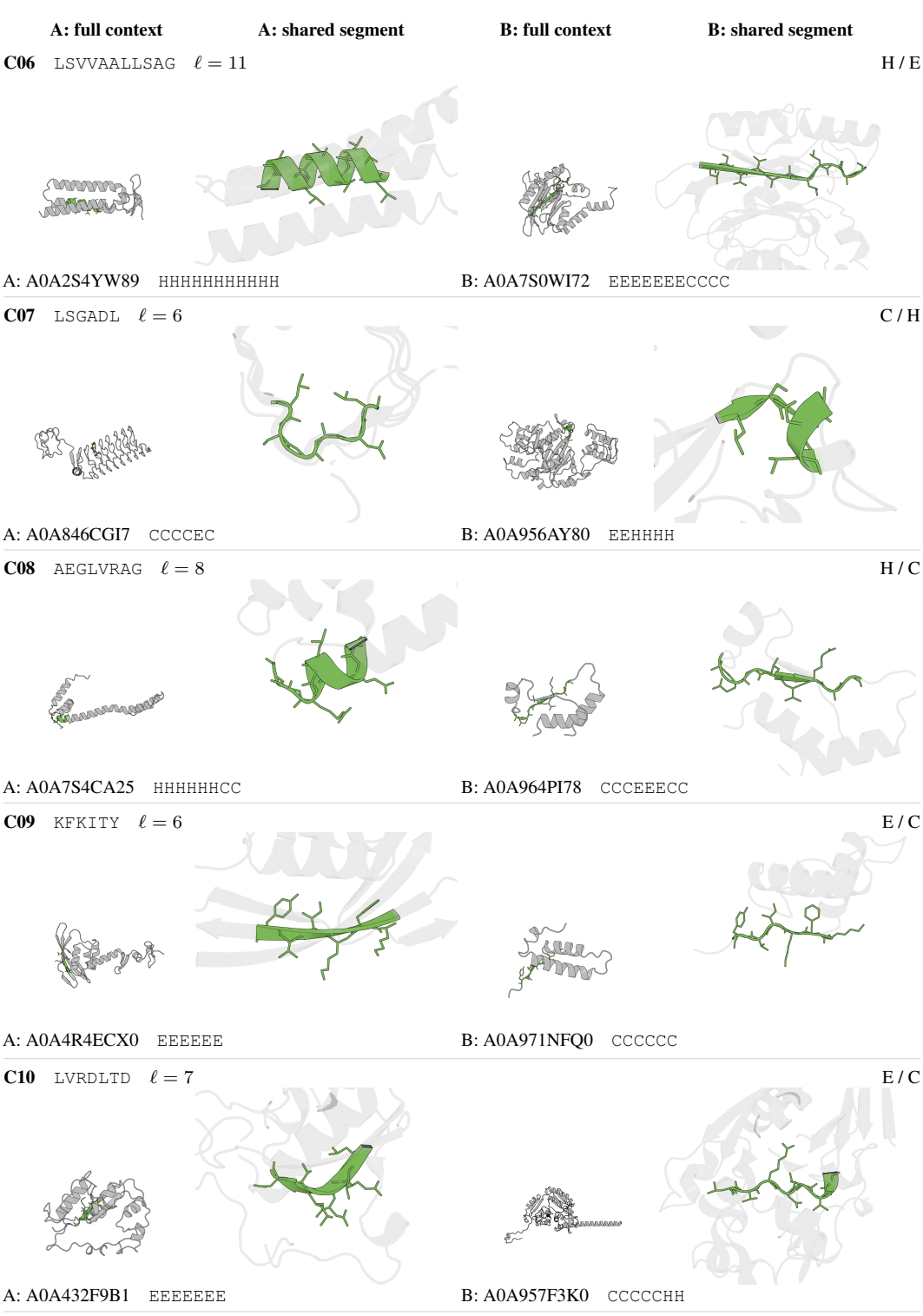


Figure 8: **Chameleon sequence pairs in distinct protein contexts (C06–C10).** Each row shows the complete parent structure and a local close-up for A and B. Green highlights the identical sequence; gray shows its protein context. Headers give the shared sequence, length, and dominant SS labels; per-residue SS strings appear below. Selection follows Figure 7.

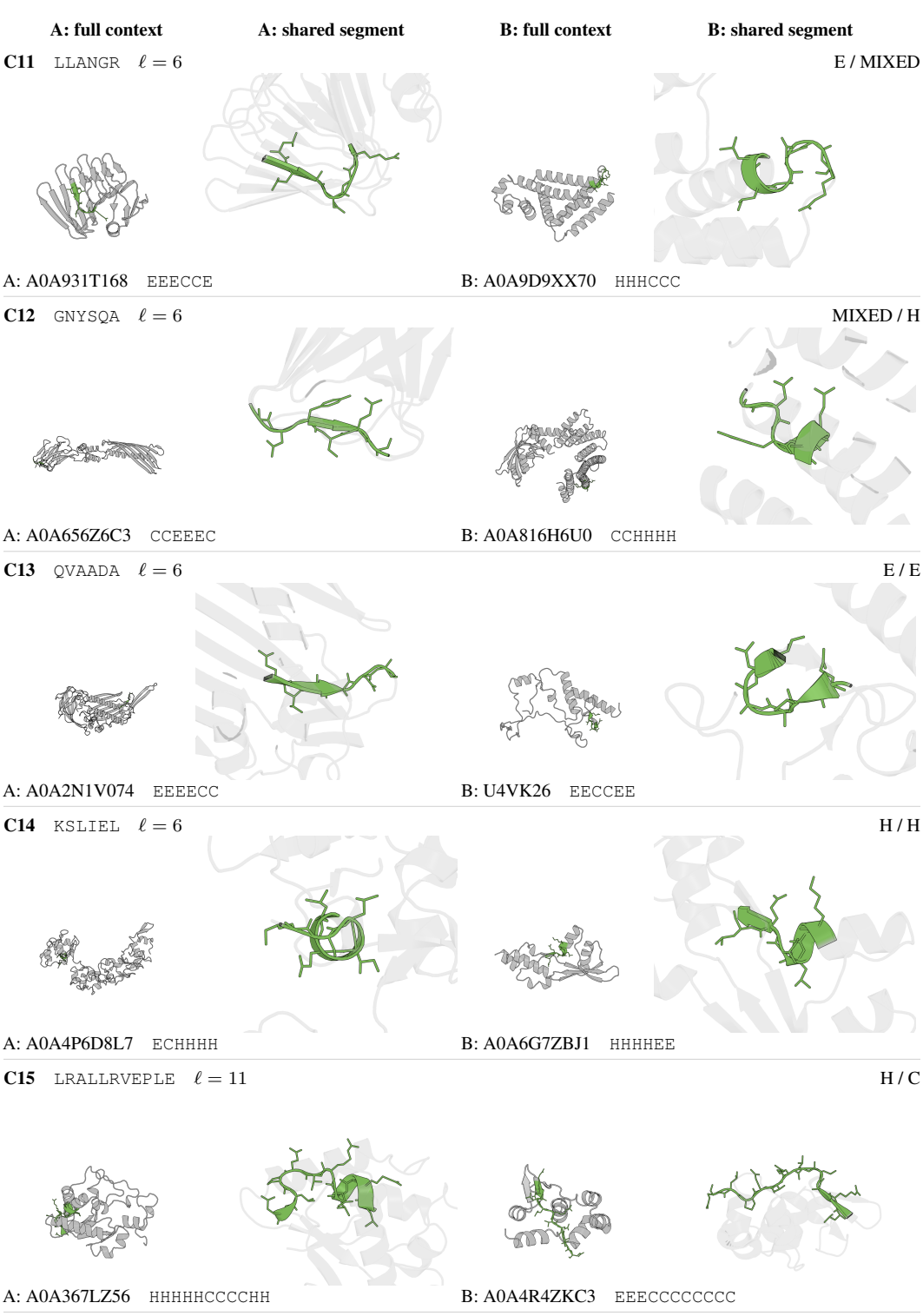


Figure 9: **Chameleon sequence pairs in distinct protein contexts (C11–C15).** Each row shows the complete parent structure and a local close-up for A and B. Green highlights the identical sequence; gray shows its protein context. Headers give the shared sequence, length, and dominant SS labels; per-residue SS strings appear below. Selection follows Figure 7.

distinct local conformations and protein contexts. Motif conditioning is applied to these three unit types, as detailed in Appendix A.3.4.

Shared segments and crops. Real pairs retain their complete identity span when each branch is cropped to at most 256 residues. For a longer parent, spatial and contiguous crop proposals are chosen with equal probability. A contiguous crop starts uniformly among windows containing the full span; if it retains less than 90% of the parent span–context contacts, it is replaced by a spatial crop. Contacts use Cβ coordinates (Cα when Cβ is unavailable), distance < 8 Å, and original sequence separation > 4, with the context endpoint outside the shared span. Spatial cropping starts with the span and up to 16 neighboring residues on either side, then adds sequence blocks of up to 16 residues around context residues ranked by their minimum spatial distance to the span, until the residue budget is filled.

Packing and corruption. A unit is packed intact, with separate branch identifiers and one logical-unit identifier; unrelated units do not communicate. Deferred units are retained for subsequent packing rather than discarded when they do not fit. Each branch is centered over its modeled atom slots, independently rotated uniformly in $SO(3)$, and translated by a draw from $\mathcal{N}(0, 3^2 I)$ in Å. One noise level σ_u is drawn per unit using Eq. 9, and

$$X^b_{\sigma_u} = X^b_0 + \sigma_u \epsilon^b, \qquad \epsilon^b \sim \mathcal{N}(0, I), \quad b \in \mathcal{B}_u, \tag{17}$$

with independent Gaussian noise in the two branches. Here X^b_0 denotes the augmented clean target; known motif coordinates receive the same branch transformation. The EDM preconditioning coefficients are unchanged from Eq. 10.

A.3.2 Cross-branch communication

Token attention. Let f^b_i indicate a residue with a known motif identity and backbone, and define the free correspondence set $\mathcal{P}^{\text{free}}_u = \{(i, \pi(i)) \in \mathcal{P}_u : f^A_i = f^B_{\pi(i)} = 0\}$. For valid tokens r, t in a packed row, let $b(r)$ be the branch identifier and let $\mathcal{E}_{\text{free}}$ contain both directions of these correspondence edges. The token-attention and geometric pair masks are

$$M^{\text{token}}_{rt} = \mathbf{1}[b(r) = b(t) \text{ or } (r, t) \in \mathcal{E}_{\text{free}}], \qquad M^{\text{pair}}_{rt} = \mathbf{1}[b(r) = b(t)]. \tag{18}$$

Matched tokens participate in the same attention softmax as within-branch keys. Pair representations and triangle operations remain within each branch. Correspondence edges use the zero-offset relative-position bucket and carry no inter-branch geometric feature; within-branch positional encoding follows Appendix A.1.1. Padding is masked, and register tokens remain local to their branch.

Atom residue consensus. The atom-communication schematic in Figure 3c is expanded below. For atom states h^b_{ia} at an Atom DiT block, with slot $a \in \{1, \dots, 14\}$, define

$$\begin{aligned} u^b_{ia} &= \text{RMSNorm}_{\text{cons}}\big(\text{RMSNorm}_1(h^b_{ia})\big), \\ q^b_i &= W_q \, \text{RMSNorm}_q\left(\frac{1}{14}\sum_a u^b_{ia} + W_t e_{\sigma_u}\right). \end{aligned} \tag{19}$$

The input uses the block's normalized states before AdaLN modulation; e_{σ_u} is the noise embedding alone. For each free pair (i, j), set $\bar{q}^A_i = \bar{q}^B_j = (q^A_i + q^B_j)/2$; other residues use $\bar{q}^b_i = q^b_i$. Each branch then reads its own 14 atom slots. For one attention head,

$$k^b_{ia} = W_k u^b_{ia}, \qquad v^b_{ia} = W_v u^b_{ia}, \qquad m^b_i = \sum_{a=1}^{14} \text{softmax}_a\left(\frac{\bar{q}^b_i \cdot k^b_{ia}}{\sqrt{d_h}}\right) v^b_{ia}. \tag{20}$$

Four heads with $d_h = 16$ are concatenated into a 64-dimensional local message, also denoted m^b_i. Corresponding messages are averaged before a shared projection:

$$\bar{m}^A_i = \bar{m}^B_j = \tfrac{1}{2}(m^A_i + m^B_j), \qquad \delta^b_i = W_o \, \text{LayerNorm}(\bar{m}^b_i). \tag{21}$$

Unlinked residues use their own local message. The same δ is broadcast to all 14 slots of both linked residues, alongside ordinary within-branch atom attention and before the feed-forward sublayer (Algorithm 4). It is not multiplied by the branch-specific AdaLN attention gate. Every Atom DiT block has its own consensus parameters, with W_o initialized to zero. The two branches retain separate atomic states; sequence agreement is learned rather than imposed by a discrete sequence replacement.

Algorithm 4 One Atom DiT block with residue consensus

Require: Atom states h, branch conditions c, noise embedding e_σ, free correspondences $\mathcal{P}^{\text{free}}$

$n \leftarrow \text{RMSNorm}_1(h)$
$u \leftarrow \text{RMSNorm}_{\text{cons}}(n)$
$q \leftarrow W_q \,\text{RMSNorm}_q(\text{Mean}_{14}(u) + W_t e_\sigma)$
$\bar{q} \leftarrow \text{AverageLinked}(q, \mathcal{P}^{\text{free}})$
$k \leftarrow W_k u$; $v \leftarrow W_v u$
$m \leftarrow \text{LocalAtomAttention}(\bar{q}, k, v)$ ▷ 14 slots per residue
$\bar{m} \leftarrow \text{AverageLinked}(m, \mathcal{P}^{\text{free}})$
$\delta \leftarrow \text{Broadcast}_{14}(W_o \,\text{LayerNorm}(\bar{m}))$
$(\alpha_1, \beta_1, \gamma_1, \alpha_2, \beta_2, \gamma_2) \leftarrow \text{AdaLN}(c)$
$h \leftarrow h + \alpha_1 \odot \text{Attention}_{\text{branch}}((1+\gamma_1) \odot n + \beta_1) + \delta$
$h \leftarrow h + \alpha_2 \odot \text{FFN}((1+\gamma_2) \odot \text{RMSNorm}_2(h) + \beta_2)$
return h, m ▷ m supplies auxiliary identity supervision

`AverageLinked` averages linked residues and leaves other residues unchanged. Token DiT precedes the atom stack, as in Appendix A.1.1.

Table 7: SS visibility modes during training. "Full" exposes all valid, non-motif positions. A pair window means one independently sampled local window per branch.

Training unit	All NULL	Local window	Full	Shared-span subset
Individual protein	0.20	0.60	0.20	–
Self-copy / real pair	0.10	0.30	–	0.60

A.3.3 Secondary-structure conditioning

Secondary-structure labels are read from full-parent annotations and gathered using the crop indices. For each residue, $c_i^b \in \{H, E, C, \text{NULL}\}$ specifies helix, strand, coil, or an unspecified state. The NULL embedding is zero and is distinct from coil. Let r_i^b indicate membership in the designated shared segment. The residue-level condition combines noise, shared-region, SS, and motif indicators:

$$e_i^b = e_{\sigma_u} + E_{\text{shared}}(r_i^b) + E_{\text{SS}}(c_i^b) + E_{\text{motif}}(f_i^b). \tag{22}$$

Projected versions of e_i^b modulate Token DiT and Atom DiT through AdaLN; the atom condition also incorporates the final token single representation, as in the base model.

Label visibility is sampled according to Table 7. For a local window, we first choose uniformly among the available H/E/C classes, then uniformly select an anchor from that class. The window length is 9 or 10 with probabilities 0.93 and 0.07, clipped to the valid continuous sequence run containing the anchor. Pair windows are drawn independently in the two branches. In shared-span mode, q is chosen uniformly from $\{0.25, 0.5, 0.75, 1\}$, and corresponding positions are revealed with probability q. The two branches expose the same offsets but their own SS labels. At least two jointly valid positions are exposed when available; if the labels differ at any eligible position, at least one such position is included. Real-pair A/B order is swapped with probability 0.5. After motif selection, SS visibility is resampled over eligible non-motif residues, preserving the selected mode and q. This supplies conditions over free regions without revealing redundant SS labels at known motif positions.

A.3.4 Motif conditioning

Selecting known regions. Motif conditioning is requested independently for each logical unit with probability 0.5. We request one, two, or three continuous segments with probabilities 0.8, 0.15, and 0.05. For a branch of length L, the total motif length is drawn uniformly.

Encoding the conditions. Known motif backbone coordinates and amino acid identities enter a separate conditioning encoder. For each atom slot, the encoder concatenates coordinates divided by 16, a 16-dimensional slot embedding, a backbone-availability bit, and a 32-dimensional amino acid embedding. A two-layer MLP of width 128 with SiLU activations produces additive atom features

and, after averaging the 14 slots, additive token features. The output projections are initialized to zero. Distances between known motif Cα atoms within a branch are encoded using 128 bins, with 127 boundaries evenly spaced from 1 to 30 Å, plus an availability bit, and projected into the pair representation with a zero-initialized linear map.

A.3.5 PAIRED TRAINING OBJECTIVES

Base reconstruction and frozen sequence readout. We retain coordinate, smooth-lDDT, sequence, and distogram terms with weights 1, 0.25, 0.05, and 0.01, respectively. Their per-branch definitions follow Appendix A.1.2, with the coordinate target evaluated directly in the augmented branch frame: Eq. 12 uses X_0 in place of the rigidly fitted target $\widetilde{X}_0$. The smooth-lDDT term uses within-branch atom14 pairs, including diagonal entries and without a distance cutoff; the distogram target uses Cα distances. Chemical-padding slots at Cα are retained under the atom14 loss convention, while unavailable named atoms are masked from coordinate supervision.

The pretrained sequence readout G_ϕ is frozen throughout fine-tuning, while gradients through its coordinate input remain enabled. For the base sequence loss, each branch uses predicted coordinates with probability 0.5 when $\sigma_u < 4$, and clean coordinates otherwise. The prediction path updates the denoiser; the clean-coordinate path through the frozen readout contributes a loss value but no denoiser gradient. The shared-identity terms below always read the predicted coordinates.

Message identity supervision. A trainable classifier H_ψ, consisting of LayerNorm followed by a 20-class linear head, reads the local message $m_i^{b,t}$ before cross-branch averaging at atom block t. The classifier is shared across the $T = 16$ blocks. Let $\mathcal{V}_u^b$ contain valid residues whose identities are not provided as motif conditions. For a nonempty branch,

$$\begin{aligned} \ell_{\mathrm{msg},u}^b &= \frac{w_{\mathrm{msg}}(\sigma_u)}{T|\mathcal{V}_u^b|} \sum_{t=1}^{T} \sum_{i \in \mathcal{V}_u^b} \mathrm{CE}\Big(H_\psi(m_i^{b,t}), s_i^b\Big), \\ w_{\mathrm{msg}}(\sigma) &= \mathrm{sigmoid}\left(\frac{\log 5 - \log \sigma}{0.5}\right). \end{aligned} \tag{23}$$

This supervision covers individual proteins and free residues outside shared spans as well as shared positions. Local messages receive identity supervision without feeding the classifier's discrete predictions back into the denoiser.

Identity from predicted coordinates. The frozen readout gives $p_i^b = \mathrm{softmax}(G_\phi(\hat{X}_0^b)_i)$ at each residue. For a correspondence (i, j), let $\bar{p}_{ij} = (p_i^A + p_j^B)/2$. Let $\mathcal{Q}_u \subseteq \mathcal{P}_u^{\mathrm{free}}$ contain correspondences with valid identity-supervision masks in both branches. Each correspondence is counted once. For an eligible pair unit,

$$\begin{aligned} D_{\mathrm{JS}}(p_i^A, p_j^B) &= \frac{\mathrm{KL}(p_i^A \| \bar{p}_{ij}) + \mathrm{KL}(p_j^B \| \bar{p}_{ij})}{2 \log 2}, \\ \ell_{\mathrm{JS},u} &= \frac{w_{\mathrm{JS}}(\sigma_u)}{|\mathcal{Q}_u|} \sum_{(i,j) \in \mathcal{Q}_u} D_{\mathrm{JS}}(p_i^A, p_j^B), \\ \ell_{\mathrm{GT},u} &= -\frac{w_{\mathrm{GT}}(\sigma_u)}{2|\mathcal{Q}_u|} \sum_{(i,j) \in \mathcal{Q}_u} \Big[\log(p_i^A)_{s_i^A} + \log(p_j^B)_{s_i^A}\Big], \end{aligned} \tag{24}$$

where $s_i^A = s_j^B$ and

$$w_{\mathrm{JS}}(\sigma) = \frac{1}{1 + (\sigma/4)^2}, \qquad w_{\mathrm{GT}}(\sigma) = \frac{1}{1 + (\sigma/2)^2}. \tag{25}$$

The normalized JS term encourages agreement, while the ground-truth term anchors both distributions to the shared training identity.

Table 8: Configuration of the paired fine-tuning protocol. The global batch is measured in packed rows, each containing one or more logical units.

Setting	Value
Initialization / additional steps	190k pretraining EMA / 30k
Individual / self-copy / real pair	50% / 25% / 25% throughout
Motif request probability	0.5 per logical unit
Token / atom blocks; heads	16 / 16 blocks; 12 heads each
Token / atom / pair widths	768 / 384 / 256
Consensus width / heads	64 / 4; zero-initialized output projection
Hardware / global batch	16 H20 GPUs / 64 packed rows
Rows per GPU / accumulation	4 / 1
Packing capacity	512 residues and at most 16 branches per row
Optimizer	Muon for selected matrix weights; AdamW for remaining trainable parameters
Learning rate / schedule	5×10^{-5}; 200-step warmup, then constant
Muon settings	Momentum 0.95, Nesterov, 5 Newton–Schulz steps, weight decay 0.01
AdamW settings	$\beta = (0.9, 0.95)$, $\epsilon = 10^{-8}$, zero weight decay
Gradient clipping / precision	Global norm 1.0 / bf16 mixed precision
EMA / activation checkpointing	Decay 0.999 / enabled

Reduction and total objective. Each base term is first normalized over its valid atoms, residues, or residue pairs within a branch. For a global batch of logical units $\mathcal{U}$, the base reduction is

$$\mathcal{L}_{\text{base}} = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \frac{1}{|\mathcal{B}_u|} \sum_{b \in \mathcal{B}_u} \left(\mathcal{L}^b_{\text{coord}} + 0.25 \mathcal{L}^b_{\text{smooth-lDDT}} + 0.05 \mathcal{L}^b_{\text{seq}} + 0.01 \mathcal{L}^b_{\text{dist}} \right). \tag{26}$$

Message losses are averaged over nonempty branches within a unit and then equally over eligible units, giving $\mathcal{L}_{\text{message-ID}}$. The JS and GT losses are averaged over eligible pair units only, giving $\mathcal{L}_{\text{shared-JS}}$ and $\mathcal{L}_{\text{shared-GT}}$; individual proteins do not enter those denominators. An empty eligible set contributes zero. Noise weights multiply the loss values without changing the averaging denominators. All unit means use global counts across devices, so two-branch units do not receive twice the sample weight of individual proteins. Equation 3 uses $\lambda_{\text{msg}} = 0.01$, $\lambda_{\text{cons}} = 0.02$, and $\lambda_{\text{gt}} = 0.01$.

A.3.6 Fine-tuning protocol

The protocol initializes from the EMA weights of the same 190k all-atom checkpoint evaluated in Section 4.1 and uses 30k additional optimizer steps. The task mixture is held at individual/self-copy/real-pair $= 50/25/25\%$, with motif request probability 0.5 throughout. Shuffled task queues implement these proportions before packing. Table 8 gives the optimization and model settings.

A.4 Generation and evaluation protocols

This section specifies the generation protocols and computational metrics used in Sections 4.1 and 4.2. Additional results and structure examples are collected in Appendix A.5; motif-specific case-selection criteria are given in Appendix A.7.

A.4.1 Sampling, baseline implementation, and refolding

Unconditional generation. We evaluate the 190k-step pretrained AtomDiff-X model described in Appendix A.1. For each sampling configuration, we generate 100 proteins at each length $L \in \{100, 200, 300, 400, 500\}$, for 500 samples in total. Each sample starts from Gaussian atom14 coordinates and follows one generation trajectory. The sequence is decoded by taking the per-residue argmax of the coordinate-based readout at the final denoising iteration. The generated sequences are used directly, without inverse-folding redesign, relaxation, or resampling based on refolding results. Baseline numbers in Table 1 are taken from Geffner et al. (2026); those models were not rerun for this comparison.

AtomDiff sampling. Our coordinate samplers use a power-law noise grid following the EDM parameterization (Karras et al., 2022):

$$\sigma_j = \sigma_{\mathrm{data}} \left[s_{\max}^{1/\rho} + \frac{j}{N-1} \left(s_{\min}^{1/\rho} - s_{\max}^{1/\rho} \right) \right]^{\rho}, \qquad j = 0, \ldots, N-1. \tag{27}$$

Here $\sigma_{\mathrm{data}} = 16$ and $s_{\min}, s_{\max}$ are noise levels relative to this scale. Table 9 lists the settings. The unconditional samplers update between the 500 positive grid points; the paired sampler additionally includes a final update to zero noise. For an update from σ_j to σ_{j+1}, let D_θ denote the preconditioned coordinate denoiser, with the relevant conditions suppressed in the notation. After centering and random rigid augmentation, we apply

$$\begin{aligned} \gamma_j &= \gamma_0 \, \mathbf{1}[\sigma_{j+1} > \gamma_{\min}], & \widehat{\sigma}_j &= (1+\gamma_j)\sigma_j, \\ \widehat{X}_j &= X_j + S_{\mathrm{noise}} \sqrt{\widehat{\sigma}_j^2 - \sigma_j^2}\, \epsilon_j, & \epsilon_j &\sim \mathcal{N}(0, I), \\ X_{j+1}^{\mathrm{pred}} &= \widehat{X}_j + a(\sigma_{j+1} - \widehat{\sigma}_j) \frac{\widehat{X}_j - D_\theta(\widehat{X}_j, \widehat{\sigma}_j)}{\widehat{\sigma}_j}. \end{aligned} \tag{28}$$

The Euler sampler takes $X_{j+1} = X_{j+1}^{\mathrm{pred}}$. For paired generation, the same update is applied jointly to the two branches, with independent Gaussian noise and rigid augmentations and a shared noise schedule.

Table 9: Coordinate-sampling settings. The endpoints $s_{\min}$ and $s_{\max}$ are relative to $\sigma_{\mathrm{data}} = 16$ in Equation 27. The PC correction is defined below.

Parameter	Unconditional Euler	Unconditional PC	Paired generation
Positive grid points N	500	500	500
$s_{\max}$ / $s_{\min}$	160 / 0.002	160 / 0.002	160 / 0.001
ρ	7	7	7
Churn γ_0	0.4	0.2	0.2
Churn cutoff $\gamma_{\min}$	1	1	1
Noise multiplier S_{noise}	1.003	1.003	1.003
Step multiplier a	1.5	1.0	1.5
PC correction	None	$\eta_j \le 0.2$	None

The stochastic PC sampler evaluates $D_j^+ = D_\theta(X_{j+1}^{\mathrm{pred}}, \sigma_{j+1})$ and uses

$$X_{j+1} = X_{j+1}^{\mathrm{pred}} + \eta_j (D_j^+ - X_{j+1}^{\mathrm{pred}}) + 0.05\, \sigma_{j+1}\, \mathbf{1}[\eta_j > 0]\, \epsilon_j', \qquad \epsilon_j' \sim \mathcal{N}(0, I). \tag{29}$$

We set $\eta_j = 0.2$ when $\sigma_{j+1} > 1$ and zero otherwise. For $L \ge 500$ and $\sigma_{j+1} > 2$, this value is capped at $\max(0, 1 - 0.6/\max(r_j, 10^{-12}))$, where $r_j = \mathrm{RMS}(X_{j+1}^{\mathrm{pred}} - D_j^+)/\sigma_{j+1}$ and the RMS averages over the three Cartesian components of all atom14 slots. This caps the correction toward the denoised prediction when its residual is already small relative to the next noise level.

Short shared-segment inputs. The paired benchmark contains 1,000 cases, with 200 cases for each $k \in \{6, 7, 8, 9, 10\}$. Both proteins have length 96, including the shared segment. In each branch, the one-based start position is sampled independently and uniformly from $\{2, \ldots, 96 - k\}$, leaving at least one residue before and after the segment. The k positions are paired in N-to-C order. To construct SS conditions, we uniformly sample an eligible natural chameleon pair, select a contiguous aligned k-residue window uniformly within its shared sequence, and exchange the two template branches with probability 0.5. Each branch receives its own H/E/C labels throughout the designated segment; all other positions are NULL. This procedure preserves contiguous, potentially mixed SS patterns and permits identical SS strings in the two branches.

Long shared-segment inputs. We generate 400 cases for each $k \in \{12, 24, 36, 48\}$, for 1,600 pairs in total. Chain lengths, random internal placement, and residue correspondence follow the short-segment protocol. We select 80 input SS-template pairs without replacement from each of the five original lengths $m \in \{6, 7, 8, 9, 10\}$, obtaining 400 template pairs reused across the four target lengths. For each case, an offset is sampled uniformly from $\{0, \ldots, k - m\}$, and the two m-residue

SS strings are inserted at this same relative offset within their respective shared segments. Only these m positions per branch are conditioned; all remaining positions are NULL. Sequence correspondence still covers the full k residues. Thus, the longer shared sequences are design targets, while the SS inputs are short windows drawn from the original benchmark.

For both cohorts, AtomDiff-Chameleon uses EMA weights and one joint trajectory per case, with the sampling settings in Table 9. The inputs specify lengths, residue correspondences, and SS labels, without providing a target sequence, coordinates, or motif. The two sequences are read out independently from the generated coordinates; shared identities are not enforced by overwriting either sequence. Each method receives the same case definitions within a cohort.

ProteinGenerator. We use the official pretrained ProteinGenerator weights (Lisanza et al., 2025), without task-specific retraining, and coordinate sequence generation over the designated shared segment. The two branches maintain separate structural trajectories and independent sequence states outside that segment. At each step, we average amino acid logits at corresponding shared positions and tie both their noisy sequence states and sequence self-conditioning. Consequently, exact shared-sequence identity is enforced during inference. The full shared segment is tied in both cohorts, including unconditioned positions in the long-segment experiments. Input coil and NULL labels are mapped to ProteinGenerator's L and X labels, respectively.

Each case uses one trajectory with 25 diffusion timesteps, four RoseTTAFold recycles per branch per timestep, and sampling temperature 1.0. We evaluate the common trajectory snapshot with the highest mean native predicted pLDDT across A and B, rather than choosing a separate snapshot for each branch. This selection does not use ESMFold or refolding scores. Atom coordinates are reconstructed from the sampled identities and each branch's backbone and torsions, without ProteinMPNN redesign or relaxation. The comparison matches task inputs and the number of trajectories, but uses model-specific noise, computation, and output-selection procedures.

Refolding. Every raw generated sequence is independently folded with ESMFold (Lin et al., 2023), using the `facebook/esmfold_v1` implementation in evaluation mode and its default recycling configuration. A and B are folded as separate single chains. The resulting structure is compared with the generated structure for that same sequence. No sequence replacement, post-refolding relaxation, or score-based regeneration is applied.

A.4.2 Evaluation metrics and reporting rules

Structural self-consistency. Let X be a generated structure and Y the structure predicted from its own generated sequence. Residues are matched by sequence position. We obtain a proper rotation Q^* and translation t^* by minimizing the squared error over corresponding Cα atoms using Kabsch alignment:

$$(Q^*, t^*) = \underset{Q\in \mathrm{SO}(3),\, t\in\mathbb{R}^3}{\operatorname{argmin}} \sum_{i\in\mathcal{I}_{\mathrm{CA}}} \|Qy_{i,\mathrm{CA}} + t - x_{i,\mathrm{CA}}\|_2^2 . \tag{30}$$

The Cα scRMSD is the square root of the mean squared residual over $\mathcal{I}_{\mathrm{CA}}$ after this alignment. In the paired benchmarks, all 96 corresponding Cα atoms are used for each branch, without outlier removal or restriction to the shared segment.

For all-atom scRMSD, let $\mathcal{H}(s)$ contain the chemically defined, named heavy atoms for the generated sequence that are present in both structures. Using the same Cα-derived transform, we compute

$$R_{\mathrm{all}} = \sqrt{\frac{1}{|\mathcal{H}(s)|} \sum_{(i,r)\in\mathcal{H}(s)} \|Q^* y_{ir} + t^* - x_{ir}\|_2^2}. \tag{31}$$

Virtual atom14 slots and hydrogens are excluded. No additional heavy-atom alignment or side-chain atom-permutation optimization is performed. Section 4.1 defines all-atom co-designability by $R_{\mathrm{all}} < 2$ Å; its reported rate is the fraction of all 500 generated samples passing this criterion. Section 4.2 instead uses full-chain Cα scRMSD, denoted R_A and R_B for the two branches.

Diversity. We pool the all-atom co-designable generated structures across the five unconditional lengths and report the number of clusters in this pooled set. The structural (Str) and joint sequence–structure (Seq+Str) counts use Foldseek `easy-cluster` with the settings in Table 10; sequence

(Seq) counts use MMseqs2. Seq+Str denotes clustering with Foldseek's combined sequence–structure alignment mode, rather than an intersection of separately computed cluster assignments. Larger counts indicate broader coverage under the selected clustering criteria, but also depend on the number of co-designable samples available for clustering.

Table 10: Clustering settings in the unconditional evaluation code. All three metrics operate on the pooled co-designable subset.

Metric	Settings
Str	Foldseek: alignment type 1, TM-score threshold 0.5, minimum sequence identity 0, coverage mode 0.
Seq+Str	Foldseek: alignment type 2, TM-score threshold 0.5, minimum sequence identity 0.1, coverage mode 0.
Seq	MMseqs2: minimum sequence identity 0.1, coverage threshold 0.7, coverage mode 1.

Shared-sequence agreement. For a pair with correspondence set $\mathcal{P}$ of size k, define

$$\text{Identity} = \frac{1}{k} \sum_{(i,j)\in\mathcal{P}} \mathbf{1}[s_i^A = s_j^B], \qquad \text{Exact} = \bigwedge_{(i,j)\in\mathcal{P}} [s_i^A = s_j^B]. \tag{32}$$

These measure agreement between the two generated segments, not recovery of a natural reference sequence. We also report each branch's refolding success and the event that both $R_A < 2$ Å and $R_B < 2$ Å hold for the same pair. The latter is computed directly per pair, rather than by multiplying the two marginal success rates. Combining it with Exact gives Success in Equation 4.

Secondary-structure agreement. We assign three-state SS labels to each full generated backbone using PyDSSP on its N, Cα, C, and O coordinates, mapping its assignments to H/E/C. Let Ω_b be the positions with an explicitly supplied SS condition in branch $b \in \{A, B\}$, c_i^b their input labels, and $\widehat{c}_i^b$ the labels assigned to the generated structure. We compute

$$q_b = \frac{1}{|\Omega_b|} \sum_{i\in\Omega_b} \mathbf{1}[\widehat{c}_i^b = c_i^b], \qquad C = \min(q_A, q_B). \tag{33}$$

NULL positions are excluded; an explicitly provided coil label is included. All benchmark cases have nonempty conditioned sets in both branches. For short segments, $|\Omega_b| = k$; for long segments, $|\Omega_b| = m$, the length of the inserted SS window, rather than the full shared length.

SS80 requires $q_A \geq 0.8$ and $q_B \geq 0.8$ separately. We implement this inclusive threshold by comparing integer counts: $5n_{\text{match}}^b \geq 4|\Omega_b|$ for each branch. This includes cases with exactly 80% agreement and avoids decisions based on rounded percentages. Combining SS80 with Success yields $\text{Success}_{\text{SS80}}$ in Equation 5. SS is evaluated on the generated structures; preservation of these labels after refolding is not an additional success requirement. The benchmark also imposes no separate threshold on the local structural difference between A and B.

A.5 Additional computational results

We report length-resolved results and selected structures for the evaluations in Sections 4.1 and 4.2. Sampling and metric definitions follow Appendix A.4.

A.5.1 Unconditional generation and structure examples

Table 11 resolves co-designability and structural diversity by length. Both samplers produce co-designable proteins throughout the 100–500-residue range. The largest PC gain occurs at $L = 100$ (60% to 82%), while Euler performs better at $L = 400$.

Figure 10 shows one uniformly sampled co-designable structure per sampler–length group, overlaid with its ESMFold refold.

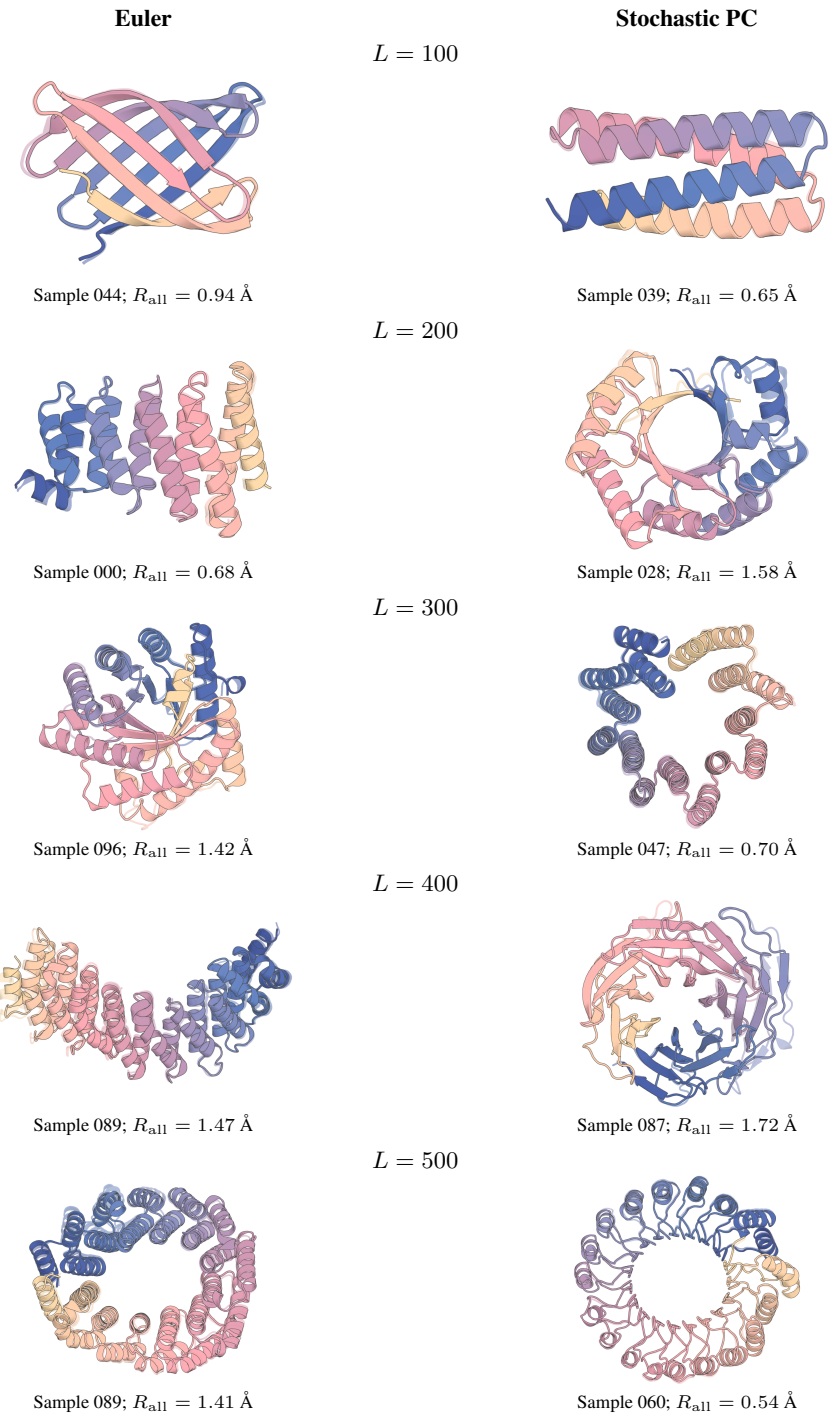


Figure 10: **Unconditional AtomDiff-X examples.** One all-atom co-designable sample is drawn uniformly from each sampler. Generated structures are overlaid with translucent ESMFold refolds using the full-chain Cα alignment from Appendix A.4.2. Both use the same sequence-order color gradient from pale yellow to dark blue. R_{all} is the all-heavy-atom scRMSD under that alignment.

Table 11: **Unconditional generation by length.** Each sampler generates 100 proteins per length. Co-designability (%) requires all-heavy-atom scRMSD below 2 Å. Str counts structural clusters among co-designable samples. The pooled row uses all 500 samples per sampler.

	Euler		Stochastic PC	
L	Co-designability	Str	Co-designability	Str
100	60.0	24	82.0	24
200	52.0	9	63.0	7
300	61.0	9	62.0	7
400	54.0	10	49.0	6
500	43.0	5	49.0	5
Pooled	54.0	52	61.0	45

A.5.2 Paired chameleon benchmark

Table 12 decomposes performance into shared-sequence agreement, refolding of each branch, and their joint success. ProteinGenerator enforces exact identity during sampling, so its both-chain refolding rate equals its Success rate. AtomDiff-Chameleon decodes the two sequences independently and achieves 96.25% mean residue identity and 75.1% exact shared-sequence agreement. Of 1,000 pairs, 332 satisfy both-chain refolding, 256 also satisfy exact shared identity, and 143 additionally meet the SS80 criterion. The corresponding ProteinGenerator counts are 71, 71, and 39.

Table 12: **Short shared-segment benchmark.** Each method generates 200 pairs per length and 1,000 pairs overall. All entries are percentages. PG denotes ProteinGenerator; its shared identity is enforced during sampling. A and B report $R_A < 2$ Å and $R_B < 2$ Å, respectively; AB requires both. S and S_{80} denote Success and $\text{Success}_{\text{SS80}}$.

k	Method	Identity	Exact	A	B	AB	S	S_{80}
6	PG	100.00	100.00	29.00	23.50	6.50	6.50	4.00
6	Ours	95.92	80.00	60.00	62.00	38.50	31.50	18.00
7	PG	100.00	100.00	25.50	23.00	9.50	9.50	3.50
7	Ours	96.36	78.50	54.50	64.50	34.50	28.00	15.00
8	PG	100.00	100.00	26.50	18.50	7.00	7.00	4.00
8	Ours	96.19	75.00	52.50	51.50	29.50	24.00	15.50
9	PG	100.00	100.00	29.50	20.50	8.00	8.00	5.00
9	Ours	96.17	71.00	56.50	58.50	34.00	25.00	10.50
10	PG	100.00	100.00	24.50	22.50	4.50	4.50	3.00
10	Ours	96.60	71.00	51.00	55.50	29.50	19.50	12.50
All	PG	100.00	100.00	27.00	21.60	7.10	7.10	3.90
All	Ours	96.25	75.10	54.90	58.40	33.20	25.60	14.30

Figure 11 shows one AtomDiff-Chameleon pair per tested length, sampled uniformly from the $\text{Success}_{\text{SS80}}$ subset.

A.5.3 Effect of paired supervision

To examine transfer from single-structure training to paired generation, we compare a single-only model with a paired reference after 30k training updates from the same 190k-step pretrained initializer. The single-only model retains the communication architecture, local message identity supervision, and single-structure SS conditioning, but receives neither self-copy nor real chameleon pairs and has no pair-specific identity losses. During paired evaluation, the prescribed correspondences activate token communication and residue-consensus fusion. Both models receive the same 1,000 short-segment cases under matched sampling settings. The paired reference in Table 13 is distinct from the main model reported in Table 3.

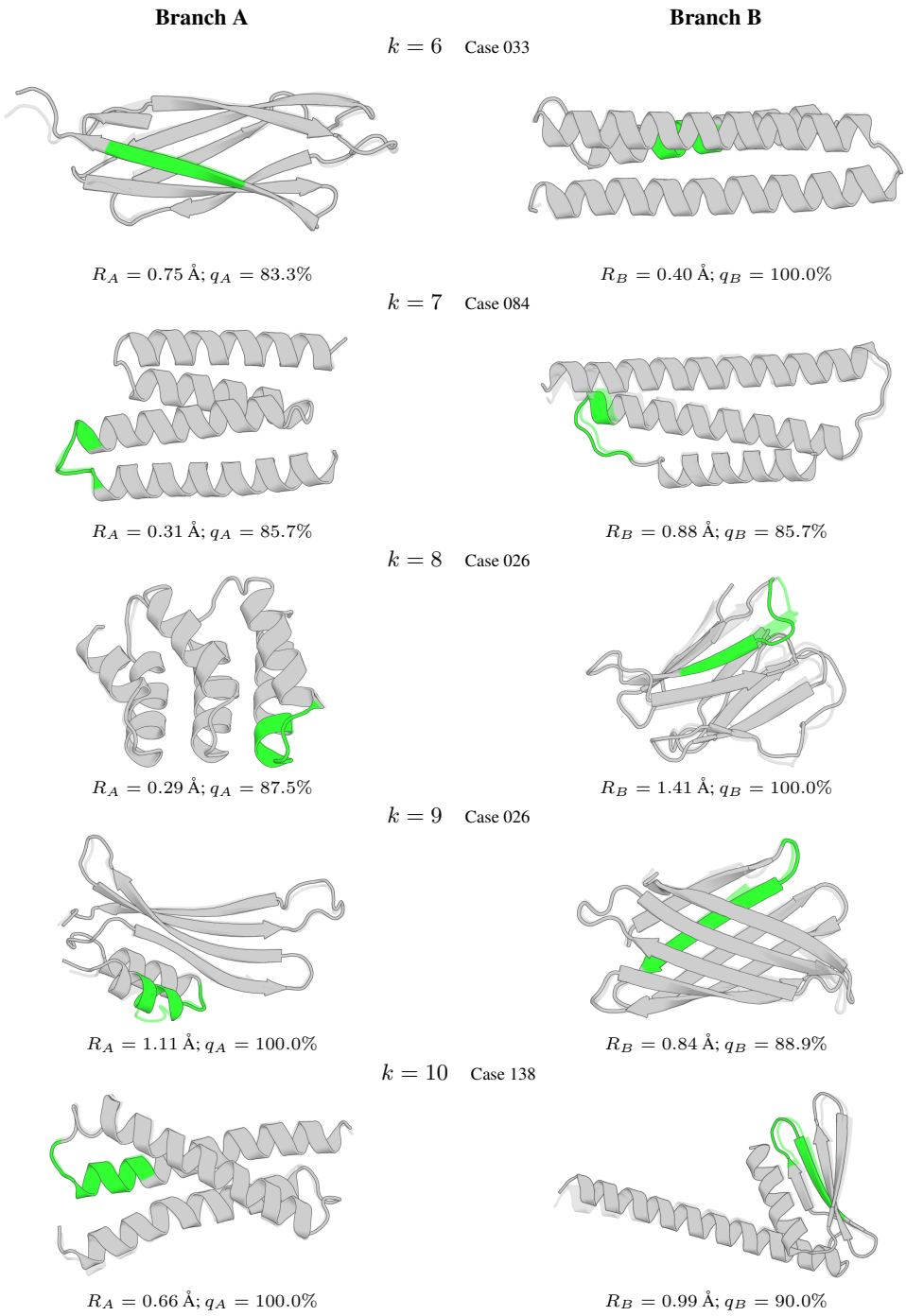


Figure 11: **Short shared-segment examples.** One AtomDiff-Chameleon pair is sampled uniformly from the $\text{Success}_{\text{SS80}}$ cases at each length. Each generated protein is overlaid with its translucent ESMFold refold, aligned independently using all 96 Cα atoms. Both layers use light gray for context and bright green for the shared segment, following Figure 4. The two branches share the full designated sequence. R_b reports Cα scRMSD; q_b reports generated-structure SS agreement at conditioned positions.

Table 13: **Single-only training and paired supervision.** Results on the same 1,000 short-segment inputs after 30k updates. All entries are percentages. The single-only column corresponds to "Ours, w/o pairs" in Table 3; the second column is the paired reference for this comparison.

Metric	Single-only	Paired reference
Residue identity	8.52	95.41
Exact	0.00	70.60
$R_A < 2$ Å	49.60	45.60
$R_B < 2$ Å	54.60	49.30
Both-chain refolding	26.50	23.20
Success	0.00	16.40
$\text{Success}_{\text{SS80}}$	0.00	10.50
Mean joint SS score C	79.39	80.17

Single-only training preserves individual-chain refolding and yields both-chain refolding in 26.5% of pairs, but shared residue identity is only 8.52% and no pair has an exact shared sequence. The paired reference achieves 70.6% exact agreement and 16.4% joint success. This comparison supports the contribution of paired supervision to cross-context sequence agreement, beyond the ability to generate individually self-consistent structures.

A.5.4 Generalization to longer shared segments

The long-segment evaluation uses 400 pairs at each $k \in \{12, 24, 36, 48\}$, with the random internal placement and partial SS conditions defined in Appendix A.4.1. Sequence correspondence covers all k positions, while SS is specified only in an embedded 6–10-residue window. Table 14 reports the component metrics. AtomDiff-Chameleon achieves 125/1,600 Success cases and 73/1,600 $\text{Success}_{\text{SS80}}$ cases, compared with 60/1,600 and 22/1,600 for ProteinGenerator.

Table 14: **Long shared-segment benchmark.** Each method generates 400 pairs per length and 1,600 pairs overall. All entries are percentages, with column definitions as in Table 12. SS80 is evaluated only at the supplied SS-window positions.

k	Method	Identity	Exact	A	B	AB	S	S_{80}
12	PG	100.00	100.00	23.00	20.00	3.25	3.25	1.00
12	Ours	97.79	76.00	43.50	44.50	21.00	16.50	9.25
24	PG	100.00	100.00	21.25	19.25	5.75	5.75	2.25
24	Ours	98.23	67.75	34.50	36.50	13.25	9.00	5.75
36	PG	100.00	100.00	16.50	14.50	3.25	3.25	1.00
36	Ours	98.58	64.25	26.50	28.25	5.75	4.25	2.00
48	PG	100.00	100.00	13.75	11.50	2.75	2.75	1.25
48	Ours	98.46	55.25	18.75	18.75	4.00	1.50	1.25
All	PG	100.00	100.00	18.62	16.31	3.75	3.75	1.38
All	Ours	98.26	65.81	30.81	32.00	11.00	7.81	4.56

Mean per-residue identity remains above 97% at all four lengths, while exact identity and both-chain refolding become less frequent as k increases. Even among pairs with exact shared sequences, the fraction satisfying both-chain refolding decreases from 66/304 (21.7%) at $k = 12$ to 6/221 (2.7%) at $k = 48$. The decline therefore involves structural self-consistency as well as the stricter requirement of matching every residue in a longer segment.

These shared lengths exceed those in the curated real chameleon pairs, whose longest spans contain 11 residues. Because the total chain length remains 96, increasing k also reduces the private context and the fraction of the shared segment with specified SS. The experiment therefore tests generalization under longer sharing constraints, rather than isolating length as the only changing factor. Figure 12 displays examples satisfying $\text{Success}_{\text{SS80}}$ for $k = 12, 24, 36, 48$.

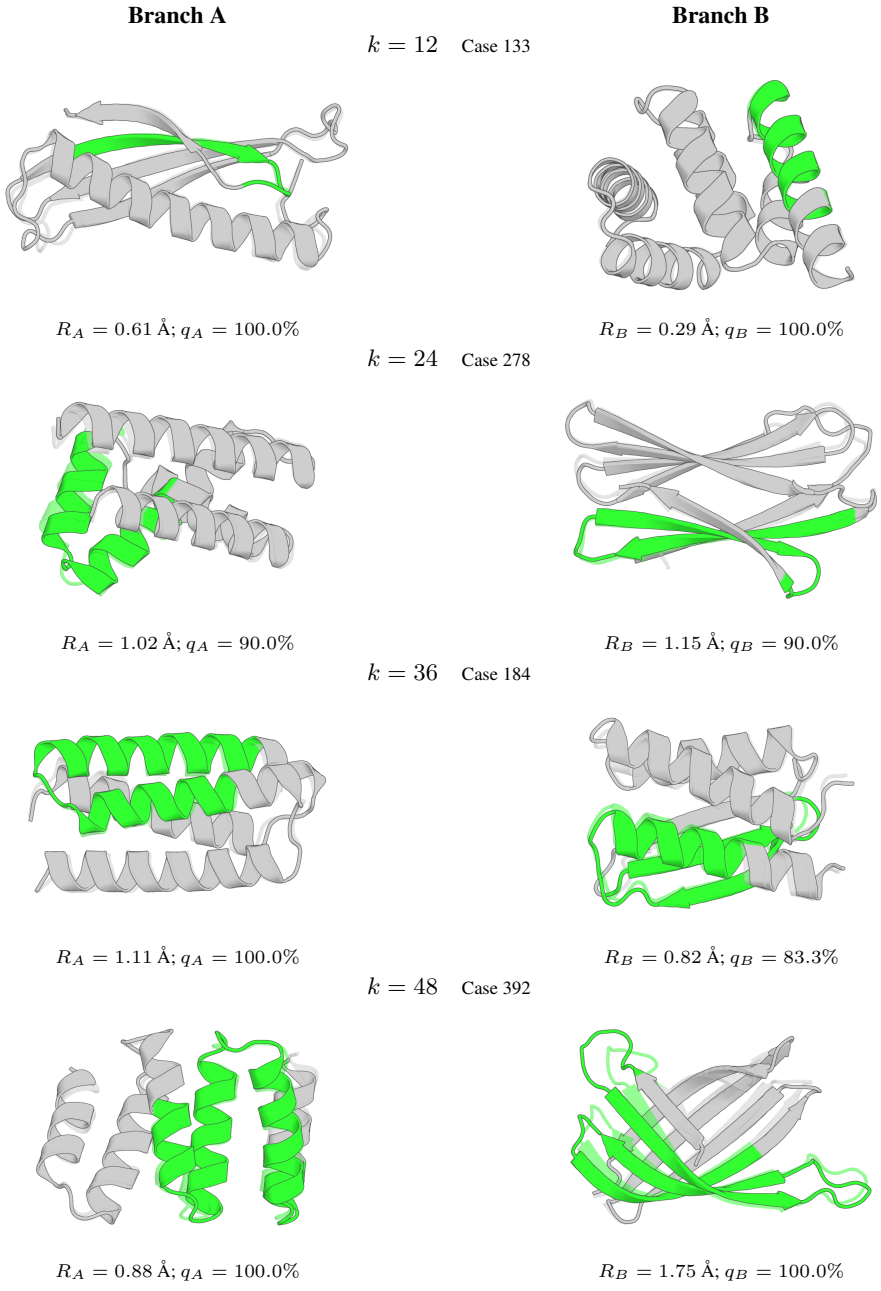


Figure 12: **Long shared-segment examples.** One AtomDiff-Chameleon pair satisfying $\text{Success}_{\text{SS80}}$ is shown per length. Each generated protein is overlaid with its translucent ESMFold refold, aligned independently using all 96 Cα atoms. Both layers use light gray for context and bright green for the shared segment, following Figure 4. The two branches share the full designated sequence. R_b reports Cα scRMSD; q_b reports generated-structure SS agreement at conditioned positions.

A.6 BIOCHEMICAL CHARACTERIZATION

We manually selected 19 AtomDiff-X designs spanning 100–500 residues for length coverage and visual structural diversity. Sequences were required to have an alanine fraction of at most 30%, without fixed scRMSD or pLDDT thresholds. Five ProteinMPNN redesigns of AtomDiff-X backbones were included as controls.

Following commercial gene synthesis, proteins were expressed in *E. coli* Rosetta (DE3) cells with 1 mM IPTG induction at 16°C for 16 h. Soluble fractions were purified by Ni-NTA affinity chromatography and assessed by SDS-PAGE. FPLC-SEC (Superdex 75 Increase 10/300, 0.75 mL/min) was used to assess solution oligomeric behavior. Experimental outcomes are summarized in Table 2.

A.6.1 INDIVIDUAL OUTCOMES AND CHARACTERIZATION DATA

Table 15 links the laboratory sample numbers to the finalized design identifiers and the reported outcomes. Fourteen all-atom designs and three ProteinMPNN controls were reported as soluble. One soluble sample, `adx-L100-04`, was lost during the SEC procedure and has no interpretable SEC classification.

A.7 MOTIF-CONDITIONED DESIGN CASE STUDIES

A.7.1 CASE INPUTS AND GENERATION SETTINGS

The case studies in Section 4.3 use constructs of the form M-X-A and M-X-B, where M supplies a reference sequence and structure, X is the designated shared segment, and A and B provide separately generated protein contexts. The motif-conditioning mechanism is described in Appendix A.3.4. Table 16 records the two case-specific input configurations used for Figure 4b. For each motif, 1,000 protein pairs were generated. The shared-segment length was sampled from 6–10 residues, and the two context lengths were sampled independently within the indicated ranges. The 3MLT motif contains 11 structurally contiguous residues within the PDB numbering range C304–318; gaps in the residue numbering do not indicate a break in the motif structure.

A.7.2 MOTIF PRESERVATION AND CASE-SELECTION CRITERIA

Each raw generated sequence is independently refolded with ESMFold. Let Exact_{M-X} denote complete sequence identity across the shared M-X region, and let R_A and R_B denote full-chain Cα scRMSDs between the generated and refolded structures, as defined in Appendix A.4.2. Motif preservation is measured by comparing the motif in each refolded structure with the input reference motif, yielding RMSDs r_M^A and r_M^B. The common criterion is

$$\begin{aligned}\mathrm{Success}_{\mathrm{motif}} = \mathrm{Exact}_{M-X} \wedge (R_A < 2\,\text{Å}) \wedge (R_B < 2\,\text{Å}) \\ \wedge (r_M^A < 1\,\text{Å}) \wedge (r_M^B < 1\,\text{Å}).\end{aligned} \tag{34}$$

For BH3 motif (1PQ1), we additionally require the functional sites are exposed in state A and buried in state B.

Table 15: **Individual expression and SEC records.** Samples 1–19 are AtomDiff-X designs; samples 20–24 are ProteinMPNN controls.

No.	Design ID	Length	Soluble	SEC annotation
1	adx-L100-01	100	Yes	Monomer
2	adx-L100-02	100	Yes	Monomer
3	adx-L100-03	100	Yes	Monomer
4	adx-L100-04	100	Yes	Sample lost
5	adx-L100-05	100	Yes	Monomer
6	adx-L100-06	100	Yes	Monomer
7	adx-L200-01	200	No	Not assigned
8	adx-L200-02	200	Yes	Monomer
9	adx-L200-03	200	Yes	Monomer
10	adx-L200-05	200	No	Not assigned
11	adx-L300-01	300	Yes	Monomer + oligomer
12	adx-L300-02	300	Yes	Monomer
13	adx-L300-03	300	No	Not measured
14	adx-L300-04	300	Yes	Monomer + oligomer
15	adx-L400-01	400	No	Not measured
16	adx-L400-03	400	Yes	Oligomer
17	adx-L500-01	500	No	Not measured
18	adx-L500-02	500	Yes	Monomer
19	adx-L500-03	500	Yes	Monomer
20	mpnn-L100-04	100	No	Not measured
21	mpnn-L200-04	200	Yes	Monomer
22	mpnn-L300-01	300	Yes	Monomer
23	mpnn-L400-01	400	No	Not measured
24	mpnn-L500-02	500	Yes	Oligomer

Sample 4 was soluble before SEC but was lost during the procedure. Samples 13, 15, 17, 20, and 23 had no target band and were not run on SEC.

Table 16: **Inputs for the two motif-conditioned case studies.** Residue ranges follow the supplied PDB numbering. Context lengths exclude M and X. NULL denotes an unspecified SS condition.

Input	3MLT case	1PQ1 case
Reference motif	Chain C, 304–318 (11 residues)	Chain B, residues 85–104
Shared X length	6–10 residues	6–10 residues
Each context length	30–90 residues	45–90 residues
Initial NULL positions in X	First four positions	First three positions
Remaining SS specification	Two consecutive positions chosen later in X: HH in A, EE in B; all other positions NULL	Two consecutive positions chosen later in X: HH in A, EE in B; all other positions NULL
Generated pairs	1,000	1,000