A-CODE: FULLY ATOMIC PROTEIN CO-DESIGN WITH UNIFIED MULTIMODAL DIFFUSION

Anonymous authors
Paper under double-blind review

ABSTRACT

We present A-CODE, a fully atomic unified one-stage protein co-design model that simultaneously refines discrete atom types and continuous atom coordinates. Unlike predominant two-stage methods that cascade structure design with amino acid-level sequence design, our approach is fully atomic within a unified multimodal diffusion framework, in which residue identities are inferred solely from atom-level predictions. Built upon the powerful all-atom architecture, A-CODE achieves superior designability for unconditional protein generation, outperforming all existing one-stage and two-stage design models. For binder design, A-CODE rivals and even outperforms existing state-of-the-art two-stage design models and, compared with the existing one-stage co-design model, achieves a drastic tenfold improvement in success rate on hard tasks. The inherent flexibility of our atomic formulation enables, for the first time, seamless adaptation to non-canonical amino acid (ncAA) modeling. Our fully atomic framework establishes a new, versatile foundation for all-atom generative modeling that can be naturally extended to complex biomolecular systems.

1 INTRODUCTION

Proteins are the fundamental molecular machines of life, orchestrating nearly every biological process through their intricate three-dimensional structures and dynamic interactions. The ability to design novel proteins with properties beyond those found in nature holds transformative potential for science and medicine, promising breakthroughs such as highly specific therapeutics and novel biocatalysts. However, as a protein's function is strictly dictated by its sequence of amino acids, which folds into a unique geometric configuration, navigating the vast combinatorial space of possible sequences remains a formidable computational challenge.

The landscape of computational biology was fundamentally altered by the emergence of AlphaFold 2 (AF2) (Jumper et al., 2021), which demonstrated that deep neural networks could learn the intricate mapping from sequence to structure with near-experimental accuracy and was further extended to possess the unified all-atom predictive power in AlphaFold 3 (Abramson et al., 2024) (AF3) more recently. This predictive breakthrough has catalyzed a new wave of generative approaches, with attempts to adapt such architectures originally designed for structure prediction into tools for protein design. Despite this progress, the prevailing paradigm for *de novo* design remains largely bifurcated.

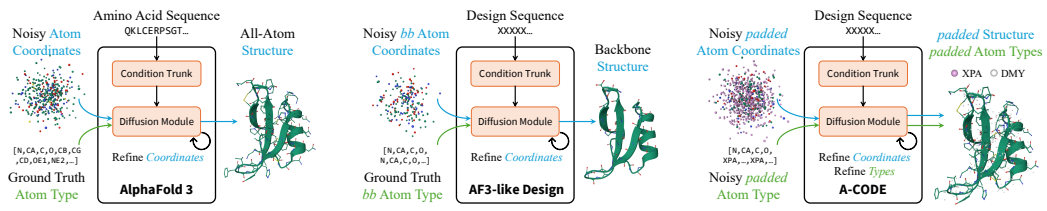


Figure 1: Comparison of the model frameworks of AlphaFold 3 (**left**), AlphaFold 3-like design model (**middle**), and our A-CODE (**right**) with atomic-scale co-design. Note how the initial noisy input features differ across the three models.

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.

Current state-of-the-art methods predominantly operate as *two-stage* cascades: they first generate a geometric backbone structure via diffusion or flow matching and subsequently employ a separate inverse folding model to condition on that fixed backbone and predict a compatible amino acid sequence. For example, PXDesign (Team et al., 2025) extends Protenix (Chen et al., 2025b) for backbone-only structure design. More recent works like RFDiffusion3 (Butcher et al., 2025) and BoltzGen (Stark et al., 2025) that attempt to leverage the all-atom power of AF3-like models still suffer from such sequential decoupling that prevents the model from capturing the full joint distribution of sequence and structure. The latent-based generative models, such as La-Proteina (Geffner et al., 2025) and PLAID (Lu et al., 2024), offer an alternative, more flexible approach but remain limited by the accuracy of their structure VAEs at recovering all-atom structures. On the other hand, a true *one-stage* framework capable of unified co-design, in which geometry and biochemical semantics are generated simultaneously in an end-to-end manner, remains a significant open challenge in the field. While recent attempts such as MultiFlow (Campbell et al., 2024) and Protpardelle (Chu et al., 2024) have been made, these models are generally less favorable for co-designability, as coordinating the structure and sequence modalities can introduce subtle trade-offs.

To address the challenges in protein structure–sequence co-design, we propose *A-CODE* (**A**tomic **CO**-**DE**sign), a unified one-stage framework that operates directly at the atomic scale. A-CODE combines mask diffusion for discrete atom types with standard diffusion for continuous atom coordinates, forming a unified multimodal diffusion process over atomic representations. During sampling, the atom types and coordinates are denoised simultaneously at each iterative diffusion step, enabling finer-grained interactions between modalities. Unlike existing co-design approaches that rely on residue-level type information, A-CODE infers residue identities purely from atom-level predictions. This atom-centric formulation removes hard constraints on residue types in existing work and naturally extends to modeling non-canonical amino acids (ncAAs) without architectural changes.

Table 1: Comparisons between existing protein co-design models. All-atom is for binder structures.

Model	All-Atom	One-Stage Co-Design	ncAA Modeling
MultiFlow [6]	✗	✓	✗
Protpardelle [12]	✓	✓	✗
ProteinGenerator [21]	✓	✓	✗
Pallatom [25]	✓	✗	✗
La-Proteina [15]	✓	✗	✗
RFDiffusion3* [5]	✓	✗	✗
BoltzGen* [29]	✓	✗	✗
PXDesign [30]	✗	✗	✗
A-CODE (Ours)	✓	✓	✓

*In principle, these methods support one-stage co-design, but redesigned sequences are used in their papers and codebases.

Through extensive experiments on unconditional protein generation and conditional binder design, together with an in-depth analysis of ncAA modeling behavior, we demonstrate that A-CODE achieves strong empirical performance and provides a flexible, unified foundation for all-atom generative modeling of diverse biomolecules. For unconditional protein design, A-CODE establishes a new state-of-the-art designability and codesignability performance, outperforming the second-best La-Proteina by a significant 6-9% increase and achieving a 2.5-12x inference-time speedup. For the binder design task, A-CODE rivals and even outperforms existing two-stage models for the first time, achieving a tenfold improvement in success rate on hard tasks. Additionally, for the first time, we present quantitative results for ncAA modeling, in which A-CODE demonstrates the ability to generate structurally plausible ncAA. In this way, our fully atomic design paves the way for a new paradigm to inspire future exploration of a unified all-atom design framework.

We summarize the main features of our A-CODE compared to existing protein binder design models in Table 1. We highlight our contributions in the following aspects:

- We present A-CODE, a fully atomic, one-stage, unified protein co-design framework that combines discrete diffusion for atom types with continuous diffusion for atom coordinates.
- *In silico* evaluations on the unconditional protein generation demonstrate A-CODE's superior generation quality, with a significant 6-9% increase in designability and codesignability and a 2.5-12x inference-time speedup compared to the second-best La-Proteina model.
- *In silico* evaluations on the binder design task further validate A-CODE's performance, rivaling and even outperforming existing state-of-the-art two-stage models. Compared to existing one-stage co-design models, a tenfold improvement in the success rates on hard tasks is observed.
- By predicting atom types without any hard constraint, our framework naturally extends to ncAA modeling, which existing models fail due to their hard-coded constraints.

2 Related Work

Recent advances in the generative framework, such as diffusion (Ho et al., 2020; Song et al., 2020; Karras et al., 2022) and flow matching (Lipman et al., 2022; Chen & Lipman, 2023), have revolutionized generative modeling in all domains, including protein co-design, where each amino acid's position and type are learned. Noticeably, there are also existing works on extending such generative frameworks for the discrete modality (Austin et al., 2021; Cheng et al., 2024; Sahoo et al., 2024). While the AlphaFold 2 (Jumper et al., 2021) model still relies on multiple rounds of structural recycling, the all-atom AlphaFold 3 (Abramson et al., 2024) already adopts the diffusion framework. Many protein co-design models are built on the expressive power of AF3-like models, which we detail below. Additional works are discussed in Appendix B.4.

As briefly discussed in the previous text, we categorize co-design models into *one-stage* and *two-stage* based on their approach to discrete sequence modality. The two-stage representatives include PXDesign (Team et al., 2025) and ODesign (Zhang et al., 2025), which adopt a backbone-only design mode, whereas RFDiffusion-3 (Butcher et al., 2025) and BoltzGen (Stark et al., 2025) have all-atom design capability. These three models are built on AF3-like structures, leveraging its strong structure prediction performance, while other models develop their own architectures. La-Proteina (Geffner et al., 2025) and PLAID (Lu et al., 2024) leverage latent flow matching and diffusion for determining the residue type and the all-atom structures based on the latents. Pallatom (Qu et al., 2024) relies on an alternative sequence decoder when maintaining the all-atom information. While existing all-atom design models predominantly rely on the 14-atom representation, APM (Chu et al., 2024) employs a side-chain structure module to model torsion angles.

In contrast, the one-stage co-design models remain relatively underexplored. MultiFlow (Campbell et al., 2024) and ProteinGenerator (Lisanza et al., 2025) combine continuous flow matching for the backbone atom positions with discrete flow matching for amino acid types and update both modalities simultaneously during flow sampling. Protpardelle (Chu et al., 2024; Lu et al., 2025) relies on an exotic 73-atom presentation which combines all sidechain atoms from the 20 standard amino acids and selectively updates a subset according to the sequence predictor in each denoising step. We also note that, in both two-stage and one-stage approaches, exploration of ncAA modeling remains extremely limited, often treating them as ligands and discarding residual-level information.

3 A-CODE: Fully Atomic Co-Design

A protein can be described by its 3D structure information (atom coordinates) $X = \{\mathbf{x}_i\}_{i=1}^{n_{\text{atom}}}, \mathbf{x}_i \in \mathbb{R}^3$ and amino acid sequence information (residue types) $S = \{c_i\}_{i=1}^{n_{\text{res}}}, c_i \in \{\texttt{ALA}, \texttt{ARG}, \ldots, \texttt{VAL}\}$. Each of the 20 canonical amino acids is associated with a unique list of atom types (names), which together build the discrete atom type information $A = \{a_i\}_{i=1}^{n_{\text{atom}}}$ that can be identified from S or vice versa (See Appendix A, Table 5). Recent structure-prediction models, such as AF3, adopt a generative perspective that relies on diffusion to refine their atomic coordinates, thereby modeling $p(X|S)$. As a folding model, AF3's condition trunk first processes the input amino acid sequence S together with other optional features, such as MSA and templates, to produce a conditioning representation C for the diffusion module, which then takes in the noisy atom coordinates $\mathbf{x}$ and the residue-level representations $\mathbf{s}$ to predict clean coordinates $\hat{X}_0 \leftarrow \text{DenoiseNet}(X_t, S, \sigma_t, C)$, where σ_t is the noise schedule. The outline of the AF3 framework is visualized in Figure 1. During sampling, the coordinates are updated according to the EDM scheduler (Karras et al., 2022) using the clean data prediction, with the noise scale decreasing exponentially to produce gradually refined coordinates.

The protein design task models of the joint distribution over structure and sequence modalities, $p(X, S)$. Existing work, such as PXDesign (Team et al., 2025), predominantly adopts a two-stage approach and decomposes the joint distribution as $p(S|X)p(X)$. In this way, these approaches first utilize a structure-design model to generate X from $p(X)$, and then leverage an inverse-folding model, such as ProteinMPNN (PMPNN) (Dauparas et al., 2022), to generate the sequence from $p(S|X)$ given the structure. Unlike PXDesign, our A-CODE framework directly models the joint distribution $p(X, S)$, effectively capturing the all-atom structure and atom-type modalities simultaneously. We highlight the difference between A-CODE and AF3 and AF3-derived design models in Figure 1, where the former uniquely refines the atom types together with atom coordinates during each iterative step. We now explain the design details of our unified atomic diffusion module.

3.1 UNIFIED ATOMIC DIFFUSION

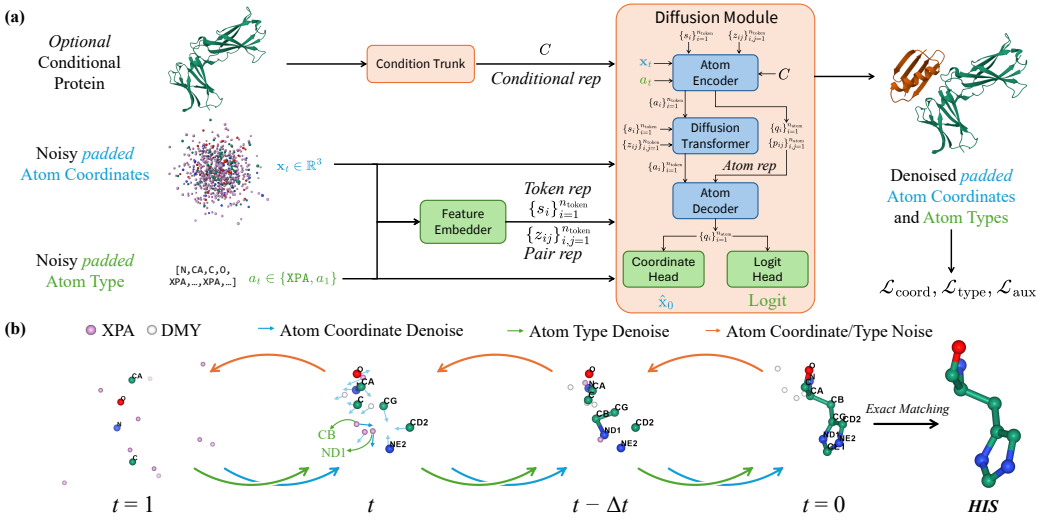


Figure 2: Overview of the architecture and sampling process of our fully atomic A-CODE. **(a)** Model architecture where additional atom-type information is injected and predicted. **(b)** One-stage noising and sampling where both modalities are coupled via the padded *atom14* representation.

The core challenge of fully atomic protein co-design lies in jointly modeling structure and biochemical identity across heterogeneous, variable atomic compositions. To address this, A-CODE adopts the *atom14* representation used in AF2, where each (canonical) amino acid is represented with at most 14 heavy atom coordinates. Unlike prior structure-only design models where *dummy atoms* act purely as coordinate placeholders, in A-CODE they are explicitly represented by a dedicated atom type (`DMY`) and are treated on equal footing with other atom types during diffusion. As a result, dummy atoms participate directly in the generative process and can also be predicted. This design allows residue identities to emerge naturally from atom-level predictions, thereby tightly coupling atomic structure generation and sequence inference within a single generative process.

Within this formulation, A-CODE predicts atom names rather than coarse element or residue types. Atom names encode both chemical and positional semantics, which are essential for unambiguous residue identification. The four backbone atoms are always present and therefore fixed during training and sampling, while the remaining side-chain positions are treated as designable and initialized with masked tokens (`XPA0`–`XPA9`). Importantly, residue-level supervision for designable positions is entirely removed to prevent information leakage; residue types are inferred only after diffusion by matching the predicted atom-name sets. Together with the standard diffusion on the atom coordinates, the model operates purely in atom space, without directly optimizing or predicting residue labels. This atom-centric formulation removes hard constraints on allowable residue types and naturally supports extensions beyond the 20 canonical amino acids, including non-canonical variants, without requiring architectural modification.

To facilitate information flow between the continuous and discrete modalities, A-CODE's architecture adopts a shared transformer-based denoiser network built upon the noisy X_t and A_t, with the final lightweight coordinate and type prediction heads for denoising as $[\hat{X}_0, \text{Logits}] \leftarrow \text{DenoiseNet}(X_t, A_t, \sigma_t, t, C)$, where C is the conditional information in the binder design task. In this way, the model effectively learns the subtle yet crucial interdependence between the coordinate and type modalities, enabling simultaneous denoising during the iterative sampling. Figure 2(a) provides an overview of the model architecture.

3.1.1 TRAINING PIPELINE

The training pipeline of A-CODE utilizes a combination of continuous and discrete diffusion losses for coordinate-type co-design. Let X_0 denote the extended clean coordinates where the coordinates of the dummy atoms are padded. The noisy coordinates are obtained via $X_t := X_0 + \sigma_t \varepsilon$, where $\varepsilon \sim \mathcal{N}(0, 1)$

is the standard Gaussian noise and σ_t are defined according to the EDM scheduler (Karras et al., 2022) used in the original AF3 model. For such noise structures, the original structure losses in the AF3 model are applied, which include the coordinate diffusion loss, a smooth LDDT loss, and additional auxiliary losses described in Appendix B.2. Following PXDesign, the structures of the conditional information are not fixed during training; instead, they are provided to the model as additional information within a unified learning framework. In this way, all structural losses also apply to the non-designable conditional structures. In practice, as with PXDesign, we observed that such information is learned quickly in the early stages of model training, with little error.

Let A_0 denote the extended clean atom types where the special `DMY` tokens are padded. The noisy atom types are obtained via the standard mask diffusion, where all the `XPA` tokens serve as the mask. Following Sahoo et al. (2024), each designable atom type (i.e., not backbone and not condition in binder design) has a probability of κ_t of being turned into the corresponding `XPA` token, where we choose a square scheduler $\kappa_t = 1 - (1-t)^2$ following Gat et al. (2024). The standard mask diffusion loss is then applied to designable atoms as

$$\mathcal{L}_{\text{type}} = \mathbb{E}_t \left[-\frac{\dot{\kappa}_t}{1-\kappa_t} \text{CE}(A_0, \phi_\theta(A_t, t)) \right], \tag{1}$$

where ϕ_θ is the additional atom-level logit head that outputs atom type logits. We weigh the cross-entropy loss for the dummy atoms at different positions according to their natural frequencies to balance training.

3.1.2 Sampling Pipeline and Improvement

As a unified co-design model, A-CODE simultaneously generates discrete atom types and continuous atom coordinates in a single diffusion process, in contrast to two-stage approaches that decouple structure and sequence generation. For the coordinate modality, we adopt the same EDM solver as used in AF3. For the discrete modality, we follow the reparameterized discrete diffusion model (RDM) (Zheng et al., 2023), which allows previously unmasked tokens to be remasked and has been shown to improve sampling robustness. Specifically, based on the scheduler-determined mask proportion, the least-confident designable atom types are remasked. We further adopt a conservative decoding strategy in which a token is remasked if its current prediction is inconsistent with the previous step (see Appendix B.3 for details). Figure 2(b) demonstrates one-stage iterative training and sampling steps where the continuous and discrete modalities are simultaneously noised or denoised.

A critical challenge in fully atomic co-design is that prematurely committing side-chain conformations can introduce steric clashes, which in turn hinder the formation of a correct global fold. To mitigate this issue, we introduce a **side-chain denoising lag** strategy. Specifically, backbone atoms follow the standard noise schedule $\sigma_{\text{bb}}(t)$, while side-chain atoms are assigned a lagged schedule $\sigma_{\text{sc}}(t) = \eta \cdot \sigma_{\text{bb}}(t)$ with $\eta > 1.0$ (we use $\eta = 1.5$), keeping side-chain atoms in a higher-noise regime during early denoising steps and biasing the model toward stabilizing the backbone topology before refining side-chain geometry. We find that applying this denoising lag at inference time alone is sufficient to improve unconditional monomer generation. For conditional binder design, we further apply the same lag during training, which improves optimization stability and downstream design performance. Unless otherwise stated, side-chain denoising lag is enabled during inference for all tasks, and additionally during training for binder design.

To ensure numerical stability when mixed-noise structures are passed to the denoiser, we apply a variance correction to the input coordinates:

$$\hat{X}_{\text{input}} = X \cdot \sqrt{\sigma_{\text{bb}}^2 + \sigma_{\text{data}}^2} \Big/ \sqrt{\sigma_{\text{atom}}^2 + \sigma_{\text{data}}^2}, \tag{2}$$

where σ_{atom} denotes the noise level assigned to a given atom (either σ_{bb} or σ_{sc}). This normalization ensures that the effective input variance remains consistent with the training distribution, while deferring side-chain commitment until a reliable backbone scaffold has emerged.

Throughout sampling, no residue-level information is used or predicted. Residue identities are inferred only after all atom-level predictions have been fully denoised, by exact matching of the predicted atom types after removing dummy atoms. Although this procedure is sensitive to atom-type errors and assigns `UNK` when matching fails, we empirically observe a very low unknown residue rate of approximately 0.05%. The full sampling procedure is summarized in Algorithm 1, with additional

Algorithm 1 Sampling From A-CODE

Calculate shareable conditional features C.
Initialize: Noisy coordinates $X_1 \sim \mathcal{N}(0, \sigma_{\text{max}}^2 \mathbf{I})$ and atom types $A_1 \leftarrow$ masked (XPA).
$\sigma_{\text{bb}}(t) \leftarrow$ EDM schedule from σ_{max} to 0. $\sigma_{\text{sc}}(t) \leftarrow \eta \cdot \sigma_{\text{bb}}(t)$.
for $t \leftarrow 1, 1 - 1/N, \ldots, 1/N$ **do**
  $\sigma_{\text{bb}}, \sigma_{\text{sc}} \leftarrow$ noise levels at current step t.
  $\triangleleft$ **Phase 1: Variance Normalization**
  for each atom i **do**
    $\sigma_i \leftarrow \sigma_{\text{bb}}$ if $i \in$ backbone, else σ_{sc}. $w_i \leftarrow \sqrt{\sigma_{\text{bb}}^2 + \sigma_{\text{data}}^2} / \sqrt{\sigma_i^2 + \sigma_{\text{data}}^2}$.
  $\triangleleft$ **Phase 2: Unified Multimodal Prediction**
  $\tilde{X}_t \leftarrow X_t \odot \mathbf{w}$ {Element-wise scaling}
  $[\hat{X}_0, \text{Logits}] \leftarrow \text{DenoiseNet}(\tilde{X}_t, A_t, \sigma_{\text{bb}}, t, C)$
  $\hat{X}_0 \leftarrow \hat{X}_0 \oslash \mathbf{w}$ {Rescale predicted coordinates back}
  $\triangleleft$ **Phase 3: Coordinate & Type Update**
  $X_{t-1/N} \leftarrow \text{EDM_Step}(X_t, \hat{X}_0, \sigma)$ using per-atom noise levels σ_i.
  $A_{t-1/N} \leftarrow \text{RDM_Step}(\text{Logits}, A_t, t)$.
Strip dummy atoms and infer residue types S_0 from A_0 via exact atom-name matching.
return X_0, A_0, S_0.

details provided in Appendix B.3. Notably, updates to the atom coordinates and atom types are performed in parallel using a single model forward pass at each diffusion step.

3.2 Non-Canonical Amino Acid Modeling

All amino acids beyond the 20 standard proteinogenic ones are referred to as *non-canonical amino acids* (ncAAs). ncAAs are widely present in natural proteins, most commonly arising from post-translational modifications. Their chemical diversity, together with relatively low per-type occurrence frequencies, makes ncAAs difficult to model using residue-level labels. In the PDB database, we identify 890 distinct ncAA types, each corresponding to a unique chemical component dictionary (CCD) code. Despite this sparsity at the type level, ncAAs appear frequently in aggregate, with approximately 46k total occurrences in the PDB, underscoring the importance of modeling ncAAs, particularly for peptide-based therapeutics.

Most existing protein design models effectively ignore ncAAs by restricting generation to backbone atoms, thereby discarding the all-atom information critical for side-chain interactions. In contrast, A-CODE naturally supports ncAA modeling due to its atom-centric formulation, which imposes no hard constraints on residue identities during atom-type decoding. To leverage the strong generative prior learned from large-scale canonical protein data, we finetune a pretrained A-CODE on structures containing ncAAs rather than training a separate model from scratch. This is achieved by constructing an atom-name mapping that assigns each non-canonical atom to a canonical atom name based on chemical semantics, including element type and relative position (see Appendix A for details). Apart from this mapping, all other featurization and training components remain identical, allowing ncAA modeling to be incorporated without architectural modification.

4 Experimental Result

To demonstrate the effectiveness and flexibility of A-CODE in all-atom generative modeling, we conduct extensive experiments spanning unconditional protein generation and conditional binder design. We also introduce a new ncAA benchmark for our new setup, which, to the best of our knowledge, has not been explored in prior work.

Following PXDesign, A-CODE adopts the general AF3 architecture, except that the pairformer stacks are discarded for efficiency, resulting in a total of 169M trainable parameters. We also follow PXDesign's datasets, which include all PDB entries weighted by their pLDDT scores, as well as the clustered and filtered AFDB, a distilled dataset containing AlphaFold-generated structures. The model is trained progressively with two separate training stages. In the first stage, single-chain datasets predominate, whereas multi-chain binder design datasets are upweighted in the second stage.

For fair comparison, we follow PXDesign to use 1000 Euler steps for diffusion sampling. Any unidentified residues will be converted to UNK during postprocessing, and later converted to glycine when fed into ESMFold for designability evaluation.

4.1 Unconditional Protein Generation Benchmark

Table 2: Unconditional protein generation benchmark results. Best results are marked in **bold**.

Model	Codesignability/%↑		Designability/%↑		Sec. Struct./%		Diversity↑	
	All-Atom	CA-Only	PMPNN@1	PMPNN@8	α	β	Str	Seq
P(all-atom) [25]	36.7	37.9	44.4	57.9	56	17	134	148
APM [8]	19.0	32.2	42.8	61.8	73	8	32	64
PLAID [22]	11.0	19.2	23.8	37.6	44	14	25	38
ProteinGenerator [21]	9.8	17.8	42.8	54.2	78	5	12	28
MultiFlow [6]	-	73.2	81.4	94.0	72	12	**209**	247
Protpardelle [12]	8.8	35.2	43.8	56.2	65	14	10	37
La-Proteina [15]	68.4	72.2	82.6	93.8	72	5	206	216
La-Proteina (w/ tri)	75.0	78.2	84.6	94.6	73	6	129	252
A-CODE	**81.2**	**87.0**	**91.6**	**97.0**	50	14	70	252
w/o side-chain lag	72.6	77.6	91.6	97.0	50	13	65	**262**

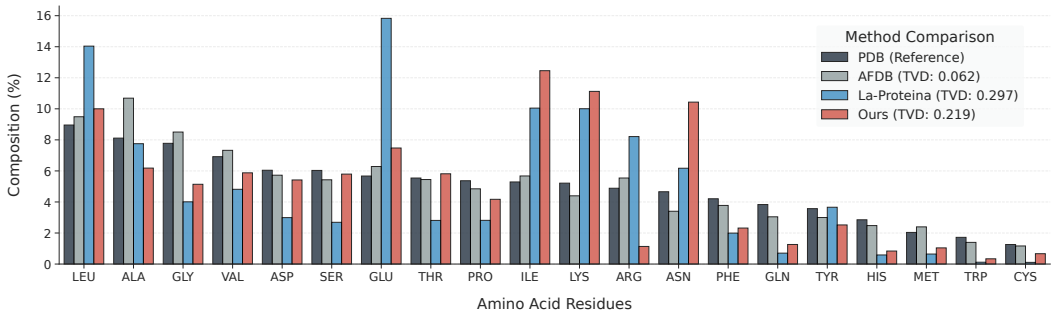


Figure 3: Amino acid type distributions with total variation distance (TVD) to the PDB database.

Table 3: Inference time (seconds) per sample across different lengths on a single NVIDIA A100-80G.

Model	100	200	300	400	500
La-Proteina	1.7	4.1	7.5	12.0	17.6
La-Proteina w/ tri	7.7	28.5	65.7	168.8	182.0
Multiflow	23.5	25.8	30.5	44.9	61.0
A-CODE	2.9	5.2	7.8	10.9	14.6

For unconditional protein design, we follow La-Proteina and draw 100 samples for each length in $\{100, 200, 300, 400, 500\}$. The metrics include: **Codesignability**, defined as the percentage of the samples whose self-consistency RMSD (scRMSD) between the generated structure and the predicted structure from the ESMFold (Lin et al., 2023) on the generated sequence is less than 2.0 Å. The all-atom and Cα-only metrics indicate the atom sets used for the scRMSD calculation. **Designability**, defined as the percentage of the samples whose scRMSD between the generated structure and the predicted structure from the PMPNN-redesigned sequence folded by ESMFold is less than 2.0 Å. For PMPNN@8, eight candidates are generated by the PMPNN, and the best one with the minimal scRMSD is selected, whereas only one sequence is generated for PMPNN@1. **Diversity**, defined as the number of designable clusters in samples, using either the sequence or structure. **Novelty**, defined as the highest TM-align scores against the target database (PDB or AFDB). A more detailed description of these metrics and the evaluation pipeline is available in Appendix C.

The experimental results are summarized in Table 2. A-CODE consistently generates high-quality designs, achieving a significant 6-9% improvement in both designability and codesignability relative to the second-best La-Proteina, which uses the expensive triangular update. Importantly, side-chain lagging plays a crucial role in maintaining the consistency between the continuous and discrete modalities. Compared to consistently high designability, the lower codesignability indicates a mismatch between the generated structures and sequences. This suggests that deferring side-chain

commitment is particularly important for joint structure–sequence generation. Our side-chain lagging effectively mitigates this issue by placing greater emphasis on the crucial backbone positions, thereby allowing the discrete types to be more accurately deciphered.

For inference time, A-CODE demonstrates a 2.5-12x speedup over the second-best, La-Proteina (see Table 3), whereas La-Proteina, with a similar runtime, must discard the triangular update, resulting in even worse performance. Together, A-CODE's superior designability and significant sampling efficiency support the effectiveness of our proposed fully-atomic unified framework.

We further provide an analysis of the distribution of the 20 canonical amino acids in the generations, as demonstrated in Figure 3. A-CODE achieves a lower total variation distance (TVD) from naturally occurring proteins in the PDB database, indicating better fitness to the overall amino acid type distribution, despite a seemingly more difficult target: exact atom-name matching. Interestingly, A-CODE's generated amino acid type distribution captures the low-frequency amino acid (`HIS`, `MET`, `TRP`, `CYS`) much better than La-Proteina, probably due to our position-wise weighting scheme to balance the residue distribution in an implicit way. La-Proteina, on the other hand, relies on latent codes to decode the residue type and thus has an inductive bias towards more frequent amino acids. A more comprehensive analysis of the atomic-level distances, bond angles, and torsion angles is provided in Appendix D, where A-CODE generates physically plausible all-atom structures.

4.2 Conditional Binder Design Benchmark

Table 4: Binder design benchmark on the 10 targets. Best results for each model type are in **bold**.

Model Type	Model	BHRF1	H1	IL17A	IL7RA	IR	PDL1	SC2RBD	TNFa	TrkA	VEGFA
Two-Stage	BoltzGen [29]	14.56	19.21	0.31	12.41	22.34	14.26	0.69	0.88	**28.65**	7.69
	ODesign [37]	42.23	7.29	0.10	7.61	16.82	10.26	3.13	0.00	9.99	3.71
	RFDiffusion-3 [5]	33.38	0.89	0.82	6.47	18.17	16.81	4.34	0.00	14.44	2.95
	PXDesign [30]	**43.90**	12.08	0.82	**29.80**	25.04	**45.33**	11.20	3.43	23.55	**16.72**
	A-CODE (PMPNN)	25.00	**65.59**	**1.79**	4.93	**30.09**	39.96	**28.05**	**6.16**	6.87	1.37
One-Stage	Protpardelle-1c [23]	3.73	0.27	0.00	0.09	0.19	7.04	0.99	0.00	3.52	0.17
	A-CODE (Co-Design)	**22.87**	**55.71**	**1.24**	**4.05**	**41.67**	**28.70**	**37.50**	**7.57**	**3.70**	**0.96**

We benchmark the conditional generation ability of A-CODE on a test set comprising 10 protein targets with diverse structural properties, as proposed in Zambaldi et al. (2024). In binder design, a target protein with known sequence and structure is fed in as the conditions for the generative model. Hotspot information of the binding site is often available as additional input. For a unified evaluation, we follow PXDesign to use the filter from AF2-IG-Easy (Watson et al., 2023). For each different target, we sweep the binder length range with 8 samples per length, following the same protocol in PXDesign, and report the percentage of designable samples as **Designability**. We evaluate both the model-generated co-designed sequence and the PMPNN-redesigned single sequence as separate variants. The designability of A-CODE, together with existing baselines, is provided in Table 4, where two-stage and one-stage (co-design) models are highlighted separately.

Used as a two-stage model, A-CODE achieves the best designability on 5 out of the 10 tasks, outperforming all existing baselines. The one-stage co-design results, on the other hand, also consistently and significantly outperform the Protpardelle baseline across all 10 tasks, with up to a tenfold increase in the success rates. Notably, the performance gap between the co-designed sequence and the 1-pass PMPNN-redesigned sequence has narrowed significantly, and the co-design variant even outperforms the best two-stage model (which is A-CODE itself) on two of the hardest tasks of SC2RBD and TNFa. Enabling the lag consistently also improves success rates across 6 of 10 targets, as demonstrated in Figure 5 in Appendix D.

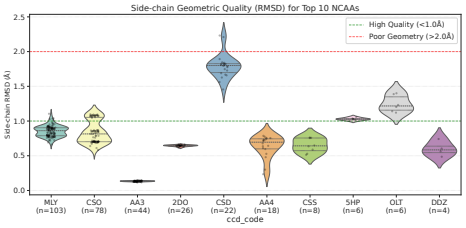
The superior performance of A-CODE demonstrates the necessity and benefit of incorporating atomic-level information to build a better design model: the co-design framework not only significantly pushes the performance boundary within one-stage design models but also benefits two-stage design by enabling more fundamental atomic-level interactions between modalities.

4.3 Extension to ncAA Modeling

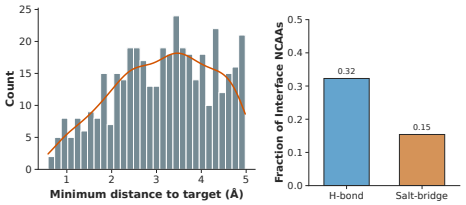
While the two training stages above are effective, with an extremely low frequency of unknown residues at 0.05%, they do not unlock the full power of A-CODE for ncAA modeling. To achieve

this, we first filter the entire PDB database and retain entries with at least one ncAA. This gives a total of 23k single chains and 29k binder pairs where the second chain contains at least one ncAA. With the additional atom name mapping described in Section 3.2, we finetuned the model trained after the two main stages using only these two datasets to emphasize its ncAA modeling ability.

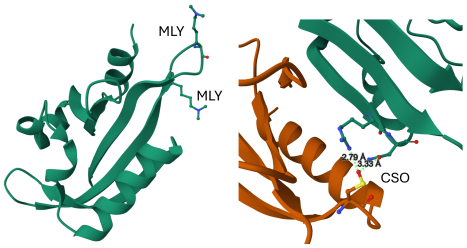
We provide a quantitative statistical analysis of the generated ncAAs in Figure 4. Figure 4a illustrates the side-chain geometric quality for the top 10 generated ncAAs. The majority of generated structures are high-quality, with a side-chain RMSD under 1.0 Å, which demonstrates the robustness of A-CODE in modeling complex chemical topologies. The distribution of the minimum distance from ncAAs to the target protein (Figure 4b) shows a significant concentration between 2 Å and 4 Å, the optimal range for biophysical interactions. Interface analysis in Figure 4c reveals that 32% of generated ncAAs participate in hydrogen bonding and 15% form salt bridges. These statistics suggest that A-CODE effectively captures the physicochemical preferences of ncAAs, leveraging their unique properties to stabilize binder-target complexes.



(a) Side-chain RMSD.



(b) Min distance to target. (c) Interaction ratio.



(d) Generation examples.

Figure 4: ncAA statistics analysis and selected generations containing ncAAs.

We further show representative generation results in Figure 4d. In the left panel, with the unconditional setup, two methyllysines (MLY) are successfully generated around the disordered loop region. This mimics the behavior of naturally occurring methyllysines in histones, in which methylation serves as a signal for gene regulation. Therefore, for signaling purposes, methyllysine residues are predominantly located in solvent-accessible regions or on the surface of protein complexes, as correctly captured by our A-CODE generation. On the right, we present the binder design setup, in which the target protein is specified as the condition. A hydroxycysteine (CSO) is generated at the interface instead of a cysteine, effectively expanding the sidechain length to promote sidechain interactions. Furthermore, with the modified geometry and the greater electronegativity of oxygen, two additional hydrogen bonds form with ARG and ASN residues in the target protein at appropriate distances, thereby reducing binding energy and stabilizing the complex. In this way, we demonstrate the flexibility and feasibility of A-CODE in ncAA modeling, offering a unified approach to all-atom protein generative modeling that can be extended to arbitrary biomolecules.

5 Conclusion & Limitation

In this work, we propose A-CODE, a fully atomic, one-stage protein co-design model with a unified multimodal diffusion framework for both discrete atom types and continuous atom coordinates. Our fully atomic design not only facilitates the synergistic generation of the two modalities but also enables natural extension to non-canonical amino acids. A-CODE demonstrate superior generation quality, achieving new SOTA designability and codesignability on the unconditional benchmark and significantly better one-stage binder design performance across all ten tasks.

Despite its current superior performance over the baselines, we note limitations in A-CODE. Most crucially, although the performance gap between the two-stage (with ProteinMPNN) and one-stage co-design models has been shrunk with our new state-of-the-art results, it remains non-negligible. Furthermore, our evaluations of the ncAA generations remain preliminary, as no widely acknowledged datasets or benchmarks have been established. In this way, we hope to inspire future design of protein generative models that naturally incorporate ncAAs and even ligands as a unified framework.

LLM USAGE STATEMENT

LLMs are *not* used in any part of this paper, including devising the idea, writing the code, running the experiments, or writing or polishing the paper.

ETHICS STATEMENT

We present A-CODE for the fully atomic generation of protein structures and sequences within a unified multimodal diffusion framework. This advancement has broad implications for computational biology and drug discovery. By enabling the seamless incorporation of non-canonical amino acids (ncAAs) and eliminating the error propagation found in multi-stage pipelines, our model facilitates the design of more complex and functionally diverse protein therapeutics.

Our high-fidelity design tool lowers the barrier to entry for exploring novel protein modalities. While this fosters innovation in medicine, it also necessitates a consideration of safety. The generalizability of our framework to arbitrary atomic compositions implies that it could potentially be applied to biological systems with unknown or hazardous properties. We advocate for the responsible release of model weights and are dedicated to ensuring the responsible use of our model for the positive benefit of society.

REPRODUCIBILITY STATEMENT

We have provided the data processing pipeline in Appendix A, model hyperparameters in Appendix B.1, training hyperparameters in Appendix B.2, training and sampling pseudocode in Algorithm 1, 4, 5, and 6, and the evaluation protocol in Appendix C. We therefore believe our results can be reasonably reproduced despite the lack of codebase due to limitations.

REFERENCES

Josh Abramson, Jonas Adler, Jack Dunger, Richard Evans, Tim Green, Alexander Pritzel, Olaf Ronneberger, Lindsay Willmore, Andrew J Ballard, Joshua Bambrick, Sebastian Bodenstein, David A Evans, Chia-Chun Hung, Michael O'Neill, David Reiman, Kathryn Tunyasuvunakool, Zachary Wu, Akvile Zemgulyte, Eirini Arvaniti, Charles Beattie, Ottavia Bertolli, Alex Bridgland, Alexey Cherepanov, Miles Congreve, Alexander Imani Cowen-Rivers, Andrew Cowie, Michael Figurnov, Fabian B Fuchs, Hannah Gladman, Rishub Jain, Yousuf A. Khan, Caroline M R Low, Kuba Perlin, Anna Potapenko, Pascal Savy, Sukhdeep Singh, Adrian Stecula, Ashok Thillaisundaram, Catherine Tong, Sergei Yakneen, Ellen D. Zhong, Michal Zielinski, Augustin Žídek, Victor Bapst, Pushmeet Kohli, Max Jaderberg, Demis Hassabis, and John M. Jumper. Accurate structure prediction of biomolecular interactions with alphafold 3. *Nature*, 630(8016):493–500, 2024.

Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. Structured denoising diffusion models in discrete state-spaces. *Advances in neural information processing systems*, 34:17981–17993, 2021.

M. Beals, L. Gross, and S. Harrell. Amino Acid Frequency. `https://legacy.nimbios.org/~gross/bioed/webmodules/aminoacid.htm`, 1999.

Avishek Joey Bose, Tara Akhound-Sadegh, Guillaume Huguet, Kilian Fatras, Jarrid Rector-Brooks, Cheng-Hao Liu, Andrei Cristian Nica, Maksym Korablyov, Michael Bronstein, and Alexander Tong. Se (3)-stochastic flow matching for protein backbone generation. *arXiv preprint arXiv:2310.02391*, 2023.

Jasper Butcher, Rohith Krishna, Raktim Mitra, Rafael I Brent, Yanjing Li, Nathaniel Corley, Paul T. Kim, Jonathan Funk, Simon V. Mathis, Saman Salike, Aiko Muraishi, Helen E. Eisenach, Tuscan Rock Thompson, Jie Chen, Yuliya Politanska, Enisha Sehgal, Brian Coventry, Odin Zhang, Bo Qiang, Kieran Didi, Max Kazman, Frank DiMaio, and David Baker. De novo design of all-atom biomolecular interactions with rfdiffusion3. *bioRxiv*, 2025.

Andrew Campbell, Jason Yim, Regina Barzilay, Tom Rainforth, and Tommi Jaakkola. Generative flows on discrete state-spaces: Enabling multimodal flows with applications to protein co-design. *arXiv preprint arXiv:2402.04997*, 2024.

Ricky TQ Chen and Yaron Lipman. Flow matching on general geometries. *arXiv preprint arXiv:2302.03660*, 2023.

Ruizhe Chen, Dongyu Xue, Xiangxin Zhou, Zaixiang Zheng, Xiangxiang Zeng, and Quanquan Gu. An all-atom generative model for designing protein complexes. *arXiv preprint arXiv:2504.13075*, 2025a.

Vincent B Chen, W Bryan Arendall, Jeffrey J Headd, Daniel A Keedy, Robert M Immormino, Gary J Kapral, Laura W Murray, Jane S Richardson, and David C Richardson. Molprobity: all-atom structure validation for macromolecular crystallography. *Biological crystallography*, 66(1):12–21, 2010.

Xinshi Chen, Yuxuan Zhang, Chan Lu, Wenzhi Ma, Jiaqi Guan, Chengyue Gong, Jincai Yang, Hanyu Zhang, Ke-Xin Zhang, Shenghao Wu, Kuang-Qi Zhou, Yan-Ping Yang, Zhenyu Liu, Lan Wang, Bo Shi, Shaochen Shi, and Wenzhi Xiao. Protenix-advancing structure prediction through a comprehensive alphafold3 reproduction. *BioRxiv*, pp. 2025–01, 2025b.

Chaoran Cheng, Jiahan Li, Jian Peng, and Ge Liu. Categorical flow matching on statistical manifolds. *Advances in Neural Information Processing Systems*, 37:54787–54819, 2024.

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 V. Anishchenko, Nathaniel R. Bennett, Hua Bai, Robert J. Ragotte, Lukas F. Milles, Basile I. M. Wicky, Alexis Courbet, Robbert J. de Haas, Neville P. Bethel, Philip J. Y. Leung, Timothy F. Huddy, Samuel J. Pellock, Doug K Tischer, Frank Chan, Brian Koepnick, H. Nguyen, Alex Kang, Banumathi Sankaran, Asim K. Bera, Neil P. King, and David Baker. Robust deep learning–based protein sequence design using proteinmpnn. *Science*, 378(6615): 49–56, 2022.

Itai Gat, Tal Remez, Neta Shaul, Felix Kreuk, Ricky TQ Chen, Gabriel Synnaeve, Yossi Adi, and Yaron Lipman. Discrete flow matching. *Advances in Neural Information Processing Systems*, 37: 133345–133385, 2024.

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. *arXiv preprint arXiv:2507.09466*, 2025.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A. A. Kohl, Andrew J. Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Michalina Pacholska, Tamas Berghammer, Sebastian Bodenstein, David Silver, Oriol Vinyals, Andrew W. Senior, Koray Kavukcuoglu, Pushmeet Kohli, and Demis Hassabis. Highly accurate protein structure prediction with alphafold. *nature*, 596(7873):583–589, 2021.

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.

Zeming Lin, Halil Akin, Roshan Rao, Brian L. Hie, Zhong-Kai Zhu, Wen-Ting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, Allan dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Salvatore Candido, and Alexander Rives. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Sidney Lyayuga Lisanza, Jake Merle Gershon, Sam Tipps, Jeremiah Nelson Sims, Lucas Arnoldt, Samuel Hendel, Miriam K. Simma, Ge Liu, Muna Yase, Hongwei Wu, Claire D Tharp, Xin-Ting Li, Alex Kang, Evans Brackenbrough, Asim K. Bera, Stacey R. Gerben, Bruce J. Wittmann, Andrew C. McShan, and David Baker. Multistate and functional protein design using rosettafold sequence space diffusion. *Nature biotechnology*, 43(8):1288–1298, 2025.

Amy X Lu, Wilson Yan, Sarah A Robinson, Kevin K Yang, Vladimir Gligorijevic, Kyunghyun Cho, Richard Bonneau, Pieter Abbeel, and Nathan Frey. Generating all-atom protein structure from sequence-only training data. *bioRxiv*, pp. 2024–12, 2024.

Tianyu Lu, Richard Shuai, Petr Kouba, Zhaoyang Li, Yilin Chen, Akio Shirali, Jinho Kim, and Po-Ssu Huang. Conditional protein structure generation with protpardelle-1c. *bioRxiv*, 2025.

Jiezhong Qiu, Junde Xu, Jie Hu, Hanqun Cao, Liya Hou, Zijun Gao, Xinyi Zhou, Anni Li, Xiujuan Li, Bin Cui, Fei Yang, Shuang Peng, Ning Sun, Fangyu Wang, Ai-Min Pan, Jie Tang, Jie-Ping Ye, Junyang Lin, Jin Tang, Xing-Yong Huang, Pheng Ann Heng, and Guang-Yong Chen. Instructplm: Aligning protein language models to follow protein structure instructions. *bioRxiv*, pp. 2024–04, 2024.

Wei Qu, Jiawei Guan, Rui Ma, Ke Zhai, Weikun Wu, and Haobo Wang. P (all-atom) is unlocking new path for protein design. *bioRxiv*, pp. 2024–08, 2024.

Michael G Rossmann and Eddy Arnold. *International tables for crystallography volume F: crystallography of biological macromolecules*. Wiley Online Library, 2001.

Subham Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. *Advances in Neural Information Processing Systems*, 37:130136–130184, 2024.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Hannes Stark, Felix Faltings, MinGyu Choi, Yuxin Xie, Eunsu Hur, Timothy O'Donnell, Anton Bushuiev, Talip Uçar, Saro Passaro, Weian Mao, Mateo Reveiz, Roman Bushuiev, Tomáš Pluskal, Josef Sivic, Karsten Kreis, Arash Vahdat, Shamayeeta Ray, Jonathan T. Goldstein, Andrew Savinov, Jacob A. Hambalek, Anshika Gupta, Diego A. Taquiri-Diaz, Yaotian Zhang, A. Katherine Hatstat, Angelika Arada, Nam Hyeong Kim, Ethel Tackie-Yarboi, Dylan Boselli, Lee Schnaider, Chang C. Liu, Gene-Wei Li, Denes Hnisz, David M. Sabatini, William F. DeGrado, Jeremy Wohlwend, Gabriele Corso, Regina Barzilay, and Tommi Jaakkola. Boltzgen: Toward universal binder design. *bioRxiv*, pp. 2025–11, 2025.

Protenix Team, Milong Ren, Jinyuan Sun, Jiaqi Guan, Cong Liu, Chengyue Gong, Yuzhe Wang, Lan Wang, Qixu Cai, Wenzhi Ma, Yuxuan Zhang, Zhenyu Liu, Hanyu Zhang, Xinshi Chen, and Wenzhi Xiao. Pxdesign: Fast, modular, and accurate de novo design of protein binders. *bioRxiv*, pp. 2025–08, 2025.

Michel Van Kempen, Stephanie S Kim, Charlotte Tumescheit, Milot Mirdita, Jeongjae Lee, Cameron LM Gilchrist, Johannes Söding, and Martin Steinegger. Fast and accurate protein structure search with foldseek. *Nature biotechnology*, 42(2):243–246, 2024.

Xinyou Wang, Zaixiang Zheng, Fei Ye, Dongyu Xue, Shujian Huang, and Quanquan Gu. Diffusion language models are versatile protein learners. *arXiv preprint arXiv:2402.18567*, 2024a.

Xinyou Wang, Zaixiang Zheng, Fei Ye, Dongyu Xue, Shujian Huang, and Quanquan Gu. Dplm-2: A multimodal diffusion protein language model. *arXiv preprint arXiv:2410.13782*, 2024b.

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, Basile I. M. Wicky, Nikita Hanikel, Samuel J. Pellock, Alexis Courbet, William Sheffler, Jue Wang, Preetham Venkatesh, Isaac Sappington, Susana Vázquez Torres, Anna Lauko, Valentin De Bortoli, Emile Mathieu, Sergey Ovchinnikov, Regina Barzilay, T. Jaakkola, Frank Dimaio, Minkyung Baek, and David Baker. De novo design of protein structure and function with rfdiffusion. *Nature*, 620(7976): 1089–1100, 2023.

Jason Yim, Andrew Campbell, Andrew Y. K. Foong, Michael Gastegger, Jos'e Jim'enez-Luna, Sarah Lewis, Victor Garcia Satorras, Bastiaan S. Veeling, Regina Barzilay, T. Jaakkola, and Frank Noé. Fast protein backbone generation with se (3) flow matching. *arXiv preprint arXiv:2310.05297*, 2023.

Vinícius Flores Zambaldi, David La, Alexander E. Chu, Harshnira Patani, Amy E Danson, Tristan O. C. Kwan, Thomas Frerix, Rosalia G. Schneider, David Saxton, Ashok Thillaisundaram, Zachary Wu, Isabel Moraes, Oskar Lange, Eliseo Papa, Gabrielle Stanton, Victor Martin, Sukhdeep Singh, Lai H. Wong, Russ Bates, Simon A A Kohl, Josh Abramson, Andrew W. Senior, Yilmaz Alguel, Mary Y. Wu, Irene M. Aspalter, Katie Bentley, David L.V. Bauer, Peter Cherepanov, Demis Hassabis, Pushmeet Kohli, Rob Fergus, and Jue Wang. De novo design of high-affinity protein binders with alphaproteo. *arXiv preprint arXiv:2409.08022*, 2024.

Odin Zhang, Xujun Zhang, Haitao Lin, Cheng Tan, Qinghan Wang, Yuanle Mo, Qian-Tai Feng, Gang Du, Yuntao Yu, Zichang Jin, Ziyi You, Peicong Lin, Yijie Zhang, Yuyang Tao, Shi-Cheng Chen, Jack Xiaoyu Chen, Chenqing Hua, Wei-Bo Zhao, Run-Ze Ma, Yu-Qin Xia, Kejun Ying, Jun Li, Yu-Yu Zeng, Lijun Lang, Peichen Pan, Hanqun Cao, Zihao Song, Bo Qiang, Jiaqi Wang, Peng-Fei Ji, Lei Bai, Jian Zhang, Chang-Yu Hsieh, Pheng Ann Heng, Si-Qi Sun, Ting-Jun Hou, and Shuangjia Zheng. Odesign: A world model for biomolecular interaction design. *arXiv preprint arXiv:2510.22304*, 2025.

Lin Zheng, Jianbo Yuan, Lei Yu, and Lingpeng Kong. A reparameterized discrete diffusion model for text generation. *arXiv preprint arXiv:2302.05737*, 2023.

Supplementary Material

A Dataset Pipeline

In this section, we describe the details of our data pipeline for adapting PXDesign to an all-atom co-design model. Specifically, we discuss our procedure for the atom type (name) modality and our ncAA pipeline.

A.1 Canonical AA Pipeline

Table 5: Padded atom names for the 20 canonical amino acids with their nature occurring frequencies taken from Beals et al. (1999).

Amino Acid	Atom Names	Freq/%
ALA (A)	N ,CA ,C ,O ,CB ,DMY,DMY,DMY,DMY,DMY,DMY,DMY,DMY,DMY	7.4
ARG (R)	N ,CA ,C ,O ,CB ,CG ,CD ,NE ,CZ ,NH1,NH2,DMY,DMY,DMY	4.2
ASN (N)	N ,CA ,C ,O ,CB ,CG ,OD1,ND2,DMY,DMY,DMY,DMY,DMY,DMY	4.4
ASP (D)	N ,CA ,C ,O ,CB ,CG ,OD1,OD2,DMY,DMY,DMY,DMY,DMY,DMY	5.9
CYS (C)	N ,CA ,C ,O ,CB ,SG ,DMY,DMY,DMY,DMY,DMY,DMY,DMY,DMY	3.3
GLN (Q)	N ,CA ,C ,O ,CB ,CG ,CD ,OE1,NE2,DMY,DMY,DMY,DMY,DMY	3.7
GLU (E)	N ,CA ,C ,O ,CB ,CG ,CD ,OE1,OE2,DMY,DMY,DMY,DMY,DMY	5.8
GLY (G)	N ,CA ,C ,O ,DMY,DMY,DMY,DMY,DMY,DMY,DMY,DMY,DMY,DMY	7.4
HIS (H)	N ,CA ,C ,O ,CB ,CG ,ND1,CD2,CE1,NE2,DMY,DMY,DMY,DMY	2.9
ILE (I)	N ,CA ,C ,O ,CB ,CG1,CG2,CD1,DMY,DMY,DMY,DMY,DMY,DMY	3.8
LEU (L)	N ,CA ,C ,O ,CB ,CG ,CD1,CD2,DMY,DMY,DMY,DMY,DMY,DMY	7.6
LYS (K)	N ,CA ,C ,O ,CB ,CG ,CD ,CE ,NZ ,DMY,DMY,DMY,DMY,DMY	7.2
MET (M)	N ,CA ,C ,O ,CB ,CG ,SD ,CE ,DMY,DMY,DMY,DMY,DMY,DMY	1.8
PHE (F)	N ,CA ,C ,O ,CB ,CG ,CD1,CD2,CE1,CE2,CZ ,DMY,DMY,DMY	4.0
PRO (P)	N ,CA ,C ,O ,CB ,CG ,CD ,DMY,DMY,DMY,DMY,DMY,DMY,DMY	5.0
SER (S)	N ,CA ,C ,O ,CB ,OG ,DMY,DMY,DMY,DMY,DMY,DMY,DMY,DMY	8.1
THR (T)	N ,CA ,C ,O ,CB ,OG1,CG2,DMY,DMY,DMY,DMY,DMY,DMY,DMY	6.2
TRP (W)	N ,CA ,C ,O ,CB ,CG ,CD1,CD2,NE1,CE2,CE3,CZ2,CZ3,CH2	1.2
TYR (Y)	N ,CA ,C ,O ,CB ,CG ,CD1,CD2,CE1,CE2,CZ ,OH ,DMY,DMY	3.3
VAL (V)	N ,CA ,C ,O ,CB ,CG1,CG2,DMY,DMY,DMY,DMY,DMY,DMY,DMY	6.8

In a standard PDB file, each of the 20 canonical amino acids has a unique and canonical ordered list of atom names. Such a list always starts with the 4 backbone atoms `N,CA,C,O` followed by the other side-chain atoms (if any). We note that identification based on element type and position to distinguish among amino acids may be ambiguous. For example, both LEU and ILE are identified by the sequence `N,C,C,O,C,C,C,C` and both PRO and VAL are identified by `N,C,C,O,C,C,C`. The atom names list is then padded with the special token `DMY` to make an atom14 representation for each amino acid as the target discrete modality. See Table 5 for a complete list of the padded names for all 20 canonical amino acids. For all amino acids, the input atom type is always fixed to be

$$A_0 := \texttt{N,C,CA,O,XPA0,XPA1,XPA2,XPA3,XPA4,XPA5,XPA6,XPA7,XPA8,XPA9}, \tag{3}$$

where the 10 special `XPA` tokens serve as the mask token in discrete diffusion with additional positional embedding. The four backbone atoms are always fixed in both the forward noising process and the reverse sampling process to facilitate training. In this way, together with the `UNK` token for unknown atoms, we obtain a total number of 40 atom type classes, as shown in Table 6.

During training, the dummy atoms are randomly sampled from a Gaussian distribution with a unit variance centered at the CA atom. In this way, we can provide an additional supervising signal for the model to learn the approximate position of the amino acid while offering robustness to potential noise compared to a deterministic target. During sampling, the dummy atoms are not deleted until the final postprocessing (matching) stage, ensuring a consistent model input. After the diffusion sampling stage finalizes, the postprocessing stage identifies the residue type based on the final atom types predictions in the procedure described in Algorithm 2. It can be seen that the exact-matching scheme requires both the generation's positions and atom types to be identical for a successful match.

Table 6: Atom names used in A-CODE. A total of 40 types are used.

Atom Category	Atom Names	Count
Backbone Atom	N,CA,C,O	4
Sidechain Atom	CB,CG,CG1,CG2,CD,CD1,CD2,CE,CE1,CE2,CE3,CZ,CZ2,CZ3,CH2, ND1,ND2,NE,NE1,NE2,NZ,NH1,NH2, OG,OG1,OD1,OD2,OE1,OE2,OH,OXT,SG,SD	33
Special Atom	DMY (dummy atom in the target) XPA (design atom in the input) UNK (unknown atom)	3

Algorithm 2 Residue Type Matching

1: **Input:** Predicted atom types A_0.
2: **for** Each residue **do**
3: Strip any dummy atom DMY.
4: Concatenate the remaining atom names as the unique identifier.
5: Matching the identifier for the 20 canonical amino acids.
6: If some amino acid matches, the residue type is assigned; else, assign UNK.
7: If more than one atom shares the same atom name (which only happens in UNK or ncAAs after atom name mapping), modify the repetitive names by appending or increasing the last numerical count in the name. This ensures that the output PDB file is legal.
8: **return:** S_0.

The featurization generally follows the original AF3 framework, with some modifications on the designable residues/atoms to prevent data leakage. Specifically, restype for designable residues is assigned a special XPB token. Any MSA or template information is disabled, with features set to zero as inputs. Additionally, the atom-level CCD reference information for the designable residues is replaced with that for glycine, which provides only backbone information. For conditional generation in binder design, additional ground-truth coordinates, residue types, and reference information are still available in a separate conditional track, enabling the model to recover the target protein structure. For binder pairs, a hotspot mask will be randomly sampled as additional information to enable hotspot-conditioned binder design during inference.

In the first two stages of the training, we follow standard practice from previous work by first ignoring any ncAA and substituting them with glycine. That is, we only extract the positions and atom names of the four backbone atoms N,CA,C,O and discard any sidechain information as if it were a glycine. In this way, we aim to minimize the occurrence of UNK at these stages, leaving the ncAA modeling to the third finetuning stage.

The datasets used in the first stage follow the original setup in PXDesign, where we have a combination of the experimentally determined PDB, split into single-chain and binder-pair splits, as well as the distillation dataset of AlphaFold-predicted structures in AlphaFold Database (AFDB), which contains only single-chain structures. The single-chain PDB split contains 433,318 data points, clustered into 36,955 unique clusters. The PDB complex split contains 266,369 data pairs, clustered into 26,394 unique clusters. The AFDB split, after filtering and clustering, contains 8,403,792 single-chain data points.

A.2 ncAA Pipeline

For non-canonical amino acids, to enable finetuning without expanding the current atom type vocabulary, we first propose a semantic atom name mapping that assigns a *chemically* closest atom name to the atom names in ncAAs. As the atom names in the PDB file often follow a naming convention, where the first character indicates the element type, the second character indicates the relative position (A, B, G, D, etc.), and the last character indicates the numbering, we may provide more semantically important chemical information in addition to pure element types. The detailed atom name mapping scheme is described in Algorithm 3. We note that the unknown atoms only occur when an element other than C, N, O, or S is present, such as P in phosphorylation.

Algorithm 3 ncAA Atom Name Mapping

1: **Input:** Atom name.
2: If the name matches a standard one, return the name.
3: Else, if the element type and positions match, return the first available name with the same first two characters.
4: Else, if the element type matches, return the last available position of that element.
5: Else, return the unknown atom type.
6: **return:** Mapping canonical atom name.

We also ensure the four backbone atoms are always the first four atoms in the ncAA, where the other sidechain atoms follow the original ordering in the PDB. For compatibility with the atom14 representation, any ncAA with more than 14 heavy atoms will be truncated to keep only the first 14. Other parts of the ncAA pipeline are the same as the standard pipeline, demonstrating the flexibility and transferability of our A-CODE framework.

In principle, it is always possible to incorporate the ncAA atom names in addition to the 40 standard AA atom name vocabulary, but this will lead to unnecessary expansion of the vocabulary with extremely low-frequency classes. Nonetheless, we do note two limitations that can be potentially improved by training a new model: a) Although the unknown atom name is uncommon after mapping, they do exist for atom types other than C, N, O, or S. Adding some additional element types to the vocabulary will mitigate this issue. b) Although the atom14 representation suffices for the 20 standard AA and most ncAAs, there are ncAAs with more than 14 atoms. We truncate those ncAAs with more than 14 heavy atoms. Using larger atom representations will mitigate this issue at the cost of higher computational cost. We will leave more comprehensive explorations for future work.

The ncAA dataset is constructed by filtering the whole PDB entries and keeping only the entries with at least one ncAA, with the exceptions that a) the already existing `UNK` residues are regarded as alanine and not ncAA, and b) any residues without the four backbone atoms are not regarded as ncAA (e.g., the amino group `NH2`, the acetyl group `ACE`, or the methylamino group `NME`). Note that the D-amino acids (or other enantiomers/diastereomers) are still kept as ncAAs. Other non-protein components, such as DNA, RNA, or ligand, are filtered away. Following the PXDesign dataset splitting scheme, we also construct separate splits for the single-chain and binder ncAA datasets, where for the latter, the second chain is fixed as the binder (designable) and should contain at least one ncAA. After filtering, we obtain 22632 single-chain data and 29252 binder pairs. The whole dataset contains 890 distinct ncAA types, out of which 263 have more than 14 heavy atoms, and the largest ncAA contains 61 heavy atoms, where a coenzyme A is attached.

B Training Pipeline

In this section, we provide additional details for training A-CODE in the three stages.

B.1 Model Specification

Following PXDesign, we do not use the full AF3 architecture and discard all pairformer layers to improve efficiency, resulting in a small model with 169M trainable parameters for efficient generative modeling. The conditional trunk and the feature embedder generally follow the original AF3 design, in which additional node features, such as position embeddings and chain IDs, are fed into the module along with noisy coordinates and types. The conditional trunk offers additional structure information as soft constraints. For unconditional generation, these features will be generated from placeholders of `XPB` residue types (noted as X in Figure 1 and Figure 2) to prevent information leakage. The embedder generations token representations $\{s_i\}_{i=1}^{n_{\text{token}}}$ and token-wise pair representations $\{z_{ij}\}_{i,j=1}^{n_{\text{token}}}$ via reading out over atom features. These features, without requiring expensive pairformer stacks for refinement, are directly fed into the diffusion module.

The diffusion module consists of an atom encoder, a diffusion transformer, and an atom decoder, followed by the final prediction heads for denoising. The atom encoder takes the token-level single and pair representations, the conditional features, and the atomic-level noisy coordinates and types as inputs. The token-level representations are directly fed into the diffusion transformer for

refinement, whereas the atom-level single and pair representations $\{q_i\}_{i=1}^{n_{\text{atom}}}, \{p_{ij}\}_{i,j=1}^{n_{\text{atom}}}$ are fed in a skip-connection fashion directly to the atom decoder. In this way, the overall computational cost will not increase dramatically when adopting the atom14 representation, which has more atoms but the same number of tokens. The output of the atom decoder is the atom-level representation, upon which the denoised coordinates $\hat{\mathbf{x}}_0$ and the atom type logits are predicted following separate heads.

We generally follow the hyperparameter selection used in PXDesign, except that the query sizes in the atom encoder and decoder are increased to 128 to accommodate more atoms. We refer interested readers to the original AF3 for the details on each module and PXDesign for detailed hyperparameters.

B.2 Training Specification

During training, the noisy interpolation data are generated using a combination of continuous and discrete diffusion. Specifically, for the continuous atom coordinates, we follow AF3 to use the EDM scheduler (Karras et al., 2022) as:

$$\mathbf{x}_t = \text{random_aug}(\mathbf{x}_0) + \sigma_t, \tag{4}$$

where random_aug denotes the random rotation augmentation and $\sigma_t \sim \sigma_{\text{data}} \cdot \exp(-1.2 + 1.5\mathcal{N}(0,1))$ with $\sigma_{\text{data}} = 16$. For the structure losses, we generally follow the original ones in AF3. The major diffusion loss is a combination of the weighted MSE loss and the smooth local distance difference test (LDDT) loss with a cutoff distance of 15 Å:

$$\mathcal{L}_{\text{coord}} = \frac{\sigma_t^2 + \sigma_{\text{data}}^2}{(\sigma_t + \sigma_{\text{data}})^2}\mathcal{L}_{\text{MSE}} + \mathcal{L}_{\text{slddt}}, \tag{5}$$

where the MSE is calculated between the denoiser prediction $\hat{X}_0$ and the ground truth aligned to the prediction X_0^{align}. The smooth LDDT loss is defined as a smooth distance difference between the ground truth and the prediction structures as:

$$\mathcal{L}_{\text{slddt}} = 1 - \frac{1}{|D|}\sum_{(i,j)\in D} F(|\hat{d}_{ij} - d_{ij}|), \tag{6}$$

where $d_{ij} = \|\mathbf{x}_i^{\text{GT}} - \mathbf{x}_j^{\text{GT}}\|$ denotes the distance between the ground truth X_0 and $\hat{d}_{ij} = \|\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_j\|$ for the prediction $\hat{X}_0$. The summation is over the set $D = \{(i,j) | i \neq j, d_{ij} < 15\text{Å}, 1 \leq i,j \leq n_{\text{atom}}\}$ of atom pairs (i,j), and the soft kernel is defined as

$$F(\delta) = \frac{1}{4}\left[\sigma\left(\frac{1}{2} - \delta\right) + \sigma\left(1 - \delta\right) + \sigma\left(2 - \delta\right) + \sigma\left(4 - \delta\right)\right]. \tag{7}$$

Additionally, auxiliary losses for predicting denoised structure distogram bins, LDDT (pLDDT), aligned error (pAE), and distance error (pDE) are also leveraged for all atoms (including conditions) following the original setup in AF3. We refer interested readers to the original AF3 paper (Abramson et al., 2024) for more details.

For the discrete atom type, we adopt the standard mask diffusion model (MDM), where the noisy discrete atom types are obtained as:

$$a_t = \begin{cases} a_0, & \text{w.p. } \kappa(t), \\ \texttt{XPA}, & \text{w.p. } 1 - \kappa(t), \end{cases} \tag{8}$$

where we follow Gat et al. (2024) to use the square scheduler $\kappa(t) = 1-(1-t)^2$ (diffusion convention is used such that $t = 0$ is the clean data). For better consistency between the two modalities, we use a coupled scheduler where $t = F^{-1}(\sigma_t)$ is inferred from the inverse CDF F^{-1} of σ_t.

The discrete diffusion loss is demonstrated in Eq. 1 and is calculated on designable atoms (designable residues and not backbone). The discrete diffusion loss is weighted by the natural occurrence frequencies of the 20 canonical amino acids to prevent data imbalance between the dummy and non-dummy targets. We follow the statistics in Table 5 to adjust the positive weight (non-dummy) as if there is an equal number of dummy and non-dummy atoms at each position. As an example, the `XPA0` position is 92.6% non-dummy, whereas the `XPA8` and `XPA9` positions only have 1.3%

Algorithm 4 Training Step for A-CODE

1: Calculate conditional features C.
2: Sample $\sigma_t \sim \sigma_{\text{data}} \cdot \exp(-1.2 + 1.5\mathcal{N}(0,1)), t = F^{-1}(\sigma_t)$.
3: $X_t \leftarrow \text{Noise_EDM}(X_1, \sigma_t)$.
4: $A_t \leftarrow \text{Noise_MDM}(X_1, \kappa(t))$.
5: $[\hat{X}_0, \text{Logits}] \leftarrow \text{DenoiseNet}(X_t, A_t, \sigma_t, t, C)$
6: Calculate the continuous diffusion loss $\mathcal{L}_{\text{coord}}(\hat{X}_0, X_0, \sigma_t)$, the discrete mask diffusion loss $\mathcal{L}_{\text{type}}(\text{Logits}, A_0, \kappa(t))$ and other auxiliary losses.
7: Optimize the model parameters.

non-dummy, so the dummy cross-entropy losses on these positions are weighted by 12.51 and 0.013, respectively.

The datasets we used for training A-CODE in the first two stages are the same, except that the binder split was upweighted in the second stage. In the first stage, the dataset weights are 1.4, 0.004, and 3.25 for the PDB single-chain, PDB complex, and AFDB, respectively; in the second stage, they are 0.4, 0.2, and 0.8. In the first two stages, ncAAs were discarded, and the ncAA datasets were not used. In the third stage, we used only the two ncAA datasets for finetuning to highlight the ncAA modeling ability of A-CODE, with weights of 1.0 and 1.0. The model was trained with a batch size of 64, with 200k iterations in the first stage and 400k in the second. The model was then finetuned for another 200k iterations in the third stage. At all stages, we used a cropping size of 640 residues, employing a mixture of spatial, sequential, and interface cropping approaches as described in the AF3 paper. We used a local batch size of 1 and a diffusion batch size of 8. A constant learning rate of 5×10^{-4} is applied for all training stages. All training was carried out on NVIDIA H20 GPUs.

B.3 Sampling Specification

The sampling component of the structure generally follows the original AF3 sampling algorithm, adapted from the EDM's second-order Heun solver. The inference-time noise schedule is defined as

$$\sigma_t = \sigma_{\text{data}} \cdot \left(s_{\text{max}}^{1/p} + t(s_{\text{min}}^{1/p} - s_{\text{max}}^{1/p}) \right)^p, \tag{9}$$

where $s_{\text{max}} = 160, s_{\text{max}} = 4 \times 10^{-4}, p = 7$ and t is uniformly distributed between $[0, 1]$ with 1000 sampling steps. The detailed sampling algorithm is summarized in Alg. 5, where each step advances the current noisy coordinates with a step size of $1/1000$ and a step scale $\eta = 2.5$. As noted in Alg. 1, the noise schedule σ_t here can be either σ_{bb} or σ_{sc} depending on the atom types.

Algorithm 5 Diffusion Sampling for Coordinates (EDM_Step)

Require: Current time t, step size Δt, current coordinates $\mathbf{x}_t$, denoised coordinates $\hat{\mathbf{x}}$, schedules σ_t, σ_s with $s = t - \Delta t$, noise scale $\lambda = 1.003$.
1: $\gamma \leftarrow 0.8$ if $\sigma_t > 1$ else $0, \hat{t} \leftarrow (\gamma + 1)\sigma_s$
2: $\tilde{\mathbf{x}}_t \leftarrow \mathbf{x}_t + \lambda\sqrt{\hat{t}^2 - \sigma_s^2} \cdot \varepsilon, \varepsilon \sim \mathcal{N}(0, I)$
3: $\mathbf{x}_s \leftarrow \tilde{\mathbf{x}}_t + \eta(\sigma_t - \hat{t})(\mathbf{x}_t - \hat{\mathbf{x}})/\hat{t}$

For the discrete diffusion sampling of atom types, instead of using the standard mask diffusion sampling where modification of an already unmasked token is not allowed, we follow RDM (Zheng et al., 2023) to allow for a more flexible control of the sampling dynamics, which is described in Algorithm 6. As in training, we use the same square schedule for the discrete modality. Additionally, we employ the *conservative* decoding scheme in RDM, where a token must be sufficiently confident to be unmasked, and less-confident predictions are remasked to allow modification. This is carried out iteratively by recording the previous logits as the confidence score and iteratively unmasking and remasking. An exception would be the last step, where we will force all the remaining masks to be unmasked. In practice, about 15%-20% of the tokens are unmasked during the last step, with the others already determined during the iterative process. In this way, our sampling scheme is fundamentally different from existing all-atom two-stage models, in which atom types are inferred solely from the final denoised structure after a single forward pass.

Algorithm 6 Discrete Diffusion Sampling for Types (RDM_Step)

Require: Current time t, step size Δt, previous selected logits s_{prev} and token a_{prev} at t, current selected logits s_{cur} and token a_{cur} at $t - \Delta t$, schedule $\kappa(\cdot)$.

1: If $t \leq \Delta t$, unmask all the remaining masked positions with current tokens and return.
2: Sort the previous selected logits s_{prev}, mask $1 - \kappa(t - \Delta t)$ proportion of the previous tokens with the lowest logits (low-confidence prediction).
3: For the current unmasked positions in the previous token, remask all the positions where $a_{\text{prev}} \neq a_{\text{cur}}$ (inconsistent prediction) or $s_{\text{prev}} > s_{\text{cur}}$ (less confident prediction).
4: For the current masked positions in the previous token, unmask all tokens with the current tokens where $a_{\text{prev}} = a_{\text{cur}}$ and $s_{\text{prev}} \leq s_{\text{cur}}$.
5: Set $s_{\text{prev}} \leftarrow s_{\text{cur}}, a_{\text{prev}} \leftarrow a_{\text{cur}}$ for the next iteration.

B.4 Additional Relative Work

We also include several notable unimodal protein design models as references. In principle, any structure-only design model can be cascaded with inverse folding models such as ProteinMPNN (Dauparas et al., 2022) to form a co-design model, and any sequence-only design model can be cascaded with folding models like AF2/AF3 as a co-design model. For example, FrameFlow (Yim et al., 2023) and FoldFlow (Bose et al., 2023) leverage Riemannian flow matching to learn the position and orientation of each amino acid's backbone atoms. Proteina (Geffner et al., 2025), the predecessor of La-Proteina, is a CA-only structure design model. Other protein language models, such as InstructPLM (Qiu et al., 2024) and DPLM (Wang et al., 2024a;b), can also be adapted for protein co-design. Within sequence design models, methods like InstructPLM (Qiu et al., 2024) extend the power of large language models to protein design tasks by framing them as natural language tasks. On the other hand, models like DPLM (Wang et al., 2024a;b) rely purely on the amino acid sequences and structure tokens. These models are generally posed to address different unimodal design tasks. Therefore, we do not include them in our experiments.

C Evaluation Pipeline

In this section, we discuss the evaluation pipeline and metrics.

C.1 Uncondtional Protein Generation

The unconditional protein generation task benchmarks the general structure-sequence co-design ability of a generative model, highlighting the need for more complex practical tasks like binder design. We generally follow the setup in PXDesign to generate 128 sequences for each length in $\{100, 200, 300, 400, 500\}$ and then average across all generated samples to compute the metric. For co-design models in Table 2, we generally either copy the originally reported metrics or follow the recommended inference setup with the official checkpoints.

For comprehensive evaluation, we categorize the metrics we used into *likelihood* and *diversity* metrics. The likelihood metrics measure the quality of the generation, determining whether they resemble a *natural* protein structure, sequence, or both. Such likelihood metrics include:

Designability Designability describes whether a structure is *foldable*, i.e., if there exists some amino acid sequence that folds into such a structure. In practice, ProteinMPNN (Dauparas et al., 2022) (PMPNN) is used as a common inverse-folding model to design amino acid sequences, which are then fed into ESMFold (Lin et al., 2023) for folding prediction. In this way, designability is defined as the average proportion of the generated structures whose smallest scRMSD to the folded structures of the ProteinMPNN redesigned sequence is less than 2.0 Å. **PMPNN@1** and **PMPNN@8** denote the different numbers of candidate amino acid sequences generated by ProteinMPNN. Designability does not rely on the sequence modality generated by the model.

Codesignability Codesignability describes whether a sequence-structure pair is feasible, offering another aspect of likelihood that also depends on the sequence modality and its coordination with the

structure modality. Utilizing ESMFold on the generated sequence, codesignability is defined as the proportion of the generated structure-sequence pairs whose scRMSD between the generated structure and the folded structure of the generated sequence is less than 2.0 Å. **All-Atom** and **CA-Only** denote the different alignment sets on either the all-atom structure or the CA atom only. The latter is generally easier. Due to the limitations of ESMFold, any generated unknown residue in this stage will be substituted with a glycine.

The diversity metrics measure the sample diversity of a generative model, preventing the model from overfitting only on high-likelihood data or memorizing the dataset, which includes:

Diversity Diversity counts the number of distinct clusters in the generated samples, and a higher diversity score indicates a broader span of the generated samples. The clustering is obtained by running the `foldseek` (Van Kempen et al., 2024) program, and the **Str/Seq** subtypes are configurable clustering types that consider either the structure or sequence modality during clustering.

C.2 Binder Design

The setup for the binder design follows the PXDesign paper (Team et al., 2025), which uses the 10 protein targets proposed in Zambaldi et al. (2024), covering a wide range of structural diversity and exhibiting real-world biological significance (see Table 7). Designability is the major metric used here, where we follow PXDesign to rely on the *in silico* filter from AF2-IG-Easy (Watson et al., 2023), where a designable binder is defined as: (a) interface predicted absolute error (ipAE) less than 10.85 Å, (b) interface predicted TM score (ipTM) greater than 0.5, (c) predicted local distance difference test (pLDDT) greater than 80%, and d) binder bound/unbound RMSD less than 3.5 Å. AF2 (Jumper et al., 2021) is used to evaluate these metrics, in which residues in the binder chain are assigned a large offset to mimic multichain co-folding. Across different binder lengths for the same target, the success counts are summed to obtain the final result.

Table 7: Protein binder design targets from AlphaProteo (Zambaldi et al., 2024).

Target	PDB	Target crop	Hotspot	Binder length
BHRF1	2WH6	A2-158	A65, A74, A77, A82, A85, A93	80-120
H1	5VLI	A5-325, B501-670	B521, B545, B552	40-120
IL7RA	3DI3	B17-209	B58, B80, B139	50-120
IL17A	4HSA	A17-29, A41-131, B19-127	A94, A116, B67	50-140
Insulin	4ZXB	E6-155	E64, E88, E96	40-120
PD-L1	5O45	A17-132	A56, A115, A123	50-120
SC2RBD	6M0J	E333-526	E485, E489, E494, E500, E505	80-120
TNF-α	1TNF	A12-157, B12-157, C12-157	A113, C73	50-120
TrkA	1WWW	X282-382	X294, X296, X333	50-120
VEGFA	1BJ1	V14-107, W14-107	W81, W83, W91	50-140

C.3 ncAA Analysis

The ncAA generation protocol follows the unconditional setup with 500 generations in total. ncAAs are identified from `UNK` residues in the binder chain and matched to PDB Chemical Component Dictionary (CCD) entries using heavy-atom topology matching via graph isomorphism. The graph topology is inferred from the generated structures using RDKit[1], and only those whose graph topology and element types exactly match a CCD entry will be successfully identified. Side-chain geometric quality was assessed by computing a side-chain RMSD to the corresponding CCD reference, excluding backbone atoms (N, CA, C, O) and taking the minimum RMSD over all symmetry-consistent atom mappings. Interface ncAAs were defined based on the minimum heavy-atom distance to the target protein (5 Å as a threshold), and interaction patterns were characterized using geometry-based proximity proxies, including hydrogen-bond-like (N/O/S atoms within 3.5 Å) and salt-bridge-like (N-O atom pairs within 4.0 Å) contacts.

[1] `https://www.rdkit.org/`

D ADDITIONAL RESULTS

In this section, we provide additional experimental results to support a more comprehensive analysis of A-CODE's generative performance.

D.1 ADDITIONAL RESULTS FOR UNCONDITIONAL PROTEIN GENERATION

Table 8: Atomic structural evaluation metrics of MAD and clashscore for A-CODE and La-Proteina.

Metric (MAD)	N-CA	CA-C	C-N	C-O	N-CA-C	CA-C-N	C-N-CA	sc_bond_length	sc_bond_angle	clashscore
Ours	0.0096	0.009	0.0185	0.0071	1.1618	1.521	1.6651	0.0188	1.8623	**53.67**
LA-Proteina (notri)	0.0062	0.0056	0.0184	0.0079	1.1781	1.3632	1.3565	0.0234	1.7959	70.06
LA-Proteina (tri)	0.0061	0.0061	0.0187	0.008	1.1577	1.9424	1.8639	0.0233	1.7862	57.50

To comprehensively evaluate the physical realism of the generated all-atom structures and explicitly validate the effectiveness of our fully atomic co-design approach, we conducted a rigorous quantitative analysis of atom-level physical metrics. The evaluations are aggregated across all generation lengths in the unconditional setup, conducted on structures generated by our method and compared against the baseline (La-Proteina) under identical settings. We focused on three primary aspects:

- **Steric Clashes (Clashscore)**. We calculated the steric clashscore (Chen et al., 2010), defined as the number of severe spatial overlaps per 1,000 heavy atoms. A steric clash is identified when the Euclidean distance between two non-bonded heavy atoms is less than the sum of their respective van der Waals (vdW) radii minus a tolerance margin of 0.4 Å. To strictly exclude covalently bonded neighbors, we parsed the molecular topology and only considered atom pairs separated by more than three bonds (i.e., shortest path > 3 in the chemical graph).
- **Backbone Geometry Deviations**. We measured the structural distortion of the protein backbone by computing the Mean Absolute Deviation (MAD) between the generated geometries and the standard ideal values (e.g., Engh & Huber parameters (Rossmann & Arnold, 2001)). Specifically, we analyzed the bond lengths for four primary backbone covalent bonds (N-Cα, Cα-C, C-N peptide bond, and C-O) and the bond angles for three primary backbone angles (N-Cα-C, Cα-C-N, and C-N-Cα).
- **Side-Chain Geometry Deviations**. Crucially, since our model generates full-atom coordinates including side chains, we explicitly evaluated the side-chain geometric quality. We utilized the standard amino acid templates from the PDB Chemical Component Dictionary (CCD) to construct ideal reference topologies. By mapping the generated side-chain heavy atoms to these standard templates, we systematically calculated the MAD for 93 distinct side-chain covalent bonds (e.g., CB-CG, CG-OD) and 134 distinct side-chain bond angles across all 20 standard amino acids.

The results are summarized in Table 8. The quantitative results demonstrate that our method achieves a notably lower steric clashscore compared to the strong baseline, indicating a more harmonious 3D spatial arrangement of atoms. Furthermore, our generated structures maintain highly precise local geometries, with both backbone and side-chain bond-length deviations strictly controlled to ~ 0.02 Å and bond-angle deviations to $\sim 2.0°$.

Additionally, we plot the side-chain torsion angle distributions in the generated samples for each canonical amino acid type in Figure 6a and 6b, together with those obtained from the PDB and AFDB databases and the La-Proteina baseline as a reference. Our A-CODE can produce structures that more closely resemble their natural occurrence, reflecting its better generative capability in capturing atomic conformations. These minimal deviations fall well within the natural range of physical thermal fluctuations, strongly confirming the physical validity and high atomic fidelity of our fully atomic generation model.

To provide a more accurate understanding of generational diversity, we present additional analysis of the clusters obtained by A-CODE and La-Proteina. In Table 9, we show that each designable structure cluster produced by our method is supported by substantially richer sequence diversity, with 4.19 sequence clusters per structure cluster on average (vs. 1.57 for La-Proteina) and much lower within-cluster pairwise sequence identity (0.410 vs. 0.872). This suggests that our method explores deeper sequence basins within the designable structural families it discovers. Indeed, these results support the conclusion that the observed behavior reflects a genuine designability–diversity

trade-off rather than a trivial artifact, as La-Proteina's structural diversity came with a pronounced drop in designability. As future work, we will investigate whether broader structural coverage can be achieved more effectively through data-level interventions, such as training-data construction and sampling strategies, beyond sampling-time tuning alone.

Table 9: Analyses on the clusters generated by A-CODE and La-Proteina.

Method	Avg. Seq Clusters / Str Cluster	Within-Cluster Pairwise Seq ID	Top-5 Largest Str Clusters: Avg. Seq Clusters
Ours	**4.19**	**0.410**	**18.6**
La-Proteina	1.57	0.872	15.0

D.2 Additional Results for Binder Design

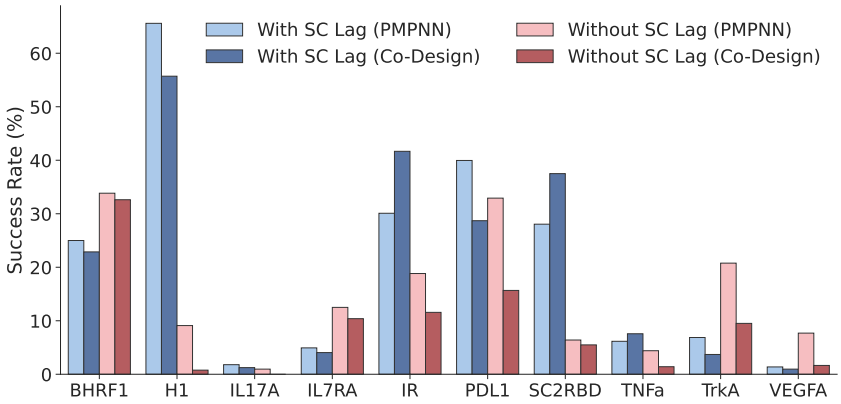
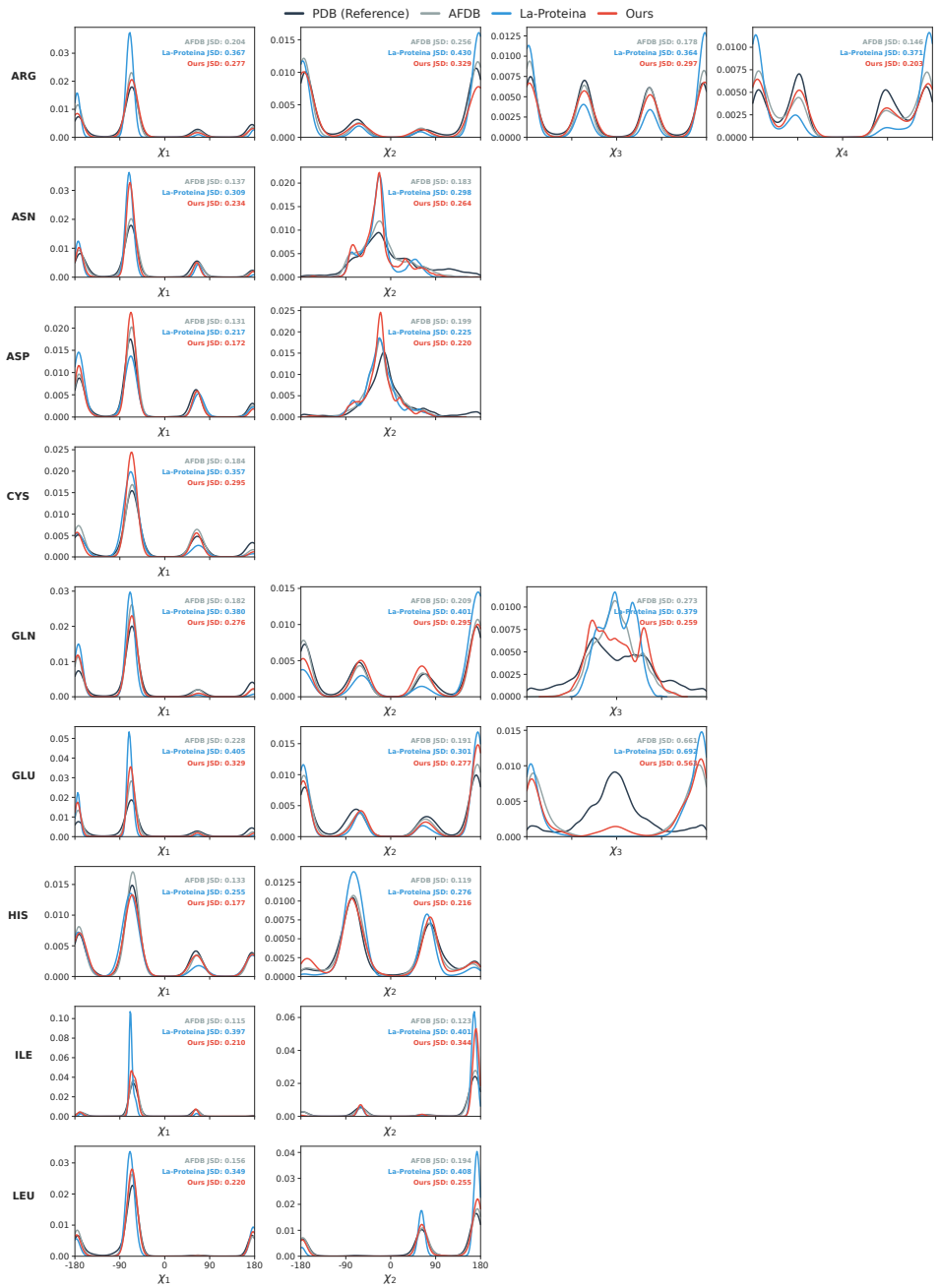
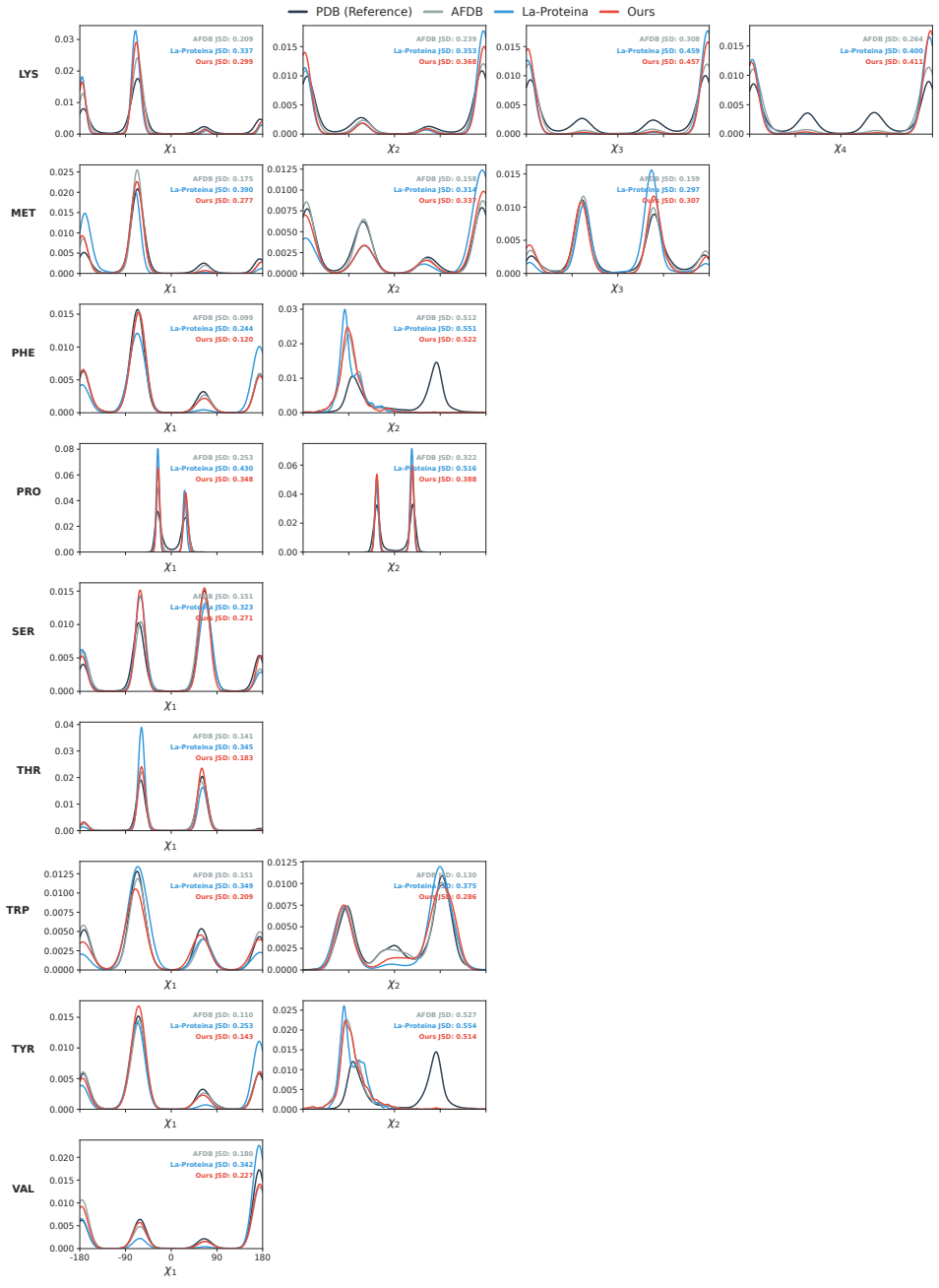


Figure 5: Ablation study of side-chain lag training on binder design

We benchmark the impact of side-chain lagging on the binder design generations in Figure 5 across all ten targets. It can be seen that enabling the lag improves the success rates across 6 of 10 targets. Similar to the unconditional setup, we observe larger gains for co-designed sequences than for ProteinMPNN-redesigned ones. This, together with the results from the unconditional benchmark, supports the effectiveness of our proposed side-chain lagging technique during sampling, mitigating the early commitment of side-chain positions that might otherwise lead to inconsistency between the continuous and discrete modalities.



(a) Distributions for residues ARG through HIS.



(b) Distributions for residues ILE through VAL.

Figure 6: Statistical distribution of side-chain dihedral angles.