COVATIDE: COVALENT TOPOLOGY-CONDITIONED ALL-ATOM FLOW MATCHING FOR LINEAR AND CYCLIC PEPTIDE DESIGN

Anonymous authors
Paper under double-blind review

ABSTRACT

Peptides offer a versatile therapeutic modality for targeting protein surfaces, but conformational flexibility and susceptibility to proteolysis can limit their utility. Cyclization can mitigate these limitations, yet cyclic peptide design requires coordinating receptor-bound conformations with the local geometry of covalent closure. Existing methods often rely on post-generation filtering and endpoint mutation to obtain cyclic peptides, decoupling closure constraints from the generation of receptor-bound conformations. We present CovaTide, a receptor-conditioned all-atom flow model that jointly generates peptide sequences and structures under a prescribed covalent topology. CovaTide uses a deep topology adapter to repeatedly integrate residue-level connectivity and atom-level bond information, coupling global peptide conformations with local closure geometry throughout generation. This enables a single model to design linear peptides as well as head-to-tail, disulfide, and isopeptide cyclic peptides without post-generation endpoint mutation. To rank generated candidates, we further introduce a generation-aware confidence model that estimates peptide and interface errors. CovaTide outperforms the evaluated baselines in sequence and binding pose recovery for linear peptides and in cyclization, bond, and energy-based success rates for cyclic peptides. Together, these results support a unified formulation of linear and cyclic peptide binder design through multiscale covalent topology conditioning.

1 INTRODUCTION

Peptides are an attractive therapeutic modality, as they can be tailored to recognize protein surfaces, including interfaces that are difficult to target with small molecules (Wang et al., 2022; Vinogradov et al., 2019). Compared with larger protein therapeutics such as antibodies, peptides can also offer lower immunogenicity (Wang et al., 2022). For many linear peptides, however, conformational flexibility and susceptibility to proteolysis can limit their therapeutic utility. Cyclization can mitigate these limitations by introducing covalent linkages that restrict flexibility, stabilize binding-competent conformations, and improve proteolytic resistance (Wang et al., 2022; Ji et al., 2024). The challenge is to satisfy cyclization constraints while achieving favorable receptor interactions. Head-to-tail amide, disulfide, and side-chain isopeptide linkages impose distinct sequence and geometric requirements that must be satisfied in the receptor-bound conformation (Figure 1a). Cyclic peptide design therefore requires jointly optimizing peptide sequences and receptor interactions under the constraints imposed by the cyclization topology.

Generative models have advanced receptor-conditioned peptide design, but their treatment of cyclization remains limited. Some methods specialize in head-to-tail macrocycles (Rettie et al., 2025a;b) or disulfide peptides (Wang et al., 2024). Other methods obtain cyclic candidates through geometric guidance or alignment, downstream screening and relaxation, or post-generation endpoint mutation (Yang et al., 2025; Jiang et al., 2025; Zhang et al., 2026). Such guidance and post-processing can facilitate cyclization, but the resulting structures may still fail to satisfy the prescribed closure geometry. None of these methods unifies the design of linear peptides and multiple types of cyclic peptides in a receptor-conditioned all-atom model that explicitly encodes prescribed endpoint roles and connectivity at both residue and atom levels throughout generation. These limitations motivate a unified treatment of covalent topology.

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

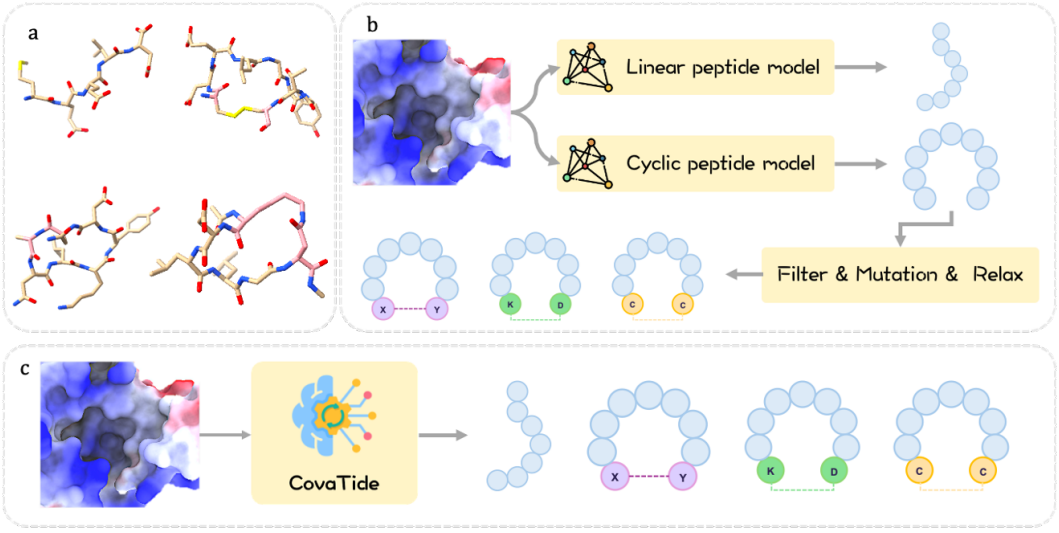

Figure 1: **Peptide design with prescribed covalent topology.** (a) Representative linear and cyclic peptide structures, covering head-to-tail, disulfide, and isopeptide linkages. (b) Two representative prior strategies: separate linear and cyclic generators, and a post hoc workflow with filtering, endpoint mutation, and relaxation. (c) CovaTide uses one generator conditioned on the requested topology and linkage endpoints.

To address this gap, we introduce CovaTide, a receptor-conditioned all-atom flow model that jointly generates peptide sequences and structures under a prescribed covalent topology and fixed endpoint residue identities (Figures 1c and 2). Its deep topology adapter repeatedly integrates residue-level endpoint roles and closure graph distances with atom-level bond information, coupling global peptide conformations with local closure geometry throughout generation. This enables a single model to design linear peptides and head-to-tail, disulfide, and isopeptide cyclic peptides without post-generation topology filtering or endpoint mutation. To rank generated candidates, we separately train a generation-aware confidence model that estimates peptide and interface errors. We evaluate CovaTide on receptor-conditioned de novo design of linear and cyclic peptides, assessing both binding pose quality and compatibility with the prescribed covalent topology. CovaTide achieves the best sequence and binding pose recovery for linear peptides and the highest cyclization, bond, and energy-based success rates for cyclic peptides among the evaluated models.

Our main contributions are:

1. **Unified linear and cyclic peptide design.** We propose a receptor-conditioned all-atom flow matching model that jointly generates peptide sequences and structures under a prescribed covalent topology and fixed endpoint residue identities. This formulation enables a single model to design both linear peptides and cyclic peptides with head-to-tail, disulfide, or isopeptide linkages.
2. **Multiscale covalent topology conditioning.** We develop a deep topology adapter that repeatedly integrates residue-level endpoint roles and closure graph distances with atom-level bond information, coupling global peptide conformations with local closure geometry throughout generation.
3. **Generalization to internal cyclization.** We demonstrate that CovaTide extends from terminal closures in the training data to prescribed internal disulfide and isopeptide linkages with favorable closure geometry and Rosetta interface energies, supporting generalization across cyclization positions.
4. **Generation-aware confidence ranking.** To rank generated candidates, we introduce a generation-aware confidence model that predicts structural and interface errors from final all-atom structures and generator representations. We validate the utility of predicted iPDE for within-target pose selection in linear peptide docking experiments.

2 Related Work

2.1 All-atom and receptor-conditioned peptide generation

Protein structure prediction and design models provide a computational foundation for peptide generation. AlphaFold 2 and AlphaFold 3 link residue-level representations to all-atom geometry for structure prediction (Jumper et al., 2021; Abramson et al., 2024). For protein design, RFdiffusion and FrameDiff generate backbones through geometric diffusion, while ProteinMPNN designs sequences for specified backbones (Watson et al., 2023; Yim et al., 2023; Dauparas et al., 2022). Flow matching provides a continuous vector-field formulation for generation, and MultiFlow extends this framework to jointly model discrete sequences and continuous structures (Lipman et al., 2023; Campbell et al., 2024). These advances support joint sequence-structure modeling but do not directly address receptor-conditioned peptide generation under a prescribed covalent topology.

Peptide-focused generative models directly incorporate side-chain geometry and receptor interactions into the generation process. PepGLAD generates all-atom peptide structures using geometric latent diffusion, while PepFlow jointly models backbone frames, side-chain torsions, and residue identities through multiple flows (Kong et al., 2024; Li et al., 2024). Extending beyond peptides, UniMoMo uses a shared block representation for peptide, antibody, and small-molecule binder design (Kong et al., 2025). Together, these methods provide strong baselines for receptor-conditioned linear peptide generation.

2.2 Cyclic peptide structure and design

Cyclic relative positional encodings have been used to adapt pretrained protein models for head-to-tail macrocycle design. AfCycDesign modifies the positional inputs of AlphaFold 2 to encode the terminal peptide bond, supporting cyclic peptide structure prediction, sequence redesign, and de novo hallucination (Rettie et al., 2025a). RFpeptides introduces a cyclic offset into RFdiffusion to generate macrocyclic backbones, followed by ProteinMPNN sequence design and filtering based on structure prediction, Rosetta scores, and interface quality (Rettie et al., 2025b). Both approaches demonstrate the value of explicitly encoding terminal connectivity in pretrained protein models, but focus only on head-to-tail cyclization.

Other work addresses cyclic peptide design through post-generation cyclization and closure guidance. CPSea constructs a large cyclic peptide-protein complex dataset and uses endpoint geometry to filter and cyclize candidates (Yang et al., 2025), while DiffPepBuilder introduces disulfide bonds through post hoc cyclization (Wang et al., 2024). CP-Composer applies a geometric latent diffusion model trained on linear peptide data to zero-shot cyclic peptide design by composing residue-type and pairwise-distance constraints with classifier-free guidance (Jiang et al., 2025). GeoCycler extends CP-Composer's architecture with reward-weighted alignment across cyclization types, retaining classifier-free guidance (Zhang et al., 2026). Both methods improve cyclization success but may fail to preserve prescribed endpoint identities or satisfy closure geometry. Their workflows also include geometric screening and structural relaxation. CpSDE explicitly models atoms and covalent bonds but relies on alternating structural denoising and residue prediction with separately trained models (Zhou et al., 2025). CovaTide keeps prescribed endpoint identities fixed and repeatedly integrates residue-level endpoint roles and closure graph distances with atom-level closure edges in a single all-atom generator. It predicts sequences from final generator representations without alternating between separate structure and sequence models.

3 Method

Problem formulation and molecular representation. We formulate peptide design conditioned on a fixed receptor pocket $\mathcal{R}$, peptide length n, and prescribed covalent topology $\mathcal{T} = (\tau, i, j)$. Here, τ specifies a linear, head-to-tail, disulfide, or isopeptide design, and i, j identify the residues forming the prescribed closure. These endpoint indices are omitted for linear peptides. For disulfide and isopeptide designs, the endpoint residue identities are fixed to Cys/Cys and Lys/Asp, respectively. We denote the constrained positions by $\mathcal{I}_{\mathcal{T}}$ and their required identities by $c_{\mathcal{T}}$. The complementary set $\overline{\mathcal{I}}_{\mathcal{T}}$ contains positions whose identities remain free to generate. Let S denote the peptide sequence and $X \in \mathbb{R}^{n\times 14\times 3}$ its atomic coordinates, represented using an Atom14-style fixed-slot

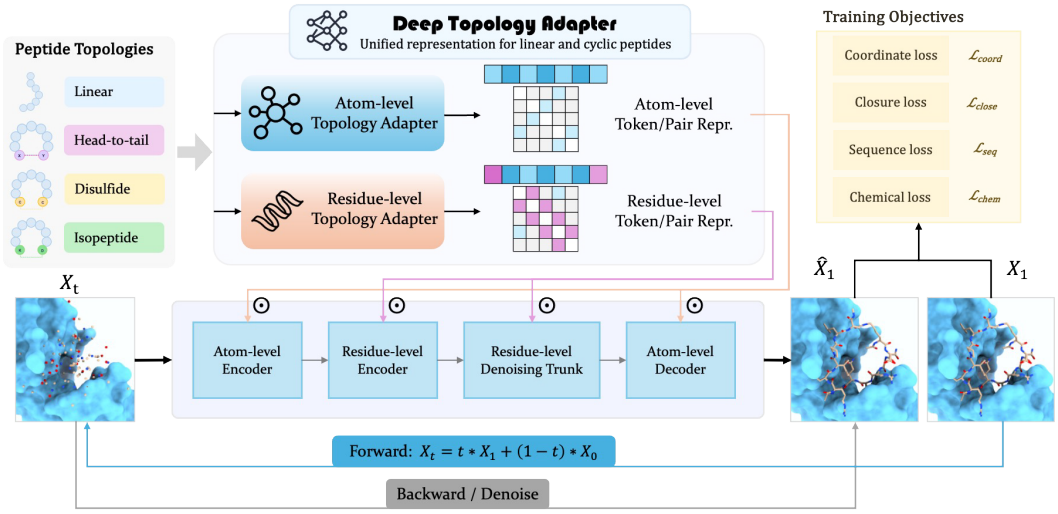

Figure 2: **CovaTide architecture.** Upper: the deep topology adapter maps a prescribed connection into residue and atom single/pair conditions. Lower: these conditions are injected at multiple layers throughout atom encoding, residue encoding, denoising trunk, and atom decoding.

representation (Jumper et al., 2021). With parameters θ, CovaTide models:

$$p_\theta\left(S_{\overline{\mathcal{I}}_\mathcal{T}}, X \mid \mathcal{R}, n, \mathcal{T}, S_{\mathcal{I}_\mathcal{T}} = c_\mathcal{T}\right), \tag{1}$$

jointly generating peptide coordinates and free residue identities with the receptor fixed. Residue identities map the fixed slots to physical atoms or padding and determine the atom mask $M(S)$.

3.1 Deep covalent topology adapter

Global residue connectivity. The prescribed topology $\mathcal{T}$ supplies two residue-level conditions: endpoint roles and closure graph distances. For an arbitrary residue u, the role code r_u identifies its role in the closure and is zero for non-endpoint residues. Adding the prescribed endpoint edge (i, j) to the peptide backbone gives the residue graph $G_\mathcal{T}$. For any two residues u, v in a cyclic peptide, their shortest-path distance $d_{G_\mathcal{T}}(u, v)$ is encoded as $\delta_{uv} = 1 + \min\{d_{G_\mathcal{T}}(u, v), 32\}$. Residue pairs in linear peptides and pairs involving the receptor use $\delta_{uv} = 0$. Learned lookup tables E_{role}, E_{dist}, and E_{type} embed the role, distance code, and linkage type τ, respectively, producing the single-residue condition e_u and residue-pair condition b_{uv}:

$$e_u = E_{\text{role}}(r_u), \qquad b_{uv} = E_{\text{dist}}(\delta_{uv}) + \mathbf{1}[\delta_{uv} > 0] E_{\text{type}}(\tau). \tag{2}$$

Here, $\mathbf{1}[\cdot]$ denotes the indicator function. The topology embedding tables are shared across layers, with index 0 reserved for the absence of a topology condition and assigned a fixed zero embedding. Standard sequence-relative positional encoding is retained for both linear and cyclic peptides.

Repeated residue-level injection. The adapter injects these conditions at multiple depths of the pairformer (residue encoder) and flow transformer (denoising trunk) stages. At site ℓ in stage $B \in \{\text{pairformer}, \text{flow}\}$, the current residue and residue-pair features are denoted by s_u and z_{uv}. The learned projection matrices W_s^B and W_z map e_u and b_{uv} into the corresponding feature spaces. A learned scalar gate g_ℓ^B controls both updates:

$$s_u \leftarrow s_u + g_\ell^B W_s^B e_u, \qquad z_{uv} \leftarrow z_{uv} + g_\ell^B W_z b_{uv}. \tag{3}$$

These projections are bias-free and shared across layers within each stage. The single-feature projection W_s^B differs between stages to accommodate their different feature dimensions, while the pair projection W_z is shared by both stages. Each layer has its own learned scalar gate. Injection occurs after each pairformer block and before each flow transformer block.

Local atom-level chemistry. The residue-level endpoint role condition is propagated to atom slot a through $e_{\pi(a)}$, where $\pi(a)$ identifies the residue. The prescribed endpoint residues i, j determine the bond-forming atom slots a_i, a_j. For a cyclic peptide, these slots define the closure edge set $\mathcal{E}_\mathcal{T} = \{(a_i, a_j)\}$. For a linear peptide, $\mathcal{E}_\mathcal{T} = \varnothing$. The symmetric closure edge indicator is defined as:

$$\chi^{\mathcal{T}}_{ab} = \sum_{(a_i,a_j)\in\mathcal{E}_\mathcal{T}} \mathbf{1}[\{a, b\} = \{a_i, a_j\}]. \tag{4}$$

A separate learned table E_{bond} embeds the linkage type τ at the atom level. The shared context comprises single-atom features c_a and atom-pair features p_{ab}. Bias-free linear projections W_c and W_p map the role and bond embeddings into these two context spaces, while a scalar gate g^{atom}_ℓ controls their injection at atom block ℓ:

$$\tilde{c}^\ell_a = c_a + g^{\text{atom}}_\ell W_c e_{\pi(a)}, \qquad \tilde{p}^\ell_{ab} = p_{ab} + g^{\text{atom}}_\ell W_p \chi^{\mathcal{T}}_{ab} E_{\text{bond}}(\tau). \tag{5}$$

The conditioned contexts $\tilde{c}^\ell_a, \tilde{p}^\ell_{ab}$ are recomputed from the shared base features c_a, p_{ab} at each block, without accumulating updates across blocks. The projections are shared across atom encoder and decoder blocks, with an independently learned gate at each conditioning site. Both edge orientations receive the same update. Sparse atom attention retains each closure partner in the other's neighbor list so that pairwise updates are applied to the prescribed closure edge (Figure 2).

3.2 Coordinate flow and sequence readout

Following flow matching (Lipman et al., 2023), the peptide coordinate tensor follows a trajectory X_t for $t \in [0, 1]$, interpolating between a Gaussian sample X_0 and the clean coordinates X_1. The learned vector field v_θ, conditioned on $\mathcal{R}$, $\mathcal{T}$, and the prescribed endpoint identities $c_\mathcal{T}$, yields a prediction of the clean coordinates $\hat{X}_1$:

$$X_t = (1-t)X_0 + tX_1, \qquad \hat{X}_1 = X_t + (1-t)v_\theta(X_t, t; \mathcal{R}, \mathcal{T}, c_\mathcal{T}). \tag{6}$$

During generation, peptide coordinates evolve while receptor coordinates remain fixed. The atom-level and residue-level encoders and decoders follow the multiscale design of AlphaFold 3 (Abramson et al., 2024) and RFdiffusion3 (Butcher et al., 2025), with topology conditioning injected across layers (Figure 2). A sequence readout predicts residue identities from detached decoder features, so sequence losses train the readout without backpropagating into the coordinate trunk. Unlike methods that jointly evolve sequences and structures (Li et al., 2024; Campbell et al., 2024), only coordinates follow the flow trajectory, while free residue identities are predicted at the end of generation.

Training objective. The overall training objective is:

$$\mathcal{L} = \lambda_x \mathcal{L}_{\text{coord}} + \lambda_c \mathcal{L}_{\text{close}} + \lambda_s \mathcal{L}_{\text{CE}} + \lambda_h \alpha(k) \mathcal{L}_{\text{chem}}, \tag{7}$$

where $\lambda_x, \lambda_c, \lambda_s, \lambda_h$ balance the contributions of the four loss terms. $\mathcal{L}_{\text{coord}}$ combines time-weighted clean-coordinate MSE with smooth lDDT. $\mathcal{L}_{\text{close}}$ is a smooth L1 loss on the pairwise distances of five atoms surrounding the closure. Sequence cross-entropy $\mathcal{L}_{\text{CE}}$ and the auxiliary geometry-based sequence penalty $\mathcal{L}_{\text{chem}}$ apply only at $t \geq 0.9$. The latter uses detached geometry, so it updates sequence logits without introducing coordinate gradients. Its warmup factor $\alpha(k)$ depends on the training step k. Appendix A.5 specifies the architecture, reductions, and weights, and Algorithm 1 summarizes the training procedure.

3.3 Generation-aware confidence

After sampling, an independently trained confidence model C_ϕ ranks generated candidates without updating the generator. Its inputs are the final sequence S, candidate coordinates $\bar{X}$, detached denoiser atom representations $\text{sg}(h^{\text{gen}})$, and the receptor and topology conditions $(\mathcal{R}, \mathcal{T})$. A reduced CovaTide trunk processes these inputs at $t = 0$ and outputs atomwise pLDDT predictions $\hat{q}$, together with candidate-level PDE $\hat{e}_{\text{pep}}$ and iPDE $\hat{e}_{\text{int}}$. We train the confidence model as:

$$(\hat{q}, \hat{e}_{\text{pep}}, \hat{e}_{\text{int}}) = C_\phi\big(S, \bar{X}, \text{sg}(h^{\text{gen}}); \mathcal{R}, \mathcal{T}\big), \tag{8}$$

$$\mathcal{L}_{\text{conf}} = \mathcal{L}_{\text{pLDDT}} + \mathcal{L}_{\text{PDE}} + 1.5\mathcal{L}_{\text{iPDE}} + \beta(k)\mathcal{L}_{\text{rank}}, \tag{9}$$

where the first three terms are bin-wise cross-entropy losses, while $\mathcal{L}_{\text{rank}}$ is the weighted within-target Huber loss on candidate-level mean pLDDT, PDE, and iPDE. Appendix A.6 defines the supervision targets and details the ranking loss. Algorithm 2 summarizes confidence model training.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Datasets. For linear peptide training, we use two datasets, ProtFrag and PepBench (Kong et al., 2024). For cyclic peptide training, we use a subset of CPBind, a dataset derived from CPSea (Yang et al., 2025) through energy-based filtering. For evaluation, we use the LNR dataset for linear peptide design and an LNR subset of 57 targets selected following CPSea for cyclic peptide design. Further dataset and preprocessing details are provided in Appendices A.1 and A.2.

Baselines. We compare CovaTide with PepGLAD (Kong et al., 2024), PepFlow (Li et al., 2024), and UniMoMo (Kong et al., 2025) for linear peptide design, and with PepGLAD$_{\mathrm{cyc}}$, PepFlow$_{\mathrm{cyc}}$, and DiffPepBuilder$_{\mathrm{cyc}}$ (Wang et al., 2024) for cyclic peptide design. The three cyclic peptide baselines are trained on the same subset of CPBind as CovaTide. We additionally include CP-Composer (Jiang et al., 2025) as a zero-shot baseline trained on linear peptide data, following its original setting.

Metrics. For linear peptides, we follow the metrics in UniMoMo. AAR measures sequence recovery, C-RMSD evaluates receptor-frame binding pose recovery, and L-RMSD assesses intrinsic peptide conformation. Rosetta dG measures interface energy (Alford et al., 2017), while IMP reports the fraction of targets where at least one generated peptide achieves a lower dG than the native peptide. Clash rates and torsion JSD assess structural rationality, while clustering metrics quantify sequence and structure diversity. For cyclic peptides, cyclization success (Cyc.) requires the endpoint C_β distance to fall within the reference p5-p95 interval for the corresponding cyclization type. Bond success requires the absolute deviation of the closure bond length from its ideal value to be at most 0.2 Å. Peptide-receptor clash rate, side-chain torsion JSD, and Ramachandran statistics assess structural quality. Rosetta and Vina scores evaluate interaction favorability, with success requiring $dG < 0$ and Vina score < -3, respectively. Complete definitions are provided in Appendix A.7.

For linear peptides, we select the best value of each recovery and energy metric per target and then average across targets. For cyclic peptides, cyclization, bond, and energy-based success rates, clash rates, and Ramachandran statistics are micro-averaged over all samples, whereas Rosetta and Vina scores are reported as the mean of per-target minima. Torsion JSDs are computed from dihedral distributions pooled by residue and torsion type and then averaged across groups. Because DiffPepBuilder$_{\mathrm{cyc}}$ generates only backbone atom coordinates, its side-chain torsion JSD cannot be computed, and its bond success rate is evaluated only for head-to-tail cyclization.

4.2 MAIN RESULTS

Linear peptide design. For each target, we generate 100 peptides per model for evaluation. CovaTide achieves the best recovery and energy metrics for linear peptide design (Table 1). Its highest AAR and lowest C-RMSD and L-RMSD indicate more accurate recovery of peptide sequences, receptor-frame binding poses, and intrinsic conformations, respectively. CovaTide also achieves the lowest Rosetta interface energy and the highest IMP, supporting favorable receptor-peptide interactions. It further yields the lowest clash rates, indicating improved steric compatibility. For torsional geometry, CovaTide ranks second in both backbone and side-chain JSD, while PepFlow and UniMoMo achieve the best backbone and side-chain JSD, respectively. PepGLAD and UniMoMo generate more diverse sequences and structures.

Table 1: Evaluation of de novo linear peptide design.

	Recovery			Energy		Rationality				Diversity	
Model	AAR ↑	C-RMSD ↓	L-RMSD ↓	dG ↓	IMP ↑	$\mathrm{Clash}_{\mathrm{in}}$ ↓	$\mathrm{Clash}_{\mathrm{out}}$ ↓	$\mathrm{JSD}_{\mathrm{bb}}$ ↓	$\mathrm{JSD}_{\mathrm{sc}}$ ↓	Seq. ↑	Struct. ↑
PepGLAD	0.389	3.194	1.710	-18.765	0.151	5.294%	3.821%	0.513	0.396	**0.654**	**0.772**
PepFlow	0.387	3.358	1.426	-24.983	0.226	1.195%	4.571%	**0.205**	0.703	0.388	0.343
UniMoMo	0.395	2.299	1.339	-33.003	0.376	0.481%	1.634%	0.236	**0.178**	0.622	0.614
CovaTide	**0.479**	**2.039**	**1.105**	**-36.459**	**0.441**	**0.396%**	**1.377%**	0.228	0.257	0.412	0.362

Cyclic peptide design. For each target, CovaTide and CP-Composer generate 10 peptides per cyclization type, while models that cannot condition on cyclization type generate 30 peptides. For these models, peptides are classified for geometric evaluation by endpoint residue identities, whereas cyclization types for relaxation are assigned based on endpoint C_β distances following the CPSea protocol. CovaTide achieves the highest cyclization, bond, Rosetta, and Vina success rates, together with the lowest peptide-receptor clash rate, side-chain torsion JSD, and Rosetta dG (Table 2). Its strong cyclization and bond success is consistent with repeated multiscale topology conditioning, which guides global endpoint geometry at the residue level and enforces local bond geometry at the atom level. PepFlow$_{\text{cyc}}$ retains the best Vina score and highest Ramachandran allowed fraction, whereas CovaTide achieves the highest favored fraction. Across cyclization types, CovaTide consistently yields the lowest Rosetta dG and the best Vina scores for disulfide and isopeptide designs, while PepFlow$_{\text{cyc}}$ performs best for head-to-tail peptides (Table 3). CovaTide also achieves the highest Ramachandran allowed fraction for isopeptide designs and remains close to the best result for disulfide designs.

Table 2: Evaluation of de novo cyclic peptide design.

	Rationality		Success					Ramachandran		Energy score	
Model	Clash$_{\text{out}}$↓	JSD$_{\text{sc}}$↓	Cyc.↑	Bond↑	Bond$_{\text{ht}}$↑	Rosetta↑	Vina↑	Allowed↑	Favored↑	Rosetta↓	Vina↓
PepGLAD$_{\text{cyc}}$	1.871%	0.510	0.504	0.184	0.206	0.761	0.899	0.734	0.417	-30.547	-7.197
PepFlow$_{\text{cyc}}$	7.781%	0.543	0.670	0.160	0.298	0.872	0.914	**0.836**	0.584	-27.658	**-8.065**
DiffPepBuilder$_{\text{cyc}}$	16.795%	-	0.475	-	0.212	0.689	0.758	0.820	0.566	-29.332	-7.021
CP-Composer	23.899%	0.398	0.275	0.007	0.007	0.524	0.492	0.594	0.314	-25.527	-6.218
CovaTide	**1.854%**	**0.286**	**0.754**	**0.870**	**0.737**	**0.975**	**0.973**	0.830	**0.597**	**-35.528**	-7.861

Table 3: Topology-specific evaluation of de novo cyclic peptide design.

	Head-to-tail			Disulfide			Isopeptide		
Model	Allowed↑	Rosetta↓	Vina↓	Allowed↑	Rosetta↓	Vina↓	Allowed↑	Rosetta↓	Vina↓
PepGLAD$_{\text{cyc}}$	0.741	-27.625	-6.171	0.744	-26.289	-6.321	0.729	-27.313	-6.885
PepFlow$_{\text{cyc}}$	**0.846**	-25.802	**-7.235**	**0.831**	-23.643	-7.030	0.818	-23.023	-7.453
DiffPepBuilder$_{\text{cyc}}$	0.829	-20.459	-5.509	0.813	-24.026	-6.255	0.811	-16.220	-6.054
CP-Composer	0.608	-10.422	-5.497	0.588	-12.721	-5.052	0.585	-10.741	-5.939
CovaTide	0.819	**-32.014**	-6.920	0.829	**-32.504**	**-7.371**	**0.830**	**-32.060**	**-7.492**

Generalization to internal cyclization. Beyond terminal closures, CovaTide generalizes to prescribed internal disulfide and isopeptide linkages, achieving 77.8% bond success and 75.2% joint success under the bond and Rosetta criteria across 540 generated candidates from 9 targets. Detailed evaluation protocols and results are provided in Appendix A.12.

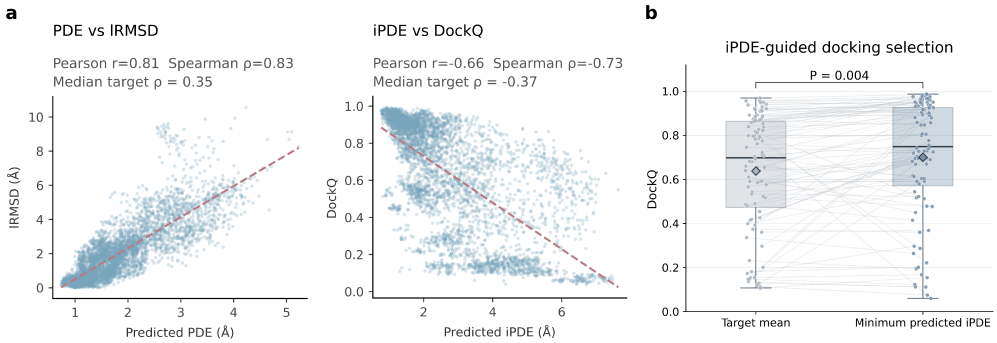

Figure 3: **Confidence assessment for generated peptides.** (a) PDE versus iRMSD and iPDE versus DockQ. (b) Target-wise mean DockQ compared with DockQ of the lowest-iPDE candidate.

Confidence-based candidate assessment. We evaluate the confidence model on eligible linear docking candidates to determine whether its predictions can identify accurate binding poses and guide candidate selection. Lower predicted PDE is associated with more accurate peptide placement, while lower predicted iPDE indicates better interface quality (Figure 3a). The corresponding Spearman correlations are 0.83 for PDE-iRMSD and -0.73 for iPDE-DockQ. Median within-target correlations retain these directions with smaller magnitudes, suggesting a weaker but still informative ranking signal among candidates for the same receptor. To test whether this signal improves candidate selection, we choose the candidate with the lowest predicted iPDE for each target. The selected candidates achieve a mean DockQ of 0.70 across targets, compared with 0.63 when averaging over all candidates for each target (paired t-test, $P = 0.004$, Figure 3b), supporting the use of predicted interface error to prioritize more accurate binding poses.

4.3 Case Study

Molecular dynamics examples. To assess whether the generated cyclic peptides maintain stable binding conformations, we performed 100-ns explicit-solvent molecular dynamics simulations for 4J86 and 3BRH, comparing a generated disulfide peptide with the reference linear binder for each receptor (Figure 4). In both cases, the cyclic designs deviated less from their initial receptor-bound poses and showed smaller RMSD fluctuations over the final 50 ns, suggesting greater conformational stability than the linear references. After alignment on receptor C_α atoms, the peptide heavy-atom RMSDs were 2.87 ± 0.27 versus 4.72 ± 0.41 Å for 4J86, and 3.40 ± 0.15 versus 6.03 ± 0.64 Å for 3BRH (cyclic versus linear). These observations are consistent with disulfide closure restricting conformational rearrangements and helping preserve the designed binding poses in these two case studies. Appendix A.10 provides the complete simulation and analysis protocol.

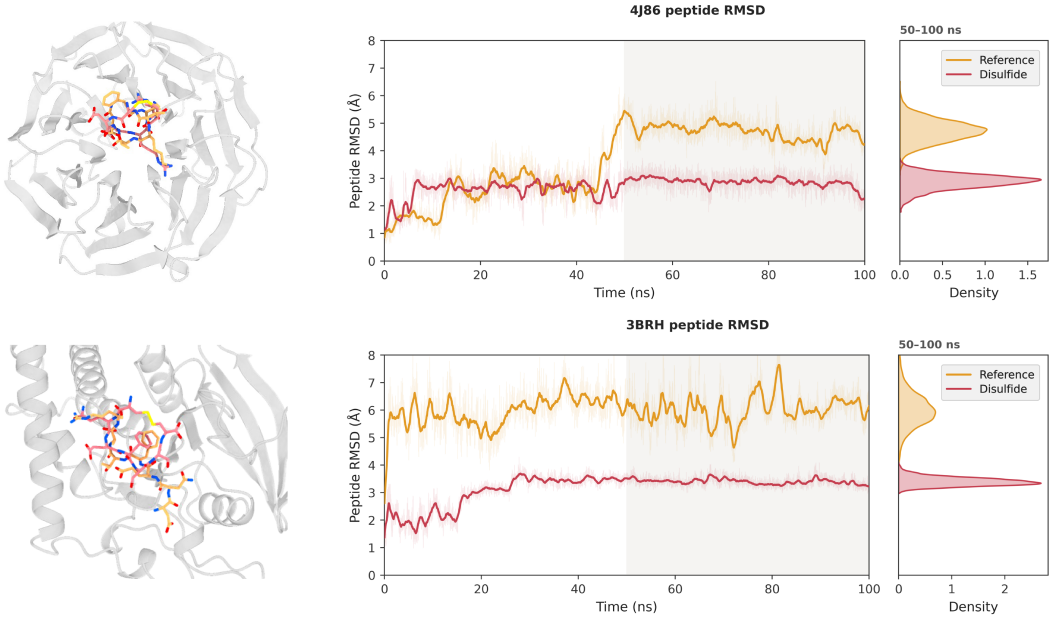

Figure 4: **Molecular dynamics examples for 4J86 and 3BRH.** RMSD trajectories of peptide heavy atoms and distributions over the final 50 ns for the reference and generated disulfide peptides.

Rhodopsin design examples. Rhodopsin is a prototypical G protein-coupled receptor (GPCR) expressed primarily in retinal rod photoreceptors, where it initiates the phototransduction cascade and mediates dim-light vision. Using this receptor as a case study, we examine representative linear, disulfide, head-to-tail, and isopeptide designs generated by CovaTide for the same binding pocket (Figure 5). These designs adopt distinct backbone conformations that fit within the pocket and position side chains along the receptor interface. Electrostatic surface maps further suggest charge complementarity between the designed peptides and their binding pockets. These observations, to-

gether with favorable Rosetta interface energies for all four designs, support the structural plausibility of the generated binding poses across the evaluated peptide topologies.

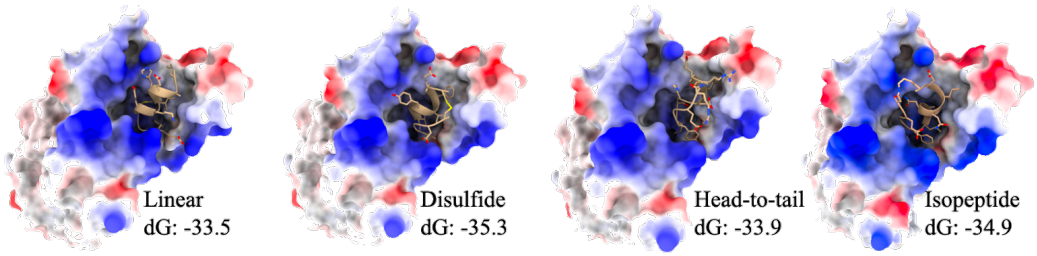

Figure 5: **Topology-conditioned peptide design targeting rhodopsin.** Representative linear and cyclic peptide designs generated by CovaTide for the same receptor pocket.

4.4 Ablation

To evaluate the topology adapter's contribution to cyclization, we compare the full model with variants that remove it, reduce its depth, or omit residue- or atom-level conditioning. Table 4 reports Cyc. and $\text{Closure}_{\text{range}}$, the fractions of generated structures for which the endpoint C_β distance and closure bond length, respectively, fall within the corresponding p5-p95 interval. Removing the adapter reduces both metrics for every topology, with the largest drop for head-to-tail closure. Because head-to-tail cyclization imposes no specific endpoint residue identities, removing the adapter leaves no explicit cyclic-specific conditioning beyond that used for linear peptides. The deep adapter also outperforms both single-level variants on both metrics across all three topologies, supporting the use of combined residue- and atom-level conditioning. It achieves the highest values for disulfide and isopeptide closures, while the shallow adapter performs slightly better for head-to-tail closure. This pattern suggests that side-chain cyclization benefits more from repeated topology injection than head-to-tail backbone closure, possibly reflecting its more complex local geometric requirements.

Table 4: Ablation of the deep topology adapter.

	Disulfide		Head-to-tail		Isopeptide	
Model	Cyc. ↑	$\text{Closure}_{\text{range}}$ ↑	Cyc. ↑	$\text{Closure}_{\text{range}}$ ↑	Cyc. ↑	$\text{Closure}_{\text{range}}$ ↑
w/o adapter	0.581	0.753	0.059	0.049	0.461	0.500
Shallow adapter	<u>0.714</u>	<u>0.900</u>	**0.792**	**0.785**	0.709	<u>0.852</u>
w/o atom level	0.674	0.714	0.757	0.465	<u>0.728</u>	0.759
w/o residue level	0.600	0.624	0.749	0.572	0.670	0.603
Deep adapter	**0.736**	**0.946**	<u>0.775</u>	<u>0.742</u>	**0.759**	**0.953**

5 Conclusion

We present CovaTide, a receptor-conditioned all-atom flow model for unified linear and cyclic peptide design. Its deep topology adapter couples residue connectivity with atom-level bond information throughout generation, allowing a single model to support linear, head-to-tail, disulfide, and isopeptide designs. Benchmark results show improved sequence and binding pose recovery for linear peptides and higher cyclization, bond, and energy-based success rates for cyclic peptides. The generation-aware confidence model further helps prioritize accurate binding poses in linear peptide docking. Molecular dynamics and structural case studies suggest that the designed peptides can maintain stable binding poses and form favorable interactions with their receptors. Together, these results support explicit covalent topology conditioning for unified peptide design. Extending the current formulation beyond a single closure and canonical amino acids to multiple closures and non-canonical residues is a natural direction for future work.

Ethics Statement

This work presents CovaTide, a computational framework for receptor-conditioned design of linear and cyclic peptides, with potential applications in peptide therapeutics research. We train CovaTide on existing structural datasets and evaluate generated candidates using computational benchmarks and molecular dynamics simulations. We recognize that generative peptide design carries potential dual-use risks, as such methods could also be applied for harmful purposes. We therefore encourage careful consideration of intended applications and potential dual-use implications, as well as responsible development and use in support of beneficial scientific and biomedical research.

The use of large language models

Large language models (LLMs) were used to assist with manuscript polishing, literature discovery, and coding. They were not used to generate data or run experiments. All technical claims, equations, experimental designs, analyses, citations, and numerical results were reviewed and verified by the authors. We take full responsibility for the scientific content of this work.

Reproducibility statement

We provide detailed descriptions of data preparation, model implementation, training, and evaluation to support reproducibility. Appendix A.1 summarizes the datasets and their sources, while Appendix A.2 describes data preprocessing and the train/validation split. Section 3.1 and Appendices A.3-A.5 specify the model representations, topology conditioning, model architecture, training objectives, and optimization settings. Appendix A.6 describes confidence model training and decoy generation. Algorithms 1 and 2 summarize the training procedures for the generator and confidence model, respectively. Section 4.1 presents the evaluation protocol, while Appendices A.7 and A.8 define the evaluation metrics and ablation configurations. Appendix A.10 provides the molecular dynamics simulation and trajectory analysis protocol.

REFERENCES

Mark James Abraham, Teemu Murtola, Roland Schulz, Szilárd Páll, Jeremy C. Smith, Berk Hess, and Erik Lindahl. GROMACS: High performance molecular simulations through multi-level parallelism from laptops to supercomputers. *SoftwareX*, 1–2:19–25, 2015.

Josh Abramson, Jonas Adler, Jack Dunger, Richard Evans, Tim Green, Alexander Pritzel, Olaf Ronneberger, Lindsay Willmore, Andrew J. Ballard, Joshua Bambrick, et al. Accurate structure prediction of biomolecular interactions with AlphaFold 3. *Nature*, 630(8016):493–500, 2024.

Rebecca F. Alford, Andrew Leaver-Fay, Jeliazko R. Jeliazkov, Matthew J. O'Meara, Frank P. DiMaio, Hahnbeom Park, Maxim V. Shapovalov, P. Douglas Renfrew, Vikram K. Mulligan, Kalli Kappel, et al. The Rosetta All-Atom Energy Function for Macromolecular Modeling and Design. *Journal of Chemical Theory and Computation*, 13(6):3031–3048, 2017.

Mattia Bernetti and Giovanni Bussi. Pressure control using stochastic cell rescaling. *The Journal of Chemical Physics*, 153(11):114107, 2020.

Jasper Butcher, Rohith Krishna, Raktim Mitra, Rafael Isaac Brent, Yanjing Li, Nathaniel Corley, Paul T Kim, Jonathan Funk, Simon Valentin 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. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 5453–5512. PMLR, 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.

Tom Darden, Darrin York, and Lee Pedersen. Particle mesh Ewald: An $N \cdot \log(N)$ method for Ewald sums in large systems. *The Journal of Chemical Physics*, 98(12):10089–10092, June 1993.

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

Peter Eastman, Jason Swails, John D. Chodera, Robert T. McGibbon, Yutong Zhao, Kyle A. Beauchamp, Lee-Ping Wang, Andrew C. Simmonett, Matthew P. Harrigan, Chaya D. Stern, et al. OpenMM 7: Rapid development of high performance algorithms for molecular dynamics. *PLOS Computational Biology*, 13(7):e1005659, 2017.

Noelia Ferruz, Steffen Schmidt, and Birte Höcker. ProtGPT2 is a deep unsupervised language model for protein design. *Nature Communications*, 13(1):4348, 2022.

Ilanit Gutman, Ron Gutman, John Sidney, Leila Chihab, Michele Mishto, Juliane Liepe, Anthony Chiem, Jason Greenbaum, Zhen Yan, Alessandro Sette, et al. Predicting the Success of Fmoc-Based Peptide Synthesis. *ACS Omega*, 7(27):23771–23781, 2022.

Berk Hess, Henk Bekker, Herman J. C. Berendsen, and Johannes G. E. M. Fraaije. LINCS: A linear constraint solver for molecular simulations. *Journal of Computational Chemistry*, 18(12): 1463–1472, September 1997.

Chad W. Hopkins, Scott Le Grand, Ross C. Walker, and Adrian E. Roitberg. Long-time-step molecular dynamics through hydrogen mass repartitioning. *Journal of Chemical Theory and Computation*, 11(4):1864–1874, 2015.

Xinjian Ji, Alexander L. Nielsen, and Christian Heinis. Cyclic Peptides for Drug Development. *Angewandte Chemie International Edition*, 63(3):e202308251, 2024.

Dapeng Jiang, Xiangzhe Kong, Jiaqi Han, Mingyu Li, Rui Jiao, Wenbing Huang, Stefano Ermon, Jianzhu Ma, and Yang Liu. Zero-Shot Cyclic Peptide Design via Composable Geometric Constraints. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pp. 27553–27568. PMLR, 2025.

William L. Jorgensen, Jayaraman Chandrasekhar, Jeffry D. Madura, Roger W. Impey, and Michael L. Klein. Comparison of simple potential functions for simulating liquid water. *The Journal of Chemical Physics*, 79(2):926–935, July 1983.

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873):583–589, 2021.

Elizabeth Jurrus, Dave Engel, Keith Star, Kyle Monson, Juan Brandi, Lisa E. Felberg, David H. Brookes, Leighton Wilson, Jiahui Chen, Karina Liles, et al. Improvements to the APBS biomolecular solvation software suite. *Protein Science*, 27(1):112–128, 2018.

Xiangzhe Kong, Yinjun Jia, Wenbing Huang, and Yang Liu. Full-Atom Peptide Design with Geometric Latent Diffusion. In *Advances in Neural Information Processing Systems*, volume 37, pp. 74808–74839. Curran Associates, Inc., 2024.

Xiangzhe Kong, Zishen Zhang, Ziting Zhang, Rui Jiao, Jianzhu Ma, Wenbing Huang, Kai Liu, and Yang Liu. UniMoMo: Unified Generative Modeling of 3D Molecules for De Novo Binder Design. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pp. 31397–31418. PMLR, 2025.

Jiahan Li, Chaoran Cheng, Zuofan Wu, Ruihan Guo, Shitong Luo, Zhizhou Ren, Jian Peng, and Jianzhu Ma. Full-Atom Peptide Design based on Multi-modal Flow Matching. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 27615–27640. PMLR, 2024.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow Matching for Generative Modeling. In *The Eleventh International Conference on Learning Representations*, 2023.

James A. Maier, Carmenza Martinez, Koushik Kasavajhala, Lauren Wickstrom, Kevin E. Hauser, and Carlos Simmerling. ff14SB: Improving the Accuracy of Protein Side Chain and Backbone Parameters from ff99SB. *Journal of Chemical Theory and Computation*, 11(8):3696–3713, 2015.

Valerio Mariani, Marco Biasini, Alessandro Barbato, and Torsten Schwede. lDDT: a local superposition-free score for comparing protein structures and models using distance difference tests. *Bioinformatics*, 29(21):2722–2728, 2013.

Shuichi Miyamoto and Peter A. Kollman. Settle: An analytical version of the SHAKE and RATTLE algorithm for rigid water models. *Journal of Computational Chemistry*, 13(8):952–962, 1992.

Stephen A. Rettie, Katelyn V. Campbell, Asim K. Bera, Alex Kang, Simon Kozlov, Yensi Flores Bueso, Joshmyn De La Cruz, Maggie Ahlrichs, Suna Cheng, Stacey R. Gerben, et al. Cyclic peptide structure prediction and design using AlphaFold2. *Nature Communications*, 16(1):4730, 2025a.

Stephen A. Rettie, David Juergens, Victor Adebomi, Yensi Flores Bueso, Qinqin Zhao, Alexandria N. Leveille, Andi Liu, Asim K. Bera, Joana A. Wilms, Alina Üffing, et al. Accurate de novo design of high-affinity protein-binding macrocycles using deep learning. *Nature Chemical Biology*, 21(12): 1948–1956, 2025b.

Oleg Trott and Arthur J. Olson. AutoDock Vina: Improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *Journal of Computational Chemistry*, 31(2):455–461, 2010.

Alexander A. Vinogradov, Yizhen Yin, and Hiroaki Suga. Macrocyclic Peptides as Drug Candidates: Recent Progress and Remaining Challenges. *Journal of the American Chemical Society*, 141(10): 4167–4181, 2019.

Fanhao Wang, Yuzhe Wang, Laiyi Feng, Changsheng Zhang, and Luhua Lai. Target-Specific De Novo Peptide Binder Design with DiffPepBuilder. *Journal of Chemical Information and Modeling*, 64(24):9135–9149, 2024.

Lei Wang, Nanxi Wang, Wenping Zhang, Xurui Cheng, Zhibin Yan, Gang Shao, Xi Wang, Rui Wang, and Caiyun Fu. Therapeutic peptides: current applications and future directions. *Signal Transduction and Targeted Therapy*, 7(1), 2022.

Joseph L. Watson, David Juergens, Nathaniel R. Bennett, Brian L. Trippe, Jason Yim, Helen E. Eisenach, Woody Ahern, Andrew J. Borst, Robert J. Ragotte, Lukas F. Milles, et al. De novo design of protein structure and function with RFdiffusion. *Nature*, 620(7976):1089–1100, 2023.

Ziyi Yang, Hanyuan Xie, Yinjun Jia, Xiangzhe Kong, Jiqing Zheng, Ziting Zhang, Yang Liu, Lei Liu, and Yanyan Lan. CPSea: Large-scale cyclic peptide-protein complex dataset for machine learning in cyclic peptide design. In *Advances in Neural Information Processing Systems*, volume 38, Main Conference. Curran Associates, Inc., 2025.

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

Jingjie Zhang, Hanqun Cao, Haosen Shi, Mutian He, Yu Wang, Zijun Gao, Fang Wu, Xiaojun Yao, Chang-Yu Hsieh, Sinno Jialin Pan, et al. GeoCycler: Reward-Aligned 3D Diffusion for Constraint-Conditioned Cyclic Peptide Design. *arXiv preprint arXiv:2605.23407*, 2026.

Xiangxin Zhou, Mingyu Li, Yi Xiao, Jiahan Li, Dongyu Xue, Zaixiang Zheng, Jianzhu Ma, and Quanquan Gu. Designing Cyclic Peptides via Harmonic SDE with Atom-Bond Modeling. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pp. 78827–78858. PMLR, 2025.

A APPENDIX

A.1 DATASETS

ProtFrag. ProtFrag is a protein fragment dataset introduced with PepGLAD to augment peptide training data (Kong et al., 2024). It contains 70,498 peptide-like fragments extracted from PDB protein chains after sequence-based deduplication. Fragments are selected based on length, amino acid composition, stability, and buried surface area. The surrounding protein residues provide interaction context, allowing intrachain contacts to serve as additional examples of peptide-receptor geometry. We use ProtFrag to supplement linear peptide training.

PepBench. PepBench is a dataset for receptor-conditioned peptide sequence and structure design curated from PDB complexes (Kong et al., 2024). Its construction retains receptors longer than 30 residues and peptides of 4-25 residues, yielding 6,105 complexes after removing duplicates at 90% sequence identity for both partners. Receptors are then clustered at a 40% sequence identity threshold, and complexes sharing clusters with the LNR test set are excluded from training and validation. The published split contains 4,157 training complexes and 114 validation complexes. PepBench provides experimentally determined protein-peptide complexes for our linear peptide training, complementing the fragment-derived examples in ProtFrag.

CPSea. CPSea is a computationally constructed dataset of approximately 2.71 million cyclic peptide-protein complexes derived from the AlphaFold Database (Yang et al., 2025). Its pipeline extracts peptide segments from predicted protein domains, uses surrounding residues as receptor context, and selects candidate closure sites based on endpoint geometry. After introducing covalent bonds and applying energy minimization and structural filtering, the dataset covers head-to-tail, disulfide, and isopeptide linkages. CPBind is an affinity-filtered subset of CPSea containing approximately 545,000 complexes selected with an AutoDock Vina score below -6 and a Rosetta dG below -25. For cyclic peptide training, we use a subset of CPBind shared by all models in the cyclic peptide comparison. The preprocessing procedure is described in Appendix A.2.

LNR. The large non-redundant dataset (LNR) is an expert-curated collection of protein-peptide complexes used to evaluate generalization across receptor targets. For linear evaluation, we use the same LNR test set as PepGLAD, comprising 93 protein-peptide complexes retained from the original collection of 96 complexes after excluding peptides containing non-canonical amino acids (Kong et al., 2024). For cyclic evaluation, we use a subset of LNR comprising 57 receptor targets selected in CPSea based on peptide length and binding site topology (Yang et al., 2025). On these targets, we evaluate head-to-tail, disulfide, and isopeptide designs under prescribed topology conditions.

A.2 DATA PROCESSING

Linear and cyclic peptide datasets are prepared separately for pretraining and fine-tuning. Table 5 summarizes the resulting training and validation sets. Figure 6 shows the peptide length distributions for linear peptides and each cyclization type.

Linear peptide datasets. For pretraining, we use ProtFrag and randomly reserve approximately 2% of its 70,498 samples for validation, resulting in 69,088 training and 1,410 validation samples. For fine-tuning, we use the predefined PepBench split, which contains 4,157 training and 114 validation samples.

Cyclic peptide datasets. For both pretraining and fine-tuning, we use CPBind and select samples separately for each cyclization type based on Rosetta dG. Samples within each type are ranked by increasing dG. To balance dataset sizes across cyclization types, we retain all 74,513 disulfide samples and the 80,000 lowest-dG samples for each of the head-to-tail and isopeptide types.

Cyclic peptide splits. Within each cyclization type, the 10% of samples with the lowest Rosetta dG are assigned to fine-tuning, while the remaining 90% are used for pretraining, ensuring that the two stages use disjoint samples. For pretraining, approximately 2% of the samples are randomly held

out for validation. For fine-tuning, approximately 2% of the unique cluster centers are randomly selected, and all samples associated with these centers are assigned to the validation set.

Cyclic peptide filtering against LNR. After splitting, we filter CPBind against the 93 LNR test receptors based on receptor sequence similarity. We perform bidirectional searches with MMseqs2 easy-search, using CPBind and LNR in turn as the query database (identity = 0.4, coverage = 0.95, cov-mode = 1, sensitivity = 7.5, alignment-mode = 3, seq-id-mode = 0, E-value = 10^{-3}). The maximum number of returned sequences is set to the size of the target database (max-seqs = 93 for CPBind-to-LNR and 234,513 for LNR-to-CPBind). A CPBind sample is removed if its receptor has a qualifying hit in either direction. Filtering is performed directly on receptor sequences and does not rely on cluster assignments.

Sequence identity is computed over the alignment length, including internal gap columns. With target-based coverage, CPBind-to-LNR requires 95% coverage of the LNR receptor, whereas LNR-to-CPBind requires 95% coverage of the CPBind receptor. Taking the union of both searches captures sequence-similar receptor pairs of different lengths that may satisfy the coverage criterion in only one direction.

CPBind-to-LNR yields no qualifying hits, while LNR-to-CPBind identifies 14 CPBind samples for removal. The remaining 234,499 samples retain their original pretraining or fine-tuning split and training or validation assignment, with no resampling or repartitioning.

Table 5: Final training and validation data sizes for linear and cyclic peptides.

Peptide type	Total	Pretraining		Fine-tuning	
		Train	Validation	Train	Validation
Linear	74,769	69,088	1,410	4,157	114
Disulfide	74,507	65,716	1,341	7,305	145
Head-to-tail	79,999	70,559	1,440	7,809	191
Isopeptide	79,993	70,554	1,440	7,809	190
Total	309,268	275,917	5,631	27,080	640

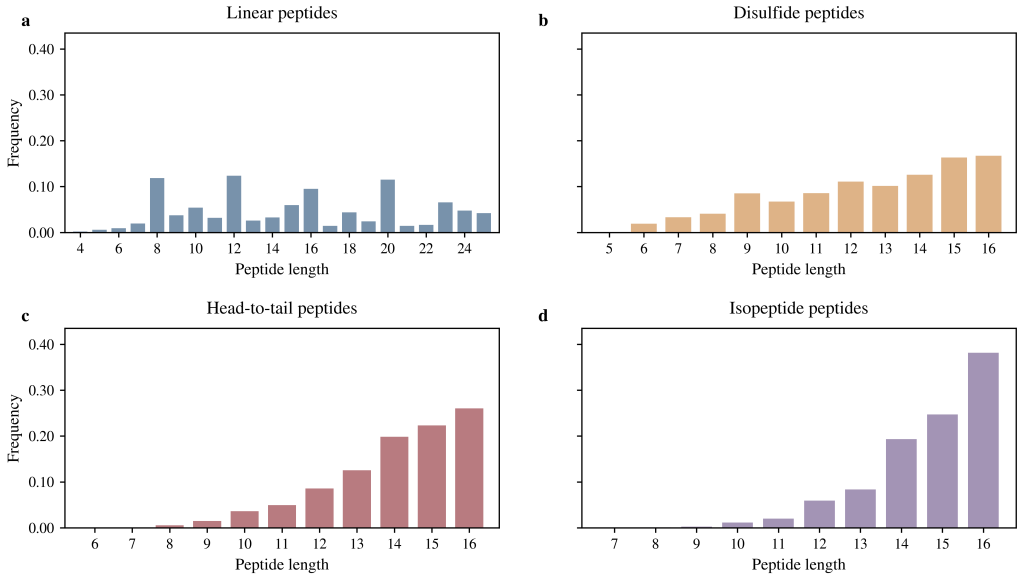

Figure 6: Peptide length distributions for (a) linear, (b) disulfide, (c) head-to-tail, and (d) isopeptide peptides. Frequencies are normalized within each peptide type.

A.3 Representation and conditioning

Atom14 representation. Atom14 represents each residue using 14 atom slots, with a residue-dependent mapping from slots to heavy-atom names (Jumper et al., 2021). These slots accommodate all 20 canonical amino acids in their standard residue templates, with tryptophan occupying all 14 slots. Residues with fewer atoms leave unused slots, represented by empty entries in the atom mapping and excluded by an atom presence mask. This compact layout allows the same slot to correspond to different side-chain atoms across residue types.

Atom37 representation. Atom37 instead assigns a fixed slot to each of 37 atom names across canonical residues (Jumper et al., 2021). A given atom name therefore occupies the same slot regardless of residue identity, while absent atoms are masked out. The vocabulary includes the terminal oxygen OXT and follows the ordering N, CA, C, CB, O, CG, CG1, CG2, OG, OG1, SG, CD, CD1, CD2, ND1, ND2, OD1, OD2, SD, CE, CE1, CE2, CE3, NE, NE1, NE2, OE1, OE2, CH2, NH1, NH2, OH, CZ, CZ2, CZ3, NZ, and OXT.

Atom73 representation. Protpardelle introduces Atom73 to represent alternative side-chain states when residue identities are not yet determined (Chu et al., 2024). It uses shared slots for N, CA, C, O, and CB, while allocating the remaining side-chain slots separately for each amino acid type. Thus, the same atom name in different residue types, such as NE2 in glutamine and histidine, occupies distinct slots. The resulting 73-slot layout represents a superposition of possible residue states, from which atoms corresponding to a selected residue identity can be extracted using a mask.

Representation used in CovaTide. We use an Atom14-style fixed-slot representation based on the compact per-residue layout of Atom14 (Jumper et al., 2021). All peptide residues share 14 generic slots: N, CA, C, O, CB, and $\mathrm{V}_0, \ldots, \mathrm{V}_8$. A residue-specific mapping assigns physical atom names and presence masks to these slots, with some assignments differing from the standard Atom14 ordering. During training, the native residue identity determines the target atom mapping and presence mask. During generation, the mapping and mask are determined by predicted residue identities at free positions and prescribed identities at fixed endpoints, with unassigned slots excluded from the physical atom output. Receptor atoms are represented explicitly, and their coordinates remain fixed throughout generation.

Sequence and topology are encoded separately. Sequence inputs use mask tokens at positions whose residue identities are to be generated, while fixed endpoints receive one-hot encodings of their prescribed identities. These identities are Cys/Cys for disulfide designs and Lys/Asp for isopeptide designs. Topology inputs encode the endpoint roles and covalent connectivity, with at most one additional bond beyond the peptide backbone. These inputs are used alongside the shared single and pair representations, sequence-relative positional encodings, and receptor context common to linear and cyclic peptides.

A.4 Deep covalent topology adapter

The adapter maps the prescribed topology $\mathcal{T} = (\tau, i, j)$ to conditioning features at both residue and atom levels. Each peptide residue u is assigned an endpoint role r_u, with $r_u = 0$ for non-endpoint residues. Head-to-tail endpoints receive distinct N-terminal and C-terminal roles, disulfide endpoints both receive the same Cys role, and isopeptide endpoints receive distinct Lys and Asp roles. A learned embedding table maps these categorical roles to residue-level embeddings:

$$e_u = E_{\mathrm{role}}(r_u). \tag{10}$$

At the atom level, each atom a is assigned the role embedding $e_{\pi(a)}$ of its parent residue, where $\pi(a)$ maps atom a to its parent residue.

Closure graph conditioning. Let $\mathcal{P}$ denote the set of peptide residues. For a cyclic peptide, $G_{\mathcal{T}}$ contains the sequential peptide edges together with the prescribed closure edge (i, j). We encode the resulting connectivity for each residue pair as:

$$\delta_{uv} = \begin{cases} 1 + \min\{d_{G_{\mathcal{T}}}(u, v), 32\}, & u, v \in \mathcal{P}, \\ 0, & \text{otherwise}, \end{cases} \tag{11}$$

and define the corresponding pairwise topology embedding as:

$$b_{uv} = E_{\text{dist}}(\delta_{uv}) + \mathbf{1}[\delta_{uv} > 0] E_{\text{type}}(\tau). \tag{12}$$

Here, $d_{G_\mathcal{T}}(u, v)$ denotes the shortest-path distance between peptide residues u and v in $G_\mathcal{T}$, while E_{dist} and E_{type} are learned embedding tables for graph distance and cyclization type, respectively. For linear peptides, $\delta_{uv} = 0$ for all residue pairs. Thus, index 0 is reserved for the absence of closure graph conditioning, whereas self-pairs in cyclic peptides use index 1. The cyclization type embedding is added only to peptide-peptide pairs with $\delta_{uv} > 0$.

Repeated residue-level injection. The adapter injects e_u and b_{uv} after each of the four Pairformer blocks and before each of the sixteen flow transformer blocks. Let $B \in \{\text{pairformer}, \text{flow}\}$ index these two stages, corresponding to the residue encoder and denoising trunk, respectively. At injection site ℓ in stage B, we denote the current residue single and pair representations by $s_u^{B,\ell}$ and $z_{uv}^{B,\ell}$. The updates are:

$$\tilde{s}_u^{B,\ell} = s_u^{B,\ell} + g_\ell^B W_s^B e_u, \qquad \tilde{z}_{uv}^{B,\ell} = z_{uv}^{B,\ell} + g_\ell^B W_z b_{uv}, \tag{13}$$

where W_s^B is stage-specific, whereas W_z is shared across the two stages. The projection layers are bias-free, and each injection site uses an independently learned scalar gate g_ℓ^B.

Typed atom-level injection. The bond-forming atom pair (a_i, a_j) is C-N for head-to-tail cyclization, S_γ-S_γ for disulfide cyclization, and Lys N_ζ-Asp C_γ for isopeptide cyclization. Using the closure edge set $\mathcal{E}_\mathcal{T}$ defined in Section 3.1, we define the symmetric closure edge indicator as:

$$\chi_{ab}^\mathcal{T} = \sum_{(a_i, a_j) \in \mathcal{E}_\mathcal{T}} \mathbf{1}[\{a, b\} = \{a_i, a_j\}]. \tag{14}$$

For linear peptides, $\chi_{ab}^\mathcal{T} = 0$ for all atom pairs. A learned embedding table E_{bond} embeds the cyclization type τ at the atom level. Before each of the three atom encoder blocks and three atom decoder blocks, the atom single context c_a and atom-pair context p_{ab} are updated as:

$$\tilde{c}_a^\ell = c_a + g_\ell^{\text{atom}} W_c e_{\pi(a)}, \qquad \tilde{p}_{ab}^\ell = p_{ab} + g_\ell^{\text{atom}} W_p \chi_{ab}^\mathcal{T} E_{\text{bond}}(\tau), \tag{15}$$

where W_c and W_p are shared bias-free projections, and g_ℓ^{atom} is the learned scalar gate at atom block ℓ. Both orientations of the closure edge receive the same update, and padding masks are reapplied after each injection. The four Pairformer gates, sixteen flow transformer gates, and six atom block gates are initialized to the inverse square root of the number of injection sites in their respective groups. To ensure that the typed closure edge condition remains accessible even when the endpoints fall outside the geometric neighborhood, we augment the sparse attention neighborhoods with the prescribed closure partner:

$$\mathcal{N}(a_i) \leftarrow \mathcal{N}(a_i) \cup \{a_j\}, \qquad \mathcal{N}(a_j) \leftarrow \mathcal{N}(a_j) \cup \{a_i\}. \tag{16}$$

A.5 Flow model and training objective

Flow architecture. Figure 2 summarizes the flow model, which consists of an atom-level encoder, a residue-level encoder, a residue-level denoising trunk, and an atom-level decoder. The atom stream maintains the dynamic single representation Q_a together with the atom-level contexts c_a and p_{ab} defined in Section 3.1. At the residue level, s_u^{pair} denotes the Pairformer conditioning stream, s_u^{flow} the dynamic denoising stream, and z_{uv} the pair representation for residues u and v. The full residue-pair tensor $z = (z_{uv})_{u,v}$ collects these pairwise features, with z^0 denoting its initialized state. Two initial Pairformer blocks construct $(s^{\text{pair},0}, z^0)$ from sequence, receptor, and relative-position features. The topology adapter described in Appendix A.4 provides topology-conditioned updates to the atom-level (c, p), residue-level (s^{pair}, z), and denoising (s^{flow}, z) representations at the corresponding blocks.

Atom-level encoder. The atom encoder integrates noisy coordinates, atom identities, fixed-coordinate indicators, and atom pair geometric features. Each of its three blocks applies pair-biased atom attention followed by a conditioned transition:

$$\bar{Q}_a^\ell = Q_a^\ell + \text{Attn}_{\text{atom}}(Q^\ell; \tilde{c}^\ell, \tilde{p}^\ell)_a, \tag{17}$$

$$Q_a^{\ell+1} = \bar{Q}_a^\ell + \text{Trans}_{\text{atom}}(\bar{Q}_a^\ell; \tilde{c}_a^\ell). \tag{18}$$

The attention neighborhood includes spatial and sequential neighbors as well as the prescribed closure partner, allowing the bond-forming atoms to exchange information directly.

Residue-level encoder. The three atom encoder blocks produce atom features $Q^{\text{enc}} = (Q_a^{\text{enc}})_a$, which are aggregated to their parent residues through Downcast and combined with a coordinate embedding to initialize the denoising stream $s_u^{\text{flow},0}$. Coordinates are normalized using a scale of $\sigma = 6$ Å. The residue-pair stream additionally receives a distogram $D_{uv}(X_t)$ computed using a representative atom for each residue. The residue-level initialization is:

$$s_u^{\text{flow},0} = E_X\left(\underset{a:\pi(a)=u}{\text{mean}} X_{t,a}/\sigma\right) + \text{Downcast}\big(\{Q_a^{\text{enc}}\}_{\pi(a)=u}, s_u^{\text{pair},0}\big)\,, \tag{19}$$

$$(s^{\text{pair}}, z) = \text{Pairformer}_2\big(s^{\text{pair},0}, z^0 + E_D(D(X_t))\big)\,, \tag{20}$$

where E_X is a linear coordinate embedding, E_D embeds the distogram bins, and Downcast performs gated cross-attention from the atoms belonging to residue u to its residue representation. Pairformer_2 denotes a stack of two consecutive Pairformer blocks. This stage transfers local atomic geometry to the residue level while preserving residue-pair context for the denoising trunk.

Residue-level denoising trunk. The sixteen residue-level transformer blocks iteratively update s^{flow}. Before block ℓ, the topology adapter produces $\tilde{s}^{\text{flow},\ell}$ and $\tilde{z}^\ell$ according to Equation 13. Each block applies pair-biased residue attention followed by a transition conditioned on s^{pair}:

$$\bar{s}_u^{\text{flow},\ell} = \tilde{s}_u^{\text{flow},\ell} + \text{Attn}_{\text{res}}(\tilde{s}^{\text{flow},\ell}; s^{\text{pair}}, \tilde{z}^\ell)_u, \tag{21}$$

$$s_u^{\text{flow},\ell+1} = \bar{s}_u^{\text{flow},\ell} + \text{Trans}_{\text{res}}(\bar{s}_u^{\text{flow},\ell}; s_u^{\text{pair}}). \tag{22}$$

Atom-level decoder and readouts. Each of the three decoder blocks first propagates the final residue representation to the corresponding atoms through cross-attention, followed by the same atom-level attention and transition updates used in Equations 17 and 18. The final atom representation Q^{dec} is used for coordinate and sequence prediction:

$$Q^{\ell+1} = \text{AtomBlock}\big(\text{Upcast}(Q^\ell, s^{\text{flow},16}); \tilde{c}^\ell, \tilde{p}^\ell\big)\,, \tag{23}$$

$$v_\theta = \sigma\, W_x\, \text{Norm}(Q^{\text{dec}}), \tag{24}$$

$$\hat{X}_1 = X_t + (1-t)v_\theta, \tag{25}$$

$$\ell_S = W_{\text{seq}}\, \text{Downcast}\big(\text{sg}(Q^{\text{dec}}), \text{sg}(s^{\text{flow},16}), \text{sg}(s^{\text{pair}})\big)\,. \tag{26}$$

where W_x and W_{seq} are the coordinate and sequence projection layers, respectively, Norm denotes RMS normalization, ℓ_S contains residue type logits, and sg denotes the stop-gradient operator. Stop-gradient prevents the sequence readout from backpropagating into the coordinate generation trunk, decoupling sequence prediction from the coordinate objective. The coordinate head predicts the denoised all-atom structure, while the sequence head predicts residue identities only at positions designated for sequence generation.

Coordinate supervision. The coordinate loss combines a weighted coordinate MSE with a smooth-lDDT term. The MSE uses a slot mask m_a that selects peptide slots with unfixed coordinates, including unused virtual slots, while excluding unresolved physical atoms in the native structure and padding. Fixed receptor atoms and peptide atoms with coordinates prescribed by the training task are excluded from this mask. For peptide slots, $x_{1,a}$ and $\hat{x}_{1,a}$ denote the corresponding atom coordinates of X_1 and $\hat{X}_1$. With atom-specific weights w_a and coordinate normalization scale σ, the per-example coordinate objective is:

$$\mathcal{L}_{\text{coord}} = \frac{\sum_a m_a w_a \,\|(\hat{x}_{1,a} - x_{1,a})/\sigma\|_2^2}{3\max(\sum_a m_a, 1)\big((1-t)^2 + 10^{-5}\big)} + 0.25\mathcal{L}_{\text{smooth-lDDT}}, \qquad \sigma = 6\,\text{Å}. \tag{27}$$

Unused peptide slots are anchored to the C_β coordinate of their parent residue, with C_α used for Gly or when C_β is unavailable, and remain supervised when their coordinates are unfixed. The atom weight w_a is 1.2 for slots belonging to polar residues and 1.0 otherwise. Virtual slots inherit the weight of their parent residue. The loss is averaged over the per-example objectives.

The smooth-lDDT term uses only resolved physical atoms, excluding virtual slots. It includes peptide-peptide and peptide-receptor pairs, while excluding pairs in which both atoms have fixed coordinates. Same-residue pairs, pairs with zero target distance, and pairs with target distances of at least 15 Å

are also excluded. Each unordered pair is counted once. For each eligible pair, the absolute error between the predicted and target distances is scored using sigmoid functions with thresholds of 0.5, 1, 2, and 4 Å. The loss is one minus the mean score over thresholds and eligible pairs, and is zero when no eligible pairs are present.

For cyclic peptides, an additional closure loss supervises the local geometry around the prescribed covalent junction. Specifically, we compare all off-diagonal pairwise distances among five physical atoms surrounding the junction with the corresponding distances in the clean target using a smooth L1 loss with $\beta = 0.1$ Å. For head-to-tail cyclization, the five atoms are C_α-C-O at one endpoint and N-C_α on the other. For disulfide cyclization, they are C_α-C_β-S_γ and S_γ-C_β, while for isopeptide cyclization they are Lys C_ϵ-N_ζ and Asp C_γ-$O_{\delta 1}$-C_β. Pairs involving unresolved atoms are excluded, and the remaining pairwise losses are averaged within each example. Linear samples contribute zero closure loss, so the effective contribution of this term to the batch objective scales with the fraction of cyclic samples. The closure loss supervises the local junction geometry rather than the newly formed covalent bond length alone.

Sequence supervision. Let S denote the native peptide sequence. For residue u, $q_u(y)$ is the softmax probability assigned to amino acid $y \in \mathcal{A}$, where $\mathcal{A}$ is the canonical amino acid alphabet. Because the sequence readout operates on stop-gradient decoder features, the sequence loss updates only the readout parameters and does not backpropagate into the coordinate trunk. The set $\mathcal{F} \subseteq \overline{\mathcal{I}}_{\mathcal{T}}$ contains valid residue positions whose identities are free to generate under the sampled training task. The per-example sequence loss is the mean cross-entropy over these positions:

$$\ell_{\mathrm{CE}} = -\frac{1}{\max(|\mathcal{F}|, 1)} \sum_{u \in \mathcal{F}} \log q_u(S_u). \tag{28}$$

Chemical consistency. Each candidate residue type y is evaluated using the same predicted clean coordinates $\hat{X}_1$. Under the fixed-slot mapping, $\mathcal{B}_y$, $\mathcal{H}_y$, and $\mathcal{V}_y$ denote the within-residue bonds, atom triples defining bond angles, and virtual slots associated with type y, respectively. For residue u, $d_{u,ab}$ denotes the Euclidean distance between slots a and b, and $\zeta_{u,abc}$ denotes the cosine of the angle centered at slot b, both computed from $\hat{X}_1$. The anchor slot a_y is C_α for Gly and C_β otherwise. Deviations from a prescribed range $[L, U]$ are penalized by:

$$\psi_\beta(z, L, U) = \mathrm{smoothL1}_\beta\left([L - z]_+ + [z - U]_+, 0\right), \tag{29}$$

where $[z]_+ = \max(z, 0)$ and β is the smooth L1 transition parameter. For a nonnegative input r:

$$\mathrm{smoothL1}_\beta(r, 0) = \begin{cases} r^2/(2\beta), & r < \beta, \\ r - \beta/2, & r \geq \beta. \end{cases} \tag{30}$$

The candidate geometry penalty is:

$$\begin{aligned} E_u(y) = {} & \frac{1}{|\mathcal{B}_y|} \sum_{(a,b) \in \mathcal{B}_y} \psi_{0.1}(d_{u,ab}, L_{y,ab}, U_{y,ab}) \\ & + \frac{1}{|\mathcal{H}_y|} \sum_{(a,b,c) \in \mathcal{H}_y} \psi_{0.05}(\zeta_{u,abc}, L_{\mathrm{ang}}, U_{\mathrm{ang}}) \\ & + \frac{1}{\max(|\mathcal{V}_y|, 1)} \sum_{a \in \mathcal{V}_y} \psi_{0.1}(d_{u,a\,a_y}, 0, 0.25). \end{aligned} \tag{31}$$

Distances are expressed in Å, and $E_u(y)$ serves as an auxiliary geometry penalty. The bond length bounds $(L_{y,ab}, U_{y,ab})$ are $(1.6, 2.2)$ for bonds involving sulfur and $(1.15, 1.75)$ otherwise. The cosine bounds $(L_{\mathrm{ang}}, U_{\mathrm{ang}}) = (-0.906308, 0.258819)$ correspond approximately to bond angles between 75° and 155°. For numerical stability, the product of the two bond vector norms used in the cosine calculation is lower-bounded by 10^{-6}. The three terms in Equation 31 are normalized separately, with the virtual slot term equal to zero when $\mathcal{V}_y$ is empty.

The chemically valid set $\mathcal{F}_{\mathrm{chem}} \subseteq \mathcal{F}$ excludes residues with any unresolved native physical atoms. The per-example chemical loss is:

$$\ell_{\mathrm{chem}} = \frac{1}{\max(|\mathcal{F}_{\mathrm{chem}}|, 1)} \sum_{u \in \mathcal{F}_{\mathrm{chem}}} \sum_{y \in \mathcal{A}} q_u(y)\ \mathrm{sg}([E_u(y) - E_u(S_u)]_+). \tag{32}$$

where $E_u(S_u)$ evaluates the native residue type using the same predicted coordinates and provides a position-specific reference geometry penalty. The loss penalizes probability assigned to candidate residue types whose geometry penalty exceeds this reference. The stop-gradient operator sg detaches the penalty difference, preventing gradients from the chemical loss from propagating into the predicted coordinates.

The terms $\mathcal{L}_{\mathrm{CE}}$ and $\mathcal{L}_{\mathrm{chem}}$ in Equation 7 are computed from ℓ_{CE} and ℓ_{chem}, respectively, using only examples with $t \geq 0.9$ and nonempty corresponding sets $\mathcal{F}$ and $\mathcal{F}_{\mathrm{chem}}$. The loss weights in Equation 7 are $(\lambda_x, \lambda_c, \lambda_s, \lambda_h) = (4, 0.1, 0.1, 0.05)$. The chemical loss warmup $\alpha(k)$ remains zero through step 20,000, increases linearly to 1 by step 30,000, and remains 1 thereafter.

The flow model is trained for 200,000 steps using the Adam optimizer. The learning rate is linearly warmed up to 5×10^{-4} over the first 1,000 steps and then decayed to 10^{-4} using a cosine schedule. Algorithm 1 summarizes the flow model training procedure.

Algorithm 1 Training the CovaTide flow model

input: Training set $\mathcal{D}$, flow model F_θ, updates K_{flow}
output: Flow parameters θ

Initialize θ
$\boldsymbol{\rho} \leftarrow (0.90, 0.05, 0.05)$ ▷ co-design, fixed-sequence, fixed-backbone
for $k = 1, \ldots, K_{\mathrm{flow}}$ **do**
  $(\mathcal{R}, X_1, S, \mathcal{T}) \sim \mathcal{D}, \quad \kappa \sim \mathsf{Cat}(\boldsymbol{\rho})$ ▷ one task per complex
  $(M_{\mathrm{fix}}, M_{\mathrm{seq}}, f_{\mathcal{T}}) \leftarrow \textsc{Condition}(\kappa, \mathcal{T})$ ▷ task masks and features
  $t \sim \mathcal{U}(10^{-4}, 1 - 10^{-4})$
  $X_0 \leftarrow \textsc{CoordinatePrior}(X_1, M_{\mathrm{fix}}, f_{\mathcal{T}})$
  $X_t \leftarrow M_{\mathrm{fix}} \odot X_1 + (1 - M_{\mathrm{fix}}) \odot [(1-t)X_0 + tX_1]$
  $(\hat{X}_1, \ell_S) \leftarrow F_\theta(X_t, t; \mathcal{R}, f_{\mathcal{T}})$
  $\mathcal{L}_X \leftarrow \lambda_x \mathcal{L}_{\mathrm{coord}}(\hat{X}_1, X_1) + \lambda_c \mathcal{L}_{\mathrm{close}}(\hat{X}_1, X_1, \mathcal{T})$
  $\mathcal{L}_S \leftarrow \mathbf{1}_{t \geq 0.9}(\lambda_s \mathcal{L}_{\mathrm{CE}}(\ell_S, S, M_{\mathrm{seq}}) + \alpha(k)\lambda_h \mathcal{L}_{\mathrm{chem}})$
  $\mathcal{L} \leftarrow \mathcal{L}_X + \mathcal{L}_S$
  $\theta \leftarrow \mathrm{Adam}(\theta, \mathrm{clip}(\nabla_\theta \mathcal{L}))$
end for
return θ

A.6 Confidence model and training objective

Architecture and outputs. The confidence model is trained independently with the generator frozen. Candidate coordinates and the generator's final atom representations are detached before being passed to the confidence model. For fixed-sequence training decoys, the native peptide sequence is retained and used to construct the residue type features and Atom14 mapping. A projected generator representation is added to the initial atom embedding, and a reduced CovaTide trunk processes the candidate structure at $t = 0$. This trunk consists of one initial Pairformer block, two atom encoder blocks, one residue-level Pairformer block, four residue transformer blocks, and two atom decoder blocks. It retains the deep topology adapter but does not include coordinate or sequence readouts.

The resulting atom representation h_a^{conf} is used by a 50-bin pLDDT head and a shared pair error head. For an atom pair (a, b), the pair representation is constructed as:

$$z_{ab}^{\mathrm{conf}} = \left[h_a^{\mathrm{conf}} + h_b^{\mathrm{conf}}, \left|h_a^{\mathrm{conf}} - h_b^{\mathrm{conf}}\right|, h_a^{\mathrm{conf}} \odot h_b^{\mathrm{conf}}, E_{\mathrm{geom}}(\|\bar{x}_a - \bar{x}_b\|_2), E_{\mathrm{pair}}(r_{ab})\right], \quad (33)$$

where E_{geom} embeds the candidate interatomic distance, r_{ab} identifies peptide-peptide or peptide-pocket pairs, and E_{pair} embeds the corresponding pair type. The pair error head predicts a categorical distribution over 33 error bins spanning 0-16 Å. PDE is computed as the mean expected error over all unordered pairs of physical peptide atoms, including pairs within the same residue, whereas iPDE is computed over all physical peptide-pocket atom pairs. The pLDDT head reports the expected per-atom confidence on a 0-100 scale. Section 3.3 combines the predicted PDE and iPDE scores for candidate ranking.

Absolute confidence targets. For candidate coordinates $\bar{X}$ and native coordinates $X^\star$, define:

$$d_{ab}(Y) = \|y_a - y_b\|_2.$$

For each supervised peptide atom a, the reference neighborhood $\mathcal{N}_a$ contains physical C_α atoms from the peptide and receptor pocket whose native distance to a is below 15 Å, excluding self-pairs. Following the distance-difference formulation of lDDT (Mariani et al., 2013), the pLDDT target for atom a with a nonempty neighborhood is:

$$q_a^\star = \frac{1}{4|\mathcal{N}_a|} \sum_{b \in \mathcal{N}_a} \sum_{\epsilon \in \{0.5,1,2,4\}} \mathbf{1}\left[|d_{ab}(\bar{X}) - d_{ab}(X^\star)| < \epsilon\right]. \tag{34}$$

The resulting targets are discretized into 50 bins. The PDE and iPDE heads use the absolute distance error target:

$$e_{ab}^\star = \left|d_{ab}(\bar{X}) - d_{ab}(X^\star)\right|, \tag{35}$$

discretized into the same 33 bins used for prediction. Cross-entropy losses are averaged within each example to obtain $\mathcal{L}_{\text{pLDDT}}$, $\mathcal{L}_{\text{PDE}}$, and $\mathcal{L}_{\text{iPDE}}$. During training, peptide and interface pairs are subsampled for efficiency, whereas inference evaluates all valid pairs.

Algorithm 2 Training the generation-aware confidence model

input: Training set $\mathcal{D}$, frozen F_θ, confidence model C_ϕ, updates K_{conf}
output: Confidence parameters ϕ

Initialize ϕ
for $k = 1, \ldots, K_{\text{conf}}$ **do**
  $(\mathcal{R}, X^\star, S, \mathcal{T}) \sim \mathcal{D}$, $f_{\mathcal{T}}^{\text{fix}} \leftarrow \textsc{FixedSequence}(S, \mathcal{T})$
  $n \sim \mathcal{U}\{5, 10, 20, 30, 40\}$ ▷ Euler steps shared by four decoys
  $(\mathcal{P}_{\text{pep}}, \mathcal{P}_{\text{int}}) \leftarrow \textsc{SamplePairs}(X^\star, \mathcal{R})$
  for $d = 1, \ldots, 4$ **do**
    $\sigma_{\text{ctr}}^{(d)} \sim \mathcal{U}\{0.5, 1, 1.5, 2, 2.5\}$ ▷ center perturbation
    $\sigma_0^{(d)} \sim \mathsf{Cat}\big(\{4, 5, 6, 7, 8\}; (0.1, 0.2, 0.4, 0.2, 0.1)\big)$ ▷ prior scale
    $\xi^{(d)} \leftarrow (n, \sigma_{\text{ctr}}^{(d)}, \sigma_0^{(d)})$
    $(\bar{X}^{(d)}, h_{\text{gen}}^{(d)}) \leftarrow \text{sg}\big(\textsc{EulerSample}(F_\theta, \mathcal{R}, f_{\mathcal{T}}^{\text{fix}}, \xi^{(d)})\big)$
    $(q^{\star(d)}, e_{\text{pep}}^{\star(d)}, e_{\text{int}}^{\star(d)}) \leftarrow \textsc{Targets}(\bar{X}^{(d)}, X^\star, \mathcal{P}_{\text{pep}}, \mathcal{P}_{\text{int}})$
    $(\hat{q}^{(d)}, \hat{e}_{\text{pep}}^{(d)}, \hat{e}_{\text{int}}^{(d)}) \leftarrow C_\phi(S, \bar{X}^{(d)}, h_{\text{gen}}^{(d)}; \mathcal{R}, \mathcal{T})$
  end for
  $\mathcal{L}_{\text{abs}} \leftarrow \mathcal{L}_{\text{pLDDT}} + \mathcal{L}_{\text{PDE}} + 1.5\mathcal{L}_{\text{iPDE}}$
  $\mathcal{L}_{\text{rel}} \leftarrow 0.05\mathcal{L}_{\text{pLDDT}}^{\text{rel}} + 5\mathcal{L}_{\text{PDE}}^{\text{rel}} + 7.5\mathcal{L}_{\text{iPDE}}^{\text{rel}}$
  $\mathcal{L}_{\text{conf}} \leftarrow \mathcal{L}_{\text{abs}} + \beta(k)\mathcal{L}_{\text{rel}}$
  $\phi \leftarrow \text{AdamW}(\phi, \text{clip}(\nabla_\phi \mathcal{L}_{\text{conf}}))$
end for
return ϕ

Within-target ranking supervision. Each native complex is used to generate four fixed-sequence structural decoys. Candidate-level pLDDT predictions and targets are obtained by averaging over supervised physical peptide atoms and are reported on a 0-100 scale. For each confidence quantity $m \in \{\text{pLDDT}, \text{PDE}, \text{iPDE}\}$, all pairs of decoys from the same native complex supervise the agreement between predicted and target pairwise score differences:

$$\mathcal{L}_m^{\text{rel}} = \frac{1}{|\mathcal{G}|} \sum_{(g,p,q) \in \mathcal{G}} H_{\Delta_m}\big((\hat{m}_{g,p} - \hat{m}_{g,q}) - (m_{g,p}^\star - m_{g,q}^\star)\big), \tag{36}$$

where $\mathcal{G}$ contains all triplets (g, p, q) that pair decoys p and q from native complex g. The quantities $\hat{m}_{g,p}$ and $m_{g,p}^\star$ denote the candidate-level prediction and target, respectively. H_{Δ_m} is the Huber loss with $\Delta_{\text{pLDDT}} = 5$ and $\Delta_{\text{PDE}} = \Delta_{\text{iPDE}} = 0.5$ Å. The complete confidence objective is:

$$\begin{aligned} \mathcal{L}_{\text{conf}} = \mathcal{L}_{\text{pLDDT}} + \mathcal{L}_{\text{PDE}} + 1.5\mathcal{L}_{\text{iPDE}} \\ + \beta(k)\left(0.05\mathcal{L}_{\text{pLDDT}}^{\text{rel}} + 5\mathcal{L}_{\text{PDE}}^{\text{rel}} + 7.5\mathcal{L}_{\text{iPDE}}^{\text{rel}}\right). \end{aligned} \tag{37}$$

The relative loss weight $\beta(k)$ is zero through step 1,000, increases linearly to 1 by step 2,000, and remains at 1 thereafter. The frozen generator produces four fixed-sequence decoys online for each native complex. The number of Euler steps is sampled once per batch, whereas center perturbations and coordinate prior scales are sampled independently for each decoy. The confidence model uses unrelaxed structures in both training and standard sampling, scoring the final coordinates together with the corresponding generator representation.

We train the confidence model for 6,000 steps using AdamW with a weight decay of 0.01. The learning rate is linearly increased from 2×10^{-4} to 5×10^{-4} over the first 200 steps and then decayed to 2×10^{-4} using a cosine schedule. Each native complex defines one ranking group, with all six pairwise comparisons among its four decoys contributing to the relative losses. Algorithm 2 summarizes the confidence model training procedure.

A.7 Evaluation metrics

Linear peptide evaluation. Linear peptide evaluation uses the evaluation metrics described in UniMoMo (Kong et al., 2025). To compute amino acid recovery (AAR), we first align the generated and reference peptide sequences using the Needleman-Wunsch (NW) algorithm with the BLOSUM62 substitution matrix. AAR is then computed as the proportion of matching amino acids between the aligned generated and reference sequences. C-RMSD is the root-mean-square deviation of peptide C_α coordinates in the receptor frame, retaining differences in both binding pose and internal conformation. L-RMSD is computed after optimal rigid-body alignment of the peptide itself and measures intrinsic conformational recovery.

$\mathrm{Clash}_{\mathrm{in}}$ and $\mathrm{Clash}_{\mathrm{out}}$ denote the fractions of peptide residues involved in at least one C_α clash within the peptide and with the receptor, respectively, where a clash is defined as a C_α-C_α distance below 3.6574 Å. For $\mathrm{Clash}_{\mathrm{in}}$, we exclude self-pairs, adjacent residues, and the prescribed closure pair, and count each clashing residue only once. $\mathrm{JSD}_{\mathrm{bb}}$ and $\mathrm{JSD}_{\mathrm{sc}}$ measure the discrepancy between generated and reference torsion distributions using 10-degree bins and the Jensen-Shannon distance, defined as the square root of the Jensen-Shannon divergence. Distances are computed separately for each amino acid type and torsion type and then averaged over the available backbone (ϕ, ψ) or side-chain (χ) distributions. Lower clash rates indicate fewer steric conflicts, while lower JSD values indicate closer agreement with the reference local geometry.

Sequence and structure diversity are computed for each target as the number of clusters divided by the number of generated samples, using single-linkage clustering with similarity thresholds of 40% sequence identity and 2 Å for direct receptor-frame C_α RMSD, respectively. The resulting ratios are averaged across targets, with higher values indicating lower redundancy among generated candidates. For linear peptide evaluation, we use the Rosetta ref2015 score function (Alford et al., 2017). Each peptide-receptor complex undergoes interface repacking followed by local relaxation, and we report the interface energy, where lower values indicate more favorable peptide-receptor interactions. IMP measures the fraction of target proteins for which at least one generated peptide achieves a lower Rosetta interface energy than the corresponding native peptide (Kong et al., 2025).

Cyclic peptide evaluation. Following CPSea, we apply OpenMM relaxation before computing Rosetta interface energies (Yang et al., 2025; Eastman et al., 2017), and additionally evaluate peptide-receptor interactions using AutoDock Vina (Trott & Olson, 2010). Lower values indicate more favorable interactions for both scoring functions. Because the two scores are on different scales, we evaluate them separately. The Cyc. criterion measures the fraction of generated peptides whose C_β distance between the closure residues falls within the p5-p95 interval computed from the training data for the corresponding cyclization type. This metric captures the proximity of the closure residues but does not ensure valid covalent bond geometry. Bond success measures the fraction of generated peptides whose closure bond length deviates from its ideal value by at most 0.2 Å. Rosetta and Vina success rates are defined as the fractions of generated peptides with $dG < 0$ and Vina score < -3, respectively. Peptide-receptor clashes and side-chain torsion JSD are computed as defined above.

Following CPSea, Ramachandran allowed and favored ratios evaluate backbone conformational plausibility using the allowed and favored Ramachandran regions implemented in PyRama. For each peptide, we compute the fraction of evaluable backbone (ϕ, ψ) pairs that fall within each region and then average these fractions across peptides. Higher values indicate a larger fraction of backbone

conformations within the corresponding region, with the favored region representing a more restrictive subset of the allowed region.

Quality and diversity trade-off. Diversity should be interpreted alongside structural quality and interaction favorability. For peptide binders, useful diversity must remain compatible with stable conformations and the geometry of the receptor interface. Exploring broader sequence or conformational space is valuable when it yields alternative plausible binding solutions. Related protein design methods illustrate how learned sequence regularities and structural compatibility can guide generation, as in ProtGPT2 (Ferruz et al., 2022) and ProteinMPNN (Dauparas et al., 2022), respectively.

These considerations also affect how a generative model explores the design space. In particular, favoring compatible local geometry and receptor interactions can concentrate sampling within a smaller set of plausible configurations, potentially reducing sequence or structure diversity. Such a reduction may accompany improved candidate quality, but lower diversity alone does not indicate better designs. For this reason, we evaluate diversity alongside recovery, geometric validity, and energy metrics to assess both the breadth of generated candidates and their suitability for the intended binding task.

A.8 Topology adapter ablation

We compare the full deep topology adapter with four variants to examine the effects of repeated conditioning and the contributions of residue-level and atom-level information. Table 4 summarizes the evaluation results for these variants.

The w/o adapter variant removes the topology adapter entirely, eliminating all explicit topology-conditioned updates to residue- and atom-level representations. The shallow adapter retains conditioning at both resolutions but applies it only once before the atom encoder. Specifically, it adds endpoint role information to both residue- and atom-level single representations, closure graph information to residue-pair representations, and closure bond type information to the corresponding atom-pair representations. Thus, topology information is introduced only once, rather than being repeatedly integrated throughout the model trunk.

The w/o atom level variant retains repeated residue-level conditioning through endpoint roles and closure graph connectivity, while removing topology-conditioned updates to atom-level single and pair representations. Conversely, the w/o residue level variant retains atom-level conditioning through endpoint roles and closure bond information, while removing the corresponding updates to residue single and pair representations. Together, these ablations isolate the contribution of each representation level while retaining conditioning at the other. The full deep adapter combines both levels and repeatedly integrates residue-level topology information into the Pairformer and flow transformer blocks and atom-level topology information into the atom encoder and decoder.

For each cyclization type, C_β range measures the fraction of generated structures whose endpoint C_β distance falls within the corresponding p5-p95 interval, while closure range measures the fraction whose newly formed covalent bond length falls within its corresponding p5-p95 interval. The former assesses the relative placement of the closure residues, whereas the latter assesses the local geometry of the bond-forming atoms. Higher values indicate better agreement with the reference geometric ranges.

A.9 Sequence composition and synthesizability

Amino acid composition. Figure 7 compares residue frequency distributions across generated and reference peptides. For linear peptides, CovaTide and PepGLAD both broadly follow the PepBench composition, whereas PepFlow shows stronger residue-specific biases, particularly toward Ala and Gly. Larger deviations from the reference composition are observed for cyclic peptides, reflecting both model-specific preferences and cyclization-dependent sequence constraints. CovaTide nevertheless maintains a broad amino acid composition without strong residue-level composition bias. PepGLAD shows a similarly broad composition, while PepFlow and DiffPepBuilder show stronger enrichment for specific residues. Overall, CovaTide maintains amino acid usage that is broadly consistent with the reference distributions across both linear and cyclic peptides.

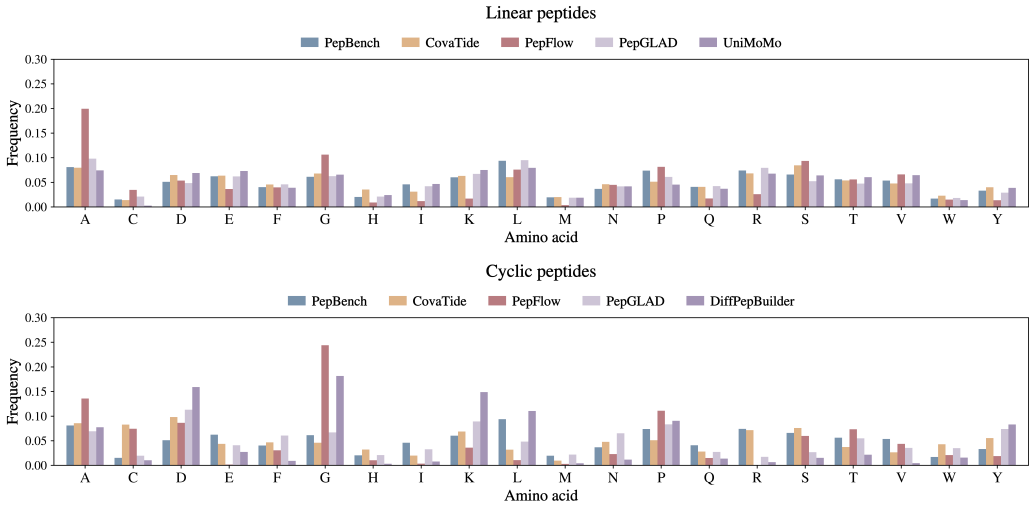

Figure 7: Amino acid frequency distributions of generated linear peptides (top) and cyclic peptides (bottom), with PepBench as the reference.

Sequence synthesizability. We assess sequence synthesizability using PepSySco (Peptide Synthesis Score), which predicts Fmoc peptide synthesis success from peptide length and the sequence-averaged Janin hydrophobicity index (Gutman et al., 2022). Scores range from 0 to 1, with higher values indicating greater predicted synthesizability. We apply the PepSySco web tool directly to generated linear and cyclic peptide sequences without length filtering and average scores within each target and then across targets. The tool supports sequences of 8-25 amino acids, whereas our evaluation also includes shorter peptides. Scores for these cases therefore involve extrapolation beyond the supported length range and should be interpreted cautiously. For cyclic peptides, PepSySco evaluates sequence-level synthesis propensity but does not explicitly model the cyclization reaction or its chemistry.

Table 6: Sequence synthesizability evaluated with PepSySco.

Method	Linear peptides ↑	Cyclic peptides ↑
CovaTide	**0.962**	<u>0.971</u>
PepGLAD	0.951	-
PepFlow	0.910	-
UniMoMo	<u>0.960</u>	-
$\text{PepGLAD}_{\text{cyc}}$	-	0.962
$\text{PepFlow}_{\text{cyc}}$	-	0.927
$\text{DiffPepBuilder}_{\text{cyc}}$	-	**0.977**

CovaTide achieves the highest PepSySco score among the evaluated methods for linear peptides and the second-highest score for cyclic peptides (Table 6), indicating favorable predicted synthesizability in both settings. $\text{DiffPepBuilder}_{\text{cyc}}$ achieves the highest score for cyclic peptides. This may partly reflect its shifted amino acid composition, including pronounced glycine enrichment and frequent glycine-rich stretches. Because PepSySco is based solely on sequence-derived features, these scores should be interpreted alongside amino acid composition and structural quality analyses rather than as a standalone measure of practical synthesis feasibility.

A.10 Molecular dynamics simulations

We perform explicit-solvent molecular dynamics (MD) simulations using GROMACS (Abraham et al., 2015). Proteins and peptides are described by the Amber ff14SB force field (Maier et al., 2015), with TIP3P water used as the solvent (Jorgensen et al., 1983). Protonation states are assigned

at pH 7.0 using PROPKA through PDB2PQR (Jurrus et al., 2018), and disulfide connectivity is explicitly specified in the topology. Each complex is solvated in a truncated octahedral TIP3P water box with a 10 Å buffer between the solute and the box boundary. Na^+ and Cl^- ions are added to neutralize the system and adjust the NaCl concentration to approximately 0.15 M.

Each system undergoes two stages of energy minimization, first with the solute held fixed and then with all atoms free. The system is subsequently heated from 5 to 310 K over 0.5 ns under NVT conditions, with positional restraints applied to protein and peptide heavy atoms. This is followed by 2.5 ns of NPT equilibration at 300 K and 1 bar, during which the restraints are progressively reduced. All positional restraints are then removed for a 100-ns production simulation at 300 K and 1 bar. Dynamics are propagated using the stochastic dynamics (Langevin) integrator, with pressure maintained using stochastic cell rescaling (C-rescale) (Bernetti & Bussi, 2020). Hydrogen mass repartitioning (HMR) (Hopkins et al., 2015) enables a 4 fs integration time step. Bonds involving hydrogen atoms in the solute are constrained using LINCS (Hess et al., 1997), while the TIP3P water geometry is maintained using SETTLE (Miyamoto & Kollman, 1992). Periodic boundary conditions are applied in all dimensions, and long-range electrostatic interactions are treated using particle-mesh Ewald (PME) (Darden et al., 1993). A 1.0 nm cutoff is used for the real-space electrostatic and Lennard-Jones interactions.

Trajectory coordinates are saved every 10 ps and subsequently corrected for periodic boundary conditions and centered before RMSD analysis. Each frame is aligned to the initial production structure using receptor C_α atoms, and the peptide heavy-atom RMSD is computed after applying the same rigid-body transformation. We report the mean and sample standard deviation of the unsmoothed RMSD values over the final 50 ns. For visualization, RMSD traces are smoothed using an approximately 1 ns moving average, whereas RMSD distributions are estimated from the unsmoothed values over the final 50 ns.

A.11 Case study

Molecular dynamics. Section 4.3 presents molecular dynamics results for two generated disulfide peptides. Figure 8 provides two additional examples for generated head-to-tail peptides targeting PDB entries 3VXW and 5ONP.

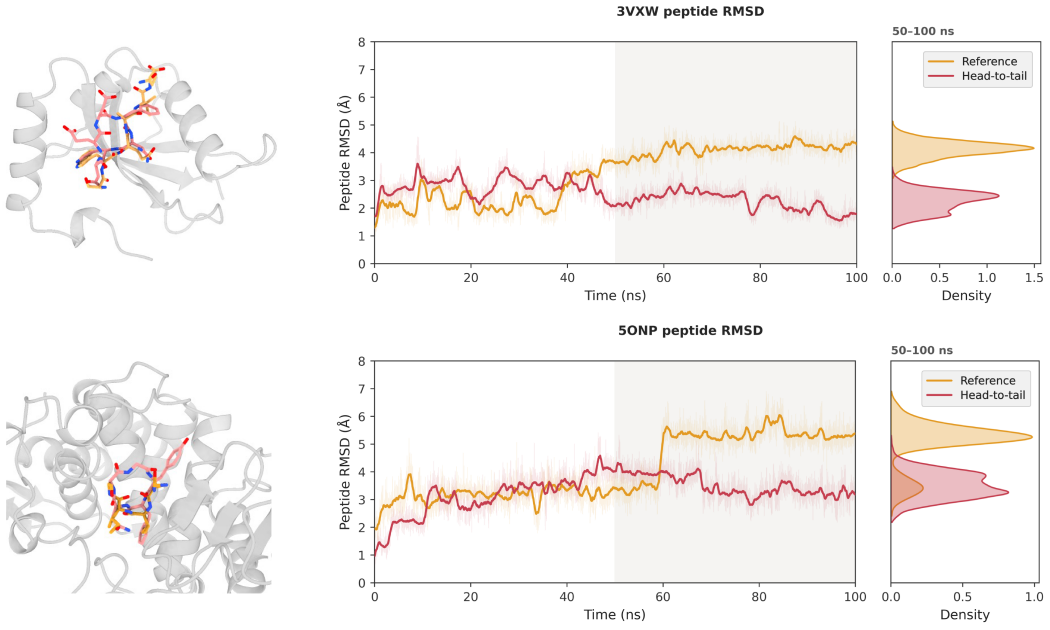

Figure 8: **Additional molecular dynamics examples for PDB entries 3VXW and 5ONP.** RMSD trajectories of peptide heavy atoms and distributions over the final 50 ns for the reference and generated head-to-tail peptides.

Linear and cyclic peptide design. Figure 9 presents representative linear and cyclic peptide designs and their Rosetta dG distributions for two targets: HTATSF1 UHM (PDB ID: 6N3E) and TEAD4 (PDB ID: 6SEO). For each target, we generate 100 linear peptides and 35 peptides for each cyclic topology (head-to-tail, disulfide, and isopeptide).

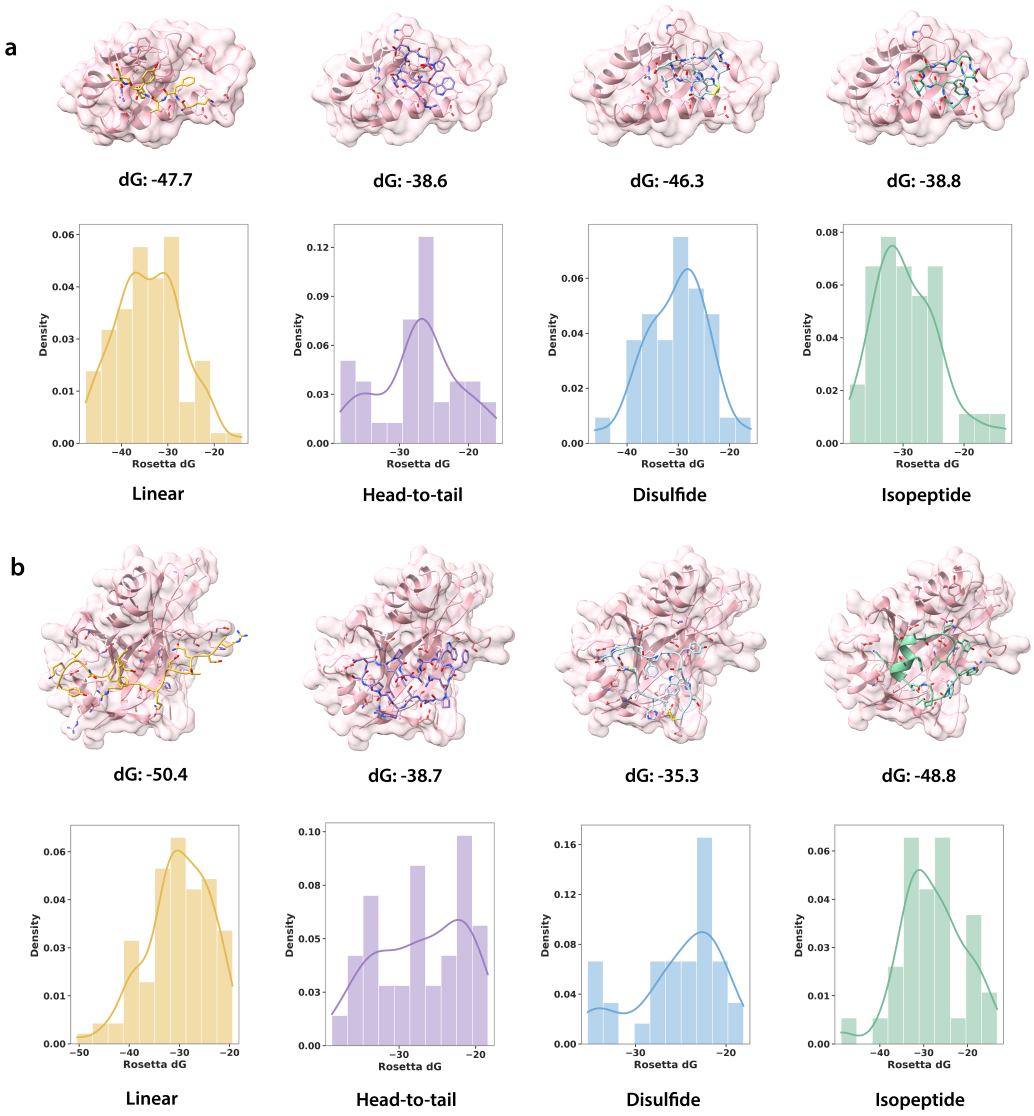

Figure 9: **Case studies of unified linear and cyclic peptide design.** Representative linear, head-to-tail, disulfide, and isopeptide designs and their Rosetta dG distributions for (a) PDB 6N3E and (b) PDB 6SEO.

A.12 Generalization to internal cyclization

Internal cyclization. The cyclic peptide data derived from CPSea place the prescribed closure bond between the first and last peptide residues. To evaluate generalization to nonterminal cyclization positions, we instead prescribe disulfide and isopeptide bonds between internal residues. We use 3 targets at each peptide length of 9, 10, and 11 residues, with closure residue pairs $(3, 7)$, $(3, 8)$, and $(2, 10)$, respectively. For each target, we generate 30 samples for each cyclization type, resulting in 90 samples per peptide length and cyclization type and 540 samples in total.

Evaluation protocol. Bond-length error is measured as the absolute deviation from reference bond lengths of 2.032 Å for disulfide cyclization and 1.343 Å for isopeptide cyclization. A generated structure is considered bond-successful when this error is at most 0.2 Å. Geometry is evaluated before OpenMM relaxation used for Rosetta interface energy scoring. Following the benchmark aggregation protocol, we select the lowest Rosetta dG among 30 candidates for each target and average these target-wise minima over the 3 targets within each peptide length and cyclization type. For the overall score, candidates from both cyclization types are pooled within each target, and the lowest dG is selected before averaging across all nine targets. We report median and mean bond-length errors over all generated samples in each group and compute success rates over the same samples. Rosetta success is defined as $dG < 0$. Joint success requires the same candidate to satisfy both the pre-relaxation bond-distance and Rosetta energy criteria.

Table 7: Evaluation of generalization to internal cyclization sites.

Length	Cyclization type	Bond error (median) ↓	Bond error (mean) ↓	Bond success ↑	Rosetta dG ↓	Energy success ↑	Joint success ↑
9	Disulfide	0.008	0.502	78.889%	-31.295	98.889%	78.889%
9	Isopeptide	0.008	0.766	63.333%	-29.677	96.667%	62.222%
10	Disulfide	0.011	0.499	80.000%	-24.197	96.667%	77.778%
10	Isopeptide	0.009	0.721	70.000%	-28.099	88.889%	64.444%
11	Disulfide	0.004	0.309	85.556%	-36.085	98.889%	84.444%
11	Isopeptide	0.002	0.169	88.889%	-34.542	93.333%	83.333%
	Overall	0.006	0.494	77.778%	-32.141	95.556%	75.185%

Generalization to internal cyclization sites. CovaTide generalizes the prescribed covalent topology from terminal closures in the CPSea-derived data to internal residue pairs (Table 7). Bond-length errors remain small across peptide lengths and cyclization types, with most generated structures satisfying the closure criterion. Disulfide cyclization is more robust for the shorter peptides, whereas isopeptide cyclization improves substantially with peptide length and performs particularly well at length 11. The gap between the low median and higher mean bond errors suggests that most samples form accurate closures, while a smaller fraction contributes larger geometric deviations.

Interface quality remains favorable after relaxation, with high Rosetta success across all tested settings. Joint success broadly follows bond success, suggesting that successful internal closure is often compatible with favorable interface energetics. Overall, these results show that topology conditioning extends beyond the terminal closure patterns present in the training data and supports internal disulfide and isopeptide cyclization across the tested peptide lengths.

Structural examples. Figure 10 shows representative peptides generated with prescribed internal disulfide and isopeptide linkages for the same receptor (PDB ID: 5NJX).

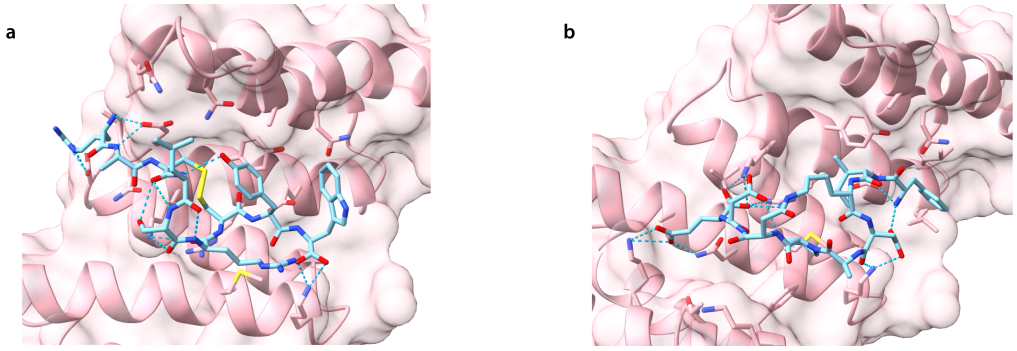

Figure 10: **Examples of internal cyclization.** Representative designs with prescribed internal (a) disulfide and (b) isopeptide linkages for PDB 5NJX.