CODESIGN: CONSISTENCY FROM DATA TO TRAJECTORY IN ALL-ATOM PROTEIN BINDER CO-DESIGN

Anonymous authors
Paper under double-blind review

ABSTRACT

The central challenge in *de novo* protein design is generating plausible, mutually compatible structures and sequences, such that each designed sequence folds into its intended structure and the structure accommodates that sequence. Compared to typical two-stage design methods, which decouple the modeling of the interdependent modalities, co-design models improve the cross-modal consistency by jointly generating sequences and structures. However, naively generating sequences and structures simultaneously does not ensure their consistency. To address this challenge, we propose CODesign framework. We improve data consistency by generating approximately 105,000 consistency-distilled dimers. We further promote consistency through a multimodal joint flow model that captures the joint distribution of sequences, backbone structures, and local atomic configurations, together with a consistency-aware joint resampling strategy that iteratively refines sequences and side chains. Experiments show that CODesign achieves state-of-the-art performance with the highest in silico success rates on both protein- and ligand-target binder design. Ablation studies also demonstrate our distilled dataset increases performance by 70.9%, and can be further improved by our proposed resampling mechanism with negligible additional computational cost. Code, model weights and the new dataset will be completely open-source.

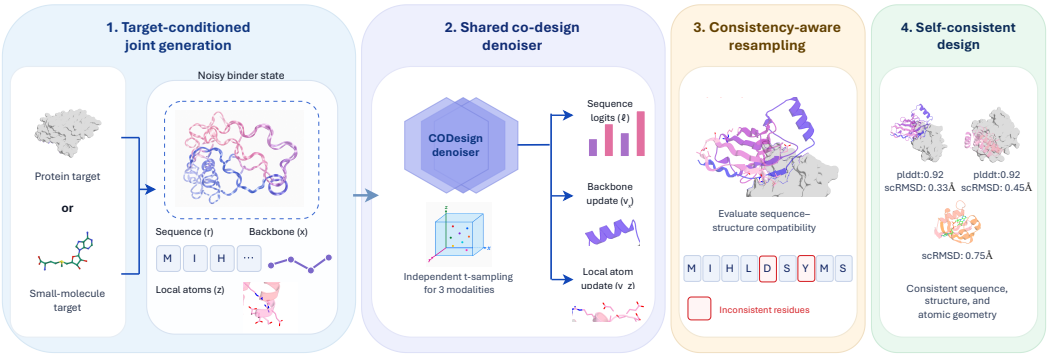


Figure 1: CODesign jointly generates sequence, backbone, and local atomic states conditioned on either a protein or small-molecule target. A shared co-design denoiser is trained with independent modality-time sampling and predicts sequence logits, backbone updates, and local-atom updates. During generation, we apply consistency-aware joint resampling for residue and side-chain refining. This overall framework generates protein binders with consistent all-atom structures.

1 INTRODUCTION

De novo protein design offers a promising avenue for developing a wide range of therapeutics (Watson et al., 2023). Protein design models can be formulated as probabilistic models of protein sequences–structure conditioned on a given target. To capture biophysical interactions under geometric constraints, prior methods typically adopt a two-stage approach, sequentially generating protein backbones and inverse-folding them into candidate sequences (Dauparas et al., 2022). Al-

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

though two-stage methods have achieved strong results in monomer and enzyme design, their low pass rates on challenging targets make them inefficient (Watson et al., 2023; Ahern et al., 2026).

Recent co-design methods jointly generate discrete sequences and structures, which could model residue-specific interactions (Butcher et al., 2025). Despite their increased expressiveness, they still underperform two-stage methods or generate a biased residue-type distribution. Latent diffusion offers an alternative by jointly generating continuous sequence embeddings and backbone coordinates (Geffner et al., 2026; Didi et al., 2026), but does not explicitly model discrete residue identities and thus falls outside explicit sequence–structure co-design.

Despite these advances, ensuring sequence–structure consistency remains a central challenge for binder co-design. We highlight two key bottlenecks: *(1) the scarcity of self-consistent training data for protein complexes* and *(2) the difficulty of maintaining sequence–structure compatibility throughout sampling*. Complexes in the Protein Data Bank (PDB) remain limited in number and structural diversity (Didi et al., 2026). Although predicted complexes can expand the training set, many synthetic sequence–structure pairs exhibit poor self-consistency: refolding their sequences does not reliably recover the paired structures. Reidenbach et al. (2025) identified a similar limitation in predicted monomers from the widely used AlphaFold Database (AFDB) (Varadi et al., 2022; Han et al., 2026) Besides, residue identities selected from partially resolved structures may become incompatible with the evolving geometry (Rector-Brooks et al., 2026) during sampling. Existing methods often rely on auxiliary sequence-revision modules or external structure-prediction guidance, providing indirect consistency signals and potentially incurring substantial computational overhead (Rector-Brooks et al., 2026; Cheng et al., 2026; Didi et al., 2026). This motivates integrating compatibility assessment and correction directly into joint sequence–structure sampling.

To address these bottlenecks, we introduce **CODesign**, a consistency-driven multimodal flow framework that extends Proteína-Atomística (Reidenbach et al., 2025) from monomer generation to target-conditioned all-atom protein binder co-design. *For improving training-data consistency,* we construct **104.5k** distilled sequence–structure pairs by jointly redesigning both partners of experimental and predicted complexes with ProteinMPNN (Dauparas et al., 2022; Reidenbach et al., 2025) and co-folding the redesigned sequences with ESMFold2-Fast. Among high-confidence candidates, we select the structure that best recovers the reference complex, emphasizing interface qualities. *For maintaining consistency during sampling,* we propose Consistency-Aware joint Resampling (**CAR**), which exploits CODesign's intrinsic inverse-folding capability enabled by training with decoupled modality times and structural conditioning. CAR reassesses residue choices against the evolving backbone, selectively resampling incompatible identities together with their local atomic states. These corrections also enter self-conditioning, modifying subsequent backbone vector fields so that C_α trajectories can adapt without direct coordinate resets. CAR thus couples sequence revision with all-atom refinement, without an external inverse-folding model or folding-based reward oracle.

We evaluate CODesign on protein- and ligand-target binder design benchmarks (Didi et al., 2026). Experiments demonstrate that our method achieves state-of-the-art performance with the highest in silico unique success rates among evaluated baselines. Ablation studies show that our distilled dataset can increase unique successes by a substantial margin of **70.9%**, which can be further improved by 4% using the proposed resampling algorithm. These results underscore the potential of CODesign as a powerful next-generation tool for developing novel protein binders.

2 Related Work

Protein generative modeling has progressed from backbone generation followed by inverse folding in RFdiffusion (Watson et al., 2023; Dauparas et al., 2022) toward joint sequence–structure modeling. In discrete all-atom design, residue identity determines atomic composition. Atom14/37 provides a fixed-dimensional representation using 14 masked atom slots per residue, enabling side-chain interaction modeling. (Qu et al., 2024; Butcher et al., 2025; Reidenbach et al., 2025). A complementary family of methods avoids explicit residue identities during generation by jointly modeling structures and continuous sequence embeddings (Geffner et al., 2025; 2026; Didi et al., 2026). Building on these paradigms, task-specific models extend co-design to target-conditioned binder design using objectives or guidance tailored to intermolecular interfaces (Cheng et al., 2026; Didi et al., 2026).

3 METHODS

Consistency in target-conditioned co-design. Let $\mathbf{c}^{\text{target}}$ denote the representation of a protein or small-molecule target. We represent a binder by its amino-acid sequence $\mathbf{r}$, C_α coordinates $\mathbf{x}$, and non-C_α coordinates $\mathbf{z}$. Target-conditioned co-design seeks to learn the conditional joint distribution

$$p^\star(\mathbf{r}, \mathbf{x}, \mathbf{z} \mid \mathbf{c}^{\text{target}}). \tag{1}$$

This distribution specifies the desired endpoint but does not impose a particular generation order. In CODesign, the three modalities evolve within a shared sampling trajectory, allowing sequence and structure predictions to influence one another as generation proceeds.

A consistent sample must satisfy two complementary conditions. First, its sequence should be compatible with the target-conditioned backbone geometry. Second, its complete all-atom structure should be recoverable from the sequence in the presence of the target. We characterize these conditional relationships as

$$\mathcal{C}(\mathbf{r}, \mathbf{x}, \mathbf{z}; \mathbf{c}^{\text{target}}) = \underbrace{\log p^\star(\mathbf{r} \mid \mathbf{x}, \mathbf{c}^{\text{target}})}_{\mathcal{C}_{\text{seq}}} + \lambda \underbrace{\log p^\star(\mathbf{x}, \mathbf{z} \mid \mathbf{r}, \mathbf{c}^{\text{target}})}_{\mathcal{C}_{\text{str}}}, \tag{2}$$

where $\mathcal{C}_{\text{seq}}$ measures sequence compatibility using residue-agnostic C_α geometry, $\mathcal{C}_{\text{str}}$ measures all-atom structural recoverability from the sequence, and λ balances the two directions. Low-consistency training pairs provide conflicting supervision for learning the target distribution in Equation 1, while initially compatible modalities may still diverge during the sampling trajectory.

CODesign addresses these two failure modes within a unified framework. At the data level, CODIMER uses inverse folding and co-folding as surrogates for the two conditional terms in Equation 2, producing more consistent training pairs (Section 3.1). At the model level, independently sampled modality times enable the multimodal flow to learn both structure-to-sequence and sequence-to-structure conditional regimes (Section 3.2). At the sampling level, consistency-aware joint resampling uses the learned conditionals to identify incompatible residue choices and jointly update their identities and local atomic states (Section 3.3).

3.1 CODIMER: CONSISTENCY-DISTILLED PROTEIN DIMERS

Limitations of existing complex data. Existing approaches either pretrain on predicted AFDB monomers (Cheng et al., 2026), or construct pseudo-complexes from domain–domain interactions in predicted monomers (Didi et al., 2026). Nonetheless, they do not explicitly ensure that the paired sequence has high probability under $\mathcal{C}_{\text{seq}}$. Predicted structures may be individually plausible while fail to recover the same complex geometry when refolded. This issue is observed in high-confidence AFDB heterodimers (Han et al., 2026): only 8.49% are recovered by folding oracles within a complex-level C_α scRMSD of 2 Å.

Consistency distillation. We construct CODIMER by approximating the two conditional terms in Equation 2 through inverse folding and co-folding. We use ESMFold2-Fast for its computational efficiency and the demonstrated correlation of its predictions with experimental binder success (Claude Science & Shanehsazzadeh, 2026). Given a reference complex structure $\mathbf{x}_0$, ProteinMPNN jointly redesigns both chains to produce $K = 4$ candidate sequence pairs , which ESMFold2-Fast then co-folds into the corresponding all-atom complexes:

$$\mathbf{r}_k \sim q_{\text{MPNN}}(\mathbf{r} \mid \mathbf{x}_0), \hat{\mathbf{x}}_k \sim q_{\text{fold}}(\mathbf{x} \mid \mathbf{r}_k), \qquad k = 1, \ldots, K. \tag{3}$$

The first step promotes sequence compatibility with the reference geometry, while the second evaluates whether the redesigned sequence recovers that geometry. We retain candidates that satisfy

$$\overline{\text{pLDDT}}(\hat{\mathbf{x}}_k) \geq 0.7, \qquad \text{ipAE}(\hat{\mathbf{x}}_k) \leq 10\,\text{Å}. \tag{4}$$

Among the qualifying candidates, we select the one with the lowest complex-level C_α RMSD to the reference structure:

$$k^\star = \arg\min_{k \in \mathcal{A}} d_{C_\alpha}(\hat{\mathbf{x}}_k, \mathbf{x}_0), \tag{5}$$

where $\mathcal{A}$ denotes the set of candidates satisfying both confidence criteria. We apply this procedure to AFDB high-confidence heterodimers (Han et al., 2026), HumanPPI predictions (Zhang et al.,

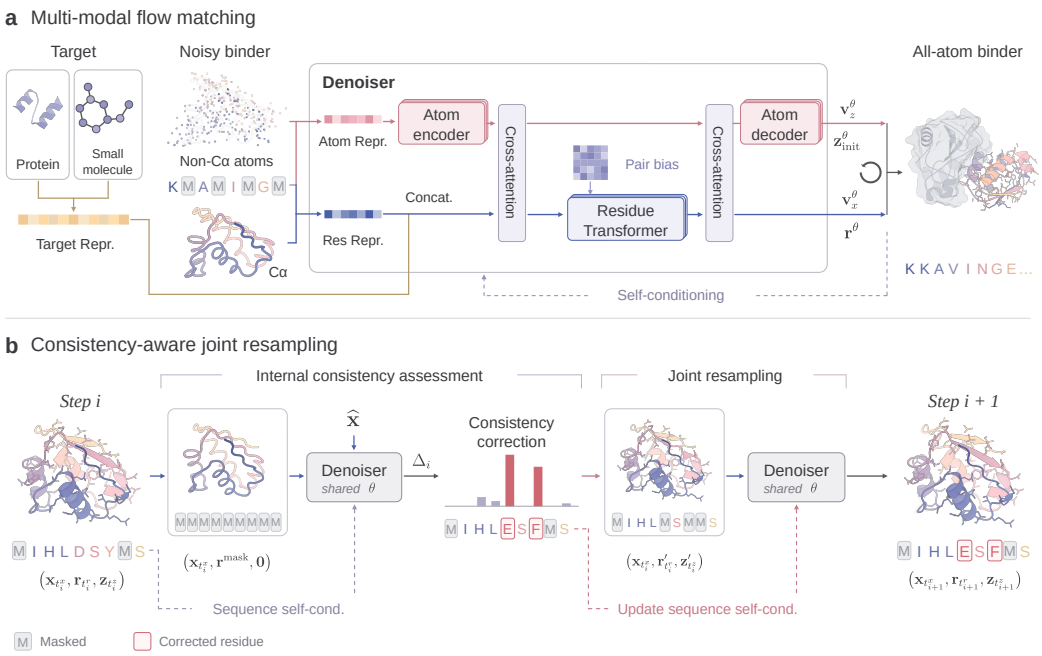


Figure 2: **Overview of CODesign and CAR. (a)** CODesign jointly generates a binder sequence and all-atom structure conditioned on a protein or small-molecule target using coupled residue- and atom-level denoising with self-conditioning. **(b)** CAR evaluates decoded residues against the evolving backbone, selects incompatible identities using the model's conditional sequence predictions, and jointly resamples their sequence and local atomic states.

2025), and experimentally resolved PINDER holo complexes (Kovtun et al., 2024). After source-specific filtering and restricting each chain to 40–300 residues, 191,836 reference complexes are processed to produce 104,528 distilled sequence–structure pairs. Each retained structure is co-folded directly from its paired sequence, yielding supervision that better approximates the consistent joint distribution in Equation 1. Further construction details are provided in Appendix A.

3.2 Multi-Modal Flow Matching

Target-conditioned explicit joint flow. A target-conditioned co-design model should approximate $p^\star(\mathbf{r}, \mathbf{x}, \mathbf{z} \mid \mathbf{c}^{\text{target}})$ while evolving sequence, backbone, and local atomic geometry within a shared sampling trajectory. This coupling allows sequence and structure predictions to influence one another throughout sampling. Because the three modalities can resolve at different stages of the trajectory, we independently sample their modality times $t_x, t_r, t_z \in [0, 1]$ during training, exposing the model to different conditional regimes. A nearly clean backbone paired with a highly masked sequence induces the structure-to-sequence prediction underlying $\mathcal{C}_{\text{seq}}$, whereas a nearly clean sequence paired with noisy coordinates induces the sequence-to-structure prediction underlying $\mathcal{C}_{\text{str}}$.

Fixed-dimensional all-atom representation. To jointly evolve sequences and all-atom structures in a fixed-dimensional state, we use Atom37, which assigns slots to the 37 possible heavy-atom positions. For a binder of length n, we represent the sequence as $\mathbf{r} \in \{0, \ldots, 19\}^n$, the C_α coordinates as $\mathbf{x} \in \mathbb{R}^{n\times 3}$, and the non-$C_\alpha$ coordinates as $\mathbf{z} \in \mathbb{R}^{n\times 36\times 3}$, where $\mathbf{z}_{ij}$ is the position of atom slot j relative to the C_α atom of residue i. Residue-specific validity masks exclude nonexistent atoms. For masked residues, we also hide the non-C_α coordinates and atom occupancy to prevent the atomic representation from revealing residue identity.

Multimodal conditional paths. To realize the conditional regimes described above, we define modality-specific paths from corrupted states at $t = 0$ to clean data at $t = 1$. We use continuous

flow matching for the coordinates and discrete flow matching for residues (Campbell et al., 2024):

$$\mathbf{x}_{t_x} = t_x \mathbf{x} + (1 - t_x)\mathbf{x}_0, \tag{6}$$

$$p_{t_r}(\cdot \mid r_i) = t_r \delta_{r_i} + (1 - t_r)\delta_{\mathtt{m}}, \qquad r_{t_r,i} \sim p_{t_r}(\cdot \mid r_i), \tag{7}$$

$$\mathbf{z}_{t_z} = t_z \mathbf{z} + (1 - t_z)\mathbf{z}_0. \tag{8}$$

Here, $\mathbf{x}_0$ and $\mathbf{z}_0$ denote the transformed backbone and local-coordinate reference endpoints, respectively; δ_a denotes a point mass at token a; and $\mathtt{m}$ is the mask token. Thus, residue i retains its clean identity with probability t_r and is masked otherwise. The backbone reference endpoint uses centered Gaussian noise with a randomized three-dimensional center, requiring the model to infer binder placement relative to the target. Exact noise transformations and modality-time distributions are provided in Appendix D.2.

Target-conditioned denoiser. The overall framework of CODesign is illustrated in Figure 2a. Our network, parameterized by θ, couples a residue-level Transformer trunk with an atom-level Transformer encoder–decoder (Reidenbach et al., 2025). Target embeddings are combined with noisy binder residue embeddings and processed using pair-biased attention over geometric features within and between the binder and target (Didi et al., 2026). Cross-attention exchanges information between the residue and atom trunks, connecting target context and global backbone geometry to local atomic structure.

Given $\mathbf{x}_{t_x}$, $\mathbf{r}_{t_r}$, $\mathbf{z}_{t_z}$, $\mathbf{c}^{\text{target}}$, and t_x, t_r, t_z, the network predicts a backbone vector field $\mathbf{v}_x^\theta$, sequence logits $\boldsymbol{\ell}^\theta$, a non-C_α vector field $\mathbf{v}_z^\theta$, and pseudo-atom initialization outputs $\mathbf{b}^\theta$ for masked residues. Self-conditioning uses a detached preliminary prediction and provides features derived from predicted C_α coordinates and sequence identities.

To strengthen the two conditional relationships in Equation 2, we additionally provide clean-backbone or clean-sequence feature channels intermittently during training. These modes retain the independently sampled noisy states and modality times while strengthening the structure-to-sequence and sequence-to-structure predictions, respectively. They are applied with probability 0.1 during protein training and 0.15 during ligand adaptation, with further details provided in Appendix D.3.

Training objective. The denoiser jointly predicts the clean backbone, residue identities, and local atomic geometry. We reconstruct the clean backbone from its vector field as

$$\widehat{\mathbf{x}} = \mathbf{x}_{t_x} + (1 - t_x)\mathbf{v}_x^\theta. \tag{9}$$

The backbone and sequence objectives are

$$\mathcal{L}_x = \mathrm{MSE}(\widehat{\mathbf{x}}, \mathbf{x}), \qquad \mathcal{L}_r = \mathrm{CE}(\boldsymbol{\ell}^\theta, \mathbf{r}), \tag{10}$$

For local atomic geometry, let

$$V_i = \mathbf{1}\{r_{t_r,i} \neq \mathtt{m}\} \tag{11}$$

indicate whether residue i is revealed, and let $M_{ia} \in \{0, 1\}$ indicate whether non-C_α atom slot $a \in \{1, \ldots, 36\}$ exists for its ground-truth identity. We denote by $\boldsymbol{\epsilon}_{z,ia}^{\text{raw}}$ the stored Gaussian draw used to construct the local-coordinate reference endpoint. Revealed residues are supervised through $\mathbf{v}_z^\theta$, whereas masked residues are supervised through $\mathbf{b}^\theta$:

$$\mathcal{L}_z = \frac{1}{n} \sum_{i,a} M_{ia} \left\| V_i \mathbf{v}_{z,ia}^\theta + (1 - V_i)\mathbf{b}_{ia}^\theta - \left(\mathbf{z}_{ia} - \boldsymbol{\epsilon}_{z,ia}^{\text{raw}}\right) \right\|_2^2. \tag{12}$$

Thus, the same atomic target supervises coordinate refinement at revealed residues and pseudo-atom initialization at masked residues. Our core training objective is

$$\mathcal{L} = \mathcal{L}_x + \mathcal{L}_r + \mathcal{L}_z. \tag{13}$$

Exact normalization, noise construction, and auxiliary training objectives are provided in Appendix D.4.

3.3 CONSISTENCY-AWARE JOINT RESAMPLING

Learning from consistent sequence–structure pairs does not prevent residue choices from becoming incompatible with the structure during sampling. We introduce Consistency-Aware joint Resampling (**CAR**), which uses the generator's own inverse-folding capability to identify such choices and jointly reset their sequence identities and local atomic states for resampling. CAR uses sequence–structure compatibility rather than random remasking to determine which residues to revisit, and, unlike search guided by external folding rewards (Didi et al., 2026), obtains its correction signal directly from the denoiser within a single sampling trajectory.

Internal consistency assessment. Independent modality times during training allow the model to observe low-noise backbones ($t_x \approx 1$) together with highly masked sequences ($t_r \approx 0$). Predicting residue identities from these inputs closely resembles inverse folding. The additional clean-backbone conditioning features used during training further strengthen this capability while retaining the independently sampled noisy states and modality times. CAR uses this learned structure-to-sequence relationship to reassess decoded residue identities as the structure evolves during sampling.

To make this assessment more reliable, we perform it during the later stages of backbone denoising. As illustrated in Figure 2b, we first compute the predicted clean C_α backbone

$$\widehat{\mathbf{x}} = \mathbf{x}_{t_x} + (1 - t_x)\mathbf{v}_x^\theta$$

from the current denoising prediction. Let $\mathcal{U} = \{i : r_{t_r,i} \neq \mathtt{m}\}$ denote the currently revealed positions. We remask these positions to obtain a fully masked sequence $\mathbf{r}^{\text{mask}}$. We also replace their non-C_α inputs with the zero-coordinate pseudo-atom state $\mathbf{z}^{\text{mask}} = \mathbf{0}$, preventing residue-specific atomic composition from revealing their identities. The predicted clean backbone $\widehat{\mathbf{x}}$ is supplied through the clean-backbone conditioning channel. Meanwhile, we retain the decoded identities in the sequence self-conditioning input $\mathbf{r}^{\text{sc}}$, encouraging refinement of the current sequence rather than an unrelated prediction from the backbone alone. Because substantially different amino-acid sequences can fold into similar structures (Koehler Leman et al., 2023), retaining the decoded sequence also prevents large deviations from the current sequence trajectory.

With these inputs, an additional forward pass performs a single-step inverse-folding assessment and produces sequence logits

$$\widetilde{\boldsymbol{\ell}}^\theta = \left[\text{Denoiser}_\theta\big(\mathbf{x}_{t_x}, \mathbf{r}^{\text{mask}}, \mathbf{z}^{\text{mask}}, \mathbf{c}^{\text{target}}, t_x, 0, t_z, \widehat{\mathbf{x}}, \mathbf{r}^{\text{sc}}\big)\right]_{\text{seq}}. \tag{14}$$

Let $q_i(a) = \text{softmax}(\widetilde{\boldsymbol{\ell}}_i^\theta)_a$ denote the predicted probability of amino-acid type a at position i. For each $i \in \mathcal{U}$, we compare the most probable type with the current identity:

$$a_i^\star = \underset{a \in \{0,\ldots,19\}}{\arg\max}\ q_i(a), \qquad \Delta_i = \log q_i(a_i^\star) - \log q_i(r_{t_r,i}). \tag{15}$$

A larger Δ_i indicates a stronger preference for replacing the current identity with $a_i^\star$. Among positions satisfying $a_i^\star \neq r_{t_r,i}$, we select those with the largest gaps, up to $\lceil \rho|\mathcal{U}| \rceil$ residues. Here, $\rho \in [0, 1]$ controls the resampling fraction, and $\mathcal{S}$ denotes the selected positions.

Joint resampling. For each selected position $i \in \mathcal{S}$, we use the preferred residue type $a_i^\star$ to guide resampling. Because changing a residue identity also changes its atomic composition, we reset its sequence and local atomic states together. Specifically, we remask the residue, replace its non-C_α input with the zero-coordinate pseudo-atom state, and update its sequence self-conditioning:

$$r_{t_r,i} \leftarrow \mathtt{m}, \qquad \mathbf{z}_{t_z,i} \leftarrow \mathbf{0}, \qquad r_i^{\text{sc}} \leftarrow a_i^\star, \qquad i \in \mathcal{S}. \tag{16}$$

The C_α coordinates and unselected residue states are not reset. We then run the denoiser again with the updated states and self-conditioning at the original times t_x, t_r, t_z, and continue sampling using the new predictions. The discrete flow samples residue identities from the updated logits $\boldsymbol{\ell}^\theta$; the preferred types $a_i^\star$ guide this process through self-conditioning rather than being assigned directly. Newly revealed atoms are initialized by the pseudo-atom initialization output $\mathbf{b}^\theta$ and subsequently refined by $\mathbf{v}_z^\theta$.

Although $\mathbf{x}_{t_x}$ is not reset directly, the additional forward pass recomputes $\mathbf{v}_x^\theta$ using the revised sequence self-conditioning. The subsequent backbone update can therefore respond to the preferred residue types. CAR thus corrects the sequence while allowing both the backbone and local atomic geometry to adjust, rather than refining the sequence alone. Sampling schedules and implementation details are provided in the Appendix E.

Table 1: Results of protein-conditioned protein binder design benchmark. scRMSD is reported as the mean value / percentage of samples with scRMSD $\leq$ 2 Å. Best and second-best results are highlighted in **bold** and underlined, respectively. Complete results with detailed standard deviations are provided in Appendix F.2.

Model	# Unique Successes ↑			scRMSD ↓ / $\leq$ 2 Å (%) ↑			Diversity ↑	Novelty ↓
	Self	MPNN@1	MPNN@8	Self	MPNN@1	MPNN@8		
two-stage								
RFDiffusion3	–	6.46	10.96	–	7.24 / 36.12	4.17 / 53.02	25.66	0.91
PXDesign	–	4.59	7.01	–	**5.21 / 51.15**	**2.29 / 74.54**	15.39	0.92
Protpardelle-1c	–	0.28	0.92	–	16.44 / 10.26	12.61 / 20.40	**75.15**	0.88
co-design								
Complexa	8.54	8.53	14.65	7.91 / 35.06	7.50 / 35.28	4.49 / 51.18	68.92	0.88
CODesign	**9.84**	**10.32**	**18.81**	**7.01 / 41.64**	7.01 / 40.53	4.09 / 57.16	73.92	**0.87**

3.4 Stagewise Training

We first train the protein-conditioned generator on Teddymer (Didi et al., 2026) and experimental PDB data for 60k iterations with a batch size of 5, and then continue training for 10k iterations with CODIMER incorporated into the mixture. The first stage enables the model to gain fundamental knowledge of protein-protein interactions from large-scale data, while the second introduces targeted consistency supervision. For small-molecule targets, we adapt the protein-conditioned model using LoRA (Hu et al., 2022) to avoid overfitting, training on a mixture of PLINDER protein-ligand complexes (Durairaj et al., 2024) and monomers from Consistency Distilled Synthetic Protein Database (CDDB). The CDDB monomers retain consistency supervision alongside ligand-specific conditioning. We adopt a batch size of 3 for the training of the ligand-conditioned generator. The two target-specific variants share the explicit generation and CAR formulations. Please refer to Appendix D for more details about training.

4 Experiments

4.1 Protein-conditioned Protein Binder Design

Experimental Configuration We follow the protein-target binder design benchmark of Proteína-Complexa (Didi et al., 2026), which includes 19 targets from AlphaProteo (Zambaldi et al., 2024) and BindCraft (Pacesa et al., 2025). Both self-generated sequences (**Self**) and the ones redesigned with ProteinMPNN (Dauparas et al., 2022) are evaluated. For each generated backbone, **MPNN@1** and **MPNN@8** redesign one and eight binder sequences respectively.Each sequence is co-folded with its target using AlphaFold2-Multimer (Evans et al., 2021) with templating.All experiments are performed 5 times with different random seeds.

Metrics We calculate average **Unique Successes** over the 19 targets using Foldseek (Van Kempen et al., 2024) for clustering. A design is considered successful if its refolded complex has ipAE $<$ 7 Å, complex pLDDT > 0.9, and binder C_α scRMSD $<$ 1.5 Å. To evaluate the overall consistency, we also calculate the average complex C_α **scRMSD** and the percentage of samples with scRMSD $\leq$ 2 Å for each sequence type. We measure **Diversity** as the average number of Foldseek clusters among all generated binder structures on each target. **Novelty** is measured by the mean best-match TM-score against PDB for successful binders, averaged across targets.

Baselines We compare CODesign against two-stage methods including RFDiffusion3 (Butcher et al., 2025), PXDesign (Team et al., 2025) and Protpardelle-1c (Lu et al., 2025); as well as a co-design method Proteína-Complexa (Didi et al., 2026). The two-stage models only generate binder structures and use ProteinMPNN for sequence design. All methods are evaluated under the same benchmark protocol without external reward-guided search or hallucination-based refinement.

Table 2: Results of ligand-conditioned protein binder design benchmark. Best and second-best results are highlighted in **bold** and underlined, respectively. Complete results with detailed standard deviations are provided in Appendix F.3.

Model	# Unique Successes ↑					Diversity ↑	Novelty ↓	Time (s) ↓
	SAM	OQO	FAD	IAI	Mean			
two-stage								
RFdiffusionAA	2.0	3.4	1.6	5.0	3.0	135.65	**0.74**	73.78
RFDiffusion3	9.4	14.8	20.8	11.0	14.0	52.50	0.83	14.96
co-design								
Complexa	3.2	3.0	5.8	8.4	5.1	**147.70**	0.76	**10.27**
Pallatom-Ligand	0.2	4.4	2.0	3.2	2.5	17.55	0.87	40.44
DISCO	**10.6**	14.0	**25.4**	12.8	15.7	60.10	0.85	181.77
CODesign (ours)	5.8	**37.2**	21.4	**18.6**	**20.8**	102.35	0.82	39.90

Results. Table 1 summarizes performance on protein-target binder design. CODesign achieves the highest number of unique successes across self-generated sequences and both ProteinMPNN redesign settings. It also obtains the strongest self-consistency among co-design methods using model-generated sequences. Although PXDesign attains the highest fraction of designs with $\mathrm{scRMSD} \leq 2$ Å after ProteinMPNN redesign, this improvement is accompanied by substantially lower diversity and novelty. Overall, these results show that Consistensy-awared model provides a stronger balance between structural consistency, diversity, and novelty than competing methods.

4.2 Ligand-conditioned Protein Binder Design

Experimental Configuration We follow the small-molecule binder design benchmark employed by Didi et al. (2026), which includes four targets: SAM, OQO, FAD, and IAI. For each method and target, we generate 200 binders of 100 residues using the same ligand inputs as Didi et al. (2026). We evaluate co-design methods using their self-generated sequences, while two-stage methods use one sequence designed by LigandMPNN (Dauparas et al., 2025) for each generated backbone. Each sequence is co-folded with its ligand target using RosettaFold-3 (RF3) (Corley et al., 2025).

Metrics Similar to the protein target benchmark, we report **Unique Successes** for each target and their average across the four targets. A design is considered successful if its refolded complex has min-ipAE < 2 Å, binder C_α scRMSD < 2 Å, and binder-aligned ligand scRMSD < 5 Å. Here, min-ipAE is the minimum entry in the protein-ligand blocks of the RF3-predicted pAE matrix. The ligand scRMSD is computed after aligning the refolded binder to the generated one. We additionally evaluate the **Diversity** and **Novelty** following the same protocol in the protein-target benchmark. The per-sample generation **Time** is recorded on a single GPU with batch size set to 1. Reported values are averaged results over five random seeds.

Baselines We select both two-stage methods (RFdiffusionAA (Krishna et al., 2024) and RFDiffusion3 (Butcher et al., 2025)) and co-design methods(Proteína-Complexa (Didi et al., 2026), Pallatom-Ligand (Wang et al., 2026), and DISCO (Rector-Brooks et al., 2026)) for baseline models. For fair comparison, no external reward-guided search or hallucination-based refinement is applied.

Results Table 2 summarizes the results. summarizes performance on ligand-conditioned binder design. CODesign achieves the highest mean number of unique successes, ranking first on OQO and IAI and second on FAD. It also substantially outperforms the strongest competing co-design method, DISCO, in mean unique successes while requiring considerably less inference time. Although Complexa attains higher diversity and faster generation, it produces substantially fewer successful designs; CODesign maintains the second-highest diversity among co-design methods while offering competitive novelty. Overall, these results demonstrate that CODesign generalizes effectively to small-molecule targets and provides a strong balance among success rate, diversity, novelty, and computational cost.

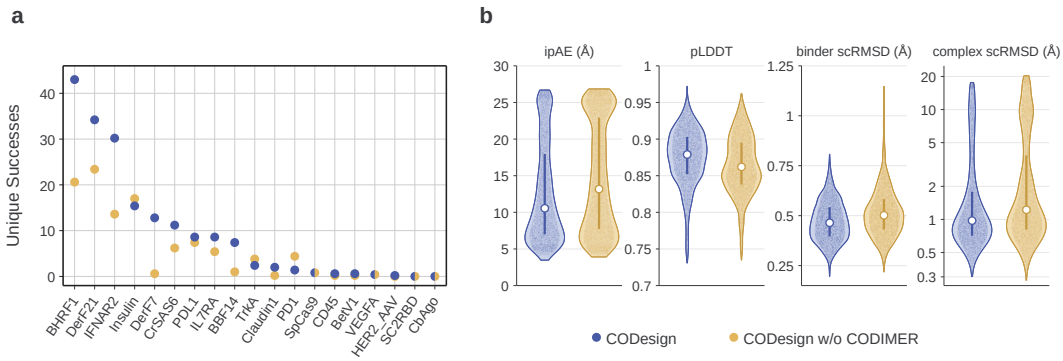


Figure 3: Impact of CODIMER on the model's performance. **(a)** Average unique successes across 5 random seeds on each target. Complete results with detailed standard deviations are provided in Appendix F.4. **(b)** Distributions of different confidence and consistency metrics.

4.3 Ablation Studies

Significance of CODIMER To investigate how CODIMER influences the model's performance, we additionally train a CODesign variant only on Teddymer and PDB as in Section 3.4 We compared generations without CAR and only the model-generated sequences with 5 random seeds. .

As demonstrated in Fig. 3a, the unique successes significantly drop across most targets when training without CODIMER. Specifically, including CODIMER in training can promote performance across 12/19 targets with a substantial average improvement of 70.9% (9.46 v.s. 5.54) on mean unique successes, as evident in Table 19. In Fig. 3b, we further identify that CODIMER increases the performance mainly by improving the physical plausibility of the interface and local structures as well as the complex-level consistency.

Impact of consistency-aware sampling To assess the impact of CAR, we evaluate the same CODesign model trained with CODIMER, disabling only CAR during sampling. We evaluate both variants on the protein-target binder design benchmark over five random seeds. As shown in Figure 4, CAR improves the average number of unique successes across 19 targets by 4%, with negligible additional computational cost. CAR therefore improves consistency by leveraging the model's intrinsic correction capability, without relying on external rewards.

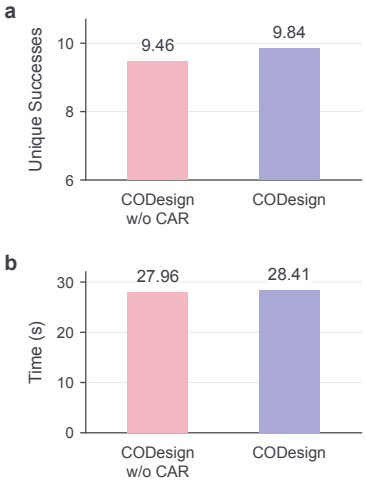


Figure 4: Impact of CAR on **(a)** average unique successes across 19 protein targets and **(b)** per-sample generation time.

5 Conclusions

We introduced CODesign, a consistency-driven framework for target-conditioned all-atom protein binder co-design. CODesign improves sequence–structure consistency through distilled training pairs and consistency-aware joint resampling, enabling joint generation of compatible sequences, backbones, and local atomic structures. It achieves state-of-the-art in silico success rates on protein- and small-molecule-target benchmarks, with ablations validating both components. These results highlight modality consistency as a key principle for *de novo* protein design. However, evaluation is limited to in silico metrics, and the benchmark's reliance on AlphaFold-based structure prediction and ProteinMPNN-based inverse folding may amplify the apparent importance of consistency. Whether these gains translate into higher binding affinity and experimental success remains to be established through wet-lab validation.

AI USE STATEMENT

In this work, we used generative AI tools for refining research hypotheses, providing feedback on experimental design, and assisting with translation. We have not used generative AI tools for generating synthetic datasets, developing theoretical models, formulating or proving mathematical claims, implementing methods, cleaning or reformatting datasets, or interpreting experimental results, and qualitative or thematic data analysis is not applicable to this work. Additionally, we used generative AI tools for drafting of the manuscript, literature search and summarization, improving readability, and debugging code. We have reviewed all AI-assisted work. AI-assisted research suggestions were independently evaluated by the authors, references and scientific claims were checked against the original sources, and AI-assisted debugging was verified through code execution and testing. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Woody Ahern, Jason Yim, Doug Tischer, Saman Salike, Seth M Woodbury, Donghyo Kim, Indrek Kalvet, Yakov Kipnis, Brian Coventry, Han Raut Altae-Tran, et al. Atom-level enzyme active site scaffolding using rfdiffusion2. *Nature Methods*, 23(1):96–105, 2026.

Jasper Butcher, Rohith Krishna, Raktim Mitra, Rafael I Brent, Yanjing Li, Nathaniel Corley, Paul T Kim, Jonathan Funk, Simon Mathis, Saman Salike, et al. 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.

Salvatore Candido, Thomas Hayes, Alexander Derry, Roshan Rao, Zeming Lin, Robert Verkuil, Bryan Wu, Jin Sub Lee, Elise S. Bruguera, Jehan A. Keval, Mykhailo Kopylov, John E. Pak, Wesley Wu, Neil Thomas, Samson Mataraso, Alvin Hsu, Ashton C. Trotman-Grant, Kilian Fatras, Allan dos Santos Costa, Rohil Badkundri, Halil Akın, Deniz Oktay, Jonathan Deaton, Elizabeth Montabana, Hrishita Sitwala, Yue Yu, Marius Wiggert, Dylan Alexander Carlin, Anthony W. Goering, Tomasz Blazejewski, McCullen Sandora, Michael Hla, Tina Z. Jia, Leon H. Kloker, Nicholas J. Sofroniew, Masatoshi Uehara, Jassi Pannu, Sharrol Bachas, Daniel S. Liu, Tom Sercu, and Alexander Rives. Language modeling materializes a world model of protein biology, 2026. URL https://www.biorxiv.org/content/10.64898/2026.06.03.729735. Preprint.

Chaoran Cheng, Jiaqi Guan, Milong Ren, Chengyue Gong, Cong Liu, Xinshi Chen, Ge Liu, and Wenzhi Xiao. A-code: Fully atomic protein co-design with unified multimodal diffusion. *arXiv preprint arXiv:2605.03360*, 2026.

Claude Science and Amir Shanehsazzadeh. Autonomous de novo protein binder design with claude. Technical report, Anthropic, August 2026. URL https://www-cdn.anthropic.com/30bf50e22a01388bb29bf077ee3f244531594b7a.pdf.

Nathaniel Corley, Simon Mathis, Rohith Krishna, Magnus S Bauer, Tuscan R Thompson, Woody Ahern, Maxwell W Kazman, Rafael I Brent, Kieran Didi, Andrew Kubaney, et al. Accelerating biomolecular modeling with atomworks and rf3. *BioRxiv*, 2025.

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.

Justas Dauparas, Gyu Rie Lee, Robert Pecoraro, Linna An, Ivan Anishchenko, Cameron Glasscock, and David Baker. Atomic context-conditioned protein sequence design using ligandmpnn. *Nature Methods*, 22(4):717–723, 2025.

Kieran Didi, Zuobai Zhang, Guoqing Zhou, Danny Reidenbach, Zhonglin Cao, Sooyoung Cha, Tomas Geffner, Christian Dallago, Jian Tang, Michael Bronstein, et al. Scaling atomistic protein binder design with generative pretraining and test-time compute. In *International Conference on Learning Representations*, volume 2026, pp. 61056–61109, 2026.

Janani Durairaj, Yusuf Adeshina, Zhonglin Cao, Xuejin Zhang, Vladas Oleinikovas, Thomas Duignan, Zachary McClure, Xavier Robin, Gabriel Studer, Daniel Kovtun, et al. Plinder: The protein-ligand interactions dataset and evaluation resource. *BioRxiv*, pp. 2024–07, 2024.

Richard Evans, Michael O'neill, Alexander Pritzel, Natasha Antropova, Andrew Senior, Tim Green, Augustin Žídek, Russ Bates, Sam Blackwell, Jason Yim, et al. Protein complex prediction with alphafold-multimer. *biorxiv*, pp. 2021–10, 2021.

Tomas Geffner, Kieran Didi, Zuobai Zhang, Danny Reidenbach, Zhonglin Cao, Jason Yim, Mario Geiger, Christian Dallago, Emine Kucukbenli, Arash Vahdat, et al. Proteina: Scaling flow-based protein structure generative models. In *International Conference on Learning Representations*, volume 2025, pp. 98803–98851, 2025.

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.

Yewon Han, Maxim I Tsenkov, Niccolò AE Venanzi, Sooyoung Cha, Nilkanth Patel, Sreenath Nair, Razan Abbara, Damian Bertoni, Alejandro Chacón, Nick Dietrich, et al. Alphafold database expands to proteome-scale quaternary structures. *bioRxiv*, pp. 2026–03, 2026.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Julia Koehler Leman, Pawel Szczerbiak, P Douglas Renfrew, Vladimir Gligorijevic, Daniel Berenberg, Tommi Vatanen, Bryn C Taylor, Chris Chandler, Stefan Janssen, Andras Pataki, et al. Sequence-structure-function relationships in the microbial protein universe. *Nature communications*, 14(1):2351, 2023.

Daniel Kovtun, Mehmet Akdel, Alexander Goncearenco, Guoqing Zhou, Graham Holt, David Baugher, Dejun Lin, Yusuf Adeshina, Thomas Castiglione, Xiaoyun Wang, et al. Pinder: the protein interaction dataset and evaluation resource. *bioRxiv*, pp. 2024–07, 2024.

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.

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.

Martin Pacesa, Lennart Nickel, Christian Schellhaas, Joseph Schmidt, Ekaterina Pyatova, Lucas Kissling, Patrick Barendse, Jagrity Choudhury, Srajan Kapoor, Ana Alcaraz-Serna, et al. One-shot design of functional protein binders with bindcraft. *Nature*, 646(8084):483–492, 2025.

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.

Jarrid Rector-Brooks, Théophile Lambert, Marta Skreta, Daniel Roth, Yueming Long, Zi-Qi Li, Xi Zhang, Miruna Cretu, Francesca-Zhoufan Li, Tanvi Ganapathy, et al. General multimodal protein design enables dna-encoding of chemistry. *arXiv preprint arXiv:2604.05181*, 2026.

Danny Reidenbach, Zhonglin Cao, Zuobai Zhang, Kieran Didi, Tomas Geffner, Guoqing Zhou, Jian Tang, Christian Dallago, Arash Vahdat, Emine Kucukbenli, et al. Consistent synthetic sequences unlock structural diversity in fully atomistic de novo protein design. *arXiv preprint arXiv:2512.01976*, 2025.

Protenix Team, Milong Ren, Jinyuan Sun, Jiaqi Guan, Cong Liu, Chengyue Gong, Yuzhe Wang, Lan Wang, Qixu Cai, Wenzhi Ma, et al. 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.

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.

Haochen Wang, Qianyi Wang, Rui Ma, Jiawei Guan, Weikun Wu, Haobo Wang, and Jiayi Dou. Pallatom-ligand: an all-atom diffusion model for designing ligand-binding proteins. In *The Fourteenth International Conference on Learning Representations*, 2026.

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.

Vinicius Zambaldi, David La, Alexander E Chu, Harshnira Patani, Amy E Danson, Tristan OC Kwan, Thomas Frerix, Rosalia G Schneider, David Saxton, Ashok Thillaisundaram, et al. De novo design of high-affinity protein binders with alphaproteo. *arXiv preprint arXiv:2409.08022*, 2024.

Jing Zhang, Ian R Humphreys, Jimin Pei, Jinuk Kim, Chulwon Choi, Rongqing Yuan, Jesse Durham, Siqi Liu, Hee-Jung Choi, Minkyung Baek, et al. Predicting protein-protein interactions in the human proteome. *Science*, 390(6771):eadt1630, 2025.

A Details of Codimer

A.1 Data Sources and Preprocessing

We collect protein dimers from three sources, requiring both chains to contain 40–300 residues. Source-specific selection proceeds as follows:

- **AFDB high-confidence heterodimers.** We use the June 8, 2026 metadata snapshot of the AFDB/NVIDIA high-confidence heterodimer collection (Han et al., 2026). The collection requires maximum ipSAE ≥ 0.6 and maximum directional pDockQ2 ≥ 0.23. Retaining entries with available structures and applying the chain-length constraints yields 25,837 dimers.
- **HumanPPI.** We use HumanPPI predictions at 80% expected precision (Zhang et al., 2025), retaining protein pairs with available two-chain structural models. When multiple segment-pair models correspond to the same protein pair, we select the model with the highest predicted inter-protein contact probability. Applying the chain-length constraints yields 9,757 dimers.
- **PINDER.** We select native holo dimers from the February 2024 release of PINDER (Kovtun et al., 2024). We retain heteromeric complexes with biological-interface annotations and distinct entities for the two chains. Both chains must be fully resolved and contain a single connected component. We additionally require no interface atom gaps within 4 Å, at least five intermolecular contacts, and per-chain `max_var` metadata values ≤ 0.98. Together with the chain-length constraints, these filters yield 156,242 dimers.

The combined collection contains 191,836 dimers, which serve as the reference structures for consistency distillation.

A.2 Consistency Distillation

We apply the same construction procedure to all three sources:

Table 3: Composition of *Codimer*. Each reference dimer produces four co-folded candidates and contributes at most one sequence–structure pair to the final dataset.

Source	Reference dimers	Codimer pairs
AFDB high-confidence heterodimers	25,837	23,823
HumanPPI	9,757	5,177
PINDER	156,242	75,528
Total	191,836	104,528

1. **Structure preparation.** For each input dimer, we use the first coordinate model and retain canonical amino-acid residues with Cα coordinates from both chains. We standardize chain identifiers and renumber residues consecutively within each chain. The prepared structure serves as the input to sequence redesign and the reference for subsequent structural comparison.
2. **Joint sequence redesign.** We use the Cα-only version of ProteinMPNN (Dauparas et al., 2022) to sample four sequence pairs per dimer at a temperature of 0.1. Both chains are redesigned jointly, rather than fixing the sequence of either interaction partner.
3. **All-atom co-folding.** Each redesigned sequence pair is independently co-folded with ESMFold2-Fast (Candido et al., 2026), using 10 loops, 68 sampling steps, and one diffusion sample. This produces four candidate complexes per input, or 767,344 candidates in total. Co-folding provides atomistic structures corresponding to the redesigned sequences, including the side-chain geometries and atom compositions associated with the new residue identities (Reidenbach et al., 2025).
4. **Confidence filtering.** We retain candidates with mean pLDDT across both chains ≥ 0.7 on the $[0, 1]$ scale and interface pAE ≤ 10 Å. Interface contacts are defined by cross-chain Cα distances ≤ 8 Å, and interface pAE is averaged over these residue pairs.
5. **Candidate selection.** For each input dimer, we select the confidence-qualified candidate with the lowest complex-level Cα RMSD to the reference structure. Inputs with no qualifying candidates are discarded. The selected sequence pair and its co-folded all-atom structure constitute one sample in *Codimer*.

This procedure yields 104,528 synthetic sequence–structure pairs. Table 3 summarizes the contribution of each source.

B Additional Training Data and Preprocessing

B.1 Training Datasets

In addition to CODIMER, CODesign uses Teddymer and PDB structures for protein-conditioned training, and CDDB and PLINDER for ligand adaptation. We describe the role of each dataset below. Their sampling probabilities are reported in Table 4.

Teddymer and PDB. We use representatives of nonsingleton Teddymer clusters to provide domain–domain interaction examples, following Didi et al. (2026). For experimental data, we use PDB structures selected by the PDB identifier list supplied with the Complexa repository. Entries with multiple protein entities and one chain per entity provide binder–target examples. Entries with repeated copies of an entity, or only one protein entity, provide monomer examples with empty target conditioning. This allows the same generator to learn from both interacting proteins and individual chains. CODIMER is added during the second protein-training stage; its construction is described in Appendix A.

CDDB and PLINDER. For ligand adaptation, CDDB provides consistency-distilled monomers of 32–250 residues (Reidenbach et al., 2025). PLINDER provides examples containing one selected protein chain and one connected small-molecule ligand (Durairaj et al., 2024). CDDB examples

Table 4: Source-sampling probabilities at each training stage. CODIMER is treated as a single distilled dataset. PDB monomers combine repeated-entity and single-entity examples.

Dataset	Protein I	Protein II	Ligand adaptation
Teddymer	0.80	0.70	–
PDB binder–target examples	0.19	0.10	–
PDB monomers	0.01	0.01	–
CODIMER	–	0.19	–
CDDB monomers	–	–	0.60
PLINDER	–	–	0.40

have no target, whereas PLINDER examples retain the ligand as fixed conditioning information. Mixing the two sources preserves protein sequence–structure supervision while adapting the model to ligand-specific interactions.

B.2 CROPPING AND GEOMETRIC PREPROCESSING

Protein targets. We identify intermolecular contacts using an 8 Å cutoff on cross-chain C_α distances. An interface residue is selected as the binder seed, and a contiguous segment containing this residue is cropped from its chain. The binder contains at most 250 residues. Target residues are selected from a 15 Å neighborhood of the cropped binder, subject to a combined budget of 500 binder and target residues. A valid conditioned crop retains at least 50 target residues and may contain multiple target chains. Examples assigned to monomer generation, or those without a valid target crop, use empty target conditioning. Contact and hotspot annotations are randomly subsampled and dropped during training.

Small-molecule targets. We use a spatial crop centered around the protein–ligand system, with a budget of 300 tokens. A protein residue counts as one token and a ligand heavy atom counts as one token. The crop-center distance cutoff is 15 Å. The ligand supplies fixed coordinates and chemical features; only the protein is generated.

Coordinate processing. Coordinates are converted to nanometers and augmented by a random rigid rotation. Conditioned examples are centered on the target, while monomers are centered on the protein. Non-C_α coordinates are then expressed relative to the corresponding C_α position. These offsets remain in the shared Cartesian coordinate system; no residue-specific rotation is applied. Protein-training metadata are shuffled with seed 42 and split into 99% training and 1% validation examples. The ligand-adaptation configuration uses the supplied CDDB and PLINDER manifests for training without an additional loader-level validation split.

C MODEL ARCHITECTURE

C.1 RESIDUE AND ATOM REPRESENTATIONS

We use the notation of Section 3.2: $\mathbf{r}$ denotes the binder sequence, $\mathbf{x}$ its C_α coordinates, $\mathbf{z}$ its non-C_α offsets, and $\mathbf{c}^{\text{target}}$ the fixed target features. The binder contains n residues. For an existing non-C_α atom slot a in residue i, the corresponding global coordinate is $\mathbf{x}_i + \mathbf{z}_{ia}$.

Following Proteína-Atomística (Reidenbach et al., 2025), the network combines a residue representation with an expanded atom representation. A revealed residue contributes its C_α token and existing non-C_α atoms. A masked residue instead contributes a C_α token and a zero-coordinate pseudo-atom token. Its residue-specific atomic composition is hidden. The pseudo-atom output provides $\mathbf{b}_i^\theta \in \mathbb{R}^{36\times 3}$, from which valid atom slots are selected after a residue identity is sampled. This keeps the state dimension fixed while allowing different amino acids to instantiate different atom sets.

Table 5: CODesign architecture. Protein- and ligand-conditioned variants use the same dimensions and layer counts, with different target feature encoders.

Hyperparameter	Protein-conditioned	Ligand-conditioned
Residue trunk		
Representation dimension d_{res}	768	768
Conditioning dimension	256	256
Pair dimension d_{pair}	256	256
Transformer layers	14	14
Attention heads	12	12
Time embedding dimension	256	256
Residue-index embedding dimension	256	256
Residue-type embedding dimension	512	512
Relative sequence-separation dimension	127	127
Noisy / self-conditioned distance bins	30 / 30	30 / 30
Distance range (nm)	0.1–3.0	0.1–3.0
Register tokens	0	0
Triangular pair updates	No	No
Atom encoder–decoder		
Representation / conditioning dimension	128 / 128	128 / 128
Encoder / decoder layers	2 / 3	2 / 3
Self-attention heads	8	8
Cross-attention heads	8	8
Atom-type / residue-type embedding dimension	128 / 128	128 / 128
Residue-index embedding dimension	128	128

C.2 Target Conditioning and Information Exchange

Input features. Binder residue features contain the noisy C_α coordinates and sequence, residue indices, self-conditioning features, and optional clean-backbone or clean-sequence features. Protein targets are represented at the residue level using amino-acid identities, Atom37 coordinates and masks, angular features, and hotspot indicators. Small-molecule targets are represented at the heavy-atom level using coordinates, element and atom-name encodings, charges, and graph Laplacian positional features. Following Didi et al. (2026), target embeddings are concatenated with binder residue embeddings along the token dimension.

Let n_{target} be the number of target residues for protein conditioning or target heavy atoms for ligand conditioning. The joint token representation has shape $(n + n_{\text{target}}) \times d_{\text{res}}$. The pair representation has shape $(n+n_{\text{target}})\times(n+n_{\text{target}})\times d_{\text{pair}}$ and combines binder–binder, target–target, and binder–target features. These features encode sequence separation, chain membership, hotspot annotations, and available geometric distances. Ligand–ligand features additionally encode bond connectivity and bond order.

Denoiser. An atom-level Transformer encoder processes local coordinates, atom types, residue identities, and residue indices. Cross-attention passes this information to the residue representation. The joint binder–target representation is then processed by a residue-level Transformer with pair-biased attention. A second cross-attention module passes its updated context to the atom decoder. Pair features remain fixed within each forward evaluation; the network does not use triangular pair updates.

The residue trunk is conditioned on t_x and t_r, and the atom branch on t_z. Time embeddings modulate attention and transition blocks through adaptive normalization and output scaling. The residue outputs predict $\mathbf{v}_x^\theta$ and $\boldsymbol{\ell}^\theta$, while the atom outputs predict $\mathbf{v}_z^\theta$ and $\mathbf{b}^\theta$. Only binder outputs are retained. The target remains unchanged throughout generation.

Table 6: Training settings. Checkpoint steps are optimizer updates within the corresponding stage. Global batch sizes equal the per-GPU batch sizes multiplied by the number of GPUs.

Setting	Protein I	Protein II	Ligand adaptation
Number of GPUs	24	64	32
Batch size per GPU	5	5	3
Global batch size	120	320	96
Updates at selected checkpoint	60,000	10,000	10,000
Trainable parameters	Full model	Full model	LoRA adapters
Optimizer	Adam	Adam	Adam
Learning rate	10^{-4}	10^{-4}	10^{-4}
Gradient accumulation steps	1	1	1
Numerical precision	Float32	Float32	Float32
EMA decay	0.999	0.999	0.999
Random seed	42	42	42
Self-conditioning probability	0.50	0.50	0.50
Clean-backbone conditioning probability	0.05	0.05	0.075
Clean-sequence conditioning probability	0.05	0.05	0.075
LoRA rank / scaling coefficient	–	–	32 / 64

D Training Details

D.1 Stagewise Optimization

We first train a protein-conditioned model on Teddymer and PDB, and then continue training after adding CODIMER. The first stage initializes shape-compatible parameters from the pretrained Complexa checkpoint. Newly introduced parameters are initialized independently. The second stage starts from the first-stage exponential moving average (EMA) weights. Both stages optimize the full generator. The data mixture changes between stages as shown in Table 4, while the architecture and training objectives remain the same.

For small-molecule targets, we initialize from the second-stage protein model and apply LoRA (Hu et al., 2022) to eligible linear and embedding layers, including the ligand-conditioning projections. For a linear weight matrix, the adapted weight is

$$W = W_0 + \frac{\alpha_{\mathrm{LoRA}}}{r_{\mathrm{LoRA}}} BA, \qquad r_{\mathrm{LoRA}} = 32, \quad \alpha_{\mathrm{LoRA}} = 64. \tag{17}$$

The base weights and biases are frozen, and only adapter parameters are optimized. Adapter dropout is zero. This stage uses the CDDB–PLINDER mixture in Table 4.

All stages use Adam with a learning rate of 10^{-4}, one optimizer update per batch, and no gradient accumulation. EMA weights are maintained with decay 0.999. Table 6 reports the training settings and the checkpoint steps selected by the stage-transition and benchmark configurations.

D.2 Backbone Noise and Modality-Time Sampling

We follow the conditional paths in Equations (6)–(8). For the backbone, the initial noise is centered across binder residues and given a random three-dimensional center:

$$\mathbf{x}_{0,i} = \boldsymbol{\epsilon}_{x,i} - \overline{\boldsymbol{\epsilon}}_x + \mathbf{d}, \qquad \boldsymbol{\epsilon}_{x,i} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_3), \quad \mathbf{d} \sim \mathcal{N}(\mathbf{0}, c_d^2 \mathbf{I}_3), \quad c_d = 0.2. \tag{18}$$

Here, $\overline{\boldsymbol{\epsilon}}_x$ is the mean of the initial Gaussian noise over the binder. Coordinates are in nanometers. Randomizing the noise center prevents the clean binder center from being determined directly from a centered-noise interpolation, encouraging the denoiser to infer placement relative to the target (Didi et al., 2026).

The three times are sampled independently. The coordinate modalities share the same marginal time distribution but do not share their sampled time:

$$\begin{aligned} t_x, t_z &\overset{\text{ind}}{\sim} 0.02\,\mathcal{U}(0,1) + 0.98\ \mathrm{Beta}(1.9, 1), \\ t_r &\sim \mathcal{U}(0,1). \end{aligned} \tag{19}$$

Independent times expose the network to different combinations of sequence and structure uncertainty. The uniform component in the coordinate distribution retains coverage across noise levels, while the beta component emphasizes more resolved structures.

D.3 Self-Conditioning and Conditional Training

With probability 0.5, we perform a preliminary denoiser evaluation without gradient tracking. Its predicted clean C_α coordinates and residue identities are supplied as self-conditioning features in the training evaluation. The preliminary predictions are detached, so the loss does not backpropagate through the first evaluation.

We additionally enable either the clean-backbone feature channel or the clean-sequence feature channel in a subset of training iterations. The two modes are mutually exclusive and equally likely. Their combined probability is 0.1 during protein training and 0.15 during ligand adaptation. They provide ground-truth conditioning features without replacing the independently sampled noisy states or their modality times. Clean-backbone conditioning strengthens structure-to-sequence prediction, while clean-sequence conditioning strengthens sequence-to-structure prediction. These are the same conditional capabilities used by CAR; no separate inverse-folding network is trained.

D.4 Loss Normalization and Auxiliary Supervision

The objectives in Section 3.2 are evaluated per example and then averaged over the batch. Using the clean-backbone estimate in Equation (9), the backbone loss includes a time-dependent normalization:

$$\mathcal{L}_x = \frac{1}{n\left[(1-t_x)^2 + 10^{-5}\right]} \sum_{i=1}^{n} \|\widehat{\mathbf{x}}_i - \mathbf{x}_i\|_2^2 . \tag{20}$$

We use the sequence cross-entropy $\mathcal{L}_r$ and local-atom objective $\mathcal{L}_z$ defined in Equations (10) and (12). The local-atom objective supervises $\mathbf{v}_z^\theta$ at revealed residues and $\mathbf{b}^\theta$ at masked residues, using the visibility indicator V_i and atom mask M_{ia} from the main text.

To supervise binder placement, we add a binder–target distance loss. Let $\mathbf{y}_j^{\text{target}}$ denote a target reference point: a C_α coordinate for a protein target, or a heavy-atom coordinate for a small-molecule target. With n_{target} such points,

$$\mathcal{L}_{\text{bt}} = \frac{1}{n n_{\text{target}}} \sum_{i=1}^{n} \sum_{j=1}^{n_{\text{target}}} \left(\left\|\widehat{\mathbf{x}}_i - \mathbf{y}_j^{\text{target}}\right\|_2 - \left\|\mathbf{x}_i - \mathbf{y}_j^{\text{target}}\right\|_2 \right)^2 . \tag{21}$$

The term is zero for monomer examples and is enabled only when $t_x > 0.3$. The complete training objective is

$$\mathcal{L}_{\text{train}} = \mathbb{E}[\mathcal{L}_x + \mathcal{L}_r + \mathcal{L}_z + \mathbf{1}\{t_x > 0.3\}\mathcal{L}_{\text{bt}}] . \tag{22}$$

All four terms have unit coefficients. The auxiliary term complements internal reconstruction by directly supervising distances to the conditioning target.

E Sampling Details

E.1 Modality Schedules

We use the same sampling settings for protein and small-molecule targets. A trajectory contains 1,000 global updates. The backbone and sequence schedules finish after 500 updates, while local atomic refinement continues for all 1,000 updates. Let $k \in \{0, \ldots, 1000\}$ denote the global step and define

$$f(u) = \frac{1 - 10^{-2u}}{1 - 10^{-2}} . \tag{23}$$

The modality times are

$$\begin{aligned} t_x(k) &= f(\min\{k/500, 1\}) , \\ t_r(k) &= \min\{k/500, 1\}^2 , \\ t_z(k) &= f(k/1000) . \end{aligned} \tag{24}$$

Table 7: Sampling settings shared by the two CODesign variants. A dash indicates a parameter that does not apply to that modality.

Setting	Backbone $\mathbf{x}$	Sequence $\mathbf{r}$	Local atoms $\mathbf{z}$
Active updates	500	500	1,000
Time schedule	Exponential	Quadratic	Exponential
Noise-variance factor	0.3	–	0.6
Categorical temperature	–	0.3	–
Discrete stochasticity η	–	5	–
ODE switching time	0.98	–	0.98
Final ODE score multiplier	1.5	–	1.5
Joint settings			
Total updates		1,000	
Self-conditioning		Enabled	
CAR backbone-time threshold		$t_x \geq 0.95$	
CAR check interval		25 global updates	
CAR resampling fraction ρ		0.2	
External reward search / refinement		Disabled	

Each update uses $\Delta t_u = t_u(k+1) - t_u(k)$ for $u \in \{x, r, z\}$. A modality with $\Delta t_u = 0$ is not advanced. The exponential backbone schedule provides informative geometry before the quadratic sequence schedule finishes, allowing CAR to revise residue decisions while sequence generation is still active.

E.2 Coordinate and Sequence Updates

Continuous coordinates. For either coordinate state $\mathbf{y} \in \{\mathbf{x}_{t_x}, \mathbf{z}_{t_z}\}$, let t_y be its current time and $\mathbf{v}_y^\theta$ its predicted vector field. The coordinate sampler uses

$$\mathbf{s}_y = \frac{t_y \mathbf{v}_y^\theta - \mathbf{y}}{1 - t_y}, \qquad g(t_y) = \frac{1}{t_y + 10^{-2}}. \tag{25}$$

For $t_y \leq 0.98$, we take an Euler–Maruyama step,

$$\mathbf{y}^+ = \mathbf{y} + \big(\mathbf{v}_y^\theta + g(t_y)\mathbf{s}_y\big)\Delta t_y + \sqrt{2g(t_y)\nu_y \Delta t_y}\,\boldsymbol{\xi}, \qquad \boldsymbol{\xi} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{26}$$

where $\nu_x = 0.3$ and $\nu_z = 0.6$. For $t_y > 0.98$, we remove the stochastic term and use

$$\mathbf{y}^+ = \mathbf{y} + \frac{\mathbf{y} + 1.5(1 - t_y)\mathbf{s}_y}{t_y}\Delta t_y. \tag{27}$$

These updates are applied only to active modalities. For local atoms, the resulting coordinate proposal is combined with the sequence transition as described below.

Discrete sequence. We sample amino-acid identities from $\pi_i(a) = \mathrm{softmax}(\boldsymbol{\ell}_i^\theta/\tau)_a$ with $\tau = 0.3$. Following purity-based discrete flow sampling (Campbell et al., 2024), we first sample the number of masked positions to reveal:

$$\begin{aligned} p_{\mathrm{reveal}} &= \min\left\{1, \frac{1 + \eta t_r}{\max(1 - t_r, 10^{-5})}\Delta t_r\right\}, \\ N_{\mathrm{reveal}} &\sim \mathrm{Binomial}(N_{\mathrm{mask}}, p_{\mathrm{reveal}}), \qquad \eta = 5. \end{aligned} \tag{28}$$

Here, N_{mask} is the current number of masked residues. We select the N_{reveal} masked positions with the largest $\max_a \pi_i(a)$ and sample their identities categorically. The base sampler also remasks positions with probability $\min\{1, \eta \Delta t_r\}$. Any remaining masks are filled at the final active sequence update. CAR adds targeted corrections to this base sampler rather than replacing its stochastic transitions.

Sequence-dependent atomic states. Let r_i and r_i^+ denote a residue identity immediately before and after a sampling update, and let $M_a(r_i^+)$ denote whether atom slot a exists for the updated identity. Local atoms follow

$$\mathbf{z}_{ia}^+ = \begin{cases} \mathbf{0}, & r_i^+ = \mathtt{m}, \\ M_a(r_i^+)\mathbf{b}_{ia}^\theta, & r_i = \mathtt{m},\ r_i^+ \neq \mathtt{m}, \\ M_a(r_i^+)\mathbf{z}_{ia}^{\mathrm{prop}}, & r_i \neq \mathtt{m},\ r_i^+ \neq \mathtt{m}, \end{cases} \tag{29}$$

where $\mathbf{z}^{\mathrm{prop}}$ is the continuous coordinate proposal. Newly revealed residues therefore receive the learned initialization directly. At the end of sampling, existing non-C_α offsets are added to the final C_α coordinates to reconstruct the all-atom binder.

E.3 Consistency-Aware Joint Resampling

Assessment context. CAR is applied every 25 global updates when $t_x \geq 0.95$ and $\Delta t_r > 0$. A normal denoiser evaluation first supplies $\widehat{\mathbf{x}}$ and the current sequence prediction. The assessment in Equation (14) then masks all sequence identities and hides their non-C_α inputs. Only the assessment sequence time is set to zero; the current noisy C_α coordinates, both coordinate times, and target features are retained. The clean-backbone channel receives $\widehat{\mathbf{x}}$, and the clean-sequence channel is disabled.

The assessment also uses $\widehat{\mathbf{x}}$ for backbone self-conditioning. For sequence self-conditioning, revealed positions retain their current sampled identities, while masked positions retain the normal denoiser's sequence prediction. This provides a structural reassessment without discarding the sequence context of the current trajectory.

Position selection. We use the untempered assessment probabilities $q_i(a)$ from Section 3.3. The preference gap can be calculated directly from logits:

$$\Delta_i = \widetilde{\ell}_{i,a_i^\star}^\theta - \widetilde{\ell}_{i,r_{t_r,i}}^\theta. \tag{30}$$

This is equivalent to Equation (15) because the log-softmax normalizer cancels. Define the eligible set $\mathcal{E} = \{i \in \mathcal{U} : a_i^\star \neq r_{t_r,i}\}$. The number of selected positions and their selection rule are

$$K_{\mathrm{CAR}} = \min\{\lceil \rho|\mathcal{U}| \rceil, |\mathcal{E}|\}, \qquad \mathcal{S} = \mathrm{TopK}_{i\in\mathcal{E}}(\Delta_i, K_{\mathrm{CAR}}), \qquad \rho = 0.2. \tag{31}$$

Thus, the budget is a fraction of revealed positions, and only disagreements are selected.

Correction and continued sampling. We apply the joint reset in Equation (16). The proposed types replace the corresponding entries in sequence self-conditioning built from the normal prediction. The primary sequence is remasked at those positions, and the associated local atomic states are set to zero. Backbone coordinates are not reset. A fresh denoiser evaluation at the original times then updates all four outputs before the next sampling step. If $\mathcal{S}$ is empty, this reevaluation is skipped.

During ordinary sampling, self-conditioning is taken from the preceding clean prediction. Once the sequence schedule has finished, its self-conditioning is fixed to the finalized sequence. Algorithm 1 summarizes the complete trajectory. With the default schedules, CAR has seven assessment opportunities, at global steps 325, 350, 375, 400, 425, 450, and 475. Each assessment adds one forward evaluation, and a nonempty reset adds one more. The maximum additional cost is therefore 14 denoiser evaluations beyond the 1,000 normal evaluations; no external folding or inverse-folding model is called.

F Supplementary Experimental Results

F.1 Reporting Conventions

The following tables supplement the benchmarks in Sections 4.1 and 4.2, and the ablation studies in the main text. We use the evaluation protocols and success criteria defined in the main text. Each benchmark run contains 200 generated binders per target, and the reported quality metrics are summarized across five independent random seeds.

Algorithm 1 CODesign sampling with CAR

Require: Denoiser parameters θ, target $\mathbf{c}^{\text{target}}$, binder length n, modality schedules
Initialize $\mathbf{x}_{t_x}$ from the backbone initial noise, $\mathbf{r}_{t_r} \leftarrow \mathtt{m}^n$, and $\mathbf{z}_{t_z} \leftarrow \mathbf{0}$
Leave self-conditioning empty for the first denoiser evaluation
for $k = 0, \ldots, 999$ **do**
  Set (t_x, t_r, t_z) and their increments using Equation (24)
  Predict $\mathbf{v}_x^\theta, \boldsymbol{\ell}^\theta, \mathbf{v}_z^\theta, \mathbf{b}^\theta$ with the current states and self-conditioning
  Recover $\widehat{\mathbf{x}}$ and predicted residue identities
  if $k \bmod 25 = 0$, $t_x \geq 0.95$, and $\Delta t_r > 0$ **then**
    Build the masked assessment and its self-conditioning as in Appendix E.3
    Compute $\widetilde{\boldsymbol{\ell}}^\theta$, $a_i^\star$, and Δ_i using the same denoiser
    Select $\mathcal{S}$ using Equation (31)
    if $\mathcal{S} \neq \varnothing$ **then**
      Form self-conditioning from the normal prediction and set $r_i^{\text{sc}} \leftarrow a_i^\star$ for $i \in \mathcal{S}$
      Reset $r_{t_r,i} \leftarrow \mathtt{m}$ and $\mathbf{z}_{t_z,i} \leftarrow \mathbf{0}$ for $i \in \mathcal{S}$
      Keep $\mathbf{x}_{t_x}$, the target, and all modality times unchanged
      Reevaluate the denoiser to refresh all four outputs
    end if
  end if
  Store the latest clean backbone and sequence predictions for the next self-conditioning input
  Advance active backbone and sequence states using Appendix E.2
  Update local atoms using Equation (29)
  If sequence generation has finished, fix sequence self-conditioning to the final sampled identities
end for
return Final sequence and existing all-atom coordinates $\mathbf{x}_i$ and $\mathbf{x}_i + \mathbf{z}_{ia}$

For a metric $m_{s,j}$ from seed s on target j, per-target entries report the mean and standard deviation across seeds. For an aggregate over targets, we first compute the target average within each seed and then summarize those five averages. This preserves the across-seed variation of the aggregate; it is not obtained by averaging the per-target standard deviations. Generation times use batch size one on a single GPU.

F.2 Protein-Conditioned Binder Design

Table 8 expands Table 1 with standard deviations for unique successes and structural consistency. To keep the statistics readable, complex scRMSD and the percentage of designs with scRMSD ≤ 2 Å are shown in separate columns. Tables 9–11 give per-target unique successes. Tables 12–17 report per-target consistency using the same metric definitions.

Complex-level consistency. The following tables report consistency for all generated binders, not only successful designs. Mean scRMSD and the fraction satisfying the 2 Å threshold are separated to avoid compressing four statistics into one cell. Both quantities are reported for each sequence type.

F.3 Ligand-Conditioned Binder Design

Table 18 provides the per-target unique successes and four-target mean corresponding to Table 2. Co-design methods use their generated sequences, and the two-stage methods use one LigandMPNN sequence per generated backbone, as in Section 4.2.

F.4 Ablation Studies

Effect of CODIMER. We compare the model trained without CODIMER with the model trained using CODIMER, disabling CAR in both cases. Both variants use the same architecture and training hyperparameters and are evaluated with self-generated sequences on the 19 protein targets. Table 19 gives the aggregate comparison, and Table 20 gives the per-target results corresponding to Figure 3.

Table 8: Protein-target benchmark: aggregate unique successes and complex C_α scRMSD. Entries are means ± standard deviations across five seeds; the threshold column is in percent.

Model	Sequence	Unique successes ↑	scRMSD (Å) ↓	≤ 2 Å (%) ↑
RFDiffusion3	MPNN@1	6.46 ± 0.43	7.24 ± 0.11	36.12 ± 0.95
RFDiffusion3	MPNN@8	10.96 ± 0.31	4.17 ± 0.04	53.02 ± 0.93
PXDesign	MPNN@1	4.59 ± 0.25	5.21 ± 0.10	51.15 ± 0.72
PXDesign	MPNN@8	7.01 ± 0.51	2.29 ± 0.04	74.54 ± 0.47
Protpardelle-1c	MPNN@1	0.28 ± 0.10	16.44 ± 0.29	10.26 ± 0.50
Protpardelle-1c	MPNN@8	0.92 ± 0.24	12.61 ± 0.21	20.40 ± 0.53
Complexa	Self	8.54 ± 0.14	7.91 ± 0.06	35.06 ± 0.53
Complexa	MPNN@1	8.53 ± 0.26	7.50 ± 0.11	35.28 ± 0.75
Complexa	MPNN@8	14.65 ± 0.51	4.49 ± 0.04	51.18 ± 0.41
CODesign	Self	9.84 ± 0.25	7.01 ± 0.05	41.64 ± 0.55
CODesign	MPNN@1	10.32 ± 0.97	7.01 ± 0.12	40.53 ± 0.71
CODesign	MPNN@8	18.81 ± 0.91	4.09 ± 0.07	57.16 ± 0.64

Table 9: Per-target unique successes with Self sequences. Entries report means ± standard deviations across five seeds.

Target	Complexa	CODesign
IFNAR2	35.00 ± 3.61	30.00 ± 4.30
BHRF1	22.00 ± 2.65	43.60 ± 6.15
BBF14	12.80 ± 1.64	8.00 ± 1.58
DerF21	20.60 ± 2.88	36.40 ± 4.51
TrkA	10.00 ± 1.73	2.60 ± 1.34
PD1	13.20 ± 3.96	2.00 ± 0.00
Insulin	12.20 ± 1.48	16.40 ± 3.91
DerF7	10.60 ± 2.30	12.40 ± 3.78
PDL1	7.40 ± 1.52	8.40 ± 2.51
IL7RA	4.60 ± 1.14	7.20 ± 2.05
CrSAS6	6.80 ± 1.48	14.20 ± 2.17
Claudin1	1.80 ± 0.45	2.80 ± 2.77
VEGFA	1.60 ± 1.14	0.60 ± 0.89
SpCas9	2.40 ± 1.14	1.80 ± 1.10
SC2RBD	0.00 ± 0.00	0.00 ± 0.00
CbAgo	0.00 ± 0.00	0.00 ± 0.00
CD45	1.00 ± 1.00	0.40 ± 0.55
BetV1	0.00 ± 0.00	0.20 ± 0.45
HER2_AAV	0.20 ± 0.45	0.00 ± 0.00
Mean	8.54 ± 0.14	9.84 ± 0.25

Table 10: Per-target unique successes with MPNN@1 sequences. Entries report means ± standard deviations across five seeds.

Target	RFDiffusion3	PXDesign	Protpardelle-1c	Complexa	CODesign
IFNAR2	24.20 ± 3.96	11.80 ± 2.95	0.80 ± 0.84	39.20 ± 3.49	41.20 ± 2.39
BHRF1	9.20 ± 1.10	5.00 ± 1.00	1.20 ± 0.84	19.20 ± 1.64	40.60 ± 9.99
BBF14	1.80 ± 1.79	3.60 ± 1.82	0.00 ± 0.00	10.40 ± 2.70	10.40 ± 1.52
DerF21	1.80 ± 0.45	8.00 ± 1.58	2.60 ± 1.14	23.60 ± 1.82	42.80 ± 5.81
TrkA	10.20 ± 1.79	7.80 ± 0.84	0.20 ± 0.45	12.80 ± 2.39	2.40 ± 2.07
PD1	10.60 ± 2.41	6.00 ± 1.87	0.00 ± 0.00	13.80 ± 1.79	2.20 ± 1.64
Insulin	4.20 ± 0.45	3.80 ± 1.30	0.00 ± 0.00	9.60 ± 1.14	9.00 ± 1.87
DerF7	4.80 ± 1.10	9.40 ± 3.51	0.00 ± 0.00	12.00 ± 3.00	16.40 ± 4.51
PDL1	9.60 ± 2.97	5.60 ± 2.70	0.00 ± 0.00	8.80 ± 2.39	7.00 ± 2.55
IL7RA	6.40 ± 2.41	4.40 ± 1.14	0.00 ± 0.00	2.20 ± 1.30	7.60 ± 2.70
CrSAS6	22.00 ± 1.41	6.80 ± 2.28	0.00 ± 0.00	5.20 ± 3.19	12.80 ± 1.92
Claudin1	5.60 ± 2.07	4.80 ± 1.30	0.00 ± 0.00	1.80 ± 1.30	2.80 ± 2.17
VEGFA	1.60 ± 0.55	3.00 ± 0.71	0.00 ± 0.00	0.60 ± 0.89	0.00 ± 0.00
SpCas9	5.40 ± 1.67	1.40 ± 0.55	0.00 ± 0.00	2.40 ± 0.89	0.60 ± 0.89
SC2RBD	3.20 ± 1.48	2.60 ± 1.52	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
CbAgo	0.60 ± 0.89	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
CD45	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
BetV1	1.60 ± 1.14	2.80 ± 1.92	0.20 ± 0.45	0.20 ± 0.45	0.00 ± 0.00
HER2_AAV	0.00 ± 0.00	0.40 ± 0.55	0.40 ± 0.55	0.20 ± 0.45	0.20 ± 0.45
Mean	6.46 ± 0.43	4.59 ± 0.25	0.28 ± 0.10	8.53 ± 0.26	10.32 ± 0.97

Table 11: Per-target unique successes with MPNN@8 sequences. Entries report means ± standard deviations across five seeds.

Target	RFDiffusion3	PXDesign	Protpardelle-1c	Complexa	CODesign
IFNAR2	35.60 ± 4.16	19.00 ± 2.92	2.60 ± 1.14	58.80 ± 2.77	62.60 ± 4.39
BHRF1	11.60 ± 0.55	7.20 ± 1.92	4.80 ± 1.10	23.00 ± 2.55	67.60 ± 6.50
BBF14	4.80 ± 1.30	7.40 ± 2.07	0.20 ± 0.45	19.20 ± 4.09	19.80 ± 3.42
DerF21	5.20 ± 0.84	11.40 ± 4.67	3.40 ± 1.14	34.80 ± 5.45	63.00 ± 5.00
TrkA	14.40 ± 2.88	11.20 ± 2.49	0.80 ± 1.30	20.20 ± 1.64	6.00 ± 2.45
PD1	17.40 ± 2.30	8.20 ± 1.10	0.40 ± 0.89	26.20 ± 2.95	4.00 ± 1.41
Insulin	7.40 ± 1.95	7.00 ± 1.58	0.40 ± 0.55	20.40 ± 3.21	23.40 ± 5.13
DerF7	9.40 ± 2.07	12.60 ± 2.07	0.40 ± 0.55	25.00 ± 4.69	35.00 ± 4.12
PDL1	17.80 ± 3.27	7.80 ± 0.45	1.80 ± 1.64	17.80 ± 3.83	14.20 ± 4.02
IL7RA	10.40 ± 2.19	6.40 ± 2.07	0.00 ± 0.00	4.80 ± 2.28	16.20 ± 2.49
CrSAS6	37.80 ± 2.28	8.20 ± 1.92	0.20 ± 0.45	12.80 ± 1.79	27.20 ± 4.82
Claudin1	8.80 ± 1.64	6.20 ± 0.45	0.60 ± 0.89	5.40 ± 1.95	10.60 ± 1.95
VEGFA	5.40 ± 0.89	5.40 ± 1.14	0.00 ± 0.00	3.60 ± 0.55	1.20 ± 1.10
SpCas9	10.80 ± 1.48	2.40 ± 0.55	0.60 ± 0.55	4.60 ± 1.67	3.80 ± 1.48
SC2RBD	7.00 ± 1.41	6.00 ± 1.00	0.20 ± 0.45	0.20 ± 0.45	0.20 ± 0.45
CbAgo	0.60 ± 0.89	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
CD45	0.40 ± 0.89	0.40 ± 0.55	0.00 ± 0.00	0.00 ± 0.00	0.20 ± 0.45
BetV1	3.20 ± 1.48	4.00 ± 1.41	0.60 ± 0.55	0.60 ± 0.89	2.00 ± 1.00
HER2_AAV	0.20 ± 0.45	2.40 ± 1.14	0.40 ± 0.55	1.00 ± 0.71	0.40 ± 0.55
Mean	10.96 ± 0.31	7.01 ± 0.51	0.92 ± 0.24	14.65 ± 0.51	18.81 ± 0.91

Table 12: Per-target mean complex C_α scRMSD with Self sequences. Values are in Å; lower is better. Entries report means ± standard deviations across five seeds.

Target	Complexa	CODesign
IFNAR2	4.26 ± 0.25	2.17 ± 0.24
BHRF1	1.44 ± 0.16	0.90 ± 0.08
BBF14	9.36 ± 0.42	8.71 ± 0.48
DerF21	3.53 ± 0.14	4.26 ± 0.24
TrkA	4.42 ± 0.20	4.73 ± 0.35
PD1	5.08 ± 0.37	4.66 ± 0.50
Insulin	3.58 ± 0.23	3.64 ± 0.07
DerF7	4.67 ± 0.27	4.61 ± 0.33
PDL1	3.83 ± 0.43	3.81 ± 0.24
IL7RA	6.34 ± 0.33	4.46 ± 0.16
CrSAS6	10.43 ± 0.26	8.59 ± 0.50
Claudin1	5.18 ± 0.34	4.03 ± 0.20
VEGFA	8.86 ± 0.41	5.64 ± 0.41
SpCas9	9.48 ± 0.27	10.27 ± 0.81
SC2RBD	10.90 ± 0.10	13.97 ± 0.23
CbAgo	14.35 ± 0.19	12.11 ± 0.18
CD45	12.53 ± 0.31	10.39 ± 0.45
BetV1	19.27 ± 0.36	16.15 ± 0.34
HER2_AAV	12.77 ± 0.40	10.21 ± 0.11
Mean	7.91 ± 0.06	7.01 ± 0.05

Table 13: Per-target mean complex C_α scRMSD with MPNN@1 sequences. Values are in Å; lower is better. Entries report means ± standard deviations across five seeds.

Target	RFDiffusion3	PXDesign	Protpardelle-1c	Complexa	CODesign
IFNAR2	2.56 ± 0.21	4.72 ± 0.36	5.39 ± 0.34	4.50 ± 0.49	2.17 ± 0.21
BHRF1	2.63 ± 0.29	2.18 ± 0.19	7.82 ± 0.45	1.22 ± 0.15	0.87 ± 0.10
BBF14	10.16 ± 0.28	7.59 ± 0.44	45.26 ± 2.20	7.37 ± 0.20	6.33 ± 0.37
DerF21	10.32 ± 0.24	3.48 ± 0.11	19.21 ± 2.51	2.85 ± 0.31	4.92 ± 0.40
TrkA	2.81 ± 0.23	2.20 ± 0.10	7.80 ± 0.33	3.77 ± 0.27	4.38 ± 0.32
PD1	3.57 ± 0.35	4.46 ± 0.65	7.81 ± 0.15	3.79 ± 0.15	4.66 ± 0.55
Insulin	4.86 ± 0.27	3.80 ± 0.28	8.74 ± 0.23	3.57 ± 0.25	4.11 ± 0.27
DerF7	9.24 ± 0.57	2.77 ± 0.26	18.56 ± 0.70	3.95 ± 0.32	4.00 ± 0.40
PDL1	3.00 ± 0.33	1.90 ± 0.29	6.63 ± 0.15	3.77 ± 0.43	3.46 ± 0.18
IL7RA	3.88 ± 0.32	2.11 ± 0.22	14.23 ± 0.47	4.61 ± 0.18	4.45 ± 0.27
CrSAS6	4.74 ± 0.32	6.21 ± 0.91	43.36 ± 1.29	11.44 ± 0.40	9.06 ± 0.39
Claudin1	3.09 ± 0.21	6.43 ± 0.34	12.56 ± 0.26	4.71 ± 0.42	3.87 ± 0.24
VEGFA	8.34 ± 0.64	5.77 ± 0.37	16.99 ± 1.23	8.92 ± 0.77	6.53 ± 0.44
SpCas9	7.28 ± 0.51	2.73 ± 0.17	9.65 ± 0.42	11.94 ± 0.46	13.62 ± 0.36
SC2RBD	6.93 ± 0.57	5.61 ± 0.17	9.89 ± 0.39	9.45 ± 0.12	9.34 ± 0.15
CbAgo	14.33 ± 0.29	8.66 ± 0.53	14.78 ± 0.43	15.46 ± 0.36	12.78 ± 0.37
CD45	13.55 ± 0.32	11.89 ± 0.40	26.76 ± 0.90	14.16 ± 0.17	14.54 ± 0.24
BetV1	12.31 ± 0.32	8.95 ± 0.29	18.19 ± 0.64	15.81 ± 0.31	15.37 ± 0.42
HER2_AAV	13.98 ± 0.23	7.62 ± 0.58	18.73 ± 0.46	11.15 ± 0.48	8.70 ± 0.36
Mean	7.24 ± 0.11	5.21 ± 0.10	16.44 ± 0.29	7.50 ± 0.11	7.01 ± 0.12

Table 14: Per-target mean complex C_α scRMSD with MPNN@8 sequences. Values are in Å; lower is better. Entries report means ± standard deviations across five seeds.

Target	RFDiffusion3	PXDesign	Protpardelle-1c	Complexa	CODesign
IFNAR2	1.09 ± 0.05	1.62 ± 0.24	2.55 ± 0.18	1.91 ± 0.20	1.06 ± 0.05
BHRF1	1.31 ± 0.13	1.15 ± 0.13	4.83 ± 0.30	0.82 ± 0.03	0.57 ± 0.02
BBF14	5.97 ± 0.30	4.07 ± 0.39	42.49 ± 2.47	3.56 ± 0.16	3.14 ± 0.27
DerF21	7.76 ± 0.22	1.32 ± 0.10	13.80 ± 2.35	1.53 ± 0.08	2.40 ± 0.25
TrkA	1.23 ± 0.09	0.86 ± 0.08	4.47 ± 0.13	1.83 ± 0.08	2.25 ± 0.27
PD1	1.47 ± 0.13	1.62 ± 0.11	4.42 ± 0.31	1.58 ± 0.17	1.91 ± 0.17
Insulin	1.82 ± 0.18	1.12 ± 0.09	4.52 ± 0.24	1.67 ± 0.09	1.68 ± 0.21
DerF7	4.27 ± 0.17	1.17 ± 0.11	15.07 ± 0.83	2.07 ± 0.20	1.87 ± 0.21
PDL1	1.15 ± 0.09	0.71 ± 0.03	3.73 ± 0.25	1.75 ± 0.23	1.36 ± 0.06
IL7RA	1.83 ± 0.09	0.95 ± 0.04	10.60 ± 0.27	2.17 ± 0.16	2.17 ± 0.17
CrSAS6	1.55 ± 0.22	2.01 ± 0.32	40.11 ± 1.22	6.74 ± 0.25	4.55 ± 0.23
Claudin1	1.03 ± 0.05	2.12 ± 0.22	7.25 ± 0.20	1.93 ± 0.14	1.51 ± 0.14
VEGFA	2.96 ± 0.29	1.20 ± 0.08	10.14 ± 1.37	3.81 ± 0.21	2.13 ± 0.33
SpCas9	3.30 ± 0.30	1.20 ± 0.04	5.99 ± 0.40	7.07 ± 0.39	8.37 ± 0.05
SC2RBD	2.78 ± 0.18	2.57 ± 0.15	5.12 ± 0.22	5.88 ± 0.14	4.99 ± 0.15
CbAgo	11.11 ± 0.38	3.59 ± 0.16	11.70 ± 0.37	12.52 ± 0.38	10.08 ± 0.28
CD45	10.07 ± 0.18	8.85 ± 0.25	23.51 ± 0.88	10.89 ± 0.21	11.59 ± 0.29
BetV1	7.98 ± 0.22	4.19 ± 0.32	15.46 ± 0.65	10.84 ± 0.49	11.18 ± 0.53
HER2_AAV	10.61 ± 0.41	3.19 ± 0.16	13.78 ± 0.62	6.71 ± 0.42	4.89 ± 0.34
Mean	4.17 ± 0.04	2.29 ± 0.04	12.61 ± 0.21	4.49 ± 0.04	4.09 ± 0.07

Table 15: Per-target percentage of designs with complex C_α scRMSD ≤ 2 Å with Self sequences. Values are percentages; higher is better. Entries report means ± standard deviations across five seeds.

Target	Complexa	CODesign
IFNAR2	58.10 ± 1.08	79.20 ± 2.95
BHRF1	94.00 ± 1.27	96.90 ± 1.14
BBF14	26.50 ± 2.78	31.10 ± 3.96
DerF21	42.10 ± 2.58	69.20 ± 3.05
TrkA	48.70 ± 2.75	34.50 ± 3.48
PD1	45.60 ± 4.36	46.10 ± 6.05
Insulin	59.50 ± 2.67	67.00 ± 2.52
DerF7	54.80 ± 2.20	53.10 ± 2.04
PDL1	53.70 ± 4.02	49.00 ± 2.94
IL7RA	32.00 ± 4.72	57.80 ± 1.15
CrSAS6	22.30 ± 1.68	33.80 ± 3.05
Claudin1	47.40 ± 3.09	56.40 ± 1.14
VEGFA	33.60 ± 1.56	50.40 ± 2.04
SpCas9	28.50 ± 1.87	22.30 ± 4.31
SC2RBD	0.50 ± 0.50	0.00 ± 0.00
CbAgo	0.20 ± 0.27	0.00 ± 0.00
CD45	1.90 ± 0.74	9.40 ± 2.27
BetV1	0.30 ± 0.27	2.50 ± 1.00
HER2_AAV	16.50 ± 2.60	32.50 ± 1.80
Mean	35.06 ± 0.53	41.64 ± 0.55

Table 16: Per-target percentage of designs with complex C_α scRMSD $\leq$ 2 Å with MPNN@1 sequences. Values are percentages; higher is better. Entries report means $\pm$ standard deviations across five seeds.

Target	RFDiffusion3	PXDesign	Protpardelle-1c	Complexa	CODesign
IFNAR2	80.90 ± 3.19	57.60 ± 2.90	40.30 ± 2.08	57.90 ± 2.41	80.20 ± 2.51
BHRF1	78.80 ± 3.63	83.90 ± 2.22	25.80 ± 3.33	94.60 ± 1.08	97.10 ± 1.47
BBF14	4.80 ± 1.68	20.30 ± 3.07	0.10 ± 0.22	27.30 ± 1.44	37.60 ± 3.17
DerF21	0.80 ± 0.57	68.60 ± 1.56	8.40 ± 2.16	57.50 ± 3.66	66.00 ± 2.32
TrkA	65.70 ± 2.51	75.10 ± 2.10	14.90 ± 3.29	54.50 ± 3.71	44.70 ± 2.68
PD1	58.10 ± 2.92	54.10 ± 1.92	15.50 ± 1.62	53.20 ± 3.09	32.20 ± 3.65
Insulin	43.10 ± 1.78	60.10 ± 3.34	6.80 ± 1.60	56.50 ± 3.62	59.90 ± 3.44
DerF7	20.50 ± 3.52	73.00 ± 3.86	3.60 ± 1.56	57.50 ± 3.45	61.90 ± 2.97
PDL1	64.80 ± 3.25	81.50 ± 4.40	13.70 ± 2.59	45.60 ± 4.25	50.80 ± 2.17
IL7RA	40.50 ± 3.08	73.50 ± 3.16	1.00 ± 0.50	33.80 ± 1.68	53.00 ± 4.51
CrSAS6	63.90 ± 2.66	44.90 ± 6.78	0.50 ± 0.61	16.50 ± 2.26	29.60 ± 3.80
Claudin1	71.80 ± 1.60	47.80 ± 2.97	7.00 ± 1.46	49.30 ± 4.80	60.70 ± 3.70
VEGFA	21.20 ± 4.45	50.70 ± 2.80	1.30 ± 0.57	27.80 ± 4.06	42.60 ± 1.92
SpCas9	50.60 ± 3.70	76.00 ± 3.04	33.80 ± 2.49	14.70 ± 1.75	8.10 ± 1.75
SC2RBD	17.90 ± 4.55	27.00 ± 1.80	5.30 ± 2.44	1.50 ± 1.41	6.10 ± 1.88
CbAgo	1.30 ± 0.76	33.70 ± 4.40	2.00 ± 1.32	0.00 ± 0.00	0.00 ± 0.00
CD45	0.40 ± 0.55	0.80 ± 0.76	6.10 ± 1.29	0.10 ± 0.22	0.10 ± 0.22
BetV1	1.20 ± 0.97	10.80 ± 1.52	2.20 ± 1.04	0.40 ± 0.22	1.90 ± 1.64
HER2_AAV	0.00 ± 0.00	32.50 ± 2.09	6.70 ± 1.35	21.70 ± 4.74	37.60 ± 1.75
Mean	36.12 ± 0.95	51.15 ± 0.72	10.26 ± 0.50	35.28 ± 0.75	40.53 ± 0.71

Table 17: Per-target percentage of designs with complex C_α scRMSD $\leq$ 2 Å with MPNN@8 sequences. Values are percentages; higher is better. Entries report means $\pm$ standard deviations across five seeds.

Target	RFDiffusion3	PXDesign	Protpardelle-1c	Complexa	CODesign
IFNAR2	92.30 ± 0.76	83.80 ± 3.37	65.00 ± 3.59	80.60 ± 2.38	90.00 ± 1.90
BHRF1	88.30 ± 2.33	89.50 ± 2.21	41.20 ± 4.91	96.50 ± 0.50	99.70 ± 0.45
BBF14	13.30 ± 2.84	39.40 ± 6.66	0.30 ± 0.27	46.90 ± 2.51	60.30 ± 3.55
DerF21	2.80 ± 2.08	90.50 ± 1.00	21.70 ± 3.58	81.50 ± 2.47	80.90 ± 2.41
TrkA	86.80 ± 2.71	94.90 ± 1.14	30.00 ± 3.43	75.20 ± 2.33	66.50 ± 5.26
PD1	83.20 ± 2.20	84.00 ± 1.22	30.80 ± 3.27	77.50 ± 4.56	63.90 ± 4.17
Insulin	73.90 ± 3.23	87.30 ± 2.44	21.90 ± 1.92	76.70 ± 2.08	82.30 ± 2.39
DerF7	50.10 ± 2.04	91.70 ± 2.61	8.10 ± 2.77	73.00 ± 2.42	79.50 ± 3.10
PDL1	87.80 ± 0.91	98.00 ± 0.79	32.90 ± 4.93	72.60 ± 6.77	79.60 ± 2.25
IL7RA	71.10 ± 2.30	97.60 ± 1.78	1.90 ± 0.89	62.50 ± 4.14	74.30 ± 0.91
CrSAS6	86.80 ± 3.01	78.10 ± 3.78	1.30 ± 0.91	31.30 ± 3.23	53.40 ± 3.03
Claudin1	93.60 ± 0.89	80.40 ± 3.15	19.90 ± 3.54	74.60 ± 2.77	82.30 ± 3.05
VEGFA	45.60 ± 5.89	87.00 ± 1.80	5.00 ± 1.27	50.50 ± 2.94	70.80 ± 4.42
SpCas9	74.30 ± 1.86	94.40 ± 0.82	52.80 ± 2.39	26.70 ± 3.01	18.50 ± 1.84
SC2RBD	45.10 ± 4.20	57.60 ± 2.82	18.60 ± 2.75	6.80 ± 1.92	21.10 ± 0.96
CbAgo	3.50 ± 2.00	61.40 ± 2.79	7.10 ± 2.43	0.10 ± 0.22	0.20 ± 0.27
CD45	2.50 ± 0.94	3.60 ± 1.02	9.80 ± 1.72	0.60 ± 0.22	1.00 ± 1.17
BetV1	6.10 ± 2.58	32.80 ± 3.21	3.80 ± 2.02	1.90 ± 0.89	8.60 ± 2.86
HER2_AAV	0.20 ± 0.27	64.30 ± 3.65	15.50 ± 2.37	37.00 ± 4.40	53.10 ± 2.77
Mean	53.02 ± 0.93	74.54 ± 0.47	20.40 ± 0.53	51.18 ± 0.41	57.16 ± 0.64

Table 18: Ligand-target unique successes. Entries report means ± standard deviations across five seeds. The final column averages the four targets within each seed.

Model	SAM	OQO	FAD	IAI	Mean
RFdiffusionAA	2.0 ± 1.4	3.4 ± 1.3	1.6 ± 1.1	5.0 ± 2.5	3.0 ± 0.4
RFDiffusion3	9.4 ± 3.2	14.8 ± 2.4	20.8 ± 2.6	11.0 ± 1.2	14.0 ± 1.1
Complexa	3.2 ± 1.5	3.0 ± 1.2	5.8 ± 2.9	8.4 ± 1.7	5.1 ± 0.7
Pallatom-Ligand	0.2 ± 0.4	4.4 ± 1.1	2.0 ± 1.0	3.2 ± 1.3	2.5 ± 0.5
DISCO	10.6 ± 3.1	14.0 ± 2.5	25.4 ± 3.8	12.8 ± 0.8	15.7 ± 1.0
CODesign	5.8 ± 1.6	37.2 ± 3.8	21.4 ± 2.3	18.6 ± 2.7	20.8 ± 1.3

For the distributions in that figure, the top 100 values of each metric are selected independently on each target: the lowest values for ipAE and scRMSD, and the highest values for pLDDT. The selected samples can therefore differ between metrics.

Effect of CAR. We use the same CODIMER-trained checkpoint with CAR enabled or disabled. All other generation and evaluation settings are unchanged. The comparison therefore measures the effect of inference-time consistency correction rather than a change in training. Table 19 also reports the mean unique successes and timing values shown in Figure 4. Quality metrics are summarized across five seeds.

Table 19: Controlled ablations on protein targets using Self sequences. Quality entries report means ± standard deviations across five seeds; times correspond to Figure 4. A dash indicates that generation time was not evaluated.

Variant	CODIMER	CAR	Unique successes ↑	Time (s)
Without CODIMER, without CAR	No	No	5.54 ± 0.48	–
Without CAR	Yes	No	9.46 ± 0.65	27.96
CODesign	Yes	Yes	9.84 ± 0.25	28.41

Table 20: Per-target unique successes in the CODIMER and CAR ablations. All variants use Self sequences. Entries report means ± standard deviations across five seeds.

Target	Without CODIMER without CAR	With CODIMER without CAR	CODesign
IFNAR2	13.60 ± 1.14	30.20 ± 7.12	30.00 ± 4.30
BHRF1	20.60 ± 6.27	43.00 ± 4.85	43.60 ± 6.15
BBF14	1.00 ± 1.00	7.40 ± 1.52	8.00 ± 1.58
DerF21	23.40 ± 3.91	34.20 ± 3.27	36.40 ± 4.51
TrkA	3.80 ± 0.84	2.40 ± 0.55	2.60 ± 1.34
PD1	4.40 ± 1.34	1.40 ± 0.55	2.00 ± 0.00
Insulin	17.00 ± 2.00	15.40 ± 1.67	16.40 ± 3.91
DerF7	0.60 ± 0.55	12.80 ± 2.68	12.40 ± 3.78
PDL1	7.40 ± 1.14	8.60 ± 1.14	8.40 ± 2.51
IL7RA	5.40 ± 1.14	8.60 ± 1.52	7.20 ± 2.05
CrSAS6	6.20 ± 1.92	11.20 ± 2.59	14.20 ± 2.17
Claudin1	0.20 ± 0.45	2.00 ± 1.58	2.80 ± 2.77
VEGFA	0.40 ± 0.55	0.40 ± 0.55	0.60 ± 0.89
SpCas9	0.80 ± 1.30	0.80 ± 0.84	1.80 ± 1.10
SC2RBD	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
CbAgo	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
CD45	0.20 ± 0.45	0.60 ± 1.34	0.40 ± 0.55
BetV1	0.20 ± 0.45	0.60 ± 0.55	0.20 ± 0.45
HER2_AAV	0.00 ± 0.00	0.20 ± 0.45	0.00 ± 0.00
Mean	5.54 ± 0.48	9.46 ± 0.65	9.84 ± 0.25